Scheduling with Multiple Channels for Rapid Convergecast in Wireless Sensor Networks

Prof. Raxit Jani,
Asst. Prof. (CE), G.I.T.,
Gandhinagar

Ms. Swinkhal Sakaria
B.E. (CE), G.I.T.,
Gandhinagar

Dr. Ramesh Prajapati
Asso. Prof. (CE), L.J.I.E.T.,
Ahmedabad

Abstract— **We search the following fundamental question - how fast can information be composed from a wireless sensor network? We believe a number of design parameters such as, power control, time and frequency scheduling, and routing. There are basically two factors that delay efficient data collection - interference and the half-duplex single-transceiver radios. We show that while power manage helps in falling the number of transmission slot to complete a convergecast below a single rate channel, scheduling transmissions on diverse frequency channels is more able in mitigating the things of interference (empirically, 6 channels suffice for most 100-node networks). With these explanation, we define a receiver-based channel assignment problem, and prove it to be NP-complete on general graphs. We then begin a greedy channel assignment algorithm that proficiently eliminate interference, and evaluate its performance with other on hand schemes via simulations. Once the interference is entirely eliminated, we demonstrate that with half-duplex single-transceiver radios the attainable schedule duration is lower-bounded by $\max(2n_k - 1, N)$, where n_k is the most number of nodes on any subtree and N is the number of nodes in the network. We transform an existing dispersed time slot assignment algorithm to attain this bound when a suitable balanced steering scheme is engaged. Through widespread simulations, we reveal that convergecast can be finished within up to 50% less time slots, in 100-node networks, using multiple channels as compare to that with single-channel statement. Finally, we also exhibit extra improvements that are possible when the sink is prepared with multiple transceivers or when there are multiple sinks to gather data.**

I. Introduction

Convergecast, namely compilation of facts from sensors towards a general sink node over a tree topology is a fundamental operation in wireless sensor networks (WSN) [1]. In many application, it is significant to deliver the data to the sink in a limited total of time and boost the speed of data collection at which the sink can receive data from the network. For example in Lites [2], which is a real time monitoring application, a typical affair may produce up to 100 packets within a few seconds and the packets need to be elated from dissimilar network location to a sink node.

Since the data has to be deliver in a short time, we believe time division multiple access (TDMA) [3] as a natural answer due to the collision free behavior. believe a schedule of t time slots where the sink receive data from all nodes in the network once every t slots. In such a context, the purpose is to diminish t to increase the speed of data collection.

We learn a set of techniques in order to crack the fundamental problem: "how fast can facts be convergecast to the sink over a tree topology?" The fundamental warning factors are interference and half-duplex scenery of the transceivers on WSN nodes. To cope with interference we believe different techniques such as transmission power control and assigning diverse frequency channels on interfering links. We show that once several frequencies are employed with spatial-reuse TDMA, the convergecast schedule becomes imperfect by the number of nodes in the network once a apposite routing tree is used. For extra improvements, we believe equipping a single sink with multiple transceivers, and also the operation of multiple sinks to gather data.

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

- assessment of transmission power manage to eliminate interference: Under idealized setting (unlimited power, continuous range) power control mechanisms may provide unbounded improvement in the speed of data collection. We appraise the performance with an optimal power control algorithm describe in [4] in a practical setting allowing for the limited discrete power levels obtainable in today's radios on WSN nodes.
- Receiver-based frequency assignment: We prove that scheduling transmissions on diverse frequency channels is more capable in mitigating the things of interference compared with broadcast power control. Accordingly, we classify a receiver-based channel project problem which is "the problem of assigning a smallest number of frequencies to the receivers such that all the interference links in an arbitrary network is distant". We show that the dilemma is NP-complete and introduce a greedy heuristic for channel assignment. By simulation and analytical calculations, we appraise the behavior of our heuristic algorithm and evaluate its performance with another channel assignment method which was recently

Proposed for WSN with tree topologies [5].

Bounds on convergecast scheduling: We prove that, once the interference is eliminated, the achievable schedule length with half-duplex transceivers is bordered by $\max(2n_k - 1, N)$ slots where n_k is the greatest number of nodes on any bough of the tree and N is the number of nodes. We adapt an existing time slot assignment algorithm and show that the algorithm requires closely $\max(2n_k - 1, N)$ slots to schedule a given network.

