UPPER AND LOWER BOUNDS IN RADIO NETWORKS

BY MIGUEL MOSTEIRO

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Computer Science

Written under the direction of

Martín Farach-Colton

and approved by

____

____

____

____

New Brunswick, New Jersey

May, 2007

ABSTRACT OF THE DISSERTATION

Upper and Lower Bounds in Radio Networks

by Miguel Mosteiro

Dissertation Director: Martín Farach-Colton

Sensor nodes are very weak computers that get distributed at random on a surface in order to achieve a large-scale sensing task. Once deployed, they must wake up and form a radio network. Given the extremely limited resources of sensor nodes, finding efficient solutions even for basic problems is very challenging. The results in this thesis concern the initialization from scratch, or *Bootstrapping*, of a Sensor Network. More precisely, we seek efficient-provable solutions to the most fundamental problem in Sensor Networks, its self organization. At the same time, we study lower bounds on the time complexity of such a problem.

The first set of results in this thesis address the three parts that Sensor Network bootstrapping research has: to model the restrictions on sensor nodes; to prove that the sensors connectivity graph has a subgraph that would make a good network; and to give a distributed protocol for finding such a network subgraph that can be implemented on sensor nodes.

A study of the Sensor Network Bootstrapping problem would not be complete without a study of lower bounds on the time complexity of solving it. Strikingly, the most basic problem in a Radio Network, i.e. to achieve a successful transmission, can be proved to be as difficult as other more complex problems under the constraints of a sensor node. The second set of results of this thesis shows new lower bounds for collision-free transmissions in Radio Networks. The main lower bound is tight for a variety of problems. An extension of this result gives the first lower bound for Sensor Network Bootstrapping. A lower bound on the expectation for fair

protocols is also shown.

Another contribuition of this thesis is a survey of research in Radio Networks. The survey includes two parts that have received extensive study: upper bounds for Sensor Network formation, and upper and lower bounds for non-colliding transmissions in Radio Networks proved under the broader context of more complex problems.

Acknowledgements

My experience during the last five years has been an exceptional journey - one that I had never dreamed of. The person responsible for opening my world to such an opportunity was my advisor, Martín Farach-Colton. I am truly indebted to him for that. My interest in research in Theoretical Computer Science started at the University of Buenos Aires. But, it was through attending a memorable Advanced Data Structures course, taught by Martín and Michael Bender, that I got excited about algorithms and data structures and the possibility of devoting myself full-time to theory research. I probably would not be writing this note if that enlightening had not taken place.

Working with Martín has been an invaluable experience that has had a profound influence throughout my academic career. First of all, he provided key insights and research directions always balanced with humor and friendliness. He also encouraged me to develop independent thinking and research skills. He continually stimulated my analytical thinking and greatly assisted me with scientific writing. He always treated me as an equal, just someone less experienced in need of advice about problem-solving, publications, collaboration, and academic politics. His continuous encouragement, guidance and advice have gone beyond research issues in many ways that helped me and my family to overcome the huge change in our life that moving to the United States involved.

I would like to acknowledge many people for helping me during my doctoral work. First, I wish to thank Michael Bender, Martín Farach-Colton, S. Muthukrishnan and William Steiger for honoring me by serving on my dissertation committee. Also, I want to give a special thanks to Michael Bender for his advice, many useful discussions and, in particular, for inviting me to collaborate with him and my advisor in a result about Insertion Sort that led to my first publication. Also, I want to thank Eric Allender for being my academic advisor during my first year at Rutgers; his wise guidance was extremely important for me in those days of adjustment.

I am amazed at all I have learned in the last five years from faculty members, postdocs and visitors at DIMACS, and fellow students. This dissertation builds upon and is influenced by the work, knowledge and wisdom of many of them. My work directly benefited from discussions with S. Muthukrishnan and Graham Cormode. In particular, the results in Chapter 6 were initially motivated by various fruitful discussions with them. I want to thank Eric Allender, Văsek Chvátal, Martín Farach-Colton, Venkatesh Srinivasan, Mario Szegedy, József Beck and Jeffry Kahn for many enlightening lectures and useful office discussions. It was particularly exciting for me to attend the Computational Geometry lectures of Bill Steiger, an awesome teacher always open to stimulating discussions. My work was also influenced by Endre Szemerédi with whom I also had the privilege of engaging in useful discussions about topics beyond this thesis. Fellow students and friends such as Matías Cuenca, Carlos Diuk, Rohan Fernandes, Michal Koucký, Marcelo Mydlarz, Christian Rodríguez, Detlef Ronneburger, Sambuddha Roy, Bin Tian and others helped to keep me going and influenced my work too. I also greatly benefited from seminars at DIMACS, the Combinatorics seminar, the IAS in Princeton, the seminars at Bell Labs and AT&T Labs, as well as from people attending the reading seminars and theory lunches organized by Michal Koucký. A particularly fruitful experience was to attended the Dagstuhl Seminar on Cache-Oblivious Data Structures. I thank my advisor for inviting me to that event.

I can not stress enough how important it is for me to work with others. In that regard, I want to specially thank my friend, fellow student and collaborator Rohan Fernandes for months of useful joint work and exciting discussions. Most of the results in Chapters 6 and 8 and in Section 3.2.2 are joint work with him and our advisor and have appeared in [FCFM05] and [FCFM06].

I want also to thank my friends and teachers at UBA: Willy, Marcelo, Irene, Víctor, Mariano, Pablo and many others for helping me to prepare for this experience. I specially want to thank my number-theory teacher there, Juan Sabia, for many inspiring lectures. A special thank you to my Galician family for their love and support while I was touring Spain during my job hunt. I am very grateful also to my friends in New Jersey: Matías & Cecilia, Carlos & Valeria, Gonzalo, José, Nachi, Roberto & Jessica, and many other members of ASADO for making the experience at Rutgers unforgettable on the social side. Thank you to my friend

Stephen Rossiter for his comments while formatting this note.

Finally, there is one person above all others who deserves my deepest thanks and respect for her continued support since I decided to devote myself to research: my wife, Silvia Romero. I could not have done it without her. Also, our children, Pablo and Martín, have been my greatest source of inspiration and strength, with their struggle to adapt to our new situation without detracting from their own fruitful social and academic life. To the three of them, my greatest gratitude for these years of love and support.

During my graduate studies I was partially supported by the following sources:

- Excellence Fellowship for Doctoral Study in Computer Science, Rutgers University, Graduate School New Brunswick.
- DIMACS, Center for Discrete Mathematics & Theoretical Computer Science, grants numbered NSF CCR 00-87022, NSF EIA 02-05116 and Alfred P. Sloan Foundation 99-10-8.
- My advisor's grants: NSF Grant: CCF 0632838, NSF Grant: CNS 0625754, NSF Grant: CCF 0541097.

Miguel A. Mosteiro

Dedication

To my father, Manuel Mosteiro, whose enduring spirit has been always my guidance.

Table of Contents

List of Figures

Chapter 1

Introduction

Advances in technology have made it possible to integrate sensing, processing and communication capabilities in a low-cost multi-function device, popularly known as a *sensor node*. Sensor nodes are randomly deployed over an area and must self-organize as a radio-communication network called a *sensor network*. Even though communication among sensor nodes is through radio broadcast, it is useful to set up explicit links between nodes in order to establish routing paths and prevent flooding.

A Sensor Network is capable of achieving large tasks through the coördinated effort of sensor nodes, but individual nodes have severe limitations on memory size, life cycle, range of communication, etc. Due to these harsh limitations, classical solutions from the Radio Networks and other areas can not be straightforwardly applied making the area of Sensor Networks very attractive. Furthermore, although problems and even results can be easily understood with little knowledge of mathematics or computer science, to find efficient solutions to basic problems is not a trivial task.

Problems in sensor networks are twofold: related to geometric properties and related to network protocols. Sensor nodes have limited range and are deployed at random over a large area. Therefore, geometric properties like path length, coverage and connectivity need to be understood. Thus, sensor networks are modeled as *random geometric graphs*. On the other hand, node limitations, shared communication channel and lack of additional infrastructure at deployment impose limitations on network protocols. Usually for parallel and distributed computing, there are many differences among computing models used in Radio Networks due to diverse technical details that may have immense impact on algorithmic issues in communication. The different models explicit or implicit in many results are incomplete or inadequate in order to reflect the various limitations under which sensor nodes operate.

The most fundamental problem in Sensor Networks is to establish efficiently the network among sensor nodes from scratch right after deployment, or the *Sensor Network Bootstrapping* problem. Sensor Network bootstrapping research has three parts: one must formally model the restrictions on sensor nodes; one must prove that the connectivity graph of the deployed sensor nodes has a subgraph that would make a good network; and one must give a distributed protocol for finding such a network subgraph that can be implemented on sensor nodes.

In this thesis, all of the three parts of Sensor Network bootstrapping research are addressed in detail: a formal Weak Sensor Model that summarizes the literature on sensor node restrictions, taking the most restrictive choices when possible, is elucidated in Section 3.2.2; in Chapter 6 it is shown that sensor connectivity graphs have low-degree subgraphs with asymptotically optimal *hop-stretch*, as required by the Weak Sensor Model; and a Weak Sensor Model-compatible $O(\log^2 n)$ [1] protocol for finding such graphs with high probability (w.h.p.), i.e. with probability $1 - n^{-\gamma}$ for some constant $\gamma > 0$[2], is given in the same chapter. This is the first network initialization algorithm that is implementable on sensor nodes.

Given the absence of provable upper bounds for Sensor Network initialization in the previous literature, it is not surprising that lower bounds for this problem where not known until recently. Any network where transmissions may collide, such as a Sensor Network, needs a protocol to achieve *collision-free transmissions*. The narrow gap between the lower bound for achieving something so simple as a clear transmission in Radio Networks and upper bounds for more complicated problems such as Maximal Indepent Set, motivates the study of this problem in order to obtain lower bounds for Sensor Network initialization.

In Chapter 8, new lower bounds for collision-free transmissions in Radio Networks are shown. More precisely, the main result is a tight lower bound of $\Omega(\log n \log(1/\epsilon))$ on the time required by a randomized protocol to achieve a clear transmission with success probability $1-\epsilon$ in a one-hop setting. The main result is tight for a variety of problems. The first lower bound for clear transmissions in the Sensor Network setting, i.e., when the topology is model by a connected random geometric graph, is obtained as an extension of that result. Finally, a lower

[1]Notation: throughout this thesis, $\log$ indicates logarithm base 2 unless otherwise stated

[2]Notation: throughout this thesis, we use *with high probability*, *w.h.p.*, or *with probability* $1 - n^{-\gamma}$ *for some constant* $\gamma > 0$ indistinctively

bound on the expectation for fair protocols is also shown.

Another contribution of this thesis is a survey of research in upper bounds for Sensor Network initialization, and bounds for non-colliding transmissions in Radio Networks, which is included in Chapters 5 and 7 respectively. The latter results are proved under the broader context of other problems such as broadcast, wake-up and leader election. In addition to the detailed description of our Weak Sensor Model, in Chapter 3 a survey of models frequently used in Radio Networks is given. Frequent terminology used in communication networks in general and Sensor Networks in particular, and definition of problems studied in Radio Networks are included in Chapters 2 and 4 respectively.

Chapter 2

Terminology

Radio Networks is a vast and active area of research in both the applied and theory communities. In order to analyze the various problems present in Radio Networks in general and Sensor Networks in particular, we summarize in this section the terminology usually adopted in the literature. Given that this terminology is mostly borrowed from the more general computer networks area, we begin with definitions for general networks, narrowing down later to our specific area of interest.

A *computer communication network* is a collection of entities with information processing and communication capabilities. In order to achieve some distributed computation, these entities, usually called *nodes*, need to exchange information among them and perhaps share distributed resources. To that extent, protocols to establish, maintain and use such a network have to be carefully designed, according with the constraints of each application.

The different classification criteria are as varied as the applications of such networks. One popular classification is based on the size and distance among nodes. Although there are other intermediate categories, we distinguish here two main ones: *local area networks* and *wide area networks*.

The communication among nodes is achieved through entities called *links*. These entities are implemented as electromagnetic links either through wires (copper, fiber optics, etc.), in which case the network is called *wired*, or through non-wired links (radio waves, infrared light, etc.) normally called *wireless networks*.

Given the existance of wireless networks, we can consider different classifications according with the stability of the topology of the network. Depending on the mobility of nodes, networks can be *stationary* or *mobile*. In cases such as Sensor Networks, even if the nodes are static, the lack of reliability can be modelled as mobile nodes. Changes in topology due to link changes are not used in the literature to classify networks but as additional constraints in problems were this issue is present.

Given that nodes are information processing devices, a standard assumption is that they all have their own clock. Each clock cycle is called a *step* or *time slot*. [1] Regarding synchronicity in the execution of the distributed protocols, networks can be *asynchronous*, *locally synchronous*, i.e., all clocks have the same frequency but perhaps different starting times, or *globally synchronous*, i.e., all clocks start at the same time and have the same frequency.

Nodes communicate among them by means of finite strings of bits usually called *messages*. These messages are padded with additional information such as origin, destination, etc. in order to facilitate the transmission of the message through the network. A message formatted in this way is called a *packet*. It is assumed that the transmission of one packet from one node to a neighboring node incurs in no delay and it is transmitted in one time step or time slot. Protocols that work under this assumption are called *slotted*. If a node a transmits a message to some other node b, we say that a *sends* a message and b *receives* a message. According with the application, in order to analyze message routing and scheduling protocols it is sometimes assumed the existance of an adversarial mechanism that generates messages to be transmitted and assigns these messages to nodes. Such mechanism may work either in advance or online. When online assignment of messages is assumed we say that a message has *arrived* to a node at a given time slot if it was assigned to such a node in that time slot.

In networks where all nodes share the communication channel, a message sent by a given node can be received by all the other nodes, either because nodes are directly connected to the transmitter or by message forwarding of the intermediate nodes. This type of configuration where a node sends a message to all its neighbors simultaneously is usually called *broadcast network*. On the other hand, in *point-to-point networks*, messages are sent from the originator to a recipient node, once again either directly or by message forwarding.

Another useful classification regarding the topology of the network is based on connectivity. If every node is connected directly to every other node, i.e., the underlying graph modelling the topology is a clique, the network is called *single-hop*. On the other hand, if there are pairs of

[1] In fact, a time slot will be the cost unit of protocols unless otherwise stated.

nodes which are not connected by a single link, the network is called *multi-hop*.

The size of the network is defined to be the number of nodes n. Nodes are assumed to have been assigned a unique *identifying number* (ID) in the range $[1..\Theta(n)]$ and that they know their own ID as well as the size of the network or an upper bound of that size given by the range of ID numbers. Distributed protocols have to be robust enough to handle at least the initialization of the network, when nodes ignore the ID numbers of their neighbors. Networks where protocols do not have this topology information available are called *unknown-topology networks* or *ad-hoc networks*.

We now focus our atention on how nodes handle messages concurrently. Depending on the type of network, a node may be able to send a message to a subset of neighboring nodes in a given time slot as well as receive messages from a subset of neighboring nodes simultaneously in a given time slot. In certain networks, in order to receive a message successfully the size of the subset of adjacent transmitting nodes has to be exactly one. In fact, this is the case for Radio Networks where only one channel of communication is available.

In a given time slot, nodes may operate in receiver mode, transmitter mode or perhaps both. If in a given time slot a node that is operating as a receiver does not receive any message, it receives what we call *noise* which is some kind of signal different from any known message. If during such a time slot, no adjacent node has transmitted any message, we say that the noise received is *background noise*. Whereas, if more than one adjacent node has transmitted a message during that time slot, we say that a *collision* has ocurred and the signal received is *interference noise*. We call a network where nodes can distinguish between background and interference noise a network *with collision detection* (with CD) and *without collision detection* (without CD) otherwise.

In shared-channel networks, nodes compete to gain access to the channel. In many cases, deterministic protocols to share the channel are not suitable due to various factors, e.g., lack of knowledge of the topology. If no centralized control is available then conflicts among nodes while trying to access the channel are unavoidable. Randomized distributed protocols intended to resolve such conflicts are called *contention resolution protocols*.

Networks are also classified depending on the type of feedback that nodes receive from the channel. As explained before, nodes might be able to distinguish between 3 states of the channel, namely, silence (background noise), transmission, and collision (interference noise). When this information is available it is said that the channel has *ternary feedback*. Whereas in the case that nodes can only distinguish between a meaningful message or a meaningless one, we say that the channel has *binary feedback*.

Regarding feedbak there is an special case depending on wether a node is able to transmit and receive in the same time slot. There are networks, specially Radio Networks, where a node can not receive while transmitting because the strength of the signal transmitted is much higher than the signal coming from neighboring nodes due to signal decay. However, when such a simultaneous reception is possible, nodes may be able to detect the collision of their own transmission.

Notice however that this type of feedback does not imply collision detection in the strong sense, because nodes would not be able to detect collisions of transmissions of other nodes. Furthermore, when the network under consideration is multi-hop, nodes would only be able to detect a collision at their location. More specifically, consider three nodes A, B and C, connected by two links only AB and BC. If A and C transmit in the same time slot, there will be a collision at B but neither A nor C will be able to detect such a collision even if they are able to receive in the same time slot. This issue is popularly known as the *hidden terminal problem*.

Regarding the information available to protocols, in single-hop networks with collision detection and simultaneous transmission and reception capabilities nodes may store the channel history and the sequence of their own attempts. Protocols that rely in such information are called *full sensing* protocols.

In some networks, nodes may be in different states according with their ability to participate in the distributed computation. When a node is fully functional, i.e., it can transmit, receive and process information, we say that the node is *active*. A node in *standby* mode is assumed that is not able to transmit or process information but it may be able to receive a message, which perhaps will put it in the active mode. Finally, when a node is not able to transmit, receive or process information is said to be *inactive*. Although at this point the later mode seems strange, nodes in Sensor Network applications are assumed to be frequently in this state due to lack of a continuous power supply.

In some randomized protocols, a node transmits a message in time slot t with some probability p_t, independently of the history of transmissions. These protocols are usually called *randomly oblivious* because the sequence of probabilities of transmission is fixed in advance. On the other hand, when protocols change the probability of transmission online, either as a function of their own transmissions or as a function of their *successful* transmissions if that information is available, they are called *randomly adaptive* protocols.

As explained before, in shared-channel networks conflicts among nodes to access the channel are unavoidable and a protocol for contention resolution has to be used. Regarding the arrival of messages to be transmitted, the problem of contention resolution can be addressed under two different assumptions. Namely, either the messages are allocated to nodes before beginning the execution of the protocol or messages are assigned to nodes as they are generated while the protocol is being executed. These scenarios are called *static* and *dynamic* respectively. In both cases the generation and assignment of messages is assumed to be adversarial.

There is also another classification depending on the number of nodes in the network. Although there are very interesting problems in scenarios where there are countably infinitely many nodes, we concentrate here in the case where there are a finite number n of them, although we analyze protocol efficiency as $n \to \infty$.

In the static model, the number of nodes is n and it is assumed that some number $d \leq n$ of nodes are assigned messages before starting the execution of the protocol. In general, it is assumed that all nodes know an upper bound on the number of nodes but they have no information regarding the magnitude of d. This is a reasonable assumption given that, in general, nodes know at least the length in bits of an ID number, information that gives an upper bound on the size of the network, and given that $d \leq n$ due to different startup times or lack of reliability.

Chapter 3

Models

In this chapter, we review models used in Radio Networks that are usually found in the literature. As explained before, not always is the case in Radio Networks that every node is connected directly to every other node. Furthermore, in many cases this connection is not even symmetrical. Therefore, a model for the topology of the network needs to be defined. Also, depending on the application, Radio Networks have very different node constraints. E.g., in some networks nodes have ternary feedback but in others the feedback is just binary. Therefore, a detailed model of the constraints present in the nodes forming the network also is needed. We summarize in this chapter models of topology and node constraints used in the Radio Networks area and in Section 3.2.2 we focus in Sensor Networks describing in detail our harsh Weak Sensor Model [FCFM05]. More details about Sensor Networks classification and taxonomies can be found in [TAGH02].

3.1 Topology Models

Regarding the topology of a network, a well known specification is given by a *directed graph*. A directed graph is a pair of sets $\{V, E\}$, where V is a set of *points* or *nodes* and E is a set of ordered pairs of distinct points taken from V. Any such pair is called an *arc* or an *edge*. In our context, the points model the nodes of the network and the arcs represent the ability to send messages directly (in one hop) from one node to another. If the communication in a network is achieved through wires, an edge AB in the graph represents the link that facilitates the communication from A to B. If on the other hand the communication in a network is wireless, an edge AB in the graph implies that B is in the range of transmission of A. Whenever this relation is symmetric, an *undirected graph* can be used as a model. For example, in a wireless network where all nodes have the same range of transmission, an undirected graph is a suitable model because if a node B is reachable from a node A, A is also reachable from B.

3.1.1 Topology in Radio Networks

The connectivity model widely used in Radio Networks where all nodes have the same range of transmission is the *Geometric Graph* (GG). The specification of a GG includes a pair of sets $\{V, E\}$ and a number $r \in \mathbb{R}^+$. The set of nodes are points in $\mathbb{R}^2$ and an edge $AB \in E$ if and only if A and B are separated by an Euclidean distance of at most r. As mentioned before, the graph is undirected because the range of transmission of all nodes is the same. If this is not the case, more sophisticated models are needed.

There are also some variations of a GG in the literature. When the distance r, modelling the range of transmission is normalized to 1, the graph is called *Unit Disk Graph* (UDG). For cases in which the connectivity beyond some distance $r \in (0, 1]$ is uncertain, there is a generalization of a GG in the literature called *Quasi-Unit Disk Graph* (QUDG). The later model can be extended with a distribution on the probability of being connected when the separation distance is bigger than the uncertainty threshold. Also, the uncertainty threshold can be defined as a function of the angle with respect to some direction of reference for cases where directional antennas are used.

Of course, any of these models can be also extended with node sets in higher dimensional spaces and with threshold distances under different metrics. The particular extension depending on the setting we are modelling. A usual simple extension is to consider the points in $\mathcal{R}^3$ to model the deployment of the network in the real world. Another possible extension is to consider a distribution on the probability of two nodes being connected. Such an extension would imply a combination of the classical random graph model of Erdos and Rènyi [ER59] with a GG. A more appropriate application of randomness to the GG model in the specific area of Sensor Networks is explained in the next section.

3.1.2 Topology in Sensor Networks

In addition to a comprehensive model for the various constraints present in Sensor Networks, a formal model of the potential connectivity of the network needs to be defined. In the past, computer networks have been modelled by means of classical random graphs. Starting in 1959 with a paper by Erdös and Renyi [ER59], the field of random graphs has been widely explored. The classical *Bernoulli* random graph model is denoted as $\mathcal{G}_{n,p}$ where n is the number of nodes and p is the probability of existence of each edge. Random graph models have been used for instance to model the web-graph [ACGL02, KKR⁺99] where the structure of the random graph gives insight into the behavior of the web-graph. However, the classical random graph model is not adequate for the Sensor Network setting because the probability of having an edge AB is either 0 or 1 depending on the Euclidean distance between A and B.

Regarding the deployment of nodes in a Sensor Network, *deterministic deployment*, i.e., the placement of nodes at specific locations, is only possible for small networks in a friendly environment. However, this scenario is not reallistic for most of the intended applications of Sensor Networks where a large area is expected to be covered and the environment is expected to be either hostile or remote. Two models of *random deployment* of nodes are used. In one model, n nodes are assumed to be distributed uniformly at random so that each node is equally likely to fall in any location of the area of interest, independently of the other nodes. The other model is a stationary Poisson point process with intensity n where the number of nodes in disjoint regions is Poisson distributed and mutually independent.

