Efficient Data Collection through Compression-Centric Routing

Hao Chen, Seapahn Megerian
Electrical and Computer Engineering Department
University of Wisconsin Madison
hao@cae.wisc.edu, megerian@ece.wisc.edu

Abstract
Efficient sensor data fusion is one of the more critical and challenging tasks in building practical sensor networks. It is widely understood that transmitting raw sensor data to a central location for processing is severely hampered by scaling in large scale wireless networks, both in terms of energy consumption and latency costs. However, many detection, classification, estimation, and phenomena modeling algorithms rely heavily on the individual data from each sensor and thus, require raw data collection if not from the entire network, then at least among localized node clusters of varying sizes. In order to make the data collection as efficient as possible, various proposed sampling and compression techniques have been, and are being investigated. As we demonstrate, in addition to the compression algorithm, the topology of the aggregation (e.g. the routes used) can play a significant role in the achievable compression rates. In this paper, we propose a compression-centric data collection algorithm for use in wireless sensor networks. The algorithm relies on the construction of a Minimum Cost Tree (MCT), using neighbor data correlation as the optimization heuristic along the tree. By selecting routes that traverse nodes with higher degrees of data correlation, it is possible to achieve superior compression results when compared to more naive minimum spanning tree algorithm variations. In addition to presenting the details of our distributed MCT algorithm, we use simulated data as well as actual data from a real sensor network to validate and demonstrate the performance of the new compression-centric routing scheme.

I. INTRODUCTION
Sensor networks have emerged as a fundamentally new tool for monitoring inaccessible environments such as non-destructive evaluation of buildings and structures, contaminant tracking in the environment, habitat monitoring, and surveillance in military zones. Many autonomous, resources efficient, sensor network applications aim to answer questions about the basic patterns, structures, and relationships in the measured data by the sensors. Such questions can often be posed as detection, classification, estimation, phenomena modeling, or other similar problems that have been widely studied in the past under the assumption that the data is stored and processed at a central location. With sensor networks, that assumption is changed: We assume that the data is not centralized, but rather distributed across all, or clusters of sensor nodes. This is driven by the fact that the cost of computation at each node is typically much less than the cost of communication between the nodes, making the option of transmitting all data to a central site for processing very expensive and unattractive in comparison.

Although relying on the data remaining distributed in the network is much more efficient in terms of energy consumption and latency, it creates severe restrictions on the kinds of algorithms, their accuracy, and implementation feasibility in the network. By collecting a subset of the data in clusters of increasing sizes, one can trade off the energy and latency costs of the data collection process with the flexibility and ability to run more powerful fusion and inference algorithms.

Having energy as the primary constraint on all aspects of design in wireless sensors networks naturally leads to the investigation of finding ways to reduce the power consumption associated with such data aggregation schemes. Finding routing paths in wireless ad hoc networks to minimize power consumption is a complex problem in itself. A general assumption with respect to transmission power has been that directly transmitting data to a sensor along a longer hop requires more power than transmitting data through multiple shorter hops. However, this is not always the case since the nodes also spend significant energy in receiving, processing, and idle states. The main motivation in using multi-hop routing approaches is fundamentally driven by the limitations in the maximum achievable transmission ranges of small, battery operated, wireless nodes.

Some of the related works discussed in section II investigate and show how data aggregation can reduce power consumption in, and extend the life time of, sensor networks. Distributed Source Coding is one potential technique to provide the data compression mechanism in sensor networks. In section III.1, we briefly outline the theory behind Distributed Source Coding (DSC), its application in sensor networks, and the associated problems. The main challenge with DSC is the difficulty in finding the correlation between all sensors in the system. Furthermore, currently it is not clear how to apply routing algorithms based on data correlation in general sensor networks. Here, our aim is to leverage the abilities of routing heuristics based on data correlation in order to enable more efficient data fusion and to help to reduce the overall power consumption of the network.

