Power Efficiency with Data Compression in Wireless Sensor Networks

Alan C. Wu

***Abstract*—Wireless sensor networks with long lifetime requirements have severe power constraints. This paper attempts to address these power constraints by taking advantage of data correlation between nodes to compress data and conserve energy in data transmission to the base station. A translationally and rotationally invariant correlation model is presented, and data compression ratios are calculated from the relationship between correlation and mutual information. Intuition about the energy trade-offs in data compression is drawn from simulations. Possible extensions of this paper, such as the use of different correlation models, are also discussed.**

I. Introduction

One of the current major research areas in wireless systems are wireless sensor networks. These networks are built to gather a large amount of data in a large spatial region, providing users with information in different points of a region. Wireless sensor networks have the advantage of not needing lines to connect them, making them a viable solution that can be placed in various locations with minimal disturbance to the surroundings.

A major difference between traditional wireless systems and wireless sensor networks is the issue of power constraints. In many applications of wireless sensor networks, the individual nodes are installed in locations under the assumption of expensive replacement. The unlikelihood of being replaced, combined with the expectation of a long working lifetime, means the nodes are limited by their power supplies. One of the main goals for wireless sensor network design are developing strategies such that these nodes can prolong their lifetime given a fixed amount of energy.

One way to overcome this is to increase the amount of energy available. Researchers have looked to allow sensor nodes to gather energy from their surroundings [1], known as energy harvesting. Another method is to reduce the sensor's power consumption. A paper discusses how this can be done in circuit architectures or power supply design to reduce power-consuming tasks such as leakage current [2]. Sensor networks communication protocols have been proposed to reduce overhead and allow for better flow of data as well [3], [4].

This paper will look at reducing data transmission power with data compression. Data compression at the nodes may provide power savings by only needing to transmit data a short distance within a network to another node which may compress the data with its own. This is feasible considering that data measured between two neighboring sensors may exhibit some correlation. Distributed source coding, where the nodes compress their data individually first, is another possible alternative.

Data compression in wireless sensor networks is not without its disadvantages. While energy is saved in transmitting less power to the base station, additional power must be used for data transmission between nodes, as well as for data compression circuitry. This paper hopes to provide a way of quantifying these trade-offs such that a making a decision on the optimal topology for a given application is straightforward.

Section II will describe the general system model. It includes the characteristics of the environment and the channel. A method of generating the correlation relation between two nodes dependent solely on distance will be described. Sections III and IV detail algorithms for data compression within the sensor network and distributed source coding. These sections will also calculate the power requirements for a target transmission rate. Simulations will be presented along with analysis of the results in section V. Finally, this paper will conclude with a discussion of the ideas shown in the simulations as well as some possible extensions of this paper.

II. System Model

A. Sensor Network Model

The wireless sensor network will be assumed to be a cluster of wireless sensor networks that are close to each other compared to the distance to the base station. This model also assumes that it is infeasible to put a high power relay to reduce the data transmission power from the sensor network. Such scenarios could exist for long term ocean floor sensing or for deep space networks. These scenarios make it difficult and expensive to place a relay between the base receiver and the sensor networks.

One of the parameters then, is the ratio of the distance between the closest edge of the network to the base station d_{base} and maximum distance between nodes d_{node}. This will be useful in approximating how much power can be saved by transmitting the data within the sensor network as opposed to transmitting it directly back to the base station.

This paper will look at performance related to the node density of the network. This is due to the fact that a network can be divided into smaller areas if the data correlation is too small, in which case the data compression scheme will intuitively be less preferable to independent transmission. In these simulations, the wireless sensor network will have a constant bandwidth over which to transmit data signals over, as well as a fixed noise.

