Energy-Efficient Data Representation and Routing for Wireless Sensor Networks Based on a Distributed Wavelet Compression Algorithm [*]

Alexandre Ciancio, Sundeep Pattem, Antonio Ortega and Bhaskar Krishnamachari
Department of Electrical Engineering–Systems, University of Southern California
Los Angeles, California, USA

ciancio@usc.edu, pattem@usc.edu, ortega@sipi.usc.edu, bkrishna@usc.edu

ABSTRACT

We address the problem of energy consumption reduction for wireless sensor networks, where each of the sensors has limited power and acquires data that should be transmitted to a central node. The final goal is to have a reconstructed version of the data measurements at the central node, with the sensors spending as little energy as possible, for a given data reconstruction accuracy. In our scenario, sensors in the network have a choice of different coding schemes to achieve varying levels of compression. The compression algorithms considered are based on the lifting factorization of the wavelet transform, and exploit the natural data flow in the network to aggregate data by computing partial wavelet coefficients that are refined as data flows towards the central node. The proposed algorithm operates by first selecting a routing strategy through the network. Then, for each route, an optimal combination of data representation algorithms i.e. assignment at each node, is selected. A simple heuristic is used to determine the data representation technique to use once path merges are taken into consideration. We demonstrate that by optimizing the coding algorithm selection the overall energy consumption can be significantly reduced when compared to the case when data is just quantized and forwarded to the central node. Moreover, the proposed algorithm provides a tool to compare different routing techniques and identify those that are most efficient overall, for given node locations. We evaluate the algorithm using both a second-order autoregressive (AR) model and empirical data from a real wireless sensor network deployment.

Categories and Subject Descriptors

C.2.1 [**Computer Communication Networks**]: Network

Architecture and Design; C.2.4 [**Computer Communucation Networks**]: Distributed Systems

General Terms

Algorithms, Performance

Keywords

Wireless Sensor Networks, Distributed Wavelets, Energy Optimization

1. INTRODUCTION

Wireless Sensor Networks (WSN) can offer mobility and versatility for a variety of applications, such as object detection/tracking, environment monitoring and traffic control [1]. The main challenge is that they often rely on batteries for power supply and limiting energy consumption becomes essential to ensure network survivability.

When data is acquired at multiple correlated sources, aggregation involving in-network data compression can offer a more efficient representation of measurements. By significantly reducing the amount of information that needs to be transmitted over the network, a potentially large reduction in energy consumption can be obtained. Prior work has addressed a number of distributed source coding (DSC) methods as a means to decorrelate data. While some rely on information exchange and additional computation inside the network to propose distributed versions of transforms, such as Karhunen-Loève [7] and Wavelets [17], others propose schemes that do not require internode communication, such as networked Slepian-Wolf coding [12, 6]. In general, DSC techniques face a trade-off between the cost for i) processing at each node to achieve compression and ii) transporting information (bits) to to the sink. This trade-off has also been addressed by previous research. For example, [11] provides an analysis on the regions in a network that should favor compression over routing based on the impact of spatial correlation of the measurements. The performance of aggregation under a more general data model is considered in [8].

However, while previous works have typically provided a number of methods to decorrelate data in a network and/or individually analyzed their performance, to the best of our knowledge, none have addressed the problem of finding an optimal assignment of compression algorithms to nodes, in

---
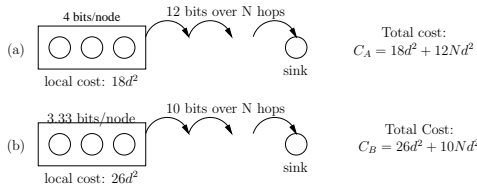
Figure 1: (a) Method "A": a simple encoding scheme is used; 12 bits are sent to the sink. (b) Method "B": a locally more expensive method is used, but better compression is achieved; 10 bits are sent to the sink. The cost of transmitting k bits over a distance d was computed as kd^2.

the sense of minimizing the energy consumption, when different methods are available. Since the distortion/energy consumption trade-off also depends on factors such as network topology and medium characteristics, different coding methods may be better suited for different parts of the network. These methods can consist of simple coding schemes such as DPCM, or more complex ones, such as Wavelet transforms with an arbitrary number of levels of decomposition.

To illustrate how the network topology can influence the performance of a given coding scheme, consider a simple example, illustrated in Fig. 1. A group of 3 sensors in an array of equally spaced nodes (at a distance d of each other) is separated from the sink by N hops, and might choose between two coding schemes to decorrelate data. Method "A" is limited in terms of performance, but requires a smaller local cost (that takes into account additional transmissions and local computation) from the group. Method "B" is locally more expensive energy-wise, but achieves better decorrelation, i.e., requires fewer bits. It is easy to show that, in this example, if the three sensors are more than 4 hops away from the sink, method "B" will lead to a lower energy consumption. In general, it is expected that while sensors that are far from the sink will benefit from using schemes that require a smaller average number of bits per sensor for a given distortion, sensors closer to the sink should use coding schemes that offer smaller local cost.

A similar conclusion is reached in [6], where a model is chosen for the data and the coding performance used in the analysis is based on entropy. A significant contribution of our work is that i) it makes no modeling assumptions about the underlying data statistics (instead all is required is training data) and ii) the algorithm directly compares the performance of actual coding schemes (instead of considering entropy) and provides more realistic network cost estimates.

