



HAL
open science

Using Data Compression for Delay Constrained Applications in Wireless Sensor Networks

Eugène Chichi, Jean-Michel Friedt, Hervé Guyennet

► **To cite this version:**

Eugène Chichi, Jean-Michel Friedt, Hervé Guyennet. Using Data Compression for Delay Constrained Applications in Wireless Sensor Networks. SENSORCOMM'10, 4-th Int. Conf. on Sensor Technologies and Applications, 2010, Venice-Mestre, Italy. pp.101 -107. hal-01313634

HAL Id: hal-01313634

<https://hal.science/hal-01313634>

Submitted on 10 May 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Using Data Compression for Delay Constrained Applications in Wireless Sensor Networks

M. Eugène Pamba Capo-Chichi
Laboratory of computer science
University of Franche-Comté
Besançon, France

Email: mpamba@lifc.univ-fcomte.fr

Jean-Michel Friedt
Time-Frequency Department/FEMTO-ST
University of Franche-Comté
Besançon, France

Email: jmfriedt@femto-st.fr

Hervé Guyennet
Laboratory of computer science
University of Franche-Comté
Besançon, France

Email: hguyennet@lifc.univ-fcomte.fr

Abstract—Data compression is a technique used to save energy in Wireless Sensor Networks by reducing the quantity of data transmitted and the number of transmission. Actually, the main cause of energy consumption in WSN is data transmission. There exist critical applications such as delay constrained activities in which the data have to arrive quickly to the Sink for rapid analysis. In this article, we explore the use of data compression algorithms for delay constrained applications by evaluating a recent data compression algorithm for WSN named K-RLE with optimal parameters on an ultra-low power microcontroller from TI MSP430 series. The relevance of the parameter K for the lossy algorithm K-RLE led us to propose and compare two methods to characterize K: the Standard deviation and the Allan Deviation. The last one allow us to control the percentage of data modified. Experimental results show that data compression is an energy efficient technique which can also perform in certain cases the global data transfer time (compression plus transmission time) compared to direct transmission.

Keywords-wireless sensor network; data compression; delay constrained application; energy saving.

I. INTRODUCTION

The interest of Wireless Sensor Networks for delay constrained application is becoming more and more important. Indeed, sensor network is formed by distributed autonomous nodes able to sense critical environmental data, compute and transmit it using wireless communication. The way to transmit data is very important as it impact on the application. Actually, there exist two main kinds of sensor network applications which are (i) demand-driven and (ii) event-driven applications. In event-driven application such as forest fire, the traffic is done on real-time, this is contrary to existing wireless sensor network technics for saving energy such as the sleep mode.

The standard designed for WSNs, IEEE 802.15.4 [1], is able to save energy for nodes which are power limited by using the sleep mode but also reveals some difficulties for heavy time constraint applications [2]. Other wireless technologies with interesting transfer time such as WIFI exist but they are not adapted for WSN because of their power consumption [3]. In this way, we study one technique that permits on the one hand to save energy and on the

other hand to respond to delay constrained applications: data compression.

Primarily, data compression is a strategy used to save energy in WSN by avoiding to send data systematically because data transmission is the main cause of energy consumption for sensor nodes [4], [5]. However, reducing the data to be transmitted by compression also minimises the time to transfert it. In this way, while data compression is designed to save energy, it can also allow us to reduce the delay during data transmission. Nevertheless, the question is to know if is it always better to compress data before transmission? Indeed, the compression is an additional action that is why it is interesting to find the cases in which it is better to compress data before transmission.

In general, energy saving and delay are two antagonist criteria in the design of WSN [6] but here, we evaluate the delay optimisation during the use of energy saving strategy which is data compression. The best known lossless compression algorithm for WSN is S-LZW [7] but recently, a new performant lossy compression algorithm for WSN named K-RLE [8], variant of the data compression algorithm RLE, has been introduced. The performance of this energy efficient algorithm depend on the choice of the parameter K which is the precision in which data are encoded. In this work, we will use and compare two different methods to choose K applied on temperature readings which are the Standard deviation and the Allan deviation [9].

