IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 2, April-May, 2013
ISSN: 2320 - 8791
www.ijreat.org

FAST AGGREGATED CONVERGECAST ENABLED TREE BASED WIRELESS SENSOR NETWORKS

S S Nandhini[1], R Saranya[2] and V Sumathy[3]

[1, 2, 3] Department of Computer Science and Engineering, Kingston Engineering College
Vellore, Tamil Nadu

Abstract

The main idea is to collect data faster using a Wireless Sensor Networks which is organised in a tree form.Wireless Sensor Networks (WSNs) consists of numerous small sensors. These sensors are wirelessly connected to each other for performing same task collectively such as monitoring weather conditions or specifically parameters like temperature, pressure, sound and vibrations etc.We explore and evaluate a number of different techniques using realistic simulation models under the many-to-one communication paradigm known as convegecast.We also construct degree constrained spanning tree and capacitated minimal spanning tree. We use BFS TimeSlot assignment and Local TimeSlot assignment to collect data as faster as possible in a tree based topology. Lastly we evaluate the impact of different interference and channel models on the schedule length.

Keywords - *Convergecast, TDMA scheduling, multiple channels, power-control, routing trees.*

I.INTRODUCTION

In recent years, wireless sensor network, has received much attention of researchers as the technological advancement has made these low power devices very cost effective. Wire-less sensor networks consist of numerous tiny low-power devices capable of performing sensing and communication tasks collectively.

Wireless sensor networks were first deployed for military applications. Gradually researchers found them to be very useful in applications like weather monitoring, habitat monitoring, agriculture, industrial applications, and recently smart homes and kindergartens. Wireless sensor network is an ad hoc network and being distributed in nature, time synchronization becomes a critical part of its functioning. Every small sensor consists of an embedded processor, memory and radio. Precise and synchronized time is needed for several reasons. For example, an accurate and synchronized time is necessary to determine the right chronological order of events as in target tracking. A lack of synchronization may lead toincorrect time stamping and misinterpretation of the readings.

For a wired network, two methods of time synchronization are most common. Network Time Protocol (NTP) and Global Positioning System (GPS) are both used for synchronization.Neither protocol is useful for wireless sensor synchronization.

Both require resources not available in wireless networks. The Network Time Protocol requires an extremely accurate clock, usually a server with an atomic clock. The client computer wanting to synchronize with the server will send a UDP packet requesting the time information. The server will then return the timing information and as a result the computers would be synchronized. Because of many wireless devices are powered by batteries, a server with an atomic clock is impractical for a wireless network. GPS requires the wireless device to communicate with satellites in order to synchronize. This requires a GPS receiver in each wireless device. Again because of power constraints, this is impractical for wireless networks. Also sensor networks consist of inexpensive wireless nodes. A GPS receiver on each wireless node would be expensive and therefore unfeasible. The time accuracy of GPS depends on how many satellites the receiver can communicate with at a given time. This will not always be the same, so the time accuracy will vary. Furthermore Global Positioning System devices depend on line of sight communication to the satellite, which may not always be available where wire-less networks are deployed.

The constraints of wireless sensor networks do not allow for traditional wired network time synchronization protocols. Wireless sensor networks are limited to size, power, and complexity. Neither the Network Time Protocol nor GPS were de-signed for such constraints.

For a wireless sensor network, there are three basic types of synchronization methods. The first is relative timing and is the simplest. It relies on the ordering of messages and events. The basic idea is to be able to determine if event 1 occurred before event 2. Comparing the local clocks to determine the order is that is needed.

Evaluation of Power Control under Realistic Setting:

It was shown recently that under the idealized setting of unlimited power and continuous range, transmission power control can provide an unbounded improvement in the asymptotic capacity of aggregated convergecast.

In this work, we evaluate the behavior of an optimal power control algorithm under realistic settings considering the limited discrete power levels available in today's radios. We find that for moderate size networks of 100 nodes power control can reduce the schedule length by 15 − 20%.

