Evaluating the Limitations of and Alternatives in Beaconing ⋆

Marc Heissenbüttel * Torsten Braun Markus Wälchli Thomas Bernoulli

Institute of Computer Science and Applied Mathematics, University of Bern, Switzerland

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. In highly dynamic networks, the information stored in the neighbor table is often outdated and does no longer reflect the actual topology of the network causing retransmissions and rerouting that consume bandwidth and increase latency. An analysis on the possible impact of beacons due outdated and inaccurate neighbor tables is needed. We quantify by analytical and simulation means the possible performance loss and explore the limitations of position-based routing protocols which use beaconing. In highly mobile ad-hoc networks, the delay can increase by a factor of 20. The neighbor table inaccuracy is the main source of packet loss in uncongested networks. We propose and evaluate several concrete mechanisms to improve the accuracy of neighborhood information, e.g., by dynamic adaptation of the timer values when beacons are broadcasted, and show their effectiveness by extensive simulation.

Key words: Ad-hoc networks, Routing, Beaconing, Energy consumption

---

⋆ 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.

* Corresponding author.
Email address: heissen@iam.unibe.ch (Marc Heissenbüttel).

1 Introduction

A wireless mobile ad-hoc network operates without any centralized administration and does not rely on any fixed infrastructure. Instead the network is completely self-organizing and the communication is maintained on a peer-to-peer basis between the mobile hosts. If two hosts that wish to communicate are not within range, other intermediate nodes act as relay stations. Due to the mobility of the nodes, changes to the network topology may be frequent and unpredictable. Furthermore, nodes may suddenly be switched on/off, causing new links to appear and established links to vanish. Routing in such a dynamic environment is a difficult task and has been subject of extensive research over the past years. A lot of topology-based routing protocols have been proposed that either establish routes 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 causing interruptions and increased latency. Flooding of control packets is costly in terms of battery power and bandwidth.

Position information available at each node is the key enabler for a new class of protocols, called position-based routing protocols. Position-based routing protocols exploit location information to enhance routing and do no longer route packets solely based on node IDs. Forwarding decisions are based on absolute or relative position of the current node (e.g. provided by GPS), the positions of neighboring nodes (by receiving periodically transmitted hello messages, called beacon) and the destination (e.g. obtained via a location service [5]). Each packet is routed independently at each node and forwarded in a greedy manner to a neighboring node that reduces the distance to the destination. If a node does not have any neighbor closer to the destination, a recovery mechanism is applied to recover from this local minimum. Many algorithms hereby apply the famous face routing where packets are routed according to the right-hand rule on the faces of a locally extracted planar subgraph [6]. These protocols are inherently more robust to changes in the network topology than topology-based routing protocols because in case of topology changes they only require local rerouting by simply selecting another neighbor as the next hop if the previously selected neighbor is no longer available. These protocols furthermore naturally support geocasting. An overview of position-based routing algorithms and location services can be found e.g. in [7] and [8].

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. This table contains all nodes within the transmission range from which it receives beacons. 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 due to 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. Although changes in the network topology do not induce overhead in terms of control packets to inform nodes about the changed topology such as in topology-based routing protocols but only require local modification of the neighbor table, inaccurate or outdated neighborhood information may severely affect position-based routing protocols. The reason is that data packets may not be delivered to the next hop or may be delivered to suboptimally located neighbors.

We first evaluate purely deductively the possible impact of inaccurate and outdated neighbor information on the performance of position-based routing protocols. These considerations are supported by theoretical analyses which show that outdated neighbors are not the exception in dynamic networks. In a next step, we propose and simulate two position-based protocols with always up-to-date neighbor information which allow providing a theoretical bound on the performance of position-based protocols. In a third step, we evaluate several optimizations to the beaconing mechanism of position-based protocols that help to improve their reliability and performance. These approaches dynamically adapt the time interval between the transmissions of beacons with respect to the encountered mobility, use predication of future position, and use a reactive beaconing to improve neighbor table accuracy. The paper shows possible limitations of position-based routing protocols which make use of beaconing to obtain local neighborhood information. The analysis of the impacts of beaconing and possible ways to alleviate them gives some insights to future solutions to these problems and provides guidelines for researchers for improving the design of position-based routing protocols.

The remainder of this paper is organized as follows. These analytical and simulation results indicate that improvements of neighbor information are required and worthwhile and that position-based routing protocols which make use of beaconing to obtain local neighborhood information have strong limitations. Section 2 gives a survey on related work. In Section 3, we give an overview of the possible direct and indirect effects caused by the periodical broadcasting of beacons. We assess analytically the impact of inaccurate and outdated neighbor information on the performance of position-based routing protocols in Section 4. In Section 5, we simulate a standard position-based protocol over various scenarios to study the effects of outdated neighbor tables. In addition we propose and evaluate several optimizations to improve their accuracy. Finally, Section 6 concludes the paper.

2 Related Work

To the best of our knowledge, most papers that propose position-based routing protocols based on beaconing do not analyze in depth the potential impact of beaconing or inaccurate neighbor tables on the performance. One notable exception is [9], where the authors compared the packet delivery ratio and routing overhead for different time intervals between beacons. In the chosen scenario, it was observed that the packet delivery ratio varied only slightly for beacon intervals of $1\,s$ and $1.5\,s$. However when the beacon interval was set to $3\,s$, the packets lost were approximately doubled caused by the increased inaccuracy of the neighbor table. Therefore, the related work discussed hereafter is broadly classified into three groups. One group includes algorithms that deal with link incidents and ways to predict them in topology-based routing protocols as outdated and inaccurate neighbor tables in position-based routing protocols may be considered as the analogue of link incidents in topology-based protocols. The transmission of beacons may be considered as simple location database updates at neighboring nodes and the second group of related work comprises various approaches that deal with updating strategies of a location database in both, mobile ad-hoc and also cellular networks. Position-based routing protocols for mobile ad-hoc networks that avoid beaconing completely fall into the third group.

Link Incidents in Topology-Based Protocols 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 minimizing the latency and induced routing overhead of link incidents. In [10], the expected lifetime of routes was investigated in order to reschedule the route discovery before actual link breakage to avoid the disruption of communication. Unlike these protocols, several other protocols were proposed that take the stability of links and paths into account to minimize the number of link incidents in the first place. In [11], the lifetime of a link is taken into account during route discovery, whereas [12] considers feedback from the link layer about signal strength as primary routing metric. With such additional information, they were able to reduce the number of route breaks by a factor of approximately two. In [13–15] results from analytical derivations and observations made by simulations are used to design new routing metrics which favor more stable paths. The average route lifetime could be increased by up to 50%. Based on link availability estimations, a metric for path selection in terms of reliability and resilience is introduced in [16] and refined in [17]. The average delay of a standard routing protocol could be reduced from $45\,ms$ to $24\,ms$ and at the same time the packet delivery ratio could be increased by 10%. 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. This information can be used to estimate when a link will break and to reconfigure routes in a timely fashion [18]. Thereby the delivery ratio could be increased by 15% and the control traffic significantly reduced as fewer costly route discoveries were necessary. Unlike [18], where GPS information is only applied to maintain routes, [19] additionally makes use of location information in the routing decision itself to establish paths in a depth-first search way with certain QoS requirements by estimating the connection time between two neighboring nodes. Factors that influence the use of hello messages for determining link connectivity in topology-based protocols were studied in [20].

Updating Location Databases Even though beaconing was not explicitly studied in ad-hoc networks, the determination of the "best" time when to update information stored at distant databases was studied in other contexts for both, cellular and ad-hoc, wireless networks. Basically all cellular networks have to implement a location management scheme to keep track of a node's position if it has moved to a new cell by updating a central database. Numerous such location management schemes were proposed for cellular networks in the literature [21]. They differ in the frequency of updating the database, the criteria when a node sends an update, when the entry in invalidated, etc. Location management schemes for mobile ad-hoc networks propose different mechanisms to store nodes' position since obviously no such fixed central data bases as in cellular wireless network exist. In [22], a central home region for each node, which can be derived from a node's identifier, is used to keep up-to-date position information. Nodes send their positions periodically to their home region and all nodes within the region store this information. A node that wishes to contact another node sends a request to the home region and receives a reply from any node which has stored the position. In [5] and [23], each node has multiple location servers distributed all over the network. They rely on a quad-tree hierarchy to structure the network area where there is one location server for each node at each hierarchy level. In [5], each node sends its current position to these location servers when it has moved a certain distance, whereas in [23] this is done periodically. Querying nodes can find the location server on the nearest common hierarchy level to obtain a node's position. [24] and [25] proposed quorum-based location update scheme where a quorum is defined as a set of nodes. Nodes distribute their positions to nodes in a "write" quorum and send requests about other nodes' locations to "read" quorums. The algorithm ensures that two arbitrary read and write quorums have a non-empty intersection and thus the position of a node can always be provided. General dissemination and replication of data in ad-hoc networks was studied in [26]. The authors propose different strategies when to trigger updates. A fundamental difference of these approaches to the beaconing mechanism studied in this paper is that while we consider the case of updating *only* neighboring nodes *frequently*, the other approaches focus on information that is transmitted infrequently to certain distant nodes. Another interesting location management scheme was proposed in [27]. Each node continuously samples its location and constructs a model of its movement. Nodes flood their model through the network. Whenever a node's distance from its actual location to its predicated location in the model is larger then a certain threshold, the updated predication model is flooded again in the network. Similarly [28] studied the effects of inaccurate location information caused by mobility on position-based routing protocols and proposed two mobility prediction schemes to mitigate these problems.