However, the choice of K also influences the percentage of data modified for this lossy algorithm. While K-RLE is based on RLE, where the compression results depend of the data source, in this work, we evaluate K-RLE with different statistics of data sources and different values of K.

This paper is organized as follows: the next section reports on related work. Experimental results are given in Section 3. Section 4 presents the conclusion.

II. THE DATA COMPRESSION ALGORITHMS

This section presents the background on data compression algorithms for WSN.

A. S-LZW

The best known data compression algorithm for WSN is S-LZW [7], which is an adaptation of the famous algorithm LZW [10].

Because of the limited RAM of sensor nodes for this dictionary based algorithm, S-LZW introduces limitations compared to LZW:

- A block size of 528 bytes which represents two flash pages. S-LZW divides the uncompressed input bit-streams into fixed size blocks and compresses each block separately.
- A 512 entries dictionary. This algorithm starts by initializing the dictionary to all standard characters of the ASCII code extended to 255 which represent the first 256 entries of the dictionary. For each block used in the compression, the dictionary is re-initialized. A new string in the input bitstream creates a new entry in the dictionary, that is why the data to be compressed are limited. However, different strategies have been developed in order to solve the problem of full dictionary. Two options exist which are to freeze the dictionary and use it as-is to compress the remainder of the data in the block, or it can be reset and started from scratch. However, this problem does not occur when the data block is small, thus the dictionary is not full.
- A mini-cache of 32 entries is added to SLZW in order to get an advantage of repetitiousness of sensor data. The mini-cache is a hash-indexed dictionary of size N, where N is a power of 2, that stores recently used and created dictionary entries.

The major problem of this dictionary based algorithm is the need of a significant RAM size that is why we cannot apply it on platforms with RAM about 2KB like those based on the TI MSP430F149. For this reason, the evaluation of S-LZW in [7] has been done on the TI MSP430F1611 with more RAM about 10 KB.

In this way, a new data compression algorithm named K-RLE, variant of RLE, has been introduced on the one hand to use it on several sensor platforms as those with limited memory and on the other hand to reduce the power consumption and to perform the data compression ratio compared to other existing data compression algorithms for WSN.

On the other side, while S-LZW and RLE are lossless, K-RLE is a lossy compression algorithm.

B. K-Run-Length Encoding

Before presenting K-RLE, it is important to present RLE.

Definition: Run-Length Encoding (RLE) is a basic compression algorithm. As described in [11], the simple idea behind this algorithm is this: If a data item d occurs n consecutive times in the input stream, we replace the n occurrences with the single pair nd .

The main problem of RLE is the compression ratio which depends on the noise level of the data source. Indeed, the noisy data are poorly compressed and this algorithm becomes efficient when there is same consecutive input streams. In this way, in order to perform it with different data sources statistics, we have introduced K-RLE.

The idea behind K-Run-Length Encoding is this:

Definition: let K be a number, If a data item d or data between $d-K$ and $d+K$ occur n consecutive times in the input stream, we replace the n occurrences with the single pair nd .

We introduce a parameter K which is a precision.

- If $K = 0$, K-RLE is RLE. K has the same unit as the dataset values.

The choice of an optimal K is very important for this algorithm as it impacts on the compression ratio and the percentage of data modified. In the next part, we introduce and compare two different functions for representing K : the Standard deviation and the Allan deviation.

C. Finding an optimal value for K

1) *The Standard deviation:* The Standard deviation is based on the variance which is for a set of sample, the mean of the square of the deviation of that sample from its mean. The Standard deviation is precisely the square root of the variance and it is defined as:

$$S = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}, \text{ where } n \text{ is the number of sample } (x_1, x_2, \dots, x_i, \dots, x_n), \text{ and } \bar{x} \text{ is the mean :}$$

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n (x_i).$$

2) *The Allan deviation:* The Allan variance is defined as the expectation of the two-sample variance, ie the classical variance evaluated for N=2 [12]. It is generally used to characterize the time dependence of spread.

