J. Basic. Appl. Sci. Res., 3(9)612-617, 2013
© 2013, TextRoad Publication
ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

Efficient Resource Preservation through Data Compression in Wireless Sensor Networks

Khalid Mahmood, Muhammad Zubair, Zeeshan Shafi Khan, Fazeel-Ud-Din Faraz

International Islamic University Islamabad, Pakistan

ABSTRACT

One possible way to achieve maximum utilization of resources in Wireless sensor networks (WSN) is to apply data compression on sensed data. However, classical compression algorithms are not applicable for sensor nodes because of their limited resources. The proposed compression framework shifts the data compression of sensed data into a dedicated layer and compresses the payload as well as the packet header. The framework splits sensed data from application into two halves, header and payload. HC1 (Header compression 1) encoding is performed to compress the header and simple stream based encoding is applied on the payload to achieve potential resource saving. The simplicity of proposed payload encoding scheme also potentially overcomes the frequent memory access that exists in the former encoding scheme and ensures maximum resource utilization of WSNs. IP is utilized in the compression framework through 6LowPAN adaptation layer so that sensing services can be provided at global scale.

KEYWORDS: Compression, 6LowPAN, IPv6, Payload, In-network processing

1-INTRODUCTION

Devices are becoming tiny and smarter due to remarkable hardware advancements over the years. These smart tiny devices are popular because these are low cost, low power and lightweight. Sensing devices are one of these tiny devices that have opened up the gate of many areas of interest. In spite of the fact that WSNs are extremely useful, resources of sensor nodes called motes in WSNs are limited. The major resources of these motes like power supply, bandwidth for communication, processing speed and memory space are all limited. This factor raises a question on the efficiency and worth of WSN but convenient and maximum utilization of these limited resources through some mechanism can maximize the efficiency and worth of WSN.

The WSN generally has two subsystems one is networking and another is sensing [1]. Both subsystems concern with different layers of data transmission, routing, sensor hardware architecture and miscellaneous network issues. Various energy conservation techniques applied to one of these different layers, are as follows [2]. Duty cycling is described as the amount of time nodes remain active during their existence either using topology control or using power management. Data-driven approaches focus on reducing the amount of data that is transmitted to the sink node. Data compression, data prediction, in-network processing and data aggregation are different methods that are utilized in data-driven approaches. In mobility-based approaches mobile data collectors are utilized to achieve energy efficient data collection by minimizing the data transmission path. Mobile nodes, mobile sink and mobile–relays are different ways to implement these approaches. Energy efficient routing protocols are used to reduce the energy consumption during network activities. All routing protocols are generally classified into data-centric, hierarchical and location based routing approaches [3].

All kinds of energy conservation schemes fall under one of these approaches. Data compression is the most common and simple energy conservation approach that we have used in this work. We have introduced an enhanced compression framework that compresses payload and header using separate encoding schemes.

The rest of the paper is organized as follows. Section 2 presents the related work. The problem analysis is presented in section 3. Section 4 presents the proposed solution including compression layer and proposed encoding scheme. The results are given in section 5 and section 6 is about conclusion.

2. Related Work

Most of the power expends during radio communication by radio transmitter on each mote of WSN. Therefore network lifetime can be increased by reducing the amount of radio transmissions and also more data can be delivered for long period of time [4] , [5]. This problem of rapid power depletion through frequent radio transmission can be sorted out through various kinds of techniques. These techniques are commonly named as energy conservation approaches.

K. Barr et al. has found that same amount of energy can be utilized for performing thousands of operations that is spent for transmitting a single byte of data [6]. C. M. Sadler et al**.** has stated in [7] that in order to

***Corresponding Author:** Khalid Mahmood, International Islamic University Islamabad, Pakistan.
khalid.phdcs74@iiu.edu.pk

compress the sensor readings, sensor system designers can have two choices at their end either develop application specific algorithm or use existing compression algorithms not designed for resource constraint sensor nodes. A Reinhardt et al. has proposed Squeeze.KOM which is a transparent compression layer and utilizes the stream-oriented compression scheme for sensor networks in order to preserve energy by reducing packet sizes [8].

