Alcom-FT Technical Report Series
ALCOMFT-TR-03-126

Performance of Distributed Algorithms for Topology Control in Wireless Networks

Stefan Rührup* Christian Schindelhauer† Klaus Volbert†
Matthias Grünewald‡

Heinz Nixdorf Institute, Paderborn University

Abstract

We try to close the gap between theoretical investigations of wireless network topologies and realistic wireless environments. For point-to-point communication, we examine theoretically well-analyzed sparse graphs, i.e. the Yao-graph, the SparsY-graph, and the SymmY-graph. We present distributed algorithms that can be used to build up these graphs in time $O(\log n)$ per node without the use of any geographical positioning system. Our algorithms are based only on local knowledge and local decisions and make use of power control to establish communication links with low energy-cost. We compare these algorithms with respect to congestion, dilation, and energy. For congestion we introduce different measures that allow us to investigate the difference between real-world wireless networks and models for wireless communication at a high level of abstraction. For more realistic simulations we extend our simulation environment SAHNE. We use a realistic transmission model for directed communication that uses sector subdivision. Finally, our experimental results show that our topologies and algorithms work well in a distributed environment and we give some recommendations for the topology control based on our simulations.

Keywords: Wireless networks, topology control, MAC, congestion, dilation, energy, distributed algorithms, simulation, power control

1 Introduction

In this paper we investigate topology control at the medium access layer (MAC) in wireless ad hoc networks. Our research aims at the implementation of such a network based on distributed robust communication protocols. We want to show how well-known sparse graphs with promising network and graph properties perform in practice. We use space multiplexing techniques and variable transmission powers to realize the topologies. Therefore, the network nodes, e.g. a colony of robots equipped with a suitable communication device, can send and receive radio or infrared signals independently in k disjoint sectors of angle θ using one communication channel. We call this ability sector subdivision. Furthermore, every device is able to regulate its transmission power for each transmitted signal.

On the one hand, a number of distributed topology control algorithms [20, 15, 14, 10], proximity graphs [11, 19], and geometric spanner graphs [3, 18, 5, 2] have been proposed that model communication networks at a high level of abstraction. On the other hand, we are currently developing a communication module for the mini robot Khepera [4] that can transmit and receive in eight sectors, using infrared light with variable transmission powers to show that these approaches are also suitable in practical situations. Here, we try to investigate theoretical results under realistic conditions given by the Khepera robots. We want to close the gap between theoretical investigations of wireless network topologies and real-world wireless environments. Since we concentrate on the interface between both parts, we have developed a simulation environment for wireless ad hoc networks, called SAHNE [17]. SAHNE allows to implement and test algorithms for topology control under realistic conditions.

The remainder of the paper is organized as follows. In Section 2, we first give the practical and the theoretical background of our work. We introduce the signal propa-

*DFG Graduate College "Automatic Configuration in Open Systems", `sr@hni.uni-paderborn.de`.

†Institute of Computer Science, {`schindel,kvolbert`}@ `uni-paderborn.de`. Partially supported by the DFG-Sonderforschungsbereich 376 and by the Future and Emerging Technologies programme of the EU under contract number IST-1999-14186 (ALCOM-FT).

‡Institute of Electrical Engineering, System & Circuit Technology, `gruenewa@hni.uni-paderborn.de`. Supported by the DFG-Sonderforschungsbereich 376.

gation model, that allows to simulate data transmissions between Khepera robots, and we introduce the topologies as well as some important definitions. In Section 3, we explain the extensions of our simulation environment that are necessary for realistic simulations. In Section 4, we give algorithms that construct the topologies without using any geographical positioning system (e.g. GPS), only based on local decisions and local knowledge. In Section 5, we measure the performance of our algorithms by presenting the results of extensive experiments. We conclude our work in Section 6 by proposing future research directions.

2 Preliminaries

In this section, we present our model and give the essential definitions which we use within our work. We introduce important aspects of the physical transmission of data that topology-control algorithms have to be aware of. Afterwards, the topologies we want to analyze are explained from a graph-theoretical point of view.

2.1 Practical Background: Signal propagation and reception

During wireless data transmissions between nodes, the bits have to be modulated in a waveform suitable for transmission over the air. The channel distorts the waveform in various ways. For example, the waveform can reach the receiver directly and via a reflection from an obstacle. The resulting two waves are phase-delayed and the superimposition of both waves is received by the destination node. Due to the phase-delay, *intersymbol interference* is observed at the receiver. This effect is called *multipath fading* and can be reduced by appropriate techniques at the receiver, e.g. maximum-likelihood sequence detection with the Viterbi algorithm [1, 13]. We assume that the physical layer solves this kind of problems. However, some aspects of the physical transmission have to be considered for performing realistic simulations suitable for developing medium access control algorithms. *Signal propagation* and *signal reception* are the most important ones. A model of the signal propagation is necessary since we want to adjust the transmission power dynamically. In wireless networks, several nodes can transmit signals simultaneously to one receiver, therefore the reception of signals has to be modelled to decide when interfering signals cause collisions.

