OPTIMIZED POSITION-BASED ROUTING AND BROADCASTING IN MOBILE AD-HOC NETWORKS

Diplomarbeit
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

Markus Waelchli
Dezember 2004

Leiter der Arbeit:
Professor Dr. Torsten Braun
Institut für Informatik und angewandte Mathematik

Abstract

In the past few years mobile ad-hoc networks came increasingly into scope of interest. Mobile ad-hoc networks are mainly confronted with the problems of unknown topologies as well as mobility. Several approaches to support functionality and reliability under such conditions have been pronounced. One group of such schemes is based on the network participant's location information. Furthermore, most of those protocols distribute their position information pro-actively within their neighborhood using "hello-messages". At routing time however, that data may lack of accuracy because of outdated neighborhood information. This may cause network unreliability as well as routing overhead. Several approaches to avoid such inaccuracies are discussed and evaluated within the first part of this diploma work. Recently location-based routing schemes which abandons neighborhood knowledge have been proposed. Those schemes deliver the position information on-demand within data messages. A dynamic delayed broadcasting protocol, constitutive on such a scheme is introduced and evaluated in the second part of this thesis.

Contents

Chapter 1

Introduction

The cumulative, task-specific demand on wireless networks in the recent years led to the mobile ad-hoc network research topic. A major characteristic of those networks is their independence of any infrastructured backbone. Thus neither local nor global network information is natively available. To enable routing in such networks, an appropriate routing protocol has to gather the necessary topology information somehow. A number of routing schemes depending on the current location information of a node have been proposed in recent years. Those protocols fulfill the task described above by distributing the position information of every node periodically within the network.

The distribution of the current position information of a node is done by hello-messages which contain the data needed. The gathering of network topology information through hello-messages suffers from impreciseness as soon as the participating nodes are moving. The inaccuracy of that data cannot be avoided, as periodically updating of information is always confronted with the possibility of out-dated information. It can be minimized through appropriate techniques. Several possible approaches are implemented and investigated within this work. It is for example thinkable to correlate the moment of distributing a hello-message to the current speed of a node, or it may depend on the distance a node covers during a time period or it may be correlated to the number of link changes. A different approach than those mentioned so far is to expand a the functionality of a node by enabling it to predict the future positions of its neighbors. To do so, each node includes into its hello-messages its current speed as well as its moving direction. A receiving node is then able to predict the neighbor's current position. Greedy Perimeter Stateless Routing (GFG/GPSR) [1] is the protocol, based on the exchange of hello-messages, we use to evaluate our improvements. We will show that appropriate neighbor update techniques improve the network reliability and decrease the network load in an important amount.

The second part of the diploma thesis introduces a broadcasting protocol based on BLR techniques called Dynamic Delayed Broadcasting Protocol (DDB). Each recipient of a broadcast packet calculates a Dynamic Forwarding Delay (DFD), which is based on the information it got so far from previous senders of the packet. Each broadcast packets is buffered for this DFD before it is relayed or discarded. The value of the DFD thereby relates to the estimated progress a node may add. Several metrics can be taken into account to assign a the progress of a node. The recipient with the best progress calculates the shortest DFD and relays the packet first. Each node receiving a recently broadcast packet recalculates its progress. If that progress is below a predefined threshold the packet is dropped, otherwise it is buffered with the newly-calculated DFD. Nodes with low progress are assumed to be covered by other nodes and should therefore be starved out. To achieve comparability we implement other well known broadcasting algorithms and test our protocol intensely against.

An implicit postulate in location-based mobile ad-hoc networks is their assumption of circular transmission ranges for all nodes. In recent works it was shown that this assumption is not fulfilled within real wireless networks. Propagation media factors as well as manufacturing influences determine irregular radio ranges among different devices placed at multiple locations. A Radio Irregularity Model (RIM) [2] is implemented to investigate and evaluate our protocols under such unpredictable network effects.

In mobile ad-hoc networks, a ready-to-transmit source node does not know anything about the position of a destination. Most simulated routing protocols take this information as being given. The availability of this information however is not self-evident. Several approaches supporting such a service are proposed in literature. These protocols are often not yet implemented. Thus, the overhead they may add to a routing protocol is not considered in the simulations. To obviate that lack the Virtual Home Region (VHR) approach (at the same time introduced by [3], [4]) is implemented.

Location-based protocols need the position of the destination in order to make routing decisions possible. A source node obtains that information from a location service (e.g. GPS, VHR) and adds it into the packet header. However, the information delivered by such a service lacks of preciseness. Furthermore, it may be out-dated when the destination position is reached. Consequently, the destination is not accessible as it is not within the neighborhood of the position entered in the packet header. To deal with that drawback, a restricted local flooding algorithm is proposed. This feature is included in the implementation of the Beaconless Routing protocol [5].

Chapter 2

Irregularities on the Physical Layer

Radio irregularity is a common phenomenon in wireless sensor networks. One fundamental consequence of radio irregularity is the variation in packet loss in different directions. It is also an important reason for asymmetric links. To further investigate these impacts, we implemented a model proposed by [2], called Radio Irregularity Model (RIM). RIM takes into account two kinds of properties, i.e. the non-isotropic properties of the propagation media (the deviation of properties according to their physical location) and the heterogeneous properties of the devices. Both characteristics are handled within the RIM model. The Degree of Irregularity (DOI) deals with the influences caused by the propagation media, whereas the Variance of Sending Power (VSP) covers the device specific manufacturing properties. Other models, also handling radio irregularity are mentioned in the literature, like for example the Lognormal Shadowing Model [6]. For our approach, the RIM model seems to be more appropriate.

2.1 Reasons of Radio Irregularity

As already indicated above, radio irregularity is caused by two categories of factors: devices and the propagation media. The antenna type, the sending power, antenna gains of sender and receiver, the sensitivity and threshold of the receiver, and the Signal-Noise Ratio (SNR) form the different types of device properties. Examples of propagation media properties are the media type, the background noise, and temperature and obstacles within the propagation media. Within the RIM model, the view is focused on two of those factors, namely, the non-isotropic path-loss and the difference in sending power. Those factors are commonly regarded as the key causes of radio irregularity.

- **Non-isotropic path-loss**: A signal may be reflected, diffracted, and scattered in a medium. In consequence, radio propagation shows non-isotropic media patterns in most environments. Furthermore, a node may have different antenna gains for its directions. Thus, hardware calibration is also an important factor for non-isotropic path-loss.

- **Variance in Transmission power**: The difference of transmission power among equivalent devices arises from random factors during their production. Furthermore, batteries of different sensor devices flush at different frequencies, due to varying workload and different application environments.

2.2 Consequences of Radio Irregularity

Radio irregularity has important impacts on the MAC layer. It is an essential reason for asymmetric radio interference and asymmetric links and may affect the MAC layer with unpredictable behavior. Thus, the correctness of MAC layer functions may be affected. If for example a node wants to reserve the wireless channel using RTS/CTS, it may fail, as neighboring nodes may not hear the CTS packet and may interrupt the communicating nodes.

2.3 Path-loss Models

Within common, isotropic radio models path-loss is the same in all directions. Thus, transmission media influences are disregarded. Two approaches to approximate the path-loss are commonly in use: the free-space model and the two-ray model. To accommodate the influences of varying path-losses, the more adaptive RIM model will be introduced.

2.3.1 Free-space Path-loss Model

In the free-space model [7] the receiving power P_r on a node at distance d is:

$$P_r(d) = \frac{P_t \cdot G_r \cdot \lambda^2}{(4 \cdot \pi)^2 \cdot d^2 \cdot L}$$

where P_t is the transmitted signal power. G_t and G_r are the antenna gains of the transmitter and the receiver, respectively. L with $L \geq 1$ is the system loss, and λ is the wavelength. It is common to select $G_t = G_r = 1$ and $L = 1$ in simulations. If a receiver is within the circular transmission range, it receives all packets. Otherwise, it loses all packets. The free-space path-loss model, thereby, requires the absence of any reflections or multipath.

2.3.2 Two-ray Path-loss Model

In the two-ray model [8] the received power P_r at distance d is calculated in a slightly different way:

$$P_r(d) = \frac{P_t \cdot G_t \cdot G_r \cdot h_t^2 \cdot h_r^2}{d^4 \cdot L}$$

where h_t is the height of the transmitter antenna and h_r the height of the receiver antenna. The two-ray model accounts for a direct line-of-sight (LoS) path, and another path, reflected from a large object such as the ground. Thus, the two-ray model has a path-loss coefficient of 4 which is twice as high as the coefficient of the free-space model. Consequently, the signal-strength attenuates much faster than in case of free space.

2.3.3 Radio Irregularity Model (RIM)

The RIM model extends the common isotropic radio models in the following properties of real world radio signals: non-isotropy, continuous variation, and heterogeneity. To do so, the sending power of the device, its energy-loss, the background noise, and the interference among different communication signals are taken into account. It is important to realize that the RIM model indicates transmission ranges without fixed upper and lower boundaries which cannot be transgressed.

Degree of Irregularity (DOI)

The DOI is defined as follows: "the maximum received signal strength percentage variation per unit degree changes in the direction of radio propagation" [2]. The DOI adjusted path-loss handles the first two properties of radio irregularity, namely, the non-isotropy and continuous variation:

$$DOI\ Adjusted\ Path\ Loss = Path\ Loss \cdot K_i \tag{2.1}$$

where K_i is the i$^{\text{th}}$ degree coefficient, which is calculated in the following way:

$$K_i = \begin{cases} 1, & i = 0 \\ K_{i-1} \pm Rand \cdot DOI, & 0 < i < 360 \wedge i \in N \end{cases} \tag{2.2}$$

where $|K_0 - K_{359}| \leq DOI$.

With this formula, 360 different K_i values for the 360 directions used in the QualNet Environment are generated. A statistical analysis of the experimental data by [2] showed that the variance of the received signal strength in different directions fits the Weibull distribution. Therefore, the random number generator used in (2.2) generates numbers accordingly to the Weibull distribution. To generate a random number q in Weibull distribution, the following formula is used:

$$q = b \cdot (-\ln(1-\alpha))^{\frac{1}{a}}$$

where a is the shape parameter and b is the scale parameter.

Variance of Sending Power (VSP)

The sending power has to be adjusted to fit the heterogeneity. Differences in hardware calibration and battery status lead to aberration of sending power among equivalent devices. The VSP is introduced to take this behavior into account. It is defined as "the maximum percentage variance of the signal sending power among different devices" [2] and is modeled by the following formula:

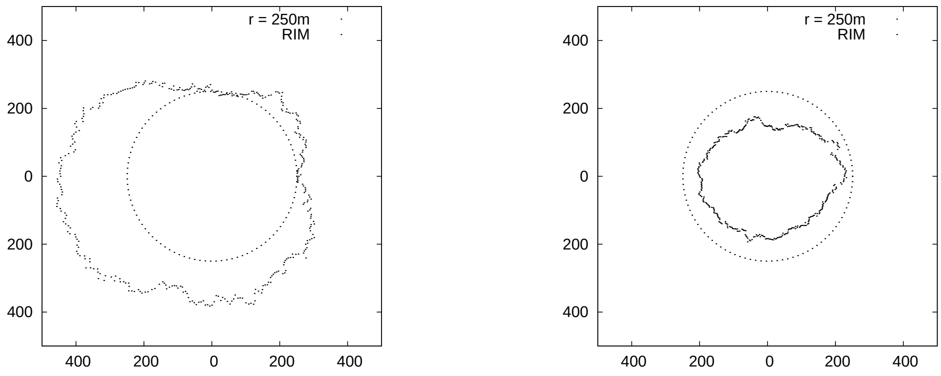
$$VSPAdjustedSendingPower = SendingPower \cdot (1 + Rand \cdot VSP)$$

where the variance of sending power follows a normal distribution. Therefore, a random generator which produces normal distributed random numbers is used.
The resulting received signal strength in the RIM model is finally:

$$Received\ Signal\ Strength = VSP\ Adjusted\ Sending\ Power - DOI\ Adjusted\ Path\ Loss + Fading$$

With the help of the RIM model, the impact of radio irregularity on routing and broadcasting protocols is investigated.

2.4 Examples of Transmission Ranges using the RIM



(a) Random example 1 (b) Random example 2

Figure 2.1: Different signal-strength for different directions.

Two examples of irregular radio ranges are shown in Figure 2.1. Both examples are derived from simulations done with the DDB protocol. A DOI of 0.01 and a VSP of 0.5 is used. Other values are possible, but [2] explained those values as being appropriate for real sensor networks. The difference of signal strength between two angles cannot be bigger than $DOI \cdot Tx$. This is obvious, as the calculation of the DOI depends on this condition.

Chapter 3

Mobile Ad-hoc Networks (MANET)

In the past few decades wireless networks have become increasingly popular in industry and an important topic of research. In the beginning, these networks were all stationary, whereas in recent years, they had to be adapted to mobility. The introduction of mobility added a lot of complexity. The topology was no longer static, thus routing protocols had to be enhanced with techniques to deal with these topology changes. Furthermore, bandwidth is highly irregular in these networks, as link changes arise proportional to the intensity of mobility. Other problems that may arise within mobile wireless networks are reliability, security policies, and privacy. In the following sections, the general schemes that operate in mobile wireless networks are introduced. The focus concentrates thereby on mobile ad-hoc networks [9]. All routing protocols used in that work are specific kinds of MANET.

3.1 Types of Mobile Wireless Networks

With progressing research, two kinds of mobile wireless networks manifested as being promising. The first scheme covers networks with a backbone infrastructure. These networks contain two kinds of nodes: wired gateways and mobile hosts. A network with fixed and wired gateways operates in the background. Each gateway covers an area where mobile nodes can move around. The bridges between the fix wired backbone network and the mobile areas are called base stations. Whereas the mobile hosts move, connect to, and communicate with the base station, they are currently within transmission range of. If a mobile host leaves the transmission range of a base station A, and enters the area of a base station B, a ”Hand Off” from A to B occurs in order to continue communication transparently.

The second group of mobile wireless networks refrains from any fixed infrastructure and is therefore called MANET. As the name indicates, these networks have no fixed routers and no backbone topology is available. As all nodes may arbitrary move around and can be connected dynamically in an arbitrary way, the network has to be totally self-organizing. It must furthermore be adaptive to topology changes. In the next sections, different kinds and approaches of routing protocols for MANET are presented.

3.2 Existing MANET Routing Protocols

[10] gives an overview of current routing protocols for MANET. The protocols introduced below have to deal with the typical limitations of ad-hoc networks, i.e. low battery power, low bandwidth, and high error rates. All existing protocols may further be divided into three typical categories:

- Table-driven routing protocols (DSDV, CGSR)
- Source-initiated routing protocols (AODV, DSR)
- Position-based routing protocols (GFG/GPSR, BLR)

The categories and their implementations are discussed below:

Table-driven routing protocols require the maintenance of a set of routing tables from each node, where the information about routes to all other nodes in the network is stored. Thus, to a specific node, the information about routes to every other node in the network is always available. To do so, update information is distributed throughout the whole network, whenever topology changes occur. The protocols implementing this scheme differ in the number of necessary routing tables and the methods how the topology changes are distributed.

Source-initiated on-demand routing operates quite different. First of all, a route is only created when needed by the source node. So, if a node needs a route to another node, a route discovery process is initiated within the network. This process terminates either when a route has been found, or when all possible route permutations have been examined. In the second case, no routing is possible, because no way exists. To assure reliability, the protocol has to support a route maintenance operation which is performed as long as the destination is accessible or the communication is desired. The route maintenance procedure has to deal with route breaks in case of topology changes.

In the third kind of protocols, no routes are created and all routing decisions are taken only locally, due to the neighbor position information of a node. To enable routing this way, a node has to know the positions of its neighbors. This is achieved with hello-messages containing the current position of a node. They are periodically distributed in the neighborhood of each node. These messages enable the setup of the routing tables. In recent time, position-based routing protocols avoiding that beaconing mechanism are proposed. These protocols use methods to forward packets in predefined regions, without knowing anything about the surrounding topology.

3.3 Table-driven routing protocols

3.3.1 Destination-Sequenced Distance-Vector Routing (DSDV)

DSDV [11] is based on the Bellman-Ford routing algorithm [12], enhanced with loop-freedom. In DSDV, each node maintains a routing table with records for all possible destinations and the hop count to reach them. Furthermore, each entry has a sequence number which is increased whenever an update message from the associated destination arrives. To insure the routing table accuracy, update messages are periodically distributed throughout the whole network. To optimize the update strategy two kinds of messages exist: The first is known as full dump and carries all available routing information in the network. These packets are transmitted infrequently. The second types are smaller, incremental packets which distribute only the information about what has changed since the last full dump.

When a data packet has to be sent by a node, it consults its routing table and choses the route labeled with the most recent sequence number. If entries with the same sequence number exist, the one with the smallest hop count is chosen in order to minimize hop count. Some optimizations to that protocol had further been proposed [11].

3.3.2 Clustered Gateway Switch Routing (CGSR)

Clustered Gateway Switch Routing (CGSR)[13] uses the DSDV Routing algorithm described in the previous section as its basis. The mobile nodes are aggregated in clusters and a node is selected as the cluster-head among them. It organizes channel access, routing and bandwidth allocation. To deal with the problem of specifying the cluster head a Least Cluster Change (LCC) algorithm is proposed. In LCC, cluster-head change occurs only if a change in network causes two cluster-heads to come into one cluster or one of the nodes moves out of the range of all the cluster-heads.

To route traffic from the source to the destination, a hierarchical cluster-head-to-gateway routing approach is used, where a node can only be a gateway to other clusters if it is at least in transmission range of two clusters. In Figure 3.1 an example of the CGSR operation is described.

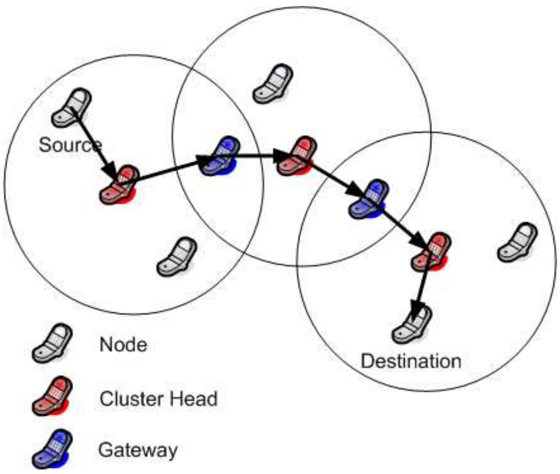


Figure 3.1: CGSR: routing from source to destination.

The source of the packet transmits the packet to its cluster-head. From this cluster-head, the packet is sent to the gateway node that connects this cluster-head and the next cluster-head along the route to the destination. The gateway sends it to that cluster-head and so on till the destination cluster-head is reached. The destination cluster-head then transmits the packet to the destination.

Each node maintains a cluster member table which contains the mapping from each node to its respective cluster-head. Each node broadcasts its cluster member table periodically and updates its table after receiving the cluster member table of other nodes. Additionally each node maintains a routing table that determines the next hop to reach the destination cluster.

3.4 Source-initiated Routing Protocols

3.4.1 Ad Hoc On-Demand Distance Vector Routing (AODV)

AODV [14] is based on the DSDV protocol which is already described in 3.3.1. In contrary to DSDV it creates routes only on demand, i.e. whenever a data packet has to be sent. Thus, it drastically minimizes the number of broadcasts. To initialize a communication with another node, a source node starts a path discovery process by sending a route request (RREQ) packet to its neighbors. This is repeated by the receiving neighbors until a route to the destination is found. Each node receiving the packet updates its routing tables with the obtained data. AODV further bases on sequence numbers to ensure loop-freedom and up-to-date route information. If the destination or an intermediate node with a valid route is reached, a unicast route reply

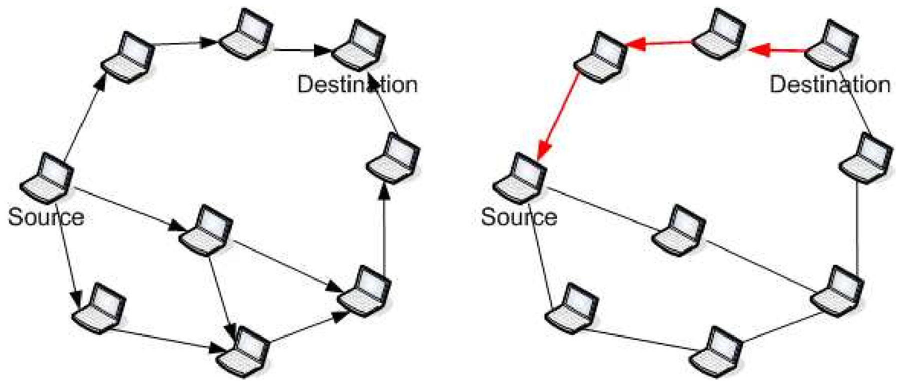


Figure 3.2: AODV route discovery with RREQ and RREP.

(RREP) is sent back on the reverse path to the source node. Each node along that reverse path sets up a forward route entry in its table. In Figure 3.2 the route request procedure and the route reply are depicted. A source node is able to reinitiate the path discovery, when it moves. If an intermediate node moves, all its upstream neighbors notify that movement and propagate a link failure notification message (RREP with infinite metric) to each of its active upstream neighbors, and so on, until the source node is reached. The source node may then reinitiate the path discovery if a route still is needed. An additional hello-message service exists to maintain local connectivity.

3.4.2 Dynamic Source Routing (DSR)

DSR [15] is another well-known on-demand routing protocol. Nodes are expanded with route caches which record all source routes the node is aware of. The route cache entries are updated whenever a new route is learned. If a source node has no valid route cache entry to the destination, the route discovery mechanism is initialized by sending a route request packet. Each intermediate receiver without a valid route cached rebroadcasts the packet extended with its own address. A node only forwards a route request if it has not already seen the request before. Thus, the number of route requests is limited. When the destination is reached, or when an intermediate node has a valid route cache entry to the destination, the route reply is initiated. The route reply is sent back the reverse path if symmetric links are supported. Otherwise, a route discovery to the source is initiated and the route reply piggybacked. Therefore, DSR supports also unidirectional links.

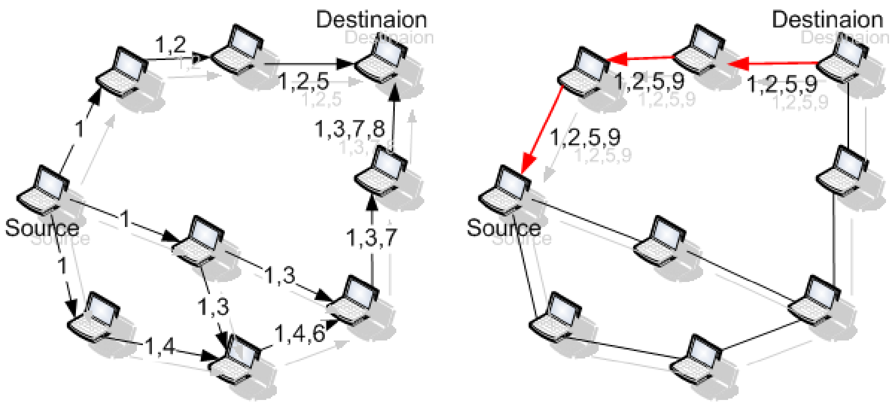


Figure 3.3: Path discovery with route records in DSR.

In Figure 3.3 the route discovery functions are shown. Route error packets and acknowledgments are used to support route maintenance. When a transmission problem is encountered on the physical layer, a route error packet is broadcasted. Each node receiving this packet truncates each route cache entry containing the hop where the error occurred. In addition to route error messages, acknowledgments are used to verify the correct operation of the route links.

3.5 Loaction-based routing protocols

GFG/GSPR and BLR which are described in the following form the underlying protocols of the investigations and refinements done in that work. For this reason they are introduced in more detail.

3.5.1 Greedy Perimeter Stateless Routing (GFG/GPSR)

GFG/GPSR is a position-based routing protocol proposed by [1]. Each node in the network periodically sends hello-messages (beacons) to update its immediate neighbors with its current position information. This information is used to setup and update the neighborhood tables on each node. The accuracy and the topicality of the position information strongly relate to the frequency the hello-messages are broadcast in.

More precisely, after each beacon interval B a node transmits a beacon to the broadcast MAC address, containing only its own identifier (e.g., IP address) and position. To avoid synchronization of neighbors' beacons, as observed by [16], each transmission of a beacon is jittered by 50% of the interval B between beacons, such that the mean inter-beacon transmission interval is B, uniformly distributed in $[0.5B; 1.5B]$. Upon not receiving a beacon from a neighbor for longer than dead interval D, a GFG/GPSR node assumes that the neighbor has failed or left transmission range, and removes the neighbor from its table. $D = 4.5B$ is used as a dead interval.

To send packets along paths, there exist two modes: Greedy Mode and Perimeter Mode. Greedy mode is quite fast and is used as long as possible. If forwarding in greedy mode fails, perimeter mode is used as a backup strategy. Its general functionality is to route along the perimeter of void areas. Both modes are explained in more detail in the next two subsections.

Greedy Mode

As a node knows its neighbors positions, it can make optimal local decisions if it chooses the closest neighbor to the destination. A packet being forwarded this way arrives successively closer to the destination. The big advantage of greedy mode is that it depends only on the immediate neighbors of the forwarding node. The disadvantage of greedy forwarding comes along if in a given topology the route to the destination requires a packet to move farther away from the destination. In such a scenario a local maximum occurs, where a route to the destination still exists, but is not accessible in greedy mode. Therefore, a backup mechanism called perimeter mode is used.

