Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Diploma Thesis

Analysis of Interference in Ad-Hoc Networks

Martin Burkhart

Prof. Roger Wattenhofer
Aaron Zollinger

Distributed Computing Group
Institute for Pervasive Computing
ETH Zürich

8^{th} April - 7^{th} August 2003

Contents

List of Figures

Abstract

Despite the focused attention to Mobile Ad-Hoc Networks (MANETs) in the past years, models of interference on the abstraction level of graphs are poorly studied. It is often argued that assuring low degree in topologies is sufficient to minimize interference.

In this work, several models of interference are defined. A classification of models is developed and relations among different models are investigated. One of the main differences between models is whether they focus on outgoing (I_{out}) or incoming (I_{in}) interference. For I_{out} a greedy algorithm (GLIT) is presented that constructs an interference optimal spanning tree. GLIT can be adapted to construct an interference optimal t-spanner (GLIS).

Based on these models, the maximum degree of a network turns out to be merely a lower bound for interference and thus the need for topologies that minimize actual interference instead of node degree becomes obvious. Despite the fact that reducing interference is the main reason for doing topology control at all, widely used topologies as the Minimum Spanning Tree, Relative Neighborhood Graph, Gabriel Graph, Delaunay Triangulation or Yao graph are shown to be prone to bad interference properties, while the introduced algorithms GLIT and GLIS are interference optimal for I_{out} and provide good heuristics for other measures.

Moreover, the MINIMUM INTERFERENCE BROADCAST (MIB) problem and its connected variant (CMIB) are defined. Solutions to CMIB can be used as interference optimal virtual backbones in ad-hoc networks. Although MIB and CMIB are shown to be NP-complete, a constant approximation of both is given. Changing the interference model from I_{in} to I_{out} makes MIB and CMIB optimally solvable in polynomial time, which is a very intriguing result concerning the complexity of interference measures.

Acknowledgements

I am profoundly grateful to my advisors Prof. Roger Wattenhofer and Aaron Zollinger for their support and many fruitful discussions. I was able to significantly influence the direction of this thesis, which was challenging at times but also a great motivation throughout the project. It was a pleasure to work in the *Distributed Computing Group*, as the ambiance in the group is very friendly and communicative.

Finally, I express my deepest gratitude to my parents and my beloved fiancée for their unceasing moral support through the entire course of my studies.

Chapter 1

Introduction

> ***Interference**: A coherent emission having a relatively narrow spectral content, e.g., a radio emission from another transmitter at approximately the same frequency, or having a harmonic frequency approximately the same as another emission of interest to a given recipient, and which impedes reception of the desired signal by the intended recipient.*
>
> Glossary of Telecommunication Terms - Federal Standard 1037C

Much research has been done in the area of mobile ad-hoc networks (MANETs) in the past years. This attention is mainly due to the tremendous progress that wireless technologies are undergoing at the moment. MANETs consist of autonomous mobile hosts that are equipped with the ability to communicate over wireless links.

Devices like mobile phones, PDAs, notebooks or mobile sensors constitute the hardware that enables completely new applications and are penetrating everyday life more and more as they are getting cheaper and perform better every day. At a steady pace society is approaching a realization of Mark Weiser's vision [Wei91] of a ubiquitous computing environment. Computing power will once be invisible and latently present as electricity is nowadays.

MANETs constitute the spine of such future communication infrastructure. They represent the paradigm shift in distributed computing from centralized client-server models towards decentralized peer to peer (P2P) and ad-hoc networks. Future networks will be large, spontaneously formed and unmanageable. Thus they need to organize themselves and must not be dependent of central entities.

As mobile hosts should be autonomous, they suffer from having limited energy resources. An easy way to conserve energy is to restrict transmission power and thus the range of coverage of hosts. This gives rise to the issue of multi-hop routing. A particular host might only see some of its immediate neighbors due to restricted transmission ranges. Anyhow the host is able to reach distant hosts all across the network if intermediate hosts relay messages.

The major part of recent research in the area of MANETs has been dedicated to the fundamental issues of topology control and routing. Topology control aims at constructing sparse network topologies that ensure desirable properties, such as short paths, low power consumption, planarity, distributed construction algorithms or easy maintenance in case of high host mobility. Routing algorithms then try to actually route information along optimal paths that are theoretically present due to topology control.

The main reason for constructing sparse networks at all is to reduce interference. Nevertheless, up to today, no topology control algorithm has been known that explicitly minimizes interference! The issue of interference has always been beaten down with the superficial argument that having a little number of direct neighbors (low node degree) will be sufficient. Yet a host potentially interferes with many hosts it has no direct connection to. The objective of this work is to thoroughly define and analyze the notion of interference in ad-hoc networks. Algorithms are presented that minimize actual interference instead of node degree.

Interference of electromagnetic waves is usually tackled by various multiplexing techniques, of which the four major ones are:

Space Division Multiplexing (SDM) divides space to gain more channels. One can for example think of directed antennas that divide space into sectors. Within each sector the full bandwidth is available.

Time Division Multiplexing (TDM) makes use of time slots. Each channel is assigned time slots during which it can use the medium exclusively and all others must wait.

Frequency Division Multiplexing (FDM) divides the frequency band into segments and assigns to each channel one of these segments. A widely used example are radio stations that are tuned to a certain frequency.

Code Division Multiplexing (CDM) is a technique which is a little more involved. Streams of data can be modulated using a unique code and are then fed into a channel. At the receiver side of the channel, the data can be regained by applying the same code. Interestingly this can be done by several senders simultaneously without distorting others seriously.

Any subset of these techniques can usually be combined. In this work only one logical channel is considered. If conflicts on one channel are minimized, multiplexing techniques can still be applied to multiply that channel.

1.1 Preliminaries

Mobile ad-hoc networks are usually modelled by graphs. A graph $G = (V, E)$ consists of a set of nodes $V \subset \mathbb{R}^2$ and a set of edges $E \subseteq V^2$. Nodes represent mobile hosts, whereas edges represent links between nodes. Edges can be directed (unidirectional) or undirected (bidirectional). A directed edge $e = (u, v)$ can only be used by u to reach v but not vice versa. Yet v could reach u by some other distinct path. An undirected graph is called *connected* if there is a path between any two nodes in the graph. If there is a path between any two nodes in a directed graph, the graph is called *strongly connected*. In the following, several terms are defined that are used throughout this work:

Definition 1 (t-spanner). *A t-spanner of a finite undirected graph $G = (V, E)$ is a subgraph $G' = (V, E')$ such that for each pair (u, v) of nodes $d_{G'}(u, v) \leq t \cdot d_G(u, v)$, where $d_{G'}(u, v)$ and $d_G(u, v)$ denote the shortest distances between u and v in G' and G respectively.*

Definition 2 (Unit Disk Graph). *Let V be the set of nodes and let E be the set of edges such that $(u, v) \in E$ if and only if the Euclidean distance between u and v is at most 1. Then the Euclidean graph $G = (V, E)$ is called the Unit Disk Graph (UDG) of the nodes in V .*

Definition 3 (Gabriel Graph). *Let V be the set of nodes and E be the set of edges. If for each pair of nodes (u, v) there is an edge in E if and only if there is no other node within the circumcircle of u and v, then $G = (V, E)$ is called the Gabriel Graph (GG) of V.*

Definition 4 (Delaunay Triangulation). *Let V be the set of nodes and E be the set of edges. Let there be edges (u, v), (u, w) and (v, w) in E iff there is no other node within the circumcircle of u, v, and w. The resulting graph $G = (V, E)$ is called the Delaunay Triangulation (DT) of V.*

Definition 5 (NP-complete). *A problem is NP-complete if it is both in NP (verifiable in nondeterministic polynomial time) and NP-hard (any other problem in NP can be translated into this problem).*

If a problem is NP-complete, there is no polynomial time algorithm to solve it, unless P=NP. Examples of NP-complete problems include the traveling salesperson and satisfiability of boolean formulae (SAT) problems. See [GJ79] for an introduction to the theory of complexity.

Let $D(w, r)$ denote the disk centered at node w with radius r and let $|u, v|$ be the Euclidean distance between nodes u and v. Then we define the *coverage* of a directed edge $e = (u, v)$ as the set of nodes that is covered by the disk induced by e:

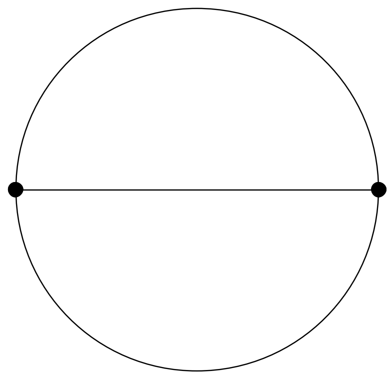

Figure 1.1: Gabriel Graph.

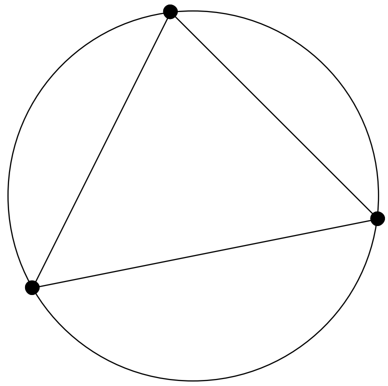

Figure 1.2: Delaunay Triangulation.

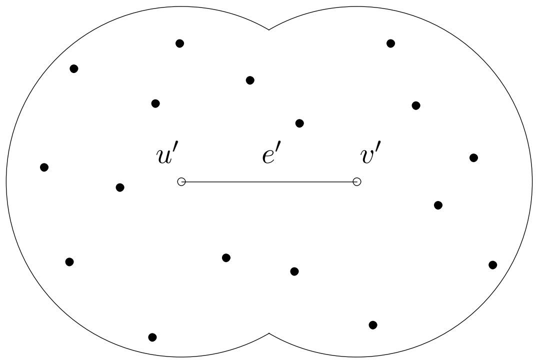

Figure 1.3: Nodes covered by an undirected edge.

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

The coverage of an undirected edge $e' = (u', v')$ is defined accordingly as

$$\begin{aligned} Cov(e') :=& \{w \in V | w \text{ is covered by } D(u', |u', v'|)\} \cup \\ & \{m \in V | m \text{ is covered by } D(v', |v', u'|)\}. \end{aligned}$$

1.2 Related Work

In the literature it is often argued that low or bounded degree goes together with low interference. This conjecture will be discussed more in depth in Chapter 5. Interestingly, there is not much to be found about ad-hoc interference in literature beyond the low degree argument.

In [MSVG02] a model of interference (denoted I_{MS}) between undirected edges is defined. Based on this interference model a time-step routing model and a notion of *congestion* is introduced. It is shown that there are inevitable tradeoffs between congestion, power consumption and dilation. For some vertex sets congestion and energy are even shown to be incompatible.

Assuming a layer architecture for ad-hoc network protocols similar to the seven ISO/OSI-layers for network models, topology control is found at a very low layer. The problem of choosing an appropriate transmission power such that connectivity is assured but interference is minimized must be addressed somewhere between the physical and the data link layer. Traffic analysis by a given routing problem is actually using application level information and can thus not be used to shape network links at a low layer. It is therefore desirable to have a static model of interference that depends solely on a vertex set and does not take a routing problem into account (as congestion does). Usually, no a priori information about the traffic in a network is available, anyway. Still one might be interested in guidelines for building interference optimal network topologies in a general sense.

In I_{MS}, an (undirected) edge $e = (u, v)$ interferes with another edge $e' = (u', v')$ if $Cov(e)$ contains u' or v' (see Figure 1.3). An *interference graph* G_{int} is defined for a graph G. Its nodes are the edges of G and there is an edge between two nodes if the corresponding edges interfere. The *interference number* of a communication link is defined as the number of incoming edges of this link in G_{int}. The interference number of an entire network is the maximum of all its link interference numbers.

