

# A DYNAMIC PROGRAMMING APPROACH TO DISTORTION-ENERGY OPTIMIZATION FOR DISTRIBUTED WAVELET COMPRESSION WITH APPLICATIONS TO DATA GATHERING IN WIRELESS SENSOR NETWORKS

Alexandre Ciancio and Antonio Ortega

University of Southern California  
Signal and Image Processing Institute  
ciancio@usc.edu, ortega@sipi.usc.edu

## ABSTRACT

We address a scenario where energy-constrained sensors in a wireless sensor network can choose among different distributed coding schemes to encode their data. We propose a framework where the network is described as a graph, with sensors representing the nodes, and where communication and processing costs are associated to edge weights and the coding schemes associated to states of operation. After describing data transitions and edge costs, we show that a shortest-path algorithm can be used to find the optimum network configuration, i.e., the one that leads to the lowest overall energy consumption.

## 1. INTRODUCTION

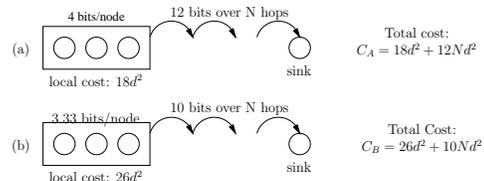
Wireless Sensor Networks (WSN) can offer mobility and versatility for a variety of applications, such as object detection/tracking, environment monitoring and traffic control [1]. Still, one of the main obstacles they face is that they often rely on batteries for power supply; thus limiting their energy 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, significantly reducing the amount of information that needs to be transmitted over the network, leading to a potentially large reduction in energy consumption. 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 [2] and Wavelets [3], others propose schemes that do not require internode communication, such as networked Slepian-Wolf coding [4, 5]. In general, DSC techniques face a trade-off between i) more processing at each node to achieve more compression and ii) less processing which would require more information (bits) to be sent to the sink. This trade-off has also been addressed by previous research. [6] 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 [7].

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 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, dif-

ferent coding methods may be better suited for different parts of the network. These methods can consist in 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 distant more than 4 hops from the sink, method “B” will lead to a lower energy consumption. In general, it is expected that sensors closer to the sink should benefit from coding schemes that offer smaller local cost, while sensors that are far from the sink should encode using schemes that require a smaller average number of bits per sensor for a given distortion. A similar conclusion is reached in [5].



**Fig. 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 need to be forwarded to the sink. The cost of transmitting  $k$  bits over a distance  $d$  was computed as  $kd^2$ .

In [8, 9] we introduced energy-aware distributed wavelet compression algorithms for WSN [8] and introduced a partial coefficient approach based on the lifting implementation [9]. Our goal was to generate the wavelet transform coefficients at the sensors, at the expense of a little extra energy spent with a few “local” transmissions, 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 transmitted. Our proposed partial coefficient approach [9] 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

This work was supported in part by CAPES, Brazil.

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.

This paper is organized as follows. In Section 2 we introduce the proposed framework, and describe states, transitions and costs for the state machine considered. Section 3 provides simulation results. We conclude the paper with a summary in Section 4.

## 2. PROPOSED FRAMEWORK

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 (Fig. 2). 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 (internode distances) 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 with using the same filterbank but with 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).

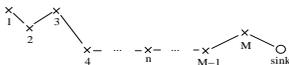


Fig. 2. 1-D path with  $M$  nodes to the sink.

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 [9], 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 [9] for details.

### 2.1. Problem Description

In order to find the best coding scheme for each of the sensors, we propose representing the network as a graph. Figure 3 illustrates

the graph associated to a 1D path including  $M + 1$  sensors (the last 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 to one choice of coding scheme for each node in the routing path from a leaf to the sink.

