
# Analysis and implementation of novel Rice Golomb coding algorithm for wireless sensor networks

S. Kalaivani, C. Tharini

PII: S0140-3664(19)30839-4
DOI: https://doi.org/10.1016/j.comcom.2019.11.046
Reference: COMCOM 6048

To appear in: *Computer Communications*

Received date : 22 July 2019
Revised date : 9 November 2019
Accepted date : 27 November 2019

# Analysis and Implementation of Novel Rice Golomb Coding Algorithm for Wireless Sensor Networks

S. Kalaivani [1], C. Tharini [2]
[1] *Research Scholar,* [2] *Professor*
[1,2]*Department of Electronics and Communication Engineering,*
*Crescent Institute of Science and Technology, Chennai, Tamil Nadu, India,*
*skalaivani@crescent.education, tharini@crescent.education*

***Abstract —*** Wireless sensor networks (WSN) comprises of several sensor nodes scattered wirelessly to accomplish a particular task. Each sensor node is empowered by a battery. The various functions of the node namely sensing, computing, storage and transmission/reception of data consumes power from the battery with limited capacity. As these batteries do not last for a long time, an efficient algorithm is required to extend its life time. Data compression algorithm is a unique method adopted to minimize the amount of data being sent or received and thereby reduces the power consumed during communication. This would further increase the lifetime of node and also the network. In this paper a simple lossless compression algorithm is proposed and is also compared with the existing Adaptive Huffman coding algorithm that is been widely used in wireless sensor network applications. The comparative analysis is based on different compression parameters like compression ratio, compression factor, saving percentage, RMSE and encoding & decoding time. The data set for comparison is acquired using a temperature sensor interfaced with NI 3202 programmable sensing node. The comparative analysis is performed and the results are simulated using MATLAB software. The NI WSN nodes are used to execute the algorithm for instantaneous data. The analysis of number of packets transmitted during wireless communication, both before and after compression is performed using Wireshark network analyzer tool. The simulation result shows that the proposed lossless compression algorithm performs better than the existing one. The hardware implementation has proven that the amount of data traffic is reduced after compression which will help in reducing the transmission power and thereby saves the lifetime of the node in a wireless sensor network.

**Keywords —** *Data compression, Huffman coding, Rice Golomb coding, Wireless sensor networks, Network Analyzer***.**

## I. INTRODUCTION

Now a day's all new technologies evolving in the field of embedded and communication is mainly based on Wireless Sensor Network. Wireless sensor network consists of sensor nodes that gather data from different sources and transmit them to a sink node depending on the applications. They are widely used in many applications namely environmental monitoring, habitat monitoring, industrial monitoring, agriculture, medical field, battlefield surveillance and target tracking [1]. The sensor node also called as source node consists of various modules to acquire the sensed data, process it and transfer it to the base station. All the nodes in the network are powered by a limited capacity battery. A typical schematic diagram of a wireless sensor network is shown in Fig.1.

The key problem in wireless sensor network is the power utilization by each node and is also the main aim to be solved as it directly has a strong effect on the functioning period of the network. There are various processes by which the power is utilized in a sensor node like the power consumed by the processor for data processing, power used for memory storage and communication power. Due to limited power supply facility for each node there is a possibility of the power drain that may lead to the death of the nodes in turn the sensor network. So it is highly required to limit the power consumption and save power for the extended lifetime of the network. From the study it is observed that over 80% of the energy is consumed by the communication module [2]. Data Compression that decreases the number of data transmitted from the source node to the sink node is one among the various techniques recommended to overcome this critical problem [3]. Data compression helps to improve the efficiency of utilizing the power supply, memory and bandwidth of the channel [4], [5].

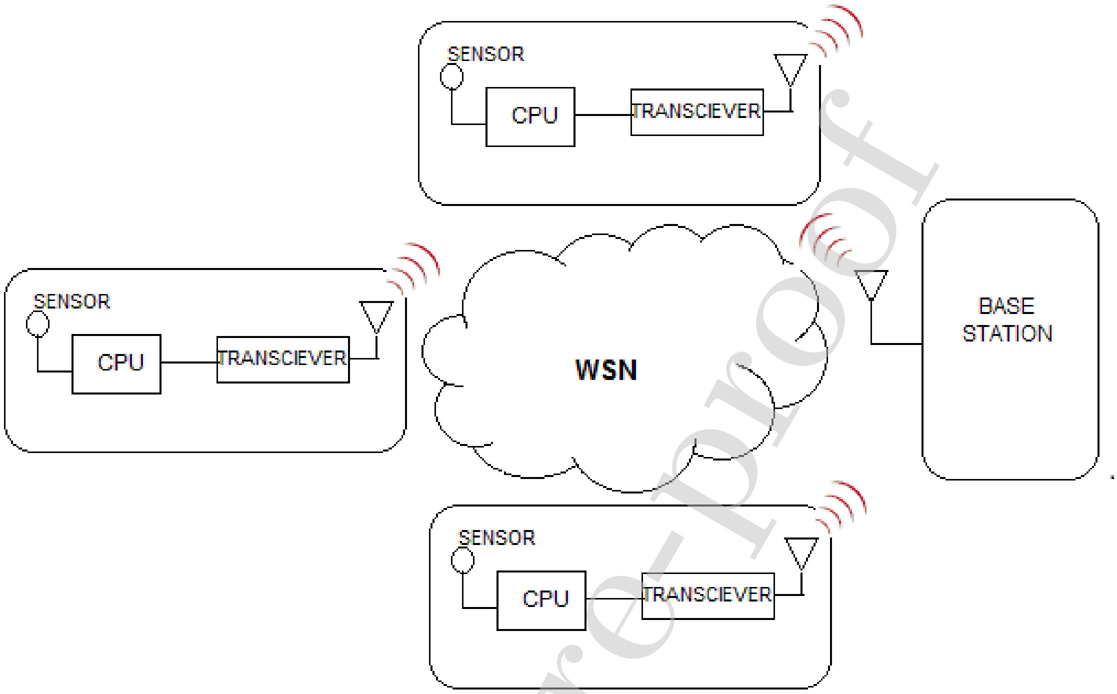



Figure 1. Schematic diagram of a WSN

Data compression technique is classified into three types namely: lossless, lossy and unrecoverable data compression [6]. Lossless compression is used to retrieve the original input data from the compressed data using decompression with no loss in data, lossy compression results with the input data recovery by decompression with some losses and unrecoverable compression is where no decompression is performed. Depending on the application a suitable type of compression technique can be considered. The proposed algorithm is based on Rice Golomb coding, a simple lossless compression algorithm.

## II. Related Work

In wireless sensor network multiple sensor nodes actively collect the data with respect to an event are highly correlated. Data compression has received attention in the field of WSN to minimize the power utilized for transferring the huge amount of highly correlated data to the sink node. Many data compression algorithms have been proposed for various applications [7], [8]. Among the proposed compression algorithms such as Distributed source coding (DSC) based on the correlation of sensor outputs [9],[10]; distributed transform coding (DTC) need extra power to determine the coefficients by inter node [11]; and compressed sensing (CS) have limitations like problem to identify sparsity requirements, ensuring of local communication and decoding complexity [12], [13]. But these methods are robust to data losses and do not guarantee optimal energy savings [14]. Lossy compression algorithm such as lightweight temporal compression (LTC), K-RLE a form of lossy run length coding and DPCM have the flexibility to tune compression ratio but with some loss in the original data [15]. Lempel Ziv Welch (LZW), Sequential Lossless Entropy Compression (S-LEC) [16], [17], Adaptive Huffman compression scheme [18], Median-Predictor-based Data compression (MPDC) are lossless compression algorithms [19]. These algorithms have computational complexity and also growing dictionary problems that would degrade the compression efficiency.

