Alcom-FT Technical Report Series
ALCOMFT-TR-03-129

A Simulation Environment for Ad Hoc Networks Using Sector Subdivision

Klaus Volbert*
Heinz Nixdorf Institute and
Department of Mathematics and Computer Science
University of Paderborn, Germany
kvolbert@uni-paderborn.de

Abstract

In this paper a new model for communication in MANETs will be presented: Instead of omnidirectional transmissions, as assumed in most papers and all existing systems, the members are allowed to submit data in a fixed number of different directions (sector subdivision) and to adjust the transmission power in each sector separately. A simulation environment (Simulation Environment for Ad Hoc Networks, SAHNE) will be presented that allows simulation of communication strategies in MANETs that use sector subdivision, and simulation results will be shown where communication paths are selected via hop-minimization or geometric spanner properties. SAHNE is based on C++ and common libraries, which ensures that it can be used on many different platforms. The experiments show the influence of different parameters in realistic szenarios, and using geometric routing seemed to be better than using hop-minimization.

1 Introduction

A Mobile Ad Hoc Network (MANET) is an autonomous system of mobile hosts connected by wireless links [CM98]. It is not centrally controlled and there is no static infrastructure such as base stations. If two hosts are not within their radio range, all message communication between them must pass through one or more intermediate hosts that act as routers. Every node organizes itself in the network and can move around without any restriction. A MANET is a very dynamic network with small bandwidth compared to normal hard-wired networks. Many routing protocols have been developed for the purpose of MANETs. Most of them work omnidirectionally and with fixed transmission power . In these protocols, the routing task is performed using either a proactive or a reactive scheme, or a combination of both. Examples of proactive protocols that have been suggested for wireless networks are OLSR, DSDV, and WRP. Examples of reactive protocols are DSR, LAR, RDMAR, AODV, and TORA. A combination of both concepts can be found in ZRP and LANMAR. Further details are given in [Per01].

Normally, mobile nodes are equipped with omnidirectional antennas and the number of frequencies for data communication is limited, so that it makes sense to deal with one frequency only. In this work the assumption is that every node is allowed to submit data in a fixed number of different directions and to adjust the transmission power in each sector separately. Information can be transmitted directionally and simultaneously in different sectors. A first guess is that power could be decreased and battery lifetime could be increased. A new simulation environment has been developed to analyze this model. It supports the simulation of power-controlled MANETs with sector-subdivision. The routing protocols, mentioned above, do not use power-control and sector-subdivision, so that new distributed algorithms are necessary to realize communication in such MANETs. Some initial simple strategies have been developed and tested with SAHNE. This work is part of a project where a prototype communication system is developed based on infrared directed communication. The prototype will be able to communicate in eight sectors independently with adjustable transmission powers. Furthermore, it can be used as an extension module for the mobile mini robot Khepera. Thus, beside computer simulations, the communication strategies will also be validated under practical conditions.

The remainder of this paper is structured as follows: In Section 2 a motivation and the theoretical background of the new model will be given. In Section 3 the new (communication) model used in SAHNE will be described. Section 4 illustrates the design and implementation of the new simulation environment and in Section 5 simple communication strategies will be presented. Finally, Section 6 concludes this work.

*Partially supported by the DFG-Sonderforschungsbereich 376 and the IST Programme of the EU under contract number IST-1999-14186 (ALCOM-FT)

2 Motivation and Background

In many kinds of MANETs, the mobile nodes operate on battery power. There are two ways that they do this. First, they might transmit data to a desired recipient. Second, a mobile node might offer itself as an intermediate forwarding node for data going between two other nodes in the network. Providing such a service is likely to be costly in terms of power consumption, but without the availability of such forwarding nodes there can be no ad hoc network.

It is known that the longer the reception range is, the larger is the transmitted power. At first sight it could be that the longer the transmission range is, the better. But it is not always like that. Although a longer transmission range reduces the number of hops that a packet needs to transverse in a MANET, it also increases the number of nodes that locally compete on the shared channel, effectively increasing the access delay and reducing the network capacity. Furthermore, a short transmission range allows better frequency reuse and longer battery lifetime. The goal in this work has been to use only as much transmission power for the communication between two nodes as necessary. In every direction of any node exist some neighbors to interact with and clearly a goal should be to bound this number of neighbors.

