Infrastructure-Establishment from Scratch in Wireless Sensor Networks

Stefan Funke and Nikola Milosavljevic*

Computer Science Department
Stanford University
Stanford, CA 94305, U.S.A.
{sfunke,nikolam}@stanford.edu

Abstract. We present a distributed, localized and integrated approach for establishing both low-level (i.e. exploration of 1-hop neighbors, interference avoidance) and high-level (a subgraph of the unit-disk graph) infrastructure in wireless sensor networks. More concretely, our proposed scheme constructs a subgraph of the unit-disk graph which is connected, planar and has power stretch factor of 1 (the well-known Gabriel graph intersected with the unit disk-graph) and – most importantly – deals *explicitly* with the problem of interference between nearby stations. Due to our interleaved approach of constructing low- and high-level infrastructure simultaneously, this results in considerable improvements in running time when applied in dense wireless networks.
To substantiate the advantages of our approach, we introduce a novel distribution model inspired by actual sensing applications and analyze our new approach in that framework.

1 Introduction

Different from wired networks, wireless networks typically consist of a set of nodes that initially have no information about how to communicate with each other. Hence before running any application-specific algorithms and procedures on such a network, a basic communication infrastructure must be established. Low-level communication infrastructure essentially comprises learning about which other stations are within direct communication range and providing for interference-free communication between such neighboring stations. The high-level communication infrastructure is typically built by selecting a subset of the links identified during the low-level infrastructure establishment such that only those links will be used for communication within the network.

Assuming disk radii correspond to transmission ranges of individual stations, the information about which stations can receive messages from which other stations is captured in the so-called disk graph *DG*, or unit-disk graph *UDG* in case of uniform transmission ranges (the former a directed, the latter a bidirectional or undirected graph). Common high-level structures on top of the *UDG*/*DG* are for example the Gabriel graph

* both authors were supported by the Max Planck Center for Visual Computing and Communication (MPC-VCC) funded by the Federal Ministry of Education and Research of the Federal Republic of Germany (FKZ 01IMC01).

$GG(UDG)$, the relative neighborhood graph $RNG(UDG)$, the Yao graph $Y(UDG)$, or the (localized) Delaunay graph $LD(UDG)$. Regarding interference-free communication there are basically two approaches. First, there is the possibility to resolve interference on the MAC-layer on a per-communication basis. Secondly, there is the concept of a so-called D2-coloring of the nodes of the disk graph. Here one color corresponds to one time slot/frequency and makes sure that no nodes at hop-distance less than or equal to 2 in the UDG transmit at the same time; hence no interference – not even at hidden terminals – can occur.

Typically low-and high-level infrastructure are constructed *sequentially*, in a sense that first the low-level infrastructure is built, and then the algorithms constructing the high-level infrastructure build upon the existing low-level infrastructure. As we will exhibit in Section 2, this inherently induces some inefficiencies, especially in scenarios where inter-station distance is small compared to their transmission range.

Related Work

There is a plethora of proposed algorithms for identifying a high-level infrastructure from a given unit-disk graph, such as [2], [9], [10], [14]. Common to all of these approaches is that they rely on a pre-existing low-level infrastructure, in particular they assume knowledge of all $1-$hop neighbors in the UDG and interference-free communication between adjacent stations in the UDG. [5] is an exception as it explicitly deals with the problem of interference and solves it by constructing a D2-coloring of all nodes in the UDG. Still, their algorithm assumes knowledge of all 1-hop neighbors in the UDG.

Most related to our work are the very recent papers by Kuhn et al. [6],[8], where the authors develop very interesting protocols for structuring a set of newly deployed wireless nodes without any assumption about a preexisting infrastructure. In [1] and [7] high-level infrastructures are examined with respect to their interference-inducing properties. Both papers introduce formal definitions of interference and propose new algorithms for which the resulting infrastructures exhibit low interference.

Our Results

In this paper we present an integrated approach for establishing both low- and high-level infrastructure for a set of deployed wireless stations. Our distributed protocol assumes no knowledge about 1-hop neighbors or interference-free communication between neighboring stations. It constructs the Gabriel graph $GG(UDG)$[1] consisting of all Gabriel edges of UDG, respective transmission ranges for each node of the network such that all nodes connected by an edge in $GG(UDG)$ can communicate with each other, as well as a valid coloring of the nodes such that interference-free communication between nodes in $GG(UDG)$ is ensured. Choosing $GG(UDG)$ as high-level structure guarantees that multi-hop communication between any two stations can be performed using the minimal amount of energy.

[1] In fact we chose the Gabriel graph as an example; other high-level structures should be constructible in a similar fashion.

Our scheme requires network nodes to have variable transmission power upper bounded by 1, to be able to detect if one or more neighboring stations, including itself, are sending a signal (the message is actually received only if exactly one station is transmitting), and to have access to a global clock and positioning device like GPS.

Furthermore, upon deployment each node is provided with a value ε, which can be set as some lower bound for the minimum inter-node distance (e.g. as derived from the physical size of the nodes), and a parameter $\widetilde{\Delta}$, which denotes the maximum number of stations nearby a node, where 'nearby' is defined for each node relative to the transmission radius assigned after the construction of the high-level infrastructure, and *not* relative to the maximum transmission radius.

The running time of our distributed protocol is $O(\widetilde{\Delta}^{3/2} \log \frac{1}{\varepsilon} \log^2 n)$ and all the above properties are fulfilled with high probability of $1 - \frac{1}{n^\phi}$ for any desired value ϕ.