B. The Correlation Model

This section provides a model from which we can obtain the correlation as a function of distance. We can imagine that we are given a 2-dimensional field of random values, whose strengths vary inversely with distance. Thus, at any point in the field, a noiseless measurement is a weighted sum of these values:

$$x = \sum_i \sum_j \alpha_{ij} \chi_{ij} \quad (1)$$

In our simulations, we will deal with a finite array of i.i.d. random variables $\chi_{ij} \quad N(0,1)$. The weights $\alpha_{ij} \propto 1/d$, with d being the distance between point ij and x. With this model for the measurements, we can then describe the correlation coefficient between two nodes x and y, each with weights α_{ij} and β_{ij} respectively, as:

$$\rho_{xy} = \frac{E[xy]}{\sqrt{Var[x]Var[y]}} \quad (2)$$

$$\rho_{xy} = \frac{\sum_i \sum_j \alpha_{ij}\beta_{ij}}{\sqrt{\sum_i \sum_j \alpha_{ij}^2 \sum_i \sum_j \beta_{ij}^2}} \quad (3)$$

It should be noted that this is not the only correlation model in which the following simulations can be applied to. The inclusion of Gaussian noise or use of some other correlation coefficient functions, such as those described in [5], can have a different effect. However, this paper will not explore the use of various correlation functions, and instead obtain understanding for this case only. This paper also does not assume data correlation in time. The time-correlation of measurements can be added into the correlation model in future work, but for now will be assumed to be independent at the current sample rate.

C. Channel Characteristics

Another characteristic of the environment are the channels between the network nodes and the network and the base station. In this paper, we will use the power loss model as follows [6]:

$$P_r = \left(\frac{d_0}{d}\right)^{\gamma} P_T \quad (4)$$

We will let $\gamma = 3$ to model the connection between the network and the base station and within the nodes of the network with the inclusion of other path loss factors, $d_0 = 1$ to reduce the number of variables. In addition, we introduce an approximation of the gain in transmitting data to a neighboring node over transmitting data to the base station. This ratio, denoted G in some of the following equations, is a conservative multiplier in received power due to transmitting at a shorter distance, and is given by:

$$G = \left(\frac{d_{base}}{d_{node}}\right)^{\gamma} \quad (5)$$

The channel will be assumed to be AWGN, and an approximation is made that all the nodes within the network are roughly the same distance d away from the base station. The power required to transmit at a target rate R with noise density $\frac{N_0}{2}$ over a distance d units is thus computed from Shannon's equation and the path-loss model:

$$P_t = \left(2^{\frac{R}{B}} - 1\right) N_0 B d^{\gamma} \quad (6)$$

D. Transmission Protocol

We will use a TDMA scheme in which each node has a data output at a rate of r. Each node will be assumed to transmit one measurement every unit of time, with measurements being independent temporally. For a given density of nodes and a constant area, we know the number of nodes k, and so are required to send at a rate of $R = kr$. This substitution will provide the base case to compare our compression schemes against. In the data compression cases, the TDMA scheme splits into two portions. A fraction t of the time is alloted for transmission of data back to the base station. This power P_{T1} is derived from the requirement of sending compressed, and uncompressed in the case where nodes decide to send data directly to the base station instead of a neighboring node, data back to the base station. For scenarios where multiple nodes transmit back to the base station, the compression rate is assumed to be fairly constant and is averaged for simplicity in calculations. The remainder $1-t$ is reserved for transmitting data between nodes in order for compression at a power P_{T2}. With the simplification of a constant inter-node transmit distance equal to the width of the network, each of the nodes observe the same channel conditions to other nodes, which eliminates the need to vary P_{T2} as different nodes transmit. The optimal TDMA scheme under these conditions is the following minimization problem that can be solved by a base station with knowledge of all correlation matrix of all the nodes:

$$min_t \quad tP_{T1} + (1-t)\, P_{T2} \quad (7)$$

The algorithms proposed in the next section will come up with equations for these power usage calculations.