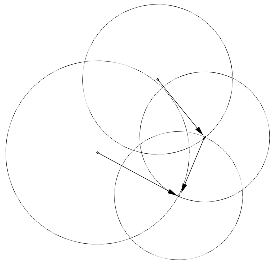

Figure 1. Omnidirectional transmissions

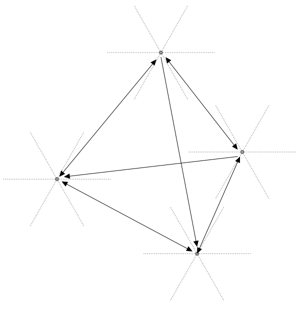

Figure 2. Transmissions with sector subdivision

On the one hand, for power-controlled MANETs with variable transmission range as they were considered in this paper, there do not exist any results for an abstract model of how to select routes or schedule transmissions of messages. Nevertheless results do exist in static power-controlled ad-hoc wireless networks [AS98] and in MANETs [CNS01], but without using any subdivision of space. In [KSV00] new MAC protocols suitable for ad hoc networks based on directional antennas were designed. On the other hand, the most prominent simulation environments for MANETs, NS-2 (USC, [FV98]) and GloMoSim (Global Mobile Simulation, UCSC [UCLA00]), are not designed for the simulation of MANETs based on this model considered in this work.

In the following point-to-point communication is considered, e.g. a node v wants to communicate with a neighbored node w. Normally, node v transmits data to w omnidirectionally with fixed transmission power. In this case it may be possible that many other nodes are blocked during this communication. This number of blocked nodes can be limited by using sector subdivision: v transmits data unidirectionally in that sector in which w lies. Additionally v could communicate with nodes in different sectors. To reduce power consumption every node uses variable transmission powers, that means node v adjusts the transmission power so that the transmission to w is possible, but it is not possible using lower transmission power. To decrease drainage and to increase battery lifetime it would also help to use omnidirectional transmissions with variable ranges, but then the main problem is, how to ensure connectivity in this one-sector case (see Figure 1). No results are known. In the case of, for example, using sector subdivision with 6 sectors, the connectivity can be guaranteed (see Figure 2).

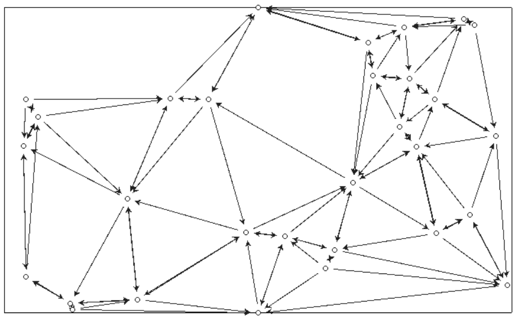

Figure 3. Θ-graph example

In the area of computer graphics, interesting and useful results have been achieved with the Θ-graph, which has been used here to ensure connectivity and to find first, simple communication strategies. In the following the results will be listed. Spanners were introduced to computational geometry by Chew [Chew86]. Let $f > 1$ be any real constant. Let the weight of an edge (p, q) be defined as the euclidean distance between p and q and let the weight of a path be defined as the sum of the weights of its edges. A graph with vertex set $V \subset \mathbb{R}^d$ is called a *spanner* of V with stretch factor f (f-spanner of V), if for every pair $p, q \in V$ there is a path in the graph between p and q of weigth at most f times the euclidean distance between p and q. If for every pair $p, q \in V$ there is only a path in the graph between p and q that does not leave the circle around p with radius $f \cdot |p - q|$, then V is named *weak spanner*. Every f-spanner is also a weak f-spanner. The basic data structure in this work will be the so-called Θ-graph or γ-angle graph (the same construction in both cases, for more details see [FMS97, KG92, RS91]). A description of this data structure follows (cf. [FMS97]). Fix an integer k. Let $\gamma = 2\pi/k$. To define the γ-sectors of a position $x \in \mathbb{R}^2$ draw k rays from x so that they form angles $2\pi(i-1)/k$, $i \in 1,..,k$, with the vertical line through x. These rays subdivide $\mathbb{R}^2$ into k sectors around x. The γ-angle graph on a (finite) set $V \subset \mathbb{R}^2$ has vertex set V. Each $x \in V$ has directed edges to the closest node in V lying in the k γ-sectors around x. Thus the graph has the out-degree k. Let E be the set of edges, then it holds: $|E| \leq n \cdot k$. Further results are (assumed that $G_k(V)$ specifics the γ-angle graph with $\gamma = 2\pi/k$):

- $G_k(V)$ is a weak f-spanner with stretch factor $f = \sqrt{3+\sqrt{5}} \approx 2.288$, if $k = 4$.
- $G_k(V)$ is a weak f-spanner with stretch factor $f = max\{\sqrt{1 + 48\, sin^4(\pi/k)}, \sqrt{5 - 4\, cos(2\pi/k)}\}$, if $k \geq 6$.
- $G_k(V)$ is a spanner with stretch factor $f = 1/(1 - 2\, sin(\pi/k))$, if $k \geq 7$.
- For weak spanner and for spanner, it holds: $\lim_{k\to\infty} f = 1$.

