Comparison and Implementation of Compression Algorithms in WSNs

B. Ananda Krishna
Professor
Department of ECE
Kallam Haranadhareddy
Institute of Technology
Guntur, AP

N. Madhuri
Assistant Professor
Department of ECE
Kallam Haranadhareddy
Institute of Technology
Guntur, AP

M. Malleswari
M.Tech.ECE (CESP)
I - II Semester
Acharya Nagarjuna University
Guntur, AP

Abstract:- **Wireless Sensor Networks consists of many autonomous devices using sensors that can process information and communicate wirelessly with their neighboring sensor nodes. There are many limitations in design of WSNs, the power utilization is the main limitation. In some cases, it is very difficult to replace the batteries of sensor nodes. Though the network is more efficient, because of the limitation of power, the life time of the network will reduce. To overcome this power limitation and to make the network more power efficient, different power saving/reduction algorithms are proposed by different authors. The power consumption can be reduced by modifying like encryption & decryption algorithms, Routing algorithms, Energy Efficient Algorithms, Compression and decompression algorithms, minimizing control packets, and many other different power reduction algorithms. Among all the algorithms, we have selected data compression techniques to reduce power utilization, memory and bandwidth reduction. To achieve our target, we have worked on different algorithms like Huffman Coding-compression algorithm, Modified Huffman Coding-compression algorithm, Run Length coding-compression algorithm and Lempel-Ziv-Welch coding-compression algorithms. We have compared all of the above algorithms, and concluded that Lempel-Ziv-Welch algorithm is the best one among all of the above algorithms, In this algorithm there is a compression ratio above 82%. These algorithms are examined and implemented in Glomosim and viewed that the information is maximum compressed which reduces the power consumption, thereby increasing battery life.**

Keywords– WSNs, Power reduction, Huffman Compression, Decompression, Modified Huffman, LZW Compression, Decompression, RLE.

1. INTRODUCTION

A Wireless Sensor Network can be defined as a network of devices that can communicate the information gathered from a monitored field through wireless links. The data is forwarded through the multiple nodes, and with a gateway, the data is connected to other networks like wireless Ethernet. A sensor is a device that responds and detects some type of input from both the physical or environmental conditions, such as pressure, heat, light, etc.

The output of the sensor is generally an electrical signal that is transmitted to a controller for further processing.WSN is a wireless network that consists of base stations and number of nodes (wireless sensors). These are used to monitor physical or environmental conditions like sound, pressure, temperature and co-operatively pass data through the network to main location. WSNs have distinct characteristics such as, power constraints, limited battery life, redundant data acquisition, low duty cycle, security and many-to-one flows. The power issue in the WSNs is one of the biggest challenges, because the sensor has a limited source of power which is also hard to replace or recharge. To improve the efficiency of WSNs and to make them more power efficient, it is necessary to implement the power reduction algorithms. To make WSNs more power efficient, different solutions have been proposed by different authors and some of them are listed below.

1. Data compression & decompression algorithms.

2. Minimization of control packets by reducing link failures, results low power consumption.

3. Specially designed power control techniques

4. By modifying encryption & decryption algorithms

Among many techniques/proposals, the authors Ruxanayasmin, et al, recommended Lempel-Ziv-Welch (LZW) compression technique to reduce power utilization [1]. The algorithm is speedy, easy to implement and efforts best for the information with lots of recurring data. The algorithm is well-organised as the output is similar to numerical data and also it doesn't need the string table for the decompression. The result of decompression decreases the number of bits to utmost extent so that the requirement of memory and bandwidth reduces. Another advantage is that the compressed text may appear like a random/noise message and an assailant in middle can't able to recognize. Therefore, the data compression decreases the size of the original text and also provides data security.

Section-II describes the related work on power minimization techniques recommended by various authors

IJERTV8IS070057

www.ijert.org
(This work is licensed under a Creative Commons Attribution 4.0 International License.)

1039

and the explanation of Huffman Coding and other algorithms is present in Section-III. Section-IV illustrates the implementation and execution. Conclusion with future work ares shown in Section-V.

2. RELATED WORK

The authors Ruxanayasmin, et al, recommended Lempel-Ziv-Welch (LZW) compression technique to reduce power utilization. The result of decompression decreases the number of bits to utmost extent so that the requirement of memory and bandwidth reduces[1].

The authors Shahin Sheihk, Ms.Hemlata Dakhore proposed Data Compression Techniques for WSNs. They proposed modified Huffman Coding for Wireless Sensor Networks and concluded that the compression ratio is better for the data with high correlation and moderate correlation [2].

