BLR: A Beacon-Less Routing Algorithm for Mobile Ad-Hoc Networks*

Marc Heissenbüttel

Torsten Braun

Technical Report
IAM-03-001

Institute of Computer Science and Applied Mathematics (IAM)
University of Bern
Switzerland
March 2003

*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.

Contents

1 Introduction **5**

2 Related Work **6**

2.1 Definitions and Network Models 6
2.2 Progress 7
2.3 Localized Routing Algorithms 8
2.4 Position-based Routing Algorithms 10
2.4.1 Basic Algorithms 10
2.4.2 Algorithms Defeated by Unit Graph 11
2.4.3 Variations 13
2.5 Updating Strategies 13
2.6 Power Consumption 14

3 The Beacon-Less Routing Algorithm BLR **15**

3.1 Model, Notions, and Assumptions 15
3.2 Packet Header Format 16
3.3 Basic Mode 16
3.4 Unicast Packets 18
3.5 Aggregation of Paths 19
3.6 Different Forwarding Strategies 20
3.7 Nodes with Approximately the same Progress 20
3.8 Duplicated Packets 21
3.8.1 Unidirectional Links 21
3.8.2 Interferences and Hidden Nodes 22
3.8.3 Obstacles 22
3.9 Discarding of Duplicated Packets 23
3.10 Node Density and Interaction with other Protocols 24
3.11 Memory Overflow 24
3.12 Sleep Operation 25
3.13 Routing in the Vicinity of the Destination 25
3.14 Backup Mode 25
3.14.1 Request-Response Approach 25
3.14.2 Half-transmission Range Approach 26
3.14.3 Clockwise-relaying Approach 26

4 Analytical Results **28**

4.1 Distribution of the Number of Nodes in a 60° Sector 28
4.2 Expected Number of Hops before Basic Mode Fails 29
4.3 Density Function for the Progress of one Node 30

4.4 Expected Progress in a Sector 31
4.5 Approximation of the Sector by a Triangle 33
4.6 Expected Progress per Hop 34
4.7 Expected Additional Delay per Hop 35
4.8 Expected Energy spent per Hop 36

5 Simulation Results **36**
5.1 Distribution of the Number of Nodes in a 60° Sector 37
5.2 Expected Number of Hops before Basic Mode Fails 38
5.3 Expected Progress in a Sector 39
5.4 Expected Progress per Hop 40
5.5 Expected Delay per Hop 41

6 Conclusions **42**

Abstract

The minimum knowledge a node must have to make useful routing decisions in a position-based routing algorithm for mobile ad-hoc network is considered to be the position of itself, of its neighbors, and the destination. In this article we present a novel routing algorithm (called BLR: Beaconless Routing Algorithm) for mobile ad-hoc networks in which nodes are not required to have information about neighboring nodes, neither about their positions nor even about their existence. We make use of the fact that the wireless medium allows broadcast transmission, which is used to determine exactly one node for relaying a packet in a completely distributed manner. More specifically, if a node has a packet to transmit, it just broadcasts the packet. Only nodes within a certain area are potentially allowed to forward the packet. Each of these nodes introduces an additional delay depending on its position before forwarding the packet. One node eventually transmits first; the other nodes detect this subsequent forwarding of the same packet and abandon their scheduled transmission. BLR especially avoids beaconing, a broadcast mechanism to inform neighbors of a node's position, which exists to our knowledge in all other position-based routing algorithms. These mechanisms use scarce battery power, interfere with regular data transmission, and degrade the performance of the network. Analytical results and simulation experiments indicate that BLR can provide efficient and battery-conserving routing in mobile ad-hoc networks.

1 Introduction

A wireless mobile ad-hoc network consists solely of wireless host which are free to move randomly. These networks operate without the support of any fixed infrastructure and are completely self-organizing and self-configuring. Nodes are connected dynamically and in an arbitrary manner to form such a network depending on their transmission power and position. Two arbitrary nodes are in general not within each others transmission range and, thus, are not able to communicate directly. They have to rely on other nodes to forward their packets, i.e. a source sends a packet to one or more of its neighbors which in turn forward the packet to their neighbors and so on until the destination is finally reached.

Routing in such an environment is a difficult task due to the mobility of the nodes which can cause frequent and unpredictable changes in the routing topology. Furthermore, devices can suddenly be switched off, making an installed route unusable. Efficient and flexible routing protocols are required to deal with these characteristics. A lot of routing protocols, which do not make use of any location information, were designed for rather small networks with up to some hundred nodes. Most of them hardly scale with large mobile ad-hoc networks, since they either use a kind of flooding to detect routes (AODV [39], DSR [22], LANMAR [36], TORA [35], FSR [37]), try to maintain pro-actively routing information (DSDV [38], TBRPF [3], WRP [32], OLSR [21], CGSR [9]), or combine both of these approaches (ZRP [16], [31]).

The availability of small and cheap GPS-receivers is one of the key enablers for position-based routing (also called geometric or directional routing) in mobile ad-hoc networks. Every node is aware of its own position and is notified of its neighbors' positions through beacons, small packets periodically broadcasted by each node to announce its position. Additionally, a node is able to determine the location of the destination through any location management scheme (VHR [14], GLS [30], [44]). This additional information allows improving routing significantly, especially for large-scale mobile ad-hoc networks which may cover continents and consist of thousands or even millions of nodes, such as considered in the terminodes project [47].

In this article, we describe an algorithm called BLR (Beacon-Less Routing Protocol) that allows further dropping one of these assumptions, namely that nodes have to transmit beacons in order to inform neighboring nodes of their position. In this way, the periodical advertisement of each node's position by sending small beacons can be avoided. A node does neither have to have knowledge of its neighbors' positions nor even of their existence. If a node wishes to send a packet, it just broadcasts the packet and every neighboring node receives it. The protocol takes care that just one of these receiving nodes relays the packet any further. This is accomplished by introducing a small additional delay at each node depending on its position relative to the last node and the destination. The node located at the most "optimal" position introduces the fewest delay and, thus, transmit the packet at first. The other nodes detect this subsequent relaying and cancel their scheduled transmission. On one hand side, the overall delay is increased, but on the other hand side, the use of battery power can be reduced significantly at the same time as well.

The remainder of this article is organized as follows. First, an overview of existing approaches is given that make use of location information. In section 3 the routing algorithm BLR (Beacon-Less Routing) for mobile ad-hoc networks is introduced and as well some possible variations and optimizations are discussed. The basic algorithm is evaluated analytically in section 4. These results are then compared in section 5 with measurements obtained through simulations. Finally section 6 concludes this article.

2 Related Work

2.1 Definitions and Network Models

Wireless network are often modeled as a graph with the nodes as points in the plane and an edge between two nodes if and only if the Euclidean distance between them is less than a certain transmission range. (It was however argued in [11] that this might not be an accurate representation for a wireless network.) Most of the approaches assume a constant transmission range throughout the network, i.e. for any two nodes under consideration, whereas others allow nodes to adjust their transmission power up to some maximum ([1], [18], [45]). If the transmission range is scaled to 1 and is fixed throughout the network, we obtain the unit disk graph as the network model.

Furthermore we have the following definitions from graph theory. A graph is planar if it can be drawn in the plane without edges crossing. All the following introduced graphs are planar as well and satisfy some additional properties.

- A convex subdivision is a planar graph such that each face of the graph is a convex polygon, except the outer face which is the complement a convex polygon.
- In a triangulation, every face is a triangle, again except the outer face.
- The Delaunay triangulation is a triangulation in which there is a triangle of edges between three points if and only if the disk determined through them contains no other points.

- In the Gabriel Graph two points are interconnected by an edge if and only if there is no other point in the circle between these two points centered at their middle point and whose diameter is equal to the distance between them.

Since a triangle is a convex polygon as well, we can state that obviously the Delaunay triangulation is a subset of the (arbitrary) triangulation which in turn is a subset of the convex subdivision. However, faces in the Gabriel graph need not to be convex. Therefore, if a given results holds for convex subdivisions it holds as well for all triangulations, especially the Delaunay triangulation, but not necessary for a Gabriel graph. Furthermore, if a result does not hold for the Delaunay triangulation, it neither holds for the set of arbitrary triangulations, convex subdivisions, nor planar graphs. Excepted for the Gabriel graph, these graphs are not locally computable, i.e. a node is not able to determine locally with only the knowledge of the position of itself and its neighbors, if a certain edge belongs to the kind of graph under consideration.

Since in the unit disk graph model edges might cross, there are edges which are not present when one of these planar graphs is used as the network model, i.e. the unit graph belongs to neither of these set of planar graphs. These planar graphs are used however in a lot of geometric routing protocols that guarantee the delivery of packets. Furthermore, the opposite is true as well, edges from one of these planar graphs may not be present in the unit disk graph because there is no restriction on the length of the edges, which can be longer than the transmission range. Fortunately, it was shown in [5] at least that if the unit disk graph is connected then the intersection of the Gabriel Graph with the unit disk graph is connected as well. (It even contains the minimum energy path for any path loss exponent ≥ 2.)

We say that a graph defeats a routing algorithm if there exist a source and destination node such that a packet never reaches the destination when beginning at the source. Otherwise the algorithm works for this graph. In a randomized algorithm a neighboring node is chosen randomly.

2.2 Progress

The term progress was introduced in ([34], [18], [46]) and is defined as the projection of the distance traveled over the last hop from P to any node A onto the line from P to the final destination D. We speak of forward progress, if the projection of the receiving node is closer to the destination than the previous node (A', B', C' in Fig. 1), and in turn, we speak of negative progress if the distance to the destination becomes larger (E').

All the position-based routing algorithms described in the following require the network (or at least a possible path) to be stable for at least a very short time, e.g. during the transmission of a packet from the sender to the destination. Otherwise

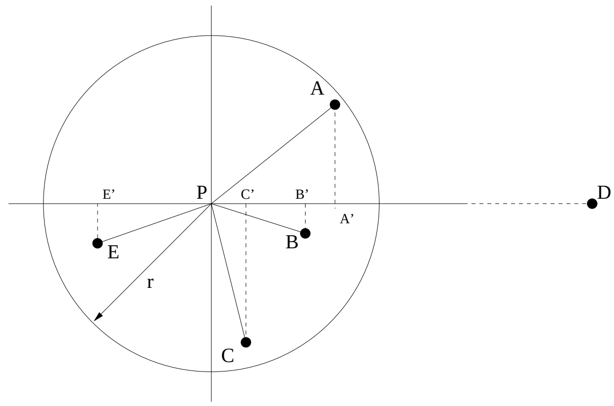


Figure 1: Forwarding strategies

it is possible that a packet may never reach its destination because the links are too short-lived. An overview and taxonomy of position-based routing algorithms can be found in [15].

2.3 Localized Routing Algorithms

Localized (or memoryless) algorithms ([7], [15], [12]) are defined as distributed algorithms that solely rely on the following three kinds of information. To our knowledge, this is assumed to be the minimum information a node must have to make useful routing decisions.

1. The position of the node itself provided, e.g., by GPS or its European counterpart Galileo that gives absolute geographical coordinates, or any other positioning service which perhaps yields relative coordinates based on signal strength [8]. Nodes are able to determine their position with accuracy in the range of some 10 meters. GPS receivers are very small, inexpensive, operate on low power and, thus, provide justification for applying position-based routing.

2. The position of the neighboring nodes which is obtained through beacons broadcasted periodically by every node. Beacons are small data packets containing information like the nodes actual position, a time stamp, a unique monotonically increasing sequence number, etc. Nodes keep track of the positions of nodes which broadcast beacons and update their neighbor data upon receiving a new beacon. If a node does not receive a beacon from a specific node within some interval, the node assumes that they are not within each others transmission range any more, whether this is due to mobility, interference caused by other transmitting nodes, power switch off or failure, etc.

3. The approximate location of the destination which must be known to the forwarding nodes. The source node of a packet is assumed to be able to determine the position of the destination node accurately enough through any location management scheme (VHR [14], GLS [30], [44]). It tags the packets with these coordinates. Thus, all nodes receiving these packets are aware of the coordinates to which the packet has to be sent as well.

A node that wants to transmit a packet selects one of its neighbors as a next hop according to some criteria like described in the next section 2.4. This process is repeated until the packet reaches the destination. Obviously, only local (optimal) routing decision can be taken with localized routing algorithms which most probably does not always result in a globally optimal routing. The only thing we can hope for is that the local decision results in a globally desired behavior of the algorithm as well.

