

# Minimizing Energy Consumptions in Wireless Sensor Networks via Two-Modal Transmission

Yao Liang  
yliang@cs.iupui.edu

Wei Peng  
pengw@iupui.edu

Department of Computer and Information Science  
Indiana University-Purdue University Indianapolis

## ABSTRACT

We present a sophisticated framework to systematically explore the temporal correlation in environmental monitoring wireless sensor networks. The presented framework optimizes lossless data compression in communications given the resource constraint of sensor nodes. The insights and analyses obtained from the framework can directly lead to innovative and better design of data gathering protocols for wireless sensor networks operated in noisy environments to dramatically reduce the energy consumptions.

## Categories and Subject Descriptors

E.4 [Coding and Information Theory]: Data compaction and compression; C.2.1 [Network Architecture and Design]: Wireless communication

## General Terms

Algorithms

## Keywords

Wireless sensor networks, energy efficiency, lossless compression

## 1. INTRODUCTION

It is anticipated that wireless sensor networks (WSNs) can fundamentally change today's practice of numerous scientific endeavors (e. g., [6]). Because wireless sensor nodes are typically battery-operated, and most environmental monitoring WSNs are deployed in harsh or even hostile environments, the replacement of batteries for sensor nodes is usually impossible. Consequently, the lifetime of WSNs depends on the power consumption of individual sensor nodes. Indeed, this severe power constraint presents one of the most critical challenges in WSNs (e. g., [3, 5]). While several approaches exist, including energy-aware routing, energy-efficient MAC protocols, adaptive sampling, and source coding, our work focuses on exploiting temporal correlation in WSN data. Among the three sensor-node operations (i. e., transmitting data, receiving data, and performing computation), transmitting and receiving are the most energy-consuming operations. For example, it was shown that about 3000 instructions could be executed for the same energy cost as sending a bit for 100 meters by radio [10], and in general, receiving has comparable energy cost as transmitting. To reduce the total energy usage at sensor nodes, one should try to

minimize nodes' transmission (and hence the corresponding receiving), probably offset by a slight increase of nodes' computing operations.

We are interested in lossless data communication in WSNs, due to the fact that in environmental monitoring tasks, particularly for new science discoveries, the accuracy of observations is often critical in understanding the underlying physical processes, because scientists may not have a priori knowledge about what observation errors are tolerable without affecting their new research findings. Hence, lossless data gathering in WSNs is essential and desirable.

However, existing studies on exploiting temporal correlation in WSNs typically focused on lossy data compression, such as the PREMON scheme [7], Lightweight Temporal Compression [12], and a few others (e. g., [11]).

In this work, we present a fundamental framework, called *two-modal transmission*, to systematically study *lossless* data compression in WSNs. Our approach exploits the principle of predictive coding [4]. In predictive coding, an error term (i. e., residue) is calculated at source node as the difference between the predicted message/signal and the actual message/signal. This error is then encoded and transmitted to the receiving node. At receiving side, with an identical predictor as the source node, the original message can be obtained by adding the received error term (decoded) to the predicted message produced at the receiving node. However, the distribution of residue signals generated at sensor node usually exhibits "long tails". A bad shape of residue distribution adversely impacts the entropy coding performance. In addition, the traditional predictive coding lacks ability to facilitate the (re)synchronization of predictors at both sensor nodes and the sink in WSNs. Unlike a recent independent work of lossless compression algorithm LEC [9], our proposed two-modal transmission is aimed to effectively overcome the above limitations of predictive coding. This presented work significantly extends and elaborates the preliminary work in [8].

## 2. ANALYTICAL FRAMEWORK

The idea of two-modal transmission is to encode only those residues which fall inside a relatively small range  $[-R, R]$  ( $R > 0$  and is called *compression radius* hereafter) by entropy coding (referred to as *compression mode*) and to transmit the original raw samples uncoded (referred to as *normal mode*) otherwise or for predictor (re)synchronization.

Let  $K$  be the size of one raw data sample in bits and  $N$  be the number of samples in a packet. The original amount of uncompressed data to be transmitted is then  $s = K \times N$  bits.

Let  $s'$  be the size of data in bits after lossless compression, the *compression ratio*  $\gamma$  is defined as

$$\gamma = (1 - \frac{s'}{s}) \times 100\% = (1 - \frac{s'}{K \times N}) \times 100\%. \quad (1)$$

Our goal is to optimize the communication performance *while meeting the constraints of the limited resources on wireless sensor nodes*. For example, the limited memory space available on sensor nodes demands that the alphabet used in entropy coding should be small.

