A Novel Data Compression Algorithm in Wireless Sensor Network

Vipin Kumar Rathi
Department of Computer Science
Ramanujan College, New Delhi
vipkrathi2013@gmail.com

Ankur
Department of Computer Science
NIT Delhi, New Delhi
ankurbohora@gmail.com

Bhawna Chaudhary
School of Computer and Systems Sciences
Jawaharlal Nehru University
bhawna0101@gmail.com

Abstract— The data compression techniques use variable length bit code to send over the network in which some of the smaller length bit codes remain unused while encoding schemes uses larger bit codes for representation. In this paper we represent a new bit representation technique for data compression that transfers lower number of bits into the network and saves transmission energy of sensor nodes. Simulation results show that the new technique for data compression reduces energy consumption and improve network lifetime.

Keywords- Bit Representation, Data Compression, Data Aggregation, Wireless Sensor Network.

I. INTRODUCTION

Energy is the main constraint in Wireless Sensor Network as sensors are deployed in an unattended area and sensors cannot be recharged regularly. Therefore sensors must operate in energy saving environment. The major tasks performed by a sensor node are data sensing, data processing, and data transmission. Data transmission consumes most of sensor’s energy as it requires radio communication and hence it is required for a sensor to send least possible number of bits over the network. For transmitting data over the network, sensors use radio transmission where most of the sensor’s energy is consumed. Hence data compression is required before transmitting data for energy saving of a sensor in the network.

To reduce the number of bits to be sent over the network, we do not send the actual sensed data rather we send the difference in previous sensed data and current sensed data. Sending the difference will reduce the number of bits as compared to send the actual sensed data. At the receiver end the current data can easily be computed by adding the difference to the previous data.

In general, Binary representation is used for encoding of data which uses more number of bits and hence more energy in transmitting the data. As we know radio communication is the main cause of energy consumption, we can increases the life time of network by reducing transmission and reception of data, by using data compression and compression. In our algorithm the difference is sent over the network which is represented by its novel data representation technique.

Compression can be helpful in power saving only if the execution of compression algorithms does not require an amount of energy greater than the one saved in reducing transmission. Basically compression and compression algorithms are designed for the storage saving, not energy.

we purposed a different bit representation for aggregating the data. It reduces the memory space as well as energy, without additional computational energy and loss less recovery of old data. The comparison is done on the basis of various performance metrics: Total data send by network, Numbers of nodes alive per round, load distribution, numbers of packet receive at base station, total remaining energy of the network and network stability.

II. PROBLEM STATEMENT

Transmitting data into network consumes most of sensor's energy. The large is number of bits the large is the energy consumption. In spite of the above reason mentioned, Currently two types of encoding schemes have been proposed for encoding, Fixed length encoding scheme and variable length encoding scheme. In fixed length encoding scheme each symbol is represented by same number of bits and hence k-bit code supports 2^k different symbols. for example 7 bit ASCII code is a fixed length code. In variable length encoding scheme like Huffman coding different symbols are represented by different number of bits. In both the schemes, there are certain blocks of bits which do not represent any symbol and hence those bit sequences become unused in encoding.

In fixed length representation, number of bits required for a number to be encoded are equal to the bits required for the largest number in the block, for instance if largest number in a block is 16 then a numbers 0, 1, 2, 3, 4, 5, 6, 7 also required 4 bits to be encoded. In this encoding scheme certain bit sequences represent nothing. In variable length scheme, there are various encoding schemes have been proposed which overcome the limitations of fixed length encoding scheme but those schemes also have certain bit sequences which do not represent any number. In such cases, larger number of bits need to be assigned to encode further numbers. For instance, bit sequences 0 represents a number 0, bit sequences 1 represents a number 1, bit sequences 10 represents number 2, bit sequences 11 represents number 3, bit sequences 100 represents number 4 and so on. But the following bit sequences 00,01,000,001,010,011,0000,0001,0010 are representing the same numbers which are already been encoded by any other bit sequence or represent nothing. This new compression protocol provides the technique to use these bit sequences which can be further utilized to encode different numbers and transmit the difference rather transmitting the actual data. At the receiver side the actual data is being computed by adding the difference to the last data received. In other words, the actual data is being computed by aggregating the last received data and the currently received difference d_i.

III. PROPOSED SOLUTION

After changing the bit representation we can use all those bit sequences which are not in use, without any extra computation energy. To reduce the bits in transmission, it is more efficient to compute the difference of consecutive data sensed by the sensor. Thus the new data is the difference of new and previous data. In general, difference is much smaller than the actual data and can be send using less number of bits. We can compress, these small variable size data using following method.

