www.ijemr.net ISSN (ONLINE): 2250-0758, ISSN (PRINT): 2394-6962

Vandana Publications
IJEMR

Volume-8, Issue-3, June 2018
International Journal of Engineering and Management Research
Page Number: 184-190
DOI: doi.org/10.31033/ijemr.8.3.25

Structural Health Monitoring by Payload Compression in Wireless Sensors Network: An Algorithmic Analysis

Md. Motaharul Islam[1] Ngnamsie Njimbouom Soualihou[2] and Safi Faizullah[3]
[1]Assistant Professor, Department of Computer Science, Islamic University of Madinah, SAUDI ARABIA
[2]Head of IT, ORYX Consulting, CAMEROON
[3]Professor, Department of Computer Science, Islamic University of Madinah, SAUDI ARABIA

[1]Corresponding Author: motahariut@gmail.com

ABSTRACT

Structural health monitoring is the fact of estimating the state of structural healthor detecting the changes in structure that affect its performance. The traditional approach to monitor the structural health is by using centralized data acquisition hub wired to tens or even hundreds of sensors, and the installation and maintenance of these cabled systems represent significant concerns, prompting the move toward wireless sensor network. As cost effectiveness and energy efficiency is a major concern, our main interest is to reduce the amount of overhead while keeping the structural health monitoring accurate. Since most of the compression algorithm is heavy weight for wireless sensor network with respect to payload compression, here we have analyzed an algorithmic comparison of arithmetic coding algorithm and Huffman coding algorithm. Evaluation shows that arithmetic coding is more efficient than Huffman coding for payload compression.

***Keywords*---**Wireless Sensor Network, TinyOS, Payload Compression, Structural Health Monitoring

I. INTRODUCTION

Wireless sensor network consists of a group of tiny sensor nodes, distributed in a wide geographic area, forming an ad-hoc network, collecting and conveying information regarding the area under surveillance. The data collected by these sensor nodes is aggregated and analyzed at a more capable node called as sink or gateway or base station. Wireless sensor network combines simple wireless communication, minimal computation facilities and some sort of sensing of the physical environment and this leads to a new paradigm of networks that can be deeply embedded in our physical environment, fueled by the low cost and wireless communication facilities.

Structural Health Monitoring is becoming a popular area of research as it offers opportunities in construction management and maintenance [1-4]. As a result, we can create post-disaster scenarios and rescue support. Thus, SHM is a multidisciplinary field where a number of different skills such as seismology, electronic and civil engineering, computer science etc. and institutions can work together to increase the performance and reliability of such systems [9-17].

The traditional approach to structural health monitoring involves conventional piezoelectric accelerometers hardwired to data acquisition boards residing in a PC. The drawbacks of such a system are (1) the high cost of installation, (2) the high cost of equipment and (3) cost of maintenance [1].

The Wireless approach to SHM offers the same functionalities as the wired approach. Besides WSN approach offers various advantages over the wired approach such as a wireless system and installation and maintenance cost reduction. Despite all advantages of the wireless system, it still needs a better compression algorithm at IP payload level. Hence, we have analyzed a presented an algorithmic comparison for payload compression for structural health monitoring in WSN. For thispurpose, we have chosen arithmetic compression algorithm and Huffman coding algorithm [5-6].

The wireless sensor network consisting of low power devices has to communicate with other regular devices and to make this possible 6LoWPAN was introduced. In 6LoPAN only the header is compressed [18-20]. To make the system more efficient in term of power consumption and the amount of data sent, we make a comparison among different algorithms for IP payload compression.

The rest of the paper is organized as follows. Section 2 reviews background and related works. The proposed architecture is presented in section 3. Section 4 describes algorithmic analysis. Section 5 shows the performance evaluation, and finally fiction 6 concludes the paper.