We have utilized IP in our approach using 6LoWPAN. Use of IPv6 over IEEE802.15.4 imposes two major challenges with respect to frames; one challenge can be described as the IPv6 header overhead and the second as IEEE802.15.4 packet size of 127 bytes as compared to MTU of IPv6 that is 1280 bytes. 6LoWPAN utilizes compression scheme and fragmentation (for ripping single IPv6 packet into multiple frames) in order to cope these challenges and offer a decent economical solution to integrate IPv6 over IEEE802.15.4 [9], [10].

3. Problem Identification

Though proposed compression layer by [A Reinhardt et al. 2009] achieves good compression gain on per packet basis but this compression framework also has some issues that can lead to inefficient resource utilization. Like in compression framework every new payload packet is compared with each element in the local transmission history table in order to find similarity. If no similarity is found then history table is updated with this new payload packet and is transmitted without compression after assigning next available index. If all indices have been used then eldest index element which has not been used for long is replaced with the new element and hash value is calculated for it. If similarity is found between the new payload and history table's elements, then difference is calculated between the each index element I_n and the new payload P. At the end calculated differential Δ_n (of each index element with the payload) is transmitted to the receiver but along with corresponding $Hash\ (I_n)$ to ensure that same I_n is referred to both at the sender and receiver.

In short this mechanism makes use of the frequent memory access and results in significant history mismatch if no similarity in the incoming new payload with the history elements is found. Since the amount of energy consumed for transmitting a single byte can be utilized for processing thousands of computation operations, hence if only single bit is compressed by hundreds or even thousands of computation then significant amount of energy should be saved, whereas memory access has proved to be over 200 times more costly than computation [K. Barr et al. 2003, [C M. Sadler et al. 2006].

In addition, compression framework makes use of 1 byte status field as shown in Figure 1 to the payload to inform necessary information to the receiver like index number of element present in the history table and flags to indicate whether payload is compressed or not etc. So this status field leads to an extra overhead and potentially affects the resources.

```
Bit 0     1     2     3     4     5     6     7
 +-----+-----+-----+-----+-----+-----+-----+-----+
 | reserved   | enc | I/Δ | History element index |
 +-----+-----+-----+-----+-----+-----+-----+-----+
```

Figure 1: Bit structure of the status field [8]

Last but not the least, the whole architecture can be referred to as stand-alone and does not provide interoperability features with the other networks or sensing services to the outside world because it makes use of proprietary protocol for communication.

4. Proposed Solution

Two major issues of IPv6, header overhead and limited payload of IEEE 802.15.4, demand an adaptation layer in order to compress header and fragment large packet in order to fit in IEEE 802.15.4 frame limit. 6LowPAN resolves both of these issues, using header compression and fragmentation (optional for large payload) respectively by providing an intermediate adaptation layer between IEEE802.15.4 and IPv6. This makes the IPv6 compatible over WSNs by reducing its significant overhead. So this feature is added in the modified compression layer.

The compression framework described in [A Reinhardt et al. 2009] is enhanced by incorporating 6LowPAN (an intermediate layer for transmitting IPv6 packet over IEEE802.15.4). The sensed data received from application is encapsulated into IPv6 packet and then dispatched into compression framework for compressing both header as well as payload with two different compressing schemes for each in order to achieve significant compressibility for maximizing resource utilization. The internal structure of the proposed compression layer is depicted in Figure 2. The header and payload compression mechanism is described in detail now.

The proposed encoding scheme in this paper makes use of last single sent packet avoiding the need of keeping bunch of successive packets in the memory. This leads towards less frequent memory access and ultimately maximizes the network lifetime. The communication starts initially by sending first packet as it is.