In the next section the new communication model will be presented and it will be described how these graph results have been used in this context.

3 Modeling

In this work a MANET will be modeled at time $t \in \mathbb{N}$ by a directed graph $G_t = (V, E_t)$. The set of vertices $V = \{1..n\}$ represents the members of a MANET. The set of edges E_t represents communication links at time t between nodes in V. A time reference is used since every node can move around without any restriction. The graph may change very quickly. At every point of time a node $v \in V$ has a well-defined position, given by the following injective function: $pos_t : V \rightarrow \mathbb{R}^d$ ($d \in \mathbb{N}$) with $pos_t(v) = (v_{x_1}, ..., v_{x_d})$. Each edge $e \in E_t$ has a weight w at every point of time. That is the power used to transmit information over an edge e, given by the function $weight_t : E_t \rightarrow \mathbb{R}$ with $weight_t(e) = w_e$ (e.g. $|e|^r$, for some $r > 1$). The communication links between nodes are defined as follows: Every node holds a constant number of directed senders with variable transmission power (e.g. infrared-sender). Let the maximal transmission power be *max_range*. Every sender will be received only within its sector and nodes are only allowed to communicate with their neighbors in these particular sectors. Now, the neighbors can be defined.

Definition 3.1 (neighbors) *Let the* nearest neighbor *of node $u \in V$ in sector j be defined as the node, having the smallest euclidean distance from u in j and being reachable from u with the given transmission power. The* l-nearest neighbor *of node $u \in V$ in sector j is located at position l in the list of nodes that is sorted ascendingly by the euclidean distance of the nodes that are in sector j and reachable from u with the given transmission power. All the nodes that are reachable from u with the given transmission power will be called* simple neighbors.

Let $|v, w|$ be defined as the euclidean distance for two nodes $u, w \in V$ and let $sect_j(v)$ be the set of nodes that are within sector j. Then the set of edges can be modeled in the following three different ways:

$$E_t^1 = \bigcup_{j=1}^{k} \Big\{ \{v,w\} \,\Big|\, v,w \in V, w \in sect_j(v) \text{ and } |v,w| = \min_{\substack{x\in V,\ x \in sect_j(v) \\ |v,x| \leq max\_range}} |v,x| \Big\}$$

Let $m \in \mathbb{N}$ be given and $l \in \{1..m\}$:

$$E_t^2 = \bigcup_{j=1}^{k} \Big\{ \{v,w\} \,\Big|\, v,w \in V, w \in sect_j(v), |v,w| \leq |v,x| \leq max\_range \text{ and } x \text{ is } l\text{-nearest neighbor of } v \Big\}$$

$$E_t^3 = \bigcup_{j=1}^{k} \Big\{ \{v,w\} \,\Big|\, v,w \in V, w \in sect_j(v) \text{ and } |v,w| \leq max\_range \Big\}$$

The remainder of the model is defined as follows: All nodes wishing to communicate with other nodes within the MANET are willing to participate fully in the protocols of the network. In particular, each node participating in the network should also be willing to forward packets for other nodes in the network. Information will be transmitted by *packets*. Every packet has a clear identification (ID), a source and a target. Furthermore, it can contain additional information such as parameters or data like files, audio- or videodata, text, etc. A packet needs one *time step* to make a single hop in the network, regardless of the distance to the destination node. In many situations it is more time- and ressource-efficient for a message to perform a sequence of hops instead of one single hop to its final destination. The sequence of nodes used for the hops is called *the route of a message*. This process is named *multi-hop-communication*. As already mentioned before, only one frequency is available for transmitting packets. That implies that a node can send out at most one packet at a time per sector. If a node v attempts to send a packet with transmission power t, then all nodes needing less than $\alpha \cdot t$ power to receive a packet from v are blocked, where $\alpha > 1$ is some fixed constant. Any information transmitted to a blocked node is not received (Interferences occur). Note, that this means that it is possible that no transmission during a single time step was succesful (cf. [AS98]). A transmission conflict caused by interferences cannot be detected by the sending node. The *hidden terminal problem* can arise due to the possibility that transmissions from two nodes which cannot hear each other, may interfere at a third node. Unfortunately, many packet radio network environments suffer from packet corruption due to this problem. All nodes work in a synchronized way.

4 Design and Implementation of SAHNE

SAHNE (Simulation Environment for Ad Hoc Networks, SAHNe) has been developed to analyze communication in MANETs based on the new model mentioned in the last section. Communication in networks is a very complex process and so a goal during the development of this tool has been to build a flexible, well-expandable and easy-to-use-environment, with which nearly every communication layer can be simulated. To achieve this goal, SAHNE has been designed with respect to an ISO/OSI reference model (see e.g. [Spohn93]) and the functionality of the nodes has been divided into several communication units. The number of these units is not fixed and at any time a new unit can be inserted. The main functional units that are already implemented in SAHNE have the following tasks.

