International Journal of Advancements in communication Technologies – IJACT
Volume 1 : Issue 2 *[ISSN 2374 – 1511]*
Publication Date : 30 September,2014

Text Compression Algorithms with Clusters for Wireless Sensor Networks

AshutoshTripathi,NarendraYadav,Reena Dadhich

Abstract—**Energy consumption not only depends on sensing the data but also on processing the sensed data and transmitting or receiving them to or from its neighbor nodes. So if it is possible to control number of transmission and receipt of messages, a significant amount of energy can be saved. Many studies have turned attention on reducing the amount of energy in wireless sensor network by controlling bits in most frequent communication and modify the way of communication between the hop to hop. All the hop follows a architecture and energy is depends on the that architecture and communication between Hop. This paper modifies the architecture of SPIN protocol and implements the cluster and clusterhead over the SPIN and proposed best way of communication between Hop to Hop using new compression algorithms over M- SPIN in Wireless Sensor Networks**.

***Keywords*—Wireless sensor Network; SPIN; Huffman Encoding;**

I. Introduction

The life time wireless sensor networks (WSN) is depends upon the energy and there is the limited energy supply in wireless sensor network. In every communication among the node energy is required and also in every communication, need of guaranteed data delivery to designated sink including energy efficient. If the energy is less then guaranteed data delivery to designated sink can not guaranteed and probability to loss of data.

Ashutosh Tripathi
Dept of ECE,Amity University Rajasthan, Jaipur
India

Narendra yadav
Dept of CSE,Sri BalaJi Technical Campus Jaipur
lndia

Reena Dhadich
Dept of CS/IT, University of Kota, Kota
lndia

M- SPIN protocol that is able to conserves more energy than the SPIN protocol [1-3], this protocol transmits information only to sink node instead of transmitting throughout the network. Total number of packet transmissions is less. Further if this protocol uses clustering and data compressing during the communication then significant amount of energy can be saved. The use of a hierarchical (cluster-based) architecture provides benefits over the flat architecture. A cluster-based structure facilitates the spatial reuse of resources to increase the system throughput. In routing, because the set of clusterheads and clustergateways normally forms a virtual backbone for inter-cluster routing, and thus the generation and spreading of routing information is restricted among the set of clusterheads and clustergateways. Next, a cluster structure makes a network smaller and more stable. On the other hand,if Huffman's idea is to replace fixed-length codes (such as ASCII) by variable-length codes, assigning shorter code-words to the more frequently occurring symbols and thus decreasing the overall length of the data and compression of more frequent data can be help full in energy saving of wireless sensor network. that compression techniques lead to life time of wireless sensor network. In this paper proposing the Architecture of SPIN with Cluster based for avoiding the problem of implosion and traffic of flooding and new algorithms that consist compression ratio is better will help in improving the life time of wireless sensor network

The remaining part of this paper is organized as follows: In Section II, consists major related work that carried out in concern of compression technique. In Section III, we presented the working of new algorithms and implementations. The paper concludes with a discussion in section IV.

II. Related Work

Authorof [8] is prposed RLE,RLE is one of the best encoding scheme with strings of repeated symbols (run). The idea leads to encode repeated symbols as a pair, the length of string and the symbol. For example, 'abbaaaaabaabbbaa' of length 16 bytes (characters) is represented as 7 integers plus 7 characters, which can be easily encoded on 14 bytes (as for eg. '1a2b5a1b2a3b2a'). The biggest problem with RLE is that in worst case the size of output data can be two times more than the size of input data. To eliminate this problem, each pair (the

lengths and strings separately) can be encoded with an algorithm like Huffman Coding,BPC and amount of Compression achieved for Shannon-Fano algorithm is presented in Table-1. The compression ratio for Shannon-Fano algorithm is in the range of 0.60 to 0.82 and the average BPC is 5.50. Compression ratio for Huffman Coding Algorithm falls in the range of 0.57 to 0.81. The compression algorithm is better compared to Shannon – Fano algorithm and the average BPC is 5.27.

Statistical Comparative results of all the compression techniques are provided. For example the BPC(bits per character) for book2 is 4.83 for Huffman coding, 8.16 for RLE & 5.08 for Shannon-Fano Algorithms.

In this paper [9], Author praposed experimental result that provides an evident that Compression Ratio is greater in Huffman Algorithm than LZ77. For example Huff3 uses only 2969 memory bits while LZ77 uses 8192 memory bits during FPGA Resource utilization. Therefore, Huffman Algorithm is better than LZ family for standalone hardware implementation of Encoder-Decoder.

Author [10], Author praposed about the Design and provides a very high Compression ratio with high efficiency using Bitwise OR Algorithm but it is very costly and complex when Hardware implementation of Lossless Data Compression Design is combined with X-Ray Spectrometer.

