Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2011, Article ID 837128, 12 pages
doi:10.1155/2011/837128

Research Article

On Reducing Communication Energy Using Cross-Sensor Coding Technique

Kien Nguyen,[1] Thinh Nguyen,[1] and Sen-Ching Cheung[2]

[1] *School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97331, USA*
[2] *Department of Electrical and Computer Engineering, University of Kentucky, Lexington, KY 40506-0046, USA*

Correspondence should be addressed to Thinh Nguyen, thinhq@eecs.oregonstate.edu

Received 15 September 2010; Revised 31 December 2010; Accepted 20 January 2011

This paper addresses the uneven communication energy problem in data gathering sensor networks where the nodes closer to the sink tend to consume more energy than those of the farther nodes. Consequently, the lifetime of a network is significantly shortened. We propose a cross-sensor coding technique using On-Off keying which exploits (a) the tradeoff between delay and energy consumption and (b) the network topology in order to alleviate the problem of unequal energy consumption. We formulate our coding problem as an integer linear programming problem and show how to construct a number of codes based on different criteria. We show that the proposed technique can extend the lifetime of a small sensor network.

1. Introduction

Recent years have witnessed an explosive growth of sensor networks designed for environmental data gathering and monitoring [1–3]. A typical sensor network for data gathering consists of a large number of battery operated sensor nodes; each is capable of sensing and forwarding its data wirelessly to the processing node through its neighboring nodes. The sensor networks are often designed to operate at low power in order to prolong their lifetimes. However, building a truly energy efficient sensor network is still a challenging problem. There has been a large body of work on reducing energy consumption of sensor networks using different approaches, from low-voltage hardware designs [3] and transmission schemes [4] to in-network processing and routing algorithms [5].

Recently, several energy efficient algorithms for sensor networks based on the correlation characteristic of data have been studied. From signal processing perspective, if the measured data among the nodes are spatially correlated, a node can jointly compress its data and its neighbor's data to increase the capacity of the network [6–8] or to minimize the energy to transmit the data [9, 10]. The larger correlation results in larger energy saving [11]. The argument for energy reduction using data compression is based on the entropy concept from information theory [12]. The entropy represents the amount of information in terms of the number of bits. Therefore, by compressing the data to a smaller number of bits, less energy is required to send these bits. However, in many situations with practical sensor networks, the assumption of data correlation is not applicable.

An important problem in sensor network is the uneven energy consumption of the nodes, specifically the energy spent on communication. The nature of data forwarding in a sensor network implies that sensor nodes closer to the processing node (the sink) tend to spend more energy than those of the farther nodes in order to relay the accumulated data. Figure 1 shows an example in which data are collected and relayed through the internal nodes 2 and 3, and eventually to the processing node 4. Thus, nodes 2 and 3 consume two and three times more energy than node 1, respectively, because they have to relay the data of other nodes. This energy unfairness problem can significantly shorten the lifetime of a sensor network.

To see the seriousness of this problem, we assume that all the nodes have identical batteries designed to continuously measure and transmit *one sample of data* to the next hop for 3 months. Then, node 2 will run out of battery in 1.5 months, and node 3 will run out of battery in 1 month. Thus, while nodes 1 and 2 still function after one month, the sensor network is unusable due to node 3's failure. In

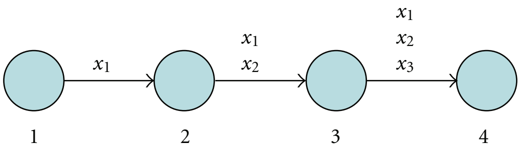


Figure 1: A simple sensor network. All measured data from nodes 1, 2, and 3 are relayed to node 4.

other words, the lifetime of this sensor network depends on the lifetime of node 3. Therefore, it is extremely important to incorporate the knowledge of the network topology into the design of a long-lived sensor network. We will show that *by allowing the internal nodes to re-encode the data they receive with appropriate codes based on its position in the network, the energy consumption can be significantly reduced.*

This work proposes a low-weight *cross-sensor* coding technique to reduce the communication energy and alleviate the problem of unequal energy consumption among nodes in a sensor network, thus improving its lifetime. Our approach is based on the following observation. Assume that digital transmission is used, and transmitting bit “1” consumes much more energy than bit “0”. This is typically true in optical communication where transmitting a bit “1” requires much energy to generate a laser pulse, while being silent represents a bit-“0” transmission. Our proposed coding technique exploits this difference to reduce the communication energy by limiting the number of bits “1” in the output codeword (*low-weight codeword*) and to use a *cross-sensor* coding technique to equalize the communication energy among the nodes. The proposed technique is designed to reduce the communication energy for fiberoptic sensor networks similar to the one used by Selker et al. as shown in Figure 2 [13]. This sensor network consists of temperature sensors connected to each other through a fiberoptic cable. Each sensor can measure temperature at temporal resolution of fractions of a minute.

2. Related Work

There has been a large amount of research on designing energy efficient sensor networks, from hardware designs and modulation schemes to MAC/routing protocols and in-network processing techniques. In this section, we list a few works in this area.

Many data aggregation algorithms for sensor networks have been recently proposed for energy optimization [6–9]. In [7], Cristescu et al. apply information network theory to sensor network by using a simple model for data compression and devise approximate algorithms for constructing shortest path tree to aggregate data from multiple sensors to a sink in order to minimize the total energy. While this approach can greatly reduce the energy consumption, it relies on large correlation of data at different sensors to achieve low energy consumption, which may not be true in many real-world scenarios.

Similar work on aggregation of correlated data in sensor network are studied by Enachescu and Sharaf in [10, 14]. In [14], the authors propose a simple randomized algorithm for routing data on a grid of sensors in a way that promotes data aggregation. They show that their randomized algorithm is a constant factor approximation to the optimal aggregation tree. On other hand, our work does not rely on data correlation.

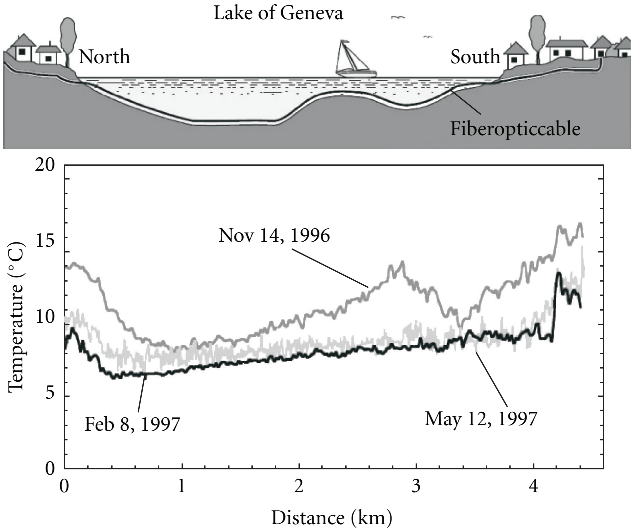


Figure 2: An fiberoptic sensor network deployed in lake Geneva.

Several researchers work on constructing the optimal prefix codes with unequal letters costs [15–17]. One solution approach to this problem is to formulate it as an integer linear programming problem. In [15], Karp also proved and proposed a method to construct the optimal codes via a set an integer linear programming problem.

Our approach is similar to the works of Liu and Asada [18], Kim and Andrews [19], and Prakash and Gupta [20], in which the authors optimize the energy consumption by exploiting different transmission energies for bits “1” and “0”. Our work is also similar to the silent based communication paradigm proposed in [21]. In [22], Dhulipala et al. also characterized the complexity of silence based communication. On the other hand, unlike previous approaches, our approach introduces a novel concept of coding data across different sensors which can alleviate the problem of uneven energy consumption at different sensors, thus prolonging the lifetime of a sensor network.

