EFFICIENT DATA COMPRESSION TECHNIQUE USING MODIFIED ADAPTIVE RICE GOLOMB CODING FOR WIRELESS SENSOR NETWORK

S. Kalaivani and C. Tharini
Department of Electronics and Communication, BSAU, India
E-Mail: skalaivani@bsauniv.ac.in

ABSTRACT

Wireless sensor networks (WSN) are energy constrained network since each node in WSNs are typically powered by batteries with limited capacity. Compressing the data sensed at each sensor node in an energy efficient manner is necessary for extending the network lifetime of wireless sensor network. In each sensor node the communication module is the main energy consuming unit and therefore data compression is one of the techniques that could be used to reduce the amount of data transmitted among the nodes. The proposed algorithm, Modified Adaptive Rice Golomb Coding (MARGC) is one such compression technique to prolong the life time of the network. Simulation results using different datasets demonstrate the feasibility and efficacy of MARGC algorithm. The algorithm has also been implemented in real time using NI WSN hardware.

Keywords: data compression, wireless sensor networks, rice golomb coding.

1. INTRODUCTION

Wireless Sensor networks are deployed with number of sensor nodes for enabling continuous monitoring in many fields and are suitable for large scale data gathering. WSN finds its application in domains like medicine, agriculture, industrial monitoring, environmental monitoring, structural monitoring and surveillance. Each sensor node transfers the collected data wirelessly to the sink either directly or through multihop communication [1]. Among different energy consuming process like sensing, processing and communication, the sensor node consumes more power during communication. There are number of algorithms developed and proposed for reducing the power consumption during transmission and reception. Designing energy efficient and a simple data compression method is a challenging issue.

Different solutions like energy conserving sleep scheduling [3], topology control [4], mobile data collectors [5], data aggregation [6] and compression techniques have been proposed to attack energy problems and conserve the energy of sensor nodes. These approaches focus on the energy efficient data collection, data aggregation and transmission of sensory data. Especially data aggregation aims at reducing the amount of data transmitted, but such mechanisms may not be applied when node redundancy is not available because of network deployment [7], [8] or sensor breakage. Compression is a solution with which the sensor nodes may regularly report their dataset that have been sensed for a long time to the remote sink [1].

Based on the recoverability, data compression can be classified into three categories: lossless, lossy and unrecoverable [9]. The original data can be perfecty recovered from the compressed data in lossless compression and lossy compression is one that allows data recovery from compressed data with some losses. An unrecoverable compression allows only compression operation and there is no decompression operation, for example transmitting only the average value of the collected dataset from which the original data cannot be reconstructed. In order to reduce the size of the data to be transmitted an efficient compression algorithm should be accomplished. MARGC algorithm aims to achieve this by exploiting temporal correlation in the data collected periodically from WSN applications.

2. RELATED WORK

Data compression scheme is classified into two main categories namely distributed data compression schemes and local data compression schemes [25]. The high spatial correlation of data in dense networks is used for distributed data compression. Some of the algorithms proposed under this approach are: distributed source coding (DSC) [10, 11], distributed transform coding (DTC) [12, 13], distributed source modeling (DSM) [14, 15], and compressed sensing (CS) [16]. But this scheme conserves energy at the expense of information loss in the source.

In local data compression scheme the temporal correlation of sensor data is taken into account to perform local compression in each sensor node. Both lossy and lossless compression algorithms have been proposed under this scheme. Lossy algorithms are: lightweight temporal compression (LTC) [19], K-RLE [21], DPCM-Optimization [24] and lossless algorithms are: Sensor - Lempel Ziv Welch (S-LZW) [20], Lossless Entropy Compression (LEC) [17], modified adaptive Huffman compression scheme [18], median-predictor-based data compression (MPDC) [22], and two-modal transmission (TMT) [23]. Most of these algorithms are computationally intensive and some also require adaptive dictionary structure that leads to growing dictionary problem and its compression efficiency also gets degraded.

The proposed MARGC algorithm can be used as lossy or lossless compression that could be adapted based on the application. Important properties of this algorithm

VOL. 10, NO. 12, JULY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

are it is very simple, adaptive and easy to use algorithm that could also be applied on non-positive values.