Perimeter Mode

GPSR uses perimeter mode if greedy forwarding fails. The functionality of the perimeter mode is described in the following: First, a planar graph is computed. A graph is defined as being planar if no two edges cross. Each subset of nodes in a network can be considered as a graph. The nodes are the vertices and existing communication links between the nodes are the edges. Generally, such a network graph is not planar, as a lot of communication links overlay each other. Therefore, GPSR uses known mechanisms to make those graphs planar. The Relative Neighborhood Graph (RNG) and the Gabriel Graph (GG) are two planar graphs used in GPRS, which reduce any graph to planarity. One important impact while reducing the graph to RNG or GG is that removing edges from the graph must not disconnect the original network graph. The RNG is defined as follows:

> An edge (u, v) exists between vertices u and v if the distance between them, $d(u, v)$, is less than or equal to the distance between every other vertex w, and whichever of u and v is farther from w. In equation form:
>
> $$\forall w \neq u, v : d(u, v) \leq max[d(u, w), d(w, v)] \tag{3.1}$$

By removing edges which are not part of the RNG we cannot disconnect the graph. This is obvious, as we eliminate only edges within the shaded area in Figure 3.4. Thus, eliminating an

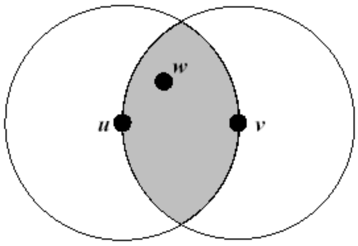


Figure 3.4: RNG: If edge (u, v) shall be included the shaded area has to be empty.

edge may lead to another path through the network, but the connectivity of the graph is granted. The GG is not further explained, as its form is almost equal to the RNG. Furthermore, the RNG is a subset of the GG and therefore the appropriate choice for GPSR. Once built, the graph is traversed with the right hand rule. If the packet arrives at a node which has a neighbor closer to the destination than itself, the perimeter mode switches back to greedy mode. Else, the packet is forwarded in perimeter mode until a loop occurs or the time to live field of the packet is exceeded. In both cases the packet is dropped. The right-hand rule works as follow:

> When arriving at node x from node y, the next node chosen is the first one sequentially counterclockwise about x from edge (x, y).

The right-hand rule traverses the interior of a closed polygonal region in clockwise edge order and the exterior of a polygonal region in counterclockwise edge order. Besides the right-hand rule, GPSR uses a face change mechanism to forward packets in perimeter mode on progressively closer faces. A graph has two kinds of faces: the interior faces are the closed polygonal areas bounded by the graph's edges, the exterior face is the unbound face outside the outer boundary of the graph. Within each face GPSR uses the right-hand rule to reach an edge, that crosses the line $\overline{xD}$, where x is the position of the node where the perimeter mode was entered and D is the position of the destination. After traversing that edge, the traversal succeeds on the adjacent face crossed by $\overline{xD}$ on its way to the destination. The Perimeter Mode guarantees delivery if there exists at least one connected path to the destination in the original network graph.

IEEE 802.11 Enhancements

To make GPSR robust on a IEEE 802.11 network, the following options are supported. These enhancements are GPSR specific and the only differences to the GFG protocol.

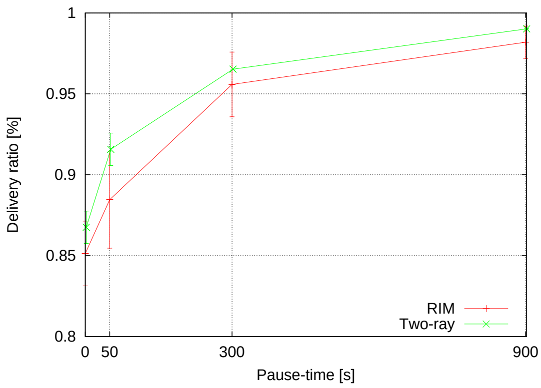
- **Support for MAC-layer failure feedback**: If a packet exceeds its maximum number of retransmitting attempts on the MAC 802.11 layer, the GPSR protocol is informed. This notification indicates either a network deadlock according to congestion or that the neighbor has left the transmission range. The maximum number of retransmissions is seven in the MAC 802.11 standard. This notification enables GPSR to chose another neighbor and forward the packet to it.

- **Interface queue traversal**: Additionally to the notification of a MAC failure feedback, the interface queue is traversed and all packets addressed to the failed recipient are removed. Those packets are returned to the routing protocol and re-forwarded to the next hop.

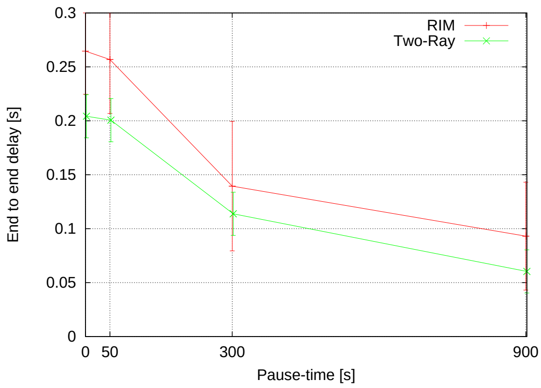
- **Promiscuous use of the network interface**: GPSR disables MAC address filtering to receive copies of all packets for all stations within its radio range. All data packets include the position of their last hop. Thus, the sending of beacons piggybacked on data packets is enabled. Consequently, the rate at which beacons must be sent is reduced. This mode is especially useful in regions under high traffic load.

Impact of radio irregularity in GFG/GPSR

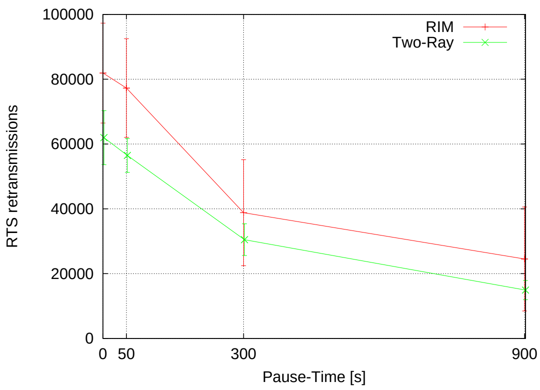
In this section, the effects of the RIM model on GFG/GPSR are investigated. However, all kinds of MANET protocols could be tested against radio irregularity. We show these results here because GFG/GPSR is in the following only simulated to evaluate neighbor table accuracy.



(a) Delivery ratio



(b) Delay



(c) RTS retransmissions

Figure 3.5: GPSR performance with radio irregularity or two-ray model.

The results obtained with the RIM model are compared to results obtained with the standard two-ray model. The DOI is renewed, whenever a node has covered more than $50m$. In the proposal of [2] the DOI is calculated only once at the beginning of the simulation. They did so because they tested static networks. Up to now, they have not gathered any results when the update mechanism should be done. We fixed the DOI update time point to $50m$. A to small value makes no sense as the transmission range should normally not change a lot if a node moves just a short distance. For the opposite reason, a too high value makes no sense, either. With our assumption we think we reproduce realistic scenarios.

The standard GFG/GPSR protocol with the parameters mentioned by [1] is used for the simulations. The simulations are run with eight seed values to gain statistical relevance and to keep the confidence intervals small. In Figure 3.5 the results are depicted. The RIM model produces slightly poorer results than the two-ray model. The increased end-to-end delay as well as the decreased delivery ratio correlates to the increased number of RTS retransmissions. This value follows from the higher number of wrong routing entries. Consequently, radio irregularity influences the reliability as well as the end-to-end delay a little, because more wrong routing table entries exist. This is obvious as the radio range changes frequently and the neighbor table update used in GFG/GPSR assumes bi-directional links. The impact of the RIM model on the other protocols is evaluated in chapter 6.

Contrary to the results obtained by [2], GFG/GPSR does not suffer much from radio irregularity. This is easy to explain. Their network was static. Furthermore, neither backup mode nor any additional handling of not deliverable packets were considered. Thus, each wrong routing decision led to the immediate drop of a packet. It is possible that a routing protocol would suffer more in a sparser network than the one we simulated. As soon as the network is dense enough, and a mechanism to handle undeliverable packets exists, the possibility that there is a path through the network is sufficiently high.

3.5.2 Beaconless Routing Algorithm for Mobile Ad-hoc Networks (BLR)

The BLR routing protocol [17] performs routing without knowledge of the neighborhood of a node. Consequently, no hello-messages are needed. The algorithm operates as follows. A source node broadcasts its data packet within its one-hop neighborhood. Furthermore, only one of its neighbors is allowed to rebroadcast the packet. This restriction is achieved through a dynamic forwarding delay (DFD). Therefore, each neighbor computes its DFD depending on its position in relation to the destination and the previous node. The node located at the best position, e.g. the node closest to the destination, calculates the shortest DFD. Consequently, it rebroadcasts the packet first. The forwarding area is restricted to ensure that all members of it detect the transmission. All nodes not participating in that area simply ignore the broadcast of a packet. All participants of the forwarding area notice the transmission and drop their copy of the packet.

The path of a data packet routed with BLR is depicted in Figure 3.6. BLR choses always the node with the best progress within the circular forwarding area (shaded circles). The other participants of the forwarding area ignore the packet release and drop their copy accordingly.

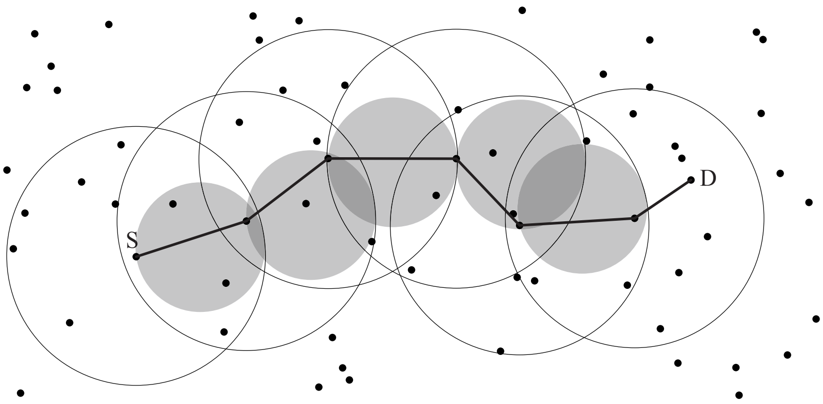


Figure 3.6: Greedy mode forwarding in BLR [5].

Forwarding Area

The forwarding area maintains the following restriction: Each participant of the forwarding area must be within the transmission range of each other node in the forwarding area. This restriction is necessary, as each node in the forwarding area must hear a relaying node. Furthermore, it avoids duplicated packets. The forwarding area should be as large as possible to increase the number of participants. The shape should favor those nodes located near to the transmission range boundary.

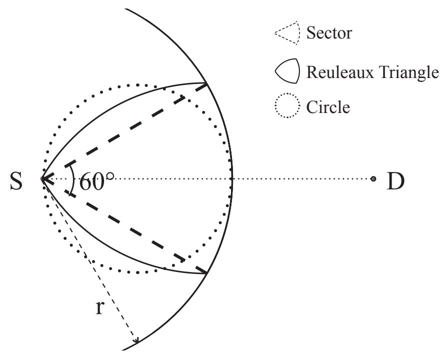


Figure 3.7: Forwarding areas in BLR [5].

The three forwarding areas proposed in the BLR routing protocol are depicted in Figure 3.7. The areas are shaped as a sector, a Reuleaux triangle, and a circle. All three areas satisfy the condition of mutual perception. Both, the sector and the Reuleaux triangle use a 60° angle to limit the distance between two randomly placed nodes to a maximum of r. The circle on the other hand has a diameter of r.

Dynamic Forwarding Delay (DFD)

The BLR protocol supports different DFD functions to calculate the time a node has to buffer a message before forwarding it. A message is dequeued and dropped if the release of a copy of the currently buffered message by another node is registered. To determine the DFD, each node has to calculate its progress p. Thereby, p is calculated depending on the position of the node itself, the position of the node it received the message from, and the destination. To do so, the node calculates its distance d from the line $\overline{SD}$, where S is the last hop and D the destination. Depending on that value, the node chooses its DFD from the interval [0, Max_Delay]. The following equations describe the three DFD functions featured by the BLR protocol:

$$DFD = Max\_Delay \cdot \left(\frac{r-p}{r}\right) \tag{3.2}$$

$$DFD = Max\_Delay \cdot \left(\frac{p}{r}\right) \tag{3.3}$$

$$DFD = Max\_Delay \cdot \left(\frac{e \cdot \sqrt{p^2+d^2}}{e}\right) \tag{3.4}$$

The first equation (3.2) implements the MFR [18]. The more progress a node has the less delay it calculates. Consequently, the node with the largest progress relays the packet first.

Equation (3.3) implements a slightly modified NFP [19]. The NFP is not directly applicable, as a node does not know its nearest neighbor. To approximate that behavior best the node with the lowest progress calculates the shortest delay and forwards the packet first. The intention of that version is to reduce energy consumption and to increase the number of simultaneous transmissions. This increases the overall capacity of the network.

In equation (3.4) an advanced approach to calculate the DFD is chosen. Not only the progress of a node, but also its distance to the previous node is taken into account. The principle is to sustain nodes which are in straight direction to the destination. If we used progress only, a node far away from the line $\overline{SD}$ may be preferred, even if there is a node with just a little less progress, but a much better position. Furthermore, [20] showed that exponentially distributed random timers can reduce the number of responses compared to uniformly distributed timers. This feature is also honored within equation (3.4).

Backup Mode

Each sending node x overhears the packet release of its chosen next hop. This is obvious, at least if we assume circular transmission ranges, as the next hop is within reachability of x. Consequently, each neighbor of a releasing node is able to passively acknowledge a packet forwarding. Furthermore, the packet has to be released by the next hop not later than the current time plus Max_Delay. Thus, if no passive acknowledgment was received on the node x, after that interval, its forwarding area can be assumed to be empty and a backup strategy is initialized. Therefore, node x broadcasts a request packet and all neighbors reply with a response message containing their current position. The backup strategy used in BLR is similar to the perimeter mode used in the GFG/GPSR protocol [1]. If a node closer to the destination exists among the replying nodes, this node is chosen as the next hop and the packet is forwarded to it in greedy mode. Otherwise, a planar graph (e.g. Gabriel Graph) is calculated for the local neighborhood of x and the packet is forwarded using the right-hand rule. The planar graph is necessary in order to prevent loops. The position where greedy mode failed and perimeter mode is started is saved within the packet. As soon as the packet arrives at a node closer to the destination than the perimeter entering point, the backup mode switches back to greedy forwarding. A packet is dropped if a loop occurs or the time-to-live field expires.

3.6 Broadcasting Mechanisms and Requirements

Network wide broadcasting is the process in which one node sends a packet to all other participants of the network. In mobile ad-hoc networks many protocols use broadcasting services to establish routes. Dynamic Source Routing (DSR), Ad-Hoc On Demand Vector Routing (AODV), Zone Routing Protocol (ZRP), and Location Aided Routing (LAR) are a few examples which use a simple form of broadcasting called Flooding, in which each node retransmits a received unique packet exactly once. Bandwidth congestion as well as inefficient use of node resources are the disadvantages of flooding. Several enhancements to a Simple Flooding algorithm have been proposed in the literature. In the last part of this chapter we give a short overview of those schemes and introduce our own approach subsequently.

MANET broadcast protocols use the IEEE 802.11 MAC [21] standard. MAC 802.11 supports Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) to deal with possible collisions. Furthermore, it includes functions to acknowledge packet delivery. The main source of collision occurrences is the hidden node problem, where a node is unable to ascertain the sending status of its neighbors and, thus, starts to send a packet although the physical channel is occupied. MAC 802.11 uses Request to Send (RTS)/Clear to Send (CTS) and Data/Acknowledge routines to deal with those problems. Using these techniques for broadcasting would intensify the disadvantages of broadcasting mentioned above. Thus, collision avoidance for broadcasting techniques is not feasible in an efficient way. The absence of an acknowledge routine disables a node from knowing if a broadcast packet is successfully delivered or not. This drawback may lead to a lot of dropped packets in congested networks where many collisions occur. All proposed schemes try to handle those problems by limiting the number of rebroadcasts. Thus, the number of rebroadcasting nodes is the determining metric to minimize overhead.

In broadcasting we are concerned with the problem of simultaneous sending. To deal with it, the transmissions of packets are jittered. To keep track of the redundant packets received, many broadcasting schemes use a time interval called Random Assessment Delay (RAD) to determine if a packet shall be broadcasted or not. The RAD is randomly chosen between 0 and T_{max}, where T_{max} is the highest possible delay. There exist two approaches how to implement the RAD. In the first one, packets are sent to the MAC layer and queued there in the interface queue (IFQ) until the channel becomes free. If meanwhile the network layer protocol decides that rebroadcasting is not necessary, the MAC layer has to be informed to discard the packet. In the second approach, the packet is kept in the network layer until the RAD expires.

3.7 Categorization of Broadcast Protocols

Broadcasting protocols fall under four categories which are called Simple Flooding, Probability-based Methods, Area-based Methods and Neighbor Knowledge Methods. In Simple Flooding, a node has to rebroadcast all packets it receives exactly once, whereas in the Probability-based Methods the rebroadcasting decision bases on the local link density of a node. In the Area-based Methods, nodes only retransmit a packet if sufficiently enough new area is covered with that broadcast. Neighbor Knowledge Methods demand the existence of neighborhood information to decide any further broadcasting.

3.7.1 Simple Flooding

In Simple Flooding protocols ([22], [23]), a source broadcasts a packet to all its neighbors. Each neighbor rebroadcasts the packet exactly once and so on, until each node has rebroadcasted the packet it initially received. Flooding has the disadvantage of high overhead, as each packet is rebroadcasted on each node and no optimizations are considered. The big amount of redundant information leads to increased reliability and the information loss through collisions is better prohibited than in other approaches. The large number of transmissions in the Simple Flooding scheme leads to additional collisions. This behavior reduces its advantage.

3.7.2 Probability-based Methods

Probabilistic Scheme

[24] introduced a broadcasting refinement, where the density of the network is assigned to a probability. According to that probability, nodes rebroadcast a packet or not. This approach is called Probabilistic Scheme. As in dense networks several nodes share the same transmission coverage, it suffices that only a subset of those nodes forward the packet. Thus, good reliability should still be achieved. Consequently, in dense networks the probability parameter is low and only a few randomly chosen nodes rebroadcast the packet. The sparser the networks are, the higher is the probability parameter. Consequently, a probability of 100% is equal to flooding.

Counter-based Scheme

[24] showed an inverse relationship between the number of packets received within a predefined interval on a node, and the area additionally reached by that node. Based on that behavior, they introduced the Counter-based Scheme. Whenever a node receives a not yet seen packet, it initializes a counter with one and sets a RAD. If during that RAD expiration more redundant packets than a predefined threshold are received, the packet is dropped, else it is rebroadcasted. [24] showed that a threshold higher than six does not provide much additional area. The Counter-based Scheme is simple and well adaptive to local topology.

3.7.3 Area-based Methods

Other broadcasting strategies proposed by [24], which depend on the area a node additionally covers, are the Area-based Methods. The calculation of the additional area is done by evaluating the redundant packets which are received during a RAD expiration. The Area-based approach takes only the coverage area of a transmission into account, whether it is void or not is of no importance. [24] introduced two schemes to calculate the additional area covered by a node.

Distance-based Scheme

Using the Distance-based Scheme [24], the forwarding decision depends on the distance between a node and each neighbor it got a broadcast packet from. Like in the Counter-based Scheme, a RAD is initialized and the distances between the receiver and the sources of each message are calculated. If there exists a node among those senders which is closer than a specific threshold, the scheduled packet is dropped, else it is rebroadcasted. The distance is mapped from the received signal strength. Therefore, no Global Positioning System (GPS) has to be used. Another approach using GPS is discussed in the following.

Location-based Scheme

This scheme [24] supports a more accurate estimation of the additional area covered by a node. Each node uses GPS to determine its own position which is included in the header of every broadcast packet. The additional area a node contributes is calculated from the position position of the node and the position it received in a packet. If it is less than a predefined threshold, it rebroadcasts the packet immediately, else a RAD is initiated. In the meantime, each reception of a redundant packet leads to the recalculation of the additional area, depending on the information already gathered. If after the expiration of the RAD the additional area of a node is still over the threshold, the packet is rebroadcasted.

3.7.4 Neighbor Knowledge Methods

[25] compared several broadcast strategies based on the knowledge of the local neighborhood of a node. Some of those protocols are shortly introduced in the following.

Flooding with Self Pruning

The simplest of them is referred as Flooding with Self Pruning [26]. Each node has to know its one-hop neighbors. This is achieved by sending hello-messages. A sending node includes its neighbor list in its broadcast packet. Each receiving node compares the neighbor list of the sending node with its own neighbor list and rebroadcasts the packet only if it reaches additional nodes.

Scalable Broadcast Algorithm (SBA)

The Scalable Broadcast Algorithm (SBA) [27] is based on the knowledge of the local two-hop neighborhood of a node. This knowledge is achieved by adding the neighbors list of a node to each of its hello-messages. Consequently, each node has the two-hop neighborhood information centered on itself, because it knows its neighbors as well as their neighbors. If additional nodes may be reached from a receiving node, a RAD is initialized, else the packet is dropped. The RAD is thereby dynamically adjusted to the network topology, by setting it to $\left(\frac{d_{Nmax}}{d_{me}}\right)$, where d_{Nmax} is the maximal neighbor degree of all the neighbors of the node and d_{me} is the current number of neighbors of the node. The packet is dropped if during the RAD expiration a packet is received which determines the node to not reach additional nodes anymore, else the packet is rebroadcasted.

Multipoint Relaying Protocol

The Multipoint Relaying Protocol [28] is based on subsets of one-hop neighbors which are forced to rebroadcast a packet. The chosen nodes are called Multipoint Relays (MPRs). Since a MPR knows the local two-hop network topology it can choose its most efficient one-hop neighbors as MPRs. The MPR set of a node x is calculated by the following algorithm:

1. Calculate the degree $D(y)$ of y, where y is a neighbor of x, for all neighbors of x. The degree is defined as the number of neighbors of node y, excluding all neighbors which are a neighbor of x themselves as well as x itself.

2. Add to the MPR set those neighbors of x which provide exactly one link to a two-hop neighbor of x. If for example two-hop neighbor b can be reached only through neighbor a, then add a to the MPR set of x. Remove all two-hop neighbors which are now covered by a node in the MPR set from the two-hop neighbor list.

3. While there exist two-hop neighbors that are not yet covered by at least one node in the MPR set:

   3.1. Calculate the reachability for each neighbor y of x, i.e. the number of 2-hop neighbors that are not yet covered by at least one node in the MPR set and that are reachable through y.

   3.2. Select as a MPR the neighbor with best reachability. In case of multiple nodes providing the same reachability, select the node with highest $D(y)$ as MPR. Remove all two-hop neighbors that are now covered by a node in the MPR set.

You can see a MPR set calculation in Figure 3.8. First, the $D(y)$ is calculated for each neighbor y of node 1. In step two of the algorithm nodes 5 and 6 are selected into the MPR set, as they are the only nodes that support accessibility to nodes 4 and 7, respectively. The covered two-hop neighbors are deleted from the two-hop neighbor list (canceled red or blue). As there still a two-hop neighbor exists (11) that is not yet covered by a node in the MPR set, step 3 of the algorithm is applied. Node 8 has a 2-hop reachability of two, whereas node 9 has a two-hop

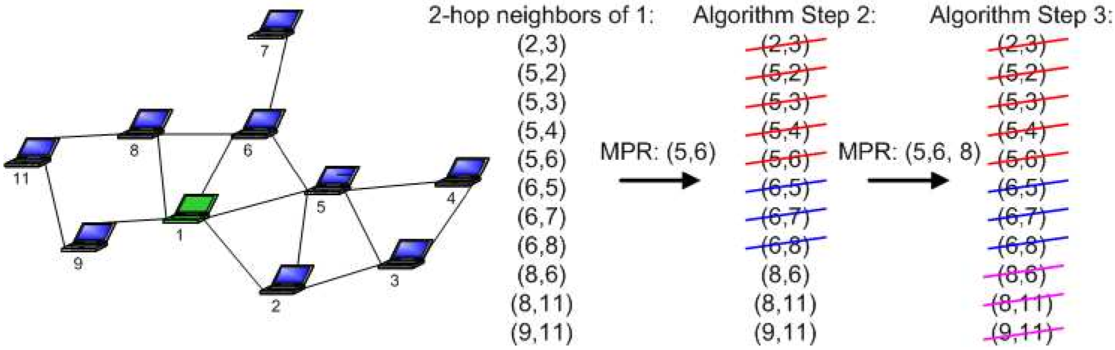


Figure 3.8: Calculation of a node's MPR set.

reachability of one. Thus, node 8 is selected as a MPR member, and all remaining two-hop neighborhood entries are removed. Furthermore, node 1 includes its chosen MPR set into its next hello-message and all neighbors receiving that hello-message check if they are contained in the MPR set of the sender. If so, they have to rebroadcast all packets they receive. The MPR of a node is recalculated, whenever a one-hop or two-hop neighborhood change occurs. MPR is part of the Optimized Link State Routing (OLSR) [29] and described in detail there.

Ad Hoc Broadcast Protocol (AHBP)

Another approach that operates similar to MPR is the AHBP [30] protocol. AHBP differs from MPR in the following points: In Multipoint Relaying the MPR designation is distributed via hello-messages. AHBP in contrast informs a node which has become a Broadcast Relay Gateway (BRG) in the header of the broadcast packets, i.e. the information about the subset of neighbors that are the BRGs of the node is added to the broadcast packet. Each receiving node that is listed as a BRG uses its two-hop neighbor knowledge to remove all neighbors that received the packet in the same transmission. Those neighbors are deleted from the neighbor graph used to calculate the next hop BRGs. Consequently, the subset of next-hop BRGs can be calculated on time. Unlike MPR, AHBP is also considered to account for high mobility networks.

3.8 Outlook

This diploma thesis concentrates on location-based MANET routing protocols. In particular, on the position based routing protocols GFG/GPSR and BLR, as well as a broadcasting protocol called DDB. Within GFG/GPSR, possibilities to optimize the accuracy of local neighborhood information are investigated. This work is done in chapter 5. In chapter 6, a location-based broadcasting algorithm (DDB) is presented. It is further simulated and evaluated in competition to some known broadcasting algorithms. Whereas in the BLR protocol a local flooding strategy to deal with out-dated destination information is implemented in chapter 3.5.2. A home-region based destination search protocol [3],[4] is implemented to investigate its impact on reliability and the additional network load, in chapter 7.1.

