Multi-Channel Scheduling Algorithms for Fast Aggregated Convergecast in Sensor Networks

Amitabha Ghosh*, Özlem Durmaz Incel†, V.S. Anil Kumar‡, Bhaskar Krishnamachari*
*Ming Hsieh Department of Electrical Engineering, University of Southern California
Los Angeles, CA 90089, USA, {amitabhg, bkrishna}@usc.edu
†NETLAB, Department of Computer Engineering, Bogazici University
34342 Bebek, Istanbul, Turkey, ozlem.durmaz@tam.boun.edu.tr
‡Department of Computer Science and Virginia Bio-Informatics Institute, Virginia Tech
Blacksburg, VA 24061, USA, akumar@vbi.vt.edu

***Abstract*—Fast and periodic collection of aggregated data is of considerable interest for mission-critical and continuous monitoring applications in sensor networks. In the many-to-one communication paradigm known as convergecast, we consider scenarios where data packets are aggregated at each hop *en route* to a sink node along a tree-based routing topology and focus on maximizing the data collection rate at the sink by employing TDMA scheduling and multiple frequency channels.**

Our key result in the paper lies in proving that minimizing the schedule length for an *arbitrary* network in the presence of multiple frequencies is NP-hard, and in designing approximation algorithms with worst-case provable performance guarantees for *geometric* networks. In particular, we design a constant factor approximation for networks modeled as unit disk graphs (UDG) where every node has a uniform transmission range, and a $O(\Delta(T) \log n)$ approximation for general disk graphs where nodes have different transmission ranges; n is the number of nodes in the network and $\Delta(T)$ is the maximum node degree on a given routing tree T. We also prove that a constant factor approximation is achievable on UDG even for *unknown routing topologies* so long as the maximum node degree in the tree is bounded by a constant. We also show that finding the minimum number of frequencies required to remove all the interfering links in an arbitrary network in NP-hard. We give an upper bound on the maximum number of such frequencies required and propose a polynomial time algorithm that minimizes the schedule length under this scenario. Finally, we evaluate our algorithms through simulations and show various trends in performance for different network parameters.

I. Introduction

Consider a large-scale wireless sensor network (WSN) deployed for a continuous and periodic monitoring application, such as a security-surveillance network for monitoring facilities, or an environmental network for monitoring critical phenomena. The successful operation of such an application depends on its ability to extract data from the network, which often comprises periodic summaries or aggregates of raw sensor readings. It also requires *fast* and *periodic* delivery of this aggregated data from the source nodes to a common sink. Typically, the routing structure used for such data collection is a spanning tree rooted at the sink. As the data flows up this routing tree, it is aggregated at each hop thereby, eliminating redundancy and minimizing the number of transmissions [13]. We refer to this process of many-to-one communication of aggregated data from the various sources to the sink node as *aggregated convergecast* [12].

In this paper, we focus on the *link scheduling problem* of maximizing the aggregated data collection rate at the sink node under the setting of TDMA protocols and *multiple frequency channels*. The key challenge in designing efficient solutions to such scheduling problems is the presence of wireless interference, which arises from concurrently transmitting nodes that are in close proximity of each other. While there is a lot of research on single-channel scheduling protocol design for WSN, exploiting parallelism using multiple channels has not been well explored. Moreover, given the fact that current WSN hardware already provides multiple frequencies, such as the 16 orthogonal frequencies with 5MHz spacing supported by CC2420 radios on TmoteSky, it is imperative to take their advantage in minimizing the interference and achieving a faster data collection rate by concurrent transmissions. In addition to multiple frequencies, we consider contention-free multiple access protocols (e.g., TDMA) due to their inherent ability in eliminating collisions and retransmissions, and thus achieving better performance in periodic data collection scenarios as opposed to contention-based protocols [14].

In our time slotted system, the duration of each slot is long enough to accommodate the successful transmission or reception of a single packet. We assume that packets are of the same size, and equal numbers of consecutive slots are grouped into non-overlapping frames that are repeated for periodic scheduling. We also assume that each node generates only one packet in the beginning of every frame, and each node has the ability to aggregate all the packets from its children as well as its own into a single packet before transmitting to its parent. The classes of aggregation functions that fall in this category include *distributive* and *algebraic* functions [13], where the size of the aggregated data is constant (e.g., fits within a single packet) and does not depend on the size of the raw measurements. Examples of such aggregation functions are MIN, MAX, MEDIAN, COUNT, SUM, AVERAGE, etc. Thus, if every node continues to transmit only once during each frame, then the sink will start receiving aggregated data from *all* the nodes in the network after a certain time. This is when we say that the network has reached a *pipelined*

978-1-4244-5113-5/09/$25.00 ©2009 IEEE

state. Note that, due to periodic scheduling, once a pipeline is established, the sink keeps receiving aggregated data from all the nodes once every frame. We call the number of time slots in each frame the *schedule length*. Under this scenario, maximizing the aggregated data collection rate at the sink node is equivalent to minimizing the schedule length.

We prove that minimizing the schedule length for an arbitrary network in the presence of multiple frequencies is NP-hard and design approximation algorithms for *arbitrarily* deployed *geometric* networks. In particular, we design a constant factor approximation for networks modeled as unit disk graphs, where every node has a uniform transmission range, and a $O(\Delta(T) \log n)$-approximation for general disk graphs, where nodes could have different transmission ranges. Here, n is the number of nodes in the network, and $\Delta(T)$ is the maximum node degree on a given routing tree T. We also prove that a constant factor approximation is achievable on UDG even when the routing topology is unknown to the algorithm designer a priori so long as the maximum in-degree of any node in the tree is bounded by a constant. We also show that finding the minimum number of frequencies required to remove all the interfering links in an arbitrary network in NP-hard. We give an upper bound on the maximum number of such frequencies required and propose a polynomial time algorithm that minimizes the schedule length under this scenario. Finally, we evaluate our algorithms through simulations and show various trends in performance for different network parameters.

A. Related Work and Paper Overview

The non-aggregated version of the convergecast problem has been considered by Gandham *et al.* in the presence of a single channel and TDMA protocols, where the goal is to minimize the schedule length [9]. The authors describe an integer linear programming formulation and propose a distributed scheduling algorithm that requires at most $3n$ time slots for general networks, where n is the number of nodes in the network. A similar study [5] is presented by Choi *et al.* in which an NP-completeness result is proved on minimizing the schedule length under a single frequency for non-aggregated convergecast. Minimizing the schedule length by using orthogonal codes or hopping sequences to get rid of interference is studied by Annamalai *et al.*, where they consider assigning different time slots and code pair to interfering links [1].

The problem of joint scheduling, routing, and transmission power control to improve network throughput and interference is studied by Bhatia *et al.* [3], and also by Bhat *et al.* [7]. A prominent recent work was by Moscibroda, in which scaling laws describing the achievable rate for aggregated convergecast in arbitrarily deployed networks were presented under the SINR (*Signal-to-Interference-plus-Noise-Ratio*) model [15]. Worst-case capacity results were also proved by employing non-linear power assignment to nodes and exploiting SINR-effects. Cruz *et al.* used a duality based approach to address the problem of finding an optimal link scheduling and power control policy, which minimized the total average transmission power and support high data rates [6].

In the context of general ad hoc networks, the use of multiple channels has been well researched. To improve network throughput, So *et al.* proposed a MAC protocol that could switch channels dynamically and could avoid the hidden terminal problem using temporal synchronization [18]. A link-layer protocol called SSCH was proposed by Bahl *et al.* that could increase the capacity of IEEE 802.11 networks by utilizing frequency diversity [2]. In the context of WSN, there exist fewer works utilizing multiple channels. The first multi-frequency MAC protocol, MMSN, was proposed by Zhou *et al.* where the goal was to increase aggregated throughput [20].

