ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

Energy Efficient Hierarchical Clustering Based Routing Protocol with TDMA for Wireless Sensor Networks

R. Swathi, S. Kirubakaran, C. Kamalanathan

PG Scholar, Dept. of ECE, Bannari Amman Institute of Technology, Sathyamangalam, India

Assistant Professor, Dept. of ECE, Bannari Amman Institute of Technology, Sathyamangalam, India

Assistant Professor, Dept. of ECE, Bannari Amman Institute of Technology, Sathyamangalam, India

ABSTRACT: A Wireless Sensor Network (WSN) is built from a network of devices referred to as nodes, which can sense the environment and are used to communicate the required information collected from monitored fields. The communication and collection of information is done with the help of wireless links, where each node is linked to one or a number of sensors. The nodes in turn are deployed in large numbers depending upon the application where it is used. In modern days the process is bidirectional where the sensor activity can be monitored. The longevity in network lifetime is necessary as they have limited battery power which is to be used efficiently. This has led to new horizons upon energy saving and reduced power consumption among sensor nodes. Hierarchical cluster-based routing protocols are considered as one of the most efficient routing protocols in wireless sensor networks (WSN) due to its higher energy efficiency, network scalability, and lower data retransmission. An energy efficient hierarchical routing protocol LEACH-EN based on LEACH-C is designed by introducing the concept of TDMA scheduling along with compression of data from each node using LZW compression technique. The proposed LEACH-EN seems to perform well as collision is avoided using TDMA and hence data can be transmitted faster.

KEYWORDS: TDMA; energy efficient algorithm; data compression; Lempel-Ziv-Welch; shortest job first

I. INTRODUCTION

Wireless Sensor Networks are tiny sensor nodes with the main purpose of sensing, computation and communication about physical and environmental activities which in turn is used in wide range of applications such as civilian, healthcare, habitat monitoring etc., in our day to day life. WSNs are battery operated sensing devices where replenishing the batteries is impossible and hence an energy saving method has to be employed to save power.

Routing is an important concept in sensor networks as they deal with the process of data dissemination and data gathering after which the shortest possible and efficient path to reach the destination is chosen. In WSNs routing may be divided into flat routing and hierarchical routing. In flat routing, nodes are similar in carrying out tasks which proves to be a disadvantage when it comes to a larger environment. In flat routing the energy in wasted at times of data processing and thus the limited bandwidth which is allocated has not been used efficiently

In hierarchical routing nodes are dissimilar in the tasks which they carry out. Thus in terms of energy they prove to be efficient when compared to flat routing as they utilize the bandwidth efficiently. Clustering technique has been introduced in which there is a Cluster Head (CH) and the respective cluster members. Nodes with less energy are chosen as cluster members and that with high energy are chosen are cluster heads which then carry out the sensing process.

II. RELATED WORK

The paper is organized as follows: Section III provides an overview of the proposed algorithm wherein cluster formation using the LEACH-C based approach is done. In LEACH-C based approach the node with the highest energy is chosen as the CH and on each round of data the node when it reaches its threshold energy will move on to the next node with the highest energy to become the CH. The LZW compression technique is used to compress data wherein a dictionary based approach is followed having index entries for each and every data. LZW technique is a lossless data compression technique wherein the original data is obtained at the receiving end. Time slots are allotted to data at each

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

round. The shortest job first algorithm is used to send data from the CM to the CH. The node with the shortest job will send data and this cycle follows depending on the length of the job. In Section IV the simulation results performed using the NS2 network simulator is shown and the paper is concluded in Section V.

III. PROPOSED ALGORITHM

A. *Cluster Formation*

During the set-up phase of LEACH-C, each node sends information about its current location and energy level to the base station. The base station runs an optimization algorithm to determine the clusters for that round. The clusters formed by the base station will in general be better than those formed using the distributed algorithm. However, LEACH-C requires that each node transmit information about its location to the base station at the beginning of each round. This information may be obtained by using a global positioning system (GPS) receiver that is activated at the beginning of each round to get the node's current location. The life time of the cluster head is increased due to the randomized rotation of the cluster. Based on the energy level the cluster head is selected. The sensor node with higher energy is converted as the cluster head.