The proposed algorithm is based on the simple and lossless data compression known as Rice Golomb coding (RGC). Due to minimum memory and low computational requirements Rice Golomb codes are widely used in many applications of WSN [20], [21]. The proposed work aims at comparing the modified Rice Golomb compression code with Adaptive Huffman coding algorithm [22], another existing and equally used compression in WSN. The proposed algorithm and Adaptive Huffman coding algorithm are applied on a temperature dataset for the analysis using MATLAB. The various compression parameters considered are: compression ratio (CR), compression factor (CF), saving percentage (SP), RMSE, encoding and decoding time. The comparison result shows that the proposed algorithm performs better than the Adaptive Huffman coding in all the parameters that were considered for performance analysis.

## III. ADAPTIVE HUFFMAN CODING

Huffman coding is a lossless coding that encodes each source symbol with variable encoding length based on the probabilities of its occurrence. The clear understanding of the source sequence is requisite for Huffman coding. If not, then the Huffman coding has to go with the following two steps [23]:

(i) Collecting the statistics about the sequence

(ii) Encoding the source symbol.

Huffman coding makes use of shorter codeword to encode very often occurring symbols and longer codeword for infrequent symbols [24]. Huffman coding can be constructed using a binary tree where the source sequence is encoded with respect to the probability of occurrence from the root node traversing towards the leaf nodes representing 0 & 1. The basic Huffman coding tree is shown in Fig.2.

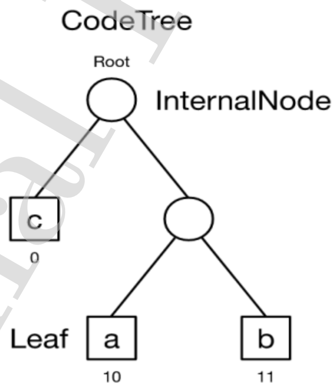



Figure 2. Huffman Code tree

The Huffman decoding begins with the root node to decode the sequence for which the node parameters are same as that of the encoder. Therefore both the transmitter and the receiver should be synchronized. Moreover this is not a good approach for encoding a huge amount of data i.e. encoding $(N+1)^{th}$ symbol based on the first symbol in WSN applications. This leads for the use of adaptive Huffman coding for WSN applications.

Adaptive Huffman coding is also called as dynamic Huffman coding where the probability of the incoming data is changed dynamically through the tree construction. The tree in the encoding and decoding procedure of the algorithm is initialized with a node known as not yet transmitted node (NYT). The number of this node is assigned as $2N - 1$ where N is the total number of alphabets [25]. If the alphabets are considered for encoding then $N = 26$. To obtain the prearranged code the parameters **e** and **r** are to be determined such that $2^e + r = 26$ where $0 \le r \le 2^e$. Therefore the values are **e** = 4 and **r** = 10 that satisfies the condition.

The encoding procedure for the adaptive Huffman coding is as follows:

1. Start building the tree to encode the symbols.
2. Read the symbol.
3. If the symbol occurs for the first time then send the codeword to the NYT node followed by the index in the NYT list.
   Go to Step 5.
4. If the symbol has occurred already then the codeword is the path from root node to the corresponding node.
5. Update the encoding tree. The update procedure is to maintain the sibling property so that the encoder and decoder work with the same information. According to sibling property the weight assigned to the nodes from left to right root till the top is in descending order as shown in Fig.3.

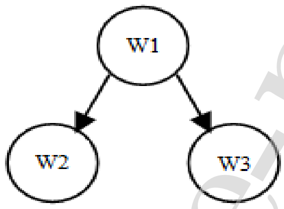



Figure 3. Code tree representing sibling property

6. If the symbol processed is the last one to be encoded then stop the encoding procedure else go to step 2.

The decoding procedure for the adaptive Huffman coding is as follows:

1. Start the decoding procedure from the root of the tree.
2. Read the binary bit until the leaf is reached.
3. If the leaf reached is a NYT node then read **e** bits, checks the respective value of the bits and let it be **y**.
4. If **y** < **r** then read one more bit and decode the symbol.
5. Else if **y** > **r** then add **r** to **y** and decode the symbol.
6. Once the symbol is decoded continue with update procedure.
7. In step 2, if the leaf node reached is not a NYT node then decode the corresponding leaf node and continue with the update procedure.
8. If the processed bit is the last one then stop the procedure for decoding else go back to read the bits from the root node.

From the procedure of encoding and decoding using adaptive Huffman coding it is observed that the transmitter or the receiver is not concerned with the analysis of the data sequence at the beginning of communication. This makes adaptive Huffman coding suitable for encoding the real time data of WSN that are highly correlated [23]. When adaptive Huffman coding is used to compress large scale of data by dynamic allocation of probabilities, the number of bits required increases. Also construction of binary tree becomes complex when used for sensor data. Thereby, both the types of Huffman algorithms are bounded with their own drawbacks.

## IV. RICE GOLOMB CODING

Rice Golomb coding (RGC) is a simple variable length coding [26]. Let Qn & Rd be the quotient and remainder on dividing an input value 'I' by 'D'. The input data is encoded using RGC by concatenating Qn and Rd in binary format. The initial step in RGC is to determine the optimal value of k where D = 2^k such that when an input data is divided by D it results with a minimum encoding length. The encoding and decoding procedure of a conventional RGC is as follows:

### *A. Encoding Algorithm*

1. Rd = Binary [I & (D - 1)].
2. Qn =Unary (I >> k).
3. Encoded data = {Qn,Rd}.

### *B. Decoding Algorithm*

1. The number of bits after the first zero in the compressed data is the tunable parameter k. Then D = 2^k.
2. Qn = Number of 1s before the first 0.
3. Rd = Binary value (the next k bits).
4. I= Qn × D + Rd.

As an example when RGC is applied for the number 11 represented in 8 bits (00001011) and for D = 4 where k = 2 then the encoding is as follows:

1. Rd = (00001011 & 011) = 11
2. Qn = Unary (000010) = 110
3. Encoded data = 11011

The encoded data results with 5 bits of encoding length, thereby 8 bit input data is compressed to 5 bits. But while applying conventional RGC to a sequence of data transmission, it requires additional bits that represent 'k' value to be sent along with each compressed data so as to decode the correct data at the receiver end. This result with more number of bits transmitted than the uncompressed data. Thereby the conventional RGC was modified to suit data transmission in WSN.

In the existing method the 'k' value for the maximum/average value of the input series is determined and is used for encoding all the data in the sequence resulting in variable length coding [27], [28]. In Modified Adaptive Rice Golomb Coding (MARGC), instead of using a constant 'k' values for all data in the sequence, adaptive 'k' value for each data is used but the encoding length of each data is kept constant resulting in fixed length coding [29]. The proposed algorithm is based on MARGC encoding and decoding algorithm.

## V. PROPOSED ALGORITHM

The proposed algorithm is a variable length compression algorithm. The proposed algorithm is suitable for WSN applications where the data communicated between the sensor nodes and gateway node are highly auto-correlated. The algorithm makes use of the redundancy or correlation of data in the sequence to reduce the encoding length whereby it encodes the auto-differenced sequence of the input data that will result in smaller encoding length. The desired auto-correlation value of the dataset is more than 0.5 for which the algorithm provides a satisfactory result on compression ratio/ saving percentage. The encoding and decoding procedure of the proposed algorithm is given below.