To substantiate the usefulness of our protocol we introduce a novel distribution model for node deployment which is inspired by actual sensing applications. We show that in this model the value $\widetilde{\Delta}$ is a constant, hence making our algorithm extremely fast in such scenarios. We emphasize though, that by choosing $\widetilde{\Delta} = n$, our algorithm works for any node distribution.

In Section 2 we recap basic procedures of low- and high-level infrastructure establishment and give an intuitive derivation of the parameter $\widetilde{\Delta}$. Section 3 describes our algorithm in detail and analyzes its performance. In Section 4 we present our new distribution model and prove that in this model the parameter $\widetilde{\Delta}$ is a constant. Finally we conclude with some remarks and open problems.

2 Disk Intersection Graphs, Interference, and Power Efficiency

In this section we will introduce the basic concepts necessary for understanding our protocol, presented in the next section. Throughout we assume the standard communication model based on the *disk graphs* (*DG*), where the transmission ranges are represented by disks (of possibly different radii), centered at node locations. A disk graph corresponding to equal communication radii for all nodes is called a *unit-disk graph* (*UDG*). We remark, as this will be important later, that *DG*s are in general directed graphs, while *UDG*s are bidirectional (equivalently, undirected).

2.1 Interference

We assume that all station use the same frequency, so due to interference, not all communication suggested by the unit-disk graph is guaranteed to succeed. Suppose two stations transmit at the same time. If they are within each other's communication range, they both experience *direct interference*. Also, any station in the intersection of their communication ranges experiences *indirect interference*. The latter is also called *hidden terminal* problem.

The problem of interference can be dealt with on per-communication basis in the MAC-layer, typically using a handshake mechanism. Clearly, if n nodes are distributed densely enough so that the communication graph is complete, it might take $\Omega(n)$ time for a handshake to succeed. Thus in the following we focus on a different approach.

Resolution by D2-coloring. A natural way to avoid collisions or interference in case of uniform transmission ranges is to make sure that stations within 2-hop distance in UDG never transmit at the same time. Abstractly this corresponds to a coloring of the vertices of UDG such that two vertices with graph distance less or equal to 2 always have different colors.

Computing a D2-coloring with the minimum number of colors is NP-hard, but several approximation algorithms are known, both centralized and distributed. See [11], [12], [13]. The distributed algorithm in [5] computes with high probability (w.h.p.) a valid D2-coloring of a given UDG, in time $O(\delta_2 \log^2 n)$, where δ_2 is an upper bound on the size of any 2-hop neighborhood [2]. A variant of this algorithm will be used in our approach as a subroutine.

The concept of D2-coloring also extends to directed disk graphs DG derived from stations with non-uniform transmission radii. A necessary and sufficient condition for interference-free communication between nodes in DG is the following: for any node u, and incoming edges $(v_1,u),(v_2,u),\dots$, each pair of nodes v_i and v_j, $i \neq j$ must be assigned different colors/time slots (also different from u). Otherwise they might transmit at the same time to u, resulting in a collision. Hence, if we denote by δ the maximum indegree of a node in DG, $\delta+1$ is a *lower bound* on the number of colors/time slots required for interference-free communication. Still, to our knowledge there is no result stating that $O(\delta)$ colors are enough to color this *directed* graph such that the above condition is satisfied.

2.2 Power Efficiency — the Gabriel Graph [4]

Wireless sensor networks typically employ indirect, multi-hop communication, so having a good routing algorithm is crucial. Among other properties, *power efficiency* is often required: total energy needed to transmit a message along a route produced by the algorithm has to be close to the minimum over all possible routes for given source and destination. If we assume that required energy grows at least quadratically with distance to be spanned, and that total energy of a path is the sum of the energies of the individual links traversed, then a simple argument shows that power-optimal paths within UDG only contain *Gabriel edges*, i.e. edges whose diametral balls are empty of other stations. Hence restricting to the Gabriel edges in UDG (we will call this graph $GG(UDG)$) ensures that power-optimal routing is still possible. Schemes for computing the Gabriel graph and variations of it have been proposed in the literature, but all of them assume knowledge of the 1-hop neighborhood as well as a solution to the interference problem.

2.3 A Typical Scenario in Sensor Networks

Suppose that many wireless sensors are dispersed over an area for the purpose of monitoring physical quantities such as temperature, humidity, exposure to light etc. The sensors have to self-organize into a network, in order to efficiently answer queries injected from the outside world. One way to achieve such an organization is as follows:

[2] In fact, they state the algorithm in terms of the maximum degree (maximum number δ_1 of 1-hop neighbors). But in the proof they derive δ_2, which is $O(\delta_1)$ for unit-disk graphs.

1. establish a *low-level infrastructure*, i.e.
   - Let every node discover its 1-hop neighborhood in the communication graph w.r.t. its maximal transmission radius.
   - Compute a D2-coloring of this communication graph to provide for conflict-free communication.
2. establish the *high-level infrastructure* and adjust the low-level infrastructure
   - Construct a suitable routing substructure, e.g. the Gabriel graph $GG(UDG)$, using the existing low-level infrastructure.
   - Adjust low-level infrastructure to the new adjacency relationships by decreasing the transmission radii where possible, and and reducing the number of colors in the D2-coloring.

See Figure 1 for a graphical illustration of the process. Observe that in the initial low-level infrastructure as many as n colors are necessary, since the nodes are densely placed. Next, the Gabriel edges are identified, and each node adjusts its transmission range to just be able to reach its furthest neighbor in $GG(UDG)$. The last phase is D2-coloring of the resulting disk graph DG, corresponding to the reduced radii.