The paper comprises an interesting result concerning I_{MS} and an algorithm for constructing a t-spanner presented by Arya and Smid [AS94] (denoted AS-spanner). Lemma 1 in [MSVG02] states that the interference number of a graph constructed by the AS-spanner on a node set V is bounded by $O(g(V))$ where $g(V)$ is the *diversity* of the node set V. $g(V)$ is defined as the number of magnitudes of distances between pairs of nodes in V:

$$g(V) := |\{m | \exists u, v \in V : \lfloor \log |u, v| \rfloor = m\}|$$

I_{MS} seems to be a little too strict in some cases. Consider the situation in Figure 1.4. According to the definition, the directed edges e, e' and e'' interfere and therefore u' cannot send a message to v' while u is communicating with v, although physically the two signals would not interfere at the target nodes. Also, node v'' could receive the message from u'' without being disturbed by u sending simultaneously to v. This restriction roots in the assumption that any node receiving a message implicitly sends an ACK to the sender. Clearly the implicit ACK of v would prevent v'' from correctly receiving a message from u''. Sending ACKs again is a matter of higher layer protocols and must not be addressed at physical layers. There exist wide spread protocols such as UDP that do not rely on ACKs at all. Therefore the implicit ACK assumption could as well be dropped and a model of directed edges could be used, leaving more freedom for link choices and thus enabling better interference properties.

In the next section additional approaches to defining an appropriate interference measure are examined and Chapter 6 contains an in-depth discussion of various interference models. The models differ mainly in terms of

- focus on edges or nodes,
- counting outgoing or incoming interference,
- choosing maximum or average node interference.

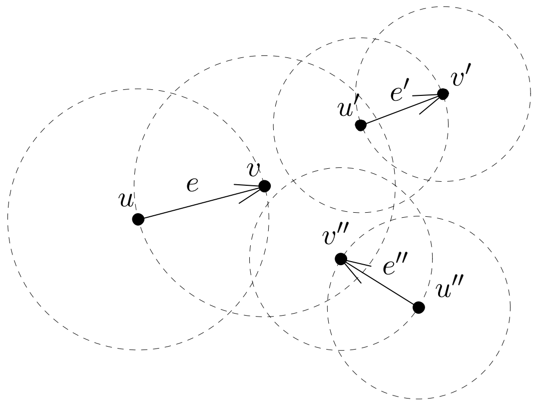

Figure 1.4: Interfering edges that cannot communicate simultaneously.

Chapter 2

The r_{max} Model

This chapter provides basic definitions summarized by the term r_{max} model. First a notion of outgoing and incoming interference of nodes is defined. Based on these node level definitions, interference measures for entire networks are developed. Properties of these measures and relations between them are analyzed considering examples.

2.1 Definitions

Let $G = (V, E)$ denote a graph with node set V and edge set E. We assume that a mobile station can adjust its transmission range to any radius between zero and its maximum range. With $r_{max}(v)$ being the distance between v and its farthest neighbor v' connected by edge $e_{max}(v) = (v, v')$, we define the potential interference area of a node v as the disk of radius $r_{max}(v)$ centered at v, denoted $D(v, r_{max}(v))$. The idea behind this definition is that a node potentially interferes with all nodes covered by $D(v, r_{max}(v))$, that is, no node within the disk can receive a message from any other node while v is sending a message to its farthest neighbor. Obviously the interference of a node v can be defined in two different ways. Either we count the nodes covered by $D(v, r_{max}(v))$, or we count the disks covering v.

Definition 6 (Interference). *The outgoing and incoming interference of a node v are defined as follows:*

$$\begin{aligned} I_{out}(v) &:= \big|Cov\big(e_{max}(v)\big)\big|, \\ I_{in}(v) &:= \big|\big\{u \big| v \in Cov\big(e_{max}(u)\big)\big\}\big|. \end{aligned}$$

Note that we consider the sending node to be covered by its own disk. Thus a sending node not reaching any other nodes still has interference 1 (see also Section 6.5). These node level interference measures are now extended to graph level interference measures. Of course, these global measures somehow need to pool over the nodes of the entire graph. This can be done either by building an average or by taking the maximum.

Note that already at the node level one could distinguish between average and maximum interference. The above definition of I_{out} and I_{in} already does a maximizing step by taking $r_{max}(v)$ as the radius of the surrounding disk. Instead of that one could for instance define the outgoing interference of a node as the average number of covered nodes when communicating with its neighbors. Definition 6 therefore leads to an estimation of the potential or worst case interference of nodes. See Chapter 6 for an in-depth discussion of a wide variety of interference models.

Definition 7 (Average and Maximum Interference). *Let $I(v)$ denote either $I_{out}(v)$ or $I_{in}(v)$. Then the average interference and maximum interference of a graph G are defined as follows:*

$$I_{avg}(G) := \frac{\sum_{v \in V} I(v)}{|V|},$$
$$I_{max}(G) := \max_{v \in V} I(v).$$

Considering I_{out}, I_{in} and both average and maximum pooling this yields four different interference measures for a graph G: $I_{avg,out}$, $I_{avg,in}$, $I_{max,out}$, and $I_{max,in}$. There is no difference between measuring the incoming or outgoing average interference. This is because every occurrence of a node covering is counted exactly once, either at the node who is the center of the covering disk (outgoing case) or at the covered node (incoming case). Therefore three measures remain: I_{avg}, $I_{out} = I_{max,out}$ and $I_{in} = I_{max,in}$.

2.2 Properties

How do these three measures relate? What is the difference between them? In this Section several example node sets are examined with respect to the previously defined interference measures.

A graph with no connected nodes obviously has interference ≤ 1 for all measures. The maximum interference can be observed in complete graphs where every node has connections to the entire node set. Interference than is as high as n for all measures, where $n = |V|$. Therefore

$$0 \leq I_{avg} \leq I_{out/in} \leq n$$

holds. For connected graphs the lower bound can be raised. For the simplest connected graph, a chain of nodes, $I_{out} = I_{in} = 3$ (if $n \geq 3$), because all the inner nodes communicate with their left and their right neighbors. I_{avg} is a little smaller than 3, as the start and end node only interfere with one neighbor: $I_{avg} = 3 - \frac{2}{n}$.

Consider the node set shown in Figure 2.1. All nodes are placed along the border of a circle with a node u in the center of the circle. Topologies as

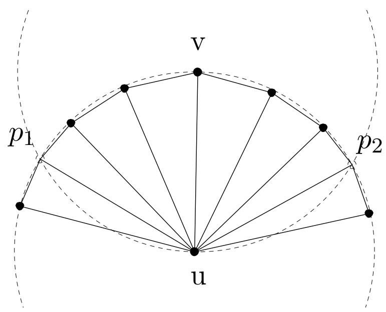

Figure 2.1: Gabriel Graph of nodes on a circle.

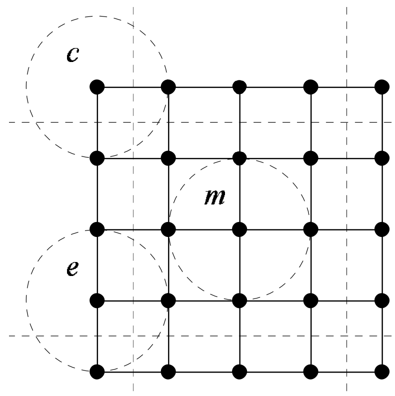

Figure 2.2: Connected grid.

the Gabriel Graph or the Delaunay Triangulation will connect all the nodes on the circle in a (closed) chain and each node directly with the center node. $I_{out} = n$, for node u interferes with all the nodes on the circle. Also $I_{in} = n$, as all nodes on the circle wish to communicate with u. For computing the average interference, we take a look at a node v on the circle. When sending a message to u, it interferes approximately with $\frac{1}{3}$ of the nodes on the circle, as the angle enclosed by (u, p_1) and (u, p_2) is $\frac{2\pi}{3}$. Taking into account node u, which interferes with all nodes, this leads to $I_{avg} \approx \frac{n}{3} + 1$.

In a grid (Figure 2.2) we have three different areas. The first area is the corners, where nodes have two neighbors (node c). Along the edges, nodes have three neighbors (node e) and all nodes inside the grid have four neighbors (node m). Due to the inner nodes, $I_{out} = I_{in} = 5$. Summing up all nodes according to their local interference leads to $I_{avg} \approx 5 - \frac{4}{\sqrt{n}}$.

Intuitively, bounded degree seems to be a good property when trying to get constant interference. An example of a bounded degree topology is the sparse or symmetric Yao graph [GLSV02]. The sparse Yao graph has in- and out-degree of at most k[1] (thus degree $2k$ in total) whereas the symmetric Yao graph has total degree of at most k. Unfortunately, the sparse (and symmetric) Yao graph can exhibit bad interference properties. Imagine a small sector originating at a node on the circle in Figure 2.1. This sector might just miss the center node by a short distance. The closest node within the sector therefore lies somewhere on the opposite side of the circle. If the opposite node symmetrically misses the center node, an edge between the two circle nodes is added to the graph (see also Figure 5.5). This edge causes both nodes to interfere with almost all other nodes when sending messages over it. Therefore a node on the circle is potentially interfered by almost all nodes on the circle, pushing the upper bound for I_{in} and I_{avg} as high as

[1] k is a parameter that specifies the number of sectors around each node.

n. $I_{out} = n$, as node u covers all nodes on the circle. Note that a simple chain of nodes from u to one of the nodes on the circle could be used to connect u to the rest of the nodes with constant interference while GG and DT would still cause interference to be $O(n)$. Thus GG and DT are not even competitive in this case. This intriguing insight will be discussed more thoroughly in Section 5.

In the examples seen so far either all interference measures were constant or all measures were $O(n)$. Anyway, as it turns out, I_{out} seems to be a more fragile measure than I_{in} or I_{avg}. Consider the cluster of nodes in Figure 2.3 with its leftmost node h. Say the cluster has constant interference. I_{out} can be pushed up to n by simply adding a new node h'. As h interferes with the entire cluster when communicating with h', the constant outgoing interference gets lost. I_{avg} and I_{in} are just increased by 1.

While I_{out} may be arbitrarily high compared to I_{in}, it can be shown that

$$I_{in} - 1 \leq 5 \cdot (I_{out} - 1)$$

always holds. The equation can be simplified to

$$I_{in} \leq 5 \cdot I_{out} \tag{2.1}$$

if self-interference is not counted (see Section 6.5).

This result is obtained by trying to cover a node with as much incoming interference as possible without increasing I_{out}. Figure 2.4 shows such a setting. Node I is the node we try to cover with incoming interference from nodes P_1, P_2, $\ldots$, P_i. Placing P_1 reduces the possibilities for placing P_2. We want to increase $I_{in}(I)$ without increasing $I_{out}(P_1)$, therefore P_2 must be placed outside the disk centered at P_1 and going through I. Additionally we want P_2 not to interfere with P_1, keeping $I_{out} = 2$ while incrementing P_{in}. This requires P_2 to be placed below the midperpendicular of P_1 and I. As can be seen in the figure, the angle enclosed by (I, P_1) and (I, P_2) is minimized to $\frac{\pi}{3}$ when P_2 is placed exactly at the intersection of the circle and the midperpendicular. Any other placement P_2' would increase the angle and as a consequence reduce the space available for placing P_3', $\ldots$, P_i'. So maximally six nodes can be placed before I_{out} is increased by at least 1. Note that if we define nodes on the circle to be within the disk, we need to marginally displace P_2 away from the intersection point, say by a distance ε. If we do this, the maximum number of nodes is even reduced to five.

Concluding this section of examples, we have seen that in a grid topology interference is constant while in other topologies interference can be as bad as $O(n)$ for all measures. Constant interference certainly is a desirable property for ad-hoc network topologies in general.

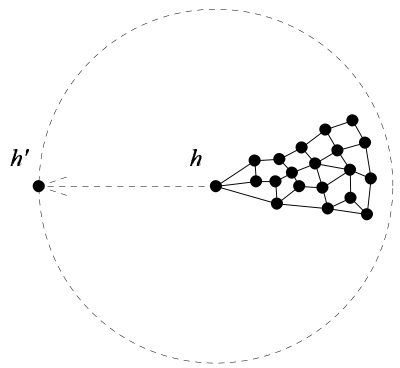

Figure 2.3: A single node can destroy a low I_{out}.

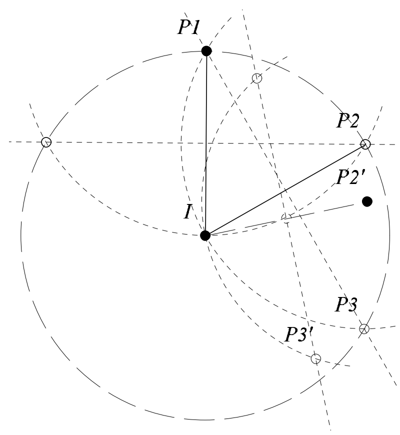

Figure 2.4: Illustration for $I_{in} \leq 5 \cdot I_{out}$.