In [2, 3] we introduced energy-aware distributed wavelet compression algorithms for WSN [2] that aimed at exploiting the *spatial* redundancy in sensor data. These algorithms could be easily extended to exploit any existing temporal redundancy as well. This could be accomplished by using a separable implementation of the wavelet transform (such as those typically used when applying the wavelet transform to images), which incorporates temporal filtering in addition to the spatial filtering we describe in this paper. In our previous work we also introduced a partial coefficient approach based on the lifting implementation [3]. Our goal was to generate the wavelet transform coefficients at sensors, at the expense of energy spent for a few "local" transmis-

sions, i.e., data transmissions between neighboring nodes that are needed to actually compute the wavelet transform coefficients, since the transform operates by filtering "across nodes". If the original data has sufficient spatial correlation, after quantization the wavelet coefficients can represent the original measurements using fewer bits, and the overall energy consumption in the network is lowered by reducing the amount of information that has to be transported. The proposed partial coefficient approach [3] essentially allows all wavelet transform operations to be causal, in the sense of that data is processed as it is being forwarded to the central node, so that only data from nodes already traversed is used to compute the wavelet coefficients. This requires the computation and quantization of "partial" coefficients, which are transmitted over a few hops, before being used to generate the final wavelet coefficients. In future discussions, we refer to *partial* coefficients as the coefficients that were computed at a node with insufficient data, and still need to be refined at other nodes until they become final, or *full* coefficients.

In this work, we consider networks where each sensor can use one of several different compression schemes, which include wavelet transforms with different number of levels of decomposition, or simpler approaches, like raw (quantized) data transmission or DPCM. We provide a framework that allows finding, for a given network topology, which among the available coding methods is more suitable for each of the sensors, such that the whole network operates with a minimum cost to achieve a desired distortion level. Our method is flexible enough to accommodate any network configuration (topology). We believe that the basic principles of our approach could be applied to other data representation selection among other sets of coding schemes (i.e., not limited to wavelets) that operate by exploiting spatial redundancy (in methods that do not involve data exchanges between nodes in the compression process, the optimization becomes straightforward). However, the exact formulation of the optimization problem, and the suitability of dynamic programming techniques to solve it, will depend on the specific algorithms involved.

1.1 Related Work

One of the first works to propose combining routing with standard vector compression techniques was [16]. The correlated data gathering problem and the need for jointly optimizing the coding rate at nodes and routing structure is also considered in [13]. The authors provide analysis of two strategies: the Slepian-Wolf or DSC model, for which the optimal coding is complex(needs global knowledge of correlations)and optimal routing is simple(always along a shortest path tree) and a joint entropy coding model with explicit communication for which coding is simple and optimizing routing structure is difficult. For the Slepian-Wolf model, a closed form solution is derived while for the explicit communication case it is shown that the optimization problem is NP-complete and approximation algorithms are presented. In [11], the approach is to simplify the optimization for the explicit communication case by using an empirically obtained approximation for the joint entropy of sources. The optimal routing structure is then analyzed under this approximation. The analysis demonstrates that the optimal routing structure also depends on where the actual data compression is performed; at each individual
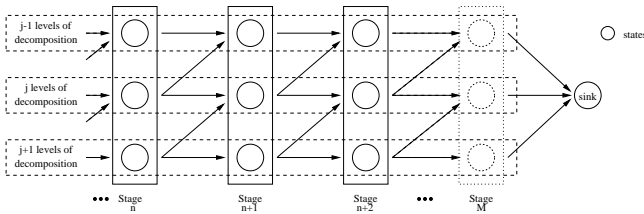
Figure 2: Data transitions for partial coefficient approach.

node or at "micro-servers" acting as intermediate data collection points. In [15], "self-coding" and "foreign-coding" are differentiated. In self-coding, a node uses data from other nodes to compress its own data, while in foreign-coding a node can also compress data from other nodes. With foreign-coding, the authors show that energy-optimal data gathering involves building a directed minimum spanning tree (DMST). For self-coding, it is shown in [13] that the optimal solution is NP-complete. Also, it is expected to be a tradeoff between a SPT and a TSP (traveling salesman path). Both these works assume that the data is compressed only once, after which it is decompressed at the sink. Techniques such as those described in this paper (and earlier in [3]) and [9] allow compression at several hops, potentially leading to greater reductions in transported data. In [11], the authors explore compression at several hops and only at cluster heads and conclude that there exist efficient correlation independent routing structures. Some recent research [19] argues that the improvement from correlation aware routing is limited. Using a correlation model proposed in [14], it is shown that in terms of energy efficiency, a shortest path tree has at least 0.5 times performance compared to an optimal correlation aware routing structure. However, this result is contingent on a limited data compression model - compression gain independent of number of neighbors and distances between nodes. In [11], [15] and [14], the details of how compression is achieved and the accompanying cost for the computation required is not taken into account. Depending on the signal field and the degree of spatial correlations, this cost can significantly impact the routing structure. To illustrate this impact, we present some initial results on a variety of network topologies and routing structures.

