Constant Density Spanners for Wireless Ad-Hoc Networks

Kishore Kothapalli Christian Scheideler
Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218, USA
{kishore, scheideler}@cs.jhu.edu

Melih Onus Andrea Richa
Department of Computer Science
Arizona State University
Tempe, AZ 85281, USA
{melih, aricha}@asu.edu

ABSTRACT

An important problem for wireless ad hoc networks has been to design overlay networks that allow time- and energy-efficient routing. Many local-control strategies for maintaining such overlay networks have already been suggested, but most of them are based on an oversimplified wireless communication model.

In this paper, we suggest a model that is much more general than previous models. It allows the path loss of transmissions to significantly deviate from the idealistic unit disk model and does not even require the path loss to form a metric. Also, our model is apparently the first proposed for algorithm design that does not only model transmission and interference issues but also aims at providing a realistic model for physical carrier sensing. Physical carrier sensing is needed so that our protocols do not require *any* prior information (not even an estimate on the number of nodes) about the wireless network to run efficiently.

Based on this model, we propose a local-control protocol for establishing a constant density spanner among a set of mobile stations (or *nodes*) that are distributed in an arbitrary way in a 2-dimensional Euclidean space. More precisely, we establish a backbone structure by efficiently electing cluster leaders and gateway nodes so that there is only a constant number of cluster leaders and gateway nodes within the transmission range of any node and the backbone structure satisfies the properties of a topological spanner.

Our protocol has the advantage that it is locally self-stabilizing, i.e., it can recover from *any* initial configuration, even if adversarial nodes participate in it, as long as the honest nodes sufficiently far away from adversarial nodes can in principle form a single connected component. Furthermore, we only need constant size messages and a constant amount of storage at the nodes, irrespective of the distribution of the nodes. Hence, our protocols would even work in extreme situations such as very simple wireless devices (like sensors) in a hostile environment.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*; F.2.8 [**Analysis of Algorithms and Problem Complexity**]: Non-numerical Algorithms and Problems—*Computation of discrete structures*

SPAA'05, July 18–20, 2005, Las Vegas, Nevada, USA.

General Terms

Algorithms, Theory

Keywords

ad hoc networks, spanner, dominating set, self-stabilization

1. INTRODUCTION

An important problem for wireless ad hoc networks has been to design overlay networks that allow time- and energy-efficient routing. Many local-control strategies for maintaining such overlay networks have already been suggested, but mostly high-level wireless models have been used for their analysis. However, since mobile ad-hoc networks have many features that are hard to model in a clean way, it is not clear how well these strategies may actually perform in practice. Major challenges are how to model wireless communication and how to model mobility. Here, theoretical work is still in its infancy. So far, people in the algorithms community have mostly looked at static wireless networks (i.e. the wireless stations are always available and do not move). Wireless communication has mostly been modeled using the packet radio network or the even simpler unit disk model. In the packet radio network model, the wireless units, or nodes, are represented by a graph, and two nodes are connected by an edge if and only if they are within transmission range of each other. Transmissions of messages *interfere* at a node if at least two of its neighbors transmit a message at the same time. A node can only receive a message if it does not interfere with any other message.

The packet radio network model is a simple and clean model that allows to design and analyze algorithms for overlay networks with a reasonable amount of effort. However, since it is a high-level model, it does have some serious problems with certain scenarios in practice because in reality the transmission range of a message is not the same as its interference range. Consider, for example, two nodes s and t and a set U of n nodes with all nodes in U being within the transmission range of s but only one node in U, v, being within the transmission range of t. All other nodes in U just interfere with t. Then no uniform protocol (i.e., at each step all nodes try to access the wireless medium with the same probability) can send a message from s to t in an expected $o(n)$ number of steps, whereas a constant expected time is easy to achieve if we can ignore interference problems between U and t.

There are a limited number of papers in the theory area that use a model that differentiates between the transmission range and the interference range [1, 12, 13, 14, 22], but they still assume a disk model in a sense that the transmission range and interference range can be modeled by two distance values that hold irrespective of the position of a node. We propose a more general model.

1.1 Wireless communication model

In order to motivate our model, we first review some commonly used transmission techniques in wireless communication. We will concentrate here on the IEEE 802.11 standard because IEEE 802.11-based radio LANs are currently dominating the market and will most probably do so also in the future. The IEEE 802.11 standard distinguishes between a Physical (PHY) layer and a Medium Access Control (MAC) layer for the transmission of messages. The 802.11 MAC protocols are based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA).

Carrier sensing

The basic approach of the CSMA/CA scheme is as follows. Whenever a node receives a packet to be transmitted, it first listens to the channel to ensure no other node is transmitting. If the channel is clear, it transmits the packet. Otherwise, it uses an exponential back-off scheme until it either finds a time point in which the channel is clear so that it can transmit its packet or aborts the transmission due to too many failed attempts.

In WLAN devices, there is usually just one antenna for both sending and receiving, and hence the stations are not able to listen while sending. For this and other reasons there is no collision detection capability like in the Ethernet. Therefore, acknowledgment packets (ACK) have to be sent from the receiver to the sender to confirm that packets have been correctly received.

In wireless ad hoc networks that rely on a carrier-sensing random access protocol, such as IEEE 802.11, the wireless medium characteristics generate complex phenomena such as the well-known hidden-station and exposed-station problems. In order to handle these problems, the MAC layer uses *physical* and *virtual carrier sensing* techniques.

The physical carrier sensing part of the CSMA scheme is provided by a Clear Channel Assessment (CCA) circuit. This circuit monitors the environment to determine when it is clear to transmit. It can be programmed to be a function of the Receive Signal Strength Indication (RSSI) and other parameters. The RSSI measurement is derived from the state of the Automatic Gain Control (AGC) circuit. Whenever the RSSI exceeds a certain threshold, a special Energy Detection (ED) bit is switched to 1, and otherwise it is set to 0. By manipulating a certain configuration register, this threshold may be set to an absolute power value of t dB, or it may be set to be t dB above the measured noise floor, where t can be set to any value in the range 0-127. The ability to manipulate the CCA rule allows the MAC layer to optimize the physical carrier sensing to its needs.

Virtual carrier sensing is usually achieved by using two control packets, Request To Send (RTS) and Clear To Send (CTS), which are exchanged before the data transmission is taking place. More precisely, before transmitting a data frame, the source station sends an RTS packet to the receiving station announcing the upcoming frame transmission. Upon receiving the RTS packet, the destination replies by a CTS packet to indicate that it is ready to receive the data frame. Both the RTS and CTS packets contain the total duration of the transmission, i.e. the overall time needed to transmit the data frame and the related ACK, so that other stations within the transmission range of either the source or the destination stay silent until the transmission is over.

Transmission range, interference range, and physical carrier sensing range

Every data transmission mechanism has a minimum signal-to-noise ratio (SNR) at which a data frame can still be transmitted with a reasonably low frame error rate. The minimum SNRs for 802.11b, for example, are 10dB for 11Mbps, 8dB for 5.5Mbps, 6dB for 2Mbps, and 4dB for 1Mbps, and for 802.11a, 23dB is usually the minimum SNR for 54Mbps. In the 802.11a standard [28], the minimum dB values are defined as the received signal strength level at which the frame error rate (FER) of a 1000-octet frame is less than 10%.

The SNRs above specify the *transmission range* of the data transmission mechanism, i.e. the maximum range within which data frames can still be received correctly. The transmission range is highly dependent on the environment. A reasonable model for determining the transmission range is the log-normal shadowing model [23, 30]. In this model, the received power at a distance of d relative to the received power at a reference distance of d_0 is given in dB as

$$-10\log(d/d_0)^\theta + X_\sigma$$

where θ is the path loss coefficient and X_σ is a Gaussian random variable with zero mean and standard deviation σ (in dB) that models the influence of the background noise. θ usually ranges from 2 (free space) to 5 (indoors) [31].

