An Efficient Tree Structure for Delay Sensitive Data Gathering in Wireless Sensor Networks

Soonmok Kwon, Jeong-gyu Kim and Cheeha Kim
Department of Computer Science and Engineering, Postech, Pohang, Korea
Emails: {smok80, haji1202, chkim}@postech.ac.kr

Abstract—It is important to design an energy efficient data gathering tree structure for wireless sensor networks. As for the energy efficiency, network’s overall energy consumption and per-node fairness have been studied. Note that the minimum degree spanning tree (MDST) is optimal in the sense of per-node fairness. We believe that the per-node fairness is of practical interest and that the delay bound associated with it must be investigated. Unfortunately, no such efforts have been made so far.

In this paper, we propose a tree structure, called the DB-MDST, which modifies MDST to reduce the height and extend its degree. With the DB-MDST, the near optimal delay bound (i.e., comparable to the one obtained using the shortest path tre) can be achieved at the cost of as much energy as consumed by the MDST while it cannot be achieved with the MDST. Simulation results support the claim stated.

Index Terms—Sensor Networks, Data Gathering Tree, Energy Efficiency, Per-node Fairness, Delay Bound

I. INTRODUCTION

Advances in processor, memory and radio technology have enabled small and cheap nodes capable of wireless communication and data processing. Wireless sensor networks are composed of a large number of sensor nodes deployed over a wide area, operating in an unattended mode, supporting applications such as monitoring environmental changes in the area [1], [2]. The networking [3] and MAC protocols [4], [5] have been widely studied.

Wireless sensor networks are similar to mobile ad-hoc networks in that both involve multi-hop wireless communications. However, by the nature of the data gathering application, sensor networks have several unique characteristics, and two major ones are as follows: First, the traffic is from multiple sources to a single destination called sink - a reverse multicast. Second, there is likely to be some redundancy in the data from the multiple sources - the correlation in the data. The technique called data-centric routing [6], in-network aggregation of data [7] or correlation-aware data gathering [8], [9] is introduced to deal with these characteristics. The idea is to combine the data coming from different sources enroute - eliminating redundancy, minimizing the number of transmission and thus saving energy [10]. The data aggregation is a typical example of the cross-layer design in sensor networks and its main benefits are as follows:

1) Overall energy efficiency: As mentioned above.

2) Lifetime elongation of nodes near the sink: The sensor nodes located close to the sink suffer premature battery depletion from relaying packets, leading to an early disconnection of the network. In-network data reduction is an efficient solution to this problem. Another solution is to introduce mobile devices in sensor networks.

Data gathering applications can be categorized by the degree of correlation existing between the source data [7], [9], [11]. For statistical queries such as SUM, MAX, MIN, etc., aggregated data is of constant size regardless of the number of data sources. This type of data query is called fully-aggregated query. The unaggregated query refers to the opposite case, where no data reduction is available. Intermediate cases are called partially-aggregated query. 1

After the development of the forwarding schemes where data is opportunistically aggregated at the intermediate nodes [3], various structured data aggregation schemes have been studied. They can be classified into two classes: First, there are cluster-based approaches [12] where the cluster header nodes perform data aggregation. Second, there are tree-based approaches [6]–[11], [13] where each node gathers data from its downstream nodes (i.e. children) and sends aggregated data to its parent node. The chaining approaches [14] can be considered as special cases of the tree-based approach. Our focus is on this tree-based approach that has somewhat higher flexibility for real implementations.

Designing an efficient data gathering tree structure is the goal in this field. As for the energy efficiency, network’s overall energy consumption and per-node fairness are the two primary research issues.

In the first issue, the overall energy efficiency, the problem is to find an energy efficient spanning tree over all the sources, sink and some of non-source nodes where the energy is measured by the number of packet transmissions and receptions. In this case, the Steiner minimum tree (SMT) is a well-known optimal solution for the fully-aggregated queries [8], [9].