To address the stringent resource constraints, we propose a flexible family of alphabets called *M-based alphabet system* to represent residues for entropy coding. With an *M-based alphabet*, a residue is represented using base  $M$ , where  $M$  ( $M > 1$ ) is an integer. A typical example is the decimal alphabet (i. e.,  $M = 10$ ), in which, for instance, 13 is represented as  $S(1)S(3)$ . A special symbol  $S(plh)$  named *placeholder* is introduced to flag the use of normal mode in our two-modal transmission. For instance, with the decimal alphabet and  $R = 9$ , a residue sequence of (5, 10, 6, 12, 7, ...) would produce a symbol sequence ( $S(5), S(plh), S(6), S(plh), S(7), \dots$ ) subject to entropy coding, with the original raw samples corresponding to the residues 10 and 12 attached in order after the sequence. At the sink, when encountering a  $S(plh)$ , the decoder would put the next attached original raw sample in the place of the placeholder. Thus, the size of an *M-based alphabet* ( $\{S(0), \dots, S(M-1), S(-), S(plh)\}$ ) is  $M + 2$ , regardless of  $R$ . The size of an *M-based alphabet* can be very compact when  $M \ll R$ .

Thus, we formulate the *two-modal transmission with an M-based alphabet* as the following optimization problem: Given the constraint  $M \leq M'$  for some fixed integer  $M'$  ( $M'$  specifying the resource constraints), find a combination  $(M, R)$  which maximize the compression ratio  $\gamma$ .

## 2.1 Residue Distribution

While some studies suggested using *Gaussian* distribution representing residues, our experiments on real-world data showed a much narrower residue distribution than Gaussian distribution. Hence, *Laplacian* distribution, which is more spiky than Gaussian distribution with the same variance, is adopted in this paper.

In the following discussion,  $f$  stands for the *cumulative distribution function* of a Laplacian distribution with zero mean. We have

$$f(x) = \frac{1}{2}(1 + \text{sgn}(x)(1 - \exp(\frac{-|x|}{b}))), \quad (2)$$

in which  $b = \frac{\sqrt{2}}{2}\sigma$  and  $\sigma$  is the standard deviation.

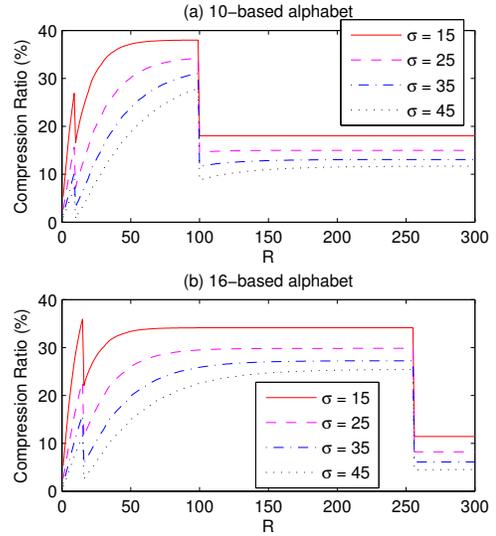
## 2.2 Entropy Computation

In information theory, *entropy* is defined as

$$H = - \sum_{s \in S} P(s) \log(P(s)), \quad (3)$$

in which  $S$  is the alphabet and  $P(s)$  is the probability model for  $S$ . The significance of entropy is that it is the *lower bound* of average bits per symbol for *any* lossless compression algorithm, assuming that occurrence of symbol in the sequence is independent.

The exact representation of a residue  $r$  in an *M-based alphabet*  $\text{rep}(r)$  depends on the choice of compression radius



**Figure 1: Compression ratio for Laplacian distribution with zero mean under different standard deviation  $\sigma$ . (a) 10-based alphabet; (b) 16-based alphabet.**

$R$ . Let  $l$  denote the number of symbols for the numerical part (excluding the possible sign symbol  $S(-)$ ) of the representation. We have

$$l = \lfloor \log_M R \rfloor + 1. \quad (4)$$

Thus,  $\text{rep}(r)$  for a residue  $r$  has length of  $l$  when  $r \geq 0$  and length of  $l + 1$  when  $r < 0$ . For example, if  $M = 10$  and  $R = 20$ , then  $l = \lfloor \log_{10} 20 \rfloor + 1 = 2$ , and  $\text{rep}(5) = S(0)S(5)$ ,  $\text{rep}(0) = S(0)S(0)$ , and  $\text{rep}(-1) = S(-)S(0)S(1)$ .

