Optimizing Neighbor Table Accuracy of Position-Based Routing Algorithms

Marc Heissenbüttel, Torsten Braun
Institute of Computer Science and Applied Mathematics
University of Bern, Switzerland
Email: {heissen, braun}@iam.unibe.ch

***Abstract*— In position-based routing protocols, each node periodically transmits a short hello message (called beacon) to announce its presence and position. Receiving nodes list all known neighbor nodes with their position in the neighbor table and remove entries after they have failed to receive a beacon for a certain time from the corresponding node. Especially in highly dynamic networks, the information stored in the neighbor table is often out-dated and does not reflect the actual topology of the network anymore such that retransmissions and rerouting are required which consume bandwidth and increase latency. Despite a considerable number of proposed position-based protocols, almost no analysis has been performed on the impact of beacons and the out-dated and inaccurate neighbor tables. We show by analytical and simulation results that performance suffers especially in highly mobile ad-hoc networks and propose several mechanisms to improve the accuracy of neighborhood information. Extensive simulations show the effectiveness of the proposed schemes to improve the network performance.**

***Index Terms*— Simulations**

I. Introduction

ROUTING in wireless mobile ad hoc networks (MANET) is a challenge due to the mobility which causes frequent changes in the network topology. Existing links disappear and new links are established when nodes move in and out of the transmission range. Routing protocols must be able to cope efficiently with this mobility. A lot of topology-based routing protocols have been proposed that either establish a route on-demand (e.g. AODV [1], DSR [2]) or proactively maintain hop-by-hop information at each node (e.g. OLSR [3], TBPRF [4]). In case of link incidents, new routes need to be discovered and updated routing information needs to be distributed by (partially) flooding the network, which may cause long latencies and interrupt communication between source and destination nodes. Furthermore, flooding of packets degrades the overall network performance and is costly in terms of consumed battery power.

The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

Position information available at each node is the key enabler for a new class of protocols, called position-based routing protocols, that exploits location information to enhance routing. Position-based routing protocols do no longer route packets based on node IDs. Forwarding decisions are solely based on absolute or relative position of the current node (e.g. provided by GPS), the positions of neighboring nodes (by nodes periodically transmitting a hello message, called beacon) and the destination (e.g. obtained via a location service [5]). Each packet is routed independently at each node and forwarded to a neighboring node which reduces the distance to the destination. These protocols are inherently more robust to changes in the network topology as they allow (almost) stateless routing and only require local rerouting in case of topology changes. Furthermore they naturally support geocasting. All these properties make them e.g. especially suited for sensor and vehicular ad-hoc networks. An overview of position-based routing algorithms and location services can be found e.g. in [6] and [7].

In position-based routing protocols, nodes periodically broadcast beacons to announce their presence and location to their neighbors. Each node stores all neighbors and their current positions in a neighbor table, i.e. all nodes within transmission range from whose it receives a beacon. If a node does not receive any beacon from one of its neighbors within a certain time interval, called neighbor time-out interval, the corresponding node is considered to have left the transmission range or is unreachable by any other reason and is deleted from the neighbor table. Routing of packets is done based on the positions of nodes in the neighbor table. One node is chosen as a next hop according to the applied routing strategy, e.g. the node closest to the destination. Even though changes in the network topology do not induce overhead by transmitting routing packets and only require local modification of the neighbor table, inaccurate or out-dated neighborhood information may severely affect position-based routing protocols.

In this paper, we analyze the impact of inaccurate and out-dated neighbor information on the performance of the network for position-based routing protocols. We propose several approaches to improve the reliability of neighbor tables by optimizing the time between the transmissions of beacons in order to mitigate the observed drawbacks such as packet loss and high latency. The remainder of this paper is organized as follows. Section II gives a survey on related work. In section III and IV, we briefly discuss the drawbacks of beaconing and analytically estimate the impact on routing, respectively. Several approaches to mitigate the drawbacks of beaconing are proposed in section V. The protocols used in the simulations are described and the simulation results are given in section VI and VII. Finally, section VIII concludes the paper.

II. Related Work

The related work can be broadly divided into three groups. One group includes approaches to deal with link incidents and ways to predict them in topology-based protocols. The second group of related work comprises various approaches that deal with updating strategies of a central database in wireless networks. Protocols that avoid beaconing completely fall into the third group.

Several approaches are described in the literature to mitigate the drawbacks of link incidents for topology-based protocols. AODV [1] implements a local route repair mechanism which aims to replace a particular broken link with an alternate path between the two nodes and to minimize the latency and induced routing overhead of link incidents. To avoid complete disruption of communication, [8] investigated the expected lifetime of routes in order to schedule the route discovery before actual link break. Unlike these protocols using hop-count as the routing metric, several other protocols were proposed, which take the stability of links and paths into account to minimize the number of link incidents in the first place. In ABR [9] the lifetime of a link is taken into account, whereas SSA [10] also considers feedback from the link layer about signal strength as primary routing metric. In [11] and [12] results from analytical derivations and observations made by simulations are used to design new routing metrics which favor more stable paths. Based on link availability estimations, a metric for path selection in terms of reliability and resilience is introduced in [13]. If nodes are equipped with GPS receivers or any other technology that provides absolute or relative positions of nodes, information about the velocity and direction are also often known and can be utilized to estimate the expiration time of a link and to reconfigure routes timely as proposed in [14]. Unlike [14], where GPS-information is only applied to maintain routes, [15] additionally makes use of location information in the routing decision itself to establish paths in a depth-first search way. In [16], factors that influence the utility of hello messages were studied for determining link connectivity in topology-based protocols. Out-dated and inaccurate neighbor tables in position-based routing protocols may be considered as the analogue of link incidents in topology-based protocols. Unlike for topology-based protocols, almost no research has been performed on link incidents and inaccurate neighbor tables in position-based routing protocols. To the best of our knowledge, the only exception is GPSR [17], where the authors compared the packet delivery ratio and routing overhead for different time intervals between beacons.

Even though beaconing was not explicitly studied, the determination of the ”best” time when to update information stored at other nodes was studied in various contexts. Many location management schemes were proposed for cellular networks in the literature (see e.g. [18] for an overview). Location management schemes deal with when to update the Home Location Register to keep track of a node's position if it has moved to a new cell. Dissemination and replication of data in repositories for mobile ad-hoc networks was studied e.g. in [19]. The authors propose different strategies when to trigger updates. These approaches differ from our investigation in many points, mainly that information is transmitted infrequently to some few central nodes over long paths whereas we consider the case of updating only neighboring nodes but rather frequently. Furthermore, entries do not need to be periodically refreshed to remain valid.

