An Energy Efficient routing with Rugged Data Compression for Wireless Sensor Networks

Ashwin G.B.[1] Dr. H.N Suresh[2]
[1]M.Tech Scholar [2]Professor
[1,2]Department of Electronics and Instrumentation Engineering
[1,2]Bangalore Institute of Technology. Bangalore, India

Abstract— One of the primary parameters to be considered in the configuration of Wireless Sensor Networks (WSN) is the energy consumption of the nodes and the data throughput. Since the nodes are controlled by batteries with lower energy limit, it is required to minimize the energy utilization. Henceforth a proficient routing technique in light of LEACH protocol is proposed alongside the utilization of A-LEC data compression strategy. The simulations are done in Network Simulator 2 (NS2).

Key words: Multi-hop routing, LEACH, LEC compression, Energy consumption

I. Introduction

In a Wireless Sensor Network, the fundamental capacity is to gather the data from the sensor nodes and transmit it to the sink in a multihop style. Sending the data is more energy effective contrasted with sending the data straightforwardly to the sink. The principle inconvenience of the multihop strategy is that nodes close to the sink must forward more number of packets contrasted with the ones far from the sink. Thus these nodes die early[2]. This is known as "The Hotspot Problem"[1].

One of the approaches to tackle the hotspot issue is to disperse the energy utilizing a superior sending method. They have demonstrated that changing the transmission energy of nodes, does not solve the hotspot problem[2]. The target here is to increase the lifetime of WSN through a superior routing calculation.

The seriousness of the hotspot issue relies on upon the condition of the sensor nodes or the sink, i.e whether they mobile or not. By conveying mobile sensors energy can be adjusted in light of the fact that the sensors can modify their position to adjust the energy utilization in territories that have nodes transmitting a gigantic measure of data. Conveying versatile sinks and nodes expands the expense and base. What's more, there are applications wherein portability is unreasonable. In this paper, the situation where nodes are fixed is considered and a routing calculation alongside data compression is proposed. By extensive simulation the demonstration of the algorithm is performed. Since detected data shows high temporal correlation, data compression is a productive way to deal with decrease power-utilization of WSN nodes. Late advances in lossless compression incorporate Sensor-LZW calculation, Lossless Entropy Compression Algorithm in which entropy coding methods are used. Data compression can be used along with an efficient routing in order to conserve the energy.

II. Survey of existing methods

The two areas to be dealt with are:

A. *Data Routing:*

The fundamental undertaking of a routing calculation is to choose a path from an arrangement of accessible paths, that is most effective as far as energy minimization is concerned. There is more focus on WSN multi-hop routing and numerous calculations have been proposed on the same [3]. They are predominantly portrayed as single level multi-hop routing and varioleveled multi-bounce calculations.

1) *Flat Multi-Hop Routing Algorithm:*

Fig1 shows illustrates how flat multi-hop routing is used to send data. Data from different nodes is forwarded to the sink node A. The design metric here is the total power consumed by the network. The energy consumed is given by the following equations:

$$\text{link-cost}(p,q) = e_t(p) + e_r(q)$$
$$e_t(p) = \varepsilon_1 r_{i,j}^2 + \varepsilon_2$$
$$e_r(q) = \varepsilon_3$$

where link-cost is the amount of energy consumed for transmitting a unit of data from node p to q. es(p) is the energy consumed by transmitting a unit of data to the receiving node q and the value is proportional to the square of distance between the nodes. er is the energy consumed for receiving a unit of data from p. ε1, The path which has minimum sum of link-costs is chosen. Thereby WSN's total power consumption is minimized.

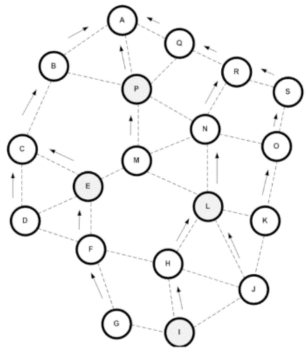

Fig. 1: Flat Multi-hop Routing

Defining link-cost in the above mentioned way, would over-exhaust certain nodes, resulting in rapid decrease in their energy. An effective algorithm which uniformly distributes the energy consumption over each node, addresses the problem by redefining Link-cost(p,q) as :

$$\text{Link-cost}(p,q)_{new} = \text{Link-cost}(p,q)/E_p^2$$

,

where Ep is the residual energy of node p. The possibility of a node being selected as a forwarding node decreases as its remaining energy decreases.