When using forward error correction mechanisms as proposed in the IEEE 802.11e MAC standard currently under development, the transition between being able to correctly receive a data frame with high probability and not being able to correctly receive a data frame with high probability is very sharp. As shown in [6], it can be less than 1 dB. Thus, in an ideal environment the transmission range is an area with a relatively sharp border that in reality, however, may be blurred due to environmental effects.

A limitation of the shadowing model is that it is only applicable in uniform environments. In non-uniform environments, the signal strength can exhibit a non-monotonic behavior. For example, it can happen that the sender position A has a smaller distance to a position B than to a position C and yet the strength of the signal from A received at B is lower than the signal strength received at C. This can even happen if B and C are close by.

For the interference and physical carrier sensing ranges there does not seem to be a commonly accepted definition in practice. So we will use a conservative model for these ranges to make sure that our results in this model are meaningful in practice.

Our model

In our model, we assume that we are given a set V of mobile stations, or *nodes*, that are distributed in an arbitrary way in a 2-dimensional Euclidean space. For any two nodes $v, w \in V$ let $d(v, w)$ be the Euclidean distance between v and w. Furthermore, consider any cost function c with the property that there is a fixed constant $\delta \in [0, 1)$ so that for all $v, w \in V$,

- $c(v, w) \in [(1-\delta)\cdot d(v, w),\ (1+\delta)\cdot d(v, w)]$ and
- $c(v, w) = c(w, v)$, i.e. c is symmetric.

c determines the transmission and interference behavior of nodes and δ bounds the non-uniformity of the environment. Notice that we do not require c to be monotonic in the distance or to satisfy the triangle inequality. This makes sure that our model even applies to highly irregular environments. In Figure 1(a), for example, the distance between u and v is greater than the distance between u and w. Yet, the cost of communicating between u and w, $c(u, w)$, is bigger than $c(u, v)$. Similar cost functions were also used in [21].

We assume that the nodes use some fixed-rate power-controlled communication mechanism over a single frequency band. When using a transmission power of P, there is a transmission range $r_t(P)$ and an interference range $r_i(P) > r_t(P)$ that grow monotonically with P. The interference range has the property that

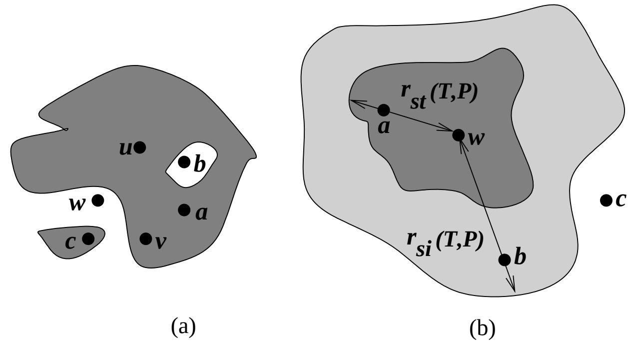

Figure 1: Figure (a) shows the notion of transmission range in terms of cost of communication. Notice that node u can communicate directly with nodes v and a and c but not with nodes b or w. Figures (b) shows the sensing ranges. When all nodes use a transmission power P and node w uses a threshold of T, in this example, node w can *always* sense transmissions of node a while it *may* sense transmissions of node b and *can never* sense transmissions of node c.

every node $v \in V$ can only cause interference at nodes w with $c(v,w) \le r_i(P)$, and the transmission range has the property that for every two nodes $v, w \in V$ with $c(v,w) \le r_t(P)$, v is guaranteed to receive a message from w sent out with a power of P (with high probability) as long as there is no other node $v' \in V$ with $c(v,v') \le r_i(P')$ that transmits a message at the same time with a power of P'.

For simplicity, we assume that the ratio $\rho = r_i(P)/r_t(P)$ is a fixed constant greater than 1 for all relevant values of P. This is not a restriction because we do not assume anything about what happens if a message is sent from a node v to a node w within v's transmission range but another node u is transmitting a message at the same time with w in its interference range. In this case, w may or may not be able to receive the message from v, so any worst case must be assumed in the analysis. The only restriction we need, which is important for any overlay network algorithm to eventually stabilize, is that the transmission range is a sharp threshold. That is, beyond the transmission range a message cannot be received any more (with high probability). This is justified by the fact that when using modern forward error correction techniques, the difference between the signal strength that allows to receive the message (with high probability) and the signal strength that does not allow any more to receive the message (with high probability) can be very small (less than 1 dB).

Nodes can not only send and receive messages but also perform physical carrier sensing, which has not been considered before in models proposed in the algorithms community. Given some sensing threshold T (that can be flexibly set by a node) and a transmission power P, there is a *carrier sense transmission (CST) range* $r_{st}(T,P)$ and a *carrier sense interference (CSI) range* $r_{si}(T,P)$ that grow monotonically with T and P. The range $r_{st}(T,P)$ has the property that if a node v transmits a message with power P and a node w with $c(v,w) \le r_{st}(T,P)$ is currently sensing the carrier with threshold T, then w senses a message transmission (with high probability). The range $r_{si}(T,P)$ has the property that if a node v senses a message transmission with threshold T, then there was at least one node w with $c(v,w) \le r_{si}(T,P)$ that transmitted a message with power P (with high probability). More precisely, we assume that the monotonicity property holds. That is, if transmissions from a set U of nodes within the $r_{si}(T,P)$ range cause v to sense a transmission, then any superset of U will also do so. The two sensing ranges are shown in Figure 1(b).

For simplicity, we will assume in the following that for the carrier sense ranges, $r_{si}(T,P)/r_{st}(T,P) = r_i(P)/r_t(P)$ for all relevant values of T.

1.2 Our contributions

Our contributions are two-fold: we present a new model for wireless networks, and we demonstrate how to develop and analyze algorithms on top of this model by presenting self-stabilizing local-control algorithms for building constant density dominating sets and spanners.

In our algorithms, the nodes do not have to have *any* a priori knowledge about the other nodes, not even an estimate on their total number. Also, fixed identification numbers of any form are not required so that our protocols may even be applicable to the important field of sensor networks. It is sufficient for us if the nodes choose identification numbers so that there are no local conflicts (which can be easily achieved with random, local-control coloring strategies). We only require that the mobile hosts can synchronize in rounds of constant length. This can be done, for example, with the help of GPS signals or any form of beacons (that are sufficiently far apart in time for a round of our protocols to complete).

In order to obtain a constant density spanner under an arbitrary distribution of nodes, we proceed in two stages. First, we show that there is a simple, distributed protocol to obtain a constant density dominating set, and then we show how to extend this protocol in order to also obtain a constant density spanner.

It is worth noting that our protocols only need a constant amount of storage at each node, irrespective of the distribution of the nodes. The constant only depends on the δ in our model. Moreover, our protocols can self-stabilize even if some of the nodes show *arbitrary* adversarial behavior. We only require the honest nodes that are outside a certain range of the adversarial nodes to be placed so that they can in principle form a single connected component. So our protocols would even work for very primitive devices in hostile environments.

Constant density dominating set

Given an undirected graph $G = (V, E)$, a subset $U \subseteq V$ is called a *dominating set* if all nodes $v \in V$ are either in U or have an edge to a node in U. A dominating set U is called *connected* if U forms a connected component in G. The *density* of a dominating set is the maximum over all nodes $v \in U$ of the number of neighbors that v has in U.

