ISSN 2229-5518

Survey on Algorithms for Data Compression in Wireless Computing Systems

Upasana Mahajan, Dr. Prashanth C.S.R
Asst. Prof. CSE, NHCE, Karnataka, Bangalore
Professor& HOD, CSE, NHCE, Karnataka, Bangalore

Abstract— Data compression is a technique in which the same amount of data is transmitted by using a smaller number of bits with the help of an algorithm. Different compression algorithms are compared and based on their performance the optimal algorithm is selected. In this paper the different data compression algorithms have been selected and their performance evaluation is done. One of the best and easy algorithms is Huffman's Encoding but the result of evaluation of performance shows that the Huffman's algorithm has some drawbacks, and can be overcome by new algorithm named as Selective Huffman's algorithm. The paper focus on a new technique for compression named as semi-lossless text compression, which states that instead of compressing the whole data if only some portion of the data is compressed the time and space can be saved. The paper also attempted to explore each layer and shows what are different compression techniques can be used in TCP/IP model.

Keywords: Data compression, Huffman algorithm, Selective Huffman Encoding, Semi-lossless text compression

1. INTRODUCTION

Compression can be identified as either lossless or lossy compression. The lossy compression is a method of data encoding, in which compression is done by discarding/losing some data. The lossless data compression can be defined as reducing the bits by identifying and eliminating statistical redundancy. The paper gives the importance to the lossless compression as it is related to data whereas the lossy compression most of the time occurs in audio and video data or in image files and choosing the right depends on the performance of the algorithm.

2. LITERATURE SURVEY

2.1. Optimal Selective Huffman Coding for Test- Data Compression [1] [2]

The Huffman coding provides the short length codewords for the decodable variable length codes, but a high hardware overhead of the required decompressors. To overcome this problem the Selective Huffman approach is used. The study shows that the Selective Huffman approach is not too optimal when compared to proposed selective Huffman approach. The proposed approach maximizes the compression ratio with similar or smaller decompressor hardware overhead. The Selective Huffman Encoding [2] technique symbolizes the codes (fixed to variable length) as statistical codes i.e. for the data blocks of fixed length the variable length codewords are used. The following example shows how the Huffman Codes are generated with the help of binary tree.

Table1.Test Data Partitioning and Occurrence Frequencies

Test Set T	Distinct Blocks	Occur. Freq.
1010 0000 1010 1111	1010	9/20
1111 0000 1010 0001	0000	5/20
1010 0000 0010 1010	1111	3/20
0000 1010 1010 0000	0001	2/20
1010 1111 1010 0001	0010	1/20

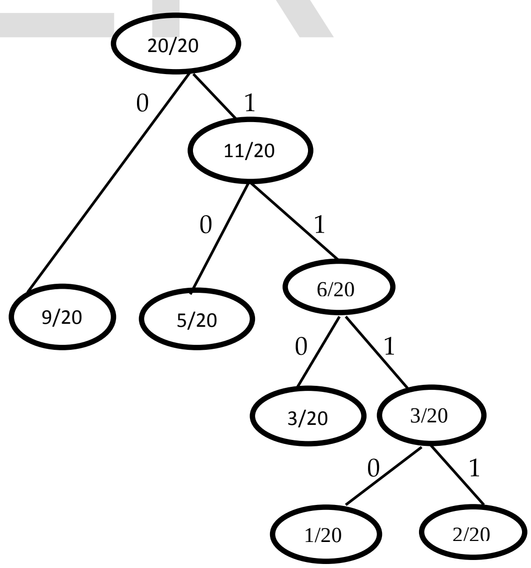
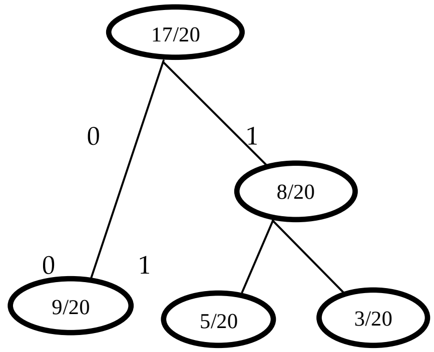