Chapter 3

A Greedy Approach

In this chapter we present a greedy algorithm that is capable of finding an interference optimal connected topology for some measures. By adapting the stopping condition of the algorithm, other topology properties can be achieved. A dynamic greedy criterion is then defined that is more suitable for minimizing incoming instead of outgoing interference.

3.1 Greedy Low Interference Tree

Prior to minimizing interference on a set of nodes one must define certain conditions that must be met by the desired topology. Clearly there is a trivial zero interference topology: A graph where no node is sending. Reasonable properties of graphs include

- dominating set property,
- (strong) connectivity,
- Euclidean, power, or hop spanner property,
- planarity,
- bounded degree.

In addition to requiring topology properties it must be guaranteed that nodes can adapt transmission ranges as proposed in [WLBW01]. For instance in a unit disk graph model, nodes might be able to lessen transmission energy and thus their range of coverage in order to save energy and local resources. It makes no sense to talk for example about a minimum interference connected topology if all nodes send at a fixed radius, as there is no room for optimization. A lower range of coverage imposes less interference to other nodes and thus gives rise to an optimization task: meeting conditions with as little interference as possible. In the following we require nodes to be spanned by a tree and present a simple greedy algorithm that computes the minimum interference tree.

Definition 8 (Minimum Interference Tree). *The* Minimum Interference Tree *(MIT) for a given set of nodes V and an interference measure I is a tree on V that minimizes I.*

Definition 9 (c-interfering). *A graph $G = (V, E)$ is called* c-interfering *according to an interference measure I if*

$$I(G) \leq c \cdot I(MIT(V))$$

As it turns out, Algorithm 1 (called GLIT) computes an MIT if I is an outgoing interference measure. Edges can be directed or undirected. Note that although the name GLIT (Greedy Low Interference Tree) implies the construction of a tree, this is only true for undirected edges. If directed edges are used, GLIT builds an interference optimal strongly connected topology.

Algorithm 1 Greedy Low Interference Tree (GLIT)

Input: V, a set of nodes

sort all (directed) edges (u, v) according to $|Cov(u, v)|$
$E = \emptyset$
$G = (V, E)$
while edges left **do**
  $e = (u, v)$ (* next edge *)
  if no path $u \to v$ in G **then**
    $E = E \cup e$
  end if
end while

Theorem 1. *The tree constructed by GLIT is 1-interfering according to interference measure I_{out}.*

Proof. The proof is by induction. Let m be the global minimum edge and thus the first edge to be added to the tree. Trivially, m belongs to an MIT. Now let τ be the tree constructed by GLIT so far. Suppose that τ is part of an MIT τ'. Let $e = (u, v)$ denote the next fringe edge chosen by the algorithm to be added to the tree. We will show that $\tau + e$ is contained in some MIT τ''.

If $e \in \tau'$ we are done, simply let $\tau'' = \tau'$. If $e \notin \tau'$, consider a path P from u to v in τ'. Some edge $e' = (u', v')$ along this path P will be a fringe edge of τ. Then $I(e) \leq I(e')$, otherwise e' would have been chosen by the algorithm instead of e. Now let $\tau'' = \tau' - e' + e$. τ'' is still a tree and $I(\tau'') \leq I(\tau')$. This is because e' was replaced by an edge e covering a smaller or equal number of nodes. Thus τ'' is an MIT. $\square$

The time complexity of GLIT is $O(n^3)$ since there are $\binom{n}{2} \in O(n^2)$ edges that must be processed ($O(n)$ for each edge). Sorting only is $O(n^2 \log n)$. Note that there are two versions of GLIT, differing in the same way as Prim's and Kruskal's algorithms for minimum spanning tree. The first version adds fringe edges to an existing tree (like Prim's algorithm) whereas the other version adds the edge that minimizes the greedy criterion globally (like Kruskal's algorithm). The above proof is for the tree growing version but can easily be adapted to fit for both. Note that if directed edges are used, $NNG \subseteq GLIT$ where NNG denotes the Nearest Neighbor Graph[1]. Of course, GLIT could alternatively be specified as a shrinking algorithm, starting with a fully connected graph and continually removing edges with high coverage as long as the desired property is maintained.

Theorem 1 holds for many other interference measures (such as I_{avg}) that consider interference on nodes (see Section 6 for an overview). This is because nodes are static and thus the number of covered nodes by a disk can be computed once for all, no matter what edges are added to the topology in the future.

Unfortunately, the shown type of greedy algorithm does not work for almost any type of incoming interference measures and measures targeting edges. Whenever an edge e is added to the intermediate tree, $I_{out}(e)$ can be computed and will never change while the algorithm is progressing. When dealing with some kind of I_{in}, the final interference of an edge is not known until the algorithm has stopped. I_{in} always depends on other edges that might be added to the tree only much later. The same is true for measures that count the coverage of edges.

3.2 The Geometric Dominating Set Problem

The problem of covering a set of nodes V using a minimum number of disks with radius r, centered at nodes in V, is known as the GEOMETRIC DOMINATING SET problem (GDS) and is NP-complete [MIH81]. Thus each node has the choice to either send with fixed radius r or not to send at all (radius zero). Minimizing the covering for I_{out} instead of the number of disks turns out to be easily solvable. If we want to know whether V is coverable with $I_{out} \leq k$ we simply add all edges that cover at most k nodes. If and only if this covers all nodes then the answer is 'yes'. As there are $2\binom{n}{2} \in O(n^2)$ directed edges, this is a polynomial algorithm and thus GDS for I_{out} is in P. The same procedure works for connected GDS which is GDS with the additional condition that nodes in the dominating set must be connected.

In Section 4 we show that minimizing GDS and connected GDS for I_{in} instead of I_{out} is NP-complete, which is a very intriguing result.

[1] In a NNG, each node has a directed edge to its nearest neighbor.

3.3 Low Interference Spanners

Algorithm 1 can easily be adapted to fit other needs. For instance one could construct an algorithm for a *Greedy Low Interference t-Spanner (GLIS)* by simply changing lines 6-8 as follows:

if shortestPath($u \to v$) in $G > t|u,v|$ **then**
  $E = E \cup e$
end if

This clearly constructs a t-spanner, as it adds edges until for all nodes u and v there is a path $u \to v$ of length at most t. GLIS resembles the greedy spanner introduced by Das and Narasimhan [DN94], denoted GS here. GS sorts edges according to their length whereas GLIS sorts edges according to coverage of nodes. Das and Narasimhan use graph clustering techniques to reduce the cost of *shortestPath*-queries such that the overall time complexity of GS becomes $O(n \log^2 n)$. Similar techniques can be used for GLIS. But what effect do the different sorting criteria have on the resulting topologies?

Consider a setting as in Figure 3.1. The figure shows for instance recovery teams that have settled down after a hard day of work in a disaster area along a river. The camps (gray disks) are equipped with ad-hoc communication networks and interconnected via some relay stations along the river that might not be placed as close to the river due to rough terrain. Assume that within the camps, network topology is such that constant interference k can be achieved. GS first adds all edges within the camps, as nodes are closest there. Then it will cross the river directly from camp to camp using the dashed edges. Usage of these edges interferes with the entire camp on one side, increasing overall I_{avg} and I_{in}. I_{out} is pushed up to the size c of the camps. GLIS on the other hand prefers the crossing edges outside camps as they have much smaller coverage. Depending on t it might not be necessary to use the direct edges between camps and therefore an increase of interference can be avoided. Table 3.1 summarizes the performance of the two algorithms for the presented scenario. Note that for n nodes and m camps $c \approx \frac{n}{m} \in O(n)$ and thus also $I_{out}^{GS} \in O(n)$.

GLIS produces an interference optimal t-spanner with respect to I_{out} as it uses the least interfering edges to build the spanner. Any other t-spanner's interference can only be equal or worse.

	GLIS	GS
I_{avg}	k	$k+1$
I_{in}	k	$k+1$
I_{out}	k	$k+c$

Table 3.1: Performance of GLIS versus GS

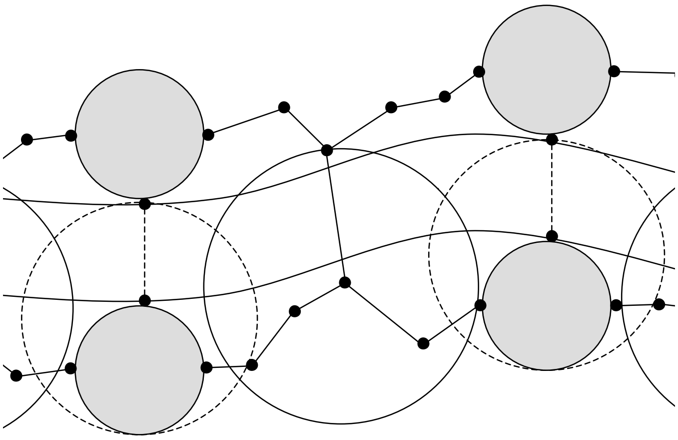

Figure 3.1: Communication structure of camps along a river.

Unfortunately, GLIS is not necessarily a planar embedding. Consider four nodes arranged as in Figure 3.2. Edge BC has already been added by the GLIS algorithm. Assume that nodes A,B and B,D are already connected by paths $\widetilde{AB}$ and $\widetilde{BD}$ of lengths $t \cdot |AB|$ and $t \cdot |BD|$ respectively. The shortest detour from A to D without using the direct edge is via B. Assume that there is a constant $c > 1$ such that $|AB| + |BD| = c \cdot |AD|$. Thus the overall length of the detour from A to D will be

$$|\widetilde{AD}| = |\widetilde{AB}| + |\widetilde{BD}| = t \cdot (|AB| + |BD|) = c \cdot t \cdot |AD|.$$

Accordingly the shortest path from A to D in the graph does not satisfy the t-spanner property and thus edge AD will be added by GLIS, crossing edge BC.

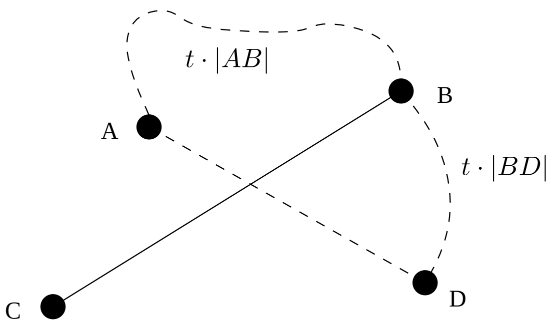

Figure 3.2: GLIS is not planar.

3.4 The Competitive Ratio

As discussed above, $|Cov|$ is not necessarily a useful greedy criterion when going for low I_{in}. We present an example where GLIT does not compute the optimum with respect to I_{in} and propose a new heuristic criterion called *impact* which is better suited to minimize I_{in}.

Consider a set of nodes as shown in Figure 3.3. Two rings of nodes are placed around a center node I. Between each pair of consecutive nodes on the second (outer) ring, there is an additional outer node. The radius of the first (inner) ring is r, while the radius of the second ring is $< 2r$. Outer nodes are placed outside the second ring. In the figure, we see the coverage annotated to some directed edges. Obviously, GLIT will first add the edges with coverage 2, connecting nodes between the two rings. Edges from the first ring to I and from outer nodes onto the second ring both have coverage 3 and thus will be added next. Because edges from the second ring to outer nodes have coverage 4 they are added later. Meanwhile, all nodes on the first ring have been connected to I directly, causing as much as 6 disks to cover I (counting self-interference of I). These inner spokes could have been saved if connections via outer nodes would have been established earlier, which was not possible due to the higher coverage of edges from the second ring to outer nodes. An optimal connected topology would connect at most three first ring nodes directly to I and connect the remaining first ring nodes via links over outer nodes, ending up with an incoming interference of 4.

The question arises whether GLIT can be arbitrarily bad in terms of I_{in}. GLIT might as well be c-interfering, meaning that there is some constant $c > 1$ such that

$$I_{in}(\text{GLIT}) \leq c \cdot I_{in}(\text{MIT}).$$

If self-interference is neglected, the above example shows that c must be at least $\frac{5}{3}$.

