An Efficient and Robust Data Compression Algorithm in Wireless Sensor Networks

Yao Liang, *Senior Member, IEEE,* and Yimei Li

***Abstract*—Data compression is a useful technique in the deployments of resource-constrained wireless sensor networks (WSNs) for energy conservation. In this letter, we present a new lossless data compression algorithm in WSNs. Compared to existing WSN data compression algorithms, our proposed algorithm is not only efficient but also highly robust for diverse WSN data sets with very different characteristics. Using various real-world WSN data sets, we show that the proposed algorithm significantly outperforms existing popular lossless compression algorithms for WSNs such as LEC and S-LZW. The robustness of our algorithm has been demonstrated, and the insight is provided. The energy consumption of our devised algorithm is also analyzed.**

***Index Terms*—Wireless sensor networks, data compression, energy efficiency, robustness.**

I. Introduction

Wireless sensor networks (WSNs) are being increasingly deployed for enabling continuous monitoring and sensing of physical variables of our world. Energy efficiency is of paramount importance in the design and deployment of wireless sensor networks, as WSN nodes are typically battery-powered, and in many real physical environments the replacement of batteries for nodes is either difficult or virtually impossible. In general, radio transmissions and receptions are most power consuming compared to the energy consumption of node microcontroller and memory in WSNs. Because sensing data usually exhibit strong temporal correlations, data compression has been considered as a useful approach to reducing power consumptions of WSN nodes. Recent advances in lossless temporal compression include compression algorithms such as Sensor LZW (S-LZW) [1] and Lossless Entropy Compression (LEC) [2], [3], and compression framework such as Two-Modal Transmission (TMT) [4], [5], in which various entropy coding algorithms can be employed.

A. State of the art

We focus on lossless compression algorithms. S-LZW [1] is a lightweight data compression algorithm for resource-constrained WSNs, which modified the well-known dictionary-based lossless compression Lempel-Ziv-Welch (LZW) algorithm [6]. On the other hand, LEC [2], [3] is based on predictive coding [7], [8], in which a predictor and an encoder are employed. For a new data item x_i in a series, $\hat{x}_i$ is generated by a given predictor, and residue $r_i = x_i - \hat{x}_i$ is calculated. This residue r_i is coded, and then transmitted to the receiving node. In LEC, the simple and popular differential predictor is adopted, that is, $\hat{x}_i = x_{i-1}$; the contribution of LEC is its encoder which is a modified version of the Exponential-Golomb code (Exp-Golomb) of order 0 [9].

While LEC demonstrated significantly better compression performance in comparison with S-LZW in [3], one critical challenge of LEC and other practical data compression algorithms is the *robustness*, which indicates whether or not a data compression algorithm can be widely and effectively applied for diverse real-world sensor data in various WSN applications, for different sensor data can exhibit very different temporal characteristics. This motivated our work. The differential predictor in LEC is generic without any training/leaning when applied to any sensed data streams, making LEC is simple to use and scalable for large-size WSNs. However, such a generic and non-adaptive predictor cannot effectively exploit temporal correlations for diverse WSN data streams for various WSN applications, because different sensed data streams exhibit different temporal correlation patterns. As a result, while LEC may show good compression performance for some sensed data streams in a WSN, it may also produce poor performance for other data streams in other WSNs. In other words, LEC suffers from its lack of robustness.

B. Contributions

In this letter, we present a novel, efficient and robust lossless data compression algorithm which is able to significantly improve temporal lossless compression performance in WSNs for data collections. Our devised algorithm extends the LEC to address its weakness of the lack of robustness, and hence is able to achieve highly robust compression performance for various sensor data streams. At the same time, our algorithm is sufficiently simple to enable its energy-efficient implementation and execution on resource-constrained WSN nodes. Our algorithm, referred to as *Sequential* Lossless Entropy Compression (S-LEC), is proposed in Sec. II. We provide careful and rigorous evaluation of our algorithm in comparison with recent S-LZW and LEC compression algorithms using diverse real-world WSN data sets in Sec. III. Finally we present energy analysis of our devised algorithm versus LEC in Sec. IV to validate our algorithm.

II. Algorithm

In LEC, residues in residue series $\{r_i\}$ ($i = 1, 2, 3, \ldots, M$, where M is the size of data block), being coded by the entropy encoder, are considered having no correlation among them,

Manuscript received October 19, 2013. The associate editor coordinating the review of this letter and approving it for publication was D. Qiao.

This work is supported in part by the U.S. National Science Foundation under grant CNS-1252066.