Chapter 4

Simulation Environment

All simulations are performed using the QualNet v3.6 [31] network simulator developed by Scalable Network Technologies. QualNet is a commercial application based on GloMoSim [32]. An Intel(R) Pentium(R) 4 (CPU 1.8GHz, 736MB RAM) was used for the simulations. The university version of QualNet is built to run on one CPU only, besides there exists a multi-threaded version of QualNet for parallel processing.

4.1 QualNet v3.6

”QualNet is a discrete event simulator developed by Scalable Networks. It is extremely scalable, accommodating high-fidelity models of networks of 10’s of thousands of nodes. QualNet makes good use of computational resources and models large-scale networks with heavy traffic and mobility, in reasonable simulation times.”, this is how Scalable Network Technologies shortly describe their network simulator.

4.1.1 QualNet overview

QualNet is a discrete event simulator. State changes occur at discrete points in simulation time (message generation, packet arrival, packet departure, etc.). These points in time are scheduled by the event manager which contains information about any event that has to be processed. The event scheduler furthermore ensures that QualNet processes all events in strict timestamp order, which means the computation consists of sequences of event computations. The computational load is thereby proportional to the number of events. The number of events is further proportional to the amount of traffic, as well as the number of node specific events. Consequently, each protocol implemented in QualNet operates as a final state machine which only changes its state when an event occurs.

QualNet is a modeling tool for wireless and wired networks. The QualNet suite is composed of QualNet Simulator which claims to be the fastest real-time traffic-modeling tool. QualNet Animator allows to graphically design the network model, using a wide library of components and displays the results of simulation runs. QualNet Designer allows to create finite state automates to describe the behavior of a protocol. With QualNet Analyzer and Designer, simulation results can be interpreted.

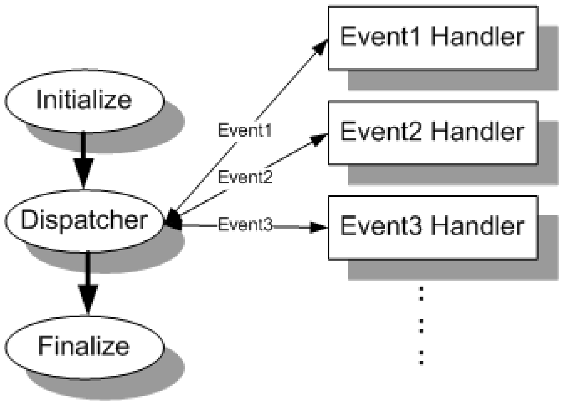


Figure 4.1: Overview of QualNet's event handling.

A typical QualNet protocol overview is given in Figure 4.1. At the beginning, each protocol starts with an initialization routine where the external protocol parameters are read from the config-file and other initialization tasks are fulfilled. The protocol execution is then passed to the event dispatcher which handles all events that are generated during the simulation. The event dispatcher calls an event handler, based on the type of event it processed just then. At the end of the simulation, a finalization procedure is automatically called for each layer in order to print all collected data into the statistic-file which may be analyzed afterwards.

4.1.2 Execution of a Routing Protocol

In the following section, the binding of a routing protocol into the network layer is shortly described. In general, only the interface between the underlying IP protocol and the routing protocol has to be implemented. More precisely the following functions must be supported by a routing protocol to allow communication with the IP protocol:

- *RoutingProtocol*RouterFunction[1]
- *RoutingProtocol*HandleProtocolPacket
- *RoutingProtocol*HandleProtocolEvent
- *RoutingProtocol*MacLayerStatusHandler (optional)

All other functions in the routing protocol are protocol specific. They specify the routing protocol and constitute its functionality. Thus, the whole interaction among the participating protocols (e.g. routing protocol, IP protocol) can be reduced to those functions above. In the following the above procedures are described shortly.

Whenever the transport layer delivers a packet down the network stack to the network layer, or a data packet arrives at the interface of a node, the *RoutingProtocol*RouterFunction is called.

[1] *RoutingProtocol* is just a dummy name for any routing protocol possible here.

Here, the decision is taken whether a packet has arrived at the destination or has to be routed further. As soon as the destination is reached, the *RoutingProtocol*RouterFunction informs the network layer that no further routing is needed and that the IP header can be removed and the packet be delivered to the transport layer. If the *RoutingProtocol*RouterFunction decides that the packet has to be forwarded, it determines the next hop. If a next hop exists, the packet is delivered to the MAC layer. If no next hop is accessible the packet is dropped.

Whenever a protocol packet arrives at the interface of a node, the *RoutingProtocol*HandleProtocolPacket routine is called. Hello-messages in GPSR, or RREQ and RREP in AODV, are examples of such protocol specific packets used to setup routing tables. Therefore, this network traffic has to be handled different than data packets.

The *RoutingProtocol*HandleProtocolEvent routine is used to handle all the events that are node specific, e.g. all the timeouts indicating protocol events, as the broadcasting of hello-messages, or the dequeuing of data packets. All these messages are node specific and do not access the network. Consequently, they do not cause any additional network traffic.

Finally, the *RoutingProtocol*MacLayerStatusHandler function is called from the network layer if anything has gone wrong on the MAC layer, e.g. the packet could not be delivered by the MAC protocol. To decide the further handling of the packet, it may be returned to the routing layer, where new routing decisions are taken, or the packet is dropped. The function is optional, and only used if a further handling of undeliverable packets is wished.

4.1.3 Configuration Scripts

The QualNet simulator is configured by a number of scripts which define the general network settings, the mobility model, the application source, and the resulting output statistics. The scripts used in this diploma thesis are described in the following sections.

default.config

default.config[2] is the main configuration file used in the QualNet Simulator. It supports most of the options to setup a simulation scenario. It includes the general parameters defining the network topology as well as a separate subsection for each network layer. It configures, among other things, the following network properties: simulation area, node density, propagation, antenna and transmission parameters, frequencies, medium access control, routing models, interface queue length, and so on. The descriptions in this section are limited to customized values used in this diploma thesis.

EXPERIMENT-NAME	default
SIMULATION-TIME	1500S
SEED	1
COORDINATE-SYSTEM	CARTESIAN
TERRAIN-DIMENSIONS	(5000, 5000)
SUBNET N16-0	{ 1 thru 5000 }

[2]the term ”default” is just chosen as an example, it may be replaced by any other name.

```
NODE-PLACEMENT                      RANDOM
# NODE-PLACEMENT                    FILE
# NODE-PLACEMENT-FILE               ./default.nodes

MOBILITY                            NONE
# MOBILITY                          RANDOM-WAYPOINT
MOBILITY-WP-PAUSE                   1500S
MOBILITY-WP-MIN-SPEED               1
MOBILITY-WP-MAX-SPEED               20

MOBILITY-POSITION-GRANULARITY       5
```

A hash at the beginning of the according line indicates comments or unused parameters. The experiment name is used to generate the needed output files. The simulation time indicates the continuance of the whole simulation. To enable reproducibility, a seed value is introduced. Therefore, the mobility pattern, the node placement, and so on, remain the same for all simulations using the same network topology. This behavior is necessary in order to enable comparability. In all simulations a Cartesian coordinates system is used. The dimensions of the area are indicated in meters. Furthermore, 5000 nodes are randomly distributed in the network area in the example above. Besides a random distribution, QualNet supports other node placement strategies like uniform distribution or the configuration via file.

QualNet supports several mobility patterns. In our simulations we only use the random waypoint model, or abandon total on mobility in some simulations. Within the Random Waypoint Model, the MOBILITY-WP-PAUSE variable indicates the pause time of a node between two moving activities, whereas the MOBILITY-WP-MIN-SPEED parameter defines the minimal speed a node has to move at and the MOBILITY-WP-MAX-SPEED the maximum one, respectively. The position granularity indicates the granularity after which position updates have to be done. A granularity of 5 means that the position has to be updated every 5 meters. Thus, node positions are discretized. This can cause problems when distance coverage is assumed, but the resulting distance is zero. This may occur, as update moments are correlated to granularity and in the meantime, the positions do not change.

```
PHY-MODEL                               PHY802.11b
PHY802.11b-TX-POWER-DQPSK               7.874
PHY802.11b-RX-SENSITIVITY-DQPSK         -91.0
PHY802.11b-RX-THRESHOLD-DQPSK           -81.0

ROUTING-PROTOCOL                        DDB

APP-CONFIG-FILE                         ./default.app

APPLICATION-STATISTICS                  NO
ROUTING-STATISTICS                      YES
NETWORK-LAYER-STATISTICS                YES
```

QUEUE-STATISTICS	YES
MAC-LAYER-STATISTICS	YES
PHY-LAYER-STATISTICS	YES

Finally, the appropriate physical layer, the routing protocol with its parameters, the application stream used, and the statistics generated in the end of the simulation, have to be defined. The multiple statistics gathered in the end of a simulation are saved in the EXPERIMENT-NAME.stat file which is used to analyze the simulation.

default.app

QualNet offers different application services (such as web browsing, file transfer, telnet) to produce traffic which flows through a network. A short description of the more important available models is given. Only the file format of CBR traffic is described in more detail.

- **FTP** represents the File Transfer Protocol initiated between a client and a server.
- **HTTP** represents a connection between a single-thread web-client and a set of web-servers. The model considers ”think time” between client requests. It further varies numbers of pages, items per page, and size of items, in the server responses. The client also alters the session times during which it makes requests on the same server.
- **Telnet** represents a plain text console connection between server and client.
- **CBR** maintains a Constant-Bit-Rate traffic between a client and a server. Its intention is to simulate multimedia traffic. The file format is as follows:

  CBR <src><dest><items_to_send><size><interval><start time><end time>

  The example listed below indicates that node 1 will send 64 byte packets to node 100. The 0 for <items_to_send> means that an unlimited number of items can be sent. Every second, two packets will be relayed, the first one starting at simulation time 180s, the last one at time 880s.

  CBR 1 100 0 64 0.5S 180S 880S

default.nodes

The placement of the nodes inside the simulation area can be specified using the default.nodes file. Each line defines a new node record. There is a predefined order, the records have to fulfill: First, the unique node identifier is set. The second parameter is a dummy-variable to ensure format consistency with the mobility trace format; it is always set to 0. Within the brackets a triple, containing the x, y, and z coordinate of the node, is defined (in meters). Optionally, one may also define the orientation of the node using the last two floating-point parameters.

NODE-PLACEMENT-FILE

Format: nodeId 0 (x, y, z) [azimuth, elevation]
1 0 (600.0, 300.0, 0.0) 0.0 0.0
2 0 (825.0, 300.0, 0.0) 0.0 0.0
3 0 (850.0, 500.0, 0.0)

4.1.4 Metrics

To enable protocol evaluation as well as comparability to other protocols the metrics which are investigated have to be defined. The metrics listed below are used in the following chapters to fulfill these tasks.

- Packet Delivery Ratio: It is defined as the number of packets received by the destination divided through the number of packets originated by the source node. It specifies the packet loss rate and thus the throughput of the network.

- End-to-End Delay: This value indicates, how long it takes for a packet to travel from the source node to the destination. It represents the average data delay of an application when transmitting data.

- Hop Count: This metric indicates the average path length between the source node and the destination.

- RTS retransmissions: Whenever a packet has to be forwarded, a free medium access has to be granted first. Therefore, a small Ready-to-Send (RTS) packet is sent to the receiving node which is confirmed by the resending of a Clear-to-Send (CTS) packet. If no CTS is received, the operation is done again. If within several attempts no CTS packet was received, the other node is assumed to be unaccessible. The data is gathered on the MAC layer and maps the number of outdated routing attempts.

- Number of retransmitting nodes: This metric is used in the broadcast simulations. It defines the number of nodes retransmitting a broadcast packet in relation to the number of nodes participating in the network. In the worst case (e.g. simple flooding), all nodes retransmit the packet.

- Percentage of network dead: Used in the energy consumption simulations. It defines the moments, when a certain percentage of the network nodes are dead, e.g. the battery power of these nodes has exceeded.

- Average battery power: This metric is defined as the average battery power throughout the whole network at a given time point t. It is only used in battery consumption simulations.

4.2 Mobility Models

The main characteristic of mobile networks is the mobility of their participating nodes. To support realistic behavior, several models have been proposed to simulate the movement of nodes. The movement is determined by the speed a node is moving at, by its direction, and the rate of mobility state changes. A survey of different mobility models is given by [33]. This paper includes the description of the Random Waypoint Model which is used in our simulations.

4.2.1 Random Walk Mobility Model

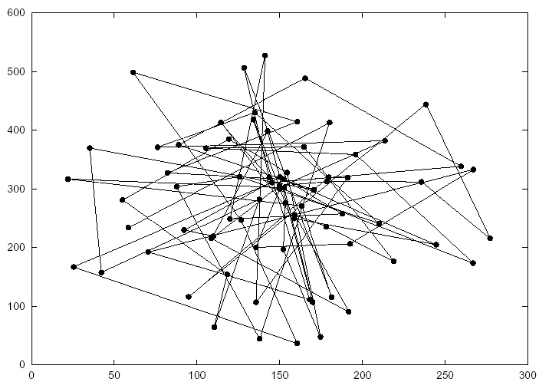


Figure 4.2: Traveling of a node using random walk mobility model [33].

The Random Walk Mobility Model (see Figure 4.2) is based on random directions as well as random speeds. A node randomly chooses a direction between 0 and 2π and a speed between 0 and v_{max}. A recalculation of both values is done, whenever a threshold-based distance was transgressed, or after a predefined timeout. The model is memoryless, which means, no previous data is taken into account when making the new-walk decisions. Thus, the successive movements are totally independent. This may cause unrealistic movements, such as sudden stops or sharp turns. The movement can furthermore be limited to a small area, if the predefined timeout or the distance threshold are chosen short.

4.2.2 Random Waypoint Mobility Model

The Random Waypoint Mobility Model is used in [15], modeling the behavior of Dynamic Source Routing (DSR) under mobility, and was later refined by the same research group [34]. Today, it is by far the most often used mobility model. The model operates as follows. Whenever a state change occurs, a node selects randomly a uniformly distributed position in the simulation area, as well as a moving speed randomly chosen between $[v_{min}, v_{max}]$. v_{min} refers to the minimum speed a node has to move at, v_{max} to the maximum speed a node is allowed to move at, respectively. As soon as a node arrives at the position previously chosen, it rests for a certain time-period (pause-time) before choosing a new destination and a new average speed. In Figure 4.3 you can see the trace of a node moving around the simulation area during a simulation.

According to [35] speed and positions are not uniformly distributed within the original proposed Random Waypoint Mobility Model. They are in fact quite different from a uniform distribution. In particular, it has been shown ([36],[37],[38]) that after a certain amount of time, the distribution of the location of a node is more concentrated in the center of the simulation area.

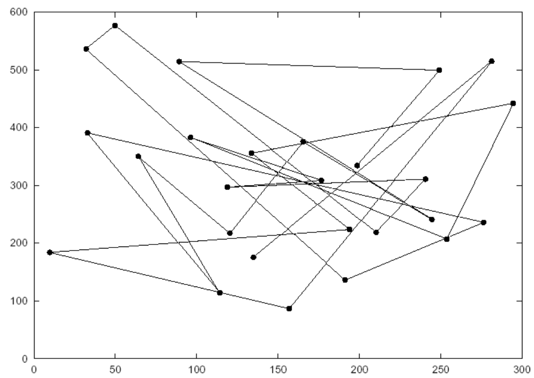


Figure 4.3: Pattern of a node using random waypoint model [33].

This is due to the circumstance that nodes traveling between uniformly chosen points spend more time near the center than near the edges. [36] observed that the distribution of the speed is not uniform, either. If a minimum speed of $0\frac{m}{s}$ is chosen the mean node speed approaches $0\frac{m}{s}$ after a given time of moving, as the mobile nodes become "lazy", traveling long distances at low speed. The simplest solution to avoid this drawback is to set the minimum speed to $1\frac{m}{s}$. A stationary random waypoint mobility model has been implemented by [5]. It ensures the uniform distribution of positions and the speed.

4.2.3 Restricted Random Waypoint Mobility Model

This model was proposed and first used by [39]. In general, it is similar to the Random Waypoint Mobility Model, but it includes several area restrictions. First, a given set of cities within the simulation area has to be defined. The nodes are then randomly distributed among those cities. A node chooses a random position within a city and as soon as it arrives at that destination, it either remains in the same city (i.e. walks to another destination in the city), or it randomly chooses a location within a different city. To do so, links between the cities must exist. Cities and links can be created and removed at any time during simulation. Furthermore, pause times are introduced like in the standard random waypoint model. The Restricted Random Waypoint Model is introduced to deal with more realistic topologies.

Chapter 5

Optimizing Neighbor Table Accuracy of Position-based Routing Algorithms

In this chapter, the impact of wrong routing decisions due to inaccurate neighbor table entries is investigated. All location-based protocols that use hello-messages (beacons) are confronted with the problems mentioned in the following sections. We use GFG/GPSR as the underlying protocol to implement and investigate the refinements to improve neighbor table accuracy we introduce in this chapter. The remainder of the chapter is as follows: First, the problem of inaccurate neighbor table entries is outlined. Second, the general simulation setup is described. In the third part, different forwarding schemes are compared. The upper limits concerning the performance for position-based protocols are identified in the fourth part. Then the enhancements to improve neighbor table accuracy are proposed and evaluated. The chapter ends with the verification and conclusion sections.

5.1 Inaccuracy of Neighbor Table Entries

In MANET, we are confronted with topologies that are not based on fixed infrastructure. Location-based schemes are one particular group of routing protocols operating in such environments. The ability of exchanging data using a location-based routing protocol demands that a given node x must be able to gather the information about its neighborhood from the other participants of a network. Only then, routing decisions can be taken.

Location-based protocols support the need of distributing position information by the periodical sending of hello-messages. These hello-messages contain the current position of a node x and are broadcasted to the one-hop neighborhood of x. Each node receiving a beacon updates its local neighbor table with the position information included in the message. This strategy enables each node in a network to gather information about its local one-hop neighborhood. Consequently, routing decisions, depending on the knowledge of the neighborhood, can be taken. The distribution of hello-messages does not depend on data traffic and is pro-actively maintained over time.

The approach has several disadvantages. The broadcasting of hello-messages leads to a routing overhead which consumes bandwidth and, even worse, blocks data transmission during the sending of hello-messages. Thus, the transmission of data may be delayed. An effect even worse is the inaccuracy of neighbor table entries. As hello-messages are periodically broadcasted, the neighbor table entries can be out of time. This happens whenever a node has moved in the interval between two hello-messages. Let's assume, a node x wants to forward a data message to a next hop y. Furthermore, x had lastly received a beacon from y at time k. The next update will soonest be one beacon interval B later at time $k + B$. If in the meantime y has left the transmission range of x, x may try to forward its packet to a neighbor that no longer exists. A routing entry is only deleted if x has not received a beacon from y until the neighbor dead timeout occurs. The neighbor dead timeout occurs after the expiration of the dead interval D. Then y is removed, as it is assumed to have left the transmission range of x.

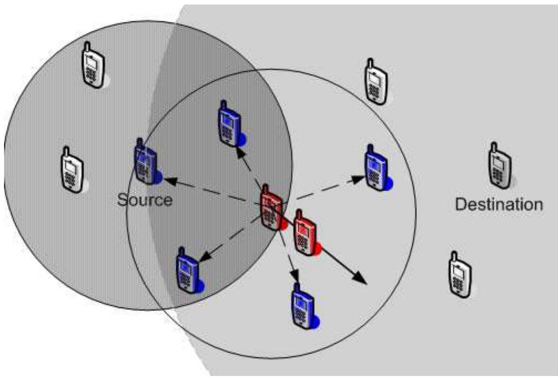


Figure 5.1: Inaccurate neighbor data caused by the moving of a node.

Figure 5.1 illustrates the case of a wrong routing entry. The red node is still in transmission range of the source, when it broadcasts a beacon (dashed arrows). During the beacon interval, the red node leaves the transmission range of the source. The filled arrow depicts the movement of the red node. Now, the source intends to send a message. The forwarding area of the source is the "dark shaded" intersection area between the transmission range of the source and the circle around the destination with the distance between the blue node and the source as radius. As the source expects the red node to be closest to the destination, it chooses the red node as next hop. The red node cannot be reached as it has left the transmission range of the source.

The amount of wrong routing entries is indicated in the following. Assume, nodes move at $20\frac{m}{s}$ and the beacon interval is $2.5s$. The position inaccuracy for routing entries of those nodes may be up to $50m$. Consequently, among the neighbors of a node x, placed in the surroundings of its transmission range, it is probable that a number of these neighbors leave transmission range of x during the beacon interval. In [40] you can see that the percentage of wrong routing entries with a beacon interval of $3s$ and a maximum node speed of $40\frac{m}{s}$ is ~13%. This value is calculated over all neighbors. For GFG/GPSR we expect even worse results because the forward strategy chooses nodes as close as possible to the transmission range as next hops. Exactely these nodes have an increased possibility of being out of date.

The impact of wrong routing entries on the IEEE MAC 802.11b layer [21] is shortly discussed. First, we estimate the induced delay if a routing protocol selects an unreachable next hop. In MAC 802.11, packets are retransmitted up to seven times before the MAC layer assumes the next hop to be unreachable and gives the packet back to the routing layer. For every failed retransmission, the contention window is doubled. Starting at a size of 31, up to a maximum of 1023, times the slot-time of $20\mu s$. Furthermore, the sending node chooses a uniformly distributed back-off time, from the current contention window size, after each RTS failure. After the expiration of that back-off time, the node starts a retransmission attempt. Thus, the expected delay, supposing a maximal number of seven retransmissions, is $\frac{31+63+...+1023+1023}{2} \times 20\mu s \approx 30ms$. Second, additional RTS retransmissions lead to unnecessary bandwidth consumption. Especially in networks with high mobility, we are confronted with a lot of wrong routing entries. The additional RTS retransmissions on the MAC layer, caused by such wrong routing entries, have a deep influence on the routing reliability.

These observations substantiate the need of more accurate neighbor table entries. A number of possible enhancements are described and evaluated in later sections. The first approaches are based on more appropriate beacon interval choices. More precisely, beacon intervals are assigned to the moving behavior of a node. Another strategy is realized by expanding hello-messages with the information needed by other nodes to draw conclusions about the future position of the sending node. A last proposed version to minimize routing entry inaccuracy is to avoid a pro-active neighbor table setup mechanism and, instead, to make the GPSR reactive.

5.2 Simulation Scenario and Parameters

The simulation setup, as long as nothing else is mentioned, is the following: All simulations are done with 400 nodes moving according to the Random Waypoint Model. Speed is randomly chosen between [$1\frac{m}{s}$, $40\frac{m}{s}$]. The simulation area is always $600m \times 3000m$. We simulate one CBR source which starts to send at simulation time $180s$ and stops at simulation time $880s$. Each configuration runs with eight different seed values to obtain statistical relevance. The confidence intervals are not calculated and therefore not shown.

The GPSR unaltered version used by [1] is referred to as GPSR_Karp. All parameters are chosen according to the values mentioned in Chapter 3.5.1. GPSR_Karp choses the best neighbor as the neighbor closest to the destination but still within transmission range. This forwarding scheme is called Most Forwarding within Radius (MFR) [18].

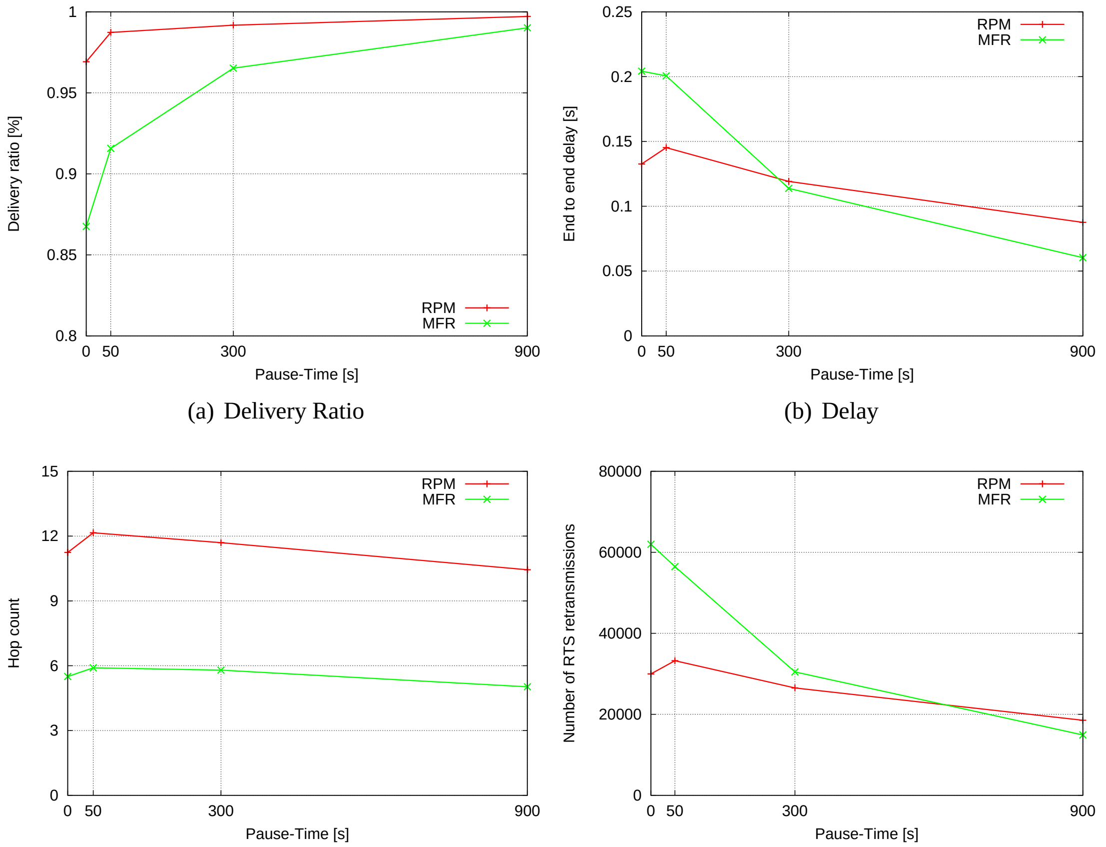
In most of our refinements the beacon interval B is determined by the movement of a node. The dead interval D in those simulations is once $4.5B$ and once $2B$. However, a beacon is released if after $5s$ no beacon send event occurs. For similar reasons, the removement of a neighbor table entry is done by the last after $10s$. The neighbor table update refinements are only simulated with a pause time of $0s$, whereas in the other simulations (optimum, forward decision) pause times of $0s$, $50s$, $300s$ and $900s$ are simulated.