3.5 Dynamic Greedy Criterion

To overcome this weakness of GLIT in terms of I_{in}, we present an new greedy criterion called impact which is better suited to minimize incoming interference. Let e be an edge, then the impact of e is defined as follows:

$$impact(e) := |Cov(e)| + \sum_{v \in Cov(e)} I_{in}(v)^2$$

$impact(e)$ not only counts the nodes covered by a potential edge e but also takes existing incoming interference in these covered nodes into account. Note that the term $I_{in}(v)$ in the above definition is dynamic, since incoming interference is dependent on edges that are added by the algorithm. Thus the values of $I_{in}(v)$ need to be updated every time a new edge is added

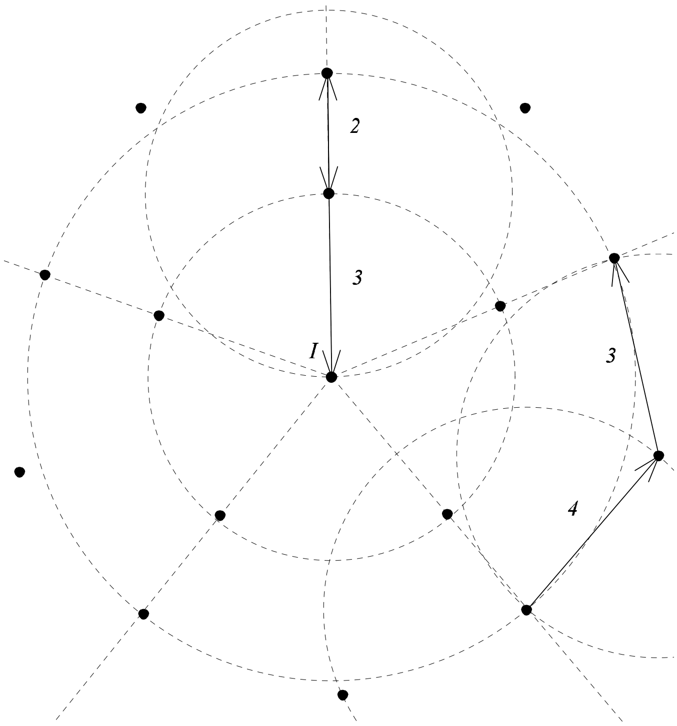

Figure 3.3: A set of nodes where GLIT is not optimal for I_{in}.

to the graph. An algorithm deploying *impact* will do everything to avoid covering nodes that are already highly interfered. In fact, GLIT using the impact criterion (denoted $GLIT_{imp}$) stops adding direct edges to I in the above example as soon as three direct edges are in place and ends up with an optimally connected topology having $I_{in} = 4$.

An already mentioned disadvantage induced by using *impact* is the need for updating the edge table after each step. This is because any new edge increases I_{in} in covered nodes. Yet the overall time complexity of the resulting greedy algorithm will not exceed $O(n^4)$. There are $\binom{n}{2} \in O(n^2)$ edges. For each edge, all n nodes are checked for coverage, which is $O(n^3)$. Thus building the edge table and sorting it is $O(n^3 + n^2 \log n) = O(n^3)$. The resulting tree has $O(n)$ edges, thus the table must be rebuilt $O(n)$ times.

Yet another approach to make GLIT perform better in terms of I_{in} is to append a second phase. In this second phase, nodes with high I_{in} are examined and edges adjacent to these nodes are deleted as long as the graph is still (strongly) connected. In the above example, this removal phase would eliminate some edges that cause high interference in I. Because nodes on the first ring are now connected via the second ring, some first ring nodes can drop their direct connection to I, reducing I_{in} to 4.

Chapter 4

NP-Completeness Results

Although one can think of many different greedy criteria for the algorithm proposed in Chapter 3 or even other types of algorithms, not all interference measures can be tackled as easily as I_{out} or I_{avg}. This chapter defines two problems in the domain of I_{in}: MINIMUM INTERFERENCE BROADCAST (MIB) and its connected variant (CMIB). NP-completeness of both problems is shown. At last, a known algorithm is shown to construct a constant approximation of CMIB.

4.1 Minimum Interference Broadcast

It is known that GEOMETRIC DOMINATING SET (GDS) is NP-complete [MIH81]. In this section we will show that a variation of GDS which minimizes incoming interference also is NP-complete.

Definition 10 (MIB). *Let V be a set of nodes in the two-dimensional space. Furthermore let R be a finite set of radii with $|R| \geq 2$. Then MINIMUM INTERFERENCE BROADCAST (MIB) is the problem of covering all nodes in V with disks $D(v_i, r_j)$ where $v_i \in V$ and $r_j \in R$ such that I_{in} of the resulting graph is minimal.*

Theorem 2. *MIB is NP-complete.*

Proof. We will reduce a restricted version of 3SAT to MIB. ONE-IN-THREE 3SAT is a restriction of 3SAT such that each clause contains exactly one true literal. The problem remains NP-complete even if no clause contains a negated literal [Sch78].

Let $R = \{r, r + \delta\}$, where $\delta \ll r$ is a positive constant. Consider a set of $3k$ nodes along a closed loop of wire as in Figure 4.1. The distance between two consecutive nodes is $s = \frac{r}{\sqrt{2}}$ such that a disk of radius r centered at a corner node of a $s \times s$ square will cover all nodes on the square. Any disk with radius r centered at a node along the wire exactly covers three nodes:

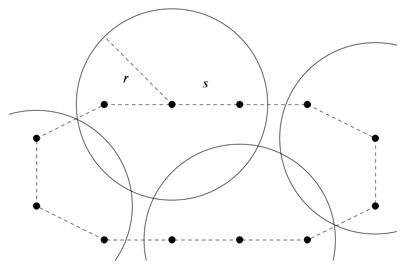

Figure 4.1: Nodes along a looping wire.

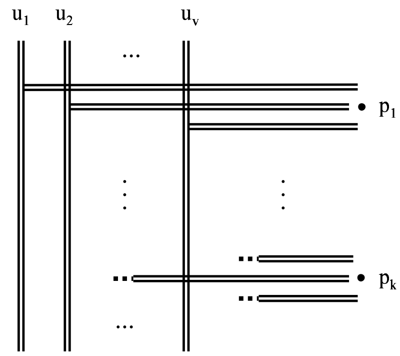

Figure 4.2: Overview of the wire construction.

itself and its two neighbors. By building groups of three consecutive nodes we see that an interference optimal covering must choose either the first, second or third node of each group as a center node of a disk. Hence a wire can be covered with minimal interference 1 in exactly three ways. Any other covering will lead to an interference ≥ 2.

Now take an instance of ONE-IN-THREE 3SAT, a boolean formula B in conjunctive normal form consisting of a set of variables U and a collection C of clauses over U such that each clause $c \in C$ has $|c| = 3$. The question is, whether there is a truth assignment for U so that each clause in C has exactly one true literal.

The construction contains a looped wire w_i for each of the variables $u_i \in U$. Let one of the three interference optimal coverings of w_i correspond to a *true* assignment and the other two to a *false* assignment of u_i. Let T_i denote the set of center nodes in the *true* covering of wire w_i. For each clause $c_j \in C$ we define a clause point p_j. The three wires corresponding to the literals of c_j are brought into close proximity of p_j such that p_j can be covered by its wires 'for free' if any of the wires is in the *true* state. This can be done by arranging the wires around p_j such that for each wire w_i exactly one node in T_i is at distance r from p_j while the other nodes in T_i are further away from p_j. We need not worry about negated literals, because the restricted problem instance containing no negated literals remains NP-complete. Figures 4.2 and 4.3 show the overall construction and a detailed vicinity view of a clause point.

This construction cannot be built without crossing wires. It can be shown that any crossing of two wires can be covered with interference 1 and that if this optimal covering is chosen, both wires will independently maintain their 'values' when passing the crossing. To achieve this, some extra nodes need to be added wherever two wires cross (see Figure 4.4). Let the coordinates of crossing point x be $(0, 0)$. Its right neighbor is at $(1, 0)$ accordingly. Then we must add helper nodes at coordinates $(1, 1)$, $(1, -1)$,

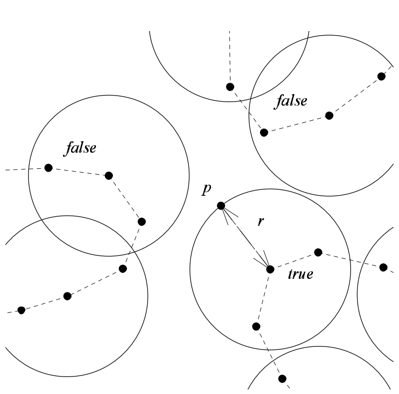

Figure 4.3: The vicinity of a clause point p with the three corresponding wires.

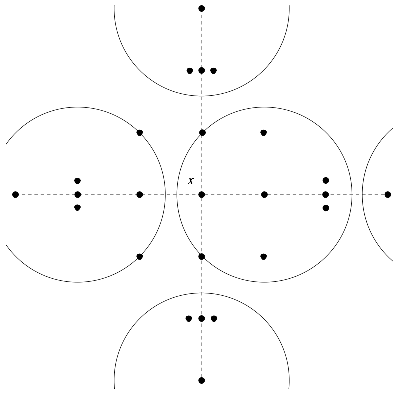

Figure 4.4: Crossing of two wires. Configuration is $[1, 0]$.

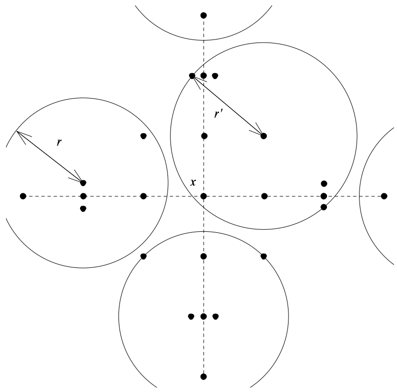

Figure 4.5: Crossing of two wires. Configuration is $[1, 1]$.

$(-1, 1)$ and $(-1, -1)$. Around nodes at distance 2 from x two helper nodes need to be placed at distance δ from the wire.

At a crossing, nine different configurations of wire values can occur. Let $[0, 0]$ denote the case where both wires center a disk at x. $[1, 0]$ means that the vertical wire places a node at x, but the horizontal wire has a phase shift of 1, that is it places a disk at node $(1, 0)$. Any configuration $[i, j]$ for $i, j \in \{-1, 0, 1\}$ is possible. Configuration $[0, 0]$ is optimally covered anyway, since both wires use the same disk at x. Figure 4.4 shows how the $[1, 0]$ case can be solved by simply dropping a disk on the vertical wire. Due to symmetry, this also solves the configurations $[-1, 0]$, $[0, 1]$ and $[0, -1]$.

The remaining cases are those where both wires are out of phase by either -1 or 1. Without loss of generality configuration $[1, 1]$ can be considered (see Figure 4.5). The center of the leftmost disk must be displaced to node $(-1, \delta)$ so that node $(-1, -1)$ will not be covered by 2 disks. Of course, the bottom disk could instead be displaced. The remaining nodes can be covered by a single disk centered at $(1, 1)$. Note that the algorithm must choose the bigger radius $r + \delta$ for this disk, otherwise it would not cover all helper nodes of $(2, 0)$ and $(0, 2)$. This is the only case where radius $r + \delta$ is needed to maintain optimal interference. If both wires keep their state beyond the crossing, the crossing can be optimally covered. It can also be seen that if any wire changes its state over the crossing, the remaining points cannot be covered with interference 1. Eventually, an algorithm seeking the optimal covering must preserve wire states across intersections.

At a clause point p_j, if exactly one literal of the corresponding clause is *true*, p_j is covered by the according wire with interference 1 as in Figure 4.3. If two or three literals are *true*, p_j is covered by two or three distinct wires respectively, increasing the interference at p_j. If none of the literals is true, p_j remains uncovered and an additional disk needs to be placed either at p_j itself or at a wire close to p_j. In either case, this disk entails an interference of 2 in at least one of the wires (the additional disk can be placed on a literal wire or on the clause point itself). Consequently there is a truth assignment over U satisfying all clauses with exactly one true literal per clause if and only if the shown construction can be covered with a minimal interference of 1.

The number of nodes and crossovers is bounded by $O(|U| \cdot |C|)$, where $|U|$ is the number of variables and $|C|$ is the number of clauses. Thus, the node set can be generated in time $O(|U| \cdot |C|)$ which is clearly polynomial in $|U|$ for $|C| \leq \binom{|U|}{3} \in O(|U|^3)$.