The copy of this sent packet is maintained at the source and also at the destination when received. Now, when the next new payload packet P_i arrives, XOR operation is performed on new arrived payload P_i and last single packet payload P_{i-1}. $P_i \oplus P_{i-1}$ operation produces an output bit-stream with a confined number of set bits. Again new payload P_i replaces P_{i-1} at the source end.

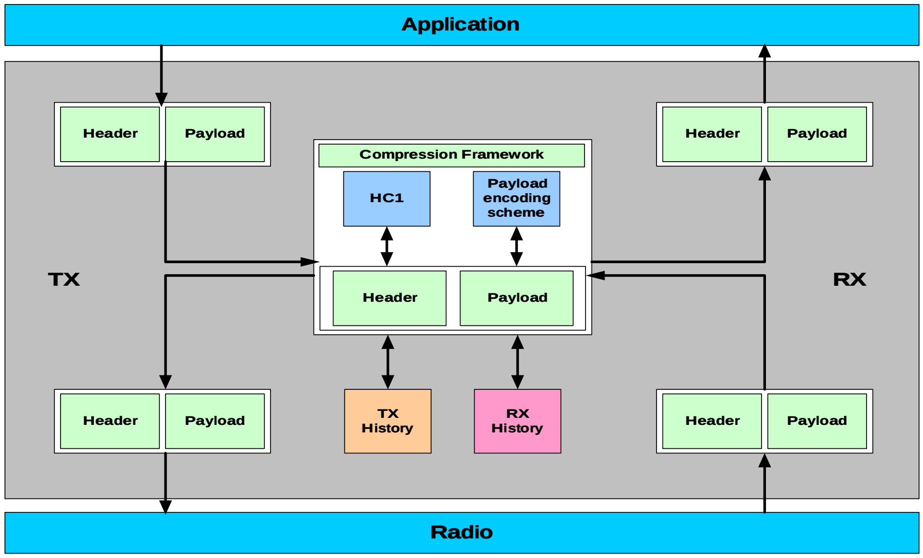

Figure 2: Internal structure of proposed compression layer

To compress the bit-stream after XOR operation, distance coding scheme is utilized. This distance coding scheme encodes the distance between set bits or "1" bits in other words length of consecutive "0" bits. The organization of distance coding scheme can be viewed in Table 1. This can be described as when there is no "0" bit between two "1" bits in the input bit-stream then the code output length is single bit, length of "0" ranging from 1 to 8 are encoded as 5 bits output code. Runs of "0" with length of 9 to 70 are encoded as 8 bits output code along with two reserved bits. Also an 8 bit output code with unset least significant bit refers to the input bit-stream that contain 71 or more than that runs of "0". End of data is represented by the output having all bits set to "1". Payload compression process is depicted in a flow chart in Figure 3.

Bit Representation								Interpretation
0								No zero bit
1	0	n_2	n_1	n_0				n (1-8) zeroes
1	1	n_5	n_4	n_3	n_2	n_1	n_0	n (9-70) zeroes
1	1	1	1	1	1	1	0	71+n(next code) zeroes
1	1	1	1	1	1	1	1	End of data

Table 1 - Organization of Binary Distance Code [8]

