July 2012 IJMIE Volume 2, Issue 7 ISSN: 2249-0558

STRATEGIES OF DATA COLLECTION IN TREE-BASED WIRELESS SENSOR NETWORKS

NITA V. JAISWAL*

VISHAL S. DHOLE*

PROF. D.M.DAKHANE**

HARSHAL N. DATIR***

ABSTRACT:

Data collection is a fundamental operation in wireless sensor networks where sensor nodes measure attributes about a phenomenon of interest and transmits their readings to a common base station. We classify the algorithms according to their common design objectives, identifying the following four as the most fundamental and most studied with respect to data collection in WSNs: (i) minimizing schedule length, (ii) minimizing latency, (iii) minimizing energy consumption. We explore and evaluate a number of different techniques using realistic simulation models under the many-to-one communication paradigm known as convergecast. In sensor networks data are transferred from the sensor nodes to a few central data collectors.

Index Terms—*Convergecast, TDMA Scheduling algorithm., multiple channels*

---

* M.E.(first year in computer Engg.), Sipna College of Engg. & Technolgy, Amravati.

** Assist Professor (Computer Sci &Engg), Sipna College of Engg. & Technology, Amravati

*** Lecturer in Computer Science & Enginnering and Information Technology Dept, Sipna College of Engg. & Technology, Amravati.

INTRODUCTION

Convergecast, namely the collection of data from a set of sensors toward a common sink over a tree based routing topology, is a fundamental operation in WSN[1]. Many sensor applications require broadcasting and convergecasting. Data collection from a set of sensors to a common sink over a tree-based routing topology is a fundamental traffic pattern in WSNs. This many-to-one communication paradigm in which data flows from many nodes to a single node is known as convergecast.

In TDMA-based scheduling algorithms for data collection in sensor networks. We first classify the algorithms based on their common objectives, and then identify different design constraints and assumptions, and provide taxonomy according to these metrics. We study two types of data collection: (i) aggregated convergecast where packets are aggregated at each hop, and (ii) raw-data convergecast where packets are individually relayed toward the sink. Aggregated convergecast is applicable when a strong spatial correlation exists in the data, or the goal is to collect summarized information such as the maximum sensor reading. Raw data convergecast, on the other hand, is applicable when every sensor reading is equally important, or the correlation is minimal. We study aggregated convergecast in the context of continuous data collection, and raw data convergecast for one-shot data collection[2].

For periodic traffic, it is well known that contention free (MAC) protocols such as TDMA are better fit for fast data collection, since they can eliminate collisions and retransmissions and provide guarantee on the completion time as opposed to contention-based protocols[1].However, the problem of constructing conflict free TDMA schedules even under the simple graph-based interference model has been proved to be NP-complete.

We start by identifying the primary limiting factors of fast data collection, which are: (i) interference in the wireless medium, (ii) half-duplex transceivers on the sensor nodes, and (iii) topology of the network. Then, we explore a number of different techniques that provide a hierarchy of successive improvements, the simplest among which is an interference-aware, minimum-length, TDMA scheduling that enables spatial reuse. Every wireless sensor network consist design constraints, which is as follws:-

Design Constraints

- Communication and Interference Models
- Fast Data Collection
- Network Topology
- Sensor Deployment
- Buffer Size
- Implementation Method
- Granularity of Assignments

This journal is based on only on data collection design constraint.

RELATED WORK

The scheduling problem with the objective to minimize the number of time slots required to complete convergecast (known as the schedule length) has been studied in [4][8] for aggregated data, and in[1][2] for raw data. Most of the algorithms aim to maximize the number of concurrent transmissions and enable spatial reuse by devising strategies to eliminate interference.

The aggregated data capacity as well as the notion of worst-case capacity, which concerns the question of how much information can each node transmit to the sink regardless of the networks topology, are investigated for typical worst-case structures, such as chains. In case of raw data convergecast, the scheduling problem using a single-channel TDMA protocol. They describe an integer linear programming formulation and propose a distributed scheduling algorithm that requires at most time slots for general networks, where is the number of nodes. In the domain of sensor networks, however, there exist fewer works using multiple channels.

Several optimization problems arising in the design of communication networks can be modeled as constructing optimal network topologies in particular spanning trees. The Minimum-Degree Spanning Tree problem, where the goal is to construct a spanning tree such that its maximum node degree is minimized, is NP-hard on general graphs[5].