The steps involved in the encoding procedure:

1. Get the input data sequence.
2. Take auto-differencing of the input data sequence.

3. The auto-differenced sequence consists of some negative values; it is then converted to a positive sequence using the DC shift with the DC value as the first data.
4. Find k for the maximum probability of occurrence of a data.
5. Apply RGC encoding procedure for each data in the positive sequence obtained in step 2 using the same k value found in step 4.
6. The encoding format of the data sequence is such that the first 8 bit of the encoded data represents the binary form of first data in the input sequence followed by 3 bit representation of k value and the variable length encoded bits for each data.

Steps involved in the decoding process:

1. The decoding procedure starts with identifying the first data value of the original input sequence from the first eight bits.
2. Find the 'k' value from the next three bits of the encoded sequence.
3. Now with the 'k' value the positive differenced data are decoded using the RGC decoding procedure.
4. The decoded sequence will have the DC value as the first data from which the auto-differenced sequence at the transmitter end is found.
5. With the help of first decoded data in step 1 the original data sequence is retrieved.

The flowchart of encoding and decoding procedure for the proposed algorithm is shown in the Fig. 4.

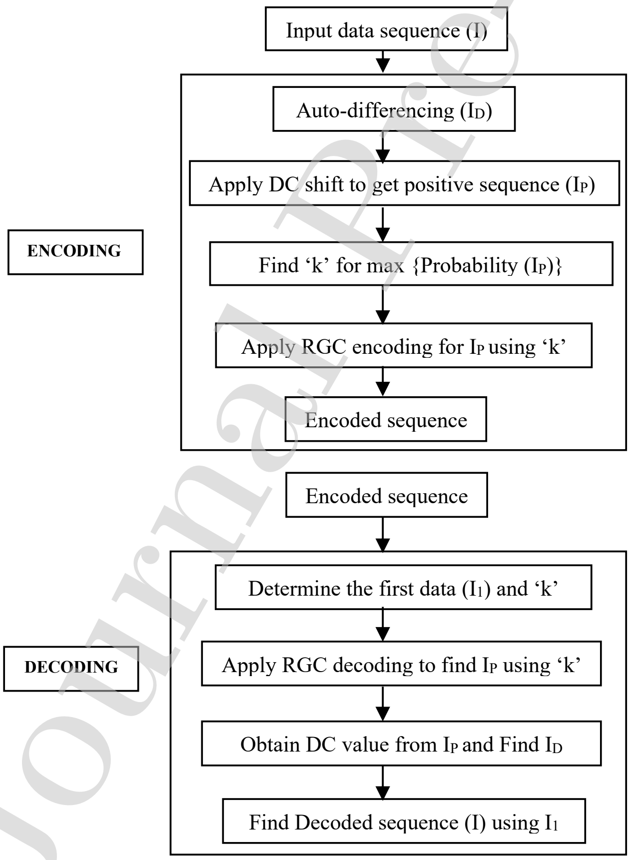



Figure 4. Flow chart of the proposed algorithm

As the procedure for proposed algorithm is framed with simple logic the time consumed by the data processing node for computation is less.

## VI. Performance Evaluation Parameters

There are number of ways to evaluate the performance of the algorithm. The performance of a compression algorithm is evaluated by measuring the complexity of the algorithm, time taken to execute the algorithm in a given machine, amount of compression obtained, memory space required for the compressed data and the originality of the decompressed data.

The performance of the proposed algorithm is evaluated using the following parameters [30]:

1. Length of encoded data (bits)
2. Compression Ratio (CR)
3. Compression Factor (CF)
4. Saving Percentage (SP)
5. RMSE (Root Mean Square Error)
6. Encoding and Decoding time

### *A. Length of Encoded data:*

It refers to the data length after the encoding process i.e. length of the compressed data in bits.

### *B. Compression Ratio:*

It is the ratio of input data length before compression to the output data length after compression.

$$\text{Compression ratio} = \frac{\text{Uncompressed data length}}{\text{Compressed data length}} \quad (1)$$

### *C. Compression Factor:*

It is the ratio of output data length after compression to the input data length before compression.

$$\text{Compression Factor} = \frac{\text{Compressed data length}}{\text{Uncompressed data length}} \quad (2)$$

### *D. Saving Percentage:*

It denotes the memory space saved in percent due to compression.

$$SP = 1 - \frac{(\text{Compressed data length})}{(\text{Uncompressed data length})} \times 100 \quad (3)$$

### *E. Root mean square error:*

It is used to evaluate the deviation between the decoded value and the input value. For a lossless compression RMSE is 0.

$$\text{RMSE} = \sqrt{\frac{(\sum_{i=1}^{N} (\text{input data (i)} - \text{output data (i)})^2}{2}} \quad (4)$$

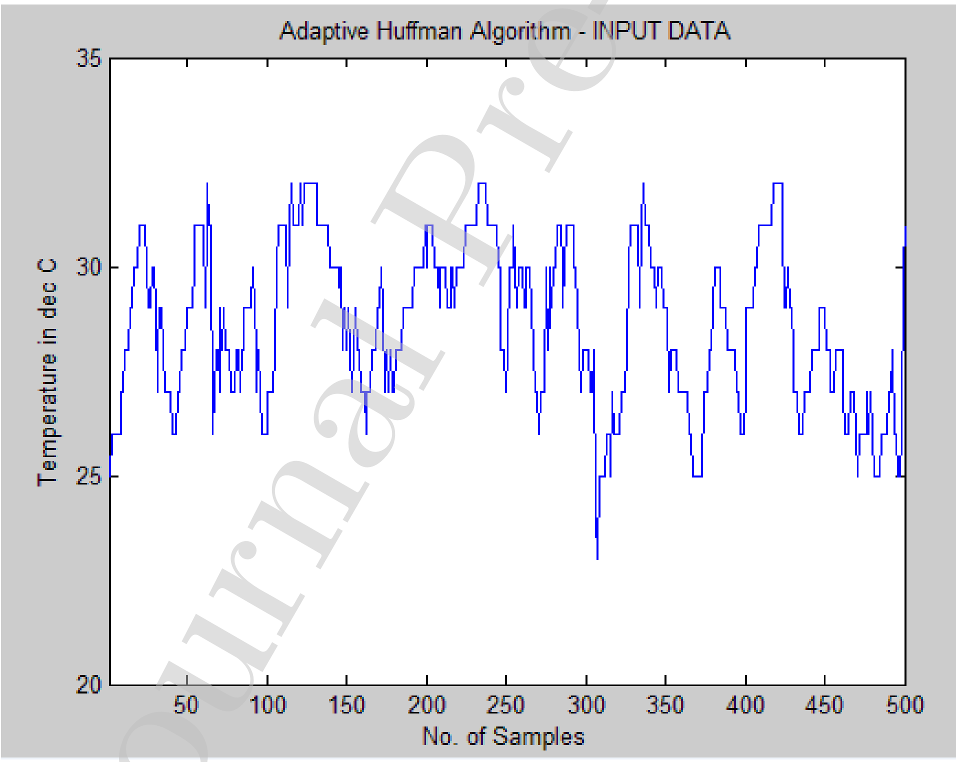
### *F. Encoding and Decoding Time:*

It refers to the computation time that denotes the duration for encoding and decoding the data sequence.