Given an arbitrary distribution of a set V of nodes in a 2-dimensional Euclidean space, let the graph $Q_r = (V, E_r)$ contain all edges $\{v, w\}$ with $d(v,w) \le r$. Suppose that we select a maximal independent set U in Q_r. Then this is also a dominating set of constant density because in the 2-dimensional Euclidean space a node can have at most five neighbors within a distance of r that are part of an independent set in Q_r [3]. Note that a constant density dominating set is also a constant factor approximation of a minimum dominating set, a well-studied problem in the algorithms and wireless networking community.

Now, let us consider the graph $G_r = (V, E'_r)$ that contains all edges $\{v, w\}$ such that $c(v,w) \le r$. Since $c(v,w) \le (1+\delta)\, d(v,w)$, it follows from [3]:

FACT 1.1. *Every node v can have at most five neighbors within a Euclidean distance of $r/(1+\delta)$ that are part of an independent set in G_r.*

Otherwise, there must be a pair $v, w \in V$ with $c(v,w) \leq (1+\delta)d(v,w) \leq (1+\delta)\cdot r/(1+\delta) = r$ that are in an independent set in G_r, a contradiction. Furthermore, because $c(v,w) \geq (1-\delta)d(v,w)$, a node can only be connected in G_r to nodes up to a Euclidean distance of $r/(1-\delta)$. Hence, it is easy to see that for every node v there is a set C_v of neighbors of v in G_r of constant size so that for every neighbor w of v in G_r there is a neighbor $w' \in C_v$ with $d(w,w') \leq r/(1+\delta)$. Combining this with Fact 1.1, we get:

FACT 1.2. *For any independent set in G_r it holds that every node v in G_r can have at most a constant number of neighbors in this set, where the constant depends on δ.*

Now, recall that any maximal independent set in a graph G_r is also a dominating set in G_r, and according to the fact above, any maximal independent set in G_r has a constant density (i.e., every node only has a constant number of neighbors in that set). Hence, in order to obtain a dominating set of constant density, it suffices to design an algorithm that constructs a maximal independent set in G_r. It turns out that constructing such a set is quite tricky, given the uncertainties in our model, but we can construct something close to that so that the following result holds.

THEOREM 1.3. *For any desired transmission range r and any initial situation, the dominating set protocol generates a constant density dominating set in G_r in $O(\log^4 n)$ communication rounds, with high probability.*

Hence, our protocol self-stabilizes within $O(\log^4 n)$ rounds. Interestingly, this result is only possible because our protocol uses physical carrier sensing. It is known that if physical carrier sensing is not available and the nodes have no estimate of the size of the network, then it takes $\Omega(n)$ steps on expectation for a single message transmission to be successful [17] in any protocol.

Constant density spanner

A subgraph H of a graph G is called a *(topological) t-spanner* of G if for every pair of nodes v, w in G there is a path in H from v to w whose length is at most t times the minimum length of a path from v to w in G. In this case, t is also called the *stretch factor* of H.

We then extend the dominating set protocol by additional protocols that connect the nodes in the dominating set via so-called gateway nodes so that the following result holds.

THEOREM 1.4. *For any desired transmission range r and any initial situation, the spanner protocol generates a constant density spanner in G_r in $O(D \log D \log n + \log^4 n)$ communication rounds, with high probability, where D is the maximum number of nodes that are within the transmission range of a node.*

All of our protocols can self-stabilize even under adversarial behavior as long as the nodes outside a range of $r' = \Theta(r)$ of adversarial nodes form a connected component in G_r.

1.3 Previous work

The problem of finding a minimum dominating set has been shown to be NP-complete even when restricted to unit disk graphs [7] and, hence, approximation algorithms are of interest. Recent research focused on developing distributed (rather than centralized) algorithms for finding good approximations of minimum dominating sets in arbitrary graphs (see, for example, [9, 16, 20]). A simple and elegant distributed approximation algorithm was proposed by Luby [27].

Alzoubi et al. [4] presented the first constant approximation algorithm for the minimum connected dominating set problem in unit-disk graphs with $O(n)$ and $O(n \log n)$ time and message complexity, respectively. Cheng et al. [5] proposed a polynomial time approximation scheme for the connected dominating set problem in unit-disk graphs.

Huang et al. [15] formally analyze a popular algorithm used for clustering in ad-hoc mobile network scenarios. They show that this algorithm actually gives a 7-approximation for the minimum dominating set problem in unit-disk graphs, while adapting optimally to the mobility of the nodes in the network.

Recently, Kuhn et. al. [19] presented a distributed algorithm that computes a constant factor approximation of a minimum dominating set in $O(\log^2 n)$ time without needing any synchronization but it requires that nodes know an estimate of the total number of nodes in the network. In [29], Parthasarathy and Gandhi also present distributed algorithms to compute a constant factor approximation to the minimum dominating set. The running time of their algorithm depends on the amount of information available to the nodes, and nodes have to know an estimate of the size of the network. Both papers extend the unit disk model taking into account signal interference.

Spanners

Suppose that we have a set of nodes V that are distributed in an arbitrary way in a Euclidean space. For $v, w \in V$, let $d(v, w$ denote the Euclidean distance between v and w. The goal of the geometric spanner problem is to find a graph $G = (V, E)$ so that for each pair of nodes $v, w \in V$ there is a path in G from v to w whose length is at most $t \cdot d(v,w)$ for some fixed constant t. In this case, G is called a *geometric t-spanner* of G where t is the stretch factor.

For geometric spanners, several structures have been proposed. Geometric spanners based on the Delaunay triangulation have been studied, e.g., [11, 24, 33]. Spanners based on the Yao graph [36] and Gabriel graph [10] are presented in [25, 35, 32].

For topological spanners, Dubhashi et. al. [9] presented a spanner with logarithmic stretch factor. Alzoubi et. al. [2] presented a spanner with constant stretch factor of 5 where the protocol is very similar to ours but uses a high-level model for wireless networks.

Our protocol for selecting gateway nodes also has similarities to the protocols presented in [34, 11]. However both these papers are based on high-level wireless models.

1.4 Structure of the paper

We start with an overview of our protocol for the constant density spanner problem. This protocol consists of three phases. A detailed description and analysis of phase I is given in Section 3, which also proves Theorem 1.3. Phases II and III are described and analyzed in Sections 4.1 and 4.2. The paper ends with possible extensions and open problems.

2. OVERVIEW OF SPANNER PROTOCOL

In the following, r_t denotes the desired transmission range and G_{r_t} represents the graph with node set V and edge set E_{r_t} containing all edges $\{v, w\}$ with $c(v,w) \leq r_t$.

Our spanner protocol for G_{r_t} consists of 3 phases:

- Phase I: The goal of this phase is to construct a constant density dominating set in G_{r_t}. This is achieved by extending Luby's algorithm [27] to our more complex model. Since the dominating set resulting from phase I may not be connected, we need further phases to obtain a constant density spanner.

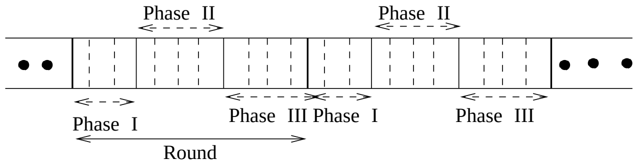

Figure 2: Two consecutive rounds of the spanner protocol.

- Phase II: The goal of this phase is to organize the nodes of the dominating set of phase I into color classes that keep nodes with the same color sufficiently far apart from each other. Only a constant number of different colors is needed for this, where the constant depends on δ. Every node organizes its rounds into time frames consisting of as many rounds as there are colors, and a node in the dominating set only becomes active in phase III in the round corresponding to its color.

- Phase III: The goal of this phase is to interconnect every pair of nodes in the dominating set that is within a hop distance of at most 3 in G_{r_t} with the help of at most 2 gateway nodes, using the coloring determined in phase II to minimize interference problems. Constructions using gateway nodes were also presented in [11, 34] but assuming a higher level model of wireless networks.