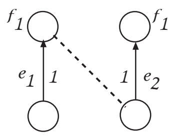
Most closely related is our previous work [12], in which we described a realistic simulation-based study on tree-based data collection utilizing transmission power control, multiple frequencies, and efficient routing topologies. It is shown that once all the interfering links are removed by use of multiple frequencies, the data collection rate becomes limited by the maximum degree of the tree. We also showed that this rate can further be increased on degree-constrained trees. Our present work is different from the rest in that we propose algorithms and prove several important theoretical results on the aggregated convergecast problem under multiple frequencies.

The rest of the paper is organized as follows. In Section II, we describe our problem formulation and assumptions. In Section III, we prove two complexity results on general graphs for the aggregated convergecast problem. In Section IV, we focus on unit disk graphs and propose frequency and time slot assignment schemes that achieve constant factor approximation on the optimal schedule length. Section V focuses on general disk graphs. In Section VI we present our evaluation results, and finally Section VII concludes the paper.

II. Preliminaries

In this paper, we are interested in designing link scheduling protocols that exhibit provably good performance on *arbitrarily* deployed networks (possibly even *worst case*) in the Euclidean plane under multiple frequencies. We model the network as an undirected graph $G = (V, E)$, where V is the set of nodes and an edge $e = (u, v) \in E$ exists between any two nodes $u, v \in V$ if they are within the communication range of each other. We are also given a fixed node $s \in V$ that represents the sink, and a spanning tree $T = (V, E_T)$ rooted at s that serves as the routing topology. All the nodes except s generate packets.

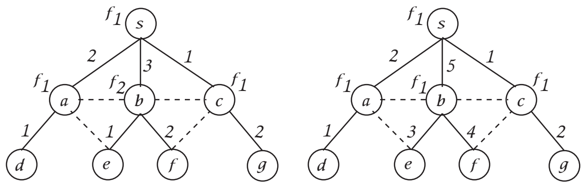
We assume that each node has a single, half-duplex transceiver, so it can either transmit or receive a single packet at any given time slot. We also assume that transmissions on orthogonal channels do not interfere with each other. Although this assumption may fail in practice depending on adjacent and alternate channel rejection values for different types of transceivers, experimental results [12] presented by Incel *et al.* show that the scheduling performance remains similar for CC2420 and Nordic nrf905 radios.



(a)

	Frame 1			Frame 2		
Receiver	Slot 1	Slot 2	Slot 3	Slot 1	Slot 2	Slot 3
s	c	a,d	b,e,f	c,g	a,d	b,e,f
a	d			d		
b	e	f		e	f	
c		g			g	

(b)



(c) (d)

Fig. 1. (a) Secondary interfering link. (c), (d) Aggregated convergecast with 7 source nodes with two frequencies and a single frequency. (b) Pipeline with two frequencies starts from frame 2 with a minimum schedule length 3.

In our formulation, we consider a *receiver-based frequency assignment* strategy, i.e., we statically assign a frequency to each of the receivers (non-leaf nodes) of T. Although in practice, every sender-receiver pair could potentially negotiate on a particular frequency before each packet transmission, assigning different frequencies to the transmitters that are children of the same parent does not significantly help in reducing the schedule length. This is because the single-transceiver radio cannot receive multiple packets simultaneously. Moreover, pair-wise, per-packet frequency negotiation might create unnecessary overhead. Thus, the children of the same parent transmit on the parent's frequency, and so a node operates on at most two frequencies.

We assume a graph-based interference model (also called the *protocol model*), where the interference range of a node is equal to its transmission range, and concurrent transmissions on two edges $e_1, e_2 \in E$ interfere with each other if either (i) they are adjacent, or (ii) both the receivers of e_1 and e_2 are on the same frequency and at least one of the receivers is within the communication range of the other transmitter. The first type of interference is known as the *primary interference* while the second type is known as the *secondary interference*, as illustrated in Fig. 1(a).

In Fig. 1(c) and 1(d), we illustrate aggregated convergecast and the advantages of using multiple frequencies on a network of 7 source nodes. The dotted lines represent interfering links and the solid lines represent tree edges. A number beside an edge represents the time slot in which the edge is scheduled to transmit. The entries in Fig. 1(b) list the source nodes from which data is received on the corresponding time slot. For example, s receives aggregated data from b, e, and f in slot 3 starting from frame 1. In this case, it takes two frames to reach a pipelined state, because data from g does not reach s in frame 1. Thus, from frame 2 onwards, s receives aggregated data from all the nodes in the network once in every three time slots; so the minimum schedule length is 3. Note that, there may exist other assignments, such as f_2 to a, c, and s, and f_1 to b also yielding the same schedule length. However, if we had only one frequency, the minimum schedule length is 5, as shown in Fig. 1(d).

III. Assignment on General Graphs

A. Scheduling With Unlimited Frequencies

From the illustration above, we see that assigning different frequencies to the receivers in an appropriate way can mitigate the effects of interference and shorten the schedule length. In this subsection, we first study the problem of finding the *minimum* number of frequencies needed to remove *all* the secondary interfering links. We say that a secondary interfering link is removed if the two receivers of an edge pair are assigned different frequencies. Note that, primary interference cannot be removed using multiple frequencies because of half-duplex radios.

Minimum Frequency Assignment: Given a tree T on an arbitrary graph G and an integer q, is there a frequency assignment to the receivers in T using at most q frequencies such that all the secondary interfering links are removed?

Theorem 1. *Minimum Frequency Assignment is NP-complete.*

The proof is by reduction from the Vertex Color problem and is omitted here due to lack of space. We now give an upper bound on the maximum number of frequencies required to remove all the secondary interfering links and describe a polynomial time algorithm that minimizes the schedule length under this scenario.

Lemma 1. *Construct a constraint graph $G_C = (V_C, E_C)$ from the original graph G as follows. For each receiver (i.e., non-leaf nodes) in G, create a node in G_C. Create an edge between two such nodes in G_C if their corresponding receivers in G are on two edges that form secondary interfering links. Then, the number K_{max} of frequencies that will remove all the secondary interfering links is bounded by: $K_{max} \leq \Delta(G_C) + 1$, where $\Delta(G_C)$ is the maximum node degree in G_C.*

Proof: Since we create an edge between every two nodes in G_C whenever their corresponding receivers in G form a secondary interfering link, assigning different frequencies to every such receiver-pair in G is equivalent to assigning different colors to the adjacent nodes in G_C. Thus, K_{max} is equal to the minimum of the number of colors needed to vertex color G_C, called its *chromatic number* $\chi(G_C)$. Since $\chi(G) \leq \Delta(G) + 1$, for arbitrary G, the lemma follows. ■

Once all the secondary interfering links are removed, the problem of minimizing the schedule length on G reduces to that on tree T. In the following, we propose an algorithm, BFS-TimeSlotAssignment (Algorithm 1), that runs in $O(|E_T|^2)$ time and minimizes the schedule length.

In each iteration (lines 2-6) of the *Breadth-First Search* (BFS) time slot assignment, an edge e is chosen in the BFS order (starting from any node), and is assigned the minimum time slot that is different from all its adjacent edges. We prove such an assignment gives a minimum schedule length that is equal to the maximum degree $\Delta(T)$ of T.

