Journal of Universal Computer Science, vol. 12, no. 9 (2006), 1292-1311
submitted: 31/12/05, accepted: 12/5/06, appeared: 28/9/06 © *J.UCS*

Resource Efficient Maintenance of Wireless Network Topologies[1]

Tamás Lukovszki
Faculty of Informatics, Eötvös Lóránd University, Budapest, Hungary
lukovszki@inf.elte.hu

Christian Schindelhauer
Institute of Informatics, Albert-Ludwig University, Freiburg, Germany
schindel@informatik.uni-freiburg.de

Klaus Volbert
Heinz Nixdorf Institute, Paderborn University, Germany
kvolbert@uni-paderborn.de

Abstract: Multiple hop routing in mobile ad hoc networks can minimize energy consumption and increase data throughput. Yet, the problem of radio interferences remain. However if the routes are restricted to a basic network based on local neighborhoods, these interferences can be reduced such that standard routing algorithms can be applied.

We compare different network topologies for these basic networks, i.e. the **Yao-graph** (aka. Θ-graph) and some also known related models, which will be called the **SymmY-graph** (aka. YS-graph), the **SparsY-graph** (aka. YY-graph) and the **BoundY-graph**. Further, we present a promising network topology called the **HL-graph** (based on **H**ierarchical **L**ayers).

We compare these topologies regarding degree, spanner-properties, and communication features. We investigate how these network topologies bound the number of (uni- and bidirectional) interferences and whether these basic networks provide energy-optimal or congestion-minimal routing. Then, we compare the ability of these topologies to handle dynamic changes of the network when radio stations appear and disappear. For this we measure the number of involved radio stations and present distributed algorithms for repairing the network structure.

Key Words: ad hoc networks, topology control, distributed algorithms
Category: F.2, G.2.3, I.3.5

1 Motivation

Our research aims at the implementation of a mobile ad hoc network based on distributed robust communication protocols. Besides the traditional use of

[1] Research was done as the authors were with the Heinz Nixdorf Institute, Paderborn University. 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). A preliminary version appeared in [Lukovszki et al. 2002].

omni-directional transmitters, we want to investigate the effect of space multiplexing techniques and variable transmission powers on the efficiency and capacity of ad hoc networks. Therefore our radios can send and receive radio signals independently in k sectors of angle θ using one frequency. Furthermore, our radio stations can regulate its transmission power for each transmitted signal. To show that this approach is also suitable in practical situations, we are currently developing a communication module for the mini robot Khepera [Mondada et al. 1999, K-Team S.A. 2000] that can transmit and receive in eight sectors using infrared light with variable transmission distances up to one meter A colony of Khepera robots will be equipped with this modules to establish ad hoc networks and to evaluate our research results under realistic conditions.

We assume that most of the time the network is stable and performs a point-to-point communication protocol according to an adequately chosen routing protocol. In [Meyer auf der Heide et al. 2004] it is shown that the quality of the routing depends on the choice of the underlying network that we call *basic network*. In this paper we investigate how such networks can be maintained when stations enter and leave the network.

Little is known about the efficient design of topology-preserving dynamic algorithms. Many approaches consider a model where a central algorithm controls the network structure, using the exact coordinates in $\mathbb{R}^2$ of the radio stations (e.g., [Chen et al. 2001, Xu et al. 2001]). In contrast to this model we want to investigate a distributed network model where the only information available is given by incoming radio signals and which sector it is received, which gives a rough estimation of the direction to the sender.

The dynamics we are investigating is that a single radio station enters or leaves the system, while the rest of the system is stable. We claim that a node entering a network knows this situation, e.g. because it is switched on or it eavesdrops on existing communication from the network. A node leaving the system is equivalent to a complete node failure. This means that it is not necessary that the leaving node informs the network. Such dynamic changes are the most frequent changes of a radio network besides the motion of radio stations.

In our view its very unlikely that all mobile radio station would start (or leave) at the same time. And even if this is enforced one can easily add a probabilistic strategy that prevents this situation. Then the establishment of the complete network turns out to be a series of single stations entering an existing network. This approach makes sense, since nobody expects that a radio connection to the network is instantly established and we will see that there exist network structures where entering and leaving will only need some logarithmic communication rounds.

In this paper, we do not address the problem of moving radio stations. However, if the movement is not too fast, the moving node can reestablish the correct network by triggering a `leave` and an `enter`-operation. Furthermore, we hope that the basic routines developed for this switching dynamics provide basic techniques for more sophisticated maintenance techniques of mobile ad hoc networks.

2 Model

Our investigations concentrate on the implementation of distributed algorithms for mobile ad hoc networks with radio stations with specific hardware features. However, some network topologies (like the HL-graph) can be used in a much more general hardware model.

2.1 Communication Model

In this paper we assume that if a station enters the system it will send out control messages to stop normal packet routing for the (hopefully short) time needed to update the network structure. All packets are stored on the radio stations and delivered when the network structure has been restored. In contrast to this reactive approach, one can also take advantage of synchronized clocks if available. If a periodically time period is reserved that is known to all nodes (including new ones), the maintenance of the network can be done in this special maintenance period. Thus, no control packets are necessary to stop the packet routing mode and collisions caused by the control packets can be prevented.

In our communication model, we assume that a radio station w, also called node, is able to detect three types of incoming signals: *No signal* indicates that no radio signal is transmitted at all or that all radio stations r in distance d send with transmission distance $d' < d$. The *interference signal* indicates that at least two radio stations u and v send in this time step t with transmission distance $d(u,t) > ||u,w||_2$ and $d(v,t) > ||v,w||_2$, where $||u,w||_2$ denotes the Euclidean distance. A *clear signal* is received by w if one radio signal with appropriated transmission power to cancel out weaker incoming signals is reaching u's antenna. Then it can read the transmitted information $m \in \{0,1\}^p$ of some length p. A communication round is the time necessary to send one packet of length p, where p is large enough to carry some elementary information like the sending station, the addressed stations (if specified), the transmission distance, and some control information.

We assume that there is a timing schedule adapted to the basic network topology that allows the stations in a static time period, i.e. no nodes enter or leave, to transmit and acknowledge packets over the network routes with only small number of interfering packets. During such a phase we can neglect the interfering impact of acknowledgment signals.

However when a connection is established the sending and answering signal have the same small length, because only control information needs to be transmitted. Then the impact of answering signals is the same as those of sending signals. Therefore, we consider two types of interferences: The *uni-directional interferences* in the routing mode and the *bi-directional interferences* when connections are established or network changes are compensated.