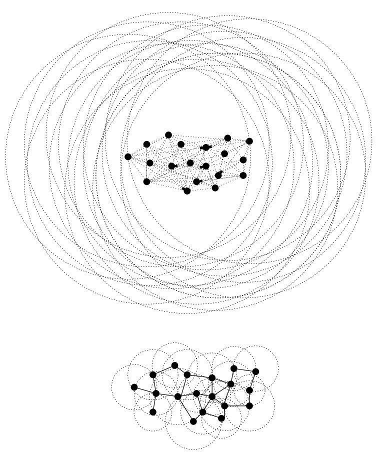

Fig. 1. The communication graph for a dense node distribution (top) and the adjusted transmission radii after computing a high-level infrastructure (bottom).

We expect the number of required colors/time slots to drop considerably due to the decreased transmission radii. In particular, for dense node distributions, after construction of the high-level infrastructure, nodes communicate only with stations at a very short distance. The amount of interference that the coloring has to cope with is drastically reduced. Still, to construct the high-level infrastructure, most known algorithms require collision-free communication between stations within distance 1, and hence a coloring w.r.t. the maximum transmission ranges has to be computed before, even though it will not be needed in the final version of the communication graph.

The goal of this paper is to avoid this potentially wasteful procedure and use time slots only up to an amount as roughly required by the final assignment of transmission powers anyway.

2.4 The Local Vicinity Size Δ_k

Before we can present our remedy for the problem sketched above, we need to have a closer look at some properties of the resulting communication graph DG after adjustment of the transmission powers. In particular, we are interested in a parameter that estimates how many colors are necessary in a D2-coloring of the final structure.

As mentioned above, if δ is the maximum indegree of a node in a DG, then $\delta + 1$ is a lower bound on the number of colors/time slots required for interference-free communication. For the analysis of our algorithm we need to get a handle on a simple and *geometric* parameter that relates the node distribution and this quantity δ. Hence let us introduce a parameter Δ_k, which provides us with an upper bound on δ (and later it will also be an upper bound on the number colors needed for a valid coloring).

Definition 1. *For a node v, let $\Delta_k(v)$ be the number of nodes contained in a disk of radius $k \cdot |vu|$, where (v,u) is the longest edge in $GG(UDG)$ adjacent to v. The quantity $\Delta_k = \max_v \Delta_k(v)$ is called the **local vicinity size**.*

For not too contrived node distributions we expect $\Delta_k = O(\delta)$, but again, showing a tight bound seems hard. See Figure 2 for an example of the local vicinity size. It is now easy to see that for large enough k, Δ_k yields an upper bound on δ.

Lemma 1. *For $k \geq 2$ we have $\delta + 1 \leq \Delta_k$.*

Proof. Consider the node v which maximizes the indegree in DG. Let (u,v) be the longest incoming edge of v. Clearly u has a Gabriel edge adjacent with length at least $|uv|$. Considering the ball B of radius $2|uv|$ centered at u, B must contain all w with $(w,v) \in DG$. Hence $\Delta_2 \geq \delta + 1$.

Our algorithm in the next section will use the local node density $\widetilde{\Delta} = \Delta_k$ for some constant $k \geq 2$ as an estimate for the number of time slots necessary to achieve interference-free communication. The intuition behind this parameter is that the number of colors required should be somewhat proportional to the maximum number of nearby stations of some node. Here 'nearby' has to be seen relative to the longest Gabriel edge adjacent to that node. In non-contrived node distributions we expect $\widetilde{\Delta}$ to be quite small or even constant, but certainly smaller than the maximum degree of the original unit-disk graph corresponding to the assignment of maximum transmission ranges.

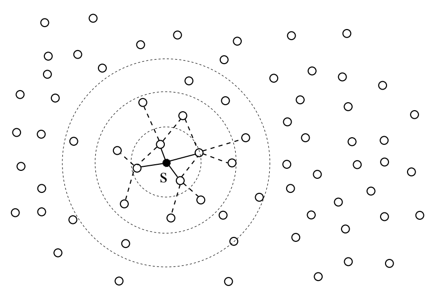

Fig. 2. Local vicinity sizes $\Delta_1(s) = 5, \Delta_2(s) = 12, \Delta_3(s) = 20$ (only Gabriel edges of S and its 1-hop neighbors are drawn).

3 An Integrated Approach to Infrastructure Establishment

As we have seen, in case of very dense sensor distributions, sequential infrastructure establishment introduces an inherent overhead. In the following we will remedy this problem by providing an integrated approach for constructing both low- and high-level infrastructure.

The basic idea of our approach is very straightforward: locally, at every node, we start exploring its neighborhood gradually by increasing its transmission range. For each transmission radius, one then constructs a local D2-coloring using a variation of the algorithm in [5]. As we will then see, one can guarantee that w.h.p. this approach actually finds exactly the edges of $GG(UDG)$.

Provided with two parameters, ε and $\widetilde{\Delta}$, every node s locally executes the protocol in Figure 3. Here ε is a parameter essentially capturing the minimum distance between any two nodes, that can be for example derived just from the physical dimensions of the wireless nodes. So each node executes for $\log \frac{1}{\varepsilon}$ rounds the exploration and announcement protocol to learn about its neighborhood and construct the high-level substructure $GG(UDG)$. At the end, a valid D2-coloring of the final communication graph is computed. Let us now describe the subroutines in more detail.