3. RICE GOLOMB CODING

Rice Golomb coding is a lossless data compression method and is a well-known compression algorithm for sensor data. A tunable parameter 'M' is used to divide an input value 'S' in Golomb coding that results into two parts: quotient 'Q', and the remainder 'R', where: S = Q × M + R. Golomb code encode non negative integer S by encoding (S/M)=Q as a unary value and (S mod M) =R as a binary value.

a) Encoding algorithm

Rice coding is fairly straightforward. M is an integer power of two say, M = 2^K. Then encoding is carried out for each input value 'S' as follows:

1. Writing S & (M - 1) in binary that represents R.
2. Writing S >> K in unary that represents Q.

b) Decoding algorithm

Decoding algorithm is also as simple as encoding. M is computed using the equation M = 2^K where the bit length K is obtained from the encoding. Then the following steps are carried out for each encoded symbol (S):

1. Q is determined by counting the number of 1s before the first 0.
2. R is determined by reading the next K bits as a binary value.
3. S is given by Q × M + R

4. MARGC ALGORITHM

The proposed MARGC algorithm can be used for coding/decoding both positive and negative dataset whereas the original Rice Golomb coding was designed to encode sequence of non-negative numbers. The sensor network that would periodically monitor the environmental condition is considered for applying this algorithm. The network is supposed to collect a set of data periodically and send it to sink node in an energy efficient manner by adopting MARGC algorithm. The algorithm can be used as lossy or lossless compression based on the need of application. Lossless compression is obtained by subjecting the dataset to MARGC encoding and decoding procedure. In order to reduce the number of bits during transmission the dataset is divided by 2 and this data is then encoded by MARGC algorithm. At the receiver end the decoded data is again multiplied by 2 to retrieve the dataset. Though this is a lossy compression technique, compression ratio is increased.

In both the cases root mean square error (RMSE) and compression ratio (CR) in % is calculated using equation (1) and (2),

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\left(input\ data(i) - output\ data(i)\right)^2}{2}} \quad (1)$$

$$CR = \left(1 - \frac{compressed\ data\ length}{uncompressed\ data\ length}\right) * 100$$

a) Encoding algorithm for negative dataset

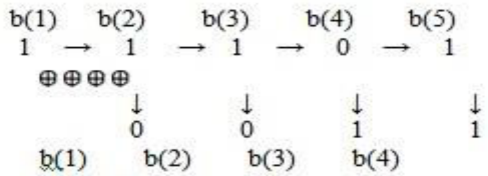
The Encoding algorithm for negative dataset using MARGC is as follows:

(i) Maximum negative number in the sequence is found and is added to all the data in the sequence. Thereby the negative sequence is converted to positive sequence.

(ii) The tunable parameter 'M' that divides the input data 'S' is made adaptive with respect to the data, such that 'Q' is made small.

(iii) Now the data is coded as per rice golombi.e. 'Q' in Unary and 'R' in binary, this results with a '0' in MSB bit followed by a '1' in the next bit for all the data in dataset while encoding.

(iv) The above operation allows us to perform bitwise-XOR for the encoded bits thereby reducing the number of encoded bits.
Example for bitwise-XOR on single operand:



(v) Now the variable length coded data is converted to a fixed length coded data in order to decode the data correctly at the decoder by appending zeros to the encoded data to match with the maximum length of the encoded data in the given sequence.

(vi) The encoded data will be in the format such that the first 3 bits represent the fixed length of encoded data and the first encoded data will be the maximum negative value in the sequence followed by the encoded data of converted positive sequence.

b) Decoding algorithm for negative dataset

The decoding algorithm for negative dataset using MARGC is as follows:

(i) At the receiver end, from the received encoded data fixed length of encoded data is determined from the first 3 bits.

(ii) The maximum negative value used for converting the negative sequence to positive sequence is found from the immediate encoded data.

(iii) The appended zeros are removed from the encoded data leaving a '1' in the MSB of each data and reverse of bitwise-XOR operation is performed.

(iv) The remaining K bit of binary value is used to determine 'R' value, M = 2^K and since 'K' is made

VOL. 10, NO. 12, JULY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

adaptive such that it results with ‘Q’ to be 0 for all the data in the sequence.

(v) This R, M & Q values can be used to determine the positive sequence from the expression S = Q × M + R.

(vi) The original negative dataset can be retrieved by subtracting the positive sequence with the determined maximum negative value in step (ii).