Fig.1 Huffman Tree for table 1

To overcome the problem of large overhead of the required decompressors Selective Huffman Encoding is used, in which all distinct blocks are not encoded. It means that the distinct block which occurs more frequently are encoded and which occurs less frequently, remains unencoded. This way new code, Selective Huffman is generated as shown in

ISSN 2229-5518

the example. Using the above test data given in table 1, the Selective Huffman tree is constructed. So in this only the distinct blocks that occur most frequently are encoded i.e. the first three blocks are encoded and other two are not (referring table 1).



distinct block	1010	0000	1111
Codeword	10	110	111

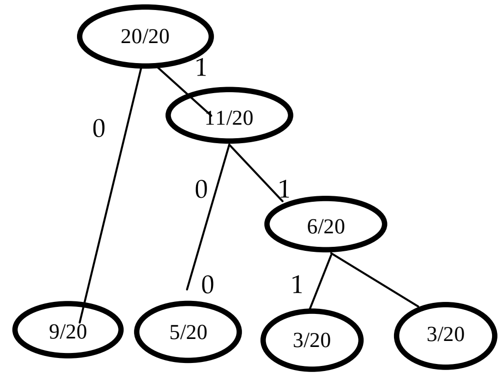
Compressed Test Set (table1 column1 as an example)

0	10	0	110
110	10	0	111 0001
0	10	111 0010	0
10	0	0	10
0	110	0	111 0001

Fig.2 Huffman Tree &Selective Huffman encoding

The main issue in Selective Huffman Encoding approach is that the extra bit which is used for all coded and uncoded data in the compressed test set. Though the extra bit is used to differentiate the data but it also increases the length of data irrespective of their occurrence frequency. So to overcome this problem the new method has been introduced known as proposed Selective Huffman Encoding. In proposed method *m+1* Huffman codewords are used, *m* for encoding *m* most frequently occurring distinct blocks and 1 for indicating the unencoded blocks.

Here is an example of Proposed Selective Huffman Encoding, using the data given in Table1 the following will be the result. Consider the test set given in table 1 there are two unencoded blocks as shown in fig. 0001 and 0010. While apply the proposed approach, just add the occurrence frequency of these two unencoded blocks i.e. 2/20 + 1/20 = 3/20 and draw a tree as shown in fig.1.3.Once the tree has been constructed, it's time to write the compressed test set as shown below:



Distinct Blocks	1010	0000	1111	Unencoded blocks 0001, 0010
Codewords	0	10	110	111

Compressed Test Set

10	110	10	111
111	110	10	00001
10	110	00010	10
110	10	10	110
10	111	10	00001

Fig.3 Proposed Selective Huffman encoding

So with this proposed approach only extra bits will be used for the unencoded distinct blocks instead of using single bit for every codewords. Both the approaches performs same functions i.e. encodes distinct blocks but the proposed approach is more efficient and optimal than the Selective Huffman Encoding proved by using theorems[1][2]. The efficiency of the Selective Huffman coding has been verified through the experimental results and comparisons in [2]. The evaluation of two approaches can be done as

Compression ratio(%) =

(1- compression bits / Mintest bits) * 100

where mintest is uncompressed test set.

This shows that for the constant block size, the proposed method always benefits from greater encoded-distinct block volumes which is not true for the approach of [2] as the use of the extra-bit overhead forced on all codewords. Using the compression improvement formula it shows that the proposed encoding provides better compression as compared to the approach [2]. So,

Compression improvement (%) =

$$1-\frac{\text{Compressed bits of proposed}}{\text{Compressed bits of [2]}}*100$$

The main reason why the proposed approach is more effective than the approach [2] is the decrease in hardware

ISSN 2229-5518

overhead which is achieved by eliminating the use of extra bits for coding the codewords.