Contrary to the classical variance described above, it is a variance term to term used to estimate the stability of the measured period. Historically, it has been introduced to characterize the temporal evolution of the frequency of oscillators. It is given by :

$$\sigma_y^2 = \frac{1}{n-1} \sum_{k=1}^n [\bar{y}_k - \langle y_n \rangle]^2,$$

$$\text{where } \bar{y}_k(\tau) = \frac{1}{\tau} \int_{kt}^{(k+1)t} x(t) dt$$

In this work, we consider a minimum estimate of the Allan deviation. Indeed, the minimum point on the Allan variance curve represents the best point to characterize the noise.

III. EXPERIMENTAL RESULTS

In this section, we use the same strategy as on [8] to evaluate the previous data compression algorithms using a

real temperature dataset of 500 bytes. The emphasis laid on the study of the data compression ratio, the energy consumption and we also focused here on the time for compressing and transmitting data. In order to evaluate the behavior of these algorithms with different data source statistics, we have collected temperatures since the 1st of January 2008 (Fig. 1) from different locations [13] which are: Libreville (Gabon), Cayenne (Guyana), Montbeliard (France), Svalbard (Norway). We have simulated the sensing of temperature as it was sensed by the ADC12 module, a high-performance 12-bit analog-to-digital converter implemented on the MSP430x14x and MSP430x16x devices [14].

Fig. 1 shows the variation of temperatures of different latitude. We notice that the higher the latitude, the greater the temperature change. Certainly, the input stream is an important factor in the performance of compression algorithm that is why we consider our algorithms in real different conditions.

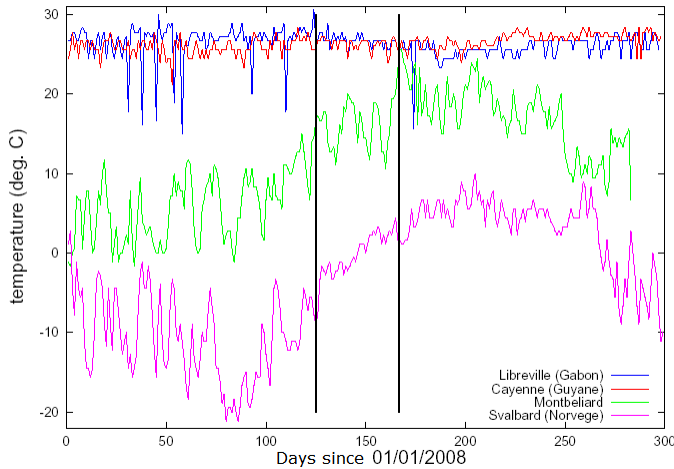


Figure 1. The representation of temperatures variation from different locations

The reference coefficient to estimate the performance of compression algorithms is the data compression ratio. It is defined as:

$$ratio = 100 * (1 - \frac{compressed_size}{initial_size})$$

Due to the RAM size limitation of the MSP430F149 to run S-LZW, we have done all the experiments on a MSP430F1611 in order to make a comparison between S-LZW and K-RLE using Standard and Allan deviation parameters on different data sources.

Fig. 2 shows the variation resulting from the Allan deviation for the locations previously chosen. We notice that while the location is close to the equator, the minimum of the Allan variance is low.

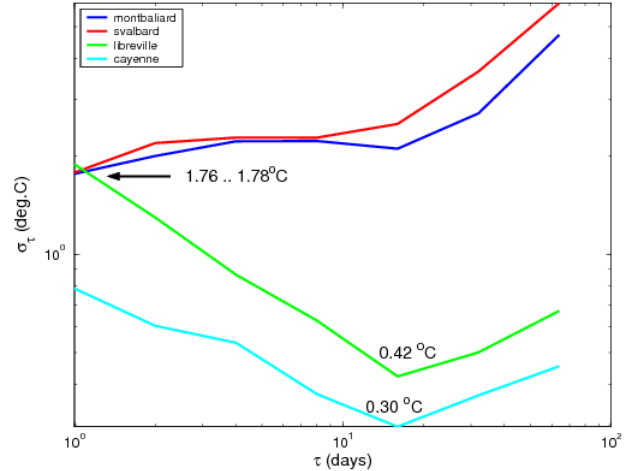


Figure 2. Allan deviation applied on different temperature dataset

Tab. I summarizes all results obtained using Standard and Allan deviation. The idea is to study the function used to determine K before deployment based on one year or more to be more precise. Ideally, the choice of K would be adaptative according to current and anticipated variation. We can see on Tab. I that for all locations, the values of K with Standard deviation is higher than Allan deviation. Parameters mentioned on Tab. I will permit to characterize K.