In closely related work [9], the authors propose using different data gathering algorithms for different classes of signal fields. They state that wavelet based processing is well-suited for deterministic signals such as piecewise constant signals and prediction based DPCM processing is optimal for random Gaussian correlated fields. The algorithms presented account for the combined costs of both communication (for data transport) and computation (for decorrelating i.e. compressing raw data). Distributed and power-optimal operation of these algorithms results in division of the network into segments (or clusters) within which incurring the cost for compression is efficient. In contrast, we consider a scenario where a number of different compression schemes are available at each node. In this case, the problem is one of making a decision at each node on which compression scheme to use based on the expected computation/communication cost tradeoff. Currently, we have addressed the assignment in a two dimensional field assuming

the routing structure is known, using a heuristic extension of the dynamic programming based optimal solution for a one dimensional network presented earlier in [4]. An important problem is to optimize jointly for the scheme assignment and routing structure.

This paper is organized as follows. In Section 2 we briefly describe the partial coefficient data representation and introduce the optimization framework, and describe states, transitions and costs for the state machine considered. The proposed algorithm is described in Section 3, and section 4 describes the routing strategies used in the simulation. Section 5 provides simulation results. We conclude the paper with a summary in Section 6.

2. DATA REPRESENTATION AND OPTIMIZATION

2.1 Partial Coefficient Approach Overview

In this paper, we assume that a sensor network acquires measurements from a correlated data field. We consider data aggregation (compression) along a 1-D path from an edge to the sink, and define a heuristic approach to represent data at merging points. This path is assumed known, which implies that a routing algorithm has been applied to the network first. Each sensor is assigned a number n, starting from the edge. The network topology (and therefore internode distance) is known, and each node in the 1-D path can operate using a coding scheme chosen from a predefined set of available coding schemes. In this paper, available schemes are discrete wavelet transforms that use the same filter bank but employ a different number of levels of decomposition: when the number of levels decomposition is increased, the potential compression efficiency also increases (if data is highly correlated across sensors), but at the cost of more local information exchange (because data from more nodes is needed to compute some of the wavelet coefficients).

Since the wavelet transform is critically sampled, the number of wavelet coefficients generated is equal to the number of nodes. Using the partial coefficient approach [3], the wavelet coefficient corresponding to node n is computed in steps: at node n a partial version of the coefficient is first generated, which becomes a full coefficient as it "incorporates" additional data from future nodes (i.e., nodes closer to the sink). The number of hops required until a partial coefficient becomes full depends on the specific transform filters being used - refer to [3] for details.

2.2 1D Path Optimization

In order to find the best coding scheme for each of the sensors, we propose representing the network as a graph. Fig. 2 illustrates the graph associated to a 1D path including $M+1$ sensors (the last one being the sink), where each sensor can use one of three available coding schemes. Each edge in the graph reflects a possible transition from one coding scheme to another, and has an associated weight that represents the transmission cost to continue in a determined scheme or the extra processing/transmission cost to change coding schemes. Each possible path in the graph is associated with one choice of coding scheme for each node in the routing path from a leaf to the sink.

We assume that the coding schemes that can be chosen from in Fig. 2 are wavelet transforms with progressively
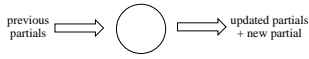
Figure 3: State I/O.

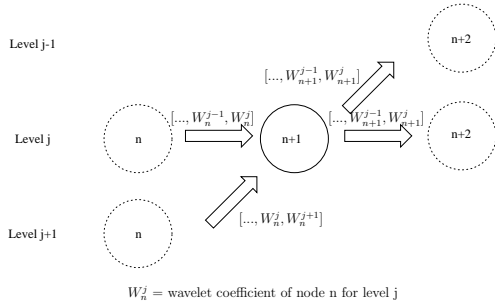W_n^j = wavelet coefficient of node n for level j

Figure 4: Data transitions for partial coefficient approach.

larger number of levels of decomposition, and that as a node gets closer to the sink it can only choose between staying on the same scheme as the previous node or encoding its data using a simpler scheme, specifically one that uses one fewer level of decomposition than the current node. As motivated by the example in Section 1, the intuition behind this limitation is that as nodes get closer to the sink, simpler coding schemes tend to be more efficient energy-wise. This idea can also be linked to results obtained in [6].

Let $c_n(i, j)$ be the cost of the transition from method i at node n to method j at node $n+1$, $n = 1, \ldots, M$, where $i, j \in S = \{A, B, C, \ldots\}$, with $A, B, C, \ldots$ representing the coding schemes considered. Let $l_n(i)$ denote the local processing cost for sensor n to encode its data using coding scheme i. Our goal is to find the sequence $\{i_1, i_2, \ldots, i_n, \ldots, i_M\}$ of coding schemes associated with each of the nodes such that $\sum_n (c_n(i_n, i_{n+1}) + l_n(i_n))$ is minimum.

2.2.1 State/Transition Descriptions

A state can be described by two parameters: the node's position in the 1-D path to the sink (n), and the coding scheme being used (j). Each state in the graph corresponds to a series of computations that depend on the node and coding scheme represented by the state. These computations are: receiving partial coefficients from previous nodes, updating them, and generating the node's partial coefficient based on the available information (see Fig. 3). Each state adds to the current path cost the costs associated with i) the corresponding partial coefficient generation and update and ii) forwarding any newly computed full coefficients to the sink.