The authors are with the Department of Computer and Information Science, Indiana University Purdue University, Indianapolis, IN 46202 USA (e-mail: yliang@cs.iupui.edu, liyim@imail.iu.edu).

Digital Object Identifier 10.1109/LCOMM.2014.011214.132319

1089-7798/14$31.00 © 2014 IEEE

TABLE I
LEC CODING TABLE FOR K=24

n_i	$h(n_i)$	r_i
0	00	0
1	010	± 1
2	011	$\pm 2, \pm 3$
3	100	$\pm 4, \cdots \pm 7$
4	101	$\pm 8, \cdots \pm 15$
5	110	$\pm 16, \cdots \pm 31$
6	1110	$\pm 32, \cdots \pm 63$
7	11110	$\pm 64, \cdots \pm 127$
8	111110	$\pm 128, \cdots \pm 255$
9	1111110	$\pm 256, \cdots \pm 511$
10	11111110	$\pm 512, \cdots \pm 1023$
11	111111110	$\pm 1024, \cdots \pm 2047$
12	1111111110	$\pm 2048, \cdots \pm 4095$
13	11111111110	$\pm 4096, \cdots \pm 8191$
14	111111111110	$\pm 8192, \cdots \pm 16383$
15	1111111111110	$\pm 16384, \cdots \pm 32767$
16	11111111111110	$\pm 32768, \cdots \pm 65535$
17	111111111111110	$\pm 65536, \cdots \pm 131071$
18	1111111111111110	$\pm 131072, \cdots \pm 262143$
19	11111111111111110	$\pm 262144, \cdots \pm 524287$
20	111111111111111110	$\pm 524288, \cdots \pm 1048575$
21	1111111111111111110	$\pm 1048576, \cdots \pm 2097151$
22	11111111111111111110	$\pm 2097152, \cdots \pm 4194303$
23	111111111111111111110	$\pm 4194304, \cdots \pm 8388607$
24	1111111111111111111110	$\pm 8388608, \cdots \pm 16777215$

TABLE II
SEQUENTIAL CODING IN S-LEC

s_i	Context Information	
00	$h_i = h_{i-1}$;	same group
01	$h_i = \begin{cases} h(n_{i-1}-1), & n_{i-1} \geq 1 \\ h(2), & n_{i-1} = 0 \end{cases}$;	neighboring group
10	$h_i = \begin{cases} h(n_{i-1}+1), & n_{i-1} < K \\ h(K-2), & n_{i-1} = K \end{cases}$;	neighboring group
11	h_i cannot be omitted in codeword, and $s_i \vert h_i \vert a_i$ is required;	otherwise

TABLE III
GROUP CODE REDUCTION

r_{i-1}	Reduced h_i when $s_i = 11$ and $n_i > n_{i-1}$	Meaning
C_1	1-reduced h_i	Remove the first '1' from LEC h_i, e.g., 1110 reduced to 110
C_2	11-reduced h_i	Remove the first two '1's from LEC h_i, e.g., 11110 reduced to 110
C_3	111-reduced h_i	Remove the first three '1's from LEC h_i, e.g., 111110 reduced to 110

and thus are encoded independently. Our insight is that since the simple differential predictor is generic for any sensed data streams, it cannot completely capture and remove the temporal correlations among an arbitrary sensor data sequence. Based on the implicit assumption of residue independence, LEC would perform reasonably if the correlation characteristic of a sensor data stream is roughly captured by the differential predictor, but would perform poorly otherwise. To address this issue, we introduce a novel idea of sequential coding and extend LEC. In the following, before we present our S-LEC algorithm, we first briefly overview LEC.

A. Overview of LEC

As a modified version of the Exponential-Golomb code, LEC organizes the alphabet of integer residues obtained from differential predictor into groups which have exponentially increased sizes [2], [3]. Generally, an LEC codeword consists of two parts: the entropy code specifying the group and the binary code representing the index in the group. Assume that any sensor reading x_i is represented in K bits, $K+1$ groups are to be formed. When residue r_i is not zero, it is represented as $h_i|a_i$, where $h_i = h(n_i)$ encodes the group number $n_i (n_i = \lfloor \log_2 |r_i| \rfloor + 1)$ and a_i represents the following computed index within the group as binary code:

$$index = \begin{cases} r_i, & r_i > 0 \\ 2^{n_i} - |r_i| - 1, & r_i < 0 \end{cases}.$$

When residue r_i equals to zero, then the corresponding group number $n_i = 0$ and also no index (i.e., no a_i code) is needed. An LEC coding table, for $K = 24$ for example, is shown in Table I.