Beacon-Less Position-Based Routing Protocols Lately several protocols have been proposed which adopt a new paradigm for position-based unicast routing [29], [30], [31], and [32]. In these protocols the next hop is not determined as usually at the sender of a packet, but instead they 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 simply broadcasted by the sender without requiring any knowledge of neighboring nodes and the protocol takes care that just one of the receiving nodes forwards the packet further. Optimized forwarding is achieved by applying a concept of dynamic forwarding delay. Consequently, the node among all receiving nodes which computed the shortest forwarding delay relays the packet first. This subsequent forwarding is detected by the other nodes and suppresses them to relay the same packet any further. Unlike other position-based routing protocols, those beacon-less protocols do not require nodes to periodically broadcast hello-messages. Therefore, they avoid drawbacks normally caused through beaconing such as extensive use of scarce battery-power and interferences with regular data transmission. Simulation results also indicated that these protocols provide efficient and robust routing in highly dynamic mobile ad-hoc networks. Even in highly dynamic scenarios where nodes toggle between active/sleep several times a second [31] or move at speed of $100\,m/s$ [32], the delivery ratio was still almost 100% and the delay was not significantly higher than for complete static scenarios. On the other hand, some new problems occur through this beacon-less routing mechanism. In these protocols, packets are frequently duplicated in case of irregular transmission ranges. A forwarding node may not overhear the forwarding of other neighbors due to obstacles or interferences and thus transmit another copy of the same packet.

3 Qualitative Assessment of Beaconing

In this section, we discuss the impact of beaconing on the performance of routing protocols and on the overall network. In the following sections, these considerations will then be also evaluated analytically and quantitatively through simulations.

The periodic broadcasting of beacons has several drawbacks such as unnecessary utilization of network resources and interferences with regular data packets. 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 broadcast beacons to update their neighbors. We can distinguish between direct and indirect impacts of beaconing. We classify the effects that are caused by the transmissions of beacons such as the additionally consumed energy and bandwidth as direct impacts. Indirect impacts comprise all effects that are not caused by transmission of beacons themselves, but by the information distributed through the beacons. Routing protocols rely on this information and may take wrong or suboptimal forwarding decision due to the fact that nodes do not have complete accurate topology information about their neighborhood as beacons are broadcasted only periodically. Thus, if nodes are mobile or the topology changes due to any other reason, the topology as perceived by the nodes never corresponds to the actual topology. Inaccurate position information provided by GPS or other position services may further increase the deviation. Even though, the indirect impacts are perhaps less obvious, the performance may degrade even more than with the direct effects in terms of increased delay, wasted bandwidth and battery power.

3.1 Direct Impacts

We can observe several direct impacts of beaconing. First, additional energy is used to transmit, receive, and process the beacons. Second, beacons interfere with regular data transmissions and thus increase the number of collisions and subsequent retransmissions, if there is no separate signaling channel. This not only reduces the available bandwidth, but at the same time also increases the delay and the congestion in the network. Third, beaconing introduces control overhead. A part of the bandwidth is used for this control traffic and not available for user data.

We briefly discuss in more detail the effects of beaconing on the power consumption in more detail as power consumption is very critical factor in ad-hoc networks. Major sources of energy consumption for wireless devices were identified in [33–35].

- The fixed costs of sending a packet are large compared to the incremental costs for larger packets.
- Receiving a message causes high costs. If a message is received by several neighbors, the total costs of receiving the message may be much larger than for sending it.
- After having received the packet header, a node can determine if it is the intended receiver and can discard the packet otherwise. Discarding is a strategy that allows nodes entering a sleep mode for the duration of the transmission of a packet if it is not the intended receiver. Thus, receiving a packet, passing it to the protocol stack and processing it costs generally much more than just discarding it at the network interface.
- A node receiving and processing packets destined for other nodes wastes a substantial amount of energy. This is called overhearing, which is, e.g., the case if nodes operate in promiscuous mode.
- Idle listening where a node just listens to the medium to receive possible traffic that is not sent causes high costs too.

Let us briefly consider the costs of beaconing in terms of energy consumption based on these facts. Due to startup costs, the transmission of beacons is costly even though beacons are small packets. Furthermore, beacons are always broadcasted such that all neighbors receive and need to process the packets and cannot discard them. Many protocols propose to piggy-back beacons on data packets to reduce the total number of transmitted packets. But, it may even increase the power consumption as piggy-backing requires nodes to process every received packet. This also includes unicast packets addressed to other nodes, such that packets can no longer be discarded at the network interface. Furthermore, in scenarios with little data traffic, nodes have to listen to the medium only to receive beacons and can enter less frequently power saving sleep states, which strongly depends on the used MAC protocol however. In view of these results, we may conclude that protocols that use beaconing are highly suboptimal in terms of power consumption.

3.2 Indirect Effects

In position-based routing protocols, nodes forward packets based on the perceived topology, which typically does not correspond to the actual topology, because nodes have moved since their last beacon transmission. Neighbor tables actually do not correspond to the physical topology and are always inaccurate, except for static networks.

We can broadly distinguish between three kinds of inaccuracies. First, nodes are listed in the neighbor table with inaccurate positions, but they are still within the transmission range. Second, a node moved into the transmission range, but it is not visible since no beacon was received yet. These two scenarios have only minor effects on the routing protocol. The routing protocol may take suboptimal decisions and not forward packets over the best-located neighbor.

The third scenario has a much stronger impact when nodes are wrongly listed in the neighbor table even though they moved out of transmission range since their last beacon transmission. If such an unreachable node is chosen by the routing protocol as the next hop, the MAC protocol will not be able to deliver the packet. After several retransmission attempts, the MAC protocol either drops the packet or notifies the routing protocol of the failed delivery and passes the packet back to the routing protocol. The routing protocol in turn selects a different next hop and hands the packet over to the MAC protocol again. This process is repeated until the packet can be delivered eventually to the next hop. Rerouting increases the delay, reduces the effective available bandwidth, and consumes energy for retransmissions. If IEEE 802.11b is used, packets are retransmitted up to seven times before the MAC layer gives up and assumes the next hop to be unreachable. Therefore we can roughly estimate that the effective used bandwidth is maximally one seventh of the total available bandwidth during the time a node tries to deliver a packet to an unreachable neighbor. As the retransmission timing depends greatly on the contention, i.e. on the number of active neighbouring nodes, especially in multihop MANETs, the ratio between actually used and available bandwidth can become even worse.

Furthermore, the power consumption can also be increased significantly depending on the ratio between sending/receving/idle/sleep. In order to roughly estimate the induced delay, we consider having IEEE 802.11 on the MAC layer. 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 a backoff time from the contention window for the next transmission. If all seven retransmissions fail because the next hop is out of transmission range, the expected delay is $\frac{31+63+...+1023+1023}{2} \cdot 20\,\mu s \approx 30\,ms$. Note that this additional delay of $30\,ms$ is introduced for every selected unreachable neighbor, which can happen multiple times at each node before the packet is successfully delivered at the next node. This also applies for the additionally used bandwidth and energy consumption as stated above. When we refer to neighbor table inaccuracy, we often only refer to this third scenario, which has by far the most severe consequences.

We furthermore mentioned typically mobility as the only 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, interferences, adjustment of transmission and reception parameters, etc. Throughout this paper, we consider speed as a proxy for any kind of topology changes.

Topology changes are not the only source of inaccurate neighbor tables. In addition, other factors contribute to the inaccuracy. Beacons are broadcasted and most MAC layer protocols do neither require nor provide acknowledgments for broadcast transmissions such that the delivery is not guaranteed. Furthermore, many position-based routing protocols apply a greedy forwarding strategy where packets are forwarded to neighbors close to the destination [36,9,37]. Therefore, the chosen neighbor is close to the boundary of the transmission range. This increases the probability that the node will soon become unreachable. A third factor is that the neighbor time-out interval is often set to a multiple of the beacon interval to avoid that nodes are constantly inserted and removed from the neighbor table if one or two beacons are lost. This longer neighbor time-out interval further increases the probability that a neighbor has meanwhile left the transmission range and is no longer available. There are also more practical factors that contribute to inaccurate neighbor tables. Packets transmitted at lower rates typically use more robust modulation schemes and thus can be still decoded at farther distances than packets transmitted at a higher rate. [38] observed that IEEE 802.11b cards transmit broadcast packets constantly at 2 Mbps whereas unicast packets can be transmitted at higher rates. Thus, the set of neighbors may vary for beacons and data packets.

4 Analytical Evaluation

After having discussed reasons for outdated and inaccurate neighbor tables and possible implications, we would like to analytically estimate the likelihood of the occurrence of such events. We use the unit disk graph network model, i.e., a fixed isotropic transmission range with radius 1 for all nodes and an unbound simulation area. 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 unbound area is to simplify our analysis by having a uniform moving direction of the nodes in the interval $[0, 2\pi]$, a uniform distribution of the nodes, and travel distances independent of the nodes' locations. All these conditions do not hold in the standard random waypoint model [39]. Furthermore, we assume that nodes do not change their direction or speed for the time interval under consideration, i.e. the neighbor time-out interval, to simplify the analysis. As this time interval is short and only in the order of a few seconds, this assumption is reasonable for realistic movement patterns. We consider two nodes A and B within each others transmission range and calculate the probability that they leave each others' transmission range within a certain time interval t, namely the neighbor time-out interval. The crucial point in the derivation is to notice that instead of having both nodes moving, we assume node B being static and node A moving with their relative speed vector. This assumption is valid as nodes move independently of each other and have symmetric transmission ranges. Therefore, we first derive the expected value of the difference of two arbitrary speed vectors in the used mobility model. Finally, we calculate the size of the area that was covered by a node's transmission range and is no longer when the node moves at the expected relative speed for the time interval t. The size of this area to the overall transmission range is the probability that a neighbor has left the transmission range.

