Efficient Compressive Sensing based Technique for Routing in Wireless Sensor Networks

WALID OSAMY [1]
AHMED SALIM[2]
AHMED AZIZ[1]

[1]CS Dept., Fac. of CS and Informatics, Univ. of Benha, Benha, Egypt
[2] Mathematics Dept., Fac. of Sci., Univ. of Zagazig, Zagazig, Egypt
[2]a.salim@zu.edu.eg
[1](ahmed.aziz,walid.osamy)@fci.bu.edu.eg

Abstract. Energy consumption and prolonging network lifetime are a primary challenge in many studies on Wireless Sensor Networks (WSN). Thus, since radio communication and routing protocol transmission are in general the main cause of power consumption, different techniques proposed in literature to improve energy efficiency have mainly focused on limiting transmission/reception of data. To this aim, we propose an adaptive and efficient technique based on compressive sensing for improving the performance of routing in wireless sensor network. The performance of our technique is evaluated by applying it to PEGASIS (power efficient gathering in sensor information systems), which is one of the most popular protocols for routing in wireless sensor network. A comparison of PEGASIS and PEGASIS with Huffman coding shows the advantage of the proposed technique in terms of reducing the energy consumption and network lifetime.

Keywords: Wireless sensor networks, Compressive Sensing, Routing.

(Received February 22nd, 2013 / Accepted May 8th, 2013)

1 Introduction

Wireless sensor networks (WSNs) can be used for a wide variety of applications dealing with monitoring (health environments, seismic, etc.), control (object detection and tracking), and surveillance (battlefield surveillance, perimeter and topology discovery [2, 1, 3].

Regardless the application in which the sensor network is serving, the data generated in the network eventually must be delivered to the sink. However, the limited network bandwidth, node/link failure along with the unreliable communication medium pose great challenges on the sensor network communication paradigms.

The simplest approach to collect data from sensor nodes is direct one where each sensor node transmits the data directly to the base station (BS) which is located far away. The Cost of data transmission from each sensor node to BS is very high, thus nodes die quickly and hence reducing the lifetime of the network. Therefore, use a few transmissions as possible leads to efficient energy utilization. Routing protocols from one of the most important communication paradigms that greatly affect the performance of the wireless sensor networks; so that designing routing protocols for sensor networks is a vital aspect. Many routing protocols have been proposed [4, 5, 6] in which consider reducing the amount of data transmissions in a WSN by fusing (or aggregating) these sensing data. In particular, the data aggregation techniques usually select a subset of sensor nodes (called aggregation nodes) to collect the sensing data sent from their neighboring sensor nodes and then fuse these sensing data. In this case, the amount of sensing data transmitted to the sink can be significantly reduced.

One example of routing protocols in WSNs is Power Efficient Gathering in Sensor Information Systems (PEGASIS) [6]. PEGASIS protocol distributes the energy load evenly among the sensor nodes in the network. PEGASIS protocol uses Greedy algorithm to form a chain of the sensor nodes starting from the node farthest to the sink node then randomly selects a leader for the formed chain. Each node in the chain receives from and transmits to close neighbors and continue communicating in their turns until the aggregated data reaches the leader which transmits them to the base station (BS). In that protocol, sensor nodes suffer from heavy data traffic. As a result, compression methods are often adopted to reduce the data size and reduce the required bandwidth for transmitting data (i.e., reducing the energy consumption arising from communications). Compressive sensing [8] and Huffman coding [9] are the representative examples of compression methods. In this paper, we propose a technique called ECST which is an adaptive and efficient compressive sensing based technique to improve the performance of routing in wireless sensor network by compress sensor reading while relaying them to the base station.

The rest of the paper is organized as follows: Section 2 briefly review related work. Compressive Sensing background is presented in section 3. In section 4, we introduce our approach to carry out the proposed problem. In Section 5, we give an example scenario. The simulation of our approach is presented in section 6. In section 7, we conclude our work.