3. Sensor Network Model

This work does not address the overall energy consumption problem for the general class of sensor networks. Instead, it is useful for reducing communication energy for a number of sensor networks in which the following assumptions hold.

(1) Data is transmitted digitally using On-Off Keying (OOK) technique with bit “1” using more energy than that of bit “0”.

(2) Largest source of energy consumption is transmission.

(3) All the sensor nodes are equipped with a loosely synchronized clock.

(4) The sensor network is designed to collect data at low frequency, that is, low bandwidth requirement.

(5) Sensors are aligned and relay data in a linear fashion.

Assumption 1 is critical to our coding technique since we utilize the characteristic of OOK that more energy is required to transmit bit "1" than bit "0". Assumption 2 typically holds in many sensor networks, and, thus, minimizing the transmission energy significantly improves the network lifetime. Therefore, in our energy performance evaluation, we only consider the energy required to transmit, ignore the energy consumed by the nodes to listen and to perform any necessary computations. We will consider these other sources of energy consumption in our future work.

For this scheme to work, both receiver and sender must have a synchronized clock; thus, assumption 3 follows. As for assumption 4, our coding technique is more effective in terms of energy reduction when there is no or little restriction on the bandwidth requirement. This typical assumption often holds since sensor networks collect data at the large intervals, for example, on the order of minutes, resulting in a very low bandwidth. Therefore, the clocks at different nodes are only needed to be synchronized to a certain resolution.

Assumption 5 enables a node to code the data effectively. This assumption arises with the fundamental characteristic of sensor networks where data is only sent in one direction, that is, from sensor nodes to the sink as most of the sensor networks are used for data aggregation. Furthermore, our work focuses on small-scale environmental sensor applications in which a typical sensor network includes tens of sensors.

Finally, the discussion on transmitted bit errors is outside the scope of this paper.

4. Coding Schemes

4.1. 1-Bit Coding. Typical entropy coding techniques, such as Huffman coding [23], aim to minimize the number of bits per symbol, given a probability distribution of the source data. The symbols with frequent occurrences are coded using fewer number of bits than those of the rare symbols. As a result, the average number of bits per symbol is reduced. However, Huffman coding only minimizes the number of bits, regardless whether the bit is "0" or "1". This may result in higher transmission energy. To reduce transmission energy, one can use the 1-bit coding scheme [20] in which, every codeword consists of at most 1 high bit as below:

$$\begin{aligned} 0 &\longrightarrow 00000 \\ 1 &\longrightarrow 00001 \\ 2 &\longrightarrow 00010 \\ 3 &\longrightarrow 00100 \\ 4 &\longrightarrow 01000 \\ 5 &\longrightarrow 10000 \end{aligned} \tag{1}$$

Assume that it takes no energy to transmit bit "0", then the transmission energy per symbol is much smaller for the 1-bit coding than the traditional coding schemes such as direct binary coding or Huffman coding. The 1-bit coding scheme employs fixed-length codes for differentiation of symbol boundary. Furthermore, the receiver and sender's clocks must be synchronized in order for the receiver to detect the *silent* interval of bit "0". As a result, the 1-bit coding scheme results in longer average time to transmit a symbol.

The 1-bit coding technique can be generalized to k-bit coding which uses slightly more energy per symbol in order to reduce the delay. A k-bit coding symbol consists of at most k bits "1" per codeword. Therefore, the number of possible k-bit codewords of length n is

$$M(n,k) = \sum_{i=0}^{k} \binom{n}{i}. \tag{2}$$

The individual k-bit codewords of length n are then simply the enumeration of the combinations $\binom{n}{i}$ for $i = 0 \cdots k$. Clearly, larger k results in larger energy consumption. We note that if the number of possible symbols exceeds 2^n, where n is the length of a k-bit symbol, then n will be increased, but k remains the same in order to keep the transmission energy constant.

4.2. Cross-Sensor Coding Technique. The previous coding scheme reduces the energy consumption at each sensor; however, it cannot solve the problem of energy unfairness as explained in Section 1. One may think that the relay architecture used in the sensor network fundamentally produces this energy unfairness problem since some nodes, (e.g., the internal nodes) will have to transmit more data, thus resulting in higher energy consumption. However, this is not necessarily the case, the cross-sensor coding technique described below aims to solve this problem.

We first begin with the definition of energy unfairness which ultimately affects the lifetime of a sensor network. The energy unfairness is defined as follows.

Definition 1. One has

$$F_E = \frac{1}{n}\sum_{i=1}^{n}\left(X_i - \frac{\sum_{j=1}^{n} X_j}{n}\right)^2, \tag{3}$$

where n denotes the number of sensors, X_i denotes the average transmission energy at node i which is proportional to the number of 1's in a codeword.

Intuitively, the energy unfairness represents the variance of the average energy consumption by the nodes. The lower value of F_E indicates higher fairness and, thus, is desirable in a long-lived sensor network. We will show that our cross-sensor coding technique can reduce F_E significantly.

We would like to mention that there are different ways of defining energy fairness. All of these definitions ensure that in an ideal case, all nodes consume the same amount of energy. For example, a metric for energy fairness in cooperative wireless networks was proposed by Chen et al. [24]. Their definition is based on a cooperation matrix, and for which they showed that in general, the energy constraint and energy fairness cannot be satisfied simultaneously if each node uses a fixed set of relays.

To illustrate our coding techniques and their advantages, we first consider the traditional data relay in a simple sensor network consisting of four sensors and a processing node as shown in Figure 3.

4.2.1. Traditional Data Relay. Suppose the sensors are designed to measure the data with 8-bit resolution. Assume that at time t, the measured data for nodes 1, 2, 3, and 4 are 10101111, 00100111, 11100111, and 11111100, respectively. Assuming that propagation delay is negligible, then to transfer these data to the processing node (from left to right in Figure 3), the traditional relay method requires: node 1 transmits 6 bit 1's, node 2 transmits 10 bit 1's, node 3 transmits 16 bit 1's, and node 4 transmits 22 bit 1's. Therefore, the total number of transmitted bits equals to 6 + 10 + 16 + 22 = 54 bits. Assume that the energy for transmitting a bit "1" is 1 joule and for transmitting bit "0" is 0 joule, then the total energy consumption per one round of data gathering in this sensor network is 54 joules. Furthermore, the energy consumption of node 4 is almost 4 times the larger than that of node 1. In general, it is easy to show that the average number of transmitted bits per sample; hence, the energy per sample, is $O(mn^2)$ where m is the number of bits per sample (data resolution) and n is the number of sensor nodes.

4.2.2. Direct 1-Bit Coding. Now, suppose we apply 1-bit coding to the 8-bit data directly, and each node relays the data in the same way as the traditional method. Since each sample data is coded with only one bit "1", the total number of transmitted bits for this scheme scheme equals 1 + 2 + 3 + 4 = 10 bits, resulting a factor of 5 in energy reduction compared to that of the traditional coding technique. The direct 1-bit coding data scheme results in much higher delay. Assume that a node relays the data in the next time slot after it receives a data sample completely, then there is an additional 255 bit delay per hop. This relay technique (also called *store and forward* scheme) is employed in packet-switched networks. The *store and forward* scheme waits until all bits in a packet are received before transmitting the first bit of the packet onto the next link. In our case, a packet is a data sample (255 time slots). Consequently, the time it takes for the first bit of node 1's data to arrive at the processing node equals to 255 × 3 = 765 time slots as compared to only 8 × 3 = 24 time slots for traditional binary coding. In general, we prove the following theorem regarding the direct 1-bit coding scheme for sensor networks.