Therefore, in our proposed work a system is implemented using source encoder and decoder as similar to simple communication system.

In [11] ON-Chip parallel architecture hardware is devised to classify text, based on Arithmetic Coding Data Compression providing very high speed up factor. Due to high speed up factor this hardware performs 26 times faster than software based implementation. But it provides relatively less compression ratio of approximately 60% as compared to high Compression ratio of Huffman Algorithm providing a compression ratio of 80%. But the concept of Pattern classifier is not used in the proposed model as we are focusing on Lossless text transmission which mainly focuses on Compression rate without any loss.

In [12] The Compression rate of LZ77 is improved by 40% and area is reduced by 30% by using Systolic-Array Approach. If the same approach is used with the Huffman Algorithm then compression rate can be increased by 70-80%. Thus real-time speed and area efficient FPGA implementation can be achieved. End-Tagged Densed Coding Algorithm which is simpler for word based compression but less efficient than Adaptive Huffman Algorithm. It represents that compression rate is lesser for smaller frequency i.e. (repetition of similar data) and compression rate worsen with smaller text collection. It uses the concept that first bit in every byte is reserved for the flag, so a data of 4-bits is converted into a data of 8-bits.Example:

If probability of an Alphabet is 1/16 then
a) Plain Huffman code = 0001
b) Tagged Huffman Code = 00 00 00 01

The Designing of Look-Up table for frequency calculation is also complex for end-tagged scheme as compared to Huffman.So Huffman Algorithm is preferred.

Author [14], presents an improvement over Static Huffman as adaptive Huffman Algorithm which is suited for changing statistics of highly correlated data. The number of bits transmitted after compression is smaller in Adaptive Huffman and it is found to be better than Static. The Compression ratio with Adaptive is 55% while 45% with Static Huffman Algorithm. Fig.(3) in this paper represents the comparison . So, as an advanced technique Adaptive Huffman Algorithm is used.

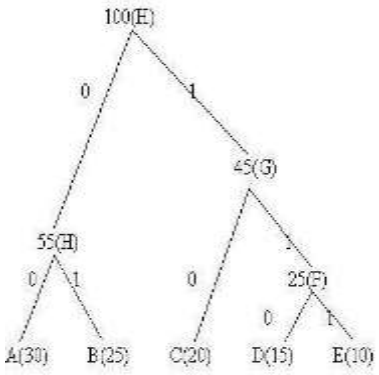

Figure: 1 Huffman Algorithms

III. Working of Adaptive Huffman Algorithm

Suppose the occurrence of letters in a text is given in percentage as:
A = 30%,B = 25%,C = 20%,D = 15%,E = 10%
The tree and frequency allocation will be in following way:
The Encoded Bits for different alphabets are:
A(00), B(01), C(10), D(110), E(111)

It depicts that the compression rate is lesser for lesser frequency. So this scheme is more beneficial when the repetition of any symbol is greater. Now let us take an example and calculate the compression rate for every technique. Suppose our Text is MISSISSIPPI RIVER-Original Size.Total no. of Characters=17 and No. of Bits=17*8=136.Adaptive Huffman-Encoded Bits for Different Characters are:I(00),S(01),P(100), R(101),M(1100),V(1101),E(1110),–(1111)Total Bits = 26, Compression Rate = (136-26)*100/136 = 80.88%

B. Implementation of Proposed Algorithm:

B1.Data Compression:

Algorithms: Compression Process

- Select the mode for compression
- Enter the string (0-N-1)
- Calculate the frequency of each character after grouping completed
- Start sorting and assigning the binary code
- Displays the encoded message after code allotted

In above algorithms, describes the latest trend in Data Compression i.e. Huffman Algorithm for variable length mainly divided into two large blocks such as Encoder and Decoder. The encoder compresses the data and transmits it fixed, the decoder has to find out the length of each code while processing.

Step 1: Ask the user to select any of two modes. Mode-1 for Compression of data and transmission, Mode-2 for decompression of received data. In this case user selected mode-1 i.e. compression mode.

Step 2: Enter the string or characters (total 52 characters of ASCII system is taken into account in the program). The entered array of characters is stored into a variable named 'INPUT'.

Step 3: The entered string is stored in an integer type variable 'MESSAGESYMBOLCOUNT' which consists of registers of 8-bits long and have a total of variable length greater than the characters length i.e. output limit is greater than 'INLIMIT'

Step 4: Check that whether every entered character is fetching exactly 8-binary bits (because we are considering only ASCII format). If any variations then display the error otherwise move to the next level.