2 Related Research

During the past few years surveillance and monitoring applications using WSNs have attracted a lot of attention from the research community. The associated functionalities form a canonical class of applications which can be feasible only with WSNs. The work presented in this paper has been inspired by various existing research efforts. Due to the severe energy constraints of the large number of densely deployed sensor nodes, various routing protocols for wireless sensor network are discussed and compared. For example, flooding is a technique in which a given node broadcasts the received data and control packets to the rest of the nodes in the network. This process is repeated until the destination node is reached. Note that the flooding technique does not take into account the energy constraint imposed by WSNs. So that, when it is used for data routing in WSNs, it leads to the problems of implosion and overlapping [17]. To overcome the shortcomings of flooding, another technique known as gossiping can be applied [18]. In gossiping, upon receiving a packet, a sensor would select randomly one of its neighbors and send the packet to it. The same process is repeated until all sensors receive this packet. Using gossiping, a given sensor would receive only one copy of a packet being sent. While gossiping tackles the implosion problem, there is a significant delay for a packet to reach all sensors in a network. LEACH (low energy adaptive clustering hierarchy) [4] is the first hierarchical cluster-based routing protocol for WSN. LEACH protocol partitions the sensor nodes of WSN into clusters; each cluster has cluster nodes (CNs) and cluster head (CH). CH receives data from CNs in their cluster, aggregates the data, and forwards them to the sink. LEACH protocol achieves even energy dissipation by randomly rechoosing CH at regular intervals. It leads to an eight times improvement compared to the direct transmission protocol. An Efficient clustering protocol in the large scale WSN [5] is self-organizing and adaptive multi-hop clustering protocol that uses efficient MAC to distribute the energy load evenly and guarantee minimum energy consumption for large scale WSNs and the ones that deployed in frequently ideal environments due to low data occurrence. PEGASIS (Power-Efficient gathering in Sensor Information Systems) [6, 11, 13], introduces only a routing protocol that is near optimal for a data-gathering problem in sensor networks and didn't assume any kind or type of data compression. The main idea of the PEGASIS protocol is the formation of a chain among the sensor nodes so that each node will receive from and transmit to a close neighbor. Gathered data moves from node to node, get fused, and eventually a designated node transmits it to the BS. The PEGASIS protocol achieves improvement varies between 100 to 300% when 1%, 20%, 50% and 100% of nodes die in the deployed field compared to the LEACH protocol.

In [23], Zytoune et al. present a Stochastic Low Energy Adaptive Clustering Hierarchy protocol (SLEACH), which outperforms the LEACH when the interesting collected data is the minimum or the maximum value in an area. SLEACH uses the same method proposed in LEACH for forming clusters. Once the clusters are formed, the cluster head broadcasts in its cluster a data message containing its measurement assuming the pertinent value. Only the nodes, having most significant data, send their messages towards the cluster-head.

In [19], the authors extend the SLEACH algorithm by modifying the probability of each node to become cluster-head based on its a required energy to transmit to the sink. Their contribution consists in rotation selection of cluster heads considering the remoteness of the nodes to the sink, and the network nodes residual energy. In [20], the authors proposed an algorithm called TB-LEACH which is an improvement of LEACH. TB-LEACH constructs the cluster by using an algorithm based random-timer, which doesn't require any global information. The work described in [22] proposed Variable-Round LEACH (VR-LEACH). VR-LEACH changes the round time according to the residual energy of the cluster head at the beginning of the round, the energy cost in every frame and the constants λ. In VR-LEACH a constant value for λ and the frame time μ are determined each round time, where the values of λ, and μ experimentally defined. The work in [21] proposed Energy-LEACH (E-LEACH) protocol which is an improvement of LEACH.

All the above routing protocols cannot satisfy the huge data traffic on wireless sensor networks. So, it is effective to apply compression before transmitting data to reduce total power consumption by a sensor node. For these reasons, researchers have therefore designed and developed various compression algorithms specifically for WSNs. There are two general approaches for data compression in WSNs . One is the distributed data compression approach [7, 10] , and the other is the local data compression approach [16].

The Local data compression approach is based on temporal correlation in WSNs. One example of this approach is Huffman coding[9]. Huffman coding is a lossless compression algorithm, it uses a variable length code according to the occurrence frequency of symbols. Therefore, short bits are allocated to high frequent symbols and long bits are allocated to the relatively low frequent symbols. The main disadvantage of local data compression algorithms is that large number of complex computations necessary by individual sensors in the network. These complex computations increase processing and power consumption and are therefore not a good fit for WSNs. Distributed data compression is based on the idea of reducing the complex computations required by the individual sensors of the network and exploiting correlated data at the sink node; example of distributed data compression can be found in [7]. The main disadvantage of this approach is that they predefine certain data to be main data and other data to be side data.

A new concept of signal sensing and compression has been developed [16, 8] called compressed sensing (CS), or compressive sensing. CS can sample a signal far below the Nyquist rate if the signal has a sparse representation in one basis. In CS, the signal is sampled and compressed simultaneously and accurately reconstructed with high probability.

The work presented in [12] is the first work completely introducing compressive sensing to wireless sensor network, named Compressive Data Gathering (CDG). It shows a routing tree in which the sink has four children. Each of them leads a subtree. Data gathering and reconstruction are performed on the subtree basis. In order to combine sensor readings while relaying them, every node needs to know its local routing structure. It focuses on the introduction of the concept, rather than analyzes and solves the energy balance problem. Furthermore, the work did not consider the effect of the choice of the coefficient matrix on the information content and dependent on grid topology assumptions which are not suitable with many real applications of wireless sensor network.