- $r \leftarrow \varepsilon$
- while ($r < 1$)
  1. **ExploreDirectNeighborhood**$(s, r, \widetilde{\Delta})$
  2. **LocalD2Color**$(s, r, \widetilde{\Delta})$
  3. **AnnounceNApproveEdges**(s)
  4. $r \leftarrow r \cdot 2$
- **FinalD2Color**$(s, |l_s|, \widetilde{\Delta})$

Fig. 3. Infrastructure-establishment protocol

3.1 Exploration of the r-neighborhood – ExploreDirectNeighborhood$(s, r, \widetilde{\Delta})$

This routine makes sure that a node s w.h.p. announces its presence to all nearby nodes that need to know about it. More precisely, we will show that for $r \leq \varepsilon \cdot 2^{\lceil \log_2(l_s/\varepsilon) \rceil}$, where l_s denotes the length of the longest edge adjacent to s in $GG(UDG)$, w.h.p. all nodes 'nearby' s in the unit-disk graph of disks of radius r get to know about their 1-hop neighborhood. **Note:** We do not guarantee this property for larger values of r, since it turns out that this information is not necessary for constructing all edges of $GG(UDG)$.

This exploration/announcement procedure consists of $\kappa \cdot \log n$ rounds, each of which lasts for $2\widetilde{\Delta}$ time steps. In each round a node chooses a random number $1 \leq t \leq 2\widetilde{\Delta}$ and transmits its ID together with its position at time step t in this round within radius r. Let us first convince ourselves that all neighbors of s get to know about s's presence w.h.p.

Lemma 2. *After $2\kappa\widetilde{\Delta}\log n$ time steps, with high probability of $1 - \frac{1}{n^{\kappa-1}}$, all neighboring nodes (w.r.t. transmission radius r) of a node s with $r \leq \varepsilon \cdot 2^{\lceil \log_2(l_s/\varepsilon) \rceil}$ know about s's presence if $\widetilde{\Delta} = \Delta_k$ for $k \geq 4$.*

Proof. For a given node, the probability that its announcement was overheard by its neighbors is at most $1/2$, since there are at most $\widetilde{\Delta}$ other nodes which could interfere with its announcement (since we have $r \leq 2l_s$ and Δ_k for $k \geq 4$ counts the number of all 2-hop neighbors of s w.r.t. transmission radius r). The probability that in none of the $\kappa \cdot \log n$ rounds its announcement was successful is bounded by $(\frac{1}{2})^{\kappa \cdot \log n} = \frac{1}{n^{\kappa}}$. Hence the probability that all nodes with $r \leq \varepsilon \cdot 2^{\lceil \log_2(l_s/\varepsilon) \rceil}$ have successfully announced their presence to their neighbors is at least $1 - \frac{1}{n^{\kappa-1}}$.

In the same setting, we can also show that all h-hop neighbors of s get to know about their 1-hop neighbors w.h.p. This will be needed later in the proofs.

Lemma 3. *After $2\kappa\widetilde{\Delta}\log n$ time steps, with high probability of $1 - \frac{1}{n^{\kappa-2}}$, all nodes within h-hop distance from some station s get to know about their neighborhood (w.r.t. transmission radius r) if $r \leq \varepsilon \cdot 2^{\lceil \log_2(l_s/\varepsilon) \rceil}$ and $\widetilde{\Delta} = \Delta_k$ for $k \geq 2(h+1)$.*

Proof. Consider some node t within h-hop distance of s. For t to get to know about the set N_1 of its 1-hop neighbors, each of the nodes in N_1 should in some round have announced its presence without interfering with another node in N_1. Hence if we have $\widetilde{\Delta} \geq |N_1|$, the probability for one announcement to get through is $\geq 1/2$ in each round. Using the same argumentation as in Lemma 2, it follows that the probability of each node getting at least one announcement through in $\kappa \log n$ rounds is at least $1 - \frac{1}{|n|^{\kappa-1}}$. Furthermore, *all* nodes within the h-hop neighborhood then get to know about their neighbors with probability $\geq 1 - \frac{1}{|n|^{\kappa-2}}$. We now need to relate that to $|l_s|$, i.e. the longest adjacent edge of s in $GG(UDG)$. This can be easily achieved by requiring that $\widetilde{\Delta}$ bounds the number of nodes contained within distance $(h+1) \cdot r$. Hence the Lemma holds for $\widetilde{\Delta} = \Delta_k$ for $k \geq 2(h+1)$.

Correctness of our whole algorithm will later be established by focusing on one particular node s and show that when $r = \varepsilon \cdot 2^{\lceil \log_2(l_s/\varepsilon) \rceil}$, all edges adjacent to s in $GG(UDG)$ are found w.h.p. and that wrong edges are never created. Towards this goal, Lemma 3 states that for the appropriate choice of $\widetilde{\Delta}$, **ExploreDirectNeighborhood**$(s, r, \widetilde{\Delta})$ with high probability ensures that all h-hop neighbors of s get to know about their immediate neighbors in the unit-disk graph induced by transmission radius r.

3.2 Local D2-coloring – LocalD2Color$(s, r, \widetilde{\Delta})$

This subroutine ensures that for a node s and transmission radius r with $r \leq \varepsilon \cdot 2^{\lceil \log_2(l_s/\varepsilon) \rceil}$ all 1- and 2-hop neighbors and s have distinct colors with high probability. Again, we do not guarantee anything for larger values of r, since this is not necessary for constructing all edges of $GG(UDG)$. Our procedure is a slight variation of the one proposed in [5]. Next, we sketch the latter and briefly discuss the changes needed for our algorithm.

The algorithm by Parthasarathy/Gandhi for D2-coloring. Parathasarathy/Gandhi in [5] have presented an algorithm for D2-coloring of a unit-disk graph. Their algorithm assumes that each node knows its 1-hop neighbors and a bound on the number of 2-hop neighbors.