We can take advantage of the embedded nature of the wavelet transform structure (octave tree) to calculate the overhead involved in changing the number of levels of decomposition in the transform. In particular, we know that the transform coefficients generated by a wavelet transform with $j-1$ levels of decomposition can be obtained from those generated with j levels. Therefore, if W_n^j corresponds to the set of wavelet coefficients for all nodes up to n for the j-th level of decomposition, we can say that $\cdots \subset W_n^{j-1} \subset W_n^j \subset W_n^{j+1} \subset \cdots$. The information that is conveyed in each of the transition scenarios corresponding to different branches of the state machine can be seen in Fig. 4.

The transition costs reflect the transmission cost of sending partial data either i) over one hop in case the data can be refined at future nodes (i.e. next node operates in the same scheme as before), or ii) over all hops until the sink, in case it cannot be refined (future nodes use a simpler wavelet transform). A more detailed description of state and transitions can be found in [4].

2.2.2 Optimization

The partial coefficient approach guarantees that any computation at any given node requires only data from previous nodes. Also, due to the wavelet property discussed in Section 2.2.1, any node always has access to the past coefficients it needs to compute its own partial coefficient, regardless of whether the data it is receiving was generated from the same wavelet transform as itself or from one with a larger number of decomposition levels. As a result, transition costs depend only on the present state: the physical position of the node in the network and the coding scheme being used, and, since the output of a node (its coefficient) is the same regardless of the previous path, costs up to a particular node do not influence the cost for a future transition. Therefore, choosing a best path arriving to a specific state does not eliminate optimal paths. Hence a path that minimizes the path metric (lowest cost) can be found using a shortest-path algorithm. At each stage n of the decision, the algorithm computes the best transition coming into each state at that stage by computing the metrics of all the possible paths coming into the state, and then selecting the path with the minimum metric as the survivor path coming into that state. At the last stage, the survivor path with the minimum path metric is selected as the optimum path. Since each state describes the coding scheme used at each of the nodes, the optimum path also provides the optimal selection of coding schemes for each sensor in the network, such that the energy consumption is minimized. A more detailed description of dynamic programming and the shortest-path algorithm can be found in [5], and we refer to [4] for further details on our proposed 1D optimization.

2.3 2D Path Merging

In the 2D networks we are considering, individual 1D paths may merge before reaching the sink. We now extend our 1D wavelet representation algorithms to the 2D case. Referring to Fig. 5, assume that the same wavelet transform is used along both paths. The node information captured along Paths 1 and 2 may be correlated but is not expected to be identical. Thus, the wavelet coefficients received by node n, from the two paths will be different in general. Assume first that we continue processing both paths independently, i.e., the coefficients received from Path 1 and Path 2 are updated to incorporate nodes n, $n + 1$, and so on. Since these new coefficients are generated with common information, and our wavelet filters have finite memory, after several iterations, say at node $n + k$ (where k will depend on the filter length), the newly produced wavelet coefficients will be identical regardless of whether the data was originated from Path 1 or Path 2. Taking this into account, from that point forward the two sets of data can be merged, so only one set of wavelet coefficients needs to be sent from node $n + k$ onwards.

In addition, Paths 1 and 2 up to node n are generally correlated and even if the correlation was not high, the pairs
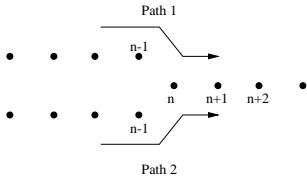
Figure 5: Multiple coefficients are generated at merge points.

of coefficients (one per path) generated in nodes n through $n + k - 1$ will become increasingly similar. This suggests a simple approach to further reduce the cost incur in merging paths. At any node where multiple coefficients are generated (due to two or more path mergings), we encode this array of coefficients using DPCM, i.e., we compute and quantize the difference between consecutive values in the array. If the values are similar, this difference is small, and can be more efficiently represented by DPCM. This additional data is forwarded to the sink without being modified by future nodes.

3. PROPOSED ALGORITHM

The proposed algorithm assumes a routing topology has been selected (in future work we will consider a joint selection of route and coding algorithm). Given the topology, we apply the 1D optimization of Section 2.2 (based on the partial coefficient approach [3]) *independently* to each of the paths connecting an edge to the sink, where we consider complete paths from edge to sink, i.e., we optimize the data representation for some nodes multiple times (e.g., selection for nodes n, $n+1$, etc in Fig. 5 will be considered twice corresponding to Paths 1 and 2). Thus, some specific nodes, e.g., node n in Fig. 5, could be assigned different data representations when optimized under different path configurations. When this happens we assign to the node the simplest representation (i.e., least number of levels in the wavelet decomposition). With this heuristic we force the merged paths to use the least computationally intensive among candidate algorithms.

Obviously, in cases where nodes are assigned the same algorithm by all 1D optimization, we use the chosen algorithm and compress the merged path information as described above. Our algorithm can be summarized as follows:

Algorithm 1
    1) Optimization:
    for *each 1D path to the sink:*
        -assign optimum scheme to node;
        if *assignment conflict with previous paths:*
            -use simpler scheme;
    end

    2) Encoding:
    for *each path:*
        -encode data using optimum node assingment;
        -if multiple coefficients, use DPCM
    end

4. ROUTING

When the proposed approach for choosing encoding schemes is used for a two dimensional network, an extra processing cost is incurred whenever two different paths merge enroute to the sink. In particular cases, as will be seen in Section 5, the extra processing and added information needed to represent merged paths could lead to inefficiency in the data gathering; so that a topology with fewer merges might be preferable, even if it has overall longer transmission paths.