We have shown a reduction of ONE-IN-THREE 3SAT to MIB and therefore MIB is NP-hard. Since it is clearly in NP (for solutions are verifiable in polynomial time) it is also NP-complete. $\square$

4.2 Connected Minimum Interference Broadcast

When studying wireless ad-hoc networks, using some kind of virtual backbone structure can be beneficial [DB97, WL99]. Most of the traffic is routed on the backbone and thus protocol overhead can be reduced and fault-tolerance increased. Other advantages include alleviation of the broadcast storm problem [TNCS02], improved network throughput or primitives for broad- and multicast.

We introduce connected MIB (CMIB) as an interference optimal virtual backbone and show its NP-completeness. CMIB is MIB with the additional requirement that the broadcasters (sending nodes) are connected.

Theorem 3. *CMIB is NP-complete.*

Proof. We will reduce planar 3SAT to CMIB. Lichtenstein showed planar 3SAT to be NP-complete in [Lic82]. In order to understand what planar 3SAT exactly is, we recapitulate its definition.

Definition 11 (Planar 3SAT). *Let B be an instance of 3SAT with m clauses c_j and n variables v_i. Then $G(B) = (N, A)$ is a graph defined on B, where*

$$N = \{c_j | 1 \leq j \leq m\} \cup \{v_i | 1 \leq i \leq n\}$$

is the set of nodes and $A = A_1 \cup A_2$ is the set of arcs[1] *with*

$$A_1 = \big\{\{c_i, v_j\} | v_j \in c_i \text{ or } \overline{v_j} \in c_i\big\},$$
$$A_2 = \big\{\{v_j, v_{j+1}\} | 1 \leq j \leq n\big\} \cup \big\{\{v_n, v_1\}\big\}.$$

Planar 3SAT is 3SAT restricted to formulae B such that $G(B)$ is planar.

There is a node in G for each clause and each variable. Variables are connected to the clauses they are part of (A_1). In addition, all variables are interconnected (A_2).

Similar to the proof for MIB we will construct wires that carry truth values of variables. A structure is needed that has exactly two (or few that can be combined) interference optimal solutions for CMIB, allowing us to code *true* and *false* for each variable. Connected topologies have a minimum interference of 3 which complicates finding a useful structure, as the higher optimal interference admits more freedom in choosing broadcasters. Figure 4.6 shows a mesh that has exactly two optimal solutions represented by the two zigzag lines (l_{true} and l_{false}). The distance between two nodes diagonally is r, thus a sending node covers exactly all diagonal neighbors.

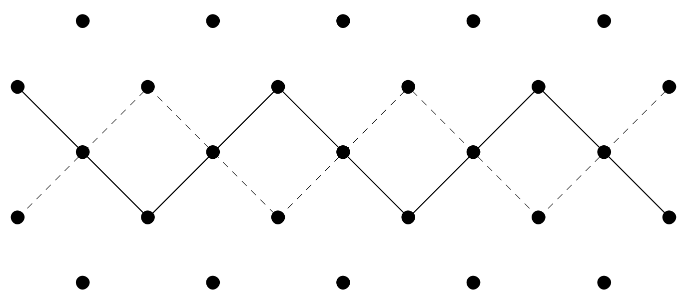

Figure 4.6: Mesh structure with only two optimal solutions for CMIB.

For each variable in B there shall be a wire consisting of the above structure visiting all connected clause points along the arcs in A_1, always returning to the variable node between two clause points. At clause points the three wires are brought into close proximity such that from each wire there is exactly one node within distance r of the clause point. If the occurrence of the variable in the particular clause is positive, a node in l_{true} is put closest. If the occurrence is negated, the wire will be slightly distorted before and after the clause point such that a node in l_{false} will be closest.

[1] The term arc is used here for edges that need not be straight but can be bowed.

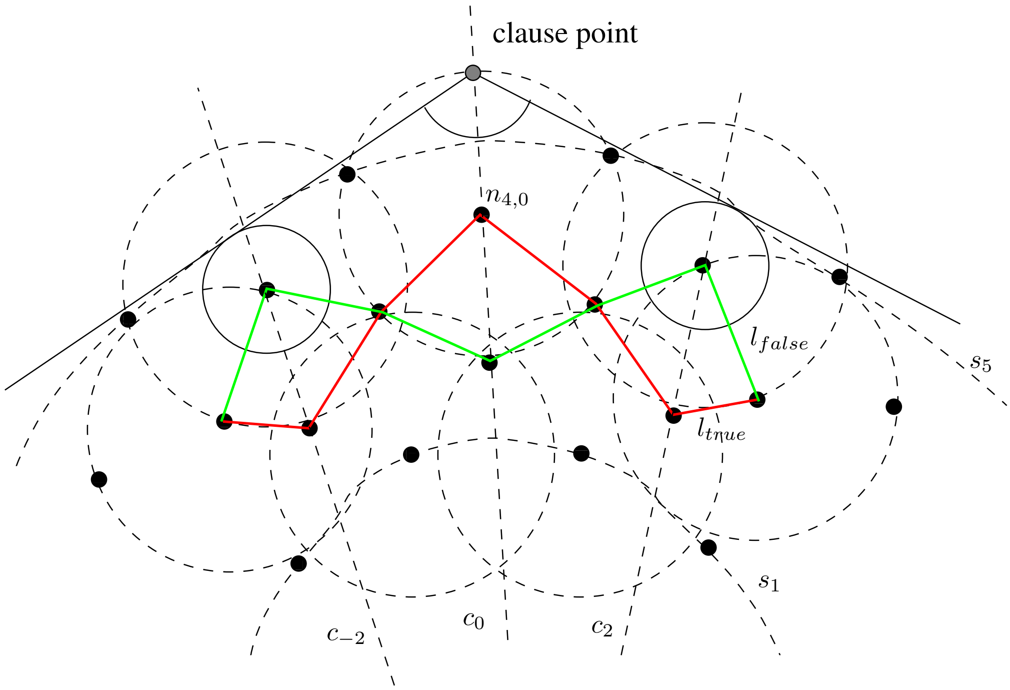

Figure 4.7: Turn of a five-stranded wire at a clause point.

In order to make the entire dominating set connected, the wire fringes must be connected by strands, corresponding to the arcs in A_2. Those connecting strands can be simple lines of consecutive nodes at distance r that are attached to the ends of two wires.

It is not quite trivial to see that the five-stranded wires can be bent tight enough so they will not interfere with each other at clause points. Figure 4.7 illustrates the situation.

As three wires need to visit each clause point, a wire must be turned within an angle of 120°. Of course there are conditions to meet that prevent the turning angle from being arbitrarily small. Let s_i denote strand number i, starting at the inner strand s_1 and c_j column number j where columns are sets of two or three nodes along radial lines as shown in the figure (c_0 being the column containing the clause point). Node $n_{i,j}$ denotes the node on strand s_i and column c_j. Then the following conditions guarantee preservation of wire properties:

(i) Nodes must not interfere with their immediate neighbors on the same strand ($|n_{i,j}, n_{i,j\pm1}| > r$).

(ii) Nodes must not interfere with their immediate neighbors in the same column ($|n_{i,j}, n_{i\pm1,j}| > r$).

(iii) Nodes must reach their diagonal neighbors ($|n_{i,j}, n_{i\pm1,j\pm1}| \leq r$).

Once the conditions are met, the question of where to put the tangential lines that enclose the turning angle arises. $n_{4,0}$ is part of the broadcasting set iff the wire satisfies the literal. In the case where the literal is *false*, nodes $n_{4,\pm2}$ are sending. In order not to interfere with its corresponding nodes (mirrored across the tangential) in a neighboring wire, there must be a distance of at least $\frac{r}{2}$ around $n_{4,\pm2}$, depicted by the small solid disks. Thus if we draw the tangential from the clause point to those solid disks we can be sure that no broadcasting node in one wire can reach a node of another wire, as also the mirrored equivalent to $n_{4,-1}$ can not be reached by $n_{4,-2}$ or $n_{4,0}$.

Note that an algorithm seeking a solution to CMIB could as well consider connected sets that jump from wire to wire in nodes as $n_{4,-1}$, $n_{4,-3}$ or the clause point. Nevertheless a solution with interference 3 admits only connected sets that constitute a single line. If some node in the set has more than two neighbors its interference will be at least 4. Thus leaving a wire w for wire v cuts of the connected line in w as well as in v. Think of the global connected set as a string. Changing wires around a clause point corresponds to cutting the string twice and knitting two of the three fragments together, ending up with two not connected fragments (Note that we don't include edge $\{v_n, v_1\}$ in A_2. If we did, the algorithm could indeed once change wires). Thus changing wires does not produce connected sets with optimum interference and therefore it is no option for the algorithm.

The turning in Figure 4.7 satisfies conditions (i) through (iii) and turns within 118°. A number of example disks of radius r allow the verification of the conditions for some of the nodes. Trigonometrical calculation of the minimum possible turning angle is fairly intricate and omitted in the scope of this work.

If at least one of the literals in each clause point resolves to *true* the corresponding wire will cover the clause point for free. If no literal resolves to true, the clause point must be covered by an additional disk centered at a wire node, inevitably leading to an interference higher than 3 on that wire. Thus it can be concluded that B is satisfiable if and only if the corresponding construction described above has a solution to CMIB with interference 3.

The construction can be built in polynomial time and thus it constitutes a reduction of planar 3SAT to CMIB. So CMIB is NP-hard. Since it is clearly in NP (for solutions are verifiable in polynomial time) it is also NP-complete. □

4.3 Constant Approximation of CMIB

Since CMIB is NP-complete, our focus now lies on finding polynomial time approximations. The CONNECTED DOMINATING SET (CDS, equal to connected GDS) algorithm presented by Alzoubi et al. [AWF02] turns out to be a constant approximation of CMIB (and MIB as well). The algorithm (called AWF here) works as follows: First, a MAXIMAL INDEPENDENT SET (MIS) S is constructed. Nodes in S (called dominators) can be connected by using at most two additional nodes (connectors) between pairs of dominators. The necessary connectors are accumulated in a set C. The resulting graph $U = S \cup C$ is a CDS with size bounded by a constant times the size of an optimal CDS, as has been proven in the paper. The algorithm is fully distributed and runs in linear time and message complexity.

Theorem 4. *AWF constructs a CDS that approximates CMIB within a constant factor.*

Proof. Consider a node $v \notin S$. Let the unit radius be 1. Following the proof of Lemma 1 in [AWF02] and by the standard area argument we can compute the number of potential dominators $d(r)$ within a disk of radius r around v by calculating the maximum number of disks of radius $\frac{1}{2}$ that can be placed within the area of the disk with radius $r + \frac{1}{2}$ around v. This leads to

$$d(r) := \left\lfloor \frac{\pi(r+0.5)^2}{\pi(0.5)^2} \right\rfloor .$$

Thus $d(1) = 9$ dominators can be placed that directly interfere with v. $d(2) = 25$ dominators can be placed within radius 2. Each of these more distant dominators could potentially be connected to any other of the 25 dominators by a path of at most two connectors. Each of these connectors could be placed within the disk of unit radius around v and thus interfere with v. There remain $d(3)-d(2) = 24$ dominators in the ring between radius 2 and 3 that can be connected to any of the radius 2 dominators in a way that at most one of the connectors enters the unit disk around v. This leads to an upper bound of interfering nodes in v.

$$I_{in}(v) \leq 9 + 2\binom{25}{2} + 24 \cdot 25 = 909 \tag{4.1}$$

Apparently one could place less than $d(2)$ dominators within radius 2 to allow the number of dominators in the outer ring between radius 2 and 3 to be increased. A closer look at the formula shows that $2\binom{r_2}{2}+r_2(d(3)-r_2)$ can be simplified to $r_2(d(3)-1)$ and thus is maximized if r_2 is chosen as big as possible, and $r_2 = d(2)$ as in Equation 4.1 actually maximizes interference.

The same argumentation leads to an upper bound of interference in a dominator node $s \in S$. Indeed the maximum interference in s is slightly smaller, as no other dominator can be placed within the disk of unit radius around s.

Thus I_{in} is bounded by a constant on all nodes. Because any solution to CMIB with more than two dominators has at least interference 3, AWF approximates CMIB within a constant factor of at most 303. □

Chapter 5

Low Degree versus Low Interference

It is often argued that in MANETs low or bounded degree topologies are well suited to minimize interference. In this section we will discuss this conjecture by looking at several well known topologies and analyzing their properties in terms of interference. It turns out that many topologies are not competitive in terms of interference while GLIT is optimal for I_{out} and I_{avg}.