In [15], the non-localized routing algorithms were further divided into global and zonal algorithms. Global routing algorithms are often variations of the shortest path algorithm. A node already has knowledge about the network topology or acquires it before data transmission in order to make (almost) global optimal routing decision with respect to a certain cost metric like minimum hop, minimum energy (DREAM [2], AODV [39], [41], [40]). The second class, called zonal class, consists in fact of hybrid approaches in which the network is divided into zones ([16], [31]). Within each zone a localized routing algorithm is applied and only between the zones some shortest path algorithm or any other global routing algorithm is employed. The algorithms from the zonal and the global class introduce a lot of signaling traffic which is required to maintain up-to-date routing information or require a kind of flooding to detect routes. This overhead even increases with the number of nodes in the network. Therefore, these two classes of algorithms hardly scale with large ad-hoc networks with thousand of nodes. Furthermore, the signaling traffic for maintaining pro-actively routing information consumes scarce battery power. (However, it was even observed in [48] that reactive routing protocols consume about the same amount of battery power as pro-active algorithms. This is due to the fact that just listening to the medium costs only few less than actually receiving.)

Consequently, localized algorithms are clearly preferred if they can nearly match the performance of non-localized algorithms. Even though, all of them require a kind of beaconing which not only wastes scarce battery power, but also interferes with regular data transmission. Data packets are destroyed and need to be retransmitted, consuming even more battery power, reducing the capacity of the network, and introducing additional delay.

2.4 Position-based Routing Algorithms

In LAR [24] location information was used in order to reduce flooding for finding a route to the destination. Unlike in LAR, a lot of position-based routing algorithms even do not require the establishment of any route prior to data transmission. A packet can just be sent to any intermediate node into the direction of the destination, making almost stateless routing feasible. In this way, nodes neither have to keep track of installed routes nor to store routing tables. Several of the hereafter described algorithms are localized in the sense of the previous section 2.3 or require just an additional constant amount of memory. A further advantage of position-based routing is that it naturally supports geocasting ([27], [33]).

2.4.1 Basic Algorithms

On the notion of progress, several algorithms were proposed. In [46], Most Forward within Radius (MFR) was introduced trying to minimize the number of hops by selecting the node from its neighbors with the largest progress (A in Fig. 1). Under the assumption that nodes are able to adjust their transmission power, Nearest with Forward Progress (NFP) was proposed in [18] in order to minimize the interference with other nodes and the overall power consumption. Another approach, Random Progress Method (RPM), is described in [34], where any node with forward progress is chosen with equal probability. In [45], Nearest Closer (NC), the packet is forwarded to the nearest node among the neighboring nodes, which are closer to the destination.

These basic algorithms described above are inherently loop-free, but fail in case no neighboring node has forward progress. For example in Fig. 2, there exists a path from P to D but since no node within the transmission range of P has forward progress, the packet is stucked.

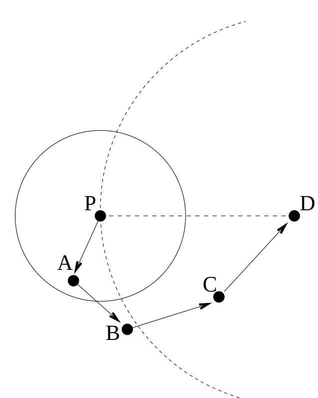


Figure 2: Failing with forward progress

2.4.2 Algorithms Defeated by Unit Graph

Other routing algorithms not making use of the notion of progress often avoid that a packet can get stucked, but do not further guarantee that no loops can occur. More exactly, they do not provide loop-freedom for unit graphs but only for certain kinds of planar graphs.

Compass Routing was introduced in [7], in which a node forwards a packet to the neighboring node minimizing the angle between itself, the previous node, and the destination (e.g. node B in Fig. 1). It was shown in [6] that this protocol does not guarantee loop-freedom for unit graphs but only on Delaunay triangulations. In Fig. 3, a situation is depicted where the graph of the network is triangulated, but since this is not a Delaunay triangulation a packet can get trapped in a loop. From P the packet is sent to the node A, which in turn forwards the packet to B and so on.

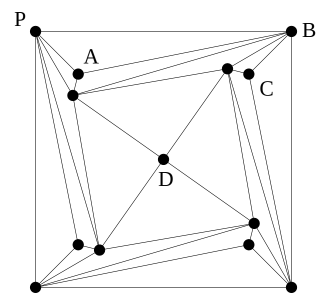


Figure 3: Failing of compass routing on triangulation

Variations, like randomized compass [7] or greedy compass [6] routing were proven to work for all convex subdivisions and all triangulations, respectively. Randomized compass routing is a memoryless randomized algorithm that uses 1 random bit per step, i.e. it chooses randomly from the two neighboring nodes minimizing the angle between itself, the previous node, and the destination in clockwise and anti-clockwise order. In greedy compass routing, the node among these two nodes is chosen that minimizes the distance to the destination. Furthermore, Bose et al. showed that there is no deterministic memoryless routing algorithm that works for all convex subdivisions in [6].

In greedy routing [7], the packet is sent to that neighboring node which is closest to the destination. With this algorithm as well a packet can loop in the network forever. For example consider the triangulation in Fig. 4, the packet is sent from P to A and back to P. (The circle does not indicate the transmission range but shows the distances of the different nodes to the destination D.) However, it was shown that greedy routing is not defeated by Delaunay triangulations. In order to detect loops, it is often proposed to store the previous nodes in the packet header in order to avoid that a packet is sent back to one of these nodes. This some times will also not help to overcome the looping-problem, since loops may consist out of an arbitrary many number of hops (as it is shown e.g. in [7] for greedy routing). Therefore an "infinite" amount of memory is required to store all previously visited nodes in the header of a packet in order to prevent loops in large networks.

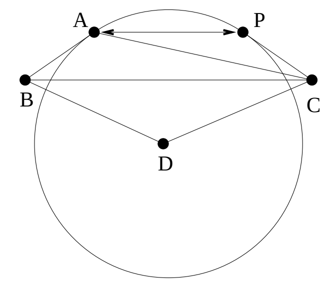


Figure 4: Failing greedy algorithm on triangulation

A very elegant approach is the concept of routing on faces of a graph and is based on the famous right-hand rule to find a way out of a maze. These algorithms is however not completely localized but needs to store at least the position of the source in the packet header, and in some optimization even other intermediate locations. In [28] this principle was described for the first time. Starting at P in Fig. 5 and determine the face F_0 incident to P intersected by the line segment $\overline{PD}$ joining P and D. Start traversing the edges (in clockwise or counter clockwise order) of this face until a second edge AB is found on the boundary of the face intersected by $\overline{SD}$.

At this point, we switch to the face F_1 of our geometric graph containing AB and start traversing again this face until another second edge intersected by $\overline{PD}$ is found. This process is repeated until the destination is reached. It was shown that this algorithm requiring O(1) memory is not defeated by any convex subdivision.

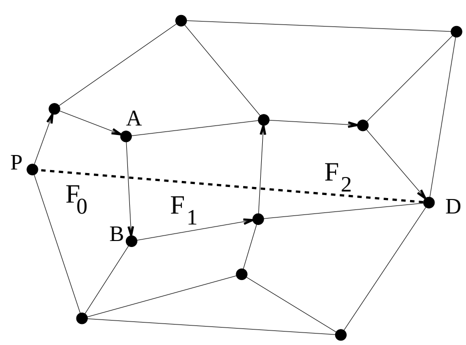


Figure 5: Routing on faces

In [5] and [25], this face routing was combined with greedy routing. Greedy routing is applied as long as there is any neighboring node closer to the destination than the current node, and face routing is only applied if the packet cannot be forwarded with greedy routing anymore. AFR was proposed in [29] which further optimizes face routing by avoiding to route beyond some radius by branching the graph within an ellipse of exponentially growing size. It was shown that AFR is asymptotically optimal.

As already mentioned, face routing would work on any convex subdivision but since these graphs are not locally computable, a subset of them is employed, namely the Gabriel graph (or sometime another planar graph known as the Relative Neighborhood graph).

2.4.3 Variations

A lot of optimizations and variations of these basic algorithms exist. For example, any intermediate node along the path with more accurate or recent information of the destinations position can update the packet header ([37], [2]). Another possibility it described in [4] where routing in the vicinity of the destination is not performed anymore with location information, but conventional pro-active or reactive algorithms are applied instead. This mechanism was introduced because it was observed that a packet can spin around the indicated position due to position inaccuracies, e.g. in case that the destination node moved to far away from its former position. It was proposed in [21] to extended the knowledge of the neighborhood to more than just the one hop neighbors, resulting in more optimal routing decision. In [4] and [17], a packet is possibly not routed directly to the destination location, but perhaps first to one or more intermediate positions. This can become necessary and improve the performance in cases where routing along a roughly straight line between the source and destination node is not possible.

2.5 Updating Strategies

A node transmits small data packets, called beacons, to inform its neighbors about its position. Beaconing is a broadcast mechanism and the strategy when a node broadcasts a beacon can have a strong impact on the number of transmitted beacons. Several of the update strategies proposed for cellular networks can be used for mobile ad-hoc networks with modifications as well (see [50] for an overview). Furthermore, additional approaches were proposed especially tailored for mobile ad hoc networks in [26]. They proposed the following four update strategies when a location updated is triggered.

- In the time-based strategy the length of the time interval between two successive location updates is exponentially distributed.

- The time and location-based strategy is very similar to the time-based strategy, solely that a node remembers its location where it triggered the last update and in case it didn't move since then it as well doesn't send an update.

- In the absolute connectivity-based strategy a node updates its position after a certain number of link incidents occurred on it, i.e. the number of link establishments plus the number of link breaks reaches a certain threshold.

- In the percentage connectivity-based strategy, an updated is triggered when a certain percentage of the total number of links of a node changed, i.e. either are newly established or broken.

The dynamic of the network is tried to be captured with the last two strategies. In a network where the node move fast and unpredictable, the number of link incidents is higher as well and thus updates are triggered more often. In [26], these update strategies where analyzed and the absolute connectivity-based strategy was found to have the best performance in mobile ad-hoc networks. But definitely more work is required in this area.

2.6 Power Consumption

In [13] and [49] several concerns of protocol designers were addressed regarding power consumption of network operations, and major sources of energy waste were identified.

- The receiving of a message causes high costs, such that if a broadcast message is received by four neighbors, the total cost of receiving the message is larger than of sending it.

- The fixed costs of sending a packet are larger compared to the incremental cost. This implies, e.g., that beaconing mechanisms with short "hello" messages incur relatively high costs.

- Receiving a packet costs generally much more than just discarding it. Discarding is a strategy that allows non-destination nodes to enter a low-energy consumption state while the media carries uninteresting traffic. The packet is marked with the length in the header and any node after having received only the header can enter a sleep mode for the duration of the transmission of the packet if it is not the intended receiver.

- Collisions and subsequent retransmission are very costly. In case a transmitted packet is corrupted and needs to be discarded, the follow-on retransmissions increase energy consumption and as well latency.

- 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.

- A further source of inefficiency is the overhead induced by control packets, since the sending of these packets consumes energy and less data packets can be transmitted as well. This actually results in less available bandwidth.

- Idle listening where a node listens to the medium to receive possible traffic that is not sent causes high costs. Especially, a lot of MAC protocols require the nodes to listen to the medium wasting a lot of energy.

Due all these observations, we may conclude that beaconing mechanisms in position-based routing algorithms use a substantial amount of battery power. First, beaconing is a broadcast mechanism and therefore a lot of nodes receive beacons, even in case this information is not needed because there are currently, or in the near future, no data transmissions through this node. But nevertheless beaconing has to be performed since a node is not able to know when this information is needed for routing. Furthermore, this periodically transmitting of beacons interferes with regular data transmission (if there is no separate signaling channel to broadcast beacons) and thus increases the number of collision and necessary retransmissions. Thus, beaconing is not only costly in terms of battery power, but reduces the capacity of the network and introduces additional delay as well.

3 The Beacon-Less Routing Algorithm BLR

3.1 Model, Notions, and Assumptions

We use the unit graph model and, thus, presume bidirectional links and omnidirectional antennas. Like in the position-based routing algorithms described in section 2, nodes are aware of their own position by means of GPS, Galileo, or any other positioning service and there is any mechanism which enables the source to detect accurately enough the destination node's position. But as opposed to other position-based routing algorithms, we do not assume that a node needs to have information about its neighboring nodes.