In order to study the impact of these merges, we consider a variety of topologies, namely, degree-constrained trees. These trees are generated using a modified version of Dijkstra's algorithm over weighted graphs. Consider a graph $G(V, E)$ with edge weights $EW(i, j)$ for nodes $i, j \in V$ and edge $(i, j) \in E$ and a given degree constraint $maxDegree$. Each node n maintains a value for its best weighted path distance to the sink $WPD(n)$ (initial value $= \infty$) and the number of nodes that are its children $C(n)$(initial value $= 0$) in the tree rooted at the sink. Starting with a set T that initially contains the sink S, at each step add to T the node $p \notin T$ for which

- there is a node $p' \in T$ such that edge $(p, p') \in E$,
- if $p' \neq S$ then $C(p') < maxDegree - 1$, and
- the weighted path cost to the sink $[= EW(p, p') + WPD(p')]$ is minimized.

The updates are made as $WPD(p) = WPD(p') + EW(p, p')$ and $C(p') = C(p') + 1$. The algorithm stops when $|S| = |V|$ or when no more nodes can be added (since all their neighbors have already hit the degree constraint). We avoid the latter case by considering well-connected graphs. When $maxDegree$ is greater than the maximum node degree in G, the algorithm reduces to finding the shortest weighted path and when $maxDegree = 2$, it results in long paths with no merges except at the sink. In the experiments that follow, we generate trees using the maximum number of merges M ($= maxDegree - 1$) as a tunable parameter.

5. EXPERIMENTS

In our experiments, we use both a second-order AR model as well as empirical data from a real wireless sensor network deployment. The simulated data consisted in generating three 600×600 2D processes using a second order AR model with low, medium and high data correlation. The nodes were placed in the 600×600 grid, and their measurements corresponded to the data value from the associated position in the grid. The data is from a subset of 19 sensor nodes from a habitat monitoring deployment [10, 18] on the Great Duck Island. The dataset used is for 200 temperature readings taken at each sensor location on August 1, 2003 at roughly 5 minute intervals.

The sensor locations for simulated data included random sensor placement and square grid, both with 100 nodes. The routing algorithm uses degree-constrained trees generated as described in Section. 4. The branch costs used for data representation optimization were proportional to the square of the distance between nodes, with the constant of proportionality being the number of bits allocated to the node transmitting data towards the sink.

In Section 5.1 we analyze the performance of the proposed method for a given routing topology. This allows us

to demonstrate that gains are achievable by selecting different data representations for different nodes. Then, in Section 5.2, we select the data representation to be that obtained with our optimization algorithm and we compare the performance of different routing topologies, using degree-constrained trees with varying degree. For a sufficiently large degree value, the tunable tree reduces to the shortest-path tree. This allows us to demonstrate that for certain cases, shortest path routing is in fact not optimal, as it leads to an undesirably large number of path merges.

In all network graphs that follow, nodes marked with "×" are encoded with a 1-level wavelet algorithm and nodes marked with "○" are encoded with a 2-level wavelet. Raw data transmission (i.e., no compression) is represented by the symbol "◇", but no node was assigned to be encoded as raw for any of the simulations performed.

5.1 Algorithm Performance for a Given Network Topology

Figs. 6, 7 and 8 illustrate the performance of the different algorithms; in terms of data representation distortion (measured by Signal-to-Noise ratio, SNR) as a function of total energy consumption in the network. In each figure we depict the routing together with the data representation algorithm used for each sensor node. The energy consumption is averaged over multiple realizations of the data. The data representation itself is generated taking as an input training data generated with the AR model, i.e., we do not use the model parameters directly and our algorithm can be applied, as shown below, to optimize representation for any training data set. Obverse that in all cases the optimized data representation, as expected, outperforms configurations where a single representation is used. Note, however, that in Figs. 6, 7 and 8 the gains provided by the optimal representation are modest. This can also be seen by noting that only a few nodes make use of a 2 level wavelet representation.

In Figs. 6 and 7 nodes were uniformly distributed. Instead, in Fig. 8 nodes were clustered to split the data field in two regions separated by a gap, and the network sink was placed in the region where the node density was lower. Observe that the behavior in Fig. 8 is intuitively reasonable: in the cluster of nearby nodes on the top right region of the sensor field a 2 level wavelet is chosen so as to minimize the overall rate required. This is because this small cluster is far from the sink and thus any excess rate in the representation (due to using a less efficient encoding) would result in significant energy required for transmission.

Fig. 9 illustrates the performance of our methods with real data. It can be seen that when no merge is allowed (Fig. 9a), the optimum algorithm performs considerably better than the other schemes. This confirms our initial assumption that merging paths tend to be costly. However, as will be seen, shortest path routing still provides overall better performance.