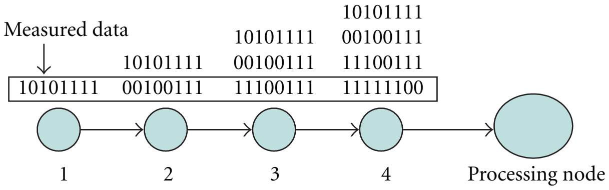


Figure 3: All measured data from nodes 1, 2, 3, and 4 are relayed to the processing node.

Proposition 2. *Giving*

(1) *a sensor network consisting of n sensor nodes and one processing node, all are aligned and forward data in a linear fashion,*

(2) *data gathering is based on store and forward scheme with a packet being a data sample of m-bit resolution,*

(3) *measured data are randomly and uniformly distributed,*

then, using direct 1-bit coding scheme with time slot of T second per bit results in

(1) *the average number of transmitted bits TB per sample data per node is*

$$\mathrm{TB} = \left(1 - \frac{1}{2^m}\right)\frac{n+1}{2}, \tag{4}$$

(2) *the achievable bandwidth B is*

$$B = \frac{mn}{T(2n-1)(2^m-1)}\,[bps], \tag{5}$$

(3) *the energy fairness is*

$$F_E = \left(1 - \frac{1}{2^m}\right)^2\left(\frac{n(n+1)(n-1)}{12}\right). \tag{6}$$

Proof. To prove (4), one notes that the first node transmits its sample data to node 2 using at most 1 bit "1". There is a $1/2^m$ chance that node 1 does not transmit any bit "1". This happens when its measured value maps to a 1-bit codeword that contains all zeros. Thus, the average number of bits transmitted by node 1 is $(1 - 1/2^m)$. Since node 1's bit must be relayed through $n-1$ nodes before arriving at the processing nodes, the total number of transmissions for this bit is therefore $(1 - 1/2^m)n$. Similarly, the bit from a node 2, will have to be transmitted through $n-2$ nodes, resulting in $(1 - 1/2^m)(n-1)$ transmissions. Thus, the average number of transmissions for all the sample data is the total number of transmitted bits per n sample data

$$\begin{aligned}&\left(1-\frac{1}{2^m}\right)n + \left(1-\frac{1}{2^m}\right)(n-1) + \cdots \left(1-\frac{1}{2^m}\right)\\&\quad = \left(1-\frac{1}{2^m}\right)\frac{n(n+1)}{2}.\end{aligned} \tag{7}$$

Divided the result by n, we obtain the average number of bits per sample data

$$\mathrm{TB} = \left(1 - \frac{1}{2^m}\right)\frac{n+1}{2}. \tag{8}$$

To prove (5), one notes that $2^m - 1$ is the number of time slots required to encode m-bit data using 1-bit coding. Since there are n nodes and the time for the first data bit of the first node to arrive at the last node is $(n-1)(2^m-1)T$. All the subsequent bits will arrive at the processing node at the rate of $1/T$ bps. There are a total of n data sample transmitted from the last node to the processing node, and the time it takes for all n data samples to arrive at the processing node is $n(2^m-1)T$. Adding the delay of the first bit and the transmission time of n samples together, we obtain the total time to receive all n samples of m bits each. Thus, the bandwidth is

$$\frac{mn}{T(2n-1)(2^m-1)}. \tag{9}$$

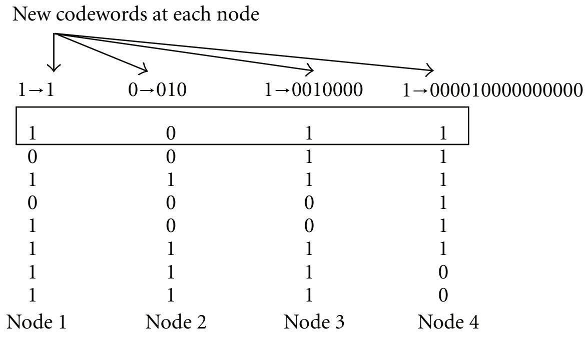
The second sample data of the first node can be sent after all the processing node has received completely the first data samples from all the nodes.

To prove (6), one can explicitly compute the formula in (3). To compute X_i, one notes that $X_{i+1} = X_i + (1 - 1/2^m)$. We know $X_1 = 1 - 1/2^m$ thus, X_i can be shown to be $i(1 - 1/2^m)$. □

We note that using direct 1-bit coding results in lower energy consumption, but the energy unfairness is still on the order of $O(n^2)$. In other words, as the number of relay nodes increases, the lifetime of the sensor network decreases significantly. Thus, we propose the following cross-sensor coding technique that saves communication energy as well as resolves the energy unfairness issue.

4.2.3. Cross-Sensor 1-Bit Coding. Unlike the previous approaches, in which each sensor encodes its data separately, cross-sensor coding technique allows each sensor to encode its data and its neighbor's data jointly using variable-length codewords where the codeword's length is based on the location of a sensor.

Figure 4 illustrates the cross-sensor coding technique for 8-bit sample data. For easy visualization, the 8-bit resolution data at each sensor nodes are now arranged vertically with the most significant bits (MSB) at the top, and the least significant bits at the bottom. First, we consider sending the MSBs of the data measured at different sensors to the processing node. The MSB of the measured data at the node 1 is "1", so node 1 will send the codeword "1" to the node 2. The MSB of the measured data at node 2 is 0, so node 2 will send the codeword "010", representing both MSBs at nodes 1 and 2 ("10"), to node 3. The MSB of the measured data at node 3 is 1, so node 3 will send the codeword "0010000" to node 4, representing MSBs at nodes 1, 2, and 3 ("101"). Finally, the MSB of the measured data at node 4 is "1", so node 4 will send the codeword "000010000000000" to the processing node. Basically, the code sent by node n is the 1-bit codeword, representing all the possible bit patterns at the nodes 1 to n.



Node 1	Node 2	Node 3	Node 4
1	0	1	1
0	0	1	1
1	1	1	1
0	0	0	1
1	0	0	1
1	1	1	1
1	1	1	0
1	1	1	0

FIGURE 4: Individual bits are transmitted through each node. Each sensor combines the cumulative data and produces a new codeword with longer length.

We note that the length of the codeword increases roughly by a factor 2 after each hop. This is because there are 4 maximum possible data patterns that node 2 can send to node 3, that is, 0 or 1 from node 1, and 0 or 1 from node 2. Similarly, there are 8 possible data patterns that node 3 can send to node 4. In general, there are 2^n possible data patterns that represent different MSB patterns for nodes 1 to n. Thus, 1-bit codeword sent out by node n will have a length of $2^n - 1$ bit times.

Using the above scheme, when the processing node receives a codeword "000010000000000", it will be able to infer that the MSBs at nodes 1, 2, 3, and 4 are 1, 0, 1, and 1, respectively, by simply converting "000010000000000" (11 in decimal) to a binary coding, that is, 1011. After the processing node receives the MSBs of all the nodes, the first node can start to transmit the second bit, and the process repeats until all the 8-bit data are transmitted. Clearly, using cross-sensor 1-bit coding, each sensor only needs to send at most 1 bit "1" per hop, thus eliminates the problem of uneven transmission energy. However, the length of an 1-bit codeword increases roughly by a factor 2 after each hop, that is, 1-bit codeword sent out by node n will have a length of $2^n - 1$ bit times, resulting in longer delay or equivalently bandwidth.