4.1 Probability Density Function of the Speed

We derive the probability density function f_S of the nodes' speed if they choose uniform randomly a speed in the interval $[v_{min}, v_{max}]$. For the probability density function f_S of the speed s, the following holds trivially by definition of the probability density function.

$$\int_{v_{min}}^{v_{max}} f_S(s)\, ds = 1$$

Since the distance to the next waypoint is independent of the speed, we may assume without loss of generality that all trips have the same distance, say 1. Because a trip with a smaller speed takes inverse proportionally longer than the same trip at a higher speed, we have that $f_S(s)$ must be proportional to $1/s$ in the interval $[v_{min}, v_{max}]$ and 0 otherwise. We immediately have for a certain constant k that

$$\int_{v_{min}}^{v_{max}} \frac{k}{s}\, ds = 1$$

which yields by integration and some algebra

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

Thus, f_S is

$$f_S(s) = \begin{cases} \frac{1}{s \ln\left(\frac{v_{max}}{v_{min}}\right)} & : \quad v_{min} \le s \le v_{max} \\ 0 & : \quad \text{otherwise} \end{cases} \tag{1}$$

From (1), we obtain the expected average node speed $E(v_{min}, v_{max})$.

$$E(v_{min}, v_{max}) = \int_{v_{min}}^{v_{max}} s \cdot \frac{1}{s \cdot \ln\left(\frac{v_{max}}{v_{min}}\right)} \, ds = \frac{v_{max} - v_{min}}{\ln\left(\frac{v_{max}}{v_{min}}\right)} \tag{2}$$

4.2 Relative Speed of Two Nodes

Let the speed vectors $\vec{a}, \vec{b}$ of two arbitrary nodes 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 relative speed vector $\vec{a} - \vec{b}$ in Cartesian coordinates is given by

$$\vec{a} - \vec{b} = (s_a \cos(\alpha) - s_b \cos(\beta), s_a \sin(\alpha) - s_b \sin(\beta))$$

The velocity of the relative speed vector is the norm of $\vec{a} - \vec{b}$ which is

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

We do not need to consider the corresponding angle of $\vec{a} - \vec{b}$ as the transmission ranges are isotropic and moving directions are uniform over the whole interval $[0, 2\pi]$. It is well known that for a random vector $\mathbf{X} = (X_1, \ldots, X_n)$ with the joint density function $f_{\mathbf{X}}(x)$ and a function $\varphi : \mathbf{R}^n \to \mathbf{R}$, the expected value is

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

If the X_i are independent, we have that

$$E_\varphi(\mathbf{X}) = \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} \varphi(x_1, \ldots, x_n) f_1(x_1) \ldots f_n(x_n) \, dx_1 \ldots dx_n$$

where the f_i are the probability density functions of X_i.

Thus, the expected value $E_{rel}(v_{min}, v_{max})$ for the norm $|\vec{a} - \vec{b}|$, which is the expected relative speed of two arbitrary nodes, is given by

$$E_{rel}(v_{min}, v_{max}) = \int_{v_{min}}^{v_{max}} \int_{v_{min}}^{v_{max}} \int_{0}^{2\pi} \int_{0}^{2\pi} \sqrt{{s_a}^2 + {s_b}^2 - 2 s_a s_b cos(\alpha - \beta)} \cdot f_S(s_a)\, f_S(s_b)\, f_A(\alpha)\, f_B(\beta)\, d_\alpha\, d_\beta\, ds_a\, ds_b$$

where f_S is the density function of the speed as given in (1) and f_A, f_B are the density functions of α and β, respectively. As the moving direction of nodes is uniform in the interval $[0, 2\pi]$, we have that $f_A(\alpha) = f_B(\beta) = \frac{1}{2\pi}$.

We can simplify this formula by substituting $\alpha - \beta$ by γ. The probability density function f_Γ of $\gamma = \alpha - \beta$ is given by

$$f_\Gamma(\gamma) = \begin{cases} \frac{2\pi - |\gamma|}{4\pi^2} & : \quad 0 \le |\gamma| \le 2\pi \\ 0 & : \quad \text{otherwise} \end{cases} \tag{3}$$

This yields for the expected relative speed $E_{rel}(v_{min}, v_{max})$ the following integral where s_a, s_b and γ are distributed according to f_S in (1) and f_Γ in (3), respectively.

$$E_{rel}(v_{min}, v_{max}) = \int_{v_{min}}^{v_{max}} \int_{v_{min}}^{v_{max}} \int_{-2\pi}^{2\pi} \sqrt{{s_a}^2 + {s_b}^2 - 2 s_a s_b cos(\gamma)}\, f_S(s_a)\, f_S(s_b)\, f_\Gamma(\gamma)\, d_\gamma\, ds_a\, ds_b$$

Unfortunately, we can not solve this integral analytically and give the values obtained by numerical integration for some specific speed intervals in Tab. 4.2. We can observe for example that even though a speed interval of $[1, 40]\, m/s$ seems to be a very high node mobility scenario, the expected average speed $E(v_{min}, v_{max})$ of the nodes is only about $10\, m/s$, because most nodes move slower than the arithmetic middle of $20.5\, m/s$. On the other hand, the expected relative speed $E_{rel}(v_{min}, v_{max})$ is approximately the arithmetic middle and approximately 50% higher than the expected speed $E(v_{min}, v_{max})$ of one single node.

Table 1
Expected speed for different v_{min} and v_{max}

$v_{min}\,[m/s]$	$v_{max}\,[m/s]$	$E_{rel}(v_{min}, v_{max})\,[m/s]$	$E(v_{min}, v_{max})\,[m/s]$
1	10	5.69	3.91
1	20	9.64	6.34
1	40	16.68	10.57
10	20	18.83	14.43
10	40	29.55	21.64

4.3 Size of Uncovered Area

We want to determine the size of the area $A(r, d)$ that was initially covered of a node's transmission range and is no longer after it has moved a certain distance d from A to A' where r is the transmission radius. The size of the area $A(r, d)$ is depicted in Fig. 1.

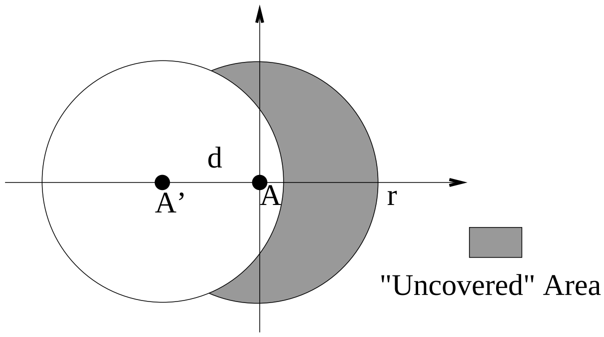


Fig. 1. Area uncovered if node moves from A to A'

We immediately obtain for the size of $A(r, d)$ above the x-axis, which is just half the size of $A(r, d)$, that

$$\frac{A(r,d)}{2} = \int_{-\frac{d}{2}}^{r} \sqrt{r^2 - x^2}\, dx - \int_{-\frac{d}{2}}^{-d+r} \sqrt{r^2 - (x+d)^2}\, dx$$

which yields by integration

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

and finally we obtain for $A(r, d)$

$$A(r,d) = \frac{d}{2}\sqrt{4r^2 - d^2} + 2r^2 \arcsin\left(\frac{d}{2r}\right) \tag{4}$$

We can now calculate the probability p that a node B is no longer within transmission range of a node A after t as follows. The given speed interval immediately yields the expected relative speed $E_{rel}(v_{min}, v_{max})$. We obtain the expected distance, which a node moves relative to any arbitrary other node within t, by multiplying E_{rel} with t. From the expected distance d and the transmission range r, we immediately obtain $A(r, d)$ from (4), i.e., the size of the area uncovered within the neighbor time-out interval t. As node B is static and uniformly and independently distributed, the probability p that B is out of transmission range after node A has moved to A' equals the ratio of $A(r, d)$ to the size of the whole transmission range $r^2\pi$.

$$p = \frac{A(r,d)}{r^2\pi} = \frac{A(r, E_{rel}(v_{min}, v_{max}) \cdot t)}{r^2\pi}$$

In other words, with a probability p an entry in the neighbor table is not valid and corresponds to a node, which is no longer available, i.e., p is also the percentage of invalid entries in the neighbor table. In Fig. 2 the respective values are exemplary given for transmission radii of $r = 250\,m$ and $r = 100\,m$ and different speed intervals.

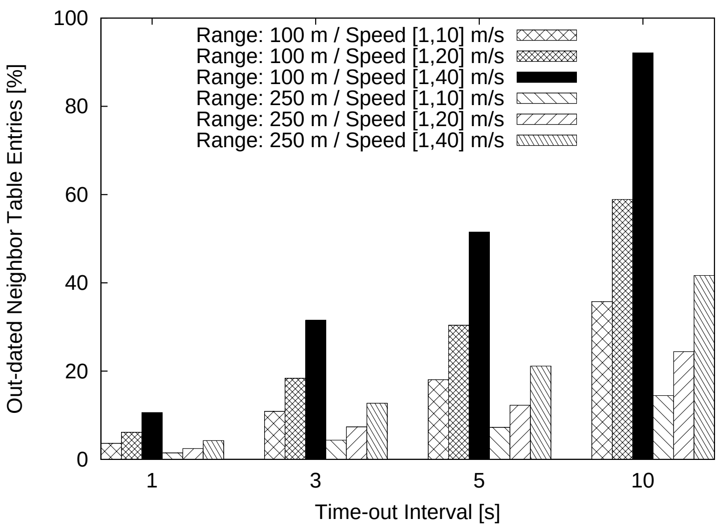


Fig. 2. Expected percentage of outdated neighbors

Even for a large $r = 250\,m$ and for slow $v_{max} = 10\,m/s$, the percentage of outdated 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 number of outdated neighbors is almost inverse proportional to the transmission radius. A 2.5

times smaller transmission radius yields an about 2.5 times higher probability p. A similar proportionality also holds for the relative speed E_{rel} and the number of outdated neighbors. This rough estimation of the percentage of outdated neighbor entries does not account for other factors such as discussed in Section 3. The probability for an entry in the routing table to become outdated depends also on the distance to the respective node, which was not considered in the analysis. Therefore, as a routing protocol typically selects a next hop close to the boundary of the transmission range, the percentage of unreachable next hops selected by the routing protocol will be even higher.