Fig. 10 shows the optimum node assignment in the network with random node placements, for different values of data correlation and maximum number of merges per node. It can be seen that, in general, more complex schemes (better data decorrelation) tend to be used far from the sink, agreeing with previous work [13, 11].

5.2 Routing Comparison With Optimum Network Representation

Now that we have established that data representation optimization leads to lower overall cost for a given network topology, we compare different network topologies in terms of their overall cost with optimized representation (see Fig. 11). In particular we compare the performance of no-merge and shortest-path routing. For both simulated random network data with uniform sampling (Fig. 11a) and real data (Fig. 11b), we can see that shortest path routing leads to better overall performance. For the denser network (simulated data) overall performance is very sensitive to the average length of the 1D paths in the network, so that the gains from using shortest-path routing are substantial. In contrast, for the smaller real-data network (Fig. 11b), performance is not as significantly affected by the routing. We believe that in the case of Fig. 11a, given the density of the network the cost of merging is not as significant (as a result of the proposed DPCM coding of merged paths) and therefore the penalty due to allowing longer paths (as in the no-merge configuration) is too high to be compensated by the reduction of merge costs. It is also interesting to notice that the no-merge and shortest-path routing have similar performance at low SNR (this can be seen in the real-data case, which we operate at lower rates). This can be explained as follows: in shortest-path routing information needs to be sent for every merge, for a number of nodes dependent on the filter length. At low rates this overhead becomes more significant as the data transmission rates in no-merge routing become lower. Fig. 11c represents results for the clustered network scenario. Note that in this case, unlike the two preceding ones, no-merge routing leads to the best overall performance.

Note that in Fig. 11a, b, c the performance curve slope is lower for no-merge routing (i.e., the quality degrades more slowly as the transmission cost decreases). We again attribute this to additional cost of data merging in the shortest path routing case.

6. CONCLUSION

In this work, we have provided a framework that allows finding, for a given network topology, which among a number of available coding methods is more suitable for each of the sensors. We have shown that by optimizing the coding algorithm selection, the overall energy consumption can be significantly reduced when compared to the case when data is just quantized and forwarded to the central node. In our simulations we were able to compare different routing techniques and identify those that are most efficient overall, for given node locations, and for both simulated data and from a real wireless sensor network deployment. We verified that the proposed optimum algorithm always finds the best coding scheme that should be used by each node in the network, and that shortest-path routing not always leads to best performance.

A number of topics can still be addressed. While in this work the algorithm analysis was limited to predefined routing topologies, joint selection of route and coding algorithm seems to be a natural next step for optimization. Also, more realistic cost functions can be incorporated to the decision process (e.g., transmission costs a constant term in addition to a term proportional to distance), and a more efficient
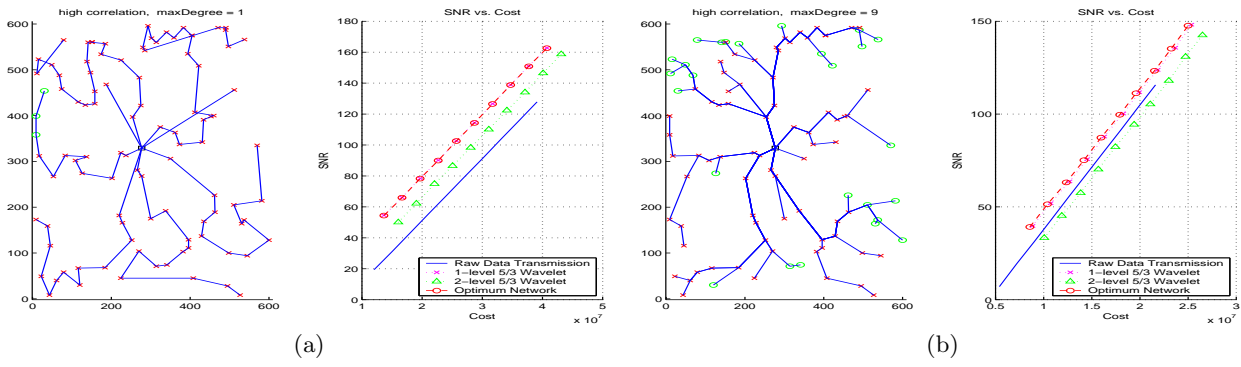
Figure 6: Algorithm performance for random network with a) no-merge routing b) shortest-path routing
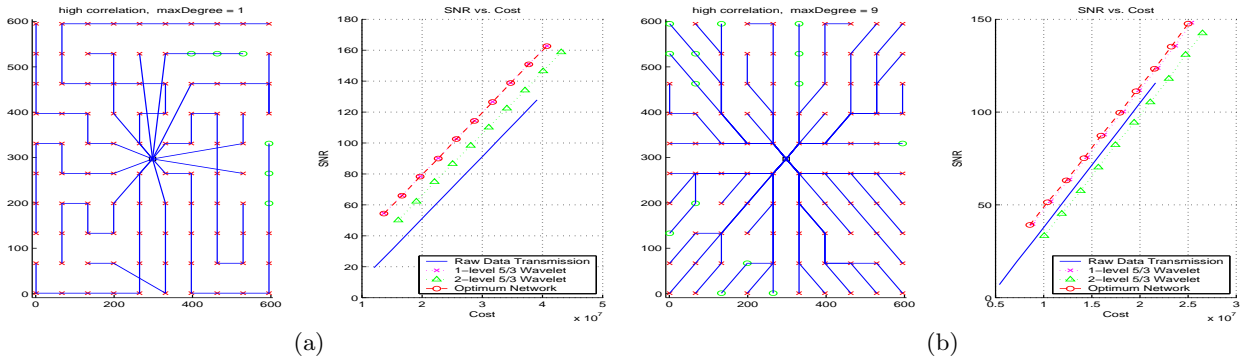