Furthermore, there are two system-wide parameters which are known by all the node. Max_delay indicates the maximum delay a packet can perceive per hop, and a maximal transmission radius r that can be achieved by every node in the network. The Max_delay can be in the order of 100μs (E.g. the CS/CCA (Carrier Sense / Clear Channel Assignment) of IEEE 802.11 is less than 50μs).

3.2 Packet Header Format

Each packet has header fields for storing the following information (cp. Fig. 6): The position of the destination (DN Pos.) as well as the IDs of the source (SN ID) and the destination node (DN ID). The source node marks the packets with a monotonically increasing sequence number (Seq. Nr.). In this way, duplicated or looping packets can be unambiguously detected. The previous node, i.e. last transmitting node, stores its ID (PN ID) and its position (PN Pos.) in another header field. Additionally, there is a bit to indicate if the algorithm is in basic or backup mode (BM-Bit, 0: Basic Mode, 1: Backup Mode). In the backup mode a fallback mechanism is applied, when a packet got stucked in the basic mode. There is another field to store the position where the packet entered the backup mode (BM Pos.). Finally, the packet length (Pkt. Len.) allows node to discard packets and enter a sleep mode during the transmission.

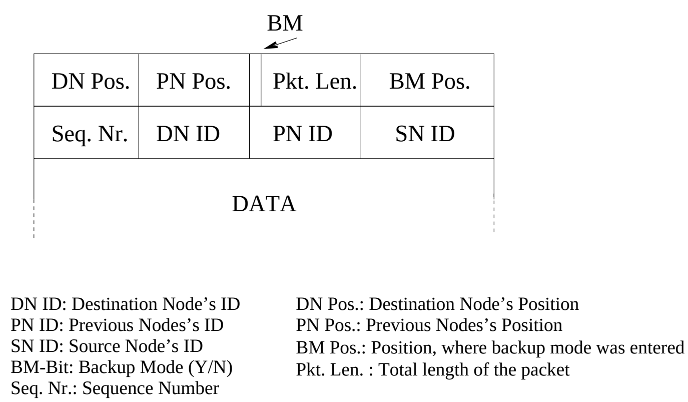


Figure 6: Packet Header Format

3.3 Basic Mode

The main feature is the absence of any beaconing mechanism. Beacons are small data packets broadcasted by a node to inform its neighbors of its position. Consequently, a node does normally not have any information about its neighbors. The algorithm takes care that an appropriate neighboring node is chosen to forward the packet. First, the source node determines the destination node's position prior to the transmission and stores these geographical coordinates in the header of the packet. From then on, basically every node, whether it is the source or any intermediate node, performs exactly the same algorithm to forward the packet. The node stores its position in the packet header and just broadcasts the packet. Several neighbors around the transmitting node receive the packet.

Upon the reception of a packet, the only available information a node has, is the position of the destination and the previous node from the packet header, and as well its own position through any location management service. Therefore, a node can easily determine if it is located within a 60° sector from the previous node towards the destination location with a radius which just equals the transmission range r. The angle of 60° results from the precondition that every node within this sector should be able to detect the transmission of any other node within the same sector for the algorithm to work properly, as described below. Only nodes within this sector are taking part in the elimination process to forward the packet (e.g. nodes A and B in Fig. 7). The other nodes just discard the packet (e.g. nodes C and E in Fig. 7).

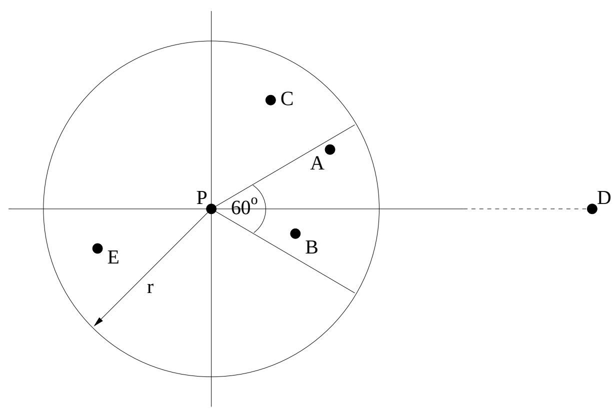


Figure 7: 60 ° Sector within transmission range

Among the nodes which are located in the 60° sector within the transmission range r, the one with the most progress is eventually chosen (i.e. node A in Fig. 7). This is accomplished through the following mechanism. Each node within this 60° sector determines its progress p towards the destination with respect to the last hop. From this value it derives a value Add_delay in the interval $[0, .., Max\_delay]$ which indicates the delay additionally introduced before relaying the packet.

$$Add\_delay = Max\_delay \, \frac{r - p}{r} \tag{1}$$

A node with less progress introduces a larger delay than a node with more progress. Consequently, the node with the most progress within this 60° sector forwards the packet at first.

Since two nodes within the same sector are located at a maximum distance of r away from each other, every node in the sector overhears the further relaying of the packet and cancels its scheduled transmission of the same packet, since obviously another node is located at a "better" position. Furthermore, we make use of passive acknowledgments (cp. [23]). The previous node P detects the further relaying of the packet as well and thus concludes that its transmission was successfully received by another node. In this way, the need to have acknowledgments on the MAC-Layer is avoided (e.g. if IEEE 802.11 was applied on the MAC-Layer, there is actually no acknowledgment provided for broadcast packets).

If a node does not detect a relaying of a packet after Max_delay, it assumes that no other node is located within the 60° sector towards the destination and enters the backup mode which provides a fallback mechanism to recover from this state (cp. section 3.14)

3.4 Unicast Packets

Since especially broadcasting incurs high cost in terms of battery power (cp. section 2.6), not all the packets are broadcasted and transmitted at full power . After a node detects through passive acknowledgment that another node forwarded the packet, it is automatically also aware of that node's position. Therefore it is not necessary to transmit subsequent packets with the same destination (or at least approximately the same destination; see section 3.5) with full power all the time. The node adjusts its transmission power and sends the subsequent packets per unicast to the node which relayed the last broadcasted packet and only every, e.g., three seconds (this parameter is called *Beacon_Threshold*), a packet is broadcasted at full power in order to detect possibly other nodes located at a better position towards this destination. The packets transmitted per unicast are forwarded immediately without introducing an additional delay. Furthermore, they are explicitly acknowledged, e.g. on the MAC-layer. In case a node is no longer able to deliver the packets per unicast because, e.g., the downstream node was switched off, it just transmits one packet again per broadcast in order to detect other nodes towards the destination. Similarly, if a node detects its downstream node to be moving out of its transmission range, a packet is transmitted at full power even before *Beacon_Threshold* in order to avoid unnecessary interruptions of links with ongoing data transmissions.

This broadcasting of packets every *Beacon_Threshold* at full power yields the positions of nodes located optimally within a 60° sector with respect to the delay function of (1). Only after transmitting a packet at full power again, i.e. after *Beacon_Threshold*, another neighboring node located at a more optimal position can be detected. However, this shortcoming also holds for "conventional" beaconing mechanisms, where a node located at a better position can only be detected after it announces its position by broadcasting a beacon. Due to these similarities with our approach which only yields the most optimal node if a packet is broadcasted at full power, we call the corresponding parameter *Beacon_Threshold* as well.

Actually, the *Beacon_Threshold* does not need to be a time interval. It can indicate as well e.g. a threshold for the traveled distance or the number of link incidents, after which a packet has to be broadcasted at full power again (cp. with the strategies described section 2.5).

As we will see in section 5, the algorithms works best if either the node density is high or the transmission range r is large. But this means as well that we encounter probably a lot of collisions, especially caused by the periodical broadcasting of packets. However, if nodes are able to transmit on different channels, this can be employed to minimize the number of collision, e.g. by broadcasting data packets on one channel and transmitting unicast packets on other channels.

3.5 Aggregation of Paths

Paths with similar destination coordinates can be aggregated in the following way. A node divides the plane into six static 60° sectors and aggregates paths that have destination coordinates within the same sector such that for all destinations located within the same sector only one packet has to be broadcasted after *Beacon_Threshold*.

For example, consider the scenario depicted in Fig. 8. Node P transmits packets to both destinations D_1 and D_2. Since the destinations are not located within its transmission range r, other intermediate nodes have to further relay the packets. Let us suppose that first a packet destined for D_1 is received by node P. P transmits the packet at full power since at that moment it is not aware of any neighboring node. Node A forwards the packet. P overhears the forwarding of A and, thus, not only gets to know through this passive acknowledgment that the packet was successfully received, but also gets to know the position of node A, since node A stored its actual position in the header. Therefore, any subsequent packet destined for D_1 within the next *Beacon_Threshold* is not broadcasted at full power but just transmitted per unicast with adjusted power level to reach node A. Not only packets for D_1, but for any other destination node located within the same 60° sector are routed per unicast over A within the next *Beacon_Threshold*. Thus, the packets for node D_2 are forwarded to node A as well, even though node B actually makes more progress with respect to destination D_2. But, it would require P again to broadcast a packet for D_2 at full power to detect node B's position. Since D_1 and D_2 are located close enough to each other, i.e. within the same 60° sector, the packets may be routed over the same next hop A. Only after *Beacon_Threshold*, node P broadcasts a packet for destinations within that sector again at full power to adapt to possible changes. In case this packet is destined for D_2, node B is detected since it forwards the packet prior to node A. Consequently, at least for the next *Beacon_Threshold* all packets to D_2 are relayed through B.

A drawback of this division of the plane into static sector is that destination nodes located close to each other are perhaps located in different sectors (e.g. D_2 and D_3 in Fig. 8). To overcome this shortcoming, a dynamic division of the plane into sectors might be used such that paths to similar destination coordinates are

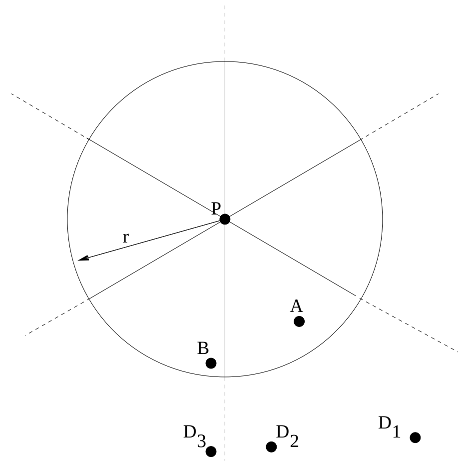


Figure 8: Division of plane into sectors for aggregation of paths

aggregated in a dynamic way.

3.6 Different Forwarding Strategies

In our basic algorithm, the node with the most progress within the 60° sector eventually forwards the packet due to the fact that it delays the packet the least. But basically all kinds of different forwarding strategies (MFR [46], NFP [18], compass [28], etc.), or any combination of them, restricted to the 60° sector can be applied, because one can argue that for the same reason as mentioned before in section 2, it is better e.g. to transmit the packets to the nearest node with forward progress in order to minimize the overall power consumption. The only thing that has to be modified to adapt the algorithm to other forwarding strategies is the delay function (1) in such a way that, e.g. the node with the least progress introduces also the least delay.

3.7 Nodes with Approximately the same Progress

It may happen that two or more nodes receive a packet and make about the same progress towards the destination. Thus, they start to relay the packets at about the same time. Unlike in typical wireless scenarios, all these nodes are able to detect each others transmissions, since they are all located within the 60° sector with radius r. As a consequence, they almost immediately detect the collision and stop their transmission. It is possible to apply ordinary CSMA/CD (Carrier Sense Multiple Access with Collision Detection) to resolve this conflict. Actually, the exponential back-off algorithm of CSMA/CD is only applied among the nodes that in fact transmitted one of the destroyed packets. The other nodes in the sector, which not yet transmitted the packet because they introduced a larger additional delay, do not try do send the packet anymore at all upon the detection of the collision.

To alleviate the problem that really only the nodes with the largest progress, i.e. with the fewest additional delay, take part in the back-off algorithm, two different strategies can be applied. If nodes are able to transmit on different channels, a node transmits a busy tone (see [19]) on a channel while broadcasting the data packet on the other channel. Nodes with a larger delay are able to detect this busy tone and abandon their scheduled transmission. In this way, only these nodes that started simultaneously to transmit at about the same time, further use a back-off algorithm to determine the node finally forwarding the packet. Similarly black bursts - pulses of energy (cp. [42]) - can be applied such that the nodes with the fewest delay first transmit the longest black burst. Again the nodes with a larger additional delay are eliminated and only these nodes with the fewest delay take part in the back-off algorithm.