Next, we compute the expected number of occurrences of symbol  $s$  due to residue  $r$  as follows,

$$c_s^r = (f(r + 0.5) - f(r - 0.5))N_s(\text{rep}(r)), \quad (5)$$

in which  $N_s(q)$  is the number of occurrence of the symbol  $s$  in a symbol sequence  $q$ . For example,  $N_{S(1)}(S(-)S(1)S(1)) = 2$  while  $N_{S(-)}(S(-)S(1)S(1)) = 1$ .

By Equation (5), the expected number of occurrences of symbol  $s \in \{S(0), \dots, S(M-1), S(-)\}$  can be obtained by

$$E(s) = \sum_{r=-R}^R c_s^r. \quad (6)$$

The expected number of occurrences of symbol  $S(plh)$  is

$$\begin{aligned} E(S(plh)) &= 1 - \sum_{i=-R}^R (f(i + 0.5) - f(i - 0.5)) \\ &= 1 - (f(R + 0.5) - f(-R - 0.5)). \end{aligned} \quad (7)$$

Hence, given a residue distribution  $f(r)$ , the alphabet  $S = \{S(0), \dots, S(M-1), S(-), S(plh)\}$  and  $R$ , we have the following probability model for entropy computation:

$$P(s) = \frac{E(s)}{\sum_{s \in S} E(s)}. \quad (8)$$

## 2.3 Compression Ratio

To establish the performance envelope of our approach, we assume the availability of an ideal encoder which can generate the perfect lossless compression indicated by entropy. To compute  $\gamma$  by Equation (1), we need to first obtain the after-compression size  $s'$ . Note that  $s'$  can be divided into the following three components:

- Size of compressed data (i. e., those whose residues falling *inside*  $[-R, R]$ )  $s_{\text{dat}}$ ;
- Size of overhead (i. e., placeholders)  $s_{\text{ovh}}$ ;
- Size of data transmitted as original raw samples (i. e., those whose residues falling *outside*  $[-R, R]$ )  $s_{\text{ori}}$ .

Thus, we have

$$s' = s_{\text{dat}} + s_{\text{ovh}} + s_{\text{ori}}. \quad (9)$$

Let  $N$  be the number of data samples in a packet to be transmitted (as in Equation (1)). We have the following:

- The number of *negative* compressed data is  $N_{<0} = N \times \sum_{r=-R}^{-1} (f(r+0.5) - f(r-0.5)) = N \times (f(-0.5) - f(-R-0.5))$ ;
- The number of *nonnegative* compressed data is  $N_{\geq 0} = N \times \sum_{r=0}^R (f(r+0.5) - f(r-0.5)) = N \times (f(R+0.5) - f(-0.5))$ ;
- The number of overhead *symbol* is  $N_{\text{ovh}} = N \times (1 - (f(R+0.5) - f(-R-0.5)))$ ;
- The number of uncompressed data is  $N_{\text{ori}} = N \times (1 - (f(R+0.5) - f(-R-0.5)))$ .

As mentioned before, nonnegative residues are represented by  $l$  symbols and negative residues are represented by  $l+1$  symbols; A placeholder requires only 1 symbol. Hence, by Equations (4) and (3), we have

$$s_{\text{dat}} = H \times l \times N_{\geq 0} + H \times (l+1) \times N_{<0}, \quad (10)$$

$$s_{\text{ovh}} = H \times 1 \times N_{\text{ovh}}, \quad (11)$$

$$s_{\text{ori}} = K \times N_{\text{ori}}. \quad (12)$$

By Equations (9), (10), (11) and (12), we have

$$s' = H \times l \times N_{\geq 0} + H \times (l+1) \times N_{<0} + H \times 1 \times N_{\text{ovh}} + K \times N_{\text{ori}}. \quad (13)$$

In Equation (13),  $l$  and  $H$  depend on both  $M$  and  $R$ ;  $N_{\geq 0}$ ,  $N_{<0}$ ,  $N_{\text{ovh}}$  and  $N_{\text{ori}}$  depend on  $R$ . Hence,  $s'$  can be seen as a function  $s'(M, R)$  over  $M$  and  $R$ . Thus, by Equation (1), we get  $\gamma$  as a function  $\gamma(M, R)$  over  $M$  and  $R$ . With Equations (1) and (13), we can now compute the theoretical compression ratio  $\gamma(M, R)$  of our two-modal transmission approach for any residue distribution with zero mean.

For illustration, Figure 1 depicts  $\gamma(M, R)$  for  $M = 10$  in (a) and  $M = 16$  in (b), respectively, for a group of different values of  $\sigma$ . We note the following important points.