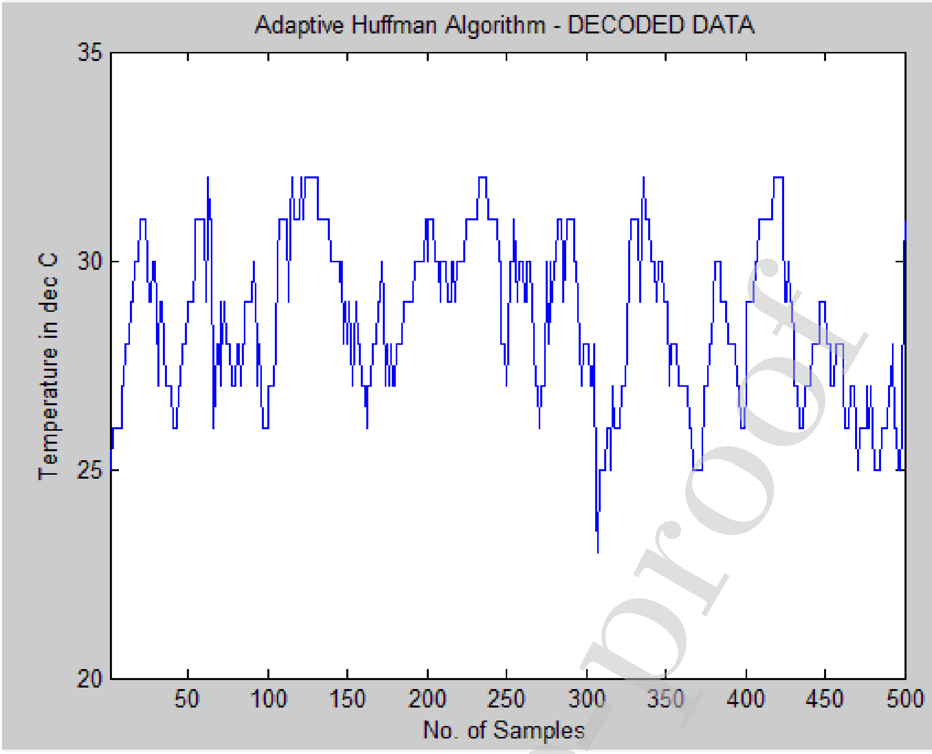
## VII. SIMULATION RESULTS

The performance of the proposed algorithm is compared with the Adaptive Huffman lossless compression algorithm that is widely used in wireless sensor networks. Adaptive Huffman coding is one of the popular and commonly used lossless coding techniques for almost all data types [31]. Also recent research combines Huffman codes with other compression algorithms to improve the data compression performance [32], [33]. Both the algorithms are applied on the temperature dataset (500 data) acquired by interfacing the temperature sensor with NI 3202 sensing node [34]. The program to acquire the room temperature for every 5 minutes is written in LabVIEW [35] and thereby a dataset of 500 data is collected. The acquired data that is highly correlated with the auto-correlation value of 0.911 is exported to an excel sheet which is then used to implement the proposed algorithm. The existing and proposed algorithm is simulated using MATLAB and the simulation results are shown below. As the number of data considered is 500 and if 8 bit is used to represent each data, then the encoded length of data before implementing the algorithm is 4000 bits. Fig.5. shown below represents the simulation result on applying the adaptive Huffman coding algorithm on the dataset.

The simulation result shown below in Fig.5(a) represents the input temperature data (500 data) considered before applying the Adaptive Huffman coding and 5(b) represents the decoded temperature data (500 data) obtained after applying the Adaptive Huffman decoding algorithm. As adaptive Huffman coding is a lossless compression algorithm, both the input data and the decoded data are same. Fig.5. represents the simulation result on applying the proposed algorithm on the same temperature dataset.



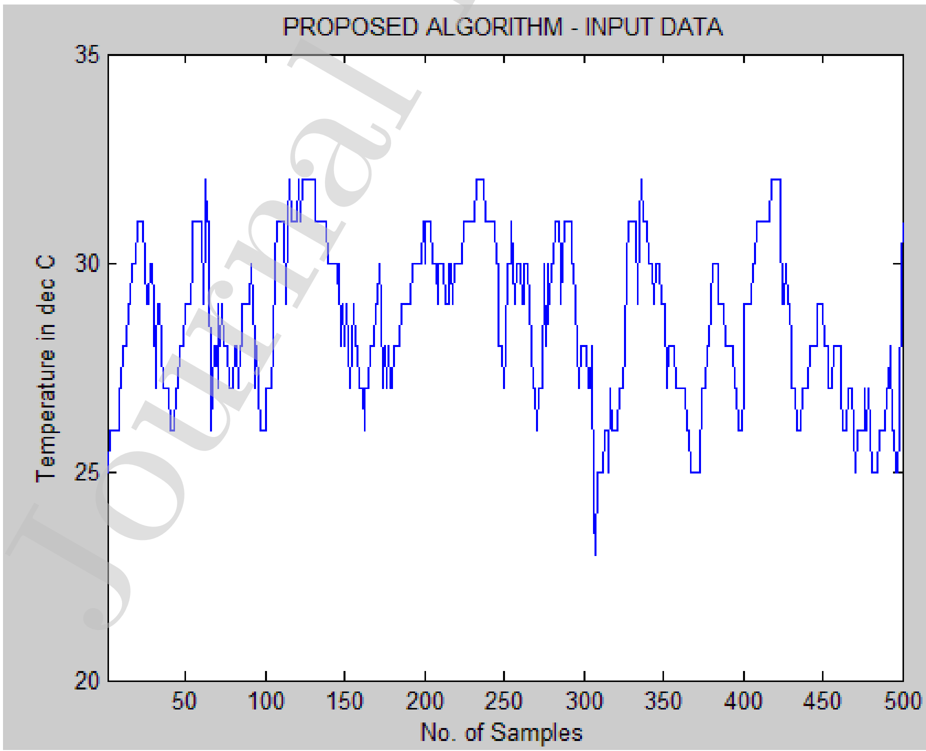

(a) Input data Plot



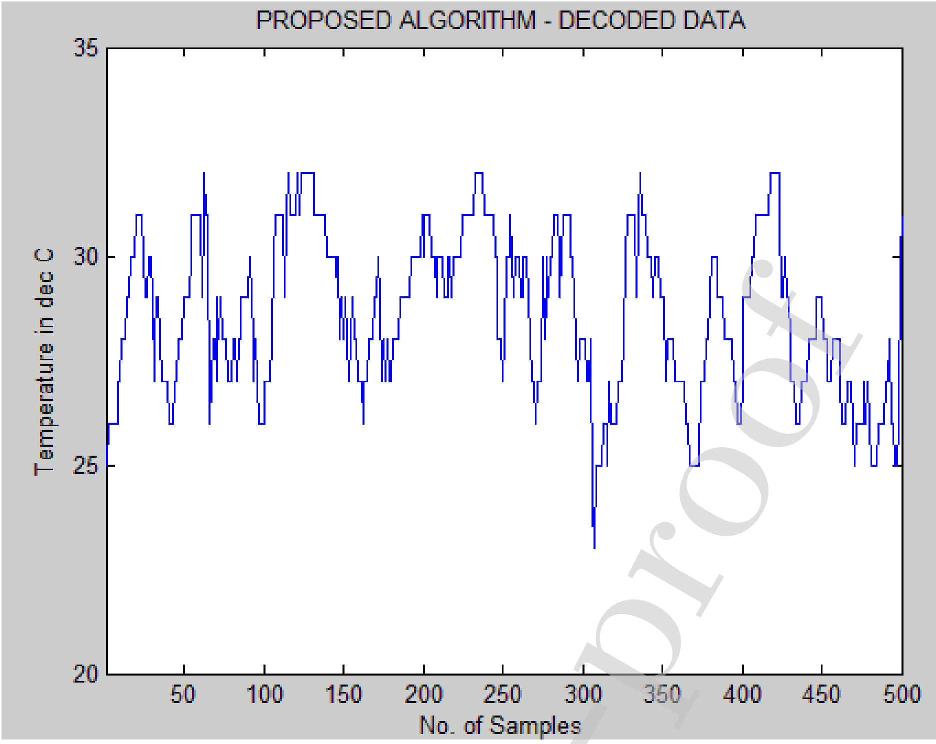

(b) Decoded data Plot

Figure 5. Simulation results of Adaptive Huffman coding algorithm

In the simulation result shown below Fig.6(a) represents the input temperature data (500 data) considered before applying the proposed encoding algorithm and 6(b) represents the decoded temperature data (500 data) obtained after applying the proposed decoding algorithm.