Impact of Routing Trees: According to the bound on con- have impartial number of nodes such that $2n_k - 1 < N$. vergecast schedules, the twigs of a routing tree should Such a tree manufacture is defined as the "Capacitated Minimal Spanning Tree Problem" and is proved to be NP-complete in [6]. Given the stiffness of the problem, we propose a heuristic algorithm and appraise the impact of such routing trees on the schedule length by simulations.

Multiple transceivers at the sink node: For further Improvements we believe the sink having multiple transceivers and manifold sinks deployed in the network. We observe improved reduction on the schedule length that are proportional with the numeral of accessible transceivers.

The remnants of the paper is controlled as follows: in Section II, we initiate the problem. In Section III we explain the mechanisms that we use to eradicate interference. In Section IV, we introduce a receiver-based greedy channel assignment algorithm. In Section V, we present 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 summarize some of the linked work. Finally, Section IX provides the finishing remarks.

Preliminaries and Problem Statement

Before explanation the studied mechanisms, we first describe our beginning design and give the details of the problem formulation. We presume time is divided into equal sized slots and each node is assigned a time slot to broadcast data. All the nodes in the network apart from the sink are sources and produce one packet for each convergecast operation.

We mold the sensor network as a graph $G = (V, E)$, where V is the set of nodes and E is the set of edges that embody communication links and interference links among 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 share (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 associated. The problem we address is the following. Given G, find an assignment of time slots to the transmitters such

2

that the the numeral of time slots to complete a convergecast is minimized with subject to the subsequent transmission constraints.

Two adjacent edges (see Fig. 1) cannot be scheduled at the same timeslot. An edge (k, l) is adjacent to edge (i, j) if $\{i, j\} \quad \{k, l\}\varphi$.

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 convey a packet before it actually receives or generate that packet and a node cannot broadcast more than one packet at a time slot.

A node has a single half-duplex transceiver such that it cannot transmit and obtain concurrently and cannot receive from more than one transmitter at the same time.

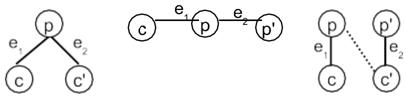


Fig. 1: Solid lines symbolize communication links whereas the dotted lines characterize the interference links.

Given a tree T on an arbitrary interference graph $G = (V, E)$

Theorem 1: The subsequent problem is NP-complete. 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 dropping the well eradicate the interference links on G, then it become G = known *Partition Problem* to the original problem. If we can T and T can be scheduled in polynomial time. Therefore, originally we focus on methods to eliminate interference.

Mechanisms to eliminate interference

Joint Scheduling and Power Control

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

The answer is collected of 2 phases: scheduling and power control. It is to be executed at the foundation of each time slot in order to survive with excessive interference levels. The scheduling phase searches for a broadcast schedule which is definite to be valid if no node is to convey and receive concurrently and no node is to receive from more than one neighbor at the same time. Power control phase iteratively searches for an allowable agenda which means that a set of communication powers is available to please 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 greatest numeral of iterations is reached and if the valid scenario is not permissible, the scheduling algorithm exclude the link with the smallest SINR. The power control algorithm is recurring until an allowable transmission scenario is found. We calculate the improvement on the schedule length with the algorithm in Section VII.

A. Frequency and Time Scheduling

The use of several frequency channels is an proficient way to improve the ability of wireless networks. concurrent transmissions on non-conflicting frequencies can take place without interference in the same spatial environs.

We disagree 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 concurrent transmissions take place on dissimilar channels for different receivers. Motivation is as follows:

close communication links (Fig. 1) cannot be assigned the equal time slot since they have to wait for each other's transmission. Conveying non-conflicting frequencies to these nodes does not advance the situation, either. Then the receiver should be assigned a frequency and the sender should use this frequency to transmit.

Interference links (Fig. 1) should not be assign the same time slot and frequency. Since our try is to diminish the number of time slots, the top option then is to allot the same time slot on non-conflicting frequencies.