Algorithm 1 BFS-TIMESLOTASSIGNMENT

```
Input: T = (V, E_T)
while E_T ≠ φ do
    e ← next edge from E_T in BFS order
    Assign minimum time slot to e respecting adjacency constraint
    E_T ← E_T \ {e}
end while
```

THEOREM 2. *Algorithm* BFS-TIMESLOTASSIGNMENT *on a tree T gives a minimum schedule length equal to $\Delta(T)$.*

Proof: The proof is by induction on i. Let $T^i = (V^i, E_T^i)$ denote the subtree of T in the i^{th} iteration constructed in the BFS order, where E_T^i comprises all the edges that are assigned a slot, and V^i comprises the set of nodes on which the edges in E_T^i are incident. Note that, $|E_T^i| = i$, because at every iteration exactly one edge is assigned a slot. For $i = 1$, clearly the number of slots used is 1, equal to $\Delta(T^1)$.

Now, assume that the number of slots $N(i)$ needed to schedule the edges in T^i is $\Delta(T^i)$. In the $(i+1)^{th}$ iteration, after assigning a slot to the next edge in BFS order, the number of slots needed in T^{i+1} can either remain the same as before, or increase by one. Thus,

$$N(i+1) = \max\{N(i), N(i)+1\} \tag{1}$$

If it remains the same, $N(i+1)$ is still the maximum degree of T^{i+1} at end of $(i+1)^{th}$ iteration. Otherwise, if it increases by one, the new edge must be incident on a node v^*, common to both T^i and T^{i+1}, such that the number of incident edges on v^* that were already assigned a time slot at the end of i^{th} iteration was $\Delta(T^i)$. This is so because in the BFS traversal, *all* the edges incident on a node are assigned a slot first before moving on to the next node, and because the slot assigned to the new edge is the minimum possible that is different from all that already assigned to the edges incident on v^* until the i^{th} iteration. Thus, at the end of $(i+1)^{th}$ iteration, the number of slots used $N(i)+1$ is equal to the number of assigned edges incident on v^* which, in turn, equals $\Delta(T^{i+1})$. This proves the inductive step. Therefore, it holds at every iteration of the algorithm until the end when $i = |V|-2$, yielding a schedule length equal to the maximum degree $\Delta(T) = \Delta(T^{|V|-1})$. Now, since assigning different time slots to the adjacent edges in T is equivalent to edge coloring T, which requires at least $\Delta(T)$ colors, the schedule length is minimum. ∎

B. Scheduling With Limited Number of Frequencies

We showed that with sufficient number of frequencies, all the secondary interfering links can be removed and a minimum length schedule can be found in polynomial time. However, typically there is a limitation on the number of frequencies over which a given transceiver can operate. In this subsection, we study the problem of minimizing the schedule length on an arbitrary graph when a limited number of frequencies are available (*Multi-Frequency Scheduling* problem). First, we state a known result in Lemma 2 on *distance-2-edge-coloring* (also called *strong edge coloring*) on trees that we use in the proof of one of our key results in Theorem 3.

DEFINITION 1. *Two edges $e, e' \in E$ in a graph $G = (V, E)$ are within distance 2 of each other if either they are adjacent or both are incident on a common edge.*

A *distance-2-edge-coloring* of G requires that every two edges that are within distance 2 of each other have distinct colors. The fewest such colors needed is called the *strong chromatic index*, $s\chi'(G)$, and finding it for general graphs is known to be NP-hard [10]. It is easy to see that even when all the receivers in G are assigned the same frequency, the minimum schedule length is no more than $s\chi'(G)$.

LEMMA 2. *The strong chromatic index $s\chi'(T)$ of a tree $T = (V, E_T)$ is given by [8]:*

$$s\chi'(T) = \max_{(u,v)\in E_T} \{deg(u) + deg(v) - 1\}$$

Multi-Frequency Scheduling: Given a tree T on an arbitrary graph G, and two positive integers p and q, is there an assignment of time slots to the edges in T using at most q frequencies such that the schedule length is at most p?

THEOREM 3. *Multi-Frequency Scheduling is NP-complete.*

Proof: It is easy to see that Multi-Frequency Scheduling is in NP. Given a particular assignment, one can verify in polynomial time that - (i) at most q frequencies and p time slots are used, (ii) either the receivers of every secondary interfering link are assigned different frequencies or their edges are on different time slots, and (iii) all adjacent edges are on different time slots.

To show NP-hardness, we reduce an instance $G' = (V', E')$ of the Vertex Color problem to an instance $G = (V, E)$ of the Multi-Frequency Scheduling problem, as illustrated with an example in Fig. 2. Our gadgets for reduction are as follows.

Let $|V'| = n$. For every vertex $v_i \in V'$, create a set S_i of q pairs of nodes $\{(u_{is}, v_{is}) : s = 1, \ldots, q\}$ in G, and join each pair with an edge e_{is}, treating u_{is} as the parent of v_{is}. Then, create $\binom{q}{2} = q(q-1)/2$ secondary interfering links between all such pairs in each S_i as follows. Consider each u_{is} in turn, for $s = 1, \ldots, q-1$, and create an edge from u_{is} to v_{il}, for all $l > s$ (see Fig. 2(b) for $q = 2$).

Next, for every edge $e_{ij} = (v_i, v_j) \in E'$, create q^2 secondary interfering links in G by considering the two sets: $S_i = \{(u_{is}, v_{is}) : s = 1, \ldots, q\}$ and $S_j = \{(u_{js}, v_{js}) : s = 1, \ldots, q\}$, and creating an edge from each u_{is} to each v_{js}. Then, for each S_i, construct a binary tree T_b^i creating additional nodes and edges, and treating the set $\{u_{is}\}$ of nodes as leaves, for $s = 1, \ldots, q$.

Finally, treating the roots of T_b^i's as leaves, create a binary tree on top of it, and designate the root of it as the sink s. The reduction clearly runs in polynomial time and creates an instance of the Multi-Frequency Scheduling problem. Next, we show that there exists a solution to the Vertex Color problem using at most p colors if and only if there exists an assignment in T using at most q frequencies and at most p plus a constant number of time slots.

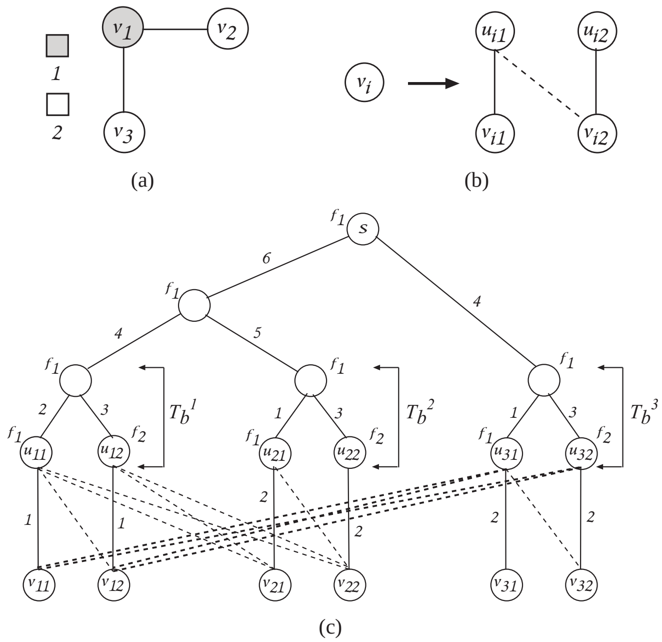


Fig. 2. Reduction for Multi-Frequency Scheduling: (a) Instance G' of the vertex color problem. (b) Gadget for each v_i in G' for $q = 2$. (c) Instance G of Multi-Frequency Scheduling as constructed from G' for $q = 2$.

