Algorithms for Fast Aggregated Convergecast in Sensor Networks

Amitabha Ghosh*, Özlem Durmaz Incel†, V.S. Anil Kumar‡, and Bhaskar Krishnamachari*
*Ming Hsieh Department of Electrical Engineering - Systems, University of Southern California
Los Angeles, CA 90089, USA, {amitabhg, bkrishna}@usc.edu
†Department of Computer Science, University of Twente, Enschede, 7522NB, the Netherlands, o.durmaz@cs.utwente.nl
‡Virginia Bio-Informatics Institute and Department of Computer Science, Virginia Tech
Blacksburg, VA 24061, USA, akumar@vbi.vt.edu
USC CENG Technical Report CENG-2008-8

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, referred to as convergecast, we focus on applications wherein data packets are aggregated at each hop *en-route* to the sink along a tree-based routing topology, and address the problem of minimizing the convergecast schedule length by utilizing multiple frequency channels. The primary hindrance in minimizing the schedule length is the presence of interfering links. We prove that it is NP-complete to determine whether all the interfering links in an arbitrary network can be removed using at most a constant number of frequencies. We give a sufficient condition on the number of frequencies for which all the interfering links can be removed, and propose a polynomial time algorithm that minimizes the schedule length in this case. We also prove that minimizing the schedule length for a given number of frequencies on an arbitrary network is NP-complete, and describe a greedy scheme that gives a constant factor approximation on unit disk graphs. When the routing tree is not given as an input to the problem, we prove that a constant factor approximation is still achievable for degree-bounded trees. Finally, we evaluate our algorithms through simulations and compare their performance under different network parameters.

I. Introduction

Convergecast in wireless sensor networks (WSN) typically refers to the many-to-one communication pattern, where data from a set of sources are routed toward a common sink. Often, many WSN applications [8], [14] require periodic summaries or aggregates of these data rather than raw sensor readings, in addition to quick delivery with minimum energy consumption. In such cases, data coming from different sources can be aggregated at each hop *en-route* to the sink - eliminating redundancy, minimizing the number of transmissions, and thereby saving energy and improving network throughput [17], [16]. In this paper, we consider the convergecast process where aggregated data are periodically streamed from a set of sources to a common sink over a tree-based routing topology, and refer to it as *aggregated convergecast* [15].

It is well known that contention-free medium access control (MAC) protocols like TDMA (Time Division Multiple Access) offer better solutions for such periodic data collection by eliminating collisions and retransmissions as opposed to contention-based protocols [18]. We therefore consider TDMA protocols where time slots are grouped into equal sized repeated frames. We call the number of time slots in each frame the *schedule length*, and assume that each node is scheduled to transmit in only one slot per frame, sending its own as well as aggregated data from its children. We also assume that the duration of each slot allows transmission for exactly one packet. Thus, once a *pipeline* is established, the sink will start receiving aggregated data from all the nodes in the network once in each frame. In this paper, we focus on the problem of minimizing the schedule length which, under this framework, is equivalent to maximizing the data collection rate at the sink.

A natural approach to avoid interference and increase throughput in wireless networks is to use multiple frequency channels. While there is a lot of research on single-channel scheduling protocol design for WSN, exploiting parallelism using multiple channels has not yet been well explored. Given the fact that current WSN hardware already provides multiple frequencies, such as the 16 orthogonal frequencies with 5MHz spacing supported by CC2420 [5] radios on TmoteSky [23], it is imperative to take their full advantage in order to minimize interference and collisions - the two most predominant causes of packet losses - and thereby achieve faster data collection rate by parallel transmissions. In this work, we thus exploit the benefits of utilizing multiple frequencies.

A. Related Work and Paper Overview

The non-aggregated version of the convergecast problem is considered by Gandham *et al.* in the presence of a single channel and TDMA protocols, where the goal is to minimize the schedule length [11]. 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 [6] 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 was studied by Bhatia *et al.* [3], and also by Bhat *et al.* [9]. A prominent recent work is by Moscibroda, in which scaling laws describing the achievable rate for aggregated convergecast in arbitrarily deployed sensor networks are presented under the SINR (signal-to-interference-plus-noise-ratio) model [19]. Worst-case capacity results are also proved by employing non-linear power assignment to nodes and exploiting SINR-effects. Cruz *et al.* use a duality approach to address the problem of finding an optimal link scheduling and power control policy, which minimizes the total average transmission power and support high data rates [7].

In the context of general ad hoc networks, the use of multiple channels has been well researched. To improve network throughput, So *et al.* propose a MAC protocol that switches channels dynamically and avoids the hidden terminal problem using temporal synchronization [22]. A link-layer protocol called SSCH is proposed by Bahl *et al.* that increases 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, is proposed by Zhou *et al.* where the goal is to increase aggregated throughput [25].

