Cluster Sizing and Head Selection for Efficient Data Aggregation and Routing in Sensor Networks

Hao Chen
Electrical and Computer Engineering Department
University of Wisconsin Madison
Madison WI, USA
hao@cae.wisc.edu

Seapahn Megerian
Electrical and Computer Engineering Department
University of Wisconsin Madison
Madison WI, USA
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 terms of energy consumption and latency costs, in large scale wireless networks. 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 compression and fusion techniques have been proposed and are currently being investigated. In addition to the compression and fusion algorithms, the topology of the aggregation, e.g. the clusters and routes used, can play a significant role in the achievable compression rates.**

In this paper, we investigate the problem of cluster formation for data fusion by focusing on two aspects of the problem: (i) how does one estimate the number of clusters needed to efficiently utilize data correlation of sensors for a general sensor network, and (ii), given the number of clusters, how does one pick the cluster-heads (sinks of information) to cover the sensor network more efficiently. We start by first analytically deriving and analyzing the number of required cluster heads. We then propose an algorithm for the head selection. Simulation results are used to investigate the performance of the algorithm compared to exhaustively found optimal solutions which show that significant improvements in energy efficiency of the fusion algorithms can be obtained through minimal efforts spent on optimizing the cluster head-selection process.

Keywords: sensor network, sink selection, data compression, power consumption

I. Introduction

Sensor networks have emerged as a fundamentally new tool for monitoring inaccessible environments such as nondestructive evaluation of buildings and structures, contaminant tracking in the environment, habitat monitoring, and surveillance in military zones. Many autonomous, resource-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 a collection of networked devices or clusters there of. 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 relatively expensive and unattractive in comparison.

Although relying on the data remaining distributed in the network is often 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. For example, W. Heinzelman et al propose LEACH [5], which randomly selects cluster-heads and provides data fusion (aggregation) in each hop to the head to reduce energy consumption. Mo Chen et al [18] demonstrate the importance of data aggregation for energy efficiency by showing that using data aggregation with LEACH can increase the lifetime of the network. Pattern et al [8] show that the routing relative to data correlation can improve data fusion performance and provide better energy efficiency. They also analyze 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.

The works mentioned above investigate and show how data aggregation can reduce power consumption in, and extend the life time of, sensor networks. Furthermore, the routing algorithms based on data correlation in the network can provide more efficient data fusion and help in reducing the overall power consumption. However, currently it is not clear how to apply routing algorithms based on data correlation in general sensor networks. For example, nodes in real sensor networks are often randomly located in the area of interest. Due to obstacles, device orientation, and observed phenomena changes, the relation of sensors cannot be simply given by the physical distances of sensors. In order to tackle the larger problem, our goals in this paper are two fold: (i) How to estimate the number of clusters needed to efficiently utilize data correlation of sensors for a general sensor network and (ii) how to pick cluster-heads according to a given number of clusters to cover the sensor network more efficiently. The answers to these questions are crucial in devising routing and other algorithms that rely on cluster-based topologies to perform data fusion, compression, and aggregation.

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. Reference [9] presents 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 constraint is based on the maximum transmission distance with minimum hops 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 [7] provides a solution for delivering messages from any sensor to a sink sensor 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 neighboring 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 investigated in [17]. There, the sensors in the network are grouped into several clusters and sensor data in the same cluster are gathered, aggregated, and combined during the data collection. The sink sensor of each cluster is regarded as a node at a higher level of the data collection hierarchy. The sink senor 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 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 senor 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 during aggregation.

The potentially heavy overlap in data and distributed nature of sensor networks requires efficient and fully distributed data compression techniques without requiring the sensors to talk to one another during data compression. Distributed source coding (DSC) is the fundamental concept in information theory applicable to this problem. Reference [2] reviews the main ideas, provides illustrations, and gives the intuition behind the theory that enables this framework. Reference [4] proposes distributed source coding (DSC) to reduce energy consumption in a sensor network, showing an estimated 10%-65% improvement. An adaptive filtering scheme is used to continuously estimate the relevant correlation in the measured data. 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.

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 [10], [11], [12], and [13] present several examples of approaches for energy aware and hierarchical clustering and data collection algorithms. Reference [14] 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 of [14] show that localized cluster head selection strategies can significantly impact the global performance of the network in different ways.