Each node maintains a list of c potential colors it can choose from. The algorithm proceeds in rounds, in each of which typically some so-far-uncolored nodes choose a color from their list, and if this color has not been chosen by any 1- or 2-hop neighbors, these color assignments become permanent, and all 1- and 2-hop neighbors remove the respective colors from their color lists. The algorithm terminates after t rounds after which with high probability all nodes have been assigned colors which are a valid D2-coloring provided a suitable choice of the parameters c and t.

One round consists of the following 4 phases: TRIAL, TRIAL-REPORT, SUCCESS, and SUCCESS-REPORT.

The **TRIAL** phase consists of c time slots. An uncolored node u decides with probability $1/2$ to wake up, choose a random color from its list, and transmit a TRIAL message $\{ID(u), \text{color}(u)\}$ at the time slot corresponding to the chosen color.

The **TRIAL-REPORT** phase consists of b blocks of c time slots each. At the beginning of this phase each node composes a TRIAL-REPORT containing all TRIAL messages that node received in the previous phase. Then in every block it chooses a random time slot among the c slots available in a block, and sends its TRIAL-REPORT.

The **SUCCESS** phase consists of c time slots. At the beginning of this phase, every awake node u determines if the color it had chosen in the first phase is safe, i.e. if TRIAL-REPORTs have been received from all neighbors, each contains u, and none contains the same color chosen by another node. If all these conditions are met, u sends a SUCCESS message $\{ID(u), \text{color}(u)\}$ in the time slot corresponding to color(u). u will not participate in future TRIAL and SUCCESS phases.

The **SUCCESS-REPORT** phase is similar to the TRIAL-REPORT phase. Nodes prepare SUCCESS-REPORTs and send them in random time slots over r rounds. But additionally, at the end of this phase, all uncolored nodes remove the colors used in SUCCESS-REPORTs that they have received.

Parthasarathy and Gandhi prove that for $c = O(\max \# \text{ D2-neighbors})^3$, $t = O(\log n)$, and $b = O(\log n)$, their algorithm computes a valid D2-coloring of the unit-disk graph with high probability $\geq 1 - \frac{1}{n^\chi}$ for arbitrary $\chi > 0$ (the constant factors for c, t, b depend on the desired success probability). Clearly, the overall running time is $O(c \log^2 n)$. They establish correctness of their algorithm by proving the following key facts (we refer to [5] for a detailed exposition of the proofs):

1. the probability that after completion of the procedure a node is still uncolored can be bounded by $\frac{1}{n^\psi}$ for arbitrary $\psi \geq 1$
2. the probability that any two nodes u, v at 1- or 2-hop distance end up with the same color can be bounded by $\frac{1}{n^\psi}$ for arbitrary $\psi \geq 1$

Can we apply their algorithm directly in our setting, where for a fixed transmission range r we want to compute a valid D2-coloring of the induced disk graph? — Unfortunately not! In fact, in this graph we do not have a bound on the number of 1- or 2-hop neighbors that is valid for *all* nodes. For a choice of $\widetilde{\Delta} = \Delta_k$ with $k \geq 4$ we can only bound the number of D2-neighbors of nodes s with $r \leq \varepsilon \cdot 2^{\lceil \log_2(l_s/\varepsilon) \rceil}$.

What we actually want is that each node s with $r \leq \varepsilon \cdot 2^{\lceil \log_2(l_s/\varepsilon) \rceil}$ w.h.p. gets colored differently from its arbitrary 1- or 2-hop neighbor t. The details of the proofs in [5] reveal that the probability of t not being colored at all depends on the number of 2-hop neighbors of t, and *not* of s (essentially, if t has too many 2-hop neighbors, chances are that each trial of t is doomed as there is some 2-hop neighbor of t which has chosen the same color). Similarly, if both s and t have been colored, they only might have the same color if a SUCCESS-REPORT of their common neighbor was lost due to interference. This probability can be kept low if the number of D2-neighbors of this common neighbor is bounded.

But D2-neighborhoods of all these nodes can be easily bounded by requiring $\widetilde{\Delta} = \Delta_k$ with $k \geq 8$, that is by bounding the number of points in a larger disk around s which also contains all 2-hop neighbors of 2-hop neighbors of s for the current choice r.

So we can prove the two key facts under this new condition. We emphasize again, though, that our Lemmas hold only for nodes s with $r \leq \varepsilon \cdot 2^{\lceil \log_2(l_s/\varepsilon) \rceil}$ and not for larger values of r.

[3] In fact they show $c = O(\max \text{1-hop-neighbors})$, but in case of unit-disk graphs, this is the same quantity up to a constant factor.

Lemma 4. *Let s be a node with $r \leq \varepsilon \cdot 2^{\lceil \log_2(l_s/\varepsilon) \rceil}$. Then with high probability of $1 - \frac{1}{n^\Psi}$, s and all its 1- and 2-hop neighbors are colored after completion of* **LocalD2Color**$(s, r, \widetilde{\Delta})$ *for $\widetilde{\Delta} = \Delta_k$ and $k \geq 8$.*

Proof. All 2-hop neighbors of 2-hop neighbors of s in the unit-disk graph w.r.t. transmission radius r are contained in a ball of radius $4r$ around s. Since $r \leq 2|l_s|$ we obtain a bound on the number of 2-hop neighbors of all 2-hop neighbors of s and can apply the proofs from [5].

By similar arguments, one can show that the coloring is valid w.h.p.