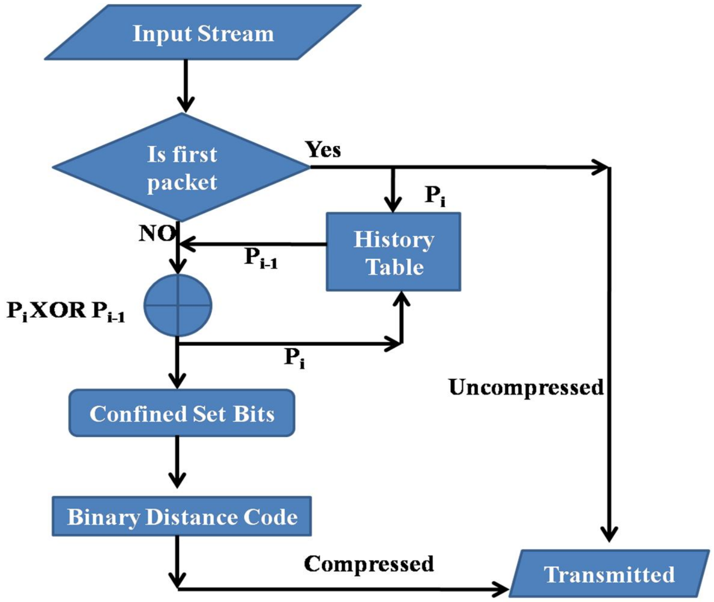

Figure 3: Flow Chart of Payload Compression Scheme

In order to elaborate payload compression scheme let us take an example. For simplicity 1 byte sensed data is taken. On start of transmission first sensed data suppose 10010110 is transmitted as it is without applying compression through binary distance code. It is also placed in single element history table. If subsequent sensed input bit-stream is 10110111. XOR operation is performed on the incoming input bit-stream P_i=10110111 and on history element P_{i-1}=10010110. Therefore $P_i \oplus P_{i-1}$ = 00100001. After XOR operation, history table is updated and P_i takes place of previous P_{i-1}. XOR operation also produces confined number of set bits in the output. These confined number of set bits are compressed using binary distance code.

Decoding is performed through reverse process of encoding. Such as to decode the payload packet binary distance code is implemented in reverse order. First decompress the output bit-stream then XOR operation is performed on this decompressed bit-stream and last payload bit-streamed that is stored previously to get the original payload that is intended to be transmitted towards receiver. If proceed with the above example of payload compression scheme then on the receiver end input bit-stream is decompressed using binary distance code. After decompressing bit-stream 00100001 is achieved. When XOR operation is performed between bit-stream 00100001 and history element 10010110, we get 10110111 that is the original sensed input bit-stream that is intended to be transmitted.

5. Simulation and Results

To evaluate the performance of proposed solution simulation model is implemented in OPNET modeler.

5.1 Energy Consumption

The simulated scenario reveals significant amount of reduction in the energy consumption in case of proposed compression framework as compared to former compression framework [A Reinhardt et al. 2009]. This can be analyzed in the obtained Figure 4, by implementing the former compression framework and proposed compression framework presented in this paper. As shown in figure 4 proposed compression framework consumes energy less rapidly due to infrequent memory access, whereas former compression framework consumes energy rapidly due to frequent memory access and other overheads like large size of history table. Therefore it is presenting the significant amount of difference in the energy consumption. It is evident that proposed compression framework helps for conserving enough energy resources to maximize the resource utilization for WSNs.