II. RELATED WORK
Routing protocols for wireless ad hoc networks and sensor networks typically optimize the performance in terms of energy consumption, end-to-end delay, and/or throughput. Variations of shortest-path routing schemes have been used in such networks for a long time. Youssef et al. [4] present an energy-aware routing protocol that minimizes the energy consumption and maintains good end-to-end delay and
throughput performance at the same time. The algorithm is based on the constraint on the maximum transmission distance with minimum hop count routing. Even through the algorithm provides a trade off method for energy consumption and end-to-end delay, its performance is heavily dependent on the value for the maximum transmission distance constraint. Reference [5] provides a solution for delivering messages from any sensor to a sink along the minimum cost path in a large sensor network. The cost field setup algorithm finds the optimal costs of all nodes to the sink with one single message overhead at each node. In this algorithm, each node broadcasts the optimal cost to its neighborhood sensors. Once a minimum-cost path is established, the messages carrying dynamic cost information flow along the path.

The problem of maximizing the overall system lifetime for data collection is a fundamental research topic associated with sensor networks. In [1], the sensors in the network are grouped into several clusters and sensor data in the same cluster are gathered and aggregated. The sink sensor of each cluster is regarded as a node at a higher level in the overall data collection hierarchy. The sink sensor in each cluster is chosen in a round-robin manner in each round to minimize the energy burden on the sink node. In [6] attempts are made to find data gathering schemes that balance the energy and delay costs, quantified by the energy×delay product. The algorithm uses a chain-based multiple level scheme to optimize the energy×delay product for the sensor network. In each level, sensors are classed as several clusters. Data from the sensors are transmitted to the sink sensor of that cluster by chain link, and fused during the transitions. This method does not use data correlation or a similar heuristic to find an optimal route for compressing the data during aggregation.

The potentially heavy overlap in data and distributed nature of the sensor network requires efficient and fully distributed data compression techniques without the sensors needing to talk to one another during data compression. Distributed source coding (DSC) is the fundamental concept in information theory applicable to this problem. Pradhan et al. [7] review the main ideas, provides illustrations, and gives the intuition behind the theory that enables this framework. Chou et al. [8] propose distributed source coding (DSC) to reduce energy consumption in a sensor network. The results show an estimated 10%-65% improvement. An adaptive filtering framework is used to continuously estimate the relevant correlation in the data of the sensors. The authors provide a simple distributed algorithm (one modulo operation) to implement the DSC. The decoding error is unavoidable by this method although error detection and correction techniques can deal with such errors.

Cristescu et al. [14] discuss two approaches that may be used in correlated data gathering on sensor network. The one approach is using Slepian-Wolf coding (DSC). This method requires the correlation structure of all sensors and potentially provides optimized data-gathering structure. Slepian-Wolf coding can be applied locally in several disjoint clusters. When each cluster contains more than 2 sensors, obtaining an optimal cluster division is an NP-complete problem. Another method mentioned in [14] is that data compression in each sensor is just based on the data of current sensor and the side information from the other sensors instead of global correlation structure. Shortest Path Tree (SPT), Traveling Salesman Path (TSP) and SPT/TSP balanced tree are discussed and evaluated with simulation in their paper.

W. Heinzelman et al. [1] propose LEACH, which randomly selects cluster-heads and provides data fusion (data aggregation) in each hop to the head to reduce energy consumption. In [2], Mo Chen et al. demonstrate the importance of data aggregation for energy efficiency by showing that using data aggregation with LEACH can increase the lifetime of the network while randomly selected cluster-heads does not save the power consumption of the whole sensor network. Pattern et al. [3] show that the routing relative to data correlation can improve data fusion performance and provide better energy efficiency. They also analyze theoretic optimal and near-optimal solutions with an assumed data correlation model with uniformly located sensors in one dimension. In their model, the data correlation is determined by the physical distance of sensors. However, the data correlation is also effected by the sensor direction and the obstructions in sensor networks in real situation.

