Partial DCT-based Energy Efficient Compression Algorithm for Wireless Multimedia Sensor Network

Rajib Banerjee
Dept. of Electronics and Communication Engineering
Dr.B.C. Roy Engineering College
Durgapur, India
Email: rajib123banerjee@gmail

Mohammad Mobashir
Dept.of Computer Science and Technology
Bengal Engineering and Science, University, Shibpur, Howrah, India
Email: mdmobashir@gmail.com

Sipra Das Bit
Dept.of Computer Science and Technology
Bengal Engineering and Science University, Shibpur, Howrah, India
Email: sdasbit@yahoo.co.in

***Abstract*-Recent advancement in hardware technology provides sensor nodes capable to process multimedia signal. However, the strict limitations in terms of processing power, storage, bandwidth etc. pose a great challenge to data processing in wireless multimedia sensor network (WMSN). In this paper we propose an energy saving audio data compression technique for WMSN using partial DCT (PDCT), where only the last DCT coefficient is propagated thereby saves energy by transmitting reduced number of bits. We have considered an application domain where data are co-related and we have also considered a 2D architecture where nodes are randomly placed. Along with the proposed PDCT based scheme a tree based routing scheme is also proposed with an aim to reduce energy consumption further. The design feasibility along with simulation results evaluate efficacy of the scheme in terms of two conflicting parameters viz. energy consumption and SNR. The comparative results confirm our scheme's supremacy over the competing scheme.**

***Index Terms*- Wireless Multimedia Sensor Network, Compression, Discrete Cosine Transform, Partial Discrete Cosine Transform, Tree Based routing.**

I. INTRODUCTION

Wireless sensor networks (WSNs) involve large number of micro sensor, called nodes and the main goal of these nodes is to collect information [1] from the target environment. With rapid development and miniaturization of hardware, a sensor node is equipped to collect multimedia signals. This has fostered the development of Wireless Multimedia Sensor Networks (WMSNs) [2],[3],[4]. However, the strict constrained resource of these nodes brings great challenges to data processing and communication procedure. Energy is one of the scarcest resources in WSN, especially in WMSN and therefore, importance should be given to save energy.

In most of the applications sensor nodes are spatially correlated, so transmitting only raw data increases transmission energy. Thus in order to save energy consumption in WMSN in-network processing is required. Data compression is one of the techniques for in-network processing. Compression can be primarily achieved by providing some transform techniques at the nodes by de-correlating data from the sensor end. Recent advancement, in WMSN has drawn a lot of attention to the development of variety of distributed algorithm to improve the performance of such networks. A number of works towards energy saving data compression for WMSN has been reported and a few of them will be focussed. A distributed data compressing technique for WSN is proposed by Jim Chou et al. [5]. The authors have considered a 2D regular architecture where only one node is elected to send uncompressed data. The other nodes are required to send compressed data. The sink node upon receiving the sensory data, decodes it through correlations between the compressed and uncompressed data. Although the scheme has achieved energy saving, it has considered only scalar data.

A. Ciancio et al. [6] has proposed a distributed wavelet compression algorithm for WMSN. They have considered linear placement of nodes and used lifting wavelet transform, thereby achieving energy saving with simultaneous quality reconstruction.

A lossless data compression algorithm based on H.264 integer DCT (Discrete cosine transform) is proposed by Ren. Xuejen et al. [7]. Original DCT algorithm is complicated to apply in sensor nodes, so H.264 is also difficult to apply to sensor nodes. Thus the contribution is to develop an algorithm for slowly varying data applications, keeping in mind about the limitation of DCT in sensor network.

Routing in Wireless Sensor Network plays a major role, since it implements the direction of data from the node to the sink. Thus energy efficient compression algorithm should be accompanied with proper routing. G.Vithya et al. [8] proposed an efficient routing algorithm which works dynamically according to the situation and network requirements.

G.Shen et al.[9] has also designed an optimized 2D transform based on wavelet lifting. Here tree based routing is accompanied with partial wavelet transform. Partial wavelet transform eliminates backward flow of data. Since Partial wavelet transform along with routing allows unidirectional transform computation, energy consumption reduces.

