Multi-Channel Scheduling for Fast Convergecast in Wireless Sensor Networks

Özlem Durmaz Incel*, Amitabha Ghosh†, Bhaskar Krishnamachari† and Krishna Kant Chintalapudi‡
*Department of Computer Science, University of Twente
Enschede, 7522NB, the Netherlands, o.durmaz@cs.utwente.nl
†Department of Electrical Engineering - Systems, University of Southern California
Los Angeles, CA 90089, USA, {amitabhg, bkrishna}@usc.edu
‡Research & Technology Center, Robert Bosch GmbH
Palo Alto, CA, USA, Krishna.Chintalapudi@us.bosch.com
USC CENG Technical Report CENG-2008-9

***Abstract*— We explore the following fundamental question - how fast can information be collected from a wireless sensor network? We consider a number of design parameters such as, power control, time and frequency scheduling, and routing. There are essentially two factors that hinder efficient data collection - interference and the half-duplex single-transceiver radios. We show that while power control helps in reducing the number of transmission slots to complete a convergecast under a single frequency channel, scheduling transmissions on different frequency channels is more efficient in mitigating the effects of interference (empirically, 6 channels suffice for most 100-node networks). With these observations, we define a receiver-based channel assignment problem, and prove it to be NP-complete on general graphs. We then introduce a greedy channel assignment algorithm that efficiently eliminates interference, and compare its performance with other existing schemes via simulations. Once the interference is completely eliminated, we show that with half-duplex single-transceiver radios the achievable schedule length is lower-bounded by $\max(2n_k - 1, N)$, where n_k is the maximum number of nodes on any subtree and N is the number of nodes in the network. We modify an existing distributed time slot assignment algorithm to achieve this bound when a suitable balanced routing scheme is employed. Through extensive simulations, we demonstrate that convergecast can be completed within up to 50% less time slots, in 100-node networks, using multiple channels as compared to that with single-channel communication. Finally, we also demonstrate further improvements that are possible when the sink is equipped with multiple transceivers or when there are multiple sinks to collect data.**

I. Introduction

Convergecast, namely collection of data from sensors towards a common sink node over a tree topology is a fundamental operation in wireless sensor networks (WSN) [1]. In many applications, it is important to deliver the data to the sink in a limited amount of time and increase the speed of data collection at which the sink can receive data from the network. For instance in Lites [2], which is a real time monitoring application, a typical event may generate up to 100 packets within a few seconds and the packets need to be transported from different network locations to a sink node.

Since the data has to be delivered in a short time, we consider time division multiple access (TDMA) [3] as a natural solution due to the collision free behavior. Consider a schedule of t time slots where the sink receives data from all nodes in the network once every t slots. In such a context, the objective is to minimize t to increase the speed of data collection.

We study a set of techniques in order to solve the fundamental problem: "how fast can data be convergecast to the sink over a tree topology?" The fundamental limiting factors are interference and half-duplex nature of the transceivers on WSN nodes. To cope with interference we consider different techniques such as transmission power control and assigning different frequency channels on interfering links. We show that once multiple frequencies are employed with spatial-reuse TDMA, the convergecast schedule becomes limited by the number of nodes in the network once a suitable routing tree is used. For further improvements, we consider equipping a single sink with multiple transceivers, and also the deployment of multiple sinks to collect data.

We evaluate the above mentioned techniques using mathematical analysis and simulations that use realistic channel models and radio parameters typical of WSN radio devices. The following are some of the findings and key contributions of this work:

- Evaluation of transmission power control to eliminate interference: Under idealized settings (unlimited power, continuous range) power control mechanisms may provide unbounded improvements in the speed of data collection. We evaluate the behavior with an optimal power control algorithm described in [4] in a practical setting considering the limited discrete power levels available in today's radios on WSN nodes.
- Receiver-based frequency assignment: We show that scheduling transmissions on different frequency channels is more efficient in mitigating the effects of interference compared with transmission power control. Accordingly, we define a receiver-based channel assignment problem which is "the problem of assigning a minimum number of frequencies to the receivers such that all the interference links in an arbitrary network is removed". We show that the problem is NP-complete and introduce a greedy heuristic for channel assignment. By simulations and analytical calculations, we evaluate the behavior of our heuristic algorithm and compare its performance with another channel assignment method which was recently

proposed for WSN with tree topologies [5].

- Bounds on convergecast scheduling: We show that, once the interference is eliminated, the achievable schedule length with half-duplex transceivers is bounded by $\max(2n_k-1, N)$ slots where n_k is the maximum number of nodes on any branch of the tree and N is the number of nodes. We modify an existing time slot assignment algorithm and show that the algorithm requires exactly $\max(2n_k-1, N)$ slots to schedule a given network.
- Impact of Routing Trees: According to the bound on convergecast schedules, the branches of a routing tree should have balanced number of nodes such that $2n_k - 1 < N$. Such a tree construction is defined as the "Capacitated Minimal Spanning Tree Problem" and is proved to be NP-complete in [6]. Given the hardness of the problem, we propose a heuristic algorithm and evaluate the impact of such routing trees on the schedule length by simulations.
- Multiple transceivers at the sink node: For further improvements we consider the sink having multiple transceivers and multiple sinks deployed in the network. We observe improved reductions on the schedule length that are proportional with the number of available transceivers.

The remainder of the paper is organized as follows: in Section II, we introduce the problem. In Section III we explain the mechanisms that we use to eliminate interference. In Section IV, we introduce a receiver-based greedy channel assignment algorithm. In Section V, we provide the bounds on the convergecast schedule when interference is eliminated and present a modified time slot assignment algorithm that achieves the lower bound. In Section VI, we discuss the impact of routing trees on the generated schedules. Section VII gives the detailed simulation based evaluation of the discussed methods. Section VIII summarizes some of the related work. Finally, Section IX provides the concluding remarks.

II. Preliminaries and Problem Statement

Before explaining the studied mechanisms, we first describe our preliminary design and give the details of the problem formulation. We assume time is divided into equal sized slots and each node is assigned a time slot to transmit data. All the nodes in the network except the sink are sources and generate one packet for each convergecast operation.