2.2. Semi-Lossless Text Compression [3]

While performing data compression the main concern is compression performance which depends on two factors time and space. The idea behind the lossy compression is that even if the whole data is not available, the human brain can often make up for the missing parts and guess, whatever has been omitted. For example, one can understand English text even if there are errors, as shown in following paragraph,

Aoccdrnig to arscheearch at CmabrigdeUinervtisy, it deosn't tmttaer in wahtoredr the ltteers in a wrod are in; the olnyiprmoetnttihng is that frist and lsatltteer be at the rghitpclae. The rset can be a toatlmses and you can still raed it wouthitporbelm. Tihs is bcusease the huamn mind deos not raederveylteter by istlef, but the wrod as a wlohe.

Semi-Lossless compression method has been introduced. It is the hybrid of lossless and lossy compressions and the original text will not be fully reconstructed. This method has restriction, i.e. it can be used only in short emails or SMS sent between cellular phones, where the correct spelling is not so important. Considering a method in which there is no loss of information and taking the rule mentioned in the quotation above, namely, that "if the first and last letters of the printed words are left in place, the remaining letters within each word can appear in any order". This law is not universal but relies on the assumption that the reader has a good knowledge of English, the following compression and decompression have been suggested.

Compression:

1. Process the words sequentially and if the current word is not special (number, proper name, etc.), do
2. keep the first and last letters in place, but rearrange the others into 'special' order,
3. apply some encoder on the rearrange text.

Decompression:

1. Decode the compressed words sequentially, and if the current word is not special, do
2. keep the first and last letters in place; choose a random permutation of the other letters and send them to output.

There are different methods for choosing a special order of the characters some of them has been discussed. One possibility is arranging the letters in alphabetic order as this strategy expected to improve compression is Burrows-Wheeler Transform (BWT) [4]. If the characters in the given problem are arrange alphabetically, by using move-to-front or run-length coding if strings are long enough the compression can be improved.

Another possibility can be arranging the characters by frequency which will not give good bonding between certain characters into considerations. But if grouping of the characters is done based on the probabilities of a given letter to appear after another one, will give more accurate rule. For example, E is most likely followed by R, so E is its most probable successor. The algorithm based on this assumption will be:

1. Start with an arbitrary character, x;
2. While not all characters are processed
   - Choose, among remaining characters, the successor s of x with the highest probability;
   - $x \longleftarrow s$

Using this scheme leads to a chances of loss, that choosing the successor with the highest probability might push the second best choice too far away.

Once the ordering of characters has been decided, it's time to check which compression technique will suites most. If Huffman or arithmetic coding, L-Z variant encoding is used for above mentioned law, there will be no additional compression because the set of encoded characters remain the same only the order of characters is altered. A better performance can be assumed when Huffman coding combined with the Markov model of order $k \geq 1$. A different approach is used for this type of text known as bigrams, trigrams, or generally *k*-grams with $k > 1$. Following are the steps involved in bigram coding.

Step1: Partitioned the text into sequence of character pairs (blanks are replaced by dashes for visualization).

Step2: The pairs are reordered (in the below example the alphabetic order is taken in account with blanks preceding the characters).

Step3: A flag bit is added for the first and last pair of each word, in order to take care of word boundaries. A '1' indicates that the corresponding pair has to be switched and '0' means that corresponding pair stays unchanged

Step4: Now, apply Huffman coding algorithm for each pair.

ISSN 2229-5518

Let's take an example, the string is given as
combining character ordering

Step1

co	mb	In	in	g-	ch	ar	ac	Te	r-

or	de	Ri	Ng

Step2

co	bm	In	In	-g	ch	ar	ac

et	-r	Or	de	ir	gn

Step3

0 1 0 1

co	bm	in	in	g-	ch	ar	ac	et	r-

0 1

or	de	ir	Ng

Using the BWT doing block sorting is also good option. Some experiments has been done and their results shows that the Lempel-Ziv and alone Huffman coding compression is not optimal. Using *k*-grams with Huffman coding gives the better compression performance. The main apprehension of semi-lossless text compression is that it is used for short messages (cellular phones) or the emails where the loss of small amount of data doesn't bother the whole information. And if this method is used for whichever language, the concern user must know that language then only it is useful.