Most closely related is our previous work [15], 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. Our key contributions are the following:

1) We prove that it is NP-complete to determine whether all the interfering links in an arbitrary network can be removed using at most a constant number of frequencies.
2) We give a sufficient condition on the number of frequencies for which all the interfering links can be removed, and propose a polynomial time algorithm that minimizes the schedule length in this case.
3) For a given number of frequencies, we also prove that minimizing the schedule length on an arbitrary network is NP-complete, and describe a greedy scheme that achieves a constant factor approximation on the optimal schedule length for the special case of unit disk graphs.
4) We also consider the case when the routing tree is not given as an input to the problem, and prove that a constant factor approximation on the optimal schedule length is still achievable for degree-bounded trees.
5) Finally, we evaluate our algorithms through simulations and show various trends in performance for different network parameters.

The rest of the paper is organized as follows: Section II describes the problem formulation and assumptions. In Section III, we prove two complexity results on 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. In Section V, we consider aggregated convergecast on arbitrary trees. Section VI presents our evaluation results, and finally Section VII concludes the paper.

II. Preliminaries and Problem Formulation

We model the sensor network as an undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of edges that represent communication links. We assume G to be connected. A fixed node $s \in V$ is a given sink, and a spanning tree $T = (V, E_T \subseteq E)$ rooted at s is a given routing tree on the network. All the nodes except s are transmitters.

Definition 1. *Two edges e_1, $e_2 \in E_T$ form an interfering edge structure if the transmitter of either edge has an interfering link in G to the receiver of the other (see Fig. 1(a)).*

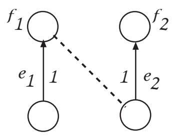
We assume each node has a single half-duplex transceiver, implying that it cannot receive multiple packets simultaneously, and cannot transmit and receive simultaneously. We also assume transmissions on orthogonal channels do not interfere with each other. Although this assumption may fail in practice depending on the adjacent/alternate channel rejection values for different types of transceivers, experimental results [15] presented by Incel *et al.* show that the scheduling performance remains similar for CC2420 and Nordic nrf905 radios.

The scheduling problem we address in this work is the following. Given a routing tree T on a graph G and K orthogonal frequencies $f_1, \ldots, f_K$, find an assignment of a frequency to each of the receivers and a time slot to each of the edges (i.e., transmitters) in T that minimizes the schedule length subject to the following constraints:

1) *Interfering Link Constraint*: Two edges forming an interfering edge structure cannot be scheduled simultaneously if their receivers are on the same frequency.
2) *Adjacent Edge Constraint*: No two adjacent edges in T can be scheduled simultaneously.

In our formulation we statically assign a frequency to each of the receivers. Although in practice, every sender-receiver pair could potentially negotiate on a particular frequency before each packet transmission, switching frequencies if necessary, we argue that assigning different frequencies to the transmitters that are children of the same parent does not help significantly 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, in our *receiver-based* frequency assignment strategy, the children of the same parent transmit on the parent's frequency, and therefore, a node in T operates on at most two frequencies.

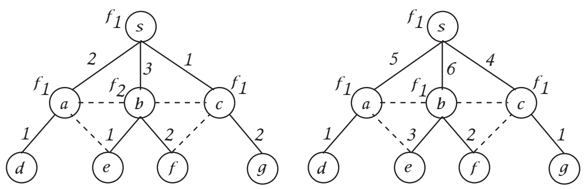
Fig. 1(c) illustrates aggregated convergecast with an example in a network of 7 source nodes and 2 frequencies. The dotted lines represent interfering links and the solid lines represent



(a)

Receiver	Frame 1 Slot 1	Frame 1 Slot 2	Frame 1 Slot 3	Frame 2 Slot 1	Frame 2 Slot 2	Frame 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) Interfering edge structure. (c) Aggregated convergecast with seven source nodes and two frequencies and (d) one frequency. (b) Pipeline with two frequencies starts from frame 2 with a minimum schedule length 3.

tree edges. A number beside an edge represents the time slot in which the edge is scheduled. The entries in Fig. 1(b) list the source nodes from which data is received on the corresponding time slot. For instance, s receives aggregated data from b, e, and f on the third time slot starting from frame 1. In this case, it takes two frames to reach a pipeline, as the 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 yielding the same schedule length. However, if we had only one frequency, the minimum schedule length would be 6, as shown in Fig. 1(d).