The pseudo-code of MARGC algorithm for negative sequence is shown in Figure-1 below:

```
\\ ENCODING

If S(N )<0
{
Max = Maximum (negative value(S));
SE = S+ Max; i=0;
for (j=1;j<(N+1);J++)
{
SE(j)= SE(i); ++i ;
}
SE(0) = Max; N= N+1;
for(i=0; i<N; i++)
{
for (K= 0; K<8; K++)
M=2^K;Q= Unary( S/M );
R= Binary(S mod M);
if  (Q==0 && R(K) == 1)
EN(i) = bitwise-XOR(concatenate{R,Q});
L(i) = Length (EN(i));
}
L = Max(L);i= 0;
do{
ENC(i)=concatenate{0,En(i)};
if (length(ENC(i)==L)
i = --i + 1;
else
i = --i;
} while(i==N-1);
ENCSEQ=   concatenate{binary3(L),ENC};\\encoded data
}end;
\\ DECODING
L = decimal(concatenate{ENCSEQ(0), ENCSEQ(1), ENCSEQ(2)});
i=0; j=3;
do{
ENC(i) = ENCSEQ(j);
–i;--j;
}while(i==N);
MAX = decimal(ENC(0));
for(i=0;i<N;i++)
{
ENC(i) = decimal(ENC(i));
ENC(i) = binary(ENC(i));
K=length(ENC(i));
R= decimal(Rbitwise-XOR(ENC(i));
M(i) = 2^K;Q(i)=0;
S(i) = (Q(i) * M(i))+ R(i);
}end;
```

Figure-1. Pseudo-code of MARGC algorithm for negative sequence.

c) Encoding and decoding algorithm for positive dataset

The steps involved in encoding positive dataset using MARGC algorithm is as follows:

(i) Average value of the given positive input sequence is found.

(ii) The difference of the sequence with the average value is taken.

(iii) Now the differenced sequence will have some negative values also and hence this differenced sequence could be encoded and decoded using the above algorithm.

(iv) The encoded data format is, the first 8 bit represents the average value found in step (i), next 3 bit representing the fixed length of each encoded data in the sequence followed by the encoded data with the maximum negative value of the differenced sequence being the first encoded data.

(v) The decoded data is then added with the average value found in the first step that can be determined from the encoded data.

The pseudo-code of MARGC algorithm for positive sequence is shown in Figure-2 below.

```
\\ ENCODING
If S(N )>0
{
AVG = average(S(N)); S(N) = S(N) – AVG;
Max = Maximum (negative value(S));
SE = S+ Max;
i=0;
for (j=1;j<(N+1);J++)
{
SE(j)= SE(i); ++i ;
}
SE(0) = Max; N= N+1;
for(i=0; i<N; i++)
{
for (K= 0; K<8; K++)
M=2^K;Q= Unary( S/M );
R= Binary(S mod M);
if  (Q==0 && R(K) == 1)
```

VOL. 10, NO. 12, JULY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences


```
EN(i) = bitwise-XOR(concatenate{R,Q});
L(i) = Length (EN(i));
}
L = Max(L);i= 0;
do{
ENC(i)=concatenate{0,En(i)};
if (length(ENC(i)==L)
i = --i + 1;
else
i = --i;
} while(i==N-1);
ENCSEQ = concatenate{binary8(AVG), binary3(L), ENC}; \\ ENCSEQ => Encoded data
}end;
\\ DECODING
AVG = decimal (ENCSEQ(0))
L = decimal(concatenate{ENCSEQ(1), ENCSEQ(2), ENCSEQ(3)});
i=0; j=4;
do{
ENC(i) = ENCSEQ(j);
–i;--j;
}while(i==N);
MAX = decimal(ENC(0));
for(i=0;i<N;i++)
{
ENC(i) = decimal(ENC(i));
ENC(i) = binary(ENC(i));
K=length(ENC(i));
R= decimal(Rbitwise-XOR(ENC(i));
M(i) = 2^K;Q(i)=0;
S(i) = ((Q(i) * M(i))+ R(i))+AVG;
} end;
```

Figure-2. Pseudo-code of MARGC algorithm for positive sequence.

5. SIMULATION RESULTS

Simulation results for two different dataset one with positive data alone and the other dataset with both positive & negative data obtained from [26] are given below. Each dataset with 144 temperature data collected for 6 days in hourly basis is considered for the simulation.

a) Simulation results for positive data sequence

The Lossless Encoding and Decoding simulation results for sequence with positive data are shown in Figure-3 below. Figure-3(a) shows the plot of the input temperature dataset considered and the decoded output is shown in Figure-3(b).