We model the sensor network as a graph $G = (V, E)$, where V is the set of nodes and E is the set of edges that represent communication links and interference links between nodes. A pair of nodes $v_i \in V$ and $v_j \in V$ form a communication link (i, j) if the signal to noise ratio (SNR) is not less than a communication threshold γ_C. A pair of nodes $v_i \in V$ and $v_j \in V$ form an interference link (i, j) if the transmission from node v_i disturbs a reception at the node v_j or vice versa, as illustrated in Fig. 1.

Let $s \in V$ be the sink node and $T = (V, E_T) \subset G$ be a spanning tree on the graph rooted at s. We assume G to be connected. The problem we address is the following. Given G, find an assignment of time slots to the transmitters such that the the number of time slots to complete a convergecast is minimized with subject to the following transmission constraints.

- Two adjacent edges (see Fig. 1) cannot be scheduled at the same time slot. An edge (k, l) is adjacent to edge (i, j) if $\{i, j\} \bigcap \{k, l\} \neq \phi$.
- Two edges (i, j) and (k, l) cannot be scheduled simultaneously if (i, l) or (k, j) is an interference link.
- A node cannot be assigned a time slot to transmit a packet before it actually receives or generates that packet and a node cannot transmit more than one packet at a time slot.
- A node has a single half-duplex transceiver such that it cannot transmit and receive simultaneously and cannot receive from more than one transmitter at the same time.

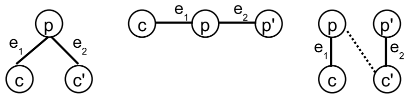


Fig. 1: Solid lines represent communication links whereas the dotted lines represent the interference links.

Theorem 1: The following problem is NP-complete. Given a tree T on an arbitrary interference graph $G = (V, E)$, and an integer t, is there an assignment of time slots to the edges in the tree using at most t slots?

This theorem has been proved in [7] by reducing the well known *Partition Problem* to the original problem. If we can remove the interference links on G, then it becomes $G = T$ and T can be scheduled in polynomial time. Therefore, initially we focus on methods to eliminate interference.

III. Mechanisms to Eliminate Interference

A. Joint Scheduling and Power Control

El Batt *et al.* [4] introduced a cross layer method for joint scheduling and power control in wireless multi-hop networks. The aim is to find a TDMA schedule which can support as many transmissions as possible in each time slot. We use their algorithm to investigate the impact of power control on the scheduling performance.

The solution is composed of 2 phases: scheduling and power control. It is to be executed at the beginning of each time slot in order to cope with excessive interference levels. The scheduling phase searches for a transmission schedule which is defined to be valid if no node is to transmit and receive simultaneously and no node is to receive from more than one neighbor at the same time. Power control phase iteratively searches for an admissible schedule which means that a set of transmission powers is available to satisfy the SINR (signal to interference and noise ratio) constraints according for all links in the given valid schedule. In each iteration nodes adjust their transmission powers.

If the maximum number of iterations is reached and if the valid scenario is not admissible, the scheduling algorithm excludes the link with the minimum SINR. The power control algorithm is repeated until an admissible transmission scenario is found. We evaluate the improvements on the schedule length with the algorithm in Section VII.

B. Frequency and Time Scheduling

The use of multiple frequency channels is an efficient way to improve the capacity of wireless networks. Simultaneous transmissions on non-conflicting frequencies can take place without interference in the same spatial neighborhood.

We argue that since interference arises at the receiver, the channels should be assigned to the receivers, i.e. to the parents on the tree, such that interfering simultaneous transmissions take place on different channels for different receivers. Motivation is as follows:

- Adjacent communication links (Fig. 1) cannot be assigned the same time slot since they have to wait for each other's transmission. Assigning non-conflicting frequencies to these nodes does not improve the situation, either. Then the receiver should be assigned a frequency and the senders should use this frequency to transmit.
- Interference links (Fig. 1) should not be assigned the same time slot and frequency. Since our aim is to minimize the number of time slots, the best option then is to assign the same time slot on non-conflicting frequencies.

Given the motivation to eliminate interference, we define the receiver-based channel assignment problem on a tree topology. First we explain the basics of the problem next study the complexity of the problem.

DEFINITION 1: *Interfering Parents:* We define interfering parents as a pair of parent nodes p and p' such that a transmission by any child of a parent interferes with a simultaneous transmission by any child of the other.

As illustrated in the last part of Fig. 1, p and $p'$are interfering parents when assigned the same frequency because simultaneous transmissions by their respective child c and c' cause interference.

DEFINITION 2: *Receiver Based Channel Assignment Problem:* Given f available channels, the problem is to assign the channels to the receivers (i.e. parents) such that all the interference links are removed.

THEOREM 2: The following problem is NP-complete. Given a tree T on an arbitrary interference graph $G = (V, E)$, and an integer f, is there a frequency assignment to the parents such that all the interference links are removed by using at most f frequencies?

Proof: To show that the problem is in NP, we reduce an arbitrary instance G of the vertex color problem to an instance G' of our problem. Our reduction is as follows, as illustrated in Fig. 2. For every vertex $v_i \in V$ construct two nodes v_{i1} and $v_{i2} \in G'$, and join them with an edge. For every edge $e_{ij} = (v_i, v_j) \in E$, construct an interference link in G' either between v_{i1} and v_{j2}, or between v_{i2} and v_{j1}, if neither of them already exists. Finally, create a root node s and add edges from each v_{i1} to s. This new graph G' is an instance of the problem. Clearly, the reduction runs in polynomial time.

Next, we show that there exists a solution to the vertex color problem using f colors if and only if there exists a solution to the original problem using f frequencies. Let G is vertex colorable using f colors, and let vertex v_i is assigned color j. Assign frequency j to node $v_{i1} \in G'$, and any of the j's to the root node s. Since no pair of adjacent vertices v_i and v_j in G are assigned the same color, no pair of vertices v_{i1} and v_{j1} in G' that have an interference link from either of them to the child of the other will have the same frequency. This is so, because by our construction an interference link is created either between the child of v_{i1} to v_{j1}, or between the child of v_{j1} and v_{i1} whenever v_i and v_j are adjacent in G. Finally, since the root does not have an interference link to any of the v_{i1}'s or their children, all the interference edges are removed.