III. Assignment on General Graphs

A. Optimal Frequency Assignment

From the illustration above, we observe that when multiple frequencies are available, assigning different frequencies to the receivers in an appropriate way could mitigate the effects of interference and shorten the schedule length. In this subsection, we study the problem of finding the *minimum* number of frequencies to remove *all* the interfering link constraints. We say that an interfering link constraint is removed if the two receivers (i.e., parents) of an interfering edge structure are assigned different frequencies. In the following, we define the *Minimum Frequency Assignment Problem* and prove its hardness result.

Minimum Frequency Assignment Problem (MFAP): 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 interfering link constraints are removed?

Theorem 1. *The MFAP is NP-complete.*

Proof: It is easy to show that MFAP is in NP. Given a particular assignment, one can verify using a non-deterministic algorithm in polynomial time if at most q frequencies are being used, and if the receivers of every interfering edge structure are assigned different frequencies.

To show NP-hardness, we reduce an instance $G' = (V', E')$ of the vertex color problem to an instance $G = (V, E)$ of the MFAP. For every $v'_i \in V'$, create two nodes u_i and v_i in G, and join them with an edge $e_i = (u_i, v_i)$, treating u_i as the parent of v_i. For every edge $e'_{ij} = (v'_i, v'_j) \in E'$, create an interfering link in G between u_i and v_j. Finally, create a root node s, and add an edge $e_{is} = (u_i, s)$ from each u_i to s, treating s as the parent of u_i. This is an instance of the MFAP, where the tree given by $T = (V = \{u_i\} \cup \{v_i\} \cup \{s\}, E_T = \{e_i\} \cup \{e_{is}\})$. Clearly, the reduction runs in polynomial time.

Suppose G' is vertex colorable using at most q colors, and suppose v'_i is assigned color j. Assign frequency f_j to u_i in G, and any one of the frequencies, say f_1, to s. Clearly, this needs at most q frequencies. Since no two adjacent vertices v'_i and v'_j in G' are assigned the same color, no two nodes u_i and u_j in G, which are the receivers of an interfering edge structure, are assigned the same frequency, because by construction an interfering link exists between u_i and v_j whenever v'_i and v'_j are adjacent in G'. Therefore, this frequency assignment removes all the interfering link constraints.

Conversely, suppose there exists a solution to the MFAP using at most q frequencies. If u_i is assigned frequency f_j, assign color j to v'_i in G'. Clearly, this requires at most q colors because the number of receivers in G is one more than the number of vertices in G'. Since all the interfering link constraints are removed by such a frequency assignment, every two nodes u_i and u_j that are receivers of an interfering edge structure are assigned different frequencies. And since their corresponding vertices v'_i and v'_j are adjacent in G', they will be assigned different colors, thus, yielding a proper coloring of G'. Therefore, the theorem follows. ■

Theorem 1 implies that finding the *minimum* number of frequencies which will remove all the interfering link constraints on an arbitrary graph is NP-hard. In the following, we give an upper bound on the number of such frequencies required.

Lemma 1. *Construct a constraint graph $G_C = (V_C, E_C)$ from the original graph $G = (V, E)$ as follows. For each receiver v_i in G, create a vertex u_i in G_C. Create an edge between two such vertices u_i and u_j if their corresponding receivers are part of an interfering edge structure. Then, the number K_{max} of frequencies that will remove all the interfering link constraints 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 vertices in G_C whenever their corresponding receivers in G are part of an interfering edge structure, assigning different frequencies to every such receiver-pair in G is equivalent to assigning different colors to adjacent vertices 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. ■

We describe a simple scheme called LargestDegreeFirst (LDF) in Algorithm 1, in which the receiver with the maximum degree in G_C is assigned the first available frequency at every step. In Section VI, we compare the upper bound of Lemma 1 with LDF.

Algorithm 1 LARGEST DEGREE FIRST

1. Input: Constraint graph $G_C = (V_C, E_C)$
2. **while** $V_C \neq \phi$ **do**
3. $u \leftarrow$ vertex with maximum degree in V_C
4. Assign the first available frequency to u that is different from u's neighbors
5. $V_C \leftarrow V_C \setminus \{u\}$
6. **end while**

Algorithm 2 BFS-TIMESLOTASSIGNMENT

1. Input: $T = (V, E_T)$
2. **while** $E_T \neq \phi$ **do**
3. $e \leftarrow$ next edge from E_T in BFS order
4. Assign minimum time slot to e respecting adjacency constraint
5. $E_T \leftarrow E_T \setminus \{e\}$
6. **end while**

Once all the interfering link constraints are removed, the problem of minimizing the schedule length on the graph G reduces to one on the tree T. The remaining constraint that still prevents simultaneous transmissions is the adjacent edge constraint, which cannot be removed by using multiple frequencies. We propose an algorithm BFS-TIMESLOTASSIGNMENT in Algorithm 2 that runs in $O(|E_T|^2)$ time and minimizes the schedule length on a tree.