B. S-LEC Algorithm

To address the robustness issue of the original LEC approach, we devise Sequential LEC (S-LEC) algorithm to exploit valuable sequential context information among adjacent residues for WSN data compression. S-LEC introduces an additional sequential code s_i into its codeword, and therefore extends the LEC codeword of r_i from $h_i|a_i$ into $s_i|h_i|a_i$ in general. The fundamental idea is that if a subsequent residue r_{i+1} belongs to the same or a neighboring group as residue r_i's, the group code h_{i+1} for r_{i+1} can be *inferred* from the previous h_i and thus *omitted* in its codeword; that is, r_{i+1}'s codeword is $s_{i+1}|a_{i+1}$ instead of $s_{i+1}|h_{i+1}|a_{i+1}$, reducing the size of codeword based on the adjacent context information among residues. To code such sequential context information, two bits of s_i are devised in our S-LEC algorithm and specified in Table II. We note that s_1, for the first residue r_1 of a sensed data block, will be always omitted, and hence h_1 has to be present.

In the case of $s_i = 11$, while group code h_i cannot be omitted completely, it can be reduced in its size in some context situations without any ambiguity. To investigate this possibility, let all groups of LEC alphabet be further grouped into three clusters as follows: $C_1 = \{n_i | i = 0, 1, 2, 3\}$, $C_2 = \{n_i | i = 4, 5\}$, and $C_3 = \{n_i | i = 6, \ldots, K\}$, where K is the number of bits of a sensor reading, that is, the precision of A/D converter (ADC). Then we give h_i reduction rule based on $r_{i-1} \in C_j (j = 1, 2, 3)$ in Table III when $s_i = 11$ and $n_i > n_{i-1}$.

To illustrate, for example, let us consider a series of residues $r_1 = 9$, $r_2 = 128$, $r_3 = -130$, $r_4 = 16$, and $r_5 = -32$, which will be encoded by S-LEC as 101|1001, 11|1110|10000000, 00|01111101, 11|110|10000, 10|011111. In contrast, the same residue series would be encoded by LEC as 101|1001, 111110|10000000, 111110|01111101, 110|10000, 1110|011111.

III. PERFORMANCE EVALUATION

To evaluate the proposed S-LEC algorithm, we compare its lossless compression performance with LEC algorithm and S-LZW algorithm, as LEC and S-LZW represent two different popular algorithms in lossless compression. We use real-world diverse WSN datasets from SensorScope [10] and volcanic monitoring [11], [12] in our evaluation. The compression performance is usually evaluated by compression ratio defined as follows:

$$CR = (1 - \frac{z'}{z}) \times 100\% = (1 - \frac{z'}{K \times N} \times 100\%),$$

where z and z' denote the original raw data size and the compressed data size in bits, respectively; N is the number of data samples, and K is the fixed bits per raw sample specified by the precision of the A/D converter used.

A. Data Sets

First, real-world environmental monitoring WSN datasets from SensorScope are used [13]. To compare with LEC and S-LZW, the same temperature and relative humidity measurements from the following two SensorScope deployments are tested: LUCE and Le Gnpi. The size of the sensor data sets used ranges from 21523 samples to 64913 samples. Both temperature and relative humidity sensors are connected to an ADC. The outputs of ADC for the raw temperature (*raw_t*) and raw relative humidity (*raw_h*) are represented with resolutions of 14 and 12 bits, respectively. These raw outputs *raw_t* and *raw_h* are then converted into physical measures t and h expressed, respectively, in Celsius degrees and percentage (%) as described in [14]. The data sets published on SensorScope corresponding to the two deployments are in the format of physical measures t and h. Therefore, one needs to convert such physical measures back to their corresponding raw measures to evaluate compression algorithms, which can be realized by using the inverted versions of the conversion functions in [14].

Then, real-world volcanic monitoring WSN seismic data sets are used [15], collected via a 19-day WSN deployment at Reventador, an active volcano in Ecuador [11], [12]. The resolution of sampled seismic signals is 24 bits, and sample rate is 100Hz. The size of the seismic data sets used ranges from 7885 samples to 8268 samples. Unlike the relatively smooth SensorScope temperature and relative humidity data, the seismic data of volcanic eruptions are highly dynamic, presenting challenges to data compression approaches. The volcanic data 2005-08-11_03.36.40 from [15] is used in our experiment, whose range is projected from the normalized (-1,1) to their original (0, 2^{24}-1) to obtain raw ADC readings.