- Evaluation of Channel Assignment Methods:

Using extensive simulations, we show that scheduling transmissions on different frequency channels is more effective in mitigating interference as compared to transmission power control. We evaluate the performance of three different channel assignment methods: (i) *Joint Frequency and Time SlotScheduling* (JFTSS), (ii) *Receiver-Based Channel Assignment*(RBCA)] and (iii) *Tree-Based ChannelAssignment (TMCP)*]. These methods consider thechannel assignment problem at different levels: the link level, node level, or cluster level. We show thatfor aggregated convergecast, TMCP performs betterthan JFTSS and RBCA on minimum-hop routingtrees, while performs worse on degree-constrainedtrees. For raw-data convergecast, RBCA and JFTSSperform better than TMCP, since the latter suffersfrom interference inside the branches due to concurrenttransmissions on the same channel.

- Impact of Routing Trees:

We investigate the effect ofnetwork topology on the schedule length, and showthat for aggregated convergecast the performancecan be improved by up to 10 times on degreeconstrainedtrees using multiple frequencies as comparedto that on minimum-hop trees using a singlefrequency. For raw-data convergecast, multi-channel scheduling on capacitated minimal spanning treescan reduce the schedule length by 50%.

- Impact of Channel Models and Interference:

Underthe setting of multiple frequencies, one simplifyingassumption often made is that the frequenciesare orthogonal to each other. We evaluate this assumptionand show that the schedules generated may not always eliminate interference, thus causingconsiderable packet losses. We also evaluate andcompare the two most commonly used interferencemodels: (i) the graph-based *protocol model*, and (ii)the SINR (Signal-to-Interference-plus-Noise Ratio)based *physical model*of synchronization methods. The first is relative timing and is the simplest. It relies on the ordering of messages and events. The basic idea is to be able to determine if event 1 occurred before event 2.

Comparing the local clocks to determine the order is all that is needed. Clock synchronization is not important. The next method is relative timing in which the network clocks are independent of each other and the nodes keep track of drift and offset. Usually a node keeps information about its drift and offset in correspondence to neighboring nodes. The nodes have the ability to synchronize their local time with another nodes local time at any instant. Most synchronization protocols use this method. The last method is global synchronization where there is a constant global timescale throughout the network. This is obviously the most complex and the toughest to implement. Very few synchronizing algorithms use this method particularly because this type of synchronization usually isnot necessary.

II. TDMA SCHEDULING ON CONVEGECAST

Periodic Aggregated Converge cast.

Data aggregation is a commonly used technique in WSN that can eliminate redundancy and minimize the number of transmissions, thus saving energy and improving network lifetime. Aggregation can be performed in many ways, such as by suppressing duplicate messages; using data compression and packet merging techniques; or taking advantage of the correlation in the sensor readings

We consider continuous monitoring applications where perfect aggregation is possible, i.e., each node is capable of aggregating all the packets received from its children as well as that generated by itself into a single packet before transmitting to its parent. The size of aggregated data transmitted by each node is constant and does not depend on the size of the raw sensor readings.

III. ASSIGNMENT OF TIMESLOTS

Given the lower bound $\Delta(T)$ on the schedule length in the absence of interfering links, we now present a time slot assignment scheme in Algorithm 1, called BFSTIMESLOTASSIGNMENT, that achieves this bound. In each iteration of BFS TIMESLOTASSIGNMENT , an edge e is chosen in the Breadth First Search (BFS) order starting from any node, and is assigned the minimum time slot that is different from all its adjacent edges respecting interfering constraints. Note that, since we evaluate the performance of this algorithm also for the case when the interfering links are present, we check for the corresponding constraint in line 4; however, when interference is eliminated this check is redundant. The algorithm runs in $O(|ET|2)$ time and minimizes the schedule length when there are no interfering links. All the interfering links removed, and so the network is scheduled in 3 time slots.

Algorithm1

```
BFS-TIMESLOTASSIGNMENT
Input: $T = (V, ET)$
while$ET$ _= $\varphi$ do
    $e \leftarrow$ next edge from $ET$ in BFS order
    Assign minimum time slot $t$ to edge $e$ respecting adjacency and interfering constraints
    $ET \leftarrow ET \setminus \{e\}$
end while
```

AlthoughBFS-TIMESLOTASSIGNMENT may not be an approximation to ideal scheduling under the physical interference model, it is a heuristic that can achieve the lower bound if all the interfering links are eliminated. Therefore, together with a method to eliminate interference the algorithm can optimally schedule the network.