II. BACKGROUND AND RELATED WORK

Reliable, data compression for wireless sensor networks is an ongoing area of research [1]. Recently several reliability protocols have been proposed. WISDEN which is a reliable data transport using a hybrid of end-to-end and hop-by-hop recovery, and low-overhead data time-stamping that does not require global clock synchronization in which data compression is been made using run length encoding This paper also study the applicability of wavelet-based compression techniques to overcome the bandwidth limitations imposed by low-power wireless radios. The architecture of Wisden was simple, a base station centrally collecting data) its design was a bit more challenging than that of other sensor networks built till date. Structural response data is generated at higher data rates than most sensing applications. Furthermore, this application required loss intolerant data transmission, and time synchronization of readings from different sensors. The relatively low radio bandwidths, the high packet loss rates observed in many environments, and the resource constraints of existing sensor platforms added significant challenges to this system [2]. Wisden used a vibration card, especially designed for structural applications. In addition to describing this card, the description of Wisden focused on its three novel software components:

Reliable Data Transport Wisden used existing topology management techniques to construct a routing tree but implemented a hybrid error recovery scheme that recovers packet losses both hop-by-hop and end-to-end.

Compression Wisden used a simple run-length encoding scheme to suppress periods of inactivity in structural response, but it also evaluated the feasibility of wavelet compression techniques to reduce Wisden’s data rate requirements and to improve latency.

Data Synchronization Wisden also implemented a data synchronization scheme that requires little overhead and avoids the need to synchronize clocks network-wide [2]. An accurate data acquisition system, high-frequency sampling with low jitter and time synchronized sampling were not provided in Wisden [1]. In [1] A Wireless Sensor Network (WSN) for Structural Health Monitoring (SHM) was designed, implemented, deployed and tested on the 4200ft long main span and the south tower of the Golden Gate Bridge (GGB). Ambient structural vibrations were reliably measured at a low cost and without interfering with the operation of the bridge. In the GGB deployment, 64 nodes were distributed over the main span and the tower, collecting ambient vibrations synchronously at 1 kHz rate, with less than 10µs jitter, and with an accuracy of 30µG. The sampled data was collected reliably over a 46-hop network, with a bandwidth of 441B/s at the 46th hop. The collected data agrees with theoretical models and previous studies of the bridge. The deployment is the largest WSN for SHM [1]. In this work a small packet size was a bottleneck for network data transmission bandwidth but increasing packet size was not a good solution for the Mica motes due to the limited amount of available RAM; a limitation resulting from an unshared buffer pool. In [3] WSN for structural health monitoring in which Huffman coding technique was used. In this work, sensor node was collecting data, and also processing the data package. They implemented this algorithm into the sensor node to reduce the packet size to be transmitted. In this work, an on-site WSN-based structure health monitoring of Chung-Sha Bridge (Taiwan) was implemented. The implemented WSN monitoring system successfully achieved the frequency analysis of the bridge structure by monitoring with 128Hz sampling rate from end nodes. A local-data processing node with ARM Cortex M3 processor was developed. Based on this local-data-processing node, Huffman compression algorithm was implemented and examined. The experimental results showed that the wireless transmission payload was reduced by 60% and node number of the implemented network could be increased by 3 times [3].

Both these two works are concerned with the performance. Wireless sensor networks in Structural Health Monitoring based on ZigBee technology and TPSN (Timing-sync Protocol for Sensor Network), ZigBee was used because it is the most popular low-cost, low-power wireless mesh networking standard available. It is Suitable for complex networks with large spatial extension, multi-hop networks and proprietary metering and automation solutions. But one of it drawback is that the interoperation between Zigbee devices and IP based devices [4]. So, our approach is to use WSN in which 6LOWPAN is used as an adaptation layer along with Payload compression using arithmetic compression. "6LoWPAN (Montenegro et al., 2007)" refers to "IPv6 over Low-Power Wireless Personal Area Network". It defines an adaptation layer which allows transportation of IPv6 packets over IEEE 802.15.4 links. 6LoWPAN reduces the IPv6 packets to fit within the MTU (127 bytes) of IEEE 802.15.4 frames. To do that 6LoWPAN uses header compression and fragmentation and reassembly schemes [18-20].