Next, let there exists a frequency assignment in G' using f frequencies. If v_{i1} is assigned frequency j, assign color j to v_i in G. Since all the interference links are removed by such a frequency assignment, every pair of parents v_{i1} and v_{j1} that have an interference link from either of them to the child of the other are assigned different frequencies. And since their corresponding vertices v_i and v_j are adjacent in G, they will be assigned different colors. Therefore, the reduction is complete. ■

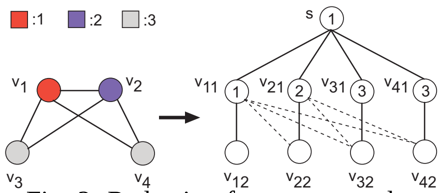


Fig. 2: Reduction from vertex color

IV. Receiver Based Channel Assignment Algorithm

As we discussed in Section III-B, the goal is to schedule the interference links on non-conflicting frequencies such that the receptions at the parents of the interfering senders are not disturbed. From Theorem 2, we know that the problem is NP-complete, in this section we introduce a greedy channel assignment algorithm. Initially, all the nodes operate on the same frequency. The method finds the interference links according to the SINR values. Accordingly, at each step the most interfered parent (the parent with the highest number of interference links) is assigned a frequency, if one is available. If not, the parent node and the associated children remain on the initial frequency and the interference conflicts have to be resolved in the time slot assignment phase.

The algorithm has a set of parents and a number of channels as an input and gives an output as the list of frequencies assigned to the parents, as illustrated in Algorithm 1. First, a list of interfering parents for each parent is created. After creating the list of interfering parents, the algorithm iteratively assigns the channels. During channel assignment, if the channels are considered to be orthogonal, the node can simply choose the next available channel. However, due to the channel overlaps, SINR value at the receiver may not be high enough to tolerate the interference. The algorithm considers the channel overlaps and assigns the channels according to the ability of the transceiver to reject the interference, i.e. adjacent channel rejection and blocking values.

Algorithm 1 Receiver-Based Frequency Scheduling

Input: P:set of parents, f:number of available channels
Output: F be the frequencies assigned to the elements in P.
I. Create list of interfering parents
for all $p \in P$ **do**
    C: set of children of p
    $P'(p)$: set of interfering parents of p
    $AC(p)$: set of available channels for parent p
    $P'(p) \Leftarrow \phi$, $AC(p) \Leftarrow \{1, 2, ..., f\}$
    for all $c \in C$ and $c' \notin C$ **do**
        if $(SINR(c, p) < \beta\ P'(p))$ **then** $P'(p) \Leftarrow$ parent of c'
    end for
end for
II. Channel Assignment
while $P \neq \phi$ **do**
    $p \leftarrow$ next most interfered parent from P
    $F(p) = i$, $i \in AC(p)$
    for all $p' \in P'(p)$ **do**
        $P'(p') = P'(p') \setminus p$
        $AC(p') = AC(p') \setminus i$
    end for
    $P'(p) = \phi$
    $P \leftarrow P \setminus p$
end while

A. Evaluation of the Greedy Algorithm

The aim of the receiver-based scheduling method is to schedule all the interfering parents on different frequencies to eliminate interference. In this section, we investigate the bounds on the required number of frequencies. We construct a constraint graph $G' = (V', E')$ from the original interference graph $G = (V, E)$ as follows: For each parent in the tree $v_i \in V$, construct a vertex $v_i^{'} \in V'$. Create a link $e'_{ij} = (v'_i, v'_j) \in E'$ if their corresponding vertices v_i and v_j in G are interfering parents.

THEOREM 3: The number of frequencies needed that would be sufficient to remove all the interference links on G is upper bounded by:

$$f \leq \Delta(G') + 1, \quad (1)$$

where $\Delta(G')$ is the maximum degree of G'.

Proof: Since interfering parents are the ones for which simultaneous transmissions by their children on the same time slot and the same frequency cause interference, so long as we assign different frequencies to every pair of interfering parents v_i and v_j in the original graph G, we can remove all the interference links.

By our construction, we create a vertex in G' for each parent in G, and a link between two such vertices if they are interfering parents in G. So assigning different frequencies to every pair of interfering parents in G is equivalent to assigning different frequencies to every pair of adjacent vertices in G'. Therefore, the minimum number of frequencies required is equal to the minimum number of colors required to vertex color G', called the chromatic number $\chi(G')$, which is bounded by one more than the maximum graph degree. ■

V. Time slot Scheduling for Tree Networks

In this section we explain how to assign time slots to the senders after frequencies are assigned to receivers. The time slot assignment algorithm, shown in Algorithm 2 is an extended version of the algorithm in [1]. The basic motivation is to schedule transmissions in parallel along multiple branches. If the sink has a single transceiver it can receive at most from only one branch at a time slot. So the algorithm should decide which branch should be transmitting at each time slot. A branch is eligible to transmit if the root of a branch has packets to transmit (root of a branch is the top-most node on a branch, connected to the sink. For instance nodes 1,2 and 3 are the roots of their branches in Fig. 3). In a given time slot, there may be more than one eligible branch (as shown on line 8 of the algorithm, E holds the list of eligible branches). In that case the branch with the highest number of remaining packets/nodes should be scheduled (line 9). We assume that all the nodes in the network have the information about the number of nodes in all the branches such that all the nodes know which branch is active at each time slot without knowing the entire topology. If there is a tie, the node with the lowest id is to be scheduled.

If a branch is active in transmitting, the nodes on the branch can be either in Tr (transmit) state or Rx (receive) state depending on their hop counts. When a branch is active, the root of the branch will be in state Tr at time slot t. The nodes with hop count h will be in Rx state if (h mod $2 = 0$) or will be in Tr state if (h mod $2 = 1$). In the next slot, nodes transit to the opposite state. The root of the branch will have data to transmit only at $t + 2$ and it will be eligible to be scheduled by then.

