IJCSMS International Journal of Computer Science & Management Studies, Special Issue of Vol. 12, June 2012
ISSN (Online): 2231 –5268
www.ijcsms.com

Temporal Data Compression in Wireless Sensor Network using LZW

Anubha Barak[1], Preeti Gulia[2]

[1]M.tech Student, Department of Computer Science and Applications,
M. D. University, Rohtak-124001, Haryana, India
Anubhabarak59@gmail.com

[2]Assistant Professor, Department of Computer Science and Applications,
M. D. University, Rohtak-124001, Haryana, India

Abstract

Wireless networks find maximum usage in present scenario due to less overhead of establishment. But these networks have constrained environment. Wireless networks have limited processing capability, memory space and battery power. These networks also have routing overhead involved with normal activities. The power required for transferring data is much more than processing the data. Therefore, data compression techniques can be used to save the power. Also it is good in reducing load on intermediately nodes. In this paper we have proposed a LZW data compression algorithm.

Keywords: *Wireless sensor networks, data compression.*

1. Introduction

Wireless sensor networks consist of sensor nodes which move in a defined environment to sense any abnormal activity. These nodes regularly send some sensor information to each other and also report a base station about it. So, these nodes need to communicate in short time. These nodes cover a wide geographical area and coordinate with each other to provide high quality information about its environment.

Each sensor node bases its decisions on its mission, the information it currently has its knowledge of its computing, communication and energy resources. Each of these scattered sensor nodes has the capability to collected compressed and route data either to other sensors or back to an external base stations.

A base-station which is also known as a sink may be a fixed node or a mobile node capable of connecting the sensor network to an existing communications infrastructure or to the Internet where a user can have access to the reported data, applying data compression in each node before transmitting data for reducing total power consumption by a sensor node. The sensor node consists of sensing unit, processing unit, transceiver and power unit as shown in figure 1.

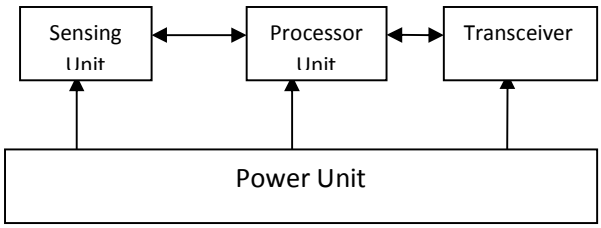


Figure 1 Components of Wireless Sensor Network

Advances in sensor and communication technology have focused interest on using wireless sensor networks, which are formed by a set of small sensor devices that are deployed in an ad hoc fashion to cooperate on sensing a physical phenomenon, making the inferences, and transmitting the data. Typically, each individual sensor can sense in multiple modalities but has limited communication and computation capabilities.

Wireless sensor networks hold the promise of revolutionizing sensing in a wide range of application domains because of their reliability, accuracy, flexibility, cost-effectiveness, and ease of deployment. Wireless sensor networks share many of the challenges of traditional wireless networks, including limited energy available to each node and bandwidth-limited, error-prone channels. Among these challenges, energy is typically more limited in wireless sensor networks than in other wireless networks because of the nature of the sensing devices and the difficulties in recharging their batteries. Usually, the following three metrics are used to evaluate the design of any wireless sensor networks.

IJCSMS International Journal of Computer Science & Management Studies, Special Issue of Vol. 12, June 2012
ISSN (Online): 2231 –5268
www.ijcsms.com

1. Energy efficiency/system lifetime: As sensor nodes are battery-operated, the design of the wireless sensor network must be energy-efficient to maximize system lifetime.
2. Accuracy: Obtaining accurate information is the primary objective.

1.1 Data Compression

In the field of computer science and information theory, data compression or source coding is the process of encoding information using fewer bits or other information-bearing units than an un encoded representation would use through use of specific encoding schemes.

Lossless compression technique, as the 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 algorithms usually exploit statistical redundancy in such a way as to represent the sender's data more concisely with fewer errors. Lossless compression is possible because most real-world data has statistical redundancy.