The cross-sensor coding technique achieves energy fairness and efficiency by exploiting the individual node's knowledge of the topology. In particular, each sensor needs to know its position in the topology, and the data from other sensors in order to encode the new codeword with appropriate length. Furthermore, to be able to receive the correct codeword, each sensor must have a synchronized clock to enable correct timing of each bit. Cross-sensor coding also introduces additional delay per hop due to the store and forward scheme. Figure 5 shows that the delay increases exponentially as data traverse to the processing nodes. Note that for two consecutive nodes, the transmission time slots of one node are the receiving time slots of another node. We emphasize that a transmission time slot of a node does not mean that node has to transmit bit "1" during this time slot. Rather, a transmission time slot means that this time slot is used to indicate the transmission of the data bits either "0" or "1". There will not be an actual transmission

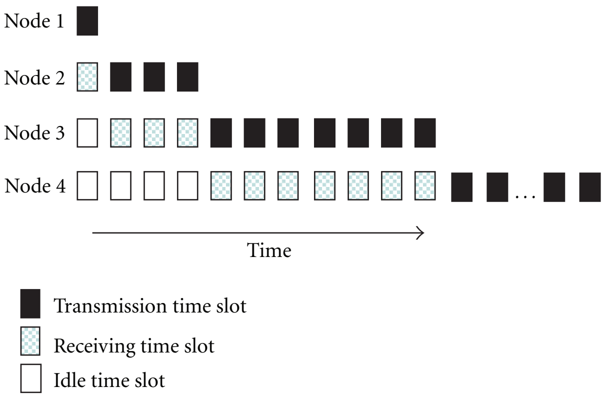


Figure 5: Delay in cross-sensor coding. Staggering delay is introduced at each node as shown by the squares representing time slots. The closer the nodes to the processing node are, the higher delay is. Blank squares represent the idle time slots, the wavy squares represent the transmission time slots, and the brick squares represent the time slots during which the node is receiving information.

if the data bit is “0”. Due to 1-bit coding, there are also idle time slots which are necessary for the node to differentiate the codewords. The following theorem characterizes the bandwidth, the transmission energy, and the energy fairness associated with 1-bit cross-sensor coding.

Proposition 3. *Given the same conditions as those of Proposition 2, using 1-bit cross-sensor coding scheme with time slot of T second per bit results in*

(1) *the average number of transmitted bits (TB) per sample data is*

$$\mathrm{TB} = m\left(1 - \frac{1}{n} + \frac{1}{n2^n}\right), \tag{10}$$

(2) *the achievable bandwidth B is*

$$B = \frac{mn}{T(2^{n+1} - n - 2)}\ [\mathrm{bps}], \tag{11}$$

(3) *the energy fairness is*

$$F_E = \frac{1}{3n}\left(1 - \frac{1}{4^n}\right) - \frac{1}{n^2}\left(1 - \frac{1}{2^n}\right)^2. \tag{12}$$

Proof. To prove (10), one notes that with probability 1/2, the first node will not transmit anything, that is, the first bit is bit “0”. With probability 1/4, node 2 will not transmit anything to node 3 (pattern “00”). With probability 1/8, node 3 will not transmit anything to node 4 (pattern “000”). In general, with probability $1/2^n$, node n will not transmit anything. Thus, the expected transmission energy equals to

$$\sum_{i=1}^{n}\left(1 - \frac{1}{2^i}\right) = n - 1 + \frac{1}{2^n}. \tag{13}$$

Hence, the average number of transmissions per node is

$$1 - \frac{1}{n} + \frac{1}{n2^n}. \tag{14}$$

If the data is m-bit resolution, we just need to transmit a total of

$$m\left(1 - \frac{1}{n} + \frac{1}{n2^n}\right) \tag{15}$$

bits per data sample.

As for (11), it is clear that each node i introduces additional $2^i - 1$ bit times as shown in Figure 5. Hence, the total number of bit times after n is

$$\sum_{i=1}^{n}\left(2^i - 1\right) = 2^{n+1} - n - 2. \tag{16}$$

Thus, the bandwidth is

$$B = \frac{mn}{T(2^{n+1} - n - 2)}. \tag{17}$$

Finally, (12) can be obtained by computing the formula in (3) with $X_i = 1 - 1/2^i$. □

Note that energy unfairness from using cross-sensor coding scheme now asymptotically approaches 0 as the number of sensors approaches infinity. We can also extend cross-sensor 1-bit coding to cross-sensor k-bit coding.

4.2.4. Cross-Sensor k-Bit Coding. To reduce the delay or increase the bandwidth, one can use k-bit coding technique where the maximum number of high bits in a codeword is k. The code length n is a variable such that:

$$2^m \leq \binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{k}, \tag{18}$$

where m is the data resolution.

5. Optimal Low-weight, Cross-Sensor Coding

Thus far, we have considered an ideal situation in which the transmission energy of bit “0” is assumed to be 0. We now show how to find an appropriate low-weight, variable-length, prefix code for each sensor based on (a) its position, (b) the transmission energy of bits “0” and “1”, and (c) a certain delay-energy consumption tradeoff. The term *low-weight* means a small number of bits “1”, leading to small transmission energy. We note that the delay-energy consumption tradeoff is equivalent to the bandwidth-energy consumption tradeoff when using the cross-sensor coding technique.

Data Model. In many sensor networks, the observed data might be correlated across different sensors in spatial proximity according to some joint distribution. In this case, it is beneficial to consider these correlation for compressing the data, leading to fewer transmissions, and thus less energy consumption. For example, one might employ differential coding technique that entropy codes the differences in values of the observed data between two spatially consecutive sensors. This differential coding would reduce the dynamic range of the data, increase the compression ratio, and reduce the overall energy consumption. We consider this approach as a high-level data modeling technique that when used in conjunction with an entropy coding technique, such as Huffman coding, will further compresses the data. Like other entropy coding techniques, Huffman coding assumes a distribution of the observed symbols to be i.i.d (identically independent distributed). In other words, each symbol i occurs with some probability p_i, and is independent of each other.

Our proposed cross-sensor technique is an entropy coding technique, thus makes the typical i.i.d assumption. However, there is one key difference. The symbols observed at a sensor k are not the actual data observed at sensor k. Rather, it is a sequence of bits, one bit from each of the previous sensors as described in Section 4.2.3. Nevertheless, each of these symbols at a sensor k will still have an occurrence probability p_i. In this paper, we assume that the distributions of these symbols at every sensors are i.i.d. In future work, we will consider higher-level data modeling techniques which take advantage of the data correlation to further reduce the energy consumption.

Optimality of Huffman Coding. An optimal code is an entropy code that produces the shortest average output codeword length. Given n source symbols $i = 1 \cdots n$, a binary entropy code produces a distinct binary string of length l_i to represent each source symbol i. Thus, the average length of a code is mathematically defined as

$$L = \sum_{i=1}^{n} \mathrm{p}_i l_i, \tag{19}$$

where p_i is the occurrence probability of the source symbol i. It is well known that Huffman code is the optimal entropy code, in the sense that it results in the shortest average code length.

Intuitively, if bits “0” and “1” require the same transmission energies, then the average delay and the transmission energies are directly proportional to the length of the code. Therefore, minimizing the expected code length using Huffman coding also minimizes the average energy consumption and delay simultaneously. In other words, there is no tradeoff between the average energy consumption and the delay.

Energy and Delay Tradeoff of Cross-Sensor Coding. However, the previous argument does not hold in the case where transmission energies of bits “0” and “1” are different. As discussed in Section 4.2.4, if a larger number of bits “1” in a codeword is allowed, more energy is used to transmit that codeword. On the other hand, with a larger number of bits “1” the average code length or equivalently the delay will be reduced. This is due to the fact that, for a prespecified maximum length over all the codewords, the number of possible codewords increases as the number of allowable bits “1” increases. Therefore, there is a tradeoff between the amount of transmission energy and delay. Importantly, there is not a single code, for example, Huffman code that simultaneously minimizes both energy and delay. Thus, the type of applications would impose the appropriate requirements on delay and energy consumption.