Thus, Sensor Networks are best modelled by *Random Geometric Graphs* (RGG) in $\mathbb{R}^2$ [Pen03]. In the Random Geometric Graph Model $\mathcal{G}_{n,r,\ell}$, n nodes are distributed uniformly at random in $[0,\ell]^2$, and nodes are connected by an edge if and only if they are at an Euclidean distance of at most $r \leq \ell$, the *connectivity radius* (Figure 3.1). The node density depends on the relative values of n,r and ℓ. A specific instance of $\mathcal{G}_{n,r,\ell}$ is a *Random Geometric Graph (RGG)*, also referred to as $G(n,r,\ell)$. A popular instance of this model is $\mathcal{G}_{n,r,1}$ or simply $\mathcal{G}_{n,r}$. Of course, sometimes, a two dimensional model may be inadequate when the terrain in which the sensors are positioned is uneven. In this case an extension to three dimensional random geometric graphs may be needed.

Some properties commonly studied for random geometric graphs within the context of sensor networks are

- *Physical Coverage*: For the region in question, what fraction of the region is covered by balls of radius r, centered on the points thrown randomly into the region with uniform

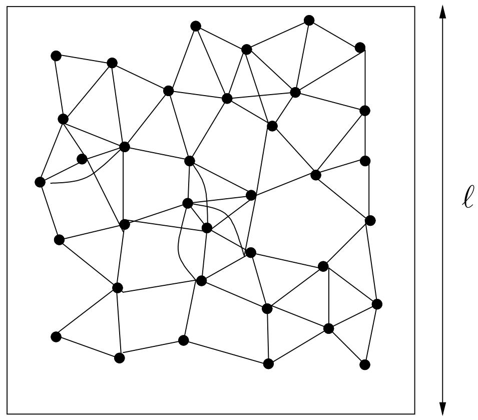


Figure 3.1: A Random Geometric Graph.

distribution? More specifically we are interested in the number of nodes we must throw such that the fraction of the region covered is $1 - o(1)$.

- *Graph Connectivity*: What is the relation among n, r and ℓ when a graph $G(n, r, \ell)$ becomes connected? In keeping with the random nature of the model we say that $G(n, r, \ell)$ is connected when it is connected *with high probability*.

- *Route Stretch*: Given two nodes u, v in a graph $G(n, r, \ell)$, $stretch(u, v)$ is defined as the ratio of the shortest distance between u and v in the graph to the normed distance between the two points in the plane. The *stretch* of $G(n, r, \ell)$ is the maximum of the *stretch* over all pairs of points (u, v) in $G(n, r, \ell)$.

The theory of random geometric graphs is a key tool to study some of the underlying properties in Sensor Networks such as connectivity or coverage. However, the results obtained in this field can not be directly applied to Sensor Networks due to the additional constraints present in them.

3.2 Node Constraints Models

Radio Networks is a vast area and there is a myriad of applications of such a technology, e.g., cellular phones, wireless computer networks, ad-hoc networks, etc. Depending on the specific application the nodes forming the network have very different constraints on their processing and communication capabilities, i.e., range of transmission, life cycle, storage size, etc. In addition to formal models of the topology or the potential connectivity among nodes, an appropriate model of the constraints of the nodes in the network has to be defined, in order to properly design and analyze protocols. We summarize here some of the models used in Radio Networks and Sensor Networks.

3.2.1 Radio Networks

In a seminal paper [BYGI92], Bar-Yehuda, Goldreich and Itai presented a formal model of a radio network that specifies many of the important restrictions on sensor nodes, including, e.g., limits on contention resolution, but they make no mention of computational limits, such as small memory. More precisely, the model consists of an arbitrary multihop undirected network. The nodes are assumed to be locally synchronous, i.e., they all have the same clock frequency but perhaps different starting times. Each node either receives or transmits within each time slot, but not both. A node receives a message successfully in a time slot if exactly one of its neighboring nodes transmits in that time slot. If more than one neighboring node transmits in the same time slot, the messages are garbled and the node receives noise. It is not possible to detect collisions, hence, a node can not distinguish the case in which no neighboring node transmits from the case in which more than one transmit in the same time slot. The topology of the network is not known a priori. The main difficulty in this model, as well as in most of the models in Radio Networks, is the possibility of message collision, therefore, any protocol for this model has to include contention resolution in order to be useful.

After this model was introduced, some papers [NO00a, KMPS04] have added more restrictions, although often such restrictions are implicit in the text or algorithms rather than fully specified. In the following section we elucidate a complete and comprehensive model for Sensor Networks.

3.2.2 The Weak Sensor Model

As explained before, nodes in Sensor Networks are designed with the goal of obtaining a device as small as possible and at a very low cost. Therefore, sensor nodes have very harsh constraints in each of its main capabilities, processing, communication and sensing. These strong constraints are the main reason why problems in Sensor Networks are challenging, because the typical solutions utilized in computer networks are not suitable in such a harsh scenario. Therefore, in order to approach any problem in Sensor Networks, and in addition of formal models of the connectivity of the network, a formal model of the various sensor node constraints has to be defined.

There is a vast body of literature on sensor networks, however most of it does not sufficiently handle all aspects of the problem. All random geometric graph results related to ad-hoc wireless networks require $\omega(1)$ degree (see e.g. [MP05]). All proposed protocols for sensor network formation include some inappropriate hardware assumptions. For example, the sensor network formation protocol in [SWLF04] builds a constant-degree network, but relies on positional information hardware. The protocol proposed in [BLRS03] also builds a constant degree network, but relies on the preëxistence of a scheme for channel-contention resolution. The different models implicit in such results are inadequate and do not reflect all the limitations under which sensor nodes operate.

Given the various limitations of sensor nodes and the absence of a reliable communication structure after deployment, any sensor network protocol must work under difficult conditions. In this section, we specify the formal Weak Sensor Model that summarizes the literature on sensor node restrictions, taking the most restrictive choices when possible. The protocol for Sensor Network formation presented in Chapter 6 is designed under this model, whereas the lower bounds in Chapter 8 are proved for the more general model of Radio Networks.

- MEMORY SIZE: Sensor nodes may have limited memory size. In fact, asymptotically speaking, if we assume that the memory size is any function in $\omega(1)$ we would be assuming that nodes can have a memory of infinite size. Therefore, in the Weak Sensor Model sensor nodes may store only a constant number of $O(\log n)$ bit words.

- SHORT TRANSMISSION RANGE: Due to costs and size restrictions, sensor nodes may not have a large range of transmission. Consequently, not all nodes are reachable from a given node leading to the well known hidden-terminal problem. This limitation has an impact on the density of sensor node deployment.

- DISCRETE TRANSMISSION RANGE: Some of the extant literature [SWLF04] assumes that nodes can vary their power of transmission. However, assuming that any number of levels can be reached is unrealistic–in particular to analyze the asymptotic behavior of the algorithm. In this model, sensor nodes can adjust their power of transmission to only a $O(1)$ number of levels.

- ONE CHANNEL OF COMMUNICATION: although it is assumed in some papers that $\omega(1)$ channels are available in order to avoid collisions, this assumption is unrealistic–specially in order to analyze the asymptotic behavior of protocols. We constraint the number of channels of communication to exactly one.

- LOCALITY: Sensor nodes are distributed over a large area and may not be reachable by a central controller. Hence, each sensor node must be capable of configuring itself automatically.

- LOW-INFORMATION CHANNEL CONTENTION:

  - SHARED CHANNEL OF COMMUNICATION: Given that this is a Radio Network and that there is only one channel available, the communication with neighboring nodes is through broadcast in a shared channel.

  - CONTENTION-RESOLUTION MECHANISM: If more than one message is placed on a multiple-access communication channel at the same time, a collision occurs and no message is delivered. Hence, sensor nodes have to implement a contention-resolution mechanism to access the channel.

  - NO INITIAL INFRASTRUCTURE: Right after deployment, the nodes of a Sensor Network have no communication infrastructure available (MAC layer). Therefore, before that any exchange of information can be carried out, nodes have to self-organize a medium access scheme bringing structure to the network.

  - NO COLLISION DETECTION: Although in many Radio Networks it is possible to detect a collision, it has been also argued that a collision can not be detected in the presence of noisy channels [BYGI92]. In this model, only two channel states are feasible, single transmission and silence/collision. This scenario is popularly known as *binary channel* or it is said that nodes have *binary feedback*.

- – NON-SIMULTANEOUS RECEPTION AND TRANSMISSION: A sensor node may not be able to receive while transmitting because, in its vicinity, its own signal overwhelms any signal transmitted by other nodes. Therefore, transmitters also cannot detect collisions.

- ASYNCHRONICITY: No global clock or other synchronizing mechanism is assumed, but all sensor nodes have the same clock frequency. We assume that time is divided into *slots*. The use of a slotted scenario instead of a more realistic unslotted one was justified in [Rob75], where it was shown that they differ only by a factor of 2 because a packet can interfere in no more than 2 time-slots. This type of synchronicity is usually called *local synchronism*.

- LIMITED LIFE CYCLE: Sensor nodes may be powered by sources such as solar energy. These sensors may go down from time to time to recharge. This necessitates simpler and fast computations and energy-efficient protocols.

- NO POSITION INFORMATION: Due to cost and size restrictions, sensor nodes may not have position information obtained using a global or local positioning system, directional antenna or other specialized hardware.

- ADVERSARIAL NODE WAKE-UP SCHEDULE: Given that the sensor nodes are deployed over large areas and given the lack of a centralized controller, we can not expect all sensor nodes to start the execution of protocols in the same time slot. Therefore, in order to analyze these protocols in a worst case scenario, we assume the existance of an adversary that determines the wake-up schedule.

- UNRELIABILITY: In addition to the lack of guarantees of a constant power supply, due to low cost sensor nodes are unreliable. Hence, sensor network protocols have to be designed to be robust in the case of failures of one or more sensors.

Chapter 4

Problems

4.1 Radio Networks

We summarize in this section some of the problems commonly studied within the Radio Networks area. Although these problems emerge from different motivations, they are all due to the main constraint present in Radio Networks, the communication channel is shared among many nodes. The models under which these problems are studied include various assumptions and constraints, the strongest being the harsh Weak Sensor Model detailed in Section 3.2.2. Of course, protocols designed under stronger models are more robust, however, weaker models are valuable in certain applications in order to achieve more efficient solutions. In any case, all models include the following assumption due to the shared use of the communication channel: *In Radio Networks, a node receives successfully a message in a time slot if and only if exactly one of its adjacent neighbors has transmitted in that time slot. If many neighbors send messages simultaneously the messages are received garbled.*

A fundamental problem in Radio Networks is to achieve a successful, i.e. non-colliding, transmission of at least one node. If the network is a single-hop one, this problem is called in the literature the *selection* problem. However, in multi-hop networks we will distinguish two variants of this problem. If in a given time slot exactly one of the adjacent neighbors of a node transmit, we say that there was a *clear reception* in that time slot. Whereas, in the case where a node transmits a message in a given time slot, and no other node within two hops of the transmitter transmits in the same time slot, we say that there was a *clear transmission*. Of course, in a one-hop network both problems are identical. Due to lack of global synchronization nodes may become active at different times. In order to analyze protocols for clear transmissions in a worst case scenario, the nodes-startup schedule is assumed to be adversarial. The time complexity of protocols for clear transmissions is computed from the time that the first node becomes active until the time slot in which the clear transmission is achieved. More details on these problems can be found in [Wil86, BYGI92, KM98, KG85, Mar94, GW85, CMS01, Kow05].

In the dynamic model messages to be transmitted are allocated to nodes as the protocol is executed. Therefore, in addition to define the number of participating nodes n, the way that messages arrive has to be specified. One popular model is to assume that in each time slot a new message arrives to a node with some probability p, independently of other nodes and time slots – a Bernoulli process. In this case, each node has a queue of messages to be transmitted. In the dynamic model the most important problem is to give a *stable* protocol, i.e., a protocol that schedules transmissions so that no queue overflows.

The problem of stability under dynamic allocation of messages has originated a very active area of research called *Adversarial Queueing Theory* (AQT). The introduction of an adversarial model in communication networks is due to Cruz [Cru91a, Cru91b]. In this model, each packet to be transmitted belongs to a *session* and has a predefined route. The adversary gets to define what are the active sessions in the network, what are their routes, the arrival rate and the burstiness. The AQT model was later introduced in [BKR+01]. In this model, the network traffic is not grouped in sessions but still the adversary gets to define the route to be followed by a packet and the arrival rate. Therefore, nodes can only choose a scheduling policy. While in the model of Cruz links with different delays and packets of different sizes were allowed, in the AQT model this is not possible since the system is assumed to be synchronous. More recently, Calzada, Fernández, López, Martínez and Santos in [CFL+04] have proposed a generalization of the AQT model to allow links with different delays and packets of differents sizes. The model proposed has been termed *Continuous Adversarial Queueing Theory* (CAQT).

A problem widely studied in Radio Networks is *broadcast*. In one-hop Radio Networks the broadcast problem is as follows. A non-empty set of *source* nodes is allocated a message to be transmitted and the message has to be delivered to all the other nodes in the network. Given that it is a single-hop setting, one non-colliding transmission is enough to solve the problem. In multi-hop Radio Networks broadcast goes beyond nodes sending a message to their neighbors. Since some nodes might not be in range of any source node, the solution of this problem relies in message forwarding. There are also a few well-studied variants of the broadcast problem. If the source nodes expect to receive an acknowledgement the problem is called *broadcasting* *with acknowledgement*. If different source nodes are allocated different messages and all nodes have to receive all messages for the problem to be solved the problem is called *gossiping* or *all-broadcast*. An important constraint of the channel for the gossiping problem is what is the maximum length of a message that can be transmitted in one time slot. Although for most of the problems in Radio Networks a message is supossed to have a length of $O(\log n)$ bits, in some literature related to the gossiping problem it is assumed that links have enough capacity as to allow many messages to be combined. In the broadcast problem it is customary to assume the existance of an adversary that gets to decide which is the set of source nodes. Interesting results and further details on the broadcast problem can be obtained from [BYGI92, KM98, BYII93, BP97, CGOR00, CGR00, CGG$^+$02, CMS01, CR03, KP03, KP04b, KP04a, LP02].

Another problem of interest in Radio Networks is the *leader election* problem. In the leader election problem the participating nodes have to choose a leader among themselves. That is, at the end of the protocol exactly one node has status of leader and all the other nodes have status of non-leader and know the identity of the leader. Leader election is a central problem in Radio Networks because many other more complex problems rely on the existance of a leader or distinguished node in the network. In the deterministic case the lack of ID numbers makes this problem unsolvable so it is customary in the literature to assume that nodes have ID numbers. For randomized solutions it is common to assume that no ID numbers are available therefore an initialization phase assigns different ID numbers to every node. The leader election problem has been studied in the scenario where the number nodes is known and where it is unknown. Like for the broadcast problem, the leader election problem in a single-hop network is reduced to achieve a succesful transmission. The node that achieves such a transmission first becomes the leader. However, in the multi-hop setting other ways of breaking the symmetry are necessary. For more details on the leader election problem, we refer the reader to [HNO99, JKZ02, NO02a, NO00b, NO02b, KR03, BIN06].

A fundamental problem in Radio Networks is then to establish a mechanism of accessing the shared communication-channel in order to avoid or efficiently resolve contention. The problem is easier to resolve if the nodes are permanently active. However, due to energy constraints, it is often desirable to let the nodes become inactive in the communication channel whenever there is no ongoing communication. In other words, after some session of communication is completed, nodes should enter a *standby* or *sleep* state and stop being alert until a new session begins. Unfortunately, this mechanism entails the loss of synchronization, if there was any. The problem of regaining synchronization or induce a change to an active state to all nodes in a Radio Network after entering a sleeping mode is usually called the *wakeup* problem. More specifically, the wakeup problem is defined as follows. Initially, all nodes are in sleep mode. Any node can wake up either spontaneously or by receiving a wake-up signal from an awake node. Nodes can wake up spontaneously in any time slot. Once a node is in awake mode, it starts running the wake-up protocol in order to wake up other nodes that are still in sleep mode. The problem is solved after all nodes in the network are in awake mode. The time complexity of a protocol that solves this problem is computed from the time slot that the first node wakes up spontaneously until the time slot when the last node is woken up. Assuming that not all nodes wake up spontaneously, otherwise there is no problem to solve, the wakeup problem in one-hop Radio Networks is the same as the clear transmission problem. However, in multi-hop Radio Networks, similarly to the leader election and broadcast problems, one clear transmission is not enough. An interesting observation is that the wakeup problem can be seen as a generalization of the broadcast problem. That is, if we have a protocol that solves the wakeup problem we can solve the broadcast problem by waking up spontaneously the source nodes only in the first time slot. More details and relevant results for the wakeup problem can be found in [CGK04, CK04, GPP01, Ind02, JS05].

4.2 Sensor Networks

We concentrate now in the most stringent setting in Radio Networks, the Sensor Network. Although the problems here described are also apply to other Radio Networks, they are particularly challenging in Sensor Networks.

Even though communication among sensor nodes in a Sensor Network is through radio broadcast, it is useful to set up explicit links between nodes in order to establish routing paths and prevent flooding. A Sensor Network is capable of achieving large tasks through the coördinated effort of sensor nodes, but individual nodes have severe limitations on memory size, life cycle, range of communication, etc. We specify these and other limitations in the

Weak Sensor Model in Section 3.2.2. Given these difficult conditions, i.e. a group of weak sensor nodes deployed in a geometric random distribution, a natural question is how to organize such a network. Therefore, the *Sensor Network Bootstrapping* problem, also called *Sensor Network Initialization*, *Sensor Network Formation* or *Topology Control* in some literature, has emerged as the most fundamental problem in the Sensor Networks area. Any sensor network initialization algorithm must be fast and distributed, and must resolve channel contention issues. The network constructed by such an algorithm must be connected and, according with the Weak Sensor Model, must have low degree and diameter. The limitations on individual sensors nodes make this problem non-trivial, and its adequate resolution is crucial for making sensors useful.

There are two main types of issues in sensor network formation: those relating to geometric properties and those relating to network protocols; and any solution achieved for either must be compatible with an accurate model of sensor nodes. On the one hand, coverage and connectivity in sensor networks are dependent on the distribution of nodes in an area and the range of transmission of each node. Additionally, the density of nodes in an area determines the minimum path length between any two nodes in the induced connectivity graph. The limited range of transmission makes these properties geometric. On the other hand, protocols for sensor network formation are limited by the fact that sensor nodes share a common channel of communication and that they do not typically have access to directional or positional information. Memory limitations in sensor nodes also impose the restriction that a node can only keep track of $O(1)$ neighbors.

As it is customary in Radio Networks, the analysis of protocols for Sensor Network initialization is done under the assumption of the existance of an adversary that knows the protocol and gets to define which nodes are active in each time slot. Given the unreliability and the lack of a constant power supply that characterizes the sensor nodes, the time complexity can not be analyzed for the network as a whole but for each individual node. I.e., if we define the running time of a protocol as the time that takes to establish all links among nodes after the first node becomes active, an adversary could potentially define a wake-up schedule that would make any protocol never build the network. Instead, the time complexity is analyzed for each node. In other words, for any node i, the running time of a Sensor Network initialization protocol is the time that takes for i to join the network after it becomes active, unless i is turned off. Given the energy constraints of a sensor node, protocol efficiency is sometimes analyzed in terms of the energy consumed. It is well known that the transmission cost is a polynomial of degree $\alpha \geq 2$ in the distance covered. In the most general setting where nodes have more than one range of transmission r available, a frequent metric of the cost of sending a message between two nodes is to consider the sum of the energy cost of each hop in the path between them. If we consider the total energy consumed in the network as the sum over all links in the network, it has been proven that to find the optimal subgraph of the connectivity graph that minimizes the energy cost is NP-hard [KKKP00]. More details on the sensor network initialization problem can be found in Chapter 5.

The other key problem in Sensor Networks is *routing* messages through the network. In fact, the meaning of routing messages in Sensor Networks differs according with the application. Some Sensor Networks are designed to monitor some physical variable such as temperature. In this application nodes may transmit continuously their measurements or only transmit whenever a significant change is observed. In any case, there are distinguished nodes in the network called *sinks* and all nodes forward their messages to them. In other applications nodes may accept queries from any node in the network which might identify itself with its ID or even with its geographical position.

Due to the harsh restrictions under which sensor nodes operate, solving the routing problem using the techniques frequently used in communication networks is not possible. Routing is particularly challenging when the following two restrictions are present at the same time, constant memory size and lack of position information. Under the first restriction is not possible to maintain routing tables. Under the second one is not possible to use geographic routing. Therefore, in some settings the lack of position information constraint is relaxed to *lack of accurate position information*. In this case, it is assumed that nodes can *estimate* distances based on measurements of transmission delays.

The low cost and low reliability requires in general redundant deployment of sensor nodes. Therefore, the importance of the idendity of a node is reduced compared to traditional communication networks. In this case the routing paradigm used is called *data-centric*. In the data-centric paradigm the objective is the retrieval of data keyed on an event where the identity of the nodes or their position is irrelevant. An event is an abstraction that may model a set of sensor measurements, the frequency or speed of change of a variable, etc. We refer the reader to [ASSC02, KW03, PB03, EGHK99, GGSE01, IGE00, BE02] for more details about routing in Sensor Networks.

Another problem of interest in Sensor Networks is *positioning*. Although in most of the applications specialized hardware for positioning is not available due to cost, sometimes the absolute position of a node is not necessary and its relative position with respect to its neighbors is enough. Furthermore, in some applications may be enough to determine the topology of the underlying graph because this information gives an upper bound on the distances between any pair of nodes. In general, no matter what is the method used to determine position, errors are introduced and the resulting distance matrix is not embbedable even in three dimensions. The problem of embbeding a distance matrix in the minimum number of dimensions is a vast area of research in engineering and other areas and it is usually called *rigidity* [JJ03, Sax03, LY82, Hen92, Lam70].

There are a number of closely related well-known graph problems whose efficient solution would be a promising approach to give structure to a Sensor Network. Namely, *Clustering*, *Dominating Set*, *Maximal Independent Set* (MIS), *Vertex Coloring*, *Edge Coloring*, etc. Unfortunately, many of the solutions included in the literature for these problems [WAF04, Bas99, ACS94, Lub86, FPS02] assume an underlying medium access scheme or contention resolution mechanism.

Chapter 5

Survey: the Sensor Network Initialization Problem

Given that Sensor Network initialization is a fundamental problem and its solution is non-trivial due to all the constraints present in sensor nodes, we concentrate in this chapter in the Sensor Network initialization problem. The extant literature related to Sensor Networks is vast and includes both theoretical and empirical research work. Although many of the solutions proposed do not sufficiently handle all the aspects of the problem, we summarize here some of the most relevant results for Sensor Network initialization and related problems. This chapter is not intended to provide a *full* overview of the existing body of work in Sensor Network initialization–a task beyond the scope of this thesis, but to give some detail on previous work before we give an optimal initialization protocol in Chapter 6.

A protocol called *Self-Organizing Medium Access Control for Sensor Networks* was presented in [SGAP00]. This protocol builds a flat topology with no local or global masters. The model for this protocol is as follows. Due to the short transmission range of the sensor nodes, it is assumed that a reception consumes the same energy as the transmission. Therefore, nodes can not have their radios on permanently. There are enough channels as to accomodate each link among neighbors in a different frequency in order to avoid collisions. Furthermore, the number of available channels is assumed to be big enough so that if two nodes choose a channel at random the probability of choosing the same channel is low. Another key assumption is that nodes have memory of size $\Omega(\Delta)$ where Δ is the maximum degree in the connectivity graph. Finally, nodes are assumed to start running the protocol (wake up) at random times under some distribution such that the probability of two nodes being synchronized is low.