- All curves exhibit sudden drops at  $R = 10$  and  $R = 100$  in (a) and at  $R = 16$  and  $R = 256$  in (b), respectively. The drops are a direct result of the residue representation adopted in the  $M$ -based alphabet system. For example, for 16-based alphabet, while the numerical part of each residue can be represented by 1 symbol when  $R = 15$ , it requires 2 symbols to be represented when  $16 \leq R \leq 255$ , and so on. This has a direct impact on the length of symbol sequence to be encoded and hence on the size after compression.

- For a Laplacian distribution with zero mean, almost all (98.56%) samples fall inside the range  $[-3\sigma, 3\sigma]$ . For example, when  $\sigma = 15$  (the red curves in (a) and (b)), almost all residues fall inside  $[-45, 45]$ . This corresponds to the rapid growth in compression gain (with the exception of those sudden drop points) when  $R \leq 45$  and gradual flattening out when  $R > 45$ .
- Both (a) and (b) indicate a reciprocal relation between compression ratio and the standard deviation of residue distribution. This can be understood by realizing that larger standard deviation corresponds to a “flatter” distribution. This in turn would result in a less skewed probability statistics among symbols and hence a larger entropy.

A key observation is that the drops of compression ratio  $\gamma$  for the two-modal transmission can *only* occur at points  $R = M^k$  ( $k$  is an integer and  $k \geq 1$ ). On each segment of  $R$  divided by those  $M^k$  (i. e.,  $[0, M-1]$ , and  $[M^k, M^{k+1}-1]$  ( $k \geq 1$ )),  $\gamma$  is piecewisely monotonically increasing. This is because that each segment of  $R$  induces the same number of symbols to represent the numerical part (e. g.,  $[0, 9]$ ,  $[10, 99]$  for base 10) and thus *larger  $R$  in the same segment* would produce better  $\gamma$  (or at least as good when  $R \gg 3\sigma$ ). This leads to the following important assertion: For a given  $M$ , there exists a global maximal  $\gamma(M, R)$  (denoted by  $\gamma_{\text{max}}(M)$ ) over the whole range of  $R$ . Moreover, all *local* maxima of  $\gamma(M, R)$  are obtained at  $R = M^k - 1$  ( $k$  is an integer and  $1 \leq k \leq \lfloor \log_M 2^K \rfloor$ ) and at the boundary point  $R = 2^K - 1$ .

Based on this structure of local maxima, instead of searching over every combination of  $(M, R)$  for the maximal  $\gamma$  given  $M'$ , we only need to check those local maxima  $\gamma(M, R)$  to obtain all  $\gamma_{\text{max}}(M)$  for  $M \leq M'$  and then select the maximal  $\gamma_{\text{max}}$  among  $\gamma_{\text{max}}(M)$ . This greatly reduces the size of searching space.

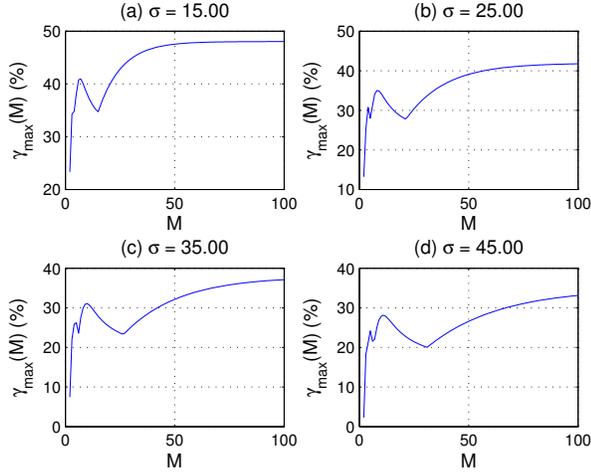
## 3. HEURISTIC METHOD

We now study  $\gamma_{\text{max}}(M)$  to further simplify the computation of the overall maximal  $\gamma(M, R)$  and its corresponding  $M$ , given constraint  $M'$ . Figure 2 illustrates the relationship between  $M$  and  $\gamma_{\text{max}}(M)$ .

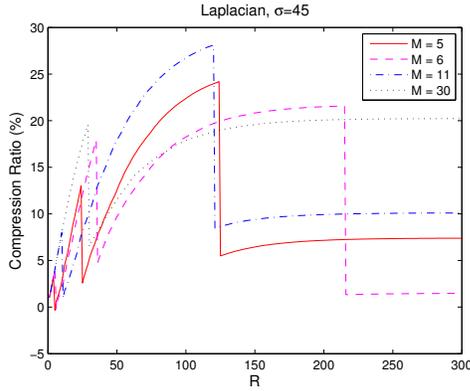
Note that before certain point, greater  $M$  (hence a larger alphabet) does not always produce a better compression ratio. For example, when  $\sigma = 45$ , in the range of 2 to 40, the  $M$  with maximal  $\gamma_{\text{max}}(M)$  is 11 rather than either 2 or 40.