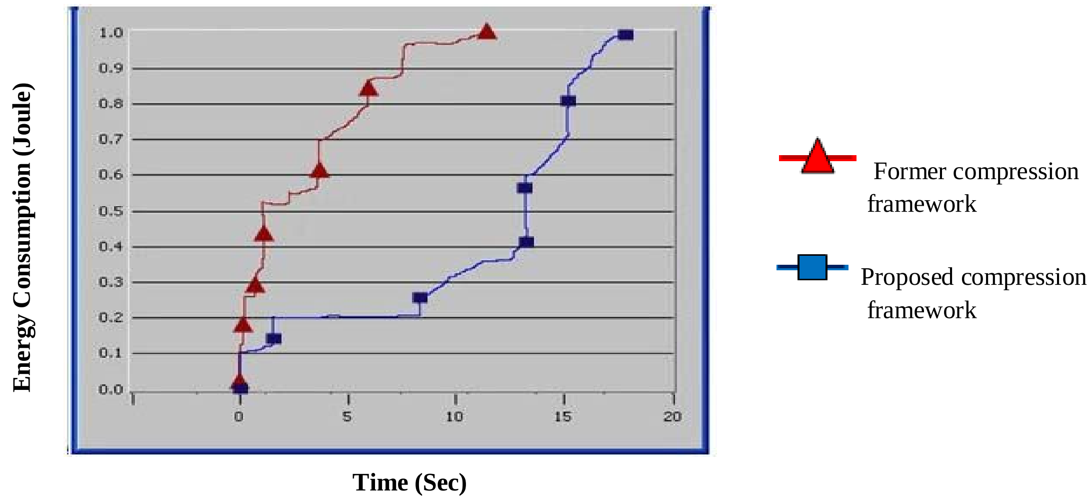

Figure 4: Energy consumption pattern

5.2 Compression ratio

The compression ratio refers to the ratio between the compressed and uncompressed size of data. It is noticeable from the figure 5 that proposed compression framework provides superior compression ratio than former compression framework. This is because proposed compression framework reduces the overheads of frequent memory access and large size history table. The figure shows that approximately 20 bytes of data is compressed into 5 bytes using proposed compression framework, whereas same amount of data can be compressed into 10 bytes using former compression framework. Therefore figure 5 depicts, no matter whatever the size of packet, proposed compression framework performs well at all.

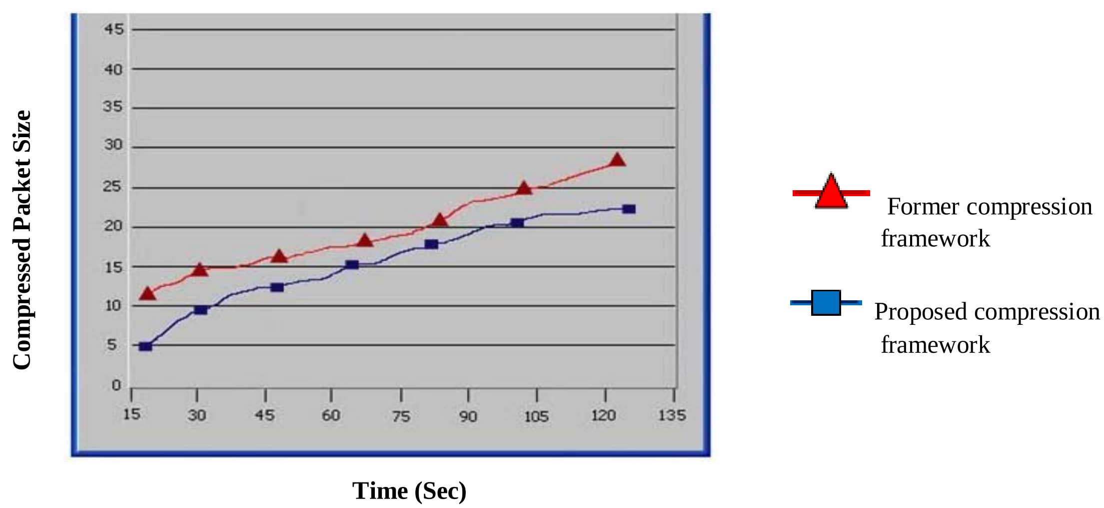

Figure 5: Compression ratio

5.3 Throughput

Throughput refers to average amount of successful data sent or received per unit time over a specific communication channel. Figure 6 presenting comparison of corressponding throughput achieved through former compression framework and proposed compression framework. As shown in figure 6, 55 Kbits of data sent or received in 150 seconds using proposed compression framework whereas 40 Kbits of data sent or received in 150 seconds using former compression framework. Therefore it is self explanatory from the figure 6 that throughput achieved through proposed compression framework is more than throughput achieved through fomer compression framework.