Lossy compression technique involves some loss of information and data that have been compressed using lossy technique generally cannot be recovered or reconstructed exactly. In return of accepting this distortion in the reconstruction we can generally obtain much higher compression ratio than lossless compression.

Data compression methods are commonly developed either under a classical rate distortion viewpoint or an operational rate-distortion viewpoint. The classical viewpoint strives to develop methods that are optimal on average, over an ensemble of realizations of a random process model; this necessarily demands a random model for the signal and knowledge of a probability structure.

The operational viewpoint specifies a compression framework whose design is often based on insights from the classical viewpoint and then optimizes the operating point of that framework for the particular signal at hand; this has the advantage of relaxing the assumptions made on the signal but has the disadvantage that side information describing the operating point must be included as overhead in the compressed bit stream, because a sensor network would likely be required to operate in an abundance of differing signal environments, in this dissertation, we focus on the operational viewpoint to remove the necessity of assuming statistical models for the signal.

1.1.1 Benefits of Data Compression for Energy efficiency in Wireless Sensor Network

Compression is useful because it helps reduce the consumption of expensive resources, such as hard disk space or transmission bandwidth. On the downside, compressed data must be decompressed to be used, and this extra processing may be detrimental to some applications.

The design of data compression schemes therefore involves trade-offs among various factors, including the degree of compression, the amount of distortion introduced, and the computational resources required to compress and uncompress the data.

Energy efficiency in wireless sensor networks has principally been addressed through routing protocols, sleeping strategies, low-power architectures, and energy-efficient modulation schemes. Accuracy is generally controlled through optimal processing strategies as well as using accurate sensors deployed in optimal ways. Latency and channel capacity issues in sensor networks can be addressed through routing strategies and data compression. It is very important to understand the interplay between the compression method and routing. The data compression can bring more energy efficiency to a network than does recently proposed combinations of routing and data aggregation.

The purpose of coding is to exploit statistical redundancy among the quantization indices. The quantization and transform elements are designed in such a way as to ensure that the redundancy is localized. Ideally, the underlying random variables are all statistically independent. In that case, the indices may be coded independently and the only form of statistical redundancy which need be considered is that associated with any non-uniformity in their probability distribution.

The transform is responsible for mapping the original samples into a form which enables comparatively simple quantization and coding operations. On the one hand, the transform should capture the essence of statistical dependencies among the original samples so that the group of adjacent transform samples and the quantization indices possess common characteristics and exhibit at most only very local dependencies, ideally, independent; On the other hand, the transform should separate irrelevant information from relevant information according to

IJCSMS International Journal of Computer Science & Management Studies, Special Issue of Vol. 12, June 2012
ISSN (Online): 2231 –5268
www.ijcsms.com

certain criteria so that the irrelevant samples can be identified and quantized more heavily or even discarded whereas relevant samples are quantized lightly. Quantization is the element of lossy compression systems responsible for reducing the precision of data (reduce the word length of samples) in order to make them more compressible. In most lossy compression systems, quantization is the sole source of distortion. Next, the most widely used quantization (scalar quantization) will be reviewed. Scalar quantization is the simplest of all lossy compression schemes. It can be described as a function that maps each element in a subset of the real line to a single value in that subset. Consider partitioning the real line into M disjoint intervals. The compression gain is achieved by limiting the number of bits assigned to the quantizers of transform coefficients. In many cases, since different parts of transform coefficients have different statistics and importance for the tasks at hand, each quantizer has to be optimized for its own transform coefficients and the quantizers are not identical. For example, it is well known that the statistics of high-frequency components of audio are significantly different from those of the lower frequency components, therefore a subband coder assigned different quantizer to different spectral bands of audio. The optimal bit allocation or operational rate distortion optimization for a particular signal at hand instead of classical operational rate distortion optimization for an ensemble of realizations of a random process models can be formulated.