In another work by Fan Bai et al. [10], DCT data aggregation technique is used for clustered sensor networks where each sensor gathers data from the environment periodically. The algorithm exploits correlation using DCT. Each node sends its sensory data to a data aggregation point where DCT is applied and transformed data is transmitted to the sink. This scheme saves energy consumption in the network up to a certain limit.

In the present work we propose a data compression scheme where nodes are placed in 2D network area. In this scheme we have modified DCT into partial DCT with a target to reduce computational and communication overhead. A pre-computed routing strategy is also proposed. Partial DCT based technique along with proposed routing technique together have made possible to achieve energy saving compression, while keeping quality of reconstruction (SNR) to an acceptable limit.

The rest of the paper is organized as follows. Section II presents system architecture for the proposed scheme. Brief on discrete cosine transform is presented in section III. In section IV proposed scheme is elaborated. We provide performance of the scheme in terms of theoretical analysis on design and simulation results in section V. Finally, the work is concluded in section VI.

II. SYSTEM ARCHITECTURE

We consider the nodes are deployed randomly in a 2D area Fig.1. Sink is placed at the centre position of the area. Each of the nodes sends sensory data to the sink through multi hop communication. Further, we consider a node's communication range as R_c and two nodes placed within this range are considered as 1-hop neighbour nodes. The nodes know their own locations and the location of the sink.

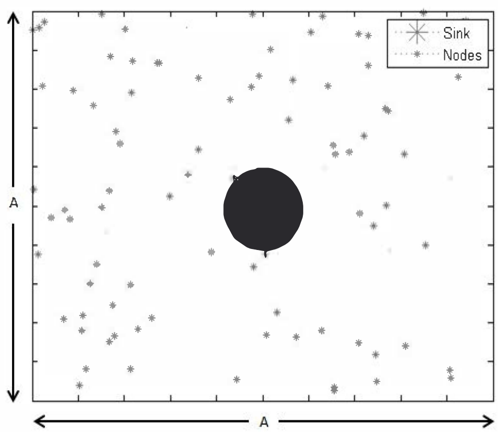


Fig 1. System Architecture

III. BRIEF ON DISCRETE COSINE TRANSFORM (DCT)

Discrete Cosine Transform (DCT) is a method to de-correlate the correlated data. The objective of employing DCT on a signal/data is to extract cosine parts from the sampled data thereby extracting real part only ensuring data compression. For example, if a set of correlated signal/data is accumulated at a point, DCT attempts to de-correlate the correlated [10] data into uncorrelated coefficients using a group of cosine functions. This helps in transmitting less data in order to save transmission energy. One can get back the original samples by employing inverse DCT (IDCT) on the uncorrelated coefficients. The most common mathematical expression of one-dimensional DCT [12] is as follows:

$$c(u)=\alpha(u)\sum_{x=0}^{n-1} f(x)\cos\left[\pi(2x+1)\frac{u}{2n}\right] \qquad (1)$$

where $c(u)$ denotes the u^{th} coefficient of n correlated data and f(x) denotes the x^{th} data for u, x = 0,1,…,n-1. Once the transformation has been performed, the original data can be reconstructed by employing IDCT using the following expression:

$$f(x)=\sum_{u=0}^{n-1} \alpha(u)c(u)\cos\left[\pi(2x+1)\frac{u}{2n}\right] \qquad (2)$$

In both the above equations, $\alpha(u)$ is a constant defined as:

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{n}} & \text{for } u = 0 \\ \sqrt{\frac{2}{n}} & \text{for } u \neq 0 \end{cases} \qquad (3)$$

Referring the above equations in this proposed scheme, n corresponds to the number of nodes along the multi-hop path through which sensory data reaches to the sink. Here $f(x)$ maps to data sensed at a node which is at x^{th} hop away from the sink. Now with the help of $f(x)$s and n, coefficient $c(u)$ is to be calculated.

IV. PROPOSED SCHEME

We propose an energy saving data compression scheme in WSN environment, where instead of the data being propagated, the components of the last DCT coefficient is propagated. Using this data accumulated at the sink, we aim to generate the remaining DCT coefficients and finally reconstruct the original signal. Prior to the start of the partial DCT computation at each node and the transmission of this data to the immediate predecessor in its route to the sink, it is necessary to provide a routing scheme.