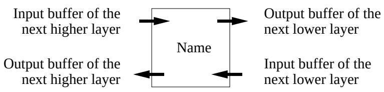

Figure 4. Layout of an unit

The *user* is equivalent to the application layer and it is comparable with a data transfer service. Data will be created there and destinations have to be chosen for the generated packets. The user can be extended by new application services that are arranged in higher layers.

As the first part of the network layer the *scheduler* determines what should be done with packets received by the user. Packets will be scheduled (e.g. forwarding packets via a random target that acts as a stopover, or splitting packets into smaller packets and vice versa).

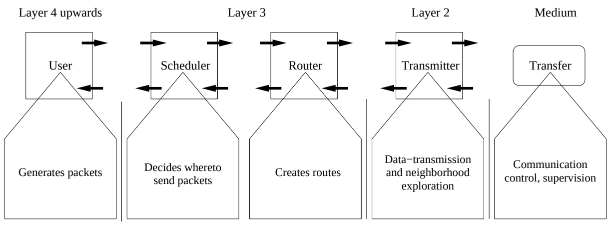

Figure 5. The communication units of SAHNE

As the second part of the network layer the *router* forwards packets received by the scheduler to its targets. Especially, there will be created and chosen routes and it will be decided how to transmit information along these routes. The router uses topology knowledge, e.g. it knows every node in the neighborhood.

The *transmitter* represents the MAC (Medium Access Control) layer of the data link layer. It is responsible for direct data transmissions between two neighboring nodes. In this layer, problems depending on the network architecture will be simulated such as the occurrence of interferences in MANETs. The transmitter acquires direct neighbors.

The *medium* models the communication channel. Data will be forwarded from its source to its target with consideration of physical channel properties.

Two nodes are able to communicate at every layer of this model. In general these functional units will be used in the following way to generate and to process network traffic. A user generates a packet with a target and forwards it to the scheduler of the same node. The scheduler decides whether to send the packet directly to the target, or via another node, or something like this, and forwards it to the router. The router knows where the target is and chooses a node that will get the packet next. This transmission will be performed by the transmitter. The last two steps will be repeated until the packet arrives at the target. Then it goes again through the scheduler to another scheduler or directly to the user. In fact, this is a very simple example.

SAHNE has been implemented in C++ using various data structures and several advanced data types of LEDA [MN99]. The development has been done with great care to be platform independent (SAHNE was tested on several operating systems: Solaris, Linux, Windows). The units in SAHNE are realized as C++ classes and the whole environment is based on an object-oriented model. SAHNE can be compiled in two different ways, either in textual mode or in interactive mode. In both cases simulation parameters will be read from a configuration file (e.g. a textfile called *sahne.cfg*), but additionally in the interactive mode they can be edited interactively via a graphical user interface (GUI). In the textual mode the simulation runs until the number of simulation steps will be finished, in the interactive mode the user is able to change parameters, check nodes/edges, view statistics, etc. online. At the end of a simulation with SAHNE, local and global statistics will be given and some files will be generated that can be used directly as an input for the gnuplot tool to produce statistics. As one can see in figure 6, in addition to communication units (network, node and the afore-mentioned modules) there exist further classes. The movement unit is the most interesting feature, because it regulates motion

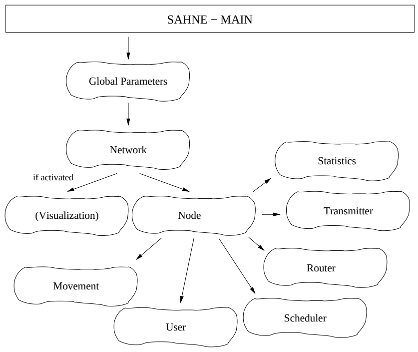

Figure 6. Concept of SAHNE

patterns. Every node can move around without restriction in a MANET and therefore motions have to be calculated at each simulation step. In SAHNE miscellaneous movement strategies have been implemented, which the following four examples will illustrate. Some known motion patterns such as the Brownian Motion Model, the Column Model and the Pursue Motion Model [Sanchez98] are also available in SAHNE.