5.1 Low Degree Spanners

The quest for low degree spanners was initiated by Dobkin et al. [DFS90]. They posed the problem of finding a *v-degree constrained t-spanner* such that every node has degree v or less and showed that for two-dimensional problems $2 < v \leq 7$. Soares provided a proof for $v \leq 5$ [Soa92] and Salowe showed that $v \leq 4$ [Sal94]. Das and Heffernan concluded the quest by showing that actually $v = 3$, as previously conjectured [DH93]. The major drawback of these solutions is a very high stretch factor t. Salowe for instance uses a technique that takes a v-degree constrained spanner and constructs a new $(\lfloor \frac{v}{2} \rfloor + 2)$-degree spanner. By iteratively applying this degree reduction he arrives at a 4-degree spanner. Let t and t' be the stretch factor before respectively after reducing degree. Then $t' \leq 117t + 864$, which is by no means tolerable in practice, even using a single reduction step. We conjecture the performance of these low degree spanners with respect to our interference measures to be bad. Up to today, v-degree constrained spanners with desirable properties such as planarity, reasonable stretch factor and a distributed construction algorithm are known having $v \geq 20$ [LW03].

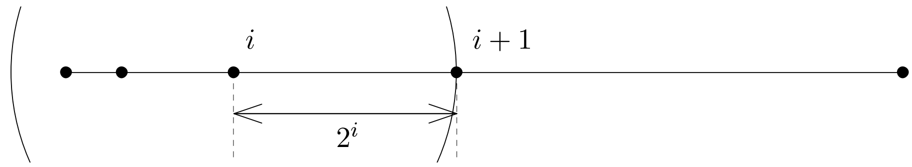

Figure 5.1: Although $\Delta = 2$, interference is $O(n)$.

5.2 Performance of Known Topologies

Obviously, the maximum degree Δ of a graph is a lower bound for interference in any reasonable measure (see Section 6 for a thorough discussion of various measures). Nevertheless interference can be as high as the number of nodes n (or edges respectively). A striking example found in [MSVG02] is the topology shown in Figure 5.1. The distance between two nodes i and $i+1$ is 2^i. Thus each node will cover all nodes to the left when communicating with its right neighbor[1], and, although $\Delta = 2$, interference is $O(n)$ for all measures.

Yet there is no way the nodes could be connected causing less interference, and thus interference of $O(n)$ is optimal. Consider an extended example shown in Figure 5.2. Again, there is a horizontal exponential line as in the previous example. In addition, each node h_i of the exponential line has a corresponding node v_i which is vertically displaced by a little more than the distance to its left neighbor. Let this vertical distance be called d_i, then $d_i > 2^{i-1}$. These additional nodes form a second (diagonal) exponential line. Between two of these diagonal nodes v_{i-1} and v_i, there is a helper node t_i such that $|h_i, t_i| > |h_i, v_i|$.

Consider Prim's algorithm for growing a minimum spanning tree (MST). As it greedily adds shortest edges not forming cycles, it will always connect nodes along the horizontal line via edges along the exponential line (see Figure 5.3), leading to interference of $O(n)$, although the degree of a MST is bounded by 6. This time however, there exists a tree on the nodes that imposes only constant interference. Figure 5.4 shows such a tree. It connects horizontal nodes via vertical edges instead of horizontal edges. By doing this, it omits edges (h_j, h_{j+1}) that cover all nodes to the left in the exponential line. GLIT, the greedy algorithm introduced in Chapter 3, will construct such a constant interference tree for all measures, being optimal for I_{out}. Also, if $t \geq 3$, GLIS will construct a constant interference t-spanner.

Let RNG denote the Relative Neighborhood Graph, GG the Gabriel Graph and DT the Delaunay Triangulation. Then the relation

$$MST \subseteq RNG \subseteq GG \subseteq DT$$

[1] due to $\sum_{i=1}^{n-1} 2^i < 2^n$

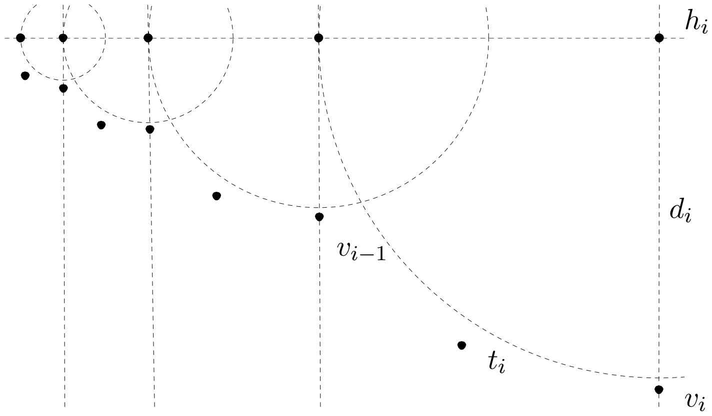

Figure 5.2: Two exponential lines.

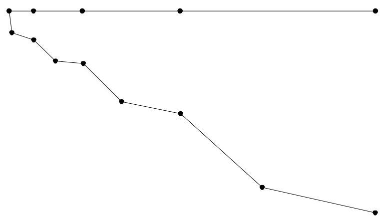

Figure 5.3: MST produces interference $O(n)$.

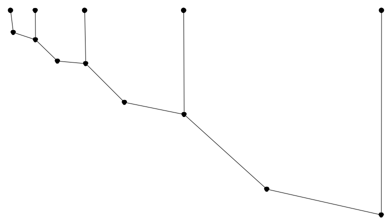

Figure 5.4: Optimal tree with constant interference.

implies that all topology control algorithms relying on one of the listed topologies are prone to interference of $O(n)$ in all measures (as they contain the MST) while optimal interference may be constant. In general, GLIT and GLIS construct I_{out}- and I_{avg}-optimal topologies, while they provide good heuristics for other measures.

Another example where low degree is no guarantee for low interference is the sparse Yao graph [GLSV02]. The basic idea underlying the Yao graph [Yao82], also known as θ-graph, is to divide the space around each node into k equal sectors. Each node is now connected to its nearest neighbor in each sector. If $k > 6$, the Yao graph is a spanner with a stretch factor depending only on k. The major drawback of this kind of graphs is that the in-degree of nodes is not restricted. This disadvantage is overcome in a variation known as the sparse Yao graph, where only the shortest of all incoming edges per sector is kept. The sparse Yao graph has in- and out-degree of at most k (thus degree $2k$ in total).

Now consider the node set shown in Figure 5.5. $I_{out} = n$ if the center node communicates with the nodes on the circle. The sectors originating at u and v both just miss the center node. This causes u and v to add an edge e between them. Sending a message over e interferes with virtually all nodes. Because all nodes might just miss the center node, each node is covered by up to n disks directly leading to $I_{in} \approx n$. Accordingly, I_{avg} can also be as bad as n. In the previous example, constructing the sparse Yao graph leads to the same topology as the MST (see Figure 5.3). It is known that the sparse Yao graph is a power spanner [JRS02] and has constant degree. Nevertheless it exhibits very bad interference properties!

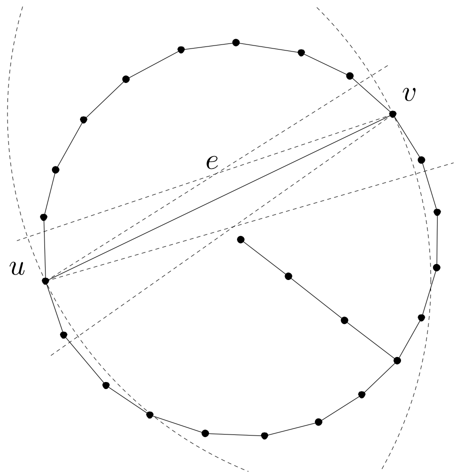

Figure 5.5: An unfavorable edge in a Yao graph.

Thus we have shown two examples where several well known and widely used topologies are bad in terms of interference, but GLIT and GLIS perform well:

Figure 5.2: MST, GG, DT and the (sparse) Yao graph are not competitive.

Figures 2.1 and 5.5: When adding a chain of nodes from the center node to a circle node, GG, DT as well as the (sparse) Yao graph are not competitive.

5.3 Concluding Observations

The shortcoming of Δ as a measure for interference mainly stems from the fact that a node v (or an edge) may be interfered by nodes that are not directly connected to v via an edge. However this is not the case in a Unit Disk Graph (UDG) where there is an edge e between two nodes u and v iff $|u,v| \leq r = 1$. In a UDG, the incoming interference (I_{in}) of a node is given by its in-degree: $I_{in}(v) = \Delta_{in}$. Interference of an edge[2] depends directly on the degree of the associated nodes, as $I_{in}(e) = I_{in}(u) + I_{in}(v) - I_{in}(u) \cap I_{in}(v)$

[2] According to the definition for nodes, incoming interference of an undirected edge $e = (u,v)$ is the number of disks covering u or v. This notion will be discussed more thoroughly in Chapter 6.

where $I_{in}(u) \cap I_{in}(v)$ denotes the number of disks that cover both u and v. It follows that

$$\Delta_{in} \leq I_{in}(e) \leq \Delta_{in}(u) + \Delta_{in}(v) \leq 2\Delta_{in}.$$

A common way to establish topologies for MANETs is to start from a UDG and then consecutively remove redundant edges without affecting desired properties such as the spanner property. This process of removing edges will reduce interference only if nodes are able to lessen their transmission radius in response to the deletion of edges! If all nodes constantly maintain a transmission radius of r, deletion of edges neither affects I_{in} nor I_{out}, as all disks considered remain exactly the same and thus all nodes are covered by the same number of disks. Equation 5.1 follows and gives bounds on I_{in}.

$$\Delta_{in} \leq I_{in} \leq \Delta_{in}^{UDG} \tag{5.1}$$

Δ_{in} denotes the maximum in-degree of the resulting graph and Δ_{in}^{UDG} stands for the maximum in-degree of the UDG on the node set. Similar bounds hold for I_{out}.

Summarizing the chapter, it can be said that there is need for topologies that do not (only) strive for low degree, as it is merely a lower bound, but try to minimize actual interference.

Chapter 6

Modeling Interference

Dealing with wireless ad-hoc networks at the abstraction level of graphs leaves open a wide area of possible interference measures. In this section we will provide an overview and comparison of various interference models. In Section 3 it is shown that minimizing some of them can be achieved by a simple greedy algorithm while minimizing others is NP-complete (see Section 4).

One of the first questions to ask is 'who interferes with whom'? Is it mainly nodes interfering with other nodes or do links (edges) disturb other links? Is there a significant difference between the two perspectives? One might argue that communication along a link imposes interference to all nodes within the link's vicinity and hence interference from edges onto nodes should be considered. Partitioned into four subsections corresponding to the categories *nodes to nodes*, *edges to edges*, *edges to nodes* and *nodes to edges* several more questions are accounted for. Where shall interference be measured, at the originators or rather at the affected nodes? Does it make a difference at all? How do we derive a measure for the entire network from local measures? Relations among these different interference definitions are studied.

6.1 Nodes to Nodes

Mobile stations are the physical entities that eventually emit radio signals and are disturbed by signals of other stations. Hence it can be argued that interference should best be measured between nodes. Figure 6.1 shows an overview of several additional decisions that must be made along the way to a thorough model of interference. Following the insight of Section 1.2 we will only consider directed edges in this section in order to not restrict communication unnecessarily. Anyway, undirected edges will be discussed in following sections.