A. Routing

We have modified, Lee[11] routing algorithm with an objective to have a pre computed routing to determine the shortest path from a node (source) to sink (destination). Lee algorithm is basically used in VLSI as a solution for maze routing problems, where initially a path is established from source to the destination (target) by performing breadth first search. The path would cross the least number of existing paths.

Our routing algorithm is also based on breadth first search technique with a difference that the searching starts in all direction around the sink. Therefore, unlike Lee's algorithm, here the algorithm starts by considering sink to propagate radio signal in multiple direction. Moreover instead of considering regular, predefined topology as in [11], our routing algorithm considers irregular placement of nodes with no knowledge of topology apriori. We generate the shortest path from each node to the sink via intermediate nodes (multi-hop). It is to be noted, that the algorithm avoids the case of loop generation in the path through the formation of a tree.

As mentioned in section II, after deployment each node knows its own location. Each of the nodes store a 3-tuple (own-location, own-hop-id, parent-location) data out of which initially two attributes own-hop-id, and parent-location are set to null. Once the nodes are deployed, the sink starts pre-computation by broadcasting a 2-tuple (own-location, own-hop-id). The hop-id of sink is assumed as -1. The nodes within communication range of sink receive the broadcast and set their own-hop-id as zero (received own-hop-id+1) and the location of sink as parent-location. Now the nodes with hop-id zero in turn broadcast the 2-tuple (own-hop-id, own-location) packet. The nodes which are yet uninitialized and located within communication range of the nodes with hop-id zero receive the broadcast packet and set their own-hop-id as 1 and parent-location as location of sender of the broadcast (received own-location). In this way, the process is continued until all the nodes have been assigned their hop-id and parent's location. Once this process of pre-computation is over, paths from each of the nodes towards sink through respective parent nodes is established and a tree is formed as shown in Fig.2.

Once a node has been assigned a parent, it would not be considered in the next iteration and thereby one node can have one and only one parent satisfying the property of a tree data structure. As a tree is formed, the path from any node to the sink is the shortest. The result after running the pre-computation process is shown in Fig.2(a) which shows hop-id assignments and path establishment up to hop-id 3 and Fig.2(b) highlights one of the established paths along with direction of data flow.

Once the nodes are deployed the routing pre-computation is initiated by the sink. However, all the nodes in turn run this algorithm. Whenever the pre-computation is over, the network is in operation. Now if a node wants to send data to sink, it sends the data along the established path.

B. *Routing Algorithm*

Algorithm

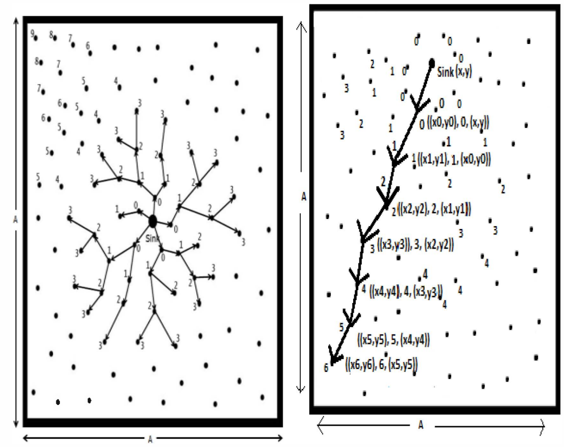
Input: Nil

Output: Establishment of unique, shortest path between each of the nodes to sink

```
Begin
for each node n_i /* Initialization*/
    hop-id_i := 0  /* own-hop-id */
    n_i_Parent:= null /* parent-location */
    store (own-location,own-hop-id, parent-location)
        /*own-location is known a priori*/
end for
for each node n_i in the R_c range of sink do
    n_i_Parent = sink /* parent-location */
end for
    tree_form(node n)
    broadcast 2-tuple (own-hop-id, own-location)
for each node n_i in the R_c range of n and n_i_Parent
    == null do
        hop-id_i  := hop-id_n  + 1   /* n's own-hop-id */
        n_i_Parent = n             /* n's own-location */
        tree_form(node n_i)
end for
```

C. *Partial DCT based compression*

Once the routing paths for all the nodes are established to send data towards sink, the network starts operation. We resort to computation of partial $(n-1)^{th}$ DCT coefficient computation where each node computes a part of the $(n-1)^{th}$ DCT coefficient and forwards it to the immediate next node in the path towards sink. Here the path comprises of n nodes with 0 to (n-1) hop-id. The steps of operation for the proposed compression scheme are given as follows.