III. PROPOSED ARCHITECTURE

Sensor Network is made up of many sensor nodes. These nodes are tiny in size. They have limited power supply, memory, and processing capability. Each sensor consists of, Transceiver, Power supply, Processor, Memory and Sensors. The overall architecture of our sensor network is shown in the figure 1.

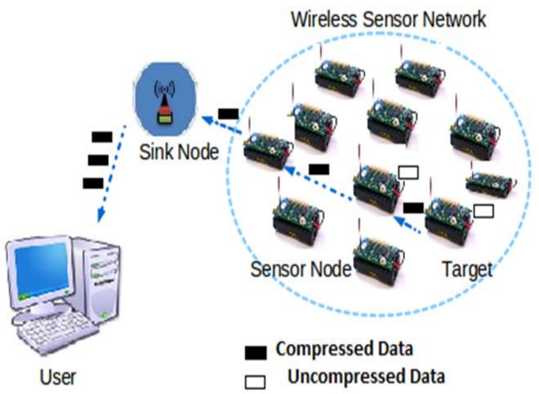


Figure 1: Architecture of a Wireless Sensor Network

The sensor nodes are connected to one another in mesh topology fashion. Each node compresses the data using the arithmetic compression algorithm, and forward it to the sink node. The compression of data in each node i.e. local processing of data compresses the data further and thus saves power as well as increases the capacity of the nodes in the network. The target node, passes the compressed data to the sink node, and a further compression on that received data is being perform again, and then, sent to the user's terminal. So, the sink node has more computation power because of the reduce amount of transfer it has to make.

During the implementation the sensors node are given id in order for us to uniquely identify them so that we can know which node is down or up. This id allocation allows us to specify the sink node, which is the node to which all the other nodes has to send the compressed data to. This node is usually given the id 500 because, the number 500 represent the root node, and any number can be assign to any other node as their id. Doing so will then help the other node to locate the sink node and pass the data to it for further processing.

IV. ALGORITHMIC ANALYSES

Arithmetic algorithm

In this algorithm, the symbols are not replaced by some code words. Instead, the symbol in the data are being assigned, values, based on some mathematical model; the Arithmetic algorithm can treat the whole symbols in a list, or, in a data messagxe to be transmitted as one unit. It does not use a discrete number of bits or some frequency distribution for the data's symbols [6], instead each symbol is assign an interval starting with the interval [0….1); So, at the beginning the probabilities of occurrence of a set of symbols together with the cumulative probabilities are taken into consideration; these cumulative probabilities are used for encoding and decoding. Figure 2 gives some pictorial representation in this regard.

The Encoding Process: The first step is to calculate the cumulative probabilities and then make ranges based on the obtained results. When we read a character, its range is considered as the new cumulative range on which our encoding will be done, that range is so divided into the sub parts, according to the probabilities of occurrence of the character being encoded; and the next symbol is read, and this process of sub part formation is repeated again and this goes on as long as we have a character to encode in our source data. Once the end of our source data is found, we take a fraction of our sub part range formed. Therefore, using a fraction taken from our range we can represent our entire source data into binary form. Let's consider the following example [7]. From the encoded interval [0.6607, 0.66303). A sequence of bits is assigned to a number that is located in this range.

During this encoding process two values are frequently calculated: The Upper bound and the Lower bound, and from those values are generated the probability distribution of the symbol to be encoded, they are obtained from the bellow formula:

- Lower Bound:

$$L = \sum_{i=1}^{n} n^{n-i} C_i \prod_{k=1}^{i-1} f_k$$

- Upper Bound:

$$U = L + \prod_{k=1}^{n} f_k$$

We can summarize the encoding algorithm using the following pseudo-code:[22]

Initialize l and u.