(a) Input data Plot




(b) Decoded data Plot

Figure 6. Simulation results of proposed algorithm

The input data length considered for analysis is 4000 bits, it is reduced to 2411 bits on applying Adaptive Huffman coding and it is found that the data length reduction of about 2102 bits is obtained when the input data is subjected to the proposed compression algorithm. The simulation output on applying Adaptive Huffman coding results with the compression ratio (CR) of 1.659, compression factor (CF) of 0.603, saving percentage (SP %) of 39.7, RMSE as 0, encoding time as 1.95 seconds and decoding time of 1.25 seconds. Similarly the outcome of the simulation on using the proposed compression algorithm results with the compression ratio (CR) of 1.903, compression factor (CF) of 0.526, saving percentage (SP %) of 47.45, RMSE as 0, encoding time as 1.70 seconds and decoding time of 0.03 seconds. The input data is also tested with the existing Rice Golomb Coding (RGC) method [36] that results with the following values: Encoding length of 3027 bits, CR of 1.321, CF of 0.757, SP% of 24.325, RMSE as 0, encoding time as 0.41 seconds and decoding time of 0.11 seconds. The performance analysis on applying the proposed algorithm, adaptive Huffman coding algorithm and the existing Rice Golomb coding on the temperature dataset with respect to different parameters is tabulated in Table I given below.

TABLE. I RESULTS ON COMPARISON OF PROPOSED ALGORITHM WITH ADAPTIVE HUFFMAN CODING AND EXISTING RGC

| **Parameters** | **Proposed Algorithm** | **Adaptive Huffman Coding** | **Existing RGC** |
|---|---|---|---|
| Length of encoded data (bits) | 2102 | 2411 | 3027 |
| CR | 1.903 | 1.659 | 1.321 |
| CF | 0.526 | 0.603 | 0.757 |
| SP (%) | 47.45 | 39.7 | 24.325 |
| RMSE | 0 | 0 | 0 |
| Encoding Time (sec) | 1.70 | 1.95 | 0.41 |
| Decoding Time (sec) | 0.03 | 1.25 | 0.11 |

From the above table it is observed that the proposed lossless data compression algorithm performs better with the compression ratio of 1.903 and saving percentage of 47.45%. The computation time i.e. the time taken for compression and decompression obtained by using the TIC & TOC functions in MATLAB is also less for the proposed algorithm by a factor of 1.47 seconds compared to the adaptive Huffman coding algorithm.

## VIII. Implementation and Analysis

The real time execution of the proposed algorithm is implemented using NI WSN [37] setup. The experimental setup shown in Fig.7 consists of a temperature sensor LM35, NI WSN 3202 a programmable sensing node, NI WSN 9792 a programmable gateway node [38] and a PC installed with software namely LabVIEW 2013 & Wireshark network analyzer tool.

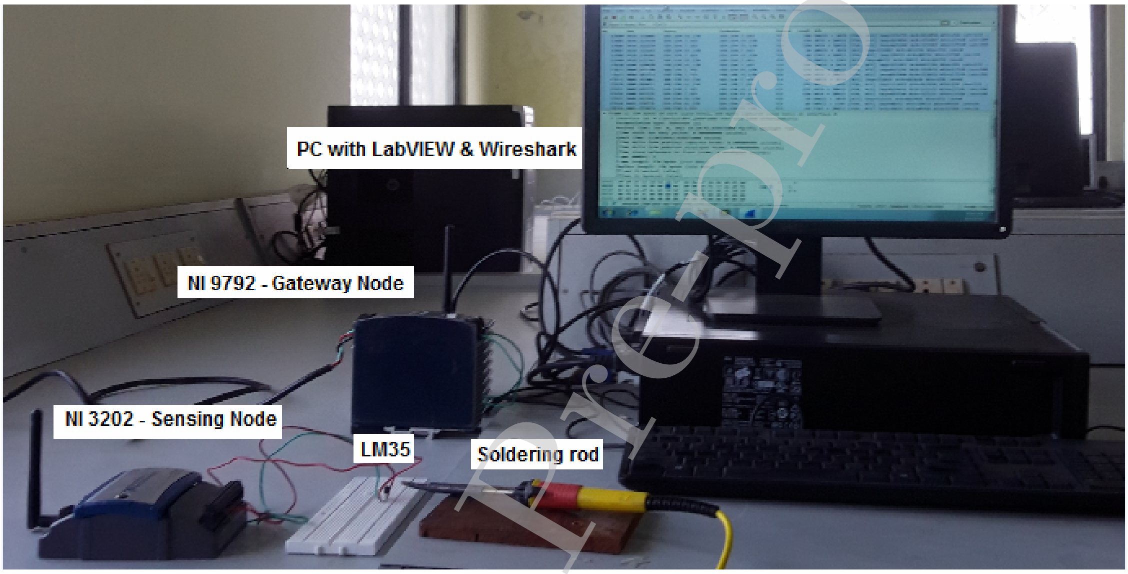


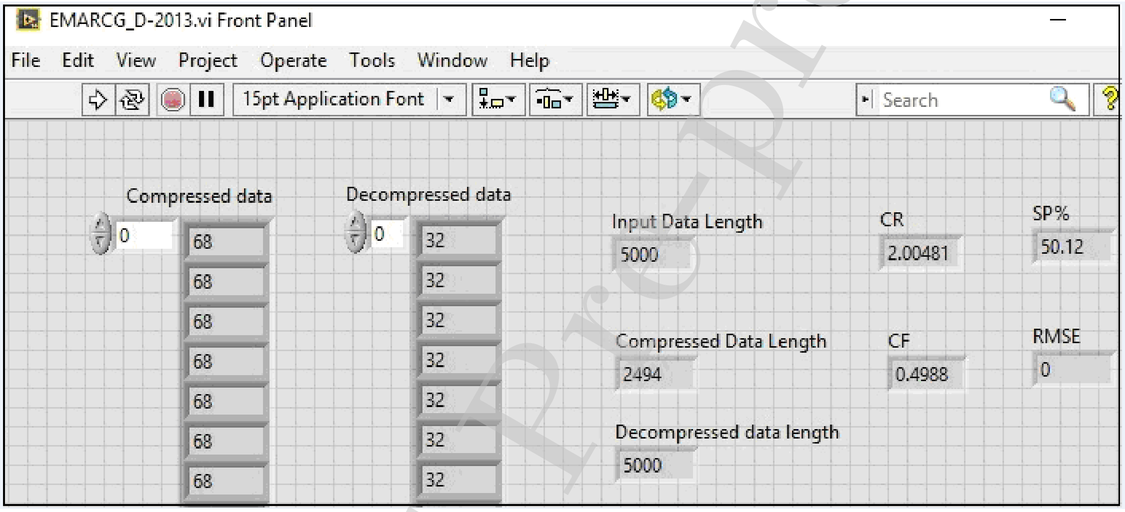

Figure 7. Experimental Set-up

The NI WSN set-up comes under the IEEE 802.15.4, Wireless Personal Area Network (WPAN) standard. NI WSN 9792 gateway node can connect to a maximum of 36 sensing nodes. It is used to connect the sensing nodes to the host computer. The gateway node is connected to the host computer via Ethernet cable. NI WSN 3202 programmable sensing node consists of 4 analog and 4 digital pins to which the sensors are connected. The sensing node is synchronized with the gateway node using the LabVIEW Measurement and Automation software package. The sensing node is programmed using LabVIEW software to acquire the sensed data, process it and transmit the data to the host computer via gateway node. The post processing of the received data is carried out in the host computer loaded with the LabVIEW. The LabVIEW software consists of two windows namely block diagram and front panel. Block diagram is used for programming and the front panel is used as an UI (User Interface) window. Wireshark [39] is network analyzer tool for which the source code is freely available. It is used for recording and analyzing the network traffic and packet details. It is installed in the PC to analyze the reduction in data traffic obtained due to the implementation of compression algorithm.