Given the enthusiasm to eradicate interference, we define the receiver-based channel assignment difficulty on a tree topology. First we clarify the basics of the trouble next study the complication 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 interfere with a instantaneous transmission by any child of the other.

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

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

Given a tree T on an arbitrary interference graph $G = (V, E)$

THEOREM 2: The following problem is NP-complete. and an integer f, is there a frequency obligation to the parents such that all the interference links are detached by using at most f frequencies?

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

Next, we show that there exists a answer to the vertex color difficulty using f colors if and only if there exists a answer to the unique trouble 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 nearby 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 assembly 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 detached.

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 distant by such a frequency obligation, 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 diverse frequencies. And since their equivalent vertices v_i and v_j are adjacent in G, they will be assigned different colors. Therefore, the lessening is absolute.

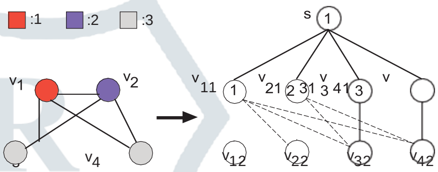


Fig. 2: Reduction from vertex color

RECEIVER BASED CHANNEL ASSIGNMENT ALGORITHM

As we discuss in Section III-B, the aim is to schedule the interference links on non-conflicting frequencies such that the reception at the parents of the interfering senders are not troubled. From Theorem 2, we know that the problem is NP-complete, in this section we bring in a greedy channel obligation algorithm. Initially, all the nodes work 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 obtainable. If not, the parent node and the associated children linger on the initial frequency and the interference conflicts have to be resolute 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 roll of frequencies assigned to the parents, as illustrated in Algorithm 1. First, a list of interfering parents for each parent is shaped. After creating the list of interfering parents, the algorithm iteratively assigns the channels. During channel assignment, if the channels are measured to be orthogonal, the node can simply decide the next available channel. However, due to the channel overlap, 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

```
1: Input: P :set of parents, f :number of available channels
2: Output: F be the frequencies assigned to the elements in P .
3: I. Create list of interfering parents
4: for all p ∈ P do
5:     C: set of children of p
6:     P′(p): set of interfering parents of p
7:     AC(p): set of available channels for parent p
8:     P′(p) ⇐ φ, AC(p) ⇐ {1, 2, ..., f}
9:     for all c ∈ C and c′ f∈ C do
10:        if (SINR(c, p) < β P′(p)) then P′(p)⇐ parent of c′
11:    end for
12: end for
13: II. Channel Assignment
14: while P f= φ do
15:    p ← next most interfered parent from P
16:    F(p) = i, i ∈ AC(p)
17:    for all p′ ∈ P′(p) do
18:        P′(p′) = P′(p′) \ p
19:        AC(p′) = AC(p′) \ i
20:    end for
21:    P′(p) = φ
22:    P ← P \ p
23: end while
```

A. Evaluation of the Greedy Algorithm

The try of the receiver-based scheduling process is to schedule all the interfering parents on diverse frequencies to eradicate interference. In this segment, we examine the boundaries on the required numeral of frequencies. We assemble a constraint graph G′ = (V ′, E′) from the original interference graph G = (V, E) as follows: For each parent in the tree $v \in V$, construct a vertex $v \in V'$.

Theorem 3: The numeral of frequencies wanted that would be enough to remove all the interference links on G is upper bounded by:

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

where Δ(G′) is the highest degree of G′.

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

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

Time slot Scheduling for Tree Networks

In this segment we explain how to assign time slots to the senders after frequencies are assigned to receivers. The time

slot obligation algorithm, shown in Algorithm 2 is an extended version of the algorithm in [1]. The basic incentive is to schedule transmissions in similar along several branches. If the sink has a distinct transceiver it can receive at most from only one branch at a time slot. So the algorithm should choose which branch should be transmitting at each time slot. A branch is entitled to transmit if the root of a branch has packets to transmit (root of a bough 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 qualified branch (as shown on line 8 of the algorithm, E holds the list of eligible branches). In that case the branch with the maximum number of residual packets/nodes should be scheduled (line 9). We suppose that all the nodes in the network have the in sequence about the figure of nodes in all the branches such that all the nodes know which branch is active at each time slot without meaningful the entire topology. If there is a tie, the node with the lowest id is to be listed.