5.3 Forwarding Strategy

Beside the MFR forward strategy, two other schemes are presented. The Random Progress Method (RMP) is introduced by [41]. This scheme choses a random neighbor closer to the destination as next hop. The second scheme is called Nearest within Forwarding Direction

(NFP) [19]. Here, the neighbor closest to the releasing node but still with progress to the destination is chosen as next hop.

With the simulation of these forwarding schemes we hope to see that the MFR forwarding strategy leads to an unproportional amount of RTS retransmissions, because neighbors as close as possible to the destination are chosen. All three forwarding schemes are simulated based on the GPSR_Karp protocol settings. Only the forwarding strategy is replaced once with the NFP and once with the RMP scheme.



(a) Delivery Ratio (b) Delay

(c) Hop Count (d) Hop Count

Figure 5.2: Best neighbor strategies.

In Figure 5.2 the results of our simulations are shown. The results with the NFP are not depicted. The density of the simulated network leads to hop counts which often exceed the time to live of the IP packet header and the packet thus is dropped. Consequently, the NFP performs bad in our dense network. In contrast to the NFP approach, the RPM scheme behaves better than the standard MFR. The delivery ratio as well as the end-to-end delay is better, at least under high mobility. The RMP strategy improves the delivery ratio up to 97% with a pause time of 0, which is rather surprising. The influence on the number of retransmissions supports the assumption mentioned above: The number of retransmissions is high with the MFR strategy. The number is halved with the RMP strategy. This is obvious, as with the random choice of a neighbor fewer nodes located near the transmission range boundary are chosen. Thus, the probability of chosing a neighbor that has left the transmission range decreases. Consequently, the number of retransmission attempts is smaller. The results show that the average hop count is doubled with the RMP forwarding strategy. The random neighbor choice is uniformly distributed among the neighbors with progress. Thus, the neighbor closest to the half transmission range is on the average chosen. This perfectly fits the result of a doubled hop count and explains the halved number of retransmissions.

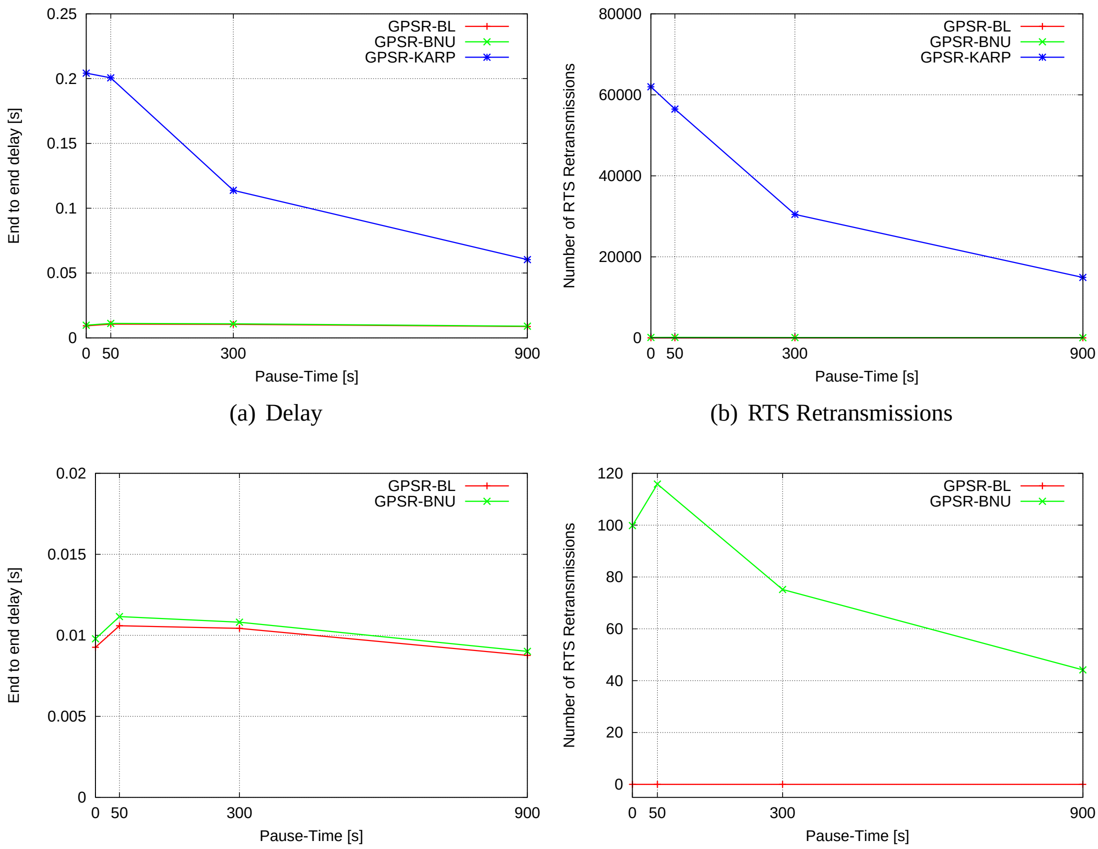
Despite these results, we perform all remaining simulations with the MFR strategy. The RMP has the advantage of choosing more often nodes that are farther away from the transmission boundary, which increases performance. However, the inaccuracy of neighbor table entries is not handled.

5.4 Optimum of Location-based Routing

In order to have a measurement on how good our refinements are, two versions of GFG/GPSR are implemented that never make wrong routing decisions. Both versions use the global simulator data to determine next hops. As that data is always up-to-date, wrong routing decisions depending on neighbor table inaccuracies are avoided. The first version is called Beacons-Not-Used (GPSR-BNU). The hello-messages are still sent, but not used to determine next hops. In the second version, the whole beaconing mechanism is disabled. This is called Beacon-Less (GPSR-BL). With both versions we hope to see the influence of the hello-message sending mechanism on network performance. The delivery ratio and the average end-to-end delay should be affected in particular. The node density is high enough to ensure total connectivity within the network. Furthermore, no wrong routing decisions can occur. Consequently, the delivery ratio is 100% for both versions. The delivery ratio is not depicted as it is only a consequence of wrong routing decisions and not a cause.

The average end-to-end delay as well as the number of retransmitted packets on the MAC layer are depicted in in Figure 5.3. The delay for GPSR-BL and GPSR-BNU is always around $10ms$, which is much lesser than simulated with the GPSR_Karp. GPSR_Karp has end-to-end delays between $60ms$ and $210ms$ and up to 60'000 RTS retransmissions. The number of RTS retransmission attempts on the MAC layer correlates thereby to the average end-to-end delay. An increased number of RTS retransmissions causes a higher end-to-end delay.

The delivery ratio as well as the delay is an upper, respectively a lower boundary for all the protocol refinements in the following sections. The results indicate that wrong neighbor table entries are the most important factor for high end-to-end delays. The low number of RTS retransmission for both, the GSPR-BNU and the GSPR-BL, simulations and the accordingly low end-to-end delays support this assumption. The few retransmissions when using the GPSR-BNU protocol can be explained by collisions between hello-messages and RTS packets. The 10% higher delay of GPSR-BNU compared to GPSR-BL, is a consequence of the additional delay caused by the RTS retransmissions. The results gathered so far support our thesis that inaccurate neighborhood table entries are the most important factor of packet loss and high end-to-end delay.



(a) Delay (b) RTS Retransmissions

(c) Delay (d) RTS Retransmissions

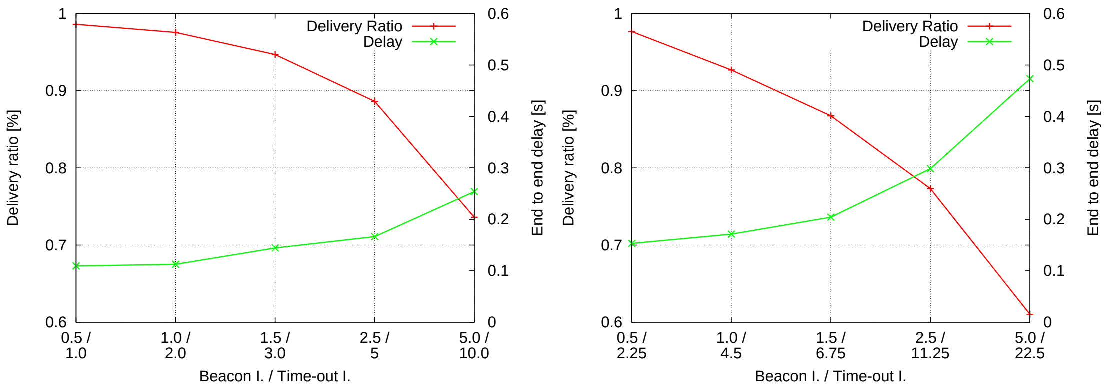
Figure 5.3: Simulations with correct neighbor decisions.

5.5 Beacon Frequency Strategies

5.5.1 Time-based Beaconing Strategy

In the standard time-based beaconing strategy the influence of a fixed beacon interval B and a fixed dead interval D is investigated. Consequently, all nodes have the same beacon and dead intervals. The speed of a node or its direction have no influence on the scheduling of its beacon broadcasting time points. We expect two main disadvantages that are strongly related for this beaconing strategy, namely the inaccuracy of the neighbor table entries and the disregarding of the speed of a node. [1] used in their paper a beacon interval of $1.5s$ and a dead interval of $6.75s$. We compared simulations with these parameters to other settings, as shown in Figure 5.4.

The simulation results indicate that a smaller beacon interval increases the reliability of the network. The choice of a shorter dead interval increases the delivery ratio and decreases the end-to-end delay, too. The best results are achieved with $B = 1s$ and $D = 2B$. The augmentation of the values for the beacon interval and the dead interval degrade performance



(a) $D = 2 \cdot B$ (b) $D = 4.5 \cdot B$

Figure 5.4: Different ratios of beacon and dead intervals.

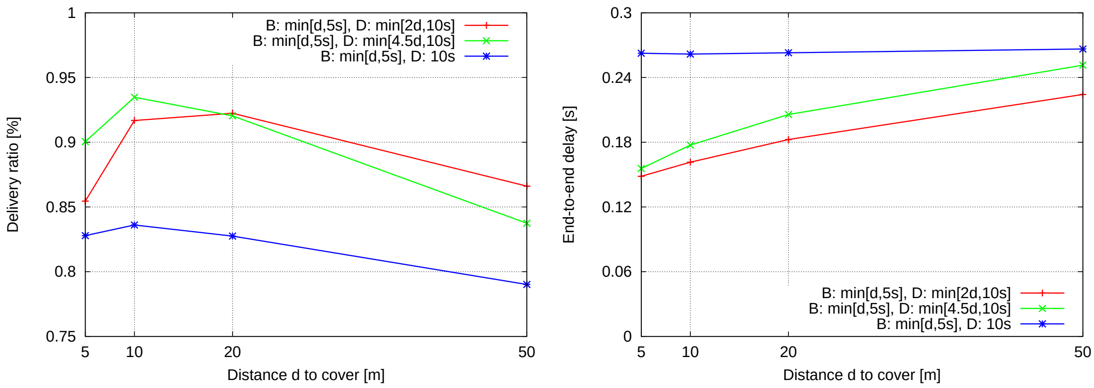
proportionally. A scaling down of the intervals raises an additional network load caused by the increased frequency of hello-messages broadcasted. This has no influence in our scenario where only one CBR source is used and congestion and number of collisions are thus minimized. [1] used in their simulations 30 CBR sources. This leads to much more traffic and may be the reason why they proposed a beacon interval of $1.5s$ and a dead interval of $6.75s$ as the most appropriate settings. The results with $B = 5s$ and $D = 10s$ are the leveling board for our refinements, as those values indicate the time points after which at latest an event must occur. In these simulations a delivery ratio of approximately 74% and an end-to-end delay of $270ms$ are achieved.

5.5.2 Distance-based Beaconing Strategy

In the Distance-based strategy a beacon is sent, whenever a given distance is covered by a node, or a timeout occurs. In order to discretize the observation of the distance coverage, we implemented a timer. This timer is reset every $0.5s$ and checks if the node has moved more than the distance threshold d in the meantime. To ensure that each station sends a beacon from time to time an absolute timeout of $5s$ is introduced. After this a beacon is automatically sent. This is the case if a node has not covered the distance in the meantime.

To determine the dead interval D, we introduce two different methods. In the first solution, D has a fixed value of $10s$ for each entry in the neighborhood table (see graph 3 in Figure 5.5). In the second approach, a node deletes an entry in its neighbor table if it has moved more than k times d or at latest after $10s$. Consequently, the dead interval is chosen as the minimum of $[k \cdot d, 10s]$. To enable this feature, a node has to save its position at the arrival of a beacon. With this information it frequently calculates the distances it has moved since it has last received the hello-messages from its neighbors. Thus, a node can easily determine if it has covered a distance bigger than $k \cdot d$ and can remove the neighbor table entry accordingly. The value of k is fixed. Once it is 2 (graph 1 in Figure 5.5) and once 4.5 (graph 2 in Figure 5.5).

With the distance-based mode, we hope to make our neighbor update strategy more correl-



(a) Delivery ratio

(b) End to end delay

Figure 5.5: Reliability with distance-based beaconing strategy.

ative to the movement of nodes. Nodes that move very fast send a beacon frequently, whereas nodes that move slowly and therefore have a little position inaccuracy send hello-messages less frequently. The distance-based scheme has one major disadvantage: As slow nodes only infrequently send hello-messages, fast moving nodes are likely not to hear them. Or if they hear them, they keep them only for a short time interval in their neighborhood table.

You can see the simulation results in Figure 5.5. All distance-based strategies have better delivery ratios than the time-based strategies (see Figure 5.4) where the delivery ratio with $BI = 5.0s$ and $DI = 10.0s$ is approximately 74%. The delay is only improved if the dead interval is determined in correlation to the distance covered by a node. We get best results when a distance threshold d of $10m$ is choosen and the neigbor daed interval is the minimum of $[4.5d, 10s]$. In those simulations the delivery ratio is improved from $\sim$ 78% to $\sim$ 94%. The delay is decreased only little from $200ms$ to approximately $180ms$. The slight improvement concerning the end-to-end delay is caused by the more frequent sending of hello-messages. A fast moving node, e.g. $40\frac{m}{s}$ sends four hello-messages per second. Consequently, the network load is increased and the end-to-end delay suffers accordingly.

5.5.3 Speed-based Beaconing Strategy

In the speed-based mode the beacon interval B, as well as the dead interval D, is correlated to the speed a node is moving at. Furthermore, B can be determined using discrete thresholds, or it is calculated continuously within predefined interval boundaries. D is calculated depending on the beacon interval, e.g. $D = kB$, where k is a positive integer. A node sends its calculated value of D within a beacon to its neighbors. All receiving nodes determine their final dead interval as the minimum of the dead interval they received and the dead interval they calculated according to their own speed. With this enhancement, we hope to satisfy the drawback from the distance-based approach, where the determination of the dead interval between two nodes moving at different speed is not solved satisfactory.

First, we introduce two continuous functions to map the speed of a node on its beacon interval. The first function is linear (5.1) and the second one is polynomial (5.2). Both functions are defined over the basic set between $[1s, 5s]$. Their equations have to be adapted according to the following constraint:

$$B = 1 + \left(\frac{4 \cdot (v_{max} - v)}{v_{max} - 1} \right) \tag{5.1}$$

$$B = 1 + \left(\left(1 - \frac{v}{v_{max}} \right)^4 \cdot \frac{4}{\left(1 - \frac{1}{v_{max}} \right)^4} \right) \tag{5.2}$$

where B is the beacon interval, v is the current speed of the node, and v_{max} is the maximal speed a node can move at. The functions ensure that the calculated beacon interval is always within range $[1s, 5s]$.

Speed $[\frac{m}{s}]$	Beacon interval [s]
$1 - 5$	5
$5 - 10$	3
$10 - 20$	2
$20 - 40$	1

Table 5.1: Speed mapped on beacon interval.

Besides the continuous calculation of the beacon interval B, we introduce a discrete calculation. The mappings in Table 5.1 are used to assign the beacon interval of a node to its current speed. The dead interval is accordingly determined as with the continuous function, $D = kB$. The resulting dead interval D at the receiver of a beacon is the minimum of the sender's and receiver's dead intervals.

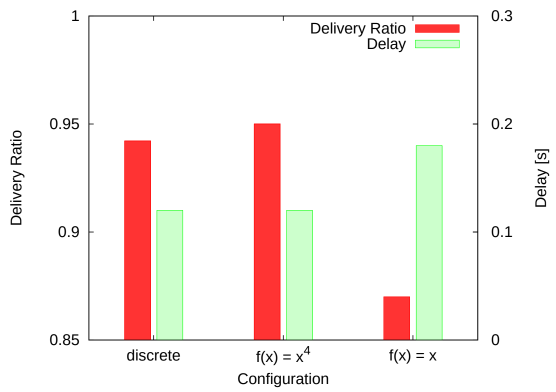


Figure 5.6: Speed-bsed modes.

The results for the simulations using these three configurations are depicted in Figure 5.6. Ten seconds are chosen as the dead interval, i.e. $k = 2$. The strategies with the discrete as well as the polynomial function perform very well, whereas the strategy with the linear function is only slightly better than the standard time-based mode. The worse performance of the linear mode can be explained by its wrong assumption of the average speed of a node. More precisely, the beacon interval is uniformly chosen from the interval $[1s, 5s]$, accordingly to the current speed of the node. In the Random Waypoint Model, the average speed is about $10\frac{m}{s}$ [36]. Consequently, most nodes move more slowly than the arithmetic middle of the simulation, which is $20.5\frac{m}{s}$. The reason for this behavior is the faster arrival of nodes moving at higher speeds at their randomly chosen destinations. This leads to a minor average speed in the simulation. The linear function, however, does not account for that behavior.

In the discrete and the polynomial configuration, the delay is reduced to about 100ms and the packet delivery ratio is increased to 94%, which are promising results compared to the 74% delivery ratio and the $270ms$ end-to-end delay (see 5.5.1). The speed-based performance is even much better than the GPSR_Karp simulations which achieved a delivery ratio of $\approx 87\%$ and an end-to-end delay of about $210ms$ with pause time $0s$.

5.5.4 Link-Break-based Beaconing Strategy

In [42] two slightly different modes to determine the number of link-breaks were introduced: an absolute connectivity-based approach as well as a percentage connectivity-based one. Both modes maintain a counter indicating the number of link change occurrences between the sending of hello-messages. This counter is increased whenever a new neighbor appears or is deleted. The counter is reset whenever a beacon is sent. Both modes differ in the method they use to schedule a beacon sending event. Within the absolute connectivity-based approach a hello-message is sent whenever a fixed number of link changes have occurred. The Connectivity Percentage-based approach on the other hand releases a hello-message whenever a fixed percentage of link changes in correlation to the total number of neighbor table entries have occurred. The upper bound of the beacon interval B is fixed to $5s$ to ensure that a beacon is at times released even if not enough link-breaks have occurred. The dead interval D is according to the other refinements $10s$. Additionally, a dead interval strategy similar to that used in the distance-based approach is

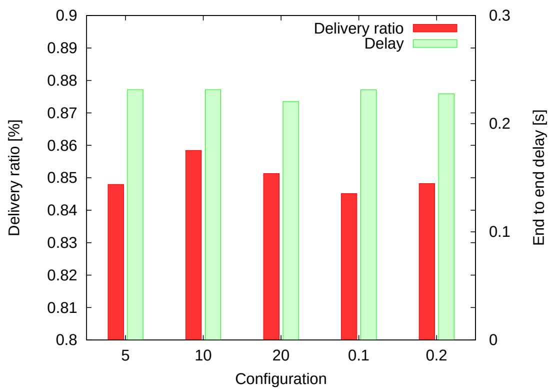


Figure 5.7: Link-break-based beacon strategies.

supported. Thereby, the dead interval is in correlation to the number of link incidents. However, this strategy has been discarded, as the associated calculations lead to recursions and cause a massive simulation time consumption. The problem is: whenever a link break occurs, we have to check if some neighbors in the table have to be deleted. If so, each of these neighbors indicate a new link break occurrence, and so on. As we simulate mobile and dense networks, these routines consume too much memory and can not be considered.

In Figure 5.7 you can see the results we got with the link-break-based beacon strategy. The dead interval is $10s$ for all simulations. The first three simulations are done using the absolute connectivity-based mode with absolute link-break thresholds of 5, 10, and 20 link-break occurrences. The last two simulations are based on the percentage connectivity-based mode. 10% and 20% of link changings relatively to the absolute number of neighbors must occur to indicate a beacon release. The results show that all simulations have about the same end-to-end delay of $240ms$. This value is more or less as bad as the end-to-end delay of the according Time-based simulation ($\approx 270ms$). Whereas the delivery ratio is improved from $\sim$73% to a maximum of $\sim$86%, with a fixed threshold of ten link-breaks. The simple link-break-based approaches are a little better than the simple distance-based approaches (see Figure 5.5).

5.6 Receiving-Power-based Update Strategy

GPSR_Karp always choses the neighbor closest to the destination (MFR). This behavior of greedy forwarding leads to the problem that, whenever possible, a node very close to the transmission boundary is selected. Exactly these nodes, however, have a much higher possibility of being out of transmission range than the others. Furthermore, if a node is still within the transmission range, it nevertheless may be unreachable due to radio irregularity. In reality, transmission ranges are irregular because of obstacles and interferences. Introducing a receiving power threshold RX we can artificially create a circular gray-zone close to the transmission area boundary where nodes are not allowed to receive packets. The information about the receiving power can be used to cope with non-isotropic transmission ranges on the one hand and wrong neighbor table entries due to beacon inaccuracy on the other. The RX-threshold excludes all nodes from routing that receive a hello-message on a power less than the minimal required receiving power RX. Consequently, only nodes with a signal strong enough, e.g. nodes nearer to the relaying node, are considered as neighbors. The Phy802.11 standard uses an RX-threshold of -81dBm. This value determines the transmission range of a node. Receiving signals with a signal strength below that value cannot be decoded. The GPSR_Karp protocol settings were used to test the RX-threshold neighbor table update strategy.

We investigated several RX-thresholds in our simulations. The refinement performance is depicted in Figure 5.8. The results indicate that the introduction of a receiving gray-zone close to the transmission boundary improves the protocol performance. We achieve best results with a RX-threshold of -79dbm that is $2dBm$ higher than the Phy802.11 threshold. The delivery ratio is increased about 7% to almost 94% and the end-to-end delay is diminished $90ms$ to approximately $120ms$. However, the cumulation of the performance reaches its climax with a RX-threshold of $-75dBm$. Afterwards, the delivery ratio begins to deteriorate again. This behavior is predictable. Incrementing the threshold ”decreases” the transmission range of a node. Thus, less neighbors exist within the remaining transmission area. A disadvantage of the

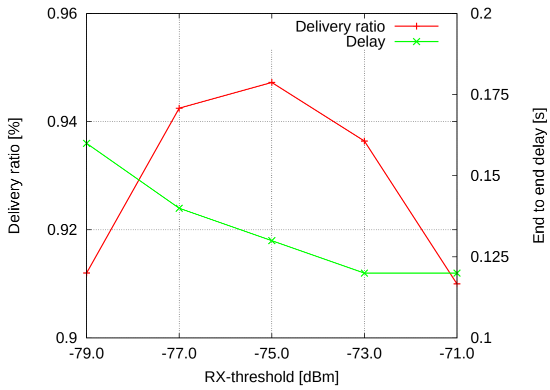


Figure 5.8: Protocol performance, if the receiving power sensitivity constricts the transmission power.

current RX-threshold based beaconing strategy is its inability of predicting the most appropriate threshold. The choice of a good threshold depends on the speed of a node. If a node is moving fast, it should only add close neighbors in its neighbor table. Consequently, the node should chose a higher RX-threshold. Whereas neighbors farther away may be accepted if the node moves slowly. Similar to the speed-based strategy the speed of a node could be mapped on its RX-threshold to solve the problem.