3.8 Duplicated Packets

It is possible that a node does not detect the forwarding of a packet by another node. In general, we can state that for each such node within the 60° sector, including the previous transmitting node, which does not detect the forwarding, one more copy of the packet is created. In the following we briefly discuss such situations, the implications, and approaches to deal with them.

3.8.1 Unidirectional Links

Unidirectional links may occur mainly if nodes have varying transmission ranges. Until now, we only considered the situation with bidirectional links where all the nodes have exactly the same maximal transmission range. This is certainly not the case for more realistic scenarios. Suppose, we have three nodes A, B, and C with different maximal transmission ranges like depicted in Fig. 9. If A broadcasts a packet, it is received at B and C. Since they are not able to detect each others transmission, both eventually forward the packet. Furthermore, since A is not able to detect the forwarding of neither B nor C, it concludes falsely that no other node is located within the 60° sector towards the destination and switch to backup mode, resulting in an third additional packet in the network (cp. section 3.14). It is easy to see that unidirectional links in the reverse direction (i.e. from B and C to A) has no influence on the algorithm, since in this case, neither B nor C overhear packets transmitted from A. (It can happen as well that either just A does not detect the forwarding of B but C does, or that A detects the forwarding of B but not C. In this case we have two copies of the packet in the network instead of three.)

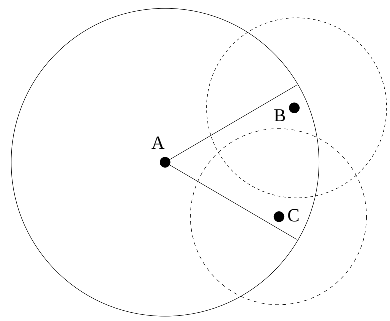


Figure 9: Duplicated packets due to varying transmission ranges

3.8.2 Interferences and Hidden Nodes

Packets may be duplicated due to the hidden node problem, or more generally if the SINR (Signal to Interferences plus Noise Ratio) is higher at one of the nodes basically resulting in a unidirectional link. Consider for example Fig. 10 where C receives successfully the packet from B since C is out of the transmission range of A. But the forwarding of node C to D may not be detected at node B due to a simultaneous transmission at node A, or any other source of noise which inhibits the correct reception at B. Like in the case with varying transmission ranges, a packet is not duplicated if the unidirectional link occurs in the opposite direction, i.e., when a transmission takes place at node C in the direction of node B, and a simultaneous transmission at node A to any other node. The signals interfere at B and thus B is not able to detect the packet transmitted from C.

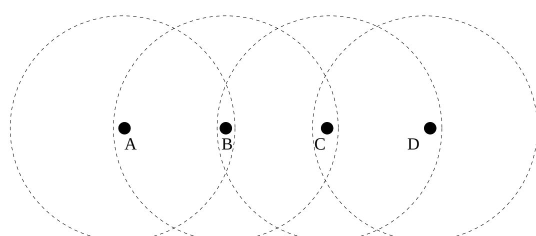


Figure 10: Duplicated packets due to interferences

3.8.3 Obstacles

In the scenario depicted in Fig. 11, all the links are bidirectional, but nevertheless we end up with a duplicated packet in the network. Node A detects the retransmission of B and C, but neither B is able to detect the forwarding of C, nor the other way around due to an obstacle in between them.

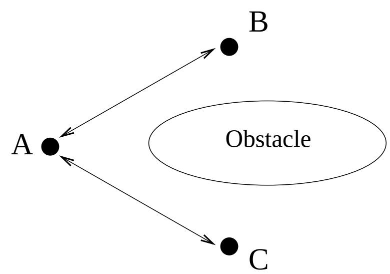


Figure 11: Duplicated packets due to obstacles

3.9 Discarding of Duplicated Packets

First, one has to keep in mind that only after *Beacon_Threshold* a packet is broadcasted for a certain destination (or area of destinations, cp. section 3.5) and, thus, possibly duplicated. All other packets for a certain destination area are transmitted per unicast between two broadcast packets and, hence, the risk of duplication is reduced greatly.

Nodes store the unique pair of source ID and sequence number of every packet they overhear. In this way, a node can unambiguously detect duplicated packets. This information is kept in the memory for only a short time. Depending on whether a broadcast or unicast packet is detected as a duplicate, different actions are taken. In the case of unicast transmissions, the receiving node just discards the packet, but nevertheless sends an acknowledgement in order to stop the previous node to further transmitting the packet. If another node than the intended receiver of the unicast transmission overhears the packet, e.g. in promiscuous mode, it notifies the intended next receiver of the duplication which abandons in turn further transmitting the packet. If the packet is broadcasted, any node which already detected the same packet before, transmits at full power immediately a short packet and notifying in this way all receiving neighbors, including the previous node, of the duplicated packet and prompt them to avoid further forwarding of the same packet. Since this notification packet is transmitted immediately, the other nodes receive this notification packet before they relayed the actual data packet because of the additional delay.

Perhaps, it is even better to completely abandon this mechanism of discarding duplicated packets. It require nodes to keep track of overheard packets what consumes battery and processing power. Furthermore, in dynamic and unstable environments, where packets are duplicated more easily, this can be exploited to increase the ratio of packets delivered at the destination.

3.10 Node Density and Interaction with other Protocols

Obviously, the BLR algorithm works best if it can operate most of the time in basic mode, i.e. if there is a rather high node density between the source and destination such that almost always at least one node is located within the 60° sector. (Something similar also holds for several of the other position-based routing protocols where greedy forwarding fails and a fallback mechanism has to be applied in case that there is no neighboring node with forward progress.) This is often not the case due to obstacles or unpopulated areas in-between. However, there exist algorithms (e.g. TRR [4], MABR [17]) which try to provide a path from the source to the destination, probably not along the line of sight but with some detours, such that the node density is always high along this determined path. The packets are not directly routed to the position of the destination node, but to (perhaps several) intermediate coordinates, and the packets are only routed between these intermediate geographical coordinates in a greedy manner. Therefore, these algorithms seem to be well-suited to combine with the algorithm described in this article.

In section 4 and 5, we will consider the impact of the node density, and as well the transmission range r, on the algorithm analytically and through simulations, respectively.

3.11 Memory Overflow

We can distinguish two different kinds of overflow. The first one deals with overflows in the packet queues. Due to the introduced additional delay, the queue is filled up with packets which are scheduled for forwarding. Unicast packets are always dropped before broadcast packets, since broadcast packets are employed to find a good path to the destination as opposed to unicast packets which are just sent to nodes already "detected" by broadcast packets. Furthermore, unicast packets need to be acknowledged explicitly. Different dropping policies can be applied for unicast packets like conventional FCFS or more sophisticated algorithms which drop the packets with, e.g., the smallest progress or the longest remaining delay. In case that broadcast packets need to be dropped, the impact should not be to large since these packets are received at several nodes anyway. Only in case the node which introduces the least additional delay has to drop the packet, another node is selected as the next hop. For all other nodes, the dropping has no influence since anyway they will detect the subsequent forwarding of the "best" node and drop the packet. Thus, the dropping policy for broadcast packets is based on the progress of a node in such a way that packet with a large introduced additional delay are dropped first, i.e. the nodes drop the broadcast packets with the fewest additional delay at last since it is more probable that they are the most optimal neighboring node for these packets.

Nodes store the unique pair of source ID and sequence number of every packet they overhear for a short time (cp. section 3.9). The second kind occurs if a node is not able to store this unique pair anymore. This kind of overflow is even less severe since it only limits the node's ability to detect duplicated packets.

3.12 Sleep Operation

Nodes which operate in sleep mode at a very low power level for energy conservation are not able to detect, receive, or transmit any packets. In case we either have a rather high node density or a large transmission range, nodes which are currently not relaying unicast traffic can be put into sleep mode. These nodes do not detect and forward broadcast packets and thus perhaps the "most" optimal path cannot be detected. However as we will see in section 4.6, the impacts are not severe since normally there are a lot of other nodes located at a almost same good position which introduce about the same additional delay.

3.13 Routing in the Vicinity of the Destination

Due to position inaccuracies, a packet may spin around the indicated destination position and never be delivered. To overcome this shortcoming, we propose to apply a reactive or pro-active protocol to route in the vicinity of the destination (as e.g. TLR [4]), if a packet cannot be delivered with geographical routing.

3.14 Backup Mode

If a node does not detect a further forwarding of its previously broadcasted packet within Max_delay, it assumes that no node is located within the 60° sector towards the destination and thus it has to take other actions to forward the packet. Several such fallback concepts are introduced to deal with these situations in the next sections. In all these approaches, the node has to set the BM-Bit and as well stores its position in the BM Pos. field in the packet header. Basically, all three approaches spent much more energy and further delay the forwarding of a packet than if transmitted in basic mode. As soon as the packet arrives at a node located closer to the destination than where it entered the backup mode, it switches back to the basic, greedy mode again.

3.14.1 Request-Response Approach

The node broadcasts a short request and all neighboring nodes respond with a packet indicating their positions. The node then forwards the packet per unicast to the node with the most forward progress towards the destination. Hence, not only nodes within a 60° sector but within a 180° sector towards the destination potentially forward the packet. If there is no node with forward progress, the actual node forwards the packet per unicast according to the right-hand rule (see e.g. AFR [29], GPSR [25]). In this Request-Response approach the overall delay is increased even more and scarce battery-power is wasted because all neighboring nodes have to transmit a beacon to announce their position. The algorithm continues in backup mode until a node is reached which is located closer to the destination than where the packet entered the backup mode (BM Pos.).

3.14.2 Half-transmission Range Approach

The node broadcasts the data packet again, and every node located closer to the transmitting node than half of the transmission range r is allowed to forward to packet. In this way every other node should be able to detect the transmission of any other node within this disk of radius $r/2$ (cp. Fig. 12). In the same way as in the basic mode, a delay is introduced per hop depending on the progress of each receiving node towards the destination. Thus, node A forwards the packet earlier. Node B detects this relaying, no matter where it is located within the disk of half the transmission range, and abandon its scheduled relaying. The advantage to the approach described in section 3.14.1 is that neighboring nodes do not have to announce their position by sending a beacon. A severe shortcoming is that a packet can get stucked completely if no node is located in the disk with radius $r/2$. Therefore, this fallback mechanism can fail as well and a approach like in section 3.14.1 has to be applied to recover.

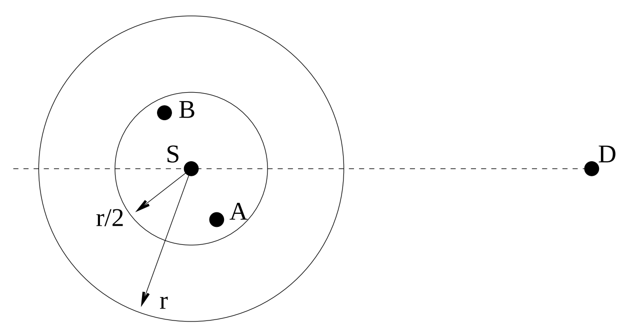


Figure 12: Area covered by half transmission range

3.14.3 Clockwise-relaying Approach

Another approach is to retransmit the packet in backup mode so that each receiving node introduces again a delay in the interval $[0, .., Max\_Delay]$, but this time solely depending on the angle α between itself, the previous node, and the destination. The corresponding delay function is given in (2).

$$Add\_delay = Max\_delay \, \frac{\alpha - 90^o}{360^o} \quad (2)$$

Thus, a node located at an angle only little less than 90° introduces almost no delay, whereas a node at 270° or at a little bit more than 90° adds $0.5 \cdot Max\_Delay$ or almost Max_Delay, respectively. With this delay function any node with forward progress relays the packet before any node with backward progress in a clockwise order. Unlike in basic mode, normally not all the neighboring nodes are able to detect a transmission of any arbitrary other neighboring node with this method. Therefore unlike in the basic mode, the previous node transmits a STN-packet (Successful Transmission Notification) as soon as it detects the forwarding of the packet through another node. The STN-packet is to inform the other neighboring nodes of the successful forwarding and prompts them to abandon their scheduled transmission.