2) *Heirarchical Multi-Hop Routing:*

In this type of routing the set of sensor nodes are divided into clusters. Each cluster consists of a cluster head which forwards the data to the sink. It is found that hierarchical routing is more efficient in conserving energy since it drains energy equally from all the nodes. The following diagram illustrates the method.

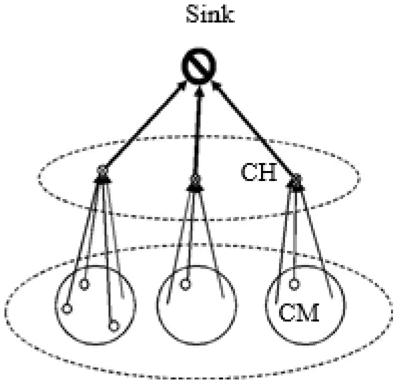

Fig. 2: Hierarchical Routing

The most notable of the multi-hop routing algorithms is LEACH (Low Energy Adaptive Clustering Hierarchy). It is a two layered hierarchical algorithm. Each node can either be a cluster head or a cluster member. Each node can be renewed in a time interval, termed as a round. At the beginning of each round a node declares itself as a CH with certain probability. CM choose the CH's close to it and transmit the data to the CH belonging to the same cluster. The diagram below is an example of LEACH.

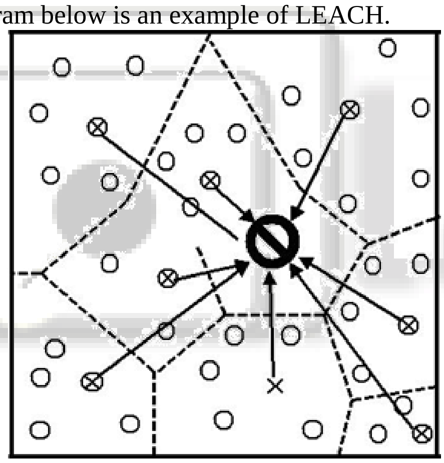

Fig. 3: Topology of LEACH

B. Lossless Compression:

The attention is on lossless compression systems. S-LZW is a lightweight compression calculation for memory constrained WSN's [3], which is an adjustment of the popular Lemplel Ziv Welch (LZW) calculation. The authors in [5] have utilized S-LZW planned particularly for resource compelled WSN's. It fundamentally utilizes versatile dictionary systems with dynamic code length. The dictionary structure permits the calculation to adjust to changes in the data and to exploit the redundancy in the detected data. Be that as it may, it experiences the developing reference issue and the span of the code increments

LEC [4] is based on predictive coding in which predictor and encoder is used. For a new data sample x_i in a series, $\hat{x}_i$ is generated and the residue $d_i = x_i - \hat{x}_i$ is calculated. The residue d_i is then entropy coded and transmitted to the receiving node. In LEC, for prediction model the differential predictor is adopted, $\hat{x}_i = x_{i-1}$. The residue or error d_i is then coded using Exponential-Golumb code of zeroth order. While LEC possess better compression performance that S-LZW, it lacks robustness wherein the predictor cannot effectively exploit temporal correlations for diverse WSN data sets, because different sets exhibits different correlation patterns. Hence, it may show good compression for a particular data set and poor compression for others.

In [4], the creators have fused Huffman coding into Wireless sensor nodes. The technique was to utilized Lossless entropy Compression (LEC) calculation which depended on static Huffman coding, abusing the temporal relationship that exist in data to acquire a compacted variant utilizing a little word reference. The calculation is static and thus does not adjust to source data measurements. In [5], the calculation proposed is an altered versatile Huffman coding. It doesn't require earlier data of the insights of the source data and compression is performed adaptively in light of the temporal relationship existing in the source data. The disadvantage of this calculation is that it is computationally economical.

The LEC calculation has less computational multifaceted nature and requires less memory for its execution. Be that as it may, it can't adjust to changing correlation in sensor measured data. Subsequently compression proportion got is not ideal. Henceforth, in this paper a powerful compression method is proposed known as A-LEC (Adjacent LEC) which misuses the transient correlation in the detected data all the more viably.

III. PROPOSED METHOD

Here to minimize the energy utilization of the nodes a strategy including effective routing with data compression is proposed. Filter calculation is utilized to course the data and A-LEC technique is utilized to pack the data got by the nodes. The underlying phase of routing includes setting up of groups. Once the groups are set up, the control parcels are transmitted to pick the bunch head. Once the bunch head is picked, the data packets are transmitted from one group head to the next lastly to the sink. The topology utilized is demonstrated as follows.