Table I
PARAMETERS RESULTING FROM ALLAN AND STANDARD DEVIATION

| | Libreville | Cayenne | Montbe | Svalbard |
|----------|------------|---------|--------|----------|
| Standard | 1,9 | 1,1 | 6,8 | 7,9 |
| Allan | 0,4 | 0,3 | 1,7 | 1,7 |

Fig. 3 illustrates the compression ratio of K-RLE using the previous parameters and respectively compared to RLE and S-LZW.

We notice that despite of the Allan deviation parameters (Allan-RLE) which reach 28% compared to the Standard deviation parameters (Std-RLE) about 80%, it controls the percentage of data modified (Fig. 4).

For locations near the equator in which the temperatures are constant, Allan-deviation parameters are close to zero; consequently, Allan-RLE results are the same as RLE.

However, for locations away from the equator and close to the polar circles with great variation in temperatures the data compression ratio obtained with Allan-RLE are close to S-LZW.

It is also interesting to notice that the performance of Std-RLE and Allan-RLE improves when the temperature changes become increasingly important. These results differs from RLE and S-LZW in which the performances becomes

worse as the temperature variation is important. The common point between S-LZW and RLE which explain this phenomenon is that these two algorithms are lossless while K-RLE is lossy.

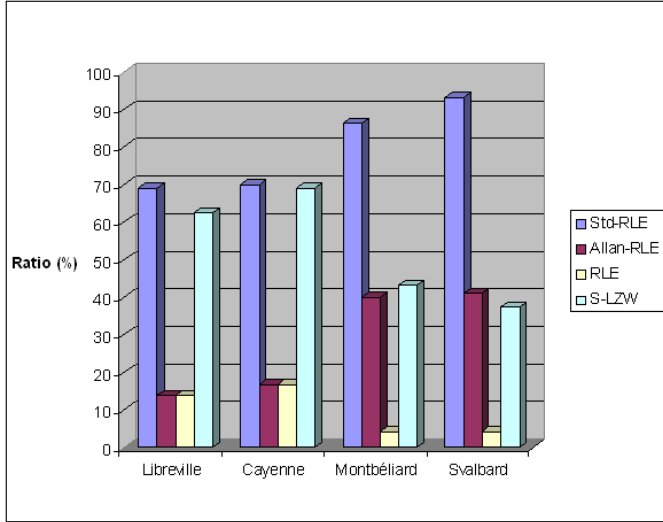


Figure 3. Comparison between S-LZW, RLE and K-RLE using different parameters

Hence, we focused on the percentage of data modified by K-RLE while it offers an interesting data compression ratio (Fig. 4).

Fig. 4 shows that K-RLE offers the best data compression ratio with Std-RLE at the cost of data modified. The percentage of data modified reaches about 22% and 71% respectively for Allan-RLE and Std-RLE. This percentage becomes higher when the temperature changes is significant as for Svalbard up to 93% of data modified with Std-RLE.