Fast data collection with the goal to minimize the schedule length for aggregated convergecast. Raw-data convergecast has been a distributed time slot assignment scheme to minimize the TDMA schedule length for a single channel.For raw-data convergecast, a time-optimal, energy-efficient, packet scheduling algorithm with periodic traffic from all the nodes to the sink. Once interference is eliminated, their algorithm achieves the bound that they briefly mention a 3-coloring channel assignment scheme, and it is not clear whether the channels are frequencies, codes, or any other method to eliminate interference. Moreover, a simple interference model where each node has a circular transmission range and cumulative interference from concurrent multiple senders is avoided. Different from their work, we consider multiple frequencies and evaluate the performance of three different channel assignment methods together with evaluating the effects of transmission power control using realistic interference and channel models. A TDMA based MAC protocol for high data rate WSNs. Maximizing the throughput of convergecast by finding a shortest-length, conflict-free schedule is where greedy graph coloring strategy assigns time slots to the senders and prevents interference.

TDMA SCHEDULING OF CONVERGECASTS

1. Algorithms on Minimizing Schedule Length

The TDMA-based scheduling algorithms that identify minimizing the schedule length as their primary objective. We first describe works that pertain to raw-data convergecast, and then focus on aggregated convergecast. Since packets are aggregated at each hop in aggregated convergecast, the number of packets transmitted and delivered to the sink is substantially lower than that of raw-data convergecast.

1.1 Raw-data convergcast

For raw-data convergecast, finding a minimum-length, conflict free assignment of time slots, such that every packet generated by a node reaches the sink, is fundamental to efficient network operations.

Assuming the protocol interference model, they propose optimal centralized algorithms for special network topologies, such as line, multi-line, and tree networks, for both Omni directional and directional antennas. For packet distribution in line networks, where the sink sends p(i) ≥ 0 packets to node i which is i hops away, the basic idea is to first transmit packets destined to the furthest node, then packets for the second furthest node, and so on, as quickly as possible respecting channel reuse constraints. Nodes between the sink and a packet's destination are required to forward that packet as soon as it arrives. In particular, a transmission from node i to i+1 occurring at time slot j for the distribution problem corresponds to a transmission from node i+1 to i in time slot T−j+1 for the convergecast problem. Here, N is the total number of nodes and T is the minimal schedule length which, for omnidirectional antennas, is given by:

$$T = \max_{1\leq i\leq N} (i-1+ p(i)+2p(i+1)+3 \textstyle\sum_{j\geq i+2}^{N} P(j)),$$

and for directional antennas is given by:

$$T =\max_{1\leq i\leq N-1} (i-1+p(i)+2 \textstyle\sum_{j\geq i+2}^{N} P(j))$$

prove that the problem of minimizing the schedule length is NP-complete by reducing it from the Graph Coloring problem. The scheduling difficulty arises since many subsets of non-conflicting nodes are candidates for transmission in each time slot, and the subset chosen in one slot affects the number of transmissions in the next slot. This is due to the fact that some eligible nodes may not have any packet to transmit because of the subset selected in the previous slot. When a graph-based interference model is used, a conflict-free schedule can be found by coloring a conflict graph. A conflict graph is one in which every node represents an edge in the original graph and two nodes are connected if their corresponding edges interfere in the original graph, i.e., give rise to primary or secondary conflicts.

A Virtual Node Expansion-based approach that also uses graph coloring to find a minimum-length, conflict-free schedule where every node generates a single packet in each frame. Once the expanded conflict graph is constructed, an approximate coloring algorithm is used to find a time slot assignment. The coloring algorithm works by finding a vertex with the least degree and removing it from all its adjacent edges. This is repeated until all the vertices are removed, after which it greedily assigns colors in the reverse order of removing the vertices. This results in a conflict-free schedule for each of the edges in the original graph.

The schedule length achieved by coloring algorithm is no more than 2δ /k+1, where δ is the largest degree in any subgraph of the conflict graph in which every vertex has a degree at least

δ, and k is the maximum size of an independent set in the neighborhood of any node in the conflict graph.