We first describe the propagation model used. For unidirectional communication, we adapt a well-known model from directed infrared (IR) communication, that shows the relation between the transmission power P_t and the received power P_e:

$$P_e = A_{\mathit{eff}}(\psi)\frac{1}{d^2}R_t(\varphi)P_t \tag{1}$$

where $A_{\mathit{eff}}(\psi)$ is the effective area of the IR sensor, d is the distace between sender and receiver, and $R_t(\varphi)$ is the radiant intensity of the IR transmitter [7]. The angular characteristics of the sensors and the diodes can be accurately modelled by

$$f(\alpha) = c[(m+1)/2\pi]\cos^m\alpha, \;\; m = -\ln 2/\ln(\cos\alpha_{1/2}) \tag{2}$$

where c accounts for the characteristics of the optical transducers (e.g. area of the receiver) and $\alpha_{1/2}$ is the semi-angle at which half of the signal intensity is emitted / detected.

Signal reception depends on the received signal strength P_e of the data signal. If it is high enough compared to interfering signals and noise, the data signal can be extracted from the received superimposed signal. The *signal-to-interference-plus-noise-ratio* ($SINR$) expresses this relation [8]:

$$SINR = \frac{P_{e,d}}{\sum_{i=1}^{m} P_{e,i} + P_n} > SINR_{min} \tag{3}$$

where $P_{e,d}$ is the signal strength of the data signal, $P_{e,i}$ are the signal strengths of the m interfering signals and P_n is the noise power. If the $SINR$ exceeds the threshold $SINR_{min}$, the data signal can be decoded with a low *bit error rate* (BER). The threshold $SINR_{min}$ is usually chosen such that the BER is below 10^{-6}.

We have extended our simulation environment SAHNE accordingly. The propagation models are used to calculate the signal strength of transmitted data packets. The packet with the highest reception power is selected as the data packet and the $SINR$ is computed. If it exceeds the threshold $SINR_{min}$, the packet has been correctly received. All other interfering packets are discarded. A collision is produced if eq. 3 is not fulfilled.

2.2 Theoretical Background: Graph theory

In this work we investigate experimentally several known topologies for wireless networks to build up a basic structure with nice communication properties. Our underlying hardware model allows to communicate in k disjoint sectors in parallel. In [3] it turned out that for this model the Yao-graph and its variants perform well.

In the **Yao**-graph a directed edge (u, v) between two vertices u and v will be established, if in the sector with u as starting point v is the nearest neighbor to u. The **SymmY**-graph consists of all edges (u, v) of the Yao-graph for which the reversed edge (v, u) is also part of the Yao-graph. The **SparsY**-graph is an indegree-bounded subset of the Yao-graph. If a vertex v has two incoming edges (u_1, v) in a receiving sector and (u_2, v) in the Yao-graph, then the SparsY-graph discards the longer edge. Figure 1 helps to get an overview how these topologies appear and how one

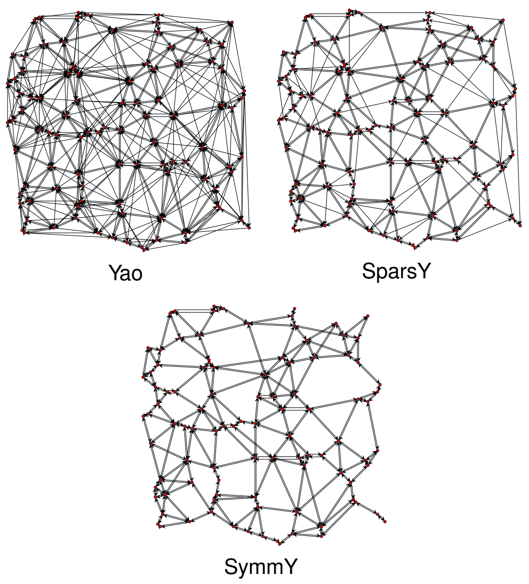


Figure 1. Yao, SparsY, and SymmY for a random vertex set with 100 nodes and 8 fixed sectors per node

topology differs from another one. Of course, problems will occur if there is no unique nearest neighbor. Therefore in [3] it is assumed that all vertices are in general position, which particularly means that there is always only one nearest neighbor in each sector. The following relationship follows directly by the definitions: Let V be a vertex set. Then,