Each phase has a constant number of time slots associated with it, where each time slot represents a communication step. Phase I consists of 3 time slots, phase II consists of 4 time slots, and phase III consists of 4 time slots. These 11 time slots together form a *round* of the spanner protocol (see also Figure 2). We assume that all the nodes are synchronized in rounds, that is, every node starts a new round at the same time step. As mentioned earlier, this may be achieved via GPS or beacons.

The spanner protocol establishes a constant density spanner by running sufficiently many rounds of the three phases. All of the phases are self-stabilizing. More precisely, once phase I has self-stabilized, phase II will self-stabilize, and once phase II has self-stabilized, phase III will self-stabilize. In this way, the entire algorithm can self-stabilize from an arbitrary initial configuration.

It is not difficult to see that our spanner protocol results in a 5-spanner of constant density: Consider any pair of nodes s and t in G_{r_t} and let $p = (s = v_0, v_1, \ldots, v_k = t)$ be the shortest path from s to t in G_{r_t}. Then we can emulate p via the connected dominating set by first going to a leader ℓ_0 of s, then (possibly via gateway nodes) to a leader ℓ_1 of v_1, then to a leader ℓ_2 of v_2, and so on, until we reach a leader ℓ_k of t, and finally to t. The length of this path is at most $3k + 2 \leq 5k$ for every $k \geq 1$. Combining this with the time bounds shown for the various phases in the sections below results Theorem 1.4.

An important feature of our protocol is that all messages sent are of constant length and the nodes only have to have a constant amount of storage, irrespective of the density of the network. We just need the assumption that a storage unit is large enough to store the ID of a node. Hence, our protocol can be used with very simple devices such as sensors.

3. PHASE I: DOMINATING SET

Let P be some fixed transmission power with transmission range r_t and interference range r_i for which we want to construct a dominating set of constant density. That is, given any set of nodes V, we want to find a subset $U \subset V$ of nodes so that every node $v \in V$

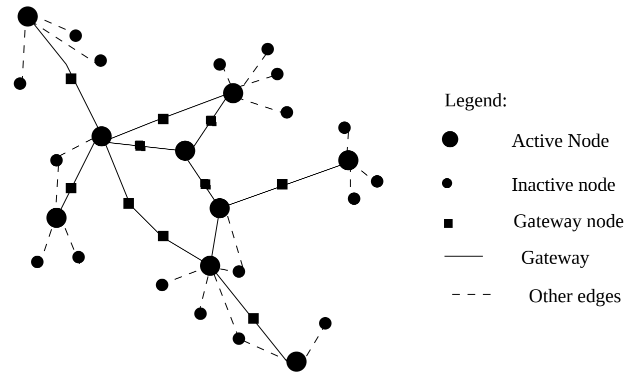

Figure 3: The spanner of the original network.

has at least one node $w \in U$ with $c(v, w) \leq r_t$ and at most some constant number of nodes $w \in U$ with $c(v, w) \leq r_t$.

As mentioned earlier, if we want to reach the goal above in a sub-linear number of steps without physical carrier sensing, then a good approximation of $\log n$ is needed, where $n = |V|$. Since our goal is to arrive at a dominating set without using any prior knowledge of the network topology, physical carrier sensing has to be used, which complicates the design as it has uncertainties (see our model). To handle these uncertainties, we use a distributed coloring strategy together with two different sensing ranges.

In our protocol, nodes can either be *active* or *inactive*. The active nodes are the candidates for the dominating set. The nodes use two different sensing thresholds, depending on their state. The sensing threshold T_a has a CSI range of r_t and the sensing threshold T_i has a CST range of r_i. To distinguish between these ranges, we speak about an aCST/aCSI-range whenever we mean T_a and iCST/iCSI-range whenever we mean T_i.

Each node cuts the time into *time frames* of k *rounds* each for some constant number k that is the same for every node. The rounds are synchronized among the nodes but we do not require the frames to be synchronized.

Initially, all nodes are inactive. Afterwards, each node executes the following protocol in each round. In this protocol, each active node has exactly one, fixed active round in a frame and a signal is just a very simple message. Each item represents a communication step.

1. If v is active and in its active round, then v sends out an ACTIVE signal.

   If v is inactive and v did not sense any ACTIVE signal for the last k rounds using a sensing threshold of T_a, v senses with threshold T_i, and if it does not sense anything, it becomes active and declares the current round number as its active round. If v did sense some ACTIVE signal in one of the last k rounds, it just performs sensing with threshold T_a and records the outcome.

2. If v is active and is in its active round, then v sends out a LEADER message containing its ID with some fixed probability p (determined later). If v decides not to send out a LEADER message but it either senses a LEADER message with threshold T_a or receives a LEADER message, v becomes inactive.

In the following, let $H_{r,k} = (V, E)$ be an undirected graph that contains an edge between two nodes v and w if and only if v and w

are active and use the same active round (or color) k and $c(v, w) \leq r$. A node v is called a *leader* if it is active and there is no other active node w of the same color with $c(v, w) \leq r_t$. Since inactive nodes sense with an iCST range of r_i before they become active, none of the inactive nodes w with $c(v, w) \leq r_i$ will become active in the active round of v. Hence, we get:

FACT 3.1. *At any time, the set of leader nodes forms an independent set in $H_{r_t,k}$ that is disconnected from all other active nodes in $H_{r_t,k}$.*

In addition, a leader node uses an aCSI range of r_t and will therefore not be affected by nodes outside of a range of r_t. Hence, we arrive at the following fact.

FACT 3.2. *Once a node becomes a leader, it will stay a leader as long as the cost function c does not change.*

Furthermore, an inactive node v can only become active if in the previous k rounds there was no active node w with $c(v, w) \leq r_s$, where r_s is the CST range for threshold T_a, because otherwise v would have sensed the ACTIVE signal of w in one of these rounds. Hence, we also get:

FACT 3.3. *There cannot be two leaders v and w with $c(v, w) \leq r_s$.*

Since r_t/r_s is a constant, the facts above and Fact 1.2 imply that the leaders must form a set of constant density in G_{r_t}. On the other hand, the following lemma is true.

LEMMA 3.4. *In any situation in which all active nodes are leaders but the leaders do not form a dominating set with respect to G_{r_t}, at least one inactive node will eventually become active.*

PROOF. From Facts 1.2 and 3.3 it follows that there can be at most some constant number k' of leaders within the iCSI range of any node. Hence, if $k > k'$ then for every inactive node that does not yet have a leader within its transmission range there must be at least one round s in which there is no leader within its iCSI range. Because the inactive node will continue to explore potential active rounds in a round-robin fashion as long as it senses a transmission with threshold T_i, it will eventually arrive at round s and become active (unless some other inactive node close to it becomes active before that). □

On the other hand, the following result is easy to check.

LEMMA 3.5. *Every connected component of active nodes in $H_{r_t,k}$ results in at least one leader.*

Thus, the algorithm eventually arrives at a situation where there is no inactive node that does not have a leader within its transmission range. At that point, the leaders must form a superset of a maximal independent set in G_{r_t}. Thus, according to Facts 1.2, 3.3, and 3.2 the leaders eventually form a static dominating set of constant density. It remains to prove how much time is needed to reach such a state.

THEOREM 3.6. *If all nodes are initially inactive, after $O(\log^4 n)$ rounds of the algorithm, the leaders form a static dominating set of constant density with respect to G_{r_t}, with high probability.*

PROOF. The next two lemmata state important properties of connected components of active nodes in $H_{r_t,k}$. Notice that a leader always represents a connected component by itself.