Suppose G' is vertex colorable using at most p colors, and v_i is assigned color i. First, assign frequency f_s to u_{is}, for $s = 1, ..., q$, in each S_i, and any one of the q frequencies, say f_1, to all the parents in the rest of tree (see Fig. 2(c)). Then, assign time slot i to all the q edges connecting the pairs (u_{is}, v_{is}), for $s = 1, ..., q$, in each S_i. Because all the receivers in S_i are on different frequencies, assigning the same time slot to all the edges in S_i does not create secondary interfering links within each S_i. Also, since only non-adjacent vertices in G' may have the same color, any two sets of edges S_i and S_j that are on the same time slot cannot have secondary interfering links between each other.

Next, the lowest level edges, which connect to the $\{u_{is}\}$ nodes, of all the binary trees T_b^i, $\forall i$, can be scheduled using at most 2 time slots, because all the edges in each S_i are assigned the same slot. Finally, all the remaining edges in the binary tree can be scheduled in polynomial time because a distance-2-edge-coloring on trees can be computed in polynomial time [17], and within number of time slots no more than its strong chromatic index which, from Lemma 2, equals at most 5.

Conversely, suppose there exists a valid assignment in G that uses at most q frequencies and at most p plus a constant number of time slots. Assign colors to the vertices in G' as follows. For each frequency f_s, consider the set of edges $E_{ts} = \{(u_{ts}, v_{ts})\}$, which are assigned time slot t, for $t = 1, ..., p$, in order. Since the edges in E_{ts} are on the same slot and their receivers are on the same frequency, they cannot form secondary interfering links, and so each one of them must lie in a different S_i. Therefore, each edge in E_{ts} has a corresponding vertex in G', no two of which are adjacent. Select those edges in E_{ts} whose corresponding vertices are unassigned, and assign color t to all of them. Repeat the above assignment for all the frequencies f_s, for $s = 1, \ldots, q$. Clearly, this uses at most p colors and assigns different colors to adjacent vertices. Also, because we run the above procedure over all frequencies and over all time slots, and select an edge from E_{ts} only if its corresponding vertex is unassigned, exactly one edge gets picked from each S_i. Therefore, every node in G' gets a proper color, and the theorem follows. ■

IV. Assignment on Unit Disk Graphs

In this section, we consider networks that are modeled as unit disk graphs and design approximation algorithms on the optimal schedule length.

We first consider the scenario when the routing topology is known a priori. In the next subsection, we consider the case when the routing topology is unknown to the algorithm designer. The significance of the latter is that an optimal algorithm can then choose any arbitrary routing tree to minimize the schedule length, while the goal of the algorithm designer will be to construct a *good* routing topology such that scheduling on top of it would still guarantee provably good performance bounds.

A. Known Routing Topology

We divide the area covering all the nodes into a set of grid cells $\{c_i\}$, each of size $\alpha \times \alpha$, as illustrated in Fig. 3. Under a UDG model, there exists an edge between any two nodes that are at most a unit distance apart from each other. We say that two cells are *adjacent* to each other if they share a common edge or a common grid point. We say that an edge e_k *belongs* to a cell c_i if the receiver of e_k lies within c_i. Thus, a cell can have 3, 5, or 8 adjacent cells depending on whether it is a corner cell, an edge cell, or an interior cell, respectively. Since the interfering links are of length at most one, interference is spatially restricted, and so time slots can be reused across cells that are spatially well separated.

In our approach to design an approximation algorithm for minimizing the schedule length, we separate the frequency and time slot assignment phases. We first assign the frequencies to the receivers in T such that the maximum number of nodes transmitting on the same frequency is minimized. Then, we employ a greedy time slot assignment scheme that guarantees a constant approximation factor.

1) Frequency Assignment: Let $R_{c_i} = \{v_1, \ldots, v_n\}$ denote the set of receivers on the given routing tree T that lie in cell c_i, and let $m : R_{c_i} \rightarrow \{f_1, \ldots, f_K\}$ be a mapping that assigns a frequency to each of these receivers. Note that if $m(v_j) = f_k$, then the children of v_j transmit on f_k.

Definition 2. *We define a load-balanced frequency assignment in c_i as an assignment of the K frequencies to the receivers in R_{c_i} such that the maximum number of nodes transmitting on the same frequency is minimized.*

To formulate this, we define the *load* on frequency f_k in cell c_i under mapping m as the total number of children of

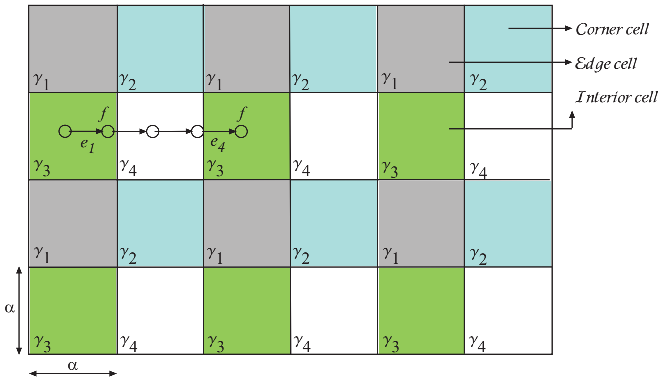


Fig. 3. Four pair-wise disjoint sets of time slots γ_1, γ_2, γ_3, γ_4 schedule the whole network. Each set maps to a distinct color.

all the receivers in R_{c_i} that are assigned f_k, and denote it by $l^m_{c_i}(f_k)$. We call the number of children of v_j its *in-degree*, and denote it by $deg^{in}(v_j)$. Thus,

$$l^m_{c_i}(f_k) = \sum_{v_j \in R_{c_i}: m(v_j)=f_k} deg^{in}(v_j) \tag{2}$$

Then, a load-balanced frequency assignment m^* in c_i is:

$$m^* = \arg\min_m \max_{f_k} \left\{ l^m_{c_i}(f_k) \right\} \tag{3}$$

We denote the load on the maximally loaded frequency under m^* in c_i by $L^{m^*}_{c_i}$. Finding a load-balanced frequency assignment is equivalent to, as shown in Lemma 3, scheduling jobs on identical machines to minimize the *makespan* (last finishing time of the given jobs), which is known to be NP-hard [11]. In the following, we describe an algorithm FREQUENCYGREEDY (Algorithm 2) that achieves a $\left(\frac{4}{3} - \frac{1}{3K}\right)$-approximation on the optimal load.

Algorithm 2 FREQUENCYGREEDY

1. In each cell c_i, do the following:
2. Sort the nodes in R_{c_i} in non-increasing order of their in-degrees. Let this order be: $deg^{in}(v_1) \geq deg^{in}(v_2) \geq \ldots \geq deg^{in}(v_n)$
3. Starting from v_1, assign each successive node a frequency from $\{f_1, \ldots, f_K\}$ that has the *least* load on it so far, breaking ties arbitrarily.

LEMMA 3. *The algorithm* FREQUENCYGREEDY *in each cell c_i gives a $\left(\frac{4}{3} - \frac{1}{3K}\right)$-approximation on $L^{m^*}_{c_i}$.*

Proof: Consider a job scheduling problem with K identical machines $m_1, \ldots, m_K$, and n jobs $1, \ldots, n$. Executing a job j on any machine takes time $t_j > 0$. Thus, if $\Psi(k)$ denote the set of jobs assigned to machine m_k, then the total time m_k takes is $\sum_{j \in \Psi(k)} t_j$, and the makespan is $\max_{1 \leq k \leq K} \{\sum_{j \in \Psi(k)} t_j\}$. The objective is to find an assignment of the jobs to the machines that minimizes the makespan.