In this paper, we propose an adaptive and efficient CS based technique (ECST) to improve routing in wireless sensor network, considering that the compression scheme designed for WSNs should be lightweight, and the computational requirements of the algorithms should be low for efficient operation due to WSNs constraints in terms of hardware, energy, processing, and memory. ECST reduce the energy consumption prolonging the life of the whole network remarkably. The Simulation shows that the proposed protocol achieves a longer network lifetime compared with PEGASIS, and PEGASIS with Huffman.

3 Compressive Sensing Background

In WSNs, the distributed sensors observe physical changes in designing area. Since each sensor observes similar physical changes, the signals observed from each sensor have much correlation. The correlated signal can be compressed for reducing data. The conventional compression (e.g., Joint Entropy) for WSN requires communication between nodes and exploits correlated data in the compression process. Such a transmission strategy makes the network system complex. In contrast to the conventional schemes, Compressive Sensing is a new decentralized compression technology that achieves low complexity at the individual sensors, and its ability to exploit correlated data.

Compressive sensing (CS) is a new theory of sampling in many applications, including data network, sensor network, digital image and video camera, medical systems and analog-to-digital convertors [14]. CS enables a potentially large reduction in the sampling and computation costs for sensing data that have a sparse or compressible representation without relying on any specific prior knowledge or assumption on data [8]. The compressed sensing theory points out that any sufficiently compressed data can be accurately recovered from a small number of measurements without going through many complex signal processing steps.

The fundamental idea behind CS is to provide a direct method which acquires compressed samples without going through the intermediate stages of conventional compression. In addition, the CS provides several recovery routines which the original signal can be regenerated perfectly from the compressed samples [8].

3.1 Mathematical Definition

Consider a real-valued,finite-length, one-dimensional, discrete-time signal x, which can be viewed as an $N \times 1$ column vector in R^N with elements $x[n], n = 1, 2, ..., N$. Any signal in R^N can be represented in terms of a basis of $N \times 1$ vectors $\{\Psi_i\}_{i=1}^{N}$. For simplicity, assume that the basis is orthonormal. Using the $N \times N$ basis matrix $\Psi = [\Psi_1|\Psi_2|\Psi_3|...|\Psi_N]$ with the vectors Ψ_i as columns, a signal x can be expressed as

$$x = \sum_{i=1}^{N} s_i \Psi_i \;\; or \;\; x = \Psi s \tag{1}$$

where s is the $N \times 1$ column vector of weighting coefficients. The matrix $\Psi \in R^{N \times N}$ is an orthonormal basis. The signal x is K-sparse if it is a linear combination of only K basis vectors; that is, only K of the s_i coefficients in equation 1 are nonzero and $(N - K)$ are zero. The case of interest is when $K << N$. The signal x is compressible if the representation in equation 1 has just a few large coefficients and many small coefficients. The compressive measurements y (compressed samples) is obtained via linear projections as follows :

$$y = \Phi x = \Phi \Psi s = \Theta s \tag{2}$$

,where the measurement vector is $y \in R^M$,with $M < N$, and $\Theta = \Phi\Psi$ is the measurement matrix $\Theta \in R^{M \times N}$.

4 Efficient Compressive Sensing based Technique (ECST)

In this section, we outline our ECST technique for improving the performance of routing protocol in wireless sensor network which based on compressive sensing (CS) technique. Unfortunately, the computational complexity of CS makes it inefficient to be adopted directly in sensor nodes. The reason behind that is the energy limitation of sensor nodes. For that, our proposed technique shifts the computational complexity from the sensor nodes to the base station, where most of the computations will be at the base station and only minor computations will be at each sensor node. The contribution of this paper is summarized in three phases; Chain Construction and Leader Selection Phase, Data Compression and Gathering Phase, and Reconstruction Phase. Fig. 1 shows the flow chart of our ECST technique.