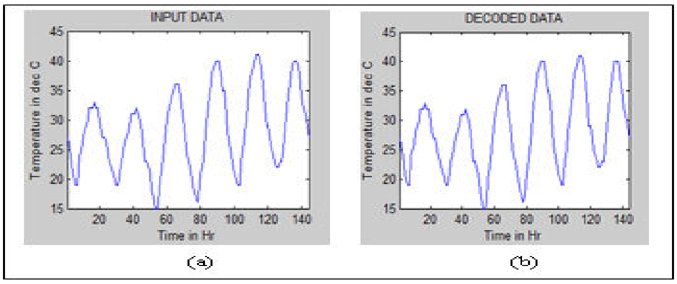


Figure-3. Lossless encoding and decoding (a) Positive sequence of input data (b) Decoded dataset after applying MARGC algorithm.

The RMSE and compression ratio of the above result are : RMSE = 0 since it is a lossless compression and CR (%) = 36.1%.

The Lossy Encoding and Decoding simulation results for positive data sequence are shown in Figure-4. Figure-4(a) shows the plot of the input temperature dataset considered. The considered data are all positive and lossy MARGC encoding was performed on the data. The decoded output is shown in Figure-4(b).

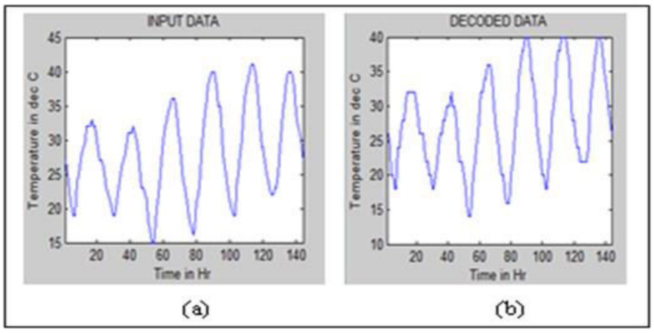


Figure-4. Lossy encoding and decoding (a) Positive sequence of input data (b) Decoded dataset after applying MARGC algorithm.

In lossy compression though there is an RMSE of 0.69, CR(%) is increased to 48.7%.

b) Simulation results for sequence with both positive & negative data

The Lossless Encoding and Decoding simulation results for sequence with both positive and negative data are shown in Figure-5 below. Figure-5(a) shows the plot of the input temperature dataset considered. The considered dataset consists of both positive and negative data. Lossless MARGC encoding proposed for negative data was performed on the data. The decoded output is shown in Figure-5(b).

VOL. 10, NO. 12, JULY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

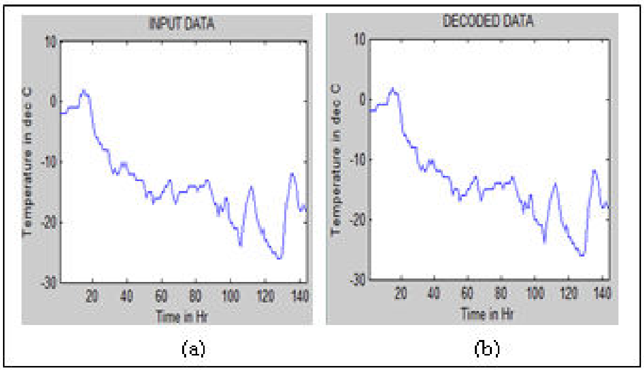


Figure-5. Lossless encoding and decoding (a) Input dataset with both positive and negative data (b) Decoded dataset after applying MARGC algorithm.

The RMSE and compression ratio of the above result are : RMSE = 0 since it is a lossless compression and CR (%) = 36.8%.

The Lossy Encoding and Decoding simulation results for sequence with both positive and negative data are shown in Figure-6. Figure-6(a) shows the plot of the input temperature dataset considered. The considered dataset consists of both positive and negative data. Lossy MARGC encoding proposed for negative data was performed on the data. The decoded output is shown in Figure-6(b).