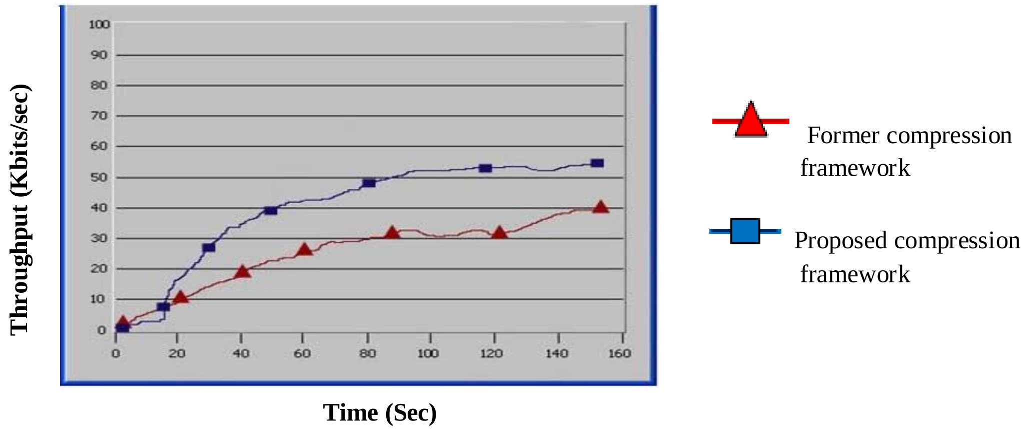

Figure 6: Throughput comparison

6. Conclusion

A compression framework has been presented, that compresses the header and payload using HC1 and stream based encoding respectively. The stream based encoding has been kept simple to avoid unnecessary memory access, because regular memory access leads to significant amount of energy consumption and can reduce the lifetime of the network. Moreover use of IP brings up the sensing services at global scale; heading towards the dream of ubiquitous computing can become true. Use of IP offers many advantages like extensive interoperability that means various kinds of devices on different network can communicate with each other. IP offers security features and network management tools. IP also provides end-to-end reliability in addition to link reliability.

According to simulation results, the implementation and analysis of proposed compression framework leads to significant resource savings that can ultimately expand the lifetime of the network. This is due to the fact that the proposed compression framework achieves infrequent memory access and has single element history table instead of large history table.

REFERENCES

[1] G. Anastasi, M. Conti, M. Di Francesco, A. Passarella, "Energy Conservation in Wireless Sensor Networks: A Survey, Ad Hoc Networks", Vol. 7, N. 3, Elsevier, May 2009.

[2] P. Liu, Y. F. Hu, G. Min and G. Dai, "Semantization Improves the Energy Efficiency of Wireless Sensor Networks", In Wireless Communications and Networking Conference (WCNC), 2010 IEEE, pp. 1-6, 2010.

[3] Kemal Akkaya, Mohamed F. Younis: "A survey on routing protocols for wireless sensor networks". Ad Hoc Networks 3(3): 325-349 (2005).

[4] Francesco Marcelloni and Massimo Vecchio, "A Simple Algorithm for Data Compression in Wireless Sensor Networks", IEEE communications letters, vol. 12, no. 6, June 2008.

[5] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Communications Magazine*, vol. 40, pp. 102–114, 2002.

[6] K. Barr and K. Asanovi, "Energy Aware Lossless Data Compression", In Proceedings of the First International
Conference on Mobile Systems, Applications, and Services (MobiSys), San Francisco, California, May 2003.

[7] C M. Sadler and M. Martonosi, "Data Compression Algorithms for Energy-Constrained Devices in Delay Tolerant
Networks", In Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys), Boulder, Colorado, USA, 2006.

[8] A Reinhardt, M Hollick, "Stream-oriented Lossless Packet Compression in Wireless Sensor Networks", In
Proceedings of 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc
Communications and Networks, Rome, Italy, June 2009.

[9] Montenegro G, Kushalnagar N, Hui J, Culler D. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. Internet Engineering Task Force, Request for Comments 4944, September 2007.

[10] ZigBee Alliance homepage. http://www.zigbee.org, [Jan 2009].