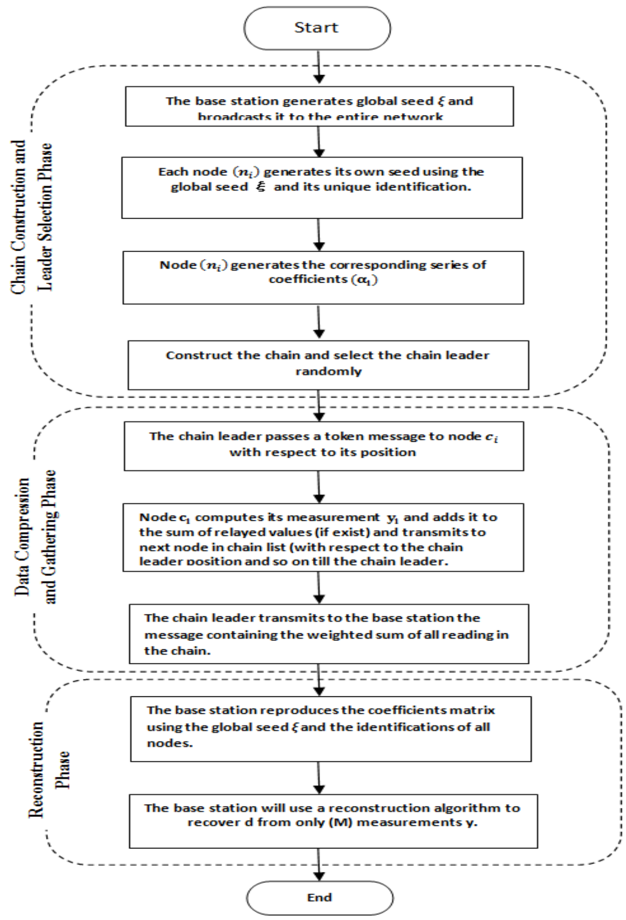

Figure 1: The flow chart describing the process of overall operations in our approach.

4.1 Chain Construction and Leader Selection Phase

4.1.1 Step One

The base station generates global seed ξ and broadcasts it to the entire network.

Upon the global seed ξ is received by each node (n_j) in the network, node n_j generates its own seed using ξ and its unique identification $(n_j.id)$, and then generates the corresponding series of coefficients $\alpha_j =$

$\begin{pmatrix} \Phi_{j1} \\ \vdots \\ \Phi_{jM} \end{pmatrix}$, that will be used to compute its measurement y_j. These coefficients can be reproduced at the base station given that the base station knows the global seed ξ and the identifications of all nodes.

Therefore, the global seed ξ can affect both the coefficient of a measurement and the information content. For that we propose an adaptive technique, in which the base station decides whether it need to dynamically change the global seed ξ or not upon calibration the measurement with a random small number of sensor reading which return to the sink from time to time. After that, the base station broadcasts the new global seed to the entire network and sends a message to chain leader and waits for the measurement vector y to return.

- ECST construct a chain as in PEGASIS, wherein it starts with the *furthest node* c_0 from the BS and add itself to the chain, then c_0 updates the chain list with the closest neighbor node c_1. Node c_1 repeats the same step by updating the chain list with the closest unvisited neighbor node and so on till all the nodes are included in the chain. This method of communication reduces the power consumption required to transmit data per round.

- At each round r, chain leader is selected randomly at a random position in the chain. A chain member node considers itself as a chain leader, if its identification number equals $r \ mod \ N$, where N represents the total number of nodes.

4.2 Data Compression and Gathering Phase

The main objective of this phase is to compress sensor readings and gather it to reduce global data traffic and distribute energy consumption evenly to the prolong network lifetime. In order to combine sensing reading while relaying them, every node c_i computes the measurement $y_i = \alpha_i d_i$ and transmits the measurement y_i to c_{i+1}. After that, node c_{i+1} computes the measurement $y_{i+1} = \alpha_{i+1} d_{i+1}$ and transmits $y_i + y_{i+1}$ to c_{i+2}. Once c_{i+2} received the values, it computes its measurement y_{i+2}, adds it to the sum of relaying values and transmits the combined value to next node in chain list (with respect to the chain leader position in the chain list)and so on till the chain leader. Finally, the chain leader transmits to the base station the message containing the weighted sum of all reading in the chain.

The operations in this phase will be as follows:

- If chain leader at the end of chain list:
  - The chain leader passes a token message to the first node c_1.
  - Node c_1 computes measurement y_1 and transmits the measurement y_1 to c_2.
  - Node c_2 computes the measurement y_2 and transmits $y_1 + y_2$ to c_3, and so on till chain leader.
  - Then the chain leader node receives data from c_{N-1} computes $\Sigma_{i=1}^{N} y_i$ and transmits it to the base station.

- If chain leader at the beginning of chain list:
  - The chain leader passes a token message to the last node of the chain c_N .
  - Node c_N computes measurement y_N and transmits the measurement y_N to c_{N-1}.
  - Node c_{N-1} computes the measurement y_{N-1} and transmits $y_N + y_{N-1}$ to c_{N-2}, and so on till chain leader.
  - Then the chain leader node receives data from c_2 computes $\Sigma_{i=N}^{1} y_i$ and transmits it to the base station.

- If chain leader at position $(j : \quad 1 < j < N))$ of chain list:
  - The chain leader passes a token message to the first node of the chain c_1 and to the last node of the chain c_N.
  - Node c_1 computes measurement y_1 and transmits the measurement y_1 to c_2.
  - Node c_2 computes the measurement y_2 and transmits $y_1 + y_2$ to c_3, and so on till chain leader at position j.
  - At the same time Node c_N computes measurement y_N and transmits the measurement y_N to c_{N-1}.
  - Node c_{N-1} computes the measurement y_{N-1} and transmits $y_N + y_{N-1}$ to c_{N-2}, and so on till chain leader at position j.
  - Then the chain leader node receives data from c_{j-1} and c_{j+1} computes $\Sigma_{j=1}^{N} y_j$ and transmits it to the base station.