Choosing the *out* branch at the *direction* node corresponds to measur-

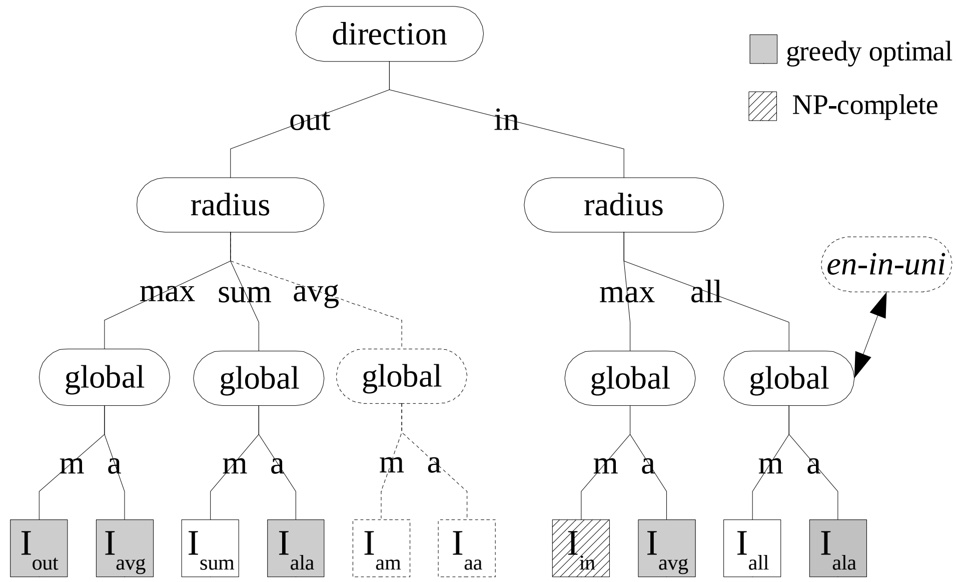

Figure 6.1: Interference *nodes to nodes*.

ing interference at originators (*outgoing interference*) whereas the *in* branch corresponds to measuring interference at the affected nodes (*incoming interference*). The next decision node (*radius*) is concerned with the way local node interference is defined. Suppose the *out* branch is chosen. Thus for each node, the impact its activity has onto other nodes is considered. A message sent along an edge e interferes with all nodes within the disk of radius $|e|$ centered at the sending node. But how can one define the interference of a node with several incident edges? Figure 6.1 shows three possibilities of assigning a value to a node v:

max A disk with radius corresponding to the longest incident edge of v is chosen. The number of nodes covered by this disk is the value for v.

sum For each incident edge of v, the number of nodes covered by the corresponding disk is summed up.

avg Similar to the *sum* case, but the value is then divided by the number of edges.

No matter how the local measure is defined, when stepping onwards to the entire graph, we need some pooling of individual nodes. This can for instance be done by taking the maximum (m) or the average (a) of all nodes, as shown in the figure. We could as well simply build the sum over all nodes, but this would induce a direct dependency on the number of nodes in the graph.

Along the *in* branch we have a bifurcation with two options:

max As in the *max* case of the *out* branch, a disk of maximum radius is considered for each node in the graph. The value assigned to a node v now is the number of disks that cover v.

all Not only the maximum edge of a node is considered, but all edges are represented by a disk with radius equal to the edge's length. Again, the number of disks covering v is counted.

Again, we have the *max* and the *all* pooling option at the bottom of the decision tree. A subtree in the tree is denoted by the decisions that lead to it. For instance I_{in} is the measure attained when following the *nn-in-max-m* branch, where *nn* stands for the diagram (*nodes-to-nodes* in this case).

Branch *nn-in-all* is equivalent to branch *en-in-uni*[1] in Figure 6.3 as indicated by the attached box.

The same notion of interference (I_{avg}) is defined by following the *max-a* branch in either *in* or *out* context. This can be seen by bringing to mind the fact that in both cases a covering of a particular node by a particular disk is counted exactly once, either at the originator or at the affected node. Therefore both approaches end up with equal sum and average, though the addends may be different. The same is true for I_{ala}.

The gray shaded measures are those that can be tackled by the greedy algorithm introduced in Section 3, at least for (strongly) connected graphs. Hatched boxes are used for measures for which an NP-completeness result has been proven (see Section 4). These results are not exhaustive, as a white box indicates that the measure has not yet been shown to belong to one of the two (or any other) classes. Note that simply by changing from the *out* to the *in* branch (choosing all other branches alike), a greedily accessible measure as I_{out} can turn into an intractable measure I_{in} for which even the construction of a dominating set is NP-complete (see Section 4)!

Measures along *out-sum* and *in-all* can grow up to n^2 which can be doubted to be reasonable. Intuitively a mobile station cannot interfere with more than all nodes. Other branches yield measures bounded by $O(n)$.

Branch *out-avg* is not considered, as the corresponding measures exhibit undesirable properties. At every node, the number of covered disks is averaged over all incident edges. Say there is a node with a very long edge covering virtually all nodes and some short edges covering merely a neighbor or two. The heavy interference of the long edge is eased by the shorter edges. Therefore a node with a heavy edge can decrease its interference by starting to communicate along additional short edges, which is contradictive. Additional communication should certainly not reduce interference. Nevertheless *out-avg* is shown in the figure for sake of completeness.

[1] *en* stands for *edges to nodes*

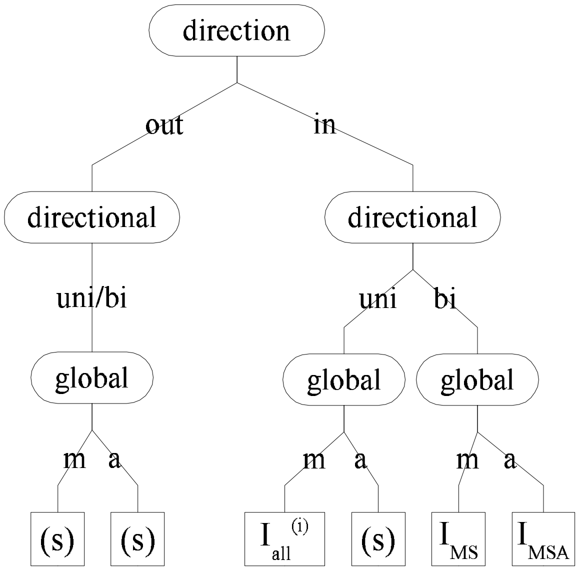

Figure 6.2: Interference *edges to edges*.

6.2 Edges to Edges

Similar to the model of Meyer auf der Heide et al. [MSVG02], denoted I_{MS} here, interference can be considered between edges. Figure 6.2 shows an overview of different *edges to edges* measures. The approaches mainly differ in terms of *outgoing* or *incoming interference* and uni- or bidirectional edges. A bidirectional edge $e = (u, v)$ is considered to be interfered by a disk if the disk covers u or v. An unidirectional edge is only interfered by disks that cover target node v. This makes sense because the transmission of a message via a directed link is only disturbed if interference occurs at the target. Interference merely at the sender does not bother, as radio signals follow the concept of linear superposition. I_{MS} can be found along *in-bi-m*. The according average measure is called I_{MSA}.

Definition 12 (Incoming Property). *A directed graph conforms to the incoming property, denoted (i), if every node has at least one incoming edge, that is, every node can be reached by at least one other node. Accordingly, an undirected graph conforms to (i) if every node has at least one incident edge.*

Figure 6.2 shows a measure with a superscript (i). This denotes that the measure at *in-uni-m* (denoted I_*) is equivalent to I_{all}, given that the incoming property holds for the underlying graph. This equivalence can easily be understood. Say the value of I_{all} is known for a graph G. Some node v in G is maximally covered by I_{all} directed edges. Now if the incoming property holds, v has an incoming edge e that is covered by exactly the same number of edges. Clearly, e cannot be covered by more disks without increasing I_{all} at v. On the other hand, if I_* is known, we know that the target node of the maximally covered edge has the same covering edges, thus $I_* = I_{all}$. If the incoming property is not guaranteed, $I_* \leq I_{all}$. Generally it can be said that all relevant topologies will be compliant with this property, because we are interested in networks in which every node can be reached.

A common relation between similar measures that only differ in terms of targets (nodes or edges) is the substitution relation which is defined below.

Definition 13 (Substitution Relation). *Let source → target denote a category of interference measures where source, target ∈ {nodes, edges}. Let I_n be a measure in category $s \rightarrow nodes$ and I_e a measure in category $s \rightarrow edges$ along an identical decision path as I_n. That is I_e and I_n are at the same place in the decision tree, only the category differs by the target. Then I_n and I_e are related by the substitution relation, denoted (s), if the following two equations hold:*

$$I_e \leq \Delta I_n$$
$$I_n \leq 2I_e$$

For directed graphs, Δ refers to the maximum in-degree, whereas for undirected graphs it simply denotes the maximum node degree. The substitution relation does not require the incoming property to hold.

If the substitution relation holds for two measures I_n and I_e, an interval for either one is given, dependent on the respective other measure. If we know the value of I_e, then $\frac{I_e}{\Delta} \leq I_n \leq 2I_e$. On the other hand if I_n is known, $\frac{I_n}{2} \leq I_e \leq \Delta I_n$ holds.

In a decision tree, a box with symbol (s) means that the particular measure is related by the substitution relation to its corresponding measure in the category where targets are swapped. For instance, according to Figure 6.2, the substitution relation must hold for *ee-out-uni-m* (I_e) and *en-out-uni-m* (I_n), which is shown in Figure 6.3 and is equivalent to I_{out}. The validity of this relation can be seen by some simple arguments as follows. In category *ee* we consider the interference imposed by directed edges on directed edges. In category *en* the interference imposed on nodes is considered. First, consider the case where I_n is known. There is an edge that covers I_n nodes. At each of these covered nodes, no more than Δ edges can be adjacent. Therefore I_e is limited by ΔI_n. If on the other hand I_e is known, some edge covers I_e edges. As stated before, a directed edge is deemed interfered if its target node is covered. For each of these edges at most two nodes are covered. If the incoming property holds, there cannot be any covered nodes which are not adjacent to a covered edge. Thus I_n cannot exceed $2I_e$. Taken together, these relations constitute the substitution relation (s).

The case where an *incoming average* pooling measure (e.g. *ee-in-uni-a* denoted $\overline{I_e}$ and *en-in-uni-a* denoted $\overline{I_n}$) is claimed to meet (s) needs different argumentation. Say $\overline{I_n}$ is given. Then every node is covered by $\overline{I_n}$ edges in average. Thus $|V| \cdot \overline{I_n}$ coverings take place. For each of these node coverings, at most Δ edge coverings may exist. Averaging the edge coverings over edges instead of nodes leads to

$$\overline{I_e} \leq \frac{\Delta |V| \overline{I_n}}{|E|} \approx \Delta \overline{I_n}.$$

The approximation works if we assume that $|E| \approx |V|$. Following the same trace as above and using the assumption, $\overline{I_n} \leq 2\overline{I_e}$ can be derived as well, which completes the substitution relation.

Hence if (s) is annotated to a measure, the incoming property is assumed to hold. If (s) is annotated to an *average incoming* measure, it is only valid if $|E| \approx |V|$, which trivially holds for (spanning) trees and the like. A proportionality factor between $|E|$ and $|V|$ leads to accordingly scaled inequalities.

For other measures listed to be complying with (s), derivation is omitted, as it follows similar reasoning. Branches *out-uni* and *out-bi* have been combined, for both yield equal results.

We conclude this section by deriving a relation between I_{in} and I_{MS}. Suppose the incoming property is assured. Then trivially, $I_{in} \leq I_{MS}$, because the node v with maximum interference $I_{in}(v)$ also has an (undirected) incident edge $e = (v, v')$ with at least the same amount of interference. For each disk covering v there is a center node c. c has at most Δ incident edges. Although there is only one disk originating in c that counts for $I_{in}(v)$, each of the Δ incident edges potentially interferes with v when the measure is changed to I_{MS}. Furthermore e might be an edge between v and v' of which both have maximum interference $I_{in}(v) = I_{in}(v')$ but no disk covers both of them. Thus I_{MS} can be at most $2\Delta I_{in}$. Put together this states

$$I_{in} \leq I_{MS} \leq 2\Delta I_{in}$$

or $\frac{I_{MS}}{2\Delta} \leq I_{in} \leq I_{MS}$, respectively.

6.3 Edges to Nodes

One could be interested in measuring the interference of communication links (edges) onto mobile stations (nodes). Figure 6.3 gives an overview of interference measures in the *edges to nodes* category.

A remarkable property of this category is the applicability of the greedy algorithm to the entire *out* branch. This stems mainly from the fact that

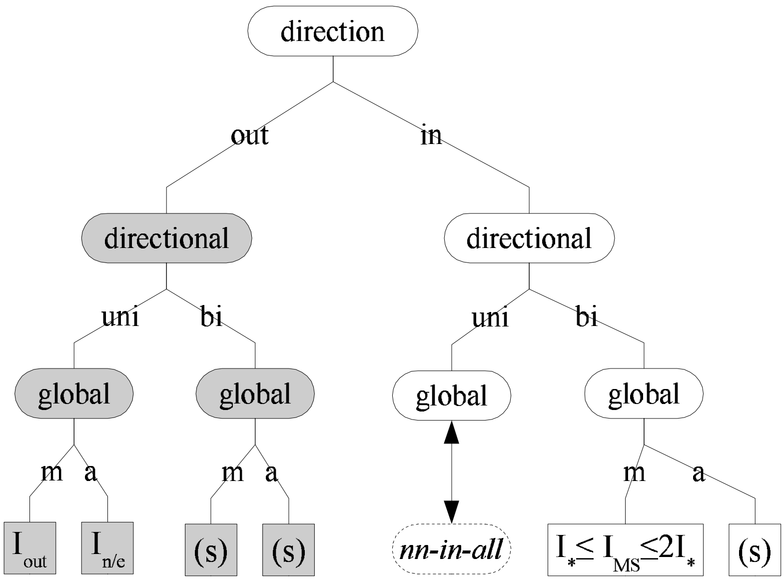

Figure 6.3: Interference *edges to nodes*.

outgoing interference to static entities (nodes) is measured. So the interference of an edge considered to be added can once for all be computed and will not change, no matter what edges are added later on.