In this paper, we first estimate the optimal cluster size and then present a method to select the cluster heads to cover whole sensor network so that total information transmitted through the sensor network is minimized.

III. PRELIMINARIES

A. Distributed Source Coding in Sensor Networks

References [1], [2], and [16] 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 large number of distributed sensors that transmit to a sink sensor independently. The redundant information from the different sensors will require more hardware resources and communication bandwidth. The relation between the information of the sensor network and individual sensors can be given by information theory as follows:

$$H(x_1, x_2, \ldots, x_N) \le H(x_1) + H(x_2) + \ldots + H(x_N) \quad (1)$$

The entropy of sensor network can be expressed by:

$$H(x1, x2, \ldots, xN) = H(x1) + H(x2| x1) + \ldots + H(xN| x1, x2, \ldots, xN\text{-}1) \quad (2)$$

Here $H(x_2| x_1)$ is entropy of x_2 given information of x_1 and $H(x_2| x_1) \le H(x_2)$. $H(x_N| x_1, x_2, \ldots, x_{N-1})$ is entropy of x_N given information of $x_1, x_2, \ldots, x_{N-1}$ and $H(x_N| x_1, x_2, \ldots, x_{N-1}) \le H(x_N)$. 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 final data size for transition. References [3] and [4] discuss how to estimate the correlation between the sensors and use those correlations to decide how many bits to send by each sensor.

We assume that each sensor in the sensor network has the same hardware architecture, sensor equipment, and data sampling frequency. So each sensor obtains the same amount of information in the local region during a given time period. However, not all of that information should be transmitted to the sink sensors due to limited hardware resource and potentially redundant information between all of the sensors. In order to apply DSC in the sensor network, each sensor has to know the relevant correlation of all sensors in network. The relative cost to learn and store the correlation information is particularly high for sensor with limited power. Thus, although not a practical solution here, DSC can be treated as the optimal policy providing a lower-bound on the bit-hop metric.

B. Power consumption in wireless communication system

The radio model discussed in [15] can be used to evaluate power consumption of data transmission. In this model, a radio dissipates E_{elec} (50 nJ/bit), defined for the transmitter or receiver circuitry, and ε_{amp} (100 pJ/bit/m^2), defined for the transmitter amplifier. We assume all sensors have transmit-power control and can use just the minimum required energy to send information to the intended recipients. The sensors could turn off their transmitter and receiver to avoid receiving uninteresting information and save energy. This is motivated by the fact that receiving is also a high cost operation in the wireless communication systems in our aim. The equations used to model power consumption of a sensor node for communication are given below.

The power consumption for transmitting sensor:

$$E_{Tx}(k,d) = E_{elec} \cdot k + \varepsilon_{amp} \cdot k \cdot d^2 \tag{3}$$

The power consumption for receiving sensor:

$$R_{Tx}(k,d) = E_{elec} \cdot k \tag{4}$$

Here d is the distance between two sensors, k is the number of bits of information sent, and E_{elec} and ε_{amp} are the constants as previously defined. The total power consumption cost is given by:

$$\begin{aligned} E_{total} &= E_{Tx}(k,d) + R_{Tx}(k,d) \\ &= \left(2E_{elec} + \varepsilon_{amp} \cdot d^2\right) \cdot k \end{aligned} \tag{5}$$

The power consumption is a second order function of distance. So the data routing with multiple shorter nearby hops will typically be more efficient than directly transmitting between two far sensor nodes. The power consumption is also a linear function of k which is bits of information transmitted through the sensor network.

C. Routing Relative to Data Aggregation

Reference [4] proposes distributed source coding (DSC) to reduce energy consumption. With an ideal DSC schedule, each sensor will send exactly the data which would not be transmitted by any other sensor. To apply DSC in a sensor network, each sensor has to know the relevant correlation of all sensors in the network, making it impractical in a real sensor network. As mentioned above, LEACH randomly selects cluster-heads and provides data fusion in each hop to reduce energy consumption. Pattern et al [8] show that the routing relative to data correlation can improve data fusion performance and provide better energy efficiency. They analyze the data correlation model in one dimension with uniformly spaced sensors.

The simple example below illustrates the main concept motivating our approach here.