2. Related work

Naoto Kimura [2] discusses that the wireless sensor network are resource constrain off limited power supply, low bandwidth, processing speed. **Sebastian** [3] discusses that the low power sensor nodes are integral parts of large scale wireless sensor networks which find extensive application in domain such as military surveillance and environment monitoring. **Tan minsheng** [4] proposes a distributed wavelet- based algorithm which can transform irregularly sample data using haar wavelet-based compression. **Raymond** [5] proposes that the Distributed wavelet processing insensor network reduces communication energy and wireless bandwidth usage at sensor nodes. **Yang** [6] proposes the constructing a data gathering tree over a wireless sensor network in order to minimize the total energy for compression and transporting information from a set of source node to the sink. **Sayood** [7] discusses the compression technique, mathematical preliminaries, information and coding, Huffman coding, Dictionary coding, quantization, differential encoding, mathematical preliminaries for transform, sub bands, and wavelet. **Salomon** [8] discusses the basic compression technique, statistical method, dictionary method, image compression, video compression and wavelet. In this we study different data compression technique. The data compression ratio is low. So to improve the compression ratio we propose lzw data compression technique.

3. Proposed Work

To design a system using Data Compression techniques as a tool for accomplishing the optimal trade-off between rate, energy, and accuracy in a wireless sensor network, whose objective is to maximize the operation lifetime of the sensor networks. Data compression shrinks raw data to smaller volumes, which is desirable for data communication, since less data requires less time and less energy for transmission and reception. Wireless sensor networks are resource constraint. They have limited power supply, less bandwidth for communication, poor processing speed and memory space.

To achieve maximum utilization of those resources is applying data compression on sensor data because processing data consumes much less power than transmitting data in wireless medium. So it is effective to apply data compression before transmitting data for reducing total power consumption by a sensor node.

3.1 System Model

IJCSMS International Journal of Computer Science & Management Studies, Special Issue of Vol. 12, June 2012
ISSN (Online): 2231 –5268
www.ijcsms.com

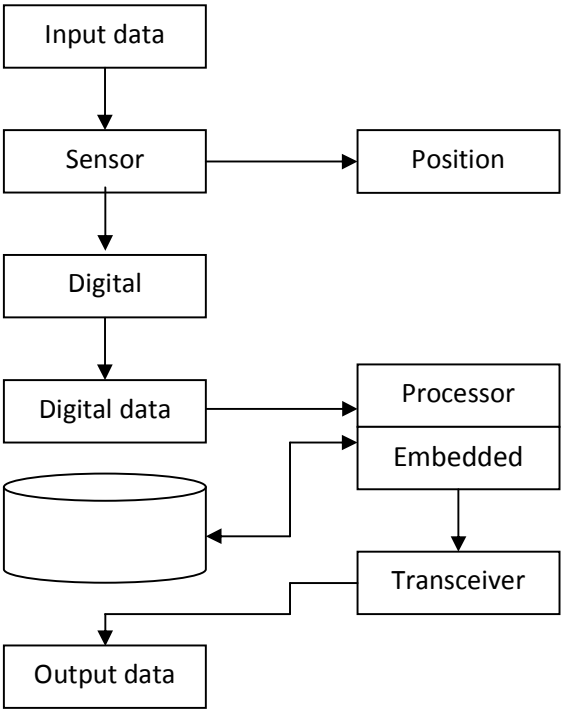


Figure 2 System Model

The above Figure 2 Illustrate the system model of the work that is carried out, the sensor grabs the data from the source which is towards the positioning finding system, the converter converts to the digital format where the processor is responsible for sending the data to the destination. The compression code LZW, Coding by ordering, Pipeline in network and Distributed Compression is embedded in the processor, so that the data are send to the destination in the compressed format.