Figure 3 is given to help in understanding this non-monotonic relationship.  $M$  corresponding to local minima and maxima in Figure 2 (d) ( $\sigma = 45$ ) are chosen. In particular, almost all residues fall inside  $[-135, 135]$  (i. e.,  $[-3\sigma, 3\sigma]$ ) in this very case. The first local maximum ( $M = 5$ ) corresponds to the  $M$  which has its cubic ( $M^3 = 125$ ) closest to 135. This means that the  $R$  corresponds to  $\gamma_{\text{max}}(M)$  is  $M^3 - 1 = 125 - 1 = 124$  and that the 3-symbol representation of the numerical part is fully used. In contrast, at the first local minimum ( $M = 6$ ), the representation is seriously wasted ( $M^3 = 216$  is much larger than 135). In the same way, the second local maximum ( $M = 11$ ) makes full use of the 2-symbol ( $M^2 = 121$ ) representation while the second local minimum ( $M = 30$ ) does not.

Thus, we propose the following heuristic for choosing an  $M$  from a given range to maximize  $\gamma_{\text{max}}(M)$ , subject to  $M \leq M'$  for some given integer  $M'$ , by making use of the



**Figure 2:**  $\gamma_{max}(M)$  for different  $M$ . Residues conform to a Laplacian distribution with zero mean and standard deviation  $\sigma$  of (a) 15; (b) 25; (c) 35; (d) 45.



**Figure 3:** Compression ratio for Laplacian distribution with zero mean and standard deviation  $\sigma = 45$ .  $M$  corresponding to local minima and maxima in Figure 2 (d) are chosen (5, 6, 11 and 30).

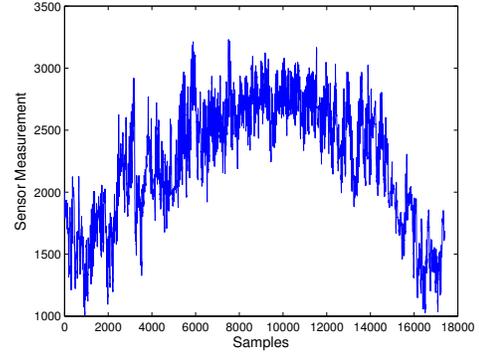
fact that almost all (98.56%) residues fall inside  $[-3\sigma, 3\sigma]$  for a Laplacian distribution with zero mean.

1.  $k \leftarrow \lceil \log_{M'} 3\sigma \rceil$ . This ensures  $M'^{k-1} \leq 3\sigma \leq M'^k$ ;
2.  $L \leftarrow \lfloor \sqrt[k]{3\sigma} \rfloor$ ;
3. Obtain  $\gamma_{max}(L) = \gamma(L, L^k - 1)$ ;
4. Obtain  $\gamma_{max}(L+1) = \gamma(L+1, (L+1)^k - 1)$ ;
5. Let  $\gamma_{max|_{M \leq M'}} = \max\{\gamma_{max}(L), \gamma_{max}(L+1)\}$ .  $M \leftarrow L$  if  $\gamma_{max|_{M \leq M'}} = \gamma_{max}(L)$  and  $M \leftarrow L+1$  otherwise.

As an illustration, in the  $\sigma = 45$  case, suppose we require  $M' = 30$ .  $k = \lceil \log_{30} 3\sigma \rceil = \lceil \log_{30} 135 \rceil = 2$ .  $L = \lfloor \sqrt[2]{135} \rfloor = 11$ . Then we compare  $\gamma(11, 120)$  and  $\gamma(12, 143)$  and find that  $\gamma(11, 120)$  is larger. Thus, we let  $M = 11$ , which is indeed the optimal choice of  $M$  for all  $M \leq M' = 30$ .

## 4. EMPIRICAL STUDY

We use a set of one-year real-world publicly-accessible environmental monitoring data[2], collected in year 2000, to



**Figure 4:** Sensor measurement for the temperature data set.

illustrate the effectiveness of the two-modal transmission scheme. These samples were taken at an interval of 30 minutes. To simulate real-world sensor communications with fidelity, we first converted the *physical* data into *sensor measurement*, with the assumption of the A/D conversion precision being 12 bits (i. e.,  $K = 12$ ).

Figure 4 shows the sensor measurement for the temperature data set.