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 equal to the maximum degree $\Delta(T)$ of T.

THEOREM 2. *The 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 $T^i = (V^i, E^i_T)$ denote the subtree of T in the i^{th} iteration constructed in the BFS order, where E^i_T comprises all the edges that are assigned a slot, and V^i comprises the set of nodes on which the edges in E^i_T are incident. Note that, $|E^i_T| = i$, 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 of T is equivalent to edge coloring T, which requires at least $\Delta(T)$ colors, the schedule length is minimum. ■

B. Scheduling with Constant Number of Frequencies

We showed that when a sufficient number of frequencies is available, all the interfering link constraints can be removed and a minimum schedule length can be found in polynomial time. However, typically there is a limitation on the number of frequencies over which a given transceiver can operate. We now study the problem of minimizing the schedule length on an arbitrary graph when a constant number of frequencies is available. 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 Theorem 3.

DEFINITION 2. *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 if they are both 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 [12]. 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 [10]:*

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

Multiple-Frequency Minimum Time Scheduling Problem (MFMTSP): Given a tree T on an arbitrary graph G, an integer p, and a constant number of frequencies q, is there an assignment of frequencies to the receivers in T using at most q frequencies, and an assignment of time slots to the edges in T, such that the schedule length is at most p?

THEOREM 3. *The MFMTSP is NP-complete.*

Proof: It is easy to show that the MFMTSP is in NP. Given a particular assignment, one can use a non-deterministic algorithm to verify in polynomial time that - (i) at most q frequencies and p time slots are used, (ii) either the receivers of every interfering edge structure are assigned different frequencies or their edges are on different time slots, and (iii) all adjacent edges are on different time slots.

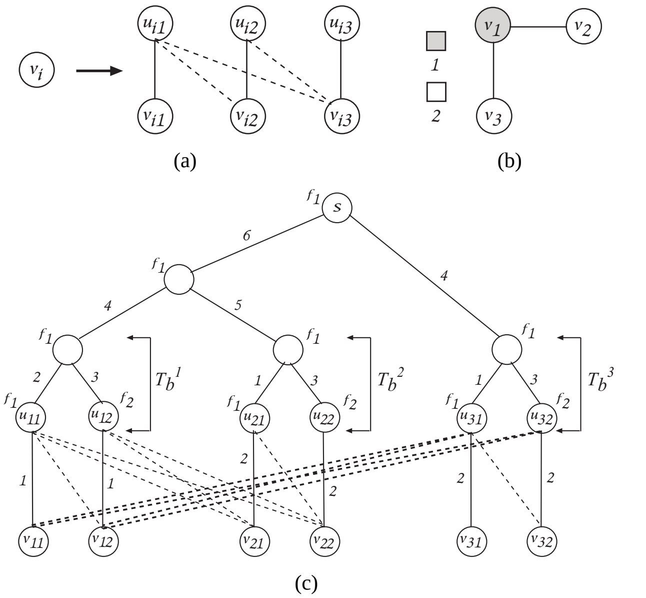


Fig. 2. Reduction for the MFMTSP: (a) Gadget for each v_i in G' for $q = 3$; (b) Instance G' of the vertex color problem; (c) Instance G of the MFMTSP as constructed from G' for $q = 2$.

To show NP-hardness, we reduce an instance $G' = (V', E')$ of the vertex color problem to an instance $G = (V, E)$ of the MFMTSP, as illustrated with an example in Fig. 2. 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 $\binom{q}{2} = q(q-1)/2$ interfering links between all such pairs in each S_i as follows. Consider each u_{is} in turn, for $s = 1, ..., q-1$, and create an interfering link from u_{is} to v_{il}, for all $l > s$. Thus, every two edges in S_i form an interfering edge structure. Next, for every edge $e_{ij} = (v_i, v_j) \in E'$, create q^2 interfering links (and hence, q^2 interfering edge structures) 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 interfering link 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 $\{u_{is}\}$ 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 MFMTSP. 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.

Suppose G' is vertex colorable using at most p colors, and v_i is assigned color t. 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. Then, assign time slot t 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 sot to all the edges in S_i does not violate the interfering link constraint within each S_i. Also, since only non-adjacent vertices in G' may have the same color, two sets of edges S_i and S_j that are on the same time slot cannot have interfering links between each other, because interfering links exist between S_i and S_j whenever v_i and v_j are adjacent in G'. 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 [21], 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 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 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 be part of an interfering edge structure, 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 aggregated convergecast in networks that are modeled as *unit disk graphs* (UDG) on the Euclidean plane and prove constant factor approximation results on the optimal schedule length.