The topology in data-gathering in wireless sensor networks is a spanning tree because the traffic is mainly in the form of many-to-one flows. References [9, 10] present several examples of energy aware and hierarchical clustering and data collection algorithms. Zhou [11] evaluates the effect of localized topology generation mechanisms on network performance metrics: node degree, robustness, channel quality, data aggregation and latency. A total of four mechanisms are used there: earliest-first, randomized, nearest-first, and weighted-randomized. The simulations show that localized cluster head selection strategies can significantly impact the global performance of the network.

III. PRELIMINARIES

As a simple illustrative example of our problem here, consider the communication network shown in Fig. 1(a) where the cost values are denoted next to each link. Fig. 1(b) shows the links that form a Minimum Cost Spanning Tree (MST). In this case, the sum of all link costs for the MST is 32. However, the sum of the costs from all nodes to the root, if they were to send their data independently, is 93. Fig. 1(c) shows another link set forming a spanning tree where the total cost is 38, which is larger than that of MST. However, the sum of the costs from all nodes sending their data independently to the root is 78 which is less than that of the MST. In addition, data correlations along the path will significantly impact the compression rates.

3.1 Distributed Source coding in Sensor Networks

Pradhan and Zixiang et al. [7, 12, 13] discuss Distributed Source Coding (DSC) in sensor networks. DSC techniques use a jointly designed codec of several sensors to
reduce the data size. If the sensor network contains \( N \) sensors \((x_1, x_2 \ldots x_N)\), the information obtained by this sensor network can be evaluated by the entropy \( H(x_1, x_2, \ldots, x_N) \). The information of the sensor network is obtained by a very large number of distributed sensor nodes. If each node were to transmit its information to the sink nodes independently, the potentially redundant information from different sensors will require unnecessary hardware resources, energy, and communication bandwidth. The relation between the data in the sensor network and an individual sensor can be given by information theory:

\[
H(x_1, \ldots, x_N) \leq H(x_1) + H(x_2) + \ldots + H(x_N) \tag{1}
\]

The entropy of sensor network can also be expressed by:

\[
H(x_1, \ldots, x_N) = H(x_1) + H(x_2 | x_1) + \ldots + H(x_N | x_1, x_2, \ldots, x_{N-1}) \tag{2}
\]

Here \( H(x_j | x_i) \) is the entropy of \( x_j \) given information of \( x_i \) and \( H(x_2 | x_1) \leq H(x_2) \). \( H(x_3 | x_1, x_2, \ldots, x_{N-1}) \) is the entropy of \( x_N \) given information of \( x_1, x_2, \ldots, x_{N-1} \) and \( H(x_1, x_2, \ldots, x_{N-1}) \). The equality is satisfied when \( x_N \) is uncorrelated with any of the other sensors. The basic idea of DSC is to use redundant information between the sensors to reduce the final data size that is transmitted.

The main challenge in DSC is that the reference information from a sensor \( x \) is not available during encoding even through it is available during decoding. Pattem et al. [3] and Youssef [4] discuss how to estimate the correlation between the sensors and use the correlations to decide how many bits to send from each sensor. It is almost impossible to estimate or provide all correlation information among the huge number of sensors that we expect to have in a large sensor network. Furthermore, applying DSC among many number of sensors is an NP-complete problem [14].

In general, two sensors that are far away from each other have very weak correlation in the information collected by the sensors. Thus, another practical method to transmit correlated data in sensor networks is to only utilize the correlation among a sensor and its closer neighboring sensors. It is not difficult to individually estimate correlation between two sensors from real collected data. In this case, data coding is easily applied and relies only on the data from neighborhood sensors. In this paper, we will provide one method to find optimal routing path for the second solution.