(a) Hop-id assignment (b) One route (zoomed)

Fig 2. Routing

To start with (say at time t) each of the nodes collects data from environment. All the nodes compute respective parts of the $(n-1)^{th}$ coefficient. Next time onwards a node senses data as well as forwards the partial coefficient (if any) received at previous instance towards its next hop along the routing path. In the next instance the nodes again compute partial coefficients and forward buffered coefficients from previous instance. In this way at each instance sink receives part of $(n-1)^{th}$ coefficients which are kept in it. When all the parts of the coefficient are received, one iteration is over. The entire process of partial DCT based compression is explained further with the help of mathematical expression of DCT. Referring equation (1), the $(n-1)^{th}$ DCT coefficient c(n-1) is to be calculated as follows:

$$c(n-1)=\alpha(n-1)\sum_{x=0}^{n-1} f(x)\cos\left[\pi(2x+1)\frac{n-1}{2n}\right] \quad (4)$$

This can be expanded as follows:

$$c(n-1)=\alpha(n-1)\left\{f(0)\cos\left[\pi\frac{n-1}{2n}\right]+......+f(n-1)\cos\left[(2n-1)\pi\frac{n-1}{2n}\right]\right\} \quad (4.1)$$

Thus a node i^{th} hop away from the sink computes a fraction of the entire c(n-1) given by

$$c(n-1)_i=\alpha(n-1)f(i)\cos\left[\pi(2i+1)\frac{n-1}{2n}\right] \quad (4.2)$$

This i^{th} node after computing a part of the coefficient forwards it along with the fraction obtained from $(i+1)^{th}$ hop-away node to $(i-1)^{th}$ until after successive such relay actions, complete c(n-1) is accumulated at the sink as follows

$$c(n-1)=\sum_{i=0}^{n-1} c(n-1)_i \quad (4.3)$$

Now from all the n partial coefficients of $(n-1)^{th}$ DCT accumulated at the sink, remaining DCT coefficients are to be computed for getting the entire DCT matrix.

The j^{th} column of i^{th} row of DCT matrix is given by

$$c(i)_j=\alpha(i)f(j)\cos\left[\pi(2j+1)\frac{i}{2n}\right] \quad (4.4)$$

Now employing Eq (4.2) & (4.4) and applying equality of matrix property on it, we get the following equation.

$$c(i)_j = c(n-1)_j \left[\frac{\alpha(i)}{\alpha(n-1)}\right]\left[\frac{\cos\left[\pi(2j+1)\frac{i}{2n}\right]}{\cos\left[\pi(2j+1)\frac{n-1}{2n}\right]}\right] \quad (4.5)$$

At the end of one iteration sink with the help of c(n-1), computes c(n-2), c(n-3), . . ., c(0).

Finally when all the n number of DCTs is computed at sink, it reconstructs the original signal by performing IDCT using equation (2) as follows:

$$f(i) = \sum_{u=0}^{n-1} \alpha(u)c(u)\cos\left[\pi(2i+1)\frac{u}{2n}\right] \quad (4.6)$$

Referring equation (4.2) it is to be noted that the parameters in the cosine part are fixed for each and every node, although unique for different nodes in any path. So the computation at every node involves multiplication of the sensed data f(i) with a constant thereby reducing computation overhead and that helps to make the scheme light-weight.

D. Compression Algorithm

The entire operation is split up into the processing at the sensor nodes, and the processing at the sink while reconstructing the original data.

1) At the sender end (Sensor nodes)

Input: Sensory data and data received from preceding nodes.

