



Data Compression Algorithm for Wireless Sensor Network

Reshma B. Bhosale¹, Rupali R. Jagtap²

^{1,2}Department of Electronics & Telecommunication Engineering, Annasaheb Dange College of Engineering & Technology, Ashta, Sangli (M.S), India

ABSTRACT- In use of wireless sensor network technology for environmental monitoring, the two main fundamental activities of wireless sensor network is data acquisition and data transmission. However, transmitting or receiving data are power consuming task. Wireless sensor network consists of a set of sensor nodes. The processing capacity varies with each node. Sensors in it have storage capacity which is limited. The main challenge today in this field is to improvise the power and energy management of sensor network. Different techniques have been introduced by various researchers. In order to reduce power consumption during transmission, we introduce data compression by processing information locally.

This paper presents the simulation results for RLE method for data compression. The MATLAB GUI model is employed to simulate the RLE method. The simulation result shows that the RLE method can effectively compress the data with minimum power consumption.

Key Words: Wireless Sensor Network, Data Compression, RLE

I. INTRODUCTION

Wireless Sensor Network (WSN) is being increased deployed for enabling continuous monitoring and sensing of physical variables of our world. Recent technological breakthrough in low power processing units and communication devices have enabled the development of distributed autonomous nodes able to sense environmental data, compute and transmit it using wireless.

Communication to a base station known as Sink for future analysis; thus, forming a Wireless Sensor Network [1]. In the wireless sensor technology, the two main activities of wireless sensor network are data acquisition and transmission. However, transmitting/Receiving data are power consuming task. We explore data compression by processing information locally. The most common technique to save energy is the use of sleep mode where significant parts of sensor's receiver is switched off. In most cases, the radio receiver on board sensor nodes is the main

cause of energy consumption. Hence it is important to keep the receiver in switched off mode most of the time to save energy. Nevertheless, using the sleep mode reduces data transmission/reception rate and thus consumption in the network [2] [3].

The question is how to keep the same data rate sent to the base station by reducing the number of transmission. In network producing techniques, reduces the amount of data to be transmitted. Data compression is the process that reduces the amount of data in order to reduce data transmitted, because the size of the data is reduces. There is many benefits to data compression. The main advantage of it, however, is to reduce storage requirements. Also, for data communications, the transfer of compressed data over a medium result in an increase in the rate of information transfer. Note that data compression can be implemented on existing hardware by software or through the use of special hardware devices that incorporate compression techniques. Below figure

shows a basic data compression block diagram.

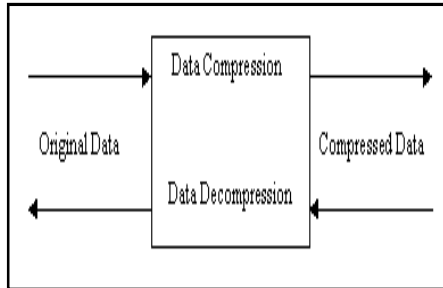


Fig.1 Data Compression Block Diagram
Modelling and Coding:

The reconstruction requirement may force the decision whether to apply loss or lossless compression on a given data. However, regardless of the compression type, all algorithms share a similar process when compressing information, namely: modelling and coding. The difference between modelling and coding has often been misunderstood. In the first stage (modelling), the algorithm tries to model the redundancy within the data. Generally, this can be achieved by collecting information regarding the frequencies of the repetitive patterns. In the second stage (coding), a description of the used model is encoded. Typically, this description is encoded using binary streams, which represent the compressed version of the information. Hence, the decision of how to encode a particular symbol (or set of symbols) is based on the model. Below figure illustrates the two stages of the compression process.

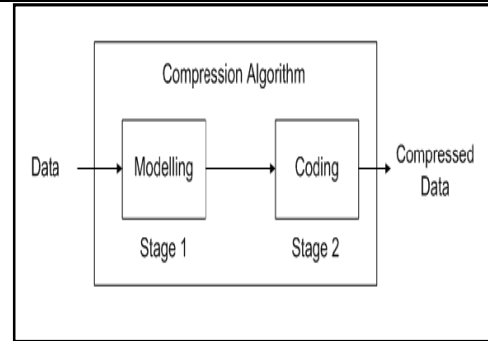


Fig.2 Stages of Compression Process

Lossless compression techniques, as their name implies, involve no loss of information. If data have been losslessly compressed, the original data can be recovered exactly from the compressed data. Lossless compression is generally used for applications that cannot tolerate any difference between the original and reconstructed data. Lossless compression is possible because most real-world data has statistical redundancy. For example, an image may have areas of color that do not change over several pixels; instead of coding "red pixel, red pixel," the data may be encoded as "279 red pixels". This is a basic example of run-length encoding; there are many schemes to reduce file size by eliminating redundancy [3].