2.3. A Survey on Data Compression in Wireless Sensor Networks [5]

The basic wireless sensor network configuration is that a large number of sensor nodes are compactly deployed over a sensor field where all the nodes are connected without wire by radio frequency, infrared or other medium. The data collected by nodes are transmitted to others by wireless medium in a network. A network can be constructed inexpensively and addition or removal of the node is easy. To attain the requirement i.e. the size of the each sensor node is to be small, the size of every component of sensor node like power source, processing and data storing memory also has to be small. Due to these reasons, the WSNs are resource constraint such as they have less power supply, bandwidth for communication, processing speed and memory space. Research study shows that data compression helps to achieve the maximum utilization of the limited sensor resource.

In terms of power consumption, operation of wireless sensor node can be divided into three parts: sensing, processing and transmission. And among of these three operations, the data transmission is most power consuming. So by reducing the size of data, the transmission power will be reduced. The research [6] has been conducted for the power consumption by data compression and data transmission in wireless medium, whose outcome indicates that if more than one bit is removed from an original data bit string by using compression which takes instruction equivalent to 480 addition instructions than it will reduce total power consumption by a sensor node. The operative is apply the data compression before transmitting data in wireless medium as there are some compression algorithms which increases total power consumption instead of reducing and it results in expensive memory access. Following sections discuss the different types of compression algorithm for wireless medium.

In Coding by Ordering [5] data compression technique firstly data is passed from sensor nodes in the interested region to a collector node, as shown below in figure. Node A, B and D are data aggregation nodes (it aggregates the sensor data using aggregation approaches and then aggregated data is transferred to the sink node by selecting the efficient path). In below figure, at node D data collected by node E is combined with data collected by node D itself. Then the aggregated data is transmitted to node B.

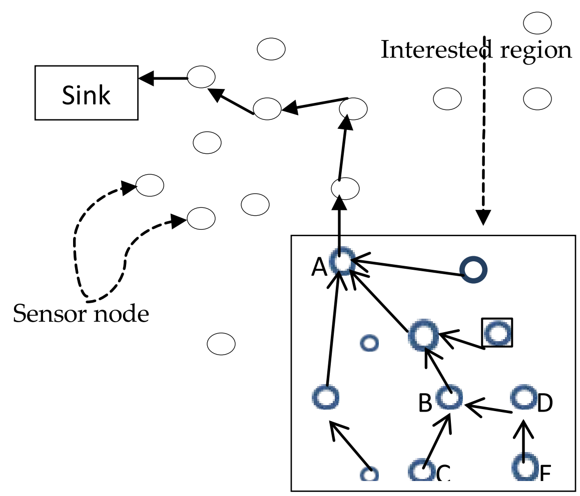


Fig.4. Data Path

In this algorithm, when data is combined at an aggregation node, some data is dropped. To include the information of dropped data in the aggregated data, the order of data packet is utilized. This technique is simple and gives the good compression ratio so can be used in WSNs only if its utilization will become efficient by using the mapping table for the mapping permutation to data value.

The pipelined In-Network Compression technique [7] basically deals with trading high data transmission latency for low transmission energy consumption. In this only for some period of time the aggregation node's buffer stores the data which collected by sensor node. During this period of time the process of combining the different data packets into one packet and removal of the redundancies will take place to minimize the data transmission. Let's take an example to explain the above process; every data packet has the following form:

[measured value, node ID, timestamp]

And after applying the above compression data packet will have the following form:

[shared prefix, suffix list, node ID list, timestamp list]

where, 'shared prefix' is the most significant bits, with all measured values in combined data packets have in common.

'suffix list' is the list of measured values excluding the shared prefix part.

'node ID list' is the list of node identifiers.

'timestamp list' is the list of timestamp.

This approach is simple and the shared prefix method can be used for node IDs and timestamps, by which more data compression can be done. The efficiency of the data compression depends on the length of shared prefix i.e. longer the shared prefix, larger the data compression ratio. There is serious issue with this technique is the efficiency. Though the long shared prefix is used, the efficiency of Pipelined In-Network Compression will decrease. Since the sensor node in this method has limited amount of memory space, enough buffer space will not be available. Combining large amount of data packets, the larger buffer space is required to store the packets temporarily which is not possible.

