Algorithmic Models of Interference in Wireless Ad Hoc and Sensor Networks

Pascal von Rickenbach, Roger Wattenhofer, and Aaron Zollinger, *Member, IEEE*

***Abstract*—Among the most critical issues of wireless ad hoc and sensor networks are energy consumption in general and interference in particular. The reduction of interference is consequently considered one of the foremost goals of topology control. Almost all of the related work however considers this issue implicitly: Low interference is often claimed to be a consequence of sparseness or low degree of the constructed topologies. This paper, in contrast, studies explicit definitions of interference. Various models of interference—both from a sender-centric and a receiver-centric perspective—are proposed, compared, and analyzed with respect to their algorithmic properties and complexities.**

***Index Terms*—Algorithmic analysis, interference, modeling, network connectivity, network spanners, topology control.**

I. Introduction

ONE MANIFESTATION of the currently observed and continuing miniaturization of electronics in general and wireless communication technology in particular is mobile *ad hoc networks*. Ad hoc networks are formed by mobile devices consisting of, among other components, a processor, some memory, a radio communication unit, and a power source, due to physical constraints commonly a weak battery or a small solar cell.

Typically, wireless ad hoc networks are intended to be employed where no communication infrastructure is present before the deployment of the ad hoc network or where reliance on previously present infrastructure is not desired or not possible. Common scenarios for ad hoc networks include communication among rescue teams, police squads, or during fire fighting or other disaster relief actions.

Sensor networks can be considered a specialization of ad hoc networks in which nodes are equipped with sensors measuring certain physical values, such as humidity, brightness, temperature, acceleration, or vibration. Usually, the sensor nodes are designed to report measured information to a data sink node. Among the most common scenarios for sensor networks are environmental monitoring tasks, for instance to warn of imminent natural disasters or for the purpose of biological or other scientific observations.

Since ad hoc and sensor network nodes are generally assumed to be autonomous and operate for a considerable period of time, in the case of sensor networks up to several years, *energy conservation* is one of the central issues in this research context.

Fig. 1. Topology control constitutes a tradeoff between node energy conservation and network connectivity.

In a very general sense, *topology control* in wireless ad hoc and sensor networks can be considered the task of, given a network connectivity graph, computing a subgraph with specific desired properties, such as connectivity, short stretches, sparsity, low interference, or low node degree. Sometimes also the construction of node clusters and dominating sets of nodes is considered topology control.

In this paper we focus on an equally popular conception of topology control: Simply put, the main goal of topology control is often understood to be the reduction of energy consumed by the network nodes in order to extend network lifetime. Since the amount of energy required to transmit a message increases at least quadratically with distance, it makes sense to replace a long link by a sequence of short links. On the one hand, energy can therefore be conserved by abandoning energy-expensive long-range connections, thereby allowing the nodes to reduce their transmission power levels. On the other hand, confining transmission ranges also reduces interference, which in turn lowers node energy consumption by reducing the number of collisions and consequently packet retransmissions on the media access layer. Dropping communication links however clearly takes place at the cost of network connectivity: If too many edges are abandoned, connecting paths can grow unacceptably long or the network can even become completely disconnected. As illustrated in Fig. 1, topology control can therefore be considered a tradeoff between energy conservation and interference reduction on the one hand and connectivity on the other hand.

If interference reduction has often been mentioned as one of the main goals of topology control, previous work (with few exceptions) has generally stated interference to be lowered implicitly, in particular as a consequence of low node degree of the constructed topology. In this paper we give an overview of attempts to analyze this statement from an algorithmic point of view. Specifically, two distinct interference models are introduced and worst case bounds for the different algorithms under consideration are given. For an average-case analysis we refer the interested reader to the simulation results in the corresponding conference papers [1] and [2], respectively. Section III comprises such a definition of an explicit interference model

Manuscript received June 19, 2006; revised January 17, 2007; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor I. Stavrakakis. Current version published February 19, 2009.

P. von Rickenbach and R. Wattenhofer are with the ETH Zurich, 8092 Zurich, Switzerland (e-mail: pascalv@tik.ee.ethz.ch; wattenhofer@tik.ee.ethz.ch).

A. Zollinger is with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720 USA (e-mail: zolling@eecs.berkeley.edu).

Digital Object Identifier 10.1109/TNET.2008.926506

and a discussion showing that almost all previously proposed topology control algorithms—even if constructing topologies with bounded degree—do not always effectively reduce interference.

The interference model defined in Section III is based on the number of nodes affected by communication over a given link. In other words, this model focuses on the *sending* process of message transmission. It can be argued that such a perspective puts the cart before the horse, as interference and in particular message collisions actually occur at the intended *receiver* of a message. Such a receiver-centric interference model is discussed in Section IV in the context of data gathering in sensor networks and is extended for application in general ad hoc networks in Section V.

II. Related Work

The assumption that nodes are distributed randomly in the plane according to a uniform probability distribution formed the basis of pioneering work in the field of topology control in ad hoc networks [3], [4].

Later proposals adopted constructions originally studied in computational geometry, such as the Delaunay Triangulation [5], the minimum spanning tree [6], the Relative Neighborhood Graph [7], or the Gabriel Graph [8]. Most of these contributions mainly considered energy-efficiency of paths preserved by the resulting topology, whereas others exploited the planarity property of the proposed constructions for geographic routing [9]–[12].

The Delaunay Triangulation and the minimum spanning tree not being computable locally and thus not being practicable, a next generation of topology control algorithms emphasized locality. The CBTC algorithm [13] was the first construction to simultaneously focus on several desired properties, in particular being an energy spanner with bounded degree. This process of developing local algorithms featuring more and more properties was continued, partly based on CBTC, partly based on local versions of classic geometric constructions such as the Delaunay Triangulation [14] or the minimum spanning tree [15]. Among the most recent such results are a locally computable planar constant-stretch distance (and energy) spanner with constant-bounded node degree [16] or a construction with similar properties additionally having low overall energy consumption [17]. Other approaches try to build on minimal assumptions about the capabilities of nodes and signal propagation characteristics [18]. Yet another thread of research takes up the average-graph perspective of early work in the field; [19] for instance shows that the simple algorithm choosing the k nearest neighbors works surprisingly well in such graphs.

A different aspect of topology control is considered by algorithms trying to form clusters of nodes. Most of these proposals are based on (connected) dominating sets [20]–[27] and focus on locality and provable properties. Cluster-based constructions are commonly regarded as a variant of topology control in the sense that energy-consuming tasks can be shared among the members of a cluster.

Topology control having so far mainly been of interest to theoreticians, first promising steps are being made towards exploiting the benefit of such techniques also in practical networks [28]. A more detailed overview of topology control techniques in general can be found in [29].

As mentioned earlier, reducing interference—and its energy-saving effects on the medium access layer—is one of the main goals of topology control besides direct energy conservation as a consequence of transmission power restriction. Astonishingly however, all the above topology control algorithms at the most implicitly try to reduce interference. Where interference is mentioned as an issue at all, it is maintained to be confined at a low level as a consequence of sparseness or low degree of the resulting topology graph.

A notable exception to this is [30], defining an explicit notion of interference. Based on this interference model between edges, a time-step routing model and a concept of congestion is introduced. It is shown that there are inevitable tradeoffs between congestion, power consumption and dilation (or hop-distance). For some node sets, congestion and energy are even shown to be incompatible.

The interference model proposed in [30] is based on current network traffic. The amount and nature of network traffic however highly depends on the chosen application. Since usually no *a priori* information about the traffic in a network is available, a static model of interference depending solely on node constellations is consequently desirable. Such a traffic-independent notion of interference was introduced in [1] and is presented in Section III. As we show, the above statement that graph sparseness or small degree implies low interference is misleading. The interference model described in Section III builds on the question of how many nodes are affected by communication over a given link. As also discussed in more depth, this sender-centric perspective can however be accused to be somewhat artificial and to poorly represent reality, interference in fact occurring at the intended *receiver* of a message. Furthermore, this interference measure is shown to be susceptible to drastic effects even if single nodes are added to or removed from a network.

An attempt to correct for this deficiency was made in [2], as presented and discussed in Section IV. As we show, this work defines a receiver-centric concept of interference in the context of data-gathering structures in sensor networks. The issue of energy efficiency in sensor networks [31]–[33], particularly extending network lifetime, has been mainly studied with respect to optimal sensor placement and energy-efficient routing. Recently also the fact that certain types of sensed data allow for aggregation at sensor nodes [34] and the existence of redundancy in acquired information [35], [36], for instance correlation between sensed data depending on the distance between sensors, has been considered.

The interference modeling approach originally presented in [37] and discussed in Section V goes beyond sensor networks by defining and employing a suitable robust interference model for the analysis of topology control in ad hoc networks in general.

The static interference models, in the sense that they are defined independent of current network traffic, as introduced in [1], [2], [37] and summarized in this paper, formed the basis for continuing research [38]–[40]. The interference issue as a scheduling problem over time, in a sense taking up the approach from [30], was later again studied in [41]. In contrast to the work presented in this paper, [41] models interference with the physical signal-to-interference-plus-noise ratio (SINR) and defines the concept of scheduling complexity for the connectivity of wireless networks. [42] extends this concept to arbitrary given network topologies and demonstrates the existence of a relation between this scheduling complexity and the receiver-centric interference model defined in Section V.