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 broadcast only at t + 2 and it will be eligible to be scheduled to the opposite state. The root of the branch will have data to by then.

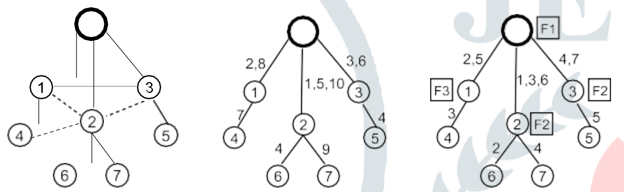
within the branches, there may be numerous sub-branches. For case believe the tree in Fig. 3. Here nodes 6 and 7 cannot be concurrently in the *Tr* state. Each node should know which one is going to transmit first. The algorithm assume the first child of a parent gets active first. Once it finishes transmitting, the second sub-branch with node 7 can befall active. So, the algorithm assumes all the nodes should know how many time slots to wait before transiting to state *Tr*, as stand for on line 5 and with the situation check on line 14.

Fig. 3 shows an instance network and the generated schedule by the algorithm. Fig. (a) shows the note links and interference links. In Fig. (b) nodes are planned on a single channel and it takes 10 time slots. In Fig. (c), frequencies are assigned to the interfering parents and the time slot task takes only 6 time slots. If the interference cannot be eliminated as in the second part of the figure, we use a adapted version of the presented algorithm such that among the nodes who are planned 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 delayed for that slot and the link is to be scheduled in the next slots.