III. Data Compression Algorithms

Given the correlation matrix of the data received at all the nodes, we will find a quantitative relation between data correlation and data compression. We assume that each node has the same amount of information, which can be found as the entropies $H(X)$ and $H(Y)$, to transmit. If we can find fraction of redundant information between two nodes, also known as the mutual information $I(X;Y)$, this fraction can be subtracted from the sum of information from the two nodes, to give the amount $H(X,Y)$ actually needed to be transmitted. This is described with the following equation found in [7]:

$$H(X,Y) = H(X) + H(Y) - I(X;Y) \quad (8)$$

We can also make the observation that the values measured at the nodes are a sum of i.i.d. Gaussian random variables, and so themselves are Gaussian. Furthermore, with the correlation known between two nodes, the mutual information between them can be found as described in [8]. This relationship is given as:

$$I\left(X;Y\right)=-\frac{1}{2}\log_2\left[1-\rho_{X,Y}^2\right] \quad (9)$$

Where $I\left(X;Y\right)$ is the mutual information between the information at nodes X and Y, and $\rho_{X,Y}$ is the correlation coefficient between the values at the two nodes.

The main problem here is that these correlations and mutual information equations are for infinite precision, and so very highly correlated values may lead to a mutual information with more bits than the individual measurements. While this does not happen in our simulations, we will make the assumption that once the correlation gets to a certain point, the two values are represented in the same way by the sensors. We cannot compress data beyond the original information of one of the measurements, and so the mutual information is the minimum between Equation 9 and $H(X)$.

The resolution choice plays a large factor into the compression ratio. For lower resolutions, medium amounts of correlation can lead to relatively large compression ratios, since the least significant bit is still a large bin. For larger resolutions, substantially higher correlation is required to achieve significant compression ratios. This paper uses 6-bit resolution to emphasize the amount of compression for the given model and the differences in the transmit powers. The effect of varying the resolution will also be simulated.

This paper proposes 2 algorithms in choosing how to compress the data.

Algorithm 1: This method finds a node with the highest summation of correlation coefficients with all its neighbors. All the nodes can send data to it, and it can only transmit the extra information back to the base station.

This algorithm goes through each node and finds the total compression it can perform based on its correlation coefficients with other nodes. It assigns the node with the highest compression ratio as the center node, and all other nodes with correlated data will send their data to the center node. We denote the size of this grouping of correlated nodes and center node with an integer x. With a compression ratio of $C_c \leq 1$, we come up with the following constraints.

$$\left(k+x(C_c-1)\right)\frac{R}{k}=t\,B\,\log_2\left(1+\frac{P_{T1}}{N_0Bd^\gamma}\right) \quad (10)$$

$$\left(x-1\right)\frac{R}{k}=\left(1-t\right)B\,\log_2\left(1+\frac{GP_{T2}}{N_0Bd^\gamma}\right) \quad (11)$$

With TDMA, we find that the capacity constraints for transmitting data back to the base station (Equation 10) is the sum of the nodes in the group transmitting at the compressed rate and the uncompressed rate for the remainder. The inter-node data transmission constraint (Equation 11) only applies to the $x-1$ nodes that have to transmit their data to the center node at the uncompressed rate.

Algorithm 2: This algorithm pairs up nodes that have the highest correlation coefficients, and combines the data between those pairs. Each pair then elects one of the two to transmit the data back to the base station. These pairs are found using a greedy algorithm, where the network cycles through each node. If the node is not already paired, then it will find the neighboring node with the highest non-zero correlation, and pair with it. If it cannot find another node that satisfies this condition, it will choose to not undergo compression and will behave as in the no-compression scenario. Although this greedy algorithm is not optimal, it is a fast heuristic with a computational time of $O(n^2)$, compared to a brute-force method with exponential computational time. The constraints for this scheme are as follows:

$$\left(k+2y(C_c-1)\right)\frac{R}{k}=t\,B\,\log_2\left(1+\frac{P_{T1}}{N_0Bd^\gamma}\right) \quad (12)$$

$$y\frac{R}{k}=\left(1-t\right)B\,\log_2\left(1+\frac{GP_{T2}}{N_0Bd^\gamma}\right) \quad (13)$$

The constraints use similar reasoning to that of *Algorithm 1* for y pairs of nodes, and assumes that each pair will have the same average compression ratio of C_c. The constraints take into account that not all nodes will be paired, and those will instead behave as in the uncompressed case.