2.2 Hardware Model

Every node can choose the transmitting power according to s discrete choices $p_1, \ldots, p_s$. The energy to send over distance d is given by $\text{pow}(d) := d^c$ for some constant $c \geq 2$ (constant factors are omitted for simplicity). This defines the transmitting distances $d_i = (p_i)^{1/c}$ for all $i \in [s]$, where $[s] := \{1, \ldots, s\}$.

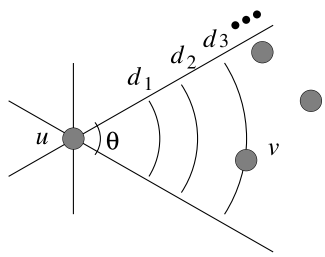

Figure 1: The sector of u containing v and transmitting distances $d_1, d_2, d_3, ...$

Every node u has k sending and receiving devices, which are located such that they can communicate in parallel within each of k disjoint sectors with angle $\theta = \frac{2\pi}{k}$. Every node u has been rotated by a angle α_u, which is unknown to u. Note that the radio stations have different offset angles α_u. If u sends a signal in the ith sector it actually sends into a direction described by the interval $R = [\alpha_u + i\theta, \alpha_u + (i+1)\theta)$ and can be received by node v in sector j if $R \cap [\alpha_v + j\theta, \alpha_v + (j+1)\theta) \neq \emptyset$. Of course v receives u only if in addition u sends this signal with transmission distance $d_i \geq ||u, v||_2$ (see Figure 1).

Furthermore, we allow that radio stations can measure distances only by sending messages with varying transmission power. Then the receiving party can only decide whether the signal arrives or not. This restricts transmission distances to the set $S = \{d_1, \ldots, d_s\}$. Define $D : \mathbb{R} \rightarrow \{\varnothing, 1, \ldots, s\}$ as the minimum discrete choice of transmission power to send over a given distance by

$$D(x) := \begin{cases} \min\{i \mid d_i \geq x\} & \text{if } x \leq d_s \ , \\ \varnothing & \text{if } x > d_s \ . \end{cases}$$

Define $\triangleleft(u, v)$ as the number of u's sector containing the edge (u, v) (note that $k\theta = 2\pi$):

$$\triangleleft(u, v) := \left\lfloor \frac{[\measuredangle(v - u) - \alpha_u]}{\theta} \right\rfloor \bmod k \ ,$$

where $\measuredangle(x)$ denotes the angle of a vector x in $\mathbb{R}^2$.

2.3 Location of Nodes

One of the most delimiting properties is that radio stations do not know their locations. The following restriction prevent the vertex set from taking abnormal positions.

Definition 1. Let V be a set of n vertices in $\mathbb{R}^2$. V is in **general position**, if there are no vertices $u, v, w \in V$ with $v \neq w$ and $||u, v||_2 = ||u, w||_2$.
We call a vertex set **normal**, if for a fixed polynomial $p(n)$ we have

$$\frac{\max_{u,v \in V} ||u, v||_2}{\min_{u,v \in V} ||u, v||_2} \leq p(n) \ .$$

Because of the discrete model for the transmitting distances we cannot distinguish distances within some interval $(d_i, d_{i+1}]$. However, we assume that these distances form a fine granular scale. Furthermore, we want to neglect problems occurring when the maximum transmission distance is shorter than distances between nodes. Therefore throughout this paper, we restrict vertex sets to be *nice*:

Definition 2. We call the locations of radio stations **nice**, if for all $u, v, w \in V$ we have $D(u, v) \neq \varnothing$ and

$$v = w \qquad \Longleftrightarrow \qquad \triangleleft(u, v) = \triangleleft(u, w) \wedge D(u, v) = D(u, w) \ .$$

Throughout this paper we consider the vertex set to be nice and normal.

3 Basic Network Topologies

The underlying hardware model allows to communicate in k disjoint sectors in parallel. Therefore a straight-forward approach is to choose as a communication partner the nearest neighbor in a sector. This leads to the definition of Yao-graphs. Beside the class of Yao-graphs we study a hierarchical topology construction, called the Hierarchical Layer Graph, which is based on a hierarchical clustering method.

3.1 Yao-Graphs and Variants

Definition 3. [Yao 1982] For a given set V of n vertices in $\mathbb{R}^2$, the **Yao-graph** (aka. Θ-graph) is defined by the following set of directed edges:

$$E := \{(u,v) \mid \forall w \neq u \ : \ \sphericalangle(u,v) = \sphericalangle(u,w) \quad \Rightarrow \quad D(u,v) \leq D(u,w)\} \ .$$

Recall that throughout this paper we assume vertex sets to be nicely located, hence every node has at most one neighbor in a sector. The out-degree is therefore bounded by k. However, a node can be the nearest node of many nodes. To overcome this problem of high in-degree resulting in time-consuming interference resolution schedules, we present three Yao-graph based topologies.

The **symmetric Yao-graph**, called (SymmY-graph) is a straight-forward solution of the high in-degree problem. An edge (u,v) is only introduced if u is the nearest neighbor of v and vice versa.

Definition 4. [Wang and Li 2002] Let G_θ be the Yao-graph of a vertex set V. Then, the edge set E of the **Symmetric Yao-graph** (**SymmY-graph**) of V is defined by

$$E := \{(u,v) \in E(G_\theta) \mid (v,u) \in E(G_\theta)\} \ .$$

Although such a graph reduces interferences to a minimum (because in every sector only at most one neighbor appears) very long detours may appear, which make such a graph incapable of bearing short routes and allowing routing without bottlenecks.

Following the approach of [Wang and Li 2002] we consider also a graph topology which allows at most two neighbors in a sector and call this graph **sparsified Yao-graph**, which is a Yao-graph where, when the in-degree of a sector exceeds one, only the incoming shortest edge will be chosen.

Definition 5. [Wang and Li 2002] For a given set V of n vertices in $\mathbb{R}^2$, the edge set of the **Sparsified Yao graph** (**SparsY-graph**) is defined by

$$E := \{(u,v) \in E(G_\theta) \mid \forall w \in V : \ ((w,v) \in E(G_\theta) \text{ and } \sphericalangle(v,w) = \sphericalangle(v,u)) \\ \implies \ ||w,v||_2 > ||u,v||_2\} \ ,$$

where G_θ denotes the Yao-graph of V.

It is an open problem whether all SparsY-graphs are c-spanners, i.e. the shortest path between vertices in the network is at most c-times longer than the Euclidean distance.