Designing Codes with Optimal Tradeoff. Our aim in this section is to design a cross-sensor code that satisfies prespecified requirements on the average delay and energy consumption. For example, one may want to employ a code that minimizes the energy consumption while the delay may not exceed a certain threshold. Or conversely, one may want a code that minimizes the delay while the energy consumption may not exceed a certain threshold. Finally, one can also design a code that minimizes the weighted sum of the energy consumption and the delay. We will show that each of these formulations can be casted as an integer linear programming problem. To do so, we must define appropriate optimization variables and represent the delay and energy consumption as functions of these optimization variables.

Our approach is based on the approach proposed by Karp [15] on minimum-redundancy coding for the discrete noiseless channel. However, we should emphasize that it is not trivial to apply Karp’s approach directly to our problem. Furthermore, formulating precisely the canonical form of an optimization problem class often requires some ingenuities. Common techniques include introducing new constraints and/or variables, or even modifying the objective as long as it can be shown that the solution to the modified problem will be identical to that of the original problem.

Code with Unequal Costs. We now briefly discussed the formulation of codes with unequal cost problem on which our derivation is based. Figure 6 shows a binary tree, representing a binary prefix code with unequal costs for bits “0” and “1”. The length of each branch represents the cost associated with bits “0” and “1”. Each leaf node represents a binary codeword which is a sequence of the branch labels starting from the root to the leaf node, similar to a Huffman tree. The goal is to design a code that minimizes the average cost given the probability distribution of the symbols.

Karp approached this problem by formulating an equivalent integer linear programming problem. (To the best of our knowledge, it is currently unknown whether the problem of optimal unequal cost letters is NP hard or polynomial time solvable.) Karp assumed that the cost of each digit is discrete and finds the necessary and sufficient conditions for the prefix code in the form of a set of linear inequalities involving integer solutions. The key to Karp’s formulation is to design a tree as shown in Figure 6. Specifically, the cost of a codeword is the total length from the root to a leaf node representing that codeword. For example, if the discrete costs of the digits ”0” and ”1” are 1 and 2, respectively, then the codeword “100” would have a cost of $2 + 1 + 1 = 4$. The optimal tree has a following special property. Denote a_j and b_j the number of leaf nodes and nonleaf nodes at level (depth) j, where j is an integer. Let E_0 and E_1 denote the cost of digits “0” and “1” and assume $E_1 \geq E_0$, respectively; then,

$$a_j + b_j \leq b_{j-E_0} + b_{j-E_1}, \quad \forall j \geq E_1. \tag{20}$$

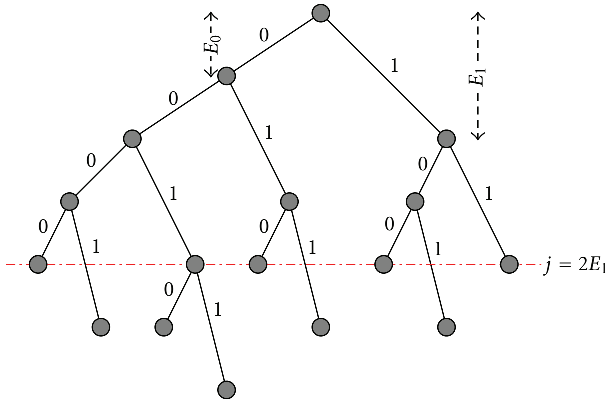


Figure 6: Prefix code as a tree with the cost of a symbol represented by the length of its branch.

This is a generalization of the well-known Kraft's inequality for a prefix code with equal cost. Importantly, this will be used as the constraint in our integer linear programming formulation. Once the integer solutions a_j and b_j are found, they can be mapped directly to a valid prefix code. We refer the interested readers to [15] for more details.

The Optimal Cross-Sensor Codes as Code with Unequal Costs. In our problem, the transmission energy can be thought as the cost of a digit in a codeword. Thus, the problem of minimizing the energy consumption can be formulated as finding a prefix codebook whose digits have unequal costs which results in minimum cost. However, a direct application of Karp's approach may result in a undesirable long average code length, or delay.

If we take both delay and energy consumption into consideration, the problem becomes nontrivial. The problem is complicated further with multiple sensors. As discussed earlier, there is no single optimal code that is simultaneously optimal for both delay and energy consumption. Rather, depending on specific applications, a certain tradeoff between delay and energy consumption must be made. Therefore, it is necessary to find mathematical expressions for both delay and energy consumption. To derive these expressions and, consequently, to formulate our problem as an integer linear programming formulation, we define the following notations.

(i) N: the number of sensors in the network.

(ii) n_k: the number of possible symbols observed at sensor k. For a straight-line topology, $n_i = 2^i - 1 > n_j$ if sensor i is closer to the sink than sensor j.

(iii) X_i^k: the input symbol i at node k, $i = 1 \cdots n_k$.

(iv) C_i^k: the codeword for symbol X_i^k.

(v) N_{ki}^0, N_{ki}^1: the number of bits "0" and bit "1" in a codeword C_i^k, respectively.

(vi) p_i^k: the probability of the symbol X_i^k observed at node k and $i = 1 \cdots n_k$.

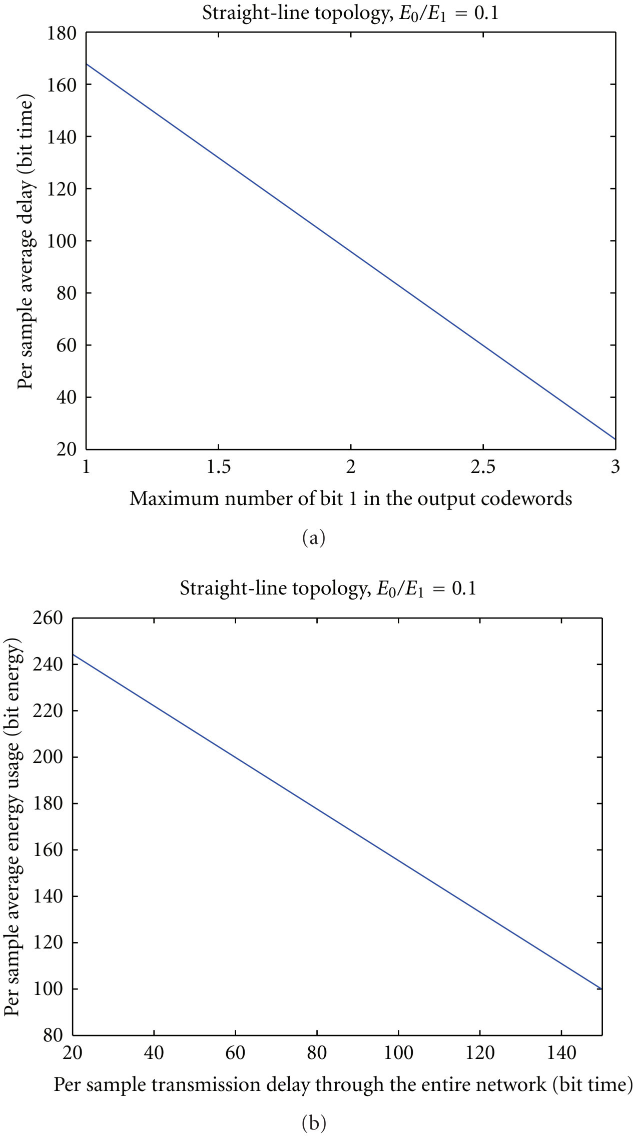


Figure 7: (a) Per sample delay as a function of K (delay optimization). (b) Average energy as a function of per sample delay (energy optimization).