The base station receives the compressed data from the chain leader, then recovers the original data from the compressed data using a CS recovery algorithm as shown in the next phase.

4.3 Reconstruction Phase

In densely deployed sensor networks, sensors have spatial correlations in their readings. Let N sensor readings form a vector ($d = [d_1, d_2, \ldots, d_N]^T$), then d is a K-sparse signal in a particular domain Ψ. Denote $\Psi = [\Psi_1|\Psi_2|\Psi_3|\ldots|\Psi_N]$ as the representational basis with vectors $\{\Psi_i\}_{i=1}^N$ as columns, and $X = [x_1, x_2, ..., x_N]^T$ are the corresponding coefficients.

Then d can be represented in the Ψ domain as:

$$d = \sum_{i=1}^{N} x_i \Psi_i \qquad or \quad d = \Psi X \tag{3}$$

Where, the Ψ matrix describes the correlation pattern among sensor readings. It is utilized only in data recovery process, and is not required to be known to sensors. According to equation (1), the base station is able to reconstruct sensor readings through solving an L_1-minimization problem:

$$min_{x \in R^N} \parallel X \parallel_{l1} \quad s.t \quad y = \Phi d, d = \Psi X \tag{4}$$

In particular, Orthogonal Matching Pursuit for Signal Recovery (OMP) algorithm [15] can be used by the base station to solve the above L_1-minimization problem.

To identify the ideal signal x, we need to determine which columns of α participate in the measurement vector y. The idea behind the algorithm is to pick columns in a greedy fashion. At each iteration the base station chooses the column of α that is most strongly correlated with the remaining part of y. Then we subtract off its contribution to y and iterate on the residual. After M iterations, the algorithm will have identified the correct set of columns.

5 Example Scenario

In this section, we provide a simple example, with step by step explanation of our algorithm. Assumed that, ten sensor nodes ($N = 10$) are randomly deployed in a region of size 100×100 with a base station (BS) located at the center of the network. Where, the nodes $n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_8, n_9$,and n_{10} are located at (65,90), (72,56), (32,69), (37,6), (87,84), (90,68), (41,57), (99,58), (17,15), and (82,84), respectively. Let $d_{N\times1} = (2, 4, 2, 4, 5, 1, 5, 3, 2, 5)^T$ is a data vector in which each data value represents the sensing data at one node. The algorithm goes as follows:-

- **Step One:** Nodes are organized to form a chain. To construct the chain, we use equation 5 to compute the distance (dis) between the nodes and the base station

$$dis[(x, y), BS] = \sqrt{(50 - x)^2 + (50 - y)^2} \tag{5}$$

we will have

$dis[n_1, BS] = 42.72$, $dis[n_2, BS] = 22.80$, $dis[n_3, BS] = 26.17$, $dis[n_4, BS] = 45.88$, $dis[n_5, BS] = 50.42$, $dis[n_6, BS] = 43.86$, $dis[n_7, BS] = 11.40$, $dis[n_8, BS] = 49.64$, $dis[n_9, BS] = 48.10$, and $dis[n_{10}, BS] = 46.96$

Therefore, n_5 will be the first node added to the chain list (as shown in table 1), since it is the furthest node to the BS.

n_5									

Table 1: Chain List contains one node

- **Step Two:**

The chain will be updated with the closest node to the last added one (n_5)by using equation 5 we will have

$dis[n_5, n_1] = 22.80$, $dis[n_5, n_2] = 31.76$, $dis[n_5, n_3] = 57.00$, $dis[n_5, n_4] = 92.64$, $dis[n_5, n_6] = 16.27$,

$dis[n_5, n_7] = 53.33$, $dis[n_5, n_8] = 28.63$, $dis[n_5, n_9] = 98.29$, and $dis[n_5, n_{10}] = 5$

Therefore, n_{10} will be the second node added to the chain (as shown in table 2).

Repeating the previous processes till all the nodes added to the chain list(as shown in table 3).

- **Step Three:**

Let the measurement matrix

$$\Phi_{M\times N} = \left(\begin{array}{cccc} -0.629 & -0.509 & -0.217 & 1.099 \\ -0.02 & -0.0409 & 0.691 & -0.250 \\ 0.0891 & -0.108 & 0.258 & -0.082 \\ -0.225 & -0.251 & -0.478 & 0.434 \\ -0.408 & -0.572 & & \\ 1.232 & 0.405 & & \end{array} \right)$$

, where $M = 2$, $N = 10$.

n_5	n_{10}								

Table 2: Chain List contains two nodes

n_5	n_{10}	n_1	n_6	n_8	n_2	n_7	n_3	n_9	n_4

Table 3: Chain List contains all the nodes

Each node in the chain compute its measurement as follows :

To compute the measurement y_1 for n_1 we use the following equation

$$y_1 = \alpha_1 \;\; d_1 \tag{6}$$

where $d_1 = 2$ and $\alpha_1 = \begin{pmatrix} \Phi_{11} \\ \Phi_{12} \end{pmatrix} = \begin{pmatrix} -0.62 \\ 0.028 \end{pmatrix}$ then $y_1 = \begin{pmatrix} -1.259 \\ -0.045 \end{pmatrix}$.