Let  $c_n(i, j)$  be the cost of the transition from method  $i$  at node  $n$  to method  $j$  at node  $n + 1$ ,  $n = 1, \dots, M$ , where  $i, j \in S = \{A, B, C, \dots\}$ , with  $A, B, C, \dots$  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  $\{i_1, i_2, \dots, i_n, \dots, i_M\}$  of coding schemes associated to each of the nodes such that  $\sum_n (c_n(i_n, i_{n+1}) + l_n(i_n))$  is minimum.

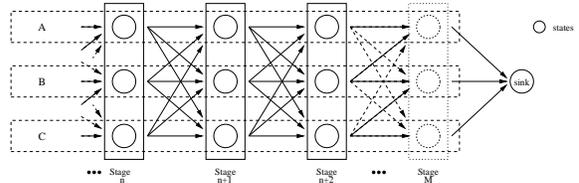


Fig. 3. Sensor network seen as a graph. Each sensor is a node that can be in one state of operation, associated with a coding scheme (A, B, C) and a position in the network. Edges correspond to transitions between states, and have transmission and processing costs (weight) associated to them.

We assume that the coding schemes in Figure 3 consist in wavelet transforms with progressively 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. These constraints limit the number of possible transitions in the graph, as illustrated in Figure 4. 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 [5], where the authors addressed the problem of joint rate allocation and transmission structure optimization for sensor networks.

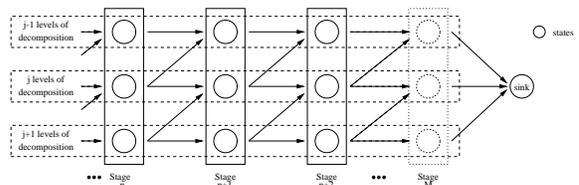


Fig. 4. Data transitions for partial coefficient approach.

We can now describe the state and transitions for the state machine of Figure 4) based on the partial coefficient approach.

### 2.2. State Description

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 ( $j$ ). We can define the following quantities of interest:

- $d_n$ : distance to next node;
- $R_{n,j}$ : rate allocation for coefficient at position  $n$  encoded with  $j$ -levels of decomposition;

- $D_n^2$ : cost to forward one bit to the sink. If we assume the cost to send  $k$  bits over  $d$  meters is  $kd^2$ , then  $D_n^2 = \sum_{k=n}^M d_k^2$  (sum of cost for each hop to the sink).

### 2.2.1. State I/O

Each state in the graph corresponds to a series of computations that depend on the node and coding scheme represented by the state. Specifically, if a node is at position  $n$  and is encoding data using a wavelet transform with  $j$  levels of decomposition, it is responsible for generating the  $j$ -level partial coefficient corresponding to node  $n$  and refining any previous partials that were also encoded with  $j$ -levels of decomposition and depend on the measurement at position  $n$  to be fully computed (Fig. 5).



Fig. 5. State I/O.

### 2.2.2. State Costs

Each state adds to the current path cost the computational costs associated to i) the corresponding partial coefficient generation and update and ii) the cost of forwarding the any new full coefficients to the sink. The computational costs are calculated in terms of the number of multiplications required, and depend on the specific state considered, while the transmission cost is given by  $R_{n,j}D_n^2$ .

### 2.3. Transition Description

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  $\dots \subset W_n^{j-1} \subset W_n^j \subset W_n^{j+1} \subset \dots$ . The information that is conveyed in each of the transition scenarios corresponding to different branches of the state machine can be seen in Figure 6.

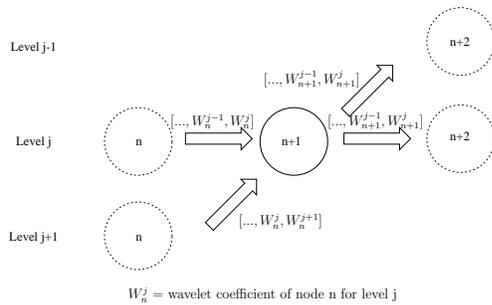


Fig. 6. Data transitions for partial coefficient approach.