$$\text{SymmY}(V) \subseteq \text{SparsY}(V) \subseteq \text{Yao}(V) \ .$$

Note, that these topologies are *sparse graphs*, since the number of edges in the graph is given by $O(n)$ where n is defined as the number of nodes in V. Let us introduce some definitions before we give further properties. As we allow only one transmission frequency (one channel), we may experience an **interference**, so when two or more packets are transmitted at the same time, either one packet or none can be received. An edge (u, v) interferes with an edge (r, s), if $\sphericalangle(s, r) = \sphericalangle(s, u)$ and $\sphericalangle(u, s) = \sphericalangle(u, v)$ and $D(u, v) \geq D(u, s)$ where $\sphericalangle(u, v)$ describes the sector number of u in which v lies and $D(u, v)$ means the distance between u and v. Now, let $\text{Int}(e)$ contain all edges e' that interfere with e. Then $|\text{Int}(e)|$ gives the *number of interfering edges*. A directed graph will be *strongly connected* if for all u, v there exists a directed path from u to v. A graph G will be called a (c, d)-*power spanner* if for all u, v there exists a path $(u = u_0, u_1, \ldots, u_m = v)$ in G such that

$$\sum_{i=1}^{m} (|u_{i-1}, u_i|_2)^d \ \leq \ c \cdot \min_{(u=v_0, v_1, \ldots, v_{m'}=v)} \sum_{i=1}^{m'} (|v_{i-1}, v_i|_2)^d$$

for some constant c. If $d = 1$, we call such a graph a c-spanner. A graph G will be called *power-spanner graph* if G is a (c, d)-power spanner for all d. In a *weak c-spanner* G for each vertex pair u, v, there exists a path P in G such that for all $w \in V(P) : |u, w|_2 \ \leq \ c \cdot |u, v|_2$ for some constant c.

Theorem 1 *[3] For a vertex set V with n vertices in general position in $\mathbb{R}^2$ the Yao-graph variants provide the following properties if sector subdivision is used.*

Topology	*Yao*	*SparsY*	*SymmY*
Out-degree	$\leq k$	$\leq k$	$\leq k$
In-degree	$\leq n-1$	$\leq k$	$\leq k$
Degree	$\leq n-1$	$\leq 2k$	$\leq k$
Interfering edges	$\leq n-1$	1	0
Strongly connected	*yes*	*yes*	*yes*
Weak spanner	*yes*	*yes*	*no*
Power spanner	*yes*	*yes*	*no*
Spanner	*yes*	*open*	*no*

The spanner-property immediately implies the power spanner property, which implies the weak spanner property, which implies connectivity. The connection between these graph properties and network properties is described by the following theorem:

Theorem 2 *[3] If a graph G is a power spanner, then for every pair of vertices there will exist a path in G which approximates the minimal energy path by a constant factor. If a weak spanner graph G has interference number I, then for every communication demand there will exist a routing using paths of the graph G such that the congestion approximates the minimal congestions by a factor of $O(I \log n)$.*

3 Simulation Environment

We have developed and investigated distributed algorithms for topology control in wireless networks using sector subdivision. To close the gap between mathematical analyses and realistic wireless environments, we have performed simulations. Since we allow sector subdivision, we use our simulation environment for mobile ad hoc networks, shortly called SAHNE [17], that enables us to simulate such communication capabilities. It is based on C++ and common libraries which ensure that it can be used on many different platforms. It has been designed with respect to the ISO/OSI reference model, e.g. [16], therefore the protocol stack of the nodes has been divided into several communication units that represent the different layers. The whole environment is based on an object-oriented model in which each unit is represented by an own class. The main unit that we have worked on for this paper is the transmitter, which represents the medium access control (MAC) layer. Here, we have implemented our algorithms. Further, we have extended the simulation kernel with the described realistic signal propagation model (see 2.1). The following table shows new SAHNE extensions and some differences between the former, idealized simulation model assumed in [17] and the newer, more realistic model.

Idealized Model	Realistic Model
Sectors with fixed borders	Ellipsoidal transmission range
Fixed sector orientation	Variable sector orientation
Transmitted signals will be received in at most one sector	Transmitted signals will be received in different sectors depending on the receiver characteristic
Interferences result always in complete data lost	Signal-to-interference-ratio (SIR)

The new simulation kernel allows to simulate effects like interfering signals (SIR), overlapping of sectors, or rotation of senders (see fig. 2).