### 3.2 Power consumption in wireless system

The radio model discussed in [1] can be used to evaluate power consumption of data transmissions. In this model, a radio dissipates \( E_{\text{elec}} \) (50 nJ/bit) as defined for the transmitter or receiver circuitry and \( E_{\text{amp}} \) (100 pJ/bit/m\(^2\)) as defined for the transmitter amplifier. All sensor nodes are assumed to have power control and can adjust to use the minimum required energy to send information to the intended recipients. Furthermore, the nodes can turn off their transmitter and/or receiver to avoid receiving uninteresting information and save energy. Since receiving is also a relatively costly operation in the wireless systems in our aim, the power consumption for the receiver should also be accounted in this model. Let us look at the equations used to evaluate the power consumption of a node for wireless communication.

The power consumption for a transmitting node:

\[
E_{\text{tx}}(k, d) = E_{\text{elec}} \cdot k + E_{\text{amp}} \cdot k \cdot d^2 \tag{3}
\]

The power consumption for a receiving node:

\[
R_{\text{rx}}(k, d) = E_{\text{elec}} \cdot k \tag{4}
\]

Here \( d \) is the distance between two sensors, \( k \) is bits of information sent through the network, and \( E_{\text{elec}} \) and \( E_{\text{amp}} \) are the constants as defined above. The total power consumption cost is thus given by:

\[
E_{\text{total}} = E_{\text{tx}}(k, d) + R_{\text{rx}}(k, d) = (2E_{\text{elec}} + E_{\text{amp}} \cdot d^2) \cdot k = L(d) \cdot k \tag{5}
\]

The power consumption is a second order function of distance and thus, the data routing path with multiple shorter hops will typically be more efficient than directly transmitting data between two sensors. The power consumption is also a linear function of \( k \), the number of bits of information sent through the link.

### 3.3 Cost function to minimize power consumption

Before introducing the algorithm, we define several variables used in the algorithm. \( L_{0,3}(d) \) defines the partial factor of power consumption when information transmitted from sensor \( s_3 \) to sensor \( s_0 \). \( L_{3,1}(d) \) defines the partial factor of power consumption when information is transmitted from
sensor \( s_3 \) to sink sensor \( s_0 \). Those two variables can be given by the equation:

\[
L_{0,2}(d) = 2E_{\text{elec}} + E_{\text{step}} \cdot d(s_0, s_2)^2
\]

\[
L_s(d) = L_{0,2}(d) + L_{2,3}(d)
\]

Here, \( d(s_0, s_2) \) is the distance between sensor \( s_0 \) and \( s_2 \). Sensor \( s_2 \) will transmit information to the sink \( s_0 \) through sensor \( s_3 \). When the information of sensor \( s_1 \) and cross-correlation between \( s_1 \) and \( s_2 \) are available, the number of bits \( k_1 \) can be estimated by information theory as (10). The cost function, when \( s_2 \) transmits its information to \( s_0 \) through \( s_1 \) can be given by the equation:

\[
C_s[s_1, s_0] = k_1 \cdot (L_{1,4}(d) + L_n(d))
\]

In the same way, the cost function when \( s_2 \) transmits its information to \( s_0 \) through \( s_1 \) is given by:

\[
C_s[s_1, s_0] = k_2 \cdot (L_{1,4}(d) + L_n(d))
\]

The parameters \( k_1 \) and \( k_2 \) can be obtained by equation:

\[
k_1 = n_{\text{data}} \times \left( \text{Entropy}(s_1, s_4) - \text{Entropy}(s_4) \right)
\]

\[
k_2 = n_{\text{data}} \times \left( \text{Entropy}(s_2, s_4) - \text{Entropy}(s_4) \right)
\]

Here \( \text{Entropy}(s_i) \) is the entropy of data obtained by sensor \( s_i \). \( \text{Entropy}(s_i, s_j) \) is joint entropy of data obtained by sensor \( s_i \) and \( s_j \), and \( n_{\text{data}} \) is the number of bits of data in \( s_i \).

The total power consumption is thus given by:

\[
\bar{C} = \sum_{i=1}^{N} C_{\text{path}_i}[s_i, R]
\]

Here \( N \) is the total number of sensors in the network, \( C_{\text{path}_i}[s_i, R] \) is the power consumption when sensor \( s_i \) sends its information through path \( i \) to the sink sensor \( R \). In the next sections we discuss how to find the optimal routing paths that minimize this cost function for compression.