LOCAL-TIMESLOTASSIGNMENT

This runs*locally* by each node at every time slot. The keyidea is to: (i) schedule transmissions in parallel alongmultiple branches of the tree, and (ii) keep the sink busyin receiving packets for as many time slots as possible. Because the sink can receive from the root of at mostone top-subtree in any time slot, we need to decide which top-subtree should be made active. We assume that the sink is aware of the number of nodes in each top-subtree. Each source node maintains a buffer and its associated state, which can be either *full* or *empty*depending on whether it contains a packet or not. Our algorithm does not require any of the nodes to store more than one packet in their buffer at any time. We initialize all the buffers as full, and assume that the sink's buffer is always full for the ease of explanation.

Algorithm2

```
LOCAL-TIMESLOTASSIGNMENT
node.buffer = full
if{node is sink} then
    Among the eligible top-subtrees, choose the one with the largest number of total (remaining) packets, say top-subtree$i$
    Schedule link ($root(i)$, $s$) respecting interfering constraint
else
    if{node.buffer == empty} then
        Choose a random child $c$ of node whose buffer is full
        Schedule link ($c$, node) respecting interfering constraint
        $c$.buffer = empty
        node.buffer = full
    end if
end if
```

The first block of the algorithm in lines 2-4 gives the scheduling rules between the sink and the roots of the top-subtrees. We define a top-subtree to be *eligible* if its root has at least one packet to transmit. For a given time slot, we schedule the root of an eligible topsubtree which has the *largest* number of total (remaining) packets. If none of the top-subtrees are eligible, the sink does not receive any packet during that time slot. Inside each top-subtree, nodes are scheduled according to the rules in lines 5-12. We define a subtree to be *active* if there are still packets left in the subtree (excluding its root) to be relayed. If a node's buffer is empty and the subtree rooted at this node is active, we schedule one of its children at random whose buffer is not empty. Our algorithm guarantees that in an active subtree there will always
be at least one child whose buffer is not empty, and so whenever a node empties its buffer, it will receive a packet in the next time slot, thus emptying buffers from the bottom of the subtree to the top.

Transmission Power Control

In wireless networks, excessive interference can be eliminated by using transmission power control i.e., by transmitting signals with just enough power instead of maximum power. To this end, we evaluate the impact of transmission power control on fast data collection using discrete power levels, as opposed to a continuous range where an unbounded improvement in the asymptotic capacity can be achieved by using a non-linear power assignment. The algorithm proposed by El Batt*et al.* is a cross layer method for joint scheduling and power control and it is an optimal distributed algorithm to improve the throughput capacity of wireless networks. The goal is to find a TDMA schedule that can support as many transmissions as possible in every time slot. It has two phases:

1)scheduling and
2) power control that are executed
at every time slot.

First the scheduling phase searches for a *valid transmission schedule*, i.e., largest subset of nodes, where no node is to transmit and receive simultaneously, or to receive from multiple nodes simultaneously. Then, in the given valid schedule the power control phase iteratively searches for an *admissible schedule* with power levels chosen to satisfy all the interfering constraints. In each iteration, the scheduler adjusts the power levels depending on the current RSSI at the receiver and the SINR threshold according to the iterative rule: *Pnew=βSINR.Pcurrent*.

IV. TREE-BASED MULTI-CHANNEL PROTOCOL (TMCP)

TMCP is a greedy, tree-based, multi-channel protocol for data collection applications [8]. It partitions the network into multiple subtrees and minimizes the *intratree* interference by assigning different channels to the nodes residing on different branches starting from the top to the bottom of the tree.The figure shows which is scheduled according to TMCP for aggregated data collection. Here, the nodes on the leftmost branch is assigned frequency *F*1, second branch is assigned frequency *F*2 and the last branch is assigned frequency *F*3 and after the channel assignments, time slots are assigned to the nodes with the BFSTimeSlotAssignment algorithm. The advantage of TMCP is that it is designed to support convergecast traffic and does not require channel switching. However, contention inside the branches is not resolved since all the nodes on the same branch communicate on the same channel.