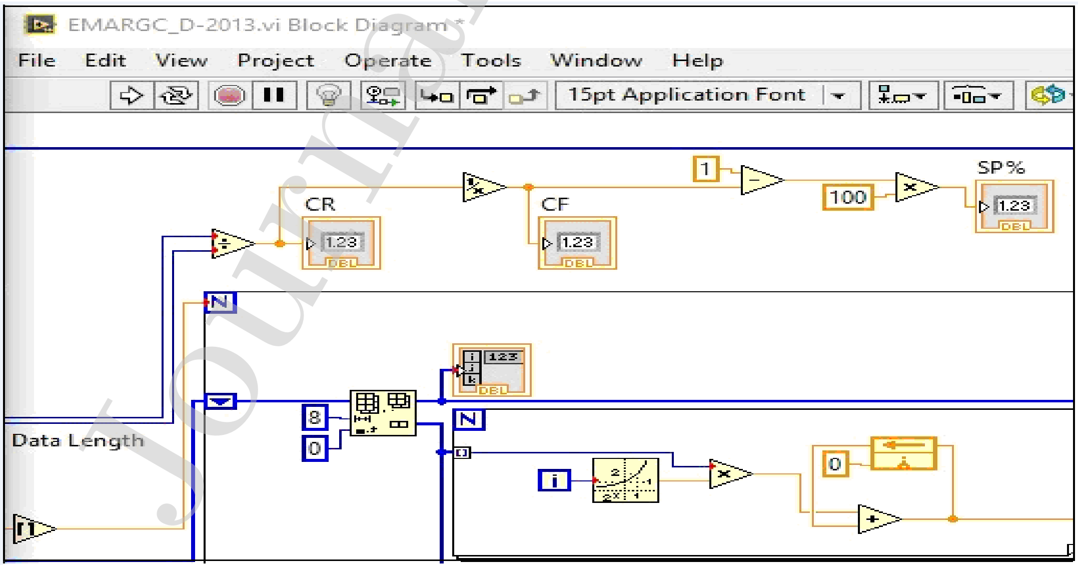
The temperature sensor LM35 is connected to the Analog input (AI0) of the sensor node NI 3202. The sensing node is powered by a set of 4 AA batteries and the gateway node is powered using its respective adapter. Both the nodes are connected wirelessly to the WSN platform by setting and verifying the necessary options in NI MAX software. The gateway node is provided with an IP address of 169.254.7.183 and the sensing node is automatically assigned with an IP address of 169.254.51.170.

The sample interval of the sensing node is set to 1 minute to acquire the temperature data. The sensing node is programmed by building and deploying the proposed compression algorithm in its memory using LabVIEW programming software. As the sample interval is set as 1 minute in the sensing node, to show variations in the acquired temperature readings a soldering rod powered randomly is placed near the temperature sensor and a dataset length of 5000 is collected. The data acquired by the sensing node is then compressed and transmitted wirelessly to the gateway node. The 5000 data length is thereby compressed to 2494 data length by the proposed algorithm loaded in the sensing node.

The gateway node is connected to a PC via the Ethernet cable and is programmed with the decompression algorithm using LabVIEW. As gateway node receives the compressed data from the sensing node it decompresses the data to retrieve the original dataset. The results of various performance evaluation parameters for the proposed compression algorithm obtained using LabVIEW programming is shown in the below Fig.8.




(a) Front Panel of the LabVIEW

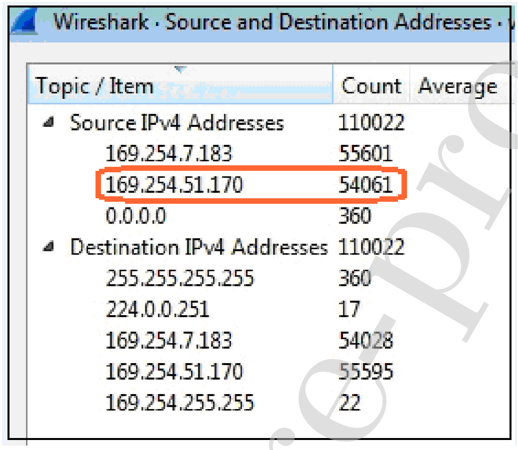



(b) Block diagram of the LabVIEW

Figure 8. Results of performance evaluation parameters in LabVIEW

From the above figure it is observed that by applying the proposed algorithm for the real time temperature data of length 5000, the compression ratio = 2.005, compression factor = 0.498, saving percentage = 50.12% and RMSE = 0 is obtained.

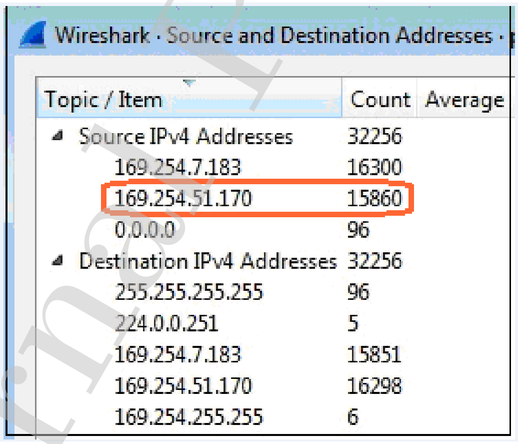
The analysis of packet count during the wireless transmission between the sensing node and the gateway node is extracted using the Wireshark network packet analyzer tool. The analysis is performed on the amount of packets transmitted before and after compression. The packet detail is obtained from the IPV4 statistics option in the tool as shown below. Fig. 9(a) represents the packet details communicated between the sensing node & gateway node before compression and Fig. 9(b) and represents the packet details communicated between the sensing node & gateway node after compression.



Wireshark · Source and Destination Addresses ·

| Topic / Item | Count | Average |
|---|---|---|
| Source IPv4 Addresses | 110022 | |
| 169.254.7.183 | 55601 | |
| 169.254.51.170 | 54061 | |
| 0.0.0.0 | 360 | |
| Destination IPv4 Addresses | 110022 | |
| 255.255.255.255 | 360 | |
| 224.0.0.251 | 17 | |
| 169.254.7.183 | 54028 | |
| 169.254.51.170 | 55595 | |
| 169.254.255.255 | 22 | |

Figure 9(a). Packet details communicated between the sensing node and gateway node before compression



Wireshark · Source and Destination Addresses ·

| Topic / Item | Count | Average |
|---|---|---|
| Source IPv4 Addresses | 32256 | |
| 169.254.7.183 | 16300 | |
| 169.254.51.170 | 15860 | |
| 0.0.0.0 | 96 | |
| Destination IPv4 Addresses | 32256 | |
| 255.255.255.255 | 96 | |
| 224.0.0.251 | 5 | |
| 169.254.7.183 | 15851 | |
| 169.254.51.170 | 16298 | |
| 169.254.255.255 | 6 | |

Figure 9(b). Packet details communicated between the sensing node and gateway node after compression

In both figures shown above the highlighted value indicates the transmitted packet count when the sensing node with an IPV4 address of 169.254.51.170 is acting as a source. It is observed that the packet count of about 54,061 packets is transmitted by the sensing node before applying compression whereas after compression the packet count transmitted by the sensing node has a good reduction of about 15,860 packets. This reduction in packet count that reduces the data traffic in turn will scale down the power required for the data transmission from the sensing node to the gateway node.