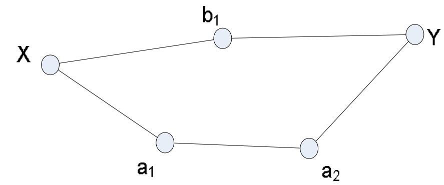


Figure 1. Data correlation and power consumption example.

Fig.1 shows a simple network composed of 5 sensors. Suppose that data from sensor X is to be transmitted to sensor Y. Here, path X→b_1→Y will be better than path X→a_1→ a_2→Y if total distance or hop count metrics are used. However, assuming $|X-b_1|<2^5$, $|X-a_1|<2^1$ and $|a_2-a_1|<2^1$, we can just send the difference between two sensors instead of the original data. In this situation, less data will be transmitted through path X→a_1→ a_2→Y than path X→b_1→Y. In real system, it's better to find routing path according to data correlation. We elaborate more on this in the next section.

IV. Routing Schemes and Clustering

Given a graph representing the communication topology and a cluster head (root), one can choose different routing schemes to optimize energy efficiency. With DSC, each sensor can just send information to root through the shortest path without any aggregation during transmition because there is no redundancy of information in the data sent by any two sensors. Thus, the routing with DSC will be straight forward as the problem reduces to the well understood shortest-path routing. However, as shown in our example in Fig.1, shortest path is not the most efficient technique for data compression. Our routing policy in this paper is based on data aggregation methods as discussed in [8]. Here, data is aggregated at each hop and routed to the next hop so as to allow for maximum possible aggregation along all paths. Consequently, the number of clusters and selection of cluster heads play a very important role in obtaining energy efficient solutions [14].

A. Number of Clusters in a Sensor Network

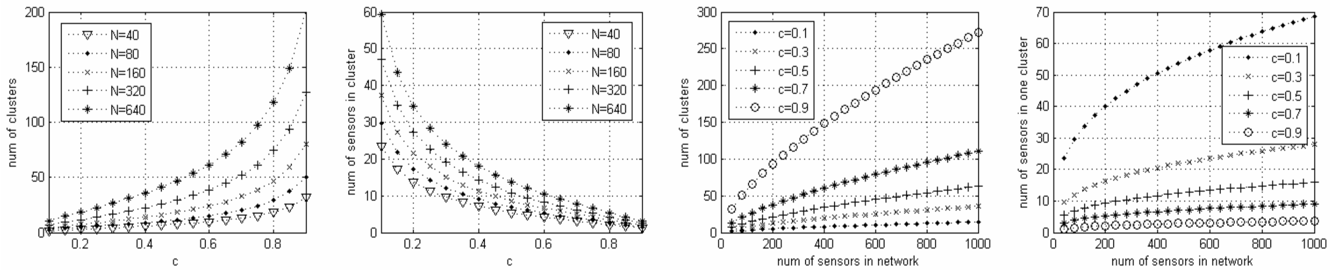
Consider the sensor network with N randomly located sensors. Suppose the average distance from a sensor S_i to the other sensors which it can directly communicate with, is d_i. Entropy of the sensor S_i is H_i. The information that is only provided by the sensor S_i can be expressed by entropy:

$$H(S) - H(S \cap \bar{s}_i) \tag{6}$$

where $H(S)$ is entropy of total sensor network; $H(S \cap \bar{s}_i)$ is entropy of total sensor network excluding sensor S_i. The coefficient c_i is defined as the percentage of unique information of sensor S_i compared to $H(S_i)$, the complete information provided by S_i:

$$c_i = \frac{H(S) - H(S \cap \bar{s}_i)}{H(S_i)},\ 0 < c_i \le 1 \tag{7}$$

The coefficient c_i can express the degree of correlation between a sensor and its neighborhood sensors. In real sensor networks, the data from two sensors will be almost decorrelative when both sensors are located far away from each other.



(a) (b) (c) (d)

Figure 2. Analytical curves for number (K) and sizes of clusters (s) with respect to the degree of correlation (c) and the size of the network (N).

Each sensor S_i can estimate c_i according to the correlation between current sensor and its nearby sensors.