In our load-balanced frequency assignment formulation, we map each receiver $v_j \in R_{c_i}$ to job j, and the in-degree $deg^{in}(v_j)$ to t_j. Map each frequency f_k to machine m_k. The load on f_k is therefore equal to the total time m_k takes. Thus, minimizing the maximum load over all the frequencies is equivalent to minimizing the makespan over all the machines. Under this mapping, FREQUENCYGREEDY is identical to Graham's list scheduling algorithm according to the *Longest Processing Time* (LPT) [11] first, which achieves a $\left(\frac{4}{3} - \frac{1}{3K}\right)$-approximation on the minimum makespan. Therefore, the lemma follows. ■

2) Time Slot Assignment: Once the receivers in each cell c_i are assigned frequencies according to algorithm FREQUENCYGREEDY, we employ a greedy time slot assignment scheme for the whole network. The following lemmas follow from greedily scheduling a *maximal* number of edges in each time slot.

LEMMA 4. *Let γ_i denote the set of time slots needed to schedule all the edges in cell c_i. Then, the minimum schedule length Γ for the whole network is bounded by: $\Gamma \leq 4 \cdot \max_{c_i} \{|\gamma_i|\}, \forall \alpha \geq 2$.*

Proof: Since in a UDG the distance between any two adjacent nodes is at most one, two edges that belong to non-adjacent cells must have their transmitters at least two hops away from the receiver of the other, for any $\alpha \geq 2$. Therefore, two such edges can be scheduled on the same time slot regardless of their receiver frequencies, such as e_1 and e_4 in Fig 3. Thus, the set γ_i of time slots needed to schedule all the edges in c_i can be reused in any other cell c_j that is non-adjacent to c_i, for any $\alpha \geq 2$. This is equivalent to vertex coloring a graph in which each node represents a cell and an edge exists between any two nodes if the corresponding cells are adjacent to each other; the colors represent pair-wise disjoint sets of time slots.

Although vertex coloring an arbitrary graph is NP-hard [10], the particular color assignment to the cells shown in Fig. 3 gives an optimal assignment because of the regular grid structure. Since we need at most four different colors, at most four pair-wise disjoint sets of time slots are sufficient to schedule the whole network. ■

LEMMA 5. *If $L^\phi_{c_i}$ denote the load on the maximally loaded frequency in c_i under mapping $\phi : R_{c_i} \rightarrow \{f_1, \ldots, f_K\}$ achieved by algorithm* FREQUENCYGREEDY, *then any greedy time slot assignment can schedule all the edges in c_i within $2 \cdot L^\phi_{c_i}$ time slots.*

Proof: Consider a multi-graph $H = (\{f_1, \ldots, f_K\}, E')$, where for each edge $e = (v_i, v_{i'})$, $v_i, v_{i'} \in R_{c_i}$ with $\phi(v_i) \neq \phi(v_{i'})$, we have an edge $(\phi(v_i), \phi(v_{i'})) \in E'$. Note that these will be multi-edges; let $n(f_k, f_{k'})$ denote the number of edges between f_k and $f_{k'}$ in H. Then, $deg(f_k) \leq l^\phi_{c_i}(f_k)$, where $l^\phi_{c_i}(f_k)$ is the load on f_k under ϕ in c_i. By Ore's theorem [16], which generalizes Vizing's theorem for edge coloring on multi-graphs, it follows that the edges in H can be colored using $\max_{f_k}\{l^\phi_{c_i}(f_k)\}$ colors. Therefore, all edges of the form $e = (v_i, v_{i'})$ between two nodes in R_{c_i} with different frequencies can be colored in $\max_{f_k}\{l^\phi_{c_i}(f_k)\} = L^\phi_{c_i}$ colors.

All remaining edges either have only one end-point in R_{c_i}, or have both end-points in R_{c_i}, with the same frequency; let $S(f_k)$ denote the set of such edges with the end-point in R_{c_i}

assigned frequency f_k. Note that $|S(f_k)| \leq l^{\phi}_{c_i}(f_k)$, and edges $e \in S(f_k), e' \in S(f_{k'})$ can be assigned the same time slot if $f_k \neq f_{k'}$. So all the remaining edges can be scheduled in $\max_{f_k} |S(f_k)| \leq \max_{f_k}\{l^{\phi}_{c_i}(f_k)\}$ time slots. Therefore, all edges in c_i can be scheduled within $2 \cdot \max_{f_k}\{l^{\phi}_{c_i}(f_k)\} = 2 \cdot L^{\phi}_{c_i}$ time slots, and the lemma follows. ■

We now prove our key approximation result on the optimal schedule length.

THEOREM 4. *Given a routing tree T on a UDG G and K frequencies, there exists a greedy algorithm $\mathcal{G}$ that achieves a constant factor $8\mu_\alpha \cdot \left(\frac{4}{3} - \frac{1}{3K}\right)$-approximation on the optimal schedule length, where $\mu_\alpha > 0$ is a constant for a given cell size $\alpha \geq 2$.*

Proof: Algorithm $\mathcal{G}$ consists of two phases. In the first phase, we run FREQUENCYGREEDY to assign the K frequencies to the receivers in each cell. In the second phase, we greedily schedule a *maximal* number of edges on each time slot. Let the schedule length of $\mathcal{G}$ be $\Gamma_{\mathcal{G}}$, and that of an optimal algorithm be OPT.

Due to the presence of interfering links, there exists a constant $\mu_\alpha > 0$, depending on the cell size α and the deployment distribution, such that at most μ_α edges in any cell, whose receivers are on the same frequency, can be scheduled simultaneously by an optimal algorithm.

Now, regardless of the assignment chosen by an optimal strategy, it will take at least $L^{m^*}_{c_i}/\mu_\alpha$ time slots to schedule the edges in c_i, because $L^{m^*}_{c_i}$ is the *minimum* of the *maximum* number of edges that are on the same frequency in c_i. Thus, $OPT \geq L^{m^*}_{c_i}/\mu_\alpha, \forall c_i, \Rightarrow OPT \geq \max_{c_i}\{L^{m^*}_{c_i}\}/\mu_\alpha$; so

$$\max_{c_i}\{L^{m^*}_{c_i}\} \leq \mu_\alpha \cdot OPT \quad (4)$$

By running FREQUENCYGREEDY in c_i, Lemma 3 implies

$$L^{\phi}_{c_i} \leq \left(\frac{4}{3} - \frac{1}{3K}\right) \cdot L^{m^*}_{c_i} \quad (5)$$

and by scheduling a maximal number of edges in each time slot, Lemma 5 implies $|\gamma_i| \leq 2 \cdot L^{\phi}_{c_i}$. Then, from Lemma 4:

$$\begin{aligned} \Gamma_{\mathcal{G}} &\leq 4 \cdot \max_{c_i}\{|\gamma_i|\} \\ &\leq 8 \cdot \max_{c_i}\left\{L^{\phi}_{c_i}\right\} \\ &\leq 8 \cdot \max_{c_i}\left\{\left(\frac{4}{3} - \frac{1}{3K}\right) \cdot L^{m^*}_{c_i}\right\} \\ &\leq 8\mu_\alpha \cdot \left(\frac{4}{3} - \frac{1}{3K}\right) \cdot OPT \end{aligned}$$

■

B. Unknown Routing Topologies

In this subsection, we consider the scenario when the routing topology is not known to the algorithm designer a priori. Our goal is to find properties of a routing tree that could still guarantee a constant factor approximation on the optimal schedule length.