Given these two algorithms, *Algorithm 2* intuitively has a higher data compression rate since it has higher correlation coefficients, so the compression ratio will be higher. In addition, *Algorithm 2* has the advantage that the data destined for the base station is distributed among at least half of the nodes in the network as compared to the concentration of data at a center node. In practical sensor networks, this will likely lead to a shorter lifetime for a sensor network applying *Algorithm 1*, since it will experience a constant decrease in the number of nodes as the center nodes run out of energy.

IV. Use of Distributed Source Coding (DSC)

DSC provides another method of reducing the amount of bits required for transmission of the measurements from the network. In this scheme, nodes are given the correlation, and use this knowledge to compress their data individually, and then send those bits to the base station. The theoretical compression limits derived by Slepian and Wolf for DSC approach that of data compression with knowledge of the other nodes' data [9]. Many implementations have come close, such as ones using LDPC codes [10], and Turbo codes [11], but have not achieved the theoretical limit. Thus, the paper will use the DSC transmit powers as theoretical achievable bounds. Another practical limitation is that many of the codes rely on fairly high correlation between nodes in order to work. In sparser sensor networks, the low correlation between nodes will likely make DSC impractical to use.

The Slepian-Wolf Theorem states that in the limit of infinite time, two correlated nodes only need to send the same amount of information as if they had knowledge of the other's data. In this scenario, we pair nodes with high correlation together again and find the amount of data compression possible. The equations to find the compression ratio will be similar to that of *Algorithm 2*. This leads us to use the following equation for calculating the required power for transmission with the compression ratio C_c derived from *Algorithm 2*:

$$RC_c=B\log_2\left(1+\frac{P_T}{N_0Bd^\gamma}\right) \quad (14)$$

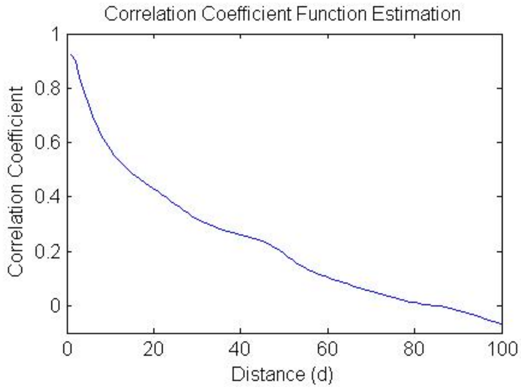


Figure 1: The simulated results of the correlation coefficient as a function of distance based on the correlation model

V. Simulation and Analysis

A. Correlation Modeling

First, we introduce our correlation function through simulation in MATLAB. In this simulation, we take a 2-dimensional array of i.i.d. normal random variables as the χ_{ij} from Equation 1. The sum of weighted values is equivalent to a filter with those weights, so we create a filter and apply this to the array. Next, we go through the new array, and look at the correlation between sets of points a distance d units away from each other. A condition that must be satisfied for this correlation function to be valid is translational and rotational invariance, else the correlation function would depend on the coordinates as well. The i.i.d. random variable array implies translational invariance, since shifting in the grid has no effect on the expected values of the variables. We attempt make the filter spherically symmetric, so that this correlation function can be approximately rotationally invariant, and thus only a function of distance.

From Figure 1, we can see that the curve fits our intuition. As the distance between the two nodes increases, the correlation between them decreases towards 0. The small negative correlations at very large distances may simply be due to the randomness in the simulations, and are too small to be of much effect. For distances larger than 100, we will set the correlation to be 0.

B. Node Power Consumption

Using the constraints for each of the data compression schemes, we are able to simulate the required power per node for data transmission for various node densities, for d in units of meters, bandwidth $B = 20kHz$, noise density $N_0 = 10^{-15}W/Hz$, base-station/node distance ratio $\frac{d_{base}}{d_{node}} = 30$, and individual node rate $r = 6b/s$. We assume that we have k nodes in a 500m x 500m unit area, giving the total rate $R = kr$.

Based on the graph, we can notice some interesting results. We see that the worst performer is *Algorithm 1*. A major problem with data compression within the nodes is the need to transmit data between nodes. In these schemes, we assume that the nodes always transmit a distance equal to that of the longest distance in the box, so this effect is magnified. Another explanation for the significantly worse performance is that no matter the node density, the center node will never be correlated with those on the boundaries, and so this compression scheme is very limited.