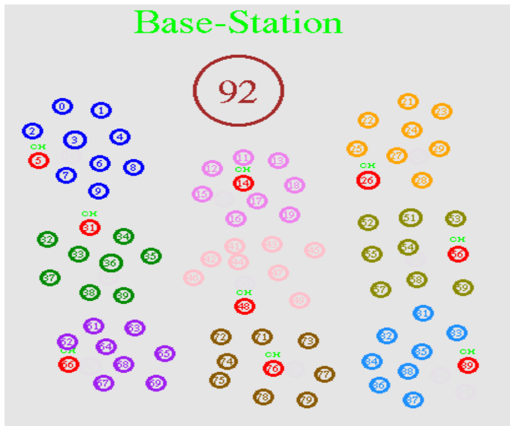

Fig. 4: Topology Showing CH in red

The nodes marked in red are the cluster heads. The rounds are repeated at constant time intervals.

The data is compressed using A-LEC method (Adjacent Lossless Energy Compression). In LEC, the data collected by the nodes is given to a difference computation block whose output is a sequence of residues r_i (i=1,2, . . M). The residue sequence is encoded by the entropy encoder and are considered to have no correlation amongst each other and hence are coded independently. The central idea here is to formulate a coding method that would exploit the temporal correlation in the residue sequence and thereby resulting in higher compression ratio. The residue sequence is coded using the following table:

` n	h(n)	d(n)	
0	00	0	
1	010	-1,+1	
2	011	-2,+2,-3,+3	
3	100	±4, · · · ± 7	
4	101	±8, · · · ± 15	
5	110	±16, · · · ± 31	
6	1110	±32, · · · ± 63	
7	11110	±64, · · · ± 127	
8	111110	±128, · · · ± 255	
9	1111110	±256, · · · ± 511	
10	11111110	±512, · · · ± 1023	
11	111111110	±1024, · · · ±	2047
12	1111111110	±2048, · · · ± 4095	
13	11111111110	±4096, · · · ± 8191	
14	111111111110	±8192, · · · ± 16383	
15	1111111111110	±16384, · · · ± 32767	

Table 1: LEC Coding

LEC Algorithm: Modified version of Golumb code is used to code the residues. The codeword consists of two parts : The entropy code specifying the group(hi) and the binary code representing the index in the group(Index of di). The residue is represented by hi|di. For example consider the coding of the sequence : r1=9,r2=128,r3= -130 and r4=16. This is encoded as : 101|10,111110|00,111110|100,110|00. The index of ri can be found out using:

$$(Index)_i = 2[|r_i| \bmod (2^{n(i)-1})] \quad \text{if } r_i > 0$$

$$(Index)_i = 2[|r_i| \bmod (2^{n(i)-1})] + 1 \quad \text{if } r_i < 0$$

A-LEC (Adjacent LEC) Algorithm : Here the LEC algorithm is modified and 4 status variables are used to track the temporal correlation.

S_i	Context Data	Codew+ord for $h(n_i)$	Group
00	$n_i = n_{i-1}$	No codeword for h	Same
01	$n_i = (n_{i-1}) - 1$	No codeword for h	Neighbor above
10	$n_i = (n_{i-1}) + 1$	No codeword for h	Neighbor Below
11	Otherwise	$h(n_i)$	Otherwise

Table 2: A-LEC Coding

For the residue sequence: r1=9,r2=128,r3= -130 and r4=16. This is encoded by A-LEC as :101|1001, 11|111110|00, 00|100,11|110|00. Since the WSN data has a high temporal correlation, this method can exploit the correlations, by omitting the codeword for h if the next residue belongs to same/neighboring group w.r.t the previous one. Here instead of using hi|di, si| hi|di is used.

IV. PERFORMANCE EVALUATION

The performance of the network is studied under 4 scenarios and comparisons are made.

1) CH-BS transmission without compression
2) CH-BS transmission with compression
3) CH-CH transmission without compression
4) CH-CH transmission with compression

The simulations are conducted in NS2 and the following graphs for the residual energy was obtained.