Output: Partial coefficient of $(n-1)^{th}$ DCT

/* n_i : node having $ID = i$ assigned during routing */

Begin

1. **for each** node n_i in the path to the sink **do**
2. calculate $c(n-1)_i^t$ /* partial coefficient at time t using equation (4.2) */
3. quantize $c(n-1)_i^t$
4. forward the following data to the n_{i-1}-hop-away node along the path
5. forward the following data
6. **end for**

2) At the receiver end (Sink)

Input: $c(n-1)^t$ the $(n-1)^{th}$ DCT coefficient received from a node at t^{th} iteration.

Output: Regenerated audio signal sensed at the node and propagated to the sink

Begin

1. **for each** $c(n-1)_i^t$ received, **do**
2. .calculate remaining $c(i)_j^t$, $0 \le j < n-1$ /*using equation (4.5) */
3. Calculate $c(u) = \sum_{j=0}^{n-1} c(u)_j^t$
4. reconstruct original data

$$f(i) = \sum_{u=0}^{n-1} \alpha(u)c(u)\cos\left[\pi(2i+1)\frac{u}{2n}\right]$$

5. **end for**

E. Illustrative Example (compression)

Let us consider there are four nodes with hop-id 0,1,2,3 along a routing path towards sink. The sink maintains a table structure Fig.3 to keep the data received from nodes where each row corresponds to 3^{rd} DCT at one time instance and each column represents partial coefficient of 3^{rd} DCT at different time instances. To start with (T=1) all four nodes calculate partial co-efficient of 3^{rd} DCT based on their respective sensory data such as Node 0 → $c(3)_0^1$,Node 1 → $c(3)_1^1$,Node 2 → $c(3)_2^1$,Node 3 → $c(3)_3^1$. In the next time instance (T=2), four partial coefficients $c(3)_0^2$, $c(3)_1^2$, $c(3)_2^2$, $c(3)_3^2$ of 3^{rd} DCT calculated at four nodes 0,1,2,3 respectively are propagated towards sink. At the end of this instance, node 0 accumulates two coefficients. One of them is $c(3)_1^1$ which was calculated at node 1 at T=1 and the other is $c(3)_0^2$ which is calculated at current instance T=2 at node 0. So the two coefficients $c(3)_1^1$ and $c(3)_0^2$ are stored at sink as shown in Figure 3. In this way, at the end of T = 4, all the partial coefficients computed at T=1 is ready with the sink. Based on these partial coefficients c(3) is computed and this will in turn help to compute c(2), c(1), c(0) using equation (4.5). Once all the c(0),c(1),c(2) and c(3) are available the original data is to be reconstructed using equation (4.6).

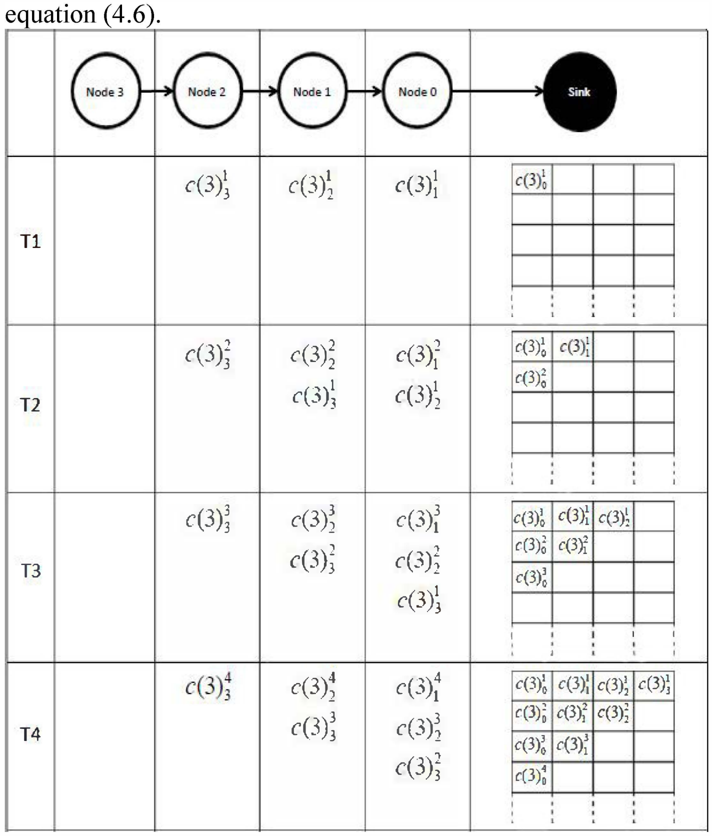


Fig 3. An illustration of compression scheme in a path

V. PERFORMANCE EVALUATION

The performance of the proposed scheme is evaluated through theoretical analysis and simulation.