Lately, several protocols have been proposed which adopt a new paradigm for position-based routing [20], [21], [22], and [23]. The next hop is not determined at the sender, but in a distributed way at the receivers. Nodes do not rely on information about neighbors anymore and allow disposing beaconing completely. These beaconless routing protocols exploit the broadcast property of the wireless medium to determine in a completely distributed way the next node after the packet has been transmitted. Data packets are just broadcasted and all receiving nodes compete to forward the packets. The first node, which succeeds to transmit, suppresses the others. To minimize the probability of more than one node transmitting simultaneously, each node introduces a small additional ”random” delay. These protocols eliminate some of the drawbacks of beaconing as discussed in the next section, but also cause new problems. To assure mutual reception of relayed packets, only nodes within a certain area are potential forwarders. This number is only a fraction of all nodes closer to the destination such that greedy routing fails more often and a time and resource consuming recovery strategy has to be applied. Furthermore, packet duplications may happen frequently if nodes, which received a packet for forwarding, do not overhear the forwarding of other neighbors due to obstacles or interferences and thus transmit another copy of the same packet.

III. Drawbacks of Beaconing

The periodical broadcasting of hello messages has several drawbacks such as unnecessary utilization of network resources, interferences with regular data packet, and consumption of scarce battery power. As beaconing is a proactive component of position-based routing, it is performed independently of actual data traffic. Even in cases where no data is transmitted, nodes constantly exchange beacons. In this paper, we do not further consider these direct consequences of beaconing, but focus on the impact of beaconing on routing as discussed in the next paragraph. This impact is perhaps less obvious but has even more severe drawbacks such as an increased delay and wasted bandwidth and battery power.

Each node routes the packets based on the perceived topology, which may not correspond to the actual topology, because nodes have moved since their last beacon transmission. This movement of nodes may result basically in three scenarios. First, nodes are listed in the neighbor table with an inaccurate position, but are still reachable. This is basically always the case expect for a static network. Secondly, a node moved into the transmission range of another node, but it is not visible since no beacon was received yet. In these both scenarios, the routing protocol may take suboptimal decisions and not route packets via the best located neighbor. More serious is the third scenario, where nodes are wrongly listed in the neighbor table even though they moved out of transmission range. If such an unreachable node is chosen by the routing protocol, the MAC layer is not able to deliver the packet. After several retransmission attempts, the MAC layer either just drops the packet or notifies the routing protocol of the failed transmission and passes the packet back. The routing protocol in turn selects a different next hop and hands the packet again down to the MAC layer. This process is repeated until the packet can be delivered eventually to the next hop. This rerouting has basically three consequences. The delay is increased, the effective available bandwidth is reduced, and energy is consumed for the retransmission.

We estimate the induced delay if a routing protocol selects and unreachable next hop in case IEEE 802.11b is used on the MAC layer. Packets are retransmitted up to seven times before the MAC layer assumes the next hop to be unreachable. For each failed transmission the contention window is doubled, starting at a size of 31 up to a maximum of 1023 times the slot time of $20\,\mu s$. A node uniformly chooses from this contention window a backoff time for the next transmission. If all seven retransmission fail because the next hop is out of transmission range, the expected delay is $\frac{31+63+...+1023+1023}{2} \cdot 20\,\mu s \approx 30\,ms$. Furthermore, the effective available bandwidth for data is consequently only a seventh of the total bandwidth. If we talk about neighbor table accuracy in the following, we refer primarily to the correct set of neighbors and less to their actual positions.

Three factors contribute further to inaccurate neighbor tables. First, beacons are broadcasted and most MAC layer protocols do neither require nor provide acknowledgments for broadcast transmissions. The delivery is not guaranteed even for nodes within the transmission range, since beacons may for example interfere with other beacons or data packets. Secondly, to avoid that nodes are constantly inserted and removed from the neighbor table if one beacon is not received, the time-out interval is often set to a multiple of the beacon interval. This longer interval increases the probability that a neighbor has meanwhile left the transmission range. And thirdly, most position-based routing algorithms apply a forwarding strategy which forwards packets to nodes close to the destination, e.g. [24], [17]. As a consequence, the selected neighbor is close to the border of the transmission range and thus has an even higher probability to have left the transmission range. All these factors contribute to out-dated neighbor tables, especially in scenarios with high mobility and/or small transmission ranges, causing the routing protocol to select unreachable nodes for several times before the packet is eventually delivered to the next hop.

We only mentioned mobility as a possible source of inaccuracy of the neighbor tables. But basically any kind of topology changes have the same effect either caused by nodes that toggle into and out of sleep states, obstacles moving between nodes, etc. Especially in some application like sensor networks were energy conservation is a major issue, putting nodes into sleep mode is often the only way to considerably save energy. Thus, we consider speed as a proxy for any kind of topology changes in this paper.

IV. Analytical Estimation

After having discussed reasons for out-dated and inaccurate neighbor tables and possible implications, we analytically estimate the likelihood of the occurrence of these events. We assume an isotropic transmission radius of r and an unbounded simulation area where nodes are distributed according to a Poisson point process of constant spatial intensity and move according to the random waypoint model with zero pause time, i.e. nodes choose randomly some destination and move there with a constant speed chosen uniformly in the interval $[v_{min}, v_{max}]$. The reason for an unbounded area is to simplify our analysis by having a uniform moving direction in the interval $[0, 2\pi]$ of the nodes, a uniform distribution of the nodes, and travel distances independent of nodes' locations, which are all not the case in the standard random waypoint model. Furthermore, we assume that nodes do not change their direction or speed within the time-out interval under consideration, which is only few seconds long. Note that these assumptions only help to simplify the analysis but do not fundamentally change the results. We consider two nodes A and B within transmission range and calculate the probability that they leave each others' transmission range within a certain time interval t. The crucial point in the derivation is to notice that instead of having both nodes moving, we assume node B being static and just node A moving with their relative speed vector. This assumption is valid as nodes move independently of each other and have symmetric transmission ranges. The probability density function f_S of the nodal speed s is derived e.g. already in [25] and is given by

$$f_S(s) = \frac{1}{s \ln\left(\frac{v_{max}}{v_{min}}\right)}$$

We first derive the expected value of the relative speed vector, i.e. the difference of two arbitrary speed vectors, in the unbounded random waypoint model. Let the speed vectors $\vec{a}, \vec{b}$ of node A and B be given in polar coordinates as (s_a, α) and (s_b, β) with $s_a, s_b \in [v_{min}, v_{max}]$ and $\alpha, \beta \in [0, 2\pi]$. The norm of $\vec{a} - \vec{b}$ is the velocity of the relative speed vector.

$$|\vec{a} - \vec{b}| = \sqrt{s_a^2 + s_b^2 - 2 s_a s_b \cos(\alpha - \beta)}$$

It is well-known that for a random vector $\mathbf{X} = (X_1, \ldots, X_n)$ with the joint density function $f_{\mathbf{X}}$ and a function $\varphi : \mathbf{R}^n \to \mathbf{R}$, the expected value is given by

$$E_\varphi(\mathbf{X}) = \int_{-\infty}^{\infty} .. \int_{-\infty}^{\infty} \varphi(x_1, .., x_n) f_{\mathbf{X}}(x_1, .., x_n)\, dx_1..dx_n$$