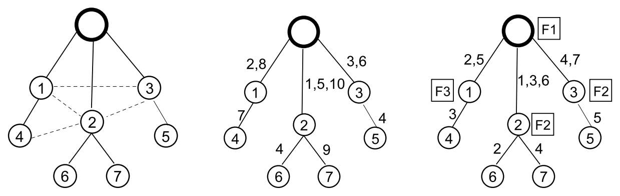
Inside the branches, there may be multiple sub-branches. For instance consider the tree in Fig. 3. Here nodes 6 and 7 cannot be simultaneously in the Tr state. Each node should know which one is going to transmit first. The algorithm assumes the first child of a parent gets active first. Once it finishes transmitting, the second sub-branch with node 7 can become active. So, the algorithm assumes all the nodes should know how many time slots to wait before transiting to state Tr, as represented on line 5 and with the condition check on line 14.

Fig. 3 shows an example network and the generated schedule by the algorithm. Fig. (a) shows the communication links and interference links. In Fig. (b) nodes are scheduled on a single channel and it takes 10 time slots. In Fig. (c), frequencies are assigned to the interfering parents and the time slot assignment takes only 6 time slots. If the interference cannot be eliminated as in the second part of the figure, we use a modified version of the presented algorithm such that among the nodes who are scheduled to be in state Tr, we check the SINR values. The algorithm schedules as many as transmissions as possible and if the SINR constraint is not met on a link, the transmission is deferred for that slot and the link is to be scheduled in the next slots.

THEOREM 4: The number of time slots to complete a convergecast is lower bounded by $\max(2n_k - 1, N)$, where n_k is the maximum number of nodes on any branch of the routing tree and N is the number of nodes in the network (in both n_k and N, the sink node is excluded).

Algorithm 2 Assignment of time slots

S∈{Tr,Rx}: Current state of a node
W: Number of packets to be forwarded by the sub-branch before the node can start transmitting
$B = 0$: Number of packets that has been forwarded by the sub-branch
Initialize S according to hop count
Initialize W with initial numbers given by the sink
$t \leftarrow 1$
while $n_i \neq 0$ for all branches **do**
    E: set of branches eligible for scheduling at t
    $j = \arg\max_{i \in E}\{n_i\}$
    Sink receives from branch j at t with nodes on j active on t and $t+1$
    $n_j \leftarrow n_j - 1$
    for all nodes that are active on t **do**
        $B \leftarrow B + 1$
        if $B \geq W$ **then**
            if $S = Tr$ **then** transmit a packet; $S \leftarrow Rx$
            if $S = Rx$ **then** receive a packet; $S \leftarrow Tr$
        end if
    end for
    t ← t + 1
end while



(a) (b) (c)

Fig. 3: (a) Communication and interference links; (b) Schedule with a single channel; (c) Schedule with 3 channels

Proof: Let n_i represent the number of nodes in branch i. Order the branches in non-increasing order of their branch sizes; let this order be $n_k \geq n_{k-1} \geq ... \geq n_1$. Assume all the interfering links are eliminated by utilizing multiple channels. Since no node can receive multiple packets in a single slot, N is a trivial lower bound to receive all the packets originated in the network. Next, consider branch k that has the highest number of children. The root of this branch has to transmit n_k packets, and the children of this root have to forward $n_k - 1$ packets in total. Due to the half-duplex nature, time slots assigned to the root of this branch must be distinct from that assigned to its children. Therefore, in total we need $n_k + (n_k - 1) = 2n_k - 1$ distinct time slots. ■

We should note that, this bound is smaller than the existing bound which was calculated as $3N$ for general networks and $\max(3n_k - 3, N)$ for tree networks, where the only limiting factor is the half-duplex transceivers. Gandham *et al.* [1] showed that the number of time slots required by the original version of the algorithm is $\max(3n_k - 1, N)$ which is 2 time slots more than the lower bound using a single channel. We now show that a modified version of the algorithm can compute schedules with a length of $\max(2n_k - 1, N)$, which is exactly the lower bound when interference is eliminated with multi-channel scheduling.

THEOREM 5: The schedule length required by Algorithm 2 is at most $\max(2n_k - 1, N)$.

Proof: The idea of the proof is based on that given in [1]. Let n_i represent the number of nodes in branch i. Order the branches in non-increasing order of their branch sizes; let this order be $n_k \geq n_{k-1} \geq ... \geq n_1$. Suppose $n_k > \sum_{i=1}^{k-1} n_i$. From Theorem 4, we know that it takes at least $2n_k - 1$ slots to schedule branch k, out of which the sink can use at most $n_k - 1$ slots to receive packets from other branches. Since the total number of packets in the other branches is at most $n_k - 1$, the schedule length is no more than $2n_k - 1$.

Now suppose $n_k \leq \sum_{i=1}^{k-1} n_i$. In this case, $\max(2n_k - 1, N) = N$. If $n_k = 1$, each of the other branches can have at most one node because k is the largest branch. Since the algorithm schedules the most loaded branch in every time slot, in total it will take N slots. Otherwise, if all the branches have equal size, the same situation repeats, and the algorithm schedules the branches in consecutive slots requiring at most $n_k \cdot k = N$ time slots in total.

For all other cases, we use induction as follows. Assume that the algorithm uses N time slots for convergecast when the most loaded branch on the network has M nodes. Next, consider a network where the most loaded branch has $M + 1$ nodes such that $n_k = M + 1$. The algorithm schedules branch k and and the next most loaded branch in the first and second time slots. At the third time slot the branch k has M packets left. If k is still the most loaded branch then according to our assumption, the remaining packets in the network can be scheduled in $N - 2$ time slots. Therefore, the complete convergecast can be completed in N time slots.

If more than two branches in the network have $M+1$ nodes, then at the third time slot the most loaded branch will still have $M+1$ packets left. Assume l branches have $M+1$ nodes such that $l \leq k$ and $l > 2$. Since the algorithm schedules the most loaded branch first, at the $(l+1)^{th}$ time slot the most loaded branch will have M packets. According to the assumption it will take $N - l$ time slots to schedule the remaining packets. Therefore, the convergecast can be completed in N time slots. ■

VI. IMPACT OF ROUTING TREE

As emphasized in [7], routing trees that allow more parallel transmissions do not always result in small schedule length. For instance, given a network, the schedule length would be N with a star topology whereas it would be $2N - 1$ with a line topology, assuming there interference links are removed. The structure of the routing tree also plays an important role on the schedule length. According to Theorem 4, the routing tree should be constructed with balanced number of nodes on branches, such that $2n_k - 1 < N$. In this section, we explore the possibilities of constructing such trees.