III. Does Topology Control Reduce Interference?

In contrast to most of the related work, where the interference issue is seemingly solved by sparseness arguments, we start out by precisely defining a first concept of interference. This definition of interference is based on the natural question of how many nodes are affected by communication over a certain link. By prohibiting specific network edges, the potential for communication over high-interference links can then be confined.

We employ this interference definition to formulate the tradeoff between energy conservation and network connectivity. In particular we state certain requirements that need to be met by the resulting topology. Among these requirements are connectivity (if two nodes are, possibly indirectly, connected in the given network, they should also be connected in the resulting topology) and the constant-stretch spanner property (the shortest path between any pair of nodes in the resulting topology should be longer at most by a constant factor than the shortest path connecting the same pair of nodes in the given network). After stating such requirements, an optimization problem can be formulated to find the topology meeting the given requirements with minimum interference.

For the requirement that the resulting topology retain connectivity of the given network, we show that most of the currently proposed topology control algorithms, already by having every node connect to its nearest neighbor, commit a substantial mistake: Although certain proposed topologies are guaranteed to have low degree yielding a sparse graph, interference becomes asymptotically incomparable with the interference-minimal topology. We also show that there exist graphs for which no local algorithm can approximate the optimum. With respect to the sometimes desirable requirement that the resulting topology should be planar, we show that planarity can increase interference.

Furthermore we propose a centralized algorithm (LIFE) that computes an interference-minimal connectivity-preserving topology. For the requirement that the resulting topology be a spanner with a given stretch factor, we present (based on a centralized variant of the algorithm) a distributed local algorithm (LLISE) that computes a provably interference-optimal spanner topology.

A. Model

We model an ad hoc network as a graph $G=(V,E)$ consisting of a set of nodes $V \subset \mathbb{R}^2$ in the Euclidean plane and a set of edges $E \subseteq V^2$. Nodes represent mobile hosts, whereas edges stand for links between nodes. In order to prevent already basic communication between directly neighboring nodes from becoming unacceptably cumbersome [43], it is required that a message sent over a link can be acknowledged by sending a corresponding message over the same link in the opposite direction. In other words, only *undirected* (symmetric) edges are considered.

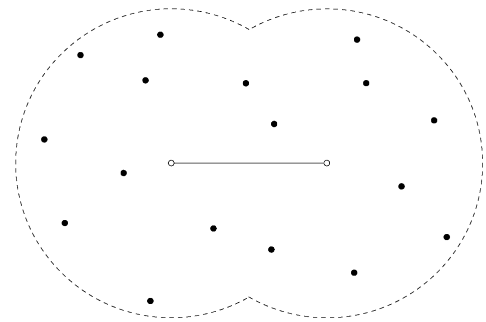

Fig. 2. Nodes covered by a communication link.

We assume that a node can adjust its transmission power to any value between zero and its maximum power level. The maximum power levels are not assumed to be equal for all nodes. An edge (u,v) may exist only if both incident nodes are capable of sending a message over (u,v), in particular if the maximum transmission radius of both u and v is at least $|uv|$, their Euclidean distance. A pair of nodes u, v is considered *connectable in the given network* if there exists a path connecting u and v provided that all transmission radii are set to their respective maximum values. The task of a *topology control* algorithm is then to compute a subgraph of the given network graph with certain properties, reducing the transmission power levels and thereby attempting to lower interference and energy consumption.

With a chosen transmission radius, for instance to reach a node v, a node u affects at least all nodes located within the circle centered at u and with radius $|uv|$. Denoting $D(u,r)$ to be the disk centered at node u with radius r and requiring edge symmetry, we consequently define the *coverage* of an (undirected) edge $e=(u,v)$ to be the cardinality of the set of nodes covered by the disks[1] induced by u and v:

$$\mathrm{Cov}(e) := |\{w \in V | w \text{ is covered by } D(u,|uv|)\} \cup \{w \in V | w \text{ is covered by } D(v,|vu|)\}|.$$

In other words, the coverage $\mathrm{Cov}(e)$ represents the number of network nodes affected by nodes u and v communicating with their transmission power levels chosen such that they exactly reach each other (cf. Fig. 2).

The edge level interference defined so far is now extended to a graph interference measure as the maximum coverage occurring in a graph:

Definition 3.1: The interference of a graph $G=(V,E)$ is defined as

$$I(G) := \max_{e \in E} \mathrm{Cov}(e).$$

Since interference reduction *per se* would be senseless (if all nodes simply set their transmission power to zero, interference will be reduced to a minimum), the formulation of additional requirements to be met by a resulting topology is

[1]The results of this section can also be adapted to the case where transmission ranges are not perfect circles centered at the sending nodes. We adhere to this simplified model for clarity of representation.

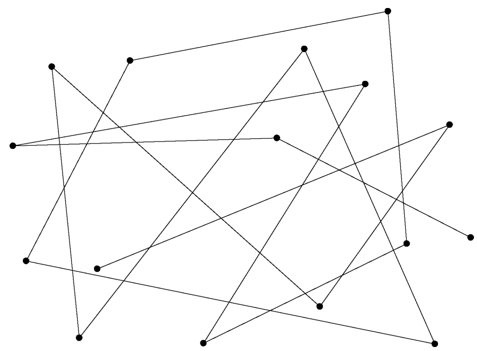

Fig. 3. Low degree does not guarantee low interference.

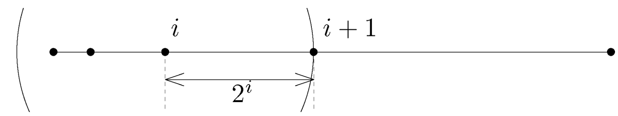

Fig. 4. Exponential node chain with interference $\Omega(n)$.

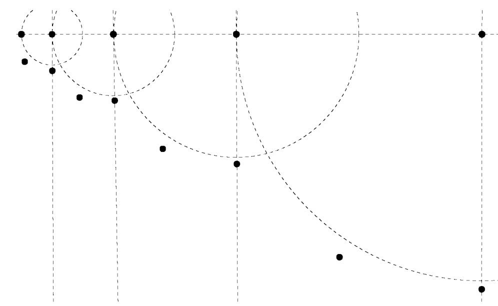

Fig. 5. Two exponential node chains.

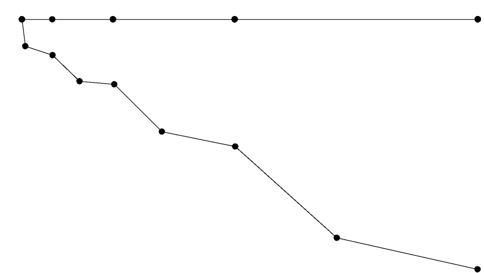

Fig. 6. The Nearest Neighbor Forest yields interference $\Omega(n)$.

necessary. A resulting topology can for instance be required for the following:

- to maintain connectivity of the given communication graph (if a pair of nodes is connectable in the given network, it should also be connected in the resulting topology graph);
- to be a spanner with constant stretch of the underlying graph (the shortest path connecting a pair of nodes u, v in the resulting topology is longer by a constant factor only than the shortest path between u and v in the given network); or
- to be planar (no two edges in the resulting graph intersect).

Finding a resulting topology which meets one or a combination of such requirements with minimum interference constitutes an optimization problem.

B. Interference in Known Topologies

It is often argued that sparse topologies with small or bounded degree are well suited to minimize interference. In this section, we show that low degree does not necessarily imply low interference. Moreover, we demonstrate that most of the currently known topology control algorithms can perform badly compared to the interference optimum, that is a topology which minimizes interference in the first place.

In particular, we consider in this section the basic problem of constructing an interference-minimal topology maintaining connectivity of the given network.

The following basic observation states that—although often maintained—low degree alone does not guarantee low interference. Fig. 3, for instance, shows a topology graph with degree 2 whose interference is however roughly n, the number of network nodes. A node can interfere with other nodes that are not direct neighbors in the chosen topology graph. Whereas twice the maximum degree of the underlying communication graph of the given network (with all nodes transmitting at full power) is an upper bound for interference, the degree of a resulting topology graph is only a lower bound.

There exist instances where also the optimum exhibits interference $\Omega(n)$, for instance a chain of nodes with exponentially growing distances (cf. Fig. 4, proposed in [30]), whose large interference is caused as a consequence of the requirement that the resulting topology is to be connected. Every node u_i (except for the leftmost) is required to have an incident edge, which covers all nodes left of u_i. Assessing the interference quality of a topology control algorithm therefore implies that its interference on a given network needs to be compared to the optimum interference topology for the same network.