We consider $K=N/s$ clusters each consisting of s sensors. The cluster head for each cluster is located at the center of cluster. The total number of bits-hop cost for the whole sensor network is expressed as:

$$E_{whole} = \sum_{i=1}^{K} \left(E_{intra}(i) + E_{extra}(i)\right) = K\left(E_{intra} + E_{extra}\right) \quad (8)$$

where $E_{intra}(i)$ and $E_{extra}(i)$ are the bit-hop cost within cluster i and the bit-hop cost for cluster i to the sink respectively. E_{intra} and E_{extra} are the average bit-hop cost within the clusters and the average bit-hop cost from the clusters to the sinks. We can obtain expressions for each of these:

$$\begin{aligned} E_{intra} &\propto ((s-1)Hc)(d\sqrt{s}) \\ E_{extra} &\propto (H + (s-1)Hc)(d\sqrt{N}) \end{aligned} \quad (9)$$

where H, c and d is the average of $H(s_i)$, c_i and d_i. $(s-1)Hc$ and $d\sqrt{s}$ are average number of bits of all sensors except the head of cluster in a cluster, and average distance from the sensors to the head of cluster respectively. $H+(s-1)Hc$ and $d\sqrt{N}$ are average number of bits of the head of cluster after data fusion and average distance from the head of cluster to the root. The number of sensors in each cluster is much larger than one (s-1≈s when s>>1) in most case. So, we have:

$$\begin{aligned} E_{whole} &= KE_{intra} + KE_{extra} \propto \sqrt{s}HcdN + s^{-1}H(1-c)dN\sqrt{N} \\ &\quad + HcdN\sqrt{N} \end{aligned} \quad (10)$$

The optimum value of the cluster size s_{opt} can be determined by setting the derivative of the above expression equal to zero:

$$\frac{\partial E_{whole}}{\partial s} = 0 \quad \Rightarrow \quad \frac{c}{2\sqrt{s}} - s^{-2}(1-c)\sqrt{N} = 0 \quad (11)$$

The optimum number of clusters can be expressed as:

$$K_{opt} = \frac{N}{s_{opt}} = \left(\frac{cN}{2(1-c)}\right)^{\frac{2}{3}} \quad (12)$$

The optimum number of clusters K_{opt} depends on the number of sensors in the entire sensor network and the degree of correlation c. Fig.2 shows how different number of clusters and cluster sizes perform across a range of correlation levels and sensor network sizes. As expected low correlation levels require small cluster sizes while high correlation levels require larger cluster sizes. Also the number of clusters is relative to sensor network size. In other words, the larger the sensor network (larger N) the more clusters are required to apply optimal data aggregation.

B. *Selecting the Heads of Clusters*

The location of cluster head is very important in minimizing the total cost of power consumption. If the cluster heads are located too close to each other, the sensors that are far away from them will have to transmit their data using more hops to reach any one of the cluster heads, making the transmissions wasteful. Conversely, if the cluster heads are located too far from each other near the boundaries of the network, again routing data to them from the sensors will prove to be inefficient. The problem is further complicated in realistic deployments since all sensor node location information is not available centrally. Here we propose an algorithm to select suitable cluster heads which cover more regions with smaller average total communication distances to the sensors in their clusters.

It's not difficult for each sensor to know how many hops it is away from another sensor. This process can be localized for larger networks by setting a limit on the maximum number of allowable hops. As an example, Fig.3 shows sensors in a network with the maximum number of minimum hops (max-min hops), by which each can reach *any* other sensor. For example, a sensor with a number 2 in Fig.3 can reach all other sensors with only 2 hops while a sensor with a number 5 requires as many as 5 hops to reach all other sensors. The sensors with smallest of this max-min hop number are said to be located at the center of the network. In general, more than one such sensor at or near the center will be obtained. An algorithm for determining the max-min hop counts at each sensor can be divided into the following three stages:

i) **Initialization:** For each sensor, the hop number to all other sensors is infinite and the distance to itself is zeros:

$$n\_hop_0(i,j) = \begin{cases} 0 & i = j \\ \infty & i \neq j \end{cases} \quad (13)$$

ii) **Find number of hops to any other sensor:** For each step, the hop number from sensor i to sensor j is updated as:

$$n\_hop_n(i,j) = \min \begin{pmatrix} n\_hop_{n-1}(i,j), \\ n\_hop_{n-1}(k_1,j)+1, \\ \cdots, n\_hop_{n-1}(k_L,j)+1 \end{pmatrix} \quad (14)$$

Here k_1, …, k_L are the neighboring sensors of sensor i. L is the number of neighbors.

iii) **Find sensors in the center:** When each sensor has the min hop number to all other sensors, find max hop number:

$$n\_max_i = \max(n\_hop(i,k), k = 1, \cdots N) \quad (15)$$

After that, each sensor will exchange information about its n_max_i and the sensor with minimum n_max_i will be a potential sink sensor.