(vii) E_0 and E_1: transmission energies of "0" and "1", respectively. We assume that E_0 and E_1 are positive integers. Note that one can approximate *real* cost with *integer* cost by scaling the costs appropriately.

(viii) m_k: the maximum cost of a codeword at node k. m_k can be set to a large value to guarantee a solution.

Expression for Energy Consumption. We first find the expression for transmission energy in terms of optimization variables which will be introduced at appropriate times. Observe that the cost (transmission energy) d_i^k of transmitting a symbol X_i^k by node k is

$$d_i^k = E_0 N_{ki}^0 + E_1 N_{ki}^1, \tag{21}$$

and the average transmission energy per symbol at node k is given by

$$E^k = \sum_{i=1}^{n_k} p_i^k d_i^k. \tag{22}$$

Note that the transmission energy of 0 can be obtained by setting each d_i^k to 0, or equivalently all the codewords are of the form $000 \cdots 0$. However, these codes are unusable as they are not uniquely decodable. Thus, the constraint for uniquely decodable codes must be incorporated into our problem.

Fortunately, Karp has shown that any prefix code of unequal cost(and therefore also uniquely decodable) must satisfy the linear constraints

$$a_j^k + b_j^k \le b_{j-E_0}^k + b_{j-E_1}^k, \quad \forall j \ge E_1, \tag{23}$$

where a_j^k denotes the number of codeword (leaf nodes) of cost j and b_j^k denotes the number of nonleaf nodes at cost j.

For example, let $E_0 = 1$, $E_1 = 2$, and $j = 4$ (level 4 of the tree) as shown in Figure 6, then $a_j^k = 4$ and $b_j^k = 1$ since there are four leaf nodes and one nonleaf node at level 4, respectively. Similarly, $b_{j-E_0}^k = 3$ and $b_{j-E_1}^k = 2$. Clearly in this case, $a_4^k + b_4^k = b_3^k + b_2^k = 5$, so the constraints (23) are satisfied.

Next, we need to incorporate these constraints into the energy consumption expression, that is, relating E^k to a_j^k and b_j^k. To do so, we introduce new binary variables y_{ij}^k defined as

$$y_{ij}^k = \begin{cases} 1 & \text{if } d_i^k = j, \\ 0, & \text{otherwise,} \end{cases} \tag{24}$$

then

$$d_i^k = \sum_{j=1}^{m_k} j y_{ij}^k, \qquad a_j^k = \sum_{i=1}^{n_k} y_{ij}^k. \tag{25}$$

Essentially, the cost d_i^k of a codeword C_i^k at sensor k can now be parametized in y_{ij}^k which in turn is related to a_j^k. Note that there is no obvious way to link d_i^k to the level j of the tree and a_j^k, more generally, to the constraints on prefix codes, that is, the constraints (23). Introducing the new variables y_{ij}^k is also a trick to convert an integer optimization problem into a binary optimization problem.

Substituting $d_i^k = \sum_{j=1}^{m_k} j y_{ij}^k$ into (22), we obtain the expression for energy consumption at node k as

$$E^k = \sum_{i=1}^{n_k} \sum_{j=1}^{m_k} p_i^k j y_{ij}^k, \tag{26}$$

and for the entire network as

$$E = \sum_{k=1}^{N} \sum_{i=1}^{n_k} \sum_{j=1}^{m_k} p_i^k j y_{ij}^k. \tag{27}$$

Expression for Delay. To find the expression for the delay, we first find a mathematical expression for the length of a codeword. Denote l_i^k as the length of a prefix codeword C_i^k at node k. We now relate l_i^k to the prefix code constraints in (23) via introducing new optimization variables. Observe that

$$l_i^k = N_{ki}^0 + N_{ki}^1, \tag{28}$$

therefore, the average cost (transmission energy) to transmit the code C_i^k is

$$d_i^k = E_0 \left(N_{ki}^0 + N_{ki}^1 \right) + (E_1 - E_0) N_{ki}^1. \tag{29}$$

Define new binary variables x_{ij}^k such that

$$x_{ij}^k = \begin{cases} 1 & \text{for } j \le N_{ki}^1, \\ 0 & \text{for } j > N_{ki}^1, \end{cases} \tag{30}$$

then

$$\begin{aligned} N_{ki}^1 &= \sum_{j=1}^{m_k} x_{ij}^k, \\ d_i^k &= E_0 l_i^k + (E_1 - E_0) N_{ki}^1 = E_0 l_i^k + (E_1 - E_0) \sum_{j=1}^{m_k} x_{ij}^k \\ \Longrightarrow \sum_{j=1}^{m_k} j y_{ij}^k &= E_0 l_i^k + (E_1 - E_0) \sum_{j=1}^{m_k} x_{ij}^k. \end{aligned} \tag{31}$$

Thus, we can represent the output codeword length of node k as a linear combination of the integer variables x_{ij}^k and y_{ij}^k as

$$l_i^k = \frac{1}{E_0} \sum_{j=1}^{m_k} j y_{ij}^k - \frac{E_1 - E_0}{E_0} \sum_{j=1}^{m_k} x_{ij}^k. \tag{32}$$

The average delay at node k can then be expressed as

$$D^k = \frac{1}{E_0} \sum_{i=1}^{n_k} \sum_{j=1}^{m_k} p_i^k j y_{ij}^k - \frac{E_1 - E_0}{E_0} \sum_{i=1}^{n_k} \sum_{j=1}^{m_k} p_i^k x_{ij}^k, \tag{33}$$

and the overall delay as

$$D = \sum_{k=1}^{N} \left(\frac{1}{E_0} \sum_{i=1}^{n_k} \sum_{j=1}^{m_k} p_i^k j y_{ij}^k - \frac{E_1 - E_0}{E_0} \sum_{i=1}^{n_k} \sum_{j=1}^{m_k} p_i^k x_{ij}^k \right). \tag{34}$$

Again, introducing the new variables x_{ij}^k is a trick to relate the length of a codeword to the constraints on the prefix. x_{ij}^k might not carry a physical meaning.

Integer Linear Programming Formulation. Given the expressions for code length (delay) and transmission energy, we now can formulate a variety of optimization problems regarding the energy consumption and delay (bandwidth). We provide three examples below.

As a first example, we formulate the problem of finding a prefix code that minimizes the average transmission energy

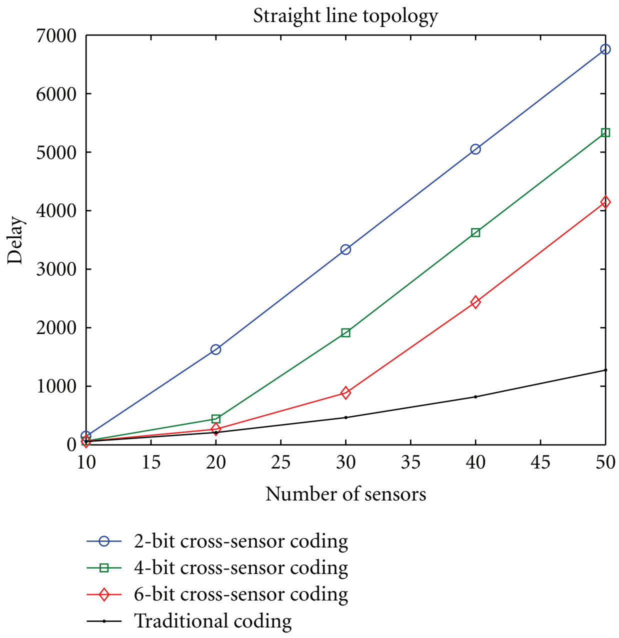


Figure 8: Delay (number of time slots) versus number of nodes for straight line topology.