For example consider Fig. 13 where P transmits a packet in backup mode that is successfully received at A, B, and C. Due to the new delay function in (2), node A introduces the smallest additional delay and thus relays the packet first. Node B detects this forwarding and cancels its transmission. However, node C is out of transmission range of A and is not able to detect the forwarding directly. But it receives the STN-Packet of P.

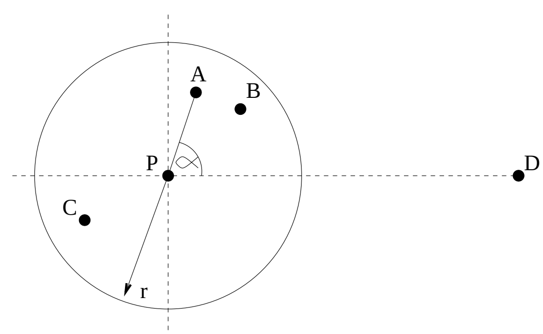


Figure 13: Clockwise retransmission in backup mode

This approach avoids any beaconing mechanism and thus conserves battery power. It actually only requires one more packet to be transmitted than in basic mode, namely the STN-packet for acknowledgement.

4 Analytical Results

4.1 Distribution of the Number of Nodes in a 60° Sector

In a first step we assume that the simulation area and the number of nodes are both bounded, where A indicates the number of square kilometers of the simulation area and N the number of nodes $(N_1, \ldots, N_N)$. Each node has a maximal transmission range of r for broadcast transmission. For unicast transmission, we use adjustable transmission ranges as described in section 3.4. The nodes are randomly distributed over the whole area. We fix an arbitrary node N_i $(1 \leq i \leq N)$ among these N nodes and a arbitrary direction between 0 and 360° (2π). Each direction is chosen with equal probability and represents the direction of the destination. The probability that there are k other nodes located in the 60° $(\pi/3)$ sector with radius r in the chosen direction is given through the following binomial distribution. X is a random variable for the number of nodes located within that 60° sector.

$$P(X = k) = \binom{N-1}{k} \left(\frac{r^2\pi}{6A}\right)^k \left(1 - \frac{r^2\pi}{6A}\right)^{N-1-k} \qquad (3)$$

This equation actually only holds, if node N_i is located at a minimum distance of r away from any border of the simulation area, i.e. the transmission area of N_i is completely located within the simulation area. But fortunately, this shows up to be irrelevant in a next step where the parameters N and A are eliminated.
Let

$$N \to \infty, A \to \infty \text{ and } \lim_{N,A\to\infty} \frac{N}{A} = n$$

N and A go to infinity, but the node density n is kept fixed (number of nodes per square kilometer). Thus, we have that

$$\lim_{A\to\infty} \frac{r^2\pi}{6A} = 0 \qquad (4)$$

and define λ as follows

$$\lambda = \lim_{N,A\to\infty} N\frac{r^2\pi}{6A} = n\frac{r^2\pi}{6} \qquad (5)$$

It is reasonable to presume that both, r and n, are in the interval $]0, \infty[$ and, thus, $0 < \lambda < \infty$ as well. We obtain from the binomial distribution in (3) a Poisson distribution for the random variable X only depending on the node density n and the transmission range r through the following steps.

$$P(X = k) \;\; = \;\; \binom{N-1}{k} \left(\frac{r^2\pi}{6A}\right)^k \left(1 - \frac{r^2\pi}{6A}\right)^{N-1-k}$$

$$= \frac{(N-1)(N-2)\dots(N-K)}{k!}\left(\frac{r^2\pi}{6A}\right)^k\left(1-\frac{r^2\pi}{6A}\right)^{N-1-k}$$

$$= \frac{N^k(1-\frac{1}{N})(1-\frac{2}{N})\dots(1-\frac{k}{N})\left(\frac{r^2\pi}{6A}\right)^k}{k!} exp\left[(N-1-k)log\left(1-\frac{r^2\pi}{6A}\right)\right]$$

We develop the last factor into a Taylor-series and obtain

$$P(X=k) = \prod_{i=1}^{k}\left(1-\frac{i}{N}\right)\frac{\left(N\frac{r^2\pi}{6A}\right)^k}{k!} exp\left[-(N-1-k)\frac{r^2\pi}{6A}+(N-1-k)\left(\frac{r^2\pi}{6A}\right)^2-\dots\right]$$

Since $N \to \infty, A \to \infty$ and together with (4) and (5), this immediately yields

$$P(X=k) = 1 \cdot \frac{\lambda^k}{k!} \cdot exp(-\lambda + 0 - \dots)$$

what is just the following Poisson distribution.

$$P(X=k) = e^{-\lambda}\frac{\lambda^k}{k!} = e^{-n\frac{r^2\pi}{6}}\frac{\left(n\frac{r^2\pi}{6}\right)^k}{k!} \tag{6}$$

(The expected value and variance of X is given by $E(X) = Var(X) = n\frac{r^2\pi}{6}$.)

From this, we easily obtain the probability p that at least one node is located within the 60° sector with radius r.

$$p = 1 - P(X=0) = 1 - e^{-n\frac{r^2\pi}{6}} \tag{7}$$

4.2 Expected Number of Hops before Basic Mode Fails

Let Y be a random variable which indicates the number of hops before the algorithms fails in basic mode, i.e. no node is located within the 60° sector. Y has a geometrical distribution with

$$P(Y=k) = (1-p)p^k$$

where k is the number of successful hops. With (7), the corresponding expected value $E(Y)$ for the number of successful hops is given by

$$E(Y) = \frac{p}{1-p} = \frac{1-P(X=0)}{P(X=0)} = \frac{1-e^{-n\frac{r^2\pi}{6}}}{e^{-n\frac{r^2\pi}{6}}}$$

(The variance of Y is $Var(Y) = \frac{p}{(1-p)^2} = \frac{1-e^{-n\frac{r^2\pi}{6}}}{e^{-n\frac{r^2\pi}{3}}}$.)

4.3 Density Function for the Progress of one Node

In section 4.1, we derived a Poisson distribution for the number of nodes located within a 60° sector with radius r. If this number is larger than 1, only the node with the most progress, i.e. with the minimum introduced additional delay, relays the packet any further (cp. section 3.3). The other nodes, which are located not so "optimal" with respect to the delay function of (1), detect this subsequent forwarding and abandon their scheduled transmission. In order to be able to calculate, e.g. the average delay per hop introduced by the algorithm, we have to look at the distribution of the progress of nodes within that 60° sector, e.g. it is much more likely that there is a node with progress $0.5 \cdot r$ than $0.1 \cdot r$.

Without loss of generality, we can assume that the transmission range is scaled to 1. Since the 60° ($\frac{\pi}{3}$) sector is symmetrical along the line in the direction of the destination, we may only consider one half of the sector as shown in Fig. 14 for the analysis of the distribution of the nodes' progress within that sector.

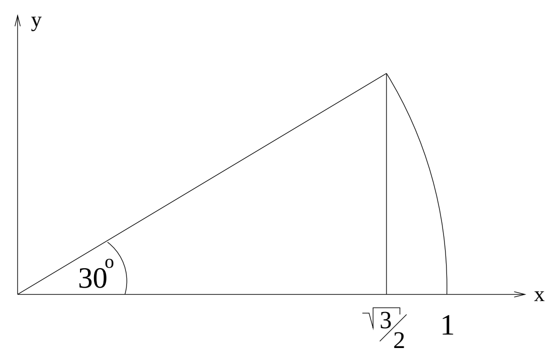


Figure 14: 30° sector scaled to a radius 1

This sector can be described analytically as follows.

$$f'(x) = \begin{cases} \frac{1}{\sqrt{3}} x & : \quad 0 \le x \le \frac{\sqrt{3}}{2} \\ \sqrt{1-x^2} & : \quad \frac{\sqrt{3}}{2} < x \le 1 \\ 0 & : \quad \text{otherwise} \end{cases} \tag{8}$$

If we normalize the area under the function f' from (8) to 1, we may consider f' as the density function f for the progress of one node located within this sector. Therefore, we first integrate the function $f'(x)$

$$A = \int_0^{\frac{\sqrt{3}}{2}} \frac{1}{\sqrt{3}} x \, dx + \int_{\frac{\sqrt{3}}{2}}^{1} \sqrt{1-x^2} \, dx = \frac{\pi}{12}$$

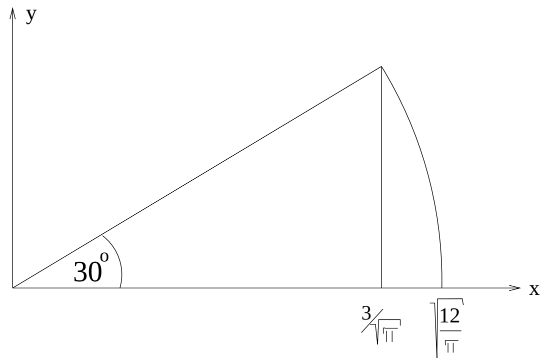


Figure 15: Density Function (”Normalized” Sector)

and afterwards have to stretch the x-axis by $\sqrt{\frac{1}{A}} = \sqrt{\frac{12}{\pi}}$ as shown in Fig. 15. From this, we immediately obtain the density function f for a node within a ”normalized” sector.

$$f(x) = \begin{cases} \frac{1}{\sqrt{3}}\,x & : \quad 0 \leq x \leq \frac{3}{\sqrt{\pi}} \\ \sqrt{\frac{12}{\pi} - x^2} & : \quad \frac{3}{\sqrt{\pi}} < x \leq \sqrt{\frac{12}{\pi}} \\ 0 & : \quad \text{otherwise} \end{cases} \tag{9}$$

4.4 Expected Progress in a Sector

Since always only the node with the most progress is relaying the packet and since all the nodes are randomly and independently distributed, we are interested in the distribution of the maximum function of independent and identically distributed (referred to as i.i.d.) random variables X_i $(i \leq n)$, where the density function of each X_i is given by (9).

The distribution function of one X_i $(i \leq n)$, we easily obtain from (9) by integration

$$F_{X_i}(t) = \begin{cases} 0 & : \quad t < 0 \\ \int_0^t \frac{1}{\sqrt{3}}\,x\,dx & : \quad 0 \leq t \leq \frac{3}{\sqrt{\pi}} \\ \int_0^{\frac{3}{\sqrt{\pi}}} \frac{1}{\sqrt{3}}\,x\,dx + \int_{\frac{3}{\sqrt{\pi}}}^t \sqrt{\frac{12}{\pi} - x^2}\,dx & : \quad \frac{3}{\sqrt{\pi}} < t \leq \sqrt{\frac{12}{\pi}} \\ 1 & : \quad t > \sqrt{\frac{12}{\pi}} \end{cases}$$

and this yields for F_{X_i} $(i \leq n)$

$$F_{X_i}(t) = \begin{cases} 0 & : \quad t < 0 \\ \frac{1}{2\sqrt{3}}t^2 & : \quad 0 \leq t \leq \frac{3}{\sqrt{\pi}} \\ \frac{t}{2}\sqrt{\frac{12}{\pi} - t^2} + \frac{6}{\pi} arcsin\left(\sqrt{\frac{\pi}{12}}t\right) - 2 & : \quad \frac{3}{\sqrt{\pi}} < t \leq \sqrt{\frac{12}{\pi}} \\ 1 & : \quad t > \sqrt{\frac{12}{\pi}} \end{cases} \tag{10}$$

The distribution of the maximum of the i.i.d. random variables X_i with $(i \leq n)$ can be calculated as follows.

$$\begin{aligned} F_{max_{i \leq n} X_i}(t) &= P(max_{i \leq n} X_i \leq t) \\ &= P(X_i \leq t, \forall i \leq n) \\ &= P(X_1 \leq t, \ldots, X_n \leq t) \\ &= [P(X_1 \leq t)]^n \\ &= [F_{X_1}(t)]^n \end{aligned} \tag{11}$$

It is well known that if for any given random variable Z and its distribution function F_Z, the expected value $E(Z)$ can be calculated in the following way.

$$E(Z) = \int_0^\infty (1 - F_Z(x))\, dx - \int_{-\infty}^0 F_Z(x)\, dx$$

Together with (10) and (11) we obtain

$$\begin{aligned} E(max_{i \leq n} X_i) &= \int_0^\infty \left[1 - (F_{X_1}(x))^n\right] dx \\ &= \int_0^{\sqrt{\frac{12}{\pi}}} \left[1 - (F_{X_1}(x))^n\right] dx \\ &= \sqrt{\frac{12}{\pi}} - \int_0^{\frac{3}{\sqrt{\pi}}} \left(\frac{1}{2\sqrt{3}}x^2\right)^n dx - \\ &\quad \int_{\frac{3}{\sqrt{\pi}}}^{\sqrt{\frac{12}{\pi}}} \left(\frac{x}{2}\sqrt{\frac{12}{\pi} - x^2} + \frac{6}{\pi} arcsin\left(\sqrt{\frac{\pi}{12}}x\right) - 2\right)^n dx \end{aligned} \tag{12}$$

Unfortunately, we are able to integrate analytically this function only for $n = 1$, which yields $E(X_1) = 4\sqrt{\frac{2}{\pi^3}}$. As expected this is just the x-coordinate of the center of gravity of the density function $f(x)$. The ratio of this value to the radius $\sqrt{\frac{12}{\pi}}$ of the "normalized" sector is $\frac{2}{\pi} \simeq 0.6367$, i.e. the center of gravity is located at about 64% of the transmission radius. Note that this is the expected progress if exactly one node is located within the sector. Actually, this holds for any 60° sector since it is independent of the transmission range (cp. 12). In order to obtain an analytical function for all n, we approximate in the next section the sector through a triangle with the same area.