If the random variables are independent, the joint density function $f_{\mathbf{X}}$ is the product of the density functions $f_{\mathbf{X_i}}$. Thus, we obtain the expected value of the speed difference $E_S(v_{min}, v_{max})$ between $\vec{a}, \vec{b}$ as

$$E_S(v_{min}, v_{max}) = \int_{v_{min}}^{v_{max}} \int_{v_{min}}^{v_{max}} \int_0^{2\pi} \int_0^{2\pi} \sqrt{{s_a}^2 + {s_b}^2 - 2 s_a s_b cos(\alpha - \beta)} \cdot f_S(s_a)\, f_S(s_b)\, f_A(\alpha)\, f_B(\beta)\, d_\alpha\, d_\beta\, ds_a\, ds_b$$

By numerical integration, we obtain for E_S the values given in Table I for different v_{min}, v_{max}. For example,

TABLE I

Expected speed difference of two nodes

$v_{min}\,[m/s]$	$v_{max}\,[m/s]$	$E_S\,[m/s]$
1	10	5.69
1	20	9.64
1	40	16.68
10	20	18.83
10	40	29.55

we can see that that for a speed interval $[1, 20]\,m/s$, two arbitrary nodes move relative to each other with almost $10\,m/s$.

Suppose now node A moves a distance d to A'. The size of the area $A(r, d)$, which was initially covered by node A's transmission range and is now not longer within transmission range after it has moved to A', is depicted in Fig. 1. It is easy to see that the expected direction of movement of $\vec{a} - \vec{b}$ is again uniformly distributed in $[0, 2\pi]$. As node B is static and uniformly distributed, the probability that B is out of transmission range after node A has moved to A' is equal to the ratio of $A(r, d)$ to the size of the whole transmission range $r^2\pi$.

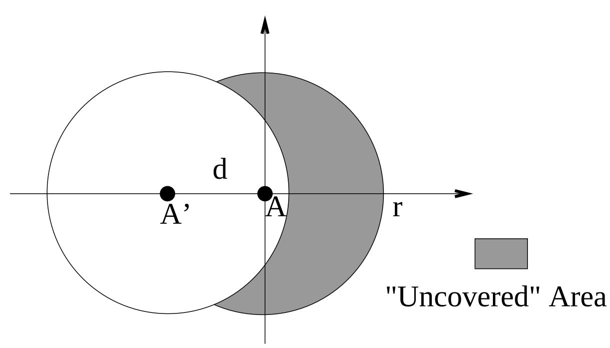

Fig. 1. Area uncovered if node moves from A to A'

The size of A(r,d) is given by

$$2 \cdot \left(\int_{-\frac{d}{2}}^{r} \sqrt{r^2 - x^2}\, dx - \int_{-\frac{d}{2}}^{-d+r} \sqrt{r^2 - (x+d)^2}\, dx \right)$$
$$= \frac{d}{2}\sqrt{4r^2 - d^2} + 2r^2 \arcsin\left(\frac{d}{2r}\right)$$

Given a certain speed interval $[v_{min}, v_{max}]$, a fixed transmission range r, and a time-out interval t, after which an entry is deleted in the neighbor table, the probability that node B is no longer within transmission range of node A after t can now be calculated as follows. The speed interval immediately yields the expected relative speed $E_S(v_{min}, v_{max})$. We obtain the distance d by multiplying E_S with t, i.e. the distance d is the expected distance which a node with speed E_S moves relative to any arbitrary other node within t. The distance d and the transmission range r are used to obtain the area $A(r, d)$ which is uncovered within the neighbor time-out interval t. Thus the probability p that an entry in the neighbor table is out-dated is given by $A(r, d)$ divided by $r^2\pi$. Or in other words, p percent of all nodes stored in the neighbor table are out-dated and the nodes are not within transmission range anymore. In Table II the respective values are given for transmission radii of $r = 250\,m$ and $r = 100\,m$. The rows and columns indicate time-out intervals t in seconds and speed intervals $[v_{min}, v_{max}]$ in meters per second, respectively.

TABLE II

EXPECTED PERCENTAGE OF OUT-DATED NEIGHBORS FOR $r = 250\,m / r = 100\,m$

	$[1/10]\,m/s$	$[1,20]\,m/s$	$[1,40]\,m/s$
1	1.45/3.62	2.45/6.13	4.25/10.61
3	4.35/10.85	7.36/18.35	12.72/31.52
5	7.24/18.05	12.25/30.39	21.14/51.51
10	14.46/35.73	24.40/58.90	41.67/92.09

Even for a large $r = 250\,m$ and for slow $v_{max} = 10\,m/s$, the percentage of out-dated entries is in the order of 10% for time-out intervals of $5\,s$ or more. For very high-speed scenarios with $v_{max} = 40\,m/s$ and long time-out intervals of $10\,s$, we may expect more than 40% of the nodes listed in the neighbor table to be actually unreachable. We can observe that the numbers out-dated neighbors are almost inverse proportional to the transmission radius. A 2.5 times smaller transmission radius thus causes about a 2.5 times higher inaccuracy. The same proportionality applies also for the relative speed E_S and the number of out-dated neighbors.

The probability of Table II may not be directly mapped to the probability that the next hop selected by the routing protocol is unreachable because most of the protocols select neighbors closest to the destination as a next hop. The probability to choose an unreachable neighbor may be even higher since most of the uncovered area is at the border of the transmission range.

V. IMPROVING NEIGHBOR TABLES

We propose three approaches to minimize the risk to choose an unreachable next hop from the neighbor table. For reason of simplicity, each approach is considered separately, even though it is possible to use them in combination.

A. Beaconing Strategies

This first proposal directly tries to improve the accuracy of the nodes' positions in the neighbor table to better match the actual positions by adapting the interval between beacons and as well the neighbor time-out interval.

1) Time: For sake of completeness, we mention the time-based strategy which is applied basically by all other position-based routing protocols proposed in the literature. Beacons are transmitted periodically by each node independent of the movement and the speed of the nodes which makes this approach susceptible to very suboptimal chosen beacon intervals. On one hand side, intervals may be too short and induce unnecessary transmissions if nodes are almost immobile. On the other hand, the interval may be too long for highly dynamic networks with fast moving nodes causing nodes to forward packet frequently to neighbors out of range, e.g. two nodes on the highway heading in different directions may only be within transmission range for a few seconds.

2) Distance: Each node transmits a beacon if it has traveled a certain distance from the position where a beacon was broadcasted for the last time. When a node has moved a certain distance, it is likely that the neighbors have changed and are not yet aware of this new node. To prevent from the possibility that a rather immobile node remains undetected, a time threshold is set after which a beacon is transmitted in any case independent of the distance traveled so far. Problems may arise in the case where nodes move as a group. Only few changes may happen to the network topology as nodes only marginally change their locations relative to other nodes and hence would not need to broadcast beacons frequently.

