16th International Conference on Clean Energy (ICCE-2018)
9-11 May 2018, Famagusta, N. Cyprus

ENERGY-AWARE TRANSFER OF COMPRESSED DATA ON MSP432 FAMILY OF MICROCONTROLLERS

Basar Koc[1], Ziya Arnavut[2], Dilip Sarkar[3], Huseyin Kocak[3]
[1]Department of Computer Science, Stetson University, USA
[2]Department of Computer Science, SUNY Fredonia, USA
[3]Department of Computer Science, University of Miami, USA
Corresponding author: Basar Koc, e-mail: bkoc@stetson.edu

REFERENCE NO

MISC-04

ABSTRACT

We previously investigated the power consumption of several lossless data compression algorithms on a Texas Instrument (TI) MSP432 microcontroller. In this study, we examine the power consumption of transmission of data between two such microcontrollers using a TI CC3100 Wi-Fi transmission module. Our experimental measurements show that compression algorithms consume very little power compared to the power required to transmit data between the two microcontrollers. For example, while the transmission of an uncompressed 450x450 image used 79,569.0 mW, the compression with move-to-front transformation along with an arithmetic coder (MTF+AC) required 238.8 mW, and the transmission of the compressed image consumed 49,082.2 mW. In conclusion, to minimize energy consumption in sensor networks we recommend the compression of data as much as possible before transmission.

Keywords:
INA219; CC3100, MSP432. microcontrollers; power measurement, data compression..

1. INTRODUCTION

Wireless sensor networks (WSN) consist of the nodes with limited computing and energy resources. The lifetime of a WSN is usually defined by the battery power of its nodes in the network. Indeed, Puthenpurayil in [1] showed that the main constraint of a WSN is the energy consumption of the nodes in the network. Thus, the amount of power that is used for the transfer of data needs to be justified because the amount of data in transmission can be reduced by utilizing data compression algorithms.

The primary goal of the data compression algorithms is reducing the amount of data so that data can be transferred using less bandwidth in a shorter amount of time. The data transmission time is highly depended on data size. Researchers have proposed different compression techniques in [2], [3], [4], [5], [6]. Zhang et al. in [7] proposed new data compression schemes based on the Lempel-Ziv-Welch coding (LZW) technique for wireless sensor networks. In [8], Capo-Chichi and Friedt introduced a new technique based on the run-length encoding. Some of the proposed data compression algorithms in [6] and [7] require low memory, that means they are

Fig. 1. TI MSP432 and CC3100.

designed for devices with very limited computing power, such as Texas Instrument (TI) MSP430 family of microcontrollers. The limited computing resources on these devices cause subpar compression rates.

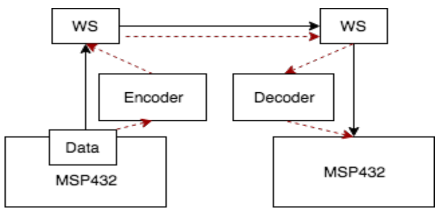

Fig. 2. Proposed data transfer scheme.

Recently, TI introduced the MSP432 family of microcontrollers shown in Fig.1., that has more computing power including float-point arithmetic and more memory than the MSP430 family of microcontrollers. With the MSP432, more computationally demanding data compression algorithms can be utilized on nodes in a WSN, and better compression gains can be obtained for example using an arithmetic coder (AC), instead of a run-length coder (RLE).

In our previous study [9], we investigated the power consumption of data compression algorithms, e.g., various linear data transformations, a run-length coder, and an arithmetic coder, on a TI MSP432 family microcontroller. Our experiments showed that the arithmetic coder consumed the least power and required the lowest time for a compression gain. However, when we pre-processed the data with a linear transformation and a run-length coder, the data compression gain is doubled on average with an extra 20 % energy usage.

In this study, we analyze the total power consumption of the wireless transfer of compressed data between two such microcontrollers using a TI CC3100 Wi-Fi transmission module [10], as shown in Fig.2. First, we apply a compression algorithm to data and transfer the compressed data to another microcontroller. We record the energy requirement for the compression and transmission. Second, we transfer the data without utilizing any compression algorithm and record the energy consumption of the system.

This paper is planned as follows: In section II, the description of the linear transformations and entropy coders used in the experiment are given. In Section III, the experimental results are discussed including how the power consumption is measured, and finally, in Section IV, the conclusions of our paper are summarized.

II. TRANSFORMATIONS AND CODERS

Several researchers have investigated different compression techniques such as LZW and Huffman Coding [5]. In this work, the compression technique we utilize consists of four different stages. The first and second stages are transformations: The Burrows-Wheeler transformation (BWT) [11] and the move-to-front (MTF) transformation [12]. Data transformed by BWT and MTF is then processed with a run-length (RLE) and arithmetic coder (AC) [13]. Here we briefly explain each stage.