The periodical broadcasting of beacons may also result in neighbors that remain undetected for a certain time. The probability that neighboring nodes exist which are unknown and thus not listed in the neighbor table can be calculated completely analogously. For symmetry reasons, the size of the area which is newly covered by a node's transmission range equals the previously calculated size of the uncovered area $A(r, d)$. The only difference is that t is no longer the time-out interval but the beacon interval. As the beacon interval is normally much shorter, the number of undiscovered neighbors is only a fraction of the number of outdated neighbors.

These considerations give an indication for the severeness of the problem of inaccurate neighbor tables, which occur frequently and have a non-negligible impact on the performance of position-based routing protocols. This also provides justification to reconsider position-based routing protocols and try to assess the impact by simulations and evaluate simple optimizations, which help to improve the accuracy of neighbor tables.

5 Simulations

In this section, we simulate a standard position-based routing protocol over various scenarios. Based on the results, we propose and evaluate several optimizations. In a first step, we identify an appropriate simulation scenario, which produces significant results and permits to assess more easily the goodness of the optimizations. Afterwards, we evaluate the performance of two optimal routing protocols, which have completely accurate neighbor tables by using the global simulator data and thus never select unreachable nodes as next hops. The significantly better performance provides justification to propose and evaluate possible optimizations, whose objectives are the improvement of the neighbor table accuracy.

GFG [36] and the latter published GPSR [9], which extends GFG by MAC-layer enhancements, are perhaps the most well-known position-based routing protocols. We refer to these algorithms together as GFG/GPSR in this paper. In all simulations we used GFG/GPSR as the underlying position-based routing protocol. A packet is routed in a greedy manner towards the position of the destination. Each node selects that 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 a recovery mode and the packet is routed according to the right-hand rule with face routing [6]. Right-hand routing is only performed on the faces of a locally extracted planar subgraph to avoid loops. In accordance with the parameter values chosen in [9], the beacon interval and the neighbor time-out interval are set to $1.5\,s$ and $6.75\,s$ respectively. As also proposed in [9], we implemented changes in the MAC layer protocol to optimize routing and make IEEE 802.11 more robust in mobile wireless scenarios. The most important optimization is that a packet is not dropped if it cannot be delivered, but handed back to GFG/GPSR for rerouting. Even if we use other more sophisticated position-based protocols, which were shown to perform better than GFG/GPSR such as GOAFR [37], GOAFR+ [40] GRA [41], they would suffer from the similar performance degradation as they also rely on neighbor information provided by beacons to forward packets greedily. GOAFR and GOAFR+ basically enhance the right-hand routing of GFG/GPSR by a bounding ellipse and circle, respectively, restricting the searchable area. On the other hand, GRA does not only take locally optimal decision. A node forwards a packet to the closest node to the destination among all nodes it is aware of, i.e., this also includes nodes that are more than one-hop away. If such a node is not found in its routing table, the node initiates a route discovery protocol.

The simulations were conducted using the Qualnet network simulator [42] and the results are averaged over eight simulation runs. Radio propagation is modeled with the isotropic two-ray ground reflection model and random packet loss is not included. The transmission power is set to $15\,dBm$ and the receiver sensitivity to $-81\,dBm$ corresponding to a nominal transmission range of $250\,m$. We use IEEE 802.11b DCF with RTS/CTS operating at a rate of 2 Mbps on the MAC layer. RTS/CTS is often used in wireless multihop networks because it is commonly assumed to alleviate the hidden node problem. However, in the view of the results from [43], there arise some doubts regarding these assumptions. The nodes are placed in a rectangular area of $600\,m$ x $3000\,m$. The simulations last for $900\,s$ and the nodes move according to the random waypoint mobility model. We implemented the stationary distribution of the random waypoint model as described in [44]. Thereby the simulations do not need an initial warm-up phase to reach a stable state.

In order to avoid possible synchronization of beacons between neighboring nodes [45], the beacons are randomly jittered by 50% of the respective beacon interval. The interface queue length is set to 1500 bytes. We have one Constant Bit Rate (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 inaccurate neighbor tables on the routing protocol.

We first conducted several simulations with the standard GFG/GPSR protocol, i.e., with a beacon interval and neighbor time-out interval of $1.5\,s$ and $6.75\,s$ respectively and without any optimizations. Thereby we are able to identify a highly challenging scenario such that the impact of the proposed optimizations can be observed more easily. In Fig. 3 and Fig. 4, the delivery ratio and the average end-to-end delay are depicted for a speed interval of $[1, 40]\,m/s$. The minimum speed was set to $1\,m/s$ as for a minimum speed of 0, the average speed of the nodes also approaches 0 [46]. We also ran simulations with a speed interval of $[1, 20]\,m/s$. The results showed the same trends with the difference that the delay and the packet loss rate were about 30% and 50% lower respectively. As already previously mentioned, 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 approximately $10\,m/s$. Secondly, we wanted to have a challenging scenario for the routing protocol to observe more clearly and definitely the differences in the results. And finally, we consider mobility as a proxy for any kind of topology changes, which could also be caused by sleep cycles of nodes, interferences, adjustment of transmission and reception parameters, etc. as discussed in Section 3.

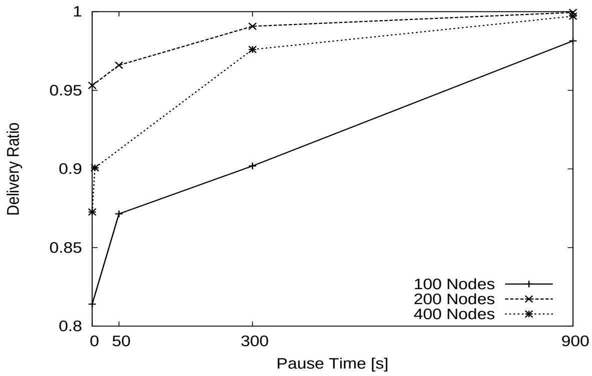


Fig. 3. Delivery Ratio of GFG/GPSR

As expected, the performance suffers in case of low pause times for all different kind of node densities. The pause time is the time that a node remains stationary between two consecutive trips. The optimum is for 200 nodes what is approximately 111 nodes per square kilometer which is also the node density used in [9]. In case of 100 nodes, the density is too low and GFG/GPSR is