A. Theoretical analysis

The design of the proposed audio compression technique resolves the trade-off between processing complexity and loss of sensory data optimally. In the proposed scheme each node performs the calculation of partial DCT coefficient. The resultant coefficient along with the data received from the previous nodes along the path is then forwarded to the parent node. The expression of partial coefficient that is generated at each node is expressed in equation (4.2) where α(n-1) cosine part are constants computed during routing and f(i) is the data sensed by the node i^{th} hop from the sink. Thus the computation carried out at each node i^{th} hop away from the sink can be expressed as

$$c(n-1)_i = K * f(i) \qquad (4.7)$$

.

where K is a constant formed by the multiplication of two constants *α(n-1)* and cosine part during the pre-computed routing phase. We claim that in the design of the scheme computational complexity for compression is not at the cost of any sensory data loss. Considering Strong ARM SA-1100 [13] micro controller based mote, we establish our claim as follows:

Number of required instruction cycle:
Multiplication: 1 (for getting K) + 1(K * f(i))
Load: 1 (LOAD A, f(i))
STORE: 1
Total number of instruction cycles required to compute partial DCT at each node is 3.
Execution time [12] for 1 instruction cycle:
$1/(206 \times 10^6)$ sec $= 4.85 \times 10^{-9}$ sec
for 206MHz Processor
$1/(133 \times 10^6)$ sec $= 7.518 \times 10^{-9}$ sec
for 133 MHz Processor
So, in case of 206 MHz processor, execution time (T_{pDCT})for partial DCT computation:
$T_{pDCT} = 3 \times 4.85 \times 10^{-9}$ sec $= 14.55 \times 10^{-9}$ sec
Similarly for 133MHz processor $T_{pDCT} = 3 \times 7.518 \times 10^{-9}$ sec $= 22.554 \times 10^{-9}$ sec

Now to maintain a reasonable audio quality, sampling rate of 44.1kHz needed (to achieve audio CD quality), which means that between two consecutive samples time interval ($T_{sampling}$)= $1/44100 = 0.267 \times 10^{-6}$ sec. As for both the processor speed $T_{pDCT} < T_{sampling}$, no samples are missed. For multiple coefficient at merged node also, only one instruction cycle will increase, thus resulting in no missed samples while sensing.

B. Simulation

We have collected sample values from a busy traffic of a metropolitan city and used as input data. Experiments are carried out by taking data from the collected audio data set for duration of 17 minutes. The simulation is performed using Matlab 7.9. First Order Radio Model is used to compute the average energy consumed by a node involved in both transmitting and receiving an N bits packet over a distance'd'[6][15][16].The performance of the proposed scheme is primarily evaluated by considering Energy consumption due to transmission and reception as well as on SNR.

C. Results and Discussion

In this paper three set of experiments are performed to evaluate the performance of the proposed scheme and to compare the results with competing schemes. In the first set of experiment, average energy consumption is plotted in Fig.4. Here quantize bits per sensor node is used as a varying parameter which represents the variation of quantizer size for the data to be transmitted.

It is evident from the plots that as number of quantize bits/ sensor node is increased, the quantizer size also increases thus average energy consumption throughout the network increases. This set of results is an important guideline in determining the size of quantizer. We observe that PDCT (Partial discrete cosine transform) consumes the least energy compared to the PDWT (Partial discrete wavelet transform) and RAW.