of the network subject to (a) the transmission energy of any node may not exceed a certain value W and (b) the average delay has to be smaller than a value X as follows:

$$\begin{aligned} &\text{minimize} \quad \sum_{k=1}^{N}\sum_{i=1}^{n_k}\sum_{j=1}^{m_k} p_i^k j y_{ij}^k \\ &\qquad \text{s.t.} \quad b_0^k = 1; \qquad b_j^k = 0, \quad \text{for } j < 0. \end{aligned} \tag{35}$$

For $1 \le k \le N$ and $1 \le j \le m_k$,

$$\sum_{i=1}^{n_k} y_{ij}^k + b_j^k \le b_{j-E_1}^k + b_{j-E_0}^k. \tag{36}$$

For $1 \le k \le N$ and $1 \le i \le n_k$,

$$\sum_{j=1}^{m_k} y_{ij}^k = 1, \tag{37}$$

$$\sum_{i=1}^{n_k}\sum_{j=1}^{m_k} p_i^k j y_{ij}^k \le W, \tag{38}$$

$$\sum_{k=1}^{N}\left(\frac{1}{E_0}\sum_{i=1}^{n_k}\sum_{j=1}^{m_k} p_i^k j y_{ij}^k - \frac{E_1 - E_0}{E_0}\sum_{i=1}^{n_k}\sum_{j=1}^{m_k} p_i^k x_{ij}^k\right) \le X, \tag{39}$$

where b_{ij}^k, x_{ij}^k, and y_{ij}^k are the optimization variables, respectively. Once the optimal optimization variables are found, the optimal prefix code can be obtained via a polynomial time mapping [15].

In the formulation above, the objective models the overall energy consumption. The inequalities (35) and (36) model the constraints of prefix codes. The (38) is required because a leaf node can only be at exactly one level. The inequality (38) models the energy constraint at any node k while (39) models the overall average delay.

As a second example, we can also find the optimal prefix code that minimizes the average delay (bandwidth) such that the average transmission energy of each sensor does not exceed W as follow

$$\begin{aligned} &\text{minimize} \quad \sum_{k=1}^{N}\left(\frac{1}{E_0}\sum_{i=1}^{n_k}\sum_{j=1}^{m_k} p_i^k j y_{ij}^k - \frac{E_1 - E_0}{E_0}\sum_{i=1}^{n_k}\sum_{j=1}^{m_k} p_i^k x_{ij}^k\right) \\ &\text{subject to} \quad b_0^k = 1; \qquad b_j^k = 0, \quad \text{for } j < 0 \end{aligned} \tag{40}$$

For $1 \le k \le N$ and $1 \le j \le m_k$,

$$\sum_{i=1}^{n_k} y_{ij}^k + b_j^k \le b_{j-E_1}^k + b_{j-E_0}^k. \tag{41}$$

For $1 \le k \le N$ and $1 \le i \le n_k$,

$$\sum_{j=1}^{m_k} y_{ij}^k = 1. \tag{42}$$

For $1 \le k \le N$, and for $1 \le i \le n_k$

$$\sum_{i=1}^{n_k}\sum_{j=1}^{m_k} p_j^k j y_{ij}^k \le W, \tag{43}$$

where b_{ij}^k, x_{ij}^k, and y_{ij}^k are the optimization variables with b_{ij}^k taking on integer values, while x_{ij}^k and y_{ij}^k are binary variables.

As a third example, one can also replace the constraint in (43) by

$$\sum_{j=1}^{m_k} x_{ij}^k \le K \tag{44}$$

to guarantee that the number of bits "1" in any codeword cannot exceed K. This is useful when $E_0 \ll E_1$ and one wants to keep the transmission energy low even when the symbol distribution changes.

6. Performance Evaluation

We now present a few results on the tradeoff between the delay and transmission energy for a straight-line topology. These results are based on the different problem formulations which are made possible by having the expressions of delay and transmission energy (26) and (32) in Section 5. We use CPLEX, an optimization package to obtain our solutions. The running times to obtain these solutions are negligible in all our experiments.

Simulation Data. First, we assume that the data sample is of 10-bit resolution. Each bit has one-half probability of being 1. Figure 7(a) shows the average transmission delay versus K, the maximum number of bit "1" allowed in any codeword.

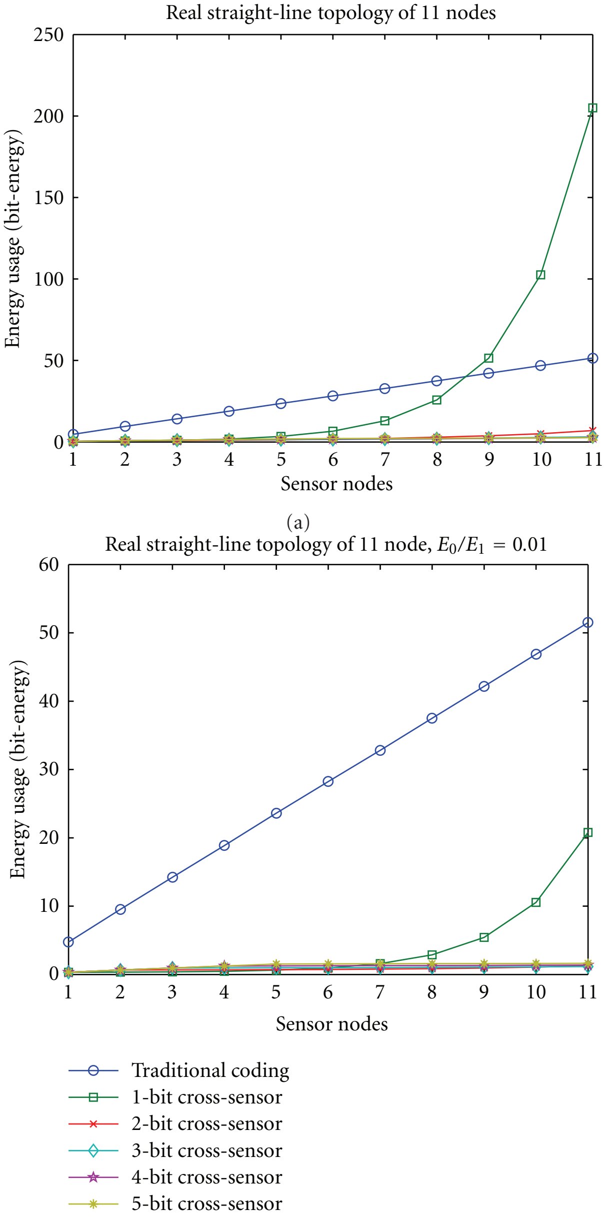


Figure 9: Per node energy usage of cross-sensor and traditional coding schemes as a function of the node number for (a) $E_0/E_1 = 0.1$ and (b) $E_0/E_1 = 0.01$.

As seen, the delay decreases with the increase of K, that is, more transmission energy is allowed. Similarly, Figure 7(b) shows the average transmission energy as a function of delay when the transmission energy is minimized subject to the delay constraint.

Figure 8 shows the average delay or code length for different values of K (the constraint on the number of bits 1 in a codeword) as a function of the number of sensors. As expected, higher K results in lower delay at the expense of energy increase.

Experimental Data. We now present the results of applying cross-sensor coding on the actual air temperature being collected by an array of sensors developed at Oregon State University. The aim of this project is to monitor the temperature and humidity levels of the forest at different elevations. These sensors are placed along a tower to continuously log the current temperatures at every 15-minute interval. The sensor network is designed to collect data in real-time, and the data is sent from the higher sensors to the lower sensors, and finally to the processing node on the ground. The measured temperature data has 10-bit resolution.