convergecast is lesser bounded by max(2nk − 1, N), where THEOREM 4: The number of time slots to absolute a nk 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 nk 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 ← 1
while n_i     0 for all branches do
    E: set of branches eligible for scheduling at t
    j = arg max{ni}
i∈E
    Sink receives from branch j at t with nodes on j active on t and t + 1
    nj ← nj − 1
    for all nodes that are active on t do
    B ← B + 1
    if B ≥ W then
        if S = Tr then transmit a packet; S ← Rx
    if S = Rx then receive a packet; S ← Tr
    end if
    end for
    t ← t + 1
end while
```



(b) (c)

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

Proof: Let ni symbolize the figure of nodes in branch i. Order the branches in non-increasing order of their branch sizes; let this order be nk ≥ nk−1 ≥ ... ≥ n1. suppose all the interfering links are eliminate by utilizing frequent channels. Since no node can receive various packets in a single slot, N is a minor lower bound to obtain all the packets originated in the network. Next, believe branch k that has the maximum number of children. The root of this limb has to transmit nk packets, and the children of this root have to forward nk − 1 packets in total. Due to the half-duplex nature, time slots Assigned to its children. hence, in total we need nk +(nk − 1) = 2nk − 1 distinct time slots.

bound which was intended as 3N for general networks and max(3nk − 3, N) for tree networks, where the only warning We should note that, this bound is lesser than the existing factor is the half-duplex transceivers. Gandham et al. [1] version of the algorithm is max(3nk − 1, N) which is 2 showed that the numeral of time slots required by the original time slots more than the lower bounce using a single channel. Compute schedules with a length of max(2nk − 1, N), which We now show that a adapted version of the algorithm can 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(2nk − 1, N).

Proof: The idea of the proof is based on that given in [1]. Let ni represent the number of nodes in branch i. Order the branches in non-increasing order of their branch sizes; let this

From Theorem 4, we discern that it takes at least $2n_k - 1$ slots $n_k - 1$ slots to accept packets from other branches. Since the total number of packets in the other kindling is at most $n_k - 1$, to schedule branch k, out of which the drop can use at most the schedule extent is no more than $2n_k - 1$.

k i=1 i k

If nk = 1, each of the other branches can have at most one node because k is the biggest branch. Since the algorithm schedules the most laden branch in every time slot, in total it will take N slots. Otherwise, if all the branches have equal size, the same provision repeats, and the algorithm schedules the branches in successive slots requiring at most nk · k = N time slots in total.

For all extra cases, we use initiation as follows. suppose that the algorithm uses N time slots for convergecast when the most burdened branch on the network has M nodes. Next, believe a network where the most loaded branch has M+1 nodes such that nk = M + 1. The algorithm schedules branch k and and the next most laden branch in the first and second time slots. At the third time slot the branch k has M packets absent. If k is still the most loaded branch then according to our supposition, the remaining packets in the network can be scheduled in N − 2 time slots. for that reason, the comprehensive convergecast can be accomplished in N time slots.

If more than two branches in the network have M +1 nodes, M +1 packets left. presume l branches have M +1 nodes such that l ≤ k and l > 2. Since the algorithm schedules the most then at the third time slot the most loaded branch will still have loaded branch first, at the (l + 1)th time slot the most loaded branch will have M packets. According to the supposition it will take N−l time slots to schedule the outstanding packets. Therefore, the convergecast can be completed in N time slots.

II. IMPACT OF ROUTING TREE

As emphasized in [7], routing trees that permit more parallel transmissions do not forever result in small schedule length. For occasion, given a network, the schedule length would be N with a star topology whereas it would be 2N − 1 with a line topology, pretentious there interference links are removed. The structure of the routing tree also plays an significant role on the schedule length. According to Theorem 4, the routing tree should be constructed with unbiased number of nodes on branches, such that 2nk − 1 < N . In this segment, we explore the possibilities of constructing such trees.

THEOREM 6: The subsequent problem is NP-complete. Given an arbitrary graph G, can we construct a tree T on G, such that nk ≤ N+1 ?
2