Similarly,

$y_2 = \begin{pmatrix} -2.037 \\ -0.163 \end{pmatrix}$, $y_3 = \begin{pmatrix} -0.435 \\ 1.383 \end{pmatrix}$, $y_4 = \begin{pmatrix} 4.398 \\ -1.003 \end{pmatrix}$, $y_5 = \begin{pmatrix} 0.448 \\ -1.129 \end{pmatrix}$, $y_6 = \begin{pmatrix} -0.108 \\ -0.251 \end{pmatrix}$, $y_7 = \begin{pmatrix} 1.293 \\ -2.393 \end{pmatrix}$, $y_8 = \begin{pmatrix} -0.246 \\ 1.303 \end{pmatrix}$, $y_9 = \begin{pmatrix} -0.817 \\ 2.465 \end{pmatrix}$, and $y_{10} = \begin{pmatrix} -2.861 \\ 2.029 \end{pmatrix}$ for n_2, n_3, n_4, n_5, n_6, n_7, n_8, n_9, and n_{10}, respectively.

- **Step Four:**

  Let n_2 be the chain leader node. Then n_2 sends a token message to n_5 and n_4 to start the data gathering process in which: n_5 send y_5 to n_{10}, then compute $y_{10} + y_5$ and send this value to n_1.

  Node n_1 add the received value to its sample and compute $y_{10} + y_5 + y_1$ then send this value to n_6. Node n_6 send the value $y_1 + y_{10} + y_5 + y_6$ to n_8, node n_8 sent the value $y_{10} + y_5 + y_1 + y_8 + y_6$ to the chain leader node n_2.

  Similarly, n_4 send y_4 to n_9 which add its sample y_9 to y_4, compute $y_4 + y_9$ and transmit the result to n_3.Then n_3 send $y_4 + y_9 + y_3$ to n_7 ,which send $y_4 + y_9 + y_3 + y_7$ to the chain leader node n_2.

  Finally, the chain leader node n_2 add its sample value y_2 to the received values from n_8 and n_7 , and transmit $y = \Sigma_{i=1}^{N} y_i = \begin{pmatrix} -20.77 \\ -0.752 \end{pmatrix}$, which containing the weighted sum of all readings in the chain to the base station. Then the base station will use a reconstruction algorithm to recover d from only ($M = 2$) measurements y.

6 Simulation Results

The network has been simulated in MATLAB, with statistical data gathered to analyze the performance of of the proposed technique. In our simulation,the sensor nodes are randomly deployed in a region of size 100 m $\times$ 100 m, and the number of deployed sensor nodes varies from 25 to 100 nodes in the increments of 25 nodes with the base station at location (x=50, y=50).

In order to measure the energy consumption of sensor nodes, we use the same energy parameters and the radio model as discussed in [4], wherein the energy consumption is mainly divided into two parts: receiving and transmitting messages. The transmission energy consumption needs additional energy to amplify the signal depending on the distance to the destination. Thus, to transmit a l-bit message a distance d, the radio power consumption will be,

$$E_{Tx}(l,d) = \begin{cases} l\;E_{elec} + l\;\epsilon_{fs}\;d^2 & d < d_0 \\ l\;E_{elec} + l\;\epsilon_{mp}\;d^4 & d \geq d_0 \end{cases} \tag{7}$$

and to receive this message, the radio expends will be

$$E_{Rx}(l) = l\;E_{elec} \tag{8}$$

Simulated model parameters are set as: $E_{elec} = 50nJ/bit$, $\epsilon_{fs} = 10pJ/bit/m^2$, $\epsilon_{mp} = \frac{13}{10000}pJ/bit/m^4$,$d_0 = \sqrt{\epsilon_{fs}/\epsilon_{mp}}$, and the initial energy per node= $2J$.

6.1 Performance Metrics

The performance of ECST technique is compared with PEGASIS [6, 11] and PEGASIS with conventional compression (we adapt Huffman coding [9] as conventional compression). The performance is evaluated mainly, according to the following metrics.

- **Energy Consumption:** It is the average energy consumed by all the nodes in sending, receiving and forwarding operations. The average energy consumption per round can be estimated as:

  $$E = \frac{\sum_{i=1}^{N} E_i(r)}{r} \tag{9}$$

  Where N is the number of sensor nodes in the considered WSN, and r is the number of rounds.