Figure 9 shows the transmission energy at each node (with node 1 being the node at the top) for different relative transmission energy levels of bits “0” and “1”. If the transmission energy is a significant part of the overall energy, then the network lifetime depends on the transmission energy of node 11. As shown in Figure 9(a), node 11 uses more energy with the traditional (binary) coding than with the cross-sensor coding for $K > 1$. This is intuitively plausible as with $K = 1$, the length of a codeword needs to be a lot longer, that is, more zeros which results in more energy usage, since transmission energy of bit “0” is not negligible, that is, 10 times less than that of bit “1”. When $K > 1$, a typical codeword resulted from cross-sensor coding is shorter, leading to lower overall transmission energy. For certain applications, one can find a suitable K-bit cross-sensor code that achieves the best performance both in terms of the delay and network lifetime.

Energy Usage and Network Lifetime. We note that our cross-sensor coding technique aims primarily at linear sensor networks for environmental monitoring applications where data is collected along the sensors arranged in a linear fashion. As such, any failed sensor will halt the data collection for the entire network. Thus, the lifetime of this type of network depends critically on the sensor with the shortest lifespan. In an ideal scenario above as shown in Figure 9, this shortest lifespan would be that of the last the sensor to the sink, specifically node 11. This is because node 11 relays the most amount of data and consumes the most amount of energy.

Scalability and Robustness. There are two major disadvantages with the proposed approach. The first issue is scalability. The cross-sensor coding approach might not be applicable to a large number of sensors since the length of the codeword increases exponentially with the number of sensors if 1-bit cross-sensor coding technique is used. It is possible to make the codeword length as a linear function of the number of sensors at the expense of using more “high” bits, resulting in higher energy consumption. At this time, we do not envision our technique to be used in future sensor networks of more than tens of sensors.

The other major disadvantage is robustness, specifically error propagation as the bits traverse through the sensors. While transmission error is out of the scope of this paper, it is an important issue in practice and deserves a discussion. In our proposed cross-sensor scheme, a bit flip from the first sensor will cause an error propagation through the downstream sensors. This is an artifact of cross-sensor coding scheme as each sensor relies on the correct bits sent by its upstream sensors to encode its data. One approach to this problem is to use special error correcting/detecting codes. Specifically, one can use error detecting/correcting codes for asymmetric channel which can detect/correct bit flips in one direction efficiently [25]. Presumably, for optical transmissions, bit “1” can turn to “0” due to attenuation, but not the other way around. In addition, weighted error correcting/detecting codes can also be used to boost up the performance [26].

7. Conclusions

We have designed and analyzed a cross-sensor energy-efficient coding technique for the ultra-low energy sensor networks using On-Off Keying. Cross-sensor coding can significantly extend the network lifetime as compared with traditional (binary) coding by solving the energy-consumption unfairness problem. We have presented the theoretical and experimental results to show that transmission energy can be reduced substantially (e.g., a factor of 15) and the unequal energy consumption among nodes can be practically eliminated.

Acknowledgment

This work is partly supported by the National Science Foundation under Grants nos. 0845476 and 0834775.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–105, 2002.

[2] J. Kahn, R. Katz, and K. Pister, “Mobile networking for smart dust,” in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, 1999.

[3] J. Rabaey, J. Ammer, J. Silva, and D. Patel, “Picoradio: ad-hoc wireless networking of ubquituous low-energy sensor/monitor nodes,” in *Proceedings of the IEEE Computer Society Annual Workshop on VLSI*, 2000.

[4] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *Proceedings of the 33rd Annual Hawaii International Conference on System Siences (HICSS ’33)*, p. 223, January 2000.

[5] A. Woo and D. E. Culler, “A transmission control scheme for media access in sensor networks,” in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pp. 221–235, July 2001.

[6] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, “On network correlated data gathering,” in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 2571–2582, March 2004.

[7] R. Cristescu, B. Beferull-Lozano, M. Vetterli, and R. Wattenhofer, “Network correlated data gathering with explicit communication: NP-completeness and algorithms,” *IEEE/ACM Transactions on Networking*, vol. 14, no. 1, pp. 41–54, 2006.

[8] R. Cristescu and B. Beferull-Lozano, “Lossy network correlated data gathering with high-resolution coding,” in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN ’05)*, pp. 218–224, April 2005.

[9] S. Lindsey, C. Raghavendra, and K. M. Sivalingam, “Data gathering algorithms in sensor networks using energy metrics,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 9, pp. 924–935, 2002.

[10] M. A. Sharaf, J. Beaver, A. Labrinidis, and P. K. Chrysanthis, “Balancing energy efficiency and quality of aggregate data in sensor networks,” *VLDB Journal*, vol. 13, no. 4, pp. 384–403, 2004.

[11] D. Slepian and J. Wolf, “Noiseless coding of correlated information sources,” *IEEE Transaction on Information Theory*, vol. 19, pp. 471–480, 1973.

[12] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley-Interscience, New York, NY, USA, 1991.

[13] J. Selker, L. Thevenaz, H. Huwald et al., “Distributed fiber-optic temperature sensing for hydrologic systems,” *Water Resources Research*, vol. 42, p. 8, 2006.

[14] M. Enachescu, A. Goel, R. Govindan, and R. Motwani, “Scale free aggregation in sensor networks,” in *Proceedings of the 1st International Workshop on Algorithmic Aspects of Wireless Sensor Networks*, 2004.

[15] R. M. Karp, “Minimum-redundancy coding for the discrete noiseless channel,” *IRE Transactions on Information Theory*, vol. 7, no. 1, pp. 27–38, 1961.

[16] P. G. Bradford, M. J. Golin, L. L. Larmore, and W. Rytter, “Optimal prefix-free codes for unequal letter costs: dynamic programming with the monge property,” in *Proceedings of the European Symposium on Algorithms*, 1998.

[17] M. J. Golin, C. Kenyon, and N. E. Young, “Huffman coding with unequal letter costs,” in *Proceedings of the 34th annual ACM symposium on Theory of computing*, 2002.

[18] C. H. Liu and H. H. Asada, “A source coding and modulation method for power saving and interference reduction in ds-cdma sensor network systems,” in *Proceedings of the American Control Conference*, 2002.

[19] J. Kim and J. Andrews, “An energy efficient source coding and modulation scheme for wireless sensor networks,” in *IEEE 6th Workshop on Signal Processing Advances in Wireless Communications*, 2005.

[20] Y. Prakash and S. K. S. Gupta, “Energy efficient source coding and modulation for wireless applications,” in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC ’03)*, 2003.

[21] Y. Zhu and R. Sivakumar, “Challenges: communication through silence in wireless sensor networks,” in *Proceedings of the Eleventh Annual International Conference on Mobile Computing and Networking (MobiCom ’05)*, pp. 140–147, September 2005.

[22] A. K. Dhulipala, C. Fragouli, and A. Orlitsky, “Silence based communication for sensor networks,” in *Proceedings of the IEEE International Symposium on Information Theory (ISIT ’06)*, pp. 212–216, July 2006.

[23] D. A. Huffman, “A method for the construction of minimum-redundancy codes,” in *Proceedings of the IRE*, 1952.

[24] W. Chen, L. Dai, K. B. Letaief, and Z. Cao, “A unified cross-layer framework for resource allocation in cooperative networks,” *IEEE Transactions on Wireless Communications*, vol. 7, no. 8, pp. 3000–3012, 2008.

[25] S. Al-Bassam and B. Bose, “Asymmetric/unidirectional error correcting and detecting codes,” *IEEE Transactions on Computers*, vol. 43, no. 5, pp. 590–597, 1994.

[26] S. Johnson, “Upper bounds for constant weight error correcting codes,” *Discrete Mathematics*, vol. 3, no. 1–3, pp. 109–124, 1972.