Lemma 5. *Let s be a node with $r \leq \varepsilon \cdot 2^{\lceil \log_2(l_s/\varepsilon) \rceil}$, and let u be its fixed 1- or 2-hop neighbor, and assume both have been assigned a color during the procedure. Then with high probability of $1 - \frac{1}{n^\Psi}$, s and u have been assigned distinct colors for $\widetilde{\Delta} = \delta_k$ and $k \geq 8$.*

Using these Lemmas and Lemma 3 with $h = 3$ it is easy to conclude the following.

Lemma 6. **LocalD2Color**$(s, r, \widetilde{\Delta})$ *computes in time $O(\widetilde{\Delta} \log^2 n)$ a coloring of the nodes such that for each node s with $r \leq \varepsilon \cdot 2^{\lceil \log_2(l_s/\varepsilon) \rceil}$, s and its D2-neighbors in $UDG(r)$ are assigned different colors with probability $1 - \frac{1}{n^\Psi}$ for arbitrary $\psi \geq 1$ if $\widetilde{\Delta} = \Delta_k$, $k \geq 8$.*

3.3 AnnounceNApproveEdges(s)

In this procedure each node first locally computes a list of adjacent Gabriel edges based upon its knowledge of the 1-hop neighbors, and then announces them one by one. Announcements take $O(\widetilde{\Delta}^{1/2})$ rounds of $2\widetilde{\Delta}$ time steps each. In each round, a node s announces in time slot $2 \cdot \text{color}(s)$ a Gabriel edge $e = (s, t)$ from its list that has not been announced by the other end-node t of e. If no neighbor of s transmits at the same time slot and no VETOs or collisions are encountered in time slot $2 \cdot \text{color}(s) + 1$, s and t regard the edge e as an edge of the final graph $GG(UDG)$. Furthermore, s listens to all announcements of other nodes and sends a VETO message in the following time slot if it either experiences collision (two or more neighboring nodes announce at the same time slot) or its own position contradicts the creation of the announced edge (i.e. s lies in the diametral circle of the announced edge).

Let us first show that non-Gabriel edges always get vetoed.

Lemma 7. **AnnounceNApproveEdges**(s) *never creates an edge that is not a Gabriel edge (for arbitrary transmission radius r).*

Proof. Assume otherwise, i.e. a node v created an edge e whose diametral circle contains a 1-hop neighbor u of v. Then e must have been announced at some point, and no VETO message, in particular none from u, was received and no collision was experienced at v. But u should have sent a VETO regardless of whether it received the announcement or experienced a collision. Thus we have a contradiction.

Next we show that w.h.p. all Gabriel edges of a fixed node s are constructed in the round where $r = \varepsilon \cdot 2^{\lceil \log_2(l_s/\varepsilon) \rceil}$. With Lemma 7, this implies that the computed $GG(UDG)$ is correct.

Lemma 8. *Let s be a node with $\varepsilon \cdot 2^{\lfloor \log_2(l_s/\varepsilon) \rfloor} \le r \le \varepsilon \cdot 2^{\lceil \log_2(l_s/\varepsilon) \rceil}$. Then with high probability all Gabriel edges adjacent to s are constructed if $\widetilde{\Delta} = \Delta_k$, $k \ge 8$.*

Proof. By Lemma 3, only correct edges are announced by s or its 1-hop neighbors. By Lemma 6, no collisions appear during the execution of **AnnounceNApproveEdges**(s). It remains to prove that $O(\widetilde{\Delta}^{1/2})$ rounds suffice to have all Gabriel edges of s announced (Note that there could be $\Theta(\widetilde{\Delta})$ many, and without the help of the other endnodes of these edges it would take $\Theta(\widetilde{\Delta})$ rounds or $\Theta(\widetilde{\Delta}^2)$ time slots for s to announce them all!).

Let us call a node *active* if not all of its adjacent Gabriel edges have been announced. Let a_i be the number of active neighbors of s in the beginning of round i. Then at every round, at least $a_i/2$ edges *in the 2-hop neighborhood* of s get announced, since each such announcement can deactivate at most two neighbors of s. The 2-hop neighborhood initially contains $O(\widetilde{\Delta})$ unannounced edges, so there are $O(\widetilde{\Delta}^{1/2})$ rounds with $a_i = \Omega(\widetilde{\Delta}^{1/2})$. Once it becomes $a_i = O(\widetilde{\Delta}^{1/2})$, it is clear that all the remaining Gabriel edges adjacent to s will be announced by s itself in additional $O(\widetilde{\Delta}^{1/2})$ rounds.

3.4 FinalD2Color$(s, |l_s|, \widetilde{\Delta})$

At this point, each node s w.h.p. knows about the topology of the *directed* disk graph *DG*, that is about the nodes that lie within its own transmission range, as well as the nodes in whose transmission range it lies. These correspond to the *outgoing* and *incoming* edges of s in *DG*, respectively. Indeed, s can learn about its outgoing edges at the time when its longest Gabriel edge is created, and about its incoming edges after running another 'announcement phase' in the spirit of **ExploreDirectNeighborhood**, with the only difference that each node actually transmits within the range determined by its longest adjacent Gabriel edge. Then with high probability all nodes get to know about their incoming edges (since again, interference is limited by the parameter $\widetilde{\Delta}$).

It remains to make the final assignment of time slots that will be used during the lifetime of the network. As explained previously, the purpose of this step is to save on the number of colors required by taking into account the nodes' effective radii of communication.

However, the algorithm of [5] cannot be directly applied to construct the final coloring. It may happen that nodes u and v propose the same color in TRIAL phase, and their messages collide *only* at node w, but w is unable to communicate that fact to u and v because its communication radius is too small. More precisely, the problem is that the *DG* induced by $GG(UDG)$ is *directed*, because the communication radii are different in general. The D2-coloring scheme requires that the *DG* be undirected, so that the nodes can communicate in "propose–respond" fashion.