We divide the area covering all the nodes into a set of equal sized grid cells $\{c_i\}$, each of size $\alpha \times \alpha$, as illustrated in Fig. 3. Under a UDG model, there exists an edge between every two nodes that are at most a unit distance apart from each other.

Definition 3. *Two cells are adjacent to each other if they share a common edge or a common grid point.*

Definition 4. *An edge e_k is in 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 thus we can reuse time slots across cells that are spatially well separated.

A. Time Slot Assignment on Unit Disk Graphs

We begin this subsection with an upper bound on the minimum schedule length. Let γ_{c_i} denote the *set* of time slots

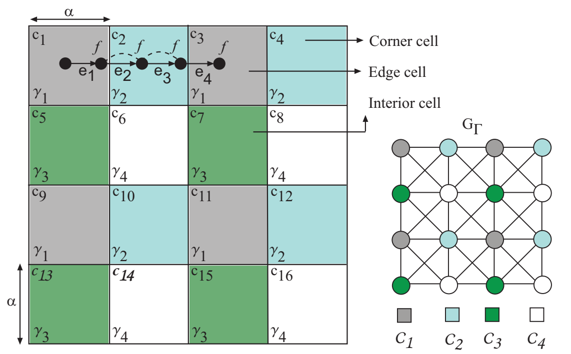


Fig. 3. Four pair-wise disjoint sets of time slots γ_1, γ_2, γ_3, γ_4 schedule the whole network. Each set γ_j maps to a distinct color C_j, for $j = 1, 2, 3, 4$.

needed to schedule all the edges in c_i.

Lemma 3. *The minimum schedule length Γ for the whole network is bounded by: $\Gamma \leq 4 \cdot \max\limits_{c_i} \{|\gamma_{c_i}|\}, \forall \alpha \geq 2$*

Proof: Since G is a UDG, the distance between any two adjacent nodes is at most one, and thus two edges that are in 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 on frequency f. Thus, the set γ_{c_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 $\alpha \geq 2$.

Construct a graph $G_\Gamma = (V_\Gamma, E_\Gamma)$, whose each vertex $v_i \in V_\Gamma$ corresponds to a cell c_i, and an edge exists between any two vertices v_i and v_j if their corresponding cells c_i and c_j are adjacent to each other, as shown in Fig. 3. The minimum number of colors needed to vertex color G_Γ is thus equal to the minimum number of pair-wise disjoint sets γ_{c_i}'s needed to schedule all the links in G. Now, although vertex coloring on a general graph is NP-complete [12], because of the regular grid structure here, we can vertex color G_Γ using at most four colors, C_1, C_2, C_3, C_4, as shown by a particular assignment in Fig. 3. The coloring pattern used is as follows: Number the vertices starting from 1 in each row of G_Γ; then (i) assign C_1 to every odd vertex and C_2 to every even vertex in the odd numbered rows, and (ii) assign C_3 to every odd vertex and C_4 to every even vertex in the even numbered rows. The corresponding assignment of the four sets of time slots, γ_1, γ_2, γ_3, γ_4, are shown within the cells. Note that no two γ_j's have a slot in common, and $|\gamma_j| \leq \max_{c_i}\{|\gamma_{c_i}|\}$, for $j = 1, 2, 3, 4$. Therefore, $\Gamma = |\gamma_1 \cup \gamma_2 \cup \gamma_3 \cup \gamma_4| = |\gamma_1| + |\gamma_2| + |\gamma_3| + |\gamma_4| \leq 4 \cdot \max\limits_{c_i}\{|\gamma_{c_i}|\}$. ■

B. Frequency Assignment on Unit Disk Graphs

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

Definition 5. *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 f_k in c_i under m as the total number of children of all 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, as shown in Lemma 4, to scheduling jobs on identical machines to minimize the *makespan* (last finishing time of the given jobs), which is known to be NP-hard [13]. Below, we describe an algorithm FrequencyGreedy in Algorithm 3 that achieves a $(4/3 - 1/3K)$-approximation on $L^{m^*}_{c_i}$.