4. Algorithm and Implementation

4.1 Encoding algorithm

The compressor algorithm builds a string translation table from the text being compressed. The string translation table maps fixed-length codes to strings. The string table is initialized with all single-character strings .As the compressor character-serially examines the text, it stores every unique two-character string into the table as a code/character concatenation, with the code mapping to the corresponding first character. As each two-character string is stored, the first character is output. Whenever a previously-encountered string is read from the input, the longest such previously-encountered string is determined, and then the code for this string concatenated with the extension character the next character in the input is stored in the table. The code for this longest previously-encountered string is outputted and the extension character is used as the beginning of the next string.

4.1.1 Algorithm

Step 1: At the start, the dictionary contains all possible roots, and P is empty

Step 2: C: = next character in the charstream

Step 3: Is the string P+C present in the dictionary

a if it is, P: = P+C (extend P with C)

b if not

i. output the code word which denotes P to the codestream;

ii. add the string P+C to the dictionary;

iii. P: = C (P now contains only the character C)

c. are there more characters in the charstream

i. if yes, go back to step 2;

ii. if not:

i. output the code word which denotes P to the codestream;

ii. END.

4.2 Decoding algorithm

The decompressor algorithm only requires the compressed text as an input, since it can build an identical string table from the compressed text as it is recreating the original text. However, an abnormal case shows up whenever the sequence character or string with the same character for each character and string for each string is encountered in the input and character or string is already stored in the string table.

When the decompressor reads the code for character or string in the input, it cannot resolve it because it has not yet stored this code in its table. This special case can be dealt with because the decompressor knows that the extension character is the previously-encountered character.

Step 1: At the start the dictionary contains all possible roots

IJCSMS International Journal of Computer Science & Management Studies, Special Issue of Vol. 12, June 2012
ISSN (Online): 2231 –5268
www.ijcsms.com

Step 2: cW: = the first code word in the codestream (it denotes a root)

Step 3: output the string.cW to the charstream

Step 4: pW: = cW;

Step 5: cW: = next code word in the codestream

Step 6: Is the string.cW present in the dictionary

i. if it is,

1. Output the string.cW to the charstream

2. P: = string.pW

3. C: = the first character of the string.cW

4. Add the string P+C to the dictionary

ii. if not,

1. P: = string.pW

2. C: = the first character of the string.pW

3. Output the string P+C to the charstream and add it to the dictionary (now it corresponds to the cW)

Step 7: Are there more code words in the codestream

iii. If yes, go back to step 4

if not, END.

5. Experimental Results

5.1 SENSE Simulator

SENSE is designed to be an efficient and powerful sensor network simulator that is also easy of use and also it has been identify the three most critical factors as:

Extensibility: The enabling force behind the fully extensibility network simulation architecture is based on component-based simulation and also introduced a component-port model that frees simulation models from interdependency usually found in an object-oriented architecture, and then proposed a simulation component classification that naturally solves the problem of handling simulated time.

The component-port model makes simulation models extensible: a new component can replace an old one if they have compatible interfaces, and inheritance is not required. The simulation component classification makes simulation engines extensible: advanced users have the freedom to develop new simulation engines that meet their needs.

Reusability: The removal of interdependency between models also promotes reusability. A component developed for one simulation can be used in another if it satisfies the latter's requirements on the interface and semantics. There is another level of reusability made possible by the extensive use of C++ template: a component is usually declared as a template class so that it can handle different type of data.

Scalability: Unlike many parallel network simulators, especially SSFNet and Glomosim, parallelization is provided as an option to the users of SENSE. It reflects the belief that completely automated parallelization of sequential discrete event models. Therefore, parallelizable models require more effort than sequential models. In SENSE, a parallel simulation engine can only execute components of compatible components. If a user is content with the default sequential simulation engine, then every component in the model repository can be reused.

The performance of the compression is evaluated in the compression ratio. The Table 1 represents the nine samples taken from range of 1 to 6 KB and then compressed using the proposed algorithm. It has found that the size of the compressed data is minimized up to 56% when compared to the original data.