To the best of our knowledge, all currently known topology control algorithms constructing only symmetric connections (and not accounting for explicit interference) have in common that every node establishes a symmetric connection to at least its nearest neighbor. In other words, all these topologies contain the Nearest Neighbor Forest constructed in the given network. In the following, we show that owing to the inclusion of the Nearest Neighbor Forest as a subgraph, the interference of a resulting topology can become incomparably bad with respect to a topology with optimum interference.

Theorem 3.1: No currently proposed topology control algorithm establishing only symmetric connections, required to maintain connectivity of the given network, is guaranteed to yield a nontrivial interference approximation of the optimum solution. In particular, interference of any proposed topology can be $\Omega(n)$ times larger than the interference of the optimum connected topology, where n is the total number of network nodes.

Proof Sketch: [2] Given a network configuration with two exponential node chains as depicted in Fig. 5, computation of the Nearest Neighbor Forest results in the topology shown in Fig. 6 with interference $\Omega(n)$. It is however possible to construct a connected topology with constant interference, as illustrated in Fig. 7.

[2]Due to space limitations, the statements in this paper are presented without proofs. We refer the interested reader to the corresponding conference papers [1] (Section III), [2] (Section IV), and [37] (Section V), respectively.

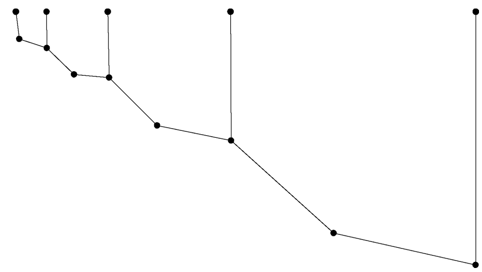

Fig. 7. Optimal tree with constant interference.

In other words, already by having each node connect to the nearest neighbor, a topology control algorithm makes an "irrevocable" error. Moreover, it commits an asymptotically worst possible error since the interference in any network cannot become larger than n.

As roughly one third of all nodes are part of the horizontal exponential node chain in Fig. 5, the observation stated in Theorem 3.1 would also hold for an average interference measure, averaging interference over all edges.

It can even be shown that for connectivity-preserving topologies no local algorithm can approximate optimum interference for every given network. Thereby the definition of a distributed *local* algorithm assumes that each network node is informed about its network neighborhood only up to a given constant distance.

Theorem 3.2: For the requirement of maintaining connectivity of the given network, there exists a class of graphs for which no local algorithm can approximate optimum interference.

As mentioned in Section III-A, another popular requirement for topology control algorithms besides bounded degree is planarity of the resulting topology, meaning that no two edges of the resulting graph intersect. This is often desired for the application of geographic routing algorithms that are only applicable to planar graphs. Topology control algorithms enforcing planarity are however not optimal in terms of interference:

Theorem 3.3: There exist graphs in which interference-optimal topologies, required to maintain connectivity, are not planar.

C. Low-Interference Topologies

In this section, we present three algorithms that explicitly reduce interference of a given network. The first algorithm is capable of finding an interference-optimal topology maintaining connectivity of the given network. The other two algorithms compute an interference-optimal topology with the additional requirement of constructing a spanner of the given network. Whereas the first spanner algorithm assumes global knowledge of the network, the second can be computed locally.

1) Interference-Optimal Spanning Forest:

In the following, we again require the resulting topology to maintain connectivity of the given network. A topology graph meeting this requirement can therefore consist of a tree for each connected component of the given network since additional edges do not contribute to graph connectivity while potentially unnecessarily increasing interference. A *Minimum Interference Forest* is therefore a set of trees maintaining the connectivity of the given network with least possible interference. It can be shown that the LIFE algorithm (Algorithm 1) computes such a forest.

Input: a set of nodes V, each $v \in V$ having attributed a maximum transmission radius r_v^{max}
1: E = all eligible edges (u, v) ($r_u^{max} \geq |uv|$ and $r_v^{max} \geq |uv|$)
// E will contain all unprocessed edges
2: $E_{LIFE} = \emptyset$
3: $G_{LIFE} = (V, E_{LIFE})$
4: **while** $E \neq \emptyset$ **do**
5: $e = (u, v) \in E$ with minimum coverage
6: **if** u, v are not connected in G_{LIFE} **then**
7: $E_{LIFE} = E_{LIFE} \cup \{e\}$
8: **end if**
9: $E = E \setminus \{e\}$
10: **end while**
Output: Graph G_{LIFE}

Algorithm 1: Low Interference Forest Establisher (LIFE).

Theorem 3.4: The forest constructed by LIFE is a Minimum Interference Forest.

With an appropriate implementation of the connectivity query in Line 6, the running time of the algorithm LIFE is $O(n^2 \log n)$. If the given network is known to consist of only one connected component, Prim's minimum-spanning-tree algorithm can be employed with running time $O(n^2)$. Algorithms computing a minimum spanning tree in a distributed way, as particularly suitable for ad hoc networks, are described in detail in [44].

2) Low-Interference Spanners: LIFE optimizes interference for the requirement that the resulting topology has to maintain connectivity. In addition to connectivity it is often desired that the resulting topology should be a spanner with constant stretch of the given network. A spanner with stretch factor t can be formally defined as follows:

Definition 3.2 (t-Spanner): A t-spanner of a graph $G = (V, E)$ is a subgraph $G' = (V, E')$ such that for each pair (u, v) of nodes $c(p^*_{G'}(u, v)) \leq t \cdot c(p^*_G(u, v))$, where $c(p^*_{G'}(u, v))$ and $c(p^*_G(u, v))$ denote the length of the shortest path between u and v in G' and G, respectively.

In this section, we consider Euclidean spanners, that is, the length of a path is defined as the sum of the Euclidean lengths of all its edges. With slight modifications, our results are however extendable to hop spanners, where the length of a path corresponds to the number of its edges.

Algorithm LISE (Algorithm 2) is a topology control algorithm that constructs a t-spanner with optimum interference. LISE starts with a graph $G_{\mathrm{LISE}} = (V, E_{\mathrm{LISE}})$ where E_{LISE} is initially the empty set. It processes all eligible edges of the given network $G = (V, E)$ in descending order of their coverage. For each edge $(u, v) \in E$ not already in E_{LISE}, LISE checks whether there exists a path from u to v in G_{LISE} with Euclidean length at most $t\,|uv|$. As long as no such path exists, the algorithm keeps inserting all unprocessed eligible edges with minimum coverage into E_{LISE}. It can be shown that the resulting topology constructed by LISE is an interference-optimal t-spanner.

Input: a set of nodes V, each $v \in V$ having attributed a maximum transmission radius r_v^{max}; a stretch factor $t \geq 1$

E = all eligible edges (u, v) ($r_u^{max} \geq |uv|$ and $r_v^{max} \geq |uv|$)
// E will contain all unprocessed edges
$E_{LISE} = \emptyset$
$G_{LISE} = (V, E_{LISE})$
while $E \neq \emptyset$ **do**
  $e = (u, v) \in E$ with maximum coverage
  while $c(p^*(u, v)$ in $G_{LISE}) > t\,|uv|$ **do**
    f = edge $\in E$ with minimum coverage
    move all edges $\in E$ with coverage $Cov(f)$ to E_{LISE}
  end while
  $E = E \setminus \{e\}$
end while

Output: Graph G_{LISE}

Algorithm 2: Low Interference Spanner Establisher (LISE).

Theorem 3.5: The graph $G_{\mathrm{LISE}} = (V, E_{\mathrm{LISE}})$ constructed by LISE from a given network $G = (V, E)$ is an interference-optimal t-spanner of G.

As regards the running time of LISE, it computes for each edge at most one shortest path. Since finding a shortest alternative path for an edge requires On^2 time and as the network contains at most the same amount of edges, the overall running time of LISE is polynomial in the number of network nodes.

In contrast to the problem of finding a connected topology with optimum interference, the problem of finding an interference-optimal t-spanner is locally solvable. The reason for this is that finding an interference-optimal path $p(u, v)$ for an edge (u, v) with $c(p) \leq t|uv|$ can be restricted to a certain neighborhood of (u, v).

In the following, we describe a local algorithm similar to LISE that is executed at all eligible edges of the given network. In reality, algorithm LLISE (Local LISE, Algorithm 3) is executed for each edge by one of its incident nodes (for instance the one with the greater identifier). The description of LLISE assumes the point of view of an edge $e = (u, v)$. The algorithm consists of three main steps:

1) collect $(t/2)$-neighborhood;
2) compute minimum interference path for e; and
3) inform all edges on that path to remain in the resulting topology.

In the first step, e gains knowledge of its $(t/2)$-neighborhood. For a Euclidean spanner, the k-neighborhood of e is defined as all edges that can be reached (or more precisely at least one of their incident nodes) over a path p starting at u or v, respectively, with $c(p) \leq k\,c(e)$. Knowledge of the $(t/2)$-neighborhood at all edges can be achieved by local flooding.

Similarly as for LISE, it can be proved that the t-spanner constructed by LLISE is interference-optimal:

Theorem 3.6: The graph $G_{LL} = (V, E_{LL})$ constructed by LLISE from a given network $G = (V, E)$ is an interference-optimal t-spanner of G.