The measure *out-uni-m* is equivalent to I_{out}, since it makes no difference whether the edge covering the most nodes is considered separately or as the maximum incident edge of a node. $I_{n/e}$ measures the average number of nodes covered by edges. This is slightly different from I_{avg} where the average is done over all nodes instead of edges.

As mentioned in Section 6.1, the branch *in-uni* is equivalent to *nn-in-all*. Measure *in-bi-m*, denoted I_*, can be closely related to I_{MS}. By changing target entities from edges (I_{MS}) to nodes (I_*), it is clear that $I_* \leq I_{MS}$. At the same time an edge can capture at most twice the incoming interference as the maximum of one of its nodes. Thus $I_* \leq I_{MS} \leq 2I_*$ or equivalently $\frac{I_{MS}}{2} \leq I_* \leq I_{MS}$. This relation does not require the incoming property to hold.

6.4 Nodes to Edges

When considering interference from nodes to edges we restrict the node model to *nn-out-max* (see Figure 6.1) due to the discussed disadvantages of *nn-out-sum* and *nn-out-avg* (see Section 6.1). Thus branches *ne-out* and *ne-in* correspond to branches *nn-out-max* and *nn-in-max* respectively. Further decisions along the path concern the target edges and global pooling.

If the incoming property holds, *in-uni-m* is equivalent to I_{in} and thus

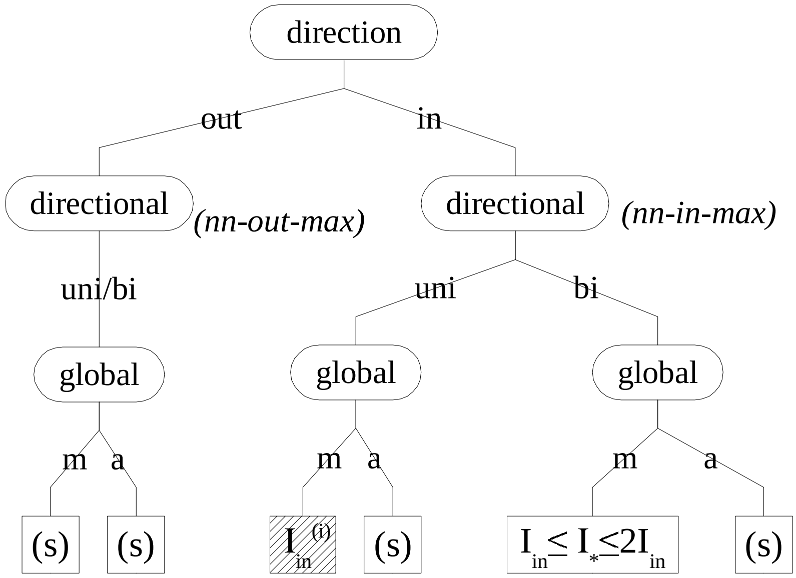

Figure 6.4: Interference *nodes to edges*.

minimizing it can be NP-complete. Measure *in-bi-m* (denoted I_*) is tightly related to I_{in} also, as $I_{in} \leq I_* \leq 2I_{in}$ holds.

For the remaining measures, the substitution relation is valid. Measures of this category need to be related to their corresponding measures in the *nn* category (Figure 6.1), just imagine the substitution of target edges by target nodes. The *directional* decision can therefore be ignored when relating these measures to each other. Remarkably, all measures marked with (s) refer to greedily tractable measures, which is to say that upper and lower bounds constituted by (s) can easily be calculated by a greedy algorithm.

6.5 Self-Interference

In all models discussed so far, interference of a node with itself is counted (I^1). Alternatively, we could argue that a sending node is not disturbed by its own transmission and thus neglect self-interference (I^0). However, there is a close relationship between the two models. If we compare the same interference model once counting self-interference and once not, we get

$$I^0 \leq I^1 \leq I^0 + 1. \tag{6.1}$$

Say I^1 is the interference of a solution to CMIB, counting self-interference. Let D be the set of dominating nodes for this solution. Then there is a set of nodes M for which interference is equal to I^1. Now assume we stop counting self-interference. If $M \subseteq D$ (all nodes with maximum interference are dominators) then D is a solution with $I^0 = I^1 - 1$ ($I^0 = I^1$ if $M \not\subseteq D$).

Anyhow, it is not possible to find a solution D' with $I^0 < I^1 - 1$, otherwise D' could be used to find a better solution to I^1 as well. Thus $I^0 \geq I^1 - 1$ which is equivalent to $I^1 \leq I^0 + 1$ and makes up the right part of Inequality 6.1. On the other hand of course, if we have a solution with interference I^0 and start counting self-interference, this may only increase the interference of any solution. Thus $I^0 \leq I^1$, which corresponds to the left part of the inequality.

Chapter 7

Conclusion and Future Work

Focused attention has been payed to the field of MANETs in the past years. The main part of recent research has been dedicated to the fundamental issues of topology control and multi-hop routing. Although it is often argued that in MANETs low degree topologies are well suited to minimize interference, models of interference on the abstraction level of graphs are poorly studied.

In this work we provided an in-depth discussion of various possible interference definitions (Chapters 2 and 6). A classification of models has been given and relations among different models have been investigated. One of the main differences between models is whether they focus on outgoing or incoming interference. For outgoing interference between nodes, we presented a greedy algorithm (GLIT) (Chapter 3) that constructs an interference optimal spanning tree (MIT). An adaptation of the algorithm yields an interference optimal t-spanner (GLIS). For incoming interference (e.g. I_{in}) these algorithms are merely heuristic. A dynamic greedy criterion was suggested to better approximate optimal incoming interference.

The MINIMUM INTERFERENCE BROADCAST (MIB) problem and its connected variant (CMIB) were defined in Chapter 4. A solution for CMIB can be used as an interference optimal virtual backbone in ad-hoc networks. MIB and CMIB were shown to be NP-complete. A constant approximation of CMIB was shown, which is at the same time also an approximation of MIB. A change of measure from I_{in} to I_{out} makes MIB as well as CMIB optimally solvable in polynomial time, which is a very intriguing result concerning the complexity of interference measures.

The common conjecture that low degree is enough to guarantee low interference was discussed in Chapter 5. The maximum degree of a network turned out to be merely a lower bound for interference. Despite the fact that reducing interference is the main reason for doing topology control at all, any algorithm relying on topologies as the MST, RNG, GG, DT or Yao graph was shown to be prone to bad interference properties, while the introduced algorithms GLIT and GLIS are interference optimal for I_{out} (and I_{avg}) and provide good heuristics for other measures.

Future work will include exhaustive analysis of the various proposed interference models. Maybe all measures can be partitioned into groups of closely related interference measures. Are there other ways of defining interference? Does low interference imply high network capacity? Of course the model could be adapted to more precisely reproduce physical properties of radio signals. The signal of a distant node could be modelled to be very faint and cause little interference, while a close-by sending node would render impossible the reception of any other signal.

Besides the general issues, there are simple questions to be answered that tie in with this work directly. It follows a list of some continuing problems that could not be answered within the time frame of this thesis:

- Is GLIT/GLIS c-interfering with respect to I_{in}?
- How well does $GLIT_{imp}$ approximate I_{in}?
- Can GLIT/GLIS be approximated by a distributed algorithm?
- Are there good non-greedy algorithms that minimize interference?
- Can a practical approximation of CMIB be found?
- Is building a (strongly) connected topology with minimum I_{in} NP-complete?
- How bad is the interference of the low degree spanners mentioned in Section 5.1 really?
- Are the presented interference measures relevant for congestion or capacity? For instance, can we say that low interference implies high capacity?
- Is there a topology control algorithm which is good in terms of interference and has other nice properties (local or distributed construction, Euclidean spanner, low power consumption, planarity, high capacity, etc.)?

The last item already states the final goal of this research: A topology control algorithm that has many of the desired properties of today's algorithms but also takes interference into account. Being a first step on the way, this work has provided many basic definitions and results that are fundamental for our understanding of interference in ad-hoc networks.

Bibliography

[AS94] Sunil Arya and Michiel Smid. Efficient construction of a bounded degree spanner with low weight. In *Proceedings of 2nd European Symposium on Algorithms (ESA '94)*, pages 48–59, 1994.

[AWF02] Khaled M. Alzoubi, Peng-Jun Wan, and Ophir Frieder. Message-optimal connected dominating sets in mobile ad hoc networks. In *MobiHOC*. EPFL Lausanne, Switzerland, 2002.

[DB97] Bevan Das and Vaduvur Bharghavan. Routing in ad-hoc networks using minimum connected dominating sets. In *ICC (1)*, pages 376–380, 1997.

[DFS90] D.P. Dobkin, S.J. Friedman, and K.J. Supowit. Delaunay graphs are almost as good as complete graphs. *Discrete Computational Geometry*, 5:399–408, 1990.

[DH93] G. Das and P. J. Heffernan. Constructing degree-3 spanners with other sparseness properties. In *Proceedings of 4th Annual Intl. Symp. on Algorithms and Computation, Lect. Notes in Comp. Sci. 762*, pages 11–20, December 1993.

[DN94] G. Das and G. Narasimhan. A fast algorithm for constructing sparse euclidean spanners. pages 132–139, 1994.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman, 1979.

[GLSV02] Matthias Grünewald, Tamas Lukovszki, Christian Schindelhauer, and Klaus Volbert. Distributed maintenance of resource efficient wireless network topologies. In *Proceedings of the 8th International Euro-Par Conference*, pages 935–946, Paderborn, Germany, 27 - 30 August 2002.

[JRS02] L. Jia, R. Rajaraman, and C. Scheideler. On local algorithms for topology control and routing in ad hoc networks. Technical report, Johns Hopkins University, 2002.

[Lic82] David Lichtenstein. Planar formulae and their uses. *SIAM J. Computing*, 11:329–343, 1982.

[LW03] Xiang-Yang Li and Yu Wang. Efficient construction of low weight bounded degree spanner. In *The Ninth International Computing and Combinatorics Conference (COCOON)*, 2003.

[MIH81] S. Masuyama, T. Ibaraki, and T. Hasegawa. The computational complexity of the m-center problems in the plane. *The Transactions of the IECE of Japan*, E64:57–64, 1981.

[MSVG02] Friedhelm Meyer auf der Heide, Christian Schindelhauer, Klaus Volbert, and Matthias Grünewald. Energy, congestion and dilation in radio networks. In *Proceedings of the 14th ACM Symposium on Parallel Algorithms and Architectures*, Winnipeg, Manitoba, Canada, 10 - 13 August 2002.

[Sal94] J. S. Salowe. Euclidean spanner graphs with degree four. *Discrete Appl. Math.*, 54:55–66, 1994.

[Sch78] Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the tenth annual ACM symposium on Theory of computing*, pages 216–226. ACM Press, 1978.

[Soa92] J. Soares. Approximating euclidean distances by small degree graphs. Technical Report CS 92-05, University of Chicago, 1992.

[TNCS02] Yu-Chee Tseng, Sze-Yao Ni, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. *Wireless Networks*, 8(2/3):153–167, 2002.

[Wei91] Mark Weiser. The computer for the 21st century. *Scientific American*, 30:94–104, 1991.

[WL99] J. Wu and H. Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proceedings of the Third International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DiaLM)*, pages 7–14, 1999.

[WLBW01] Roger Wattenhofer, Li Li, Paramvir Bahl, and Yi-Min Wang. Distributed topology control for wireless multihop ad-hoc networks. In *INFOCOM*, pages 1388–1397, 2001.

[Yao82] A. C. Yao. On constructing minimum spanning trees in k-dimensional spaces and related problems. *SIAM Journal on Computing*, 11(4):721–736, 1982.