Get symbol.

$$l \leftarrow l + \left\lfloor \frac{(u - l + 1) * Cum\_count(x-1)}{TotalCount} \right\rfloor$$

$$u \leftarrow l + \left\lfloor \frac{(u - l + 1) * Cum\_count(x)}{TotalCount} \right\rfloor - 1$$

While (MSB of u and l are both equal to b or E_3 condition holds)

if (MSB of u and l are both equal to b)

{

Send b

Shift l to the left by 1 bit and shift 0 into LSB

Shift u to the left by 1 bit and shift 1into LSB

While (Scale3>0)

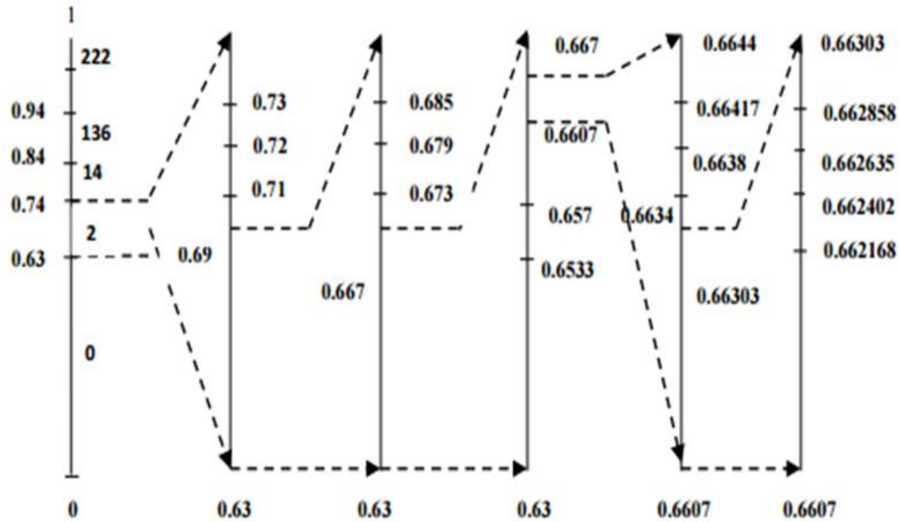
{

Send complement of b
Decrement Scale3
}
}

if (E_3 condition holds)
{
Shift l to the left by 1bit and shift 0 into LSB
Shift u to the left by 1bit and shift1 into LSB
Complement (new) MSB of l and u
Increment Scale3
}

Image representing the full encoding process:

Input symbols : 2 0 0 136 0



Output : [0.6607 , 0.66303)

Figure 2: Encoding process

The Decoding Process: Once the compress data reach the other end, it needs to be decompressed and this is being done by following some mathematical model, and then, applying the inverse process of the encoding. But to do that, the receiver needs to know the number of symbols that were sent as well as the probability l frequency distribution.

We can summarize the decoding process with the following algorithm: [22]
Initialize l and u.
Read the first m bits of the received bit stream into tag t.
k = 0

$$while\left(\left\lfloor\frac{(t-l+1)*Total\ Count-1}{u-l+1}\right\rfloor \geq Cum\_Count(k)\right)$$

k<— k+1

decode symbol x

$$l \leftarrow\ l+\left\lfloor\frac{(u-l+1)*Cum\_count(x-1)}{TotalCount}\right\rfloor$$

$$u \leftarrow l + \left\lfloor \frac{(u - l + 1) * Cum\_count(x)}{TotalCount} \right\rfloor - 1$$

While (MSB of u and l are both equal to bor E_3 condition holds)

If(MSB of u and l are both equal to b)

shift l to the left by 1bit and shift 0 into LSB

shift u to the left by1bit and shift 1 into LSB

hift u to the left by1bit and shift 1into LSB

Shift t to the left by 1bit and read next bit from received bitstream into LSB

Complement (new) MSB of l, u, and t.