In the second issue, per-node fairness, the focus is on the most overburdened node and the network lifetime is defined as the time until the first node dies. Among the previously proposed tree structures focused on the fairness, the minimum degree spanning tree (MDST) is considered as an optimal solution for the fully-aggregated queries [11]. With a fully aggregated query, the difference in the energy consumption among nodes solely comes from the packet reception cost,

1We used the terms from [11]
which is proportional to the number of children. Then, the node with the largest number of children will suffer the battery depletion first. This logic leads to the conclusion that the optimal solution for energy efficiency is the MDST where the number of children of each node is reduced as much as possible.

We focus on the second issue for following reasons. First, overall energy consumption is sometimes not controllable. It is not rare that the sensor application periodically gathers data from all the nodes with fully-aggregated query. In this case, without non-source nodes, the difference in the overall energy consumption among spanning trees is negligible. Second, load balancing by role rotation is not an essential solution. In cluster-based data aggregation, the problem of non-uniform energy drainage is addressed by the rotation of the cluster heads. A similar role rotation can be adopted for tree-based approach. However, rebuilding the entire tree structure takes considerable energy and time. To reduce the roll rotation frequency and thus save energy, we need to build a tree tailored to the per-node fairness at first.

For these reasons, we believe that the per-node fairness is of practical interest and the delay bound associated with it must be investigated. Unfortunately, no such efforts have been made so far.

In this work, we address the efficiency of the data gathering structures measured by both the per-node fairness in energy consumption and the delay bound. We focus on the height and maximum degree of a tree and propose a tree structure called DB-mdst, which modifies MDST to reduce the height and extend its degree. Then we conduct a comprehensive simulation study in a realistic system model considering various issues including the MAC effects. Our simulation shows encouraging results: The DB-mdst shows the near optimality in both the delay bound and energy efficiency. The delay bound of DB-mdst is comparable to the one obtained using the shortest path tree (SPT), and the energy efficiency of DB-mdst is comparable to that of the MDST, while the MDST incurs about four times of delay than is achievable using the SPT.

The remainder of the paper is organized as follows. In Section 2, we discuss some of the related works in this field. Then we describe the system model in Section 3. In Section 4 we present the DB-mdst algorithm. In Section 5, we describe the simulation results. Finally, we present our conclusions in Section 6.

II. RELATED WORKS

Here, we briefly discuss tree-based data aggregation schemes. We start from the first issue: To reduce the energy consumption in the entire network. In [6], the authors designed a greedy algorithm to build a tree that is a sort of an approximated SMT. This structure was evaluated with the metric ‘number of transmissions’ for energy consumption and ‘average distance between sink and source’ for delay. In [7], the authors studied the data aggregation issue in implementing real system using the SPT. In [8], the authors dealt with the energy efficiency considering the degree of correlation, proved NP-hardness of building solution and proposed an approximation algorithm based on the shallow light tree (SLT), a spanning tree that approximates both the SMT and SPT. In [9], the authors studied the achievable benefits of data aggregation considering the delay bound. They designed the delay-bounded SMT based on the bounded shortest multicast algorithm (BSMA) and evaluated this structure with the metric ‘number of edges on the tree’ for energy consumption and ‘tree height (maximum distance from sink in hop counts)’ for delay. Their result is that the SMT incurs about two times of delay than was achievable using SPT and that this delay overhead can be eliminated at the cost of 7% - 12% more energy.

Next, we discuss the second issue: The per-node fairness in energy consumption. In [10], the authors focused on the energy level differences among nodes and showed that finding the optimal routing tree is NP-complete for aggregated, unaggregated or partially aggregated queries. For fully aggregated queries they proposed a tree based on the MDST as a near-optimal solution with worst case constant factor performance guarantee. For unaggregated and partially aggregated queries, they found the maximal performance by formulating the routing problem as an integer program and proposed an approximation algorithm. In [11], the authors showed that the MDST problem is approximable to within optimal+1 in terms of the maximum degree. In [12], a distributed version of the MDST approximation algorithm is proposed.