Under these assumptions, the protocol works as follows. Upon waking up, nodes start running a discovery phase. The discovery phase starts with a listening period. If a node A receives a discovery message from some neighboring node B, A sends a reply message with its current schedule of transmissions/receptions. Then, the node B finds a time slot available in the schedules of both and sends back to A that information. After the discovery phase nodes enter the normal communication phase. In the communication phase, each node repeatedly follows its own schedule of transmissions/receptions so that from now on the system is synchronous.

The discovery phase length is chosen according with the application and it is assumed to be long enough so that the probability of not discovering a neighboring node is small. Also, collisions will not occur because all links are established in different channels. If two nodes do not find a time slot available in their schedules during the discovery phase, simply give up and the link is not established. The period of the communication phase is chosen according with the application and it is long enough as to handle most of the links of a node and the probability of having an isolated node is low. This work is empirical and does not include any running time or energy consumption analysis. However, as we will see in the analysis of the protocol below, the inclusion of a phase long enough to ensure a low probability of collisions introduces a factor of at least $\Omega(n^\gamma \log^2 n), \gamma > 0$ in the overall running time of the protocol. The requirement of $\Omega(\Delta)$ different channels and memory size makes it infeasible from a theorical standpoint.

The protocol *K-Neigh* [BLRS03] builds a network where every node has at most k neighboring nodes, where k is tuned to ensure connectivity w.h.p. The model under which this protocol works includes the following assumptions. Nodes are deployed in the plane uniformly at random. Although the transmission power can be adjusted, all nodes are constrained to the same maximum P. P is a function of n and it is chosen so that the network is connected w.h.p. The protocol also relies in some distance estimation mechanism such as measuring the radio signal strength received or comparing the time of arrival of different kinds of signals. Given that information about all neighboring nodes is collected, the memory size is assumed to be in $\omega(1)$. Although the synchronization is local, the difference between node wake up times is upper bounded by a constant Δ. The protocol is as follows.

Upon waking up at time $t_i \in [0, \Delta]$, node i waits for Δ time slots and chooses a time slot within the next δ time slots to transmit its ID number at the maximum power. δ is chosen big enough so that the probability of collision is low. From every message received, i stores the ID number and the estimated distance to that node. At time $t_i + 2\Delta + \delta$, i ends its discovery phase, sorts the list of neighbors by distance and selects the k closest discarding the rest.[1] At a time slot randomly chosen among the next δ steps, node i broadcasts its list at maximum power. Based on the lists received during the next Δ time steps, at time $t_i + 3\Delta + 2\delta$ node i removes from its list all nodes that do not include i in their lists, and adjust its transmission power to the power needed to reach the farthest node in its list.

It is easy to see that the total running time of this protocol is $4\Delta + 2\delta$. Given that Δ is assumed to be a constant, the dominating factor is δ which is tuned to guarantee that the probability of collisions is low. The probability can be bounded as follows. If d is the number of nodes in a one-hop neighborhood, the probability of not having a collision is

$$\begin{aligned}
Pr &= \prod_{i=1}^{d-1} \left(1 - \frac{i}{\delta}\right) \\
&\approx \prod_{i=1}^{d-1} e^{-i/\delta} \\
&= exp\left(-\sum_{i=1}^{d-1} \frac{i}{\delta}\right) \\
&= exp\left(-\frac{d(d-1)}{2\delta}\right) \\
&\approx exp\left(-\frac{d^2}{2\delta}\right) \\
&\in \Omega\left(1 - \frac{1}{n^\gamma}\right), \gamma > 0 \text{ for } \delta \in \Omega(d^2 n^\gamma).
\end{aligned}$$

It was proved in [XK04] that in an RGG the minimum number of neighbors needed to ensure connectivity w.h.p. is $d \in \Omega(\log n)$. Therefore the overall running time is $\Theta(n^\gamma \log^2 n), \gamma > 0$.

As in the previous protocol, the assumption of an $\omega(1)$ memory size makes this protocol infeasible asymptotically speaking. Also, the assumption of having the capability of adjusting the power of transmission to *any* level is too strong. In particular, given that the energy cost is at least quadratic in the distance of each link, the optimal path might include sub-constant distances.

[1] Although the authors included here an aditional time delay for the computation of the list, for the sake of clarity we do not take it into account based on the standard assumption for analytical purposes that in Radio Networks nodes have unbounded processing power.

An energy efficient topology control scheme called *OrdYaoGG* was presented in [SWLF04]. The assumptions of this model are as usual n nodes distributed uniformly in the $\mathbb{R}^2$ plane, each node with a maximum transmission range normalized to 1. Therefore, the connectivity graph is a UDG. All nodes have different ID numbers and each node knows its position information by means of a GPS or other specialized hardware such as a directional antenna and signal strength measurement capabilities. At a minimum, the assumption is that every node knows in advance or will collect the position information of all its neighbors. Therefore, the memory size is assumed to be in $\omega(1)$. Global synchronization is also necessary given that the proposed algorithm works in phases. Finally, an underlying contention resolution mechanism is assumed in order to collect information of neighboring nodes.

The topology obtained by this protocol has optimal power stretch, constant degree and it is planar. The power stretch is defined to be the ratio between the energy cost of a path connecting two nodes in the subgraph of the connectivity graph, to the cost in the optimal path in the connectivity graph. Obtaining a planar topology is a requirement of many routing algorithms to guarantee message delivery.

A brief description of the protocol can be given as follows. The algorithm consists of three phases. The first phase constructs a Gabriel subgraph of the connectivity graph. In a Gabriel graph, two nodes u and v are conected by an edge uv if and only if there is no other node in the circle of diameter uv. A Gabriel subgraph has been proven to have optimal power stretch. However, the degree can be as big as $n-1$ and one of the goals of the protocol is to obtain a constant degree topology. The second phase establishes an ordering among neighbors in the Gabriel graph, and in the third phase the node with the higher priority among its neighbors in the ordering splits the neighborhood in a constant number of slices and chooses the closest neighbor in each slice. Therefore, the resulting topology has constant degree.

Once again, as in the previous protocols, assumptions such as non-constant memory size or non-constant number of levels of transmission power, make this protocol infeasible from a theoretical perspective. Also, in practice, it is assumed in general that specialized hardware such as GPS or directional antennas is too costly for this application. Regarding the running time analysis, we omit the details here since the protocol does not include any contention resolution mechanism.

More general information about sensor networks can be obtained from the surveys [RMSG, ASSC02, KW03, PB03, YKR06].

Chapter 6

Bootstrapping a Hop-optimal Network in the Weak Sensor Model

The most fundamental problem in Sensor Networks is to efficiently establish the network among sensor nodes from scratch right after deployment. As explained before, its solution is non-trivial due to all the constraints present in sensor nodes. Previous work does not sufficiently handle all aspects of the problem because all solutions include some strong assumptions that yield to an inaccurate efficiency analysis or even make some solutions non-implementable in sensor nodes. The protocols detailed in this chapter are the first network initialization algorithms that are implementable on sensor nodes.

Problems in sensor networks are twofold: related to geometric properties and related to network protocols. Sensor nodes have limited range and are deployed at random over a large area. Therefore, geometric properties like path length, coverage and connectivity need to be understood. Thus, sensor networks are modeled as RGGs. On the other hand, node limitations, shared communication channel and lack of additional infrastructure at deployment impose limitations on network protocols. We specified these and other limitations in Section 3.2.2 where the harsh Weak Sensor Model is defined.

Until recently, the existing literature on sensor network initialization did not sufficiently handle all aspects of the problem. All random geometric graph results related to ad-hoc wireless networks required $\omega(1)$ degree (see e.g. [MP05]). All proposed protocols for sensor network formation included some inappropriate hardware assumptions such as the availability of positional information hardware [SWLF04] or the preëxistence of a scheme for channel-contention resolution [BLRS03]. The different models implicit in such results are inadequate and poorly reflect the various limitations under which sensor nodes operate, and indeed, there seems to be considerable confusion in the literature as to what are or are not reasonable assumptions about the capabilities of sensor nodes.

Sensor Network initialization research has three parts: (i) to specify a comprehensive model that captures all the restrictions present in sensor nodes; (ii) given that under those restrictions is not possible to establish all the links of the connectivity graph, to show that there exists a subgraph of the connectivity graph that would make a connected network without asymptotically increasing the cost of delivering messages; and (iii) to give a fast distributed protocol that works under the constraints of the specified model. We have already adressed the first part in Section 3.2.2 where the Weak Sensor Model is detailed. We concentrate in this chapter in the other two parts. Given the formal Weak Sensor Model, we show that a good Sensor Network must have constant degree and low *hop-stretch*. We also show that any appropriate RGG has such a subgraph *with high probability*. In other words, given any connected RGG, we show that, with high probability, there exists a subgraph, where the path length between any two nodes in terms of number of edges is asymptotically optimal even if the degree is restricted to a constant number of neighbors. Finally, we also give a $O(\log^2 \ell)$ localized algorithm that builds the network modelled by such a graph, under the Weak Sensor Model.

Throughout this chapter, our node constraints model is the Weak Sensor Model detailed in Section 3.2.2 and the potential connectivity of the nodes is modelled by a random geometric graph. As explained in Section 3.1.2, the deployment of nodes in a random geometric graph can also be interpreted as a Poisson process in the plane where the number of points in $[0, \ell]^2$ is given by the Poisson distribution with mean n. In our proofs, we assume the uniform deployment, i.e., each of the sensors is equally likely to fall at any location in $[0, \ell^2]$ independently of the other sensors, although the results hold for the Poisson model as well with almost no change in the proof techniques.

6.1 Related Work

The Sensor Networks area is very active and includes a vast body of theoretical and empirical research work impossible to completely include here. Before describing the hop-optimal bootstrapping protocol, we summarize in this section some of the most closely related work. The details of some of this work are given in Chapter 5.

6.1.1 Threshold properties in $\mathcal{G}_{n,r}$ and $\mathcal{G}_{n,r,\ell}$

Gupta and Kumar [GK98], in a seminal paper in the field of random geometric graphs computed the minimum radius needed to obtain a large connected component with high probability. This and other results [Pen03] give us a critical radius such that each node will have many neighbors. Of course, sometimes, a two dimensional model may be inadequate when the terrain in which the sensors are positioned is uneven. In this case an extension to three dimensional random geometric graphs may be needed.

In the $\mathcal{G}_{n,r,\ell}$ model, tight thresholds for connectivity, coverage and route stretch, were shown by Muthukrishnan and Pandurangan [MP05] using an overlapping dissection technique called bin-covering. More recently, Goel, Rai and Krishnamachari [GKR04] showed that in fact all monotone graph properties have a sharp threshold for random geometric graphs. Other properties of random geometric graphs such as vertex degree or k-connectivity were studied in [AR97a, AR97b, Pen99].

6.1.2 Sensor Networks initialization

A protocol for bootstrapping sensor networks was presented in [SGAP00]. In order to avoid collisions, the number of channels needed is a function of the density, which makes it infeasible. A network formation protocol, where node degree k is a constant tuned to ensure connectivity w.h.p., is given in [BLRS03]. This protocol relies on expensive distance estimation hardware such as GPS. Recently, an energy efficient topology control scheme was presented in [SWLF04]. This algorithm requires the use of a directional antenna and distance estimation hardware. In all these schemes, no contention resolution mechanism is given, and $\omega(1)$ memory size is assumed. Refer to Chapter 5 for further details.

6.1.3 Bluetooth

Bluetooth [BS01, Blu, MB00],which also limits the local connectivity of nodes, is a local area wireless technology designed to enable voice and data communication between various electronic devices. In these networks the nodes have less restrictive constraints (like power supply, range of transmission, memory capacity, etc.) than in sensor networks. In Bluetooth, a group of devices sharing a common channel is called a piconet. Each piconet has a master unit that selects a frequency hopping sequence for the piconet and controls the access to the channel. Other participants of the group known as slave units are synchronized to the hopping sequence of the piconet master. The maximum number of slaves that can simultaneously be active in a piconet is seven. A slave in one piconet can be a master or slave in another piconet. Piconets can also be interconnected via bridge nodes to form a bigger ad hoc network known as a scatternet.

There has been considerable work on schemes for the formation of scatternets. Barrière et al. [BFNO03] proposed a distributed construction technique for Bluetooth scatternets of low degree and fixed diameter. This technique is useful even in the dynamic case where nodes are assumed to come alive and drop dead from time to time. However this technique is restricted to networks where all nodes are within transmission range of each other and hence is unrealistic for the purpose of sensor network formation. Salonidis et al. [SBTL01] earlier proposed an algorithm for constructing scatternets, but this technique suffers from the same limitations as above and further is restricted to 32 nodes and static layout. Schemes proposed for scatternet formation in [LS01, SBTL01, WTH02, ZBC01, FMPC04] are designed to work in the more general case where all nodes may not be within transmission range of each other. Techniques proposed in these are strictly heuristic or do not fit in the *weak sensor model*.

6.1.4 Cellular Systems

There are various reasons why medium access control protocols used in cellular systems can not be used in Sensor Networks. In a cellular system, mobile nodes are a single hop away from distinguished nodes called *base stations* and the base stations form a wired backbone. The primary goal of a medium access scheme in a cellular system is to guarantee quality of service and efficient bandwith use, but power efficiency has a secondary rol given that the base stations have constant power supply and the users can replenish the batteries of the mobile nodes. In Sensor Networks there is no central control such as a base station and power efficiency dominates the life cycle of the network, therefore existent solutions for cellular systems can not be applied.

6.2 Geometric Analysis of Sensor Networks

Recall that sensor nodes may only set up links with a constant number of neighbors, a consequence of the memory size limitation in the Weak Sensor Model, and since sensor nodes are distributed uniformly at random, the potential connectivity relation defines a *Random Geometric Graph* (RGG). Hence, any protocol for network formation must set up links defining a constant-degree spanning subgraph of the RGG. However, ignoring potential links may result in an increase in path lengths in the subgraph. This increase in path length can be measured in two ways: in terms of increase in the number of *hops* or increase in route stretch.

In applications where the propagation delay is significant, route stretch is an appropriate measure of optimality. However, sensor networks have small inter-node distances, and propagation delay is low. One of our primary concerns in the Weak Sensor Model is that we should minimize energy consumption at each node so as to maximize the life cycle. Thus, a Sensor Network is optimal when it minimizes the number of transmissions, which is to say, minimizes the number of hops in each path, rather than the weighted path length. Note that schemes have been proposed that attempt to minimize energy consumption [SWLF04], and these favor many short hops over a few long ones. However, any such scheme requires an $\omega(1)$ number of transmission power levels and, furthermore, ignores the contention resolution overhead of the extra hops. A formal definition of stretch in terms of hops follows.

Let the *length* of a path connecting two nodes in a given graph be the number of edges of such a path. Let $d_{min}(u, v)$ be the shortest path between two nodes u and v in the RGG $G(n, r, \ell)$. Let $D(u, v)$ be the Euclidean distance between u and v in the plane. Note that in $G(n, r, \ell)$, $\lceil D(u, v)/r \rceil$ is a lower bound on $d_{min}(u, v)$. Call this lower bound, $d_{opt}(u, v)$. The *hop-stretch* of (u, v) is defined as the ratio $d_{min}(u, v)/d_{opt}(u, v)$. The *hop-stretch* of $G(n, r, \ell)$ is the maximum of the *hop-stretch* of (u, v) over all pairs of points (u, v) in $G(n, r, \ell)$. In the rest of this section we will outline a scheme to obtain a constant degree hop-optimal subgraph from a sufficiently dense random geometric graph.

6.2.1 Disk Covering Scheme for Sensor Network Formation

The Disk Covering Scheme presented in this section shows the existence of a bounded degree, bounded stretch subgraph of a RGG. The description and analysis of a distributed algorithm is presented in later sections 6.3.1,6.3.2 and 6.3.3. Before describing the scheme, we introduce some necessary terminology.

Definition 1. *A* Random Geometric Graph *or $G(n, r, \ell)$ is an instance of $\mathcal{G}_{n,r,\ell}$ where r is the connectivity radius.*

Given a sufficiently dense $G(n, r, \ell) = \langle V, E \rangle$, the goal of the disk covering scheme is to produce as output a spanning subgraph $\langle V', E' \rangle$ such that $V' = V$, $E' \subset E$, the maximum degree is bounded by a constant and the path length is asymptotically optimal. The precise nature of the path length optimality is given in the proof of Theorem 11.

Definition 2. *The graph obtained as a result of the disk covering scheme is called the* Constant-degree Hop-optimal Spanning Graph *(CHSG)*

The following definitions will be relevant here and their meaning will be clear after the disk covering scheme is fully described.

Definition 3. *All nodes covered by the same disk at the end of the disk covering scheme are connected to each other in the RGG and will be referred to as a* **disk-clique***.*

Definition 4. *Some (possibly all) of the nodes covered by the same disk at the end of the disk covering scheme are connected to each other by a spanner in the CHSG and will be referred to as a* **disk-spanner***.*

Definition 5. *A* **bridge** *is a node, lying at the center of a disk, that is designated to communicate between two or more disk-cliques.*

The following pseudocode summarizes the Disk Covering Scheme. a and b are tunable parameters that affect the maximum degree and hop-stretch of the CHSG. Figure 6.1 illustrates this protocol.

1. Add all nodes from the RGG to the CHSG.

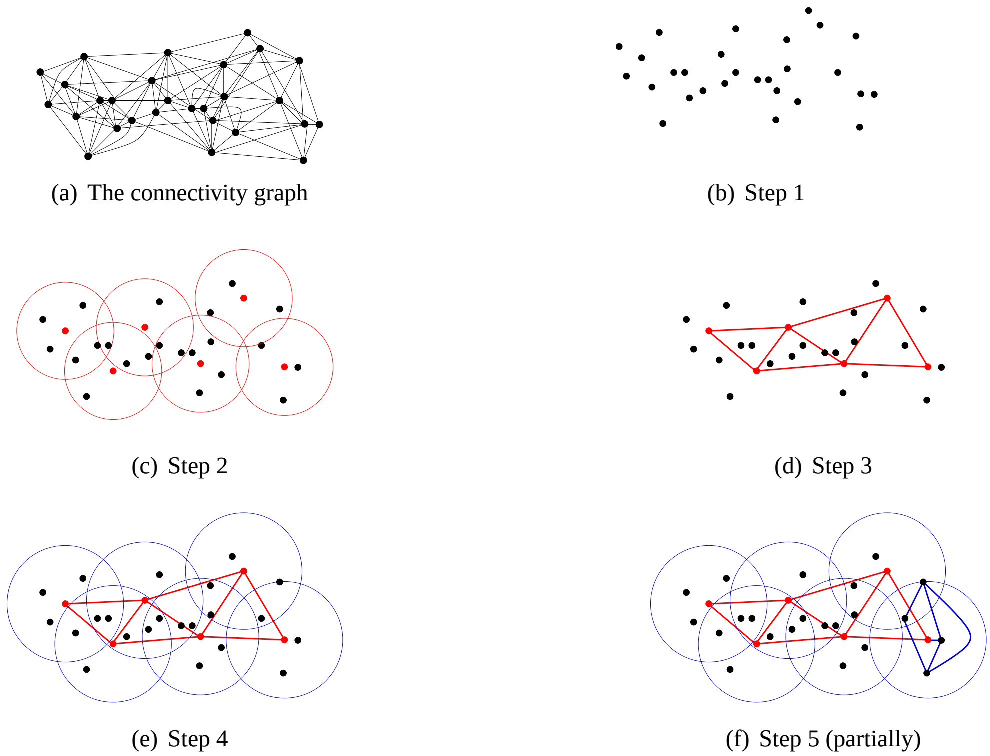


Figure 6.1: Illustration of the Disk Covering Scheme

2. Lay down *small* disks of radius $ar/2, 0 < a < 1$ centered on nodes, such that no central node is covered by more than one small disk and no node is left uncovered. We call each central node a *bridge*. Note that the bridges form a Maximal Independent Set (MIS) of the spanning subgraph $G(n, ar/2, \ell) \subseteq G(n, r, \ell)$.

3. Add to the CHSG all edges from the RGG that connect bridges.

4. Expand the small disks into *big* disks of radius $br/2, a < b \leq 1$.

5. Add to the CHSG the necessary edges to form a spanner of constant degree among nodes covered by the same big disk. We call this spanner a *disk-spanner*.

6.2.2 Analysis of the Disk Covering Scheme

In this section the Disk Covering Scheme described in Section 6.2.1 is proved to produce a CHSG with asymptotically optimal path length. In Section 6.2.2 we establish a bound on the maximum degree of a node in the CHSG. In Section 6.2.2 two useful results for a connected $G(n, r, \ell)$ are established: A bound on hop-stretch and bounds on the node density. Finally, in Section 6.2.2 we prove a theorem on the hop-optimality of the CHSG.

Degree Bound

Lemma 6. *At the end of the Disk Covering Scheme, each edge of length at most $(b-a)r/c$ has both endpoints within a single big disk w.h.p, for any constant $c > 1$.*

Proof. For the sake of contradiction, assume there exists such an edge e of length $l \leq (b-a)r/c$ not covered completely by one big disk. All nodes are covered by small disks. Each endpoint of e has to be covered by a different big disk, otherwise e is already covered. Call C the center of e. Call D the center of any big disk partially covering e. Since e has at least one point outside the big disk, the distance $d(D, C) > br/2 - l/2$ as shown in figure 6.2.

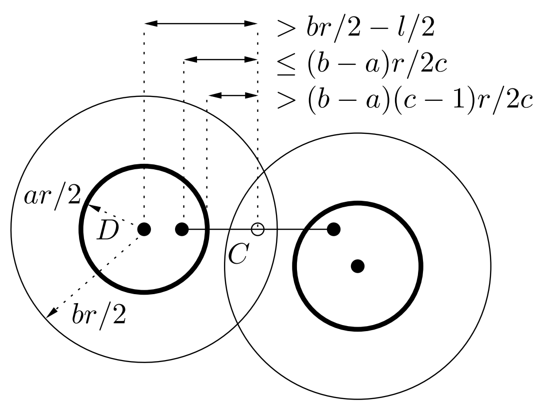
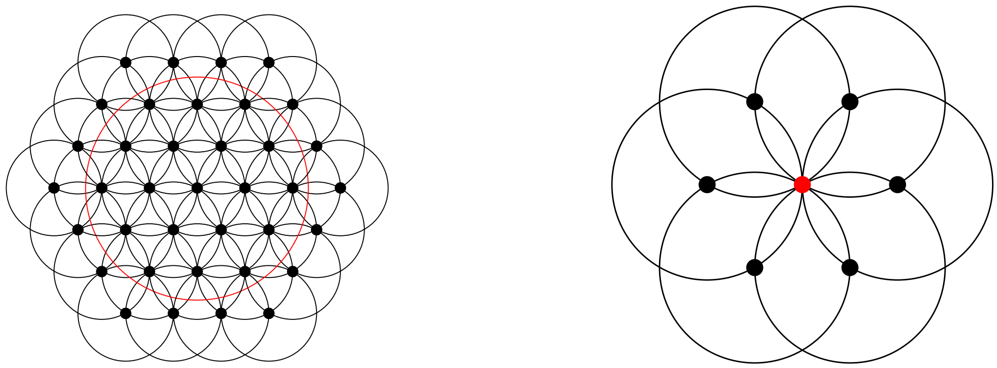


Figure 6.2: Illustration for Lemma 6

Therefore, all centers of big disks that partially cover e are located outside a circle of radius $(r-l)/2$ centered on C. Then, the corresponding small disks leave an uncovered area bigger than the area of a circle of radius $r' > br/2 - l/2 - ar/2 \geq (b-a)(c-1)r/2c$. Since there is no small disk in this area, there is no node in this area, otherwise it would be a disk center. But, as proved in Lemma 9, in any circle of radius $\Theta(r)$ there are $\Theta(\log l)$ nodes w.h.p. This is a contradiction. ☐

Lemma 7. *The degree of any node in the CHSG is in $O(1)$.*



(a) Bridge nodes (b) Non-bridge nodes

Figure 6.3: Illustration of the upper bound on the degree.