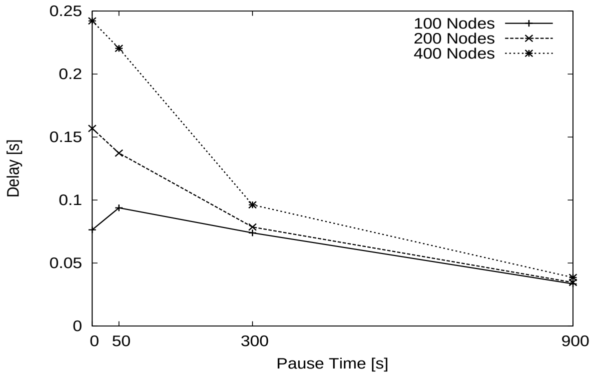


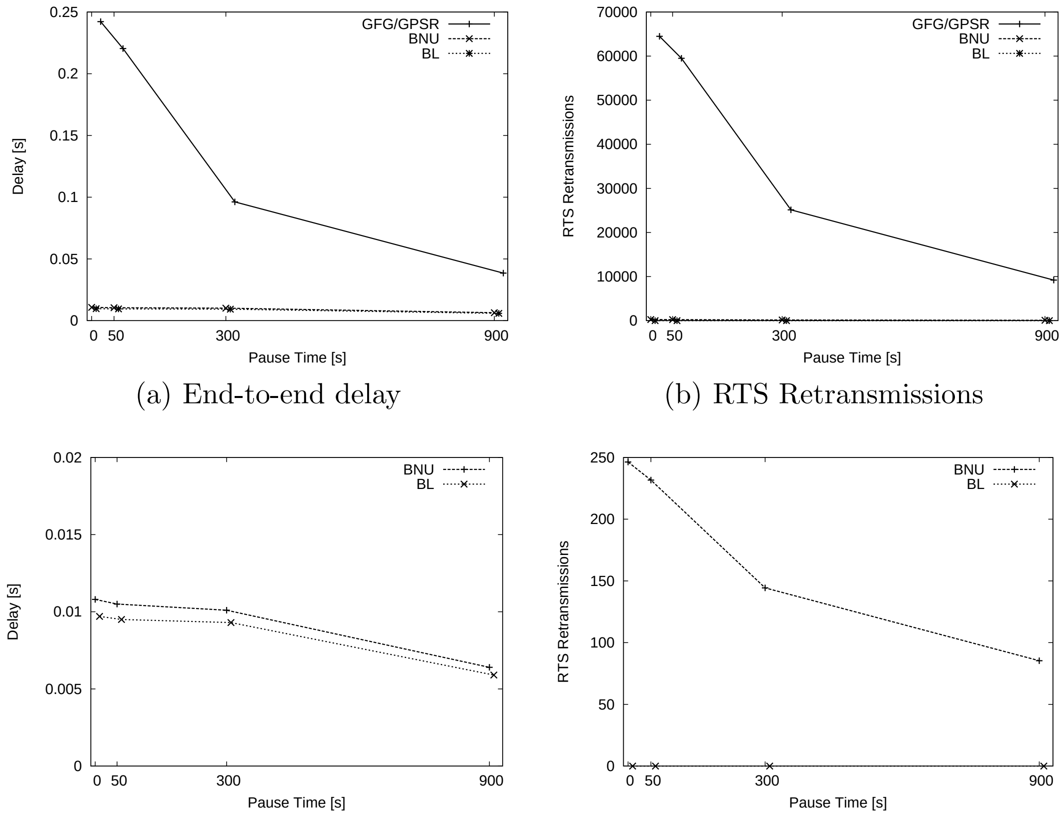
Fig. 4. Delay of GFG/GPSR

not able to achieve a high packet delivery ratio because the network is temporarily disconnected and also because face routing is often applied to forward packets, which then may loop. We observed that in highly mobile networks, a large fraction of the dropped packets is due to cycles when face routing is applied in recovery mode. Actually, face routing only guarantees delivery for static networks. The shorter end-to-end delay with 100 nodes is because packets routed over longer paths with longer delays are more likely to be dropped when face routing is applied. Packets received at the destination have often traveled only short paths and thus show a short delay. More surprisingly is the fact that the performance also suffers with a higher node density of 400 nodes. The reason is that the selected next hop is generally farther away and thus has a higher probability of having left the transmission range causing wrong routing decisions. Therefore, we choose to run all following comparative simulations with 400 nodes in a speed interval of $[1, 40]\, m/s$ and a pause time of $0\, s$, unless given otherwise.

5.2 Optimal Position-Based Routing

We first evaluate two protocols, called BNU (Beacons Not Used) and BL (Beacon Less), which use completely accurate neighborhood information provided by global data of the simulator to determine the next hop. We only use them in order to obtain a "theoretical" limit about the possible performance gain if beaconing of position-based routing is optimized. Except for these accurate neighbor tables, BNU and BL are identical to GFG/GPSR. This enables us to do a kind of "best-case" simulation analysis for position-based routing. These optimal protocols allow assessing explicitly the impact of inaccurate neighbor tables and beaconing on the performance. In BNU, nodes broadcast beacons as with GFG/GPSR, but the position information obtained via these beacons is not used, but taken from the global data. Comparing BNU and GFG/GPSR, we can quantify the influence of inaccurate and outdated neighbor tables. In BL on the other hand, the beacon mechanism is disabled completely. The performance difference between BNU and BL is an indicator for the performance loss solely due to the additional traffic caused by beacons, e.g., by collisions with data packets.

The end-to-end delay and the number of retransmitted RTS packets on the MAC layer are depicted in Fig. 5. An RTS packet is transmitted by the source prior to the data packet transmission to mitigate the hidden node problem in IEEE 802.11 [47]. The intended receiver acknowledges the RTS with a CTS packet. Afterwards the actual data packet is transmitted by the source and acknowledged by the receiver. An RTS retransmission occurs if the source does not receive the CTS from the next hop within a certain time-out interval. In our scenario with very little traffic, RTS and CTS should not collide with other packets. Thus, RTS retransmissions are an indication for the unavailability of the next hop. If the routing protocol selects an unavailable next hop, the MAC layer protocol retransmits seven RTS before giving up the delivery of the packet and handing the packet back to the routing protocol. Consequently, RTS retransmissions are a direct indication for the accuracy of the neighbor tables. Fig. 5(c) and Fig. 5(d) show the same results as Fig. 5(a) and Fig. 5(b) on a different scale for clarity reasons, as the difference between BL and BNU are hardly visible. The delivery ratio for both protocols, BL and BNU, was always 100% and thus not shown. Only very infrequently one packet was lost due to temporary network partition caused by the mobility of the nodes. The delay of BNU and BL is approximately $10\,ms$ independent of the pause time. This is much shorter than of the GFG/GPSR with delays between $60\,ms$ and $210\,ms$. The much higher end-to-end delay of GFG/GPSR is directly correlated with the number of retransmitted RTS packets, where up to 60'000 RTS packets are sent. As an unreachable neighbor causes seven retransmissions, we can assume $60000/7 \simeq 8600$ wrong routing decisions, where the next hop was not available. With 1400 packets transmitted in total, each packet is tried approximately $8600/1400 \simeq 6$ times to be routed to an unavailable neighbor. Each of these wrong routing decisions adds on average $30\,ms$ delay as seen before. Thus, $6 \cdot 30\,ms \simeq 180\,ms$ of the total end-to-end delay is caused by wrong routing decisions due to the outdated neighbor tables. On the other hand, we did not observe any RTS retransmissions for BL and only very few for BNU. The reason is that no unreachable nodes are listed in the neighbor table and packets are always routed to neighbors within transmission range. The few RTS retransmissions for BNU are due to collisions of RTS packets with beacons. These retransmissions are also the reason for the 10% higher delay of BNU compared to BL. The delay of the two optimal protocols remains almost constant for all mobility rates. These results indicate that outdated neighbor tables do not only cause long delays but are also a main reason for packet loss in uncongested networks. For all position-based protocols, which only use local information to forward packets, the delivery ratio and the delay of the two optimal protocols provide an upper and lower bound, respectively. We may therefore conclude from the results in Fig. 5(a) and Fig. 5(b) that bad entries in the routing table cause much delay, whereas Fig. 5(c) and Fig. 5(d) show that the impact of the transmission of beacons itself is not bad and only causes very little performance loss.



(a) End-to-end delay

(b) RTS Retransmissions

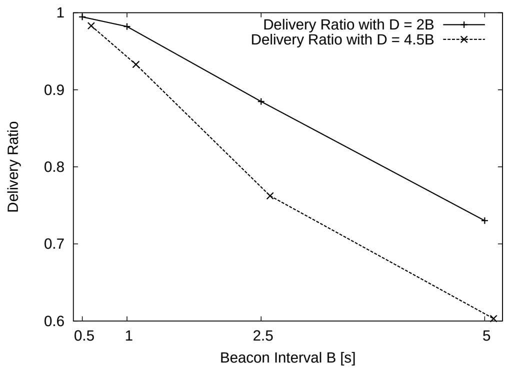
(c) End-to-end delay

(d) RTS Retransmissions

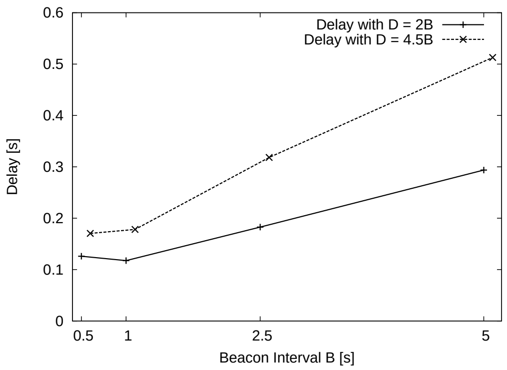
Fig. 5. Comparison of GFG/GSPR, BNU, and BL

These results also provide justification to investigate in more depth approaches to increase the accuracy of the neighbor tables of GFG/GPSR, because the performance gap to protocols without wrong forwarding decisions is significant. A number of possible approaches are described and evaluated in the following sections. In a first step, we simply study the performance of different fixed beacon and neighbor time-out intervals. Then, we improve the accuracy of neighbor information by adapting the beacon interval according to the nodes' mobility. In a third approach, nodes close to the boundary are not considered neighbors as they have the highest risk to become outdated. The forth approach adds additional information to the beacons such that nodes can estimate future positions of neighbors. In a last approach, we make beaconing reactive. A node requests its neighbors to transmit a beacon only when it has a packet to send. For reason of simplicity, each approach is considered separately, even though it is possible to use them in combination.

We try to assess the impact of different purely time-based beacon intervals B and neighbor time-out intervals D on the performance, i.e., all nodes have the same beacon and time-out intervals during the whole simulation independent of their speed, location, etc. We denote the ratio of the neighbor time-out D to the beacon interval B as k. Similar simulations were conducted in [9] where it was found that the values $B = 1.5\,s$ and $D = 6.75\,s$ are appropriate for GFG/GPSR. Many other authors also have chosen such high ratios k. In our simulations, we did not only want to study the impact of a longer or shorter beacon interval, but also the impact of the ratio k between beacon and time-out interval. The obtained simulation results for $k = 2$ and $k = 4.5$ are given in Fig. 6.



(a) Delivery Ratio



(b) End-to-end delay

Fig. 6. Time-based beaconing

The time between two consecutive beacons can be up to $2 \cdot B$ due to the 50% jitter, the time needed by IEEE 802.11 to acquire the medium, and the transmission delay. For shorter time-out intervals than $2 \cdot B$, nodes could be removed from the neighbor table between two consecutive beacons erroneously even when no beacons were missed. The results indicate that a smaller beacon interval generally increases the reliability of packet delivery at the destination. A shorter neighbor time-out interval increases the delivery ratio and at the same time also decreases the end-to-end delay. The best results are achieved with $B = 1\,s$ and $D = 2 \cdot B$. The prolongation of the beacon and the time-out intervals degrades the performance significantly. Shorter intervals however incur additional network load and power consumption by the higher frequency of broadcasted beacons. A pure time-based approach to improve neighbor table accuracy has therefore several shortcomings. On one hand, intervals may be too short and induce unnecessary transmissions if nodes are almost immobile or the network is congested. On the other hand, the interval may be too long for highly dynamic networks with fast moving nodes causing nodes to forward packets frequently to unreachable neighbors, e.g., two nodes on the highway heading in opposite directions may only be within the transmission range for a few seconds. In the following, we try to cope with these circumstances by making the beacon and time-out intervals adaptive to the movement and the speed of the nodes.