3) Speed: In the speed-based approach the frequency of broadcasting beacons is set according to the speed of nodes. Fast moving nodes transmit beacons more frequently than slow moving nodes. Drawbacks may occur in the situation if nodes move fast within a bounded area and thus do not really change their locations. As with the distance-based strategy, beaconing may be suboptimal in case of group mobility. We also have to set again an upper bound for the time interval in order to assure that stationary nodes also occasionally transmit beacons.

Problems may arise in both approaches, the speed- and distance-based, if a fast moving node crosses an area with rather static neighbors which transmit beacons very infrequently. The neighbor table of the fast moving nodes may be very inaccurate and hence it may be difficult to find a reachable neighbor.

4) Neighbor Time-Out Interval: The neighbor time-out interval is of particular importance as it directly maps to the accuracy of the neighbor table. It is normally set to a multiple of the beacon interval to avoid a too high fluctuation of the nodes in the neighbor table. We also ran simulations with smaller time-out intervals such that basically a node is deleted from the neighbor table after having not received one beacon. In the speed-based beaconing strategy, we set the time-out depending on the nodes' speeds.

B. Estimation of Link Availability

In the second approach, the velocity and direction of nodes are used to predict the time when two nodes are not within transmission range anymore. Unlike for conventional position-based routing algorithms, we do not only transmit information about a node's position in beacons, but also its speed vector, i.e. its velocity and direction. Thus, a node lists in its neighbor table all neighbors with their respective positions and speed vectors. Each entry is labeled with the time when the beacon was received. When a node at position A has to transmit a data packet, it calculates the distance to a neighbor which was located at position B and moving with speed vector $\vec{b}\,t$ seconds ago. This node is predicted to be at position $B' = B + \vec{b}t$. Assuming a circular transmission range r, the neighbor is no longer reachable if the distance $AB' > r$, i.e.

$$|B + \vec{b}t - A| > r$$

In this case, the neighbor is not selected as a next hop. This estimation helps to reduce the probability of selecting an unreachable neighbor. The underlying assumption is that nodes keep approximately the same velocity and direction for some seconds which seems to be reasonable for realistic node movement patterns.

C. Rx Threshold

A possible way to reduce the number or unreachable nodes in the neighbor table is to restrain the too greedy selection of next hops. In reality transmission ranges may be strongly irregular due to obstacles and interferences. If information from the physical layer is available such as the power of the received signal, this information may be used to cope with non-isotropic transmission ranges. We introduce a safety margin and do not consider nodes for routing from which a beacon was received with a power just above the minimal required receiving power. Only nodes from which beacons are received sufficiently strong are considered as neighbors. Unlike for beacons, data packets received at any power level are processed.

VI. PROTOCOLS

As a representative of a position-based routing protocol, we use GPSR [17]. GPSR also serves as basis for all proposed enhancements, i.e. only the enhancement under consideration is modified in the original GPSR. We consider also a reactive GPSR were nodes only transmit beacons on demand. Finally, we also discuss two theoretical versions of GPSR which retrieve perfect neighborhood information from the global data of the simulator. The aim is to determine the performance limits of any position-based routing protocol.

A. GPSR

As the underlying routing protocol, we use GPSR [17] which is basically an extension of GFG [24] with MAC-layer enhancements. A packet is routed in a greedy manner towards the position of the destination. Each node selects the node from its neighbor table which is geographically closest to the packet's destination. This process is repeated until the packet reaches the destination. If a node does not have any neighbor closer to the destination, it enters perimeter mode and the packet is routed according to the right-hand rule on the faces of a locally extracted planar subgraph to avoid loops and to recover from this local minimum (see [17] for more details). As soon as the packet arrives at a node closer to the destination than where it entered perimeter mode, the packet switches back to greedy routing. It was shown that this protocol guarantees delivery for static and connected networks. In Fig. 2, the packet is first routed in greedy mode to node X, which has no closer neighbor within transmission range to destination D, and enters perimeter mode. At node Z, the packet is again routed in greedy mode.

In accordance with the parameter values chosen in [17], beacons are sent in $1.5\,s$ intervals and the time-out is set to $6.75\,s$ after which a node is removed from the neighbor table. Furthermore, the following changes are implemented in the MAC-layer to optimize and make IEEE 802.11 more robust in mobile scenarios. Two operations are triggered, if a packet can not be delivered on the MAC-layer within the maximum number of retransmission to the next hop (normally seven retransmission with 802.11), which is an indication that

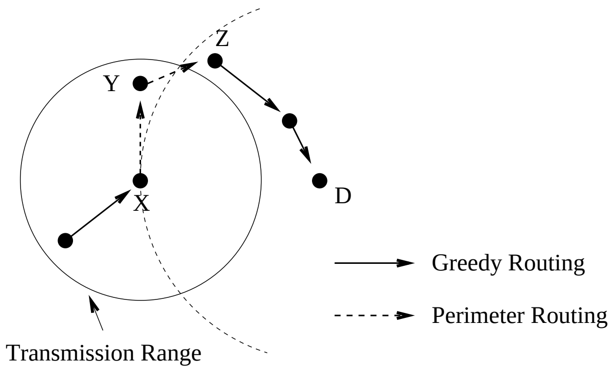

Fig. 2. Greedy and Perimeter Routing

the intended recipient has left transmission range. The unreachable node is removed from the neighbor table even before the actual neighbor time-out interval and all further packets with the same next hop already in the queue for the MAC-interface are removed and passed back to the routing protocol. Finally, nodes operate in promiscuous listening mode to receive all packets which allows to piggy-back beacons on data packets.

B. Reactive GPSR

As already proposed in [17], we make the beacon mechanism of GPSR fully reactive. Only when a node has to transmit data packets, it solicits beacons by transmitting a ”beacon request packet”. Each node overhearing this request transmits a beacon after introducing a short random delay to avoid that all nodes respond simultaneously and interfere at the receiver. Only then the node forwards the data packet to the ”best” neighbor from which it received a beacon. We implemented this reactive version to check the performance if a node operates on most accurate neighbor information. On one hand side the number of out-dated neighbor is minimized, on the other hand side the latency is increased.

C. Theoretical Optimum of Position-based Routing

We derive two slightly modified versions, called GPSR_bnu (beacons not used) and GPSR_bl (beaconless), which use perfect neighborhood information provided by global data of the simulator, i.e. impossible to realize in reality. To our knowledge, no such ”best-case” bound analysis for position-based routing has been done in any previous work. These protocols are introduced to determine the theoretical limits of any position-based routing protocol which only uses local information. At the time a node has to send a packet, the simulator provides the actual node closest to the destination which is within transmission range. This information is retrieved from the global data of the simulator where the current positions of all nodes in the network are stored. Thus, GPSR_bnu transmits beacons but does not use the information provided through beacons and maintained in the neighbor table. As neighborhood information is always correct and all selected hops are within transmission range, GPSR_bnu allows quantifying the influence of inaccurate and out-dated neighbor tables. As stated before, the periodical broadcasting of beacons has a non-negligible impact on the performance of the network, e.g. through collisions with data packets. We further introduce GPSR_bl to identify the impact of the transmissions of beacons. Additional to GPSR_bnu, nodes do not transmit beacons at all in GPSR_bl. Perfect positions of neighbors are again retrieved from global data of the simulator. The performance difference between GPSR_bnu and GPSR_bl is an indicator for the performance loss solely due to the additional traffic caused by beacons.