D. Concluding Remarks

The results in this section disprove the widely advocated assumption that sparse topologies automatically imply low interference. In contrast to most of the related work we provide an intuitive definition of interference. With this interference model we show that currently proposed topology control constructions, although claiming so, do not in the first place focus on reducing interference. In addition, we propose provably interference-minimal connectivity-preserving and spanner constructions.

As important as an explicit definition of interference is for its analysis, a clear drawback of the definition presented in this section is that it assumes the perspective of the *sender* of a message. Formally, this notion is reflected by the definition of the coverage of an edge, which counts the number of network nodes affected by communication over the considered edge. This stands in opposition to the fact that signal disturbance and message collisions actually occur at the intended *receiver* of a message. This characteristic of interference forms the core of the following sections discussing approaches to model interference which assume the signal receiver's perspective.

collect $(\frac{t}{2})$-neighborhood $G_N = (V_N, E_N)$ of $G = (V, E)$
$E' = \emptyset$
$G' = (V_N, E')$
repeat
  f = edge $\in E_N$ with minimum coverage
  move all edges $\in E_N$ with coverage $Cov(f)$ to E'
  p = shortestPath$(u - v)$ in G'
until $c(p) \leq t\,|uv|$
inform all edges on p to remain in the resulting topology.

Note: $G_{LL} = (V, E_{LL})$ consists of all edges eventually informed to remain in the resulting topology.

Algorithm 3: LLISE.

IV. Receiver-Centric Interference in Sensor Networks

The previous section states two main points. First, an implicit notion of interference, as advocated by most of the previous work, can lead to topology control algorithms that fail to effectively reduce interference. Second, it introduces an explicit definition of interference, based on the number of nodes potentially disturbed by communication over a link.

In contrast to this approach, we assume in this section a receiver-centric perspective. Particularly, we formulate an interference definition at the heart of which lies the question by how many other nodes a given network node can be disturbed. Compared to the sender-centric interference definition proposed in the previous section, the definition of interference presented in this section reflects intuition more closely in the sense that interference is considered at the receiver, where message collisions prevent proper reception. Informally, this interference definition can also be considered to correspond to the effort required to avoid collisions, be it by means of time division multiplexing, assigning transmission time slots such that no two messages collide at a receiving node, by means of frequency division multiplexing, having messages sent in different assigned frequency bands, or by means of code division multiplexing, where small interference allows for reduced spread factor.

In this section, we consider interference in sensor networks. As mentioned earlier, a sensor network consists of sensors deployed in a given region with the task of sensing a certain physical value (such as temperature, humidity, brightness, or motion). The sensors are equipped with radio devices and, in the popular monitoring scenario model, periodically transfer the sensed data to a designated data sink node. To allow all data to be gathered at the sink, a topology control algorithm therefore constructs a *sink tree*, a directed tree with all arcs (directed edges), modeling unidirectional communication links, pointing towards the sink node. In the context of interference reduction, the task of the topology control algorithm is to find such a sink tree with least possible interference. We thereby account for the fact that in the monitoring scenario communication from the sink to the sensors occurs rarely and can therefore be neglected with respect to interference.

Assuming a worst case perspective, we show that there are network instances in which any topology control algorithm will construct a resulting network with interference at least $\log n - 1$. We furthermore propose the *Nearest Component Connector (NCC)* algorithm, which provably produces at most $O(\log n)$ interference in any network in polynomial time. In this sense, the NCC algorithm is asymptotically optimal.

A. Model and Notation

In this subsection we describe our model of a sensor network and formally define receiver-centric interference and interference minimization in the context of this model.

Our model of a sensor network is a directed graph $G(V, E)$ where nodes $v_1, \ldots, v_n$ placed in the plane represent the set of sensors including the sink. Communication links between sensors are modeled as (directed) arcs. We also assume that the transmission power of each node can be adjusted. Higher transmission power allows a node to send messages over a longer distance. We assume that the covered area of a sending node v_i is a disk with v_i in its center. Furthermore we assume that a node can reach another node only if it is at most 1 distance unit away. In other words, the graph consisting of all eligible arcs if all nodes set their transmission power to the maximum possible values corresponds to the unit disk graph constructed given the node set V; the unit disk graph is defined such that it contains an edge (or two symmetric arcs) between two nodes if and only if their distance is at most 1. Finally, we only consider *connectable* graphs, which means that, with all transmission radii set to their maximum values, a path from any node to any other node in the network is constructible or, more technically, the unit disk graph given the node set V consists of one connected component.

If we want to minimize interference in sensor networks, we have to look at topologies in which each node sends its data to at most one other node, and a valid graph contains a path from every sensor to the sink. These two requirements result in a tree with the sink as its root and all arcs pointing towards the root. We call such a tree a *sink tree*. Fig. 8 shows a sample sink tree with 6 nodes.

Definition 4.1: Given a set of nodes V and a sink s, a sink tree is a tree spanning V with all arcs pointing towards s.

We use an explicit model of interference. We explicitly count the number of nodes potentially disturbing reception of a message. This definition reflects the fact that interference is a problem occurring at the receiver. Minimizing the interference at each possible receiver (each node in the network) reduces the number of potential message collisions in the network and therefore lowers the probability of required retransmission. This approach intends to save energy and extend the lifetime of sensors equipped with batteries.

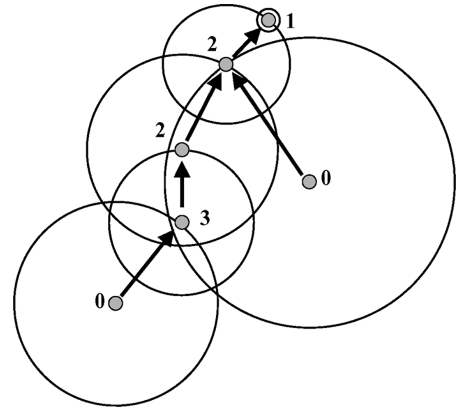

Fig. 8. A sink tree with 6 nodes, the uppermost node being the sink node. Each node is labeled with its interference value. The interference of the whole network is 3.

In particular, the interference value of a single node is defined to be the number of transmission circles by which the node is covered.

Definition 4.2: The interference value of a single node v is defined as

$$I(v) = |\{u | u \in V \setminus \{v\}, v \in D(u, r_u)\}|.$$

where $D(u, r_u)$ stands for the transmission circle with node u in its center and radius r_u.

The interference of a whole network is defined as the maximum of all interference values in the graph (see Fig. 8).[3]

Definition 4.3: The interference of a Graph $G(V, E)$ is defined as

$$I(G) := \max_{v \in V} I(v).$$

The problem we study in this section consists in finding a sink tree with least possible interference for a given sensor network.

Definition 4.4: The Minimum-Interference Sink Tree (MIST) problem is defined as the problem of finding a sink tree for a given node set with minimal interference.

In the remainder of this section we consider topology control algorithms with the goal of solving the MIST problem.

B. A Lower Bound

With a recursively defined network as illustrated in Fig. 9, it can be shown that n nodes in a sensor network can be arranged in a way that there exists no algorithm able to construct a sink

[3]It can be argued similarly that the interference of a whole network can be defined as the *average* of the node interference values. Such a definition is not considered in this section. Note that with this alternative definition of interference, the problem of finding a valid data-gathering structure with minimum interference can be solved optimally by constructing a Minimum Directed Spanning Tree with arc weights corresponding to the number of nodes covered by each edge.

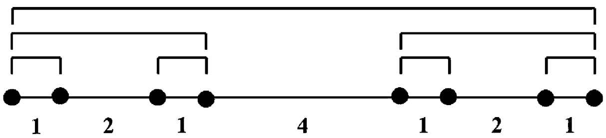

Fig. 9. A recursive arrangement of eight nodes on a horizontal line. The labels indicate distances.

Input: V: a set of nodes placed in the plane
$s_g \in V$: a predefined global sink

$G := (V, E := \emptyset)$
$lsinks := V$ // set of local sinks
while $|lsinks| > 1$ **do**
  for all $s \in lsinks$ **do**
    $E' := \emptyset$
    $C :=$ component containing s
    if s cannot reach any node outside C **then**
      $s' :=$ nearest node to s (hop metric) capable of reaching a node outside C
      $movesink(G, s, s')$
      $s := s'$
    end if
    $E' := E' \cup \{e\}$, where e is the arc from s to its nearest neighbor (Euclidean distance) outside C
  end for
  if $G' := (V, E \cup E')$ contains cycles **then**
    remove one of the arcs in each cycle from E'
  end if
  $G := G'$
  $lsinks :=$ sinks in G // sinks are nodes having no outgoing arc
end while
$s :=$ only remaining sink in $lsinks$
if $s \neq s_g$ **then**
  $movesink(G, s, s_g)$
end if

Output: G

Algorithm 4: Nearest Component Connector Algorithm (NCC).

tree with interference less than $\log(n) - 1$ no matter which node is acting as a sink. The existence of such examples constitutes a lower bound with respect to interference.

Theorem 4.1: There exist sensor networks with nodes arranged in a way that no algorithm can construct a sink tree with interference less than $\log(n) - 1$.