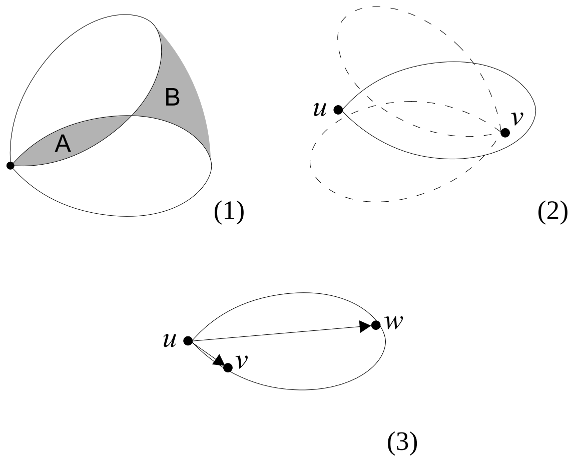


Figure 2. (1) Because of the directional characteristic there are overlapping regions (A) and regions where no reception is possible (B). That leads to the non-reciprocity of the channel (2): A node u can reach v but not vice versa, though both u and v use the same transmission power. A difference to the idealized transmission model is also shown (3): node u can reach v and w with the same transmission power.

4 Distributed Algorithms

A main advantage of the topologies Yao, SparsY, and SymmY is that they can be constructed locally without using any geographical positioning system (e.g. GPS). We have implemented a distributed algorithm for constructing these topologies: The Sector-Based Topology Control (SBTC)-Algorithm builds up the Yao-graph. It takes interferences into account and does not need a reciprocal channel. Transmissions with maximum power are avoided if possible. The algorithm can easily be extended with the capability to build up the SparsY or SymmY topology.

According to our simulation model the algorithm must comply with the following preconditions: A node can transmit and receive messages in k sectors independently. It can vary the transmission power. There is a fixed number of power levels available. It can detect interferences, but it cannot determine how many signals interfere. A node knows neither its position nor its orientation.

The SBTC-Algorithm, that builds up the Yao topology, is presented in figure 3. The main task of the algorithm is to find the nearest neighbor in each sector. The nearest neighbor is the node that can be reached with the minimal power level. It is determined by exchanging the so called *control messages*. The algorithm consists of two phases:

During the **first phase** a node (called the *initiator*) searches for a neighbor independently in each sector: First it sends a “Hello”-message with the minimal power. If no one answers, it increases the power so that the range is doubled. This is repeated until an acknowledgement is received or the maximum power is reached. Interferences between two or more acknowledgments, that are received simultaneously, are resolved by an exponential backoff algorithm. This is necessary because the initiator cannot distinguish if an interference is caused by nodes within the sector or nodes outside the sector. Acknowledgements are transmitted with the same power as the transmission power of the “Hello” message. During the **second phase** the power is adapted to the neighbor found in the first phase (called the *responder*): With repeated transmission of power control messages the initiator performs a binary search. The upper bound is the power level from the first phase. The lower bound is the minimum power level. Acknowledgements from the responder are transmitted with the power of the last acknowledgement in the first phase. This is crucial for the reachability of the nodes, because the channel is not reciprocal.

After these two phases, the initiator can establish an edge to the responder. If a neighbor cannot receive a “Hello”-message due to interferences, the initiator does not find him or establishes an edge to another node. Therefore the search is repeated. If the initiator finds the same neighbor again, the search interval is increased to reduce power consumption and interferences. That way we obtain the Yao topol-

Preliminaries

CREATEEDGE($addr$, p, s) establishes an edge to the node with the address $addr$, that can be used by sending a message in sector s with the transmission power p.
UPDATEEDGE($addr$, p, s) updates the transmission power.
DELETEEDGE($addr$, s) deletes an edge.
SEND($type$, $target$, p, s) sends a message with to the node with the address $target$ in sector s with transmission power p.

$1/\delta$ is the expected length of an interval between updating a neighbor.
M is a list of messages a node receives.
$N[s]$ denotes information about the neighbor in sector s, containing the following data:
$addr$ is the address of the neighbor.
p_{send} is the transmission power needed to reach the neighbor.
p_{recv} is the received power of a message received from the neighbor.