The Low-Complexity Video Compression algorithm [8] is specifically aimed for the wireless video surveillance system. Each video frame is divided into small blocks each contain 8 x 8 (64) pixels. To reduce the computational complexity, only the subset of the blocks (all while in this case) in each frame is considered. The number assigned to pixels indicates their importance ('1' most important and '3' means least important).

The algorithm works: For one block pixel assigned '1' is compared with the referenced frame's pixels. If the difference (D_i) of two pixels is greater than the threshold (M) value, then counter (P) is incremented by one, which initially set as zero. The maximum difference value is stored in U.

Let say S is the sensitivity parameter which is set by user. Now if P>S, then the block is marked as active block and scanning procedure moves to a next block. But if P≤S, then after checking all '1' pixels average difference 'A' is calculated as

$$A = \sum_{i=1}^{n} D_i / n$$

If A<N & U<M, where N is another threshold value, and N<M then next block is processed. Otherwise the algorithm start checking the pixels assigned as '2' in the same block. For pixels assigned '2'and'3' in the same block same process is followed as for pixel assigned '1' but the difference is that the block is marked and procedure moves to next block for '2' assigned pixels only if A>N and U>M. And in case of pixels assigned '3', after checking all pixels, if (A>N and U>M) or (A<N, U>M, and P>S), then only the block is marked as active.

After applying above procedure for set of white blocks in a frame, the non-scanned are checked. If a non-scanned block is adjacent to at least two active blocks, the non-scanned block is marked as active. Once the all marked blocks are encoded by JPEG, they transmitted in WSNs. The experimental analysis shows that the outcome image using this algorithm is same as the image processed by MPEG-2 but the difference is that the above algorithm saves little amount of energy more than MPEG-2. The Distributed Compression Technique [9] [10] uses the side information to encode the source information. Let's assume there are two sources (X and Y) which are correlated and discrete-alphabet independent identically distributed as shown in below figure. The X can be compressed with the help of conditional entropy [H(X|Y)] which can be expressed as follows:

$$H(X|Y) = -\sum_{y} P_Y(y) \sum_{x} P_X(x|y) \log_2 P_X(x|y)$$

This method starts composing cosets, whose codevector are source of X. An index value is assigned to each coset, so while transmitting data to a decoder the source X only sends an index value of coset, to which the codevector belongs. Then the source Y sends the codevector as side information. The decoder looks up the coset, which has the same index received from X. then , the decoder selects one codevector, which has a closet value to the codevector sent by Y, in the coset.

ISSN 2229-5518

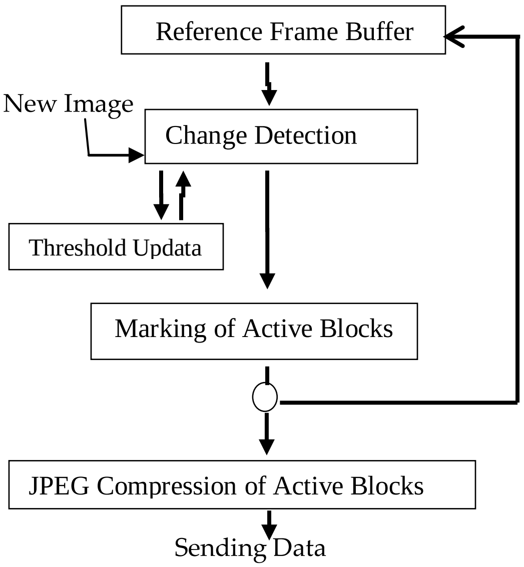


Fig.5 Image Data Processing Flow

2.4. A Survey on Sensor Networks [18]

The sensor nodes have a distinctive feature that instead of sending the raw data to the node responsible for the fusion, they use their processing abilities to locally carry out the simple computation and transmit only the required and partially processed data. The functionality of sensor nodes is that it collects the data and then route data back to the sink. As shown in below figure that the data routed back to the sink by a multihop infrastructureless architecture through the sink. The sink may communicate to the task manager node via internet or satellite. There so many factors which influence the design of the sensor networks like fault tolerance, scalability, production costs, operating environment, sensor network topology, hardware constraints etc. In the following section brief discussion about how these factors affect the design.