- **Network lifetime:** It is the time interval from the start of operation (of the sensor network) until the death of the first alive node (or until the death of half of the nodes).

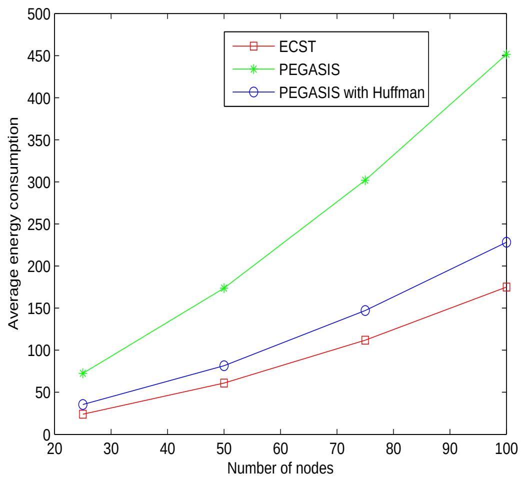

Figure 2: Average consumed energy per round as a function of number of sensor nodes

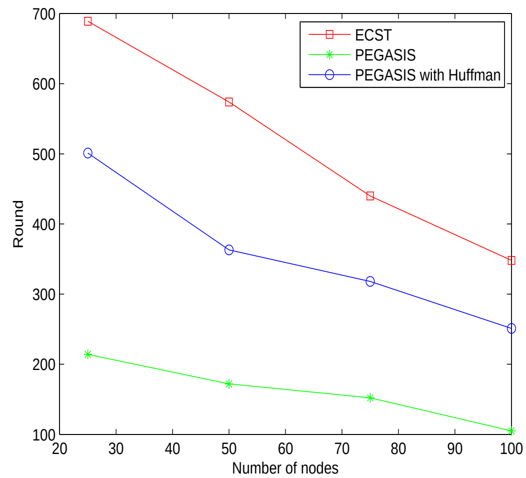

Figure 3: Network life time as a function of number of sensor nodes

	Number of Nodes			
	25	50	75	100
ECST	896	757	647	570
PEGASIS	340	298	276	248
PEGASIS with Huffman	692	664	588	477

Table 4: Comparison of Network lifetime with respect to Half Node Die

Fig. 2, shows the average energy consumption per round. In fig 2,the number of sensor nodes are varied from 25 to 100 in increment of 25 sensor nodes. It shows that, our approach has less energy consumption compared with standard PEGASIS and PEGASIS with Huffman coding.

Fig.3, shows the comparison of ECST, PEGASIS, and PEGASIS with Huffman code in terms of network lifetime (first node die). From fig. 3, it can be observed that ECST clearly improves network lifetime over other techniques. Moreover, fig. 3 illustrates the effectiveness of ECST in prolonging network lifetime than its counterparts. ECST offers improvements in network lifetime by factor of 70%,30% over PEGASIS, and PEGASIS with Huffman code, respectively.

Table 4, shows the comparison of network lifetime of ECST, PEGASIS and PEGASIS with Huffman coding with respect to half node die. It can be observed that the half node dies in ECST happens in the 896th round,in PEGASIS it appears in 340th round and in PEGASIS with Huffman, it comes in 692th round. ECST is 23% better than PEGASIS with Huffman and 63%better than PEGASIS with respect to Half node die for network size of 25 nodes.

In summary, our simulation results show that our CS-based technique minimizes the overall energy consumption and therefore, extends the lifetime of the WSN. This is due to the fact that if the original data are compressed by CS-based technique, each sensor node produces much smaller traffic volume which can be transmitted to the base station at a much lower transmission power and with a smaller time delay. Moreover, only the joint recovery at the base station is needed. Thus, no intermediate stages are required to gather all of the data at a single location and carry out compression. In contrast to the PEGASIS with the conventional compression, it needs to exploits correlated data for compression in order to made joint compression, which cause a transmission delay and spend a lot of energy.

7 Conclusion

In this paper, we have proposed an adaptive and efficient compressive sensing based technique for improving the performance of routing in wireless sensor network. The proposed technique achieves both minimum energy consumption and increase network lifetime. Experimental results demonstrate that the proposed ECST outperformed PEGASIS and PEGASIS with Huffman coding in terms of the network life time and energy consumption. As part of our future work, we are planning to investigate how to optimize ECST to work with mobile sensor network.

References

[1] A. M. Khedr and Walid Osamy, "Effective Target Tracking Mechanism in a Self-Organizing Wireless Sensor Network," Journal of Parallel an Distributed Computing, Vol. 71, pp. 1318-1326, 2011.