IV. MIN COST TREE ALGORITHM

In this section we first discuss an algorithm with proof for solving the problem of finding the Minimum Cost Tree (MCT). This method can also be extended to provide the optimal solution when multiple sinks (roots) are present. Different from MST, this algorithm is sensitive to root selection which we assume is given.

4.1. Minimum Cost Tree (MCT): Centralized Algorithm

Let us first consider the centralized scenario. We use a dynamic programming approach to solve the problem. The cost function of each edge is defined according to equations (8) and (9) above. The cost function could not be just limited by (8) and (9). In a complex sensor network, more parameters can be introduced to describe the cost of the communication between sensors. We assume that the root is given in this case. Beginning with the root \( R \) (sink node) in set \( \Omega \), \( \Omega \) will grow until it contains all nodes in the sensor network according to the given cost function. The cost function of \( R \) to itself is zero. The cost of every other sensor is set equal to the sum of the cost function of the corresponding edge to that sensor and the cost function of that sensor which it will be connected to. At each stage, one of the sensors, selected from the immediate neighborhood sensors of any node in set \( \Omega \) that are not yet in \( \Omega \), is selected and inserted into set \( \Omega \). This sensor should have minimum cost function among all the sensors that do not yet belong to the set \( \Omega \).

![Fig. 2 Optimal path found by the MCT algorithm.](image)

Proof of optimality: If \( C_{\text{path}_i}(s_i, R) \) is minimized for each \( i \), the sum of costs (11) will be minimized at the same time. So we only need to prove that the path given by this algorithm will minimize the cost for \( s_i \) through that path by contradiction. Suppose path \( A = [a_1, a_2, a_3, \ldots, a_m] \) is the path selected by our algorithm as shown in Fig. 2. If this is not an optimal path for the given cost function, there must exist a path \( B = [b_1, b_2, b_3, \ldots, b_n] \) in Fig. 2 that has a lower cumulative cost than path \( A \). If \( C_{\text{path}}(x, y) \) is defined as the cost function from node \( y \) to \( x \) using path \( D \), the previous case should satisfy: \( C_{\text{path}}(R, S) < C_{\text{path}}(R, S) \). Thus, we have the following:

\[
C(R, S) = C(R, b_n) + C(b_n, S)
\]

\[
C(R, S) = C(R, a_n) + C(a_n, S)
\]

Finally, this inequality should be satisfied:

\[
C(R, b_n) + C(b_n, S) < C(R, a_n) + C(a_n, S)
\]

We rely on the two properties to complete the proof:

Property 1: When extending the path from \( a_n \) to \( S \) by our algorithm, \( b_m \in \Omega \). If sensor \( a_n \in \Omega \) and \( b_m \notin \Omega \) during tree spanning, the path could not extend from \( a_n \) to \( S \) according to our algorithm because of the inequality:

\[
C(R, b_n) < C(R, b_n) + C(b_n, S) < C(R, a_n) + C(a_n, S)
\]

Property 2: When \( b_m \in \Omega \) and \( a_n \in \Omega \), \( S \) will be connected to the root \( R \) through \( b_m \) according to our algorithm because of the inequality:

\[
C(R, b_n) + C(b_n, S) < C(R, a_n) + C(a_n, S)
\]

From property 1 and 2, path \( B \) will be selected instead of path \( A \) by our algorithm. This however is in contradiction to our supposition that path \( A \) was selected. This algorithm will find the path which minimizes the total cost sum from all sensors to root. The tree obtained by this algorithm is called Minimum Cost Tree (MCT).

The algorithm can also be applied where multiple sinks are present in the sensor network. In order to handle this...
case, we must initialize the set \( \Omega \) such that all sinks are initially contained in the set and the cost to them set to zero. This in essence creates a dummy sink with 0 cost that is connected to all the actual sinks present in the network.