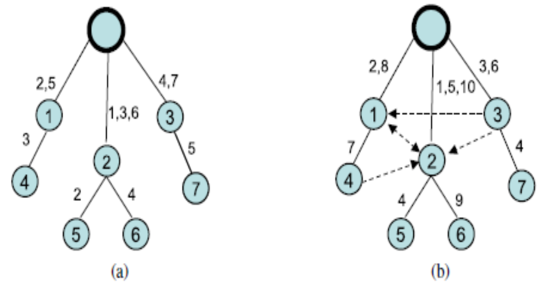
The key idea behind our algorithm, which is formally presented in following Algorithm1 as LOCAL TIME SLOT ASSIGNMENT, is to: (i) schedule transmissions in parallel along multiple branches of the tree, and (ii) keep the sink busy in receiving packets for as many time slots as possible. Each node maintains a buffer and its associated state, which can be either full or empty depending on whether it contains a packet or not. Initially, all the buffers are full because every node has a packet to send.

Algorithm1:-LOCAL TIME SLOT ASSIGNMENT

```
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
```

The first block of the algorithm in lines 2-4 gives the scheduling rules between the sink and the roots of the top-subtrees. A top-subtree TS(r) is defined as one whose root r is a child of the sink, and it is said to be eligible if r has at least one packet to send. For instance, in Fig.1(a), the top-subtrees are {1,4}, {2,5,6}, and {3,7}. For a given time slot, the root of an eligible top-subtree which has the largest number of total remaining packets is scheduled. 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. A subtree is defined to be active if there are still packets left in it (excluding its root) to be relayed. If a node's buffer is empty and the subtree rooted at this node is active, one of its children is scheduled at random whose buffer is not empty. The 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. Fig. 1(a) shows an illustration of the working of the algorithm. In slot 1, since the eligible top-subtree containing the largest number of remaining packets is {2,5,6}, link (2, s) is scheduled and the sink receives a packet from node 2. In slot 2, the eligible top-subtrees are {1,4} and {3,7}, both of which have 2 remaining packets.

We choose one of them at random, say {1,4}, and schedule the link (1, s). Also, in the same time slot since node 2's buffer is empty, it chooses one of its children at random, say node 5, and schedule the link (5,2). In slot 3, the eligible top-subtrees are {2,5,6} and {3,7}, both of which have 2 remaining packets. We choose the first one at random and schedule the link (2, s), and so the sink receives a packet from node 5 (relayed by node 2). We also schedule the link (4,1) in slot 3 because node 1's buffer is empty at this point. This process continues until all the packets are delivered to the sink, yielding an assignment that requires 7 time slots. In this example e, $2n_k - 1 = 5$, and so $\max(2n_k - 1, N) = 7$. In Fig. 1(b), an assignment is shown when all the interfering links are present, yielding a schedule length of 10.



(a) (b)

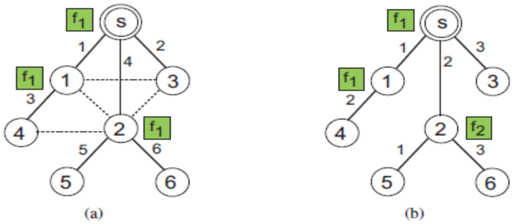
fig:1 Raw-data convergecast using algorithm local-time slot assignment: (a) Schedule length 7 when secondary conflicts are eliminated. (b) Schedule length 10 when secondary conflicts are present.

A similar result of max($2n_k - 1$,N) is also extended it to the case when the nodes have different number of packets to send. Assuming node i generates d_i packets, their proposed algorithm takes max($2\Sigma_{i\epsilon TS}(r_k)\ d_i - d_{rk} + \sum_{i=2}^{k}(dri - 1)N'$) time slots, where $r_k..r_1$ are the roots of the top-subtrees sorted in descending order of the total number of packets generated in it, k is the total number of top-subtrees, and N′ is the total number of packets in the whole network.

1.2 Aggregated Data Convergcast

Aggregated convergecast requires less number of time slots than raw-data convergecast because of the reduced volume of traffic route to the sink. Under this setting, it is assumed that every node generates a single packet at the beginning of every frame and perfect data 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. This means that the size of aggregated data is constant and does not depend on the actual raw sensor readings. Since the goal is to minimize the schedule length, each parent node ideally should wait to receive all data from its children and then aggregate those with its own data before transmitting. Thus, in aggregated convergecast, a node transmits only once per frame and it maintains an intrinsic order of transmission with respect to its children. When the routing tree is not specified as part of the application requirements, the algorithms in this category also construct the routing tree suitable for aggregation and then perform scheduling.