III. SYSTEM MODEL

In this section we describe the system model. To make it realistic, we used hardware datasheets, standards and various measurement results currently available.

A. Node and Sink Hardware

We assume that the sensor nodes are homogeneous and their radio transmission range is fixed. For detailed parameters, we assumed the CC2420 radio chip from Texas Instruments designed for IEEE 802.15.4 MAC standard. The CC2420 transceiver supports four states: shutdown, idle, transmit and receive. We used the energy measurement result on this chip shown in Figure 1 [17].

In general, the sink node is considered to be much more powerful than the sensor node. In this work, we assume that the sink has unlimited energy. This should be considered in the data gathering algorithm design. For example, the sink’s large degree in the tree can be ignored now and the MDST algorithm should be changed accordingly.

B. Topology

The nodes are uniformly distributed in a round area with sink at its center as shown in Figure 2, where $R$ is the network radius and $r$ is the node’s transmission range. We set the transmission range as 25m, where the packet error rate is measured below 0.01 when the packet size is smaller than or equal to 127 bytes [18].

$\text{127}$ is the maximum size of PHY service data unit defined in IEEE 802.15.4 standard.
If the nodes are uniformly distributed, then the probability of all-node connectivity can be formulated for the given network radius $R$, the node transmission range $r$ and the number of nodes $n$ as follows [19]:

$$r = R \cdot \sqrt{\frac{1}{n} \log\left(\frac{1}{e}\right)}$$

Here, $1 - e$ is the connectivity probability. In this paper, all the simulations are conducted under the condition that the connectivity probability is larger than 90%.

### C. The Application

We assume an application, where sensing data from all the nodes are gathered at the sink periodically. As for the degree of data correlation, we assume the fully-aggregated query type with which the MDST shows optimality in the energy efficiency.

### D. MAC

We assume that the unslotted CSMA-CD of IEEE 802.15.4 MAC standard is used in the sensor network. Basic operation of this scheme is shown in Figure 3. Table I shows the MAC parameters we used. There are several issues to be considered for MAC operation as follows:

1) Overhearing avoidance: Overhearing the frames destined to other nodes incurs considerable energy consumption. In [20], the authors proposed a special preamble containing the destination information, so that the non-destination nodes can stop listening just after receiving the preamble. We assume that similar overhearing avoidance is used. Specifically, we assume that the non-destination nodes can stop listening just after receiving the destination address field of 802.15.4 MAC header.

2) Packet loss rate: We assume that packet loss rate is 0.01. This is a worst-case assumption for $25m$ transmission range [18].

3) Permanent packet loss: In most of data gathering performance evaluations, an aggregation node is assumed to wait for all downstream nodes in the tree to reach it. However, if permanent packet loss occurs, then the nodes should not behave in this way. In this case, novel issues such as ‘waiting time control for upstream nodes’ and ‘fault tolerant data encoding’ arise. We do not address these issue in this work. Instead, we assume that no permanent packet loss occurs. For this, we assume that no node hardware failure occurs and eliminate the limit on frame transmission retries in 802.15.4 standard.

4) Fragmentation: With the partially-aggregated and non-aggregated query, some nodes can have aggregated data larger than the maximum payload size defined in IEEE 802.15.4. We assume that the packet fragmentation occurs in this case. Specifically, we assume that no additional data is attached for fragmentation.