The Algorithm

```
SBTC()
for t ← 1 to ∞
do for s ← 1 to sectors
    do parallel
        with Probability δ
        do UPDATENEIGHBOR(s)

UPDATENEIGHBOR(s)
N_old ← N[s]
FINDYAONEIGHBOR(s)
if N[s] ≠ NIL
  then if N_old = N[s]
          then UPDATEEDGE(N[s].addr, N[s].p_send, s)
               decrease δ
          else CREATEEDGE(N[s].addr, N[s].p_send, s)
               reset δ
               if N_old ≠ NIL
                 then DELETEEDGE(N_old.addr, s)
  else if N_old ≠ NIL
          then DELETEEDGE(N_old.addr, s)
               reset δ
```

```
FINDYAONEIGHBOR(s)
p ← 1                                         // phase 1
a ← 1
while M = ∅ and p ≤ p_max
do M ← ∅
   interferences ← false
   SEND(Hello-Packet, undef, p, s)
   wait 2^a time steps
   and add received acknowledgements to M
   update interferences
   if M = ∅
     then if interferences
            then a ← a + 1
            else p ← 4 · p                    // doubling the range
                 a ← 1
if M = ∅
  then ABORT
p_low ← 1                                     // phase 2
p_high ← p
a ← 1
while p_low + 1 ≤ p_high
do M ← ∅
   interferences ← false
   SEND(Power-Control-Packet, undef, p, s)
   wait 2^a time steps
   and add received acknowledgements to M
   update interferences
   if M = ∅
     then if interferences
            then a ← a + 1
            else p_low ← p
                 a ← 1
                 p ← (p_low + p_high)/2
     else p_high ← p
          a ← 1
          p ← (p_low + p_high)/2
if M = ∅
  then ABORT
msg ← message with the max. received power in M
if (p < N[s].p_send) or
   (p = N[s].p_send and msg.p_recv > N[s].p_recv)
  then N[s].no = msg.sender
       N[s].p_send = p
       N[s].p_recv = msg.p_recv
```

Figure 3. The Sector-Based Topology Control Algorithm (SBTC)

ogy. To avoid interferences while searching neighbors the initiator must not start the search in adjacent sectors at the same time. After the neighbors have been found the transmission powers are adjusted, so data transmission in adjacent sectors is possible, unless the angular characteristics of the receiver does not allow it. If the opening angle of the receiver is high, a node has to shedule the communication on ingoing edges (this is a problem when using non-reciprocal channels) or it has to acknowledge every message. Furthermore, we want the algorithm to react on node failures and mobility so we infinitely repeat the search.

A modified version of this algorithm constructs the SparsY or the SymmY topology: For the SparsY topology it is necessary that every node keeps track of its ingoing edges. If the initiator wishes to establish an edge to the responder, he first has to apply for this edge. If the responder knows no other ingoing edge in the corresponding sector that is “shorter”, then the new edge is accepted. If the new edge replaces another ingoing edge, the responder has to inform the owner of the old edge.

In the case of the SymmY topology, the nodes also have to apply for an edge to a neighbor. If the initiator applies for an edge and if he is already known to the responder as a Yao-neighbor, then the requested edge can be established on both sides. So the nodes do not have to store information about ingoing edges.

Theorem 3 *For a vertex set V in general position with n nodes and s power levels per node Yao, SparsY, and SymmY can be constructed in time $O(\log n \cdot \log s)$ (the time one node needs to find its neighbors).*

Proof: Phase 1 uses power doubling and needs $O(\log s)$ steps until some first nodes will be reached. The time needed for sending a successful acknowledgement can be bounded by $O(\log n)$, since at most all nodes could answer and in this case we need the time to resolve the collisions by the binary exponential backoff algorithm. Phase 2 is just a binary search algorithm based on the number of power levels. In this phase we need at most $O(\log s)$ steps to adjust the transmission power to the nearest neighbor and at each of these steps $O(\log n)$ time slots to resolve collisons. ■

5 Experimental Results

In [12, 3] we investigate the basic network parameters congestion (that takes interferences into account), dilation, and energy. In this work we extend the definition of congestion to practical environments where interferences are modeled by using the signal-to-interference-ratio (SIR) and the fact, that transmitted signals are received in more than one sector at a time. In our simulations we consider three types of congestions to measure the quality of topologies and algorithms. We begin each simulation with a set of nodes randomly placed in the simulation area. No edges are established at the beginning. Then we start an algorithm to build up one selected topology, e.g. Yao, SparsY, or SymmY. At some time steps we stop the topology control and calculate network and communication properties. For our congestion values we construct a permutation routing problem: every node u creates one packet for each possible destination node v. Now, we consider two path systems on the constructed topology. The path system $\mathcal{P}_d$ that optimizes **dilation**, which is given by the maximum of the lengths of all paths in $\mathcal{P}_d$, and the path system $\mathcal{P}_e$ that optimizes **flow energy**, which is defined by $\sum_{e \in E(\mathcal{P}_e)} \ell(e)|e|^2$. Both schemes can be computed in polynomial time. Now, we simulate the transport of all packets and count the number of packets that go through an edge e and define it as the **load** $\ell(e)$ of e (This load is often called congestion in wired networks, compare [9]). We define the load L of a path system as $L := \max_{e \in E} \ell(e)$. In [12] we extend this definition to an intuitive definition of congestion in wireless networks. The **congestion** of an edge e is given by

$$C(e) := \ell(e) + \sum_{e' \in \mathrm{Int}(e)} \ell(e') \, .$$

The congestion C of a path system is defined by $C := \max_{e \in E} C(e)$. In this work we modify this parameter further and introduce the **realistic congestion** C_r. The realistic congestion combines load, interferences, power attenuation and SIR. The definition is the same as for congestion, but for the definition of interferences we take the realistic SIR into account. Let us assume, that transmissions take place on all edges. An edge interferes with another edge only if the receiver can not extract the transmitted signal from the received superimposed signal (cp. 2.1).