Lossy compression techniques involve some loss of information, and data that have been compressed using lossy techniques generally cannot be recovered or reconstructed exactly. In return for accepting this distortion in the reconstruction, we can generally obtain much higher compression ratios than is possible with lossless compression. Lossy image compression can be used in digital cameras, to increase storage capacities with minimal degradation of picture quality. Similarly, DVDs use the

lossy MPEG-2 videocoding format for video compression.

II. The Compression Algorithms

Compression techniques are predominately used to increase the energy efficiency and the life time of sensors. It also helps to cut communication cost and computation cost. A few of the compression techniques for wireless sensor networks are

Run length Encoding

The Idea behind this algorithm is, if a data item *d* occurs *n* consecutive times in the input data we replace the *n* occurrences with the single pair *nd*. Run-Length Encoding (RLE) is a basic compression algorithm. It is very useful in case of repetitive and slowly varying data items. This is most useful basic compression algorithm on data that contains many such runs: for example, relatively simple graphic images such as icons, line drawings, and grayscale images. Which is a lossless data compression algorithm used for slowly varying sensor and image data. It is not useful with files that don't have many runs as it could double the file size. This is most useful basic compression algorithm on data that contains many such runs: for example, relatively simple graphic images such as icons, line drawings, and grayscale images. Which is a lossless data compression algorithm used for slowly varying sensor and image data. It is not useful with files that don't have many runs as it could double the file size.

For example, consider a screen containing plain black text on a solid white background. There will be many long runs of white pixels in the blank space, and many short runs of black pixels within the text. Let us take a hypothetical single scan line, with

B representing a black pixel and W representing white:

```

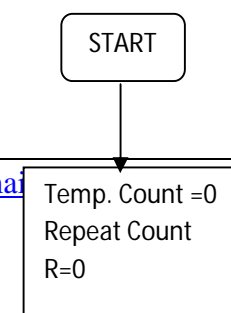
WWWWWWWWWWWWBWWWWWW
WWWWWWBBBWWWWWWWWWWW
WWWWWWWWWWWWWWWWBWWWW
WWWWWWWWWWWW
  
```

Apply the run-length encoding (RLE) data compression algorithm to the above hypothetical scan line, the encoded text is as follows:

12W1B12W3B24W1B14W Interpret this as twelve W's, one B, twelve W's, three B's, etc.

The run-length code represents the original 67 characters in only 18. Of course, the actual format used for the storage of images is generally binary rather than ASCII characters like this, but the principle remains the same. Run-Length Encoding (RLE) is a basic compression algorithm. The graphical representation of the RLE algorithm applied on temperature readings.

2.2 Flowchart for RLE:

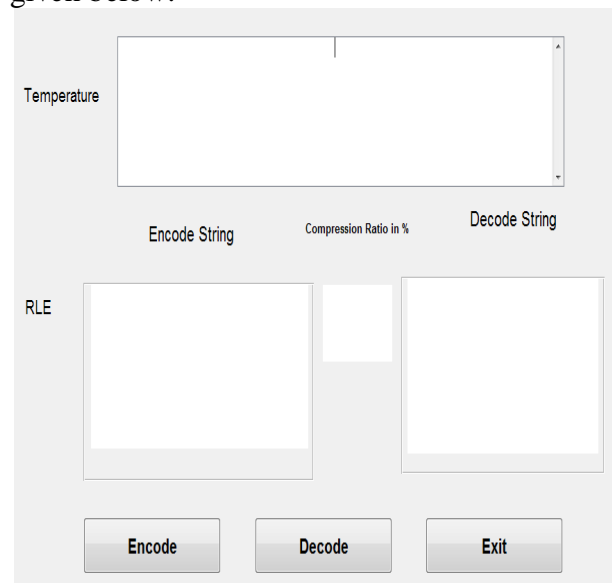


2. It is particularly well suited to palette-based bitmapped images such as computer icons, and was a popular image compression method on early online services such as CompuServe before the advent of more sophisticated formats such as GIF.
3. Run length encoding is especially valuable in the compression of images and changing images. It lies behind many of the methods being used to compress video information.

SAMPLE MATLAB RESULTS

Above flowchart are the graphical representations of the RLE algorithm applied on temperature readings. However, because RLE is based on the same consecutive input string.

Its results depend on the data source. Here the GUI model for the RLE Algorithm is given below:



Above figure shows the graphical representation of the RLE algorithm applied on temperature readings. However, because RLE is based on the same consecutive input stream, its results depend on the data source. In this way, in order to perform RLE results with different data sources statistics.