4.5 Approximation of the Sector by a Triangle

In a first step, we consider the distribution of a node in an arbitrary triangle, and only then derive the values for the triangle with an angle of 30°. In the same way as in section 4.3, we scale the area of the triangle to 1 in order to be able to consider the function describing the triangle as the density function of a random variable X for the progress of a node within the triangle.

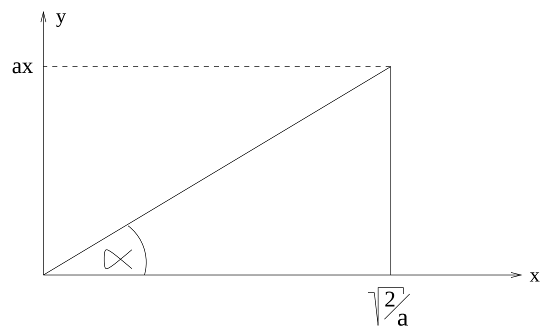


Figure 16: Sector of the Transmission Area

$$f(x) = \begin{cases} ax & : \quad 0 \leq x \leq \sqrt{\frac{2}{a}} \\ 0 & : \quad \text{otherwise} \end{cases}$$

From that we can easily derive the distribution function F_X of X

$$F_X(t) = \int_0^t ax\,dx = \frac{a}{2}t^2 \qquad \text{with } 0 \leq t \leq \sqrt{\frac{2}{a}}$$

and this yields completely analogously as in (12) the expected progress for a given number of nodes within the triangle.

$$\begin{aligned} E(max_{i \leq n} X_i) &= \int_0^{\sqrt{\frac{2}{a}}} [1 - (F_X(t))^n]\,dt \\ &= \int_0^{\sqrt{\frac{2}{a}}} \left[1 - \left(\frac{a}{2}t^2\right)^n\right] dt \\ &= \sqrt{\frac{2}{a}}\,\frac{2n}{2n+1} \end{aligned} \qquad (13)$$

In our case with $\alpha = 30°$, we have $a = \frac{1}{\sqrt{3}}$ and obtain

$$E(max_{i \leq n} X_i) = \sqrt[4]{12}\ \frac{2n}{2n+1}$$

We divide this value by the radius $\sqrt{\frac{12}{\pi}}$ of the "normalized" sector in order to obtain the value for a radius of 1 of the approximated sector.

$$E(max_{i \leq n} X_i) = \sqrt{\frac{\pi}{\sqrt{12}}}\ \frac{2n}{2n+1} \tag{14}$$

The reason that the expected progress does not approach asymptotically the normalized transmission range 1 for $n \to \infty$, but only the constant $\sqrt{\frac{\pi}{\sqrt{12}}} < 1$ stems from the fact that we approximated the 30° sector with a triangle of the same area and same angle as shown in Fig. 17. As we can see, the "length" of the triangle on the x-axis is slighly smaller than of the sector (exactly $\sqrt{\frac{\pi}{\sqrt{12}}}$ times smaller). Due to these reasons, the expected approximated value of (14) is slightly smaller than it actually is, as derived in (12).

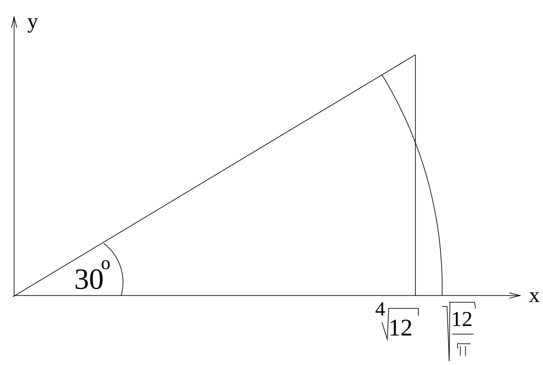


Figure 17: Sector and its approximation by a triangle

The distribution function of the maximum of i.i.d. random variables X_i with a triangle as the density function for the X_i always yields $\frac{2n}{2n+1}$ for any angle, if the side on the x-axis is of length 1. This can be seen in e.g. in (13) where the constant $\sqrt{\frac{2}{a}}$ is only used to scale to a radius of 1.

4.6 Expected Progress per Hop

In the previous section we just considered the expected progress for a given number of nodes located within the sector. We did not take into account the actual distribution of the number of nodes within this sector (6). To obtain the actual expected progress per hop for the sector with radius 1 (by using its approximation through a triangle), we can derive the following function from (6) and (14) for the expected progress EP depending on the transmission range r and node density n.

$$\begin{aligned} EP(n,r) &= \sum_{k=1}^{\infty} e^{-n\frac{r^2\pi}{6}} \frac{\left(n\frac{r^2\pi}{6}\right)^k}{k!} \sqrt{\frac{\pi}{\sqrt{12}}} \frac{2k}{2k+1} \\ &= \sqrt{\frac{\pi}{\sqrt{12}}} e^{-n\frac{r^2\pi}{6}} \sum_{k=1}^{\infty} \frac{\left(n\frac{r^2\pi}{6}\right)^k}{k!} \frac{2k}{2k+1} \end{aligned} \qquad (15)$$

If we want to know the expected progress under the condition that there is at least one node located within that 60° sector, we just divide EP by $1-e^{-n\frac{r^2\pi}{6}}$, which is the probability that at least one node is within the sector (cp. (7)), and conclude

$$EP' = \sqrt{\frac{\pi}{\sqrt{12}}} \frac{e^{-n\frac{r^2\pi}{6}}}{1-e^{-n\frac{r^2\pi}{6}}} \sum_{k=1}^{\infty} \frac{\left(n\frac{r^2\pi}{6}\right)^k}{k!} \frac{2k}{2k+1} \qquad (16)$$

Actually, EP' is of more relevance than EP because if there is no node located within the sector, the basic method fails anyway and a fallback mechanism has to be applied (cp. section 3.14).

4.7 Expected Additional Delay per Hop

Let us assume that the delay a node adds per hop is as described in (1) (Even though, other delay functions might be used as well, cp. section 3.6). The expected value for a random variable Y indicating the delay per hop can be derived analogously as in (14).

$$E(Y) = \sqrt{\frac{\pi}{\sqrt{12}}} \frac{1}{2n+1} + \left(1 - \sqrt{\frac{\pi}{\sqrt{12}}}\right)$$

The constant $1 - \sqrt{\frac{\pi}{\sqrt{12}}} \simeq 0.048$ is due to the way we approximated the sector by a triangle. Since a node can in our approximation never have a progress larger than $\sqrt[4]{12}$, every node introduces at least a small delay of $\sqrt{\frac{12}{\pi}} - \sqrt[4]{12}$ what is just the difference between the length of the sector and the triangle on the x-axis (cp. Fig. 17). This difference, if normalized to a radius of 1, i.e. if divided by $\sqrt{\frac{\pi}{12}}$, just yields that constant. Therefore we obtain for a normalized radius of 1 completely analogous as in (15) and (16) the following functions for the expected delay ED

introduced per hop

$$ED(n,r) = \left(1 - \sqrt{\frac{\pi}{\sqrt{12}}}\right) + \sqrt{\frac{\pi}{\sqrt{12}}}\; e^{-n\frac{r^2\pi}{6}} \sum_{k=1}^{\infty} \frac{\left(n\frac{r^2\pi}{6}\right)^k}{k!} \frac{1}{2k+1}$$

and under the precondition that at least one node is in the sector, respectively.

$$ED'(n,r) = \left(1 - \sqrt{\frac{\pi}{\sqrt{12}}}\right) + \sqrt{\frac{\pi}{\sqrt{12}}}\, \frac{e^{-n\frac{r^2\pi}{6}}}{1 - e^{-n\frac{r^2\pi}{6}}} \sum_{k=1}^{\infty} \frac{\left(n\frac{r^2\pi}{6}\right)^k}{k!} \frac{1}{2k+1} \qquad (17)$$

4.8 Expected Energy spent per Hop

We assume a path loss factor of b, i.e. the energy that has to be spent to reach a node at distance d is proportional to d^b. After *Beacon_Interval* a node has to transmit at full power, therefore wasting some battery-power in comparison to other position-based routing algorithms which are aware of the position of all neighboring nodes and, thus, can adjust their transmission power to an optimal level all the time. But this is a small price for the advantage that we can completely abandon any beaconing mechanism, especially because we will see that the wasted energy is really few. In order to calculate the waste of energy due to this ignorance, we first have to derive the expected value of the energy E_S that is spent in case a node knows the distance to the neighboring nodes with the most progress in the 60° sector.

$$E_S = e^{-\lambda} \sum_{k=1}^{\infty} \frac{\lambda^k}{k!} \left(\sqrt{\frac{\pi}{\sqrt{12}}}\, \frac{2n}{2n+1}\right)^b \text{ with } \lambda = n\frac{r^2\pi}{6}$$

Since the radius is already normalized to 1 in this formula and thus the energy spent if a packet is broadcasted at full power equals 1 as well, E_S indicates the ratio between the energy spent in the optimal case to the case in which a node transmits at full power. We will see in section 5 that the energy wasted can be kept reasonably low, especially if we consider the mechanism described in section 3.4, which requires to send only every *Beacon_Intervall* a packet at full transmission power.

5 Simulation Results

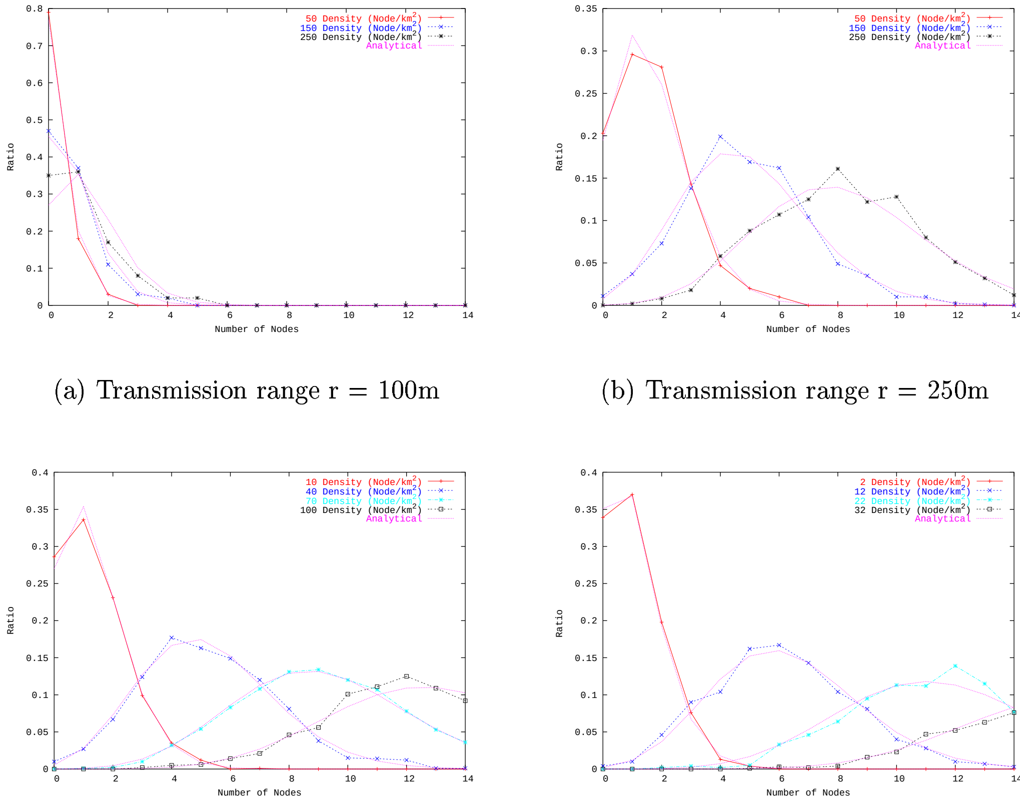
In this section some results obtained through a very basic simulator are presented, which are compared to the results obtained analytically in section 4. Basically, nodes are modeled just as points in the plane with no functionality. Our simulator does not take into account any layers like physical-, MAC-, network-, transport-, or application-layers. In order to simulate an unbounded simulation area, only nodes located at a distance larger than the transmission range r away from any border of the area are considered for transmissions (cp. section 4.1). In all the figures, the corresponding analytical function is drawn as well.