Algorithm 3 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 4. *The algorithm* FrequencyGreedy *in each cell c_i gives a $(4/3 - 1/3K)$-approximation on $L^{m^*}_{c_i}$.*

Proof: In the job scheduling problem, there are 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 the load-balanced frequency assignment formulation, map each receiver $v_j \in R_{c_i}$ to job j, and $deg^{in}(v_j)$ 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-first* (LPT) [13], which achieves a $(4/3 - 1/3K)$-approximation on the minimum makespan. Therefore, the lemma follows. ■

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* FrequencyGreedy, *then any greedy time slot assignment can schedule all the edges in c_i within $2L^\phi_{c_i}$ time slots, i.e., $|\gamma_{c_i}| \leq 2L^\phi_{c_i}$.*

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 [20], 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 $2\max_{f_k}\{l^{\phi}_{c_i}(f_k)\} = 2L^{\phi}_{c_i}$ time slots, and the lemma follows. ■

We now prove a constant factor approximation result on the optimal schedule length.

THEOREM 4. *Given a tree T on a UDG G, and K frequencies, there exists a greedy algorithm $\mathcal{G}$ that achieves a constant factor $8\mu_\alpha(4/3 - 1/3K)$-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: (i) run FREQUENCYGREEDY in each c_i, and (ii) run *any* greedy time slot assignment scheme for the whole network. One possible scheme is to greedily schedule a *maximal* number of edges simultaneously at each iteration.

Let the schedule length of $\mathcal{G}$ be $\Gamma_{\mathcal{G}}$, and that of an optimal algorithm OPT be Γ_{OPT}. We seek a lower bound on Γ_{OPT}.

Because of UDG, the tree edges are of length at most one, and thus for a given cell size α, at most a constant number of them can fit within any cell c_i. Moreover, because of interfering links, there exists a constant $\mu_\alpha > 0$, depending on α and the deployment distribution, such that at most μ_α edges in *any* c_i whose receivers are on the same frequency can be scheduled simultaneously by OPT.

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

$$\max_{c_i}\{L^{m^*}_{c_i}\} \leq \mu_\alpha \cdot \Gamma_{OPT} \tag{4}$$

By running FREQUENCYGREEDY in c_i, Lemma 4 implies

$$L^{\phi}_{c_i} \leq (4/3 - 1/3K) \cdot L^{m^*}_{c_i} \tag{5}$$

and by running any greedy time slot assignment scheme in the whole network, Lemma 5 implies:

$$|\gamma_{c_i}| \leq 2L^{\phi}_{c_i} \tag{6}$$

Then, from Lemma 3 and (6) it follows that the number of time slots needed to schedule the entire network by $\mathcal{G}$ is:

$$\begin{aligned}\Gamma_{\mathcal{G}} &\leq 4 \cdot \max_{c_i}\{|\gamma_{c_i}|\} \leq 8 \cdot \max_{c_i}\left\{L^{\phi}_{c_i}\right\} \\ &\leq 8 \cdot \max_{c_i}\left\{(4/3 - 1/3K) \cdot L^{m^*}_{c_i}\right\} \\ &\leq 8\mu_\alpha (4/3 - 1/3K) \cdot \Gamma_{OPT} \end{aligned} \tag{7}$$

Therefore, the theorem follows. ■

V. ASSIGNMENT ON ARBITRARY TREES UNDER UDG

In our discussion so far, we assumed that the routing tree T on G was given as an input to the problem. We now consider the case when it is not (sink s is still given), thus implying that OPT can construct any *arbitrary* tree T rooted at s to minimize the schedule length. In this section, we incorporate the construction of T as part of the greedy algorithm, and seek for properties of T that would still guarantee a constant factor approximation on the optimal schedule length in UDG.

THEOREM 5. *Given a UDG G and K frequencies, there exists an algorithm $\mathcal{H}$ that achieves a constant factor $8\mu_\alpha\Delta_C$-approximation on the optimal schedule length, where $\mu_\alpha > 0$ is a constant for a given cell size $\alpha \geq 2$, and $\Delta_C > 0$ is a constant.*

Proof: Since OPT can construct any arbitrary tree T on G, we seek for a lower bound on Γ_{OPT} independent of T.

Let V_{c_i} denote the set of nodes in c_i. Note that V_{c_i} is independent of T, and depends only on G. Because OPT can schedule simultaneously at most a constant number $\mu_\alpha > 0$ of nodes (i.e., edges) in *any* c_i whose parents are on the same frequency, the best it could do with K frequencies is to distribute 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, $\Gamma_{OPT} \geq \lceil |V_{c_i}|/K \rceil/\mu_\alpha, \forall c_i, \Rightarrow \Gamma_{OPT} \geq \max_{c_i}\{\lceil |V_{c_i}|/K \rceil\}/\mu_\alpha$; so

$$\max_{c_i}\{\lceil |V_{c_i}|/K \rceil\} \leq \mu_\alpha \cdot \Gamma_{OPT} \tag{8}$$

Suppose $R_{c_i}(T) = \{v_1, \ldots, v_n\}$ denote the set of receivers in c_i for any arbitrary tree T, and suppose $\Delta^{in}(T)$ be the maximum in-degree of a node in T. Then, $\max_{v_j \in R_{c_i}(T)}\{deg^{in}(v_j)\} \leq \Delta^{in}(T)$, and $|R_{c_i}(T)| \leq |V_{c_i}|$.