5) Carrier sense range: We use the setting of the NS2 simulator, where a default carrier sensing range of 2.2 times the transmission is used. This static value is sub-optimal in many network scenarios [21].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datarate</td>
<td>250 kb/s (32 µs/byte)</td>
</tr>
<tr>
<td>Symbol period</td>
<td>16 µs/symbol</td>
</tr>
<tr>
<td>PHY preamble and SFD</td>
<td>6 bytes</td>
</tr>
<tr>
<td>MAC data frame overhead</td>
<td>9 bytes</td>
</tr>
<tr>
<td>MAC ack frame size</td>
<td>5 bytes</td>
</tr>
<tr>
<td>Unit backoff period</td>
<td>20 symbol periods</td>
</tr>
<tr>
<td>CCA detection time</td>
<td>8 symbol periods</td>
</tr>
<tr>
<td>Tx/rx turnaround time</td>
<td>12 symbol periods</td>
</tr>
<tr>
<td>Max. backoff exponent</td>
<td>5</td>
</tr>
<tr>
<td>Min. backoff exponent</td>
<td>3</td>
</tr>
<tr>
<td>Max. backoff trials</td>
<td>4</td>
</tr>
<tr>
<td>Max. retransmission trials</td>
<td>Unlimited (3 in standard)</td>
</tr>
<tr>
<td>SIFS</td>
<td>12 symbol periods</td>
</tr>
<tr>
<td>LIFS</td>
<td>40 symbol periods</td>
</tr>
<tr>
<td>Ack wait duration</td>
<td>54 symbol periods</td>
</tr>
</tbody>
</table>

**TABLE I**

**MAC PARAMETERS**
Fig. 4. Tree examples generated with the setting: transmission range=25m, network radius=90m, # of nodes=180

Build_DB-MDST (Tree, Graph)
repeat while any modification to tree occurs
for-each max-degree vertex v
edges (e_add, e_delete) ← Benefiter_Edges(Tree, Graph, v)
if (e_add, e_delete) have been found
add e_add to the tree
delete e_delete from the tree
get out of the for-each loop

Benefiter_Edges (Tree, Graph, v, height_limit)
assume v is deleted from tree and build a forest
let e_add ← one of v’s incident edge
for each edge e that connects two different trees
whose degree < v’s degree − 1
if degree of e_add > degree of e and
height of the tree built using e ≤ height_limit
e_add ← e
if e_add ≠ one of v’s incident edge
return e_add, one of v’s incident edges connected to e_add
else
return failure code

Fig. 5. DB-MDST algorithm pseudo-code

IV. DELAY BOUNDED MDST

Even when the MAC effects are not taken into account, analyzing tree performance is rather a complicated task, and the integer programming formulation is often used. This approach is useful to find an optimal solution but somewhat lacks applicability when the implementation cost is considered. Approximation algorithms are used in this case.

Our approximation algorithm focuses on the two properties of tree structures, the height and the maximum degree, assuming that these properties exert critical influence on the performance. We use the aforementioned performance metrics. The energy consumption is measured by the energy consumption of the most overburdened node. The data gathering delay is measured from the time when the sensing data is generated in the entire network to the time when the sink gathers all the data. Prior to describing the algorithm, we briefly discuss the effect of the two tree properties on the energy and delay performance as follows:

1) A node’s energy consumption is proportional to its degree and the energy consumption of the most overburdened node is proportional to the maximum degree of the tree.
2) A node’s waiting time for its sub-tree data is proportional to the number of its descendants. The number of descendants is upper bounded by $\sum_{i=1}^{h} k_i$, where h is the height of the subtree rooted at this node and k is the maximum degree of the subtree. From this, we can assume that the data gathering delay is mainly proportional to the height of the tree.

Now, we describe the novel tree structure that can control both the height and maximum degree. Specifically, our algorithm minimizes the maximum degree while maintaining the tree height below a predefined limit. We call this the DB-MDST because it is based on the approximated MDST algorithm. Simply put, the approximated MDST algorithm is an iteration of removing one edge from the maximum degree vertex and adding an appropriate edge to maintain connectivity. The key idea of the DB-MDST is that the tree height can be maintained below a predefined limit by modifying the edge-adding part of the MDST iteration. Figure 5 shows the pseudo code of the DB-MDST, where the function Benefiter_Edges is used to select nodes to be removed or added. The height_limit parameter of this function is the predefined height limit to be maintained.