[2] Hande Alemdar, Cem Ersoy, Wireless sensor networks for healthcare: A survey, Computer Networks, Volume 54, Issue 15, 28 October 2010, Pages 2688-2710, ISSN 1389-1286, 10.1016/j.comnet.2010.05.003.

[3] Antoniou, M.; Boon, M.C.; Green, P.N.; Green, P.R.; York, T.A.; , "Wireless sensor networks for industrial processes," Sensors Applications Symposium, 2009. SAS 2009. IEEE , vol., no., pp.13-18, 17-19 Feb. 2009, doi: 10.1109/SAS.2009.4801768.

[4] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy Efficient Communication Protocol for Wireless Microsensor Networks", in Proceedings of the 33rd Annual Hawaii International Conference, 2000, Vol. 2, pp. 3005- 3014.

[5] K. Kwon, "Efficient Clustering Protocol in the Large-Scale WSN", MS thesis, Electrical and Electronic Engineering, chsei, Korea, 2009.

[6] S. Lindsey and C. Raghavendra, "PEGASIS: Power Efficient gathering in sensor information systems",in Aerospace Conference Proceedings, 2002, Vol. 3, pp.1125 - 1130.

[7] J. Chou, D. Petrovic, and K. Ramchandran," A distributed and adaptive signal processing approach toreducing energy consumption in sensor networks", In Proc. of IEEE Infocom, pages 1054-1062, Mar. 2003.

[8] D. Donoho," Compressed sensing", IEEE Trans. Inform. Theory, 52(4):1289-1306, Apr. 2006.

[9] M. Nelson and J. L. Gailly, "The data compression book", MIS Press, 1996.

[10] D. Slepian and J. K,"Wolf. Noiseless encoding of correlated information sources". 19:471-480, Jul. 1973.

[11] Samia A. Ali and Shreen K. Refaay,"Chain-Chain Based Routing Protocol", in IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 2, May 2011 ISSN (Online): 1694-0814.

[12] Chong Luo and Feng Wu, "Compressive Data Gathering for Large-Scale Wireless Sensor Networks", Proceedings of the 15th annual international conference on Mobile computing and networking, Beijing, China, 2009.

[13] Meenu M.Techand Vandana A.P,"Modified PEGASIS in WSN to increase Network Life Time", in International Journal of Computer Applications (0975 - 8887) Volume 52- No.19, August 2012.

[14] K. Choi, J. Wang, L. Zhu, T. S. Suh, S. Boyd and L. Xing, "Compressed sensing based cone-beam computed tomography reconstruction with a first-order method," in Med. Phys., vol. 37, pp. 5113-5125, Sep, 2010.

[15] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," IEEE Trans. Inform. Theor., vol. 53, no. 12, pp. 4655-4666, Dec. 2007.

[16] Jonathan Gana Kolo , S. Anandan Shanmugam , David Wee Gin Lim , Li-Minn Ang ,and Kah Phooi Seng ," An Adaptive Lossless Data Compression Scheme for Wireless Sensor Networks", Journal of Sensors Volume 2012, Article ID 539638, 20 pages doi:10.1155/2012/539638

[17] Luis Javier Garca Villalba, Ana Lucila Sandoval Orozco, Alicia Trivio Cabrera, and Cludia Jacy Barenco Abbas, "Routing Protocol in Wireless Sensor Networks", Sensors 2009, vol. 9, pp. 8399- 8421.

[18] E. Zanaj, M. Baldi, and F. Chiaraluce, "Efficiency of the Gossip Algorithm for Wireless Sensor Networks", In Proceedings of the 15th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split-Dubrovnik, Croatia, September, 2007.

[19] Ouadoudi Zytoune, Youssef Fakhri and Driss Aboutajdine: A Balanced Cost Cluster-Heads Selection Algorithm for Wireless Sensor Networks, International Journal of Computer Science ,vol 4,no.1, 2009,pp.21-24

[20] Hu Junping, Jin Yuhui, and Dou Liang: A Time-based Cluster-Head Selection Algorithm for LEACH, In proceeding of IEEE Symposium on Computers and Communications 2008 (ISCC 2008), July 6 - 9, 2008,Marrakech,Morocco.

[21] Fan Xiangning; Song Yulin; , "Improvement on LEACH Protocol of Wireless Sensor Network," Sensor Technologies and Applications, 2007. SensorComm 2007. International Conference on , vol., no., pp.260-264, 14-20 Oct. 2007 doi: 10.1109/SENSORCOMM.2007.4394931.

[22] Zhiyong, P. and L. Xiaojuan. The improvement and simulation of LEACH protocol for WSNs, Software Engineering and Service Sciences (ICSESS), 2010 IEEE International Conference,2010.

[23] O. Zytoune, M. El aroussi, M. Rziza, D. Aboutajdine: Stochastic Low Energy Adaptive Clustering Hierarchy, ICGST-CNIR, Volume (8), Issue (1), (2008) pp.47-51.