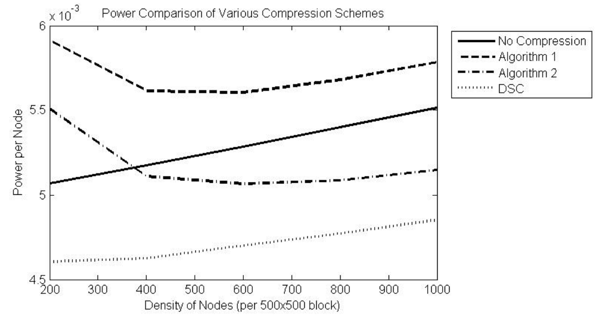


Figure 2: Simulation for required power for $B = 20kHz$, $N_0 = 10^{-15}$, $\frac{d_{base}}{d_{node}} = 30$, and $r = 6$

The density of the network plays a role in determining whether the uncompressed method or *Algorithm 2* has lower power per node. This makes sense since for sparse sensor networks, the correlation between nodes is smaller, so the cost of transmitting data between nodes for compression outweighs the savings of sending less data back to the base station.

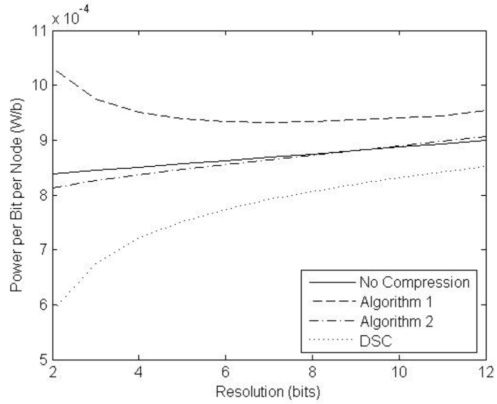
DSC does not have the cost of inter-node data transmission, and since there is some non-zero correlation in the network, it always performs better than the no compression case. Also interesting is the comparison between *Algorithm 2* and DSC. Since these two methods are very similar in terms of their grouping and transmission, the main difference between the two is the requirement of inter-node data transmission in *Algorithm 2*. The difference in power consumptions is therefore a rough estimate of the overhead accrued by data transmission within the sensor network. Of course, it is not the perfect measure of overhead because of bandwidth sharing. This overhead is largest for low node densities, due to the fact that since the data compression savings are so little that the bandwidth is essentially feeding almost completely uncompressed data to the base station, while also allocating resources to transmit data between nodes. This extra transmission greatly increases the power resource usage.

Figure 2 also provides a way of knowing our power budget for the circuitry concerning data compression and correlation knowledge. If given these values beforehand, we can simply sum the powers and compare the results in order to determine which is the better strategy for the particular sensor network.

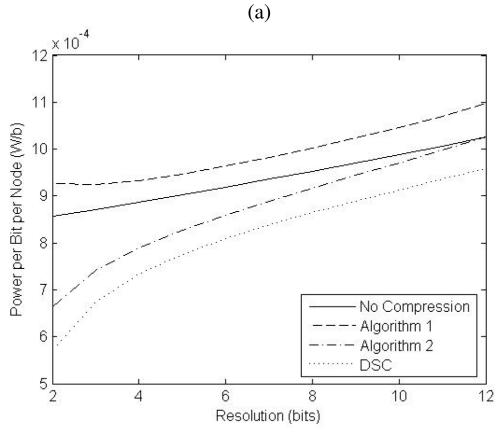
C. Power Consumption and Resolution

This paper will also look at the effect of sensor resolution on power consumption within this correlation model. Higher resolution measurements do not have as high a compression ratio as similarly correlated data simply because the extra precision bits have less correlation than the most significant bits, and so have lower compression ratios.