VII. SIMULATIONS

A. Simulation Setup

Implementation and simulation were conducted using the Qualnet network simulator. Radio propagation is modeled with the isotropic two-ray ground reflection model. The transmission power and receiver sensitivity are set corresponding to a nominal transmission range of 250m. We use IEEE 802.11b with RTS/CTS operating at a rate of 2 Mbps on the physical and MAC-layer. Nodes are placed in a rectangular area of 600m x 3000m. The nodes move according to the random waypoint mobility model. We implemented the stationary distribution of the random waypoint model as described in [25]. Thereby the simulations do not need an initial warm-up phase, whose duration is difficult to predict, to reach a stable state. The simulations last for 900s and data transmission starts at 180s and ends 880s such that emitted packets arrive at the destination before the end of the simulation. The beacons are randomly jittered by 50% of the beacon interval, in order to avoid possible synchronization of beacons between neighboring nodes [26]. We have one CBR traffic flow with 64 Byte packets at a rate of 2 packets per second between a randomly selected source and destination. We choose this low traffic scenario to prevent congestion and interference in order to isolate the effects of the routing protocols. The results are averaged over six simulation runs. Unless given otherwise, the pause time is $0\,s$ and the speed interval ranges from 1 to $40\,m/s$, the number of nodes is 400, and the beacon interval and the time-out interval is set to $1.5\,s$ and $6.75\,s$ respectively. The minimum speed is set unequal

$0\,m/s$ as otherwise the stationary distribution would be static [27]. The following subsection VII-B provides further explanation for the chosen values. The possible combinations and variations for all the parameters of the beaconing strategies, the prediction, and Rx threshold approach, yield a large set of possible simulations. To limit the number of simulations, each approach was tested independently of the others because it was not our intention to identify exactly optimal values for the parameters, but to find indications for the usability of the proposed approaches. We ran the simulations for many more parameter values than given in the next subsections. Only the most significant results are presented.

B. Identification of an Appropriate Scenario

We first conducted several simulations with the standard GPSR protocol, i.e. with a beacon interval and neighbor time-out interval of $1.5\,s$ and $6.75\,s$ respectively, to identify scenarios with poor performance such that the impact of the proposed enhancements can be observed more easily. In Fig. 3 and Fig. 4, the delivery ratio and the end-to-end delay are depicted for a speed interval of $[1, 40]\,m/s$. There are three reasons why we decided to use this high speed interval. First, even though the speed interval may seem high, the average speed of the nodes is only about $10\,m/s$. Second, we wanted to have a highly challenging scenario for the routing protocol to observe more definitely the differences in the results. And finally we consider speed as a proxy for any kind of topology changes as mentioned in section III. We ran the simulation also with a speed interval of $[1, 20]\,m/s$. The results are similar, except that the delay and the packet loss rate are about 30% and 50% lower respectively.

As expected the performance suffers in case of low pause times for all different kind of node densities. The optimum is for 200 nodes what is approximately 111 nodes per square kilometer and the value used in [17]. In case of 100 nodes the density is too low and GPSR is not able to achieve a high packet delivery ratio because packets are often routed in perimeter mode. More surprisingly is the fact that a higher node density also causes the performance to suffer. The reason is that the selected next hop is generally farther away and thus has a higher probability to have left the transmission range as explained in section III. Fig. 5 shows the number of RTS Retransmission caused by a CTS time-out and the number of packets handed from the MAC layer back to the routing protocol for the scenario with 400 nodes and a speed interval of $[1, 40]\,m/s$. These numbers are directly related to the number of out-dated entries in the routing table and are proportional to the delay in Fig. 4. Therefore, we choose to run all following comparative simulations for the enhancements to GPSR with 400 nodes and a pause time of $0\,s$, unless indicated otherwise.

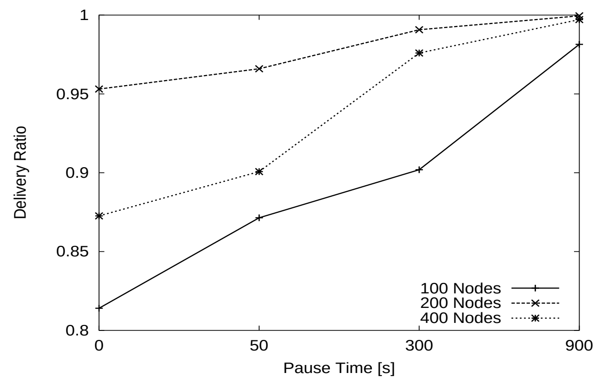

Fig. 3. Delivery Ratio of GPSR

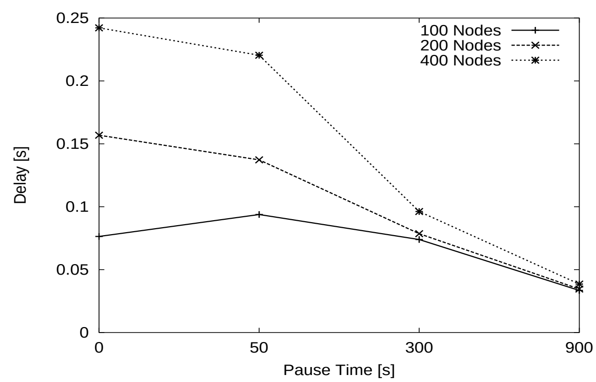

Fig. 4. Delay of GPSR

C. Time-based Beaconing Strategy

We tested several beacon intervals from $0.5\,s$ to $5\,s$ with a neighbor time-out interval which was set to 2 and 4.5 times the beacon interval. The values in Fig. 6 indicate that a neighbor time-out interval of 4.5 times the beacon interval as proposed in [17] is too long. It always performs much worse than an only twice as long time-out interval. The performance increases constantly for shorter beacon intervals such that if battery power and bandwidth are not an issue, the beacon interval and the neighbor time-out interval should be chosen very short.