A variant of the aggregated convergecast problem where a parent node need not wait to receive all data from its children within a single frame before transmitting. This is particularly applicable for continuous and periodic monitoring applications that sustain over long durations of time. As explained in the following, the transmission ordering constraint between a parent node and its children within a single frame disappears once a pipeline is established, after which the sink starts receiving aggregated data from all the nodes in the network once every frame.



	Frame 1						Frame 2					
	S1	S2	S3	S4	S5	S6	S1	S2	S3	S4	S5	S6
s	1	2	3	-	-	-	{1,4}	{2,5,6}	3	-	-	-
1	-	-	-	4	-	-	-	-	-	4	-	-
2	-	-	-	-	5	6	-	-	-	-	5	6

(c)

fig:-2 Aggregated convergecast: (a) Schedule length of 6 for single frequency. (b) Schedule length of 3 when multiple frequencies are used to eliminate interference. (c) Node ids from which aggregated data is received by their corresponding parents in each time slot over different frames.

In Fig.2, shows a network of 6 source nodes where the solid lines represent tree edges and the dotted lines represent interfering links. The numbers beside the links represent the time slots at which the links are scheduled to transmit. The frequencies assigned to the receivers of the tree are shown in boxes. The entries in the table list the nodes from which packets are received by their corresponding receivers in each time slot for Fig. 2(a). We note that at the end of frame 1, the sink does not have packets from nodes 5 and 6; however, as the same schedule is repeated, it receives aggregated packets from nodes 2, 5, and 6 in slot 2 of the next frame. Similarly, the sink also receives aggregated packets from nodes 1 and 4 starting from slot 1 of frame 2. The entries {1,4} and {2,5,6} in the table represent single packets comprising aggregated data from nodes 1 and 4, and from nodes 2, 5, and 6, respectively. Thus, a pipeline is established from frame 2, and the sink continues to receive aggregated packets from all the nodes once every 6 time slots. Thus, the minimum schedule length is 6. However, if node 2 is a assigned a different frequency, as shown in Fig.2(b), then the minimum schedule length turns out to be 3.

A number of different techniques that provide a hierarchy of successive improvements, the simplest among which is an interference aware, minimum-length, TDMA scheduling that enables spatial reuse. To achieve multiple frequencies are assumed to be orthogonal, and a Receiver-Based Channel Assignment (RBCA) scheme is proposed where the receivers (i.e.,parents) in the tree are statically assigned different frequencies to eliminate interference. It is shown through extensive simulations that once multiple frequencies are used along with spatial-reuse TDMA, the data collection rate often no longer remains limited by interference, but by the topology of the network. Thus, in the final step, Degree Constrained Trees are constructed that further enhances the data collection rate.

1.3 Degree Constrained Trees

First to construct balanced trees and compare their performance with unbalanced trees. We observe that in both cases the sink often creates a high-degree bottleneck. To overcome this, as described in following Algorithm, by modifying Dijkstra's shortest path algorithm to construct degree constrained trees. Note that constructing such a degree-constrained tree is NP-hard. Each source node i in our heuristic keeps track of the number of its children, C(i), which is initialized to 0, and a hop count to the sink, HC(i), which is initialized to ∞. The algorithm starts with the sink node, and adds a node i' ∉ T at every iteration to the tree such that HC(i') is minimized. It stops when |T| = |V |, or when no more nodes can be added to the tree because the neighbors of all these new nodes have reached the limit on their maximum degree. To illustrate the gains of degree-constrained trees, consider the case when all the N nodes are in range of each other and that of the sink. If the nodes select their parents according to minimum-hop without a degree constraint, then all of them will select the sink, and this will give a schedule length of N. However, if we take limit the number of children per node to 2, then this will result in two subtrees rooted at the sink, and if there are enough frequencies to eliminate interference, the network can be scheduled using only 2 time slots, thus achieving a factor of N/2 reduction in the schedule length.