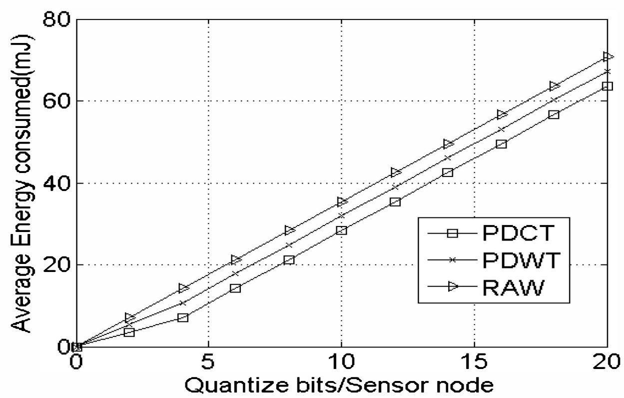


Fig 4. Energy consumption

To be more specific, our scheme PDCT saves on an average 8.9% and 17.3% more energy than the PDWT and the raw transmission respectively.

The second set of experiment deals with measuring SNR (signal to noise ratio) by varying quantize bit per sensor node as shown in Fig. 5. It is well known that energy consumption and SNR are two conflicting requirements. If high quality SNR is to achieve more energy is to be spent in transmission. On the other hand, if energy is to be conserved at any cost, then the SNR has to be compromised. Thus it is important to optimize these two parameters to get considerable amount of energy conservation in the wireless nodes, still getting acceptable value of SNR.

We observe that SNR of PDCT is better than RAW and PDWT. RAW is slightly under performed compared to PDCT since in RAW transmission as no transform technique is adapted, quantization accuracy is affected. However in between 10 to14 bit/sensor SNR of PDCT is almost similar with RAW transmission. The reason for PDWT's relative under performance is due to the fact that at each node, full as well as partial coefficients are quantized and the same is again de-quantized at the next node as it propagates along the path towards the sink. This increases the chance of introducing error in the signal, thereby resulting poor signal reconstruction.

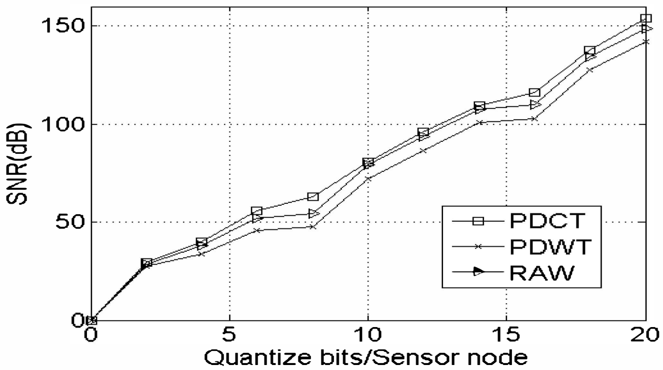


Fig 5. SNR measurement

This is in contrast to our PDCT scheme where a quantized value needs to be de-quantized at sink only.

Finally in the last set of experiment, SNR is plotted against average energy consumed in Fig.6 which helps us to fix the design trade-off between these two parameters . Thus the size of the quantizer can be known and designed for real time application by checking these two parameters. The nature of the curve is justifiable since the better the signal we need at the sink, the more the energy has to be consumed at each node.

Further, considering acceptable SNR in wireless network [14] as 25-40 dB, to achieve this acceptable SNR, PDCT scheme require 7.07 mJ whereas PDWT requires 10 mJ while Raw requires 14.145 mJ. This establishes the supremacy of our scheme PDCT where energy consumption due to transmission and receiving k bit packet at a distance d is least along with achieving appreciable SNR.