Fault Tolerance is the ability to sustain sensor network functionalities without any interruption due to the sensor node failure and also known as reliability. The reliability or fault tolerance of a sensor node can be calculated using the Poisson distribution to capture the probability of not having a failure within the time interval (0, t).

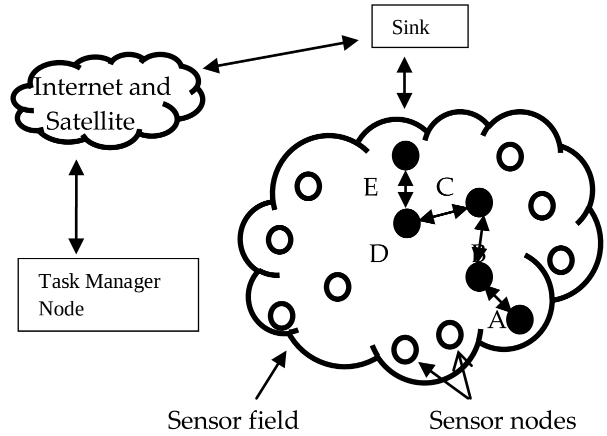


Fig.6 Sensor nodes scattered in a sensor field

Scalability indicates that how many sensor nodes can be used in a network i.e. the density of the sensor nodes, which ranges from few sensor nodes to some hundred sensor nodes in a region.

Production Costs indicates that if large numbers of sensor nodes are used in a region, then the cost of a single sensor node makes a huge difference. If the overall cost of the network is more than the deploying traditional sensors than the sensor network is not cost justified. So to make the cost of single sensor node low, the bluetooth radio system is used.

Hardware Constraints indicates what are the different hardware are present in a sensor node. A sensor node is made up of four components, a sensing unit, a processing unit, a transceiver unit and a power unit as shown in below figure. Sensor node may also have the additional-application dependent components like location finding system, power generator and mobilizer. The processing unit contains one small storage unit and manages the procedures that make the sensor node collaborate with the other nodes to carry out the assigned tasks. A power unit is supported by solar cells. The mobilizer is used to move the sensor nodes when it is required to carry out the assigned tasks.

The location finding system is an application-dependent subunit which performs the tasks related to the network routing. As there are some subunits in a sensor node, so the size is the big constraint. To overcome this, these all components should be combined and packed in a small size. There are some other hurdles related to hardware and to overcome those sensor node must consume extremely low power, operate in high volumetric densities, and have low production cost, be expendable and autonomous and be adaptive to the environment.

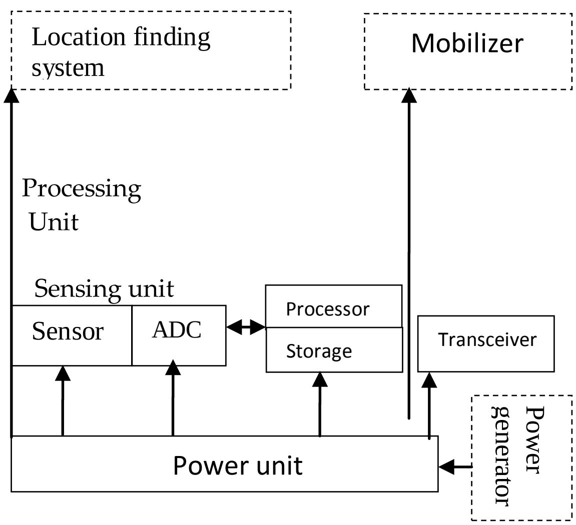


Figure7. The components of a sensor node

Sensor Network Topology discusses the issues related to the topology of the sensor network. As there can be few to few hundreds of sensor nodes in a network and deploying the sensor nodes densely requires the attention towards the topology maintenance. There are three phases where this topology maintenance will be a challenge they are pre-deployment and deployment phase, post-deployment phase and redeployment of additional nodes phase.