- **Sender Side:**

```
START
d_i = data − data_last
set n_bits = 0
if d_i is equal to zero
then set data = 0
exit
else if  d_i is greater than zero
        set  s_bits = 0
else
        set  s_bits = 1
end  if

 set   t_n = 2^n_bits
 repeat while t_n ≤ d_i
        increment, n_bits = n_bits + 1
    t_n = t_n + 2^n_bits
end  while

d_new = d_i − (2^n_bits − 1)
then aggregate data
data↑comp = s↓bits | 〖(d↓new)〗↓(n↓bits )
EXIT
```

At sender side, sender first calculates the difference (d_i) between last sensed data ($data_{last}$) and current sensed data ($data$). Initially set number of bits (n_{bits}) to zero. If difference is zero then send '0' bit sequence as data. If difference is greater than zero then set sign bit (s_{bits}) to zero, otherwise set sign bit to one. Then calculate a temporary number (t_n) which is equal to n_{bits} th power of 2. Then repeat while loop until temporary data is less than or equal to difference. In each iteration increment number of bits and increment temporary data with n_{bits} th power of 2.Then calculate the new difference by equation

$$d_{new} = d_i - (2^{n_{bits}} - 1)$$

After that send the data to receiver using above information by this format

$$(d_{new})_{n_{bits}})$$

first bit is sign bit and after that n_{bits} bits data. If d_{new} binary bit representations have fewer bits compare then n_{bits} then preamble zeros with d_{new} bit sequence. Then send $data^{comp}$ to the receiver.

- **Receiver Side:**

```
get data from sender data↑comp = s↓bits |(d↓new )↓(n

calculate
d_i = d_new + (2^n_bits − 1)
if s_bits is equal to zero
    data = data_last + d_i
else
    data = data_last − d_i
end
```

Using the $data^{comp}$ receiver first calculate the difference of last to data using following formula

$$d_i = d_{new} + (2^{n_{bits}} - 1).$$

Then the receiver checks if sign bit (s_{bits}) is equal to zero and then adds difference to last received data, if it is not equal to zero then subtract from the last data and calculate the original data by sending very less bits compared to the original data.

IV. SIMULATION AND RESULTS

Figure 1, shows the total data compression using the proposed technique for 250 bits of data. Data set is generated using binomial distribution with probability increasing from 0 to 1 in increment of 0.05. Figure 1 shows that at the start, probability of binomial distribution is 0.05 and data is 250 bits in size then we generate new data set size by using the following function: binornd(250, 0.05).

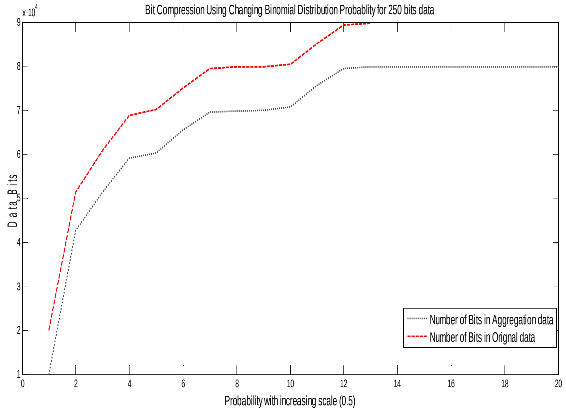

Figure 1: Bit Compression using changing Binomial distribution function with increasing probability 0.05 at each scale for 250 bit data

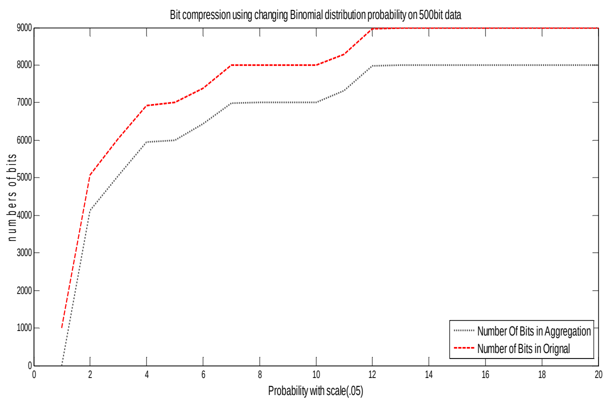

Figure 2: Bit Compression using changing Binomial distribution function with increasing probability 0.05 at each scale for 500 bit data

Like Figure1, Figure2 shows the same pattern for 500 bits data and results show that percentage of savings done by the proposed compression method decreases if data size increases. It works well for small size data.

V. CONCLUSION

The proposed algorithm reduces the number of bits works for both large size and small size data. Though it is found that for small size data this algorithm will work effectively. For small size data we have seen a greater percentage of saving bits as compared to large size data.

References

[1] Sadler, Christopher M., and Margaret Martonosi. “Data compression algorithms for energy-constrained devices in delay tolerant networks.” In *Proceedings of the 4th international conference on Embedded networked sensor systems*, pp. 265-278. ACM, 2006.

[2] Barr, Kenneth C., and Krste Asanović. “Energy-aware lossless data compression.” ACM Transactions on Computer Systems (TOCS) 24, no. 3 (2006): 250-291.

[3] Kimura, Naoto, and Shahram Latifi. "A survey on data compression in wireless sensor networks." *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*. Vol. 2. IEEE, 2005.

[4] Marcelloni, Francesco, and Massimo Vecchio. "A simple algorithm for data compression in wireless sensor networks." IEEE communications letters 12.6 (2008).

[5] Luo, Chong, et al. "Compressive data gathering for large-scale wireless sensor networks." Proceedings of the 15th annual international conference on Mobile computing and networking. ACM, 2009.