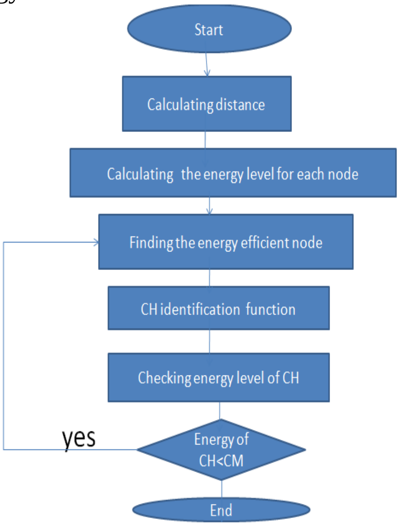

Fig.1. Flow chart for LEACH-C

B. *Implementation of Lempel-Ziv-Welch Algorithm*

Lempel–Ziv–Welch (LZW) is a universal lossless data compression algorithm . The algorithm is simple to implement, and has the potential for very high throughput in hardware implementations. It was the algorithm of the widely used UNIX file compression utility compress, and is used in the GIF image format. A particular LZW compression algorithm takes each input sequence of bits of a given length (for example, 12 bits) and creates an entry in a table (sometimes called a "dictionary" or "codebook") for that particular bit pattern, consisting of the pattern itself and a shorter code. As input is read, any pattern that has been read before results in the substitution of the shorter code, effectively compressing the total amount of input to something smaller. Unlike earlier approaches, known as LZ77 and LZ78, the LZW algorithm does include the look-up table of codes as part of the compressed file. The decoding program that un-compresses the file is able to build the table itself by using the algorithm as it processes the encoded input.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

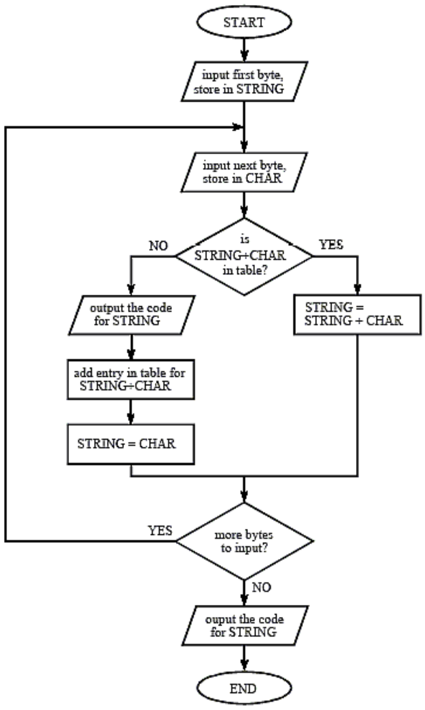

Fig. 2. Flow diagram for compressing data

Step 1: Encoding

- Initialize the dictionary to contain all strings of length one
- Find the longest string W in the dictionary that matches the current input
- Emit the dictionary index for W to output and remove W from the input
- Add W followed by the next symbol in the input to the dictionary
- Go to Step 2

A dictionary is initialized to contain the single-character strings corresponding to all the possible input characters (and nothing else except the clear and stop codes if they're being used). The algorithm works by scanning through the input string for successively longer substrings until it finds one that is not in the dictionary. When such a string is found, the index for the string without the last character (i.e., the longest substring that *is* in the dictionary) is retrieved from the dictionary and sent to output, and the new string (including the last character) is added to the dictionary with the next available code. The last input character is then used as the next starting point to scan for substrings.

Step 2: Decoding

The decoding algorithm works by reading a value from the encoded input and outputting the corresponding string from the initialized dictionary. In order to rebuild the dictionary in the same way as it was built during encoding, it also obtains the next value from the input and adds to the dictionary the concatenation of the current string and the first character of the string obtained by decoding the next input value, or the first character of the string just output if the next value cannot be decoded (If the next value is unknown to the decoder, then it must be the value that will be added to the dictionary this iteration, and so its first character must be the same as the first character of the current string being sent to decoded output). The decoder then proceeds to the next input value (which was already read in as the "next value" in the previous pass) and repeats the process until there is no more input, at which point the final input value is decoded without any more additions to the dictionary.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

C. *TDMA based scheduling*

The TDMA based scheduling is done based on the set-up phase and the steady state phase. Since each source node does not always have data to send, the duration of each frame is not fixed. Each frame consists of a schedule period and a data transmission period. During each schedule period, cluster head set up a TDMA schedule for collision-free intra-cluster communication. In the first schedule of each round, cluster head assigns the same time slot to cluster members. In the other schedule period, cluster head assigns different time slot to source nodes. Each data transmission period is divided into time slots equal to the number of source nodes in a cluster. Each source node sends its data and expected traffic load of next frame to the cluster head over its allocated transmission time slot, and keeps its radio off at all other time. All non-source nodes have their radios off during the data transmission period.