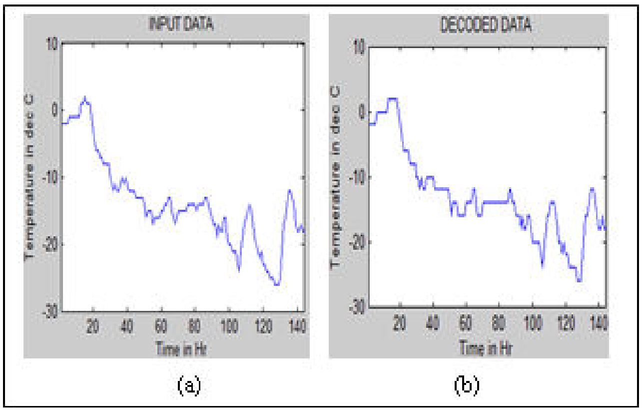


Figure-6 Lossy Encoding and Decoding (a) Input dataset with both positive and negative data (b) Decoded dataset after applying MARGC algorithm.

In lossy compression though there is an RMSE of 0.73, CR(%) is increased to 49.3%.

Lossless or lossy encoding/decoding is used based on the application requirements. The proposed algorithm is adaptive in nature, where the algorithm for positive sequence or sequence with negative data is adapted based on the input dataset.

Parameters like length of encoded data, Compression Ratio (CR) in % and RMSE are considered to analyse the efficiency of proposed alogorithm and the results are tabulated in Table-1. The actual length of encoded data before using MARGC algorithm is 1152 bits i.e. (8bits*144 data).

Table-1. Length of encoded data , compression ratio (cr) and rmse by applying margc algorithm.

Data sequence	**Type of encoding**	**Length of encoded data** (bits)	**CR** (%)	**RMSE**
Positive values	Lossless	736	36.1	0
	Lossy	591	48.7	0.69
Positive and negative values	Lossless	728	36.8	0
	Lossy	583	49.3	0.73

From Table-1 it is observed that the proposed algorithm gives better CR and reduced RMSE for the considered dataset.

6. EXPERIMENTAL SETUP

The MARGC algorithm is implemented in real time using NI WSN hardware. In our experimental setup the NI 3202 measurement node is used for sensing the temperature data and NI 9792 is used as the gateway as shown in Figure-7. The temperature of various laboratories in the department is measured using the NI measurement node and the link quality of the node with the gateway is also measured.

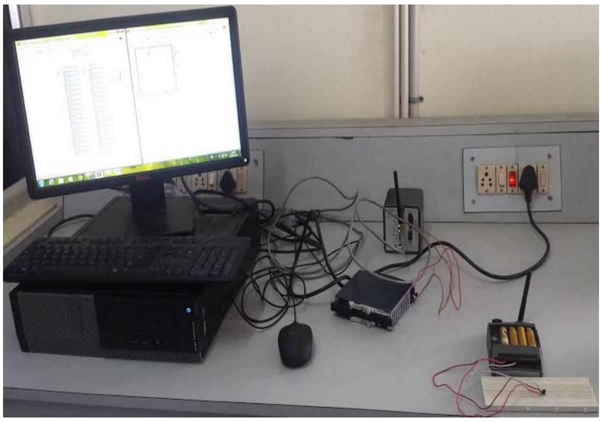
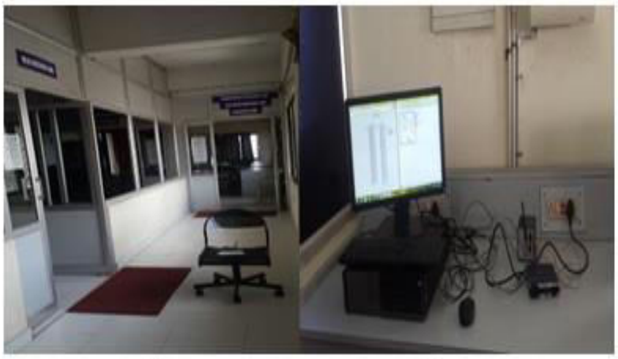


Figure-7. NI WSN hardware module.

The MARGC algorithm is implemented to one of the dataset obtained by placing the gateway node in the Networks laboratory and the measurement node in the VLSI laboratory with the link quality of 40% as shown in Figure-8.

VOL. 10, NO. 12, JULY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com



(a) (b)

Figure-8. Experimental setup with NI WSN modules (a) Measurement node in VLSI Lab (b) Gateway node in Networks Lab.