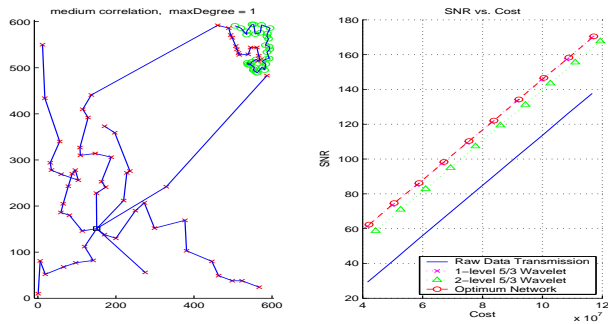**Figure 7: Algorithm performance for square grid network with a) no-merge routing b) shortest-path routing**

Figure 8: Algorithm performance for clustered network with shortest-path routing.

Figure 10: Node assignment for different values of data correlation and maximum number of merges per node (M).

representation of natural phenomena (e.g. different regions of the field are modeled with different parameters) can be used. While in this work we assumed nodes exchanged information by data hoping, in a practical application broadcast capabilities could be used to improve even more the network efficiency. For example, coefficients in the paths that were already fully computed could reach the central node by broadcasting, eventually skipping hops and saving more energy; nodes that need a measurement from a common sensor could acquire it via broadcasting in one single step, also reducing the energy consumption.
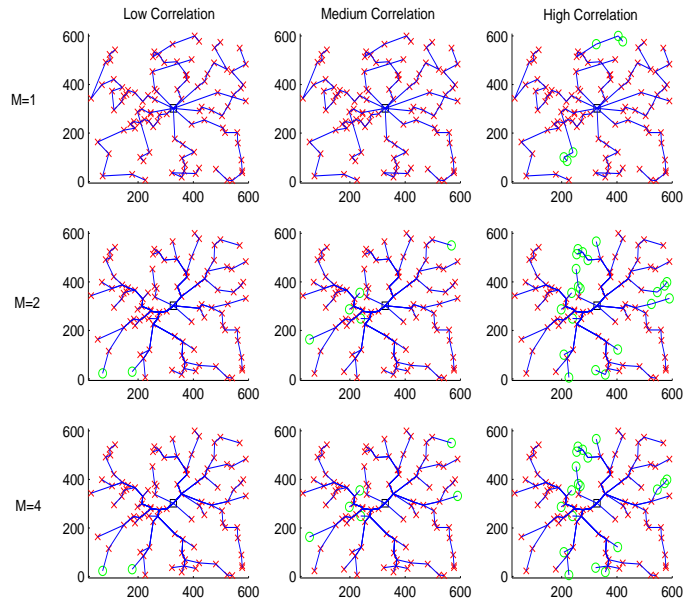
7. REFERENCES

[1] C. Chong and S. P. Kumar. Sensor networks: Evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, August 2003.
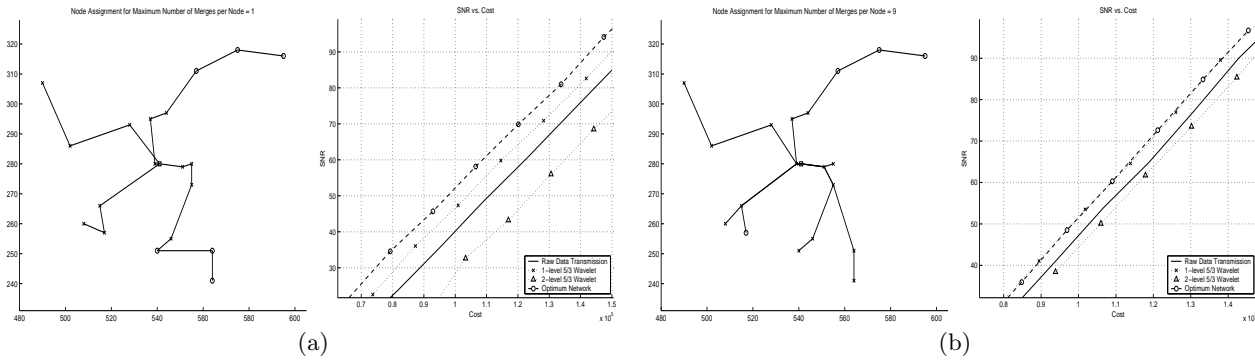
Figure 9: Algorithm performance for real network data with a) no-merge routing b) shortest-path routing
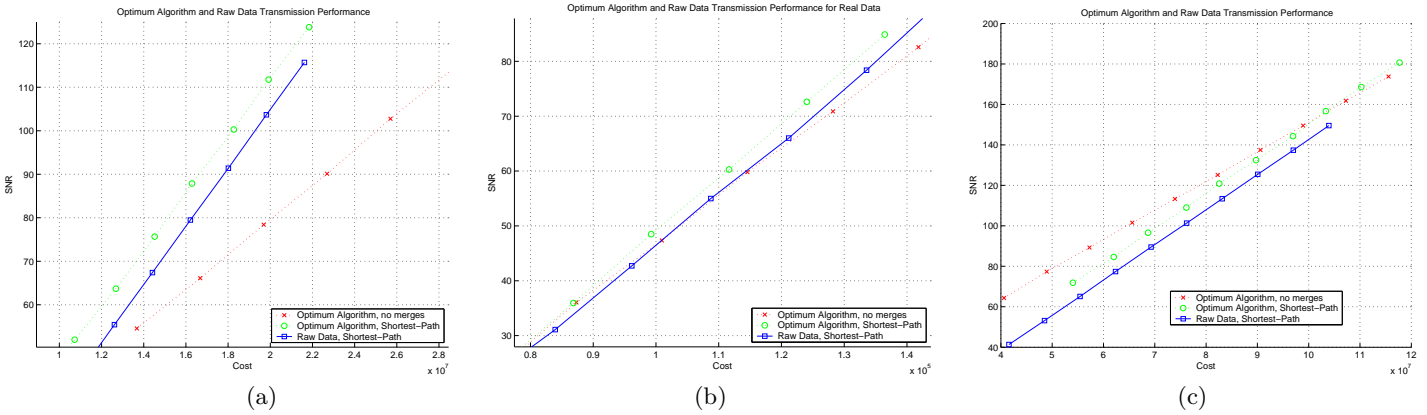