The process outlined above is one method of determining nodes that are in, or near the "center" of the network. The Cluster Head Selection (CHS) algorithm we propose uses this information as a starting heuristic based on the n_max_i of each sensor. Note that it is not necessary for nodes to determine their exact locations with respect to the center of the network. However, in order to find a suitable solution, we not only must ensure that the roots are near the center (have a means of estimating such), but also that they are adequately away from each other so as to avoid the problem of having cluster heads that are located in a very small region. To do this, we utilize the following cost function:

$$f(s_i, i = 1,..., K) = \sum_{i=1}^{K} n\_max_i - \lambda \sum_{i=1}^{K} d(i, i_{near}) \quad (16)$$

where $d(i, i_{near})$ denotes the distance between cluster head s_i and another nearest cluster head. Given the parameter λ, the cluster heads can be obtained by minimizing the function $f(s_i, i = 1,..., K)$. Fortunately, it is often not necessary to search all possible cluster head combination to find a good solution. To solve this minimization problem in our simulations for example, we randomly select K sensors as initial cluster heads. We then perform a "move" operation which swaps a selected cluster head with a neighboring node. The set with small $f(s_i, i = 1,..., K)$ is kept as the potential cluster head assignment. If no moves lead to lower cost solutions, we randomly try a different set of K sensors and compare it with previous set. With randomly selecting initial cluster head assignments like this, it is possible to change cluster heads after a certain time period to avoid cluster heads that exhaust their energy supplies prematurely. In the simulations below, we use 30 iterations to converge on the final result while λ is set equal to 0.5.

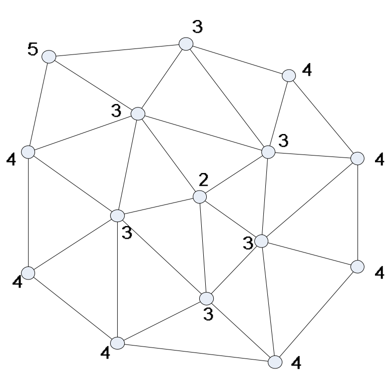


Figure 3. Sensors with maximum hop number to any other sensor

V. SIMULATION RESULT AND EVALUATION

In order to illustrate and evaluate the performance of this algorithm, let us consider a network with 200 light sensor nodes, with 4 light sources located above the sensors with a constant movement speed in a randomly chosen direction. The power of each light source changes as it moves. The light amplitude which is detected by each sensor j is given by:

$$data_j(t) = \sum_{i=1}^{K} \frac{P_i(t)}{dist(L_i(t), S_j)^2} + n \quad (17)$$

Here, K is the total number of light sources, $P_i(t)$ is power of light i and $dist(L_i(t), S_j)$ is distance between light i and sensor j, n is system noise (different in each sensor). The data signal is digitized to 256 digital levels. Thus, most of the sensors have large data correlation while a few are uncorrelated. This simulation evaluates the cluster head selection (CHS) algorithm and includes a total of 40 cases. Each sensor can directly communicate with up to eight of its nearest sensors. Both sensor network and light sources are randomly generated in each case. With our CHS algorithm, cluster head can be selected with the given cluster number (2 and 3 in this simulation). All sensors will be connected to the cluster head by a routing driven by data aggregation metric as previously described. Each sensor will thus transmit the data through the corresponding path (lowest cost) to the cluster head. At each passing hop, data fusion is applied to reduce data size. The power consumption of the sensor network is calculated using Eq.(5). The average power consumption of one sensor data to corresponding cluster head is calculated to evaluate performance of the cluster heads.