THEOREM 5. *Given a network modeled as a UDG and K frequencies, there exists an algorithm $\mathcal{H}$ that achieves a constant factor $8\mu_\alpha \cdot \Delta_C$-approximation on the optimal schedule length so long as the maximum in-degree of any node in the routing tree is bounded by a constant $\Delta_C > 0$, where $\mu_\alpha > 0$ is a constant for a given cell size $\alpha \geq 2$.*

Proof: Let V_{c_i} denote the set of nodes in cell c_i. We note that the set of receivers in the tree depends on the routing topology, but the total number of nodes V_{c_i} depends only on the graph. Because an optimal algorithm can simultaneously schedule at most a constant number $\mu_\alpha > 0$ of nodes (edges) in any c_i whose parents are on the same frequency, the best it could do with K frequencies is distributing the nodes in V_{c_i} *evenly* among all the frequencies so that $\lceil |V_{c_i}|/K \rceil$ is the *minimum* of the *maximum* number of nodes transmitting on the same frequency. Thus, $OPT \geq \lceil |V_{c_i}|/K \rceil/\mu_\alpha, \forall c_i, \Rightarrow OPT \geq \max_{c_i}\{\lceil |V_{c_i}|/K \rceil\}/\mu_\alpha$; so

$$\max_{c_i}\{\lceil |V_{c_i}|/K \rceil\} \leq \mu_\alpha \cdot OPT \quad (6)$$

Suppose $R_{c_i}(T) = \{v_1, \ldots, v_n\}$ denote the set of receivers in c_i on an arbitrary routing tree T, and suppose $\Delta^{in}(T)$ be the maximum in-degree of any node in T.

Define a *cyclic frequency assignment* under mapping $\psi : R_{c_i}(T) \rightarrow \{f_1, \ldots, f_K\}$ as follows:

$$\psi(v_i) = \begin{cases} i \bmod K, & \text{if } \; i \neq qK \\ K, & \text{if } \; i = qK \end{cases} \quad (7)$$

where $q \in \mathbb{N}^+$ is a positive integer. It is easy to see that the maximum number of receivers that are on the same frequency is $\lceil |R_{c_i}(T)|/K \rceil$. Therefore, the load $L^{\psi}_{c_i}$ on the maximally loaded frequency in c_i is bounded by:

$$\begin{aligned} L^{\psi}_{c_i} &\leq \lceil |R_{c_i}(T)|/K \rceil \cdot \max_{v_j \in R_{c_i}(T)}\left\{deg^{in}(v_j)\right\} \\ &\leq \lceil |V_{c_i}|/K \rceil \cdot \Delta^{in}(T) \end{aligned} \quad (8)$$

The load $L^{\phi}_{c_i}$ on the maximally loaded frequency produced by FREQUENCYGREEDY cannot be more than $L^{\psi}_{c_i}$; thus

$$L^{\phi}_{c_i} \leq L^{\psi}_{c_i} \leq \lceil |V_{c_i}|/K \rceil \cdot \Delta^{in}(T) \quad (9)$$

Then, scheduling a maximal number of edges in each time slot and using Lemma 4, Lemma 5 as before, and (9) it follows that:

$$\Gamma_{\mathcal{H}} \leq 8 \cdot \max_{c_i}\left\{\lceil |V_{c_i}|/K \rceil \cdot \Delta^{in}(T)\right\} \quad (10)$$

Since $|V_{c_i}|$ and $\Delta^{in}(T)$ are independent of each other, we can take the maximum separately on the two terms as:

$$\begin{aligned} \Gamma_{\mathcal{H}} &\leq 8 \cdot \max_{c_i}\{\lceil |V_{c_i}|/K \rceil\} \cdot \max_{c_i}\left\{\Delta^{in}(T)\right\} \\ &= 8 \cdot \max_{c_i}\{\lceil |V_{c_i}|/K \rceil\} \cdot \Delta^{in}(T) \\ &\leq 8\mu_\alpha \cdot \Delta^{in}(T) \cdot OPT \end{aligned} \quad (11)$$

Thus, (11) implies that so long as the maximum in-degree of a node in T is bounded by a constant $\Delta_C > 0$, the theorem holds. Although finding a degree-bounded spanning tree on a general graph is known to be NP-hard [10], for any UDG it is always possible to find a spanning tree of degree at most 5 [19]. Therefore, the theorem follows. ■

V. Assignment on General Disk Graphs

In this section, we relax our assumption of nodes having a uniform transmission range, and consider networks where nodes could transmit at different power levels resulting in different transmission ranges. Such networks could be modeled as general disk graphs, where a directed edge from node u to v exists if $d(u,v) \leq r(u)$, where $r(u)$ is the transmission range of u. We consider only those edges that are bidirectional, i.e., both $r(u)$ and $r(v)$ are greater than or equal to their Euclidean distance.

For an edge $e = (u,v)$, we use the convention that u is the transmitter and v is the receiver. The Euclidean length of e is denoted by $l(e)$. Define $I(e)$ as the set of edges that are either adjacent to e or form a secondary interfering link with e. Also, define $I_{\geq}(e) \subseteq I(e)$ as the subset of edges of $I(e)$ whose end points have "larger" disks than those of e, i.e.,

$$I_{\geq}(e) = \{e' = (u',v') : e' \in I(e) \text{ and } \max\{r(u'), r(v')\} \geq \max\{r(u), r(v)\}\}.$$

As before, we separate the two phases of frequency assignment and time slot assignment. Our approach in the frequency assignment phase is to assign the frequencies in such a way that minimizes the maximum number of edges that interferes with any given edge. Once this is done, we devise a time slot assignment strategy that optimizes the schedule length.

We formulate the frequency assignment subproblem as a 0-1 Integer Linear Program (ILP). Define the indicator variables X_{vk} for edge $e = (u,v)$ as follows:

$$X_{vk} = \begin{cases} 1, & \text{if } v \text{ is assigned frequency } f_k \\ 0, & \text{otherwise} \end{cases}$$

A frequency assignment is therefore a 0-1 assignment to the variables X_{vk}, $\forall e = (u,v) \in E_T$, $\forall f_k$. Furthermore, for an edge $e = (u,v)$ on frequency f_k, define Z_{ek} as the total number of edges in $I_{\geq}(e)$ that are also on frequency f_k. Thus,

$$Z_{ek} = \sum_{e'=(u',v') \in I_{\geq}(e)} X_{v'k} = \sum_{v'} n(e,v') X_{v'k},$$

where $n(e,v') = |\{e' = (u',v') \in I_{\geq}(e) : \ell(e') \geq \ell(e)\}|$. If we are given a frequency assignment $\vec{X}$, the following lemma (based on [21], [22]) shows how the schedule length is related to the Z_{ek}'s.

Lemma 6. *([21], [22]) Let X_{vk} and Z_{ek} be as defined above. Then, $\Omega(\max_{e,f_k}\{Z_{ek}\})$ is a lower bound on the length of any schedule for the edges. Also, it is possible to schedule all the edges using $\max_{e,f_k}\{Z_{ek}\}$ time slots.*

Proof: (sketch) The lower bound directly follows from [21], [22]. We briefly sketch a greedy scheduling algorithm (also from [21], [22]), which implies the upper bound. Let $E_T = \{e_1, e_2, \ldots, e_{n-1}\}$, with the edges numbered so that $\ell(e_1) \geq \ell(e_2) \ldots$. Our scheduling algorithm assigns a time slot $t(e_i)$ for each edge e_i in the following manner: consider the edges in the order $e_1, \ldots, e_{n-1}$, and for edge e_i, assign the smallest available time slot $t = t(e_i)$ so that (i) for each edge e_j with $j < i$ having the same receiver as e_i, we have $t(e_i) \neq t(e_j)$, and (ii) for each e_j such that $e_i, e_j \in E_T^k$ (i.e., they are assigned the same frequency), we have $t(e_i) \neq t(e_j)$. It can now be shown that the number of slots needed is at most $\max_{e,f_k}\{Z_{ek}\}$. ■

The above lemma implies that we should find a frequency assignment that minimizes $\max_{e,f_k}\{Z_{ek}\}$. We formulate this by the following ILP.