Proof. All bridges are separated by a distance of at least $ar/2$. Connected bridges are at a distance of at most r. In figure 6.3(a) consider the smallest regular hexagon whose side is a multiple of $ar/2$ and covers completely a circle of radius r. Consider a tiling of such hexagon with equilateral triangles of side $ar/2$. As proved by Fejes-Tóth in 1940 [FT40], the hexagonal lattice is indeed the densest of all possible plane packings. Therefore, the number of vertices in such a tiling is an upper bound on the number of bridges that connect to a bridge located in the center of such a hexagon. That number is:

$$3\left\lceil \frac{4}{a\sqrt{3}} \right\rceil \left(\left\lceil \frac{4}{a\sqrt{3}} \right\rceil + 1 \right) \tag{6.1}$$

There is an extra edge that is needed to connect a bridge with its disk-spanner. Since a is any constant such that $0 < a < 1$, the degree of any bridge is in $O(1)$.

Using a simple geometric packing argument, it can be proved that a non-bridge node, is covered by at most $\pi/\arcsin(a/2b)$ big disks. By construction, a non-bridge node is connected to a constant number of neighbors within the same big disk (see figure 6.3(b)). Therefore, the degree of any node is in $O(1)$. □

Hop Stretch and Density in $G(n, r, \ell)$

Theorem 8 demonstrates the existence of a path with an asymptotically optimal hop-stretch. The proof of the theorem uses an overlapping dissection technique, called bin-covering, presented by Muthukrishnan and Pandurangan [MP05].

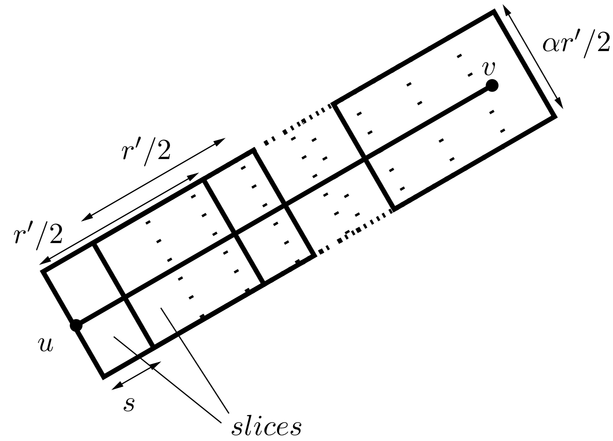


Figure 6.4: Strip between nodes u and v showing bin covering and slices.

Theorem 8. *Given a $G(n, r, \ell)$ where the following conditions are satisfied: $r^2 n = k\ell^2 \ln \ell$, $r = \theta(\ell^\epsilon f(\ell))$, $f(\ell) \in o(\ell^\gamma)$, $\gamma > 0$, $0 \le \epsilon < 1$, and $0 < \alpha \le 1$ is a fixed constant. For any constant $k > 5\frac{4+\alpha^2}{\alpha}$, the hop-stretch is $1 + \sqrt{\alpha^2 + 4}$ w.h.p.*

Proof. It is enough to show that for any pair of nodes (u, v), there is a path P defined by a sequence of nodes $\langle u = x_0, x_1, \ldots, x_m = v\rangle$ such that the ratio between the length of P and the number of hops, m is bounded upwards by $1 + \sqrt{\alpha^2 + 4}$ *w.h.p.*

For a given pair of nodes (u, v), the bin covering technique is applied as follows. Let r' be the shortest horizontal projection of a segment of length r contained in the strip, i.e. $r' = r/\sqrt{1 + (\alpha/2)^2}$. The line connecting u and v is covered with overlapping bins of dimension $r'/2 \times \alpha r'/2$ with a spacing parameter s, as shown in figure 6.4. This bin layout will be referred to as a *strip*.

The coordinate system is rotated such that the line segment $\overline{u,v}$ is parallel to the x axis. In what follows all distances are specified within this rotated frame of reference. Let $D_h(x, y)$ and $D_v(x, y)$ be the horizontal and vertical distances respectively between the nodes x and y.

Given a node x_j in the path P the node x_{j+1} is selected using the following criteria:

- The node x_{j+1} lies within the strip.
- $D_h(x_j, x_{j+1}) \le r'$.
- The horizontal distance $D_h(x_{j+1}, v)$ is minimized.

A *hole* is a rectangle of dimension $r'/2 \times \alpha r'/2$, within a strip, that is devoid of nodes and adjoins a node on the side closest to u.

Consider any 3 consecutive nodes along the path x_{i-1}, x_i, x_{i+1} where $0 < i < m$, and assume that along any strip there is no hole, then $D_h(x_{i-1}, x_i) \geq r'/2$. To see that this claim is true, assume for the sake of contradiction that $D_h(x_{i-1}, x_i) < r'/2$. The distance $D_h(x_{i-1}, x_{i+1}) > r'$, otherwise x_{i+1} would have been selected as the successor of x_{i-1}. Thus, the distance $D_h(x_i, x_{i+1}) > r'/2$. Since there cannot be any hole in the strip, there exists a node y such that $D_h(x_i, y) < r'/2$. This implies that $D_h(x_{i-1}, y) < r'$. Note that $D_h(y, v) < D_h(x_i, v)$, therefore y should have been chosen as the successor of x_{i-1} by the construction criteria, which is a contradiction. The initial assumption of $D_h(x_{i-1}, x_i) < r'/2$ is thus proven false which proves the truth of the claim.

Since $D_h(x_{i-1}, x_i) \geq r'/2$ for $0 < i < m-1$, the number of hops in the path P is

$$m \leq \left\lceil \frac{D(u,v)}{r'/2} \right\rceil = \left\lceil \sqrt{\alpha^2+4}\frac{D(u,v)}{r} \right\rceil .$$

If $D(u,v) \leq r$ the path is simply the edge connecting u and v and the hop-stretch is trivially 1. Otherwise, $D(u,v) > r$ and so, the hop-stretch is $1 + \sqrt{\alpha^2+4}$.

It remains to show that there is no hole *w.h.p.*

To bound the probability that there is a hole in any strip, consider the sequence of small rectangles (call them *slices*) defined by the spacing parameter, of size $s \times \alpha r'/2$. The slices are numbered in ascending order from u to v.

For any node x_i that is contained in some slice j, let E_i be the event that the node x_{i+1} is contained in the slice $j - 1 + \lceil r'/2s \rceil$ at a horizontal distance greater than r' from x_i. Then,

$$Pr[E_i] \leq \binom{n-1}{1} \frac{\alpha r' s}{2\ell^2} \left(1 - \frac{\alpha r'^2}{4\ell^2}\right)^{n-2} .$$

If x_{i+1} is contained in a slice closer to x_i then there is no hole. If x_{i+1} is contained in a slice farther than $j - 1 + \lceil r'/2s \rceil$ then there is at least one empty bin in the strip. The probability that some bin is empty is bounded by

$$Pr[\text{EmptyBin}] \leq \frac{\max_{(u,v)} D(u,v)}{s} \left(1 - \frac{\alpha r'^2}{4\ell^2}\right)^{n} .$$

Therefore, the probability that there is a hole within any strip is

$$Pr[\text{Hole}] \leq \binom{n}{2}\left(n(n-1)\frac{\alpha r's}{2\ell^2}\left(1-\frac{\alpha r'^2}{4\ell^2}\right)^{n-2} + \frac{\max_{(u,v)} D(u,v)}{s}\left(1-\frac{\alpha r'^2}{4\ell^2}\right)^{n}\right)$$
$$\leq n^2\frac{1}{e^{n\alpha r'^2/4\ell^2}}\left(\frac{n^2\alpha r's}{2\ell^2}e^{\alpha r'^2/2\ell^2} + \frac{\sqrt{2}\ell}{s}\right) .$$

This expression is minimized when

$$s = \left(\frac{2\sqrt{2}\ell^3}{n^2\alpha r' e^{\alpha r'^2/2\ell^2}}\right)^{1/2} .$$

Then,

$$Pr[\text{Hole}] \leq \frac{2k^3\ell^6 \ln^3 \ell}{r^6\ell^{1+(k\alpha/(4+\alpha^2))}}\left(\frac{\alpha r' e^{\alpha r'^2/2\ell^2}}{2\sqrt{2}\ell}\right)^{1/2}$$
$$\in O(\ell^{-\gamma}) \text{ for } k > 5\frac{4+\alpha^2}{\alpha} .$$

□

A simpler proof of theorem 8 is also possible and follows, though the constant obtained is worse.

Proof. Consider a strip S_j, the probability that a node x_i is contained in S_j is at most $\alpha r'/\sqrt{2}\ell$. The probability that there is a hole within S_j adjoining x_i is at most $(1-\alpha r'^2/4\ell^2)^{n-1}$. Then, the probability that there is a hole in any strip is

$$Pr[\text{Hole}] \leq \binom{n}{2} n\frac{\alpha r'}{\sqrt{2}\ell}\left(1-\frac{\alpha r'^2}{4\ell^2}\right)^{n-1}$$
$$\in O(\ell^{-\gamma}) \text{ for } k > 6\frac{(4+\alpha^2)}{\alpha}$$

□

Lemma 9. *In a $G(n,r,\ell)$ satisfying the parameter conditions of Theorem 8, the number of nodes contained in a circle of radius $\Theta(r)$ is $\Theta(\log \ell)$ w.h.p.*

Proof. To prove this lemma it is enough to show that the probability that the number of nodes, within any circle of radius βr for some constant β, deviates from $\log \ell$ by more than a constant factor, is polynomially small. Consider the random process of dropping nodes in a square of side length ℓ. Define the random variable X as the number of nodes contained in that circle. For a given node, the probability of falling in the circle is $\pi\beta^2 r^2/\ell^2$. Using Chernoff bounds

$$Pr(X \geq (1+\epsilon)\frac{\pi\beta^2 r^2}{\ell^2} n) \leq e^{-\frac{\epsilon^2}{3} n \frac{\pi\beta^2 r^2}{\ell^2}}$$
$$Pr(X \leq (1-\epsilon)\frac{\pi\beta^2 r^2}{\ell^2} n) \leq e^{-\frac{\epsilon^2}{2} n \frac{\pi\beta^2 r^2}{\ell^2}}$$

Using the parameter conditions

$$Pr(X \geq (1+\epsilon)\pi\beta^2 k \ln \ell) \leq \ell^{-\frac{\epsilon^2 \pi\beta^2 k}{3}}$$
$$Pr(X \leq (1-\epsilon)\pi\beta^2 k \ln \ell) \leq \ell^{-\frac{\epsilon^2 \pi\beta^2 k}{2}}$$

□

Hop Optimality of the CHSG

Lemma 10. *Consider the RGG $G(n, r, l)$, where n satisfies the parameter conditions of Theorem 8 for a reduced connectivity radius of $r' = (b-a)r/c$. For any pair of nodes (u, v) in the RGG at Euclidean distance $D(u, v)$, there exists a path between them in the CHSG of at most $\lceil c\sqrt{\alpha^2+4}D(u, v)/(b-a)r\rceil - 1 + O(\log \ell)$ edges w.h.p.*

Proof. Theorem 8 states that: In the RGG that satisfies the parameter conditions of Theorem 8, there exists a path of $\lceil \sqrt{\alpha^2+4}D(u, v)/r\rceil$ edges w.h.p. We can thus imply that: If the RGG satisfies the same parameter conditions for a reduced connectivity radius of $r' = (b-a)r/c$, there exists a path between u and v using $\lceil c\sqrt{\alpha^2+4}D(u, v)/(b-a)r\rceil$ edges of length at most $(b-a)r/c$. Let p be such a path and $e_1, e_2, \ldots, e_m$ be its sequence of edges.

In the description of the Disk Covering Scheme, two kinds of disks were defined for clarity: big disks and small disks. In order to prove hop-optimality of the CHSG, we only refer to big disks and simply call them disks. The rest of the proof is illustrated in figure 6.5.

Lemma 6 states that every edge in the path p is completely covered by one disk. Therefore, there exists a sequence $d_1, d_2, \ldots, d_{m'}$ of overlapping disks, where any edge e_i in p is covered by some disk d_j in this sequence. A disk may completely cover more than one edge, hence $m' \leq m$. Let D_i be the bridge (center) of disk d_i.

Define a path p' using only edges of the CHSG as follows. Connect u and the bridge D_1 with a path p_1 of disk-spanner edges defined by the disk d_1. For each edge i, $1 \leq i \leq m$,

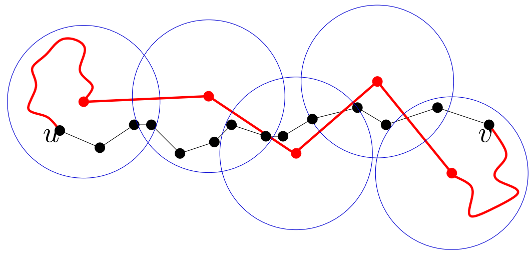


Figure 6.5: Illustration for Lemma 10

replace the edge e_i in p with the node D_i. Connect all consecutive bridges D_i and D_{i+1} within the path of overlapping disks with edge $\overline{D_i D_{i+1}}$. Consecutive bridges are adjacent to each other in the RGG, because their disks overlap and the radius of each disk is $br/2$ with $b \leq 1$. Finally, connect the bridge D_m and v with a path p_m of disk-spanner edges defined by the disk d_m. The length of p' is given by: $length(p') \leq length(p_1) + (m-1) + length(p_m)$. Using the stretch bound, $length(p') \leq \lceil c\sqrt{\alpha^2+4}D(u,v)/(b-a)r \rceil - 1 + length(p_1) + length(p_m)$ w.h.p. Only disk-spanner edges are used in p_1 and p_m. It is shown in Lemma 9 that the number of nodes within a disk is $O(\log \ell)$ w.h.p. Therefore, $length(p_1) + length(p_m) = O(\log \ell)$ w.h.p. completing the proof. □

The following theorem shows the main result.

Theorem 11. *For every pair of nodes in an RGG, there is a path in the CHSG, whose length is asymptotically optimal w.h.p.*

Proof. The optimal path between any pair of nodes (u,v) separated by a distance $D(u,v)$ has at least $\lceil D(u,v)/r \rceil$ edges. If $\log \ell$ is also an asymptotic lower bound on the length of such a path w.h.p., then $(D(u,v)/r + \log \ell)/2$ is also an asymptotic lower bound, and the result proved in Lemma 10 is a constant factor approximation. It remains to show that $\log \ell$ is an asymptotic lower bound on the length of an optimal path in a constant-degree random geometric graph w.h.p.

In a δ-regular graph, the expected distance between any pair of nodes randomly chosen is at least $\log_{\delta-1} n$. A $\Theta(1)$ degree random geometric graph is a subgraph of some regular graph. Hence, in a $\Theta(1)$ degree random geometric graph, the expected distance between any pair of nodes randomly chosen is in $\Omega(\log n)$. The previous result is true w.h.p. because for some constant β

$$Pr(D(u,v) < \beta \log n) \leq \frac{1}{n-1} \sum_{i=0}^{\beta \log n - 2} \delta(\delta-1)^i$$
$$\in O(n^{-\gamma}) \ .$$

Using the union bound, under the parameter conditions of Lemma 10, $D(u,v) \in \Omega(\log \ell)$ for all pairs of nodes (u,v) w.h.p. □

6.3 Distributed Algorithm

In this section we describe how to distributedly implement the steps of the Disk Covering Scheme for network formation. Step 2 of the Disk Covering Scheme can be achieved distributedly by means of a Maximal Independent Set (MIS) computation with nodes transmitting in a range of $ar/2$. An algorithm to compute an MIS in a weak model is presented in [MW05]. This algorithm can be tailored to our setting and can be shown to have a running time of $O(\log^2 \ell)$. The details are presented in Section 6.3.1

Steps 3 and 4 of the Disk Covering Scheme require uncolliding transmissions of each bridge in a radius of r and $br/2$ respectively. All nodes assigned to the same bridge will participate in a common spanner construction. Additionally bridge nodes must set up links with all bridge nodes at a distance of at most r. The details are presented in Sections 6.3.2 and 6.3.3. Finally, the constant-degree spanner construction is described in Section 6.3.3.

6.3.1 MIS Computation (Step 2)

Algorithm

Step 2 of the Disk Covering Scheme can be achieved distributedly by means of an MIS computation with nodes transmitting in a range of $ar/2$. The algorithm detailed in this section is the first $O(\log^2 \ell)$ MIS distributed algorithm with contention resolution in a one-channel environment for application to nodes in a connected random geometric graph and borrows heavily from the algorithm in [MW05] for arbitrary graphs. In the algorithm that follows, δ_1, δ_2, δ_3 and δ_4 are constants.

1. Transmit the local counter with probability $1/\delta_1 \log \ell$.

2. If not transmitting in the current time slot then:

   (a) If a neighbor's counter is received and the difference between the local and neighbor's counter is $\leq \lfloor \delta_2 \log \ell \rfloor$ then set local counter to $-\lfloor \delta_2 \log \ell \rfloor$.

   (b) Else if a neighbor's ID is received then set the local state to *covered* and stop.

3. Increase counter if transmitted at least once.

4. If the counter is $\lceil \delta_3 \log^2 \ell \rceil$ then set the local state to *MIS member* and transmit ID forever with probability $q = 1/\delta_4$.

5. Goto step 1 at end of time slot.

Analysis

The analysis of the MIS algorithm turns out to be difficult because nodes running different phases interfere with each other. Hence, necessary assumptions regarding bounds on the total probability of transmission of nodes in other phases cannot be made leading to a circular argument. In order to break the circularity we prove the following lemmas by induction on the time slots in which a given node joins the MIS.

Before the analysis, we recall the following basic fact [MR95]:

Fact 12. *For all $n \geq 1$ and $|x| \leq n$*

$$e^x \left(1 - \frac{x^2}{n}\right) \leq \left(1 + \frac{x}{n}\right)^n \leq e^x.$$

Lemma 13. *Given any node that joins the MIS in a given time slot, the counter of all neighboring nodes is at most $\lceil \delta_3 \log^2 \ell \rceil - \lfloor \delta_2 \log \ell \rfloor$ in the same time slot w.h.p.*

Lemma 14. *Every MIS node transmits its MIS status message successfully in the $\lfloor \delta_2 \log \ell \rfloor$ time slots after it joins the MIS w.h.p.*

Proof. We prove both preceding lemmas simultaneously by employing induction on the order in which the nodes join the MIS, with ties broken arbitrarily.

Base case: Consider the first node within the whole network, call it μ_1, that joins the MIS at time t_1.

For the sake of contradiction, assume that there is a node x contained in μ_1's neighborhood whose counter is greater than $L = \lceil \delta_3 \log^2 \ell \rceil - \lfloor \delta_2 \log \ell \rfloor$ at t_1. By the definition of the algorithm, μ_1 has first transmitted at time $t_1 - \lceil \delta_3 \log^2 \ell \rceil$ and x has first transmitted within the next $\lfloor \delta_2 \log \ell \rfloor$ time slots. Afterwards, neither μ_1 nor x have sent without collision otherwise one of their counters would have been reset. Let $E(k)$ denote the event that neither μ_1 nor x have sent without collision within k time slots. Using the fact that there are at most $\delta_6 \log \ell$ nodes within the 2-hop neighborhood of μ_1 w.h.p., for some constant $\delta_6 > 0$, as shown in Lemma 9:

$$Pr[E(L)] \leq \left[1 - 2\frac{1}{\delta_1 \log \ell} \left(1 - \frac{1}{\delta_1 \log \ell} \right)^{\delta_6 \log \ell} \right]^{\lceil \delta_3 \log^2 \ell \rceil - \lfloor \delta_2 \log \ell \rfloor}$$
$$\in O(\ell^{-\gamma_1}) \text{ (Using fact 12, for some } \delta_3, \delta_1 > \sqrt{\delta_6 / \log \ell}\text{).}$$

Now we must additionally prove that within $\lfloor \delta_2 \log \ell \rfloor$ time slots of $\mu_1$joining the MIS, all nodes within range of it receive a message declaring its MIS status. For at least $\lfloor \delta_2 \log \ell \rfloor$ time slots after the node μ_1 joins the MIS, no other nodes in its neighborhood join the MIS w.h.p. as shown. If in this time its MIS status message is received by all its neighbors, then they will all stop counting and transition into the *covered* state. We will now show that this message is received by all its neighbors w.h.p. Let $E(k)$ denote the event that μ_1 does not transmit without collision in k consecutive time slots. The probability of failure in $\lfloor \delta_2 \log \ell \rfloor$ consecutive time slots is:

$$Pr[E(\lfloor \delta_2 \log \ell \rfloor)] = \left[1 - \frac{1}{\delta_4} \left(1 - \frac{1}{\delta_1 \log \ell} \right)^{\delta_6 \log \ell} \right]^{\lfloor \delta_2 \log \ell \rfloor}$$
$$\in O(\ell^{-\gamma_2}) \text{ (Using fact 12 and for some } \delta_2\text{).}$$

This shows that μ_1 sends its MIS status message without collision successfully in $\lfloor \delta_2 \log \ell \rfloor$ time slots w.h.p.

*Inductive Step:*Consider the ith node μ_i, $i > 1$, that joins the MIS at time t_i.

Inductive hypothesis: For all nodes μ_j such that $j < i$, joining the MIS at time t_j, the counters of all nodes in the neighborhood of μ_j are at most $\lceil \delta_3 \log^2 \ell \rceil - \lfloor \delta_2 \log \ell \rfloor$ at time

t_j w.h.p. Additionally all nodes μ_j transmit their MIS status message successfully within the interval $t_j \ldots t_j + \lfloor \delta_2 \log \ell \rfloor$ w.h.p.

Therefore by time $t_j + \lfloor \delta_2 \log \ell \rfloor$ all nodes in the range of all MIS nodes $\mu_1 \ldots \mu_{i-1}$ will be in the *covered* state. From the previous statements of the inductive hypothesis we can conclude that none of the MIS nodes μ_j (where $j < i$) are neighbors of each other w.h.p.

We want to show that the counters of all nodes in the neighborhood of μ_i are at most $\lceil \delta_3 \log^2 \ell \rceil - \lfloor \delta_2 \log \ell \rfloor$ at time t_i w.h.p. and that all neighbors of μ_i are in the *covered* state by time $t_i + \lfloor \delta_2 \log \ell \rfloor$ w.h.p.

If μ_i is out of the two-hop neighborhood of all the previous MIS members, the claim can be easily proved using the same argument as in the base case. Otherwise, μ_i is within a two-hop neighborhood of some MIS members. Since all nodes that previously joined the MIS are not in range of each other, μ_i is within the two-hop neighborhood of at most 12 other MIS members. This is true because a regular polygon with side of length at least r and distance from the center to the vertices at most $2r$ has at most 12 sides.