Such a tree manufacture is defined as “Capacitated Minimal Spanning Tree Problem” and is proved to be NP-complete [6]. Given the stiffness 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 definite as a appraise to grow a branch outwards a node [9] and such an information is very significant at tree construction to decide which nodes and branches to procedure first.

We suggest a heuristic algorithm that considers such growth potential and breaks down the tree construction instrument into two parts. First every node collects information about the potential branches that it can attach to. preliminary from the sink, all the nodes broadcast information about their hop-count to the sink and the probable branch id's (direct neighbors of the sink become the roots of the brushwood and branch id is the id of the root of a branch). In the moment part the tree is constructed as shown in Algorithm 3. The erratic *growth set*, *GS* of a node *n* includes *n* and the neighbor nodes that are not yet related to the tree. likewise, *growth set* of a branch includes the unconnected neighbors of the nodes that are already on the branch. *Weight* of a branch is the numeral of nodes that are already connected to the branch. *BA* represent 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 associated. Next, the node with the major growth set should be added to tree via the branch with the smallest 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 example consider a situation in Fig. 4. Node 3 is processed (if the cardinality of the *increase 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 process, again branch B1 should be selected considering only the weights of the branches. However, if node 4 also connect to the branch B1 the nodes 8,9 and 10 have only access to the branch B1 and branch B1 will be more packed 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, opening 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*.

though, this basic algorithm tries to keep the numeral of nodes on each branch as smallest, an additional harmonizing may still be needed. We use the alteration 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.

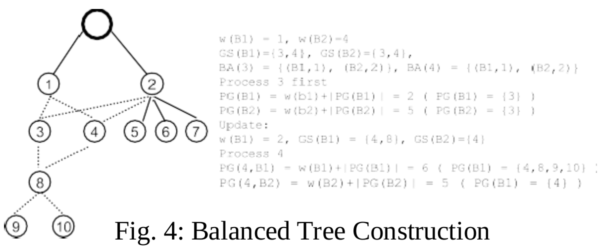


Fig. 4: Balanced Tree Construction

Algorithm 3 Capacitated Minimal Spanning Tree [6]

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: $PG(n, b) \leftarrow \varphi$ 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**

EVALUATION

We use a imitation based advance using Matlab to evaluate the impact of different mechanisms on the scheduling presentation. Nodes are randomly deployed over the area.

Terrain proportions are varied between 20 × 20 and 300 × 300 m^2 to simulate dissimilar levels of density whereas the number of nodes is kept as 100 (the node with id "1" is always repeat the simulations 1000 epoch. selected as the sink node). For different parameter settings, we gation with a path loss exponent $\alpha = 3.5$ which is a typical We use an exponential path loss model for signal propavalue for internal environments. We reproduce the behavior of the CC2420 radio which is used on the Telosb and Tmote sensor mote platform. The transmission power can be adjusted between -24dBm and 0dBm over 8 different levels. SINR on 16 different frequency channels. threshold is β=-3dB.

A. Impact of Power Control

In this segment we evaluate the brunt of transmission power control on the scheduling performance. We scrutinize two cases: nodes transmit with the greatest communication power and nodes adjust their transmission power according to the power control algorithm which is explained in Section III-A. The results are accessible in Fig. 5. The x-axis shows the length of a square area. The y-axis shows the numeral of time slots necessary to schedule all the transmissions in the network. dissimilar lines display the domino effect with different path loss exponents $\alpha = 3.0, 3.5, 4.0$ to converse the impact of $L > 200$, $\alpha = 4.0$ since it's hardly possible to produce physical layer parameters. We cannot provide the results for connected trees.

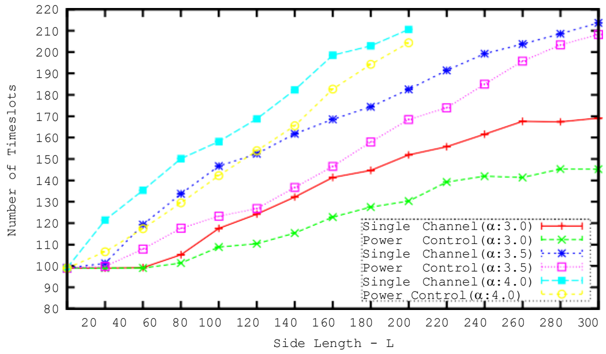


Fig. 5: Joint Scheduling and Transmission Power Control necessary number of time slots, i.e. the schedule length, Increases as the network gets sparser. One would guess the other way around since in sparser deployments the reuse of the time slots should be superior which would result in a lesser schedule length. However, as the network gets sparser, the numeral of nodes that can directly reach the sink decrease such that the packets have to be relayed over multi-hops. In this case more packets have to be scheduled compare with scheduling packets in a single-hop setting. In the simulation we observe that the number of packets to be scheduled increase faster than the recoup ratio. In the densest setting ($L =$ 20), where all 99, equal to the number of transmitting nodes in the network. the