Environment and Transmission Media concern about the issue related to the environment where the sensor nodes are deployed and which type of media is used for transmitting the data.

Power Consumption means that the wireless sensor node being a microelectronic device can only be equipped with a limited power source. One of the important factors on which the sensor node lifetime depends is the battery lifetime. So the design of the node should be in such a way that it should consume less power. Therefore the power consumption can be divided into three domains: sensing, communication and data processing i.e. the node detect the event, quickly perform data processing and transmit the data.

These are some of the design factors on which the architecture of the sensor network depends. Let's discuss about the different protocols used by the sink and sensor nodes in a sensor network in the following section. The protocol stack combines power and routing awareness, integrates data with the networking protocols, communicates power efficiently through the wireless medium. Protocol stack compromises of different layers as discussed below.

The Physical Layer: The task of this layer is frequency selection, carrier frequency generation, signal detection, modulation and data encryption.

The Data Link Layer: Ensuring reliable connections (i.e. point-to-point and point-to-multipoint) in a communication network. It support the multiplexing of the data streams, data frame detection, medium access and error control. As in wireless network, several sensor nodes are present which are densely scattered therefore the MAC first creates the network infrastructure i.e. establish the communication links for data transfer. Secondly, it allows sharing of the communication resources between the nodes fairly and efficiently. The major focus of the MAC is to provide high QoS and bandwidth efficiency so the MAC used in wireless networks cannot be used in sensor networks.
Two MAC protocols have proposed they are Self-Organizing Medium Access Control for Sensor Networks (SMACS) and the Eavesdrop-And-Register (EAR) algorithm [11, 12].There is one more significant feature of the data link layer is error control of the transmission data.

Network Layer: The networking layer of the sensor networks is designed according to the following principles:

a) Sensor networks are mostly data-centric.
b) Power efficiency is always an important consideration.
c) An ideal sensor network has attribute-based addressing and location awareness.
d) Data aggregation is useful only when it does not hinder the collaborative effort to the sensor nodes.

Selection of an energy-efficient route from sink to the source node can be done by one of the following approaches [18]:

- Maximum PA route
- Minimum energy (ME) route
- Minimum hop (MH) route
- Maximum minimum PA node route

There is one more important functionality of the network layer is that it provides internetworking with external networks like other sensor networks, command and control systems and the internet. There are some schemes used in network layer like Minimum Energy Communication Network (MECN) [12] protocol use for computing an energy-efficient subnetwork, flooding [15] and its derivation gossiping [16] used for routing, Sensor Protocols for Information via Negotiation (SPIN) [15] designed for addressing the deficiencies of classic flooding by negotiation and resource adaptation etc.

Transport Layer: This layer comes in use when the system is planned to be accessed through the internet or other external networks. The TCP and UDP protocols match with the current characteristics of the sensor network environment. But these protocols are not based on global addressing, so there is need of the some new protocols for the transport layer considering the factors like power consumption and scalability, data-centric routing etc.

Application Layer: Large number of protocols available for this layer but according to research there are only three optimized protocols [18] are available for the application layer of the sensor networks. They are: Sensor Management Protocol (SMP), Task Assignment and Data Advertisement Protocol (TADAP), and Sensor Query and Data Dissemination Protocol (SQDDP).

The realization of the sensor networks needs to satisfy the constraints. Since these constraints are highly stringent and specific for sensor networks, new wireless ad hoc techniques are required.

3. CONCLUSION

The WSNs are very popular and in progress so in future it will grow more and more people can have its access. There will be some hurdles which have to overcome like the limited resource, space etc. for wireless networks. So, to overcome these types of problems it is advisable to use the data compression algorithms for transmitting the data. This study shows the different data compression algorithms. It depends on the user and data that which type of compression will be used. Use of compression algorithm can diminish resource constraint of wireless sensor nodes.