To construct a c-spanner with constant degree Arya et al. [Arya et al. 1995] introduced the following transformation. Like in [Lukovszki 1999] we apply this technique to the Yao-graph and call the resulting graph a **Bounded Degree Yao-graph** (**BoundY graph**).

For this, let $G = (V, E)$ be a c'-spanner with bounded out-degree. Let $N(v) = \{w \in V : wv \in E\}$ the set of in-neighbors of $v \in V$. For each $v \in V$, the star defined by the edges $\{wv \in N(v)\}$ will be replaced by a so-called v-single sink c''-spanner, $c'' = c/c'$, $T(v)$, which has a bounded in- and out-degree, i.e. $G^* = (V, E^*)$, where $E^* = \bigcup_{v \in V, uw \in E(T(v))} uw$.

A graph with a vertex set U is called a *v-single sink c''-spanner (a (v, c'')-SSS)*, if from each vertex $w \in U$ there is a c''-spanner path to the vertex v. Such a (v, c'')-SSS for U can be constructed as follows.

Let $\alpha = 2 \arcsin \frac{c''-1}{2c''}$. We divide the plane around v into sectors of an angular diameter at most α. For each sector C, let U_C be the set of all vertices of $U \setminus \{v\}$ contained in C. If a subset U_C contains more than $|U|/2$ vertices, then we partition it arbitrarily into two subsets $U_{C,1}$ $U_{C,2}$, each of size at most $|U|/2$ For each subset U_C, let $w_c \in U_C$ be the vertex which is closest to v. We add the edge $w_C v$ and then we recursively construct a (w_C, c'')-SSS for each subset U_C. This recursion ends after $\log |U|$ steps, since we halve (at least) the number of vertices at each level of the recursion. In this way we obtain a directed tree $T(v)$ with root v which is a (v, c'')-SSS for $N(v) \cup \{v\}$. Since each vertex v had a bounded out-degree in G, and therefore it can be contained in a constant number of in-neighborhoods $N(u)$, $u \in V$, its degree in G^* will be also bounded. This completes the construction of the BoundY graph.

The above recursive construction is allows the distributed construction of the BoundY graph given the Yao-graph. Furthermore, for compass routing it provides suitable rerouting information: If a message wants to use an edge uv in the Yao-Graph, then it will use the tree-path from u to v in $T(v) \subset G^*$, which has at most $O(\log n)$ hops.

3.2 The Hierarchical Layer Graph

Adopting ideas from clustering [Gao et al. 2001a, Gao et al. 2001b] and generalizing an approach of [Adler and Scheideler 1998] we present a graph consisting of w layers $L_0, L_1, \ldots, L_w$. The union of all this graphs gives the **Hierarchical Layer graph** (**HL-graph**). Let V be a set of n vertices in $\mathbb{R}^2$. The lowest layer L_0 contains all vertices V. For $1 \leq i \leq w$, the vertex set of a higher layer L_i is a subset of the vertex set of a lower layer L_{i-1} until in the highest layer there is only one vertex, i.e.

$$V = V(L_0) \supseteq V(L_1) \supseteq \cdots \supseteq V(L_w) = \{v_0\} .$$

The crucial property of these layers is that in each layer L_i vertices obey a minimum distance:

$$\forall u, v \in V(L_i) \ : \ ||u, v||_2 \ \geq \ r_i \ .$$

Furthermore, all nodes in the next-lower layer must be covered by this distance:

$$\forall u \in V(L_i) \;\; \exists v \in V(L_{i+1}) \; : \; ||u,v||_2 \; \leq \; r_{i+1} \; .$$

Our construction uses parameters $\alpha \geq \beta > 1$, where for some $r_0 < \min_{u,v \in V} ||u,v||_2$ we use radii

$$r_i := \beta^i \cdot r_0$$

and we define in layer L_i the edge set $E(L_i)$ by

$$E(L_i) := \{(u,v) \mid u,v \in V(L_i) \; \wedge \; ||u,v||_2 \; \leq \; \alpha \cdot r_i\} \; .$$

Clearly, for a normal vertex set we have a maximum number of $w = O(\log n)$ layers. For HL-graphs we need not assume nice or normal locations, as long as our hardware models supports the following transmission distances:

1. If there are layers $L_0, \ldots, L_w$, then $\{r_i \mid i \in \{0,\ldots,w\}\} \subseteq S$ and $\{\alpha r_i \mid i \in \{0,\ldots,w\} \subseteq S$, where $S = \{d_1,\ldots,d_s\}$,
2. $d_0 \leq \min_{u,v \in V} ||u,v||_2$,
3. $d_w \geq \max_{u,v \in V} ||u,v||_2$.

4 Elementary Graph Properties

We can show the following inclusions. Note that $A \not\# B$ denotes $A \not\subseteq B$ and $B \not\subseteq A$.

Lemma 6. *Let V be a nice vertex set. Then,* SymmY(V) $\subseteq$ SparsY(V) $\subseteq$ BoundY(V) *and* SparsY(V) $\subseteq$ Yao(V). *For some V it holds that* BoundY(V) $\not\#$ Yao(V).

The proof follows directly by the definitions.

4.1 Degree

Lemma 7. *For normal and nice vertex sets V consisting of n nodes we observe the following maximum in- and out-degrees:*

Topology	*Yao*	*SymmY*	*SparsY*	*BoundY*	*HL*
in-degree	$n-1$	k	k	$(k+1)^2$	$O(\log n)$
out-degree	k	k	k	k	$O(\log n)$
degree	$n-1$	k	$2k$	$k+(k+1)^2$	$O(\log n)$

4.2 Spanners, weak spanners and power spanners

In section 5 we will see that spanner-properties have implication for the energy optimality of the network as well as the weak spanner property for the congestion minimization.

Definition 8. A graph $G = (V, E)$ is a c-**spanner**, if for all $u, v \in V$ there exists a (directed) path p from u to v with $||p||_2 \leq c \cdot ||u, v||_2$.

G is a **weak** c-**spanner**, if for all $u, v \in V$ there exists a path p from u to v which is covered by a disk of radius $c \cdot ||u, v||_2$ centered at u.

G is a (c, d)-**power spanner**, if for all $u, v \in V$ there is a path $p = (u = u_1, u_2, \ldots, u_m = v)$ from u to v in G such that

$$\sum_{i=1}^{m-1} (||u_i, u_{i+1}||_2)^d \leq c \min_{(u=v_1, v_2, \ldots, v_w=v)} \sum_{i=1}^{m-1} (||v_i, v_{i+1}||_2)^d .$$

If for all $d > 1$ there exists a constant c such that G is a (c, d)-power spanner we call G a **power spanner**.