nodes can straight reach the sink, the schedule length is If the nodes adjust the level of transmission power, we Observe that the schedule length is smaller since some interfer- ence 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 construct as shortest path number of nodes on a branch such that $2n_k - 1 > N$. though, when $\alpha \geq 3.5$ the transmission power control approach cannot spanning trees and the limiting factor is due to the maximum always eliminate the interference. In this case, the generated $\alpha = 3.5$ and 23m with $\alpha =$ 4.0 while it is around 68m with α = 3.0). In sparse scenario, the nodes cannot reduce networks are sparser (transmission range is around 37.5m with the transmission power further than the greatest level since level which is $-95dBm$. particularly in sparser deployments, $L \geq 200$, the results are similar either the nodes transmit with the transceiver cannot decode the signals below the sensitivity mid-sparse deployments ($60 \leq L \leq 180$) the discrete power the maximum power or adjust the power level. Moreover, in levels (8 levels) and the limited range of the broadcast power limits the nodes to adjust their transmission power.

Impact of Receiver Based Scheduling and Routing Trees

In this segment we initiate the results on the performance of receiver-based frequency and time scheduling process which is explained in Section III-B. Fig. 6 present the results with the x-axis showing the duration of a side of the square deployment area and the y-axis showing the schedule length. Different lines show the results when dissimilar number of channels are available and the last line shows the schedule length when the routing tree is constructed with balanced

numeral of nodes on each branch, such that $2n_k - 1 < N$. When the number of available channels increases, we monitor a reduction on the schedule length. though, when the number of channels is 6 or higher, the schedule length cannot be condensed any more since the interference is eliminated and the preventive factor is due to the half-duplex operation of the sink node.

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

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

Compare the upper bound $\Delta(G') + 1$ on f as per Theorem 3*1) Bounds on the number of frequencies:* In Fig. 7, we 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.

when the network is very dense $L < 40$, since the trans. The number of required frequencies is originally very low missions cannot be scheduled in similar since the number of

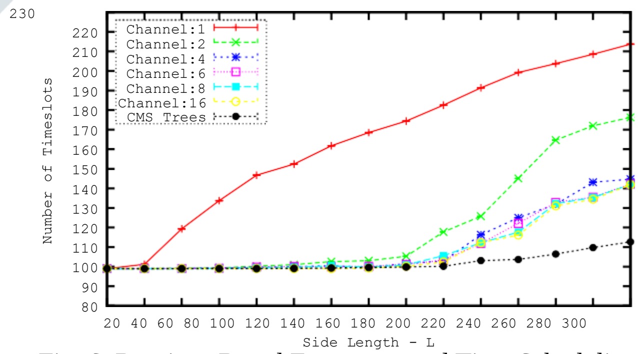
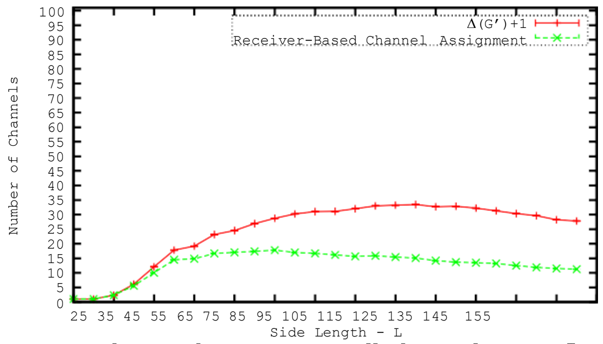
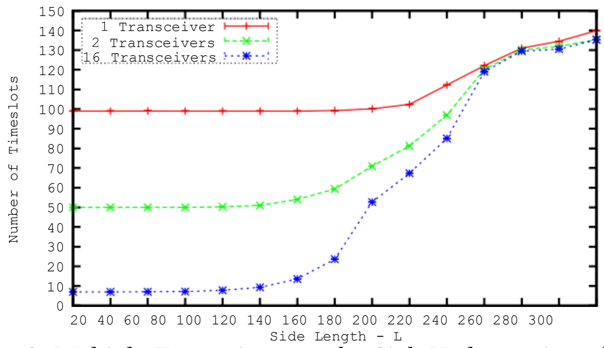


Fig. 6: Receiver-Based Frequency and Time Scheduling





receivers is short. When $L \leq 30$, all the nodes can dyFig. 7: Bounds on the add up to of frequencies reach the sink node, one frequency is satisfactory. As the network gets sparser the numeral of receivers (i.e. parents) increase. in view of that, the level of interference in the network increase and more frequencies are required to support parallel transmissions. However, when $L \geq 80$, the numeral of compulsory channels decrease since the level of interference decreases. Trends of both of the lines are quite analogous. Receiver-based channel obligation actually requires less time slots than the calculated upper bound and the compulsory 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 process with the TMCP protocol [5]. TMCP is a tree based channel assignment method such that dissimilar channels are allocated to each branch of the tree. The goal is to divider the network into multiple subtrees with minimizing the intra-tree interference. It is a greedy algorithm and assigns the channel one by one to the nodes from top-to-bottom on a fat tree. When a node is to be added to a sub tree, the subtree where the node bring the least interference is selected.

After the channel assignment, the time slots are assigned to the nodes with the equal method as explained in Section V. Fig. 8 presents the comparison between the receiver-based channel obligation and the TMCP protocol with 2 and 16 channels with the x-axis showing the side length of the consumption area and the y-axis showing the schedule length. We use straight path routing trees (not the balanced trees) with the receiver-based channel assignment method for a fair comparison. Receiver-based channel assignment performs ap-

Fig. 9: Multiple Transceivers on the Sink Node proximately the same only with 2 channels while TMCP uses 16 channels since the process allows more nodes to transmit in corresponding. For instance while a node is in receipt of from its children, the parent of this node can transmit at the same time which would not be possible if they communicate on the same channel.

Multiple Transceivers at the sink node, multiple sinks