## IX. Energy Model

The key factor to be considered while transmitting the data from one node to the other in a WSN application is the energy consumed during transmission of data. The transmission energy is minimized by reducing the length of the data transmitted using compression algorithm. The energy consumed during transmission of n bytes per hop [40] is calculated by the following expression (5):

$$E_{radio}(n) = n\, I_{TX}\, V\, T_{TX} + n\, I_{RX}\, V\, T_{RX} \qquad (5)$$

The current drawn for sending and receiving the data ($I_{TX}$ & $I_{RX}$), the Voltage supply (V) and the operating time over 1 byte ($T_{TX}$ & $T_{RX}$) for IEEE 802.15.4 standard is given in Table II [40].

TABLE II. PARAMETERS OF THE ENERGY MODELS

| Parameter | Value |
|---|---|
| $I_{TX}$ | 17.4 mA |
| $I_{RX}$ | 19.7 mA |
| $T_{TX}$ | $3.2\times10^{-5}$s |
| $T_{RX}$ | $3.2\times10^{-5}$s |
| V | 3.3 V |

The transceiver of NI nodes operates in 2.4GHz band and is compatible with IEEE 802.15.4 radio standard for communication. Thereby the above table is used to evaluate the energy consumption during transmission of data from the sensing node to gateway node. In the proposed work the energy consumed on transmitting the uncompressed data from the sensing node to the gateway node is calculated by substituting the above parameter values in equation (5) as Euncomp = 5000 ($I_{TX}$ V $T_{TX}$) + 5000($I_{RX}$ V $T_{RX}$) = 19.6 mJ. Similarly the energy consumed on transmitting the compressed data from the sensing node to the gateway node is calculated as Ecomp = 2494 ($I_{TX}$ V $T_{TX}$) + 2494 ($I_{RX}$ V $T_{RX}$) = 9.77 mJ. Therefore energy saving of about 9.83 mJ is obtained during data transmission using the proposed lossless compression algorithm.

## X. Conclusion

One of the hard drawbacks of WSN is energy consumption during data communication. In this work an algorithm based on a lossless compression is proposed and its performance is compared with the existing adaptive Huffman coding algorithm with respect to different parameters using MATLAB simulation. Simulation result shows that the proposed lossless compression algorithm performs better than the existing adaptive Huffman coding algorithm with respect to compression ratio, compression factor, saving percentage and computation time. The proposed algorithm is also implemented in real time using NI WSN hardware and LabVIEW software. The result of various compression performance analysis parameters on real time implementation is displayed in the LabVIEW front panel window. Packet sniffing is performed to analyze the data traffic before & after compression using Wireshark network analyzer tool. The packet trace obtained from the tool results with a good reduction in number of packets transmitted from the source node to destination node after compression. The results obtained conclude that the proposed lossless compression algorithm is simple with less computation time, has a better compression ratio & saving percentage than the existing algorithm, reduces the amount of packets transmitted from the source node and minimize the energy consumed during communication. Thus the proposed algorithm will help to increase the lifetime of the network by saving the energy of each node for computation and communication. As a future work the proposed algorithm has to be tested for various real time applications of WSN and internet of things (IoT).

## REFERENCES


[1] Poonam Prasad, "Recent trend in wireless sensor network and its applications: a survey", Sensor Review, Vol.35 Issue:2, pp.229-236, 2015, https://doi.org/10.1108/SR-08-2014-683.

[2] Fangxin Chen, Lejiang Guo, and Chang Chen, "A survey on energy management in wireless sensor networks," Sciverse Science Direct, Elsevier, IERI Procedia 3, Pages: 60-66, 2012, https://doi.org/10.1016/j.ieri.2012.09.011.

[3] Tossaporn Srisooksai, Kamol Keamarungsi, Poonlap Lamsrichan and Kiyomichi Araki, "Practical data compression in wireless sensor network: A survey," Journal of Networks and computer applications, Elsevier, vol. 35, Issue 1, pp. 37-59, 2012, https://doi.org/10.1016/j.jnca.2011.03.001.

[4] J.Uthayakumar, T.Vengattaraman and P. Dhavachelvan, "A Survey on data compression techniques: From the perspective of data quality, coding schemes, data types and applications", Journal of King Saud University – Computer and Information Sciences, 17 May 2018.

[5] M.A. Razzaque, Chris Bleakley and Simon Dobson, "Compression in wireless sensor network: A survey and comparative evaluation," in Journal, ACM Transactions on Sensor Networks, Volume 10, Issue 1, Article No.5 November 2013, https://doi.org/10.1145/2528948.

[6] You-Chiun Wang "Data Compression Techniques in Wireless Sensor Networks," Pervasive Computing, Nova Science Publishers, 2012, https://people.cs.nctu.edu.tw/~wangyc/publications/books/b004-nova11-compression.pdf.

[7] Jonathan Gana Kolo, S.Anandan Shanmugam, David Wee Gin Lim, Li-Minn Ang and Kah Poohi Seng, " An Adaptive Lossless Data Compression Scheme for Wireless Sensor Networks", Journal of Sensors, Hindawi Publishing, Volume 2012, Article ID 539638, 2012, https://www.hindawi.com/journals/js/2012/539638/.

[8] Ikjune Yoon, Hyeok Kim and Dong Kun Noh, "Adaptive Data Aggregation and Compression to Improve Energy Utilization in Solar-Powered Wireless Sensor Networks", Sensors 2017, doi: 10.3390/s17061226.

[9] Cheng, S. Tzong, et al, "Hierarchical Distributed Source Coding Scheme and Optimal Transmission Scheduling for Wireless Sensor Networks", Wireless Personal Communications, Springer, pp. 1-22, 2013, https://link.springer.com/article/10.1007/s11277-012-0725-0.

[10] Metin Aktas, Murat Kuscu, Ergin Dinc and Ozgur B. Akan, "D-DSC: Decoding Delay-based Distributed Source Coding for Internet of Sensing Things", PLOS ONE, https://doi.org/10.1371/journal.pone.0193154 March 14, 2018.

[11] Francesco Marcellon and Massimo Vecchio, "Enabling energy-efficient and lossy-aware data compression in wireless sensor networks by multi-objective evolutionary optimization", Information Science, Elsevier, 2010, doi:10.1016/j.ins.2010.01.027.

[12] Caione. C, Brunelli. D, and Benini. L, "Distributed compressive sampling for lifetime optimization in dense wireless sensor networks", IEEE Transaction on Industrial Informatics, Volume 8, Issue 1, pages: 30 –40, Feb 2012, 10.1109/TII.2011.2173500.