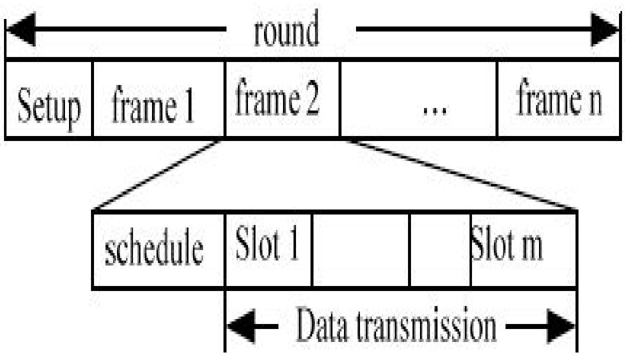

Fig. 3.Scheduling data by means of time slots

The shortest job first algorithm is used in order to get data from the cluster members. The cluster member with less amount of data packets after compression is allowed to send its data to the CH and hence this process continues as time progresses. The main objective of using this algorithm is that it reduces time delay and hence wastage of energy at each node is avoided. If a node has large amount of data to be transferred and a nearby node has less amount of data packet then, the node with lesser data has to wait for a longer period of time until the other node finishes sending. The shortest job first concept avoids this problem and allows the node with lesser data packets to send the data. The collision in sending data packets is also avoided using this concept.

IV. SIMULATION RESULTS

The simulation studies involve the network with 22 nodes in Fig.1. The proposed energy efficient algorithm is implemented with NS2. Node 22 is considered as the Base Station (BS) denoted by green color. The LEACH-C concept is used and nodes group into clusters and the node with the highest energy in each cluster is elected as the CH. 1, 6, 13 are elected as CH. Fig. 2 shows that after a certain period of time, there is an energy drop in each of the nodes and now 2, 7, 10 become CHs. Fig.3 shows the window where data to be transmitted is entered for each node. Fig.4 shows that nodes 1, 4, 5, 17 the CMs of CH 1 compressing data. The white color denoted that nodes are compressing data. In Fig.5 nodes in pink color are requesting the CH to receive data and are ready to send the data. In Fig.6 the nodes are ready to send the compressed data to CH. When all nodes send at the same time collision may occur and hence in Fig.7 the node 1 with the shortest job sends its data to the CH while nodes 4, 5, 17 are waiting to send. After a few seconds the node with the shortest job sends the data and thereby the process continues. In Fig.8 the BS decompresses the sent compressed data and hence the original data is obtained on the receiver side.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