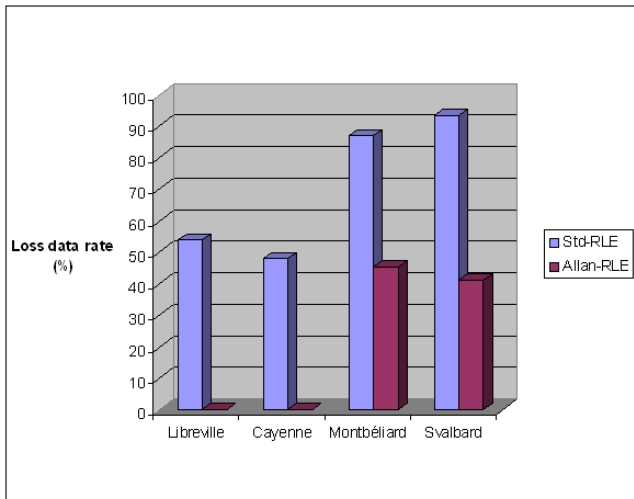


Figure 4. Representation of loss data rate for Std-RLE and Allan-RLE

We note that while Allan-RLE provides worse performance than Std-RLE, it reduces the percentage of data modified and it keeps significant information like RLE for locations in which temperature variations are constant.

This foregoing work shows the relevance of the parameter K in the use of K-RLE. We noticed that the parameters resulting from the Allan deviation allow K-RLE to keep full information in some cases at the expense of compression performance and the opposite effect is produced by Std-RLE.

Nevertheless, there are questions: what about delay and energy consumption when the data compression ratio increases? Is it always possible to use data compression for improving the global data transfert time?

A. Energy consumption and delay evaluation

In this section, we focused on time execution and energy consumption of the previous data compression algorithms using WSim. WSim is an accurate cycle hardware platform simulator. It is a part of Worldsens [15] which is an integrated environment for development and rapid prototyping of wireless sensor network applications. This simulator allows us to use the real target binary code on it. In this way, we have directly used the same program files developed for our real platform on this simulator in which the MSP430 platform has been emulated.

For evaluating energy consumption, we have focused on time execution using led2 and led3 defined on WSim.

Time execution permits to estimate the energy consumption considering the Microcontroller power mode (Fig. 5).

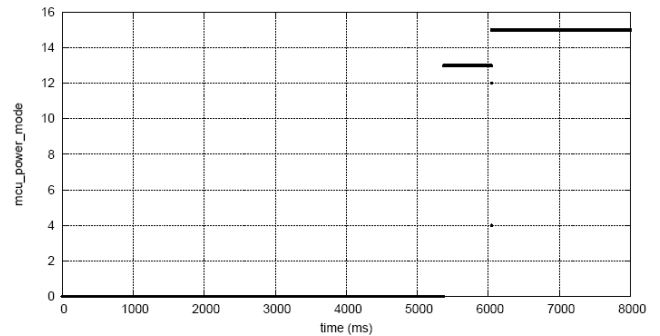


Figure 5. Example of Microcontroller power mode in WSim

In this way, the description of the consumption of each mode (Fig. 6) help us to determine the consumption of each activities.

Data compression is defined as a technique to save energy in WSN that is why we focused on energy consumption of the previous algorithms.

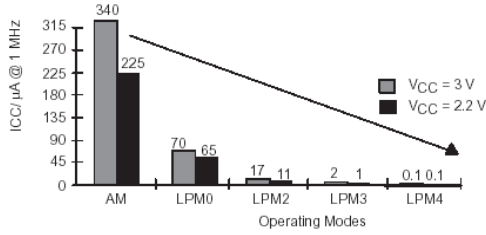


Figure 6. Operating modes [14]

Fig. 7 shows that while S-LZW is lossless with interesting performance, it consumes more energy than others about 0,0224 mJ. The consumption is about 0,0053 mJ and 0,0103 mJ respectively for RLE and K-RLE. Allan-RLE and Std-RLE have the same energy consumption. Actually, we also notice that while RLE and K-RLE have constant consumption, S-LZW uses more energy when there is more change on data.

These results show the antagonism between energy consumption and good compression ratio. Indeed, RLE offers a considerable consumption improvement for compression while it does not give a very good compression ratio.