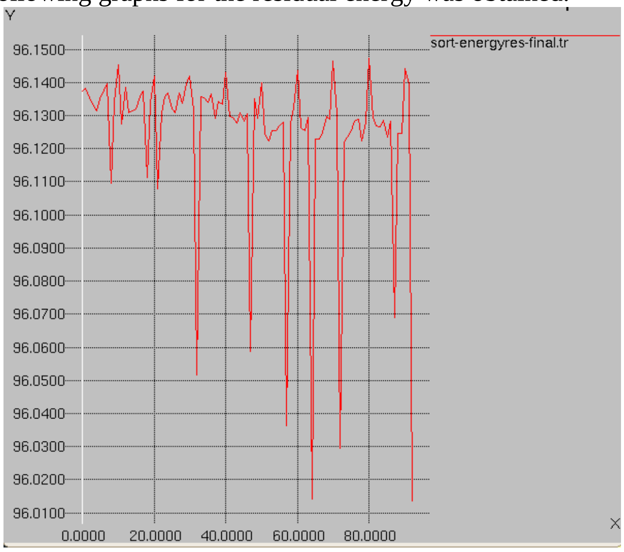

Fig. 5: CH-BS without Compression

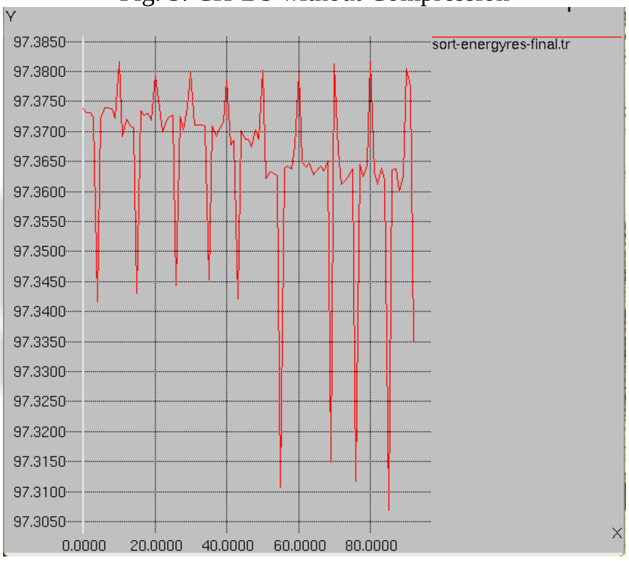

Fig. 6: CH-BS with Compression

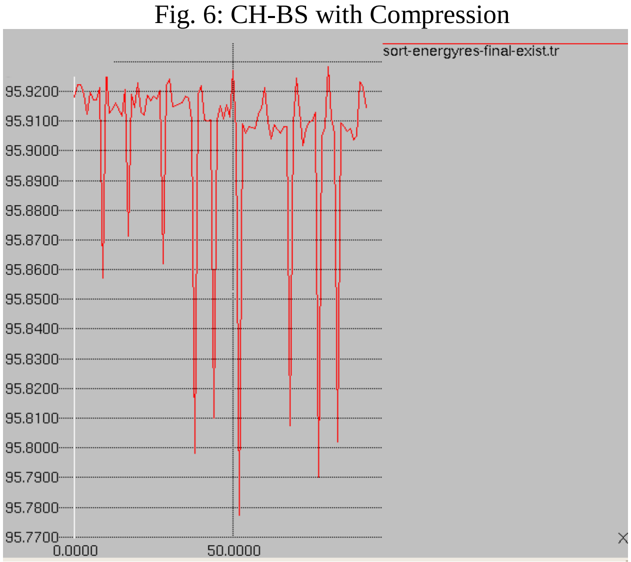

Fig. 7:

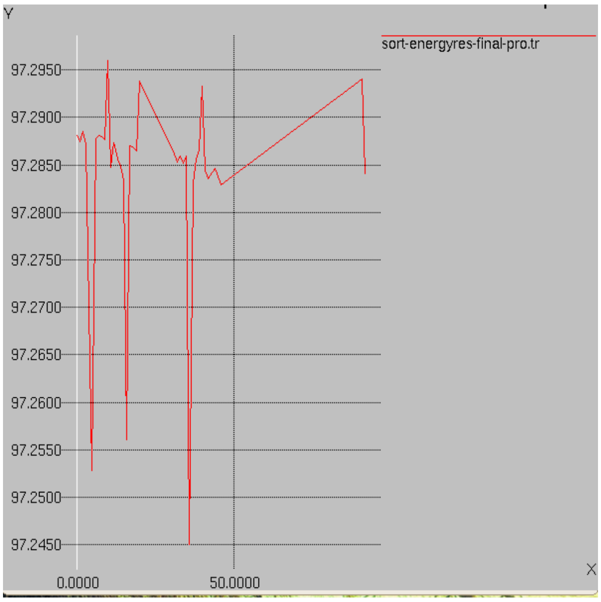

Fig. 8: CH-CH with Compression

Initially the energy of the nodes was 100 Joules. After several rounds of operation the final energy possessed was found to be more in CH-CH transmission with the use of A-LEC data compression. The compression performance was evaluated by compression ratio as follows:

CR = [1-(z/$\overline{z}$)] where z and $\overline{z}$ denote the original raw data size and compressed data size in bits respectively.