On the positive side the following results are known.

Lemma 9.

1. *Let $V \subset \mathbb{R}^2$. For $k > 6$ the Yao-graph is a c-spanner with $c = 1/(1 - 2\sin\frac{\theta}{2})$ [Ruppert and Seidel 1991].*
2. *For $k \geq 6$ and $c = \max\left(\sqrt{1 + 48\sin^4(\theta/2)}, \sqrt{5 - \cos\theta}\right)$ the Yao-graph is a weak c-spanner [Fischer et al. 1997].*
3. *For $k = 4$, the Yao-graph is a weak c-spanner with $c = \sqrt{3 + \sqrt{5}}$ [Fischer et al. 1998].*
4. *For $k > 6$ the BoundY-graph is a c-spanner for a constant c [Arya et al. 1995].*
5. *For $k > 6$ the SparsY-graph is a power spanner [Wang and Li 2002].*

It is an open problem whether SparsY-graphs are c-spanners. Here, we show that they are also weak spanners (and the proof of this theorem can be used give a proof of the power spanner property without assuming that the angle k is depending on V as done in [Wang and Li 2002]).

Lemma 10. *For $k > 6$ the SparsY-graph is a weak c-spanner where $c = \frac{1}{1-2\sin\frac{\theta}{2}}$.*

Proof. Let $G = (V, E)$ be the SparsY-graph and $G_Y = (V, E_Y)$ be the underlying Yao-graph. Starting from two vertices u, v we will show how to find a directed path from u to v in the SparsY-graph that is inside a disk with center at u of radius $||u,v||_2/(1-2\sin\frac{\theta}{2})$. For a sector i, define the Yao-neighbor v of a vertex u as the (unique) vertex v with $(u,v) \in E_Y$. Then we know:

- If a node u has no directed edge in a sector i, then either the sector is empty (i.e. no edge in the Yao-graph), or there is a Yao-neighbor v (i.e., $(u,v) \in E_Y$) incident to an edge $(w,v) \in E$, where w is in another sector of u. Furthermore, $||u,w||_2 < ||u,v||_2$, because $\theta < \pi/3$ and $||v,w||_2 < ||u,v||_2$.
- Every node u has at least one neighbor v, i.e. $\exists v \in V \ : \ (u,v) \in E$.

Now, we recursively construct the path $P(u,v)$ using some of the Yao-neighbors of u (see Figure 2). If $(u,v) \in E$ then $P(u,v) = ((u,v))$, if $u = v$ then $P(u,v) = ()$. If in sector $i = \sphericalangle(u,v)$ the Yao-neighbor, called q_0, is not directly connected to u. Then, we know that there exists an edge $(p_0,q_0) \in E$, where p_0 is in a sector $i_1 \neq i_0$ of u and $||p_0,u||_2 < ||q_0,u||_2$. Furthermore we have that $||q_0,u||_2 \leq ||u,v||_2$. Then, we repeat this consideration for the sector i_1 and replace v by p_1. This iteration ends when a Yao-neighbor q_m or p_m is directly connected to u, i.e. $(u,q_m) \in E$ or $(u,p_m) \in E$. Because every node has at least one neighbor in E this process terminates.

Now we recursively define the path $P(u,v)$ from u to v that terminates at node q_m (for p_m the path can be defined analogous: replace (u,q_m) by $(u,p_m) \circ (p_m,q_m)$) by

$$P(u,v) = (u,q_m) \circ P(q_m,p_{m-1}) \circ (p_{m-1},q_{m-1}) \circ \ldots \circ P(q_1,p_0) \circ (p_0,q_0) \circ P(q_0,v).$$

Note that all nodes p_i, q_i are inside the disk with center u and radius $||u,v||_2$. Furthermore, we have $||q_i,p_{i-1}||_2 < ||u,v||_2$. In the next recursion vertices of the path may lie outside of this disk. However it is straight-forward that the maximum disk amplification of a recursion step will be achieved, if q_l and p_{l-1} for some $l \in \{0,\ldots,m\}$ are placed as given in Figure 3 ($v := p_{l-1}, x := q_l$). There we have

$$||x,v||_2 = ||q_l,p_{l-1}||_2 < 2\sin\frac{\theta}{2}\ ||u,v||_2\ .$$

That means, that the maximum amplification of the disk with center u and radius $||u,v||_2$ can be at most $||u,v||_2 + 2\sin\frac{\theta}{2}\ ||u,v||_2$ in each recursion step. Let r be the depth of the recursion, then by $\sum_{i=0}^{r}(2\sin\frac{\theta}{2})^i\ ||u,v||_2 \leq ||u,v||_2/(1-2\sin\frac{\theta}{2})$ it follows, that $P(u,v)$ is inside the disk with center u of radius $||u,v||_2/(1-2\sin\frac{\theta}{2})$ and so we get $c = 1/(1-2\sin\frac{\theta}{2})$. □

Lemma 11. *If $\alpha > 2\frac{\beta}{\beta-1}$ the HL-graph is a c-spanner for*

$$c = \max\left\{\beta\frac{\alpha(\beta-1)+2\beta}{\alpha(\beta-1)-2\beta}, \frac{\alpha}{\beta}\right\}.$$

Proof. Define a directed tree T on the vertex set $V_0 \times \{0,\ldots,w\}$ as follows. The leafs of T are all pairs $V_0 \times \{0\}$. If $u \in V(L_i)$, then (u,i) is a vertex of T. T consists

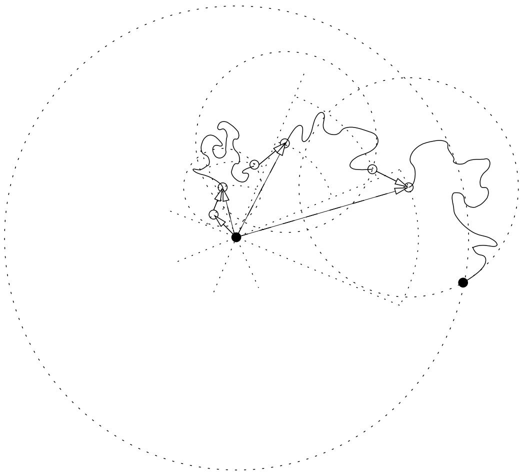

Figure 2: Proof idea for the weak spanner property of the SparsY-graph

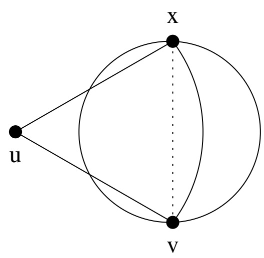

Figure 3: Maximum disk amplification

of the following edges: For $i > 0$ if $u \in V(L_i)$, then $((u, i-1), (u, i)) \in E(T)$. If $u \in V(L_i) \setminus V(L_{i+1})$ then chooses arbitrary $v \in V(L_{i+1})$ with $(u, v) \in E(L_i)$ and add $((u, i), (v, i+1))$ to the edge set of the tree T. Note that the tree has depth w and the root (v_0, w).