Figure 11: Optimum algorithm performance compared to raw data transmission for no-merge and shortest-path routing for a) random network b) real data network c) clustered network

[2] A. Ciancio and A. Ortega. A distributed wavelet compression algorithm for wireless sensor networks using lifting. In *International Conference on Acoustics, Speech and Signal Processing - ICASSP04*, Montreal, Canada, May 2004.

[3] A. Ciancio and A. Ortega. A distributed wavelet compression algorithm for wireless multihop sensor networks using lifting. In *International Conference on Acoustics, Speech and Signal Processing - ICASSP05*, Philadelphia, USA, March 2005.

[4] A. Ciancio and A. Ortega. A dynamic programming approach to distortion-energy optimization for distributed wavelet compression with applications to data gathering in wireless sensor networks. *International Conference on Acoustics, Speech and Signal Processing - ICASSP06*, 2006.

[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2nd edition, 2001.

[6] R. Cristescu, B. Beferull-Lozano, and M. Vetterli. Networked Slepian-Wolf: Theory and algorithms. *1st European Workshop on Sensor Networks EWSN 2004*, 2004. Berlin, Germany.

[7] M. Gastpar, P. Dragotti, and M. Vetterli. The distributed Karhunen-Loève transform. In *International Workshop on Multimedia Signal Processing*, St. Thomas, US Virgin Islands, December 2002.

[8] A. Goel and D. Estrin. Simultaneous optimization for concave costs: Single sink aggregation or single source buy-at-bulk. In *SODA*, pages 499–505, 2003.

[9] B. B.-L. J. Acimovic and R. Cristescu. Adaptive distributed algorithms for power-efficient data gathering in sensor networks. In *Under Submission*.

[10] H. M. on Great Duck Island. Online data-set located at<http://www.greatduckisland.net>.

[11] S. Pattem, B. Krishnamachari, and R. Govindan. The impact of spatial correlation on routing with compression in wireless sensor networks. *Journal version, Under Submission*, 2005.

[12] S. S. Pradhan, J. Kusuma, and K. Ramchandran. Distributed compression in a dense microsensor network. *IEEE Signal Processing Magazine*, pages 51–60, March 2002.

[13] B. B.-L. R. Cristescu and M. Vetterli. Networked Slepian-Wolf: Theory, algorithms and scaling laws. *To appear in IEEE Transactions on Information Theory*, 2005.

[14] M. V. R. Cristescu, B. Beferull-Lozano and R. Wattenhofer. Network correlated data gathering with explicit communication: Np-completeness and algorithms. *To appear in IEEE/ACM Transactions on Networking*, 2005.

[15] P. Rickenbach and R.Wattenhofer. Gathering correlated data in sensor networks. In *DIALM-POMC Joint Workshop on Foundations of Mobile Computing, Philadelphia, PA, USA*, pages 60–66. ACM, Oct. 2004.

[16] A. Scaglione and S. D. Servetto. On the interdependence of routing and data compression in multi-hop networks. *International Conference on Mobile Computing and Networking - Mobicom'02*, 2002.

[17] S. D. Servetto. Sensing Lena - massively distributed compression of sensor images. *ICIP - International Conference on Image Compression*, September 2003.

[18] R. Szewczyk, A. Mainwaring, J. Polastre, and D. Culler. An analysis of a large scale habitat monitoring application. In *Second ACM Conference on Embedded Networked Sensor Systems (SenSys)*, November 2004.

[19] K. S. Y. Zhu and R. Sivakumar. Practical limits on achievable energy improvements and useable delay tolerance in correlation aware data gathering in wireless sensor networks. In *IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON), Santa Clara, California, USA,*, September 2005.