#### 2.3.1. Transition Costs

In order to minimize impact of quantization, partial coefficients are quantized using  $N$  more bits than the corresponding full coefficients [9] (a typical value for  $N$  for the application considered is  $N = 3$ ). Assume that a given state  $(n, j)$  receives  $P(n, j)$  partial coefficients from the previous node. Figure 7 shows the state and edge transmission costs. Since partial coefficients are only forwarded during a few hops, the transmission cost for those  $N$  extra bits is only added to the branch costs corresponding to

transmission of these partials (the cost for the full coefficient was incorporated at the state where they were generated). If the next node maintains in the same coding scheme, partials for that level will be refined, and the transmission cost added to the path cost considers only the current hop length. However, if the next node is encoded using a simpler scheme, the partials from the higher level of decomposition will not be further refined (they will be sent as they are to the sink), and the cost of forwarding the  $N$  extra bits related to unprocessed partials to the sink is added to the path cost.

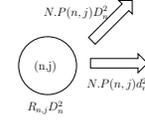


Fig. 7. State and edge transmission costs.

In addition to the transmission costs depicted in Figure 7, when there is a change in coding schemes a, typically small, processing cost is added to the branch to reflect a few extra operations the node has to perform to extract the lower level coefficients from the higher level ones.

### 2.4. Path Optimization

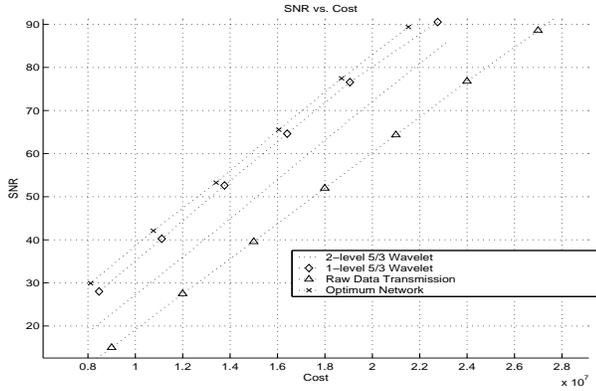
The partial coefficient approach guarantees that any computation at any given node requires only data from previous nodes. Also, due to the wavelet property discussed in Section 2.3, any node always has access to the past coefficients it needs to compute its own partial coefficient regardless of whether the data it is receiving was generated from the same coding scheme 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, and so 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 about dynamic programming and the shortest-path algorithm can be found in [10].

## 3. PERFORMANCE EVALUATIONS

For the simulations, we considered two different simple network configurations. We compared our optimization technique to solutions where the same coding scheme is used for all nodes in the network. This was done with the three available coding schemes. The coding schemes considered are raw (quantized) data transmission, wavelets with one level of decomposition, and wavelets with two levels of decompositions.

The input process data was created using a second order AR model, with poles placed such that a reasonably smooth output would be generated from white noise (poles were at  $0.99e^{\pm j\frac{\pi}{64}}$ ). Figure 8 shows the energy consumption of different single-scheme

methods (only one coding scheme for the whole network) at different distortion levels, in a network with 3 clusters of 5 sensors each (internode distance of 2m, intercluster distance of 37m).



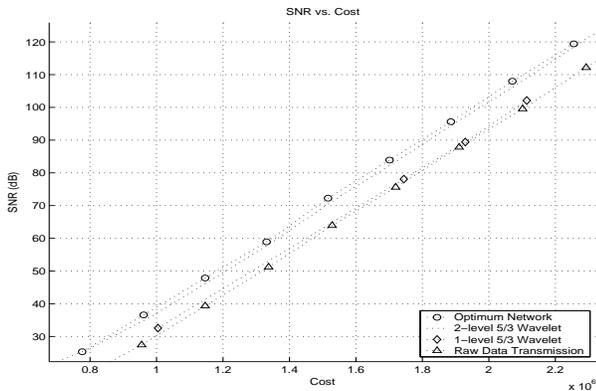
**Fig. 8.** Energy consumption comparison between methods with 3 clusters of 5 sensors each.