D. Distance-based Beaconing Strategy

In these simulations, a node transmits a beacon after it has moved 5, 10, 20, and $50\,m$. Due to the results

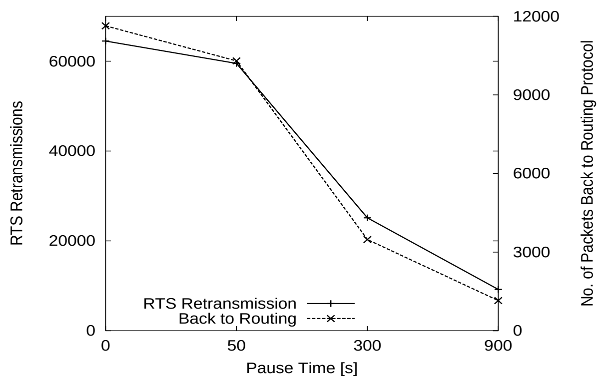

Fig. 5. RTS Retransmission / Back to Routing Protocol

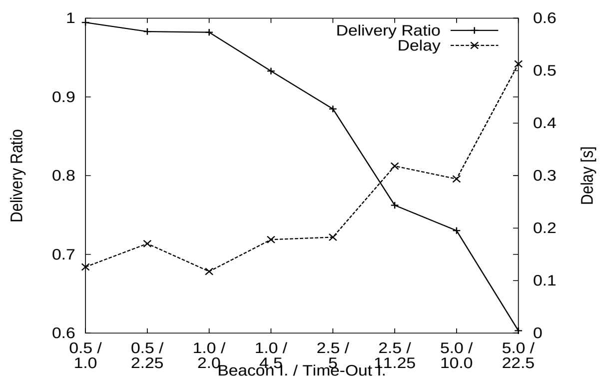

Fig. 6. Time-based Beaconing Strategy

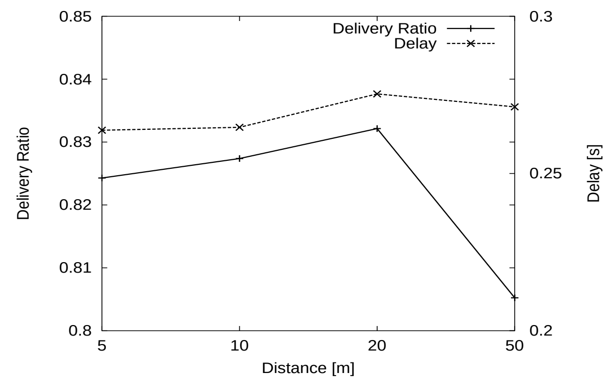

Fig. 7. Distance-based Beaconing Strategy

TABLE III

BEACON INTERVAL AT DIFFERENT SPEEDS

Speed $[m/s]$	Beacon Interval $[s]$
1-5	2.5
5-10	1.5
10-20	1
20-40	0.5

in Fig. 6, which showed that a long beacon and time-out interval decreases performance drastically, we set a maximal time of $5\,s$ after which a beacon transmitted even if the node has not yet moved the required distance. Accordingly also a maximal time-out interval is set to $10\,s$. The results in Fig. 7 show that the delivery ratio and the delay remain almost constant for different distance values with only a slight decrease for distances of $50\,m$. Compared to the pure time-based approach with beacon and time-out interval of $5\,s$ and $10\,s$ respectively, the packet delivery ratio is increased by 10% whereas the delay is only slightly reduced.

E. Speed-based Beaconing Strategy

We used the mapping in Table III to assign beacon intervals to node speeds. We tested four different configurations with two different policies to determine the neighbor time-out interval. In configuration 1, the time-out interval is set to twice the beacon interval of the sending node. This value is transmitted with the beacons such that each receiving node marks the corresponding entry in the neighbor table with a time when to delete it. The second configuration uses the minimum of the time-out interval of the node which sent the beacon and the node which received the beacons. The rationale behind this can be most easily understood with an example. Suppose a fast moving node passing by a static node and both received a beacon from the other node. The fast node would have a long time-out interval but the static node is out of transmission range very quickly and has to be removed quickly, i.e. within the time-out interval of the fast node. The same also applies for the static node which also has to delete the fast node quickly from the neighbor table. Configuration 3 and 4 are analogous to 1 and 2 with a twice as long beacon interval, i.e. with $5, 3, 2$ and $1\,s$. In Fig. 8 the results for these four configurations are depicted. Especially configuration 2 and 4 which use the minimum of both time-out intervals show very good results. The delay is reduced to about $100\,ms$ and also the packet delivery ratio is increased to 94% and more.

F. Rx Threshold for Beacons

The minimum required energy for a successful decoding of an incoming signal is set to $-81\,dBm$ in the simulator which equals a transmission range of $250\,m$ with a transmission power of $15\,dBm$. The Rx threshold for the beacons was set to $-79, -77, -75,$ and $-75\,dBm$. In the case of the isotropic two ray radio propagation model, where the signal attenuates with $\frac{1}{d^4}$, we can map this power levels to distances of 223, 199, 177, and $158\,m$. Thus, only beacons from nodes closer

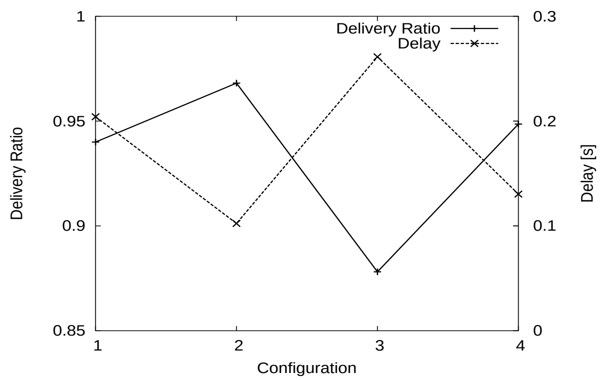

Fig. 8. Speed-based Beaconing Strategy

than these distances are processed. This reduces the probability to select an neighbor out of transmission range but at the same time also reduces the number of neighbors. Fig. 9 shows that the values for $-79\,dBm$ are less good than for the other Rx thresholds. Thus, the RX threshold should not be chosen too close to the minimal power because the performance benefit is limited. As -77, - 75, and $-73\,dBm$ yield similar results and the number of neighbors is decreasing for higher Rx thresholds, which may become a problem in less dense networks, $-77\,dBm$ is the preferred value for our simulations. Compared to the standard GPSR without Rx threshold and pause time $0\,s$ of Fig. 3, all of them improve the delivery ratio by 5% and more. At the same time the delay is also reduced by approximately 40%.