$$\textbf{Minimize} \quad \lambda$$

$$\begin{aligned} &\text{subject to} &:& \\ &\forall e = (u,v), \forall f_k &:& \sum_{v'} n(e,v') X_{vk} \leq \lambda \qquad (12) \\ &\forall e = (u,v) &:& \sum_{f_k} X_{vk} = 1, \qquad (13) \\ &\forall e = (u,v), \forall f_k &:& X_{vk} \in \{0,1\} \qquad (14) \end{aligned}$$

The second constraint guarantees that each of the receivers is assigned a single frequency. Since solving this ILP is NP-hard, we first find the solution to the linear programming (LP) relaxation of it, which is obtained by modifying the third constraint to include fractional values for the indicator variables as $X_{vk} \in [0,1]$. Let the optimum (fractional) values for the indicator variables X_{vk} obtained by solving the LP relaxation be X_{vk}^*, and let λ^* be the corresponding objective value. We now construct integral random variables Y_{vk} by rounding the fractional values X_{vk}^* in the following manner: for each v and f_k, we choose $Y_{vk} = 1$ with probability X_{ik}^*. this rounding is done in a mutually exlusive manner, so that for each v, there is exactly one frequency f_k with $Y_{vk} = 1$; this can be done since $\sum_k X_{vk}^* = 1$.

Lemma 7. *Let Y_{vk} be the rounded solution, as described above. Then,*

$$\max_{e=(u,v),f_k} \left\{ \sum_{e'=(u',v') \in I_{\geq}(e)} Y_{v'k} \right\} = O\left(\Delta(T) \log n \cdot \lambda^*\right),$$

with probability at least $1 - 1/n$.

Proof: Because of our randomized rounding strategy,

$$E[Y_{ik}] = Pr[Y_{ik} = 1] = X_{ik}^*$$

Let $\widehat{Z}_{ek} = \sum_{e'=(u',v') \in I_{\geq}(e)} Y_{v'k} = \sum_{v'} n(e,v') Y_{v'k}$. Therefore, by linearity of expectation, $E[\widehat{Z}_{ek}] = \sum_{v'} n(e,v') E[Y_{v'k}] = \sum_{v'} n(e,v') X_{vk}^* \leq \lambda^*$. Next, note that $n(e,v') \leq \Delta(T)$ for any e, v'. Therefore, by the weighted version of the Chernoff bound [23], it follows that for any edge e and frequency f_k, we have $Pr[\widehat{Z}_{ek} \geq \lambda^* \Delta(T) \log n] \leq 1/n^3$. Since the number of edges in T and the number of frequencies are both $O(n)$, we have $Pr[\max_{e,k} \widehat{Z}_{ek} \geq \lambda^* \Delta(T) \log n] \leq \sum_{e,k} Pr[\widehat{Z}_{ek} \geq \lambda^* \Delta(T) \log n]] \leq 1/n$, where the first inequality follows from the union bound. ■

The above rounded solution, along with the scheduling algorithm from Lemma 6, leads to the following approximation guarantee.

THEOREM 6. *The schedule constructed by Lemma 6, along with the frequency assignment using the above randomized rounding procedure results in a schedule of length* $O(\Delta(T) \log n)$ *times the optimum.*

VI. EVALUATION

In this section, we evaluate the performance of our algorithms through simulations on UDG. We generate connected networks by randomly placing nodes in a square region of maximum size 200×200 $unit^2$ and connecting any two nodes that are at most 25 units apart (scaled up for convenience).

A. Frequency Bounds

Fig. 4 compares the number of frequencies needed as a function of density to remove all the secondary interfering links on shortest path trees (SPT) as calculated from the upper bound, $\Delta(G_C) + 1$, and that from a *Largest Degree First* (LDF) assignment (where a node is assigned the first available frequency in non-increasing order of degrees). Here, the number of nodes N is fixed at 200 and the length l of the square region is varied from 200 to 20 so the density $d = N/l^2$ varies from 0.005 to 0.5.

The trend shows that the number of frequencies initially increases with density, reaches a peak, and then steadily goes down to one. This happens because of two opposing factors: as the density goes up, the parents link up with more and more new nodes, thus increasing the number of secondary interfering links; however, at the same time the number of parents (on the SPT) gradually decreases because the deployment region gets smaller in size. As we go right in the graph, the latter effect starts dominating until the network finally turns into a single hop network with the sink as the only parent. We also observe that for sparser networks there is a significant gap between the upper bound and the LDF scheme as compared to that in denser networks. This is because in sparser settings there are many parents, resulting in higher $\Delta(G_C)$, and assigning a distinct frequency to the largest degree parent according to LDF removes more interfering links at every step than it does for denser settings when the parents are fewer and have similar degrees.

B. Schedule Length

We evaluate the performance of our greedy algorithm $\mathcal{G}$ of Theorem 4 for $l = 200$ on shortest path trees and *Minimum Interference Trees* (MIT). We note that the constant factor approximations in our algorithms depend on the parameter μ_α, which decreases with decreasing α. However, the smallest α for which Lemma 4 holds is 2. Thus, in our experiments we chose $\alpha = 50$, which is again scaled up 25 times, as is the UDG. We do not present our evaluation results on general disk graphs due to lack of space.

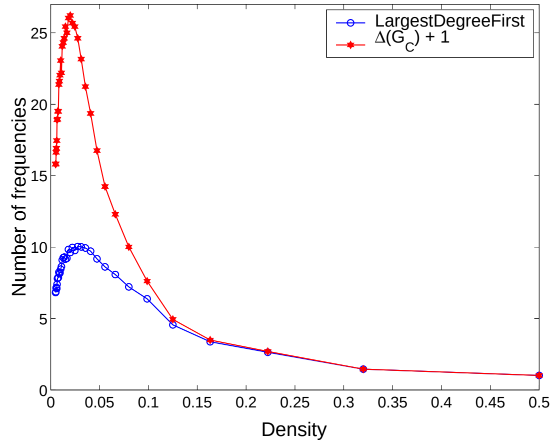


Fig. 4. Number of frequencies required to remove all the secondary interfering links as a function of network density for shortest path trees.

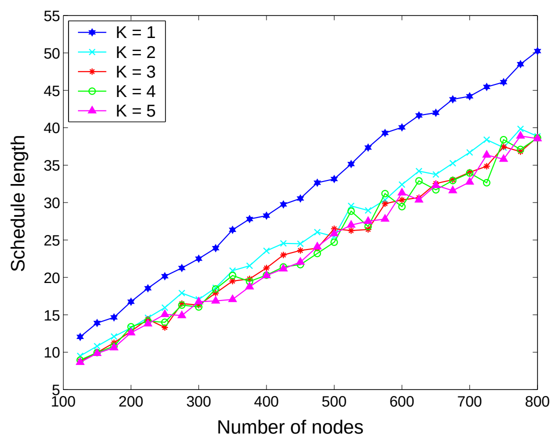


Fig. 5. Schedule length of the greedy algorithm $\mathcal{G}$ with different network sizes on shortest path trees; K is the number of frequencies.

1) Shortest Path Tree: Fig. 5 shows the schedule length of $\mathcal{G}$ with different number of nodes on SPT for different number of frequencies. We observe that multiple frequencies help in reducing the schedule length, and this effect dominates with increasing network size as the curve for single frequency and those for multiple frequencies diverge from each other. We also note that the schedule lengths with three or more frequencies do not differ much, implying that for larger networks the schedule length is constrained by the number of children any receiver has rather than the number of secondary interfering links.