LEMMA 3.7. *At any time step t, $H_{r_t,k}$ consists of connected components of active nodes where all nodes in a connected component were reactivated at the same round.*

PROOF. Suppose that there are two adjacent nodes, v and w, in some active, connected component in $H_{r_t,k}$ that were not reactivated at the same round. W.l.o.g. let v be the first node that became active. Then w could not have become active because v is in its iCST range, leading to a contradiction. □

For the next lemma, given an active node v, we define $ls(v)$ as the bit sequence in which the ith bit is 1 if and only if v sent out a LEADER message in round i since it joined its current component. $ls(v)_i$ denotes the first i bits of $ls(v)$.

LEMMA 3.8. *Every connected component of active nodes in $H_{r_t,k}$ needs at most $O(\log n)$ rounds, w.h.p., until every node in it either becomes inactive or becomes a leader.*

PROOF. Consider any connected component C of active nodes in $H_{r_t,k}$ at some time point t_0, and let C' be the union of the connected components of active nodes in $H_{r_t,k}$ that have at least one node within the interference range of a node in C.

Whenever a node becomes active after t_0, it cannot interfere with the remaining nodes in C because it will be guaranteed to be outside of their interference range (and therefore also of their aCSI range). Hence, we only need to focus on the remaining active nodes in $C \cup C'$.

We prove the lemma in two steps. First, we show that it only takes $O(\log n)$ rounds, w.h.p., until there are no two active nodes v and w in $C \cup C'$ where w is within the aCST range of v or vice versa. Then we show that it only takes $O(\log n)$ further rounds, w.h.p., until there are no two active nodes v and w in C that are within the transmission range of each other.

The probability that for any two fixed, active nodes v and w it holds that $ls(v)_i = ls(w)_i$ is equal to p^i. Hence, if $i = c \log_{1/p} n$, then the probability that there are two nodes v and w in $C \cup C'$ with $ls(v)_i = ls(w)_i$ that are within their aCST range is at most $n^2/p^{c \log_{1/p} n} = n^{2-c}$. Thus, the probability that after $c \log_{1/p} n$ rounds there are still two nodes within the aCST range in $C \cup C'$ that are both active is polynomially small in n for $c > 2$.

Hence, after $O(\log n)$ rounds, there can only be at most some constant number d of active nodes within the interference range of any active node in C, where d depends on the ratio between the interference range and the aCST range. Thus, when choosing $p = 1/d$, then the probability that exactly one of the active nodes within the interference range of an active node v in C is transmitting a LEADER message in a round is $\Theta(p)$. Therefore, it takes at most $O((1/p) \log_{1/p} n) = O(d \log_d n)$ rounds until for every node v in C that is still active there is no other active node in the transmission range of v, with high probability. □

Next we give a lower bound on the number of leaders that emerge from a connected component of active nodes in $H_{r_t,k}$. For the rest of the proof, we assume w.l.o.g. that $r_t = 1$ and $r_i = 1 + \alpha$ for some constant $\alpha > 0$. We define the area covered by an active node v as the area that is within the transmission range of v.

LEMMA 3.9. *For any time step in which the currently existing connected components of active, non-leading nodes cover an area of $A = \Omega(\log^3 n)$, the number of leaders emerging from these components is $\Omega(A/\log^2 n)$, w.h.p.*

PROOF. Consider any set C of connected components of active, non-leading nodes that cover an area of A. Given any node v, let

$\Gamma(v)$ denote the set of nodes $w \in C$ with $c(v,w) \leq 1$ and let $\gamma(v) = |\Gamma(v)|$. Let H be the directed graph resulting from C by connecting two active nodes v and w by an edge (v,w) if and only if $c(v,w) \leq 1$ and $\gamma(w) \geq 2\gamma(v)$. A node is called a *sink* if it does not have any outgoing edges. H has the following important property:

CLAIM 3.10. *Every node v in H has a directed path to a sink s of length at most* $\log n$.

PROOF. First of all, H cannot contain a directed cycle. Thus, every directed path must eventually end in a sink. Suppose now that some node v has a directed path p to a sink s of length more than $\log n$. Because of the definition of the edges, it follows that $\gamma(s) \geq 2^k \cdot \gamma(v) > n \cdot \gamma(v)$, which cannot happen because there are only n nodes in the system. $\square$

Recall that our cost function must satisfy $c(v,w) \in [(1-\delta) d(v,w), (1+\delta)d(v,w)]$. Thus, if we consider disks of radius $(1+\log n)/(1-\delta)$, around the sinks of H, then the complete area A of active, non-leading nodes is covered. To extract out of all sinks a set of sinks useful for our analysis below, we consider these sinks one by one. For each sink s that has not already been eliminated, eliminate all sinks s' that are of distance at most 4 from s and add s to a set S. At the end, we arrive at a set S of sinks of pairwise distance at least 4 such that disks of radius $r = (5+\log n)/(1-\delta)$ around these sinks cover the entire area A. Thus, the area A can be decomposed into areas of size at most $a = \pi r^2$ each containing a sink in S, and therefore $|S| \geq |A|/a$. It is not difficult to show that these sinks have the following property:

CLAIM 3.11. *For any sink $s \in S$, the expected number of active nodes in $\Gamma(s)$ that become a leader is $\Theta(1)$.*

For any sink s, let the random variable X_s denote the number of active nodes in $\Gamma(s)$ that become leaders and let $X = \sum_s X_s$. From Claim 3.11 it follows that $\mathrm{E}[X] \geq \alpha|S|$ for some constant $\alpha > 0$, and because the distance between any two sinks in S is at least 4, the X_s variables are independent. Thus, we can use Chernoff bounds to obtain

$$\Pr[X \leq (1-\epsilon)\alpha|S|] \leq e^{-\epsilon^2 \alpha|S|/2}$$

for any $\epsilon > 0$. This is polynomially small if $\epsilon = 1/2$ and $|S| = \Omega(\log n)$ is sufficiently large. Hence, in this case,

$$\Pr\left[X \leq \alpha|S|/2\right] = \Pr\left[X \leq \frac{\alpha|A|}{2\pi r^2}\right]$$

is polynomially small, which completes the proof of the lemma. $\square$

Now, let us call a node *unfinished* if it is active but not a leader or it is inactive and it does not have a leader within its transmission range. We know that an unfinished node is either active or must have at least one node within its iCSI range, r_{ii}, that was active within the previous k rounds (because otherwise it would become active). Hence, when drawing disks of radius $r_{ii}/(1-\delta)$ around all nodes that were active in at least one of the k previous rounds, the entire area that the nodes can transmit messages to is covered.

Let A_0 be the area covered by the transmission ranges of all the nodes in the system. If $A_0 = \Omega(\log^3 n)$, then Lemma 3.8 and Lemma 3.9 imply that after $O(\log n)$ rounds, the area covered by the unfinished nodes is at most

$$A_0 - c \cdot \frac{A_0}{\log^2 n} = \left(1 - \frac{c}{\log^2 n}\right) A_0$$

for some constant c, with high probability. Thus, after k stages of $O(\log n)$ rounds each, the area covered by the unfinished nodes is at most

$$\left(1 - \frac{c}{\log^2 n}\right)^k A_0 \leq e^{(c \cdot k)/\log^2 n} A_0 \ ,$$

with high probability. The right hand side is less than $\log^3 n$ if $k \geq (\log A_0)(\log^2 n)/c$. Once an area of size $O(\log^3 n)$ is reached, it follows from Lemma 3.5 that it takes only $O(\log^3 n)$ more stages of $O(\log n)$ rounds each until there are no unfinished nodes any more. Since $A_0 = O(n)$, it follows that the total runtime needed for the set of active nodes to stabilize is $O(\log^4 n)$. $\square$

The dominating set algorithm can be easily extended so that it self-stabilizes [8] and it is robust against malicious behavior. Self-stabilization means that it can recover from *any* initial configuration.