REFERENCES

[1] X. Kavousianos, E. Kalligeros, D. Nikolos, “Optimal Selective Huffman Coding for Test-Data Compression,” *IEEE Trans. Computers*,vol.56 pp. 1146-1152, Aug. 2007.

[2] A. Jas, J. Ghosh-Dastidar, M. E. Ng, and N. A. Touba, “An Efficient Test Vector Compression Scheme Using Selective Huffman Coding," *IEEE Trans. Computer-Aided Design*, vol. 22, pp. 797-806, June 2003.

[3]. Y.Kaufman and S.T.Klein, Dept. of CSE, “Semi-lossless Text Compression,” *International Journal of Foundation of Computer Science.*

[4]. http://en.wikipedia.org/wiki/BurrowsWheeler_transform.

[5]. N.Kimura and S. Latifi,” A Survey on Data Compression in Wireless Sensor Networks,” ITCC, 2005.

[6]. Kenneth Barr and Krste Asanovic, “Energy Aware Lossless Data Compression,” In First International Conference on Mobile System, Applications, and Services, May 2003.

[7]. T. Arici, B.Gedik, Y. altunbasak, and l.Liu, “PINCO: a Pipelined In-Network Compression Scheme for Data Collection in Wireless Sensor Networks,” In Proceedings of 12th International Conference on Computer Communications and Networks, Ocober 2003.

[8]. E. Magli, M. Mancin, and L. Merello, “Low- Complexity Video Compression for Wireless Sensor Networks,” In proceedings of 2003 International Conference on Multimedia and Expo, July 2003.

[9] J.Kusuma,L.Doherty,K.Ramchandran,” Distributed Compression for Sensor Networks,” In Proceedings of 2001 International Conference on Image Processing, October 2001.

[10].S.S.Pradhan, J.Kusuma, and K.Ramchandran, “Distributed Compression in a Dense Microsensor Network,” IEEE Signal Processing Magazine, Vol.19, issue2, pp.51-60, March 2002.

[11]. A.Woo, and D.Culler, “A Transmission Control Scheme for a Media Access in Sensor Networks,”Proc. ACMMobiCom01, Rome Italy, July 2001, pp.221-35.

[12]. K.Sohrabi et al., “Protocols for Self- Organization of a Wireless Sensor Network,” IEEE Pers. Commun., Oct. 2000, pp. 16-27.

[13]. E.Shih et al., “Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks,”Proc.ACM MobiCom 01, Rome, Italy, July 2001, pp.272-86.

[14]. V. Rodoplu and T.H.Meng, “Minimum Energy Mobile Wireless Networks,” IEEE JSAC, vol.17, no.8, Aug.1999, pp. 1333-44.

[15]. W. R. Heinzelman, J.Kulik and H. Balakrishan, “Adaptive Protocols for Information Dissemination in Wireless Sensor Networks,” Proc. ACM MobiCom 99, Seattle, WA, 1999, pp. 174-85.

[16]. S. Hedetniemi, and A. Lietman, “A Survey of Gossiping and Broadcasting in Communication Networks,” Networks, vol.18,1988.

[17]. C. Shen, C. Srisathapornphat, and C. Jaikaeo, “Sensor Information Networking Architecture and Application,” IEEE Pers. Commun., Aug. 2001, pp. 52-59.

ISSN 2229-5518
[18]. Ian F. Akyildiz, Weilian Su, Yogesh S. Subramaniam, and ErdalCayirci, "A Survey on Sensor Networks," IEEE Communication Magazine, Aug. 2002.

BIOGRAPHIES

Upasana Mahajan is M.Tech in Computer Science Engineering (2010) from VTU and passed B.Tech (CSE) from Guru Nanak Dev University in year of 2006. She is currently pursuing her PhD and has active research interest in area of Data Compression and their applications in wireless networks.

Dr. Prashanth C.S.R holds B.E, M.S (USA), PhD (USA). At present he is a Professor and Head of the Department for Computer Science in New Horizon College of Engineering Bangalore, Karnataka.