For the sake of contradiction, assume that there is a node y contained in μ_i's neighborhood whose counter is greater than $L = \lceil \delta_3 \log^2 \ell \rceil - \lfloor \delta_2 \log \ell \rfloor$ at t_i. By the definition of the algorithm, μ_i has first transmitted at time $t_i - \lceil \delta_3 \log^2 \ell \rceil$ and y has first transmitted within the next $\lfloor \delta_2 \log \ell \rfloor$ time slots. Afterwards, neither μ_i nor y have sent without collision otherwise one of their counters would have been reset. Let $E(k)$ be the event that neither μ_i or y send without collision for k consecutive time slots.

$$Pr[E(L)] \leq \left[1 - 2\frac{1}{\delta_1 \log \ell} \left(1 - \frac{1}{\delta_1 \log \ell} \right)^{\delta_6 \log \ell} \left(1 - \frac{1}{\delta_4} \right)^{12} \right]^{\lceil \delta_3 \log^2 \ell \rceil - \lfloor \delta_2 \log \ell \rfloor}$$

$$\in O(\ell^{-\gamma_3}) \text{ (Using fact 12,for some } \delta_3\text{,}\delta_1 > \sqrt{\delta_6 / \log \ell}\text{).}$$

Now we will show that all neighbors of MIS node μ_i will be in the covered state by time slot $t_i + \lfloor \delta_2 \log \ell \rfloor$. Any neighbor of an MIS node has a counter that lags the MIS node's counter by at least $\lfloor \delta_2 \log \ell \rfloor$. Additionally no MIS node can be within range of any other. Hence every MIS node can be subjected to interference by at most 18 other MIS nodes (by a simple geometric packing argument). Let $E(k)$ denote the event that a neighbor of an MIS node does not receive its MIS status message for k consecutive time slots. Thus the probability that a MIS node does not transmit its MIS status message without collision is given by:

$$Pr[E(\lfloor \delta_2 \log \ell \rfloor)] \leq \left[1 - \left(\frac{1}{\delta_4}\right) \left(1 - \frac{1}{\delta_4}\right)^{18} \left(1 - \frac{1}{\delta_1 \log \ell}\right)^{\delta_6 \log \ell} \right]^{\lfloor \delta_2 \log \ell \rfloor}$$

$$\in O(\ell^{-\gamma_4}) \text{ (Using fact 12 and for some } \delta_2\text{).}$$

□

Lemma 15. *No two nodes belonging to the MIS are within transmission range of each other w.h.p.*

Proof. This is a direct conclusion of Lemmas 13 and 14. □

Lemma 16. *For any node running the MIS algorithm with radius r, there is at least one node, in its immediate $r/2$ neighborhood, that transmits without collision within $\lceil \delta_5 \log^2 \ell \rceil$ steps w.h.p., for some constant $\delta_5 > 0$.*

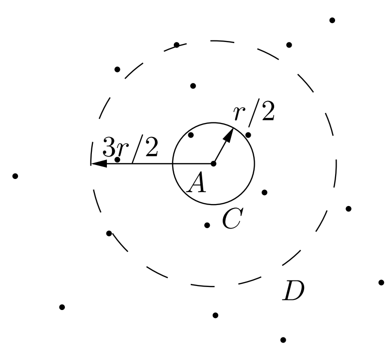


Figure 6.6: Illustration for Lemma 16

Proof. Consider a node A running the MIS algorithm (refer to figure 6.6). Since A is awake, there is at least one node awake in C at time t. From Lemma 15 it can be seen that no MIS nodes can be within range of each other, therefore there can be at most 9 MIS nodes within D (If there were more then one of them would be in range of A). Let $E(k)$ denote the event that no node in A's $r/2$ neighborhood (including A) transmits without collision in k consecutive time slots. Lemma 9 shows that there are at most $\delta_6 \log \ell$ nodes in D w.h.p., for some constant

$\delta_6 > 0$.

$$Pr[E(\lceil \delta_5 \log^2 \ell \rceil)] \leq \left[1 - \left(\frac{1}{\delta_1 \log \ell}\right)\left(1 - \frac{1}{\delta_1 \log \ell}\right)^{\delta_6 \log \ell} \left(1 - \frac{1}{\delta_4}\right)^9\right]^{\lceil \delta_5 \log^2 \ell \rceil}$$
$$\in O(\ell^{-\gamma_5}) \text{ (Using fact 12, } \delta_1 > \sqrt{\delta_6 / \log \ell}, \text{ for some } \delta_5).$$

□

Theorem 17. *For a given node running the MIS algorithm, at least one node within its transmission range joins the MIS in $O(\log^2 \ell)$ time slots and no two MIS nodes are within range of each other w.h.p.*

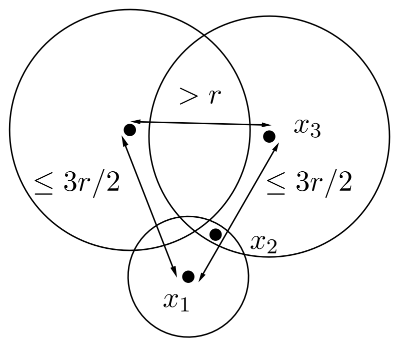


Figure 6.7: Illustration for Theorem 17

Proof. The proof is illustrated in figure 6.7. In Lemma 16, it was shown that within a circle of radius $r/2$ centered on any node x_1, there will be a node x_2, transmitting without collision, in less than $\lceil \delta_5 \log^2 \ell \rceil$ steps w.h.p. After this single transmission, there is at least one node, namely x_2, within the neigborhood of x_1 increasing its counter. If x_2 joins the MIS after its counter reaches the value $\lceil \delta_3 \log^2 \ell \rceil$, then the statement of the theorem is proved. Otherwise, some other node, call it x_3, within range of x_2, reaches this value and joins the MIS before. If x_3 is within range of x_1, then the statement of the theorem is proved. Otherwise, x_3 covers at least one node within the $r/2$ neighborhood of x_1, namely x_2, within the next $\lceil \delta_2 \log \ell \rceil$ time slots w.h.p. (as shown in Lemma 14).

Note that the distance between x_1 and x_3 satisfies the following relation :

$$r < D(x_1, x_3) \leq 3r/2 \ . \tag{6.2}$$

All uncovered active nodes within the $r/2$ neighborhood of x_1 are still counting. Hence, the same argument can be repeatedly applied with the restriction that the next MIS node is at least at a distance of r from x_3 (by Lemma 15). There can be at most 9 MIS nodes around x_1 before x_1 or one of its neighbors joins the MIS, as explained in Lemma 16. Thus, this process terminates in at most $10(\lceil \delta_3 \log^2 \ell \rceil + \lceil \delta_5 \log^2 \ell \rceil + \lfloor \delta_2 \log \ell \rfloor)$ time slots. □

6.3.2 Broadcast (Steps 3 and 4)

After a node is covered by some neighboring MIS node, it needs to be assigned to that MIS node. All nodes assigned to the same MIS node will participate in a common spanner construction. Additionally MIS nodes must set up links with all MIS nodes at a distance of at most r. Any of these steps only require each MIS node to achieve an uncolliding transmission. In this section an algorithm for achieving this is detailed and a time bound is proved.

Algorithm

The algorithm is simple to describe:

> With probability $1/\beta_1$, each MIS node transmits its ID, within range $\beta_2 r$.

Where β_1 and β_2 are constants whose values depend on which of the aforementioned steps is implemented. For informing the non-MIS nodes about assignment, the transmission is made with $\beta_2 = b/2$. For setting up connections with neighboring MIS nodes, the transmission is made with $\beta_2 = 1$.

Analysis

Lemma 18. *Any MIS node running the broadcast algorithm achieves a transmission without collision within $O(\log \ell)$ steps w.h.p.*

Proof. Let Δ denote the maximum number of interfering MIS neighbors (which depends on β_2). Let $Pr[\text{fail}]$ denote the probability that any node fails to transmit without collision after $\beta_3 \log \ell$ steps for some constant β_3. For appropriate values of β_2 and β_3, using the parameter conditions of theorem 8 and the union bound,

$$Pr[\text{fail}] = n\left(1 - \frac{1}{\beta_1}\left(1 - \frac{1}{\beta_1}\right)^{\Delta}\right)^{\beta_3 \log \ell}$$
$$\in O(\ell^{-\gamma}) \text{ For some } \gamma > 0$$

□

6.3.3 Spanner Construction (Step 5)

After nodes are covered by one or more bridges (MIS members), they have to connect locally to neighboring nodes covered by the same bridge, i.e. within the same disk. Nodes can be covered by more than one bridge. Hence, interference of transmissions not only from the local disk but also from neighboring disks must be taken into account to analyze the performance of any spanner construction algorithm. However, any node is covered by at most a constant number of disks as explained in Lemma 7, then the number of interfering transmissions with respect to the local disk is increased only by a constant factor that we fold into the constants involved in this analysis.

Algorithm

Our goal here is to construct a constant-degree spanner graph on the set of nodes assigned to a given bridge node. Since the diameter is not constrained, we adopt the simplest topology, i.e., a linked list. In order to minimize the running time, we avoid handshaking among nodes and all the construction is done by broadcasting. We start with every node choosing an integer index uniformly at random from the interval $[1, \ell]$. Since there are $O(\log \ell)$ nodes within the same range w.h.p. as shown before, no two nodes choose the same index w.h.p.

Analysis

Lemma 19. *Any node running the spanner algorithm joins the spanner within $O(\log^2 \ell)$ steps w.h.p.*

Proof. In order to prove this lemma it is enough to show that every node covered by the same bridge that is running the spanner algorithm achieves at least one single (i.e. uncolliding)

for *each non-bridge node in parallel* **do**
  predecessor.ID $\leftarrow$ *bridge.ID*;
  successor.ID $\leftarrow$ *bridge.ID*;
  choose an integer index uniformly at random from the interval $[1, \ell]$;
  while *true* **do**
    $transmit \leftarrow \begin{cases} true & \text{with probability } p = 1/\beta_4 \log \ell \\ false & \text{with probability } 1 - p \end{cases}$
    if *transmit* **then** broadcast $\langle$index,ID$\rangle$;
    else if *an index is received* **then**
      update *predecessor.ID* or *successor.ID* accordingly;
    end
  end
end

Algorithm 1: Spanner construction. β_4 is a constant.

transmission within $O(\log^2 \ell)$ steps w.h.p. It was shown in lemma 9 that there are $\Theta(\log \ell)$ nodes within any disk of radius $O(r)$. Hence, it is enough to show that within any disk with at most $\beta_4 \log \ell$ nodes there are $\beta_4 \log \ell$ *different* single transmissions within $\beta_5 \log^2 \ell$ steps w.h.p., where β_4 and β_5 are constants.

To show that, we use the following balls and bins analysis. Let each node be represented by a bin and each transmission step be represented by a ball. A node achieving a single transmission at a given step is modeled with the ball representing that step falling in the bin representing that node. If at a given transmission step there is no single transmission, we say that the ball falls outside the bins. Now, to prove this lemma it is enough to show that after dropping $\beta_5 \log^2 \ell$ balls in $\beta_4 \log \ell$ bins, no bin is empty w.h.p.

For a given ball, the probability of falling in a given bin is the probability of achieving a single transmission, i.e.

$$Pr = \frac{1}{\beta_4 \log \ell}\left(1 - \frac{1}{\beta_4 \log \ell}\right)^{\beta_4 \log \ell - 1}$$

Hence, the probability of some empty bin is

$$\begin{aligned} Pr(fail) &\leq \sum_{i=1}^{\beta_4 \log \ell} \binom{\beta_4 \log \ell}{i} \left(1 - i\frac{1}{\beta_4 \log \ell}\left(1 - \frac{1}{\beta_4 \log \ell}\right)^{\beta_4 \log \ell - 1}\right)^{\beta_5 \log^2 \ell} \\ &\leq \left(1 - \frac{1}{\beta_4 \log \ell}\left(1 - \frac{1}{\beta_4 \log \ell}\right)^{\beta_4 \log \ell - 1}\right)^{\beta_5 \log^2 \ell} \sum_{i=1}^{\beta_4 \log \ell} \binom{\beta_4 \log \ell}{i}. \end{aligned}$$

Using the binomial theorem,

$$Pr(fail) \leq \left(1 - \frac{1}{\beta_4 \log \ell}\left(1 - \frac{1}{\beta_4 \log \ell}\right)^{\beta_4 \log \ell - 1}\right)^{\beta_5 \log^2 \ell} 2^{\beta_4 \log \ell}$$

$$\in O(\ell^{-\gamma}), \gamma > 0 \text{ (using fact 12, for a large enough } \beta_5 > e\beta_4\text{).}$$

□

A small-diameter spanner

In the previous construction, the distance between any two nodes is at most the number of nodes within the disk, i.e. $O(\log \ell)$. Although a diameter of $\Theta(\log \ell)$ for the disk spanner is optimal (theorem 11) for a constant-degree random geometric graph, a constant-degree spanner with diameter $o(\log \log \ell)$ is also possible as shown in this section.

The structure we utilize, is popularly known as a *butterfly network*. Butterfly networks are used in many parallel computers to provide paths of length $\log m$ connecting m inputs to m outputs. A labeled instance of a butterfly network with $m = 8$ is shown in figure 6.8. The inputs of the network are on the left and the outputs are on the right. In our case, all nodes have the same role and a message between any pair of nodes can be sent in $O(\log m)$ hops. Then, given that there are $\Theta(\log \ell)$ nodes in any disk, the diameter obtained is $o(\log \log \ell)$. Notice that, once unique consecutive labels are assigned to all nodes, each node can easily compute to which nodes is connected. Then, our goal is to assign unique consecutive indexes to all nodes within the disk.

The distributed algorithm for non-bridge nodes to construct such a network within one disk consists of three phases, as follows. First, every node chooses an index uniformly at random from the interval $[1, \ell]$. As explained before, no two nodes will choose the same index w.h.p. Then, every node broadcasts its index and ID as in algorithm 1, but in this case they keep track of the ID of its predecessor only and the process runs for just $O(\log^2 \ell)$ steps. As shown in lemma 19, at this point all nodes have achieved at least one transmission without collision so, all nodes know who is their predecessor.

To obtain consecutive indexes, the nodes now have to pack the indexes one by one as follows. Upon receiving the new index i of its predecessor, a node redefines its index as $i + 1$

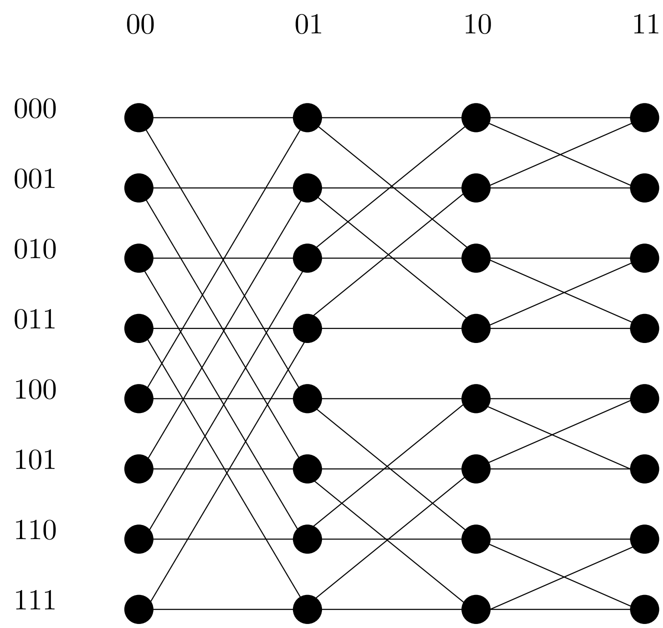


Figure 6.8: A butterfly network with 32 nodes

and broadcasts its new index and ID with constant probability for $O(\log \ell)$ steps. As shown in lemma 6.3.2, there will be at least one transmission without collision w.h.p. Obviously, the first node in this ordering will not have any predecessor and will start this phase of the algorithm redefining its index as 1. At this point, all nodes have consecutive indexes and have to connect as a butterfly accordingly but, they do not know yet the ID's of their butterfly neighbors with smaller index so, a final round broadcasting the new index and ID is necessary. The details can be seen in Algorithm 2

The first and third phase take $O(\log^2 \ell)$ time by definition of the algorithm. In the second phase, each of $\Theta(\log \ell)$ nodes in turn transmit for $O(\log \ell)$ steps. Hence, the overall running time of this algorithm is $O(\log^2 \ell)$.

6.3.4 Overall analysis

The bootstrapping protocol described in this chapter, builds a hop-optimal constant-degree Sensor Network under the constraints of the Weak Sensor Model in $O(\log^2 \ell)$ time w.h.p. The time bounds are for the MIS algorithm $O(\log^2 \ell)$, for the broadcast algorithm $O(\log \ell)$, and for the spanner algorithm $O(\log^2 \ell)$. Hence, the total running time is upper bounded by $O(\log^2 \ell)$.

for *each non-bridge node in parallel* **do**
    predecessor.ID $\leftarrow$ *NULL*;
    choose an integer index uniformly at random from the interval $[1, \ell]$;
    for $\beta_6 \log^2 \ell$ *steps* **do**
        $transmit \leftarrow \begin{cases} true & \text{with probability } p = 1/\beta_7 \log \ell \\ false & \text{with probability } 1 - p \end{cases}$
        if *transmit* **then** broadcast $\langle index, ID \rangle$;
        else if *an index is received* **then**
            update *predecessor.ID* accordingly;
        end
    end
    $index \leftarrow 0$;
    if predecessor.ID$\neq$NULL **then**
        wait until an index from *predecessor.ID* is received;
    end
    $index \leftarrow index + 1$;
    for $\beta_8 \log \ell$ *steps* **do**
        broadcast $\langle index, ID \rangle$ with probability $1/\beta_9$;
    end
    for $\beta_6 \log^2 \ell$ *steps* **do**
        $transmit \leftarrow \begin{cases} true & \text{with probability } p = 1/\beta_7 \log \ell \\ false & \text{with probability } 1 - p \end{cases}$
        if *transmit* **then** broadcast $\langle$index,ID$\rangle$;
        else if *an index is received* **then**
            store ID's of butterfly neighbors according with the index;
        end
    end
end

Algorithm 2: A small-diameter spanner construction. β_4 is a constant.

There is a trade-off among the maximum degree, the length of the optimal path and the density given by

There is a path of $\leq \left\lceil \frac{D(u,v)}{r} \frac{c\sqrt{4+\alpha^2}}{b-a} \right\rceil - 1 + O(\log \ell)$ hops w.h.p.

The degree of any bridge is $\leq 3\lceil \frac{4}{a\sqrt{3}} \rceil \left(\lceil \frac{4}{a\sqrt{3}} \rceil + 1 \right) + 1$ w.h.p.

The density of nodes is $\frac{n}{\ell^2} > 5\frac{4+\alpha^2}{\alpha} \left(\frac{c}{b-a} \right)^2 \frac{\ln \ell}{r^2}$.

Where $0 < a < 1$, $a < b \leq 1$, $c > 1$ and $0 < \alpha \leq 1$.

The longer the edges covered, the lower density and smaller number of hops in the optimal path but, the degree is bigger.

Notice that in our construction, only three ranges of transmission are used, namely $ar/2$, $br/2$ and r. Hence, the specific values of a and b are hardware dependent.

Notice that for any of the various parts of the bootstrapping algorithm no synchronicity assumption is needed. Furthermore, neighboring disks do not need to be running the same phase of the algorithm. Regarding failures, the MIS algorithm and its final broadcast algorithm as well as the linked list spanner construction algorithm are also maintenance algorithms since both bridge and non-bridge nodes keep broadcasting forever. If a bridge node fails, after some time non-bridge nodes will detect the absence of their bridge broadcast and will restart the MIS algorithm to obtain a new bridge. On the other hand, if a non-bridge node fails, its successor and predecessor will interconnect within the next round of the spanner construction. If the butterfly network spanner is used instead and a link is lost, the butterfly network can be simply rebuilt locally from scratch.

6.4 Extensions

In this section we briefly describe how to extend this protocol in order to achieve load balance and to work in settings where the density of nodes is non-uniform and the area of coverage of a node is not a circle.

6.4.1 Load Balance

The topology obtained by the protocol detailed in this chapter is not homogeneous because the node set is partitioned into two subsets, the bridge nodes and the non-bridge nodes. Given that the bridge nodes handle the communication of all the nodes covered by them, the load of work is not uniform among the different nodes in the network. Furthermore, given that the disk spanner is implemented as a linked list, nodes closer to the bridge within the list have to handle messages of all nodes behind them in the list. This issue can be addressed distributedly by simply resetting the bridge status at a random time. As explained in Section 6.3.4, given the unreliable nature of sensor nodes, the bootstrapping algorithm has to be extended to a network maintenance algorithm. More specifically, every bridge node transmits its bridge status periodically and every non-bridge node transmits its ID periodically in order to maintain the disk spanner. Therefore, non-bridge nodes can handle status resets as they handle bridge failures, i.e., re-running the algorithm. Given that re-running the bootstrapping algorithm introduces extra cost, the periodicity of this status reset gives a trade-off between load balance and throughput.

6.4.2 Non-uniform Radius

A frequent assumption in Radio Networks is that the area of coverage of a node is not a perfect circle. More precisely, it is assumed that nodes are connected with probability 1 if they are at a distance of at most r_{min}, beyond that and up to a distance of r_{max} the connectivity is uncertain, and beyond a distance of r_{max} the nodes are assumed not to be connected. We term such a model $G(n, r_{min}, r_{max}, \ell)$. Nevertheless, the main goal for any network formation protocol is to obtain a connected network. Therefore, the conditions on the minimum range of transmission r_{min} are still the same as for the case in which the radius is assumed to be unique. The following theorem, that establishes such conditions was proved in [MP05].

Theorem 20 ([MP05],Theorem 3.5)**.** *Given a $G(n, r, \ell)$ where the following conditions are satisfied: $r^2 n = k\ell^2 \ln \ell$, $r = \theta(\ell^\epsilon f(\ell))$, $f(\ell) \in o(\ell^\gamma)$, $\gamma > 0$, $0 \leq \epsilon < 1$, and $n \in \Omega(1)$. For any constant $k > 2 - 2\epsilon$, the graph is connected w.h.p.*

Under these conditions, the following lemma shows a lower bound on the density of nodes.

Lemma 21. *In a $G(n, r_{min}, r_{max}, \ell)$ satisfying the parameter conditions of Theorem 20, the number of nodes contained in a circle of radius $\Theta(r_{min})$ is $\Omega(\log \ell)$ w.h.p.*

Proof. Same as in Lemma 9. □

However, since we do not have any upper bound on the maximum radius r_{max}, we can not give an upper bound on the number of neighbors of any node better than n, the total number of nodes. Therefore, in order to use the bootstrapping algorithm as detailed in this chapter, we have to add an initial phase that upper bounds the number of neighbors of any node to $O(\log \ell)$. Such a phase can be easily implemented using the Increase From Square algorithm presented in [JS05] and detailed in Section 7.1.3. As in [MW05], instead of running the algorithm forever, every node stops running the first phase upon receiving some transmission. Nodes that transmit at least once during the first phase (successfully or not) go ahead and run the bootstrapping algorithm as a second phase. The rest of the nodes enter a waiting period of $\lceil \delta_1 \log^2 \ell \rceil$ for a constant $\delta_1 > 0$. If none of their neighbors become a bridge within the waiting period, they simply re-run the protocol. The details of the algorithm for the first phase are included here for completeness. In this algorithm, δ_1, δ_2 and δ_3 are constants.

- For $\lceil \delta_1 \log^2 \ell \rceil$ steps:
  - If a bridge node transmission is received then stop and proceed to the disk spanner formation step of the Disk Covering Scheme.
  - Else If a non-bridge node transmission is received then re-start.
- For $i = 0$ to $\lceil \log \ell \rceil$:
  - Repeat for $\lceil \delta_3 \log \ell \rceil$ steps:
    - Transmit ID with probability $2^i/\ell\delta_2$.
    - If transmitting in the current time step then stop and proceed to the Disk Covering Scheme.
    - Else If a bridge node transmission is received then stop and proceed to the disk spanner formation of the Disk Covering Scheme.
      - Else If a non-bridge node transmission is received then re-start.