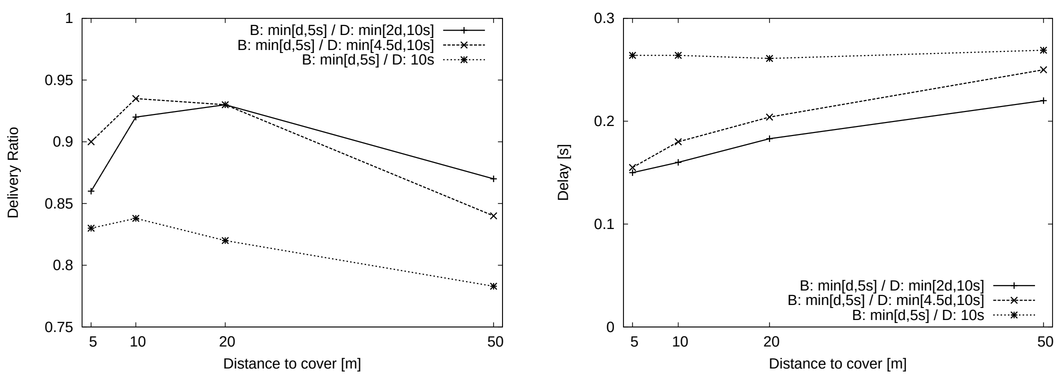
5.4 Adaptive Beacon Intervals

We evaluate two approaches based on traveled distances and nodes speeds. They improve the accuracy of the node positions in the neighbor table by adapting the interval between beacons B and as well the neighbor time-out interval D.

5.4.1 Distance

In the distance-based approach, a beacon is sent whenever a node has moved a given distance d, a "beacon distance", since its last transmission. Furthermore, we introduce two different methods to determine the neighbor time-out interval. The first one follows the same idea as the time-based approach, i.e., a simple fixed ratio between beacon and time-out interval. A node deletes an entry if it has moved more then k-times the distance d, or after a maximum time-out of $10\,s$. Consequently, the neighbor time-out interval is the minimum of $[k \cdot d, 10\,s]$. Actually, the term time-out interval is somehow misleading as it is also distance based. However, we keep the term "time-out interval" for all approaches to remain consistent. In the second approach, a neighbor is only deleted from the neighbor table after $10\,s$ independent of the distance the node moved. For the distance-based approach, nodes have to store additionally their positions each time they receive a beacon with the corresponding entry. Similar as in the time-based approach, we conducted simulations with two different values of k. The values are set again to $k = 2$ and $k = 4.5$. With the distance-based approach, we hope to map the movement of nodes to the beacon and neighbor time-out interval. Fast moving nodes send beacons frequently, whereas slow moving nodes send beacons less frequently. Problems with the distance-based approach arise if nodes move at significantly different speeds. Slow nodes only infrequently transmit beacons and a fast moving node passing by may only be within the transmission range for a few seconds. Likely, the fast node will not detect the slow moving nodes and thus perceive a reduced connectivity of the network which makes it more difficult to forward packets efficiently.

As depicted in Fig. 7(a), we have the best delivery ratios of approximately 94% for distances between $d = 10\,m$ and $d = 20\,m$. The results indicate that it is necessary to make also the time-out interval dependent on the moved distances. A pure time-based D is not able to cope efficiently with fast moving nodes and entries are deleted too late. For shorter distances, the delivery ratio decreases due to the increased network load caused by the large number of beacon transmissions. A fast node at $40\,m/s$ may transmit up to eight beacons per second. Unlike the time-based approach, a higher $k = 4.5$ does not perform much worse than $k = 2$. The reason is that fast nodes remove entries in the neighbor tables quickly even for $k = 4.5$ whereas in the time-based approach, entries are kept in the neighbor table independent of the node speed. The delivery ratio for $k = 4.5$ is even better than of $k = 2$ for short distances d because beacons collide frequently and cannot be received at the neighbors. With $k = 2$, already one not received beacon may cause the respective node to be removed from the neighbor table, which causes high fluctuation and many nodes within transmission range are temporarily not listed in the neighbor tables. Thus, nodes perceive a lower connectivity of the network than it actually is. For $k = 4.5$ up to four beacons can be missed before a node is deleted from the neighbor table, which increases the perceived connectivity. In this case, wrongly listed neighbors harm less than the removal of too many nodes within transmission range with respect to the delivery ratio. Higher k values may increase the delivery ratio, but still the delay is always shorter for smaller k values. This can be explained by the fact that the increased delivery ratio comes at the cost of numerous attempts to forward packets to unreachable neighbors. For lower k values, there are less nodes stored in the neighbor table but which have a lower probability to be unavailable than for higher k values. Thus, the delivery ratio may suffer due to the poor perceived connectivity, but if a packet is received at the destination, it is so without choosing too many unreachable neighbors. Furthermore, the end-to-end delay is only improved if the neighbor time-out interval is the minimum of the covered distance and the maximal time-out of $10\,s$ and not for a simply time-based time-out.



(a) Delivery Ratio (b) End-to-end delay

Fig. 7. Distance-based beaconing

5.4.2 Speed

In the speed-based approach, the beacon interval B and the neighbor time-out interval D are correlated to the speed a node is moving at. Each node calculates its neighbor time-out interval D again as a multiple k of the beacon interval B. Unlike before, nodes send their calculated values of D in their beacons. A receiving node then determines the time-out for this neighbor which equals the minimum of the neighbor's D as indicated in the beacon and its own D calculated from its current speed. With this enhancement, we hope to overcome the drawback from the distance-based approach where the determination of a correct time-out interval between two nodes moving at different speeds is not solved satisfactorily. The beacon interval B can be determined using either a discrete or continuous function of the nodes' speed within a predefined time range $[a, b]$. The continuous function to calculate the beacon interval B is given in (5) where v indicates the current speed of a node and v_{max} and v_{min} the maximal and minimal speed a node can move at, respectively.

$$B = a + (b - a) \cdot \left(\frac{v_{max} - v}{v_{max} - v_{min}}\right)^n \tag{5}$$

We set the range of the functions to $[1\,s, 5\,s]$ and thus have $a = 1$ and $b = 5$. We conducted simulations with three different values for n. For $n = 1$, the mapping of the speed to the beacon interval is linear and for $n = 2$ and $n = 4$ we obtain polynomial functions. Although other formulas are possible, this formula was chosen because it allows expressing in a simple way the intuitive approach that speed and beacon intervals either have a simply linear or polynomial correlation. The polynomial mapping can be used when higher speeds should yield disproportional shorter beacon intervals. The corresponding graphs are depicted in Fig. 8.

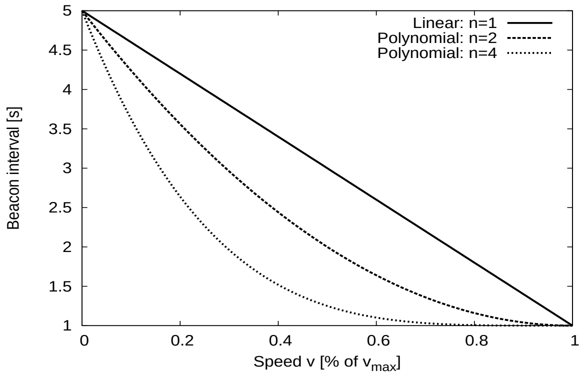


Fig. 8. Continuous functions with n=1 and n=4

The discrete function for the mapping of the speed interval $[1, 40]\,m/s$ to the beacon interval is given in Tab. 2.

Table 2
Beacon Interval at Different Speeds

Speed $[m/s]$	Beacon Interval $[s]$
1-5	5
5-10	3
10-20	2
20-40	1

The end-to-end delay and the delivery ratio for these three functions are given in Fig. 9 with $k = 2$, i.e., the time-out interval D is always $2 \cdot B$. The discrete and polynomial function with $n = 4$ perform very well whereas the linear function is only slightly better than the standard time-based approaches, cf. Fig. 6. With $n = 2$ the performance is about in between the two others as expected, since the graph is still similar to the linear function. The better performance of the discrete and polynomial function with $n = 4$ is due to the distribution of the speed of the nodes. It was shown in [44] that in the random waypoint mobility model more nodes move at lower speed than at higher. This results in an average speed of $10\,m/s$, only almost half of the arithmetic middle of the speed interval $[1\,m/s, 40\,m/s]$ in our simulations. The reason is that fast nodes arrive more quickly at their destination and then have a uniform probability of choosing a low speed. The linear function does not account for this fact. The polynomial function distributes beacon intervals over a larger range for low speeds. The discrete function was defined with the same objective in mind. The delay is reduced to around $120\,ms$ and the delivery ratio increased at the same time to 94%. These results are very promising compared to the 87% delivery ratio and $210\,ms$ of the standard GFG/GPSR in the same scenario, i.e., the delay and packet loss rate could be approximately halved.