For comparison, we also try all possible cluster heads using a exhaustive search. The optimal cluster head selection with the minimum power consumption found through this method is reported for reference. In addition, average and standard deviation is calculated with result of all possible cluster heads. Fig.4 (a) and (c) shows the power consumption with two and three cluster heads selected by CHS correspondingly. The power consumption of CHS is less than mean-STD of all possible cases in most situations. The performance of CHS is very close to the optimal result in most cases. Fig.4 (b) and (d) also show increasing power consumption of CHS compared with optimal result for two and three clusters correspondingly. On average, power consumption increases by 8.50% and 9.52% for the two and three clusters when compared to the optimum.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have presented a practical approach for estimating the number of clusters and selecting cluster heads, to efficiently utilize data correlation for aggregation driven routing. The results indicate that here, minimal optimization efforts using heuristic starting solutions and probabilistic search methods can yield near optimal results when compared to exhaustive search methods for cluster head selection. The cost of this optimization effort can be amortized in the long run in the savings that are achieved by the increased efficiency in data aggregation, fusion, and compression phases that will potentially require far fewer total numbers of bits to be transmitted.

There are several interesting and challenging questions that yet remain to be answered here. For example, modeling and quantifying the savings in data aggregation that result through the optimizations at the topology formation and management phases pose a number of challenges. Furthermore, the routing used in this paper allows for maximum possible aggregation at each hop which may not always be practically possible to implement. The correlation between sensors also considers spatial relations among sensors. Temporal correlation in sensor data can also reduce power consumption which opens other avenues for future study.

REFERENCES

[1] S.S. Pradhan and K. Ramchandran. "Distributed source coding: Symmetric rates and applications to sensor networks." *IEEE Data Compression Conference (DCC)*, March 2000.

[2] S.S. Pradhan, J. Kusuma, and K. Ramchandran. "Distributed Compression in a Dense Microsensor Network." *IEEE Signal processing Magazine*, March 2002.

[3] J. Chou, D. Petrovic, and K. Ramchandran. "Tracking and Exploiting Correlations in Dense Sensor Networks." *Signals, Systems and Computers,* 2002.

[4] J. Chou, D. Petrovic, and K. Ramchandran. "A Distributed and Adaptive Signal Processing Approach to Reducing Energy Consumption in Sensor Networks." *INFOCOM 2003, Volume: 2*, 2003.

[5] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. "Energy-Efficient Communication Protocol for Wireless Microsensor Networks." *In the Proceedings of the Hawaii International Conference on System Sciences*, Maui, Hawaii, January 4-7, 2000. (LEACH)

[6] S. Lindsey, C. Raghavendra, and K. Sivalingam. "Data Gathering in Sensor Networks using the Energy*Delay Metric." *Parallel and Distributed Processing Symposium.,* 23-27 April 2001.

[7] F. Ye, A. Chen, S. Lu, and L. Zhang. "A Scalable Solution to Minimum Cost Forwarding in Large Scale Sensor Networks." *10th International Conference on Computer Communications and Networks (ICCCN2001)*, Scottsdale, Arizona USA. October 15-17, 2001.

[8] Sundeep Pattern, Bhas kar Kris hnamachari, Ramesh Govindan, "The Impact of Spatial Correlation on Routing with Compression in Wireless Sensor Networks.", *IPSN*, April 26-27, 2004.

[9] M.A. Youssef, et. al. "A Constrained Shortest-Path Energy-Aware Routing Algorithm for Wireless Sensor Networks." *Wireless Communications and Networking Conference (WCNC'02),* 2002.

[10] S. Bandyopadhyay and E.J. Coyle. "An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks." *INFOCOM 2003. Volume: 3* , 30 March - 3 April 2003.

[11] F. Ye, G. Zhong, S. Lu, and L. Zhang. "A Robust Data Delivery Protocol for Large Scale Sensor Networks." *IPSN*, Palo Alto, CA, Apr. 22-23, 2003.

[12] A. Boukerche, X. Cheng, and J. Linus. "Energy-Aware Data-Centric Routing in Microsensor Networks." *MSWiM'03*, San Diego, California, USA. September 19, 2003.

[13] S. Tilak, A. Murphy, and W. Heinzelman. "Non-uniform Information Dissemination for Sensor Networks." *11th IEEE International Conference on Network Protocols*, 2003.

[14] C. Zhou and B. Krishnamachari. "Localized Topology Generation Mechanisms for Self-Configuring Sensor Networks." *Globecom*, 2003.