### 4.2 Minimum Cost Tree: Distributed Algorithm

Now that we have the basic centralized algorithm in place, let us briefly list how it can be implemented under a distributed computation model:

**Initialization:** The cost of each sensor to the sink sensor \( C(s_{\text{sink}}, s_i) \) is infinite except the sink sensor. The cost of the sink sensor is zero. The sink sensor broadcasts a message to its immediate neighbors with the minimum cost to any sink, which is 0 to itself in this case.

**Main operation:**

While \( s_i \) receives the message from \( s_j \):

- If message contain cost \( C(s_{\text{sink}}, s_j) \) and \( C(s_{\text{sink}}, s_j) + C(s_j, s_i) < C(s_{\text{sink}}, s_i) \)

  \[ C(s_{\text{sink}}, s_i) = C(s_{\text{sink}}, s_j) + C(s_j, s_i); \]

  Set \( s_j \) as \( s_i \) parent sensor;

- \( s_j \) broadcast a message to its directly connected neighbor sensors with \( C(s_{\text{sink}}, s_j) \);

### V. EVALUATION

In order to gain a deeper understanding on the tradeoffs and performance issues involved in the MCT data collection structure we have presented, we use a simple and efficient data compression schedule in our simulations. The sensor networks we consider, observe several light sources that may have a constant velocity in a given direction. The intensity of the light source may change over time. Ignoring the directionality of the sensors, the simulated light intensity amplitude detected by each sensor \( j \) is given by:

\[ \text{data}_j(t) = \sum_{i=1}^{K} \frac{P_i(t)}{\text{dist}(L_i(t), S_j)} + \epsilon_j \]  

(20)

Here, \( K \) is the total number of light sources, \( P_i(t) \) is power of light \( i \) and \( \text{dist}(L_i(t), S_j) \) is distance between light \( i \) and sensor \( j \). \( \epsilon_j \) is the measurement noise, which is different in each sensor. The data will be digitized to an \( N \) level digital signal. \( N \) is set to 256 in our simulation.

We use two methods as reference in performance evaluation of the MCT algorithm. The first is a simple cost minimum spanning tree (SCST) that uses the MST method with cost function defined by the distance between two sensors: \( C(s_i, s_j) = \text{dist}(s_i, s_j) \). The second uses the MST algorithm but with the cost function we use, as defined by equation (11). Each sensor will transmit information to the sink through the path given by MCT, SCST and MST. If sensor \( i \) sends data through path \( i \rightarrow p_1 \rightarrow \ldots \rightarrow p_n \rightarrow \text{sink} \) to the sink sensor, the original data in sensor \( i \) will be compressed by arithmetic coding and sent to \( p_1 \). In the sensor \( p_i \), the difference between data in sensor \( i \) and data in sensor \( p_i \) will be calculated and compressed by arithmetic coding. Only difference between data in sensor \( i \) and data in sensor \( p_i \) will be sent through \( p_i \) to sink to save power. Finally, the power consumption of all sensors in the sensor network is obtained and compared among the three methods.

### 5.1. Evaluating MCT with respect to SCST and MST

As expected, in all simulations, MCT requires less power than SCST and MST. The percentage of power consumption saving for MCT, which is compared with method \( A \) (either SCST or MST), is given by:

\[ \text{per} = \left( \frac{PC_A - PC_{\text{MCT}}}{PC_A} \right) \times 100 \]  

(21)

Here, \( PC_A \) is power consumption of method \( A \) and \( PC_{\text{MCT}} \) is power consumption of MCT. The sensor networks in this simulation have 100 sensors with coordinate \([x, y, z]\) randomly located in three dimensional space \([-30, 30; -30, 30; 0, 0.5]\). Each sensor has at most eight neighbors in this simulation. The number of lights sources is four. The simulation is run in three different sets, each with 1000 random cases. Set A has same light sources for all cases. The sensor network in set A is randomly generated for each case. Set B has the same sensor network for all cases. The light sources in set B are randomly generated for each case. Set C has randomly generated sensor networks and light sources in each case. As shown in Table 1, the MCT algorithm can save total power consumption in the sensor network by more than 60% on average than SCST and MST in all three sets. The standard deviation (std dev) is about 11%, indicating that the performance improvement of MCT is stable even through the minimum performance improvement is just 12.41% in all 3000 cases.