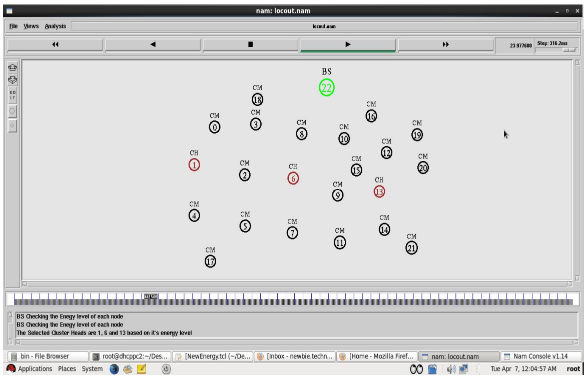

Fig. 4. Formation of nodes

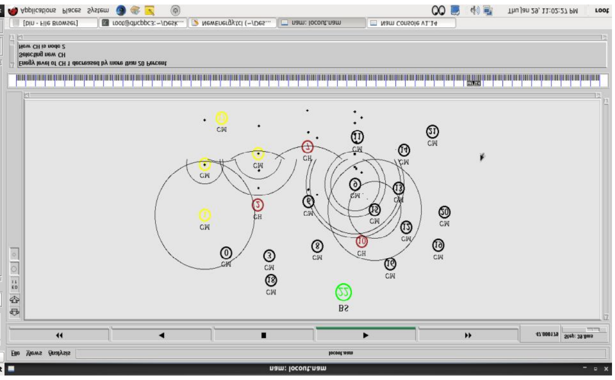

Fig. 5. Cluster head selection

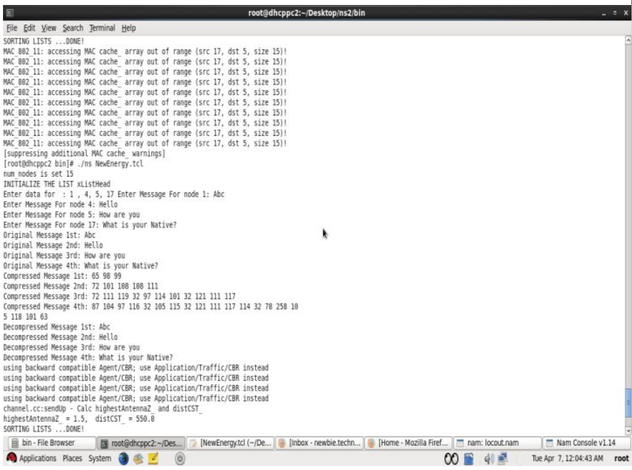

Fig. 6. Enter the data

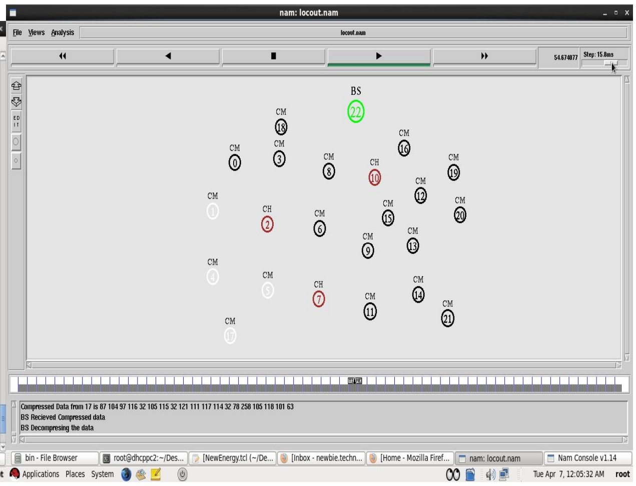

Fig. 7. Compression of data by nodes

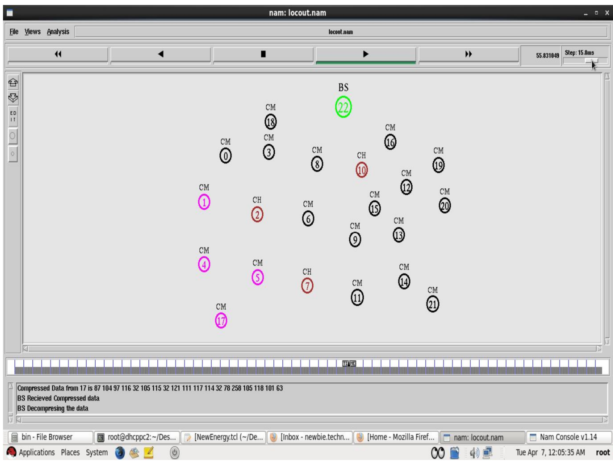

Fig. 8. Requesting CH to send data

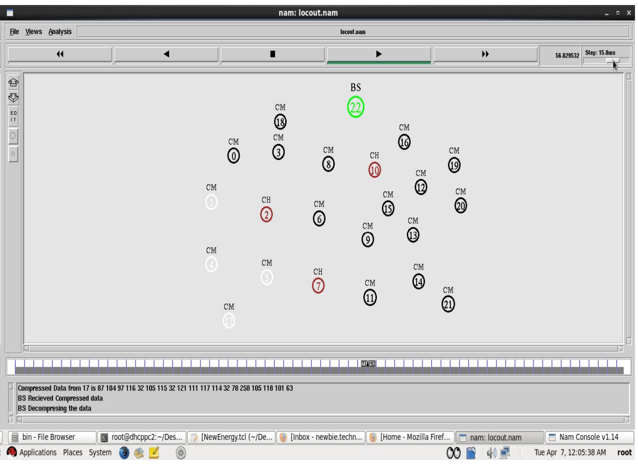

Fig. 9. CM ready to send data

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

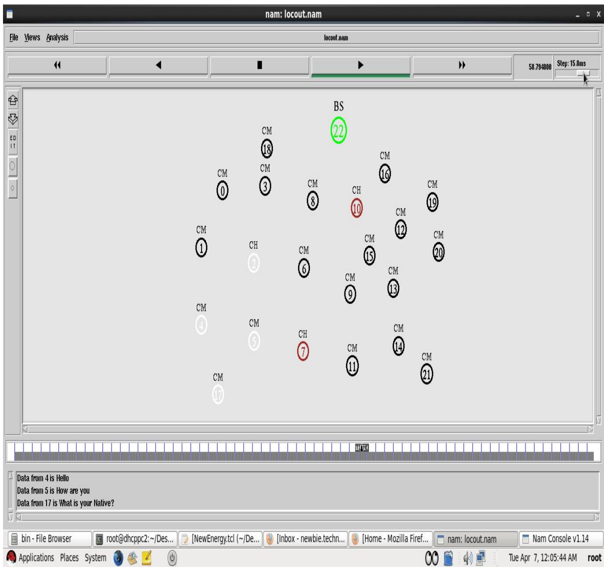

Fig.10. CM with SJF sends data to CH

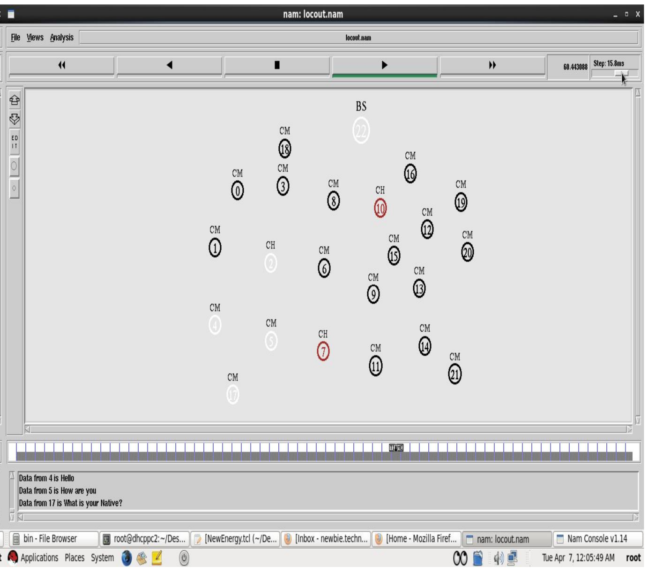

Fig. 11. BS decompresses data

V. CONCLUSION

The energy efficient hierarchical cluster based routing protocol seems to provide efficient routing based on two features. Firstly, the movement of nodes among the clusters will thus improvise the energy, as a node move closer to a base station hence distance is reduced for transmission. After consideration of this scenario the energy level is measured using LEACH-C based approach. Secondly, compressing data at the sending end and decompressing it with LEACH-C based approach for the energy level monitoring can prove to be energy efficient. TDMA scheduling avoids collision, control overhead and idle listening thus improving network lifetime.

REFERENCES

1. Y. Yu, V.K. Prasanna, and B. Krishnamachari, Information Processing and Routing in Wireless Sensor Networks, chapter 1, World Scientific Pub Co Inc, 2006.