Figure 3 provides a look at the effect of measurement resolution on node transmission power for a couple different



(a)



(b)

Figure 3: The effect of resolution on power consumption for (a) $k = 400$ and (b) $k = 1000$

node densities. These figures show the required increase in power per bit to transmit the extra bits in the same time and bandwidth. With a lower node density, not only does data compression not work as well, but it surpasses the uncompressed method at a lower bit resolution than in the higher node density scenario.

Studying the resolution also provides a rough idea on the power savings with lossy coding, in which the least significant bits are dropped. Since the correlation usually means redundancy in the most significant bits, sensors can in effect increase their compression rate and reduce energy consumption. This, coupled with the fact that the sensors need to send less bits in the first place, allows the sensors to greatly reduce energy consumption when correlation is high.

VI. Extensions and Conclusion

One way to view the two proposed algorithms is to think of them as the boundary cases of a more general algorithm. This more general algorithm picks the number of groups, and the size allowed for each group. This more general algorithm has another degree of freedom in which to optimize the sensor network transmission, especially if the joint entropy of multiple nodes have high correlations between each other, or a correlation matrix where most of the values are close to 1. While not studied in this paper, using iterative steps to vary these groupings may also help data transmission in the long run, especially if correlations are time-variant.

As mentioned previously, this correlation model is not the only model to which we can apply the simulation to. Any other correlation function derived from Gaussian random variables, such as the use of a different filter, can be used in these simulations as well since the relation between mutual information and correlation coefficients is derived for Gaussian random variables. For correlation models that take into account correlation in binary sequences, [12] provides a way of relating the mutual information between nodes to their correlation.

Based on these simulations, in order for wireless sensor networks to successfully apply data compression, the correlation between nodes must be determined to be very high. Our results show that for sparser networks, data compression is actually harmful for sensor energy usage. While we do not account for the computational power required for data compression, these simulations provide a metric in which we can make decisions, if given the power requirements for data compression, on whether or not data compression is a better solution for the sensor network.

VII. Acknowledgments

The author would like to thank Andrea Goldsmith and Yao Xie for their guidance during discussions and feedback on previous related assignments.

References

[1] Z. A. Eu, H. P. Tan, and W. Seah, "Routing and relay node placement in wireless sensor networks powered by ambient energy harvesting," in *Wireless Communications and Networking Conference*, April 2009, pp. 1–6.

[2] S. Mahlknecht and M. Roetzer, "Energy supply considerations for self-sustaining wireless sensor networks," in *Proceedings of the Second European Workshop on Wireless Sensor Networks*, July 2005, pp. 397–399.

[3] S. Mahlknecht and M. Bock, "CSMA-MPS: A minimum preamble samping MAC protocol for low power wireless sensor networks," in *IEEE International Workshop on Factory Communication Systems*, September 2004, pp. 73–80.

[4] M. Kuorilehto, J. Suhonen, M. Kohvakka, M. Hannikainen, and T. D. Hamalainen, "Experimenting TCP/IP for low-power wireless sensor networks," in *IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications*, September 2006, pp. 1–6.

[5] A. Jindal and K. Psounis, "Modeling spatially-correlated sensor network data," in *Sensor and Ad Hoc Communications and Networks, First Annual IEEE Comm. Society Conf. on*, October 2004, pp. 162–171.

[6] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[7] R. M. Gray, *Entropy and Information Theory*. Springer Verlag, 1990.

[8] A. M. Fraser, "Reconstructing attractors from scalar time series: A comparison of singular system and redundancy criteria," *Physica D: Nonlinear Phenomena*, vol. 34, no. 3, pp. 391–404, March 1989.

[9] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, pp. 471–480, 1973.

[10] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Compression of binary source with side information at the decoder using LDPC codes," *IEEE Communications Letters*, vol. 6, no. 10, pp. 440–442, October 2002.

[11] I. Deslauriers and J. Bajcsy, "Serial turbo coding for data compression and the slepian-wolf problem," in *IEEE Information Theory Workshop*, August 2003, pp. 296–299.

[12] W. Li, "Mutual information functions versus correlation functions," *Journal of Statistical Physics*, vol. 60, no. 5/6, pp. 823–837, 1990.