THEOREM 6: The following problem is NP-complete. Given an arbitrary graph G, can we construct a tree T on G, such that $n_k \leq \frac{N+1}{2}$?

Such a tree construction is defined as "Capacitated Minimal Spanning Tree Problem" and is proved to be NP-complete [6]. Given the hardness of the problem, we rely on heuristics. Esau *et al.* [8] use a greedy heuristic in solving the problem, using a cost function according to the load that a node may bring to a branch. However, they do not consider the growth possibilities of the branch and the node. The growth possibility is defined as a measure to grow a branch outwards a node [9] and such an information is very important at tree construction to decide which nodes and branches to process first.

We propose a heuristic algorithm that considers such growth possibilities and breaks down the tree construction mechanism into two parts. First every node collects information about the potential branches that it can connect to. Starting from the sink, all the nodes propagate information about their hop-count to the sink and the potential branch id's (direct neighbors of the sink become the roots of the branches and branch id is the id of the root of a branch). In the second part the tree is constructed as shown in Algorithm 3. The variable *growth set*, GS of a node n includes n and the neighbor nodes that are not yet connected to the tree. Similarly, *growth set* of a branch includes the unconnected neighbors of the nodes that are already on the branch. *Weight* of a branch is the number of nodes that are already connected to the branch. BA represents the branch access.

The algorithm iteratively grows a tree hop by hop outwards the sink node. At each hop, first the nodes that have a single potential parent are connected. Next, the node with the largest growth set should be added to tree via the branch with the minimum weight. This balances the number of nodes on different branches and prevents a branch to grow faster than the other branches. However, selecting the branch with the minimum weight is not always the best option for the nodes that are downwards on the tree. For instance consider a situation in Fig. 4. Node 3 is processed (if the cardinality of the *growth set* of two nodes is the same, the node that has a smaller id is processed first) and is added to to the branch 1 (B1). When node 4 is processed, again branch B1 should be selected considering only the weights of the branches. However, if node 4 also connects to the branch B1 the nodes 8,9 and 10 have only access to the branch B1 and branch B1 will be more crowded than branch B2. But if node 4 connects branch B2, the nodes are balanced over the two branches and $n_k \leq \frac{N+1}{2}$. In Algorithm 3, starting from line 16 a search set (SS) is created for node n for each branch b and it is initialized with the *growth set* of b. If n joins b, and if a node in the search set will have access only to branch b, the node is added to the potential growth (PG) set of n.

Although, this basic algorithm tries to keep the number of nodes on each branch as minimum, an additional balancing may still be needed. We use the adjustment algorithm used in [9] by moving the nodes on the most-loaded branch to the neighboring branches that can decrease the value of n_k.

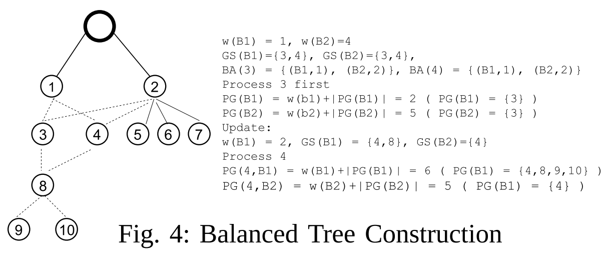


Fig. 4: Balanced Tree Construction

Algorithm 3 Capacitated Minimal Spanning Tree

1: **Input**: $G(V, E)$ be the communication graph, s be the sink, GS is the *growth set*, BA includes the tuples to represent branch access via a node
2: **Output**: T represents the tree
3: $T \leftarrow s$, $B \leftarrow$ id's of the roots of the branches
4: $\forall n \in V$, $GS(n) \leftarrow n$, unconnected neighbors of n
5: $\forall b \in B$ $weight(b) \leftarrow 1$, $GS(b) \leftarrow$ unconnected neighbors of b
6: $h = 2$
7: **while** $h \neq max(hop\_distance)$ **do**
8:   N: set of nodes at hop distance h
9:   Connect the nodes that have a single potential parent first
10:   Sort N according to the $|GS|$ values in ascending order
11:   **for all** $n \in N$ **do**
12:     **for all** $b \in B$ that n can connect to **do**
13:       $SS \leftarrow GS(b)$ Search set
14:       $PG(n, b) \leftarrow \phi$ Potential growth set that n brings on b
15:       **for all** $i \in SS$ **do**
16:         **if** $BA(i, :)$ == b **then**
17:           $PG(n, b) \leftarrow i$, $SS \leftarrow GS(i)$
18:         **end if**
19:       **end for**
20:       $PG(n, b) = W(b) + |PG(n, b)|$
21:     **end for**
22:     Connect n to the branch b where $PG(n, :)$ is the minimum
23:     Update the *growth set*'s and *weights* of the related branches
24:     h++
25:   **end for**
26: **end while**

VII. Evaluation

We use a simulation based approach using Matlab to evaluate the impact of different mechanisms on the scheduling performance. Nodes are randomly deployed over the area. Terrain dimensions are varied between 20×20 and 300×300 m^2 to simulate different levels of density whereas the number of nodes is kept as 100 (the node with id "1" is always selected as the sink node). For different parameter settings, we repeat the simulations 1000 times.

We use an exponential path loss model for signal propagation with a path loss exponent $\alpha = 3.5$ which is a typical value for indoor environments. We simulate the behavior of the CC2420 radio which is used on the Telosb and Tmote sensor mote platforms. The transmission power can be adjusted between -24dBm and 0dBm over 8 different levels. SINR threshold is β=-3dB. The transceiver is capable of operating on 16 different frequency channels.

A. Impact of Power Control

In this section we evaluate the impact of transmission power control on the scheduling performance. We investigate two cases: nodes transmit with the maximum transmission power and nodes adjust their transmission power according to the power control algorithm which is explained in Section III-A.