Algorithm2 DEGREE-CONSTRAINED TREES

```
Input: G (V, E), s, max_ degree
T ← {s}
for all i ∈ V do
    C(i) ← 0; HC(i)←∞
end for
HC(s) ← 0
```

```
7.      while $|T| \neq |V|$ do
8.          Choose $i' \notin T$ such that:
9.  (a) $(i, i') \in E$, for some $i \in T$ with $C(i) < max\_degree-1$
10  (b) $HC(i')$ is minimized
11. $T \leftarrow T \cup \{i'\}$
12. $HC(i') = HC(i) + 1$
13. $C(i) \leftarrow C(i) + 1$
14. if $\forall i \in V$ , $C(i) = max\_degree$ then
15. break
16. end if
17. end while
```

1.3 Assignment of Timeslots

Given the lower bound $\Delta(T)$ on the schedule length in the absence of interfering link, now time slot assignment scheme in Algorithm, called BFS TIME SLOT ASSIGNMENT, that achieves this bound. In each iteration of BFS TIME SLOT ASSIGNMENT(lines 2-6), 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, as proved in following theorem. To illustrate, show the same network of Fig.2(a) in 2(b) with all the interfering links removed, and so the network is scheduled in 3 time slots.

Algorithm 3 BFS-TIMESLOTASSIGNMENT

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

Although BFS TIME SLOT ASSIGNMENT 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, a method to eliminate interference the algorithm can optimally schedule the network.

2. Algorithms on Minimizing Latency

Minimizing the schedule length to complete convergecast certainly contributes to minimizing latency in data collection; however, in certain cases, it does not guarantee minimizing the average latency for individual packets. For instance, in aggregated data collection, where each sensor node is scheduled once per frame, and instead of relaying individual packets, they aggregate packets before forwarding toward the sink node; the minimal schedule length is equal to the maximum degree of the routing tree[3].

In this focus on minimizing latency and analyzing the energy-latency trade-off for data collection in WSNs where each node generates the same number of packets within each frame of length T. Data is transmitted to a relay node which forwards it toward the sink node; relay nodes are assumed to be not generating data. First present sufficient conditions on link scheduling in order to achieve the minimum worst case latency T, and then present a link scheduling algorithm satisfying these conditions. They propose and prove that it is sufficient for every node to schedule its outgoing links after its incoming links in order to achieve the minimum possible latency T.The minimum length scheduling does not automatically guarantee minimum latency, and a heuristic is proposed to minimize latency by scheduling the incoming links before the outgoing links.

To enable quick convergecast operations with minimum latency and complying with the Zigbee standard. Minimize latency by minimizing the schedule length and assigning slots to the senders, this study considers receiver-based scheduling. This is due to the fixed wakeup/sleep scheduling specified in the Zigbee stack[10] in each cycle, nodes wake up twice, first to receive packets from their children and second to transmit to their parents in a Zigbee beacon-enabled tree network. First define a minimum latency beacon scheduling problem for quick convergecast in Zigbee networks and prove it to be NP-complete. The algorithm is also extended for tree-based schemes as a heuristic. The performance of the heuristics decrease when the number of interference neighbors is high.

3. Algorithms on Minimizing Energy:

Energy efficiency is the biggest challenge in designing long-living sensor networks. Since radio communication consumes a lot of energy, a common method is to operate the radio with duty cycling that periodically switches the radio between sleep and wake-up modes. TDMA-based protocols offer the advantage of permitting nodes to enter into sleep mode during inactive periods, thus achieving low duty cycles and conserving energy. Additionally, TDMA-based medium access efficiently eliminates collisions and prevents overhearing, which are the main sources of energy consumption in wireless communication. Therefore, all the TDMA-based protocols proposed for WSNs have the inherent objective of minimizing energy consumption. Transmission power control is one of the well-studied methods in minimizing energy consumption and alleviating interference in wireless networks. Excessive levels of interference can be eliminated if the signals are transmitted with just enough power instead of maximum power.

TDMA-based communication provides a common energy-saving advantage by allowing nodes to turn their radio off when not engaged in communication; however, too much state transitions between the active and sleep modes can waste energy. Accordingly, the desired objectives in this to minimize the total time for data collecting as well as to minimize the energy consumed on switching between the active and sleep states. The goal is to find a TDMA schedule that can support as many transmissions as possible in every time slot. It has two phases: (i) scheduling, and (ii) power control. 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. The scheduling phase searches for a valid transmission schedule where no node is to transmit and receive simultaneously, or to receive from multiple nodes simultaneously. The power control phase then iteratively searches for an admissible schedule with power levels chosen to satisfy all the interfering constraints in the given valid schedule. 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:

$$P_{new} = \frac{\beta}{SINR} \cdot P_{current}$$

which is the well-known power control algorithm by Foschini, G., Miljanic, Z[6]. If the maximum number of iterations is reached and there are nodes which cannot meet the interfering constraints, the scheduling phase excludes the link with minimum SINR.