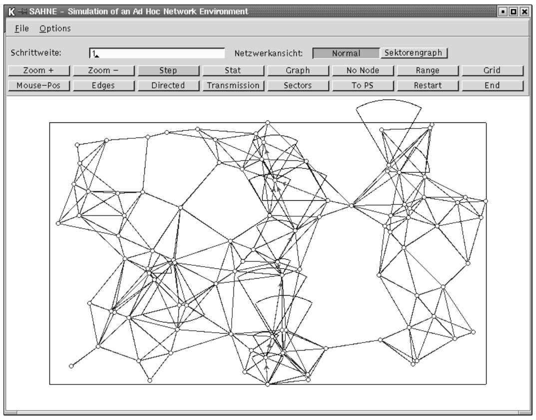

Figure 7. Visualization of data-transmissions

In SAHNE the assumption is that nodes will be placed (e.g. randomly or via a given scenario) in a rectangular box of size $[size\_x_1 \times ... \times size\_x_d]$ at the beginning of a simulation. After this placement they move around in this box, e.g. with a velocity v between min_speed and max_speed.

Random motion pattern In this model, a node chooses a destination with a uniform random distribution over the area, moves there with the above mentioned velocity v, waits for $pause\_time$ simulation steps, and then repeats this behaviour (cf. [Per01]).

Random motion pattern with variable reference point Here, a node chooses a destination with a uniform, random distribution over a given rectangular area of size $[mov\_rect\_x \times mov\_rect\_y]$, not over the whole area. The actual position of the node defines the center of this rectangle. Through this construction the elbowroom can be limited.

Random motion pattern with fixed reference point The first position of a node at the beginning of a simulation defines the center of the rectangular area as before in the random motion pattern with variable reference point. A node can only move around in this fixed and limited box.

Motion pattern with global arrival point In this model, there exists a global arrival point. This point will be chosen at the beginning of a simulation and is defined as the global target of all the nodes. During a simulation every node tries to take a step forward in the destination's direction with velocity v.

5 Simple Communication Strategies

In this section the algorithms implemented in SAHNE will be described. To enable communication in every layer of SAHNE several strategies have been developed under different assumptions. Note that these algorithms can be regarded as exemplary implementations. SAHNE can easily be extended by other strategies. Furthermore, note that the user can choose one algorithms from each unit in order to perform a simulation. The algorithms can be mixed arbitrarily and the selections can be done via a configuration textfile, via the given interface or online during a simulation.

Transmitter The transmitter is responsible for construction and maintenance of the Θ-graph. Nodes of a MANET are not centrally controlled and so distributed algorithms are needed. In this model the assumption is made that every node has a *GPS-Module* (Global Positioning System [DJ96]) to get its physical location. In reality, position information provided by GPS includes some amount of error, which is the difference between GPS-calculated coordinates and the real coordinates. In this case it will be assumed that each node knows its current location precisely. The transmitter collects the following information about each neighbor, depending on Definition 3.1: Number (resp. Name), Physical Location (GPS), Euclidean Distance (calculated via the position) and the time, at which the last information update by a packet took place. If the set of edges is E_t^1, the number of neighbors and the space requirement for neighbor information will be constant.

Let k, which is an adjustable parameter in SAHNE, be the number of sectors of the Θ-graph. If in sector $i \in \{1..k\}$ of a node more than one packet will be received, then all the packets will be removed. Otherwise the packet will be received correctly. The transmitter processes only received packets and uses this to collect the necessary information about direct neighbors. To produce neighbor information, every transmitter sends a new neighbor-control-packet in sector i per simulation step with the probability of $1/neighbor\_interval$. The transmission range of this will be chosen randomly from $\{1, .., max\_range\}$, may be fixed or something else. The transmitter rests for one step with constant probability of $1/bandwidth\_not\_use$ to reduce the number of collisions. Packets to higher layers will be forwarded as soon as they will be received completely. Packets from higher layers will be split before being forwarded if they are oversized. The maximum size of a packet and some other values can be set by the simulation parameters.

Router The routing unit handles and forwards packets in the order of their arrivals: First-In-First-Out (FIFO). But this can easily be extended to support other packet switching schemes. At the beginning of a simulation, the user is able to choose whether packets should be sent with acknowledgments or not. Some other parameters can also be adjusted. There exist two main routing strategies: the so-called *pq-routing* and the *hop-minimization*. Both algorithms use the information collected by the transmitter to route packets.

The *pq-routing* works as follows: assume a packet p arrives at node u. Let the target of p be the node w and i the sector of u in which w lies. Then p will be forwarded over a neighboring node v in sector i (via the *target-sector*). In this way p will be transmitted until the destination is reached. A precondition is that the targets of the packets are known.