Now for two vertices $u, v \in V$ we define a **clamp** of height j, which is a path connecting u and v. The clamp consists of two paths $P_u^j := (u, p(u), p^2(u), \ldots, p^j(u))$ and $P_v^j := (v, p(v), p^2(v), \ldots, p^j(v))$ of length $j-1$, where $p^i(w)$ denotes the ancestor of height i of a vertex w in the tree T. These two path are connected by the edge $(p^j(u), p^j(v))$.

Claim 1 *If for vertices u, v the distance is bounded by $||u, v||_2 \leq d_j$, then a clamp of height j is contained in the HL-graph, where*

$$d_0 = \alpha r_0 \qquad \text{and} \qquad d_{j+1} := (\alpha\beta - \alpha - 2\beta) r_j + d_j \ .$$

Proof. Note that the path $(u, p(u), p^2(u), \dots, p^j(u))$ is contained in the HL-graph and that $||p^i(u), p^{i+1}||_2 \le r_{i+1}$. A clamp of height j is contained in G if $||(p^j(u), p^j(v))||_2 \le \alpha r_j$. This gives for the maximum distance of u and v:

$$||u, v||_2 \;\le\; \alpha r_j + 2\sum_{i=1}^{j} r_i \;=:\; d(j) \;.$$

Now note that $d(j+1) = d(j) + \alpha(r_{j+1} - r_j) + 2r_{j+1} = d(j) + \alpha\beta r_j - \alpha r_j + 2\beta r_j$. □

Claim 2 *A clamp C of height j has maximum length ℓ_j, where*

$$\ell_0 := \alpha r_0 \qquad \text{and} \qquad \ell_{j+1} := (\alpha\beta - \alpha + 2\beta) r_j + \ell_j \;.$$

Proof. Recall that the length of the paths P_u^j and P_v^j is bounded by $2\sum_{i=1}^{j} r_i$ and the edge $(p^j(u), p^j(v))$ has length of at most αr_j. This gives

$$||C|| \;\le\; \alpha r_j + 2\sum_{i=1}^{j} r_i \;=:\; \ell(j) \;.$$

Now $\ell(j+1) - \ell(j) = \alpha r_{j+1} - \alpha r_j + 2\beta r_j = (\alpha\beta - \alpha + 2)r_j$. □

Given two vertices u, v with distance $d = ||u, v||_2$ we determine the minimum j with $d_j \ge d$ and then get a clamp of length at most ℓ_j. Let $k := \alpha\beta - \alpha - 2\beta$ and $K := \alpha\beta - \alpha + 2\beta$.

Then we have:

$$d_j = \alpha r_0 + k\sum_{i=1}^{j} \beta^j = r_0\left(\alpha + k\frac{\beta^{j+1} - 1}{\beta - 1} - k\right) \;.$$

Hence the minimal choice of j is $j = \lceil x \log_\beta(1 + (\frac{d}{r_0} - \alpha - k)\frac{\beta - 1}{k})\rceil - 1$. We will substitute j into ℓ_j and get for $j \ge 1$:

$$\begin{aligned}
\ell_j &= \alpha r_0 + K\sum_{i=1}^{j+1} \beta^i \\
&= r_0\left(\alpha + K\frac{\beta^{j+1} - 1}{\beta - 1} + K\right) \\
&\le r_0\left(\alpha + K\frac{\beta(\frac{d}{r_0} - \alpha - k)\frac{\beta - 1}{k}}{\beta - 1} + K\right) \\
&\le r_0\left(\alpha + K\frac{\beta(\frac{d}{r_0} - \alpha)}{k}\right) \\
&\le \alpha r_0 - \frac{\beta K}{k} r_0 + d\frac{\beta K}{k} \\
&\le d\left(\frac{\beta K}{k} + \left(\alpha - \frac{\beta K}{k}\right)\beta^{-j}\right)
\end{aligned}$$

$$\leq d\left(\beta\frac{\alpha(\beta-1)+2\beta}{\alpha(\beta-1)-2\beta}+\alpha\beta^{-j}\right)$$
$$\leq d\max\left\{\beta\frac{\alpha(\beta-1)+2\beta}{\alpha(\beta-1)-2\beta},\frac{\alpha}{\beta}\right\}.$$

□

Now, we solve the open problem stated in [Wang and Li 2002], whether the SymmY-graph is a c-spanner, or a power spanner by giving a negative answer:

Lemma 12. *The SymmY-graph is* **not** *a weak c-spanner for any constant $c \in \mathbb{R}$, nor a (c,d)-power spanner for any $d > 1$.*

Proof. We show an example for n points in the plane, such that the SymmY-graph of that points is not a weak c-spanner for any c. Let ℓ_1 and ℓ_2 be two vertical lines of unit distance from each other, such that ℓ_2 is right to ℓ_1. Rotate ℓ_1 clockwise around its intersection point with the x-axis by a very small angle δ_c, and rotate ℓ_2 counterclockwise around its intersection point with the x-axis by an angle δ_c. We denote the rotated lines by ℓ'_1 and ℓ'_2. Consider the vertex sets $U = \{u_1, \ldots, u_m\}$ and $V = \{v_1, \ldots, v_m\}$, $m = n/2$, placed on ℓ'_1 and ℓ'_2, respectively, as follows. Assume that for each point $u \in U$, the half-line, halving the ith sector of u is horizontal and directed in positive x-direction, and for $v \in V$, the half-line, halving the i'th sector of v is horizontal and directed in negative x-direction. The vertex u_1 is placed on the intersection point of ℓ_1 and the x-axis. We place v_1 on ℓ'_2 such that v_1 is in the ith sector of u_1 and it is very close to the upper boundary of the ith sector of u_1. The vertex u_2 is placed on ℓ'_1 in the i'th sector of v_1 close to the upper boundary of that sector. The vertex v_2 is placed on ℓ'_2 in the ith sector of u_2 close to the upper boundary of that sector, etc... Then the SymmY-graph does not contain any edge (u,v) such that $u \in U \setminus \{u_m\}$ and $v \in V \setminus \{v_m\}$. The nearest neighbor of u_1 in sector i is v_1, while v_1 has u_1 and u_2 also in sector i', where u_2 is nearer, etc... Only the last link u_m, v_m will be established. Therefore, even if there is a path from u_1 to v_1 in the SymmY-graph, its length is at least $||u_1, u_m||_2 + ||u_m, v_m||_2 + ||v_m, v_1||_2$. For any given c we can choose δ_c appropriately small, in order to get $||u_1, u_m||_2, ||v_m, v_1||_2 \geq c/2$. This proves the claim.