}

shift t to the left by 1 bit and read next bit from received bitstream into LSB

}

If (E_3 condition holds)

{

Shift l to the left by1 bit and shift 0 into LSB

}

If (E_3 condition holds)

{

Shift l to the left by1 bit and shift 0 into LSB

Shift u to the left by1bit and shift 1into LSB

Shift t to the left by 1bit and read next bit from received bitstream into LSB

Complement (new) MSB of l, u, and t.

}

Huffman algorithm

The Huffman encoding algorithm work by, replacing a symbol by a set of bits representing that symbol. Those set of bit are generated as follow: at the beginning there is a frequency operation that is performed to group identical symbols together and then order them from the symbol with the smaller frequency to the one with the biggest frequency, then, there is a join operation which is performed between the two smallest symbol's frequencies in order to generate a parent node, then the same process is been performed again between the next two smallest symbol's frequencies including the parent generated, and this process goes on until we have a root node and the tree fully generated. Once the tree is generated, there is an assignment that is performed as follow: the left edges of the tree are assigned the bit 0 and the right edges will be assigned the bit 1. And the end a table is generated containing symbols and their corresponding code bits, this table has to be known to both end device that are communicating. For the encoding process, to encode a symbol, you go down the tree and the read the bit on the path from the root node to the desired symbol. And for the decoding process you use the table generated and find the symbol corresponding to the code that you received.

V. PERFORMANCE EVALUATION

In performance evaluation, we have used the Huffman Coding as a reference as this algorithm was used in SHM [3].The performance is measured in terms of Compression ratio, data size, memory consumption and compressed data size.

In the figure 3, we compared the compression ratio (CR) of the Huffman coding with that of the arithmetic coding. It is being observed that, with the increasing in the size of the data, the compression ratio of arithmetic coding gets better than that of Huffman coding, and the compression ratio of arithmetic is almost more than the double of that of the Huffman coding as the data get higher. On the other hand, Huffman Coding does not give any better compression with the increase of size. We would like to mention that the data size in SHM is large and varied. Which is suitable for our algorithm.

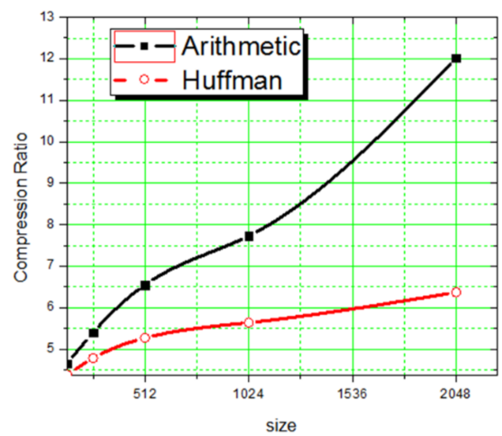


Figure 3: Algorithm Comparison (CR vs. Size)

The following Table 1 show the data's result used to draw the above graph.

Table 1: Comparison between Huffman and Arithmetic

File Size	CompressionRatio	
	Arithmetic	Huffman
128	4.65	4.38
256	5.4	4.78
512	6.55	5.27
1024	7.73	5.64
2048	12.02	6.37

The above graph (Figure 4), shows, the amount of data capable to be compressed given the same memory boundary. It is seen from the graph that, the Arithmetic coding, can compress the whole data (100%) provided to it at a specific time, while the Huffman coding, can only compress 60% of the data provided to it, as well the RLE can only compressed 75 % of the data provided, then the node running these algorithms will run out of memory at some time. The compression ability of arithmetic algorithm when provided with the same memory block is better than that of the Huffman and RLE.