In our experiments we chose the following parameters: The nodes are placed randomly in an area of size 50m × 30m and also the sector orientations of the nodes are chosen at random. Every node has 8 sectors (transceivers) and

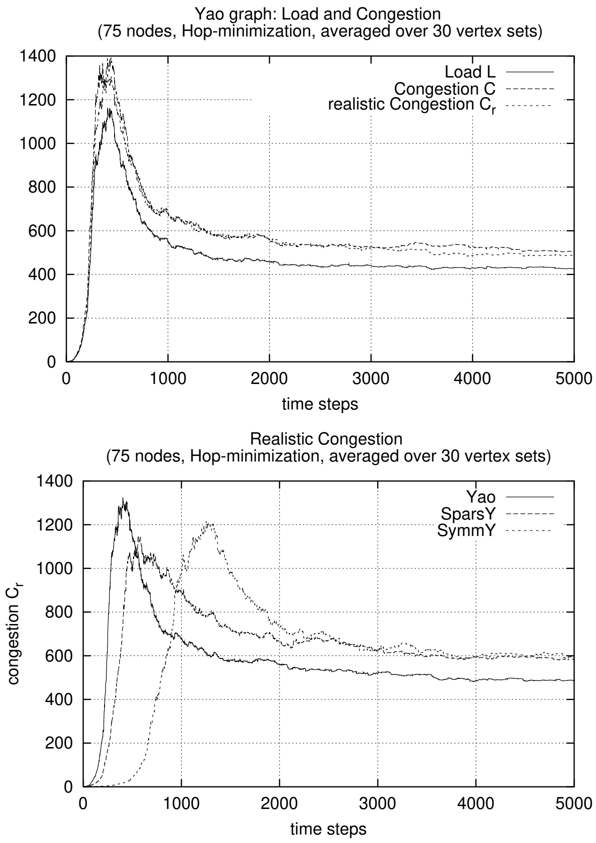


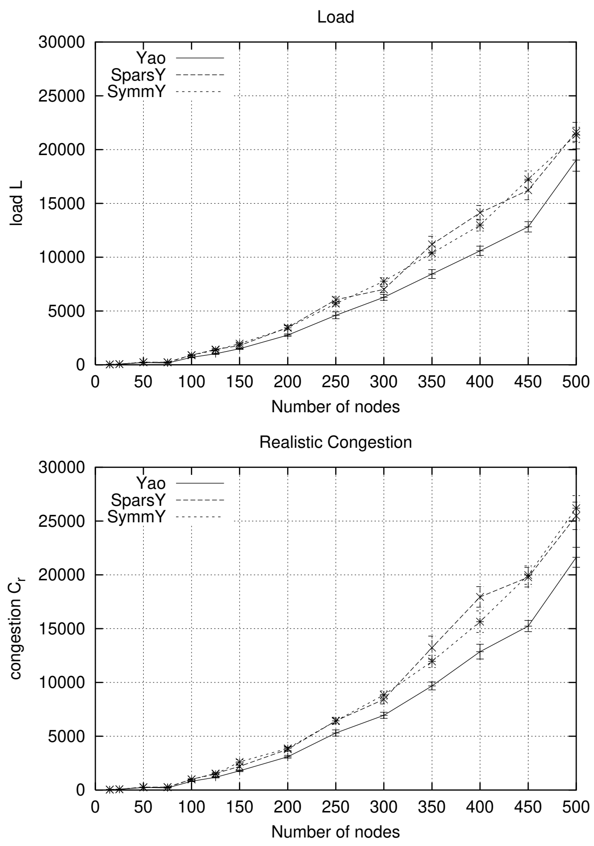
Figure 4. Load and congestion during network build-up. A time step is the transmission time for one control packet.

can change its transmission power at 256 power levels. The transmission range at maximum power is about 50m. The directional characteristic is based on the specification of the IR communication modules: The transmitter has a semi-angle of 20, the receiver a semi-angle of 50. The probability δ for repeating the UPDATENEIGHBOR procedure is initially set to $1/500$ (cp. fig. 3).