### 4.1 Predictor

Linear predictor was chosen both for its simplicity. In particular, we used *second-order linear predictor*. For a sample  $x(k)$  ( $k > 2$ ), its prediction  $\hat{x}(k)$  is computed as

$$\hat{x}(k) = c_1 x(k-1) + c_2 x(k-2) + c_0, \quad (14)$$

where coefficients  $c_1$ ,  $c_2$  and bias term  $c_0$  were determined by minimizing the mean square error (MSE) of predictor on a training data set as

$$\text{MSE} = \frac{1}{m} \sum_{k=1}^m (x(k) - \hat{x}(k))^2 = \frac{1}{m} \sum_{k=1}^m r(k)^2, \quad (15)$$

in which  $m$  is the number of training samples.  $r(k) = x(k) - \hat{x}(k)$  is called *residue*, or *prediction error*.

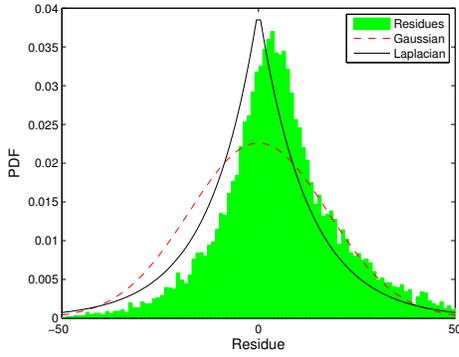
In our experiments, the first 15 days' data were used in predictor training, and another 15 days' data were used in generating symbol statistics for entropy coding. The rest of 11 month data were then used for testing.

Figure 5 shows the residue distribution generated by the predictor in our simulation. Note that  $\sigma = 17.65$  and there was a small deviation of the peak from zero. Laplacian and Gaussian distributions with the same variance and mean of zero are drawn in Figure 5 for comparison.

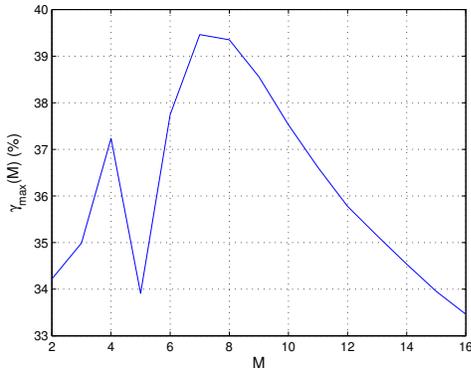
### 4.2 Simulation Results

Arithmetic Coding (AC) algorithm was chosen as our entropy coding scheme due to its near-optimal performance. Because of the daily periodicity of the physical data, the size of data payload per packet is chosen as  $48 \times 12 = 576$  bits (72 bytes).

In our simulation,  $M' = 16$  was chosen as sensor nodes' resource constraint. Solving this constrained optimization problem based on our formal approach, we obtained those local maxima for all  $M \leq 16$ . The results are shown in Figure 6 in terms of  $\gamma_{max}(M)$ . On the other hand, based



**Figure 5: Residue distribution.** Laplacian and Gaussian distribution with same  $\sigma$  ( $\sigma = 17.65$ ) are drawn for comparison.



**Figure 6:**  $\gamma_{max}(M)$  for different  $M$  of the temperature data set.

on our heuristic method, we quickly got  $k = 2$ , and either 7 or 8 being the optimal  $M$  because  $\lfloor \log_2 3\sigma \rfloor + 1 = 2$  and  $7^2 \leq 3\sigma \leq 8^2$ . As clearly shown in Figure 6,  $M = 7$  was the best among all  $M \leq M'$  in terms of  $\gamma_{max}(M)$  achieved.

Moreover, a comparison between theoretical and empirical results is shown in Figure 7 in which  $M = 7$ . We can see that the performance derived from our theoretical framework matches the empirical results obtained using real-world data set favorably.

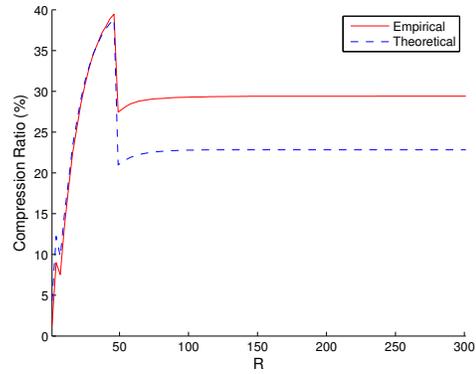
## 5. ENERGY CONSUMPTION MODEL

### 5.1 Radio Transceiver Model

Let  $I_{TX}$  and  $I_{RX}$  be the current draw of sending and receiving by the radio respectively;  $T_{TX}$  and  $T_{RX}$  be the corresponding operating time over 1 byte;  $V$  be the voltage supply, which we assume to be constant throughout the transmission. Therefore, transmitting  $k$  bytes per hop requires  $E_{radio}(k) = kI_{TX}VT_{TX} + kI_{RX}VT_{RX}$ .