3.1 Self-stabilization

An extra rule is necessary to provide self-stabilization because if the protocol above starts in a configuration violating Fact 3.3, it may not succeed in establishing a dominating set.

Consider adding a third step to each round of the protocol above. In this step, every active node sends a leader message with probability p and a transmission power so that its transmission range is only equal to the aCST range. Adding now the rule that whenever an active node receives a leader message in that step for a round different from its active round, then it becomes inactive, we do not have to assume anything about how the nodes are initially activated in order to satisfy Fact 3.3. So we get:

COROLLARY 3.12. *For any initial situation, the extended protocol needs at most $O(\log^4 n)$ rounds to arrive at a static dominating set of constant density with respect to G_{r_t}, w.h.p.*

3.2 Robustness

Our dominating set algorithm is also highly robust against adversarial nodes. For any node v, let the $r_1 \oplus r_2$-range of v be defined as the union of the r_2-ranges of all the nodes within the r_1-range of v. Given any distribution of nodes, let A be the area covered by the $r_{ii} \oplus r_t$-ranges of adversarial nodes, where r_{ii} is the iCSI range of a node. Because in our protocol adversarial nodes can directly influence only nodes within their iCSI range, nodes beyond the r_t range of these nodes can only have leaders outside of A, and leaders outside of A will stay leaders forever, one can show:

COROLLARY 3.13. *If the honest nodes outside A are connected in G_{r_t}, then after $O(\log^4 n)$ rounds, the active honest nodes outside A form a dominating set of constant density with respect to G_{r_t}, w.h.p.*

4. CONSTANT DENSITY SPANNER

In the next two subsections, we describe phases II and III in detail. We use the following notation. The constant d_1 refers to the number of active nodes that are within the interference range r_i of any node. The constant d_2 refers to the number of active nodes that are within the $r_i \oplus r_i$-range of any node, and the constant g refers to the maximum number of required gateway connections for any active node. Finally, D refers to the density of the network, i.e. the maximum number of nodes within the transmission range of a node.

4.1 Phase II - Distributed Leader Coloring

Similar to phase I, each node organizes the time into time frames consisting of cd_1 rounds for some constant c that is the same for every node. Also here, the rounds are synchronized but frames do not have to be synchronized among the nodes. We again assign active nodes to distinct rounds using a coloring mechanism. While the coloring in phase I was done with respect to G_{r_t}, we now need a coloring of the active nodes with respect to $G_{r_i \oplus r_i}$, that is, we need the active nodes to be at least $r_i \oplus r_i$ apart in order to receive the same color.

Every active node from phase I tries to own one of the rounds. An active node u is said to *own* a round if no other active node within its $r_i \oplus r_i$ range is using that round. Active nodes are in one of the states $\{\mathrm{owner},\ \mathrm{volatile}\}$. An active node is in owner state if it already owns a round and is in volatile state if it is still trying to own a round. Active nodes in owner state always send their ID in the first time slot of their round. Initially, every active node is volatile. Active nodes in volatile state choose an active round from the cd_1 possible rounds uniformly at random. Active nodes in owner state use a sensing threshold T_o with CST range r_i and active nodes in volatile state use a sensing threshold T_v with a CST range being equal to the CSI range of T_o, r_{ii}.

Active nodes do the following repeatedly. Every time a node reactivates, it sets its time stamp to 0. This time stamp is used by active nodes in Phase III to compare entries.

1. Every active node in owner state that is in its active round sends out a LEADER message containing its ID and its current time stamp and increases its time stamp by one afterwards.

2. Every active node in owner state that is in its active round decides with probability $1/2$ to send out an OWNER message either in the first or second substep of step 2.

3. Every inactive node that sensed a LEADER message with threshold T_v sends out a BUSY signal. Every active node in volatile state that senses a BUSY signal in its active round chooses a new active round uniformly at random.

4. Every inactive node that sensed OWNER messages in both substeps of step 2 with threshold T_o sends out a COLLISION signal.

   If an active node in owner state senses a COLLISION signal and sent an OWNER message in the second substep, it changes into volatile state and chooses a new active round uniformly at random.

   If an active node in volatile state did not sense a BUSY or COLLISION signal in its active round, it becomes an owner.

It is not difficult to show the following result:

THEOREM 4.1. *Once a stable set of active nodes is available, it holds: If $c \geq 4$, then all active nodes will be in owner state after $O(\log n)$ rounds of the protocol, w.h.p.*

The theorem implies that after $O(\log n)$ rounds, all active nodes have chosen rounds so that for any two active nodes ℓ and ℓ' with the same round and any inactive node v within the interference range of ℓ, ℓ' is outside of the interference range of v. Hence, ℓ can transmit messages to nodes within its transmission range without interference problems, and these nodes can transmit messages to ℓ without causing interference problems at ℓ. Both properties are important for phase III to work correctly.

Without the two types of signals BUSY and COLLISION and the two different sensing thresholds the coloring achieved may fail to be $r_i \oplus r_i$ distinct. For any active node ℓ in volatile state, the threshold T_v and the BUSY signal helps to identify the presence of active nodes in owner state with the same active round so that active nodes in owner state without another active node in owner state within the $r_{ii} \oplus r_{ii}$-range will also keep this property in the future and are therefore safe from becoming volatile again. The COLLISION signal is necessary to resolve conflicts among close by active nodes in owner state with the same active round, which can happen if volatile nodes become an owner in the same round, or this may be part of the initial state when looking at self-stabilization. In any case, the monotonicity assumption on the sensing in our model is important to make sure that there will either never be a conflict among owner nodes or immediately a conflict when a collision is detected.

4.2 Phase III - Gateway Discovery

In this section we describe the protocol for phase III. The goal of this phase is for the active nodes from Phase I to discover gateway connections to other leaders that are within a hop distance of at most 3 in G_{r_t}.

During this phase, the active nodes use an aCST range of r_t. The active nodes use the rounds reserved in phase II to achieve interference-free communication with the inactive nodes within their transmission range. Each round consists of four time slots for communication in phase III, where each time slot represents a communication step as shown in Figure 2. In the first time slot, inactive nodes send CLIENT messages and in the second time slot the active node sends a response accordingly; in the third and fourth time slots, an inactive node u may broadcast to its (active and inactive) neighbors all the information it has regarding possible gateways between the leader owning the reserved round and other leader nodes it has heard about. For simplicity, we assume that all active nodes are reactivated at the same time and hence that we can directly compare the time stamps with respect to the different active nodes. In reality, each inactive node u would keep track of the offsets of the (constant number of) time stamps it receives (in the corresponding slots allocated to the different leaders in phase II) and use these offsets when comparing time stamps from different leaders.

We first describe the data structures that are maintained during this phase. Each inactive node u maintains a cache, called $\mathcal{P}_u$, which has entries of the form (ℓ, v, t_ℓ) where ℓ is an active node, v is an inactive node (with $u = v$ possibly), and t_ℓ is the time stamp with respect to ℓ at which the entry (ℓ, v) is added to $\mathcal{P}_u$. When comparing entries in the cache, a $*$ acts as a wild card that matches any value. The operation *enqueue*(ℓ, v, t_ℓ) on $\mathcal{P}_u$ is used to add the new entry (ℓ, v, t_ℓ) to $\mathcal{P}_u$. *Enqueue* performs the following checks before actually adding the new entry to $\mathcal{P}_u$. When adding a new entry (ℓ, v, t_ℓ), any entry of the form $(\ell, *, t')$ with $t' < t_\ell$ is evicted. If no such entry exists and $\mathcal{P}_u$ is full, then the least recently added entry $(*, *, t')$, that is $t' = \min\{t | t < t_\ell \text{ and } (*, *, \mathrm{t}) \in \mathcal{P}_\mathrm{u}\}$, is evicted to make room for the new entry. The cache $\mathcal{P}_u$ has space enough to store a constant, d_2, number of entries. Inactive nodes also maintain a state that is either *awake* or *asleep* with respect to each active node that is within their transmission range. The *asleep* nodes just listen the channel and becomes *awake* when they receive a FREE or a ACK message.