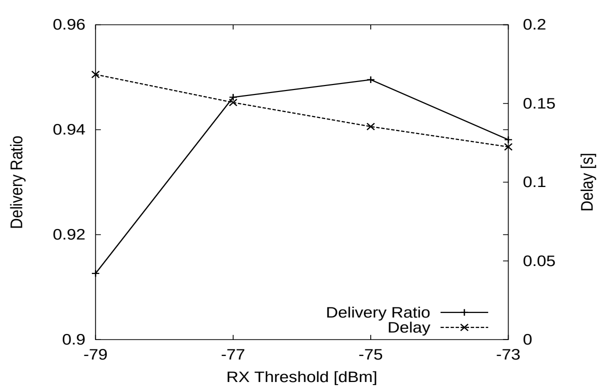

Fig. 9. Rx Threshold for Beacons

G. Estimation of Link Availability

We simulated the GPSR enhanced with the prediction if a neighbor is still within transmission range for different beacon and neighbor time-out intervals. As we can see in Fig. 10, the delay is between 20 and $50\,ms$ compared to 100 and $500\,ms$ of the pure time-based approach in Fig. 6. The increase of the packet delivery ratio, again compared to Fig. 6, is higher for longer time-out intervals. Unlike in the time-based approach,

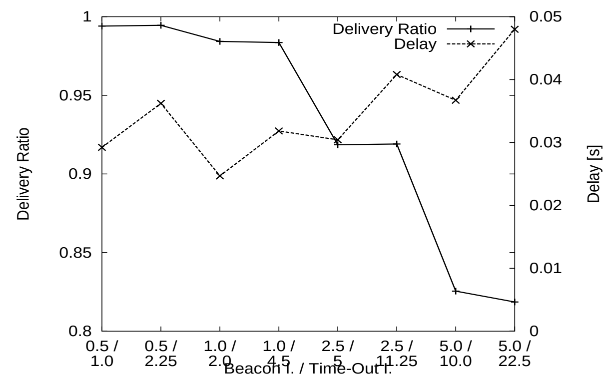

Fig. 10. Prediction of Link Availability

the delivery ratio is almost independent whether the neighbor time-out interval is 2 or 4.5 times the beacon interval. This implies that prediction allows to reliably and timely remove nodes from the neighbor table. The prediction only improves the delivery ratio substantially for long beacon and neighbor time-out intervals. But in all cases, the delay is decreased by a factor of at least four compared to the pure time-based results in Fig. 6. This is again to the reduced number or required RTS retransmission, which is in the same order of magnitude lower.

H. Reactive GPSR

We implemented the reactive GPSR in the following way. The node which broadcasts the ”beacon request packet” waits $4\,ms$ for incoming beacons. Afterwards it selects, among all nodes from which it received a beacon, the node closest to the destination. Nodes receiving a ”beacon request packet” jitter the transmission of their beacons by $2\,ms$ to avoid that all neighbors respond simultaneously. This reactive GPSR achieved an average delivery ratio of 96% and an average delay of 138 ms at a pause time of $0\,s$. The time saved through the more accurate neighbor table outweighs the additional introduced delay of $4\,ms$ at each hop. These results are promising, especially if we consider that this is a very basic reactive version where no optimizations are implemented, e.g. no caching of positions or overhead packets.

I. Theoretical Optimum of Position-based Routing

In a last step, we would like to compare the performance of all the simulations so far with the theoretical optimum what a position based routing algorithm may achieve. The delay of GPSR_bnu and GPSR_bl is depicted in Fig. 11 and is an order of magnitude lower than the delay of the all other simulated protocols, except for the version with prediction. These results show again that the by far most severe reason for the increased delay are the out-dated neighbor tables because GPSR_bnu and GPSR_bl always have up-to-date neighbor table and never select an unreachable next hop. Consequently, no RTS retransmission were observed for GPSR_bl and only very few for GPSR_bnu due to collisions of RTS packets with beacons. These few RTS retransmissions are also the reason for the 10% higher delay of GPSR_bnu. GPSR_bnu and GPSR_bl always achieved 100 percent delivery ratio (not shown). Thus, inaccurate neighbor tables are also a main source of packet loss if the network is not congested. These both protocols are hardly affected by mobility. The delivery ratio and delay are almost constant for all pause times.

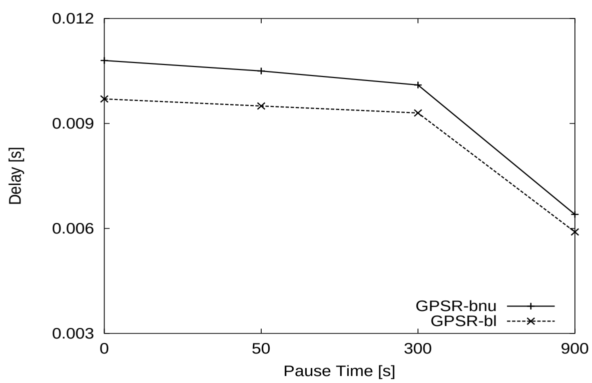

Fig. 11. Delay of GPSR_bnu and GPSR_bl

J. Verification of Theoretical Results

In this section, we briefly verify the consideration and theoretical results from section III and IV with the results of the simulations. We focus on one specific simulation of the standard GPSR with a pause time of $0\,s$ and 400 nodes. From Fig. 5 we know that approximately 11000 packets are handed from the MAC layer back to the routing protocol. Each of these causing an additional delay of $30\,ms$ on average as explained in section III. As 1400 data packets are transmitted during the simulation, the end-to-end delay of each packet is increased by $\frac{11000}{1400} \cdot 30\,ms \approx 235\,ms$. From the results shown in Fig. 11, we know that actual delay without RTS retransmission is only $10\,ms$. The $235\,ms + 10\,ms$ delay correspond to the measured delay of approximately $250\,ms$ in Fig. 4.

From the theoretical results in Table II of section IV, we would expect for this speed interval and a time-out interval of $6.75\,s$ that approximately 30% of the next hops are unreachable. In all the simulations, the average path length was approximately six hops. As 1400 packets were transmitted, we have for each hop that $\frac{11000}{6 \cdot 1400} \approx 1.3$ unreachable nodes are chosen before the packet can be delivered to the next hop what equals a 55% probability to choose an unreachable next hop compared to the predicted 30%. We explain this difference mainly with the assumption made for the theoretical analysis and the argument already given in IV that greedy routing more often selects nodes close to the transmission range which have a higher probability to leave the transmission range within the neighbor time-out interval. The possible degree of influence of these factors is difficult to estimate.

VIII. CONCLUSIONS

In this paper, we first discussed the reasons for and the possible impact of inaccurate neighbor tables in position-based routing. The probability that the routing protocol selects an unreachable node was estimated analytically which showed that this may happen often, especially for small transmission ranges, long time-out intervals, and high node speeds. The simulations support the analytical result and show that the delay can increase by a factor of more than 10 due to inaccurate neighbor tables. We further observed that if a node had perfect accurate information about its neighbors, no packet loss would occur in uncongested networks. The different proposed enhancements to improve the accuracy of the neighbor tables show very different results. Even though, all of them improved the performance in terms of delay and delivery ratio compared to the standard GPSR, the reactive GPSR, the GPSR enhanced with prediction, and the speed-based approach show the best results. For low traffic scenarios where only few packets are sent, the reactive GPSR seems to be the appropriate choice because it eliminates the proactive broadcasting of beacon which exists in all the other GPSR versions. This may be especially important for sensor networks where traffic is transmitted very infrequently and which have high requirements on energy consumption. For delay critical applications in highly mobile network, the prediction-based and speed-based GPSR may be preferred due to the much shorter delays, up to 5 times compared to the reactive GPSR. Furthermore, we state that the neighbor time-out interval should be very short in order to reduce the risk of out-dated neighbor entries because false routing decision are a major source of delay and packet loss. The results of GPSR_bl and GPSR_bnu show that the proposed enhancements are a step in the right direction but also that they are still far from the theoretical optimum such that there remains a lot of room for further improvement. Even if we did not simulate these approaches under other network conditions like lower node density and higher traffic load, the main conclusions should also be valid for other scenarios. To the best of our knowledge, all position-based routing protocols proposed in the literature use a simple time-based beaconing strategy. Thus, if this beaconing was optimized as proposed in this paper, the performance could be easily improved significantly. We would like to mention again that it was not the aim of this work to obtain optimal parameter settings for the different protocols, but only to determine the potential of the proposed enhancements and mainly to point out by simulations and theoretical results that inaccurate neighbor tables may deteriorate severely network performance.