C. NCC Algorithm

Having presented a lower bound on interference in the previous section, we introduce in this section the Nearest Component Connector algorithm (NCC) matching this lower bound and being described in detail in Algorithms 4 and 5.

The general idea of this algorithm is to connect components to their nearest neighbors. This is done in several rounds and leads to a sink tree. A component can be a single node or a group of previously connected nodes. When the algorithm starts, each node in the given sensor network forms a component of its own.

Input: Graph $G = (V, E)$
s_1: a local sink in G
s_2: a node in the same component as s_1

$sp :=$ shortest path from s_1 to s_2 according to the hop metric
remove all arcs originating at nodes on sp (including s_2) from E
add arcs on sp to E

Algorithm 5: Procedure $\mathrm{movesink}(G, s_1, s_2)$.

First, the predefined global sink is treated exactly as a normal node. Whenever two or more components are connected in one round, they form a single component in the following round of NCC. Considering an arbitrary component at any point of time of the algorithm execution, we observe that this component has exactly one node all other component members have a directed path to. This means that there is one node which gathers all sensed data of the component. We call this node the *local sink* of its component.

Whenever a new arc is established during the execution of NCC, it goes from a local sink of a component C to the nearest node not in C. However, due to the fact that all nodes have maximum transmission range 1, it is possible that the current sink s of a component C cannot connect to any node outside C. In this case another node s' is designated to become the new sink of C, particularly the nearest node to s (with respect to the number of hops) capable of reaching any node outside C. This is accomplished by removing all arcs originating at nodes on the shortest path $p^*(s, s')$ from s to s' and subsequently adding the arcs along $p^*(s, s')$ (cf. Algorithm 5). Note that every component contains at least one node capable of reaching another node outside its component since we only consider connectable networks.

If a round, connecting every sink to its nearest neighbor outside its component, produces a cycle, this cycle is broken by removing one of its arcs at the end of the round. This leads to the construction of a valid sink tree topology. It is possible that, after the last round of NCC, the root of the resulting tree is not necessarily the global sink. In this case, the root of the resulting tree is moved to the global sink again by means of the movesink procedure (Algorithm 5). Fig. 10 shows a sample execution of the NCC algorithm.

It can be proved that the presented NCC algorithm constructs a valid sink tree topology for a given sensor network consisting of n nodes with an interference value in $O(\log n)$. It can also be shown that the execution of NCC takes polynomial time only:

Theorem 4.2: The NCC Algorithm constructs a sink tree in a given Graph $G = (V, E)$ with $|V| = n$ producing an interference value of at most $O(\log n)$ in polynomial time.

Proof Sketch: The proof consists of three parts. First, it can be shown that NCC does not need more than $\log n$ rounds (while-loop iterations) to build the sink tree. Second, the interference value of a node is not incremented by more than a constant value in each of these rounds. Third, it can be shown that NCC terminates in time polynomial in the total number of nodes.

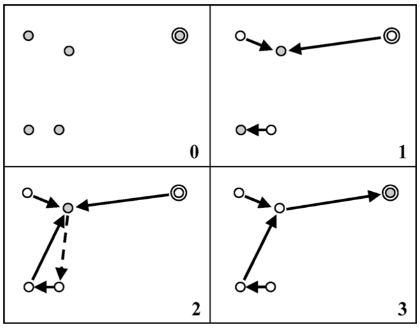

Fig. 10. A sample execution of NCC on a given set of 5 nodes. Situation 0 shows the given nodes and the predefined sink (top right node). In each of the following two rounds, every local sink connects to the nearest node not in its own component. In Round 2, a cycle is produced. It is broken at the end of the round by removing one of the involved arcs (dashed arrow). After the last round (Situation 3) the arc originating from the global sink is removed and an arc is added from the only remaining local sink to the predefined global sink. For clarity of representation, the node distances are assumed to be sufficiently small such that execution of the $\mathrm{movesink}$ procedure is not required.

We present NCC in a centralized manner. This reflects the fact that in a sensor network we have an instance (the sink) commonly assumed to have much more computing power and energy than all other nodes (sensors). Therefore, the sink can run NCC and distribute the topology information of the constructed sink tree in an initialization phase.

Nevertheless a distributed variant of NCC without the coordination of a central instance is feasible. This variant would require counters in each node which keep track of the number of component unions the node has been involved in since the start of the algorithm. These counters then guarantee that only components in the same "round" can establish new arcs between each other. Also the computation of shortest paths and the movesink procedure are implementable in a distributed way using a variant of flooding and by sending according messages over the shortest path thereby found.

D. Concluding Remarks

The approach assumed in this section in order to study interference in wireless and particularly sensor networks differs from most of the previous work in two ways: First, an explicit definition of interference is introduced. Second, in contrast to the interference model presented in the previous section, this definition of interference is receiver-centric and reflects the fact that message collisions prevent proper message reception only if they occur at the receiving node.

With this formalized notion of interference, we show on the one hand that there exist instances of sensor networks with n nodes in which it is impossible to construct a sink tree, a valid data gathering structure, with interference less than $\log n - 1$. On the other hand, we describe the NCC algorithm asymptotically matching this lower bound in that it provably builds a sink tree with interference at most $O(\log n)$ in any given sensor network.

The considerations presented in this section are restricted to sensor networks. Technically, this is reflected in the formulation of the Minimum-Interference Sink Tree problem. In the following section we will endeavor a first step towards extending a receiver-centric approach to the modeling of interference in more general ad hoc networks, particularly dropping the requirement of constructing a sink tree and instead focusing on graph connectivity.

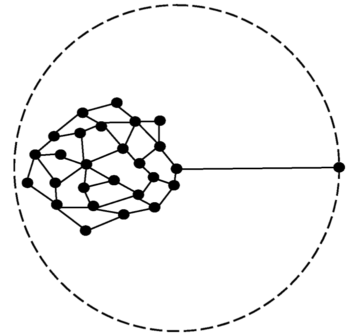

Fig. 11. In the interference model presented in Section III, addition of a single node increases interference from a small constant to the maximum possible value, the total number of network nodes.

V. A Robust Interference Model for Ad Hoc Networks

As mentioned in the previous section, the definition of interference introduced in Section III is problematic in two respects. First, it is based on the number of nodes affected by communication over a given link. In other words, interference is considered to be an issue at the sender instead of at the receiver, where message collisions actually prevent proper reception. It can therefore be argued that such a sender-centric perspective hardly reflects real-world interference. Section IV presents an approach to the modeling of interference from a receiver-centric perspective in the context of sensor networks. The second weakness of the model introduced in Section III is of more technical nature. According to its definition of interference, adding a single node to a given network can dramatically influence the interference measure. In the network depicted in Fig. 11, addition of the rightmost node to the cluster of roughly homogeneously distributed nodes entails the construction of a communication link covering all nodes in the network; accordingly, merely by introduction of one additional node, the interference value of the represented topology is pushed up from a small constant to the maximum possible value, that is the number of nodes in the network. This behavior contrasts to the intuition that a single additional node also represents but one additional packet source potentially causing collisions.

In contrast to that sender-centric interference definition, this section adopts the model presented in the previous section, explicitly considering interference at its point of impact, particularly at the receiver. Informally, the definition of interference considered in this section is based on the natural question by how many other nodes a given network node can be disturbed. The fact that this section adapts the receiver-centric concept of interference, introduced in Section IV for sensor networks, to be suitable also in general ad hoc networks is technically reflected in the consideration of arbitrary undirected networks as opposed to the directed data gathering trees studied in Section IV.

Interestingly, this interference definition not only reflects intuition due to its receiver-centricity. It also results in a robust interference model in terms of measure increase due to the arrival of additional nodes in the network. Particularly, an additional node causes an interference increase of at most one at other nodes of the network. In clear contrast to the sender-centric model from Section III, this corresponds to reality, where one added node contending for the shared medium constitutes only one additional possible collision source for nearby nodes in the network.

As already mentioned earlier, interference reduction as an end in itself is meaningless, every node setting its transmission power to a minimum value trivially minimizes interference, without the formulation of additional requirements to be met by the resulting topology. In this section, we study the fundamental requirement that the considered topology control algorithms preserve connectivity of the given network. Similarly as in Section III, we show that for this requirement most of the currently proposed topology control algorithms trying to implicitly reduce interference commit a substantial mistake, even by having every node connect to its nearest neighbor. Based on the intuition that already one-dimensional networks exhibit most of the complexity of finding minimum-interference topologies, we precisely anatomize networks restricted to one dimension—a model also known as the *highway* model. We first look at a particular network where distances between nodes increase exponentially from left to right. [30] introduces this network as a high-interference example yielding interference $O(\Delta)$, where Δ is the maximum node degree. We show that it is intriguingly possible to achieve interference $O(\sqrt{\Delta})$ in our model for this network, which matches a lower bound also presented in this section. Based on the insights thereby gained, we then consider general highway instances where nodes can be distributed arbitrarily in one dimension. For the problem of finding a minimum-interference topology while maintaining connectivity, we propose an approximation algorithm with approximation ratio $O(\sqrt[4]{\Delta})$.

