Journal of Network and Computer Applications 47 (2015) 23–31

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Comparison of local lossless compression algorithms for Wireless Sensor Networks

G. Campobello *, O. Giordano, A. Segreto, S. Serrano

Dipartimento di Ingegneria Elettronica, Chimica e Ingegneria Industriale (DIECII), University of Messina, C.da di Dio, 98166 Messina, Italy

ARTICLE INFO

Article history:
Received 6 December 2013
Received in revised form
23 July 2014
Accepted 8 September 2014
Available online 5 October 2014

Keywords:
Lossless compression
Wireless Sensor Networks

ABSTRACT

Compression algorithms are deeply used in Wireless Sensor Networks (WSNs) for data aggregation in order to reduce energy consumption and therefore increasing network lifetime. In this paper we compare several lossless compression algorithms by means of real-world data. Moreover we present a simple and effective lossless compression algorithm that is able to outperform existing solutions and that, considering its inherent low complexity and memory requirements, is well suited for WSNs.

1. Introduction

The use of wireless sensor networks (WSNs) for data acquisition is now widespread. Today, in fact, WSN represents a great tool for military, telemedicine, home automation, farming and, in general, for scientific applications. Basically, a WSN is made by a distributed set of sensor nodes, each of which consists of four basic elements: a sensing module (sensors and A/D converter), a micro-controller, a communication unit, and a power supply module (Akyildiz et al., 2002). Due to economical reasons, power supply modules are usually based on small size batteries that in most application scenarios are difficult or even impossible to replace or recharge. So the most important design parameter for WSNs is usually energy consumption. A sensor node uses its energy mainly to the activities of sensing, data processing and communication. Among these three functions, the last is the one that requires the greatest amount of energy. In commonly used sensor nodes, in fact, transmitting a single bit on the radio channel needs the same energy required to perform around 3000 instructions (Wang et al., 2010).

As a consequence energy saving can be achieved through mainly two approaches: duty cycling (Campobello et al., 2010; Dargie, 2012), e.g. by switching-off the transmitter when it is not in use, and data aggregation (Campobello et al., 2013; Liao et al., 2008), e.g. an on board pre-processing within the sensor node, with the aim of reducing the number of bits to be transmitted.

Most data aggregation techniques adopt compression algorithms for reducing the number of bits to be transmitted and therefore reducing energy consumption and increase network lifetime (Srisooksai et al., 2012). In general lossy compression algorithms allow much higher compression ratios, however lossless compression algorithms are widespread in sensor networks and are mandatory in several scenarios. For instance this is true in new scientific experiments (where accuracy of observations is critical and there is no knowledge about tolerable errors) and in the case of WSNs developed for biomedical and health-related signals (where it is necessary to ensure that medically important details are not lost causing errors in medical diagnosis).

Despite several lossless compression algorithms exist (for instance the well-known Lempel–Ziv algorithm LZWC: Lempel-Ziv-Welch Codec), most of them are not suitable when only limited storage and computational resources are available (Barr and Asanovic, 2006). This is the case of WSNs, where central processing units of sensor nodes (motes) are low-cost and low-speed micro-controllers with just a few kilobytes of memory.

A recent survey on data compression algorithms for WSNs is reported in Srisooksai et al. (2012). In particular, authors emphasizing the fact that "only a few literature works have discussed the lossless compression for local data compression approaches".

Motivated by the above paper we compared several lossless compression algorithms using both Gaussian distributed data and real-world data.

Moreover we derived a simple and effective lossless compression algorithm, henceforward named MinDiff, that is able to outperform conventional lossless compression algorithms and that, considering its inherent low complexity and memory requirements, is well suited for low-cost micro-controller and embedded devices as

* Corresponding author. Tel.: +39 090 3977378.
E-mail addresses: gcampobello@unime.it (G. Campobello), ogiordano@unime.it (O. Giordano), asegreto@unime.it (A. Segreto), sserrano@unime.it (S. Serrano).

http://dx.doi.org/10.1016/j.jnca.2014.09.013
1084-8045/

those used in WSNs. The proposed algorithm has been tested with different real-world data sets related to different physical parameters.

2. Related works

Basically, when local lossless algorithm is considered, the common approach is to exploit temporal correlation and a simple method is to use differences among two consecutive samples (commonly called residues). As shown in Srisooksai et al. (2012) and references therein, residues of different real-world data (temperature, humidity, solar radiation, etc.) fit well with Gaussian or Laplace distributions. As a consequence the basic idea behind several compression algorithms is to use a dictionary approach to encode residues near Shannon's entropy on the basis of preliminary information about their distribution.