Each active node ℓ maintains a list, called $\mathcal{G}_\ell$, and each entry in $\mathcal{G}_\ell$ contains two fields. The first field has gateways represented as quadruples of the form (ℓ, u, v, ℓ') where $\ell' \neq \ell$ and $u = v$ possibly, ℓ' is an active node and u, v are inactive nodes. The second field contains the time stamp t_ℓ at which the entry was added to $\mathcal{G}_\ell$. The operation *enqueue* on $\mathcal{G}_\ell$ is used to add a new entry $((\ell, u, v, \ell'), t_\ell)$ to $\mathcal{G}_\ell$. Before adding the new entry $((\ell, u, v, \ell'), t)$ to $\mathcal{G}_\ell$, any entry of the form $((\ell, *, *, \ell'), t')$ is evicted from $\mathcal{G}_\ell$ for $t' < t$. If the list $\mathcal{G}_\ell$ is full, then the entry corresponding to t' such that $t' = \min\{t'' | t'' < t \text{ and } ((\ell, *, *, \ell'), \text{t}'') \in \mathcal{G}_\ell\}$, that is the entry of $\mathcal{G}_\ell$ with smallest time stamp, is deleted to make room for the new entry. (Similar to *enqueue* on $\mathcal{P}_u$ for inactive node u). The list $\mathcal{G}_\ell$ has space enough to store a constant, g, number of entries.

In the following, ℓ refers to the ID of the active node that owns the current slot and u is an inactive node that received the ID message from ℓ and the state of u is with respect to ℓ.

1. If u is awake then u sends out a CLIENT message of the form $\langle \text{CLIENT}, \ell, u \rangle$ with probability $1/2$.

2. Node ℓ responds with a reply in the next time slot which can be of three forms. If ℓ receives a CLIENT message from node u then ℓ adds u to N_ℓ by calling *enqueue*(u) and also sends an acknowledgment containing the ID of u as $\langle \ell, \text{ACK}, u \rangle$. If ℓ only senses a busy channel but does not receive any message, then ℓ sends a collision message of the form $\langle \ell, \text{COL}-\text{LISION} \rangle$. If ℓ does not receive any message and also does not sense a busy channel, the ℓ sends a free channel message of the form $\langle \ell, \text{FREE} \rangle$.

   If u is awake and decided not to send a CLIENT message in the previous slot and receives a collision message then u goes to asleep state. If u is asleep and receives a free channel or an acknowledgement message then u becomes awake.

3. If u is awake and receives an acknowledgment containing the ID of u then u will store (ℓ, u, t_ℓ) in $\mathcal{P}_u$, where t_ℓ is the current time stamp associated with ℓ, by calling *enqueue*(ℓ, u, t_ℓ).

   Node u also deletes any entries of the form $(*, \ell)$ from $\mathcal{P}_u$ (since ℓ is no longer inactive). Node u then broadcasts, in the third time slot, a message $\langle \text{ADV}, \ell, u, t_\ell \rangle$ to its neighbors.

4. Node u builds one GATEWAY message containing all quintuples of the form $((\ell, u, v_j, \ell_j), t)$ for each j such that $\ell_j \neq \ell$ with $(\ell_j, v_j, t_j) \in \mathcal{P}_u$, where $t = \min\{t_\ell, t_j\}$, and sends the message to its neighbors. The GATEWAY message is sent in the fourth time slot.

   If v is not active and received an ADV message $\langle \text{ADV}, \ell, u, t_\ell \rangle$ then it calls *enqueue*(ℓ, u, t_ℓ) on $\mathcal{P}_v$. Node v also deletes any entries of the form $(u, *)$ or $(*, \ell)$ from $\mathcal{P}_v$ (as u is no longer an active node nor is ℓ inactive).

   If ℓ is active and receives a GATEWAY message containing $((\ell, u, v, \ell'), t)$, then ℓ stores $((\ell, u, v, \ell'), t)$ in $\mathcal{G}_\ell$ by calling *enqueue*$((\ell, u, v, \ell'), t)$.

Before we analyze the protocol, we start with the following fact, which follows from the observation that a necessary condition for an inactive node u to transmit in step 3 and step 4 is to receive an ACK from an active node in step 2.

FACT 4.2. *During steps 3 and 4 of the protocol there are at most a constant number d_1 of nodes that are transmitting any message.*

Using this fact, we can prove the following theorem.

THEOREM 4.3. *In $O(D \log n \log D)$ rounds, each active node learns about a gateway to each of the currently active nodes in its 3-neighborhood with respect to G_{r_t}, w.h.p.*

PROOF. We prove the convergence of phase III to a set of valid gateway connections in $O(D \log n \log D)$ rounds after phase I and phase II have reached a stable state. Since, at that point the active nodes have reserved rounds that are distinct within the $r_i \oplus r_i$ range, we can treat the actions of active nodes independent of each other.

Let (v, ℓ) be an inactive node-active node pair such that v has to send a CLIENT message to ℓ. Node v has at most $O(D)$ inactive nodes in its interference range sending a CLIENT message to some leader node. If more than one node in awake state, with respect to ℓ, decides to send a CLIENT message, then ℓ will send a collision message. Since the collision message will be received by the inactive nodes, within r_t range of ℓ, awake nodes that decided not to send a CLIENT message to ℓ in the previous slot will go to asleep state.

Consider time to be partitioned into groups of consecutive rounds such that each group ends with a round where the active node ℓ sends either an ACK message or a FREE message. (A group ending with an ACK message signifies a successful group and a group ending with a FREE message is a failed group). Notice that at the end of every group, whether successful or not, all the inactive nodes within the r_t range of ℓ go to awake state (by step 2 of the protocol).

It is not difficult to show that the expected number of rounds in each group, successful or failed, is $O(\log D)$ and any group is successful with constant probability. Due to symmetry reasons any inactive node is equally likely to be send a CLIENT message in a successful group. Thus, during any successful group, for a given pair (v, ℓ) ,

$$\Pr[\, v \text{ sends a CLIENT message successfully to } \ell] \geq 1/cD$$

for some constant $c > 1$.

Using Chernoff bounds, for any given pair (v, ℓ) the probability that it takes more than Dk groups so that v sends a CLIENT message to ℓ successfully will be polynomially small for $k = O(\log n)$. It can also be shown that each group has $O(\log D)$ rounds not only on expectation but also with high probability. Thus any node v requires at most $O(D \log n \log D)$ rounds to send a CLIENT message to ℓ successfully w.h.p.

To proceed further, let ℓ and ℓ' be active nodes, with $d(\ell, \ell') \leq 3$ and let (ℓ, u, v, ℓ') be a gateway between ℓ and ℓ'. Notice that once ℓ and ℓ' receive CLIENT messages from u and v respectively, ℓ and ℓ' can establish a gateway connection between them as successful CLIENT messages are followed by ADV and GATEWAY messages in the next time slots reserved for this phase. Without loss of generality, we assume that u sends the ADV message that v receives and adds the entry (ℓ, u, v, ℓ') to the GATEWAY message that v sends. Along with Fact 4.2 it holds that during every group the probability that u gets an ACK message and sends the ADV message is $\geq 1/c'D$ for a constant $c' > 1$. And similarly the probability that v gets an ACK message from ℓ' and sends a GATEWAY message is $\geq 1/c'D$. Thus, in each group,

$$\Pr[\ell \text{ and } \ell' \text{ discover a gateway connection}] \geq 1/c''D$$

for some constant $c'' > 1$. Using calculations similar to the above, it holds that ℓ and ℓ' can establish a gateway connection in $O(D \log n \log D)$ rounds w.h.p.