The results are presented in Fig. 5. The x-axis shows the length of a square area. The y-axis shows the number of time slots required to schedule all the transmissions in the network. Different lines display the results with different path loss exponents $\alpha = 3.0, 3.5, 4.0$ to discuss the impact of physical layer parameters. We cannot provide the results for $L > 200, \alpha = 4.0$ since it's hardly possible to generate connected trees.

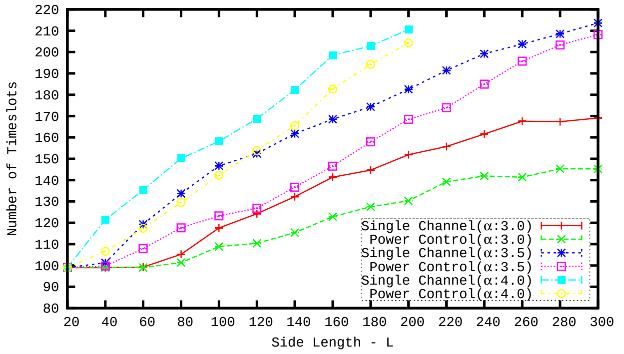


Fig. 5: Joint Scheduling and Transmission Power Control

Required number of time slots, i.e. the schedule length, increases as the network gets sparser. One would expect the other way around since in sparser deployments the reuse of the time slots should be higher which would result in a smaller schedule length. However, as the network gets sparser, the number of nodes that can directly reach the sink decreases such that the packets have to be relayed over multi-hops. In this case more packets have to be scheduled compared with scheduling packets in a single-hop setting. In the simulations we observe that the number of packets to be scheduled increases faster than the reuse ratio. In the densest setting ($L = 20$), where all the nodes can directly reach the sink, the schedule length is 99, equal to the number of transmitting nodes in the network.

If the nodes adjust the level of transmission power, we observe that the schedule length is smaller since some interference is eliminated and the reuse of the time slots is increased. When $\alpha = 3.0$, most of the interference is eliminated and the limiting factor is the routing tree structure. In this set of simulations the routing trees were constructed as shortest path spanning trees and the limiting factor is due to the maximum number of nodes on a branch such that $2n_k - 1 > N$. However, when $\alpha \geq 3.5$ the transmission power control approach cannot always eliminate the interference. In this case, the generated networks are sparser (transmission range is around 37.5m with $\alpha = 3.5$ and 23m with $\alpha = 4.0$ while it is around 68m with $\alpha = 3.0$). In sparse scenarios, the nodes cannot decrease the transmission power further than the maximum level since the transceiver cannot decode the signals below the sensitivity level which is $-95dBm$. Especially in sparser deployments, $L \geq 200$, the results are similar either the nodes transmit with the maximum power or adjust the power level. Moreover, in mid-sparse deployments ($60 \leq L \leq 180$) the discrete power levels (8 levels) and the limited range of the transmission power limits the nodes to adjust their transmission power.

B. Impact of Receiver Based Scheduling and Routing Trees

In this section we introduce the results on the performance of receiver-based frequency and time scheduling method which is explained in Section III-B. Fig. 6 presents the results with the x-axis showing the length of a side of the square deployment area and the y-axis showing the schedule length. Different lines show the results when different number of channels are available and the last line shows the schedule length when the routing tree is constructed with balanced number of nodes on each branch, such that $2n_k - 1 < N$.

When the number of available channels increases, we observe a reduction on the schedule length. However, when the number of channels is 6 or higher, the schedule length cannot be reduced any more since the interference is eliminated and the limiting factor is due to the half-duplex operation of the sink node.

This set of simulations verifies that the receiver-based frequency and time scheduling method can achieve a schedule length which is bounded by $\max(2n_k - 1, N)$ as long as the number of available channels on the transceiver is sufficient to eliminate the interference. Compared with single-channel results in sparser scenarios, we achieve a reduction of up to 40% on the schedule length. In very dense scenarios (low L) reduction is small since most of the nodes can directly reach the sink node and the limiting factor is the half-duplex capability of the transceiver of the sink node.

In Fig. 6, the first six lines represent the results collected with shortest path spanning trees. In the last line results are presented according to the tree construction method explained in Section VI (results are displayed only with 16 channels due to the limited space). The impact of such routing trees is more visible in sparser scenarios ($L \geq 200$) where a further reduction on the schedule length is observed. When $L \leq 200$, the schedule length is bounded by N. Beyond this point it is mostly not possible to construct trees where the $2n_k - 1 < N$ constraint can be met and the schedule length is limited by $2n_k - 1$ where n_k is minimized by the tree construction algorithm. As a result of this set of simulations, the receiver-based channel assignment method combined with a suitable tree construction mechanism can achieve a reduction of up to 50% on the schedule length compared with single-channel communication on shortest path spanning trees.

1) Bounds on the number of frequencies: In Fig. 7, we compare the upper bound $\Delta(G') + 1$ on f as per Theorem 3 with the actual number of frequencies required to remove all interference links on different kinds of trees. The x-axis presents the terrain length whereas the y-axis shows the number of channels. The top line presents the upper bound, the lower line presents the actual number of frequencies used by the receiver-based channel assignment method.

The number of required frequencies is initially very low when the network is very dense $L < 40$, since the transmissions cannot be scheduled in parallel since the number of