2. A.M. Zungeru, L. Ang, and K.P. Seng, “Classical and swarm intelligence based routing protocols for wireless sensor networks: A survey and comparison,” Journal of Network and Computer Applications, vol. 35, no. 5, pp. 1508–1536, September 2012.
3. M.C. Arboleda and N. Nasser, “Comparison of clustering algorithms and protocols for wireless sensor networks,” in Canadian Conference on Electrical and Computer Engineering, CCECE ’06’.
4. W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan,“Energy-efficient communication protocol for wireless microsensor networks,” in Proceedings of the 33rd Annual Hawaii International Conference on System Sciences.
5. O. Younis and S. Fahmy, “Heed: a hybrid, energy efficient, distributed clustering approach for ad hoc sensor networks,” IEEE Transactions on Mobile Computing, 2004.
6. M. Chatterjee, S.K. Das, and D. Turgut, “WCA: A Weighted Clustering Algorithm for M obile Ad Hoc Networks,” ClusterComputing, pp. 193-204, 2002.
7. M . J. Handy, M . Haase, D. Timmermann, Low energy adaptive clustering hierarchy with deterministic cluster-head selection [C], Proceedings of IEEE International Workshop on Mobile and Wireless, Communications Networks, 2002, 368-372
8. Jian Xu, Dandan Qin “A new LEACH-Based Routing Clustering Protocol in WSN”, Journal of information & computational Science (2013), 6005-6011, http://www.joics.com.