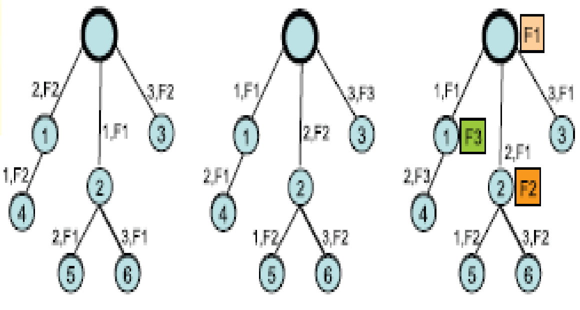
MULTICHANNEL SCHEDULING

Multi-channel communication is an efficient method to eliminate interference by enabling concurrent transmissions over different frequencies[7]. Although typical WSN radios operate on a limited bandwidth, their operating frequencies can be adjusted, thus allowing more concurrent transmissions and faster data delivery. In this section, We explain three channel assignment methods that consider the problem at different levels for both types of convergecast. These methods consider the channel assignment problem at different levels: the link level (JFTSS), node level (RBCA), or cluster level (TMCP).

- **Joint Frequency Time Slot Scheduling (JFTSS)**

JFTSS offers a greedy joint solution for constructing a maximal schedule, such that a schedule is said to be maximal if it meets the adjacency and interfering constraints, and no more links can be scheduled for concurrent transmissions on any time slot and channel without violating the constraints[9] and[12] respectively.

JFTSS schedules a network starting from the link that has the highest number of packets (load) to be transmitted. When the link loads are equal, such as in aggregated convergecast, the most constrained link is considered first, i.e., the link for which the number of other links violating the interfering and adjacency constraints when scheduled simultaneously is the maximum. The algorithm starts with an empty schedule and first sorts the links according to the loads or constraints. The most loaded or constrained link in the first available slot-channel pair is scheduled first and added to the schedule. All the links that have an adjacency constraint with the scheduled link are excluded from the list of the links to be scheduled at a given slot. The links that do not have an interfering constraint with the scheduled link can be scheduled in the same slot and channel whereas the links that have an interfering constraint should be scheduled on different channels, if possible. The algorithm continues to schedule the links according to the most loaded (or most constrained) metric. When no more links can be scheduled for a given slot, the scheduler continues with scheduling in the next slot. Fig. 3(a) shows the same tree given in Fig. 2(a) which is scheduled according to JFTSS where aggregated data is collected. JFTSS starts with link (2, sink) on frequency 1 and then schedules link (4, 1) next on the first slot on frequency 2. Then, links (5, 2) on frequency 1 and (1, sink) on frequency 2 are scheduled on the second slot and links (6, 2) on frequency 1 and (3, sink) on frequency 2 are scheduled on the last slot. An advantage of JFTSS is that it is easy to incorporate the physical interference model; however, it is hard to have a distributed solution since the interference relationship between all the links must be known.



(a) (b) (c)

Fig.:-3Scheduling with multi-channels for aggregated convergecast:(a) Schedule generated with JFTSS. (b) Schedule generated with TMCP.(c) Schedule generated with RBCA. (b) Schedule generated with RBCA.

- **Tree-Based Multi-Channel Protocol (TMCP)**

TMCP is a greedy, tree-based, multi-channel protocol for data collection applications[11]. 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. Fig.3(b) shows the same tree given in Fig. 2(a) which is scheduled according to TMCP for aggregated data collection. Here, the nodes on the leftmost branch is assigned frequency F1, second branch is assigned frequency F2 and the last branch is assigned frequency F3 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.

- **Receiver-Based Channel Assignment (RBCA)**

By 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. Fig. 3(c) shows the same tree given in Fig. 2(a) scheduled with RBCA for aggregated convergecast. Initially all nodes are on frequency F1. RBCA starts with the most interfered parent, node 2 in this example, and assigns F2. Then it continues to assign F3 to node 3 as the second most interfered parent. Since all interfering parents are assigned different frequencies sink can receive on F1.