For this network, the optimum configuration such that energy consumption is minimized, obtained by the proposed dynamic programming framework is shown in Fig. 9.



**Fig. 9.** Optimum network configuration obtained for simulation in Fig. 8.

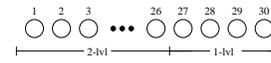
Figure 10 shows the energy consumption of different single-scheme methods at different distortion levels, in a network with 1 cluster of 30 sensors (internode distance of 1m).



**Fig. 10.** Energy consumption comparison between methods with 1 cluster of 30 sensors.

For this network, the optimum configuration obtained by the proposed dynamic programming framework is shown in Fig. 11.

Although the results suggest changes for just a few sensors when compared to the best single-scheme method, in general, such a behavior cannot be predicted beforehand. Also, as seen in Figures 8 and 10, different single-scheme methods perform differently for different network configurations. Network performance can be affected by a number of factors like the coding schemes being used, network topology, number of sensors, medium properties, data correlation, just to cite a few. Thus, a single-scheme approach might not necessarily result in the near-optimal performance. Optimization still proves to be necessary to point out the



**Fig. 11.** Optimum network configuration obtained for simulation in fig. 10.

configuration that will lead to the lowest energy cost. For the simulated case shown in Figures 8 and 10, for same distortion levels, the optimum network consumed around 6% less energy than the best single-scheme method (1-lvl wavelet for Figure 8 and 2-lvl wavelet for Figure 10) and around 32% less energy than simple raw (quantized) data transmission.

## 4. CONCLUSION

We have proposed a dynamic programming framework that can be used to assign coding schemes to each of the nodes in a WSN such that the energy consumption in the network is minimized. This algorithm is flexible enough to accommodate any network configuration, and could also be used as a benchmark to evaluate the relative performance of fast heuristics. Simulation results have shown that different coding algorithms can perform very differently (in terms of distortion and energy consumption) depending on a number of factors like network topology, medium properties, and data correlation, and that the proposed methodology provides a framework that can be used to minimize energy consumption in a WSN by efficiently assigning different coding schemes to different regions of the network.

## 5. REFERENCES

- [1] C. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, August 2003.
- [2] M. Gastpar, P. Dragotti, and M. Vetterli, "The distributed karhunen-loève transform," in *Proceedings of the 2002 International Workshop on Multimedia Signal Processing*, St. Thomas, US Virgin Islands, December 2002.
- [3] S. D. Servetto, "Sensing lena - massively distributed compression of sensor images," *ICIP - International Conference on Image Compression*, September 2003.
- [4] S. S. Pradhan, J. Kusuma, and K. Ramchandran, "Distributed compression in a dense microsensor network," *IEEE Signal Processing Magazine*, pp. 51–60, March 2002.
- [5] 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.
- [6] S. Patten, B. Krishnamachari, and R. Govindan, "The impact of spatial correlation on routing with compression in wireless sensor networks," in *Proceedings of the Third International Symposium on Information Processing in Sensor Networks*, April 2004.
- [7] A. Goel and D. Estrin, "Simultaneous optimization for concave costs: Single sink aggregation or single source buy-at-bulk," in *SODA*, 2003, pp. 499–505.
- [8] A. Ciancio and A. Ortega, "A distributed wavelet compression algorithm for wireless sensor networks using lifting," in *Proceedings of the 2004 International Conference on Acoustics, Speech and Signal Processing - ICASSP04*, Montreal, Canada, May 2004.
- [9] A. Ciancio and A. Ortega, "A distributed wavelet compression algorithm for wireless multihop sensor networks using lifting," in *Proceedings of the 2005 International Conference on Acoustics, Speech and Signal Processing - ICASSP05*, Philadelphia, USA, March 2005.
- [10] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, 2nd edition, 2001.