Note that, after phase II stabilizes and after we let phase III run for $O(D \log n \log D)$ time steps, time stamping will be enough to guarantee that we will always keep information received at a leader node ℓ about a valid gateway connection between leader nodes ℓ and ℓ', if at least one such connection exists (since we have at most a constant number of leader nodes within cost $3r_t$ from any given leader node, and since we have at most a constant number of leader nodes adjacent to any inactive node, constant size $\mathcal{P}_u$ and $\mathcal{G}_\ell$ lists at inactive nodes u and active nodes ℓ respectively will suffice). $\square$

5. FUTURE WORK

We feel that our model provides a realistic model for wireless communication and it would therefore be highly interesting to see how algorithms in our model perform in practice. Also, it would be very interesting to develop protocols for other problems on top of our wireless model (e.g. broadcasting and service discovery), in particular, protocols that can self-stabilize under adversarial influence.

6. REFERENCES

[1] M. Adler and C. Scheideler. Efficient communication strategies for ad hoc wireless networks. *Theory of Computing Systems*, 33:337–391, 2000.

[2] K. Alzoubi, X.-Y. Li, Y. Wang, P.-J. Wan and O. Frieder Geometric spanners for wireless ad hoc networks. In *IEEE Transactions on Parallel and Distributed Systems*, pages 408-421, 2003.

[3] K. Alzoubi P.-J. Wan and O. Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. In *IEEE Infocom '02*, 2002.

[4] K. Alzoubi, P.-J. Wan, and O. Frieder. New distributed algorithm for connected dominating set in wireless ad hoc networks. In *Proc. of the 34th Annual Hawaii International Conference on System Science (HICSS-35)*, 2002.

[5] X. Cheng, D. Huang, W. Li, W. Wu, and D. Z. Du. A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks. *Networks: An International Journal*, 42, 2003.

[6] S. Choi. IEEE 802.11e MAC-level FEC performance evaluation and enhancement. In *GLOBECOM '02*, 2002.

[7] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86:165–177, 1990.

[8] E. W. Dijkstra. Self stabilization in spite of distributed control. *Communications of the ACM*, 17:643–644, 1974.

[9] D. Dubhashi, A. Mei, A. Panconesi, J. Radhakrishnan, and A. Srinivasan. Fast distributed algorithms for (weakly) connected dominating sets and linear-size skeletons. In *Proc. of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-03)*, pages 717–724, 2003.

[10] K.R. Gabriel and R.R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18:259–278, 1969.

[11] J. Gao, L.J. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Geometric spanner for routing in mobile networks. In *Proc. of the 2nd ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '01)*, pages 45–55, 2001.

[12] P. Gupta and P.R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, IT-46(2):388–404, 2000.

[13] P. Gupta and P.R. Kumar. Internets in the sky: The capacity of three dimensional wireless networks. *Communications in Information and Systems*, 1(1):33–50, 2001.

[14] P. Gupta and P.R. Kumar. Towards an information theory of large networks: An achievable rate region. In *IEEE International Symposium on Information Theory (ISIT 2001)*, 2001.

[15] H. Huang, A. W. Richa, and M. Segal. Approximation algorithms for the mobile piercing set problem with applications to clustering in ad-hoc networks. *ACM Baltzer Journal on Mobile Networks and Applications (MONET)*, pages 141–149, April 2004.

[16] L. Jia, R. Rajaraman, and R. Suel. An efficient distributed algorithm for constructing small dominating sets. In *Proc. of the 20th ACM Symposium on Principles of Distributed Computing (PODC)*, 2001.

[17] T. Jurdzinski and G. Stachowiak. Probabilistic algorithms for the wakeup problem in single-hop radio networks. In *Proc. of the 13th Int. Symposium on Algorithms and Computation*, pages 535–549, 2002.

[18] J. Keil and W. Gutwin. The delaunay triangulation closely approximates the complete euclidean graph. In *1st Workshop on Algorithms and Data Structures (WADS)*, 1989.

[19] F. Kuhn, T. Moscibroda, and R. Wattenhofer. Radio network clustering from scratch. In *European Symposium on Algorithms (ESA)*, 2004.

[20] Fabian Kuhn and Roger Wattenhofer. Constant-time distributed dominating set approximation. In *Proc. of the 22nd ACM Symposium on Principles of Distributed Computing (PODC)*, pages 25–32, 2003.

[21] F. Kuhn, R. Wattenhofer, Y. Zhang and A. Zollinger. Geometric ad-hoc Routing: Of theory and practice. In *Proc. of the 22nd ACM Symposium on Principles of Distributed Computing (PODC)*, 2003.

[22] Fabian Kuhn and Roger Wattenhofer and Aaron Zollinger Ad-hoc networks beyond unit disk graphs. In *Proc. of the 2003 Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*, pages 69–78, 2003.

[23] J. Kuruvila, A. Nayak, and I. Stojmenovic. Hop count optimal position based packet routing algorithms for ad hoc wireless networks with a realistic physical layer. In *1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, 2004.

[24] X.-Y. Li, G. Calinescu, and P.-J. Wan. Distributed construction of planar spanner and routing for ad hoc wireless networks. In *IEEE INFOCOM*, pages 1268–1277, 2002.

[25] X.-Y. Li, P.-J. Wan, and Y. Wang. Power efficient and sparse spanner or wireless ad hoc networks. In *IEEE International Conference on Computer Communications and Networks (ICCCN)*, pages 564–567, 2001.

[26] X.-Y. Li, P.-J. Wan, Y. Wang, and O. Frieder. Sparse power efficient topology for wireless netowrks. In *IEEE Hawaii International Conference on System Sciences (HICSS)*, 2002.

[27] M. Luby A simple parallel algorithm for the maximal independent set problem. In *Proc. of the 17th ACM Symposium on Theory of Computing (STOC)*, pages 1–10, 1985.

[28] IEEE P802.11a/D7.0-1999. Wireless lan medium access control (mac) and physical layer (phy) specification: High speed physical layer in the 6 ghz band. IEEE, New York, July 1999.

[29] S. Parthasarathy and R. Gandhi. Distributed algorithms for coloring and domination in wireless ad hoc networks. In *Proc. of FSTTCS*, 2004.

[30] L. Quin and T. Kunz. On-demand routing in MANETs: The impact of a realistic physical layer model. In *Proc. of the 2nd International Conference ADHOC-NOW*, pages 37–48, 2003.

[31] T.S. Rappaport. *Wireless Communications: Principle and Practice*. Prentice Hall, 1996.

[32] W-Z. Song, Y. Wang, ,X-Y. Li and O. Frieder Localized algorithms for energy efficient topology in wireless ad hoc networks. In *Proc. of the 5th ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 98–108, 2004.

[33] Y. Wang and X.-Y. Li. Localized construction of bounded degree and planar spanner for wireless ad hoc networks. In *Proc. of the 2003 Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*, pages 59–68, 2003.

[34] Y. Wang and X.-Y. Li. Geometric Spanners for Wireless Ad Hoc Networks. In *IEEE Trans. on Parallel and Distributed Systems*, pages 408–421, vol. 14, 2003.

[35] R. Wattenhofer, J. Li Li, P. Bahl, and Y.-M. Wang. Distributed topology control for wireless multihop ad-hoc networks. In *IEEE INFOCOM '01*, pages 1388–1397, 2001.

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