A. *Data Sets:*

The environmental monitoring real world data sets from Sensor-Scope [7] are used. The comparison of LEC and S-LZW is made for temperature and relative humidity measurements. Le-Genepi and stbernanrd from Sensor Scope deployments are tested. The size of the data sets range from 31253 to 71536 samples. The temperature and humidity readings are connected to an ADC. The outputs of ADC for the raw relative humidity and raw temperature are represented with resolutions of 14and 12 bits. These raw outputs are then converted [8] to physical measures expressed as t and h as shown by the code below. The t and h values are then used as inputs for A-LEC Compression

```
void main()
//----------------------------------------------------------------------------------
// calculates temperature [°C] and humidity [%RH]
// input : humi [Ticks] (12 bit)
// temp [Ticks] (14 bit)
// output: humi [%RH]
// temp [°C]
{ const float C1=-2.0468; // for 12 Bit RH
 const float C2=+0.0367; // for 12 Bit RH
 const float C3=-0.0000015955; // for 12 Bit RH
 const float T1=+0.01; // for 12 Bit RH
 const float T2=+0.00008; // for 12 Bit RH
 int *p_humidity;
 int *p_temperature;
 float rh; // rh: Humidity [Ticks] 12 Bit
 float t; // t: Temperature [Ticks] 14 Bit
 float rh_lin; // rh_lin: Humidity linear
 float rh_true; // rh_true: Temperature compensated humidity
 float t_C; // t_C : Temperature [°C]
 printf("Enter the ADC output for Humidity");
 scanf("%d",p_humidity);
 printf("Enter the ADC output for temperature");
 scanf("%d",p_temperature);
 rh_lin=*p_humidity;
 t_C=*p_temperature;
 t = (t_C + 40.1)/0.01; //calc. temperature[°C]from 14 bit temp.ticks @5V
 rh = (-C2 + pow [(C2^2)-4*C3*(C1-rh_lin)),0.5)]/2*C1;                    //calc. hu
 *p_temperature=t; //return temperature [°C]
 *p_humidity=rh; //return humidity[%RH]
 printf("The humidity is %f",*p_humidity);
 printf("The temperature is %f",*p_temperature);
 getch();
clc:
```

Fig. 9: C Code for conversion to raw t and rh values

B. *Lossless Compression:*

The size of each data block for compression is 256 samples The table below shows the comparison between different methods of compression.

Variable	Compression Ratio		
	A-LEC	S-LZW	LEC
Temp. LU_ID53	71.19	51.23	69.46
Temp. LG_ID12	55.87	22.04	51.39
RH LU_ID71	65.13	36.25	64.59
RH LG_ID15	50.08	18.26	49.84

Table 3: Comparison of Compression Ratios for Temperature and Relative Humidity Data

It can be seen from the table that the A-LEC method outperforms the LEC and the S-LZW methodIt can be seen from the table that the A-LEC method outperforms the LEC and the S-LZW method.

V. CONCLUSION

The routing of data in the networks is performed using hierarchical multi-hop method. Simulations are conducted in NS2 [6] in different scenarios and the one with Hierarchical Multi-hop routing is found to be most efficient. A-LEC compression method is adopted and is compared with S-LZW and LEC compression methods. The A-LEC shows a higher degree of ruggedness and hence it is adaptive to changes in the data pattern.

REFERENCES

[1] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wirelesssensor networks: A survey," Comput. Netw., vol. 38, no. 4, pp. 393–422, Mar. 2002.
[2] G. Anastasi, M. Conti, M. D. Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," Ad Hoc Netw., vol. 7, no. 3, pp. 537–568, May 2009.
[3] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," Ad Hoc Netw., vol. 3, no. 3, pp. 325–349, May 2005.
[4] C. M. Sadler and M. Martonosi, "Data compression algorithms for energy-constrained devices in delay tolerant networks," in Proc. 2006 International Conference on Embedded Networked Sensor Systems, pp. 265–278.
[5] F. Marcelloni and M. Vecchio, "A simple algorithm for data compression in wireless sensor networks," IEEE Commun. Lett., vol. 12, no. 6, pp. 411–413, 2008.
[6] The Network Simulator – ns-2, http://www.isi.edu/nsnam/ns/
[7] 7.(2013, Oct.) Sensorscope homepage. Available: http://lcav.epfl.ch/op/ edit/sensorscope-en
[8] 8.(2013, Oct.) Sensirion homepage. Available: http://www.sensirion.com