A. Network and Interference Model

As customary, we model also in this section the wireless network by a unit disk graph G, where Δ refers to the maximum node degree in G. Again, in order to prevent already basic communication between neighboring nodes from becoming unacceptably cumbersome, we require that a message sent over a link can be acknowledged by sending a corresponding message over the same link in the opposite direction. In other words, only *undirected* (symmetric) edges are considered.

We assume that each node can adjust its transmission power to any value between zero and its maximum transmission power level. As usual, the main goal of a topology control algorithm is then to compute a low-interference subgraph of the given network graph G that maintains connectivity.

Let N_u denote the set of all neighbors of a node $u \in V$ in the resulting topology. Then, each node u features a value r_u defined as the distance from u to its farthest neighbor. More precisely $r_u = \max_{v \in N_u} |uv|$, where $|uv|$ denotes the Euclidean distance between nodes u and v. Since we assume the nodes to use omnidirectional antennas, $D(u, r_u)$ denotes the disk centered at u with radius r_u covering all nodes that are possibly affected by message transmission of u to one of its neighbors. The transmission radii of the network nodes having been fixed, the definitions of node-level and graph-level interference correspond exactly to Definition 4.2 and Definition 4.3, respectively.

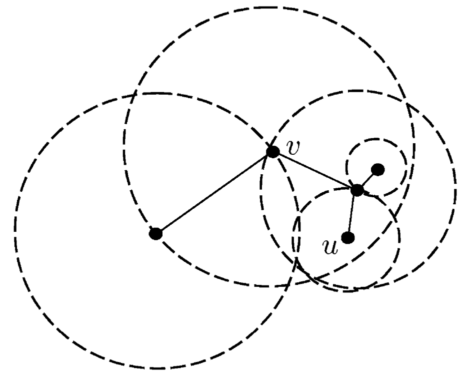

Fig. 12. A sample topology consisting of five nodes with their corresponding interference radii (dashed circles). Node u experiences interference $I(u) = 2$ since it is covered not only by its direct neighbor but also by node v.

Note that Δ, the maximum node degree of the given unit disk graph $G = (V, E)$, is an upper bound for the interference of any subgraph G' of the given graph since in G each node is directly connected to all potentially interfering nodes. However, in arbitrary subgraphs of G the degree of a node only lower-bounds the interference of that node because a node can be covered by non-neighboring nodes (cf. Fig. 12).

In this section, we study the combinatorial optimization problem of finding a resulting topology which maintains connectivity of the given network with minimum interference. Throughout the section we only consider topologies consisting of a tree for each connected component of the given network since additional edges might unnecessarily increase interference.

B. Interference in Known Topologies

As justified earlier, we restrict our considerations to resulting topologies consisting exclusively of symmetric links (edges). To the best of our knowledge, all currently known topology control algorithms (with the exception of the algorithms presented in Section III) constructing only symmetric connections have in common that every node establishes a link to at least its nearest neighbor. As also mentioned in Section III, this means in a technical sense that these topologies contain the so-called *Nearest Neighbor Forest* as a subgraph. With the example configuration also used in Section III, it can be shown that this is already a substantial mistake, as thus interference becomes asymptotically incomparable with the interference-minimal topology, also with this receiver-centric interference definition.

Theorem 5.1: Any algorithm containing the Nearest Neighbor Forest can have $\Omega(n)$ times larger interference than the interference of the optimum connected topology.

Although the topology control algorithms presented in Section III do not necessarily include the Nearest Neighbor Forest, it can be shown that also those algorithms perform badly for this receiver-centric interference model.

C. Analysis of the Highway Model

In this subsection we study interference for the highway model, in which the node distribution is restricted to one dimension. After analyzing an important artificially constructed problem instance, we provide a lower bound for interference of

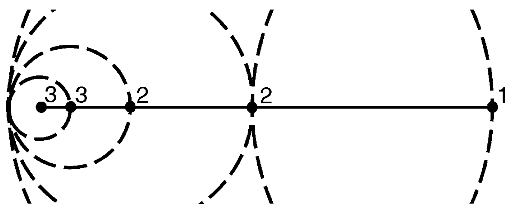

Fig. 13. Connecting the exponential node chain linearly yields interference $n-2$ at the leftmost node since each node connected to the right covers all nodes to its left. The nodes are labeled according to their experienced interference.

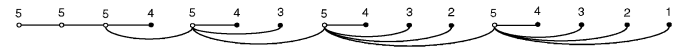

Fig. 14. The interference of the exponential node chain—shown in a logarithmic scale—is bounded by $O(\sqrt{n})$ by the topology control algorithm $\mathcal{A}_{\text{exp}}$. Only hubs (hollow points) interfere with the leftmost node. For clarity of representation, some edges are depicted as curved arcs.

general problem instances in the highway model as well as an asymptotically optimal algorithm matching this bound. Finally, an approximation algorithm is presented.

1) The Exponential Node Chain: How can n nodes arbitrarily distributed in one dimension connect to each other minimizing interference while maintaining connectivity? [30] introduces an instance which seems to yield inherently high interference: The so called *exponential node chain* is a one-dimensional graph $G=(V,E)$ where the distance between two consecutive nodes grows exponentially from left to right as depicted in Fig. 4. The distance between two nodes v_i and v_{i+1} in V is thus 2^i. Throughout the discussion of the exponential node chain, we furthermore assume that the whole node configuration is normalized in a way that the distance between the leftmost and the rightmost node is not greater than 1: Each node can potentially connect to all other nodes in V and therefore $\Delta=n-1$, where $n=|V|$. The nodes are termed *linearly connected* if each node, except for the leftmost and the rightmost, maintains an edge to its nearest neighbor to the left and to the right; in other words, node v_i is connected to node v_{i+1} for all $i=1,\ldots,n-1$ in the resulting topology. In addition to the disks $D(v_i,r_{v_i})$ for each node $v_i\in V$, Fig. 13 depicts their interference values $I(v_i)$. Since all disks but the one of the rightmost node cover v_1, interference at the leftmost node is $n-2\in\Omega(n)$; consequently also interference of the linearly connected exponential node chain is in $\Omega(n)$.

As we show in the following, the exponential node chain can surprisingly be connected in a significantly better way. According to the construction of the exponential node chain, only nodes connecting to at least one node to their right increase v_1's interference. We call such a node a *hub* and define it as follows:

Definition 5.1: Given a connected topology for the exponential node chain $G=(V,E)$, a node $v_i\in V$ is defined to be a *hub* in G if and only if there exists an edge (v_i,v_j) with $j>i$.

The following algorithm $\mathcal{A}_{\text{exp}}$ constructs a topology for the exponential node chain G which yields interference $O(\sqrt{n})$. The algorithm starts with a graph $G_{\text{exp}}=(V,E_{\text{exp}})$, where V is the set of nodes in the exponential node chain and E_{exp} is initially the empty set. Following the scan-line principle, $\mathcal{A}_{\text{exp}}$ processes all nodes in the order of their occurrence from left to right. Initially, the leftmost node is set to be the current hub h. Then, for each node v_i, $\mathcal{A}_{\text{exp}}$ inserts an edge (h,v_i) into E_{exp}. This is repeated until $I(G_{\text{exp}})$ increases due to the addition of such an edge. Now node v_i becomes the current hub and subsequent nodes are connected to v_i as long as the overall interference $I(G_{\text{exp}})$ does not increase. Fig. 14 depicts the resulting topology if $\mathcal{A}_{\text{exp}}$ is applied to the exponential node chain. The exponential node chain is thereby depicted in a logarithmic scale.[4] For clarity of representation, some edges in E_{exp} are drawn as curved arcs. In addition, Fig. 14 shows the individual interference values at each node.

It can be shown that $\mathcal{A}_{\text{exp}}$ reduces interference in the exponential node chain:

Theorem 5.2: Given the exponential node chain G, applying $\mathcal{A}_{\text{exp}}$ results in a connected topology with interference $I(G_{\text{exp}})\in O\sqrt{n}$.

This is an intriguing result since $\sqrt{n}$ is a lower bound for the interference of the exponential node chain. In particular, it can be proved that there exist network instances—again the exponential node chain—where every possible topology exhibits interference at least $\sqrt{n}$:

Theorem 5.3: Given an exponential node chain $G=(V,E)$ with $n=|V|$, $\sqrt{n}$ is a lower bound for the interference $I(G)$.

From Theorems 5.2 and 5.3 it follows that the $\mathcal{A}_{\text{exp}}$ algorithm is asymptotically optimal in terms of interference in the exponential node chain.

2) The General Highway Model: We consider an important artificially constructed instance in the highway model in the previous subsection, yielding a lower bound for the interference in arbitrary network graphs. In this subsection we go beyond the study of particular network instances and consider arbitrarily distributed nodes in one dimension.