Step 5: After verifying that every character is 8-bits long, it counts the frequency or occurrence of each data and lists encoder-decoder. Using the decoder we can obtain the original data from the compressed data. The System is and decoder receives the data and decompresses the encoded symbol. Since the Huffman code length is not the different text according to their frequency of occurrence. It prepares a table based on the occurrence (probability) of each text data in decreasing order. For this purpose variable 'UNIQUESYMBOL' is used in the Verilog program written.

Step 6: Load the total count in integer variable 'SYMBOLSCOUNT'. Apply the Temp input counting and after getting any different character the 'SYMBOLSCOUNT' is incremented using Loop.

Step 7: Sorting of character frequency is done using comparison process using loop.

Step 8: After sorting of data assign different-different binary codes based on Static/Dynamic/Adaptive/Hybrid Huffman Algorithm process.

Step 9: Display the encoded binary codes of every character using $display encoded message.

Step 10: Stop the encoding process if every character is encoded.

B2.Decoding Process

Algorithms: De-compression Process

- Select the mode for compression
- Enter the encoded message
- Fetch the binary code after possible decoding process
- Allocate the binary code for different character
- Displays the message after complete encoding process

Step 1: The input is taken bit by bit in a shift register.

Step 2: A new bit gets shifted into the shift register for every clock pulse.

Step 3: The output of the shift register is given to a comparator whose other input is from the frequency table.

Step 4: The comparator compares the output from the shift register with the frequency table.

Step 5: If a match is found the output is stored in an output register and the contents of the input shift register is cleared.

Step 6: If no match is found the next bit is shifted and a comparison is made again with the frequency table.

Step 7: The same process is done till a match is found.

Step 8: A counter is used to keep track of the number of the bits being shifted into the shift register.

Step 9: The above process is done till all bits are decoded.

Step 10: If any encoded scheme doesn't match with our decoding process then a message is displayed to use another methods.

V. Result

The above process has implemented using ModelSim Software with Verilog HDL. The Verilog code is then simulated and synthesized. The Xilinx design Software is used to place, route the gate components according to the design specifications. The Software generates a bit which is then burned into a FPGA Spartan kit. The hardware is tested with a given set of bit stream.

VI. Conclusion

New Suggested Adaptive Huffman Algorithm is best amongst other text Compression and Decompression Techniques. It will play a role in energy conservation of wireless sensor network. The code will be implemented using Verilog programming and Simulated over Xilinx Spartan-3e tool.

REFERENCES

[1] W. B. Heinzelman, A.P. Chandrakasan and H.Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Micrsensor Networks," IEEE Transaction on Wireless Communication, Vol. 1, No. 4, pp. 660-670, 2002.

[2] Zeenaat Rehna and Krishnu Kumar, "Spin Implemetation in tinny OS Environmentbusing nesC," Proc. 1st IEEE Conf. Computing Communication and Networking Technologies, , June 2010.

[3] Zeenaat Rehna and Surbani Roy, "A Modified Spin for Wireless Sensor Network," IEEE Computing Communication and Networking Technologies, , June 2011.

[4] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," IEEE Trans. Info. Theory, vol- IT 46.2,pp. 388-404, Mar. 2000.

[5] X. Y. Hong, K. X. Xu and M. Gerla, "Scalable Routing Protocols for Mobile Ad Hoc Networks," IEEE Network, pp. 11-21, July-Aug. 2002.

[6] X. Y. Hong, K. X. Xu and M. Gerla, "An Ad Hoc Network with Mobile Backbones," Proc. IEEE ICC 2002, vol. 5, pp. 3138-43, Apr.-May 2002.

[7] E. M. Belding-Royer, "Hierarchical Routing in Ad Hoc Mobile Networks," Wireless Commun. And Mobile Comp., vol. 2, no. 5, pp. 515-32, 2002.

[8] Senthil Shanmugasundaram & Robert Lourdusamy "A comparative study of Text Compression Algorithm", International Journal of Wisdom based Computing, Vol. 1, December 2011.

[9] Suzanne Rigler, William Bishop & Andrew Kennings "FPGA-Based Data Compression using Huffman and LZ77 Algorithms", NSERC CANADA,2007.

[10] Ruimin Ma & Huan Yu Wang "A High-speed FPGA-based Lossless Data Compression Design for the X-ray Spectrometer Solar Energy Spectra", International Conference of Information Technology, Computer Engineering and Management Sciences, 2011.

[11] Joel Ratsaby & Denis Zavielov "An FPGA-based Pattern Classifier using Data Compression", 26-th Convention of Electrical and Electronics Engineers, IEEE - Vol. 1, pp. 320-324, March 2010.

[12] Mohamed A. Abd El ghany & Aly E. Salama and Ahmed H. Khalil "Design and Implementation of FPGA-based Systolic Array for LZ Data Compression", IEEE- Vol. 1, pp. 3691-3695, July 2007.

.