The upper diagram in figure 4 shows the progression of load and congestion during the build-up of the Yao topology. In every time step we do an offline-computation of an all-pairs shortest-path algorithm to obtain a path system on which congestion is calculated. The path system is computed with either hop minimization or energy minimization. As hop minimization yields better congestion, we do not present the results of energy minimization. The resulting values are averaged over 30 vertex sets.

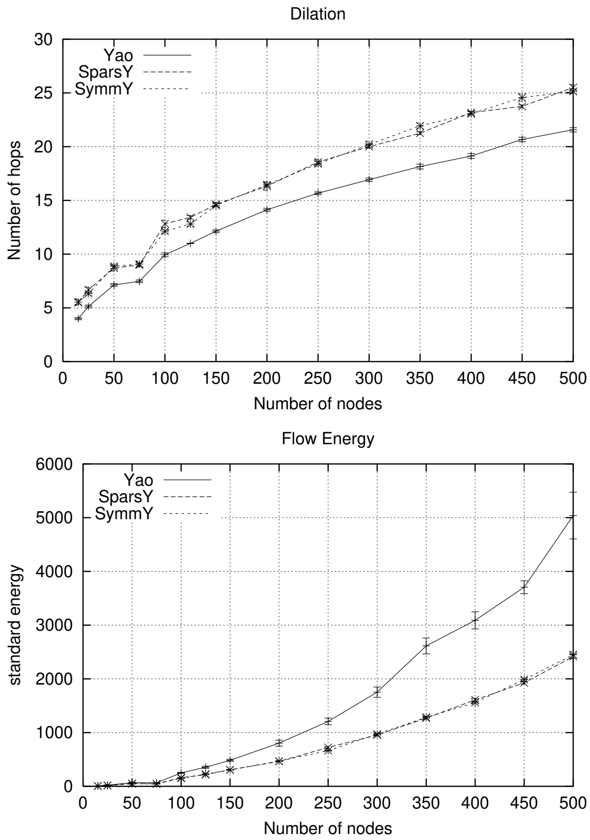
In the diagram all curves grow until a peak at 500 time steps is reached. One time step stands for the time needed to transmit one control packet. At this time the last edge that is necessary to make the network connected has been established. Then the major part of the load is allotted to this edge. When more edges are established, the load is distributed over more paths, so load and congestion decrease. Finally the curves balance out and the build-up process converges after nearly 4000 simulation steps. The diagram also shows that the difference between idealized and realistic congestion is small.

The lower diagram in figure 4 compares the realistic congestion of Yao, SparsY and SymmY during the network construction. It shows that the Yao topology can be built up quickly. Constructing SparsY and SymmY takes longer, because the nodes have to apply for an edge and so additional messages have to be exchanged. The diagram also shows that the Yao topology provides smaller congestion than SparsY and/or SymmY. There are two reasons: First, the load of the Yao topology is usually lower than that of SparsY or SymmY. Second, SparsY and SymmY do not prevent interferences in our simulation model due to the angular characteristic of the receiver (in contrast to the idealized sector model)!



For a given number of nodes, the value represents the average over 15 vertex sets. The vertical bars depict the standard error.

Figure 5. The relation of load, congestion and the number of nodes.



For a given number of nodes, the value represents the average over 15 vertex sets. The vertical bars depict the standard error.

Figure 6. Dilation and flow energy.

Figure 5 also points out this behavior. It shows congestion and realistic congestion for different numbers of nodes. The values are averaged over 15 vertex sets and were taken after the network had been constructed. Note that the area has always the same size so that a growing number of nodes imply a growing density. If we compare the two diagrams, we can see that the congestion of the Yao-graph is similar to the load of SparsY and SymmY.

Figure 6 shows dilation and flow energy for Yao, SparsY and SymmY, based on a hop-optimal path system. Flow energy is measured in standard energy which is defined as the energy needed to transmit 1 bit relative to the energy consumption of a transmission at maximum power, divided by the number of sectors. It turns out that SparsY and SymmY have similar dilation and flow energy values, because for randomly distributed vertex sets there are no significant differences between SparsY and SymmY. The edges in the Yao-graph that are not allowed in SparsY or SymmY are usually longer edges. So in the Yao-graph the distances can be spanned by a path over fewer hops than in SparsY or SymmY. Thus dilation for the Yao-graph is smaller than for

SparsY and SymmY. Though, paths that contain long edges are not energy efficient, so SparsY and SymmY provide better values for flow energy in our simulation.