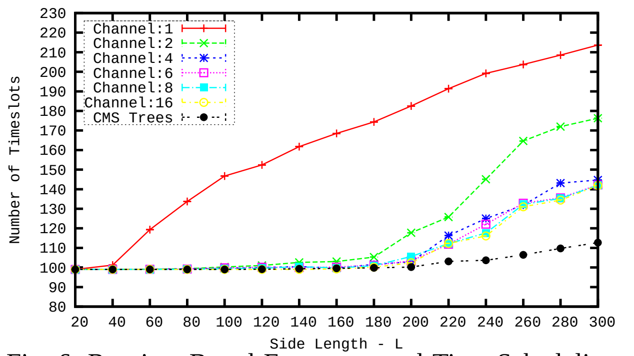


Fig. 6: Receiver-Based Frequency and Time Scheduling

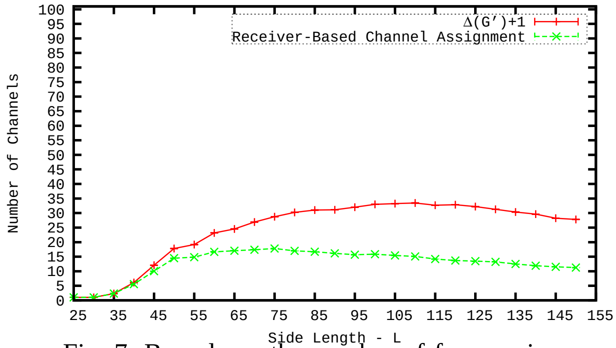


Fig. 7: Bounds on the number of frequencies

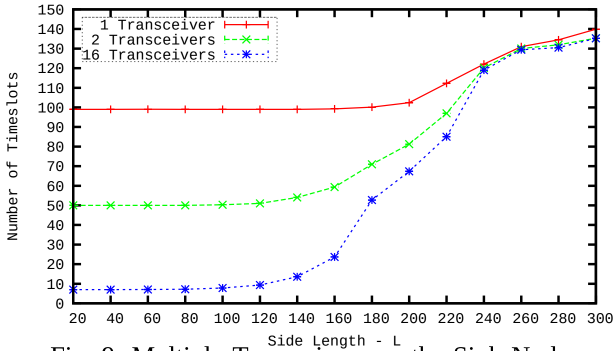


Fig. 9: Multiple Transceivers on the Sink Node

receivers is low. When $L \leq 30$, all the nodes can directly reach the sink node, one frequency is sufficient. As the network gets sparser the number of receivers (i.e. parents) increases. Accordingly, the level of interference in the network increases and more frequencies are required to support parallel transmissions. However, when $L \geq 80$, the number of required channels decreases since the level of interference decreases. Trends of both of the lines are quite similar. Receiver-based channel assignment actually requires less time slots than the calculated upper bound and the required number of channels to eliminate interference is lower than or equal to the available number of 16 channels on CC2420 radios, with 100-node networks.

2) Comparisons: In this section, we compare the performance of the receiver-based scheduling method with the TMCP protocol [5]. TMCP is a tree based channel assignment method such that different channels are allocated to each branch of the tree. The goal is to partition the network into multiple subtrees with minimizing the intra-tree interference. It is a greedy algorithm and assigns the channels one by one to the nodes from top-to-bottom on a fat tree. When a node is to be added to a subtree, the subtree where the node brings the least interference is selected.

After the channel assignment, the time slots are assigned to the nodes with the same method as explained in Section V. Fig. 8 presents the comparisons between the receiver-based channel assignment and the TMCP protocol with 2 and 16 channels with the x-axis showing the side length of the deployment area and the y-axis showing the schedule length. We use shortest path routing trees (not the balanced trees) with the receiver-based channel assignment method for a fair comparison. Receiver-based channel assignment performs approximately the same only with 2 channels while TMCP uses 16 channels since the method allows more nodes to transmit in parallel. For instance while a node is receiving from its children, the parent of this node can transmit simultaneously which would not be possible if they communicate on the same channel.

C. Multiple Transceivers at the sink node, multiple sinks

In this section we analyze the schedule length when the sink is equipped with multiple half-duplex transceivers such that the transceivers can receive in parallel from different senders. We vary the number of transceivers, tx, from 1 to 4 and accordingly tx trees are created in parallel in the simulations.

Fig. 9 presents the results (shortest path spanning trees are used). In denser scenarios ($L < 140$), reduction on the schedule length is proportional to the number of available transceivers at the sink node. However, in sparser scenarios, especially when $L > 220$, there is almost no reduction on the achievable schedule length if the sink has a single transceiver or multiple transceivers. In sparser scenarios, the number of neighbors that a node can connect to is limited. Therefore, it is difficult to balance the number of nodes transmitting to a particular transceiver of the sink node such that $2n_{kt} - 1 < N_t$ where n_{kt} is the maximum number of nodes on any branch of tree t and N_t is the number of nodes on tree t. In sparser scenarios n_{kt} with multiple transceivers and n_k with a single transceiver is mostly the same.

Next, we evaluate the schedule length if there are multiple sinks deployed within the network. We vary the number of sinks from 1 to 16. Sinks are randomly deployed as well as the nodes. Fig. 10 shows the results. Compared with the results in Fig. 9, we can achieve a reduction on the schedule length also

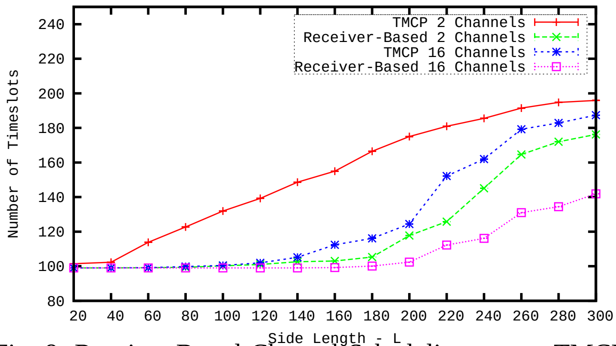


Fig. 8: Receiver-Based Channel Scheduling versus TMCP

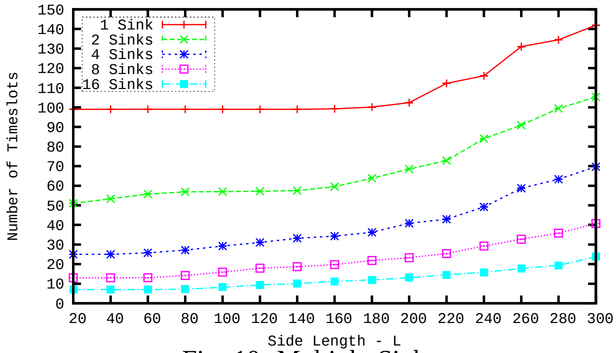


Fig. 10: Multiple Sinks

in sparser scenarios since the number of transmitting nodes to different sink nodes can be balanced. In denser deployments ($L < 100$) the reduction is proportional with the number of available channels. However, when $L \leq 200$, a factor of half of the available sinks is achieved due to the sparseness and less connectivity.