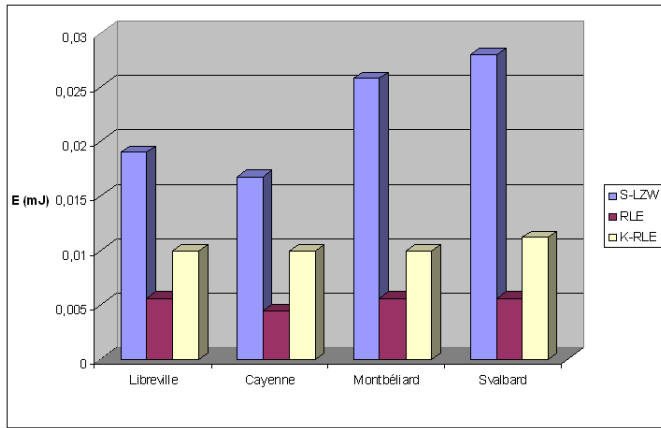


Figure 7. Compression consumption

Since K-RLE offers the best trade-off between energy consumption and data compression performance, we focus on the evaluation of Total Energy (E_{Total}) of Std-RLE and Allan-RLE.

Total Energy is defined as :

$$E_{Total} = E_{compression} + E_{Transmission},$$

where $E_{compression}$ and $E_{Transmission}$ are respectively the energy to compress and the energy to send the compressed data.

For evaluating the delay, we decompose the previous formula using the definition of the Energy ($E = P * T$):

$$E_{Total} = P_{microcontroller} * t_{compression} + P_{transceiver} * t_{transmission}$$

It is important to note that $t_{transmission}$ is much smaller when the compression is effective and generally $P_{transceiver} \gg P_{microcontroller}$.

Since the power of the microcontroller ($P_{microcontroller}$) and the transceiver ($P_{transceiver}$) are constants, we initially focus on the variations of time needed for data compression and transmission.

Fig. 8 describes the time variation for each locations using Std-RLE and Allan-RLE. We study the time variation during data compression and transmission. While the percentage of data modified for Allan-RLE is controlled compared to Std-RLE, for locations such as libreville and Cayenne with constant temperature there is no gain on delay.

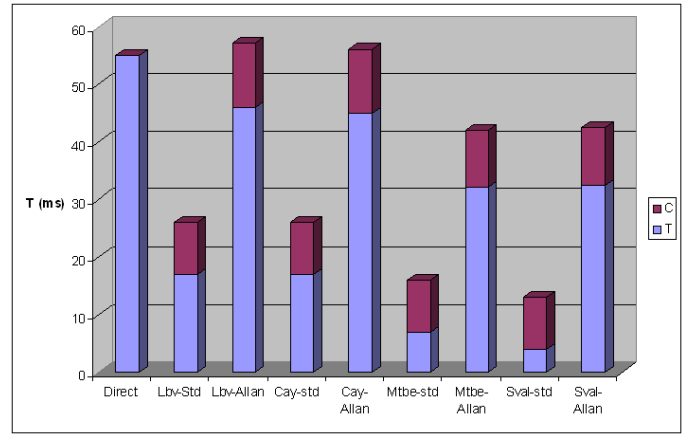


Figure 8. Representation of time variation during compression and transmission for Std-RLE and Allan-RLE

Despite the fact that the transmission time is reduced, we notice by adding compression time that the total time to compress data and transmit compressed data becomes higher than direct transmission. In these cases, a direct transmission is better than compression plus transmission and it is not adapted for delay constrained activities.

However, for all other cases the compression is interesting for improving delay as it reduces the time to transfert data compared to direct transmission even if the time improves only to 10 ms for Allan-RLE.

For Std-RLE, in all cases the results are adapted for improving delay since there is a gain of time more than half compared to direct transmission in average about 35 ms and it reaches up to 42 ms for Svalbard.

We note that it is possible to use the data compression for delay constrained applications as it reduces the time to transfer data compared to direct transmission when the compression is effective, i.e. a good compression ratio is obtained. Nevertheless, while Std-RLE gives the best results for delay constrained applications, it is at the cost of data modified.

However, Allan-RLE gives interesting delay only for locations far away from the equator with more temperature change. In this way, the choice of K is not enough for having good performances, the locations where the algorithm is applied is also important.