The authors Swati C.Pakhale, et al, proposed Data Compression Techniques Using Huffman Code for Wireless Sensor Network with an objective of reducing the bit rate that implies less power. The coding is done by taking the difference between present and previous sample of the sensed data which reduces the bit rate [3].

The authors Peng Jiang, et al, suggested an Optimal Estimation Model and Distributed Coding of data compression for WSNs. In their proposed model, the sink correlation parameters are obtained based on optimum estimation in which the time and space redundancy is included in data is acquired by sensors. As a result, the reduction in redundancy, the average energy cost per node and the life of the WSNs is increased [4].

The authors M.Malleswari et al proposed Implementation of Modified Huffman Coding in Wireless Sensor Networks. In this; they secured the algorithm by implementing ones complement and circular shifting operations and concluded that any text can be compressed more than 50% of its original size [5].

3. DESCRIPTION OF ALGORITHMS

3.1 Description of Huffman Coding Algorithm

Huffman coding is a source coding technique used for lossless data compression. The term refers to the use of a variable-length code table for encoding of text files, where the variable-length code table has derived in a particular way based on the estimated probability of occurrence for each possible value of the source symbol. Huffman coding uses a specific method for choosing the representation for each symbol, resulting in a prefix code sometimes called "prefix-free codes", that is, the bit string representing some particular symbol is never a prefix of the bit string representing any other symbol that expresses the most common characters using shorter strings of bits than are used for less common source symbols. The concept of Huffman Coding is obtained from binary trees which are used for data compression

The algorithm is explained with an example.

- Assume five different symbols with relative frequencies are:
  - K: 30
 - L: 20
  - M: 40
  - N: 5
  - O: 5
-

The Huffman tree is constructed by considering the symbols with less frequency. Here those symbols N and O have frequency 5 and 5, so combine those two frequencies. NN and O have already been used, and the new node is placed above them (named as N+O) with the sum 10.
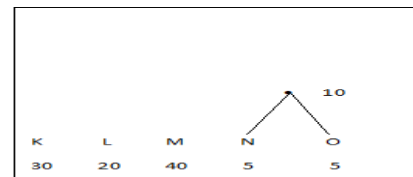

- Next, L, N+O are low with value 20 and 10 respectively.
- Combine L and N+O which yields a new node of value 30.
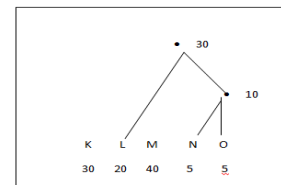
- Now the symbols K and L+N+O has less probability, so combine these two frequencies and form a new node called as K+L+N+O.
- Now combine the node K+L+N+O and M and form a root node .
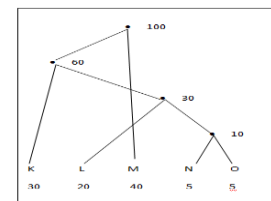
- Allot binary 0 to left branch and binary 1 to right branch and repeat the same for all branches as shown below.
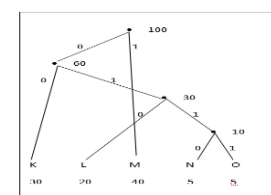
- The bits assigned for the given symbol are

K=00
L=010
M=1
N=0110
O=0111

- Each path terminates at a leaf.

3.2 Description of Modified Huffman Coding

The original Huffman coding is modified to obtain an algorithm which is more useful by means of security. The existing Huffman coding is slightly enhanced to get the new algorithm by introducing one's complement and shifting operations. By means of the one's complement and shifting operations the algorithm is more powerful and secured. The description of the modified algorithm is given below.

At the Transmitter:-

At the Transmitter, the binary group of data is divided into 128 groups of bits each; the one's complement is applied on the 128bits group of data, followed by shifting operations. Finally the Huffman coding is applied for the circular shifted data.

At the Receiver:-

At the Receiver, Huffman decoding, circular shifting and one's complement are implemented.

Advantages of Modified Huffman coding

1. The information can be secured from the attackers.
2. Transmission bandwidth and memory requirement can be minimized.
3. Less power is required to transmit less number of bits

3.3 Description of Run length Coding

Run length coding is a simple form of lossless data compression technique, in which the binary sequence of data is represented in the form of successive runs and counts. It is the most useful technique for text compression, when the data has more redundancy. The Run length coding is explained by a simple example as follows.