The *hop-minimization* is a distance-vector algorithm that operates like the Destination Sequenced Distance Vector-protocol (DSDV, [Per01]). Every node creates a routing table that includes the destination's address (e.g. number of node), the number of hops required to reach the destination, the neighbor required to reach the destination with a minimal number of hops and the creation time of the information received regarding that destination, as originally stamped by the destination. To keep the data up to date, each node periodically (per simulation step with probability of $1/rout\_table\_interval$) broadcasts to all of its neighbors its current estimate of the shortest distance to every other node in the network without the information about the neighboring node (to avoid loops). To forward a packet p the router looks in the routing table at the entry regarding the destination of p and if it is not empty, p will be transmitted via the located neighbor. Otherwise the packet will be held. Storing of the routing table requires O(n) space per node for a network with n nodes.

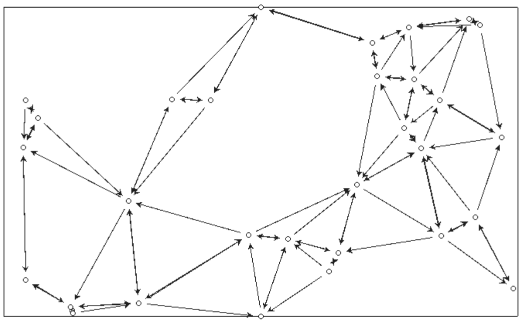

Figure 8. Θ-graph: limited transm. range

A problem of the pq-routing: in case that the maximum transmission range is very large or even unlimited, connectivity and correctness of pq-routing (all packets will reach their destinations) is guaranteed (see section 2, [FMS97]). Since the maximum transmission range is limited in MANETs, nodes can exist that do not have neighbors in certain sectors. In this case the Θ-graph could have very long edges (see figure 8, the same construction as in figure 3, but with limited transmission power). The number of these bad edges grows with an increasing number of sectors. A simple but not usable solution is to define spare-strategies. The first strategy forwards packets in the target-sector or it will hold packets, if the target-sector does not exist. Another strategy will use the sector on the left/on the right (each with probability of $1/2$), if the target-sector does not exist. This and further schemes have been implemented in SAHNE.

Scheduler As mentioned earlier, the scheduler is responsible for the scheduling of packets. Data packets can reach a size of 64 kByte (cf. max. TCP/IP packet size). To transmit packets error free and to reduce the number of collisions, all the packets should have a limited size that is much smaller than 64 kByte. For this reason the scheduler splits data packets that are larger than $max\_sched\_pack\_size$ into many several packets. These smaller packets will be forwarded and routed seperately. The scheduler forwards packets either directly or via a randomly selected node that acts as a stopover.

Parameter	*Value(s)*
Simulation area:	500 m × 300 m
Max. transm. range:	200 m
Number of nodes:	60 (2 fixed)
Simulation time:	10 s (= 100,000 simulation steps)
Bandwidth:	2 MBit/s (≈ 209 Bit/simulation step)
Link utilization:	80 %
Fixed packet size:	512 Byte
Injection:	100 packets/s (= exp. every 1,000 simulation steps)
Motion pattern:	random motion pattern, 15 km/h
Pause time:	2 s (= 20,000 simulation steps)
Number of sectors:	6
Upd. of routing inf.:	10 updates/s
Every node of the network creates data packets. The destinations for this packets will be chosen randomly. The transfers like a permutation routing.	

Table 1. Interesting simulation parameters

Node In the node unit, network traffic will be generated. Several selected nodes create data packets, choose destinations for it and forward it to the scheduler. The creation of packets will take place in one simulation step with the probability of $1/injection$, if packets are generated asynchronously. In case of synchronous creation packets will be injected every $injection$ simulation steps. The size of packets can be chosen randomly or it can be fixed.

A number of experiments were carried out modeling different possible situations. The most important parameters of the simulations are listed in table 1. At the beginning of a simulation the nodes will be placed randomly over the given simulation area. The considered parameters are the expected injection distance, the number of sectors of the Θ-graph, the transmission range and the velocity of the nodes. Finally, the accessibility will be illustrated depending on the number of sectors and on the transmission range.

In figure 9 the exploration of the expected injection distance is illustrated. The rate of successfully received packets of the pq-routing as well as of the hop-minimization decreases with an increasing number of generated packets. The pq-routing is better than the hop-minimization, since no further overhead is needed for transmissions. The number of interferences of the pq-routing is reduced.

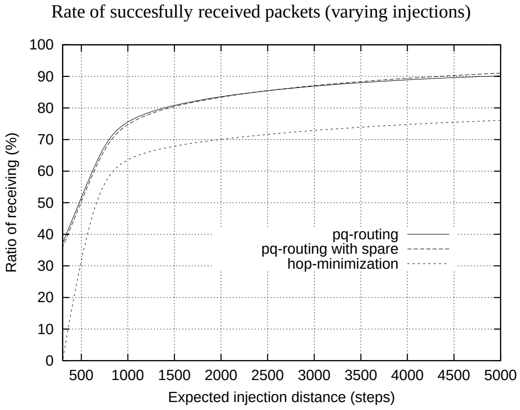

Figure 9. The expected injection distance

The number of sectors in the new model is a very interesting parameter. In figure 10 one can see that the ratio of receiving will be better, if very much sectors are available. But this conclusion is not always correct. Be careful, because in this simulation the transmission range is high (200 m).