The analysis showing that the number of nodes running the bootstrapping algorithm in any circle of radius $\Theta(r)$ is $O(\log \ell)$ w.h.p., can be done assuming that the first phase uses a different channel of communication in the presence of some source of interference that produces a transmission with constant probability. This source of noise models the interference of nodes running the bootstrapping algorithm in a one-channel setting given that due to lack of global synchronization different nodes may be running different phases. The analysis of the bootstrapping algorithm can be easily re-done with a similar assumption for the interference of the first phase. Choosing the constants involved in both phases of the algorithm (probabilities, counters, number of rounds, etc.) adequately, the sum of probabilities of transmission of neighboring nodes in the same phase is in fact a constant. This analysis can be done as a simple generalization to this assumption of the analysis in [MW05].

6.4.3 Non-uniform Distribution of Nodes

Another feasible assumption in Radio Networks is that the deployment of nodes is not uniform. Although some papers analyze problems in Radio Networks under the assumption of arbitrary distribution of nodes, this assumption is unreallistic since the layout of nodes is not a result of an uncontrolled random experiment where the probability of some highly undesirable outcome is positive. However, a uniform distribution of nodes in the plane in situations where the environment is hostile or remote may be difficult to achieve. An example of a feasible model for the distribution of nodes that reflects the random nature of the deployment leaving aside highly unlikely arbitrary distributions is a multiple bivariate normal distribution. In other words, the node density is described as a composition of normal distributions in the plane.

Notice that, independently of the model of the non-uniform distribution of nodes chosen, in order to guarantee connectivity a minimum density of nodes has to be ensured. Furthermore, the problem analyzed in the previous section, i.e., find the minimum radius in order to achieve connectivity, can be easily stated as the problem of finding the minimum number of nodes to ensure connectedness given a fixed transmission range. In fact, the conditions on the minimum density proved in Lemma 21 still hold but, as before, we can not give an upper bound on the number of neighbors of any node better than n, the total number of nodes. In order to address this issue, we follow the same approach as in the previous section.

Chapter 7

Survey: the Clear Transmission Problem

Any network where transmissions may collide needs a protocol for *collision-free transmissions*. The problem of achieving a successful transmission of at least one node is fundamental since, indeed, to solve any problem in a communication network at least one successful transmission is necessary. In some networks, such as a Radio Network, a node receives succesfully a message only if exactly one of its adjacent neighbors has transmitted in that time slot. If many neighbors send messages simultaneously messages collide and the node receives only noise.

We recall from Chapter 4 the definition of the problem. A *clear reception* at a node A is achieved if in a given time slot exactly one of the adjacent neighbors of A transmits. On the other hand, we say that a node B has achieved a *clear transmission* if B transmits and no other two-hop neighboring node of B transmits in a given time slot. Notice that although both problems look similar they are not the same in a multi-hop network. When a clear transmission occurs, all the adjacent nodes of the transmitter receive the message. Whereas if a clear reception occurs only the receiver is guaranteed to get the message because other nodes adjacent to the receiver may be receiving at the same time the transmission of some other node which is not in the range of the receiver (hidden-terminal problem). Of course, both problems are identical in a one-hop network since for a given node to receive a message exactly one node in the whole network has to transmit.

Algorithms for achieving a clear transmission have been studied in several shared-channel contention settings. We summarize here related work in clear transmissions and clear receptions, and we give the details of our results in this area in Chapter 8. In a one-hop Radio Network, the clear transmission problem is equivalent to the so-called *broadcast*, *wake-up* and *leader election* problems (refer to Chapter 4). These problems differ in multi-hop networks because, although a clear transmission is still necessary, it may not be sufficient. Therefore, while considering lower bounds, we will cite bounds for the clear transmission problem, even when the bounds were originally stated for the other problems.

7.1 Randomized Upper Bounds in One-hop Networks

The clear transmission problem in a one-hop network is also called the *selection* problem. Recall that in the most general version of the selection problem the number of nodes in the network is n but only some subset of nodes participate in the protocol. There are many models under which protocols for this problem can be studied, depending on the knowledge of the size of the network n, the number of active nodes or *participants* in the protocol d, the type of synchronization (global or local), the availability of collision detection, etc. In the following sections we summarize some protocols under these various conditions.

7.1.1 Active-Nodes-Set Size Known and Global Synchronization

If the number of participating nodes d is known, a simple approach is to use controlled-Aloha [Met75]. That is, each participating node transmits in each time slot with probability $1/d$.

Theorem 22. *Given a one-hop network where the number of participating nodes d is known, the protocol controlled-Aloha [Met75] achieves a clear transmission w.h.p. in $O(\log n)$ steps.*

Proof. The probability of not achieving a transmission in t steps is

$$\begin{aligned} Pr_{fail} &\leq \left(1 - \left(1 - \frac{1}{d}\right)^{d-1}\right)^t \\ &\leq \frac{1}{n^\gamma}, \gamma > 0, \text{ for any } t \in \Omega(\log n). \end{aligned}$$

□

7.1.2 Active-Nodes-Set Size Unknown and Global Synchronization

Unfortunately, the number of participating nodes is not known in general and smarter solutions are needed. The lack of such information makes a big difference in order to obtain fast protocols. A common observation in literature is that a *fair* protocol, i.e., a protocol where all nodes are assumed to use the same probability of transmission p in the same time step (global synchronization), has a high probability of achieving a successful transmission when p and the number of participating nodes d agree up to a constant factor and this probability is low otherwise. Therefore, a main challenge for any protocol is to estimate d accurately and fast.

Network Size Known with Collision Detection

In [Wil86] Willard presented randomized protocols to achieve a successful transmission in a one-hop network when collision detection is available. When the size of the network n is known, the protocol called *super exponential binary search* (SEBS) works as follows. In a first phase, nodes guess the number of participating neighbors. To that extent, the protocol works in rounds. Nodes transmit in each round with probability 2^{-i}. The value of i for each round is chosen by binary search in the space $[1, \lceil \log n \rceil]$. The decision regarding the value of the exponent for the next round is taken based on the feedback from the channel in the current round. More precisely, if there is silence the probability is increased, if there is collision the probability is reduced, and if there is a successful transmission nodes stop running the protocol. Upon completion of the first phase, nodes use controlled Aloha with the guessed density of participating nodes as a parameter until they achieve a successful transmission.

Theorem 23 ([Wil86],Theorem 2.10)**.** *Given a one-hop network with collision detection where the size of the network n is known by the nodes, the protocol SEBS achieves a clear transmission in $\log\lceil \log n \rceil + O(1)$ expected time slots.*

It is easy to see that the running time of the first phase is at most $\log\lceil \log n \rceil$ time steps. So, the proof goes mainly about proving that the probability of guessing a density of participating nodes away from the actual value for more than a constant in the first phase is sufficiently low as to guarantee that the second phase runs in expected $O(1)$ time. We refer the reader to the original paper for details. For protocols where all nodes are assumed to use the same probability of transmission, Willard proved in the same paper a matching lower bound in the expected running time, showing the optimality of SEBS under that assumption.

Network Size Unknown with Collision Detection

For the case in which the size of the network n is unknown, the protocol proposed in [Wil86], called QSEBS, uses a first phase that guesses the number of participating nodes using the search algorithm of Bentley and Yao [BY76], and the same second phase of SEBS. The algorithm of B-Y is based on a progression of binary search algorithms. More precisely, consider the standard unbounded binary search of a number $d \in \mathbb{N}$ that sucessively evaluates 2^i, for each $i = 1, 2, \ldots$, until an interval such that $d \in [2^{i-1}, 2^i]$ is found so that a bounded binary search can be applied. Notice that, in this algorithm, the appropriate value of i is found by exhaustive search. Performing an unbounded binary search of the appropriate value of i, we can obtain a faster algorithm called *double binary search*. The algorithm of B-Y applies recursively this idea, choosing the depth of the recursion appropriately depending on the magnitude of n. B-Y proved in that paper that in order to determine d, when no upper bound on d is known, their algorithm incurs in $\sum_{j=2}^{\log^* d} \log^{(j)} d$ comparisons, where $\log^{(j)} d$ is the jth iterated logarithm of d and $\log^* d$ is the least integer k such that $\log^{(k)} d \leq 1$. This result combined with the analysis of the second phase used to prove Theorem 23 results in a total running time of $O(\log \log d)$.

Network Size Known without Collision Detection

Although in many Radio Networks it is possible to detect a collision, it has been also argued that a collision can not be detected in the presence of noisy channels [BYGI92]. We briefly review now previous work in clear transmissions under the assumption of a binary channel, i.e., only two channel states are feasible, single transmission and silence/collision.

Hayashi, Nakano and Olariu [HNO99] presented the first $O(\log^2 n)$ algorithm for clear transmission with high probability in one-hop Radio Networks without collision detection. The protocol given, called *Election-with-no-CD*, is intended to solve the problem of leader election which, as mentioned before, in the one-hop scenario is the same as the clear transmission problem. The approach is simple, for each non-negative integer i starting with $i = 0$, nodes transmit with probability 2^{-j} for all $j \in [0, i]$. If any of the nodes achieve a successful transmission all nodes stop running the protocol. The intuition behind this protocol is that, as explained before, the probability of achieving a successful transmission is high when the number of participating nodes and the probability agree up to a constant factor. So the approach consists in repeatedly try all possible values of $1/d$ as probabilities of transmission.

Theorem 24 ([HNO99],Corollary 3.2)**.** *Given a one-hop network without collision detection where the size of the network n is known by the nodes, the protocol Election-with-no-CD achieves a clear transmission w.h.p. in $O(\log^2 n)$ time slots.*

A sketch of the proof follows. Observing that the probability of a successful transmission in a given round is $Pr_{succ} \geq dp\,(1-p)^{d-1}$. It is easy to see that whenever the probability of transmission is within $[1/2d, 2/d]$ the probability of success is at least $Pr_{succ} \geq (1-p)^{p-1} \geq 1/e^2$. Then if we have at least $\Omega(\log n)$ rounds with such probability of transmission, we have a success w.h.p. By a simple observation of the algorithm, the number of rounds needed is $O(\log^2 n)$.

In [GPP01], Gasieniec, Pelc and Peleg presented algorithms to wake up a one-hop Radio Network in different scenarios such as global or local synchronism and known or unknown network size. Recall that in the wakeup problem the goal is to wakeup all nodes in the network and nodes can be woken up by a successful transmission or spontaneously. Given that this is a single-hop network, and that in the worst case nodes do not wakeup spontaneously after time $t = 0$, this problem is the same as the clear transmission problem. When n is known and the synchronism is global, the algorithm proposed, called *repeated-decay*, is based on the algorithm *decay* of [BYGI92]. The algorithm decay works in rounds up to a maximum of $2\lceil\log n\rceil$ rounds. In each round, each node transmits a wakeup message and continues the protocol with probability $1/2$. The algorithm repeated-decay just executes phases of decay repeatedly until a successful transmission is achieved. If a node wakes up spontaneously waits to the beginning of the next phase to start running the protocol.

Theorem 25 ([GPP01],Theorem 2.3)**.** *Given a one-hop Radio Network where n is known by all nodes and global synchronization is available, the algorithm repeated-decay achieves a clear transmission with probability $1-\epsilon$ in time $O(\log n \log(1/\epsilon))$*

The algorithm decay achieves a clear transmission with probability $\Omega(1/2)$ as proved in [BYGI92]. Therefore, the probability of failing for $\log(1/\epsilon)$ rounds is at most $1-1/2^{\log 1/\epsilon}$. Given that the algorithm decay works in rounds up to a maximum of $2\lceil\log n\rceil$ rounds, the theorem follows.

7.1.3 Active-Nodes-Set Size Unknown and Local Synchronization

Network Size Known without Collision Detection

Unfortunately, given the lack of synchronization we can not guarantee a number of participants in a series of rounds, i.e., in a given time step the number of participants can be any integer in $[1, n]$. Of course, this number can not decrease in this model. A simple approach, also included in [GPP01], is to follow the approach of controlled-Aloha being conservative, i.e., assuming that all nodes participate and transmitting in each round with probability $1/n$.

Theorem 26. *Given a one-hop Radio Network where n is known by all nodes and only local synchronization is available, if all nodes transmit with probability $1/n$ a clear transmission is achieved with probability $1 - \epsilon$ in time $\Theta(n \log(1/\epsilon))$.*

Proof. The probability of failing to have a successful transmission in t steps is

$$\begin{aligned} Pr_{fail} &\leq \left(1 - d\frac{1}{n}\left(1 - \frac{1}{n}\right)^{d-1}\right)^t \\ &\leq \left(1 - \frac{1}{n}\left(1 - \frac{1}{n}\right)^{n}\right)^t \\ &\leq \left(1 - \frac{1}{2en}\right)^t \\ &\leq \epsilon, \text{ for any } t \in \Theta(\lceil 2en \log 1/\epsilon \rceil). \end{aligned}$$

□

As proved in [FCFM06], there is a lower bound of $\Omega(\log n \log(1/\epsilon))$ time to achieve a clear transmission with probability $1 - \epsilon$, so the previous upper bound would imply an exponential gap between the upper and lower bounds in this model. However, this upper bound is not tight as shown in the next section.

Network Size Unknown without Collision Detection

Recall that a protocol where all nodes transmit with probability p has a high probability of achieving a successful transmission when p and the number of participating nodes d agree up to a constant factor, and this probability is low otherwise. However, we consider in this section protocols where d is not known. A simple approach, if all nodes use the same probability, would be to try with each possible probability for enough number of steps so that whenever the probability is $p \in \Theta(1/d)$ a successful transmission is achieved. More specifically, the algorithm would consist in $\lceil \log n \rceil$ rounds. In round r, a node transmits with probability $p = 2^{-r}$. However, due to lack of global synchronization, the protocol is not a fair one, i.e., nodes may be using different probabilities. Nevertheless, it can be proved that the protocol still achieves a clear transmission fast. The intuition behind such a proof is as follows. The goal is to reach a round where some node transmits successfully but to achieve this we do not need all the nodes to use the same probability, it is enough if the summation of the probabilities of transmission of all nodes is in $\Theta(1)$ during one complete round. In order to achieve that, nodes should transmit with a small probability $\Theta(1/n)$ in the first round and increase the probability for each new round, though this would imply that nodes know the size of the network n. Nevertheless, as shown in [JS05] the labels of processors can be used as local approximations of the size of the network yielding the algorithm 3 called *Increase From Square*.

while *true* **do**
    $p \leftarrow 1/2^{\lceil \log(\pi^2 (i+1)^2/3) \rceil}$;
    if $p \leq 1/2$ **then**
        transmit with probability p for $\delta \log(1/\epsilon)$ steps;
    end
    $p \leftarrow 2p$;
end

Algorithm 3: Algorithm Increase From Square [JS05]

Theorem 27 ([JS05],Theorem 7.1)**.** *Given a one-hop Radio Network where n is not known and only local synchronization is available, the algorithm Increase From Square achieves a clear transmission with probability $1 - \epsilon$ in time $O(\log n \log(1/\epsilon))$.*

Proof. Let V be the set of nodes and let p_i be the probability of transmission of node i. Consider the first time slot t when the sum of the probabilities of transmission of all nodes becomes $\sum_{i \in V} p_i \in \Theta(1)$. Such a time slot exists because nodes increase the probability until reaching a constant. Given that nodes duplicate their probability of transmission every $\delta \log 1/\epsilon$ steps, the sum of probabilities of the nodes participating at time t will still be a constant at time $t + \delta \log 1/\epsilon$. Due to lack of global synchronization, nodes may begin to participate in the protocol during this period, however, their sum of probabilities of transmission during the first $\delta \log 1/\epsilon$ time steps is at most a constant. Therefore, after the time step t, the sum of probabilities of transmission of all nodes is $\sum_{i \in V} p_i \in \Theta(1)$ for at least the next $\delta \log 1/\epsilon$ time steps. Now it is easy to see that under these conditions the probability of not having a successful transmission is low.

$$
\begin{aligned}
Pr_{fail} &\leq \left(1 - \sum_{i \in V} p_i \prod_{j \in V, j \neq i} (1 - p_j)\right)^{\delta \log 1/\epsilon} \\
&\leq \left(1 - \sum_{i \in V} p_i \prod_{j \in V} (1 - p_j)\right)^{\delta \log 1/\epsilon}.
\end{aligned}
$$

Observing that $(1-p) \geq (1/4)^p$ for every $0 < p \leq 1/2$,

$$
\begin{aligned}
Pr_{fail} &\leq \left(1 - \sum_{i \in V} p_i \prod_{j \in V} (1/4)^{p_j}\right)^{\delta \log 1/\epsilon} \\
&= \left(1 - \sum_{i \in V} p_i \left(\frac{1}{4}\right)^{\sum_{j \in V} p_j}\right)^{\delta \log 1/\epsilon} \\
&\leq \epsilon, \text{ for some } \delta > 0.
\end{aligned}
$$

□

Given the lower bound shown in Chapter 8, this algorithm is optimal. Obviously, the same upper bound holds for the case where n is known by simply ignoring such information.

7.2 Randomized Upper Bounds in Multi-hop Networks

Recall that in a multi-hop network the clear reception problem and the clear transmission problem are not the same. When a clear transmission occurs, all the adjacent nodes of the transmitter receive the message. However, if a clear reception occurs only the receiver is guaranteed to get the message due to the hidden-terminal problem. Therefore, in this section we will analyze these problems separately.

Much of the research in shared-channel contention settings is not specific for the clear transmission problem but for more general problems such as broadcast, wake-up, leader election, etc. Although a solution for any of these problems implies a solution for the clear reception problem, it is not clear that solving the clear transmission problem is a necessary condition to solve any of the aforementioned problems. For instance, in the broadcast problem a non-empty subset of nodes are allocated messages and the goal is that all nodes receive some message. To that extent it is enough that all nodes receive a message but it may not be necessary that all nodes in each one-hop neighborhood have received the message in the same time slot. In the worst case of the wake-up problem a non-empty subset of nodes wakes up spontaneously at some initial time t_0 and the rest of the nodes have to be woken up by a successful reception so the same argument holds. Nevertheless, some solutions in the literature actually solve also the clear transmission problem or can be used to solve it without extra cost asymptotically. We will survey some of these results in this section that obviously solve the clear reception problem too.

In a seminal paper [BYGI92], Bar-Yehuda, Goldreich and Itai gave a $O((D+\log(n/\epsilon))\log n)$ randomized algorithm to broadcast a message with probability at least $1-\epsilon$ in a multi-hop Radio Network with diameter D, when the nodes know an upper bound on n and an upper bound on the maximum degree Δ. This protocol is based on an algorithm called *Decay* that resolves contention by randomly eliminating half of the transmitters. The process is repeated enough number of times in order to achieve the desired probability of success. This process of cutting by half can be implemented distributedly by letting each node to eliminate itself after tossing a fair coin. The protocol relies in some form of synchronicity which can be achieved by assuming that there is a unique source node and taking time $t=0$ when the source transmits the message. Therefore, in the most general model where only local synchronization is allowed, a protocol for clear transmission or reception based on Decay relies on actually solving the broadcast problem which implies the same time bound.

The Algorithm 3, Increase From Square, shown before can be used to solve the clear reception problem in a multi-hop Radio Network efficiently. In fact, there is only one minor modification to be introduced. Namely, instead of running the algorithm forever, every node stops running the protocol upon receiving some transmission. Furthermore, the time bound is still the same as shown in the following theorem.

Theorem 28. *Given a multi-hop Radio Network where n is not known and only local synchronization is available, the algorithm Increase From Square solves the clear reception problem with probability $1-\epsilon$ in time $O(\log n \log(1/\epsilon))$.*

The proof for each one-hop neighborhood is the same as in Theorem 27. Interference among one-hop neighborhoods is not a problem. To deal with it, induction in the sequence of time slots in which the sum of probabilities of transmission of some one-hop neighborhood reach a constant can be used. The base case is the first time slot t_0 when such an event occur. Recall that in the algorithm Increase From Square nodes double the probability of transmission in each round. Therefore, if such an event does not occur within $O(\log n)$ steps is only because every node has received some transmission. Hence, the problem is solved.

As explained in Chapter 6, when $\epsilon = 1/n$, a $O(\log^2 n)$ time bound can be obtained for the much more complicated problem of computing a Maximal Independent Set (MIS) in the multi-hop Weak Sensor Model w.h.p. This algorithm can be used to solve the clear transmission problem without extra cost as follows. In a first phase the MIS is computed using a $O(log^2 n)$ MIS algorithm and in a second phase all MIS nodes repeatedly transmit with constant probability. Although due to lack of global synchronization different nodes may be running different phases, as shown in Chapter 6 this is not a problem if we do the analysis under the assumption of the existance of a source of noise that transmits with constant probability.

Theorem 29. *Given a multi-hop Radio Network without collision detection where the size of the network n is known, and the synchronization is local, the algorithm of the previous paragraph solves the clear transmission problem in $O(\log^2 n)$ time slots w.h.p.*

Proof. By definition of an MIS, there is a constant number of MIS nodes in any two-hop neighborhood. Therefore, if the probability of transmission is p, the number of MIS nodes in any two-hop neighborhood is at most δ, and the probability of transmission of the source of noise is q, the probability that a given MIS node does not achieve a clear transmission is

$$\begin{aligned} Pr_{fail} &\leq \Big(1 - p(1-p)^{\delta}(1-q)\Big)^{t} \\ &\leq \frac{1}{n^2}, \text{ for some } t \in \Omega(\log n) \end{aligned}$$

Using the union bound, the probability that any MIS node does not achieve a clear transmission is $O(1/n)$. Given that there exists exactly one MIS node in every one-hop neighborhood w.h.p., the clear transmission problem is solved. □

Bar-Yehuda, Goldreich and Itai showed in [BYGI91] that the algorithms developed for one-hop Radio Networks with collision detection can be emulated in multi-hop Radio Networks without collision detection. Each round of the one-hop network can be emulated by $O((D + \log(n/\epsilon)) \log \Delta)$ rounds of the multihop network and succeeds with probability at least $1 - \epsilon$, where D is the diameter of the network and Δ is the maximum degree. Thus, algorithmic results concerning single-hop radio networks may have some impact on the multi-hop model.

7.3 Lower Bounds

Kushilevitz and Mansour [KM98] proved the first lower bound of $\Omega(\log n)$ on the expectation of the running time of any randomized algorithm for clear transmissions in Radio Networks. Notice that the algorithm of Willard [Wil86] gives an expected $O(\log \log n)$ running time to achieve a clear transmission in a single-hop radio network with collision detection. Hence, the lower bound of Kushilevitz and Mansour shows also an exponential gap between both models.

The specific problem for which the lower bound is proved is broadcast. In a one-hop Radio Network the broadcast problem is the same as the clear transmission problem. To see this, it is enough to assume in the broadcast problem that there is an adittional node called *the originator* that it is connected only to a non-empty subset of size d of the n nodes forming the network. In a first time slot the originator broadcasts its message and only its d neighbors receive it. From now on we have the same setting as in the clear transmission problem, i.e., a subset of d participating nodes trying to achieve a non-colliding transmission. However, nodes know neither which are the participating nodes nor the magnitude of d.

The lower bound is proved under the assumption that all nodes execute the same protocol. If the protocol is non-uniform, i.e., nodes may run different protocols, a simple reduction as the one used in Section 8.2 from the non-uniform case to the uniform one shows that the same lower bound holds.