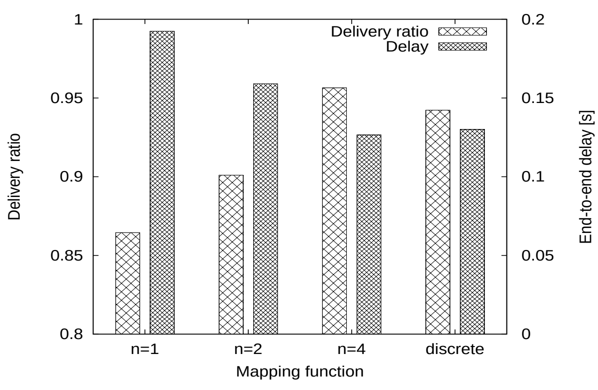


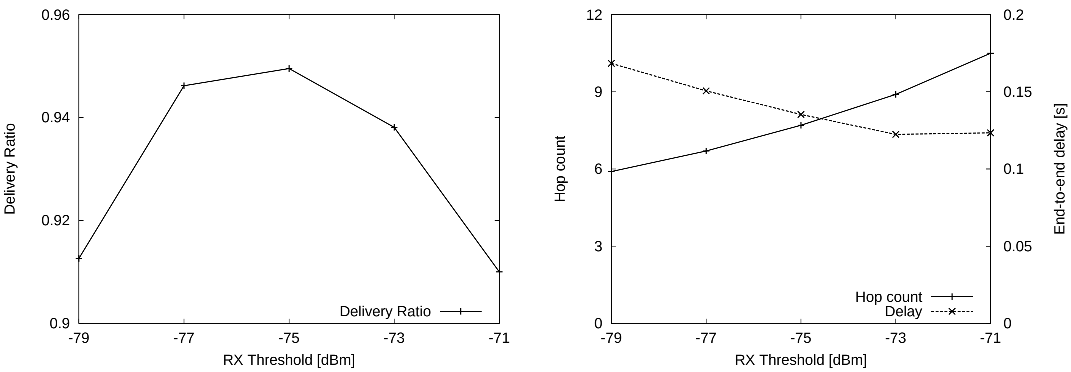
Fig. 9. Speed-based beaconing

Most position-based protocols such as GFG/GPSR forward packets to the neighbor by reducing the distance to the final destination. Thus, this neighbor is normally located close to the transmission boundary. Exactly these nodes however have the highest probability of becoming unavailable soon. By introducing a receiver (Rx) power threshold, we can create a circular gray zone at the transmission range boundary. Beacons received at a power level less than this Rx-threshold are not processed, i.e., nodes in this zone are not considered as neighbors and data packets are not forwarded to them. Unlike beacons, data packets received from these nodes are processed and forwarded as normal. In reality, transmission ranges may be highly irregular due to obstacles and interferences. The use of an Rx-threshold instead of a distance-based threshold has the advantage that it allows to cope with irregular transmission ranges.

The physical layer of IEEE 802.11 has a typically receiver sensitivity of approximately $-81\,dBm$. Together with the transmission power of $15\,dBm$, this determines the maximal transmission range of $250\,m$ in the two-ray ground reflection model. We conducted simulations with several Rx-threshold between $79\,dBm$ and $71\,dBm$. We can map this power levels to distances of 223, 199, 177, 158, 140 m when using the two-ray ground model where the signal attenuates with $\frac{1}{d^4}$ for distant nodes. Thus, only beacons from nodes closer than these distances are processed.

The delivery ratio first increases and reaches its maximum of 95% for a Rx-threshold of $75\,dBm$ as shown in Fig. 10. The difference to the delivery ratio with other Rx-threshold is only approximately 2% and not really significantly. The explanation for the shape of the graph is that for high thresholds the delivery ratio values degrades because a larger threshold reduces the effective transmission range of a node and, thus, the number of neighbors. For a too high threshold, the connectivity of the network is not guaranteed and packets start being dropped because no path exists to the destination. On the other hand, the ratio decreases for lower thresholds as more unreachable nodes are listed in the neighbor table and, thus, more incorrect forwarding decisions are taken. The hop count increases steadily from about six hops to over 10 hops because for a higher Rx-threshold the distance to the neighbors is limited. At the same time, the end-to-end delay is constantly reduced for higher thresholds. The reason is that a higher threshold reduces the wrong routing decisions and thus also the end-to-end delay, if a packet arrives at the destination. The delay first decreases and then remains rather constant as the higher hop count and the time to acquire the medium by IEEE 802.11 at each node introduce delay as well. We may conclude that wrong routing decisions have a larger impact on the delay than the actual hop count in uncongested networks.

A shortcoming of the current Rx-threshold implementation is its inability to select the most appropriate threshold. A fast moving node should only add close nodes in its neighbor table and consequently choose a large Rx-threshold depending on the node density. On the other hand, nodes close to the transmission boundary may be accepted as neighbors for slow moving nodes. Similarly as in the speed-based approach, we could map speed of nodes to Rx-thresholds to solve this problem.

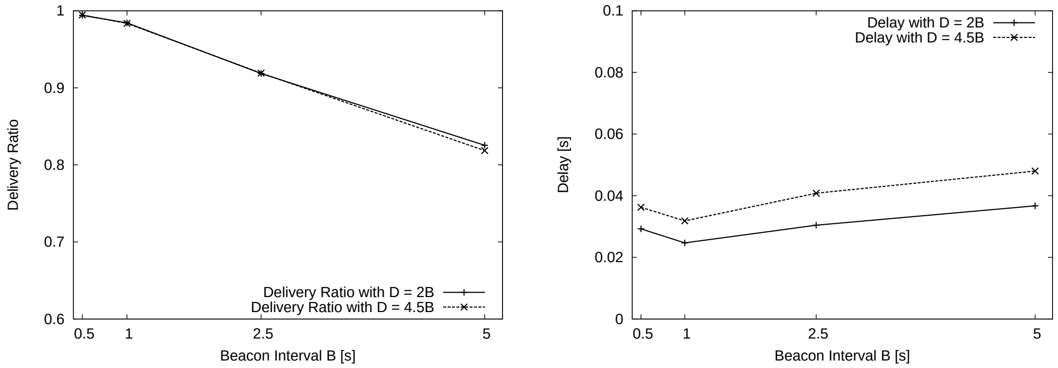


(a) Delivery Ratio (b) Hop count and delay

Fig. 10. Rx threshold for beacons

5.6 *Estimation of Link Availability*

In this approach, the velocity and direction of nodes are used to estimate the time when two nodes are no longer within each others' transmission range. In most cases, a node keeps its speed and direction during a time long enough to reliably predict its future position during the next few seconds. Consequently, each node transmits its current speed and direction in the beacons. Nodes store in their neighbor table all neighbors with their speed vectors and label the entry 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 that 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., $|B + \vec{b}t - A| > r$. The simulations were conducted with the same beacon and time-out intervals as for the pure time-based approach and also again with two k-values for the ratio between the intervals. Unlike the time-based approach, the delivery ratio and the average end-to-end delay is almost independent whether the neighbor time-out interval is 2 or 4.5 times the beacon interval as seen in Fig. 11. The reason is that the prediction of the nodes' future positions is quite accurate also for a time interval of $4.5 \cdot B$. We observe an at least five times lower end-to-end delay between $25\,ms$ and $50\,ms$ and a less steep increase for longer beacon intervals than in the pure time-based scenarios. The link availability is predicted accurately and wrong routing decisions are strongly reduced. Note that this approach again decreases the perceived connectivity because nodes in the neighbor table may be removed early. Furthermore, the delivery ratio increases significantly, e.g. for $B = 1.5\,s$ and $D = 6.75\,s$, we obtained a ratio of approximately 96% compared to 87% with the standard GFG/GPSR without prediction in Fig. 3.



(a) Delivery Ratio (b) End-to-end delay

Fig. 11. Prediction with time-based beaconing

5.7 Reactive Beaconing

As already proposed in [9], we make the beaconing mechanism of GFG/GPSR fully reactive. Only when a node has to transmit data packets, it solicits beacons from its neighbors by transmitting a beacon request packet. Each node overhearing this request replies with a beacon to announce its position. Nodes randomly jitter the transmissions of their beacons by $1\,ms$ to avoid that all nodes respond simultaneously and packets interfere at the receiver. We conducted two simulations where the requesting node waits $5\,ms$ and $10\,ms$ for incoming beacons and only then it forwards the data packet to the "best" node. This time has to be set much higher than the actual jitter of the transmissions as neighbors may have to wait some time to acquire the medium if many neighbors transmit almost simultaneously. The neighbor table is deleted and the whole process is repeated for the next packet. Nodes operate on almost accurate neighbor information as the interval between the beacon and effective packet transmission is very small.

With this reactive beaconing, we achieved a delivery ratio of 95% and an average end-to-end delay of $138\,ms$ when the requesting node waits $5\,ms$ as shown in Fig 12. The time saved through the more accurate neighbor table outweighs the additionally introduced delay of $5\,ms$ per node to acquire neighbor information. As seen before, we can expect $30\,ms$ delay per attempt to route to an unreachable neighbor. Thus, as long as there is more than one wrong routing decision in seven hops, the reactive beaconing should perform better. For a waiting time of 10 ms, the delivery ratio was even further increased to 98%, but at the same time also the end-to-end delay increased to 220 ms. The results are rather positive, 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.

Fig. 12. Reactive beaconing

6 Conclusion

In this paper, we first discussed the reasons for and the possible impact of inaccurate neighbor tables in position-based routing. We showed a strong relation between inaccurate neighbor tables and the reliability and performance of position-based routing protocols. These considerations were emphasized by a theoretical analysis that indicated that outdated entries in the neighbor tables are the rule rather than the exception in dynamic networks. Factors that amplify the inaccuracy are small transmission ranges, long beacon intervals, and high node mobility. The simulations with two optimal protocols supported the analytical results and showed that the delay can increase by more than an order of magnitude due to inaccurate neighbor tables. Furthermore, packet losses in uncongested networks are also mostly due to outdated neighbor information and wrong routing decisions. These analytical and simulation results indicate that improvements of neighbor information are required and worthwhile and that position-based routing protocols which make use of beaconing to obtain local neighborhood information have strong limitations.