Fortunately, there is a simple modification that gets around this problem without affecting the asymptotic performance. The TRIAL and SUCCESS phases do not change, while in the TRIAL-REPORT and SUCCESS-REPORT phase each node transmits within the radius equal to the length of its longest incoming edge. By keeping the TRIAL and SUCCESS unchanged, we make sure that the collisions generated during rounds of coloring are exactly those that would occur in the final communication scheme. On the other hand, modification to the REPORT phases makes sure that the coloring is correct, but may hurt the running time; increasing the radii for reporting purpose may create a large number of new D2-neighbors, and therefore many collisions among the REPORT messages. However, we show that the new neighborhood size has already been accounted for in the preceding analysis. The actions for fixing a color after reception the TRIAL-REPORT as well as for removal of colors from the list after reception of a SUCCESS-REPORT of course then restrict to reports received from nodes within the transmission range of a node.

Lemma 9. *Let G be the DG induced by $GG(UDG)$, and let H be the undirected version of G obtained by setting the radius of each node to the length of its longest incoming edge in G. Then any node in H has a D2-neighborhood of size at most Δ_k, for any $k \geq 5$.*

Proof. Consider a fixed node u and its D2-neighborhood N_u in H. Clearly, the edges in G that connect nodes in N_u are also present in H. Let (v,w) be the longest incoming edge of the nodes in N_u (w is in N_u). Then the whole N_u at most $5|vw|$ away from v. Also, (v,w) emanates from v in G, therefore $|vw| \leq l_v$. Thus, $B(v, 5l_v)$ contains all nodes in N_u, which implies $N_u \leq \Delta_5(v) \leq \Delta_5 \leq \Delta_k$, for any $k \leq 5$. This completes the proof.

In other words, the increased transmission radii (to the length of the longest *incoming* edge) create only limited potential interferences, so w.h.p. the TRIAL-REPORT and TRIAL-SUCCESS phases succeed.

Lemma 10. *In time $O(\widetilde{\Delta} \log^2 n)$,* **FinalD2Color**$(s, |l_s|, \widetilde{\Delta})$ *w.h.p. computes a valid D2-coloring of the final communication graph DG.*

3.5 Summary and Further Remarks

We summarize the statements of Lemmas 3, 6, 7,and 8, 10 and give our main theorem.

Theorem 1. *Our algorithm runs in time $O(\widetilde{\Delta}^{3/2} \log \frac{1}{\varepsilon} \log^2 n)$ and w.h.p. computes the $GG(UDG)$, as well as a coloring of its nodes which ensures interference-free communication between nodes adjacent in $GG(UDG)$.*

Here $\widetilde{\Delta} = \Delta_8$ denotes the maximum number of points within the distance of $8 \cdot |l_s|$ from s, where $|l_s|$ is the length of the longest Gabriel edge adjacent to s. Thus, we can relate the running time of the algorithm to the number of 'nearby' stations relative to its final assigned transmission range, instead of the number of stations within its maximum transmission range. ε can be set to the minimum inter-node distance (as e.g. derived from the physical size of the nodes) or even larger, as long as the number of nodes contained in any disk of radius 8ε is bounded by $\widetilde{\Delta}$.

We want to remark that k can be decreased from 8 to a value arbitrarily close to 4, by instead of doubling the radius r, multiplying it by factor smaller than 2.

4 Lipschitz-type Node Distributions

Consider an application of monitoring physical quantities (such as temperature, humidity, exposure to light etc.) over an area A of a nature preserve. Suppose that the area contains a set H of interesting 'hotspots' (e.g. breeding areas of a bird species) which should be monitored more accurately. Scientists want to deploy more wireless sensors close to the hotspots, and fewer sensors further away.

Formally, they might assume that there is an 'interest function' $f : A \to \mathbb{R}$ which is small in regions of high interest and vice versa. An example is $f(x) = d(x,H)$, the distance to the closest hotspot. Then, if the deployment of the set of sensors S satisfies the condition $\forall x \in A : \exists s \in S : d(x,s) \leq \varepsilon \cdot f(x)$ for some small $\varepsilon \in (0,1)$, enough data is gathered. On the other hand, sensor nodes are expensive, so only the necessary number should be used; in terms of the interest function f, the required condition is $\forall x \in A$: $|\{s \in S : d(x,s) \leq \varepsilon \cdot f(x)\}| \leq \beta$ for some constant β.

If the distribution of the deployed sensors adheres to the above conditions, our algorithm performs very well, since the value $\widetilde{\Delta}$, which has to be provided to every node upon deployment, is a reasonably small constant. Roughly speaking, our algorithm performs well if the node density has *bounded variation* as a function of the spatial coordinates. Below we give a more formal statement of this fact.

We want to remark that this property of gradually changing node-densities naturally also arises in other settings. For example, consider the application of tracking a set of point-sized *slowly moving* objects. If we are interested in minimizing the effort needed to adapt the sensor distribution to possible movement of objects, it is reasonable to have a gradually increasing sensor density as the distance to an object (in its current position) becomes smaller. Let us formalize the situation described above.

Definition 2. *Let S be a set of wireless nodes placed within some area of interest A. We call S a* Lipschitz-type node distribution,*if there exists an* α*-Lipschitz function* $\varphi : A \to \mathbb{R}$ *with* $\alpha > 0$ *and constants* $\varepsilon > 0$, $\beta \geq 1$ *such that for all* $x \in A$, $1 \leq |S \cap B(x, \varepsilon\varphi(x))| \leq \beta$.