VIII. Related Work

A closely related study by Gandham *et al.* [1] focuses on finding a TDMA schedule that minimizes the total time required to complete a convergecast in WSN. Although, they address the same problem as ours, we focus on methods that can reduce the schedule length by eliminating the limitations due to interference, half-duplex transceiver and routing tree where they are interested in showing the bounds on the schedule length with different network organizations. Furthermore, we improve the presented bounds on the length of convergecast schedules. Another similar study is presented in [7] where the NP-completeness of the problem is proved with single-channel communication. However, the authors don't address how to overcome the limitations on schedules, either. Duarte-Melo *et al.* [10] discuss the convergecast operations in WSN with flat and hierarchical topologies using probabilistic models. Their objective is not minimizing the schedule length but maximizing capacity and they consider simpler graph-based interference models.

We have addressed a similar problem in [11] where each link of the tree is scheduled only once assuming that the data is aggregated before relayed towards the sink node. In the aggregated-convergecast work, similarly we discussed the impact of transmission power control and using multiple channels on the schedule length. However, the presented results were based on simulations. In this work, we study the hardness of finding the minimum schedule length on arbitrary graphs, the hardness of the frequency assignment problem and the hardness of the tree construction where the $\max(2nk - 1, N)$ constraint is met. In the aggregated-convergecast problem, the constraints were much simpler and the nature of the problem in terms of the achievable schedule length was totally different. The schedule length in the former was found to be bounded by the maximum degree of a tree and in the current problem it is found and proved to bounded by the number of nodes in the network if a suitable tree construction method is used. Scheduling with aggregation was also addressed in [12] using orthogonal codes, considering the impact of routing tree and in [13] by using non-linear transmission power control mechanisms.

The use of multiple frequency channels has been extensively studied for both cellular and ad hoc networks. In the WSN domain, there exist recent studies that utilize multiple channels [5], [14], [15]. Different than the previous work, we introduce a simple frequency and time scheduling method. Instead of assigning frequencies to the links or branches we consider a receiver-based frequency assignment which is suitable for data collection on a tree topology.

IX. Conclusions

We have explored fast convergecast scheduling in wireless sensor networks where the nodes communicate on a TDMA schedule and the objective is to minimize the schedule length to complete convergecast operations. By addressing the fundamental limitations due to interference and half duplex nature of the radios on the nodes, we explored techniques to eliminate those limitations. We found that while power control is helpful in reducing the schedule length, scheduling transmissions on different frequency channels is more efficient in mitigating the effects of interference. Once the interference is eliminated, we proved that with half-duplex radios the achievable schedule length is lower-bounded by $\max(2n_k - 1, N)$, where n_k is the maximum number of nodes on a subtree and N is the number of nodes in the network. Using an optimal convergecast scheduling algorithm, we showed that the lower bound is achievable once a suitable balanced routing scheme is used. Through extensive simulations, we demonstrated up to 50% reduction on the schedule length by using the mentioned improvements compared with single-channel communication on minimum spanning trees.

References

[1] S. Gandham, Y. Zhang, and Q. Huang, "Distributed time-optimal scheduling for convergecast in wireless sensor networks," *Comput. Netw.*, vol. 52, no. 3, pp. 610–629, 2008.

[2] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. G. Gouda, Y. ri Choi, T. Herman, S. S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita, "A line in the sand: a wireless sensor network for target detection, classification, and tracking," *Computer Networks*, vol. 46, no. 5, pp. 605–634, 2004.

[3] J. Mao, Z. Wu, and X. Wu, "A tdma scheduling scheme for many-to-one communications in wireless sensor networks," *Comput. Commun.*, vol. 30, no. 4, pp. 863–872, 2007.

[4] T. A. ElBatt and A. Ephremides, "Joint scheduling and power control for wireless ad-hoc networks," in *Proceedings of IEEE INFOCOM 2002*, vol. 2, Jun 2002, pp. 976–984.

[5] Y. Wu, J. Stankovic, T. He, and S. Lin, "Realistic and efficient multi-channel communications in wireless sensor networks," in *Proceedings of IEEE INFOCOM 2008*, 2008, pp. 1193–1201.

[6] C. H. Papadimitriou, "The complexity of the capacitated tree problem," *Networks*, vol. 8, no. 3, pp. 217–230, 1978.

[7] H. Choi, J. Wang, and E. Hughes, "Scheduling for information gathering on sensor network," *Wireless Networks*, 2007.

[8] L. R. Esau and K. C. Williams, "On teleprocessing system design part ii: A method for approximating the optimal network," *IBM Systems Journal*, vol. 5, no. 3, pp. 142–147, 1966.

[9] H. Dai and R. Han, "A node-centric load balancing algorithm for wireless sensor networks," in *Proceedings of IEEE GLOBECOM '03*, vol. 1, Dec. 2003, pp. 548–552 Vol.1.

[10] E. J. Duarte-Melo and M. Liu, "Data-gathering wireless sensor networks: organization and capacity," *Comput. Netw.*, vol. 43, no. 4, pp. 519–537, 2003.

[11] O. D. Incel and B. Krishnamachari, "Enhancing the data collection rate of tree-based aggregation in wireless sensor networks," in *Proceedings of SECON 2008*, 2008, pp. 569 – 577.

[12] V. Annamalai, S. Gupta, and L. Schwiebert, "On tree-based convergecasting in wireless sensor networks," in *Proceedings of WCNC 2003*, vol. 3, March 2003, pp. 1942–1947 vol.3.

[13] T. Moscibroda, "The worst-case capacity of wireless sensor networks," in *Proceedings of IPSN '07*, 2007, pp. 1–10.

[14] G. Zhou, C. Huang, T. Yan, T. He, J. Stankovic, and T. Abdelzaher, "Mmsn: Multi-frequency media access control for wireless sensor networks," in *Proceedings of IEEE Infocom*, 2006, pp. 1–13.

[15] Y. Kim, H. Shin, and H. Cha, "Y-mac: An energy-efficient multi-channel mac protocol for dense wireless sensor networks," in *Information Processing in Sensor Networks, 2008. IPSN '08. International Conference on*, April 2008, pp. 53–63.