In this segment we examine the schedule length when the sink is equipped with multiple half-duplex transceivers such that the transceivers can accept in parallel from dissimilar 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 comparative to the number of available especially when $L > 220$, there is almost no lessening on the transceivers at the sink node. However, in sparser scenarios, achievable schedule length if the sink has a single transceiver or multiple transceivers. In sparser scenarios, the numeral 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 greatest 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 calculate the schedule length if there are multiple sinks from 1 to 16. Sinks are erratically deployed as well as the sinks deployed within the network. We vary the number of nodes. Fig. 10 shows the results. Compared with the results in Fig. 9, we can achieve a reduction on the schedule length also

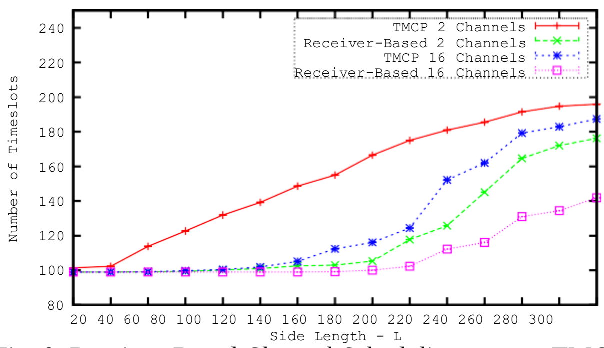


Fig. 8: Receiver-Based Channel Scheduling versus TMCP

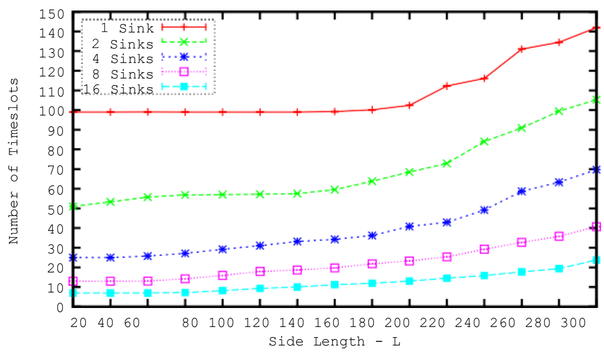


Fig. 10: Multiple Sinks

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

VIII. Related Work

A strongly related study by Gandham *et al.* [1] focuses on ruling a TDMA schedule that minimize the total time required to complete a convergecast in WSN. even though, they address the same problem as ours, we center on methods that can decrease the schedule length by eliminating the limitations due to interference, half-duplex transceiver and steering tree where they are interested in showing the boundaries on the schedule length with dissimilar network organizations. in addition, we progress 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 boundaries 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 pretentious that the data is aggregated before relayed towards the sink node. In the aggregated-convergecast work, likewise we discussed the impact of transmission power manage and using multiple channels on the schedule length. though, the presented results were based on simulations. In this work, we study the hardness of finding the smallest schedule length on arbitrary graphs, the inflexibility of the frequency assignment problem and the stability of the tree construction where the $\max(2nk - 1, N)$ constraint is met. In the aggregated-convergecast difficulty, the constraints were much simpler and the nature of the problem in terms of the attainable schedule length was totally different. The schedule length in the former was found to be bounded by the highest degree of a tree and in the current problem it is found and prove to bounded by the number of nodes in the network if a suitable tree structure method is used. Scheduling with aggregation was also address in [12] using orthogonal codes, allowing for the impact of routing tree and in [13] by using non-linear transmission power control mechanisms.

The use of manifold frequency channels has been extensively studied for both cellular and ad hoc networks. In the WSN domain, there exist recent studies that exploit 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 obligation which is suitable for data compilation on a tree topology.

9

Conclusions

We have explored fast convergecast scheduling in wireless sensor networks where the nodes communicate on a TDMA schedule and the purpose 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 eradicate those boundaries. We found that while power control is helpful in dropping the schedule length, scheduling transmissions on dissimilar frequency channels is more efficient in mitigating the effects of interference. Once the interference is eliminate, we length is lower-bounded by $\max(2n_k - 1, N)$, where n_k is the proved that with half-duplex radios the attainable schedule highest number of nodes on a sub tree 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 unbiased 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

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.

A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mit- tal, 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.

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.

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.

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.

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

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

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.

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.

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.

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.

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

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

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

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