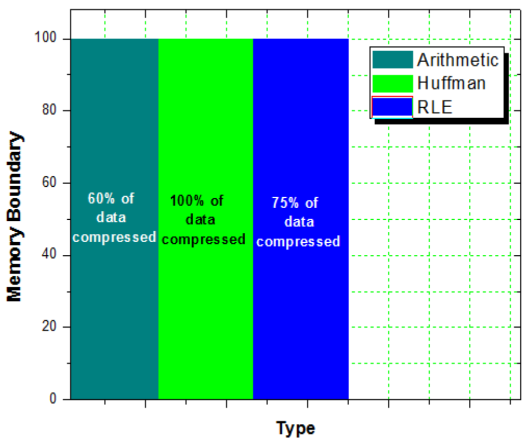


Figure 4: Data comparison capable to be compressed by both algorithm

VI. CONCLUSION

Wireless sensor network is getting popular day by day. Its mobility and capability of real time communication has led many applications use this technology. Many developers and researchers are providing their support to enhance this sector. Many new protocols, architectures are being proposed in this purpose.

Among many application areas, Structural Health Monitoring is very important. SHM has many advantages and it can be used to save both money and lives of the people of the country. By monitoring the important structures and by renovating or maintaining these structures when necessary can save a lot of money as well as keep the structures fit. Besides it can save many lives which are lost in different natural disasters.

Using WSN technology in SHM can save lives and make the system less costly and affordable. The structure's health can be easily monitored, as well as structures in the remote areas can be monitored and can be easily maintained at a very low cost which will save lives in case of emergencies.

One of the main disadvantages of the WSN node are, the power constraint (because small size implies small battery) and limited memory. It is useful to mention that WSN node can be deployed in extreme environmental conditions and no human can go over to change a damaged node. So, we need to make use of the node as efficiently and as long as possible. So, the data should be transferred efficiently between sensors.

The compression algorithm and technique we proposed will increase the life time of the sensors as less data will be sent. Besides, the local processing will increase the life time as well as the number of sensors deployed in the network. We have chosen the arithmetic algorithm over other type of algorithm because of the advantages that it was offering on the compression ratio plan as well as the memory required for the processing of data. By observing the results generated by all the algorithms in our work, we can easily show that arithmetic algorithm is far more better that the other algorithm.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their very useful comments that helped us enrich the quality and presentation of the paper a lot. This work was supported by a grant from the Islamic University of Madinah in 2017-2018.

REFERENCES

[1] Kim, Sukun, et al. (2007). Health monitoring of civil infrastructures using wireless sensor networks. *IPSN '07 Proceedings of the 6th international conference on Information processing in sensor networks, New York, USA*.254-263.Available at: https://dl.acm.org/citation.cfm?doid=1236360.1236395

[2] Xu, Ning, et al. (2004). A wireless sensor network for structural monitoring.*Proceedings of the 2nd international conference on Embedded networked sensor systems*,ACM Press.13-24. Available at: http://graphics.stanford.edu/courses/cs321-05-fall/Readings/structural.pdf

[3] Hsu, Chia-Hao, et al. (2014). An implementation of light-weight compression algorithm for wireless sensor network technology in structure health monitoring.*IEEE World Forum on Internet of Things (WF-IoT), Seoul, South Korea*. Available at: https://ieeexplore.ieee.org/document/6803227/

[4] Jiang, Xiang-dong, Yu-liang Tang, & Ying Lei. (2009). Wireless sensor networks in structural health monitoring based on zigbee technology.*3rd International Conference on Anti-counterfeiting, Securityand Identification in Communication, Hong Kong, China*. Available at: https://ieeexplore.ieee.org/document/5276977/

[5] Shahbahrami, Asadollah, et al. (2011). Evaluation of huffman and arithmetic algorithms for multimedia compression standards. *arXiv preprint arXiv:1109.0216*, 1-11. Available at: https://arxiv.org/ftp/arxiv/papers/1109/1109.0216.pdf

[6] Langdon Jr& Glen G. (1984). An introduction to arithmetic coding. *IBM Journal of Research and Development, 28*(2), 135-149.