Original data(KB)	Compressed data using LZW(KB)
5.5	3.08
5	2.8
4.6	2.576
4.5	2.52
4	2.24
3.5	1.96
3	1.68
2	1.12
1	0.56

IJCSMS International Journal of Computer Science & Management Studies, Special Issue of Vol. 12, June 2012
ISSN (Online): 2231 –5268
www.ijcsms.com

5.2 Screens

The below Figure 3 represents the screen shot of the dictionary based LZW algorithm using VC++. The compressor algorithm builds a string translation table from the text being compressed. The string translation table maps fixed-length codes to strings.

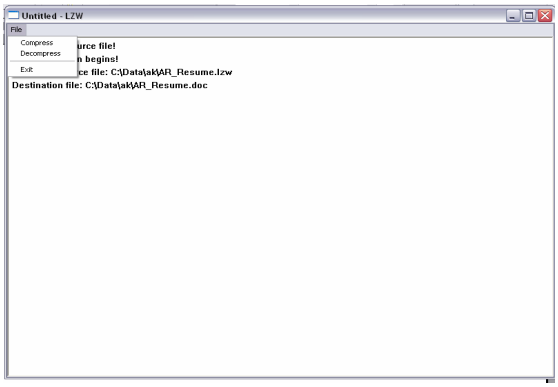


Figure 3 Output screen for compress and decompress

The compressor character-serially examines the text, it stores every unique two-character string into the table as a code/character concatenation, with the code mapping to the corresponding first character. As each two-character string is stored, the first character is output.

The below Figure 4 represents the screen shot of the dictionary based LZW algorithm using VC++ for compression. The compressor algorithm builds a string translation table from the text being compressed. The string translation table maps fixed-length codes to strings. The string table is initialized with all single-character strings. As the compressor character-serially examines the text, it stores every unique two-character string into the table as a code/character concatenation, with the code mapping to the corresponding first character.

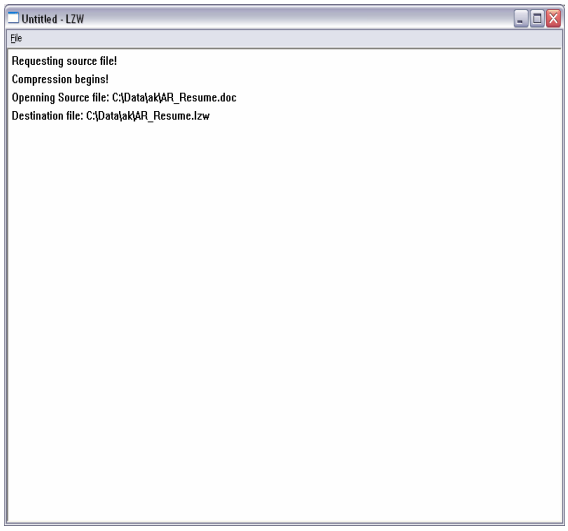
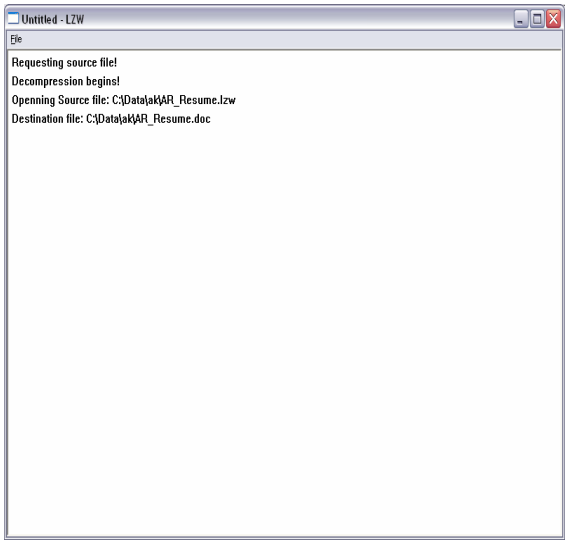


Figure 4 Output screen for compression

The below Figure 5 represents the screen shot of the dictionary based LZW algorithm using VC++ for decompression

The decompression algorithm requires the compressed text as an input, since it can build an identical string table from the compressed text as it is recreating the original text. When the decompressor code reads the code for character or string in the input, it cannot resolve it because it has not yet stored this code in its table. This special case can be dealt with because the decompressor knows that the extension character is the previously-encountered character.