### 5.2 Algorithm Complexity Model

A simplified energy consumption model on source sensor nodes for the proposed two-model transmission can be obtained by counting the number of basic operations (e. g., shifts, additions) conducted in the following outlined logic.



**Figure 7: Empirical versus theoretical (with same standard deviation) compression ratio in 7-based alphabet.**

It is worth mentioning that the optimal  $M$  and  $R$  in our scheme as well as predictor training are computed at the WSN sink which is not energy-limited. A source sensor node receives its symbol table,  $R$ , and predictor coefficients from the sink after the initial training and validation phases, during which samples could be transmitted uncompressed.

```

if (not (predictor_resynchronization))
    pred = predictor();
    residue = x - pred;
else
    residue = R+1;
if ((residue <= R) and (residue >= -R))
    encode_and_send(residue);
else
    encode_and_send(placeholder);
    send(x);

```

Since `predictor` is simply a linear predictor, the computation cost mainly rests on the entropy encoder adopted.

#### 5.2.1 AC encoder

AC encoder was employed. Based on the implementation of AC encoder described in [1], except for the interval updating and checking (4 additions, 2 integer multiplications, 2 shifts, 2 comparisons) performed on *every* symbol fed into the encoder, the coding cost depends on the number of  $E1$ ,  $E2$  and  $E3$  scalings performed [1]. These scalings have a uniform cost of 3 additions and 2 shifts. Let  $P(S)$  be the probability of a symbol  $S$ ,  $N(S) = \lceil -\log_2 P(S) \rceil$  is an adequate estimate for the number of scalings once  $S$  is encountered.

For the optimal parameter setting ( $M = 7$  and  $R = 48$ ) in our empirical study, the frequency of the symbols is summarized in table 1. From these statistics, we can estimate the number of scalings for each symbol. For instance, we estimate the number of scalings for  $S(6)$  to be  $\lceil -\log_2(\frac{105}{1779}) \rceil \times 2175 = 10875$ . Hence, we estimate the total number of scalings being 124049 by adding up the estimates for each symbol.

Thus, in our empirical study, the energy consumption for AC coding process on the entire testing data can be obtained by  $E_{AC} = 36406 \times (4\epsilon_{add} + 2\epsilon_{mul}) + 2\epsilon_{sht} + 2\epsilon_{cmp} + 124049 \times (3\epsilon_{add} + 2\epsilon_{sht}) = 517771\epsilon_{add} + 72812\epsilon_{mul} + 320910\epsilon_{sht} +$

**Table 1: Frequency of Symbols for 15 Days' Validation and 11 Months' Testing Data for  $M = 7$  and  $R = 48$** 

	$S(0)$	$S(1)$	$S(2)$	$S(3)$	$S(4)$	$S(5)$	$S(6)$	$S(-)$	$S(plh)$	Total
Valid.	388	317	219	153	118	104	105	359	16	1779
Test	8274	6912	4818	3605	2956	2432	2175	4862	372	36406

**Table 2: Parameters of the Energy Models**

Parameter	Value	Parameter	Value
$I_{TX}$	17.4 mA	$f_{cpu}$	48 MHz
$I_{RX}$	19.7 mA	$\epsilon_{add}$	2.13 nJ
$T_{TX}$	$3.2 \times 10^{-5}$ s	$\epsilon_{mul}$	6.39 nJ
$T_{RX}$	$3.2 \times 10^{-5}$ s	$\epsilon_{cmp}$	2.13 nJ
$V$	3.3 V	$\epsilon_{sht}$	4.26 nJ
$I_{cpu}$	31 mA		

$72812\epsilon_{cmp}$ , with  $\epsilon_{add}$ ,  $\epsilon_{mul}$ ,  $\epsilon_{cmp}$  and  $\epsilon_{sht}$  being the energy consumption of addition, multiplication, comparison and shift instruction respectively.

### 5.2.2 Two-Modal Transmission

In our empirical study, the total number of 11 months' testing samples was 15958. Apart from AC coding, the rest computing of the two-modal transmission including the second order linear predictor operations requires at most 4 additions, 2 multiplications and 3 comparisons per sample. Thus, the total energy consumption for this part is simply  $15958 * (4\epsilon_{add} + 2\epsilon_{mul} + 3\epsilon_{cmp}) = 63832\epsilon_{add} + 31916\epsilon_{mul} + 47874\epsilon_{cmp}$ . Adding this part with  $E_{AC}$  above, we have an estimation of the total computation energy consumption  $E_{comp} = 581603\epsilon_{add} + 104728\epsilon_{mul} + 320910\epsilon_{sht} + 120686\epsilon_{cmp}$ .