Nevertheless, we can prove the following positive property.

Lemma 13. *For $k \geq 6$ and for general vertex sets the SymmY-graph is connected.*

Proof. We consider a directed link (u,v) in the Yao-graph and show, that there is a path from u to v in the SymmY-graph. We prove the claim by induction over the length of all links. First of all, we consider the shortest directed edge (u,v)

in the Yao-graph. Then for $k > 6$ (v, u) is a directed edge in the θ-Yao-graph. Otherwise it exist an shorter edge (v, w). It follows, that (u, v) is a link in the SymmY-graph. Now, we consider any edge (u, v) and assume that the claim is true for all (r, s) with $||r, s||_2 < ||u, v||_2$. Case 1: (v, u) is a link in the Yao-graph. Then this link exists also in the SymmY-graph. Case 2: (v, u) is not a link in the Yao-graph. Then a node w exists with $\sphericalangle(v, u) = \sphericalangle(v, w)$, $||w, v||_2 < ||u, v||_2$ and $||u, w||_2 < ||u, v||_2$. By induction there exists a path from u to w and from w to v in the θ-SymmY-graph. Therefore a path from u to v exists. □

5 Network Properties

In [Meyer auf der Heide et al. 2004] we investigate the basic network parameters **interference number**, **energy**, and **congestion**. In this paper we extend the definition of interference number to directed communication. The reason is that we allow two communication modes. In the packet routing mode acknowledgment signals are very short and we can neglect its impact on the interferences. When control messages have to be exchanged sending and answering signals are both short, then we have to consider all combination of interferences. Therefore we distinguish the following of interferences.

Definition 14. The edge (r, s) has a **uni-directional interference** caused by (u, v), denoted by $(r, s) \in \mathrm{UInt}(u, v)$, if

$$\sphericalangle(s, r) = \sphericalangle(s, u) \text{ and } \sphericalangle(u, s) = \sphericalangle(u, v) \text{ and } D(u, v) \geq D(r, s) \ .$$

The edge (r, s) **bi-directionally interferes** with (u, v), denoted by $(r, s) \in \mathrm{BInt}(u, v)$, if

$$(r, s) \in \mathrm{UInt}(u, v) \text{ or } (s, r) \in \mathrm{UInt}(u, v) \text{ or } (r, s) \in \mathrm{UInt}(v, u) \text{ or } (s, r) \in \mathrm{UInt}(v, u).$$

The (bi-directional) **interference number** of a basic network G is defined by $\max_{e \in E}\{1 + |\mathrm{BInt}(e)|\}$, where $\mathrm{BInt}(e)$ denotes the set of edges that interfere with e if packets are simultaneously transmitted.

Analogously, we define the **uni-directional interference number** of a graph, by replacing $\mathrm{BInt}(e)$ by $\mathrm{UInt}(e)$.

Note that both types of interferences are asymmetric, i.e. $u \in \mathrm{BInt}(v) \not\Leftrightarrow v \in \mathrm{BInt}(u)$ and analogously for UInt. This stems from the fact that we use adjustable transmission distances.

A routing protocol can be described by a set of paths $\mathcal{P}$, called path system, that optimizes network parameters. We assume that the path system is chosen according to a demand $w : V \times V \to \mathbb{N}$ representing the point-to-point communication traffic within the network. Since the locations of the vertex sets are nice for every combination of vertices there is at least a path p from u to v in the path system if $w(u, v) > 0$.

Definition 15. The load $\ell(e)$ of an edge e is the number of packets that are using this edge. The interfering load of an edge is $\ell(e) + \sum_{e' \in \mathrm{UInt}(e)} \ell(e')$. The edge with the maximum interfering load defines the **congestion of a path system**. The **energy of a path system** $\mathcal{P}$ is given by $\sum_{P \in \mathcal{P}} \sum_{e \in P} \mathrm{pow}(e)$, where $\mathrm{pow}(e) = (||e||_2)^2$. This is $\sum_e \ell(e)(||e||_2)^2$.

It turns out that energy and congestion are connected to power spanners and weak spanners. The link between these geometric properties and the networking features is described by the following theorem:

Theorem 16.
(i) If the basic network is a (c, d) power-spanner, then it allows a path system that approximates the optimal energy path system by a constant factor of c.
(ii) Every c spanner is (c^d, d) power-spanner.
(iii) If for a normal vertex set the basic network is a weak c-spanner G with uni-directional interference number q then there is a path system in G that approximates the optimal path system minimizing the congestion by a factor of $O(q \log n)$.

Proof. (i) follows from the definition of the (c, d) power spanner.
(ii) Let $G = (V, E)$ be a c-spanner, $u, v \in V$, and $P = uu_1u_2...u_rv$ be an energy optimal path from u to v in G. Let $u_0 = u$ and $u_{r+1} = v$. We show that for each edge $u_iu_{i+1} \in P$, $0 \leq i \leq r$, there is a path $P_i = u_iw_1w_2...w_{r_i}u_{i+1}$ in G, $w_0 = u_i$, $w_{r_i+1} = u_{i+1}$, for which

$$\sum_{j=0}^{r_i} \mathrm{pow}(w_j, w_{j+1}) \leq c^d \mathrm{pow}(u_i, u_{i+1}). \tag{1}$$

Substituting each edge $u_iu_{i+1} \in P$ by P_i, after summation of equation (1) for each edge of P we obtain the claim. Equation (1) follows from the fact that G is a c-spanner, and therefore, for each edge $u_iu_{i+1} \in P$ there is a path $P_i = u_iw_1w_2...w_{r_i}u_{i+1}$ with $\sum_{j=0}^{r_i} ||w_jw_{j+1}|| \leq c||u_iu_{i+1}||$. Thus, $\sum_{j=0}^{r_i} \mathrm{pow}(w_j, w_{j+1}) = \sum_{j=0}^{r_i} ||w_jw_{j+1}||^d \leq (\sum_{j=0}^{r_i} ||w_jw_{j+1}||)^d \leq (c||u_iu_{i+1}||)^d = c^d \mathrm{pow}(u_i, u_{i+1})$.
(iii) We use the following Lemma:

Lemma 17. *[Meyer auf der Heide et al. 2004] Let C^* be the congestion of the congestion-optimal path system $\mathcal{P}^*$ for a normal vertex set V. Then, every weak c-spanner N can host a path system $\mathcal{P}'$ such that the induced load $\ell(e)$ in N is bounded by $\ell(e) \leq c' \log n\ C^*$ for a positive constant c'.*

Since the interference number of the network is bounded by q this implies $C_{\mathcal{P}}(V) = O(q\ (\log n)\ C^*)$. □

Combining this Lemma with the basic graph properties investigated in Section 3 we obtain:

Theorem 18. *For a nicely located vertex set V the following table describes the worst case behavior, whether it hosts optimal path systems approximating energy or congestion:*

Topology	*uni-directional interference number*	*Spanner*	*Energy approx. factor*	*Congestion approx. factor*
Yao-graph	$n-1$	*yes*	$O(1)$	—
SymmY-graph	1 *(bi-direct.!)*	*no, but connected*	—	—
SparsY-graph	1	*weak and power spanner*	$O(1)$	$O(\log n)$
BoundY-graph	$\Theta(n)$	*yes*	$O(1)$	—
HL-graph	$O(\log n)$	*yes*	$O(1)$	$O(\log^2 n)$

6 Maintaining the Network

The standard mode of an ad hoc network is the packet routing mode. In the lucky case of SymmY-graphs there are no interferences between messages and acknowledgments of different edges. For the SparsY-graph packets sent along the direction of the edges cannot interfere with other packets on different edges. However, acknowledgment signals of such edges can interfere. Since in the normal transportation mode data packets are long compared to the short acknowledgments, we neglect this interaction.

In all other graphs we have to resolve (uni-directional) interference. There are two strategies:

- Non-interfering deterministic schedule.

  In general it is an $\mathcal{NP}$-hard problem to compute a schedule that resolves all interferences within optimal time.

  However, in the HL-graph in each layer the bi-directional interference number is a constant. Hence, it is easy to define a deterministic schedule that ensures each edge a time frame of $\frac{1}{c \log n}$, which in the worst case slows down communication only by this logarithmic factor.

  For the Yao-graph the (uni-) directional interferences are given by the in-degree. Hence a straight-forward strategy is to assign each of these incoming senders a time frame of same size. Unlike as for the HL-graph this schedule is far from being optimal, since it does not reflect the actual load on the edges.

  The main advantage of such a non-interfering schedule is that collisions immediately indicate that dynamic changes have occurred.

- Interfering probabilistic schedule.

Following the ideas presented in [Adler and Scheideler 1998] every edge e of the basic network is activated with some independent probability $p(e) \leq \frac{1}{2}$, where for all edges e it holds

$$p(e) + \sum_{e' \in \mathrm{UInt}(e)} p(e') \;\leq\; 1 \;.$$

Then, there is a constant probability of at least $\frac{1}{4}$ that a packet is transferred without being interfered by another packet.

The detection of dynamic network changes may need more time than in non-interfering schedules. Here, since with probability of at least $\frac{1}{4}$ every receiver does not get an input signal, it suffices to repeat the dynamic change signal for some $O(\log n)$ rounds. Then all nodes are informed with probability $1 - 1/p(n)$ (for some polynomial $p(n)$).

The only information necessary to maintain such a probabilistic schedule is the local number of uni-directional interferences, or an approximation of that number. In the case of the BoundY-graph this number is not given by a graph property as in the other topologies. Therefore, a node has to inform all m interfering nodes, that they interfere and how many of them interfere. A straight-forward approach shows that this takes time $O(m)$. However later we state a general approach that computes and transmits an appropriate approximation of that number in time $O(\log m)$.

We investigate two elementary dynamic operations necessary to maintain dynamic wireless networks:

– **Enter:** While the network is distributing some packets, one radio station wants to enter the network. It will send a special signal causing a special interference signature that will cause all radio station in some specified distance to stop the point-to-point communication mode and switch to a special enter node.

  Then, this part of the network devotes its communication to insert the new node into the network topology. After this, it will resume to the normal transportation mode.

– **Leave:** A single station stops sending and receiving. At some time a neighbored node notices this failure and signals it to other nodes of the network. These nodes halt routing packets and rebuild the network.

6.1 Topology Induced Costs

The two important resources in these update processes are **time** and **number of involved processors**. If these parameters are minimized, then the impact of the network disturbance can be kept to a minimum.

Theorem 19. *For a normal and nicely located vertex set V the $\Theta(|V|)$ edges need to be changed if an enter/leave operation happens in a Yao-, SymmY-, SparsY-, or BoundY-graph.*

For the HL-graph this number is bounded by $O(\log |V|)$.

Proof. A bad situation for all Yao based graphs (i.e. for Yao-, SymmY-, SparsY-, and the BoundY-graph) occurs, when two rows of vertices $U := \{u_1, \ldots, u_m\}$ and $V := \{v_1, \ldots, v_m\}$ are placed on two parallel lines, such that the edge (u_i, v_i) is orthogonal to (u_1, u_n) and (v_1, v_n) and all nodes in U are in the same sector of a node in V and vice versa.

In this situation we have $m = n/2$ edges for all Yao-based topologies, which all have to be erased if a node w pops up in the middle of the network. The inverse situation occurs if we switch off this node.

For the HL-graph we consider each of the $O(\log n)$ layer separately. If a station enters a layer, then at most a constant number of edges have to be added while no edges have to be erased. When a node disappears in a layer, we might have to determine some (at most 6) replacement nodes. These are chosen from the lower layer. Again in this level only a constant number of new edges have to be added. □

Clearly, this worst case behavior is not the typical situation. Therefore we introduce the number of involved vertices m as an additional parameter into the analysis of the time behavior of the enter/leave algorithms.

Theorem 20. *For a normal and nicely located vertex set V and m edges are involved an enter/leave operation can be performed in the Yao based graphs in time $O(m \log s)$.*

For the HL-graph the time is bounded by $O(\log |V| + \log s)$.

Proof Sketch: We will show how the **Yao-graph** can be established using $O(m \log s)$ rounds. It turns out that these ideas can be extended to construct the SymmY-, SparsY-, and the BoundY-graph in the same time. A complete proof can be found in the full paper.