5.7 Prediction-based Next-hop Decision

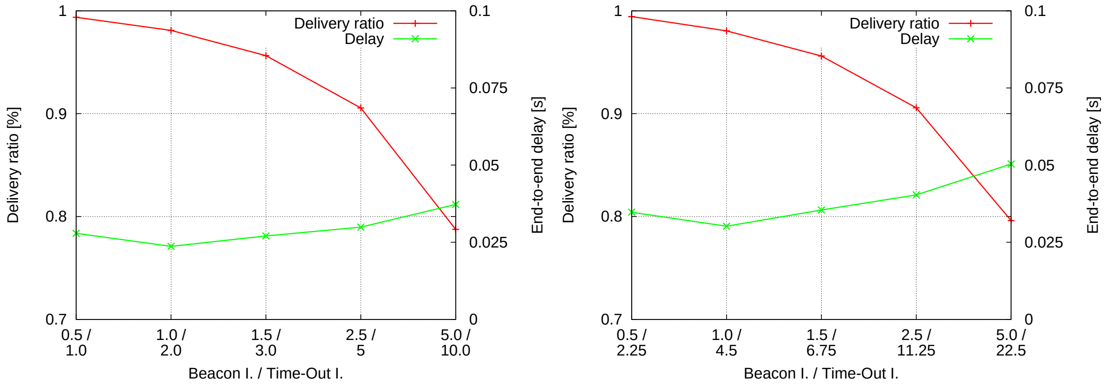
To deal with the problem of inaccurate neighbor table entries, we use the assumption that a node is moving at a certain speed in a fixed direction. In the majority of all cases, a node maintains speed and direction for a time interval long enough to enable the near future prediction. Therefore, we have to expand the neighbor table entry with two additional informations: the current speed a node is moving at and the direction it is moving along. Furthermore, each entry is labeled with the time t_r when the beacon was received. Thus, whenever a routing decision has to be done at a node x, x is able to calculate the prospective positions P_p of a neighbor y for all neighbors. With this enhancement, we hope to reduce wrong routing decisions and improve delivery ratio as well as end-to-end delay. In equation (5.3) you can see, how a node x calculates the position of a neighbor y, dependent on the information it gathered in its neighborhood table:

$$P_p(y) = P_n(y) + \overline{v}(t_c - t_r) \tag{5.3}$$

$$|P_p(y) - P_s(x)| > r \tag{5.4}$$

where P_n is the position of node y, entered in in the neighbor table, when last a beacon was received from y, t_c is the current time when the calculation is done and P_s is the current position of the releasing node x. Assuming a circular transmission range r, then a neighbor is no longer reachable if the distance $\overline{P_s P_p} > r$ (equation 5.4).

In Figure 5.9 we can see the results we get for different beacon intervals and dead intervals. We used the same values as for the Time-based strategy. Compared to the Time-based



(a) $D = 2 \cdot B$ (b) $D = 4.5 \cdot B$

Figure 5.9: Different ratios of beacon and dead intervals.

approach, the end-to-end delay is at least five times better for each configuration. The delivery ratio is also much better with prediction. We got a delivery ratio of about 97% which is a big improvement compared to the 87% we obtained with the time-based configuration ($BI = 1.5s$, $DI = 6.75s$). The same is true for the average end-to-end delay of approximately 40ms compared to the 210ms with the time-based strategy. Both results are traceable. The better delivery ratio is obviously due to the improved neighbor knowledge. This advancement leads to fewer wrong routing decisions, which is the reason for the minimized end-to-end delay.

5.8 Reactive GPSR

[1] mentioned the possibility of making GFG/GPSR reactive. That means GFG/GPSR generates the neighbor tables only when a node has to transmit data packets. The neighbor knowledge gathering mechanism is started by a sending node that broadcasts a beacon request packet (GPSR_BR) to show its sending disposition. Each node overhearing this request transmits a beacon after a short random jitter to avoid simultaneity and interference at the receiver. The source node introduces a beacon gathering interval G which is k times the maximal jitter interval of the responding nodes. After that interval, the node forwards the data packet to the best neighbor it received a beacon from. Afterwards, the source deletes its neighbor table and the whole process is repeated at the next hop. Using GPSR reactive, we hope to achieve good performance, as a node operates on almost accurate neighbor information. The interval between a beacon request and the effective sending of a packet is very small. Consequently, the positions in the neighbor table are quite exact, and thus, the number of out-dated neighbors is minimized. A disadvantage is the cumulative latency gained by the additional beacon gathering interval G on each hop. However, G is shorter than the time consumption each wrong routing entry adds. Consequently, as soon as we assume at least one wrong routing decision per hop using the normal GFG/GPSR and the accuracy of neighbor positions using the reactive GPSR, the reactive GPSR should perform better than GFG/GPSR.

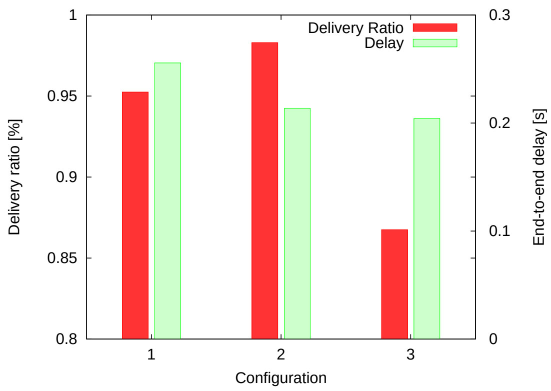


Figure 5.10: Reactive GPSR.

For the current evaluation, we have chosen the following parameters: All nodes receiving a GPSR_BR packet jitter their beacon answers within $1ms$. The source node which broadcasts the GPSR_BR has a beacon gathering interval G of $15ms$ (15 times the jitter interval of $1ms$) in the first configuration and a G of $10ms$ in the second. Subsequently, it selects the node closest to the destination among the nodes it has received a beacon from. The third configuration in Figure 5.10 is the GPSR_Karp protocol.

The reactive GPSR version (Figure 5.10) achieves an average delivery ratio of 98% and an average end-to-end delay of $220ms$ with a beacon gathering interval of $15ms$. The reactive GPSR increases the delivery ratio more than 10%, whereas the average end-to-end delay is even a little worse than in the GPSR_Karp implementation. The results are promising, especially if we consider that the current reactive GPSR version is a very basic one, where no optimizations are implemented.

5.9 Verification

First we calculate the additional delay caused by wrong routing entries. We concentrate on the standard GPSR implementation with a beacon interval of $1.5s$ and a dead interval of $6.75s$. Figure 5.2 shows that approximately 62'000 RTS retransmissions are done on the MAC layer. Each undeliverable packet adds seven retransmissions. Thus, if we assume that all RTS retransmissions relate to non-existing neighbors we have almost 9000 packets scheduled back to the routing layer. Each of these wrong routing decisions causes an additional delay of approximately $30ms$ (see Section 5.1). Furthermore, 1400 data packets are transmitted during a simulation. Thus, we obtain an additional end-to-end delay which is caused by the RTS retransmissions of $\frac{9000}{1400} \times 30ms \approx 192ms$ per packet. Figure 5.3 indicates an average end-to-end delay of $10ms$ for simulations with correct neighbor decisions, e.g. without RTS retransmissions. Consequently, we obtain a resulting end-to-end delay of $192ms + 10ms = 202ms$ which corresponds to the measured delay of approximately $210ms$ in Figure 5.4.

Second, the theoretical assumptions done in [40] indicate the average number of out-dated neighbors. Expecting a speed interval between $[1\frac{m}{s}, 40\frac{m}{s}]$ and a dead interval of $6.75s$, approximately 30% of all neighbors should be out-dated. The average hop-count in all simulations (excluding those with RMP decisions) is around six hops. Furthermore, 1400 packets are transmitted within one simulation. Under those conditions $\frac{9000}{6 \times 1400} \approx 1.07$ wrong routing decisions are taken per hop before the packet is delivered to the next hop. This equals to a 51% probability of choosing a wrong routing entry. This is bigger than the expected 30% wrong routing entries. We explain this difference mainly by the assumption already given in Section 5.1 saying that greedy routing selects nodes close to the radio range boundary. Furthermore, those nodes have an increased probability of having left the radio range during the current beacon interval. Thus, the augmented possibility of choosing a wrong neighbor table entry in the transmission boundary environment is explainable.

The results collected in the evaluation chapters fit to our theoretical assumptions. Therefore, all the different GFG/GPSR simulations seem to supply appropriate results.

5.10 Conclusions

In the beginning of the chapter, we suspected a strong relationship between inaccurate neighbor tables and the reliability of location-based routing protocols. Furthermore, we estimated the probability that the routing protocol selects an unreachable node. This indicates that wrong routing decisions happen very often, depending on the accuracy of the neighbor table entries. Factors amplifying the inaccuracy are small transmission ranges, long beacon intervals, and high node mobility.

The simulations confirm the analytical assumptions. They show that the end-to-end delay is increased up to a factor of 10 accordingly to the inaccuracy of neighbor table entries. The delivery ratio is also badly influenced by wrong routing decisions. To gain those insights, we first demonstrated that in a uncongested network with correct routing decisions (see Section 5.4) no packet-loss occurs and an end-to-end delay of about $10ms$ can be expected. In a second step, we have shown that the MFR forwarding strategy adds a high possibility of wrong routing decisions. The RMP forwarding approach on the other hand indicates much better results, even when the average hop count is doubled. Thus, we conclude that wrong routing decisions which cause a lot of RTS/CTS traffic have a much bigger influence on the end-to-end delay than the number of hops. The RMP forwarding strategy is one possibility of gaining better network performance. It is a displacement of the problem and we prefer solutions that enhance the reliability of the distributed location information.

Among the proposed refinements to improve the accuracy of the neighbor tables very different results are achieved. Nevertheless, all of them improve the delivery ratio as well as the delay compared to the standard GPSR. The best results we achieve with the prediction mode, where the end-to-end delay is improved approximately five times and the delivery ratio reaches up to 100% for certain scenarios. The speed-based scheme as well as the reactive GPSR show very good delivery ratios, too. The average end-to-end delay is not as good as with the prediction mode.

The results achieved with GPSR-BL and GPSR-BNU indicate that improvements of the standard GPSR are possible and worthwhile. The refinements proposed so far improve the reliability of the network but are still far away from the theoretical optimum.

It is possible to combine the prediction mode with an improved beacon strategy to reduce routing overhead. Nevertheless, the advantages of prediction remains. Thus, the routing protocol takes the movement of the nodes into account and estimates its neighbors positions. We have not yet done those simulations as we investigate the influences of our refinements independently.

At the time, the reactive GPSR implements no kinds of optimizations. However, we expect better end-to-end delays by implementing enhancements that incremental gather information from areas not yet known. Those refinements could make the reactive GPSR protocol appropriate, especially for low traffic scenarios as it eliminates the proactive broadcasting of hello-messages. One possible application area are sensor networks, where traffic is transmitted seldom. Furthermore, the requirements on low energy consumption could thus be satisfied. On the other hand the prediction-based mode could be chosen accordingly to the much shorter delays for delay critical applications. Furthermore, it could be enhanced with a speed based strategy. If possible, the beacon interval should be chosen as short as possible in order to minimize the risk of out-dated neighbor entries. This may not be possible in very dense networks, where the frequent exchange of beacon messages would cause too much routing overhead which could badly influence the network reliability.

It is not the aim of the current work to obtain best performance parameter settings for the given protocols. We rather want to determine the potential of the implemented refinements. The main goal is to show the impact of wrong routing decisions due to inaccurate neighbor table entries. Thus, we do not simulate the current implementations under higher traffic or other node densities. All those tasks as well as combinations of the proposed refinements could be done in future works.

Chapter 6

Dynamic Delayed Broadcasting Protocol for Mobile Ad-hoc Networks

In this chapter another area-based broadcasting scheme is proposed. That scheme is based on the dynamic forwarding delay strategy (DFD) introduced in the BLR routing protocol. The benefit of area-based schemes is their independence of any neighborhood information. The dependency of the neighborhood knowledge approaches on hello-messages may cause inaccurate position information as well as additional bandwidth and energy consumption. Therefore, those approaches are not appropriate for highly mobile networks, or energy and bandwidth critical systems.

6.1 Problems of Area-based Methods

[25] shows in extensive simulations that area-based schemes degrade disproportionally in dense or congested networks. This behavior is caused by their inability of minimizing the number of rebroadcasting nodes. [25] proposes two improvements to deal with those disadvantages of area-based protocols. The first one is the insertion of a neighbor count that depends on neighborhood information. Consequently, the distribution of neighbor information is necessary, which removes the advantages of area-based approaches. The second suggestion is to add a congestion level on each node that determines the RAD for that node.

We assume the random calculation of the RAD as another limiting factor. It is not appropriate, as the location of the node is not taken into account. Nodes located at the transmission border should schedule a shorter RAD than nodes closer to the relaying node. An area-based broadcasting algorithm supporting the position-based RAD calculation is proposed in the following.

6.2 Dynamic Delayed Broadcasting Protocol (DDB)

Despite the disadvantages of area-based broadcast schemes, we propose another broadcast protocol depending on the progress of a node. It is defined as an assumption of how much additional area a node is supposed to feed. We propose three different metrics to estimate the progress of a node. These metrics are discussed later. Whenever a packet is broadcast all nodes located within transmission range of that node calculate a Dynamic Forwarding Delay (DFD). The DFD thereby depends on the current progress of a node. The packet is buffered for that DFD and broadcast if no interruption by an earlier sending node occurs in the meantime. If a duplicate of the buffered packet is received during the expiration of the DFD, it is recalculated. Our protocol supplies three refinements to the location-based scheme proposed by [24].

The main improvement is the calculation of the DFD in correlation to the progress of a node. Furthermore, the DFD is recalculated dynamically. Thus, nodes that are covered by other nodes should be starved out iteratively.

A second refinement is the possibility of a node to remove packets from its network queue. When a packet is scheduled to be broadcast, but almost coeval another node relays a copy of that packet, the packet is removed from the network queue and further handled by the broadcast protocol. We call this refinement Cancel on MAC. It is discussed in detail later.

The last refinement we implemented suppresses the threshold decision if no message is buffered. Thus, the neighbor farthest away of a node broadcasts the packet at any rate, not depending on the progress it covers.

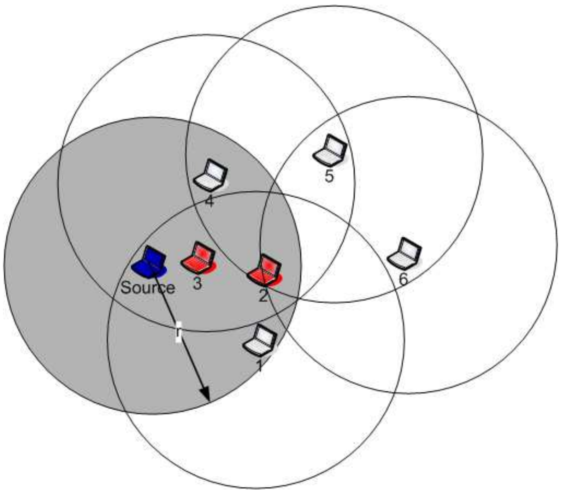


Figure 6.1: Broadcasting with Dynamic Forwarding Delay.

In Figure 6.1 you can see how DDB works. The source node intends to broadcast a packet. All nodes within its transmission range receive the packet and calculate their DFD. Node 1 is the first node that rebroadcasts the packet as it is farthest away from the source (node 1 has most progress and consequently calculates the shortest DFD). Node 2 and 3 hear that relay and recalculate their DFD. It is important to remark that their DFD increases when their progress decreases. As node 4 did not hear the initial broadcasting of the source, it calculates the shortest DFD and is the third node that broadcasts. Node 2, 3 and 5 hear that rebroadcasting. Node 3 drops the packet immediately, as it adds no progress anymore, whereas nodes 2 and 5 recalculate their DFDs. Node 5 is the next node that broadcasts because its DFD is much shorter than that of node 2. Finally, node 6 rebroadcasts, as its progress is by far bigger than that of node 2. Node 2 registers that transmission and drops its packet, as its progress becomes zero. The dynamic recalculation of the DFD allows DDB to operate similar to the perimeter mode in GFG/GPSR. Nodes that are covered by other nodes are circumvented and starve out.

6.2.1 Progress Schemes of DDB

The progress of a node is its assumption of how much additional area it reaches with the relay of a packet. We assume that the more additional area a node feeds the higher is the number of neighbors covered in that transmission. In the following three different metrics to determine the additional area are introduced. The distance and the area metric depend on the knowledge of the position of the last hop. Thus, both require a location service. The signal strength metric in contrast maps the additional area directly from the received signal strength. Consequently no location information is needed.

Distance

One metric to determine the progress of a node is its distance to the node it receives a packet from. To do so, each broadcasting node adds its current position in the message header. A receiving node is able to calculate the distance between itself and the sender according to its own position and the position it gets with the packet.

During the DFD expiration of a node, it may receive multiple copies of a broadcast packet. Consequently, it calculates its distance to all those neighbors. Furthermore, it saves the shortest distance to any node it received a packet from. If the closest neighbor is closer than a threshold d, the node covers too little new area and drops its copy of the broadcast packet.

Additional Area

The progress of a node is calculated as the additional area AC it covers. Thereby, the area is not estimated and approximated like with the distance metric but really calculated. To do so, the position information of a broadcasting node is again added in the message header. The area AC a node covers is incrementally calculated whenever a packet arrives at a node. In contrast to [24] that uses a polygonal representation of the additional area, we decided to maintain a multidimensional array on each node. This array contains space for each unique packet that arrives and the area that is covered by the node in respect to the packet.

The approach of [24] has several drawbacks. The calculation of the additional area may have an inaccuracy of up to 22% [24]. Furthermore, it is difficult to introduce a progress threshold. In the polygonal approach a receiving node just tests if it is within the boundary of the polygon it calculates from the data it received from other nodes. Thus, the threshold must be determined as the distance the node is away from the polygon. Furthermore, the additional area does not only depend from the distance to the polygon, but also from the position of the node. Therefore, an exact calculation of the additional area is not possible and the threshold decision imprecise.

This can easily be done with our approach, as a two-dimensional array represents the transmission range of a node. During the incremental calculation of the intersection between the transmission ranges of the nodes that have already transmitted a packet and the receiving node

x, we mark each position in the transmission range of x that is covered by another node. Consequently, we can decide the ratio between the area AC covered by x and the maximal area A_{Max} reachable by x.

Both, our approach as well as the approach of [24] assume a circular transmission range. Therefore, the intersection between two nodes with circular transmission ranges can be calculated easily. The maximum progress A_{Max} of a node is the percentage of area it may cover, if it is exactly at the transmission border of the node it received a message from. A_{Max} is $\approx 0.61\%$ of a node's overall transmission range.

Signal Strength

The signal strength scheme operates similarly to the approaches proposed above, apart from the metric which is the received signal strength in this case. The received signal strength is an appropriate measurement to estimate the distance between two nodes. The farther away a neighbor node from a relaying node is, the weaker the received signal strength is. In mapping the signal strength on the distance we get a similar approach as with the distance metric. Consequently, the weaker the receiving power of a signal is the more additional area is covered by a node. However, the receiving signal power must be higher than -81dBm which is the receiving power necessary on the PHY802.11 layer to decode a signal.

During the DFD expiration again multiple copies of the packet may be received. Those packets are received at different signal strengths which indicate the distance to the sender of the packet. Consequently, the actual progress, e.g. the weakest signal a duplicate packet is received at has to be saved. A node x drops its broadcast packet if it has received a copy with a signal strength higher than the threshold d. This again indicates that another neighbor close enough to x has already relayed the packet.

The signal strength is not dependent on any location information. Thus, no position information has to be gathered and distributed, and no service like GPS is needed. Consequently, the use of the signal strength metric is an appropriate choice if no location service is available.

6.2.2 Dynamic Forwarding Delay (DFD)

The DFD is chosen from the interval $[0, T_{Max}]$, where T_{Max} is the maximal delay a node can buffer a packet. Furthermore, the DFD is calculated depending on the gathered topology information a node has. To calculate the DFD several functions, depending on the progress metric, are introduced. Furthermore, the functions may be linear or exponential (see Figure 6.2).

The linear function calculates the DFD linear to the progress a node adds. The exponential function favors the nodes closer to the transmission boundary even more. Thus, a polarization is done where only the nodes very close to the transmission boundary calculate very short DFDs. The exponential function should perform in particular well if neighbors very close to the transmission range of a broadcasting node exist. Functions 6.1 and 6.2 depend on the distance between sender and receiver. Functions 6.3 and 6.4 calculate the DFD according to the additional area a node covers. Functions 6.5 and 6.6 depend on the receiving signal power of a node.

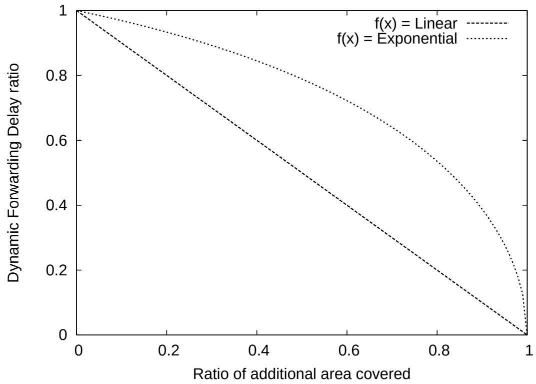


Figure 6.2: Shape of the DFD functions.

Function 6.7 is the random DFD (RAD) calculation used in the location-based scheme [24]:

$$DFD = T_{Max} \cdot \left(1 - \frac{d(x,y)}{r}\right) \quad (6.1)$$

$$DFD = T_{Max} \cdot \sqrt{\left(\frac{e - e^{\frac{d(x,y)}{r}}}{e-1}\right)} \quad (6.2)$$

Functions 6.1 and 6.2 calculate the DFD depending on the distance d between a sending node x and a receiving node y. In the first function the DFD is calculated linearly, whereas an exponential function is used in the second. The farther away two nodes are, the smaller the DFD becomes. If the receiving node is very close to the transmission range of the sending node the DFD is approximately 0, whereas a $DFD \approx 1$ results, when the receiving node is close to the sender. Thus, a receiver farther away will rebroadcast a packet earlier than a closer node. All nodes nearer than the resending one hear the relay of the packet and recalculate their DFD according to the information they received with the packet.

$$DFD = T_{Max} \cdot \left(1 - \frac{AC}{0.61 \cdot r^2 \pi}\right) \quad (6.3)$$

$$DFD = T_{Max} \cdot \sqrt{\left(\frac{e - e^{\left(\frac{AC}{0.61 \cdot r^2 \pi}\right)}}{e-1}\right)} \quad (6.4)$$

In the functions 6.3 and 6.4 the value of the DFD no longer depends on the distance between two nodes, but on the additional area AC a node covers. We again use a linear or an exponential function to calculate the current DFD. Whenever a packet is received, the DFD is recalculated if the interim progress is over the threshold, else the packet is dropped and no DFD has to be generated. Both functions again recalculate their progress as well as the DFD dynamically depending on the actual knowledge a node has about its neighborhood.

$$DFD = T_{Max} \cdot \left(1 - \sqrt[4]{10^{\left(\frac{RX_{min} - RX}{10}\right)}}\right) \quad (6.5)$$

$$DFD = T_{Max} \cdot \sqrt{\left(\frac{e - e^{\left(\sqrt[4]{10^{\left(\frac{RX_{min} - RX}{10} \right)}} \right)}}{e - 1} \right)} \quad (6.6)$$

The DFD functions 6.5 and 6.6 operate on the signal strength a packet is received at. RX is the receiving power a signal is received at and RX_{min} is the receiving power threshold below which the signal cannot be decoded anymore. Again a linear and an exponential calculation are supported. A weaker signal means a larger distance between two nodes. Whenever a packet is received, the DFD is recalculated according to the weakest signal the node has received a packet at.

$$DFD = T_{Max} \cdot RAND \quad (6.7)$$

The last function 6.7 is used by [24]. The DFD is randomly chosen between 0 and T_{Max}. Furthermore it is not recalculated according to the progress of a node. Thus, the local topology is not taken into account in the DFD calculation.

6.2.3 Cancel on MAC

If the DFD expires and the additional area coverable by a node is larger than the threshold, the packet is sent to the MAC layer and scheduled for relay there. The MAC 802.11 does not immediately rebroadcast the packet. It first has to carrier-sense the medium. Thus, it is possible that meanwhile another copy of the packet is received by the node. Therefore, the immediate broadcast of the packet is canceled and a recalculation of the progress initialized (the progress may have changed according to the new information gathered, which should be taken into account). Consequently, the packet is removed from the interface queue. If the progress is above the threshold the packet is handled to the routing layer and rebuffered with a new DFD, else it is dropped.

6.2.4 Threshold Decision

The Location-based routing protocol [24] proposes progress decisions on each node. Thus, even the first node scheduled to retransmit a packet depends its forwarding decision on the progress threshold. We insist that the first relaying node rebroadcasts a packet anyway, not depending on how much progress it adds. The refinement makes sense, as it is no restriction in dense networks, but may lead to a better reliability in sparse networks. We define that a node should only depend its broadcast on a progress threshold if it has heard another node relaying the same packet before.

Figure 6.3 depicts the influence a threshold-based broadcasting decision may have. The connected network cluster will never be reached, as node 2 does not rebroadcast the packet it received from node 1. Within our proposal, node 2 rebroadcasts the packet, as it is the first (only) node scheduling the packet for rebroadcasting.

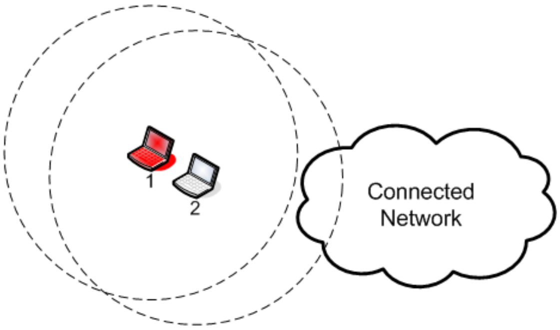


Figure 6.3: Reliability depending on the threshold.

6.2.5 Energy Consumption of DDB

An important drawback of mobile ad-hoc networks is the small availability of battery power. Therefore, a broadcast protocol is favored that consumes as little battery power as possible. To test DDB against power consumption, we implemented battery power metrics to enable these evaluations. Within our energy consumption scheme three states are distinguished. The sending of a message consumes a certain amount of energy which is described by a transmission power ratio Tx. Whenever a message is sensed another quantity of energy is consumed to receive and decode the message. The according parameter is called Rx. The active device state, whenever no sending or receiving activities are sensed, is called idle mode. The $Idle$ weight means that the idle sensing of the channel consumes as much power as the receiving of a message (see [43],[44]). The energy consumption evaluations are done in the end of the diploma thesis.

6.3 Simulation Scenario and Parameters

To properly test the DDB protocol we investigate several node densities in different topologies. As long as there is no mobility, the nodes are randomly distributed over the simulation area. In the mobility scenarios the Random Waypoint Model [15] is used. The several node densities we simulated and tested are depicted in Table 6.1.