Examples of dictionary-based approaches are

- *S-LZW* (Sadler and Martonosi, 2006) where the authors simplify the well-known algorithm of Lempel–Ziv–Welch by taking into account limited resources of sensor nodes. Basically, the algorithm divides data to be compressed into blocks of fixed size, and then separately compresses each block by using a dictionary of 256 words.
- *SHuffman* (Marcelloni and Vecchio, 2008), where for each new measurement x_i acquired by the sensor, the residue $r_i = x_i - x_{i-1}$ is calculated and encoded by a binary sequence on the basis of the dictionary reported in Table 1. More precisely, each binary sequence $(s_i|a_i)$ is composed by a prefix (s_i) that specifies the subset in which r_i lies and an m_i-bit index (a_i) that codifies the position of the value r_i within the subset referred by s_i.
  For sake of completeness in the last column of Table 1 we report the number of bits l_i needed to encode r_i.
  This compression technique exploits the correlation between two consecutive samples allowing us to achieve high compression ratios when differences have small values and, as reported in Marcelloni and Vecchio (2008), in this case it outperforms the S-LZW.
- *ND-Encoding* (Xuejun and Dingyi, 2010) is a compression technique able to achieve high compression ratios in the case of slowly varying data with Normal Distribution. Basically, the ND-Encoding calculates the residues and encodes them using the dictionary shown in Table 2 optimized for normal distributed data with a very small variance (i.e. $\sigma^2 = 7$).

A second class of compression algorithms is based on a predictive coding approach. Basically, this class of algorithms is based on the fact that in most cases it is sufficient to encode only those residues, resulting from the difference between the predicted value and the actual value, which fall inside a relatively small range $[-R, R]$ and to transmit the values outside this range (i.e. outliers) as the original raw data. This approach is commonly known as Two-Modal (TM) transmission.

In order to estimate R and the best code for residues in the range of $[-R, R]$, the TM approach proposed in Liang and Peng (2010) uses a second-order linear predictor and represents residues by an M-based alphabet. The authors propose a heuristic method for the choice of the pair (M,R), but they admit that is still necessary to develop a formula to optimize the problem of finding the best pair (M,R) to minimize the size of the compressed data.

The main problem of this approach is that only the sink can implement the needed estimator due to "huge" complexity of predictive algorithms in comparison to the limited storage and computational resources available in sensor motes. So, despite the sink is energy capable, in the case of sparse multi-hop WSNs, the energy (and delay) needed for forwarding update messages should also be considered.

In this paper we propose a simple lossless compression technique that, in comparison to the previous algorithms, is able to achieve greater compression ratios exploiting the fact that a simple prediction can be done directly on sensor nodes using the range of the actual set of data.

In the following sections, after formally defining the compression ratio, we present the basic idea underlying the proposed method and its performance.

Table 1
SHuffman dictionary.

r_i	s_i	m_i	l_i
0	00	0	2
$-1,+1$	010	1	4
$-3,-2,+2,+3$	011	2	5
$-7,\ldots,-4,+4,\ldots,+7$	100	3	6
$-15,\ldots,-8,+8,\ldots,+15$	101	4	7
$-31,\ldots,-16,+16,\ldots,+31$	110	5	8
$-63,\ldots,-32,+32,\ldots,+63$	1110	6	10
$-127,\ldots,-64,+64,\ldots,+127$	11110	7	12
$-255,\ldots,-128,+128,\ldots,+255$	111110	8	14
$-511,\ldots,-256,+256,\ldots,+511$	1111110	9	16
$-1023,\ldots,-512,+512,\ldots,+1023$	11111110	10	18
$-2047,\ldots,-1024,+1024,\ldots,+2047$	111111110	11	20
$-4095,\ldots,-2048,+2048,\ldots,+4095$	1111111110	12	22
$-8191,\ldots,-4096,+4096,\ldots,+8191$	11111111110	13	24
$-16383,\ldots,-8192,+8192,\ldots,+16383$	111111111110	14	26

Table 2
ND-Encoding dictionary.

r_i	s_i	m_i	l_i
0	00	0	2
$-1,+1$	01	1	3
$-3,-2,+2,+3$	10	2	4
$-5,-4,+4,+5$	110	2	5
$-7,-6,+6,+7$	1110	2	6
All others' data	1111	w	$4+w$

3. Entropy and compression ratio

We can measure performance of a compression algorithm using the compression ratio hereby defined as

$$r_c = 1 - \frac{\text{of bits after compression}}{\text{of bits before compression}} \tag{1}$$

Obviously better compression algorithms have greater compression ratios.

As well known, when compression of discrete sources is considered, Shannon's entropy H gives the lossless compression limit. Therefore ideal (maximum) lossless compression ratio considering blocks of N correlated values of w-bit each can be obtained as

$$r_{C,ideal} = 1 - \frac{H(X_1, \ldots X_N)}{w \cdot N} \tag{2}$$

where $H(X_1, \ldots X_N)$ is the joint entropy. In the particular case of Gaussian correlated data, under suitable assumptions and without loss of generality, it can be shown that, considering $X_1, \ldots, X_N$ obtained from quantization of continuous Gaussian variables $Y_1, \ldots, Y_N$, it follows Cover and Thomas (1991)

$$H(X_1, \ldots X_N) = h(Y_1, \ldots Y_N) = \tfrac{1}{2} \log_2((2\pi e)^N \cdot |\Sigma|) \tag{3}$$