## 5.3 Numerical Result

Consider the widely used CC2420 radio transceiver and ARM7TDMI microprocessor of motes. Table 2 summarizes the parameters obtained based on their data sheet and reported experience.

We first evaluated the consumed energy (per hop) by transmitting all the testing data uncompressed (the baseline scenario). The energy was  $E_{baseline} = E_{radio}(15958 \times 1.5) = 23937I_{TX}VT_{TX} + 23937I_{RX}VT_{RX} = 93.7$  mJ.

In the two-modal transmission, total 14417 bytes were transmitted in the simulation. So the total energy consumption of per-hop transmission is  $E_{radio,2m} = E_{radio}(14417) = 14417I_{TX}VT_{TX} + 14417I_{RX}VT_{RX} = 56.5$  mJ.

The total energy consumed by mote computation in our approach was  $E_{comp} = 581603\epsilon_{add} + 104728\epsilon_{mul} + 320910\epsilon_{sht} + 120686\epsilon_{cmp} = 3.5$  mJ.

Therefore, for the total testing data, the energy saved by our approach in an  $n$ -hop WSN is  $n \times (E_{baseline} - E_{radio,2m}) - E_{comp} = (37.2n - 3.5)$  mJ. Clearly, the larger  $n$  of WSN, the higher energy gain. Even in a 1-hop WSN, the computation energy cost is less than 1/10 of the communication energy gain, and the pure saving would be 33.7 mJ per hop. This energy saving is sufficient to transmit extra 8594 bytes over a wireless link.

## 6. CONCLUSIONS

A major contribution of this paper is the rigorous development of a sophisticated formulation of the two-modal transmission as a constrained optimization problem  $\gamma(M, R)$

subject to  $M \leq M'$  ( $M'$  representing resource constraint). We then study the structure of local maxima in the solution space and enable significantly reducing the size of searching space for the optimal solution. Furthermore, we propose a simple heuristic method to solve the constrained optimization easily. Empirical study with real-world data and energy consumption analysis are provided. The implementation of the two-modal transmission is ongoing. Our future work includes developing an adaptive framework to address the nonstationary characteristic of sensed data in WSNs.

## Acknowledgment

This work is supported by National Science Foundation under grant CNS-0758372. We are grateful to Dr. Kenneth J. Davis for his permission of using their collected real-world sensed data in this work.

## 7. REFERENCES

- [1] E. Bodden et al. Arithmetic coding revealed - a guided tour from theory to praxis. (SABLE-TR-2007-5), 2007.
- [2] K. Davis et al. [http://cheas.psu.edu/data/flux/wcreek/wcreek2000\\_met.txt](http://cheas.psu.edu/data/flux/wcreek/wcreek2000_met.txt).
- [3] L. Doherty et al. Energy and performance considerations for smart dust. *Int. J. of Paral. and Distr. Sys.*, 2001.
- [4] P. Elias. Predictive coding-i and ii. *Info. Theory, IRE Trans.*, 1(1):24-33, March 1955.
- [5] A. Ephremides. Energy concerns in wireless networks. *Wireless Comm., IEEE*, 9(4):48-59, Aug. 2002.
- [6] D. Estrin et al. Connecting the physical world with pervasive networks. *Per. Comput., IEEE*, 1(1):59-69, Jan 2002.
- [7] S. Goel and T. Imielinski. Prediction-based monitoring in sensor networks: taking lessons from mpeg. *Comput. Commun. Rev.*, 31(5):82-98, 2001.
- [8] F. Huang and Y. Liang. Towards energy optimization in environmental wireless sensor networks for lossless and reliable data gathering. In *Proc. IEEE MASS*, pages 1-6, Oct. 2007.
- [9] F. Marcelloni and M. Vecchio. An Efficient Lossless Compression Algorithm for Tiny Nodes of Monitoring Wireless Sensor Networks. *The Comp. J.*, 2009.
- [10] G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. *Commun. ACM*, 43(5):51-58, 2000.
- [11] I. Ragoler et al. Adaptive probing and communication in sensor networks. In *Proc. of 3rd Int. Conf. on Ad-Hoc Networks and Wireless*, July 2004.
- [12] T. Schoellhammer et al. Lightweight temporal compression of microclimate datasets. *Local Computer Networks, Annual IEEE Conf.*, 0:516-524, 2004.