The Burrows and Wheeler transformation (BWT) aims to collect lexically similar contexts close to each other. It is a valuable linear transformation for text and image data which is used by well-known compressors such as bzip. The efficiency of BWT is highly related to window size used in transformation. While the selection of bigger window sizes helps to obtain better compression gain, it requires more memory space.

The move-to-front (MTF) [12] coder (or recency ranking) outputs the index value of the most recently seen symbol, which is the count of “How many elements are in front of the symbol seen most recently?” This yields a better compressible data, with a lot of runs, when the data have locality of reference. Fenwick [13] notes that when the MTF coder is applied to Burrows-Wheeler transformed files, the resulting recency ranks yield a high number of zeros and ones.

The run-length coder (RLE) is the third stage. It detects long run sequences and regroups the repetitive sequences with the current symbol and the number of repetitions. RLE has two main advantages. First, RLE increases the efficiency of entropy coder by reducing the number of repeated symbols. Second, RLE decreases the length of the sequences by encoding symbols more efficiently and helps to decrease computation time of the entropy coder.

The last stage of our technique is an Entropy Coding (EC) such as the Huffman or Arithmetic coder. It has been found that an Arithmetic Coder (AC) [14] is a better choice to achieve higher compression. In contrast to the Huffman coder, AC evaluates the probability of data sets and optimizes the length of the required code.

III. EXPERIMENTAL RESULTS

In our experiments, we used *Lenna* image file obtained from the USC-SIPI Image Database and converted from TIFF to color-index-based bitmap format using GIMP program. The linear transformations and entropy coders that are used in this study from [15], and in BWT, the window size is selected as 450 characters mainly due to the limitation of MSP432 memory space.

To measure the energy consumption of the MSP432, we used our custom circuit [9] assembled from two off-the-shelf hardware: the TI INA219 DC current sensor breakout and Arduino Uno R3 boards, as shown in Fig. 3.

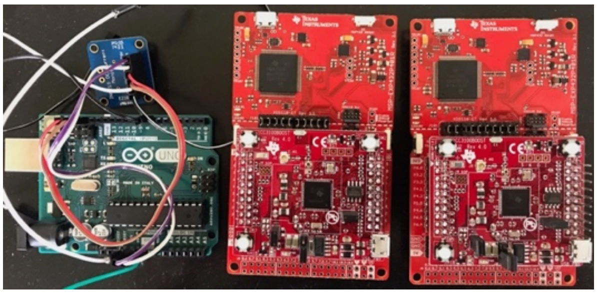

Fig. 3. Arduino UNO with INA219.

Our experimental measurements using TI INA219 with Arduino UNO showed that lossless compression algorithms consume very little power as shown in Table 1, compared to the power required to transmit data between the two microcontrollers.

Table 1. Total compression time and energy usage of various compression algorithms, as reported in [9].

Algorithms	Input File Size	Output File Size	Time(ms)	Power (mW) Compression	Compression Ratio
AC	202,500	158,172	6,987	196.6	21.89%
BWT+AC	202,500	167,508	14,105	386.9	17.28%
MTF+AC	202,500	124,764	8,519	238.8	38.39%
RLE+AC	202,500	165,877	7,494	210.9	18.09%

In Table 2 and 3, we reported the total energy consumption and run time of the compressed data including the compression process.

Table 2. Total energy consumption of data compression (MTF+AC) and transfer of compressed data.

Image Size	Input File Size	Power (mW) Compression	Power (mW) Transmission	Power (mW) Total
100 x 100	10,000	16.25	3,042.16	3,058.41
150 x 150	22,500	32.69	6,500.94	6,533.63
200 x 200	40,000	54.62	11,085.23	11,139.85
250 x 250	62,500	82.72	16,840.27	16,922.99
300 x 300	90,000	110.20	23,443.10	23,553.30
350 x 350	122,500	148.24	31,099.84	31,248.08
400 x 400	160,000	183.05	39,736.55	39,919.60
450 x 450	202,500	238.80	49,082.16	49,320.96

Table 3. Total time of data compression (MTF+AC) and transfer of compressed data.

Image Size	Input File Size	Compression Time (milisecond)	Transmission Time (milisecond)	Total Time (milisecond)
100 x 100	10,000	599	7,210	7,809
150 x 150	22,500	1,229	16,215	17,444
200 x 200	40,000	2,055	28,825	30,880
250 x 250	62,500	3,083	45,046	48,129
300 x 300	90,000	4,228	65,182	69,410
350 x 350	122,500	5,519	88,290	93,809
400 x 400	160,000	6,966	115,298	122,264
450 x 450	202,500	8,519	145,931	154,450

For example, while the transmission of an uncompressed 450x450 image used 79,569.0 mW, the compression with move-to-front transformation along with an arithmetic coder (MTF+AC) required 238.8 mW, and the transmission of the compressed image consumed 49,082.2 mW, as shown in Table 2.

We found that there is a linear correlation, as in Fig. 4 and 5, between the amount of data transmitted and the duration of the transmission process and power consumption of the microcontroller.