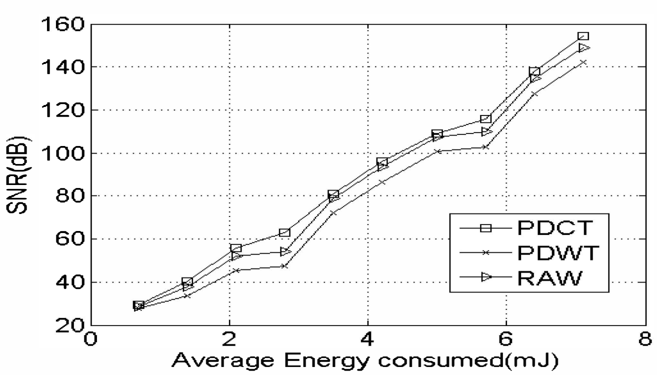


Fig 6. SNR Vs. Average Energy consumed

VI. CONCLUSION

In this work we have proposed an audio compression algorithm for WMSN in a 2D grid. Considering the spatial correlation of data, we use PDCT scheme where the last DCT coefficient are propagated. The sink, using this last coefficient generates the remaining DCT coefficients and finally reconstructs the original signal. Moreover, a tree based routing scheme provides loop-free and shortest path from the nodes to the sink is also designed. Finally the scheme is compared with the existing schemes and the simulation results show the supremacy of our scheme in terms of average energy consumption and SNR. The merit of our scheme lies on the fact that instead of computing the entire DCT matrix in resource-constrained nodes, the sink which is relatively resource-rich, computes the matrix with the help of partial data sent by the nodes and reconstructs the original data.

As a future work, the proposed scheme may be improved by reducing energy consumption further by exploiting temporal correlation of data accompanied by adaptive path merging techniques.

REFERENCES

[1] Ian F. Akyildiz , Tommaso Melodia , Kaushik R . Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks* 51, pp. 921-960, 2007.

[2] Ipsita Dutta, Rajib Banerjee, Tamaghna Acharya, Sipra DasBit, "An energy efficient audio compression scheme using wavelet with dynamic difference detection technique in wireless sensor network", *ICACCI'12, ACM*, pp: 360 – 366, August 3-5, 2012.

[3] Ipsita Dutta, Rajib Banerjee, Sipra DasBit, "Energy efficient audio compression based on red black wavelet lifting for wireless multimedia sensor network",accepted in *ICACCI, IEEE*, pp:1070-1075, 2013

[4] S.Halder, A.Ghoshal and S.DasBit, "A Pre-determined node deployment strategy to prolong network lifetime in wireless sensor network," *Elsevier, Journal Computer Communications*, Vol: 34 Issue 11, pp. 1294-1306, 2011.

[5] J. Chou, D. Petrovic.K. Ramchandran, "Signals,Tracking and exploiting correlations in dense sensor networks," *The Thirty-Sixth Asilomar Conference on Systems and Computers*, Vol: 1, pp. 39 – 43, 2002.

[6] A. Ciancio and A. Ortega, "A distributed wavelet compression algorithm for wireless sensor networks using lifting," in Proceedings of the International Conference on Acoustics, Speech and Signal Processing *ICASSP04, IEEE, Vol:4, pp. 633-636, Montreal, Canada, May 2004.*

[7] Ren Xuejun,Fang Dingyi"A sensor node compression algorithm based on H.264 integer DCT transform in WSN," *IET WSN*, pp: 214 - 217 , 2010.

[8]G.Vithiya, B.Vinayagasundaram, "Actuation sensor with adaptive routing and QOS aware checkpoint arrangement on wireless multimedia sensor network",*ICRTIT, IEEE*, pp: 444 – 449, chennai, june 3-5 2011.

[9] Godwin Shen and Antonio Ortega , "Optimized distributed 2D transforms for irregularly sampled sensor network grid using Wavelet lifting", *ICASSP, IEEE*, pp: 2513 – 2516, March, April 2008.

[10] Fan Bai, Abbas Jamalipour, "3D-DCT data aggregation technique for regularly deployed wireless sensor networks", *ICC, IEEE*,pp: 2102-2106, 2008.

[11] Lee,c,Y, "An algorithm for path connection and its application" *Electronic Computers, IRE Transactions*, vol:10,Issue3, pp:346-365,Sept 1961.

[12] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete Cosine Transform", *IEEE Trans. Computers*, vol: C-23, issue 1, pp 90-93, Jan 1974."

[13] Intel strong ARM SA-1100 Microprocessor developer's manual, august 1999.

[14] wireless.nets.com/resources/tutorials/define_SNR_values.html.

[15] Wendi B. Heinzelman, P. Chandrakasan, Hari Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks", *IEEE transaction on wireless communication* Vol-1, No-4, pp 660-670 October 2002.

[16] M.A. Labrodor, P.M.Wightman, "Topology control in wireless sensor networks", *Springer Science and Business Media* B.V. 2009..