Application of RLE:

1. Run-length encoding schemes were employed in the transmission of television signals.

In above GUI model the temperature readings are to be inserted and then we have to run the model. It will execute the RLE algorithm and we will get encoded string

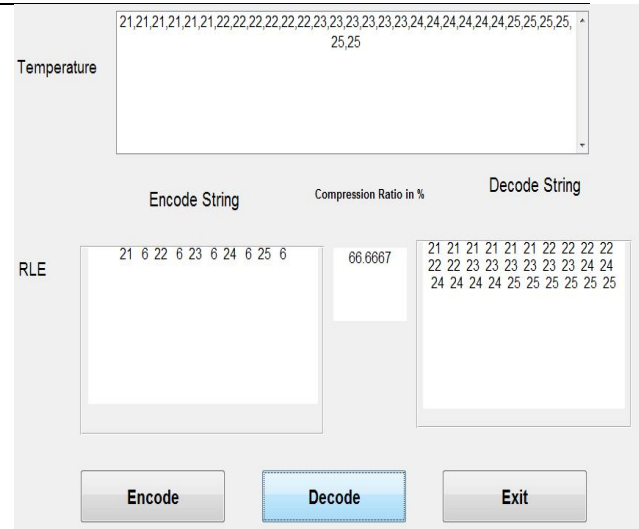
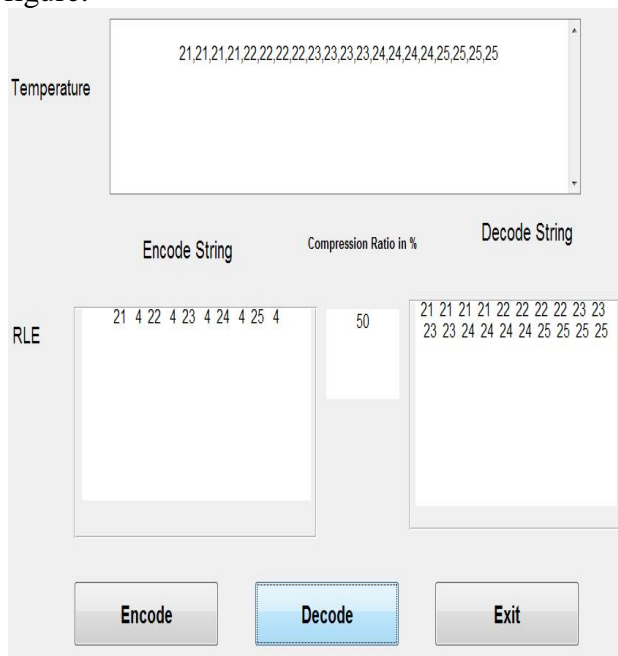
along with the compression ratio & decoded string.

Let us take a hypothetical example of Temperature readings:

21,21,21,21,22,22,22,22,23,23,23,23,24,24,24,24,25,25,25,25

Apply the run-length encoding (RLE) data compression algorithm to the above hypothetical example; the encoded text is as follows:

21 4 22 4 23 4 24 4 25 4 Interpret this as four 21, four 22, four 23, four 24, four 25etc. The run-length code represents the original 20 characters in only 10 and hence compression ratio is 50% as shown in below figure.



Above figure shows that RLE algorithm is generally used for the data compression. We can continue to increase the RLE's ratio by increasing the number of bits in the string. Indeed, the feature of lossy compression is that compressing data and then decompressing it retrieves data that may well be different from the original, but is close enough to be useful that is why the precision is chosen by the user according to the application.

CONCLUSION

This paper discusses the RLE data compression algorithm. Using this algorithm a better compression ratio has been achieved when compared to the previously proposed algorithm. We have taken a RLE data compression algorithm for WSN using real temperature datasets. There is a major limiting constraint in RLE, for RLE to achieve good compression ratio, the input data must contain long sequences of repeated characters, and this rarely occurs in

the data generated from sensors. The energy consumption using RLE offers a better compression ratio. It consumes less energy as compression ratio is increased. In this paper, we have shown the trade-off between energy consumption and compression efficiency. Energy consumption is less for transmission & it require less RAM for data storage.

REFERENCES

- [1] Eugene Pamba Capo-Chichi, Hervé Guyennet, Jean-Michel Friedt, “K-RLE: A new Data Compression Algorithm for Wireless Sensor Network” 2009 IEEE
- [2] Shengchun Long; Pengyuan Xiang, "Lossless Data Compression for Wireless Sensor Networks Based on Modified Bit-Level RLE," Wireless Communications, Networking and Mobile Computing (WiCOM), 2012 8th International Conference on, vol., no., pp.1, 4, 21-23 sept.2012,
- [3] Fransisco Marcelloni, Massimo Vecchio, Member, IEEE, “A simple Algorithm for Data compression in wireless sensor network” 2008 IEEE.
- [4] V.KRISHNAN1, MR. R.TRINADH, “A low power new data compression for wire/ wireless sensors network using K-RLE” International Journal of Electronics Signals and Systems (IJESS), ISSN: 2231-5969, Vol-3, Iss-2, 2013.
- [5] P. Suneel Kumar, Patibandla Swapna, “Performance evaluation of K-RLE compression algorithm Technique for Text data” International Journal of Advanced Research in Computer and Communication Engineering, Vol. 4, Issue 6, June 2015.