The goal is to show that for every clique of n nodes, there exists *some* non-empty subset of size d such that, if the nodes in this subset try to transmit, the expected number of time steps until exactly one of them transmits is $\Omega(\log n)$. To that extent it is enough to take the expectation over all choices of d of the form 2^i where $i \in [1, \log n]$ of the expected running time when 2^i nodes transmit, because then there exists *some* choice of d for which it is not possible to do it faster.

Theorem 30 ([KM98],Lemma 1)**.** *Given a one-hop Radio Network of size n without collision detection where the number of participating nodes d is unknown and with global synchronization, any randomized uniform protocol requires at least $\Omega(\log n)$ time steps in expectation to achieve a clear transmission. More precisely, $E_i[E[T_i]] \in \Omega(\log n)$. Where $E[T_i]$ is the expected running time of the protocol when the number of participating nodes is $d = 2^i$ and E_i is the expectation over the uniform choices of an integer $i \in [1, \log n]$.*

Sketch of the proof. First we observe that the decisions made by participating nodes are independent since before achieving a successful transmission they do not exchange any information. Therefore, each participating node can decide to transmit or not in a given time slot based only in its own history of transmissions. Therefore, w.l.o.g., we can assume that participating nodes make their decisions in advance before running the protocol. Thus, at a given time step the probability of transmission of any node is the same for all participating nodes. Under these assumptions, we can simply compute the expectation as follows. Let $Pr(t, i)$ be the probability of achieving the first non-colliding transmission at the time step t when 2^i nodes participate in the protocol. Then,

$$
\begin{aligned}
E_i[E[T_i]] &= \sum_{i=1}^{\log n} \frac{1}{\log n} \sum_{t=1}^{\infty} t\dot{P}r(t,i) \\
&\geq \frac{T}{\log n} \sum_{i=1}^{\log n} \sum_{t=T}^{\infty} Pr(t,i) \\
&= \frac{T}{\log n} \sum_{i=1}^{\log n} \left(\sum_{t=1}^{T-1} Pr(t,i) \right) \\
&\geq \frac{T}{\log n} \sum_{i=1}^{\log n} \left(\sum_{t=1}^{T-1} 2^i p(t)(1-p(t))^{2^i-1} \right) \\
&\geq \frac{T}{\log n} \left(\log n - \sum_{t=1}^{T-1} \sum_{i=1}^{\log n} 2^i p(t)(1-p(t))^{2^i-1} \right) \\
&\geq \frac{T}{\log n} \left(\log n - \sum_{t=1}^{T-1} 2 \right) \\
&= \frac{T}{\log n} \left(\log n - 2(T-1) \right) \\
&\geq \frac{1}{8} \log n + \frac{1}{2} \text{, for } T = \frac{1}{4} \log n.
\end{aligned}
$$

A lower bound of $\Omega(\log n \log(1/\epsilon)/(\log\log n + \log\log(1/\epsilon))$ for achieving a clear transmission with probability $1-\epsilon$ in a one-hop, globally-synchronized Radio Network was proved in [JS05] by Jurdzinski and Stachowiak. The lower bound of Jurdzinski-Stachowiak is tighter than the previous one of Kushilevitz-Mansour if $\epsilon \in o(1/\log n)$.

The specific problem for which the lower bound is proved is wakeup. In a one-hop Radio Network the wakeup problem is the same as the clear transmission problem. Nodes waking up spontaneously at different time steps in the wakeup problem are the same as nodes starting a clear-transmission protocol. Also, at any time step, nodes know neither which are the participating nodes (awake nodes) nor the number of them d. In order to solve the wakeup problem all nodes need to be woken up either spontaneously or by a successful reception. Given that this is a single-hop Radio Network a successful reception requires a clear transmission.

Theorem 31 ([JS05],Theorem 5.2)**.** *Given a one-hop Radio Network of size n without collision detection where the number of participating nodes d is unknown and with global synchronization, any randomized uniform protocol requires at least $\Omega\left(\frac{\log n \log(1/\epsilon)}{\log(\log n \log(1/\epsilon))}\right)$ time steps in order to achieve a clear transmission with probability at least $1-\epsilon$.*

Sketch of the proof. We assume that all the participating nodes begin running the protocol at the same time slot and that no new nodes start afterwards. Let us call a *lost step* to a time step when the probability of achieving a successful transmission Pr_{succ} is below some threshold p_ℓ to be defined later. If p is the probability of transmission of the participating nodes in a time step, using that $Pr_{succ} = dp(1-p)^{d-1}$ and with some algebra, it is easy to show that the step is lost unless $p_\ell \leq dp \leq 1/p_\ell$ as long as $d \geq max\{10, \log(1/p_\ell)\}$. Therefore, for a given probability of transmission p, we can give $\log n/2\log(1/p_\ell)$ different values to d such that only one of them produces a non-lost step. We now bound the minimum number of non-lost steps that are needed in order to achieve the clear transmission with probability $1-\epsilon$. The probability of not achieving a succesful transmission in t non-lost steps is

$$\begin{aligned} Pr_{fail} &= \left(1 - dp(1-p)^{d-1}\right)^t \\ &\leq \left(\frac{1}{e}\right)^{t\frac{p_\ell}{e^{p_\ell}}} \\ &\leq \epsilon \text{ , for } t \in \Omega\left(\frac{e^{p_\ell}}{p_\ell}\log(1/\epsilon)\right). \end{aligned}$$

So, the overall running time is at least

$$\begin{aligned} T &\in \Omega\left(\frac{e^{p_\ell}}{p_\ell}\log(1/\epsilon)\frac{\log n}{\log(1/p_\ell)}\right) \\ &\in \Omega\left(\frac{\log n \log(1/\epsilon)}{\log(\log n \log(1/\epsilon))}\right). \end{aligned}$$

This lower bound was proved for uniform protocols, i.e., all nodes run the same algorithm. A simple reduction from the non-uniform case to the uniform one as the one used in Section 8.2 is enough to extend this result to the non-uniform case.

Both lower bounds summarized in this section were proved under the assumption of global synchronization which implies bounds for the worse case of local synchronization. In Chapter 8 we improve these lower bounds showing the optimality of the Increase From Square algorithm and we also show lower bounds for the well-studied case of multi-hop Radio Networks where nodes are deployed as a Random Geometric Graph.

Chapter 8

Lower Bounds for Clear Transmissions in Radio Networks

Any network where transmissions may collide needs a protocol for *collision-free transmissions*. Different networks provide different information about collisions. For example, on some hardware, transmitters can distinguish amongst three states at each time step: no transmission, single transmission, and collision, whereas on other hardware, transmitters can not distinguish between no transmission and collisions. In some networks, transmitters know an upper bound on their number. Sometimes, transmitters may not *snoop*, i.e., listen to the channel when not transmitting; whereas at the other extreme, transmitters may only snoop, i.e., they get no information on the channel when they are transmitting. In some networks collisions are transitive. The properties of a shared channel have a profound impact on the protocols usable on such a channel.

Sensor Networks are a heavily studied example of a shared-channel network. A Sensor Network consists of small devices with processing, sensing and communication capabilities. These *sensor nodes* are randomly deployed over an area in order to achieve sensing tasks after self-organizing as a wireless radio network. Sensor nodes have strong limitations and operate under harsh conditions. Some of the important limitations of sensor nodes include: lack of collision detection hardware, non-simultaneous transmission and reception, and one channel of communication. We call any such network a Radio Network. Additionally, nodes in sensor networks wake up at arbitrary times. Sensor networks are even more restricted in various ways that will not concern us here. The Radio Network restrictions, along with these further restrictions, are part of the Weak Sensor Model described in Chapter 3.

The gap between the lower bound for achieving something so simple as a clear transmission and upper bounds for more complicated problems such as MIS was tantalizingly narrow: respectively $\Omega(\log^2 n/\log\log n)$ and $O(\log^2 n)$, when ϵ is $\Theta(1/n^c)$. In this chapter we prove a stronger lower bound: it takes time $\Omega(\log n \log(1/\epsilon))$ to solve the problem of achieving a clear transmission with probability $1 - \epsilon$ in a one-hop setting, closing that gap. This result implies, for example, the $\Omega(\log n)$ lower bound on the expectation of any randomized algorithm for clear transmission. These lower bounds go beyond the context of Sensor Networks and apply to any network with the following characteristics:

- *Shared channel of communication:* All nodes communicate with their neighbors using broadcasts that are transmitted on a shared channel.
- *Lack of a collision detection mechanism:* Nodes do not have the ability to distinguish between a collision on the channel or lack of a transmission.
- *Non-simultaneous transmission and reception:* Nodes cannot snoop on the channel while transmitting.
- *Local synchronization:* Time is assumed to be divided into slots and all nodes have the same clock frequency.
- *Adversarial wake-up schedule:* Nodes are woken up by an adversary.

Indeed, we prove the lower bound with the following weak adversary: the adversary may chose an $i \in [1, \log n]$, and 2^i nodes wake up at time 0. These techniques also give a lower bound of $\Omega(\log \log n \log(1/\epsilon))$ on clear transmissions in the well-studied case of sensor nodes distributed uniformly at random with enough nodes to ensure connectivity, and thus for more complicated problems such as MIS. There was no non-trivial lower bound known for this problem, and the best upper bound known is $O(\log^2 n)$ with high probability, proved for the more complicated problem of sensor network initialization in Chapter 6.

8.1 Randomized Uniform Protocols in One-Hop Radio Networks

In this section, we prove a lower bound on randomized uniform protocols and we extend this result to nonuniform protocols in Section 8.2.

We first define what the clear transmission problem is in the one-hop setting. The nodes are all connected to a common broadcast channel and each transmission is available for snooping to all non-transmitting nodes. The connectivity of the nodes can be modelled as a clique. In this case we assume that all nodes know an upper bound on the number of their neighbors. In this setting, a clear transmission is achieved if exactly one node transmits in a time slot.

As explained before, we prove our lower bounds under the assumption of the existence of a weak adversary that, at a given time, wakes up (i.e. turns on) some subset of nodes. We call them *active* nodes. Upon waking up, the active nodes start the execution of a protocol to achieve a clear transmission. All non-active nodes do not participate in the protocol.

We define a *randomized uniform protocol* for clear transmission to be a sequence $p_1, p_2, \ldots$ where each node transmits with probability p_ℓ in the ℓ^{th} time step after waking up. Given our adversary, this means that all active nodes transmits with same probability as each other in each time slot.

We seek a lower bound on the number of time-slots required to achieve a clear transmission with probability $(1-\epsilon)$. We simplify the analysis in two ways. First, we further weaken the adversary by requiring that the number of nodes participating can only be one of $\{2^i | 0 \le i \le \log_2 n\}$. Secondly, we assume that all $p_\ell \in \{2^{-j} | 1 \le j \le \log_2 n\}$. If this assumption is not true of a particular algorithm A, we can always produce an algorithm A' from A by replacing one attempt in A by a constant number of attempts in A' where the probabilities of transmission in A' have been rounded off to the closest power of $1/2$.

One of the principal benefits of our weak adversary is that, the probability P_ℓ of a clear transmission by time ℓ is the same for any permutation of $p_1, p_2, \ldots, p_\ell$. Therefore, we need not bother with what order the steps are taken in, but only how many times the protocol fires with each probability.

Let t_j be the number of time-slots that nodes are transmitting with probability 2^{-j}. Let p_{ij} denote the probability that 2^i nodes fail to clear when they all transmit with probability 2^{-j}. Thus we know that:

$$\begin{aligned} p_{ij} &= 1 - 2^i \frac{1}{2^j} \left(1 - \frac{1}{2^j}\right)^{2^i - 1} \\ &= 1 - 2^{i-j}(1 - 2^{-j})^{2^i - 1} \end{aligned}$$

The total probability of failure for any number of active nodes, 2^i, needs to be bounded by:

$$\prod_j p_{ij}^{t_j} \leq \epsilon$$
$$\Longleftrightarrow \sum_j t_j \ln(p_{ij}) \leq \ln(\epsilon).$$

A lower bound is achieved by minimizing the total number of time-slots needed to satisfy the previous constraints. This can be formulated as the following *primal* linear program:

Minimize $\mathbf{1}^T\mathbf{t}$,

subject to:

$$\mathbf{Pt} \geq \boldsymbol{\epsilon}$$
$$\mathbf{t} \geq \mathbf{0}$$

where:

$$\mathbf{t} \triangleq [t_j],$$
$$\boldsymbol{\epsilon} \triangleq [-\ln(\epsilon)],$$
$$\mathbf{P} \triangleq [-\ln(p_{ij})],$$

which yields the following *dual*:

Maximize $\boldsymbol{\epsilon}^T\mathbf{u}$,

subject to:

$$\mathbf{P}^T\mathbf{u} \leq \mathbf{1}$$
$$\mathbf{u} \geq \mathbf{0}.$$

The primal linear program has a finite minimum solution, and hence its dual has a finite maximum solution. The value of the objective function for every feasible solution of the dual is a lower bound on the minimum value of the objective function for the primal. Thus any feasible solution for the dual will give a lower bound on the number of time-slots required to achieve a clear transmission with failure probability ϵ.

Suppose that the j^{th} row, $\mathbf{P}_j^T$, of $\mathbf{P}^T$ has the maximum row sum, and let $r(\mathbf{P}^T) = \mathbf{P}_j^T\mathbf{1}$. Now we set $\mathbf{u} = [1/r(\mathbf{P}^T)]$. This value of $\mathbf{u}$ satisfies all constraints of the dual. The value of the objective function of the dual is simply $\boldsymbol{\epsilon}^T\mathbf{u}$. To obtain the value of the objective function of the dual we need to find the row of $\mathbf{P}^T$ with the largest row sum which is the same as the column of $\mathbf{P}$ with the largest column sum.

Lemma 32. *The trace of every column vector of the constraint matrix* $\mathbf{P}$ *of the primal is in* $O(1)$.

Proof. We begin by stating the following useful inequality [Mit64, §2.68]:

$$e^{-x/(1-x)} \leq 1 - x \leq e^{-x}, 0 < x < 1. \tag{8.1}$$

The sum of the elements of a column j of $\mathbf{P}$ is:

$$\begin{aligned} S_j &\leq \sum_i -\ln(1 - 2^{i-j}(1-2^{-j})^{2^i-1}) \\ &\leq \sum_i -\ln\left(e^{-2^{i-j}(1-2^{-j})^{2^i-1}/(1-2^{i-j}(1-2^{-j})^{2^i-1})}\right) \text{ (By Inequality 8.1)} \\ &= \sum_i \frac{2^{i-j}(1-2^{-j})^{2^i-1}}{1-2^{i-j}(1-2^{-j})^{2^i-1}}. \end{aligned}$$

Let $y_{ij} \triangleq 2^{i-j}(1-2^{-j})^{2^i-1}$.

$$\begin{aligned} S_j &= \sum_i \frac{y_{ij}}{1-y_{ij}} \\ &\leq \sum_i \frac{y_{ij}}{1-y_{max}} \text{ (where } y_{max} = \max_{ij}\{y_{ij}\}). \end{aligned}$$

Now we derive an upper bound on y_{max}:

$$\begin{aligned} y_{max} &= \max_{ij} y_{ij} \\ &= \max_{ij} 2^{i-j}(1-2^{-j})^{2^i-1} \\ &\leq \max_{ij} 2^{i-j}e^{-2^{i-j}+2^{-j}} \text{ (By Inequality 8.1)} \\ &\leq \max_{ij} \sqrt{e}\frac{2^{i-j}}{e^{2^{i-j}}} \ (\because j \geq 1) \\ &\leq \frac{1}{\sqrt{e}} \text{ (The function is maximized, when } i = j). \end{aligned}$$

Therefore:

$$S_j \leq \frac{\sqrt{e}}{\sqrt{e}-1}\sum_i y_{ij}$$

We derive an upper bound on the right hand side sum.

$$\begin{aligned}
\sum_i y_{ij} &= \sum_i 2^{i-j}(1-2^{-j})^{2^i-1} \\
&\leq \sum_i 2^{i-j}(e^{-2^{-j}})^{2^i-1} \text{ (By Inequality 8.1)} \\
&= \sum_i 2^{i-j} e^{-2^{i-j}+2^{-j}} \\
&\leq \sqrt{e}\left(\sum_{i\geq j} 2^{i-j} e^{-2^{i-j}} + \sum_{i<j} 2^{i-j} e^{-2^{i-j}}\right) (\because j \geq 1) \\
&\leq \sqrt{e}\left(\sum_{k\geq 0} 2^{k} e^{-2^{k}} + \sum_{k\geq 1} 2^{-k} e^{-2^{-k}}\right) \\
&\leq \sqrt{e}\left(\sum_{k\geq 0} 2^{k} e^{-2^{k}} + \sum_{k\geq 1} 2^{-k}\right) \\
&\in O(1) \text{ (Because both the sums are bounded by a constant)}
\end{aligned}$$

$$\implies S_j \in O(1).$$

□

Theorem 33. *Every uniform randomized algorithm to achieve a clear transmission with probability* $1-\epsilon$ *in a one-hop Radio Network requires* $\Omega(\log n \log(1/\epsilon))$ *time-slots.*

Proof. From lemma 32, we know that $r(P^T) \in O(1)$, then $\boldsymbol{\epsilon}^T\mathbf{u} = [-\ln(\epsilon)] \cdot [1/\mathbf{P}_{max}^T\mathbf{1}] \in O(\log n \log(1/\epsilon))$. From this we can conclude that the dual linear program has a feasible solution with objective function evaluating to $\Omega(\log n \log(1/\epsilon))$. Since we showed earlier that the solution to the primal linear program gives a lower bound on the number of time-slots required to achieve a clear transmission with probability $1-\epsilon$, the statement of the theorem holds. □

8.2 Randomized Non-uniform Protocols in One Hop Radio Networks

In this section we prove our lower bound for the case in which processors may run different algorithms using their unique ID's to break symmetry. We call this a *nonuniform protocol*. Recall that we model a randomized protocol to achieve a clear transmission as a schedule, or temporal sequence, of probabilities of transmission such that, at time slot i an active node transmits with probability p_i. In the case of the randomized uniform protocols, we assume that nodes either have no ID or the protocol does not make use of it to break symmetry. Then, given that no information can be obtained from a shared-channel before a clear transmission, all active nodes transmit with the same probability in the same time slot. On the other hand, if nodes have unique ID's, they may use different schedules of probabilities of transmission and achieve a clear transmission faster. We prove in this section that in fact having unique ID's does not help.

As in [KM98], we prove our lower bound by showing a reduction from a nonuniform protocol to a uniform one. We first state our result formally.

Theorem 34. *Every randomized nonuniform protocol to achieve a clear transmission with probability* $1-\epsilon$ *in a one-hop Radio Network requires* $\Omega(\log n \log(1/\epsilon))$ *time slots.*

Proof. For the sake of contradiction, assume that there exists a randomized nonuniform protocol $\mathcal{A}$ that achieves a clear transmission with probability $1-\epsilon$ in T time slots, where $T \in o(\log n \log(1/\epsilon))$. Then, we can define a randomized uniform protocol $\mathcal{A}'$ that achieves the same running time as follows.

For each node

Choose uniformly at random an integer $i \in [1, n^2/\epsilon]$. [1]

Simulate protocol $\mathcal{A}$ *using* i *as ID.*

Each node running the protocol $\mathcal{A}'$ obtains a unique ID with probability at least $1-\epsilon$. This is true because the probability that some pair of nodes chooses the same ID is ϵ/n^2 and there are $\binom{n}{2}$ possible pairs. Given that the random choice of the ID can be done in constant time, the protocol $\mathcal{A}'$ is a randomized uniform protocol that achieves a clear transmission with probability $1-2\epsilon$ in $o(\log n \log 1/\epsilon) = o(\log n \log 1/2\epsilon)$ time slots, which is a contradiction with Theorem 33. □

[1] *Under the assumptions of the Weak Sensor Model, nodes have only* $O(\log n)$ *bits of memory. Therefore, this lower bound applies also to sensor networks when* $\epsilon \geq 1/n^\gamma$*, for some constant* $\gamma > 0$.

8.3 Randomized Protocols for Geometrically Distributed Nodes

Here, we consider the problem of achieving a clear transmission under the following conditions:

The nodes are connected by a broadcast channel to some subset of nodes and each transmission made by a node is available to its neighbors only, but it can interfere with all transmissions originating in a two-hop neighborhood. The specific case we will derive a lower bound for is the case of nodes consistent with the Weak Sensor Model distributed randomly in the plane with limited transmission range but adequate density to ensure connectivity. The connectivity of the nodes can be modelled as a *Random Geometric Graph (RGG)* (see Section 3.1.2) where the parameter conditions to ensure connectivity are always satisfied. In this case, we assume that nodes know an upper bound on the number of their neighbors with a probability given by the parameter conditions for connectivity.

In this setting, we say that a clear transmission occurred if exactly one node is transmitting and no other nodes within two hops of it are transmitting. Then, the clear transmission problem in a multi-hop setting is solved after every node either produces or receives a clear transmission.

In a $G(n,r,\ell)$ satisfying the connectivity conditions explained previously, the number of nodes contained in any circle of radius $\Theta(r)$ is $\Theta(\log n)$ with high probability, as proved in Lemma 9. Then, we complete our lower bounds with the following corollary, which can be obtained as a simple application of Theorems 33 and 34 to this setting.

Corollary 35. *Every randomized protocol to solve the clear transmission problem with probability* $1-\epsilon$ *in a Radio Network with geometrically distributed nodes requires* $\Omega(\log\log n \log(1/\epsilon))$ *time slots, where* $\epsilon \geq 1/n^\gamma$ *for some constant* $\gamma > 0$.

Proof. Replacing the appropriate density for any one-hop neighborhood in this setting, i.e. $\Theta(\log n)$ instead of n, in theorem 34 the corollary follows. □

8.4 Randomized Fair Protocols for Geometrically Distributed Nodes

In this section we prove lower bounds on the time required by any *fair* protocol to solve the Clear Transmission problem in Radio Networks where the node deployment is modelled by an RGG, i.e. nodes are deployed uniformly at random in the plane with limited transmission range but adequate density to ensure connectivity. A fair protocol is a protocol in which the probability of transmission of every node in the same execution step is the same. The difference between fair and uniform protocolos is that, in uniform protocols, nodes may use messages received to break symmetry. Notice that for the Clear Transmission problem in single-hop networks a uniform protocol is a fair protocol because the problem is solved when a message is received, but this is not the case in multi-hop networks. Given that the nodes are assumed to be unreliable, the density of *active* nodes can be upper bounded as shown in Lemma 9 but it can not be lower bounded. Hence, the probability that a node transmits after a message is received in an non-fair protocol may not be the same for all nodes running the same execution step.

We exploit the unreliability assumption to obtain our lower bounds assuming the existence of a weak adversary that, at a given time, wakes up (i.e. turns on) some subset of nodes. We call them active nodes. Upon waking up, the active nodes start the execution of a protocol to achieve a Clear Transmission. All non-active nodes do not participate in the protocol.

We define a randomized fair protocol to be a sequence $p_1, p_2, \ldots$ where each node transmits with probability p_t in the t^{th} time step after waking up. Given our adversary, this means that all active nodes transmit with the same probability as each other in each time slot.

The topology of active nodes chosen by the adversary consists of a set of disjoint pairs of cliques connected by a single node. One clique of the pair has node density in $\Theta(1)$, the other in $\Theta(\log n)$ and the intermediate node connects to all nodes in both cliques as depicted in figure 8.1. We call this construction a *clique-pair*. In order to be disjoint, nodes are woken up so the resulting clique-pairs are separated by a distance of r, the maximum range of transmission of any node.