[15] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks." *In Proceedings of the Hawaii Conference on System Sciences*, 2000.

[16] Zixiang Xiong, et. al., "Distributed Source Coding for Sensor Networks", *IEEE Signal Processing Magazine*, Sep. 2004

[17] K. Dasgupta, K. Kalpakis, and P. Namjoshi. "An Efficient Clustering-based Heuristic for Data Gathering and Aggregation in Sensor Networks", *Wireless Communications and Networking (WCNC 2003). IEEE*, Mar. 2003

[18] Mo Chen and Mark L. Fowler, "The Importance of Data Compression for Energy Efficiency in Sensor Networks" *Conference on Information Sciences and Systems*, The Johns Hopkins University, 2003

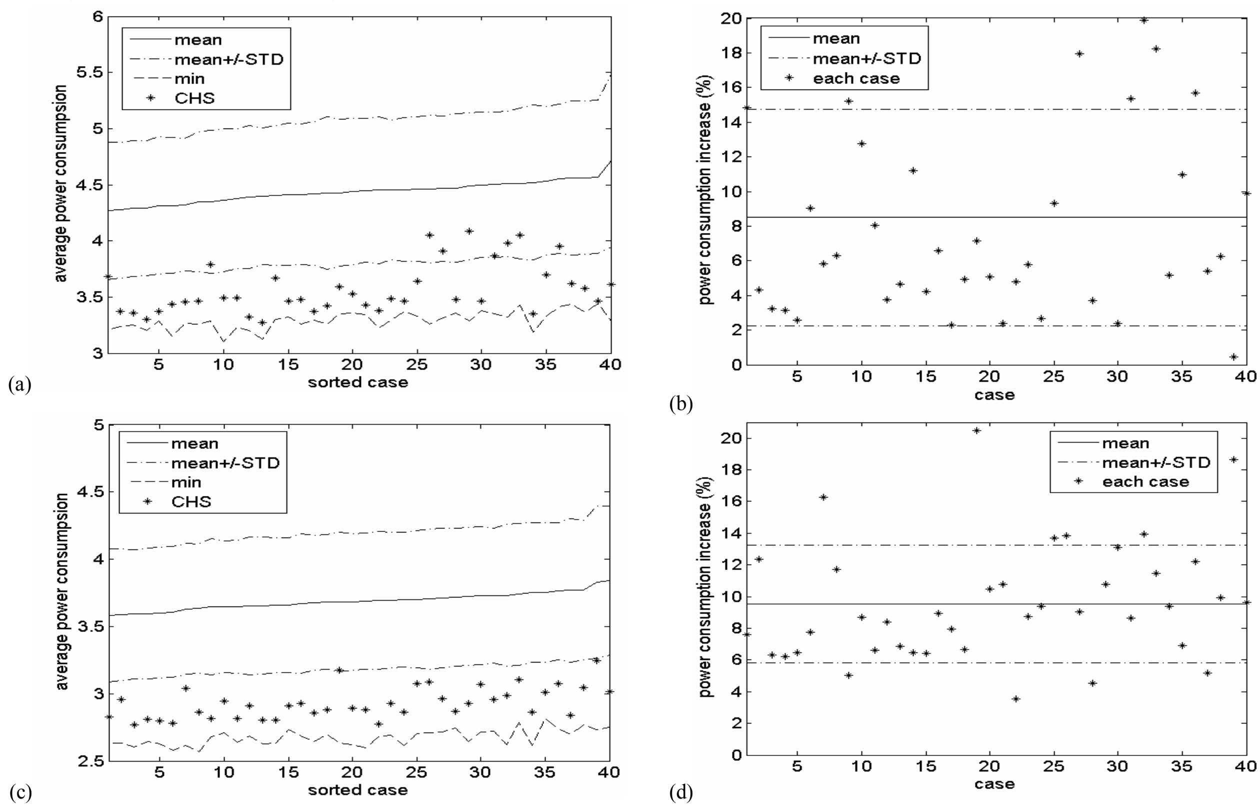


Figure 4. Comparisons in performance of CHS with optimal cluster heads with mean and standard deviation of all case. Figures (a) and (b) are result for two clusters while (c) and (d) are the result for three clusters.