A straightforward question is whether there are instances in the general highway model that are asymptotically worse than the exponential node chain, that is, where a minimum-interference topology exceeds $\Omega(\sqrt{\Delta})$. We answer this question in the negative by introducing the $\mathcal{A}_{\text{gen}}$ algorithm, which yields interference in $O(\sqrt{\Delta})$ for *any* given node distribution.

In a first step, the algorithm determines the maximum degree Δ of the given unit disk graph $G=(V,E)$ and partitions "the highway" into segments of unit length 1. Within such a segment σ, each node can potentially connect to every other node in σ.

In a second step, $\mathcal{A}_{\text{gen}}$ considers each segment independently as follows: Starting with the leftmost node of the segment, every $\lceil\sqrt{\Delta}\rceil$th node (according to their appearance from left to right) becomes a *hub*. A hub is thereby redefined along the lines of Definition 5.1 as a node that has more than one neighboring node, in contrast to *regular* nodes, which are connected to exactly one hub. In order to avoid boundary effects, the rightmost node of each segment is also considered a hub. Then the $\mathcal{A}_{\text{gen}}$ algorithm connects the hubs of a segment linearly. That is, each hub, except the leftmost and the rightmost, establishes an edge to its nearest hub to its left and to its right. Two consecutive hubs enclose an *interval*. $\mathcal{A}_{\text{gen}}$ connects all regular nodes in a particular interval to their nearest hub; ties are broken arbitrarily. Fig. 15 depicts one segment of an example instance after the

[4]In other words, the exponential node chain is viewed through a pair of glasses with "logarithmic cut."

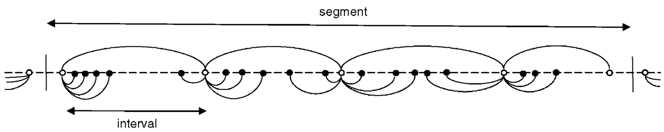

Fig. 15. $\mathcal{A}_{\text{gen}}$ partitions the highway into segments of length 1. In each segment, every $\lceil\sqrt{\Delta}\rceil$th node becomes a hub (hollow points). While the hubs are connected linearly, each of the remaining nodes in the interval between two hubs is connected to its nearest hub.

application of $\mathcal{A}_{\text{gen}}$. The nodes within a segment form one connected component.

Finally, the $\mathcal{A}_{\text{gen}}$ algorithm connects every pair of adjacent segments by connecting the rightmost node of the left segment with the leftmost node of the right segment. This yields a connected topology provided that the corresponding unit disk graph is also connected. Note that with this construction, the hubs may have a comparatively high transmission range (smaller than one unit, though). However, the interference range of regular nodes is restricted to their corresponding intervals. This is due to the fact that regular nodes are connected to their nearest hub only, which determines their transmission ranges.

Separate analysis of the interference caused by hubs and regular nodes leads to the following theorem:

Theorem 5.4: The resulting topology constructed by the $\mathcal{A}_{\text{gen}}$ algorithm from a given graph $G = (V, E)$ yields interference $O(\sqrt{\Delta})$.

3) Approximation Algorithm: The $\mathcal{A}_{\text{gen}}$ algorithm discussed in the previous subsection achieves interference in $O(\sqrt{\Delta})$ for any network instance. This subsection in contrast introduces an algorithm that approximates the *optimum solution* for the given network instance. Particularly, it yields interference at most a factor in $O(\sqrt[4]{\Delta})$ times the interference value resulting from an interference-minimal connectivity-preserving topology.

The $\mathcal{A}_{\text{gen}}$ algorithm is in a sense designed for the *worst case*. This is best displayed with an instance where the distances between consecutive nodes are identical. Connecting these nodes linearly, that is connecting each node to its nearest neighbor in each direction, yields constant interference. The $\mathcal{A}_{\text{gen}}$ algorithm however constructs a topology resulting in $O(\sqrt{\Delta})$ interference since a hub connects to one half of the nodes in its corresponding interval for this instance and an interval contains $\lceil\sqrt{\Delta}\rceil$ nodes. Based on this observation, we introduce the $\mathcal{A}_{\text{apx}}$ algorithm, a hybrid algorithm which detects high interference instances and applies $\mathcal{A}_{\text{gen}}$ or otherwise connects the nodes linearly.

In the following, we first present a suitable criterion to identify "high-interference" instances. Given a network graph $G = (V, E)$ in the highway model, let the graph $G_{\text{lin}} = (V, E_{\text{lin}})$ denote the graph where all nodes in V are linearly connected. For the considered instance to result in high interference at a node v in G_{lin}, many nodes are required to cover v with their corresponding disks. However, with increasing distance to v, these nodes require to have increasing distances to their nearest neighbors in the opposite direction of v in order to interfere with the latter. This leads to an exponential characteristic of these nodes since the edges in E_{lin} accounting for the interference at v form a possibly fragmented exponential node chain. Consequently, the *critical nodes* of v are defined as follows:

Definition 5.2: Given a linearly connected graph $G_{\text{lin}} = (V, E_{\text{lin}})$, the critical node set of a node v is defined as

$$C_v = \{u | u \neq v, |uw| \geq |vu|, \{u, w\} \in E_{\text{lin}}\}.$$

In other words, the critical nodes of a node v are those nodes interfering with v if the graph G is connected linearly. Based on the results from Section V-C1, we are able to lower-bound the interference of a minimum-interference topology of G as follows.

Lemma 5.5: Given a graph $G = (V, E)$, let $\gamma = \max_{v \in V} |C_v|$ be the maximum number of critical nodes over all network nodes. A minimum-interference topology for G yields interference in $\Omega(\sqrt{\gamma})$.

The $\mathcal{A}_{\text{apx}}$ algorithm makes use of Lemma 5.5 in order to decide whether the existing instance inherently exhibits high interference. In particular, the $\mathcal{A}_{\text{apx}}$ algorithm works as follows: $\mathcal{A}_{\text{apx}}$ first computes γ. If $\gamma > \sqrt{\Delta}$, $\mathcal{A}_{\text{gen}}$ is applied to the graph. Otherwise, if $\gamma \leq \sqrt{\Delta}$, $\mathcal{A}_{\text{apx}}$ connects all nodes of the given graph linearly. A straightforward case analysis leads to the following theorem:

Theorem 5.6: Given a graph G, the $\mathcal{A}_{\text{apx}}$ algorithm computes a topology which approximates the optimal interference of G up to a factor in $O(\sqrt[4]{\Delta})$.

D. Concluding Remarks

The results presented in this section extend the receiver-centric approach to interference modeling, as studied in the context of sensor networks in Section IV, to the analysis of connectivity in general ad hoc networks. The advantages of this interference model are twofold: On the one hand, this definition corresponds to intuition, owing to its receiver-centricity, particularly modeling interference as an effect occurring at the intended receiver of a message, where collisions actually prevent proper reception. On the other hand, this interference model is robust with respect to addition or removal of single nodes, in contrast to the sender-centric interference model proposed in Section III.

Based on this interference model we show that there exist network instances where, to the best of our knowledge, all currently known topology control algorithms (establishing exclusively symmetric connections) fail to effectively confine interference at a low level if required to maintain network connectivity. Led by the observation that already one-dimensional networks exhibit the main complexity of finding low-interference connectivity-preserving topologies, we then focus on the so-called highway model. Starting out to study the special case of the exponential node chain, we finally obtain an algorithm that is guaranteed to always compute a $\sqrt[4]{\Delta}$-approximation of the optimal connectivity-preserving topology in the highway model in general.

In [40] our approach was advanced, leading to a structure with interference in $O(\sqrt{\Delta})$ in two-dimensional node distributions without restriction to the highway model. Their results rely on computational geometric tools such as local neighbor graphs, ϵ-nets, and quad-tree decomposition. Moreover, there exists a simpler randomized algorithm computing low-interference topologies for such networks. This algorithm chooses each node with probability $\sqrt{\log n}/\sqrt{n}$ as a hub. The hubs are connected using a minimum spanning tree to form a backbone network; all other nodes are then connected to their nearest hubs. It can be shown that the interference of the resulting topology is in $O(\sqrt{n \log n})$ with high probability.

VI. Conclusion

This paper presents various approaches to explicit modeling of interference in wireless ad hoc and sensor networks. Different sender-centric as well as receiver-centric models are compared and their properties and complexities are analyzed algorithmically by means of constructive lower and upper bounds.

Finally, it is to be emphasized that this paper presents various *approaches* to the task of reducing interference in ad hoc and sensor networks, particularly focusing on interference models in graph representations of networks. Many questions remain yet unanswered. In particular, the consequences and effects of our theoretical models and analytical studies with respect to practical networks is predictable with difficulty only. We consider our work to be a first step towards understanding the complex interplay between interference and energy efficiency in wireless ad hoc and sensor networks.

References

[1] M. Burkhart, P. von Rickenbach, R. Wattenhofer, and A. Zollinger, "Does topology control reduce interference?," in *Proc. ACM MobiHoc*, 2004, pp. 9–19.

[2] M. Fussen, R. Wattenhofer, and A. Zollinger, "Interference arises at the receiver," presented at the Int. Conf. Wireless Networks, Communications, and Mobile Computing (WirelessCom 2005), Maui, HI, Jun. 2005.