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

$$\psi(v_i) = \begin{cases} i \bmod K, & \text{if} \quad i \neq qK \\ K, & \text{if} \quad i = qK \end{cases} \tag{9}$$

where $q \in \mathbb{N}^+$, 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 the following:

$$\begin{aligned} L^{\psi}_{c_i} &\leq \lceil |R_{c_i}(T)|/K \rceil \cdot \max_{v_j \in R_{c_i}(T)}\left\{deg^{in}(v_j)\right\} \\ &\leq \lceil |V_{c_i}|/K \rceil \cdot \Delta^{in}(T) \end{aligned} \tag{10}$$

Now, the load $L^{\phi}_{c_i}$ on the maximally loaded frequency produced by FREQUENCYGREEDY is no more than $L^{\psi}_{c_i}$; thus

$$L^{\phi}_{c_i} \le L^{\psi}_{c_i} \le \lceil |V_{c_i}|/K \rceil \cdot \Delta^{in}(T) \tag{11}$$

Then, doing any greedy time slot assignment and using Lemma 3 and Lemma 5 as before, and (11) it follows that:

$$\Gamma_{\mathcal{H}} \le 8 \cdot \max_{c_i} \left\{ \lceil |V_{c_i}|/K \rceil \cdot \Delta^{in}(T) \right\} \tag{12}$$

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

$$\begin{aligned} \Gamma_{\mathcal{H}} &\le 8 \cdot \max_{c_i} \left\{ \lceil |V_{c_i}|/K \rceil \right\} \cdot \max_{c_i} \left\{ \Delta^{in}(T) \right\} \\ &= 8 \cdot \max_{c_i} \left\{ \lceil |V_{c_i}|/K \rceil \right\} \cdot \Delta^{in}(T) \\ &\le 8\mu_\alpha \Delta^{in}(T) \cdot \Gamma_{OPT} \end{aligned} \tag{13}$$

Thus, (13) 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 [12], for any UDG it is always possible to find a spanning tree of degree at most 5 [24]. Therefore, the theorem follows. ■

VI. EVALUATION

In this section, we evaluate the performance of our algorithms through simulations on UDG. We construct connected networks by randomly placing nodes on a square region of maximum size $200 \times 200\ unit^2$ and connecting any two nodes that are at most 25 units from each other. Note that we scale up the UDG by a factor of 25 just for convenience. We assume that the interference range for each node is also 25 units.

A. Frequency Bounds

Fig. 4 shows the number the frequencies needed as a function of density to remove all the interfering links as calculated from algorithm LARGESTDEGREEFIRST (LDF) and from the upper bound $\Delta(G_C)+1$ on the constraint graph G_C, for given shortest path trees. Here, we keep the number of nodes N fixed at 200 and vary the length l of the square region 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 because of increasing interference. However, as the network gets denser, it reaches a peak and then steadily decreases to 1, because the number of parents on the tree becomes fewer and the network gradually 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.

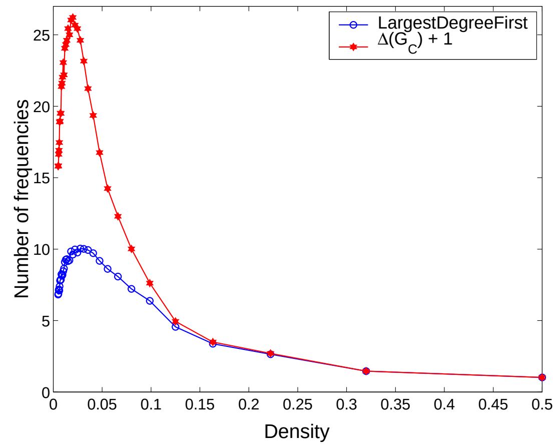


Fig. 4. Number of frequencies required to remove all the interfering links as a function of network density for shortest path trees.