We first give the intuition of why this structure gives a good lower bound on the number of time steps needed to solve the Clear Transmission problem. Recall that in a multi-hop setting a transmission is a Clear Transmission if no node within two hops of the transmitter transmits in the same time slot. To solve the Clear Transmission problem every node has to receive or produce a Clear Transmission so, in order to solve the Clear Transmission problem an uncolliding transmission in the low density clique or from the intermediate node with silence in the neighboring high-density clique has to occur. Given the different densities and that the protocol is fair, when the sum of probabilities of transmission in the low density clique

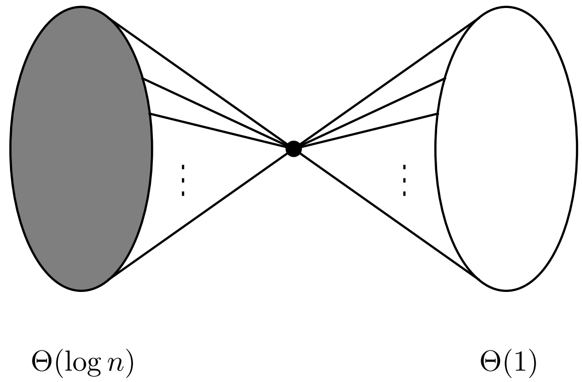


Figure 8.1: A clique-pair

reaches a constant and therefore the probability of having a succesful transmission in that clique is constant, the sum of probabilities of transmission in the 2-hop neighboring high density clique is asimptotically more than a constant and the probability of silence is low. On the other hand, when the sum of probabilities of transmission in the whole clique-pair reaches a constant and the probability of having an uncolliding transmission is high, the probability that the transmitting node is in the low-density clique or it is the intermediate node is low. Then, the probability that nodes in the low density clique produce or receive a Clear Transmission fast is low.

Lemma 36. *Given a Radio Network with nodes deployed as a connected RGG, the total number of clique-pairs activated by the adversary is in* $\Theta(n/\log n)$ *w.h.p.*

Proof. It follows from Lemma 9. □

Theorem 37. *Every fair randomized algorithm takes* $\Omega(\log^2 n)$ *expected time in order to solve the Clear Transmission problem in a multi-hop Radio Network where nodes are deployed uniformly at random.*

Proof. The probability of failing to achieve a Clear Transmission in a low-density clique in one time slot is

$$P_{fail} = 1 - \delta p(1-p)^{\delta+\Delta-1},$$

where δ and Δ are the low and high node densities respectively and p is the probability of transmission of a node. Let $\delta = c$ and $\Delta = c \log n$ for some constant $c > 0$. We now compute a lower bound on such probability of failure as follows

$$\begin{aligned} P_{fail} &= 1 - cp(1-p)^{c+c\log n-1} \\ &\geq 1 - cp(1-p)^{c(1+\log n)} \\ &\geq 1 - cpe^{-cp(1+\log n)} \\ &= 1 - \frac{cp}{(e^{1+\log n})^{cp}}. \end{aligned}$$

Using calculus, we find the value of $p = 1/c(1 + \log n)$ that minimizes this expression. Replacing,

$$\begin{aligned} P_{fail} &\geq 1 - \frac{1}{(1+\log n)e} \\ &\geq 1 - \frac{1}{1+\log n} \\ &\geq e^{-1/\log n}. \end{aligned}$$

Then, the probability of a failure in a low-density clique after t time slots is

$$P_{fail}(t) \geq e^{-t/\log n}.$$

Therefore, the probability of failure in one of the w low-density cliques after t time slots is

$$P_{fail}(w,t) \geq 1 - \left(1 - e^{-t/\log n}\right)^{w}. \tag{8.2}$$

The time step t can be seen as a discrete random variable that takes only non-negative values, then

$$\begin{aligned}
\mathbf{E}[t] &= \sum_{i=1}^{\infty} Pr(t \geq i), \text{ replacing 8.2} \\
&\geq \sum_{i=1}^{\infty} \left(1 - \left(1 - e^{-(i-1)/\log n}\right)^{w}\right) \\
&\geq \sum_{i=1}^{1+\log n \ln w} \left(1 - \left(1 - e^{-(i-1)/\log n}\right)^{w}\right) \\
&\geq (1 + \log n \ln w) \left(1 - \left(1 - \frac{1}{w}\right)^{w}\right) \\
&\geq (1 + \log n \ln w) \left(1 - \frac{1}{e}\right), \text{ using Lemma 36,} \\
&\in \Omega(\log^2 n).
\end{aligned}$$

Thus, the expected time is in $\Omega(\log^2 n)$ with probability at least $1 - n^{-\gamma}$ for some constant $\gamma > 0$. Then,

$$\begin{aligned}
\mathbf{E}[t] &\in \Omega\left(\log^2 n(1 - \frac{1}{n^\gamma}) + \frac{1}{n^\gamma}\right) \\
&\in \Omega\left(\log^2 n - \frac{1 + \log^2 n}{n^\gamma}\right) \\
&\in \Omega(\log^2 n).
\end{aligned}$$

□

Finally, we show a lower bound on the time needed to solve the problem with probability at least $1 - \epsilon$.

Theorem 38. *Every fair randomized algorithm requires $\Omega(\log n \log(n/\epsilon))$ time-slots in order to solve the Clear Transmission problem with probability at least $1 - \epsilon$ in a multihop Radio Network where nodes are deployed uniformly at random.*

Proof. We compute the minimum number of steps needed to reduce the probability of failure to ϵ, even if nodes transmit in each and every step with the probability $p = 1/c(1 + \log n)$ that minimizes that probability. More precisely, using 8.2, we want to find t such that

$$
\begin{aligned}
1 - \left(1 - e^{-t/\log n}\right)^{w} &\leq \epsilon \\
1 - e^{-t/\log n} &\geq e^{-\epsilon/w} \\
t &\geq \log n \ln \frac{w}{\epsilon} + 1 \\
t &\in \Omega\left(\log n \log(n/\epsilon)\right).
\end{aligned}
$$

□

References

[ACGL02] W. Aiello, F. Chung-Graham, and L. Lu. *Random evolution of massive graphs*, pages 97–122. Kluwer Academic Publishers, Boston/Dordrecht/London, November 2002.

[ACS94] B. Awerbuch, L. J. Cowen, and M. A. Smith. Efficient asynchronous distributed symmetry breaking. In *Proc. of the 26th Ann. ACM Symp. on Theory of Computing*, 1994.

[AR97a] M. J. Appel and R. P. Russo. The maximum vertex degree of a graph on uniform points in $[0,1]^d$. *Adv. Applied Probability*, 29:567–581, 1997.

[AR97b] M. J. Appel and R. P. Russo. The minimum vertex degree of a graph on uniform points in $[0,1]^d$. *Adv. Applied Probability*, 29:582–594, 1997.

[ASSC02] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cyirci. Wireless sensor networks: A survey. *Computer Networks*, 38(4):393–422, 2002.

[Bas99] S. Basagni. Distributed clustering for ad hoc networks. In *Proc. of the International Symposium on Parallel Architectures, Algorithms and Networks*, 1999.

[BE02] D. Braginsky and D. Estrin. Rumor routing algorthim for sensor networks. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 22–31. ACM Press, 2002.

[BFNO03] L. Barrière, P. Fraigniaud, L. Narayanan, and J. Opatrny. Dynamic construction of bluetooth scatternets of fixed degree and low diameter. In *Proc. of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 781–790. Society for Industrial and Applied Mathematics, 2003.

[BIN06] J. L. Bordim, Y. Ito, and K. Nakano. An energy efficient leader election protocol for radio network with a single transceiver. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science*, E89-A(5):1355–1361, 2006.

[BKR$^+$01] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. P. Williamson. Adversarial queueing theory. *Journal of the ACM*, 48(1):13–38, 2001.

[BLRS03] D. M. Blough, M. Leoncini, G. Resta, and P. Santi. The k-neigh protocol for symmetric topology control in ad hoc networks. In *Proc. of the 4th ACM international symposium on Mobile ad hoc networking and computing*, 2003.

[Blu] Bluetooth Special Interest Group. http://www.bluetooth.com.

[BP97] D. Bruschi and M. Del Pinto. Lower bounds for the broadcast problem in mobile radio networks. *Distributed Computing*, 10(3):129–135, 1997.

[BS01] J. Bray and C. Sturman. *Bluetooth: Connect Without cables*. Prentice Hall, 2001.

[BY76] J. Bentley and A. Yao. An almost optimal algorithm for unbounded search. *Information Processing Letters*, 5:82–87, 1976.

[BYGI91] R. Bar-Yehuda, O. Goldreich, and A. Itai. Efficient emulation of single-hop radio network with collision detection on multi-hop radio network with no collision detection. *Distributed Computing*, 5(2):67–71, 1991.

[BYGI92] R. Bar-Yehuda, O. Goldreich, and A. Itai. On the time-complexity of broadcast in multi-hop radio networks: An exponential gap between determinism and randomization. *Journal of Computer and System Sciences*, 45:104–126, 1992.

[BYII93] R. Bar-Yehuda, A. Israeli, and A. Itai. Multiple communication in multihop radio networks. *SIAM Journal on Computing*, 22:875–887, 1993.

[CFL+04] D. Calzada, A. Fernández, L. López, A. L. Martínez, and A. Santos. A continuous model of adversarial queueing theory. In *In Proc. Jornadas de Concurrencia y Sistemas Distribuídos*, 2004.

[CGG+02] B. S. Chlebus, L. Gąsieniec, A. Gibbons, A. Pelc, and W. Rytter. Deterministic broadcasting in unknown radio networks. *Distributed Computing*, 15:27–38, 2002.

[CGK04] M. Chrobak, L. Gąsieniec, and D. Kowalski. The wake-up problem in multi-hop radio networks. In *Proc. of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2004.

[CGOR00] B. S. Chlebus, L. Gąsieniec, A. Östlin, and J. M. Robson. Deterministic radio broadcasting. In *Proc. of 27th International Colloquium on Automata Languages and Programming*, 2000.

[CGR00] M. Chrobak, L. Gąsieniec, and W. Rytter. Fast broadcasting and gossiping in radio networks. In *Proc. of the 41st IEEE Annual Symp. on Foundation of Computer Science*, 2000.

[CK04] B. Chlebus and D. Kowalski. A better wake-up in radio networks. In *Proceedings 23rd Annual ACM Symposium on Principles of Distributed Computing*, 2004.

[CMS01] A.E.F. Clementi, A. Monti, and R. Silvestri. Selective families, superimposed codes, and broadcasting on unknown radio networks. In *Proc. of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2001.

[CR03] A. Czumaj and W. Rytter. Broadcasting algorithms in radio networks with unknown topology. In *Proc. of the 44th IEEE Annual Symp. on Foundation of Computer Science*, 2003.

[Cru91a] R. L. Cruz. A calculus for network delay. part i: Network elements in isolation. *IEEE Transactions on Information Theory*, 37(1):114–131, 1991.

[Cru91b] R. L. Cruz. A calculus for network delay. part ii: Network analysis. *IEEE Transactions on Information Theory*, 37(1):132–141, 1991.

[EGHK99] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *Proc. of the 5th ACM Annual International Conference on Mobile Computing and Networking*, 1999.

[ER59] P. Erdos and A. Renyi. On random graphs–i. *Publicationes Matematicae*, 6:290–297, 1959.

[FCFM05] M. Farach-Colton, R. J. Fernandes, and M. A. Mosteiro. Bootstrapping a hop-optimal network in the weak sensor model. *Lecture Notes in Computer Science*, 3669:827–838, 2005.

[FCFM06] M. Farach-Colton, R. J. Fernandes, and M. A. Mosteiro. Lower bounds for clear transmissions in radio networks. *Lecture Notes in Computer Science*, 3887:447–454, 2006.

[FMPC04] F. Ferraguto, G. Mambrini, A. Panconesi, and C.Petrioli. A new approach to device discovery and scatternet formation in bluetooth networks. In *Proc. of the 18th International Parallel and Distributed Processing Symposium*, 2004.

[FPS02] I. Finocchi, A. Panconesi, and R. Silvestri. Experimental analysis of simple, distributed vertex coloring algorithms. In *Proc. of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2002.

[FT40] L. Fejes-Tóth. über einen geometrischen satz. *Mathematische Zeitschrift*, 46(1):83–85, 1940.

[GGSE01] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(4):11–25, 2001.

[GK98] P. Gupta and P. R. Kumar. Critical power for asymptotic connectivity in wireless networks. In *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W. H. Fleming.*, pages 547–566. Birkhauser, Boston, 1998.

[GKR04] A. Goel, B. Krishnamachari, and S. Rai. Sharp thresholds for monotone properties in random geometric graphs. In *Proc. of the 36th Ann. ACM Symp. on Theory of Computing*, 2004.

[GPP01] L. Gąsieniec, A. Pelc, and D. Peleg. The wakeup problem in synchronous broadcast systems. *SIAM J. Discret. Math.*, 14(2):207–222, 2001.

[GW85] A.G. Greenberg and S. Winograd. A lower bound on the time needed in the worst case to resolve conflicts deterministically in multiple access channels. *Journal of the ACM*, 32:589–596, 1985.

[Hen92] Bruce Hendrickson. Conditions for unique graph realizations. *SIAM Journal on Computing*, 21(1):65–84, 1992.

[HNO99] T. Hayashi, K. Nakano, and S. Olariu. Randomized initialization protocols packet radio networks. In *IPPS*, 1999.

[IGE00] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Mobile Computing and Networking*, pages 56–67, 2000.

[Ind02] P. Indyk. Explicit constructions of selectors and related combinatorial structures, with applications. In *Proc. of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2002.

[JJ03] Bill Jackson and Tibor Jordán. Connected rigidity matroids and unique realizations of graphs. Technical Report TR-2002-12, Egerváry Research Group, Pázmány P. sétány 1/C, H-1117, Budapest, Hungary, March 2003.

[JKZ02] T. Jurdziński, M. Kutyłowski, and J. Zatopiański. Efficient algorithms for leader election in radio networks. In *Proceedings 21st Annual ACM Symposium on Principles of Distributed Computing*, 2002.

[JS05] T. Jurdziński and G. Stachowiak. Probabilistic algorithms for the wakeup problem in single-hop radio networks. *Theory of Computing Systems*, 38(3):347–367, 2005.

[KG85] J. Komlòs and A.G. Greenberg. An asymptotically nonadaptive algorithm for conflict resolution in multiple-access channels. *IEEE Trans. Inf. Theory*, 31:303–306, 1985.

[KKKP00] L. M. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc. Power consumption in packet radio networks. *Theor. Comput. Sci.*, 243:289–305, 2000.

[KKR$^+$99] J. Kleinberg, S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. The web as a graph: Measurements, models and methods. In *Annual Conference on Computing and Combinatorics*, 1999.

[KM98] E. Kushilevitz and Y. Mansour. An $\Omega(D \log(N/D))$ lower bound for broadcast in radio networks. *SIAM Journal on Computing*, 27(3):702–712, 1998.

[KMPS04] V. S. A. Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan. End-to-end packet-scheduling in wireless ad-hoc networks. In *Proc. of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2004.

[Kow05] D. R. Kowalski. On selection problem in radio networks. In *Proceedings 24th Annual ACM Symposium on Principles of Distributed Computing*, 2005.

[KP03] D. R. Kowalski and A. Pelc. Broadcasting in undirected ad hoc radio networks. In *Proceedings 22nd Annual ACM Symposium on Principles of Distributed Computing*, 2003.

[KP04a] D. R. Kowalski and A. Pelc. Faster deterministic broadcasting in ad hoc radio networks. *SIAM J. Discret. Math.*, 18(2):332–346, 2004.

[KP04b] D. R. Kowalski and A. Pelc. Time of deterministic broadcasting in radio networks with local knowledge. *SIAM Journal on Computing*, 33(4):870–891, 2004.

[KR03] M. Kutyowski and W. Rutkowski. Adversary immune leader election in ad hoc radio networks. In *Proc. of the 11th Annual European Symposium on Algorithms*, volume 2832 of *Lecture Notes in Computer Science*, pages 397–408. Springer-Verlag, Berlin, 2003.

[KW03] H. Karl and A. Willig. A short survey of wireless networks. Technical Report TKN-03-018, Technical University Berlin, October 2003.

[Lam70] G. Laman. On graphs and rigidity of plane skeletal structures. *J. Eng. Math.*, 4:331–340, 1970.

[LP02] D. Liu and M. Prabhakaran. On randomized broadcasting and gossiping in radio networks. *Lecture Notes in Computer Science*, 2387:340–349, 2002.

[LS01] C. Law and K. Y. Siu. A Bluetooth scatternet formation algorithm. In *Proceedings of the IEEE Symposium on Ad Hoc Wireless Networks*, November 2001.

[Lub86] M. Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM Journal on Computing*, 15:1036–1053, 1986.

[LY82] L. Lovász and Y. Yemini. On generic rigidity in the plane. *SIAM Journal on Algebraic and Discrete Methods*, 3(1):91–98, 1982.

[Mar94] C. U. Martel. Maximum finding on a multiple access broadcast network. *Inf. Process. Lett.*, 52:7–13, 1994.

[MB00] B. Miller and C. Bisdikian. *Bluetooth Revealed: The Insider's Guide to an Open Specification for Global Wireless Communication*. Prentice Hall, 2000.

[Met75] B. Metcalfe. Steady-state analysis of a slotted and controlled aloha system with blocking. *SIGCOMM Comput. Commun. Rev.*, 5(1):24–31, 1975.

[Mit64] D. S. Mitrinović. *Elementary Inequalities*. P. Noordhoff Ltd. - Groningen, 1964.

[MP05] S. Muthukrishnan and Gopal Pandurangan. The bin-covering technique for thresholding random geometric graph properties. In *Proc. of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2005.

[MR95] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[MU05] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.

[MW05] T. Moscibroda and R. Wattenhofer. Maximal independent sets in radio networks. In *Proceedings 24th Annual ACM Symposium on Principles of Distributed Computing*, 2005.

[NO00a] K. Nakano and S. Olariu. Energy-efficient initialization protocols for radio networks with no collision detection. In *ICPP*, 2000.

[NO00b] K. Nakano and S. Olariu. Randomized leader election protocols in radio networks with no collision detection. In *ISAAC*, volume 1969 of *Lecture Notes in Computer Science*, pages 362–373, 2000.

[NO02a] K. Nakano and S. Olariu. A survey on leader election protocols for radio networks. In *Proc. of the 6th International Symposium on Parallel Architectures, Algorithms and Networks*, 2002.

[NO02b] K. Nakano and S. Olariu. Uniform leader election protocols for radio networks. *IEEE Trans. Parallel Distr. Syst.*, 13(5):516–526, 2002.

[PB03] V. A. S. Ponduru and A. Bharathidasan. Sensor networks: An overview. Technical report, University of California, Davis, 2003.

[Pen99] M. D. Penrose. On k-connectivity for a geometric random graph. *Random Struct. Algorithms*, 15(2):145–164, 1999.

[Pen03] M. Penrose. *Random Geometric Graphs*. Oxford University Press, May 2003.

[RMSG] P. Rentala, R. Musumuri, U. Saxena, and S. Gandham. Survey on sensor networks. http://citeseer.nj.nec.com/479874.html.

[Rob75] L. G. Roberts. Aloha packet system with and without slots and capture. *Computer Communication Review*, 5(2):28–42, 1975.

[Sax03] J. B. Saxe. Embeddability of weighted graphs in k-space is strongly NP-hard. In *Proc. of the 17th Allerton Conf. on Communication, Control and Computing*, pages 480–489, 2003.

[SBTL01] T. Salonidis, P. Bhagwat, L. Tassiulas, and R. LaMaire. Distributed topology construction of bluetooth personal area networks. In *INFOCOM*, 2001.

[SGAP00] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie. Protocols for self-organization of a wireless sensor network. *Personal Communications, IEEE*, 7(5):16–27, 2000.

[SWLF04] W. Song, Y. Wang, X. Li, and O. Frieder. Localized algorithms for energy efficient topology in wireless ad hoc networks. In *Proc. of the 5th ACM international symposium on Mobile ad hoc networking and computing*, 2004.

[TAGH02] S. Tilak, N. Abu-Ghazaleh, and W. Heinzelman. A taxonomy of wireless microsensor network problems. *Mobile Computing and Communications Review*, 6(2), 2002.

[WAF04] P. Wan, K. M. Alzoubi, and O. Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. *Mob. Net. and App.*, 9(2), 2004.

[Wil86] D. E. Willard. Log-logarithmic selection resolution protocols in a multiple access channel. *SIAM Journal on Computing*, 15:468–477, 1986.

[WTH02] Z. Wang, R. J. Thomas, and Z. Haas. Bluenet - A new scatternet formation algorithm. In *HICSS*, 2002.

[XK04] F. Xue and P. R. Kumar. The number of neighbors needed for connectivity of wireless networks. *Wireless Networks*, 10(2):169–181, 2004.

[YKR06] O. Younis, M. Krunz, and S. Ramasubramanian. Node clustering in wireless sensor networks: Recent developments and deployment challenges. *IEEE Network Magazine*, 2006.

[ZBC01] G. V. Zaruba, S. Basagni, and I. Chlamtac. Bluetrees - Scatternet formation to enable Bluetooth-based ad hoc networks. In *Proc. of IEEE ICC*, 2001.

Vita

Miguel A. Mosteiro

2006 Ph. D. in Computer Science, Rutgers University.

2006 M. Farach-Colton, Rohan Fernandes and M. A. Mosteiro. Lower Bounds for Clear Transmissions in Radio Networks. *Proceedings of the 7th Latin American Theoretical Informatics Symposium (LATIN)*, volume 3887 of *Lecture Notes in Computer Science*, pages 447-454, Springer-Verlag.

2006 M. A. Bender, M. Farach-Colton and M. A. Mosteiro. Insertion Sort is $O(n \log n)$.*Theory of Computing Systems*, Issue: Online First, Springer-Verlag. Preliminary version in *Proceedings of the 3rd International Conference on Fun with Algorithms (FUN)*, pages 16-23, 2004.

2005 Instructor, Computer Science Department, Rutgers University.

2005 M. Farach-Colton, Rohan Fernandes and M. A. Mosteiro. Bootstrapping Hop-optimal Networks in the Weak Sensor Model. *Proceedings of the 13th Annual European Symposium on Algorithms (ESA)*, volume 3669 of *Lecture Notes in Computer Science*, pages 827-838, Springer-Verlag.

2003-2006 DIMACS, Center for Discrete Mathematics and Theoretical Computer Science Graduate Support.

2003-2004 Instructor, Management Science and Information Systems Department, Rutgers University.

2003 M. Sc. in Computer Science, Rutgers University.

2001-2006 Teaching Assistant, Computer Science Department, Rutgers University.

2001-2003 Excellence Fellowship for Doctoral Study in Computer Science, Graduate School New Brunswick, Rutgers University.

2000-2001 Head Teaching Assistant, Computer Science Department, Universidad de Buenos Aires, Argentina.

1999-2001 First Class Teaching Assistant, Computer Science Department, Universidad de Buenos Aires, Argentina.

1991-1992 First Class Teaching Assistant, Department of Electronics, Universidad de Buenos Aires, Argentina.

1989-1996 Head Teaching Assistant, Department of Computer Engineering, Buenos Aires Institute of Technology, Argentina.

1988-1996 Development Engineer, Codas Electrónica SA, Buenos Aires, Argentina.

1984 Engineer in Electronics, Universidad Tecnológica Nacional, Argentina.

1974-1978 Excellence Fellowship for the Education of Emigrants Descendants, Ministry of Education and Science, Spain.