In our example above we had $\varphi(x) = f(x)$ with $\alpha = 1$ (as f was defined to be the distance to a set of points, it is 1-Lipschitz). Observe though that this definition of Lipschitz-type node distributions also allows for 'oversampling' of the domain of interest, as long as the oversampling happens in a smooth manner.

Lemma 11. *Let S be a Lipschitz-type node distribution,with parameters* $\alpha, \beta, \varepsilon$*, and* $\varepsilon < \frac{1}{2\alpha(k+1)}$*, then we have* $\Delta_k = O(1)$.

Proof. Consider a fixed node u, and let (u,v) be its longest adjacent edge in $GG(UDG)$, $|uv| = l_u$. Let $x \in A$ be the midpoint of (u,v). Clearly, $l_u/2 \leq \varepsilon\varphi(x)$. Let us define $R = (k+1)l_u$, and hence $\Delta_k \leq |B(x,R) \cap S|$. So in the following we will concentrate on bounding the number of nodes within $B(x,R)$.

Using the inequalities above we get $R \leq (k+1) \cdot 2 \cdot \varepsilon\varphi(x)$. As φ is α-Lipschitz, we have for any $y \in B(x,R)$ that $\varphi(y) \geq \varphi(x) - \alpha R \geq \varphi(x)(1 - 2(k+1)\alpha\varepsilon)$, that is, any ball of radius at most $r = \varepsilon\varphi(x)(1 - 2(k+1)\alpha\varepsilon)$ centered within $B(x,R)$ contains less than β nodes. It remains to bound the number of balls of radius r to cover $B(x,R)$. But that number is $O((\frac{k+1}{1-2(k+1)\varepsilon\alpha})^2)$, which for constant values of k, ε with $\varepsilon < \frac{1}{2\alpha(k+1)}$ remains a constant. That is, at $\Delta_k \leq |B(x,R) \cap S| = O(\beta) = O(1)$.

It follows that for the above example with $\alpha = 1$, the condition of Definition 2 is satisfied with $\varepsilon < 1/18$ in order to guarantee $\tilde{\Delta} \leq \Delta_8 = O(1)$, so the algorithm runs in

$O(\log \frac{1}{\varepsilon} \log^2 n)$. We emphasize, however, that this theoretical analysis is very pessimistic and we expect the algorithm to perform very well for many practical node distributions.

5 Conclusions

In this paper we have presented a distributed protocol for constructing both the low- and the high-level infrastructure for a set of newly deployed wireless stations. Our *integrated approach* remedies some of the inherent problems incurred by a sequential construction of first low- and then high-level infrastructure, which is particularly apparent for dense and varying node distributions. We believe that other high-level infrastructures can be similarly constructed in this manner. It might be interesting to see whether our approach can also be adapted to work under the less restrictive model of node capabilities as used in [6] and [8].

References

1. M. Burkhart, P. von Rickenbach, R. Wattenhofer, and A. Zollinger, "Does topology control reduce interference?" *5th ACM Interational Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2004.
2. X. Cheng, X. Huang, D. Li and D.-Z. Du, "Polynomial-time approximation scheme for minimum connected dominating set in ad hoc wireless networks," *Networks*, to appear.
3. B. Clark, C. Colbourn and D. Johnson, "Unit Disk Graphs," *Discrete Mathematics*, Vol. 86, pp. 165-177, 1990.
4. K.R. Gabriel and R.R. Sokal, "A new statistical approach to geographic variation analysis", *Systematic Zoology*, vol 18, pp. 259–278, 1969
5. R. Gandhi and S. Parthasarathy, "Fast Distributed Well Connected Dominating Sets for Ad Hoc Networks," *CS-TR-4559*, UM Computer Science Department.
6. F. Kuhn, T. Moscibroda, and R. Wattenhofer. "Initializing Newly Deployed Ad Hoc and Sensor Networks" *10th Ann. Int. Conf. on Mobile Computing and Networking (MOBICOM)*, Philadelphia, USA, Sept. 2004.
7. F. Meyer auf der Heide, C. Schindelhauer, K. Volbert, and M. Grünewald, "Congestion, Dilation, and Energy in Radio Networks" *Theory of Computing Systems 37 (3), 2004*
8. T. Moscibroda and R. Wattenhofer "Efficient Computation of Maximal Independent Sets in Unstructured Multi-Hop Radio Networks" *1st IEEE Int. Conf. on Mobile Ad-hoc and Sensor Systems (MASS)*, Fort Lauderdale, USA, Oct. 2004.
9. X.-Y.Li, W.-Z.Song, Y.Wang. "Localized Topology Control for Heterogenous Wireless Ad Hoc Networks", *1st IEEE Int. Conf. on Mobile Ad hoc and Sensor Systems (MASS)*, 2004.
10. X.-Y. Li, P.-J. Wan, W. Yu, O. Frieder. "Sparse power efficient topology for wireless networks", *IEEE Hawaiian Int. Conf. on System Sciences (HICSS)*, 2002.
11. S.Ramanathan. "A unified framework and algorithm for channel assignment in wireless networks", *Wireless Networks*, 5(2):81-94, 1999
12. A.Sen and M.L.Huson. "A new model for scheduling packet radio networks", *Wireless Networks*, 3(1):71-82, 1997
13. A. Sen and E. Melesinska. "On approximation algorithms for radio network scheduling", *Proc. 35th Allerton Conf. on Communication, Control, Computing*,pp.573–582, 1997.
14. P. J. Wan, K. Alzoubi and O. Frieder, "Distributed Construction of Connected Dominating Set in Wireless ad hoc networks," *Proc. of INFOCOM 2002*.