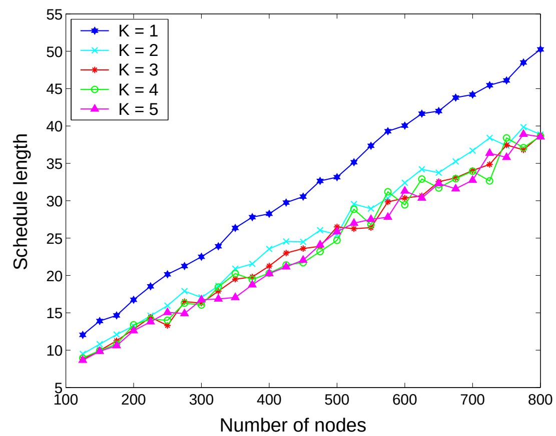


Fig. 5. Schedule length of the greedy algorithm $\mathcal{G}$ with different network sizes on shortest path trees; K is the number of frequencies.

B. Schedule Length

We evaluate the performance of our greedy algorithm $\mathcal{G}$ of Theorem 4 for $l = 200$ on two types of trees: (i) shortest path trees (SPT), and (ii) 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 3 holds is 2. Thus, in our experiments we chose $\alpha = 50$, which is again scaled up 25 times, as is the UDG.

1) Shortest Path Tree: Fig. 5 shows the schedule length of the greedy algorithm $\mathcal{G}$ with different number of nodes on shortest path trees. The different curves are for different number of frequencies. We observe that multiple frequencies help in reducing the schedule length, and this reduction increases with increasing network size, as the curve for single frequency and those for multiple frequencies diverge from each other. We also notice that the schedule lengths with three or more frequencies do not differ much, implying that interference is mostly eliminated with three frequencies and so having more frequencies is redundant.

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 algo-

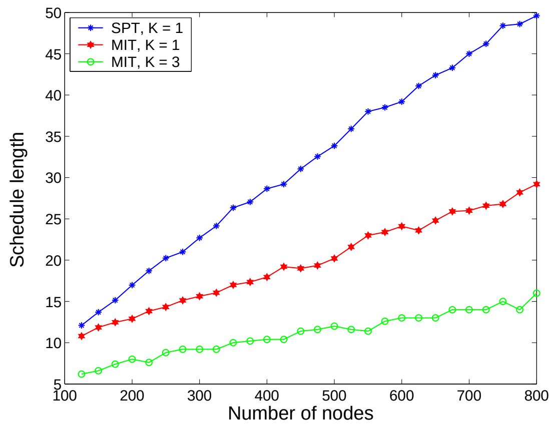


Fig. 6. Schedule length of the greedy algorithm $\mathcal{G}$ on SPT and MIT for different network sizes; K is the number of frequencies.

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

Fig. 6 shows the schedule length computed by algorithm $\mathcal{G}$ on SPTs with one frequency, and on MITs with one and three frequencies, for different network sizes. As expected, we observe a significant reduction in the schedule length for larger networks on MITs. Comparing Fig. 5 and Fig. 6, we notice 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 of aggregated convergecast in sensor networks and proposed algorithms that achieve constant factor approximations on unit disk graphs. We also evaluated our algorithms through simulations and showed various trends in performance for different network parameters. Even though we considered protocol/graph-based network and interference models as opposed to physical/SINR-based models [19] as a first step in this paper, the results presented in [15] show that graph-based models provide a decent approximation to SINR-model behavior. Studying scheduling protocols utilizing multiple frequencies under SINR-based models remain 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.

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

[3] R. Bhatia and M. Kodialam, "On Power Efficient Communication over Multi-hop Wireless Networks: Joint Routing, Scheduling and Power Control", in *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] CC2420 single-ship, 2.4GHz, IEEE 802.15.4 compliant radios, http://focus.ti.com/lit/ds/symlink/cc2420.pdf

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

[7] 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.

[8] D.M. Doolin and N. Sitar, "Wireless Sensors for Wildfire Monitoring", in *Proceedings of Smart Structures and Materials, SPIE'05*, 5765(1):477–484, March 2005.

[9] 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.

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

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

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

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

[14] T. He, P. Vicaire, T. Yan, L. Luo, L. Gu, G. Zhou, R. Stoleru, Q. Cao, J.A. Stankovic, and T. Abdelzaher, "Achieving Real-Time Target Tracking Using Wireless Sensor Networks", in *RTAS'06*, San Jose, pp. 37–48.

[15] 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.

[16] B. Krishnamachari, D. Estrin, and S. Wicker, "The Impact of Data Aggregation in Wireless Sensor Networks", in *Proceedings of International Workshop on Distributed Event-Based Systems, DEBS'02*, Vienna.

[17] 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.

[18] 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.

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

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

[21] 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.

[22] 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*, Roppongi, Japan, pp. 222-233.

[23] Tmote Sky, IEEE 802.15.4 compliant device for wireless mesh networking, http://www.sentilla.com/pdf/eol/tmote-sky-brochure.pdf

[24] 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.

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