REFERENCES

[1] Charles E. Perkins and Elizabeth Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, New Orleans, USA, Feb. 1999, pp. 90–100.

[2] David B. Johnson, David A. Maltz, and Josh Broch, "DSR: The dynamic source routing protocol for multihop wireless ad hoc networks," in *Ad Hoc Networking*, chapter 5, pp. 139–172. Addison-Wesley, 2001.

[3] Thomas Clausen, Philippe Jacquet, Paul Muhlethaler, Anis Laouiti, Amir Qayyum, and Laurent Viennot, "Optimized link state routing protocol," in *Proceedings of IEEE International Multi-topic Conference (INMIC '01)*, Lahore, Pakistan, Dec. 2001.

[4] Richard Ogier, F. Templin, and M. Lewis, "Topology dissemination based on reverse-path forwarding (TBRPF)," RFC 3684, Feb. 2004.

[5] Jinyang Li, John Jannotti, Douglas S. J. De Couto, David R. Karger, and Robert Morris, "A scalable location service for geographic ad-hoc routing," in *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '00)*, Boston, USA, Aug. 2000, pp. 120–130.

[6] Martin Mauve, Jörg Widmer, and Hannes Hartenstein, "A survey on position-based routing in mobile ad-hoc networks," *IEEE Network*, vol. 15, no. 6, pp. 30–39, Nov. 2001.

[7] Ivan Stojmenovic, "Position based routing in ad hoc networks," *IEEE Communications Magazine*, vol. 40, no. 7, pp. 128–134, 2002.

[8] Damla Turgut, Sajal K. Das, and Mainak Chatterjee, "Longevity of routes in mobile ad hoc networks," in *Proceedings of the IEEE semiannual Vehicular Technology Conference (VTC '01)*, Rhodes, Greece, May 2001, pp. 2833–2837.

[9] Chai-Keong Toh, "Associativity based routing for ad hoc mobile networks," *IEEE Personal Communications Magazine*, vol. 4, no. 2, pp. 103–139, Mar. 1997, Special Issue on Mobile Networking and Computing Systems.

[10] Rohit Dube, Cynthia D. Rais, Kuang-Yeh Wang, and Satish K. Tripathi, "Signal stability-based adaptive routing (SSA) for ad hoc mobile networks," *IEEE Personal Communications Magazine*, vol. 4, no. 1, pp. 36–45, Feb. 1997.

[11] Michael Gerharz, Christian de Waal, Peter Martini, and Paul James, "Strategies for finding stable paths in mobile wireless ad hoc networks," in *Proceedings of the 28th Annual IEEE Conference on Local Computer Networks (LCN '03)*, Köngiswinter, Germany, Oct. 2003, pp. 130–139.

[12] A. Bruce McDonald and Taieb F. Znati, "A path availability model for wireless ad hoc networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '99)*, New Orleans, USA, Sept. 1999, pp. 35–40.

[13] Shengming Jiang, "An enhanced prediction-based link availability estimation for MANETs," *IEEE Transactions on Communications*, vol. 52, no. 2, pp. 183–186, Feb. 2004.

[14] William Su, Sung-Ju Lee, and Mario Gerla, "Mobility prediction and routing in ad hoc wireless networks," *International Journal of Network Management*, vol. 11, no. 1, pp. 3–30, Jan. 2001.

[15] Ivan Stojmenovic, Mark Russell, and Bosko Vukojevic, "Depth first search and location based localized routing and QoS routing in wireless networks," *Computers and Informatics*, vol. 21, no. 2, pp. 149–165, 2002.

[16] Ian D. Chakeres and Elizabeth M. Belding-Royer, "The utility of hello messages for determining link connectivity," in *Proceedings of the 5th International Symposium on Wireless Personal Multimedia Communications (WPMC '02)*, Honolulu, Hawaii, Oct. 2002, pp. 504–508.

[17] Brad Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '00)*, Boston, USA, Aug. 2000, pp. 243–254.

[18] Jingyuan Zhang, "Location management in cellular networks," in *Handbook of wireless networks and mobile computing*, pp. 27–49. John Wiley & Sons, Inc., 2002.

[19] Goutham Karumanchi, Srinivasan Muralidharan, and Ravi Prakash, "Information dissemination in partitionable mobile ad hoc networks," in *Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems (SRDS '99)*, Lausanne, Switzerland, Oct. 1999, pp. 4–13.

[20] Michele Zorzi and Ramesh R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Multihop performance," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 337–348, 2003.

[21] Holger Füssler, Jörg Widmer, Michael Käsemann, Martin Mauve, and Hannes Hartenstein, "Contention-based forwarding for mobile ad-hoc networks," *Elsevier's Ad Hoc Networks*, vol. 1, no. 4, pp. 351–369, Nov. 2003.

[22] Brian Blum, Tian He, Sang Son, and Jack A. Stankovic, "IGF: A state-free robust communication protocol for wireless sensor networks," Tech. Rep. CS-2003-11, Department of Computer Science, University of Virginia, USA, 2003.

[23] Marc Heissenbüttel and Torsten Braun, "A novel position-based and beacon-less routing algorithm for mobile ad-hoc networks," in *Proceedings of the 3rd IEEE Workshop on Applications and Services in Wireless Networks (ASWN' 03)*, Bern, Switzerland, July 2003, pp. 197–210.

[24] Prosenjit Bose, Pat Morin, Ivan Stojmenovic, and Jorge Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," in *Proceedings of the 3th International ACM Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM '99)*, Seattle, USA, Aug. 1999, pp. 48 – 55.

[25] William Navidi and Tracy Camp, "Stationary distributions for the random waypoint mobility model," *IEEE Transactions on Mobile Computing*, vol. 3, no. 1, pp. 99–108, 2004.

[26] Sally Floyd and Van Jacobson, "The synchronization of periodic

routing messages," *IEEE/ACM Transactions on Networking*, vol. 2, no. 2, pp. 122–136, Apr. 1994.

[27] Jungkeun Yoon, Mingyan Liu, and Brian Noble, "Random waypoint considered harmful," in *Proceedings of the 22st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)*, San Francisco, USA, Mar. 2003.