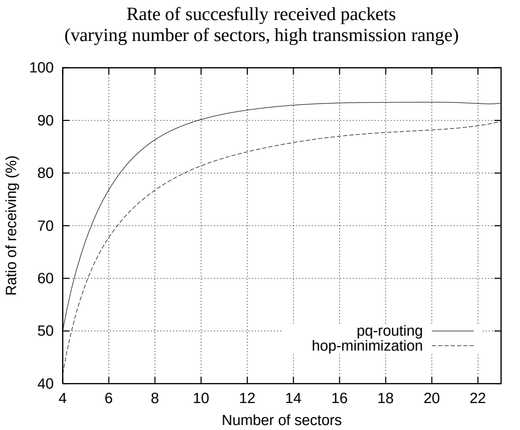

Figure 10. The influence of sector number

In figure 11, the expected result can be seen. The transmission range is low (100 m) and the pq-routing will be bad, if more than 7 sectors are available. The explanation is simple. The more sectors available the longer are the edges of the Θ-graph. Thus, many packets will be held at some nodes, if pq-routing is used. In this case, a spare strategy gives better results. But the hop-minimization will be the best, because it only uses edges that exist (pq-routing possibly tries to use sectors in which no edges are available).

A simple assumption is that the ratio of receiving will be better, if the transmission power is very high. Figure 12 shows that this is correct, but beginning with a certain transmission range the best ratio cannot be exceeded. If the transmission range is too high, the ratio will be worse because of the high number of collisions.

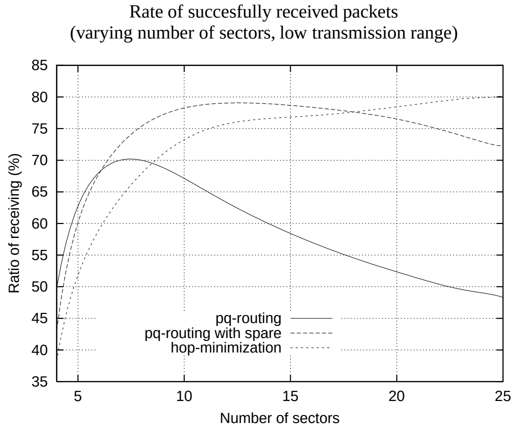

Figure 11. The influence of sector number by using small transmission ranges

The exploration of the influence of motion in figure 13 clarifies that a higher velocity entails a worse ratio of receiving. The hop-minimization is more concerned as the pq-routing. The necessary information to update the routing tables cannot arrive in time.

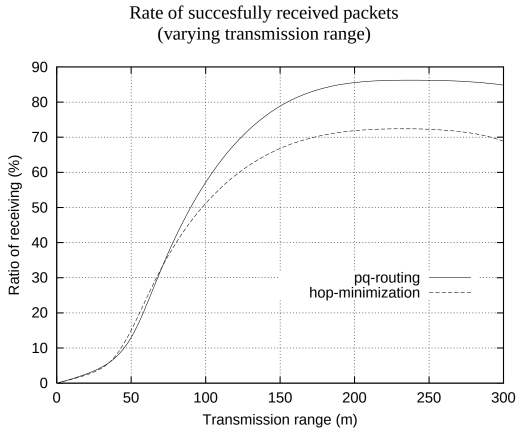

Figure 12. The transmission range

The last figure (14) gives an overview of the accessibility in the simulated MANETs. A node v will be accessible, if all other nodes can send v a packet. The number of correct routes was counted and compared to the number of all possible routes.

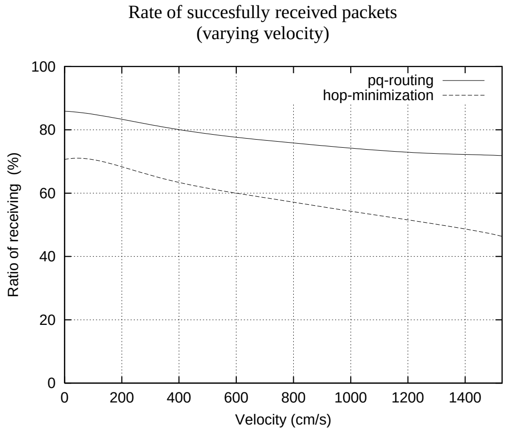

Figure 13. The influence of motion

6 Conclusions and Further Work

In this paper, a novel model for communication in MANETs was introduced. Based on this the basic concepts for the design of a simulation environment for mobile ad hoc networks were presented. SAHNE has already been used to analyze first, simple communication strategies for the directed communication model using sector subdivision and power-control.