2) Minimum Interference Tree: Since interference is one of the limiting factors in minimizing the schedule length, we study the performance of our approximation algorithms on interference-optimal trees. We use an existing greedy algorithm LIFE [4] to construct minimum interference spanning trees. LIFE uses a particular interference model, in which the *outgoing edge interference* $I_{out}(e)$ for an edge $e = (u, v)$ is defined as the number of nodes covered by the union of the two disks centered at u and v, each of radius $|uv|$, where $|uv|$ denotes the Euclidean distance between u and v. The interference $I_{out}(G)$ of a graph G is defined as the maximum

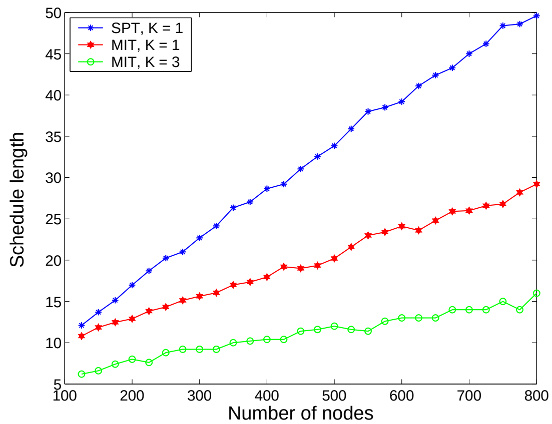


Fig. 6. Schedule length of the greedy algorithm $\mathcal{G}$ on SPT and MIT for different network sizes; K is the number of frequencies.

edge interference over all edges. The greedy strategy in LIFE is to construct a minimum spanning tree with the weight of an edge e equal to $I_{out}(e)$.

Fig. 6 shows the schedule length computed by algorithm $\mathcal{G}$ on SPT with one frequency, and on MIT with one and three frequencies, for different network sizes. As expected, we observe a significant reduction in the schedule length for larger networks on MIT. Comparing Fig. 5 and 6, we note that the curve for MIT with even one frequency is lower than those for SPT with multiple frequencies, implying that interference-optimal trees can also give benefits similar to multiple frequencies in terms of reducing the schedule length. The increasing gain in larger networks is due to smaller maximum node degree on MIT compared to that of SPT. For this particular plot with one frequency, the average maximum node degree on MIT is between 4 and 9, whereas on SPT it is between 8 and 34, with more than 20 beyond 450 nodes.

VII. Conclusions

We proved two NP-completeness results on the problem of minimizing the schedule length for aggregated convergecast and proposed algorithms that achieve constant factor approximation on UDG and $\Delta(T)\log n$ approximation on general disk graphs. We also evaluated some of our algorithms through simulations and showed various trends in performance for different network parameters. Even though we considered a graph-based network and interference model as opposed to the SINR-based model [15] as a first step in this paper, the results presented in [12] show that graph-based models provide a decent approximation to SINR-model behavior. Studying scheduling protocols utilizing multiple frequencies under the SINR model remains as part of our future work. From the simulation results, we observed that the schedule length improved significantly for minimum interference trees; however the trees are not guaranteed to be degree-bounded, which is a necessary condition for Theorem 5 to hold. Exploring the problem of constructing interference-optimal, degree-bounded trees is also part of our future work.

Acknowledgements. We would like to thank the anonymous referees and the members of the Autonomous Networks Research Group (ANRG) for their critical comments. The third author was supported partially by the following grants: DTRA CNIMS Grant HDTRA1-07-C-0113, NSF Nets Grant CNS-0626964, NSF HSD Grant SES-0729441, NSF NETS CNS-0831633, and NSF CNS-0845700.

References

[1] V. Annamalai, S. Gupta, and L. Schwiebert, "On Tree-Based Convergecasting in Wireless Sensor Networks", in *Proceedings of WCNC'03*, vol. 3, March 2003, pp. 1942-1947.

[2] P. Bahl, R. Chandra, J. Dunagan, "SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks", *Proceedings of Mobicom'04*, Philadelphia, pp. 216–230.

[3] R. Bhatia, M. Kodialam, "On Power Efficient Communication over Multi-hop Wireless Networks: Joint Routing, Scheduling and Power Control", *Proceedings of INFOCOM'04*, Hong Kong, pp. 1457–1466.

[4] M. Burkhart, P. von Rickenbach, R. Wattenhofer, and A. Zollinger, "Does Topology Control Reduce Interference?" in *Proceedings of MobiHoc'04*, Tokyo, Japan, pp. 9–19.

[5] H. Choi, J. Wang, and E.A. Hughes, "Scheduling on Sensor Hybrid Network", in *Proceedings of ICCCN'05*, San Diego, pp. 505–508.

[6] R.L. Cruz and A. Santhanam, "Optimal Routing, Link Scheduling, and Power Control in Multi-hop Wireless Networks", in *Proceedings of INFOCOM'03*, San Francisco, CA, pp. 702–711.

[7] T.A. ElBatt and A. Ephremides, "Joint Scheduling and Power Control for Wireless Ad-hoc Networks", *IEEE Transactions on Wireless Communications*, 3(1):54–85, 2004.

[8] R.J. Faudree, A. Gyrfs, R.H. Schelp, Z. Tuza, "The Strong Chromatic Index of Graphs", *Ars Combinatoria*, B29 (1990), pp. 205-211.

[9] S. Gandham, Y. Zhang, and Q. Huang, "Distributed Time-Optimal Scheduling for Convergecast in Wireless Sensor Networks", *Computer Networks*, 52(3):610–629, 2008.

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

[11] R.L. Graham, "Bounds on Multiprocessing Timing Anomalies", *SIAM Journal on Applied Mathematics*, 17(2):416–429, March 1969.

[12] O.D. Incel and Bhaskar Krishnamachari "Enhancing the Data Collection Rate of Tree-Based Aggregation in Wireless Sensor Networks", in *Proceedings of SECON'08*, San Francisco, CA, pp. 569–577.

[13] S. Madden, M.J. Franklin, J.M. Hellerstein, and W. Hong, "TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks", in *Proceedings of OSDI'02*, Boston, MA, USA, pp. 131–146.

[14] J. Mao, Z. Wu, and X. Xu, "A TDMA Scheduling Scheme for Many-to-One Communications in Wireless Sensor Networks", *Computer Communications*, 30(4):863–872, 2007.

[15] T. Moscibroda, "The Worst-Case Capacity of Wireless Sensor Networks", in *Proceedings of IPSN'07*, Cambridge, MA, pp. 1–10.

[16] O. Ore, The Four-Color Problem, *Academic Press*, 1967.

[17] M.R. Salavatipour, "A Polynomial Time Algorithm for Strong Edge Coloring of Partial k-Trees", *Discrete Applied Mathematics* 143:(1-3) 2004, pp 285–291.

[18] J. So and N. Vaidya, "Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using A Single Transceiver", in *Proceedings of MobiHoc'04*, Japan, pp. 222-233.

[19] W. Wu, H. Du, X. Jia, Y. Li, and S.C.-H. Huang, "Minimum Connected Dominating Sets and Maximal Independent Sets in Unit Disk Graphs", in *Theoretical Computer Science*, 352(1):1–7, 2006.

[20] G. Zhou, C. Huang, T. Yan, T. He, J. A. Stankovic, T. F. Abdelzaher, "MMSN: Multi-Frequency Media Access Control for Wireless Sensor Networks", in *Proceedings of INFOCOM'06*, Barcelona, pp. 1–13.

[21] V. S. Anil Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan, "End-to-End Packet-Scheduling in Wireless Ad-hoc Networks", in *Proceedings of SODA'04*, New Orleans, Louisiana, pp. 1021–1030.

[22] B. Han, V. S. Anil Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan, "Distributed Strategies for Channel Allocation and Scheduling in Software-Defined Radio Networks", *Proceedings of Infocom'09*, Rio de Janeiro, Brazil.

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