Rather high node densities are considered in our simulations as well, which perhaps seem to be too high for the overall node density in a network, or only might occur in certain areas and along appropriate paths in the network such as cities and roads, respectively. However, e.g. TRR [4] provides paths with high node densities to enhance routing and can be combined with the BLR algorithm which should enable BLR to operate most of the time in basic mode, i.e. that there is (almost) always a node in the 60° sector towards the destination.

5.1 Distribution of the Number of Nodes in a 60° Sector

For a randomly chosen node (i.e. any node which transmission area is completely located with the simulation area) and direction, we count the number of other nodes located within a 60° sector depending on the radius r and node density n in every simulation run. The number of nodes (on the x-axis) is drawn versus their frequency of occurrence on the y-axis. Please notice that the axes have a different scale in each of the figures. The four graphs in Fig. 18 show the results for four different transmission ranges, whereas in each figure the distribution for different node densities is considered. The calculated Poisson distribution from (6) for the corresponding values of r and n is drawn in the same figures as well. What immediately is apparent is that already with few simulation runs (100 - 1000), the values obtained through simulations fit very nicely with the analytical ones. We can expect to have several nodes in the corresponding sectors already with a rather low density if the transmission range is between 250 and 1000 meter as shown in Figures 18(b), 18(c), 18(d). Just the opposite applies for Fig. 18(a) with a rather small transmission range of 100 meter. The maximum of the distributions is around 0, i.e. most frequently no node is located within the sector.

Since our algorithm does not have to apply any fall-back mechanism as along as there is at least one node within the 60° sector towards the destination, the probability for this case is of special importance and is shown separately in Fig. 19. With a transmission range $r = 100m$ in Fig. 19(a), we observe that even with a rather high node density the risk is still high that no node is within the sector. However, this transmission range is perhaps more typical for indoor environments. These are often denser populated and therefore the density of the nodes is probably higher as well. Already with a transmission range of 250 meters (e.g. a typical value as considered for IEEE 802.11b), we obtain quite low ratios that no nodes is within the sector. If we increase further the transmission radius to 500 and 1000 meters in Fig. 19(b), the ratios are negligible for already very low node densities. (The x-axes



(a) Transmission range r = 100m

(b) Transmission range r = 250m

(c) Transmission range r = 500m

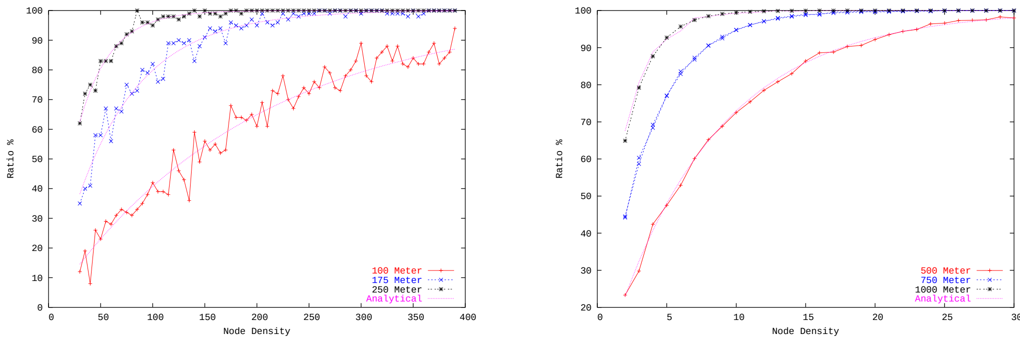
(d) Transmission range r = 1000m

Figure 18: Node Distribution

have different scales in the two figures.)

5.2 Expected Number of Hops before Basic Mode Fails

Actually, no simulation results are presented in this section. The graphs are obtained from the analytical derivation in section 4.2. We are confident however, that these results are adequate because of the exact match of the simulation and analytical results in the previous section 5.1. In Fig. 20 the expected value of the number of hops before there is no node located in the 60° sector anymore is shown on a logarithmic y-axis depending on the node density n and transmission radius r. Again, we can state that for a low transmission range $r = 100m$, we can not expect the algorithm to work in the basic mode for a large number of hops, since even for high node densities the number of successful hops is relatively low before



(a) r = 100m - 250m

(b) r = 500m - 1000m

Figure 19: Probability for at least one node in sector

the fallback mechanism has to be applied. Completely the opposite holds in the case with $r = 1000\,m$, the number of successful hops increases very strongly with only a minor increase of the node density.

Somehow, this result is similar to the observation made in [10], where the connectivity of a wireless network depending on the node density was considered. The network stays disconnected for node densities below a certain threshold and almost gets completely connected for values over that threshold.

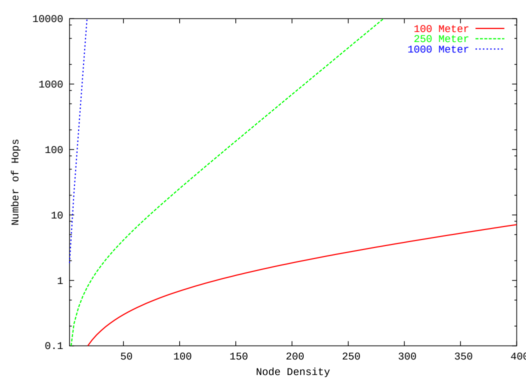


Figure 20: Number of Hops before failure

5.3 Expected Progress in a Sector

In this section, simulation results for the progress per hop are presented and compared with the analytical result (14) from section 4.5. We choose a transmission range r of 250 meters and a node density of 100 nodes per square kilometer. (Even though, the results are primarily independent of the parameters for the experiment.) Furthermore, the values are averaged over 10000 simulation runs and the expected progress is normalized to a radius of 1. In Fig. 21, the expected progress is shown

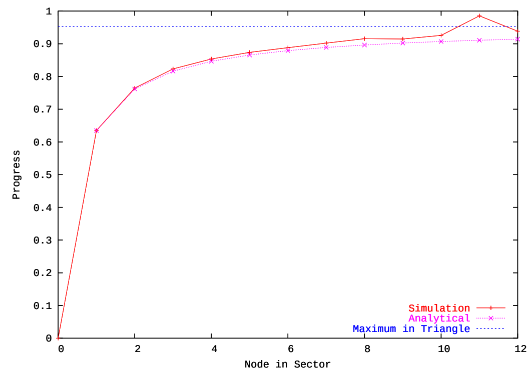
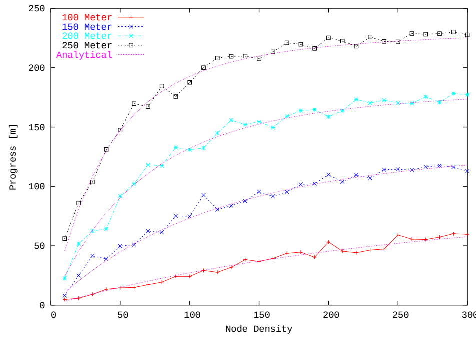


Figure 21: Expected progress for a given number of nodes within the sector

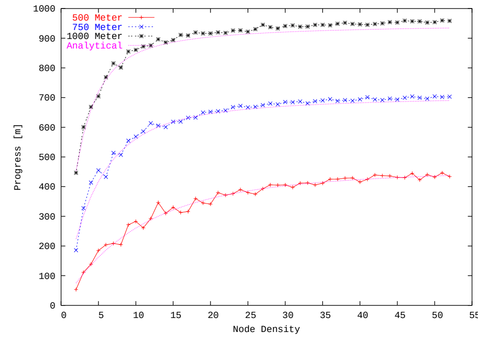
depending on the number of nodes within the triangle. As expected, the values for the simulation are slightly higher than for the analytical curve due to the way we approximated the sector with a triangle since the maximum progress in the triangle is smaller than in the sector (cp. section 4.5). The dotted horizontal line indicates the maximal achievable progress in the triangle $\sqrt{\frac{\pi}{\sqrt{12}}}$. The values of the simulation can actually be higher since packets can be forwarded to nodes within the sector and not the triangle. (cp. the peak at a node density of 11).

5.4 Expected Progress per Hop

In the previous section, we considered the progress per hop for a given number of nodes located within the sector, but we did not take into account the node density and, thus, the Poisson distribution of the number of nodes within the 60° sector. In Fig. 22 the expected progress for a certain node density n and transmission range r is depicted (cp. section 4.6). The x-axes of the two figures have again a different scale. The values are averaged over 1000 Simulation runs for every pair of n and r. Simulation runs with no node within the sector, i.e. with 0 progress, are included (cp. (15)). For a transmission range over 200 meters, the expected progress per hop increases steeply at the beginning. This is in accordance with (18) where we observed that the probability for no node within the 60° sector is negligible already for quite low node densities. Therefore, the progress per hop approaches quickly the maximal transmission range. By the same reason, the additional delay introduced per hop according to (1) is very small, what we will consider in more detail in the



(a) r = 100m - 250m



(b) r = 500m - 1000m

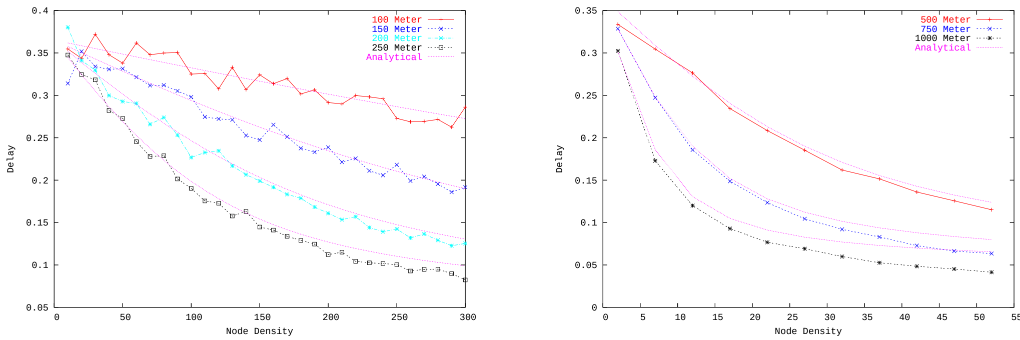
Figure 22: Expected progress per hop

next section. The simulated values are slightly higher than the ones obtained by analytical means; again, this is because we approximated the sector by a triangle.

If a node operates in sleep mode to conserve scarce battery power, it is not able to detect, receive, or transmit any packets. In case we consider e.g. a transmission range of 1000 meters, we might observe that the expected progress does not increase much for node densities higher than 10 nodes per square kilometer. E.g. this means that with a node density of $n = 100$, 90% of the nodes can be in sleep mode and the performance of the network is still almost the same, i.e. normally a node at almost the most optimal position will be chosen as a next hop anyway. Or in other words a node can operate in sleep mode at very low power level for about 90% of the time. This percentage even increases for higher node densities.