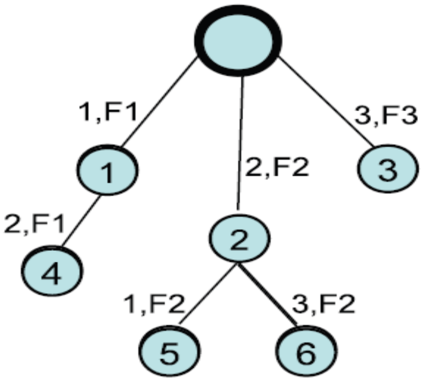

FIG.1

Tree based multi-channel protocol

V. RECEIVER-BASED CHANNEL ASSIGNMENT (RBCA)

We proposed a channel assignment method called RBCA where we statically assigned the channels to the receivers (parents) so as to remove as many interfering links as possible. In RBCA, the children of a common parent transmit on the same channel. Every node in the tree, therefore, operates on at most two channels, thus avoiding pair-wise, per-packet, channel negotiation overheads. The algorithm initially assigns the same channel to all the receivers. Then, for each receiver, it creates a set of interfering parents based on SINR thresholds and iteratively assigns the next available channel starting from the most interfered parent (the parent with the highest number of interfering links). However, due to adjacent channel overlaps, SINR values at the receivers may not always be high enough to tolerate interference, in which case the channels are assigned according to the ability of the transceivers to reject interference. We proved approximation factors for RBCA when used with greedy scheduling. Initially allnodes are on frequency *F*1. RBCA starts with the mostinterfered parent, node 2 in this example, and assigns *F*2.Then it continues to assign *F*3 to node 3 as the secondmost interfered parent. Since all interfering parents areassigned different frequencies sink can receive on *F*1.

VI. CONCLUSION AND FUTURE ENHANCEMENT

In this paper, we studied fast convergecast in WSN where nodes communicate using a TDMA protocol to minimize the schedule length. We found that while transmission power control helps in reducing the schedule length, multiple channels are more effective. We also observed that node-based (RBCA) and link-based (JFTSS) channel assignment schemes are more efficient in terms of eliminating interference as compared to assigning different channels on different branches of the tree (TMCP). Once interference is completely eliminated, we proved that with half-duplex radios the achievable schedule length is lower-bounded by the maximum degree in the routing tree for aggregated convergecast, and by $\max(2nk - 1, N)$ for raw-data convergecast. Through extensive simulations, we demonstrated up to an order of magnitude reduction in the schedule length for aggregated, and a 50% reduction for raw-data convergecast. In future, we will explore scenarios withvariable amounts of data and implement and evaluate the combination of the schemes considered.

VII.FUTURE ENHANCEMENT

Most of the surveyed algorithms considerFixed traffic patterns, i.e, every node generate a fixed number of packets in each data collection cycle. In a real scenario, some nodes may have lot of packets that require more than one time slot per frame, while some others may not have any data to send in a time slot, thus wasting bandwidth. It will be interesting to explore the performance in such scenarios with random packet arrivals and combining the solutions of TDMA scheduling with rate allocation algorithms, especially in applications where high data rates are necessary. Another possibility is to investigate different levels of aggregation, i.e, how much of the data received from the children is forwarded to the parent node.

VIII. REFERENCES

[1] S. Gandham, Y. Zhang, and Q. Huang, "Distributed time-optimal scheduling for convergecast in wireless sensor networks," *ComputerNetworks*, vol. 52, no. 3, pp. 610–629, 2008.
[2] K. K. Chintalapudi and L. Venkatraman, "On the design of mac protocols for low-latency hard real-time discrete control applications over 802.15.4 hardware," in *IPSN '08*, pp. 356–367.
[3] I. Talzi, A. Hasler, S. Gruber, and C. Tschudin, "Permasense: investigating permafrost with a wsn in the swissalps," in *EmNets07*, Cork, Ireland, pp. 8–12.
[4] S. Upadhyayula and S. Gupta, "Spanning tree based algorithms for low latency and energy efficient data aggregation enhanced convergecast (dac) in wireless sensor networks," *Ad Hoc Networks*, vol. 5, no. 5, pp. 626–648, 2007.
[5] T. Moscibroda, "The worst-case capacity of wireless sensor networks," in *IPSN '07*, Cambridge, MA, USA, pp. 1–10.
[6] T. ElBatt and A. Ephremides, "Joint scheduling and power control for wireless ad-hoc networks," in *INFOCOM '02*, Jun, pp. 976–984.
[7] O. D. Incel and B. Krishnamachari, "Enhancing the data collection rate of tree-based aggregation in wireless sensor networks," in *SECON '08*, San Francisco, CA, USA, pp. 569–577.