Future Research Directions:-

As we have seen, extensive research has been done with many different objectives in the field of TDMA scheduling for data gathering in WSNs. However, there still exist some open questions to be addressed, especially related to real implementation and evaluation of the proposals on testbeds or on real deployments. There also exist several theoretical questions that need to be addressed.

Some of the surveyed algorithms provide cross layer solutions, where the schedules are computed together with methods such as transmission power control, optimal routing trees, and multi-frequency scheduling. It is indeed essential to address the problems from a cross layer perspective to achieve the target functions and offer better performances. Along this line of research, Chafekar *et al.* in [15] extend the work by Moscibroda [14] in designing cross-layer protocols using the SINR model and proposed polynomial time algorithms with provable worst-case performance guarantee for the latency minimization problem. Their cross-layer approach chooses power level for all transceivers, routes for all connections, and constructs an end-to-end schedule such that SINR constraints are satisfied. A prominent research direction is to consider such cross-layer approaches from a theoretical point of view. More research can be done in this direction to combine the existing work with the solutions at different layers. In most studies, static topologies are assumed. Problems related to dynamic topologies, such as topological changes and addition of new nodes are open. In addition, the time complexity of data gathering under various hypothesis [16], such as when some nodes have no packet to transmit, or when no buffering is allowed remain open

CONCLUSION:-

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). In terms of the design objectives, most of the surveyed algorithms aim at minimizing schedule length, minimizing latency, minimizing energy, and multichannel scheduling.

REFERENCES

1. S. Gandham, Y. Zhang, and Q. Huang, "Distributed time-optimal scheduling for convergecast in wireless sensor networks," Computer Networks, vol. 52, no. 3
2. H. Choi, J. Wang, and E. A. Hughes, "Scheduling on sensor hybrid \network," in Proc. ICCCN, 2005
3. 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
4. A. Ghosh, O. D. Incel, V. A. Kumar, and B. Krishnamachari, "Multi-channel scheduling algorithms for fast aggregated convergecast in sensor networks," in MASS '09, Macau, China.
5. M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness. San Francisco, CA: Freeman, 1979.
6. Foschini, G., Miljanic, Z.: A simple distributed autonomous power control algorithm and its convergence. IEEE Transactions on Vehicular Technology 42(4), 641–646 (1993).
7. P. Kyasanur and N. H. Vaidya, "Capacity of multi-channel wireless networks: impact of number of channels and interfaces," in MobiCom '05, Cologne, Germany, pp. 43–57.
8. Annamalai, V., Gupta, S., Schwiebert, L.: On tree-based convergecasting in wireless sensor networks. In: WCNC '03, vol. 3, pp. 1942–1947 vol.3 (2003)
9. G. Sharma, R. Mazumdar, and N. Shroff, "On the complexity of scheduling in wireless networks," in MobiCom '06, Los Angeles, CA, USA, pp. 227–238.
10. M. Pan and Y. Tseng, "Quick convergecast in zigbee beaconenabled tree-based wireless sensor networks," Computer Communications, vol. 31, no. 5, 2008.

11. Y. Wu, J. Stankovic, T. He, and S. Lin, "Realistic and efficient multi-channel communications in wireless sensor networks," in INFOCOM '08, pp. 1193–1201.
12. X. Lin and S. Rasool, "A distributed joint channel-assignment, scheduling and routing algorithm for multi-channel ad-hoc wireless networks," in INFOCOM '07, pp. 1118–1126.
13. Chafekar, D., Kumar, V.A., Marathe, M., Parthasarathy, S., Srinivasan, A.: Cross-layer latency minimization in wireless networks with SINR constraints. In: MobiHoc '07, pp. 110–119. ACM, New York, NY, USA (2007)
14. Moscibroda, T., Wattenhofer, R., Zollinger, A.: Topology control meets SINR: the scheduling complexity of arbitrary topologies. In: MobiHoc '06, pp. 310–321 (2006)
15. Cui, S., Madan, R.,Goldsmith, A., Lall, S.: Energy-delay tradeoffs for data collection in tdmabased sensor networks. In: ICC '05, vol. 5, 2005.
16. Gargano, L.: Time optimal gathering in sensor networks. In: SIROCCO '07, pp. 7–10 (2007)