No. of nodes n	area side s	No. of neighbors
250	4000	≈ 3
500	4000	≈ 6
500	3000	≈ 11
500	2000	≈ 24
1000	2000	≈ 49
2000	2000	≈ 98

Table 6.1: Number of neighbors according to network densities.

[18] shows that in general six to eight neighbors are necessary to achieve connectivity in a network. The following equation calculates the number of nodes n necessary to fulfill the above

condition:

$$n = \frac{m \cdot s^2}{r^2 \cdot \pi}$$

where m is the minimal number of neighbors, r the transmission radius, and s a network area side. In the sparse networks (three or six neighbors) it is possible that the network is not completely connected. As we want to measure the delivery ratio as the percentage of connected nodes reachable, we implemented an algorithm to determine the connected cluster around a sending node. Consequently, the delivery ratio is calculated corresponding to that cluster.

Distance [m]	Additional area [in % of 0.61 $\cdot r^2\pi$]	Signal strength [dBm]
25	10	$\approx$ 40
50	20	$\approx$ 27.96
100	40	$\approx$ 15.92

Table 6.2: Thresholds used in our simulations.

In Table 6.2 the thresholds used in the different progress calculations are shown. The thresholds are mapped to cover 40% of the maximal progress possible. This has been done to ensure consistency among the different approaches. In all simulations a 95% confidence interval was calculated to insure statistical relevance of our simulations.

Tx	Rx	$Idle$
10	1.0	1.0
10	1.0	0.1
1.5	1.0	1.0
1.5	1.0	0.1

Table 6.3: Different transmission, receiving and idle energy consumption values.

In the second part of the evaluation we investigate the energy consumption behavior of the implemented flooding schemes. To do so, different proportions of Tx, Rx and $Idle$ parameters are chosen. You can see the parameter settings in Table 6.3.

We compare the DDB protocol to a Simple Flooding scheme as well as the Location-based broadcast strategy proposed by [24]. If no other settings are mentioned, the simulation and protocol parameters listed in Table 6.4 are used to run the different protocols. The DDB protocol performs best with those parameters in most configurations, especially if the network is dense enough. For the Location-based protocol we use the values suggested by [24].

The simulations that support the mentioned values are discussed in later sections and substantiate the choices we did so far. Within the DDB protocol all three refinements are enabled. That means the calculation of the DFD depends always on the progress of a node, the message Cancel on MAC function is enabled, and the first relaying node scheduled rebroadcasts the packet anyway. One source packet is broadcast, as long as we do not test congestion, mobility, or energy consumption. In those cases it is necessary to send multiple packets to investigate the performance of the protocols. The exponential calculation of the DFD is used. The Location-

Parameter	Value
Simulator	Qualnet 3.6
Tx distance	250 m
Simulation Time	900s
T_{Max} (Location)	10ms
T_{Max} (DDB)	2ms
Threshold (Location)	100m
Threshold (DDB)	$0.4 \cdot A_{Max}$
Jitter (Flooding)	2ms

Table 6.4: Standard parameters used in our simulations.

based protocol is based on the distance metric as proposed in [24], whereas the additional area covered is used in the DDB protocol.

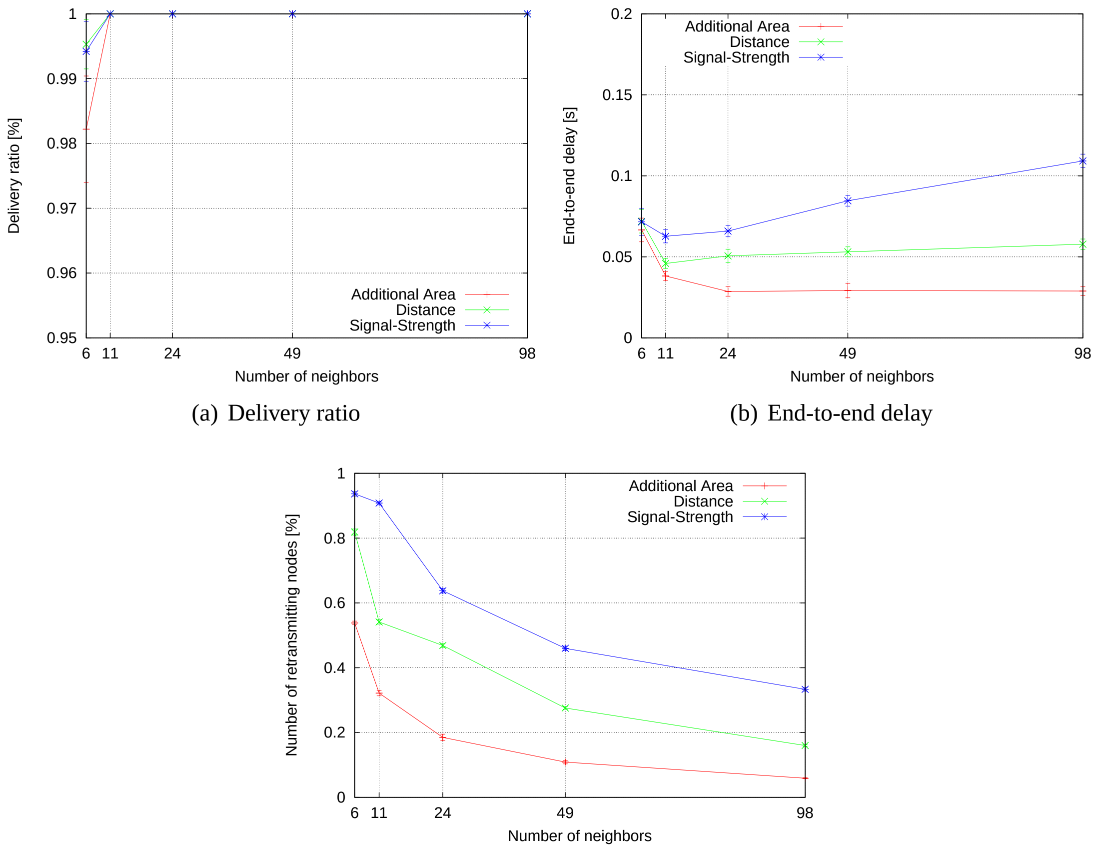
6.4 Evaluation

In the following the DDB protocol is compared to a Simple Flooding protocol on the one hand and to the Location-based protocol using the distance between two nodes for rebroadcasting decisions on the other.

6.4.1 Progress Schemes

We compare the different DDB schemes to figure out the most appropriate version to further deal with. The thresholds are defined as 40% of the overall progress possible. The results (see Figure 6.4) indicate that all progress schemes reach almost 100% delivery ratio, even in sparse networks. In contrast to the delivery ratio the different broadcasting algorithms differ much in respect to the number of retransmitting nodes as well as the average end-to-end delay. The additional area progress scheme produces by far the best results among the three approaches. Therefore, the additional area mode is chosen for our remaining investigations, as it provides the best results concerning the number of retransmitting nodes and the end-to-end delay.

The number of retransmitting nodes is decreased depending on the node density in all three modes. Furthermore, the ratios concerning the differences among the modes remain almost constant independent of the node density. If an even better mapping from the signal strength to the resulting DFD was found, the signal strength performance should approximate the distance results. This is obvious, as the DFD calculation operates similarly for both modes and the signal strength correlates with distance. However, The path loss is only calculated with a path loss coefficient of 4 for the nodes near to the transmission boundary. For all other nodes the coefficient is just 2. Our DFD calculation in contrast assumes a coefficient of 4 for all receiving nodes. Thus, nodes closer to the releasing node are proportionally favored. Enhanced with a better mapping, the signal strength becomes an appropriate mode for our simulations, as it avoids the drawbacks of distributing position information.



(a) Delivery ratio

(b) End-to-end delay

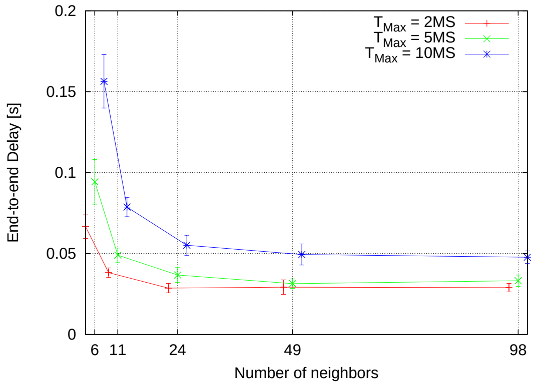
(c) Number of retransmitting nodes

Figure 6.4: DDB performance under different progress metrics.

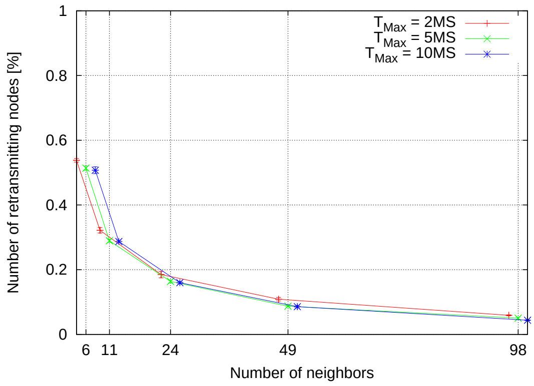
6.4.2 Influence of T_{Max}

Next we investigate the influence of the maximal packet delay T_{Max}. Provided the results above we tested the additional area progress with exponential DFD (see equation 6.4). We analyzed two metrics: the number of retransmissions necessary to broadcast a packet within the network and the end-to-end delay until the last node in the network receives the packet. The delivery ratio is not depicted, as the packet buffering time has no influence on network reliability. Thus, the ratios are similar to the results obtained in Figure 6.4 (almost 100% for all simulations).

Figure 6.5 depicts the results we got in our simulations. In sparse network a little value of T_{Max} decreases the end-to-end delay importantly, whereas the number of retransmission is almost not affected. The advantage of a higher T_{Max} only occurs in very dense networks and has just little influence. Therefore we decide to take a T_{Max} of $2ms$ for our further investigations. The results we obtain include the Cancel on MAC routine. If this option is disabled, the T_{Max} value has to be chosen higher, else the number of retransmitting nodes increases unproportionally. This is shown in section 6.4.4.



(a) End-to-end delay



(b) Number of retransmitting nodes

Figure 6.5: Influence of the maximal packet buffering time T_{Max}.

6.4.3 Influence of the Progress Threshold

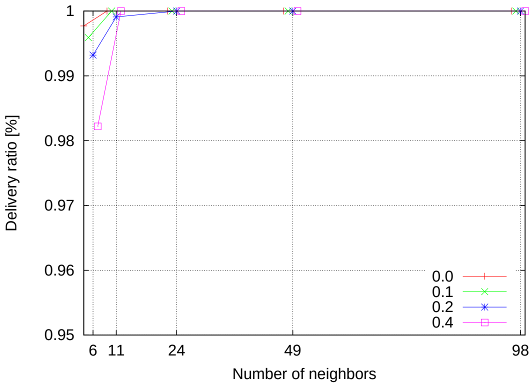
Another factor is the determination of the appropriate progress threshold. If the threshold of a buffering node is under-run, the packet is dropped. Thus, battery power is saved and congestion may be prevented, whereas the increased number of packet drops leads to a higher probability of network unreliability.

The results of the simulations with different thresholds are depicted in Figure 6.6. The average end-to-end delay as well as the number of retransmitting nodes decrease as soon as the threshold is increased. The diminution of the number of retransmitting nodes is obvious as a higher threshold leads to more rejected packets and therefore less retransmissions. The minor end-to-end delay is a side effect of the lower number of retransmitting nodes. As fewer rebroadcasting nodes correlate with fewer packets received by a node, it has to perform less rebufferings. Consequently, on average a node relays the packet sooner. The simulations support the progress threshold of 0.4 as being most appropriate. This choice is supported by the delivery ratio which still approximates 100% even in sparse networks. However, the results indicate that the choice of even higher thresholds would lead to less reliability, especially in sparse networks.

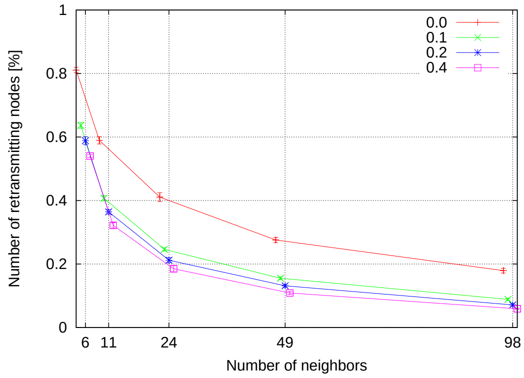
6.4.4 The Specific Protocol Refinements

In this section, the impacts of the individual DDB refinements are considered in more detail. First, we consider the scenario, where the threshold is applied to the first node releasing a packet in the vicinity of an initial sender. The advantage of our refinement is depicted in Figure 6.7.

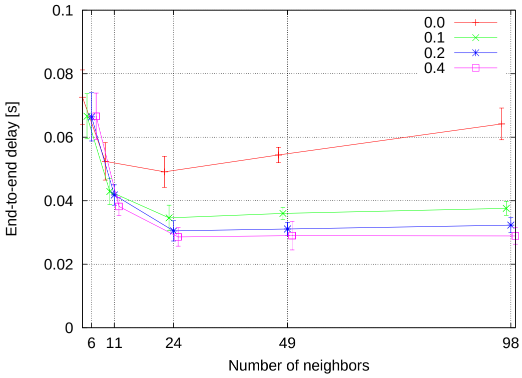
The broadcast of a packet by the ”best” neighbor without a progress threshold decision leads to an improved reliability in sparse networks, whereas no drawbacks concerning end-to-end delay and number of retransmitting nodes are added. The dependency of the first relaying node on a threshold diminishes the delivery ratio to less than 90% in a network with approximately six neighbors. This results substantiate our assumption that an unconditioned relay by the first node has no drawbacks. Therefore, we keep that refinement in use.



(a) Delivery ratio



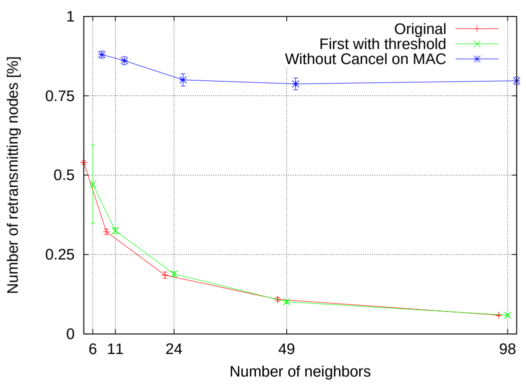
(b) Number of retransmitting nodes



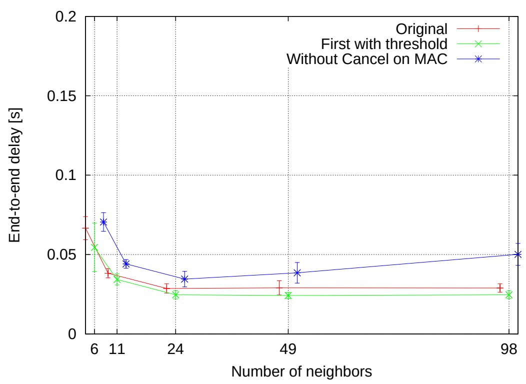
(c) End-to-end delay

Figure 6.6: Progress threshold.

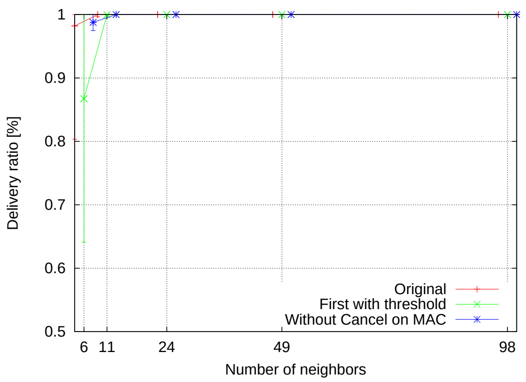
The second refinement is the Cancel on MAC option. Subfigure 6.7(a) indicates an important influence on the number of retransmitting nodes. Furthermore, the end-to-end delay seems to be affected, too. However, the bad effect on the number of retransmitting nodes correlates to the chosen T_{Max}. A T_{Max} value of $2ms$ is only appropriate in combination with the Cancel on MAC routine. Extending simulations show that $2ms$ is an interval too short to benefit from the progress calculations, as too few packets are received during such a short buffering interval. This applies to the DDB protocol as well as to the Location-based protocol. In subfigure 6.7(d) you can see the influence of the Cancel on MAC routine if we chose a T_{Max} of $10ms$. There are still more retransmitting nodes than with the standard DDB configuration, but the amount is belittled distinctively. The increased number of retransmitting nodes is obvious, as no canceling in the interface queue is possible anymore. We conclude that the Cancel on MAC routine is an important refinement. It enables especially the choice of a smaller T_{Max}, which improves the average end-to-end delay in sparse networks (see Figure 6.5). However, the abandonment of the Cancel on MAC is possible by increasing the T_{Max}.



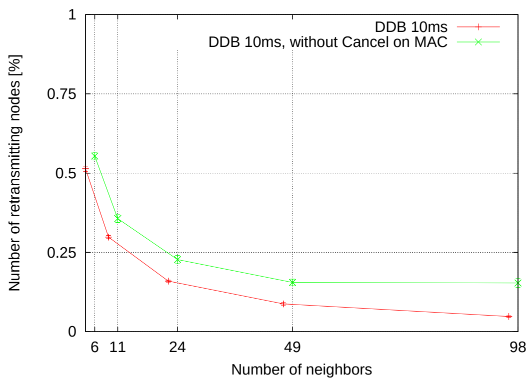
(a) Number of retransmitting nodes



(b) End-to-end delay



(c) Delivery ratio

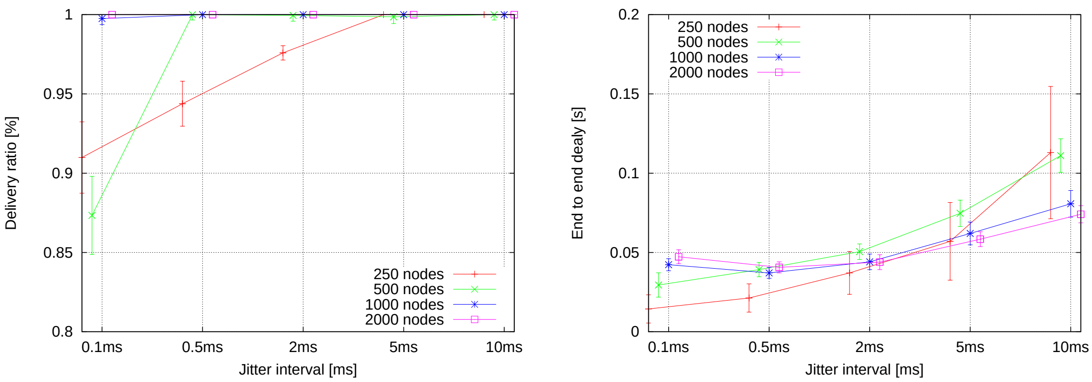


(d) Number of retransmitting nodes

Figure 6.7: The Specific Protocol Refinements.

6.4.5 Comparison of DDB with other protocols

The configuration of the DDB protocol is evaluated. It is shown that the parameters proposed in 6.3 are an appropriate choice. Thus, those protocol parameters are used for the remainder of the chapter. We will now compare the performance of DDB to a Simple Flooding protocol and the Location-based protocol with parameters as proposed in [24]. In Simple Flooding, we are confronted with the following problem: If multiple neighbors of a sending node are at similar distances away from it, they relay the packet almost simultaneously, which may lead to collision and packet loss. To avoid this behavior each packet broadcast is delayed by a random jitter interval. To properly test the influence, we additionally simulated a very sparse network with a mean of only three neighbors. Furthermore, we tested multiple numbers of nodes randomly distributed over an area of $4000m \times 4000m$. In sparse networks, collisions may lead to disconnections of nodes, or even worse to unattainability of a whole cluster of nodes. The cluster around a sending node is calculated and the delivery ratio determined relatively to the cluster.



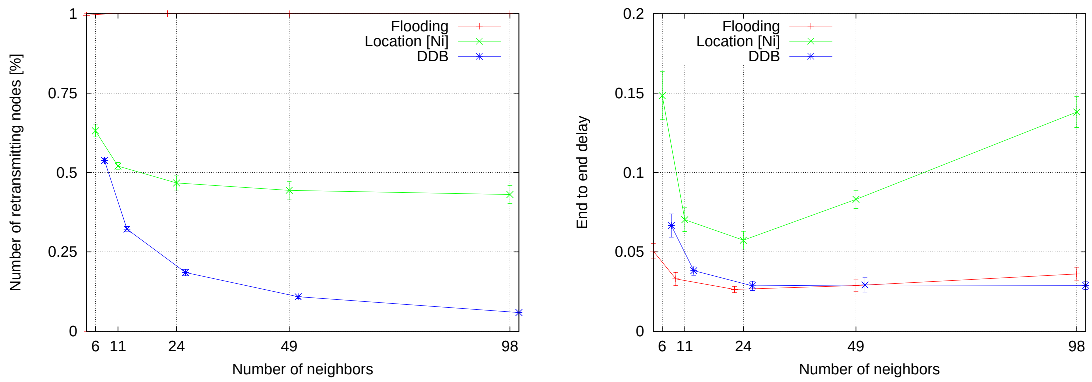
(a) Delivery ratio (b) End-to-end delay

Figure 6.8: Broadcast jitter in flooding.

The evaluation of different jitter intervals is depicted in Figure 6.8. The advantage of a high jitter interval is the decreased feasibility of collisions. Contrary, a high jitter interval increases the end-to-end delay. The simulations indicate that a jitter interval of $2ms$ is most appropriate, as minor values negatively affect the reliability and higher values increase the end-to-end delay unnecessarily.

Having outlined the settings of the different protocols, we are now able to compare the protocols among each other. In the first simulation setup, one packet is broadcast in static networks. Neither mobility nor congestion is taken into account. Thus, those simulations represent the inherent effect of the protocols on the number of retransmitting nodes, the end-to-end delay, and the delivery ratio.

The simulation results are shown in Figure 6.9. The delivery ratio for the DDB protocol is depicted in Figure 6.4. It represents the lower boundary for all simulations and is approximately 100%. Therefore, the delivery ratio is not depicted. The diminished number of retransmitting nodes with the Location-based protocol is reduced even more using the DDB protocol. The



(a) Number of retransmitting nodes

(b) End-to-end delay

Figure 6.9: DDB compared to flooding and location [24].

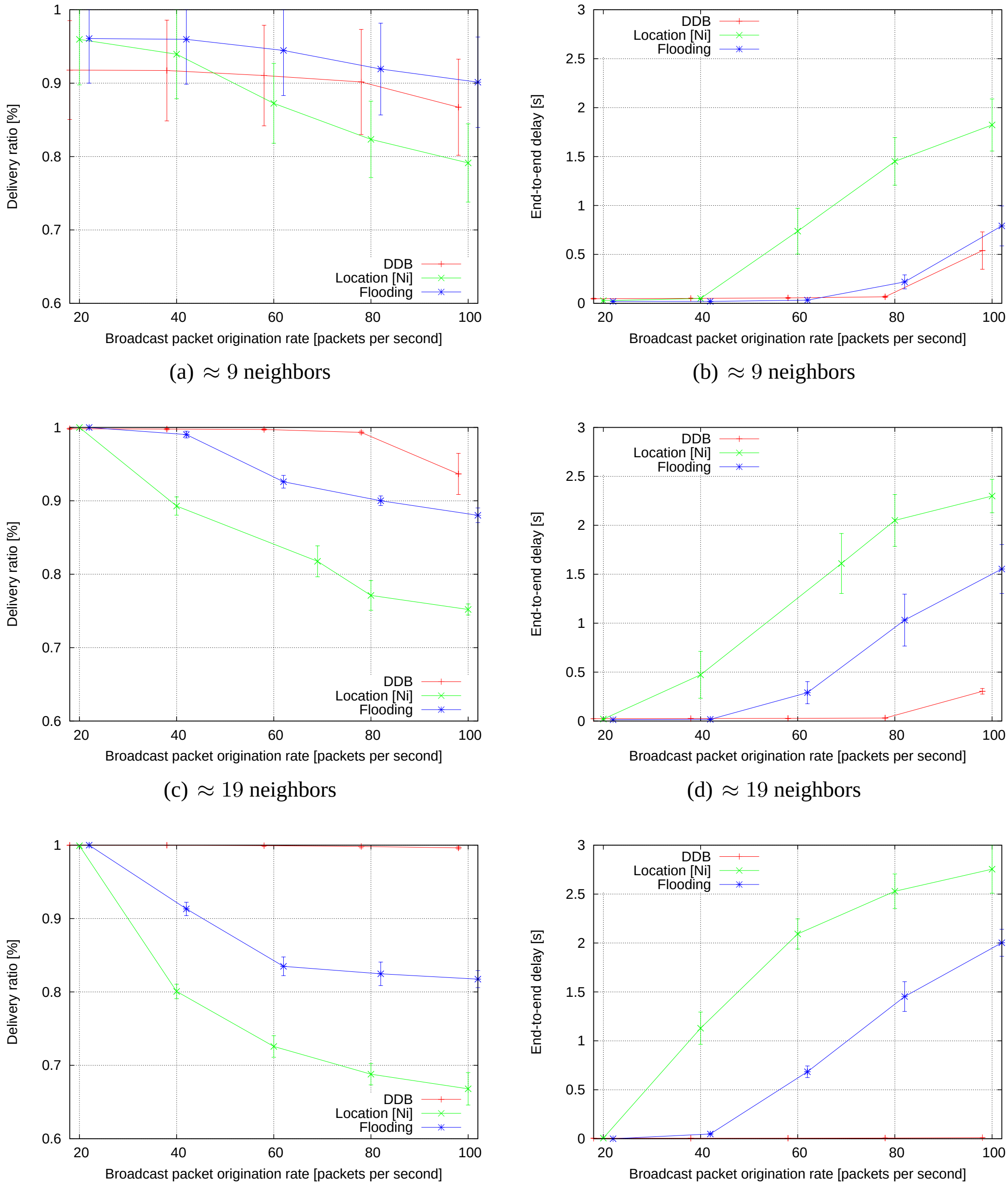
Location-based protocol calculates the DFD randomly, i.e. independent from the progress a node has. This indicates the stagnation of the number of retransmitting nodes around 40 to 50%. The DDB protocol in contrast uses the progress not only for the rebroadcasting decision, but also for the DFD calculation. Therefore, the progress affects the buffering time of a packet, which improves the best effort decisions on the nodes. This results in the continuous descending of the number of retransmitting nodes in correlation to the augmentation of the network density.