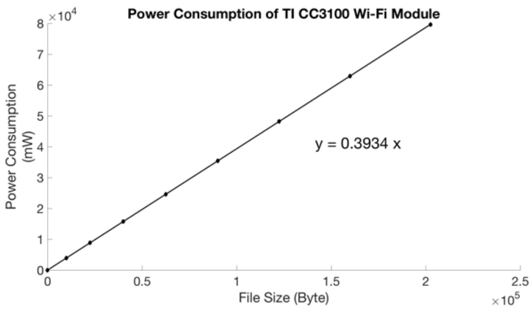

Fig. 4. Power consumption during data transmission.

IV. CONCLUSIONS

In this study, we investigated the total energy consumption of wireless data transfer between MSP432 microcontrollers with and without utilizing a lossless compression algorithm.

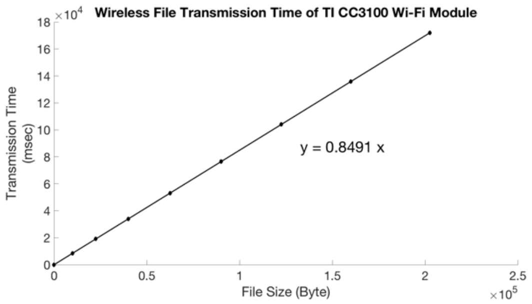

Fig. 5. Duration of wireless file transmission.

We found that compression algorithms consume very little power compared to the power required to transmit data between the two microcontrollers in a WSN. In order to minimize energy consumption of the nodes in a WSN and increase the lifetime of the system, we recommend the compression of data before transmission.

Acknowledgements

The authors thank the State University of New York at Fredonia and the University of Miami for providing the test devices used in this research.

References

[1] Puthenpurayil, S.; Ruirui Gu; Bhattacharyya, S.S., "Energy-Aware Data Compression for Wireless Sensor Networks," in Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on, vol.2, no., pp. II-45-II-48, 15-20 April 2007.

[2] Kim, S., Cho, C., Park, K. and Lim, H. (2017) Increasing network lifetime using data compression in wireless sensor networks with energy harvesting, Distributed Sensor Networks, Vol. 13(1), SAGE.

[3] Imon, K., Khan, A., and Das, S., (2014) EFFECT: An energy efficient framework for data compression in tree-based wireless sensor networks, Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks.

[4] Marcelloni, F.; Vecchio, M., "A Simple Algorithm for Data Compression in Wireless Sensor Networks," in Communications Letters, IEEE, vol.12, no.6, pp.411-413, June 2008.

[5] Kimura, N.; Latifi, S., "A survey on data compression in wireless sensor networks," in Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on, vol.2, no., pp.8-13 Vol. 2, 4-6 April 2005.

[6] Reinhardt, A.; Christin, D.; Hollick, M.; Steinmetz, R., "On the energy efficiency of lossless data compression in wireless sensor networks," in Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on, vol., no., pp.873-880, 20-23 Oct. 2009.

[7] Huan Zhang; Xiao-Ping Fan; Shao-qiang Liu; Zhi Zhong, "Design and realization of improved LZW algorithm for wireless sensor networks," Information Science and Technology (ICIST), 2011 International Conference on, vol., no., pp.671,675, 26-28 March 2011.

[8] Capo-Chichi, E.P.; Guyennet, H.; Friedt, J.-M., "K-RLE: A New Data Compression Algorithm for Wireless Sensor Network," Sensor Technologies and Applications, 2009. SENSORCOMM '09. Third International Conference on, vol., no., pp.502,507, 18-23 June 2009.

[9] Koc, B., Sarkar, D., Kocak, H. and Arnavut, Z.,"A study of power consumption on MSP432 family of microcontrollers for lossless data compression," 2015 HONET.

[10] CC3100 SimpleLinkTM Wi-Fi® and IoT Solution Getting Started Guide, url: www.ti.com/lit/ug/swru375c/swru375c.pdf

[11] Burrows M.; Wheeler D.J., "A block-sorting lossless data compression algorithm," Technical Report, Digital Equipment Corporation, Palo Alto, CA, 1994.

[12] Bentley J. L.; Sleator D.D.; Tarjan R.E.; Wei V., "A locally adaptive data compression scheme," CACM, 29(4), 320-30, 1986.

[13] Fenwick, P.m "The Burrows-Wheeler Transform for Block Sorting Text Compression: Principles and Improvements," The Computer Journal, Vol. 39, No. 9, 731-740, 1996.

[14] Witten, I.H.; Neal, R.; Cleary, J.G., "Arithmetic coding for data compression," Communications of the Association for Computing Machinery, 30 (6) 520-540, 1987.

[15] Nelson, M.; Gailly, J., "The Data Compression Book, second edition", M&TBooks, New York, 1986, url: http://marknelson.us/1996/09/01/bwt/