In the second part of this paper, we then proposed and evaluated several optimizations, which alleviate the drawbacks of an existing position-based protocol GFG/GPSR. Already with these rather simple optimizations, we were able to achieve significant performance gain. However, the optimizations come at a certain cost and several shortcomings remain. They require either a higher frequency of beacon transmissions or a larger size of the beacons, which results in an increased utilization of network resources and additional overhead. Furthermore, some of the proposed schemes remove entries in the neighbor table very early in order to minimize the risk of selecting an unreachable neighbor which may however reduce significantly the connectivity by not using all available links. Especially in sparse networks where nodes have only few neighbors, the network may become disconnected and packets may be lost. An unsolved problem is the case of group mobility often encountered in reality, where nodes move quickly but their relative positions remain invariant. In such scenarios, no beacons may be required as the topology is almost static. The reactive approach, the approach enhanced with prediction, and the speed-based approach showed the best performance in our simulations. For delay critical applications in highly mobile networks, a combination of the prediction-based and speed-based GFG/GPSR may be a preferred choice because of the shortest delays. The reactive GFG/GPSR is more appropriate for low traffic scenarios as it eliminates the proactive broadcasting of hello messages and, thus, conserves scarce network resources. One possible application area are sensor networks with strict constraints on power-consumption and where traffic may be transmitted very rarely.

Position-based routing protocols are appropriate for many scenarios ranging from networks with high mobility to networks with very limited bandwidth such as vehicular ad-hoc networks and sensor networks. However, the results in this paper show that the impact of outdated neighbor information caused by mobility and the proactive transmission of beacons is non-negligible. Therefore, the beacon-less position-based routing protocols described in the related work in Section 2 may be a better alternative for many scenarios with frequently changing topologies or scarce network resources. We hope that the analysis of these impacts and the evaluation of some possible optimizations in this paper can give helpful insights and provide guidelines to other researchers for the future design of practical position-based routing protocols. In future work, we want to evaluate the performance of position-based routing protocols with higher traffic loads. As RTS/CTS does not protect fully from the hidden node problem, the impact on the performance and the beaconing strategies may not be negligible.

Acknowledgements

We would like to thank Prof. Bharat Bhargava from the Department of Computer Sciences at Purdue University for reviewing this paper prior to its release and for his valuable input.

References

[1] C. E. Perkins, E. 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, 1999, pp. 90–100.

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

[3] T. Clausen, P. Jacquet, P. Muhlethaler, A. Laouiti, A. Qayyum, L. Viennot, Optimized link state routing protocol, in: Proceedings of IEEE International Multi-topic Conference (INMIC '01), Lahore, Pakistan, 2001.

[4] R. Ogier, F. Templin, M. Lewis, Topology dissemination based on reverse-path forwarding (TBRPF), RFC 3684 (Feb. 2004).
URL `http://www.ietf.org/rfc/rfc3684.txt`

[5] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, R. 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, 2000, pp. 120–130.

[6] E. Kranakis, H. Singh, J. Urrutia, Compass routing on geometric networks, in: Proceedings of the 11th Canadian Conference on Computational Geometry (CCCG '99), Vancouver, Canada, 1999, pp. 51–54.

[7] M. Mauve, J. Widmer, H. Hartenstein, A survey on position-based routing in mobile ad-hoc networks, IEEE Network 15 (6) (2001) 30–39.

[8] I. Stojmenovic, Position based routing in ad hoc networks, IEEE Communications Magazine 40 (7) (2002) 128–134.

[9] B. Karp, 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, 2000, pp. 243–254.

[10] D. Turgut, S. K. Das, M. Chatterjee, Longevity of routes in mobile ad hoc networks, in: Proceedings of the IEEE semiannual Vehicular Technology Conference (VTC '01), Rhodes, Greece, 2001, pp. 2833–2837.

[11] C.-K. Toh, Associativity based routing for ad hoc mobile networks, IEEE Personal Communications Magazine 4 (2) (1997) 103–139, Special Issue on Mobile Networking and Computing Systems.

[12] R. Dube, C. D. Rais, K.-Y. Wang, S. K. Tripathi, Signal stability-based adaptive routing (SSA) for ad hoc mobile networks, IEEE Personal Communications Magazine 4 (1) (1997) 36–45.

[13] M. Gerharz, C. de Waal, M. Frank, P. Martini, Link stability in mobile wireless ad hoc networks, in: Proceedings of the 27th Annual IEEE Conference on Local Computer Networks (LCN '02), Tampa, Florida, USA, 2002, pp. 30–39.

[14] M. Gerharz, C. de Waal, P. Martini, P. 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, 2003, pp. 130–139.

[15] A. B. McDonald, T. 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, 1999, pp. 35–40.

[16] S. Jiang, D. He, J. Rao, A prediction-based link availability estimation for mobile ad hoc networks, in: Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01), Anchorage, USA, 2001, pp. 1745–1752.

[17] S. Jiang, An enhanced prediction-based link availability estimation for MANETs, IEEE Transactions on Communications 52 (2) (2004) 183–186.

[18] W. Su, S.-J. Lee, M. Gerla, Mobility prediction and routing in ad hoc wireless networks, International Journal of Network Management 11 (1) (2001) 3–30.

[19] I. Stojmenovic, M. Russell, B. Vukojevic, Depth first search and location based localized routing and QoS routing in wireless networks, Computers and Informatics 21 (2) (2002) 149–165.

[20] I. D. Chakeres, E. 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, 2002, pp. 504–508.

[21] J. Zhang, Location management in cellular networks, in: Handbook of wireless networks and mobile computing, John Wiley & Sons, Inc., 2002, pp. 27–49.

[22] S. Giordano, M. Hamdi, Mobility management: The virtual home region, Tech. Rep. SSC/1999/037, EPFL, Lausanne, Switzerland (Oct. 1999).

[23] P.-H. Hsiao, Geographical region summary service for geographical routing, ACM SIGMOBILE Mobile Computing and Communications Review 5 (4) (2001) 25–39.

[24] Z. J. Haas, B. Liang, Ad hoc mobility management with uniform quorum systems, IEEE/ACM Transactions on Networking 7 (2) (1999) 228–240.

[25] I. Stojmenovic, B. Vukojevic, A routing strategy and quorum based location update scheme for ad hoc wireless networks, Tech. Rep. TR-99-09, SITE, University of Ottawa, Ottawa, Canada (Sep. 1999).

[26] G. Karumanchi, S. Muralidharan, R. Prakash, Information dissemination in partitionable mobile ad hoc networks, in: Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems (SRDS '99), Lausanne, Switzerland, 1999, pp. 4–13.

[27] A. Agarwal, S. R. Das, Dead reckoning in mobile ad hoc networks, in: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '03), New Orleans, Louisiana, USA, 2003, pp. 1838–1843.

[28] D. Son, A. Helmy, B. Krishnamachari, The effect of mobility-induced location errors on geographic routing in mobile ad hoc and sensor networks: Analysis and improvement using mobility prediction, IEEE Transactions on Mobile Computing 3 (3) (2004) 233–245.

[29] M. Zorzi, R. R. Rao, Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Multihop performance, IEEE Transactions on Mobile Computing 2 (4) (2003) 337–348.

[30] H. Füssler, J. Widmer, M. Käsemann, M. Mauve, H. Hartenstein, Contention-based forwarding for mobile ad-hoc networks, Elsevier's Ad Hoc Networks 1 (4) (2003) 351–369.

[31] B. Blum, T. He, S. Son, J. 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).

[32] M. Heissenbüttel, T. 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, 2003, pp. 197–210.

[33] L. M. Feeney, An energy consumption model for performance analysis of routing protocols for mobile ad hoc networks, ACM/Baltzer Mobile Networks and Applications 6 (3) (2001) 239–249.

[34] L. M. Feeney, M. Nilsson, Investigating the energy consumption of a wireless network interface in an ad hoc networking environment, in: Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01), Anchorage, USA, 2001.

[35] W. Ye, J. S. Heidemann, D. Estrin, An energy-efficient mac protocol for wireless sensor networks, in: Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02), New York, USA, 2002, pp. 1567–1576.

[36] P. Bose, P. Morin, I. Stojmenovic, J. 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, 1999, pp. 48 – 55.

[37] F. Kuhn, R. Wattenhofer, A. Zollinger, Worst-case optimal and average-case efficient geometric ad-hoc routing, in: Proceedings of the 4th ACM International Symposium on Mobile and Ad Hoc Networking and Computing (MobiHoc '03), Annapolis, Maryland, USA, 2003, pp. 267 – 278.

[38] S. Krco, M. Dupcinov, Improved neighbor detection algorithm for AODV routing protocol, IEEE Communications Letters (12) (2003) 584–586.

[39] J. Y. Le Boudec, On the stationary distribution of speed and location of random waypoint, IEEE Transactions on Mobile Computing. To appear.

[40] F. Kuhn, R. Wattenhofer, Y. Zhang, A. Zollinger, Geometric ad-hoc routing: Of theory and practice, in: Proceedings of the 22nd ACM Symposium on the Principles of Distributed Computing (PODC '03), Boston, USA, 2003, pp. 63–72.

[41] R. Jain, A. Puri, R. Sengupta, Geographical routing using partial information for wireless ad hoc networks, IEEE Personal Communications Magazine 8 (1) (2001) 48–57.

[42] Qualnet (May 2005).
URL http://www.qualnet.com/

[43] A. Jayasuriya, S. Perreau, A. Dadej, S. Gordon, Hidden vs. exposed terminal problem in ad hoc networks, in: Proceedings of the Australian Telecommunication Networks and Applications Conference, Sydney, Australia, 2004.

[44] W. Navidi, T. Camp, Stationary distributions for the random waypoint mobility model, IEEE Transactions on Mobile Computing 3 (1) (2004) 99–108.

[45] S. Floyd, V. Jacobson, The synchronization of periodic routing messages, IEEE/ACM Transactions on Networking 2 (2) (1994) 122–136.

[46] J. Yoon, M. Liu, B. Noble, Random waypoint considered harmful, in: Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03), San Francisco, USA, 2003.

[47] IEEE 802.11b standard for local and metropolitan area networks.
URL http://standards.ieee.org/getieee802/802.11.html