The average end-to-end delay of the DDB protocol and the Simple Flooding protocol are almost identical if only one packet is broadcast. Both have a maximal packet delay of $2ms$ per node. The slightly better end-to-end delay of the DDB protocol in dense networks corresponds to its minor network load. However, the exponential calculation of the DFD in the DDB protocol leads to a higher end-to-end delay in sparse networks. The Location-based protocol has no Cancel on MAC function. Thus, the maximal packet buffer time T_{Max} has to be chosen $10ms$ [24], else the number of retransmitting nodes increases unproportionally. Consequently, the high T_{Max} leads to the increased end-to-end delay of the Location-based protocol.

The results indicate that the DDB protocol adds important benefits to the Location-based protocol. The refinements diminish the number of retransmitting nodes drastically. The Cancel on MAC routine allows the choice of a small T_{Max} value, which eminently affects the end-to-end delay.

6.4.6 Congestion

One of the major drawbacks of the Location-based protocol is its inability to deal with congestion. Under heavy network load, the Location-based protocol does not perform better than Simple Flooding. We hope to avoid this drawback with the DDB protocol, as it diminishes the number of retransmitting nodes and the end-to-end delay. We restricted the simulation time to $100s$. 80 nodes are randomly distributed in the simulation area which is fixed to $1350m \times 1350m$, $900m \times 900m$ and $600m \times 600m$. Consequently, node densities of a minimum of 9 neighbors up to 44 neighbors are simulated. The settings differ from those used so far, as congestion con-



(a) $\approx$ 9 neighbors

(b) $\approx$ 9 neighbors

(c) $\approx$ 19 neighbors

(d) $\approx$ 19 neighbors

(e) $\approx$ 44 neighbors

(f) $\approx$ 44 neighbors

Figure 6.10: Performance in congested networks.

sumes much more simulation memory. Therefore, the simulation settings are scaled down. One randomly chosen source node broadcasts a packet at the following origination rates: 20, 40, 60, 80 and 100 packets per second.

Figure 6.9 indicates an average end-to-end delay for one single packet of 50 to 80ms for the Location-based protocol and 25 to 40ms for the DDB and Simple Flooding protocols, respectively. Consequently, the Location-based protocol is assumed to suffer heavily under congestion. The Simple Flooding algorithm on the other hand should be badly affected by its high number of retransmitting nodes.

You can see the congestion simulation results in Figure 6.10. In sparse networks the DDB protocol and the Location-based protocol seem to suffer a little under the decreased number of retransmitting nodes, whereas the Simple Flooding is only little affected by congestion. Nevertheless, the DDB protocol approximates the delivery ratio of Simple Flooding. It has even a slightly better end-to-end delay. The Location-based protocol again lacks on its high packet buffering time. Therefore, it performs badly even in sparse networks. As soon as the network density is elevated, the advantage of the DDB protocol comes into account. Even under high packet origination rates a delivery ratio of 100% is approximated, whereas the end-to-end delay remains almost constant at 50ms. The augmented network density, which leads to congestion in the Location-based and the Simple Flooding protocols, does not affect the DDB protocol. This is achieved by the improved rebroadcasting decisions that correlate to the raised number of neighbors.

The simulations show that the DDB protocol performs excellent under congestion, at least if the network density is high enough. Even in sparse networks, the DDB protocol is able to perform nearly as well as Simple Flooding. The disadvantage of Location-based broadcasting protocols is avoided, which makes the DDB protocol an appropriate choice for heavy loaded networks.

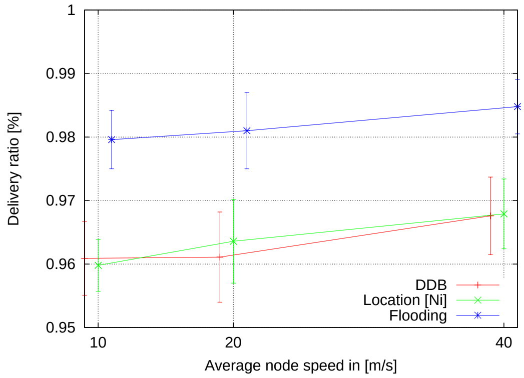
6.4.7 Mobility

Next, the DDB performance under mobility is investigated. All three protocols operate stationary without knowledge of their neighborhood. Therefore, the protocols do not maintain neighbor tables which may contain outdated neighbor entries. Consequently, all three protocols are expected to perform well under mobility.

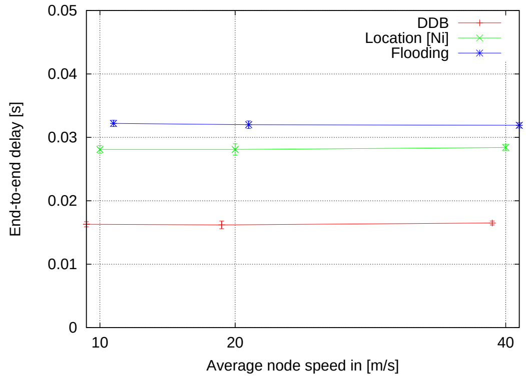
$S_{Avg} - 10\%[\frac{m}{s}]$	$S_{Avg} + 10\%[\frac{m}{s}]$
9	11
18	22
36	44

Table 6.5: Speeds used in our simulations.

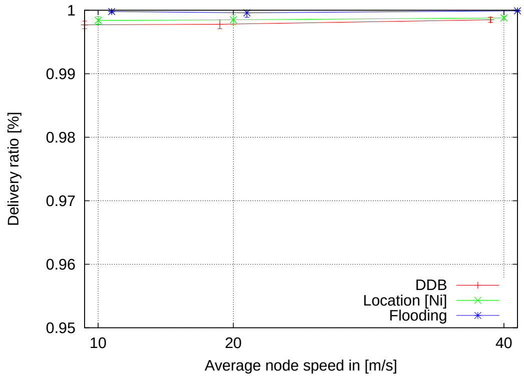
Both, the DDB and the Location-based protocol make their rebroadcasting decision due to the current position information they have gathered. Afterwards, the packet is buffered for a certain time. Meanwhile, the network topology may have changed. Thus, the packet broadcasts could be inaccurate according to the changed topology. That objection can be neglected as the packet



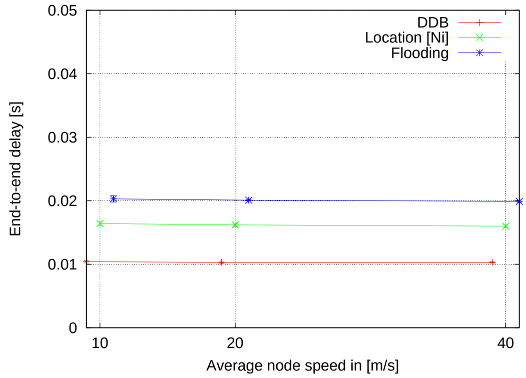
(a) ≈ 9 neighbors



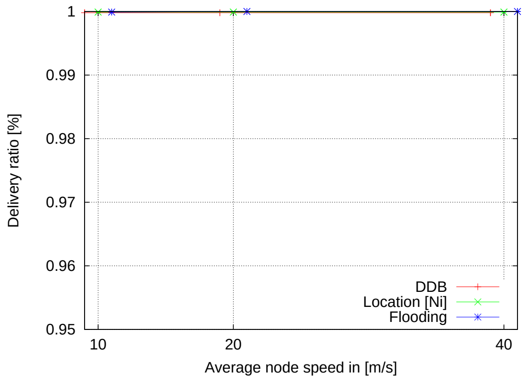
(b) ≈ 9 neighbors



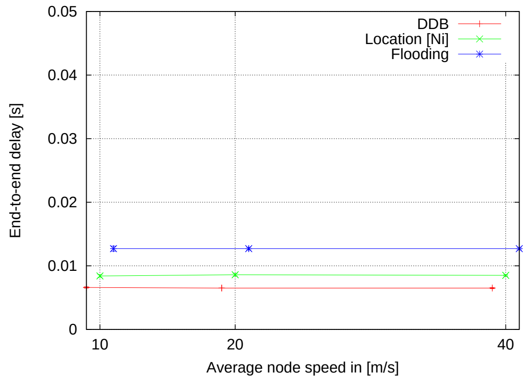
(c) ≈ 19 neighbors



(d) ≈ 19 neighbors



(e) ≈ 44 neighbors



(f) ≈ 44 neighbors

Figure 6.11: Influence of mobility.

buffering time is at maximum $10ms$ per node. Furthermore, the best progress decisions are taken locally, e.g. depending only on the neighbors of a node.

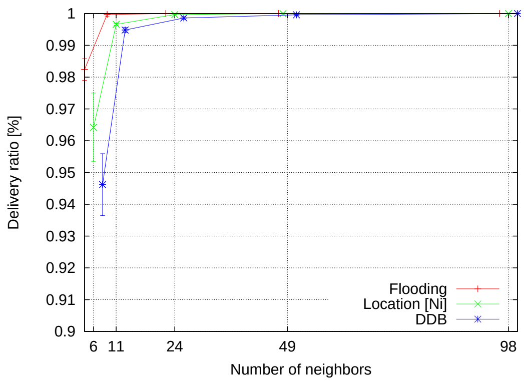
The simulation setup is the same as for the congestion simulations, apart from the packet-originating rate which is fixed to 10 packets per second. Thus, congestion is avoided. Furthermore, we use the Random Waypoint Model explained in Chapter 4. The pause time is set to $0s$. The average speeds S_{Avg} we simulate are depicted in Table 6.5. All nodes move at a speed of $S_{Avg} \pm 10\%$.

The simulation results are depicted in Figure 6.11. The results support our assumption that mobility does not have much influence on our protocols. The protocols do not suffer from out-dated neighbor information. Thus, mobility is almost transparent to them. The topology remains almost static for the duration of one network-wide broadcast. Figure 6.11 shows that the end-to-end delays are constant for all protocols over all node speeds. The packet loss in the sparse network can be explained by fragmentations of the network during simulation time. According to the frequent mobility changes no clusters around the source node are calculated. This assumption is supported by the higher confidence intervals of these simulations. Thus, the delivery ratio is absolute and not in relation to the cluster size. In denser networks, we can assume network connectivity throughout the whole simulation time. Consequently, the delivery ratio is 100% for all protocols. The simulations show that mobility has no influence on the protocols.

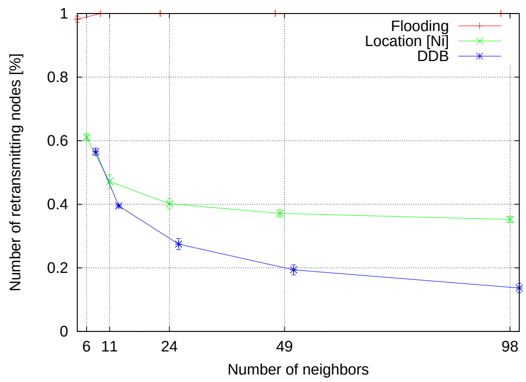
6.4.8 Radio Irregularity

The RIM model described in Chapter 2 is used. The DOI is set to 0.01 and the VSP to 0.5. The DDB protocol should suffer most under radio irregularity according to its modeling of circular transmission ranges. The Simple Flooding as well as the Location-based protocol does not depend on any suggestions concerning the radio range. Therefore, irregular transmission ranges should not affect them. However, the network connectivity is no longer predictable. Consequently, the delivery ratio cannot be calculated dependent on the cluster size around the source node.

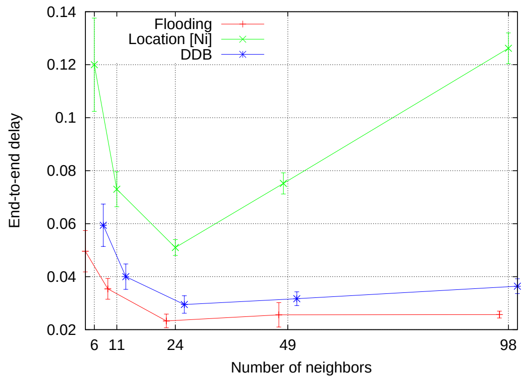
The results with radio irregularity are shown in Figure 6.12. Commonly the delivery ratio is affected a little in sparse networks, whereas in denser networks a 100% delivery ratio is achieved. The average end-to-end delay remains for all protocols the same as in the simulations using circular radio ranges. This indicates independence of the end-to-end delay on radio irregularity. The Location-based protocol uses fewer retransmitting nodes, whereas DDB needs an increased number of retransmitters under irregular transmission ranges. However, DDB remains by far the best performing protocol. The improved performance of the Location-based protocol can be explained by its distance progress metric which is less affected by irregular transmission ranges. Consequently, the threshold decisions seem to benefit from that effect. Whereas the inferior performance of the DDB is obvious. The misled radio area assumptions mentioned above do not map the irregular transmission ranges. We conclude that the effect of radio irregularity results in negligible drawbacks.



(a) Delivery ratio



(b) Number of retransmitting nodes



(c) End-to-end delay

Figure 6.12: Protocol performance under irregular radio ranges.

6.4.9 Energy Consumption of Broadcasting Protocols

A major determinant of mobile ad-hoc networks is the consumption of battery power. Mobile devices may be equipped with feeble batteries. Therefore, it is desirable to load such devices as little as possible. Concerning battery power, the Simple Flooding protocol is an upper bound of wasted energy. This is obvious as the energy consumption conductively correlates to the number of rebroadcasting nodes.

We will show that the DDB protocol on the other hand performs well, because the number of retransmitting nodes can be diminished considerable. Furthermore, contrary to neighbor knowledge methods no additional distribution of hello-messages is necessary. The additional energy consumption to receive the position of a node (GPS, VHR), which is necessary in position-based protocols, is neglected. This is acceptable, as it is proportionally low.

The simulation setup is as follow. A randomly chosen node initiates a broadcast packet every ten seconds. The network area is $2000m \times 2000m$ and 1000 nodes are randomly distributed within those boundaries. This results in a node density of approximately 49 neighbors. The decision to send a packet every ten seconds is taken in order to avoid high network loads which would overstress the simulator. The $Idle$ value is adjusted (see Table 6.3). An $Idle$ value of 1 does not mean that the idle sensing of the carrier consumes as much energy as the receiving of a packet, but that the idle energy consumption between the broadcasting of two packets is weighted with 1. The fixing of the $Idle$ parameter makes sense, as we simulate networks which are most of the time in idle mode and we want to investigate the influences of the number of retransmitting nodes as well as the number of packets received.

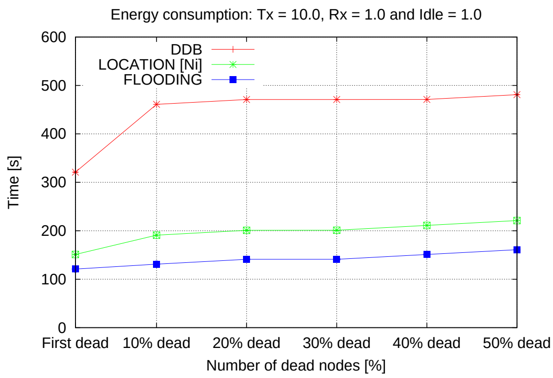
Ratio of dead nodes [%]
First dead
10
20
30
40
50

Table 6.6: Percentage of dead nodes.

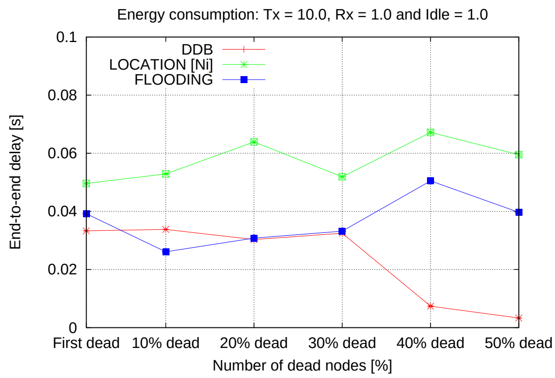
To properly analyze the simulations we collect the ongoing data whenever a certain ratio of nodes is dead. You can see the ratios of dead nodes in Table 6.6. At these time points, we collect the average energy power and the current end-to-end delay of a broadcast packet. We stop our simulations as soon as 50% of the network participants are dead. This is reasonable as afterward the fragmentation of the network increases fast. A vigorous decrease of the average end-to-end delay indicates the network fragmentation, too.

The results with one source node are depicted in Figure 6.13. The transmission power is weighted with a value of ten and the receiving power with 1. That means the transmission of a packet needs ten times more power than the reception. The idle time is set with two values, once with 1 and once with 0.1. In the second setting, the idle sensing has almost no influence as it takes about 100 seconds until one power unit is consumed by idle listening. The results show that the small number of retransmitting nodes in the DDB protocol has a big influence on the battery life expectancy. This is obvious as fewer retransmitting nodes indicate less packet receiving operations. The comparatively strong ascending between the death time of the first node and the death time of 10 percent of the network within the DDB protocol correlates to the selective choice of most appropriate neighbors. The progress dependencies within the DDB protocol lead to a selective disconnection of neighbors. The fast network drop as soon as more than ten percent of the network is dead is common to all protocols. The end-to-end delay remains almost constant for the Location-based and the Simple Flooding protocol. This corresponds to the uniform load of the participants. DDB in contrast suffers as soon as more than 30% of the network is dead. The selective disconnections in the DDB protocol lead to fragmentations which cause the decrease of the average end-to-end delay.

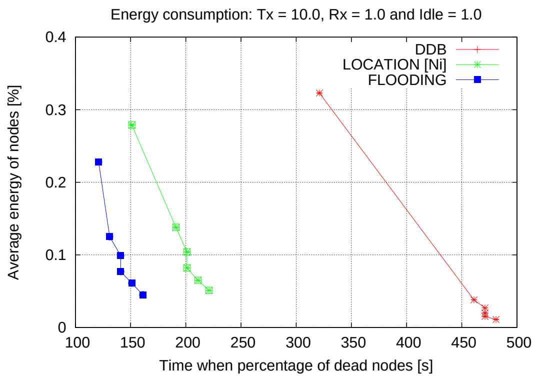
In a second scenario, we fix the number of sources to three randomly chosen nodes. The DDB protocol and the Simple Flooding protocol perform almost similar to the simulations with only one node. The death times are obviously much shorter, but the ratios remain the same. However, the end-to-end delay of the DDB protocol suffers much less under the increased network load. It remains almost the same as with only one source. The simultaneous broadcast



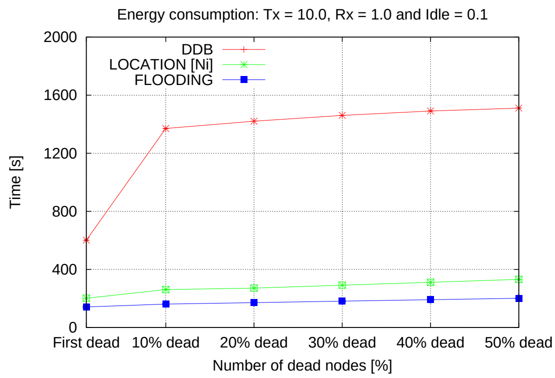
(a) Percentage of dead nodes



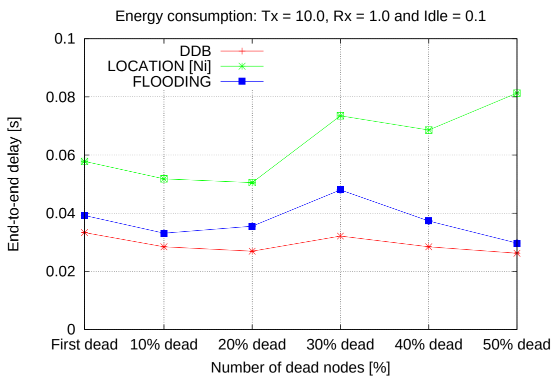
(b) Delay when percentage of nodes dead



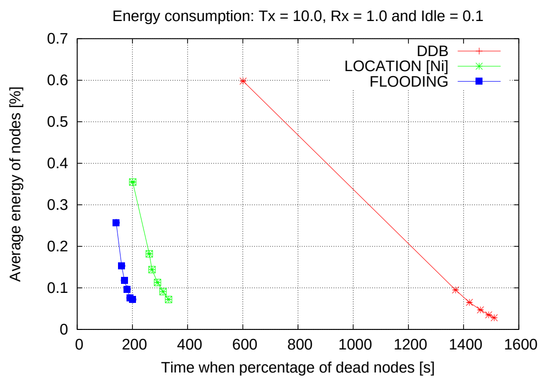
(c) Average battery power whenever a given amount of nodes is dead



(d) Percentage of dead nodes

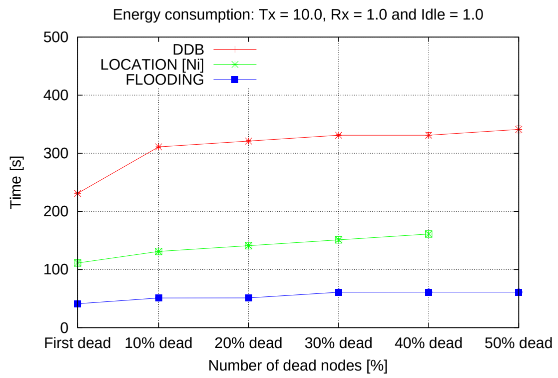


(e) Delay when percentage of nodes dead

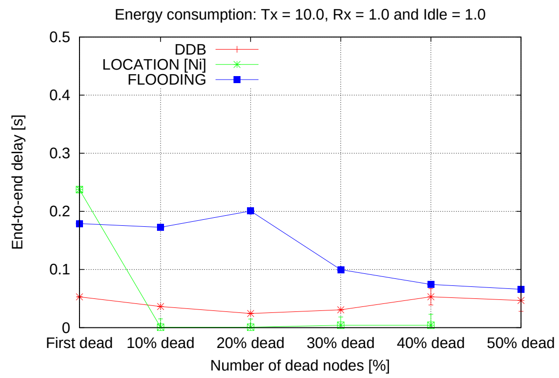


(f) Average battery power whenever a given amount of nodes is dead

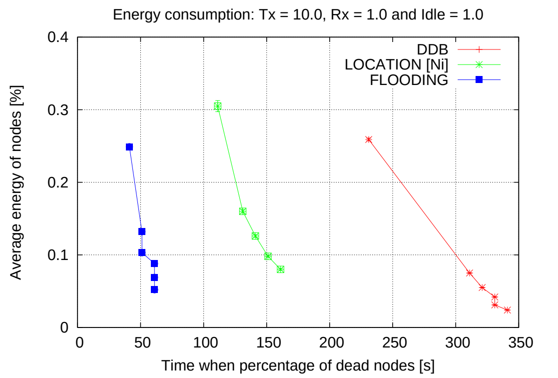
Figure 6.13: Energy consumption with one source node broadcasting a packet every ten seconds.



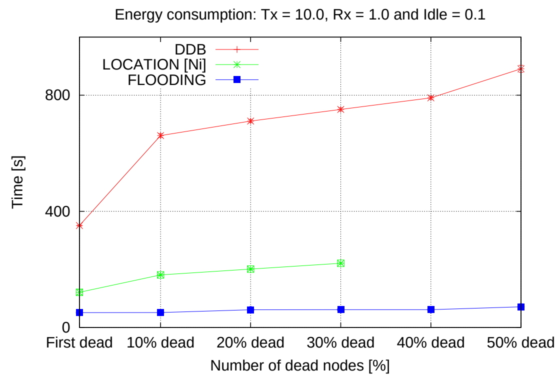
(a) Percentage of dead nodes



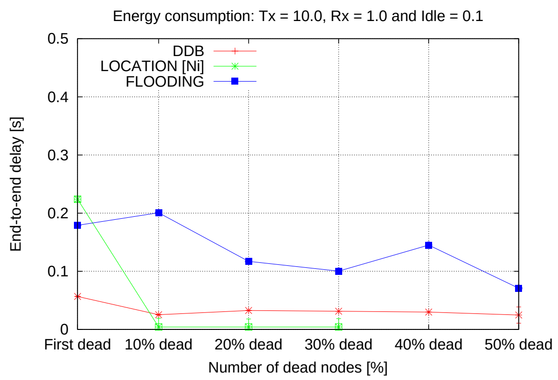
(b) Delay when percentage of nodes dead



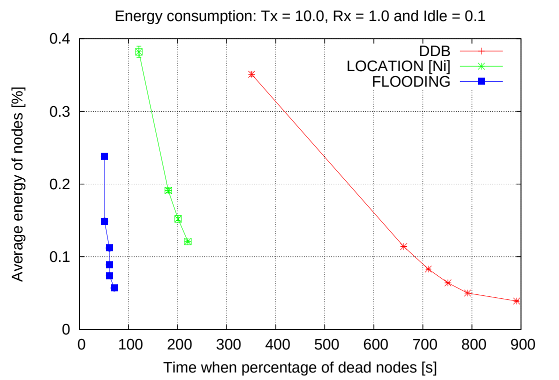
(c) Average battery power whenever a given amount of nodes is dead



(d) Percentage of dead nodes



(e) Delay when percentage of nodes dead



(f) Average battery power whenever a given amount of nodes is dead

Figure 6.14: Energy consumption with three source node broadcasting a packet every ten seconds.

of three packets does not affect the DDB protocol, which is a promising result. The results we got with the Location-based protocol are a bit astonishing. It takes almost as long as with one source node until a given percentage of nodes in the network is dead. The difference is at least lower than in the other two protocols. The random choice of neighbors adds some benefit here. The abrupt decreasing of the average end-to-end delay after the death of the first node indicates that the network is rapidly fragmented. This assumption is supported by Subfigure 6.14(a) and Subfigure 6.14(d) which show that the neighborhood of the source node is disconnected before 50% of the whole network is dead.