[13] CuicuiLv, QiangWang, WenjieYan and JiaLi, " Compressive Sensing – based sequential data gathering in WSN's", Computer Networks, Science Direct, Volume 154, 8 May 2019, Pages 47 – 59, https://doi.org/10.1016/j.comnet.2019.03.004

[14] Thanh Dang, Nirupama Bulusu and Wu-chi Feng, "Robust Data Compression for Irregular Wireless Sensor Networks Using Logical Mapping", ISRN Sensor Networks, Hindawi, Volume 2013, http://dx.doi.org/10.1155/2013/253257 .

[15] Zordan, D., Martinez B., Vilajosana, I. and Rossi M., "On the Performance of Lossy Compression Schemes for Energy Constrained Sensor Networking" ACM Trans. Sensor Network, Article A, 34 pages, January 2012, doi:10.1145/2629660 .

[16] C.M. Sadler, and M. Martonosi, "Data Compression Algorithms for Energy-Constrained Devices in Delay Tolerant Networks," Proc. 4th ACM Int. Conf. Embedded Networked Sensor Systems, Boulder, Colorado, USA, October 31–November 3, pp. 265–278. ACM, New York, USA, 2006, doi: 10.1145/1182807.1182834.

[17] Y. Liang, Y. LiAn, "Efficient and robust data compression algorithm in wireless sensor networks", IEEE Communication Letter, Volume 18, Issue 3, pp. 439-442, March 2014, doi: 10.1109/L.COMM.2014.011214.132319.

[18] C. Tharini and P. Vanaja Ranjan, "Design of modified adaptive Huffman data compression algorithm for wireless sensor network," Journal of Computer Science, vol. 5, no. 6, pp. 466– 470, 2009, doi: 10.3844/jcssp.2009.466.470.
[19] A. K. Maurya, D. Singh, and A. K. Sarje, "Median predictor based data compression algorithm for Wireless Sensor Network," International Journal of Smart Sensors and Ad Hoc Networks, volume -1, Issue-1, pages: 62–65, 2011, http://interscience.in/IJSSAN_Vol1Iss1/paper14.pdf.
[20] S. Chang, et al., "Energy and storage reduction in data intensive wireless sensor network applications", TR-15-07, Harvard University, 2007, http://ftp.deas.harvard.edu/techreports/tr-15-07.pdf.
[21] Shaik. Mahaboob Basha and B. C. Jinaga, "A Novel Optimized Golomb-Rice Technique for the Reconstruction in Lossless Compression of Digital Images", ISRN Signal Processing, Volume 2013 , Article ID 539759, 5 pages, http:// dx.doi.org/10.1155/2013/539759 .
[22] Mo Yuanbin, et al., "A Data Compression Algorithm Based on Adaptive Huffman Code for Wireless Sensor Networks", IEEE Explore, Intelligent Computation Technology and Automation (ICICTA), 2011, Volume 1, doi: 10.1109/ICICTA.2011.8.
[23] P. Rachelin Sujae and S. Selvaraju, "Power Efficient Adaptive Compression Technique for Wireless Sensor Networks," Middle-East Journal of Scientific Research 20 (10): 1286-1291, 2014, doi: 10.5829/idosi.mejsr.2014.20.10.114158.
[24] Khalid Sayood, "Introduction to Data Compression" (Fourth Edition), Elsevier, Science Direct, pages 43 -89, 2012, https://doi.org/10.1016/C2010-0-69630-1.
[25] Dragos Ioan Sacaleanu et al., "An adaptive Huffman algorithm for data compression in Wireless Sensor Networks", IEEE Explore, International Symposium on Signals, Circuits and Systems (ISSCS), Pages 1 -4, 2011, doi: 10.1109/ISSCS.2011.5978764.
[26] S. Golomb, "Run-Length Encodings," IEEE Transactions on Information Theory, vol. IT-12, no. 3, pp. 399–401, July 1966, doi: 10.1109/TIT.1966.1053907.
[27] Henrique S. Malvar, "Adaptive Run-Length / Golomb-Rice encoding of quantized generalized Gaussian sources with unknown statistics" Data Compression Conference, In-proceedings, IEEE, 2006, 10.1109/DCC.2006.5.
[28] A. Kiely, "Selecting the Golomb Parameter in Rice Coding", IPN Progress Report 42-159, November, 2004, https://ipnpr.jpl.nasa.gov/progress_report/42-159/159E.pdf.
[29] S.Kalaivani, C.Tharini, "Efficient Data Compression Technique Using Modified Adaptive Rice Golomb Coding For Wireless Sensor Network", ARPN Journal of Engineering and Applied Sciences, Vol. 10, No. 12, July 2015, http://www.arpnjournals.com/jeas/research_papers/rp_2015/jeas_0715_2258.pdf.
[30] Nishad P M, 2014, "A novel approach to reduce computational complexity of Multiple Dictionary Lempel Ziv Welch (MDLZW) using Indexed K-Nearest Twin Neighbor (IKNTN) clustering and binary insertion sort algorithm", Ph.D Thesis, Bharathiyar university, Tamil Nadu, India, 2015, pp. 22-32.
[31] Shuxia Wang, "Multimedia data compression storage of sensor network based on improved Huffman coding algorithm in cloud", Multimedia Tools and Applications, Springer, pp 1–14, 27 May 2019, doi: 10.1007/s11042-019-07765-0.
[32] Jianlin Liu, Fenxiong Chen, Jun Yan and Dianhong Wang, "CBN-VAE: A Data Compression Model with Efficient Convolutional Structure for Wireless Sensor Networks", Sensors 2019, 19(16), 3445; https://doi.org/10.3390/s19163445.
[33] Liang Sun, Lei Wang, Jian Fang, Jialin Liu and Can Ma, "MIHBS: A Mobile Interface of High Bandwidth for Wireless Sensor Networks", Special Section on Collaboration for Internet of Things, IEEE Access, Volume 7, 2019.
[34] www.ni.com/pdf/manuals/372775e.pdf
[35] www.ni.com/en-in/shop/labview.html
[36] S.Kalaivani and C.Tharini, "Analysis and Modification of Rice Golomb Coding Lossless Compression Algorithm for Wireless Sensor Networks", Journal of Theoretical and Applied Information Technology, June 2018, Vol. 96 No.12.
[37] http://www.ni.com/white-paper/8710/en/

[38] www.ni.com/pdf/manuals/372998b.pdf
[39] https://www.wireshark.org
[40] Y. Liang and W. Peng, "Minimizing energy consumptions in Wireless Sensor Networks via two-modal transmission", Computer Communication Review, vol. 40, no. 1, pp. 13–18, 2010, doi: 10.1145/1672308.1672311.

BIBILIOGRAPHY

Ms. S. Kalaivani received her M.Tech., degree from Anna University, Chennai, India. Currently she is pursuing Ph.D. at BSA Crescent Institute of Science & Technology, Chennai. Her interest lies in the area of wireless communications. At present she works as Assistant Professor (Senior Grade) in B.S.A Crescent Institute of Science & Technology, Chennai. She has a total teaching experience of 12 years. She has published papers in International conferences and Journals. She is a member of ISTE and IE.

Dr. C. Tharini received her Ph.D. in Information and Communication from Anna University, Chennai. At present she is the Head of the Department of Electronics and Communication Engineering in B.S.A Crescent Institute of Science & Technology, Chennai. She has a total teaching experience of 23 years. She is a life member of Computer Society of India. She has published papers in International conferences, Journals and few of her papers are cited in many peer reviewed journals. Her research interests include application of signal processing algorithms for energy efficient wireless sensor network design and Wireless Communication. She is a member of IE.

**Conflict of Interest**

This paper has not communicated anywhere till this moment, now only it is communicated to your esteemed journal for the publication with the knowledge of all co-authors.

**Ethical approval**

This article does not contain any studies with human participants or animals performed by any of the authors.