B. Compression Results

To make a direct comparison with the results of LEC and S-LZW reported in [3], we follow the same assumptions made in [3] for SensorScope WSN data sets: (1) a raw temperature (*raw_t*) sample and a raw relative humidity (*raw_h*) sample will take 16 bits when transmitted uncompressed; and (2) the size of each data block M for compression is 264 samples. Table IV lists the lossless compression performance of S-LEC, LEC and S-LZW, using SensorScope temperature and relative humidity data sets.

For volcanic data sets, the data block size used for compression is 176 samples. Five sensor nodes' sampled data are employed and the lossless compression performance of S-LEC and LEC is listed in Table V.

TABLE IV
COMPRESSION PERFORMANCE COMPARISONS ON SENSORSCOPE DATA

Variable	Data Set	Compression Ratio CR(%)		
		S-LEC	LEC	S-LZW*
Temp.	LU_ID84	72.07	70.46	48.99
	LG_ID20	54.80	53.54	22.02
RH	LU_ID84	63.71	62.00	31.24
	LG_ID20	51.40	48.12	21.93

*Compression ratios by S-LZW are adopted from [3].

TABLE V
COMPRESSION PERFORMANCE COMPARISONS ON VOLCANO DATA

Data Set	Compression Ratio CR(%)	
	S-LEC	LEC
Node 200	30.99	12.65
Node 201	28.31	9.76
Node 202	31.27	13.91
Node 205	26.76	4.40
Node 251	25.08	-0.39

It can be seen from Table IV that the compression performance of the proposed S-LEC achieves better compression performance than both LEC and S-LZW on all different SensorScope WSN data sets.

Next, we focus on comparisons between S-LEC and LEC, as both LEC and S-LEC have significantly better compression ratios than S-LZW shown in Table IV. We observe that, from Table V, S-LEC and LEC have dramatically different performance results on volcano WSN data sets, although their performance difference for SensorScope data sets are mild. As we can see in Table V, S-LEC improves the compression ratios by a factor of more than two in comparison with LEC for the first four sensor nodes' data sets. In particular, for node 251, while S-LEC can still achieve more than 25% compression ratio, LEC completely fails.

C. Discussions

These two groups of comparison results on two different real-world WSN dada sets reveal some interesting fact that LEC can perform extremely different depending on the characteristics of WSN data sets. While LEC could perform well on Sensorscope WSN temperature and relative humidity measurements that are quite smooth, it performs poorly on WSN volcanic data that are very dynamic. The insight is that LECs coding table (Table I) is optimized when residues generated from sampled data concentrate on zero and follow a Laplace-like distribution. Since the smooth WSN data at SensorScope largely satisfy this assumption, LEC can lead to good compression performance. However, volcanic data are very dynamic and their residues do not follow the distribution assumed by the LEC entropy encoder, which eventually fails LEC approach. In contrast, the proposed S-LEC can perform well for both smooth WSN data and dynamic WSN data, due to its capability to exploit temporal information remained in residue series, which demonstrates a high degree of robustness of S-LEC for diverse WSN data sets in broad real-world applications and deployments.

TABLE VI
ENERGY CONSUMPTION COMPARISONS ON SENSORSCOPE DATA

Variable	Data Set	Energy Consumption($E_{radio} + E_{comp}$)(mJ)		
		S-LEC	LEC	Uncompressed
Temp.	LU_ID84	142.34+1.53	150.27+1.04	508.61+0
	LG_ID20	76.29+0.75	78.34+0.56	168.64+0
RH	LU_ID84	184.85+1.88	190.91+1.35	508.61+0
	LG_ID20	82.03+0.78	86.99+0.60	168.64+0

IV. ENERGY CONSUMPTION ANALYSIS

A simplified energy consumption model on source sensor nodes for the proposed S-LEC can be obtained by counting the number of basic operations (e.g., shifts, additions) conducted in S-LEC algorithm. For example, it needs n shifts in the encoder to obtain the code if the residue is in the n^{th} group of the coding table. Thus, based on simulation, we can obtain the actual operation numbers needed for each data set in the following energy consumption analysis. Considering the widely used CC2420 radio transceiver and ARM7TDMI microprocessor of motes, we use the parameters of energy consumption model given in [4]. Consider one hop transmission in the following analysis. Therefore, transmitting k bytes per hop requires

$$E_{radio}(k) = kI_{TX}VT_{TX} + kI_{RX}VT_{RX}.$$

Denote N_{add}, N_{sht}, and N_{cmp} the number of *add*, *shift*, and *comparison* operation, respectively. The total computation energy consumption would be