After having studied the time variation, we take into account the various power consumption described on Tab. II. For the microcontroller power, we consider the consumption of MSP430 in active mode at 1 MHz (Fig. 6) and we obtain:

$$P_{microcontroleur} = U * I = 3,3V * 340\mu A = 1,12 \text{ mW}.$$

The power of several radio modules is defined in [7] (Tab. II) and we choose the popular Chipcon CC2420 used by several platforms such as those designed by Crossbow [16].

We obtain: $P_{transceiver} = 46,7 \text{ mW}$ and we note here the distinguished difference between $P_{microcontroleur}$ and $P_{transceiver}$.

Table II
RADIO PROFILES OBTAINED VIA MEASUREMENTS [7]

| Radio | Range | Tx power | RX power | Baud Rate |
|-----------------|-------|---------------|----------------|-----------|
| CC2420 | 125 m | 46,7 mW - 3 V | 50,9 mW - 3 V | 70,677 |
| Chipcon CC1000 | 300 m | 64,5 mW - 3 V | 21 mW - 3 V | 12,64 |
| MaxStream XTend | 15 km | 2,43 W - 5 V | 444,5 mW - 5 V | 7,394 |

Fig. 9 shows the total energy consumption. Since the power of the transceiver is higher than the microcontroller power, the consumption of the transmission is phenomenal compared to compression consumption.

In all cases, the compression, plus transmission is better than direct transmission. The direct transmission consumes 2,57 mJ while Allan-Std and Std-RLE uses in average respectively 1,83 mJ and 0,54 mJ. These results illustrate the benefit of data compression in WSN.

Despite of the high percentage of data modified with Std-RLE, it offers good performance in terms of delay and total consumption. However, Allan-RLE gives interesting parameters to control the percentage of data modified compared to Std-RLE.

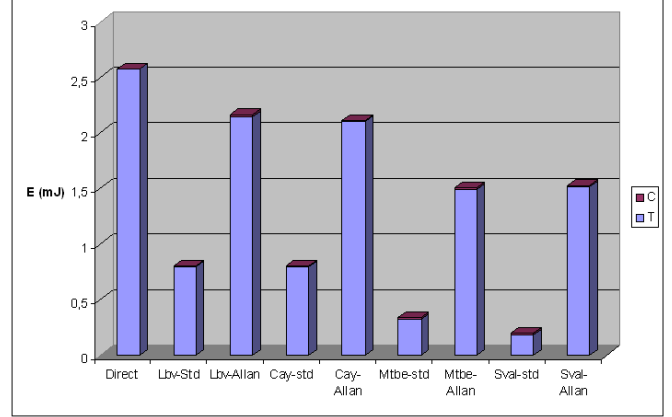


Figure 9. Evaluation of Total energy consumption

In this way, the Allan deviation is the best method to choose K and it is adapted for locations with significant temperature changes as it controls the percentage of data modified for compression ratio close to S-LZW with less energy consumption.

IV. CONCLUSION AND FUTURE WORK

In this paper, we have shown the possibility to use data compression algorithms for delay constrained applications. In this way, we have evaluated a recent performant data compression algorithm for WSN named K-RLE on a ultra-low power microcontroller from Texas Instrument known as MSP430.

This lossy data compression algorithm inspired from RLE is more efficient than the famous algorithm for compressing data in WSN known as S-LZW. The comparison between S-LZW and K-RLE using real temperature datasets shows that K-RLE gives best results in terms of compression ratio and energy consumption.

However, the data compression ratio of K-RLE depends on the parameter K. In this way, this study has characterized the choice of the parameter K using Allan deviation (All-RLE) and Standard deviation (Std-RLE) and it highlights the antagonism between compression efficiency and the quality of information.

Actually, while K-RLE is a simple energy efficient algorithm, usable by several platforms with limited RAM compared to S-LZW which is a dictionary-based algorithm and lossless, it significantly improves the data compression performance at the cost of data modified.

This work confirms the energy consumption gain provided by the data compression in WSN, in all cases during total energy consumption evaluation, compression, plus transmission use less energy than direct transmission. While K-RLE is energy efficient, it is also adapted for delay constrained applications. In some cases, it offers gain in terms of time.