[7] Bormann, Carsten. (2014). 6LoWPAN-GHC: Generic header compression for IPv6 over low-power wireless personal area networks (6LoWPANs). *Internet Engineering Task Force (IETF)*, 1-24. Available at: http://www.rfc-base.org/txt/rfc-7400.txt

[8] Kodituwakku, S. R., & U. S. Amarasinghe. (2010). Comparison of lossless data compression algorithms for text data.*Indian journal of computer science and engineering*, *1*(4), 416-425.

[9] Elson, J., and Estrin, D. (2001).Time synchronization for wireless sensor networks. *Proceedings of the 15th International Parallel & Distributed Processing Symposium, IEEE Computer Society, San Francisco, USA*. Available at: https://ieeexplore.ieee.org/document/925191/

[10] Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., & Pister, K. (2000). System architecture directions for networked sensors. *SIGPLAN Not*, *35*(11), 93–104.

[11] Intanagonwiwat, C., Govindan, R., and Estrin, D. (2000).Directed diffusion: A scalable and robust communication paradigm for sensor networks. *In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking*, ACM Press.1-12. Available at: https://www.isi.edu/division7/publication_files/directed_diffusion_scalable.pdf

[12] Lynch, J. P., Sundararajan, A., Law, K. H., Kiremidjian, A. S., & Carryer, E. (2003). Power-efficient data management for a wireless structural monitoring system.*In Proceedings of the 4th International Workshop on Structural Health Monitoring (Stanford, CA)*, USA. 1-8. Available at: http://eil.stanford.edu/publications/jerry_lynch/StanfordSHMWorkshop03Paper.pdf

[13] Stann, F., & Heidemann, J. (2003). RMST: Reliable data transport in sensor networks.*Proceedings of the First International Workshop on Sensor Net Protocols and Applications, IEEE*, Anchorage, Alaska, USA. 1-11. Available at: https://www.isi.edu/~johnh/PAPERS/Stann03a.pdf

[14] Wan, C.-Y., Campbell, A. T., and Krishnamurthy, L. (2002) Psfq: A reliable transport protocol for wireless sensor networks.*Proceeding of First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*,Atlanta, Georgia, USA. 1–11. Available at: https://www.cs.ubc.ca/~krasic/cpsc538a/papers/PSFQ-wsna2002-workshop.pdf

[15] Tang Yu-liang, Luo Yu, Huang Lian-fen, Guo Jian,& Lei Ying. (2012). Wireless sensor network for on-line structural health monitoring. *7th International Conference on Computer Science & Education (ICCSE)*, Melbourne, VIC, Australia. 386-389. Available at: https://ieeexplore.ieee.org/abstract/document/6295098/

[16] Han Zhi-gang& Cui Cai-hui. (2009). The application of zigbee based wireless sensor network and GIS in the Air pollution monitoring.*International Conference on Environmental Science and Information Application Technology*,Wuhan, China. 546-549. Available at: https://ieeexplore.ieee.org/document/5199951/

[17] Domenico Balsamo, Giacomo Paci, Luca Benini, Brunelli Davide. (2013). Long term low cost passive environmental monitoring of heritage buildings for energy efficiency retrofitting.*IEEE Workshop on Environmental Energy and Structural Monitoring Systems*, Trento, Italy. 1-6. Available at: https://ieeexplore.ieee.org/document/6661695/

[18] Md. Motaharul Islam, M. Abdullah-Al-Wadud, & Eui-Nam Huh. (2015). Energy Efficient Multilayer Routing Protocol for SPMIPv6 based IP-WSN. *International Journal of Sensor Network*, *18*(3-4), 114-129.

[19] Md. Motaharul Islam& Eui-Nam Huh. (2011). A novel addressing scheme for pmipv6 based global ip-wsns.*Sensors*, *11*(9), 8430-8455.

[20] Md. Motaharul Islam& Eui-Nam Huh. (2011). Sensor proxy mobile ipv6 (spmipv6)-a novel scheme for mobility supported ip-wsns.*Sensors*, *11*(2), 1865-1887.