6 Summary and Future Work

In this paper, we have shown how the theoretically well-studied Yao-, SparsY- and SymmY-graph can be used as congestion- and/or energy-efficient topologies for wireless networks. The communication devices we have studied use sector subdivision to transmit signals in several directions simultaneously and variable transmission powers. We have proposed distributed algorithms to maintain the in- and outgoing communication links of such topologies. To close the gap between abstract communication models used in the theoretical studies and realistic signal propagation and reception, we have extended our ad hoc network simulator SAHNE with well-known models for signal propagation and reception. The results of our simulational studies show that the Yao-graph can be constructed faster and yields smaller congestion values than the SparsY- and SymmY-graph. However, the SparsY-graph is more energy-efficient than the Yao-graph since it uses shorter edges.

We are currently constructing a test-bench consisting of several mini robots equipped with a self-developed communication device that provides the transmission feature we assumed in this paper. Our future research will investigate the performance of the presented topologies in this testbed. Additionally, we are extending SAHNE with ad hoc routing protocols such as Dynamic Source Routing (DSR) [6] to study their applicability to such network topologies. Finally, research on the mobility of the nodes and its impact on the topology maintenance and routing has to be addressed.

References

[1] S. Benedetto and E. Biglieri. *Principles of Digital Transmission*. Kluwer Academic / Plenum Publishers, 1999.

[2] J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Geometric spanner for routing in mobile networks. In *ACM Symposium on Mobile Ad Hoc Networking and Computing*, 2001.

[3] M. Grünewald, T. Lukovszki, C. Schindelhauer, and K. Volbert. Distributed Maintenance of Resource Efficient Wireless Network Topologies (Ext. Abstract). In *8th European Conference on Parallel Computing (EURO-PAR'02)*, pages 935–946, 2002.

[4] M. Grünewald, U. Rückert, C. Schindelhauer, and K. Volbert. Directed power-variable infrared communications for the mini robot Khepera. In *Proceedings of the 2nd International Symposium on Autonomous Minirobots for Research and Edutainment*, Brisbane, Australia, February 18-21 2003.

[5] Y. Hassin and D. Peleg. Sparse communication networks and efficient routing in the plane (extended abstract). In *Proc. of the 19th ACM symposium on Principles of distributed computing*, pages 41–50. ACM Press, 2000.

[6] IETF MANET Working Group. *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)*, 2002. Internet draft.

[7] J. M. Kahn and J. R. Barry. Wireless Infrared Communications. *Proceedings of the IEEE*, 85:265–298, February 1997.

[8] P. H. Lehne and M. Pettersen. An Overview of Smart Antenna Technology for Mobile Communications Systems. *IEEE Communications Surveys*, 2(4):2–13, 1999.

[9] F. T. Leighton. *Introduction to Parallel Algorithms and Architectures Arrays, Trees, Hypercubes*. Morgan Kaufmann Publishers, Inc., San Mateo, California, 1992.

[10] L. Li, J. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer. Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks. In *ACM Symposium on Principle of Distributed Computing (PODC'01)*, 2001.

[11] X.-Y. Li, P.-J. Wan, and Y. Wang. Power Efficient and Sparse Spanner for Wireless Ad Hoc Networks. In *IEEE International Conference on Computer Communications and Networks (ICCCN01)*, 2001.

[12] F. Meyer auf der Heide, C. Schindelhauer, K. Volbert, and M. Grünewald. Energy, Congestion and Dilation in Radio Networks. In *14th ACM Symposium on Parallel Algorithms and Architectures (SPAA'02)*, pages 230–237, 2002.

[13] H. Meyr, M. Moeneclaey, and S. A. Fechtel. *Digital Communication Receivers – Synchronization, Channel Estimation And Signal Processing*. John Wiley & Sons, Inc., 1998.

[14] R. Ramanathan and R. Hain. Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment. In *INFOCOM (2)*, pages 404–413, 2000.

[15] V. Rodoplu and T. Meng. Minimum energy mobile wireless networks. In *Proc. of IEEE International Conference on Communications (ICC'98)*, pages 3:1633–1639, 1998.

[16] A. S. Tanenbaum. *Computer networks (3rd ed.)*. Prentice-Hall, Inc., 1996.

[17] K. Volbert. A Simulation Environment for Ad Hoc Networks Using Sector Subdivision. In *10th Euromicro Workshop on Parallel, Distributed and Network-based Processing (PDP'02)*, pages 419–426, 2002.

[18] Y. Wang and X.-Y. Li. Distributed spanner with bounded degree for wireless ad hoc networks. In *Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, 2002.

[19] Y. Wang, X.-Y. Li, P.-J. Wan, and O. Frieder. Sparse power efficient topology for wireless networks. In *Journal of Parallel and Distributed Computing*, 2002.

[20] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang. Distributed Topology Control for Wireless Multihop Ad-hoc Networks. In *INFOCOM*, pages 1388–1397, 2001.