We make two more remarks on the DB-MDST algorithm. First, to notify the height limit of the DB-MDST, we use the constant HL that is defined as the ratio of the predefined height limit to the height of the SPT built on the same topology. Second, the assumption that the sink has unlimited battery resource should be considered. The sink’s large degree in the tree does not matter in this case. This is considered in our simulation.

The DB-MDST, a simple greedy algorithm, performs well in controlling the height and maximum degree. An example is shown in Figure 4, where the SPT, MDST and DB-MDST are
Fig. 6. Energy consumption of the most overburdened node plotted against network size and node density

Fig. 7. Data gathering delay plotted against network size and node density

built on common node connectivity. In the example, the SPT shows a small height and a large maximum degree, while the MDST shows a small maximum degree and a large height. The DB-MDST with HL=1.0 shows a maximum degree similar to that of the MDST while the height is upper bounded by the given limit.

V. PERFORMANCE RESULTS

In this Section, we present the simulation results. Our simulations are conducted for two purposes: To verify the effects of the height and maximum degree of a given tree on the performance metrics, and to assess the performance of the DB-MDST compared to prior solutions.

We compare the energy and delay performances of the SPT, MDST and DB-MDST configured with following height limits: $HL = 1.0, 2.0$. The SPT is selected as the near optimal solution for delay. The MDST is selected as the near optimal solution for energy efficiency. We use the aforementioned performance metrics for the energy consumption and delay.

We first present the effect of changing the network size and node density on the energy consumption and delay, where the network size is related to the tree height and the node density is related to the maximum degree. The network size is defined as the ratio of the network radius to the transmission range and its effect is measured in the setting where the network size is varied from 2 to 7 and the node density is fixed to 10. The node density is defined as the average number of neighbors and its effect is measured in the setting where the node density is varied from 8 to 16 and the network size is fixed to 5. Then, we discuss the optimality of the DB-MDST compared to the SPT and MDST.

A. Energy Consumption

Figure 6 shows the energy consumptions plotted against the network size and node density. The graph plotted against the node density shows a steeper slope than that plotted against the network size. This means that the maximum degree exerts more influence on the energy consumption than the height. Due to the maximum degree, the energy consumption of the SPT is far from the optimal (i.e. the energy consumed in the MDST), and the DB-MDST with a large height limit shows better performance than that with a smaller one. Note that, the DB-MDST’s energy consumption is comparable to that of the MDST even with the strongest height limit ($HL = 1.0$).
B. Delay

Figure 7 shows the delay plotted against the network size and the node density. The graph plotted against the network size shows a much steeper slope than that plotted against the node density. This means that the height exerts more influence on the delay than the maximum degree. Due to the height, the delay of MDST is far from the optimal (i.e. the delay of the SPT), and the DB-MDST with a small height limit shows better performance than that with a larger one. Note that, the DB-MDST’s delay is comparable to that of the SPT even with a large height limit ($HL = 2.0$).

C. Energy and Delay Optimality

Now we discuss the optimality issue using Figure 8, which shows the energy and delay measurements in the setting where the network size is 6 and node density is 10. As shown in the figure, DB-MDST with $HL = 1.0$ achieves 73.2% delay improvement (i.e. 8.5% more delay than the best case,) at the cost of 7.1% additional energy expense than the MDST. Thus we can conclude that the DB-MDST achieves the near optimality in both the delay bound and energy efficiency while the MDST incurs about four times of delay than is achievable using the SPT.

VI. CONCLUSION

In this paper, we addressed the delay bound associated with the per-node fairness issue where the MDST is known to be optimal. We propose a tree structure focused on the height and maximum degree of a tree, called the DB-MDST, which modifies MDST to reduce the height and extend its degree. Through quantitative study and analysis, we conclude that, with the DB-MDST, the near optimal delay bound (i.e., comparable to the one obtained using the SPT) can be achieved at the cost of as much energy as consumed by the MDST while it cannot be achieved with the MDST.

REFERENCES