IJCSMS International Journal of Computer Science & Management Studies, Special Issue of Vol. 12, June 2012
ISSN (Online): 2231 –5268
www.ijcsms.com

Figure 5 Output screen for Decompression

The performance of the compression is evaluated using the simulator SENSE. The below Figure 6 represents the output of the SENSE simulator for node 1 to node 4 which is using the LZW algorithm. Node 1 sends 10 messages, Node 2 sends 15 messages and Node 3 sends 25 messages. The total power consumption is calculated in mA. Node 1 consumes 110 mA, Node 2 consumes 117 mA and Node 3 consumes 169 mA

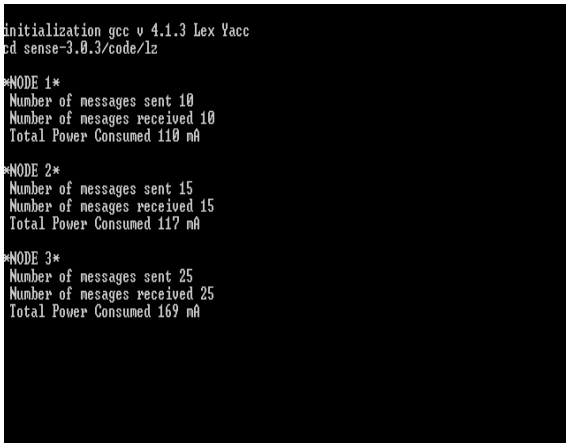


Figure 6 Simulation result for energy-consumption using LZW compression

6. Conclusion

It has been proved that people are discussing wide range of application areas for wireless sensor network. In this project data compression scheme LZW was presented.
The experimental results indicates that their compression ratio. They are one possible method to diminish resource constrain of wireless sensor node. The proposed work has been concluded with the light weight algorithms. In future we can use the same algorithms with merging with effective routing protocols so that it can achieve more energy efficiency.

7. References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks” IEEE Communications Magazine, Volume: 40 Issue: 8, pp.102-114, August 2002.

[2]Naoto Kimura and Shahram Latifi “A Survey on Data Compression in Wireless Sensor Networks” Proceedings of the International Conference on Information Technology: Coding and Computation, Berkeley, March 2005.

[3] Sebastian Puthenpurayil, Ruirui Gu and Shura S Bhattacharyya “Compression Techniques for Minimum energy Consumption” In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, Volume 2, pp. 45-48, Honolulu, Hawaii, April 2007.

[4] Tan Minsheng, XIE Zhijun and Wang Lei “Voronoi Tesselation based Haar Wavelet Data Compression for Sensor Network”, Journal of Software, 2005 Vol.17, No. 4, April 2006, pp. 860-867.

[5] Raymond S. Wagner, Shu Du and Albert Cohen “An Architecture for Distributed Wavelet Analysis and Processing in Sensor Network”, ACM Portal, April 2006, pp. 19-21.

[6] Yang Yu, Bhaskar Krishnamachari and Viktor K. Prasanna “Data Gathering with Tunable Compression in Sensor Network” IEEE Transactions on Parallel and Distributed Systems, April 2007, pp. 345-349.

[7] Sayood Khalid, “Introduction to Data Compression,” 2nd Edition, Morgan Kaufmann Publishers Inc, 1996.

[8] Salomon David, “Data Compression: The Complete Reference,” 3rd Edition, Springer 2004.

[9] C. S. Raghavendra, Krishna M. Sivalingam and Taieb Znati, “Wireless Sensor Networks”, Springer US, 2004.

[10] Holger Karl, Andreas Willig, “Protocols and Architectures for Wireless Sensor Networks”, John Wiley & Sons, 2005.

[11] Sense simulator tutorial - http://www.ita.cs.rpi.edu/sense