### 5.2. Multiple Roots

Let us now consider the case where multiple sinks are present in the network. For any given sensor network, we evaluate the performance of MCT with one, two, three, four, five and six sink sensors. The sensor network in this simulation contains 120 sensors. The sink sensors are randomly selected in this simulation. A total of 400 instances are generated and evaluated for each case. Fig. 3 shows the average power consumption of MCT with different number of sink sensors. All cases are sorted according to the average power consumption of one sink sensor. As expected, lower power consumption is needed with more sink sensors. The improvement between one sink sensor and two sink sensors are larger than improvement between two sink sensors and three sink sensors, and so on. We can say that MCT algorithm can work well at multiple sinks sensor network with this result.

### Tab 1. Compare power saving of MCT to SCST and MST

<table>
<thead>
<tr>
<th>Set</th>
<th>SCST %</th>
<th>MST %</th>
<th>SCST %</th>
<th>MST %</th>
<th>SCST %</th>
<th>MST %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set A</td>
<td>Mean</td>
<td>60.78</td>
<td>64.40</td>
<td>64.25</td>
<td>67.45</td>
<td>63.06</td>
</tr>
<tr>
<td></td>
<td>Std dev</td>
<td>11.86</td>
<td>10.80</td>
<td>13.58</td>
<td>11.40</td>
<td>13.83</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>84.57</td>
<td>87.03</td>
<td>86.09</td>
<td>86.47</td>
<td>85.44</td>
</tr>
<tr>
<td></td>
<td>Min</td>
<td>16.56</td>
<td>28.87</td>
<td>12.41</td>
<td>25.61</td>
<td>9.66</td>
</tr>
</tbody>
</table>
5.3. Real Sensor Data

In order to evaluate the performance of the MCT-based compression routing approach, we consider the real data collected by a 16 node light sensor network. The 16 sensors are located in different positions of a room with two windows. Each sensor can detect the brightness of sunlight as well as other artificial lights which may be present. The sensor data was obtained on September 14, 2004 by recording the sunlight level at the various positions in the room during the day. Two data sets from different time periods during the day are used. Data A is collected with rate 12.53 sample/second during the late afternoon from 4:55PM to 6:17PM. The data in all sensors range between [390, 490]. Most sensors have similar and high correlative data in data set A. Data B is collected at a rate of 12.63 sample/second from 6:13PM to 6:59PM at dusk. Data set B values range between [20, 220]. The data correlation of data set B is less than that of data set A. Fig. 4 (a) and (b) show power consumption of MCT, SCST, and 1-Beacon with 16 given sink sensors. Improvement for set B is less than that of set A because set B is more noisy and non-correlative than A.

VI. CONCLUSION

In this paper, we proposed a compression-centric data collection algorithm for use in wireless sensor networks. The algorithm relies on the construction of a Minimum Cost Tree (MCT), using neighbor data correlation as the optimization heuristic along the tree. By selecting routes that traverse nodes with higher degrees of data correlation, it is possible to achieve superior compression results and thus communication power savings. We presented an efficient centralized algorithm based on dynamic programming and proved its optimality. We then presented an optimal distributed implementation of the algorithm that relies on token passing to compute the MCT. In addition to the algorithmic and technical discussions, we compared the MCT approach to minimum spanning tree algorithm variations using an arithmetic coding scheme for data compression. Simulated data as well as actual data from a real sensor network were used to validate and demonstrate the performance of the new compression-centric routing scheme.

Fig. 3 Average power consumption of MCT with different number of sink sensors

Fig. 4 Power consumption of MCT, SCST, and 1-Beacon with given sink sensor for real light sensor data sets A and B.

REFERENCE