The acquired temperature data is compressed using MARGC algorithm in the NI measurement node. The algorithm is programmed in the node using LabVIEW software [27] and the compressed data is transmitted at 2.4GHz to the sink node where it is decompressed to get back the original data. The results of the MARGC algorithm implementation in NI WSN module is shown in Figure-9.

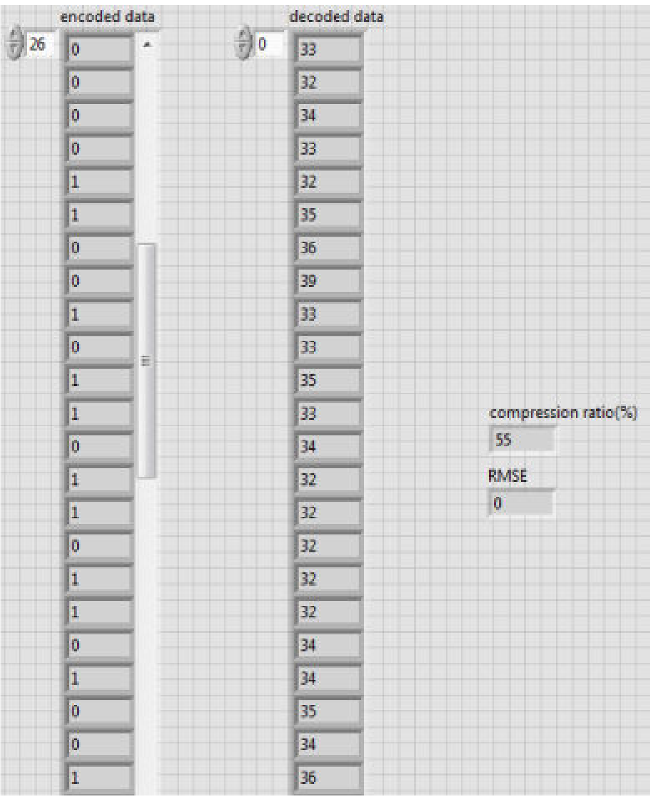


Figure-9. Real time implementation result using NI WSN hardware module.

Thus the real time implementation of MARGC algorithm results with the compression ratio of 55% and the RMSE is 0 since the algorithm implemented is a lossless one.

7. CONCLUSIONS

In this paper, a modified adaptive Rice Golomb coding is used for both lossy and lossless compression of positive/negative data sequence for wireless sensor network. The proposed MARGC algorithm is efficient, simple and is also suitable for resource-constrained wireless sensor nodes. The algorithm reduces the amount of data being sent from the source node to the sink node that in turn reduces the energy spent during transmission and thereby increase the lifetime of the network. Additionally, the proposed algorithm can be used in monitoring systems with different types of data and provides satisfactory compression ratios. The algorithm is also implemented in real time using NI WSN hardware. As a future work the performance of this algorithm will be tested for biomedical signals.

REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci. 2002. "A survey on sensor networks," IEEE Communications Magazine, Vol. 40, No. 8, pp. 102-114, August.

[2] J. Yick, B. Mukherjee and D. Ghosal. 2008. "Wireless sensor network survey," Computer Networks, Vol. 52, No. 12, pp. 2292–2330.

[3] R. Subramanian and F. Fekri. 2006. "Sleep Scheduling and Lifetime Maximization in Sensor Networks: Fundamental Limits and Optimal Solutions," in Proceedings of the IEEE/ACM Fifth International Conference on Information Processing in Sensor Networks (IPSN 2006), Nashville, TN, pp. 218–225, April.

[4] X.Y. Li, W.Z. Song and W. Wang. 2005. "A Unified Energy-Efficient Topology for Unicast and Broadcast," in Proceedings of the 11th annual international conference on Mobile computing and networking.

[5] G. Xing, T. Wang, W. Jia and M. Li. 2008. "Rendezvous Design Algorithms for Wireless Sensor Networks with a Mobile Base Station," in Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing.

[6] S. He, J. Chen, D. Yau and Y. Sun. 2010. "Cross-layer Optimization of Correlated Data Gathering in Wireless Sensor Networks," in Proceedings of the 7th Annual IEEE Communications Society Conference on Sensor Mesh and Ad Hoc Communications and Networks (SECON).

[7] N. Heo and P.K. Varshney. 2005. "Energy-efficient deployment of intelligent mobile sensor network," IEEE Trans. Systems, Man and

cyberneticPart A: Systems and Humans, Vol. 35, No. 1, pp. 78-92, January.