Example 1 - Let the redundant text be "nnnnggkkkknnn". When the Run Length coding is applied for the above text , the Run Length is as follows,

n	n	n	n	g	g	k	k	k	k	n	n	n

n	4	g	2	k	4	n	3

Example 2- Let the binary data is " 1110011110000", the Run Length code is given by the following table.

1	1	1	0	0	1	1	1	1	0	0	0	0

1	3	0	2	1	4	0	4

Advantages of RLE are, if the data has high redundancy, then the compression ratio will be high, which leads reduction of memory, bandwidth, power. If the data has low redundancy, then the above said case may be reversed. Therefore, the Run length coding is more useful for the data with high redundancy. The most important characteristic of the RLE is that any text can be compressed without any loss of data.

3.4 Description of LZW Algorithm

LZW compression works by reading a sequence of symbols, grouping the symbols into strings, and converting the strings into codes. Because the codes take up less space than the strings they replace, we get compression. Characteristic features of LZW includes,

- LZW compression uses a code table, with 4096 as a common choice for the number of table entries. Codes 0-255 in the code table are always assigned to represent single bytes from the input file.
- When encoding begins the code table contains only the first 256 entries, with the remainder of the table being blanks. Compression is achieved by using codes 256 through 4095 to represent sequences of bytes.
- As the encoding continues, LZW identifies repeated sequences in the data, and adds them to the code table.
- Decoding is achieved by taking each code from the compressed file and translating it through the code table to find what character or characters it represents.

4. IMPLEMENTATION AND SIMULATION RESULTS

The proposed work is implemented in Glomosim (Global Mobile Information System Simulation) and the performance is evaluated. It is the simulation tool for large wireless networks. It provides C-based Parallel Simulation Environment for complex System (PARSEC). The proposed algorithms are implemented in Glomosim and it's performance is evaluated and given in below table.

S.No	Original data (bits)	Huffman coding			Modified Huffman coding			Run length coding			LZW algorithm		
		CD (bits)	E(%)	T(S)	CD (bits)	E(%)	T(S)	CD (bits)	E(%)	T(S)	CD (bits)	E(%)	T(S)
1	951	264	73.8	0.025	264	73.80	0.032	482	49.35	0.05	171	82.01	107.52
2	2870	929	68.17	0.056	929	68.17	0.059	1448	49.54	0.07	312	89.12	104.24
3	4525	1446	68.12	0.076	1446	68.12	0.093	2237	50.56	0.11	399	91.18	97.33
4	9159	2914	68.02	0.164	2914	68.02	0.193	4450	51.41	0.21	571	93.76	81.06

Figure 1: Comparison of compression algorithms

Figure 1 shows the comparison between Huffman coding, Modified Huffman Coding, Run Length Coding and LZW algorithm. The experiment has done on different data bit sizes and calculated the efficiency and time required for compression.

5. CONCLUSION AND FUTURE WORK

In this work, the Huffman Coding Technique, Modified Huffman Coding Technique, Run length Coding Technique, Lempel-Ziv-Welch techniques are implemented and compared with each other in terms of efficiency and time. By using any of the compression techniques, any text can be compressed more than 50percent of its original size with no loss of data. The above all the algorithms are more efficient for large input text. Among all of them LZW algorithm has better efficiency and has high security. Reduction of number of bits requires low power, bandwidth and memory that fulfil our objectives. At outlook, other features to improve the battery life can address, so that the quality of service in WSNs can be improved. In the future, the other compression techniques are needed to implement and compare their performance.

6. REFERENCES

[1] B.Ruxanayasmin, "Implementation of Data Compression Techniques in Mobile Ad hoc Networks", International journal of Computer Applications, Volume 80, No. 08, Oct. 2013.

[2] Shahina Sheik, Ms. Hemlata Dakhore, "Data Compression Techniques for Wireless Sensor Network", International journal of Computer Science and Information Technologies,Vol.6 (1),2015,818-821

[3] Swati C.Pakhale, et al, "Data Compression Technique Using Huffman Code for Wireless Sensor Network", Internationa journal Engineering Sciences & Research Technology,4(3): March,2015.

[4] Peng jiang and Sheng-Qiang Li," A Data Compression Algorithm for Wireless Sensor networks Based on an Optimal Estimation Mdel and Distributed Coding", Sensors 201,10,9065-9083.

[5] Malleswari, "Implementation of Modified Huffman Coding in Wireless Sensor Networks", IJCA, 177(1):14-17, November 2017