We conclude that the DDB protocol consumes much less energy than the other two protocols. The specific disconnections lead to comparatively long intervals until the lower percentages of dead nodes occur. The simulations show that DDB does not only reduce the average energy consumption, but is also hardly affected by simultaneous broadcasting of packets.

6.5 Conclusion

The DDB protocol adds important features to the position-based broadcasting techniques known so far. The protocol performs well under all network metrics we tested. It shows its qualities especially in congested networks and when energy consumption is taken into account. Furthermore, it does not depend on mobility and is well scalable.

The impressing scalability of the DDB protocol correlates to the improved forwarding decisions in dense networks. This is obvious, as in dense networks much more nodes are covered by other nodes. Consequently, those nodes are prohibited to rebroadcast a packet.

The dependency on location information, which is the main disadvantage of the protocol, was not considered further. The overhead to gather this information is considered rather negligible. If not, the signal-strength progress decision could be investigated in more detail. Another drawback, the protocol suffers from is its assumption of circular radio ranges, even if the nodes have very inhomogeneous radio ranges. However, the simulations show that radio irregularity barely restricts the DDB protocol.

A future task is to compare our protocol to neighbor-knowledge-based broadcasting protocols. The Multipoint relaying protocol (MPR) is implemented, but not yet tested extensively. We suppose our protocol to perform better than the MPR protocol. Especially in respect of mobility and battery power, where the additional distribution of neighborhood information in the MPR protocol means an immense drawback.

Chapter 7

Further Investigations

7.1 Destination Search Schemes in Mobile Ad-hoc Networks

In location-based routing algorithms, we are confronted with the problem that the position of a destination is not known when a source wants to initialize a communication. Therefore, a location service is needed to deliver the position of a specific node. Furthermore, all mobile nodes have to register their current position with that service. In opposition to classical cellular networks where a central server does such a service, the necessary information has to be distributed in MANET. This is obvious, as within a mobile ad-hoc network we would not even know the position of the server. Therefore, we have to look for other solutions. One strategy to get the needed information is to simple flood the whole network. However, flooding operations have a dramatic effect in large networks and, thus, are not further considered as an approach to solve the location service problem. In [45] several approaches to supply location services are introduced. All those protocols are proposed, but none of theme has been implemented. Consequently, no data on the effectiveness of such a location service is available. Thus, we decided to implement one of those location services [3] called virtual home region (VHR). A similar approach has been proposed by [4].

7.1.1 The Virtual Home Region (VHR)

A VHR is an area, somewhere located in the network that supplies the needed information about a node. Thus, a mapping of each node to its VHR is necessary. In our approach, each node has its own VHR, uniformly distributed over the whole network area. We chose that approach to prevent congestion areas, what could affect the location service reliability. Centralized regions for multiple nodes could lead to congestion if enough queries were sent to the same region simultaneously. We maintain the diameter of the home region variable, in a way that there are always between two and ten nodes within its range. The position of the owner node of the VHR is distributed among those nodes currently located in the VHR. The minimum number of two nodes grants that the position is always available and the maximum number of ten nodes ensures that the communication overhead does not burst. The relation between a node and its VHR is defined by a well-known hash function H. The function H operates on the end-system unique identifier (EUI) space. The EUI is a key uniquely identifying a node. H returns an image in the location dependent address (LDA) space. The LDA is simply a triplet of geographic coordinates. The equation is $H(EUI) = C$, where C is the center of the VHR of the node with identifier EUI.

The update of the VHR operates as follows. The owner node x periodically sends, after a time interval t, a unicast message containing its current position to its VHR. Within the VHR the update message is distributed to each member node. If a node y is willing to send a packet to x, it sends a unicast query message into the VHR of x. The first node in the VHR receiving the position query sends a unicast message, containing the LDA of x, back to y. The VHR update as well as a position request are depicted in Figure 7.1.

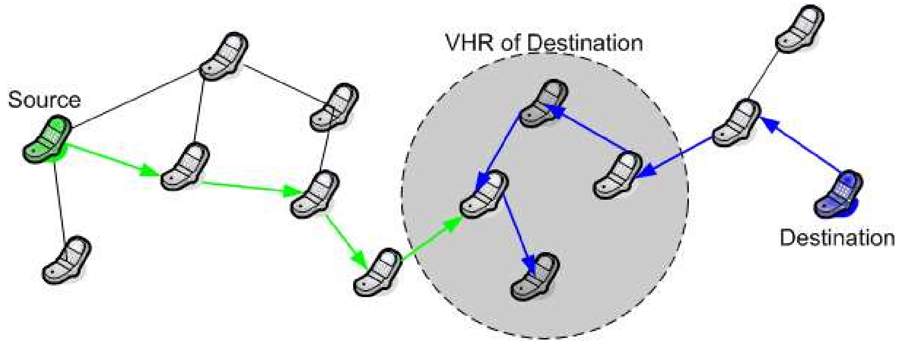


Figure 7.1: VHR update (blue) and position request using GPSR (green).

The approach has several technical difficulties. First, the VHR needs some management procedures to ensure that the VHR contains the predefined number of nodes. A second difficulty is the distribution of the LDA within the VHR. New nodes may enter the VHR and nodes may leave it. Thus, the VHR must be equipped with multiple self-organizing functions. To solve the first task, the radius of the VHR is distributed among its members and is updated, whenever one of the ranges which constrict the number of members is transgressed. That indicates broadcasting of the appropriate data, within the VHR, whenever such boundary violations occur. The distribution of the LDA among the VHR members is handled likewise. To do so, each node knows the current VHR it is a member of, or it sets a mark that it is currently within no VHR. Additionally each node frequently checks its current VHR membership. Whenever a VHR membership change occurs, a message is broadcast to inform all VHR members of the new state.

7.1.2 Implementation and Verification

During the implementation of the VHR, we were confronted with multiple difficulties. It does not seem possible to implement the distribution of the data within the VHR in an efficient way. As we simulate dense networks with approximately 220 nodes per square kilometer, the protocol overhead to organize 400 different home areas is much to high. Each node has 44 neighbors in average. Furthermore, the nodes are moving at a speed between $1\frac{m}{s}$ and $40\frac{m}{s}$. If we expect an average of five neighbors in our VHR, the expected radius is:

$$r = \sqrt{\left(\frac{5 \cdot 1000^2}{220 \cdot \pi}\right)} \approx 86m$$

Consequently, we have 400 circles, each having a diameter of about $170m$, containing five nodes in average moving at rather high speeds. The frequency of member change in the VHRs throughout the whole network, therefore, is very high and causes much overhead.

There are several modifications and optimizations possible, but the efficiency of such a protocol may still not be given. A big problem is that the distribution within each VHR depends on broadcasting mechanisms. Even if the number of broadcasts can be minimized, it still leads to a capacious protocol overhead. Furthermore, the enhanced traffic in the network may lead to even more collisions. Those collisions may also influence the reliability of the basic routing protocol. A possible solution is to map multiple nodes on one VHR. Thus, the overhead could be minimized by the same factor, as the VHR houses nodes. The disadvantage is the concentration of traffic on those areas then.

The update strategy as well as the request/respond routines are very easy to implement. The organization of the VHR in contrary leads to inefficiency that can hardly be solved in a highly mobile and dense network. Additionally, the few simulations done showed that in combination with GPSR the additional delay through the location service is much higher than expected.

7.2 Inaccuracy of Destination Information

In this section a restricted local flooding mode is proposed. Its intention is to deal with inaccurate destination information in Location-based routing protocols. The algorithm is implemented to enhance BLR. The inexact position information may originate in imprecise data delivered by a destination search scheme, or the position may be out-dated as soon as the packet arrives at the neighborhood of the destination. This is possible if the packet was routed over multiple hops and the destination moved away from its initial location in the meantime. That occurrence is rather improbable as the packet normally has short end-to-end delays. The destination may not have moved far enough in the meantime. Whereas the combination of imprecise destination position supply with a high end-to-end delay makes this possible to happen.

7.2.1 Restricted Local Flooding

If a packet includes an out-dated destination position, it is not deliverable with BLR. In this case the backup mode has no chance to deliver a packet. Consequently, the packet remains in backup mode until either a loop occurs or the packet is dropped. In order to deliver messages in such cases, we refine the BLR protocol with a restricted local flooding mechanism.

The local flooding is initiated whenever a packet arrives in the neighborhood of the destination position, included in its header, but the destination is not reachable. That means, the destination has moved farther away from its initial position than the transmission range r. Thereby, the local flooding is initiated before the backup mode is entered.

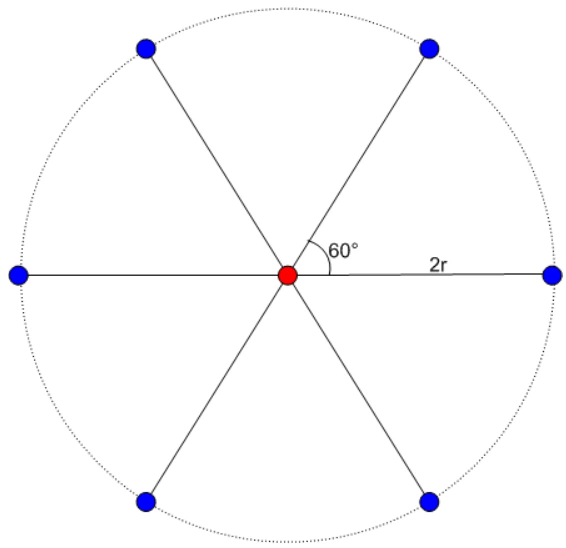


Figure 7.2: Geocast destinations of local flooding.

The local flooding operates as follows: Six positions around the initial destination position are chosen. Each of them twice the transmission range r away from the initial position and with an angle of 60° between any two pairs of these points (Figure 7.2). A duplicate of the packet is sent to all these temporary destinations, using BLR as routing protocol. If the destination has not left that circle which has a radius of the double transmission range and is centered at the initial destination coordinates, it should be reachable using the local flooding mechanism. The value of twice the transmission range should be sufficient to ensure availability.

7.2.2 Conclusion

The restricted flooding algorithm is a refinement to the BLR protocol implemented in [5] and has to be inserted there. This task has not yet been done and all future investigations and evaluations, therefore, are referenced to [5].

Chapter 8

Conclusion

In this diploma thesis some problems of location-based routing protocols for mobile ad-hoc networks are discussed. The difficulties are mainly caused by ignorance of the network topology and the difficulty to predict network performance. These restrictions lead to routing overheads when gathering or distributing the needed information. This overhead is faced by low bandwidth and low battery power which are natural constraints in mobile ad-hoc networks.

One way to counter the problems of inaccurate topology information is to improve the distribution and reliability of the exchanged location information. A number of conceivable techniques to enhance the distribution and correctness of hello messages are proposed and evaluated. GPSR is chosen as underlying routing protocol, but could be replaced by any other protocol based on hello-messages. The simulations show that a beaconing strategy expanded with the possibility of predicting neighbor positions adds network reliability, scales down routing overhead, and improves the end-to-end delay. In other simulations a correlation between the movement of a node and the frequency of sending beacons is done. The mapping of the moving characteristics of a node on its beacon-sending interval enhances the network performance. The refinements are easy to realize and improve location-based routing protocols considerably.

An astonishing effect of all routing protocols is their weak dependence on radio irregularity. The multiple simulations show only little influence of the circular transmission range assumptions done by GPSR and DDB. GPSR is highly adaptable to wrong neighborhood information. In the case of a wrong routing decision, it simply choses another neighbor as next hop and repeats that process until the packet is forwarded or dropped. The delivery ratio of DDB suffers a little more, but is still almost negligible. Another reason is the absence of complex, time, and resource consuming route detections in location-based protocols. Those services would degrade the routing performance under radio irregularity.

The investigation of destination search schemes shows that those approaches cause high routing overheads. The lack of time and the concentration on the other problems delay the more specific analysis of the drawbacks. Thus, the assumption of the knowledge of a node about its current position is taken into account in our simulations. We assume that the gathering of the location information can be obtained via a service like GPS. However, GPS is not always available and the information obtained is charged with impreciseness. However, the using of a location service like VHR would also provide imprecise position information and would additionally load the network.

The position-based DDB broadcasting algorithm proposed in the second part of the diploma thesis provides satisfactory results. The enhancements to the Location-based scheme considerable improve reliability. The most important drawbacks of the Location-based protocol, the bad performance under congestion and in dense networks, are avoided. The simulations in contrast show that our protocol performs extremely well in dense networks with high loads. The simplicity of the DDB protocol and its scalability make it an appropriate choice for broadcasting tasks in dense or highly mobile networks. The energy consumption simulations again show good performance for the DDB protocol, which makes it attractive for sensor networks.

Chapter 9

Future Work

The diploma thesis addresses some drawbacks of position-based routing protocols. For some of them, satisfactory solutions are proposed. On the other hand, a lot of new and additional questions and problems rose up during the work. They are listed below and could be investigated in the future in more detail.

- The combination of the position prediction mechanism with the mapping of the speed of a node on its beacon sending frequency could be evaluated. Thus, the advantage of the increased reliability caused by the prediction of future positions is combined with the reduced network load caused by a more appropriate beacon sending strategy. Both techniques are disjunctive. A combination is meaningful and a better performance can be expected.
- A reactive distribution of hello-messages is introduced by the GPSR reactive protocol. The protocol is implemented without considering any refinements. Nevertheless, it shows good results for the simple version. Therefore, a closer consideration of possible enhancements could be promising. The regathering of neighbor information on each hop to the destination can be optimized. Already known information can be buffered. If the interval is chosen short enough, inaccuracy of that data is avoided. Thus, the number of neighbors responding to a hello query can be minimized and the latency to gather that information can be diminished.
- Radio irregularity is only considered for position-based routing protocols (GPSR, DDB) and the simple flooding broadcast protocol. All these protocols perform quite well in correlation to irregular transmission ranges. However, it would be interesting to see the performance of other protocols, e.g. table-driven or route-based on-demand protocols.
- The virtual home region (VHR) protocol is not yet implemented definitely. Particularly, it could be enhanced with better algorithms. The major problem is an efficient distribution of the position information of a node within its VHR. Furthermore, the whole organization of the VHR, e.g. who is the coordinator, how to insure population of the VHR, is quite difficult. Appropriate methods have to be added.

- The restricted local flooding implemented to enhance BLR is not yet included in the protocol. This task as well as the subsequent testing and evaluating of the service remains to future work.

- The Multipoint Relaying (MPR) broadcast protocol has been implemented apart from the diploma thesis. The DDB protocol will be tested and evaluated intensively against the MPR protocol.

- The DDB protocol uses the additional area coverage mode to decide the progression of a node. That service postulates the knowledge of the own position information of a node. The usage of signal-strength as progress metric on the other hand does not depend on any additional data. Therefore, it should be investigated in more detail.

- The performance of the DDB extended with a location gathering service could be evaluated. This task could be done to investigate battery consumption more adequately.

- The consideration of sleeping nodes could be taken into account. The behavior of routing as well as broadcasting techniques under that condition could be evaluated.

Glossary

Beacon interval The frequency hello-messages are sent in.

BLR (Beaconless Routing Protocol) BLR is a position-based routing protocol that abandons on hello-messages.

Cancel on MAC A function performed by the DDB protocol in order to remove packets from the network queue.

DDB (Dynamic Delayed Broadcasting Protocol) DDB is an area-based broadcasting protocol. A node rebroadcast a packet if the progress of the node is high enough.

Dead interval It determines how long an entry remains in the neighbor table.

DFD (Dynamic Forwarding Delay) The delay is calculated depending on the progress of a node.

DOI (Degree of Irregularity) Addresses the properties of the propagation media.

GFG/GPSR (Greedy Perimeter Stateless Routing) GFG/GPSR is a position-based routing protocol that uses hello-messages (beacons) to pro-actively distribute position information.

Location-based broadcast protocol A broadcast protocol that uses local position information to decide the broadcast of a packet by a node.

MANET (Mobile Ad-hoc Networks) Mobile networks which do without any fixed infrastructure

MFR (Most Forwarding within Radius) The neighbor closest to the destination, but still within transmission range of the relaying node is chosen as next hop.

MPR (Multipoint Relaying Protocol) A broadcast protocol based on the knowledge of the local two-hop neighborhood of a node.

NFP (Nearest within Forwarding Direction) The closest neighbor to the relaying node, but still with progress to the destination is chosen as next hop.

Number of retransmitting nodes The number of nodes necessary to rebroadcast a packet in order to reach any participant of a network.

Progress of DDB It is defined as an assumption of how much additional area a node is supposed to feed.

QualNet A discrete event simulator for wired and wireless networks.

RAD (Random Assessment Delay) A delay, similar to the DFD, packets are buffered for. However, the delay is calculated randomly and not due to the progress of a node.

Random Waypoint Model A model to simulate the mobility of nodes.

RIM (Radio Irregularity Model) It generates irregular transmission ranges on the physical layer.

RMP (Random Progress Method) A random neighbor closer to the destination than the relaying node is chosen as next hop.

Seed A seed value generates random numbers that remain the same in equivalent simulations. It determines the sequence of pseudo-random numbers.

Threshold Decision A packet is only broadcast if the progress of the node is higher than the threshold.

VHR (Virtual Home Region) A destination search protocol. Supplies a source node with the position information of the destination.

VSP (Variance of Sending Power) Covers device specific manufacturing properties.

Bibliography

[1] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *MOBICOM'00*, Boston, USA, Aug. 2000, pp. 243–254.

[2] G. Zhou, T. He, S. Krishnamurthy, and J. Stankovic, "Impact of radio irregularity on wireless sensor networks," in *MobiSYS'04*, June 2004.

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

[4] I. Stojmenovic, "Home agent based location update and destination search schemes in ad hoc wireless networks," in *Advances in Information Science and Soft Computing*. WSEAS Press, 2002, pp. 6–11.

[5] T. Bernoulli, "Beacon-less routing in mobile ad hoc networks," Nov. 2004, diploma Thesis.

[6] T. S. Rappaport, *Wireless Communications, Principles and Practice*. Upper Saddle River Prentice-Hall PTR, 2002.

[7] H. Friis, "A note on a simple transmission formula," in *In Proceedings IRE*, vol. 34, 1996.

[8] H. Xia, H. Bertoni, L. Maciel, A. Lindsay-Stewart, and R. Rowe, "Radio propagation characteristics for line-of-sight and personal communications," in *IEEE Transactions on Antennas and Propagation*, oct 1993, vol. 41, no. 10, pp. 1439–1447.

[9] (2003, Sept.) Mobile ad-hoc networks (manet) Working Group. Internet Engineering Task Force. [Online]. Available: http://www.ietf.org/html.charters/manet-charter.html

[10] E. M. Royer and C.-K. Toh, "A review of current routing protocols for ad-hoc mobile wireless networks," *IEEE Personal Communications Magazine*, vol. 6, no. 2, Apr. 1999.

[11] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Comp. Commun. Rev*, London, UK, Aug. 1994, pp. 234–244.

[12] L. R. F. Jr and D. R. Fulkerson, *Flows in Networks*. Princeton University Press, 1962.

[13] C.-C. Chiang, H.-K. Wu, W. Liu, and M. Gerla, "Routing in clustered multihop, mobile wireless networks," in *Proceedings of IEEE Singapore International Conference on Networks (SICON'97)*, Singapore, Apr. 1997, pp. 197–211.

[14] C. E. Perkins and E. Royer, “Ad-hoc on-demand distance vector routing,” in *Proceedings of MobiCom ’99*, New Orleans, USA, Feb. 1999, pp. 90–100.

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

[16] S. Floyd and V. Jacobson, “The synchronization of periodic routing messages,” *IEEE/ACM Transactions on Networking*, vol. 2, no. 2, pp. 122–136, 1994. [Online]. Available: citeseer.ist.psu.edu/floyd94synchronization.html

[17] M. Heissenbüttel, T. Braun, T. Bernoulli, and M. Wälchli, “Blr: Beacon-less routing algorithm for mobile ad-hoc networks,” *Elsevier’s Computer Communications Journal (ECC)*, vol. 27, no. 11, pp. 1076–1086, 2003, submitted.

[18] H. Takagi and L. Kleinrock, “Optimal transmission ranges for randomly distributed packet radio terminals,” *IEEE Trans. Commun.*, vol. 32, no. 3, pp. 246–257, Mar. 1984.

[19] T.-C. Hou and V. Li, “Transmission range control in multihop packet radio networks,” *IEEE Trans. Commun.*, vol. 34, no. 1, pp. 38–44, Jan. 1986.

[20] J. Nonnenmacher and E. W. Biersack, “Scalable feedback for large groups,” *IEEE/ACM Trans. Networking*, vol. 7, no. 3, pp. 375–386, June 1999.

[21] I. S. Committee, *Wireless LAN medium access control (MAC) and physical layer (PHY) specifications*, IEEE IEEE 802.11 Standard, 1997, iSBN 1-55937-935-9.

[22] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath, “Flooding for reliable multicast in multi-hop ad hoc networks,” in *Proceedings of the International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication (DIALM)*, 1999, pp. 64–71.

[23] J. Jetcheva, Y. Hun, D. Maltz, and D. Johnson, *A simple protocol for multicast and broadcast in mobile ad hoc networks*, IETF Internet Draft, July 2001, draft-ietf-manet-simple-mbcast-01.txt.

[24] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, “The broadcast storm problem in a mobile ad hoc network,” in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, 1999, pp. 151–162.

[25] B. Williams and T. Camp, “Comparison of broadcasting techniques for mobile ad hoc networks,” in *MOBIHOC’02*, EPFL, Lausanne, Switzerland, June 2002.

[26] H. Lim and C. Kim, “Multicast tree construction and flooding in wireless ad hoc networks,” in *Proceedings of the ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*, 2000.

[27] W. Peng and X. Lu, “On the reduction of broadcast redundancy in mobile ad hoc networks,” in *Proceedings of MOBIHOC*, 2000.

[28] A. L. A. Quayyum, L. Viennot, “Multipoint relaying: An efficient technique for flooding in mobile wireless networks,” INRIA, Sophia Antipolis, France, Tech. Rep., 2000.

[29] T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Quayyum, and L. Viennot, *Optimized link state routing protocol*, IETF Internet Draft, Sept. 2001, draft-ietf-manet-olsr-06.txt.

[30] W. Peng and X. Lu, “Ahbp: An efficient broadcast protocol for mobile ad hoc networks,” *Journal of Science and Technology*, 2002.

[31] (2003, Sept.) Qualnet. Scalable Network Technologies (SNT). [Online]. Available: http://www.qualnet.com/

[32] U. of California. (2000, Dec.) Glomosim scalable mobile network simulator. UCLA Parallel Computing Laboratory. Software Package. [Online]. Available: http://pcl.cs.ucla.edu/projects/glomosim/

[33] T. Camp, J. Boleng, and V. Davies, “A survey of mobility models for ad hoc network research,” *Wireless Communications and Mobile Computing (WCMC)*, vol. 2, no. 5, pp. 483–502, 2002, special issue on Mobile Ad Hoc Networking.

[34] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, “A performance comparison of multi-hop wireless ad hoc network routing protocols,” in *4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM’98)*, Oct. 1998, pp. 85–97.

[35] W. Navidi and T. Camp, “Stationary distributions for the random waypoint mobility model,” *IEEE Transactions on Mobile Computing*, vol. 3, no. 1, pp. 99–108, 2004.

[36] J. Yoon, M. Liu, and B. Noble, “Random waypoint considered harmful,” in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*, 2003, to appear.

[37] E. M. Royer, P. M. Melliar-Smith, and L. E. Moser, “An analysis of the optimum node density for ad hoc mobile networks,” in *Proceedings of the IEEE International Conference on Communications (ICC 2001)*, 2001, pp. 857–861.

[38] C. Bettstetter and C. Wagner, “The spatial node distribution of the random waypoint mobility model,” in *Proceedings of the First German Workshop on Mobile Ad-Hoc Networks (WMAN)*, 2002, pp. 41–58, gI Lecture Notes in Informatics.

[39] L. Blazevic, S. Giordano, and J. Y. Le Boudec, “Self-organizing wide-area routing,” in *Proceedings of SCI 2000/ISAS 2000*, Orlando, July 2000.

[40] M. Heissenbüttel, T. Braun, M. Wälchli, and T. Bohbot, “Optimizing neighbor table accuracy of position-based routing algorithms,” in *Submitted to Infocom 2005*, Miami, USA, Mar. 2005.

[41] R. Nelson and L. Kleinrock, “Spatial TDMA: A collision-free multihop channel access protocol,” *IEEE trans. on commun.*, vol. COM-33, 9, pp. 934–944, 1985.

[42] G. Karumanchi, S. Muralidharan, and R. Prakash, “Information dissemination in partitionable mobile ad hoc networks,” in *Symposium on Reliable Distributed Systems*, 1999, pp. 4–13. [Online]. Available: citeseer.ist.psu.edu/karumanchi99information.html

[43] M. Stemm and R. H. Katz, “Measuring and reducing energy consumption of network interfaces in hand-held devices,” *IEICE Transactions on Communications*, vol. E80-B, no. 8, pp. 1125–31, 1997. [Online]. Available: citeseer.ist.psu.edu/stemm97measuring.html

[44] L. M. Feeney, “An energy consumption model for performance analysis of routing protocols for mobile ad hoc networks,” *Mobile Networks and Applications*, vol. 6, no. 3, pp. 239–249, 2001. [Online]. Available: citeseer.ist.psu.edu/feeney00energyconsumption.html

[45] M. Mauve, J. Widmer, and H. Hartenstein, “A survey on position-based routing in mobile ad-hoc networks,” *IEEE Network Magazine*, vol. 15, no. 6, pp. 30–39, Nov. 2001.