[8] Y.C Wang, C.C. Hu and Y.C. Tseng. 2008. "Efficient placement and dispatch of sensors in a wireless sensor network," IEEE Transactions on Mobile Computing, Vol.7, No. 2, pp. 262-274, February.

[9] You-Chiun Wang. 2012. "Data Compression Techniques in Wireless Sensor Networks," Pervasive Computing, Nova Science Publishers.

[10] S. Pradhan, J. Kusuma and K. Ramchandran. 2002. "Distributed compression in a dense microsensor network," IEEE Signal Processing Magazine, vol. 19, no. 2, pp. 51–60.

[11] J. Chou, D. Petrovic and K. Ramchandran. 2003. "A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks," in: Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies, pp. 1054–1062, April 2003.

[12] Ciancio and A. Ortega. 2005. "A distributed wavelet compression algorithm for wireless multihop sensor networks using lifting," in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05), Vol. 4, pp. IV825–IV828, March.

[13] Ciancio, S. Pattem, A. Ortega and B. Krishnamachari. 2006. "Energy-efficient data representation and routing for wireless sensor networks based on a distributed wavelet compression algorithm," in: Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN '06), pp. 309–316, April.

[14] J. J. Xiao, S. Cui, Z. Q. Luo and A. J. Goldsmith. 2006. "Power scheduling of universal decentralized estimation in sensor networks," IEEE Transactions on Signal Processing, Vol. 54, No. 2, pp. 413–422.

[15] J. B. Predd, S. R. Kulkarni and H. V. Poor. 2006. "Distributed learning in wireless sensor networks," IEEE Signal Processing Magazine, Vol. 23, No. 4, pp. 56–69, 2006.

[16] D. L. Donoho. 2006. "Compressed sensing," IEEE Transactions on Information Theory, Vol. 52, No. 4, pp. 1289–1306.

[17] F. Marcelloni and M. Vecchio. 2009. "An efficient lossless compression algorithm for tiny nodes of monitoring wireless sensor networks," Computer Journal, Vol. 52, No.8, pp.969 – 987.

[18] C. Tharini and P. Vanaja Ranjan. 2009. "Design of modified adaptive Huffman data compression algorithm for wireless sensor network," Journal of Computer Science, Vol. 5, No. 6, pp. 466– 470.

[19] T. Schoellhammer, E. Osterweil, B. Greenstein, M. Wimbrow and D. Estrin. 2004. "Lightweight temporal compression of micro- climate datasets," in: Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN '04), pp. 516–524, November.

[20] M. Sadler and M. Martonosi. 2006. "Data compression algorithm for energy-constrained devices in delay tolerant network," in: Proceedings of the 4th international Conference on Embedded Networked Sensor Systems (SenSys '06), pp. 265–278, November.

[21] E. P. Capo-Chichi, H. Guyennet and J. M. Friedt. 2009. "K-RLE: a new data compression algorithm for wireless sensor network," in Proceedings of the 3rd International Conference on Sensor Technologies and Applications, (SENSORCOMM'09), pp.502– 507, June.

[22] K. Maurya, D. Singh and A. K. Sarje. 2011. "Median predic- tor based data compression algorithm for Wireless Sensor Network," International Journal of Smart Sensors and Ad Hoc Networks, Vol. 1, No. 1, pp. 62–65, 2011.

[23] Y. Liang and W. Peng. 2010. "Minimizing energy consumptions in Wireless Sensor Networks via two-modal transmission," Computer Communication Review, Vol. 40, No. 1, pp. 13–18.

[24] F. Marcelloni and M. Vecchio. 2010. "Enabling energy-efficient and lossy-aware data compression in wireless sensor networks by multi-objective evolutionary optimization," Information Sciences, Vol. 180, No. 10, pp. 1924–1941.

[25] Jonathan Gana Kolo, S. Anandan Shanmugam, David Wee Gin Lim, Li-MinnAng and KahPoohiSeng. 2012. "An Adaptive Lossless Data Compression Scheme for Wireless Sensor Networks", Journal of Sensors, Hindawi Publishing, Vol. 2012, Article ID 539638.

[26] http://www.climate.weather.gc.ca/

[27] http://sine.ni.com/nips/cds/view/p/lang/en/nid/2069 15

[28] http://www.ni.com/white-paper/11529/en/