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

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

[5] L. Hu, "Topology control for multihop packet radio networks," *IEEE Trans. Commun.*, vol. 41, no. 10, pp. 1474–1481, Oct. 1993.

[6] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," in *Proc. IEEE INFOCOM*, 2000, vol. 2, pp. 404–413.

[7] S. Borbash and E. Jennings, "Distributed topology control algorithm for multihop wireless networks," in *Proc. 2002 Int. Joint Conf. Neural Networks (IJCNN)*, May 2002, vol. 1, pp. 355–360.

[8] V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, pp. 1333–1344, Aug. 1999.

[9] E. Kranakis, H. Singh, and J. Urrutia, "Compass routing on geometric networks," in *Proc. 11th Canadian Conf. Computational Geometry*, Vancouver, Canada, Aug. 1999, pp. 51–54 [Online]. Available: urlciteseer.nj.nec.com/kranakis99compass.html

[10] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," in *Proc. 3rd DIAL-M*, 1999, pp. 48–55 [Online]. Available: urlciteseer.nj.nec.com/bose00routing.html

[11] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric routing: of theory and practice," presented at the 22nd ACM Symp. Principles of Distributed Computing (PODC) Boston, MA, 2003.

[12] A. Zollinger, "Networking unleashed: Geographic routing and topology control in Ad Hoc and sensor networks," Ph.D. dissertation, ETH, Zurich, Switzerland, 2005, diss. ETH 16025..

[13] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang, "Distributed topology control for power efficient operation in multihop wireless ad hoc networks," in *Proc. IEEE INFOCOM*, 2001, pp. 1388–1397.

[14] X.-Y. Li, G. Calinescu, and P.-J. Wan, "Distributed construction of planar spanner and routing for ad hoc wireless networks," in *Proc. IEEE INFOCOM*, 2002, vol. 3, pp. 1268–1277.

[15] N. Li, C.-J. Hou, and L. Sha, "Design and analysis of an MST-based topology control algorithm," in *Proc. IEEE INFOCOM*, 2003, vol. 3, pp. 1702–1712.

[16] Y. Wang and X.-Y. Li, "Localized construction of bounded degree planar spanner," presented at the DIALM-POMC Joint Workshop on Foundations of Mobile Computing, San Diego, CA, 2003.

[17] X.-Y. Li, W.-Z. Song, and W. Wang, "A unified energy efficient topology for unicast and broadcast," presented at the ACM MOBICOM, Cologne, Germany, 2005.

[18] R. Wattenhofer and A. Zollinger, "XTC: A practical topology control algorithm for ad hoc networks," presented at the 4th Int. Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN), Santa Fe, NM, Apr. 2004.

[19] D. Blough, M. Leoncini, G. Resta, and P. Santi, "The lit K-neigh protocol for symmetric topology control in ad hoc networks," presented at the 4th ACM MobiHoc, Annapolis, MD, 2003.

[20] J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu, "Discrete mobile centers," in *Proc. 17th Annu. ACM Symp. Computational Geometry (SCG)*, 2001, pp. 188–196.

[21] J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu, "Geometric spanner for routing in mobile networks," in *Proc. 2nd ACM MobiHoc*, Long Beach, CA, 2001, pp. 45–55.

[22] L. Jia, R. Rajaraman, and R. Suel, "An efficient distributed algorithm for constructing small dominating sets," in *Proc. 20th ACM Symp. Principles of Distributed Computing (PODC)*, 2001, pp. 33–42.

[23] K. Alzoubi, P.-J. Wan, and O. Frieder, "Message-optimal connected dominating sets in mobile ad hoc networks," presented at the 3rd ACM MobiHoc, Lausanne, Switzerland, 2002.

[24] H. Huang, A. W. Richa, and M. Segal, "Approximation algorithms for the mobile piercing set problem with applications to clustering in ad hoc networks," in *Proc. 6th DIAL-M*, 2002, pp. 52–61.

[25] Y. Wang and X.-Y. Li, "Geometric spanners for wireless ad hoc networks," presented at the 22nd Int. Conf. Distributed Computing Systems (ICDCS), Vienna, Austria, Jul. 2002.

[26] F. Kuhn and R. Wattenhofer, "Constant-time distributed dominating set approximation," in *Proc. 22nd ACM Symp. Principles of Distributed Computing (PODC)*, Boston, MA, 2003, pp. 25–32.

[27] F. Kuhn, T. Moscibroda, T. Nieberg, and R. Wattenhofer, "Local approximation schemes for ad hoc and sensor networks," presented at the 3rd ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC), Cologne, Germany, Sep. 2005.

[28] V. Kawadia and P. Kumar, "Power control and clustering in ad hoc networks," presented at the 22nd IEEE INFOCOM, San Francisco, CA, 2003.

[29] P. Santi, *Topology Control in Wireless Ad Hoc and Sensor Networks*. New York: Wiley, 2005.

[30] F. Meyer auf der Heide, C. Schindelhauer, K. Volbert, and M. Grünewald, "Energy, congestion and dilation in radio networks," presented at the 14th ACM Symp. Parallel Algorithms and Architectures (SPAA'02), Winnipeg, Manitoba, Canada, Aug. 2002.

[31] D. Estrin *et al., Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers*. Washington, DC: National Academies Press, 2001.

[32] I. Akyildiz, W. Su, Y. Sanakarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw. J.*, vol. 38, no. 4, pp. 393–422, 2002.

[33] C.-Y. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proc. IEEE*, vol. 91, no. 8, pp. 1247–1256, Aug. 2003.

[34] A. Goel and D. Estrin, "Simultaneous optimization of concave costs: Single sink aggregation or single source buy-at-bulk," presented at the 14th ACM-SIAM Symp. Discrete Algorithms (SODA), Baltimore, MD, Jan. 2003.

[35] J. Chou, D. Petrovic, and K. Ramchandran, "A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks," presented at the 22nd IEEE INFOCOM, San Francisco, CA, 2003.

[36] P. von Rickenbach and R. Wattenhofer, "Gathering correlated data in sensor networks," presented at the ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC), Philadelphia, PA, Oct. 2004.

[37] P. von Rickenbach, S. Schmid, R. Wattenhofer, and A. Zollinger, "A robust interference model for wireless ad hoc networks," presented at the 5th Int. Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN), Denver, CO, Apr. 2005.

[38] K. Moaveni-Nejad and X.-Y. Li, "Low-interference topology control for wireless ad hoc networks," presented at the 2nd IEEE Communications Society Conf. Sensor and Ad Hoc Communications and Networks (SECON), Santa Clara, CA, 2005.

[39] T. Moscibroda and R. Wattenhofer, "Minimizing interference in ad hoc and sensor networks," in *Proc. DIAL-M*, Sep. 2005, pp. 24–33.

[40] M. Halldórsson and T. Tokuyama, "Minimizing interference of a wireless ad hoc network in a plane," presented at the Int. Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS), Venice, Italy, Jul. 2006.
[41] T. Moscibroda and R. Wattenhofer, "The complexity of connectivity in wireless networks," presented at the 25th IEEE INFOCOM, Barcelona, Spain, Apr. 2006.
[42] T. Moscibroda, R. Wattenhofer, and A. Zollinger, "Topology control meets sinr: The scheduling complexity of arbitrary topologies," presented at the 7th ACM MobiHoc, Florence, Italy, May 2006.
[43] R. Prakash, "Unidirectional links prove costly in wireless ad hoc networks," presented at the 3rd DIAL-M, Seattle, WA, 1999.
[44] D. Peleg, *Distributed Computing, A Locality-Sensitive Approach.*. Philadelphia, PA: Soc. Ind. Appl. Math. (SIAM), 2000.

Pascal von Rickenbach received the M.Sc. degree in computer science (Dipl. Informatik-Ing. ETH) from the Swiss Federal Institute of Technology (ETH), Zurich, Switzerland. in 2004. In the same year, he joined the Distributed Computing Group of Prof. Roger Wattenhofer at ETH Zurich as a Ph.D. student and research assistant.

Roger Wattenhofer received the Doctorate in computer science from ETH Zurich, Switzerland, in 1998.

From 1999 to 2001, he was in the U.S., first at Brown University in Providence, RI, then at Microsoft Research in Redmond, WA. In 2002, he returned to ETH, first as an Assistant Professor in the Computer Science Department, and since July 2004 as an Associate Professor in the Information Technology and Electrical Engineering Department. His research interests include a variety of algorithmic and systems aspects in networking and distributed computing, in particular, peer-to-peer computing and ad hoc and sensor networking.

Aaron Zollinger (M'05) received the M.Sc. degree in computer science (Dipl. Informatik-Ing. ETH) from the Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, in 2001. In the same year, he joined the Distributed Computing Group of Prof. Roger Wattenhofer at ETH Zurich as a Ph.D. student and research assistant. In 2005, he received the Ph.D. degree for his work on geographic routing and topology control in wireless ad hoc and sensor networks.

He is currently with the Department of Electrical Engineering and Computer Science, University of California, Berkeley.