5.5 Expected Delay per Hop

We plot the ratio of encountered delay per hop to the maximum delay Max_delay (cp. section 3.1) versus the node density in the network. Basically, the plotted graphs are just the inverse of Fig. 22, due to the way the progress and delay are correlated (cp. (1)). Therefore, this time the simulation results are slightly lower than the analytical ones and we have a sharp decrease of the delay at the beginning for transmission ranges r over 200 meters. For $r = 250\,m$ already with a node density of 100, the additional delay is less than 10% of the maximum delay (e.g. less than 20 μs per hop with CS/CCA). For larger transmission ranges, we obtain even shorter delays for much lower node density. In the case that $r =$ 1000m, less than 10% of the maximum delay are introduced per hop if the density is higher than 20 nodes per square kilometer. Unlike in the previous section, we took only simulations into account which had at least one node in the sector (cp. 17).



(a) r = 100m - 250m

(b) r = 500m - 1000m

Figure 23: Expected delay per hop

6 Conclusions

In this article a novel position-based routing algorithm BLR for mobile ad-hoc networks is described which avoids any beaconing mechanisms. Since beaconing mechanisms are costly and nodes are battery-powered, it is especially important to avoid any unnecessary transmission, reception, and processing of packets. In other position-based routing protocols the location information disseminated through beacons is often not even used because there is no data transmission. BLR only requires nodes to be aware of their own position. The additional information used for routing is obtained upon the reception of a data packet, namely the position of the destination and the upstream node which are stored in the packet header. Solely this location information is used at each node in order to determine in a completely distributed way a node which forwards the packet further by introducing a small additional delay depending on the relative position to the destination location and previous node.

The behavior of the BLR algorithm in basic mode was evaluated by analytical means and the obtained results were verified through simulations. Rather high node densities are required for small transmission ranges. However for transmission ranges $> 250\,\text{m}$ the algorithm is able to operate most of the time in basic mode. Especially if BLR operates in basic mode, it is able to save a substantial amount of energy compared to other position-based routing algorithms that require beaconing. Furthermore, it was shown that the additional delay introduced per hop through the BLR algorithm can be kept small. The results indicate as well that the energy wasted because a node has to transmit some packets at full power, and not at the optimal power level due to the lack of position information about its neighbors, is negligible even for sparse networks. Therefore, BLR is able to provide efficient and battery-conserving routing in mobile ad-hoc networks. The algorithm continues to work if it fails in basic mode, but additional delay is introduced and some transmissions are evoked in backup mode to recover from this failure. However, there exist routing algorithms which provide paths with high node density (e.g. TRR [4]) along roughly straight lines between intermediate coordinates. These approaches can be combined with the algorithm presented in this article. We believe that these first considerations and results provide justification for further pursue this approach.

In future work, we intend to study the impact of beaconing on position-based routing algorithms in order to be able to quantify e.g. the energy wasted or the performance degradation of the network through these beaconing mechanisms. Furthermore, we will implement the whole algorithm in a more realistic network simulator to obtain more representative results about the behavior and the performance of BLR.

References

[1] Barriere, L., et al.: Robust Position Based Routing in Wireless Ad hoc networks with unstable transmission Ranges, Proceedings of the 5^{th} ACM International Workshop on Discrete Algorithms Methods for Mobile Computing and Communication, 2001

[2] Basagni, S., Chlamtac, I., Syrotiuk, V.R., Woodward, B.A.: A Distance routing effect algorithm for mobility (DREAM), Proceedings of MOBICOM, 1998, 76-84

[3] Bellur, B., Ogier, R.G.: A Reliable, Efficient Topology Broadcast Protocol for Dynamic Networks, Proceedings IEEE INFOCOM '99, New York, USA, March 1999

[4] Blazevic, L., Giordano, S., Le Boudec, J.-Y.: Self Organized Terminode Routing, Cluster Computer Journal, Vol. 5, No. 2, April 2002

[5] Bose, P., Morin, P., Stojmenovic, I., Urrutia, J.: Routing with guaranteed delivery in ad hoc networks, 3^{rd} International Workshop on Discrete Algorithms and methods for mobile computing and communications (DIAL M), August 1999

[6] Bose, P., et al.: Online routing in convex subdivisions, Internal Symposium on Algorithms and Computation (ISAAC), 2000, 47-59

[7] Bose, P., Morin, P.: Online routing in triangulations, In Proceedings of the tenth International Symposium on Algorithms and Computation (ISAAC'99), Volume 1741 of Springer LNCS, 1999, 113 - 122

[8] Capkun, S., Hamdi, M., Hubaux, J.P.: GPS-free Positioning in Mobile Ad-Hoc Networks, In Cluster Computing Journal, Vol. 5, No. 2., April 2002

[9] Chiang, C.-C., Wu, H., Liu, W., Gerla, M.: Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel, Proceedings of IEEE SICON '97, April 1997, 197-211

[10] Dousse, O., Thiran, P., Hasler, M.: Connectivity in ad-hoc and hybrid network, IEEE Infocom, New York, USA, June 2002

[11] Ephremides, A.: Energy Concerns in Wireless Networks, IEEE Wireless Communications, August 2002

[12] Estring, D., Govindan, R., Heidemann, J., Kumar, S.: Next century challenges: Scalable coordination in sensor networks, Proceedings of MOBICOM, Seattle, USA, 2001, 263 - 270

[13] Feeney, L.M.: An energy consumption model for the performance analysis of routing protocols for mobile ad hoc networks, ACM/Baltzer Mobile Networks and Applications, Vol. 6, Issues 3, June 2001

[14] Giordano, S., Hamdi, M.: Mobility Management: The Virtual Home Region, Technical Report No. SSC/1999/037, EPFL, Switzerland, October 1999

[15] Giordano, S., Stojmenovic, I., Blazevic, L.: Position based routing algorithms for ad hoc networks: A taxonomy, in 'Ad Hoc Wireless Networking', Cheng, X., Huang, X., Du, D.Z. (eds), Kluwer 2003 (http://www.site.uottawa.ca/ ivan/routing-survey.pdf)

[16] Haas, Z., Pearlman, M.R.: A Hybrid Framework for Routing in Ad Hoc Networks, Ad Hoc Networking, Addison-Wesley Longman Publishing Co., Inc., 2001 221 - 253

[17] Heissenbüttel, M., Braun, T.: Ants-Based Routing in Large-Scale Mobile Ad-Hoc Networks, Kommunikation in verteilten Systemen (KiVS03), Leipzig, Germany, March 25-28 2003

[18] Hou, T.C., Li, V.O.K.: Transmission range control in multihop packet radio networks, IEEE Transactions on Communications, 34, 1, 1986, 38-44

[19] Huang, Z., Shen, C.-C. , Srisathapornphat, C., Jaikaeo, C.: A Busy-Tone Based Directional MAC Protocol for Ad Hoc Networks, IEEE MILCOM, Anaheim, California, USA, 2002

[20] Jain, R., Puri, A., Sengupta, R.: Geographical Routing Using Partial Information for Wireless Ad Hoc Networks, IEEE Infocom, 2001

[21] Jacquet, P., Muhlethaler, P., Qayyum, A.: Optimized link state routing (OLSR) protocol, Internet Draft, draft-ieft-manet-olsr-06.txt, July 2002

[22] Johnson, D., Maltz, D.: Dynamic Source Routing in Ad Hoc Wireless Networks, Mobile Computing Systems and Applications, December 1994, 158-163

[23] Jubin, J., Tornow, J.D.: The DARPA Packet Radio Network Protocols. Proceedings of the IEEE, 75(1):21-32, January 1987

[24] Ko, Y.B., Vaidya, N.: Location-Aided Routing (LAR) in Mobile Ad Hoc Networks. ACM/Baltzer WINET J. Vol. 6, No. 4, 2000, 307-321

[25] Karp, B., Kung, H.T.: (GPSR): Greedy perimeter stateless routing for wireless networks, Proceedings ACM/IEEE International Conference on Mobile Computing and Networking, 2000, 243 - 254

[26] Karumanchi, G., Muralidharan, S., Prakash, R: Information dissemination in partitionable mobile ad hoc networks, Proceedings IEEE Symposium on Reliable Distributed Systems, Lausanne, October 2000

[27] Ko, Y.-B., Vaidya, N.H.: Geocasting in mobile ad hoc networks: Location-based multicast algorithms. Technical Report TR-98-018, Texas A&M University, USA, September 1998

[28] Kranakis, E., Singh, H., Urrutia, J.: Compass routing on geometric networks, Proceedings 11^{th} Canadian Conference on Computational Geometry, Vancouver, Canada, August 1999

[29] Kuhn, F., Wattenhofer, R., Zollinger, A.: Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing, MobiHoc 2003

[30] Li, J., Jannotti, J., De Couto, D., Karger, D., Morris, R.: A scaleable location service for geographic ad hoc routing, Proceedings of ACM MOBICOM, ACM, August 2000

[31] Li, Q., Aslam, J., Rus, D.: Online power-aware routing in wireless ad-hoc networks, Proceedings of MOBICOM, 2001

[32] Murthy, S., Garcia-Luna-Aceves, J.: An Efficient Routing Protocol for Wireless Networks, ACM Mobile Networks and Applications, Special Issue on Routing in Mobile Communication Networks, October 1996, 183-197

[33] Navas, J.C., Imielinski, T.: Geocast - geographic addressing and routing. In ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom '97), 1997, 66 - 76

[34] Nelson, R., Kleinroch, L.: The spatial capacity of a slotted ALOHA multihop packet radio network with capture, IEEE Transactions on Communications, 32, 6, 1984, 684 - 694

[35] Park, V., Corson, M.: A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks, Proceedings of INFOCOMM '97, April 1997

[36] Pei, G., Gerla, M., Hong, X: LANMAR: Landmark Routing for Large Scale Wireless Ad Hoc Networks with Group Mobility, Proceedings of the first ACM International Symposium on Mobile and Ad Hoc Networking, Boston, USA, 2000, 11-18

[37] Pei, G., Gerla, M., Chen, T.-W.: Fisheye State Routing in Mobile Ad Hoc Networks, Proceedings of the 2000 ICDCS Workshops, Taipei, Taiwan, April 2000, 71 - 78

[38] Perkins, C., Bhagwat, P.: Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers, Proceedings of the SIGCOMM '94 Conference on Communications Architecture, Protocols and Applications, August 1994, 234-244

[39] Perkins, C., Royer, E.:Ad-Hoc On-Demand Distance Vector Routing, Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, February 1999

[40] Rodoplu, V., Meng, T.H.: Minimum energy mobile wireless networks, IEEE Journal on selected Areas in Communications, Vol. 17, No. 8, August 1999, 1333-1344

[41] Singh, S., Woo, M., Raghavendra, C.S.: Power-aware routing in mobile ad hoc networks, Proceedings of MOBICOM, 1998

[42] Sobrinho, J. L., Krishnakumar, A. S.: Quality-of-Service in Ad Hoc Carrier Sense Multiple Access Wireless Networks, IEEE Journal on Selected Areas in Communications, 17(8), August 1999

[43] Stojmenovic, I.: Voronoi Diagram and convex hull based geocasting and routing in wireless networks, SITE, University of Ottawa, Canada, TR-99-11, December 1999

[44] Stojmenovic, I: A routing strategy and quorum based location update scheme for ad hoc wireless networks, Computer Science, SITE, University of Ottawa, TR-99-01, September 1999

[45] Stojmenovic, I., Lin, X.: Power-aware localized routing in ad hoc wireless networks, IEEE International Parallel and Distributed Processing Symposium, Cancun, Mexico, May 2000, 371 -376

[46] Takagi, H., Kleinrock, L.: Optimal transmission ranges for randomly distributed packet radio terminals, IEEE Transactions on Communications, 32, 1984, 246-257

[47] Hubaux, J. P., Gross, T., Le Boudec, J. Y., Vetterli, M.: Towards self-organized mobile ad hoc networks: the Terminodes project IEEE Communications Magazine, January 2001 (http://www.terminodes.org)

[48] Xu, Y., Heidemann, J., Estrin, D.: Geography-informed energy conservation for ad hoc routing. In seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking, July 2001

[49] Ye, W.; Heidemann, J.; Estrin, D., An energy-efficient MAC protocol for wireless sensor networks, Proceedings of INFOCOM 2002.

[50] Zhang, J.: Location Management in Cellular Networks, in Handbook of Wireless Networks and Mobile Computing, Ivan Stojmenovic (Editor), John Wiley & Sons, 2002, 27-49