`enter`: First, inform all nodes V that a new node u_0 has entered. All informed nodes immediately halt normal packet transportation. This needs only one round (or $O(\log n)$ rounds with high probability for the BoundY-graph, see discussion above).

Find next neighbor in each sector. Due to nice locations this can be accomplished in $O(\log s)$ rounds using a binary search algorithm. Note that since other communication is inhibited, we can interpret interference signals as answers.

For each sector all nodes with free sectors are asked to establish edges ending at u_0. For this, we iteratively determine the nearest node in time $O(\log s)$. This takes $O(m \log s)$ rounds

At last, for each sector all nodes u who already have a next neighbor v in a sector are asked to test whether u_0 is closer than this neighbor by sending with transmission distance $d_{D(u,v)-1}$. If u_0 now receives a signal it can successively determine all these m nodes and establish links in time $O(m \log s)$.

`leave`: During the normal packet routing scheme a neighbor u' of u_0 notices that u_0 has left. Then, u' informs all nodes V that u_0 has left. All informed nodes immediately halt packet transportation. This needs $O(\log n)$ rounds with high probability.

After all nodes have been informed all nodes v adjacent to u_0 need to determine new neighbors. We can assume that u_0 has prepared its m neighbored nodes by assigning them ranks. According to this ordering each of these nodes performs a reduced version of the `enter`-algorithm. This costs $O(m \log s)$ rounds since m is the number of involved edges.

For the **HL-graph** it is straightforward, that connections of each layer can be established and added in constant time (because each layer has constant degree and a constant bi-directional interference number). At the beginning each entering node has to learn which layer is active and starts a binary search over the transmission distances. In Theorem 19 we have seen that in each layer only a constant number of vertices change. Because of the constant degree of each layer only a constant number of links have to be changed or established. This leads to the additional update time linear in the number of layers, i.e. $O(\log |V|)$. □

7 Conclusions

The following table summarizes the results concerning communication and dynamic performance of the five graph topologies. It turns out that the best dynamic behavior can be achieved by the HL-graph. From the Yao-graph variants the SparsY graph outperforms the HL-graph on the approximation factor of congestion. In this overview the SymmY-graph gives the worst impression. Nevertheless, it guarantees that no signals interfere at all. Therefore for a small number of radio stations or average locations it may outperform all the other graph types.

Topology	Congestion approx. factor	Energy approx. factor	time for `enter` & `leave`	`enter`/`leave` involved nodes
Yao-graph	—	$O(1)$	$O(n \log s)$	$\Theta(n)$
SymmY-graph	—	—	$O(n \log s)$	$\Theta(n)$
SparsY-graph	$O(\log n)$	$O(1)$	$O(n \log s)$	$\Theta(n)$
BoundY-graph	—	$O(1)$	$O(n \log s)$	$\Theta(n)$
HL-graph	$O(\log^2 n)$	$O(1)$	$O(\log n + \log s)$	$O(\log n)$

References

[Arya et al. 1995] S. Arya, G. Das, D. M. Mount, J. S. Salowe, and M. H. M. Smid. Euclidean Spanners: Short, Thin, and Lanky. In *Proc. 27th Annual ACM Symposium on Theory of Computing (STOC'95)*, pages 489–498, 1995.

[Adler and Scheideler 1998] M. Adler and Ch. Scheideler. Efficient Communication Strategies for Ad-Hoc Wireless Networks (extended Abstract). In *Proc. 10th ACM Symposium on Parallel Algorithms and Architectures (SPAA'98)*, pages 259–268, 1998.

[Chen et al. 2001] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. In *Proc. 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom)*, Rome, Italy, July 2001.

[Fischer et al. 1998] M. Fischer, T. Lukovszki, and M. Ziegler. Geometric searching in walkthrough animations with weak spanners in real time. In *Proc. 6th Annual European Symposium on Algorithms (ESA'98)*, pages 163–174, 1998.

[Fischer et al. 1997] M. Fischer, F. Meyer auf der Heide, and W.-B. Strothmann. Dynamic data structures for realtime management of large geometric scenes. In *Proc. 5th Annual European Symposium on Algorithms (ESA'97)*, pages 157–170, 1997.

[Gao et al. 2001a] J. Gao, L. J. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Discrete mobile centers. In *Proc. 17th Symposium on Computational Geometry (SoCG'01)*, pages 188–196, Medford, MA, USA, 2001.

[Gao et al. 2001b] J. Gao, L. J. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Geometric spanner for routing in mobile networks. In *Proc. ACM Symposium on Mobile Ad Hoc Networking and Computing*, pages 45–55, 2001.

[K-Team S.A. 2000] K-Team S.A. *Khepera miniuature mobile robot*, 2000. http://www.k-team.com/robots/khepera.

[Lukovszki 1999] T. Lukovszki. New results on fault tolerant geometric spanners. In *Proc. 6th Workshop on Algorithms and Data Structures (WADS'99)*, pages 193–204, 1999.

[Lukovszki et al. 2002] T. Lukovszki, Ch. Schindehauer, K. Volbert, and M. Grünewald. Distributed maintenance of resource efficient wireless network topologies. In *Proc. European Conference on Parallel Computing (EUROPAR'02)*, pages 935–946, 2002.

[Mondada et al. 1999] F. Mondada, E. Franzi, and A. Guignard. The development of khepera. In *Proc. 1st International Khepera Workshop*, pages 7–13, 1999.

[Meyer auf der Heide et al. 2004] F. Meyer auf der Heide, Ch. Schindelhauer, K. Volbert, and M. Grünewald. Congestion, Energy and Delay in Radio Networks. In *Theory of Computing Systems (TOCS)*, Vol. 37, 343–370, 2004.

[Ruppert and Seidel 1991] J. Ruppert and R. Seidel. Approximating the d-dimensional complete Euclidean graph. In *3rd Canadian Conference on Computational Geometry (CCCG'91)*, pages 207–210, 1991.

[Schindelhauer et al. 2003] Ch. Schindelhauer, T. Lukovszki, S. Rührup, and K. Volbert. Worst case mobility in ad hoc networks. In *Proc. 15th ACM Symposium on Parallel Algorithms and Architectures (SPAA'03)*, pages 230–239, 2003.

[Wang and Li 2002] 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.

[Xu et al. 2001] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed Energy Conservation for Ad Hoc Routing. In *Proc. 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom)*, 2001.

[Yao 1982] A. C. Yao. On Constructing Minimum Spanning Trees in k-Dimensional Spaces and Related Problems. In *SIAM Journal on Computing*, Vol. 11, 721–736, 1982.