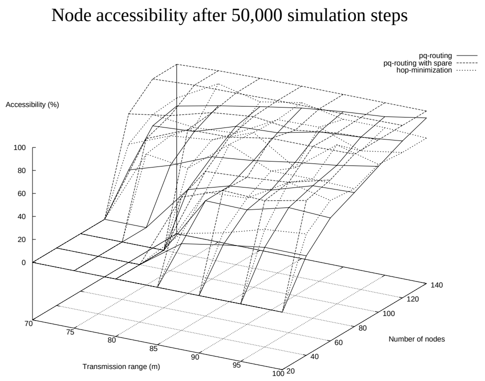

Figure 14. The accessibility

Further work can be done to extend the functional units of SAHNE and to create new modules. In the routing unit, other switching schemes can be implemented and analyzed in conjunction with routing algorithms. Finally, the channel model of the medium and transmitter layers can be upgraded to support a more practical modeling of the physical layer.

Acknowledgements

I developed this work based on my diploma thesis at the University of Paderborn which I finished in May 2001. I would like to thank my advisors Friedhelm Meyer auf der Heide and Christof Krick for their encouragement during the creation of my thesis and Andre Brinkmann for helpful discussions during the development of the simulation environment. Finally, I thank Matthias Fischer for the fundamental routine to calculate the Θ-graph with 6 sectors that I have used and extended in SAHNE.

References

[AS98] M. Adler and Ch. Scheideler. *Efficient Communication Strategies for Ad-Hoc Wireless Networks*. In Proc. 10th Annual Symposium on Parallel Algorithms and Architectures, 1998.

[Chew86] L.P. Chew. *There is a planar graph almost as good as the complete graph*. In 2nd Annual ACM Symposium on Computational Geometry, pages 169-177,1986.

[CM98] C. Corson and J. Macker. *Mobile ad hoc networking (Internet-Draft)*. Mobile Ad-hoc Network (MANET) Working Group, IETF, Oct. 1998.

[CNS01] I. Chatzigiannakis, S. Nikoletseas and P. Spirakis. *An Efficient Communication Strategy for Ad-Hoc Mobile Networks*. In Proc. 15th International Symposium on Distributed Computing, 2001.

[DJ96] G. Dommety and R. Jain. *Potential networking applications of global positioning systems (GPS)*. Tech. Rep. TR-24, CS Dept., The Ohio State University, April 1996.

[FMS97] M. Fischer, F. Meyer auf der Heide and W.-B. Strothmann. *Dynamic data structures for realtime management of large geometric scenes*. In 5th Annual European Symposium on Algorithms, pages 157-170, Springer Verlag, 1997.

[FV98] K. Fall and K. Varadhan *ns Notes and Documentation*. Web site at http://www.isi.edu/nsnam/ns, 1998.

[KG92] J.M. Keil and C.A. Gutwin. *Classes of graphs which approximate the complete Euclidean Graph*. Discrete and Computational Geometry, 7:13-28, 1992.

[KSV00] Y.B. Ko, V. Shankarkumar and N.H. Vaidya *Medium access control protocols using directional antennas in ad hoc networks*. In Proc. of IEEE INFOCOM '2000, 2000.

[MN99] K. Mehlhorn and S. Näher. *LEDA: A Platform for Combinatorial and Geometric Computing*. Cambridge University Press, 1999.

[Peleg00] D. Peleg. *Deterministic radio broadcast with no topological knowledge*. Manuscript, 2000.

[Per01] Ch. Perkins. *Ad Hoc Networking*. Addison Wesley, 2001.

[RS91] J. Ruppert und R. Seidel. *Approximating the d-dimensional complete Euclidian Graph*. In 3rd Canadian Conference on Computational Geometry, pages 207-210, 1991.

[Sanchez98] M. Sanchez. *RE: Mobility pattern in a MANET*. IETF MANET Mailing List, Sender: <misan@disca.upv.es>, http://www.disca.upv.es/misan/mobmodel.htm, June 25, 1998.

[Spohn93] D.L. Spohn. *Data Network Design*. McGraw-Hill, Inc., 1993.

[UCLA00] UCLA Parallel Computing Laboratory and Wireless Adaptive Mobility Laboratory. *GloMoSim: A scalable simulation environment for wireless and wired network systems*. http://pcl.cs.ucla.edu/projects/glomosim, 2000.