However, while Allan-RLE offers best results than Std-RLE in terms of loss data rate, it does not offers all the time the gain in terms of delay because it can use more time to compress and to transmit compressed data than direct transmission in locations without much temperatures change. In this way, it is not always adapted for delay constrained applications.

Nevertheless, since the percentage of data modified for Std-RLE is high, it is always adapted for delay constrained applications. It gives more than haff time better than direct transmission.

These results show the relevance of the parameter K in K-RLE and also the only way to increase the performance of a data compression algorithm which is necessarily here by losing information.

In this article, we highlight the benefit to use data compression in WSN for improving data transfert delay and energy saving at the cost of data modified. The Allan deviation used by K-RLE allow us to control the percentage of data modified that is why it is the best solution to characterize K applied on locations with high temperature changes.

Future work will focus on the improvements of the control of data modified by K-RLE in all locations.

REFERENCES

- [1] IEEE-TG15.4, *Part 15.4: Wireless Medium Access Control(MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE standard for Information Technology, 2003.
- [2] N. Salles, N. Krommenacker, and V. Lecuire. Performance study of ieee 802.15.4 for industrial maintenance applications. In *IEEE International Conference on Industrial Technology, ICIT 2008*, pp. 1–6, Chengdu, China, 2008.
- [3] M. A. Hanson, H. C. Powell Jr., A. T. Barth, K. Ringgenberg, B. H. Calhoun, J. H. Aylor, and J. Lach. Body area sensor networks: Challenges and opportunities. *Computer*, Vol. 42, No. 1, pp. 58–65, 2009.
- [4] F. Marcelloni and M. Vecchio. A simple algorithm for data compression in wireless sensor networks. *IEEE Communications Letters*, Vol. 12, No. 6, pp. 411–413, June 2008.
- [5] N. Kimura and S. Latifi. A survey on data compression in wireless sensor networks. In *International Conference on Information Technology: Coding and Computing, ITCC 2005* volume 2, pp. 8–13, 2005. .
- [6] E. Pamba Capo-Chichi, H. Guyennet, and J.-M. Friedt. Ieee 802.15.4 performance on a hierarchical hybrid sensor network platform. In *Proceedings of the Fifth International conference on Networking and Services, ICNS 2009*, pp. 303–308, Valencia, Spain, 2009.
- [7] C. M. Sadler and M. Martonosi. Data compression algorithms for energy-constrained devices in delay tolerant networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, SenSys 2006*, pp. 265–278, Colorado, USA, 2006.
- [8] E. Pamba Capo-Chichi, H. Guyennet, and J.-M. Friedt. K-rle: A new data compression algorithm for wireless sensor network. In *Proceedings of the Third International Conference on Sensor Technologies and Applications, SENSORCOMM 2009*, pp. 502–507, Athens/Glyfada, Greece, 2009.
- [9] D. W. Allan. Time and frequency (time-domain) characterization, estimation, and prediction of precision clocks and oscillators. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, Vol. 12, No. 6, pp. 647–654, 1987.
- [10] T. A. Welch. A technique for high-performance data compression. *Computer*, Vol. 17, No. 6., pp. 8-19, 1984.
- [11] D. Salomon. *Data Compression: The Complete Reference*. Second edition, 2004.
- [12] E. Rubiola. *Phase noise and frequency stability in oscillators*. Cambridge University Press, November 2008.
- [13] (2009) The Weather Underground website. [Online]. Available: <http://french.wunderground.com/>.
- [14] (2009) Texas Instruments MSP430x1xx Family User’s Guide. [Online]. Available: <http://focus.ti.com>.
- [15] A. Fraboulet, G. Chelius, and E. Fleury. Worldsens: development and prototyping tools for application specific wireless sensors networks. In T. F. Abdelzaher, L. J. Guibas, and M. Welsh, editors, *IPSN*, pp. 176–185. ACM, 2007.
- [16] (2008) The Crossbow website. [Online]. Available: <http://www.xbow.com>