$$E_{comp} = N_{add}\epsilon_{add} + N_{sht}\epsilon_{sht} + N_{cmp}\epsilon_{cmp},$$

where ϵ_{add}, ϵ_{sht} and ϵ_{cmp} are the energy consumption of addition, shift and comparison instruction respectively, given in [4]. Then, the total energy consumption is computed as

$$E = E_{radio} + E_{comp}.$$

Based on the above energy consumption model, we computed the total energy consumption of lossless compression in a one-hop WSN for both S-LEC and LEC, as shown in Table VI (including both E_{radio} and E_{comp}). The energy analysis was conducted on SensorScope data sets for which LEC can perform well. We can observe that S-LEC is overall more energy efficient than LEC, even in a one-hop WSN. This indicates that the communication energy savings achieved by S-LEC exceeds its computation energy cost of slightly more processing compared to LEC. Clearly, the energy savings of S-LEC over LEC on the dynamic volcano data sets will be dramatically higher. Furthermore, as the number of hops in a WSN increases, significantly more energy savings of S-LEC over LEC will be achieved.

V. CONCLUSION

We extend and modify the previous LEC and devise Sequential LEC (S-LEC) algorithm for efficient and robust data compression in WSNs. We evaluate the S-LEC using the real-world WSN diverse data sets including smooth temperature and relative humidity data and dynamic volcanic data that exhibit dramatic different characteristics. Our results show that S-LEC significantly outperforms the recent popular data compression algorithms S-LZW and LEC. In particular, while the performance of LEC is very sensitive to the characteristics of WSN data series, S-LEC shows a high degree of robustness regardless of the dramatic characteristics exhibited in diverse WSN data series. We also conduct energy consumption analysis for S-LEC versus LEC on SensorScope data sets, indicating the better overall energy efficiency of S-LEC than LEC even for those smooth sensor data sets in a one-hop WSN. For dynamic WSN data series such as volcano data in multi-hop WSNs, the energy efficiency of S-LEC compared to LEC will be significantly greater. Hence, the robustness and efficacy of S-LEC make it very suitable for data collection in various real-world WSN deployments.

REFERENCES

[1] C. M. Sadler and M. Martonosi, "Data compression algorithms for energy-constrained devices in delay tolerant networks," in *Proc. 2006 International Conference on Embedded Networked Sensor Systems*, pp. 265–278.

[2] F. Marcelloni and M. Vecchio, "A simple algorithm for data compression in wireless sensor networks," *IEEE Commun. Lett.*, vol. 12, no. 6, pp. 411–413, 2008.

[3] ——, "An efficient lossless compression algorithm for tiny nodes of monitoring wireless sensor networks," *The Computer J.*, vol. 52, no. 8, pp. 969–987, 2009.

[4] Y. Liang and W. Peng, "Minimizing energy consumptions in wireless sensor networks via two-modal transmission," *ACM SIGCOMM Computer Commun. Review*, vol. 40, no. 1, pp. 12–18, 2010.

[5] F. Huang and Y. Liang, "Towards energy optimization in environmental wireless sensor networks for lossless and reliable data gathering," in *Proc. 2007 IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems*, pp. 1–6.

[6] T. A. Welch, "A technique for high-performance data compression," *Computer*, vol. 17, no. 6, pp. 8–19, 1984.

[7] P. Elias, "Predictive coding-i," *IRE Trans. Inf. Theory*, vol. 1, no. 1, pp. 16–24, 1955.

[8] ——, "Predictive coding–ii," *IRE Trans. Inf. Theory*, vol. 1, no. 1, pp. 24–33, 1955.

[9] J. Teuhola, "A compression method for clustered bit-vectors," *Inf. Process. Lett.*, vol. 7, no. 6, pp. 308–311, 1978.

[10] G. Barrenetxea, F. Ingelrest, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "Sensorscope: out-of-the-box environmental monitoring," in *Proc. 2008 International Conference on Information Processing in Sensor Networks*, pp. 332–343.

[11] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh, "Deploying a wireless sensor network on an active volcano," *IEEE Internet Computing*, vol. 10, no. 2, pp. 18–25, 2006.

[12] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, "Fidelity and yield in a volcano monitoring sensor network," in *Proc. 2006 Symposium on Operating Systems Design and Implementation*, pp. 381–396.

[13] (2013, Oct.) Sensorscope homepage. Available: http://lcav.epfl.ch/op/edit/sensorscope-en

[14] (2013, Oct.) Sensirion homepage. Available: http://www.sensirion.com

[15] (2013, Oct.) Volcano homepage. Available: http://fiji.eecs.harvard.edu/Volcano