IJCTA, 9(13) 2016, pp. 6009-6016

Effective Algorithm for Data Gathering with Compressive Sensing in Wireless Sensor Networks

S. Kayalvizhi[1], Nikhil S. Hage[2] and S. Malarvizhi[3]

ABSTRACT

Data gathering is one of the most important functions provided by WSNs. As we know the WSNs networks are used to check the surrounding condition and monitor those without any direct human interface and without any failure of this network. Normally, all other data gathering algorithm of WSNs network will lead to loss of high energy as they have to follow the total path, and this directly affect the communication cost of the networks. In this paper; we are going to use random walk algorithm for data gathering in WSNs. Random walk algorithm nearly reduced all energy and cost constrain in WSN networks efficiently and give a non-uniform measurement. We describe compressive sensing mathematically and form random walk path detection. We obtain the random matrix which we can use for recovery of the signal in the compressive sensing and also this matrix will give connection of nodes by forming expander graph form the matrix and for reconstructing the original signal with the use of compressive sensing technique. In this paper, we discuss about sparctiy and its effect on path selection for networks and also find the probability for random walk theory. We also show simulation for our scheme. Our simulation results show that probability for our algorithm will be high as compare to normally used an algorithm with less path of connection between nodes. This leads to the reduction in communication cost for the network. Also result shows low recovery error with compressive sensing.

Key Words: Random walk, Compressive sensing, l1-minimization, sparcity, Sparse signals, Data gathering.

1. INTRODUCTION

[1]The use of wireless sensor networks (WSNs) has been increase because of its advantage and application benefits like any type of surveillance. The main future application of Wireless sensor network will be Internet of things (IoT'S) based application which will make life much easier.

On the other side, compressive sensing is one of the most used techniques in the field of signal and image processing. As per the conventional approach i.e., sampling signals: the sampling rate must be at least the twice the maximum frequency present in the signal so-called Nyquist rate.

$$R>=2F$$

For some signals such as a signal that are not actually band limited, the Shannon theorem is not applicable as they won't work at that sampling rate. In this conditions compressive sensing work efficiently. As in sampling theorem, where we have to consider the entire sampling signal for recovery, this will be not necessary in the case of compressive sensing. Only a few sample signals do this work. Another disadvantage of sampling theorem is that few missing signals lead to the wrong recovery. This disadvantage will overcome in the case of compressive sensing technique which will work effectively with only few, or little-sampled signal which are may be less than sampling frequency also. So in the case of low or very weak range of

1 Assistant Professor, *E-mail: kayalvizhi.s@ktr.srmuniv.ac.in*,

2 PG-Student, M-Tech, SRM University, Chennai-603203, India, *E-mail: nikhil0724@gmail.com*, [3]Professor, Department of Electronics and Communication Engineering, SRM University, Chennai-603203, India.

signals this will be advantageous.

In [2]L. Lovasz *et al.* investigated the problems of graph theory and random walk. In his survey, explained the basic concept and important concept about the probability of random walk and random walk based on dimension of the networks and establishing the probability from the graph. Similarly, in [3]Peter. G. Doyle et al. Also gives information about basic concepts of random walk. Author has given network solution for the model of graph G(n,p) in which an edge exist between two nodes with probability *p* in the network with node *n* and *p* depend on the dimension of the network. Here we will investigate the application of CS with *random walk* approach for collecting the data in Wireless Sensor Networks. [1]Compressive sensing will applicable to the non-uniform selection of measurement and data, different than uniform sampling in the traditional compressive sensing (CS) theory and hence for this CS will be appropriate and this will again show that this approach can be used to recover sparse signals in WSN scenario.

2. BASIC CONCEPTS

2.1. Compressive sensing basics

[4]According to the compressive sensing technique we can reconstruct the sparse or compressible input signal with high probability from far smaller measurement, than the length of the input signal used to produce output.

Let's consider an n-dimensional signal vector $x\ (x_1 \ldots . . xn)^T$) This vector x is perfectly k-sparse if it has at most $k(k<<n)$ nonzero entries. Further, suppose that x can be represented as $X = \Sigma_{i=1}^{n} \theta i \Psi i$ in some domain Ψ = (Ψ1.......Ψn). Generally, we say that the vector x is also k-sparse (in domain Ψ) if there are at most k nonzero entries in the vector $\theta = (\theta 1 \ldots \theta n)^T$. The theory of CS states that the k-sparse signal x can be recovered from m (m< n) linear combinations of measurements with high probability, which can be obtained through an m x n measurement matrix A, i.e., y = Ax. Each element of y is called a projection. Cand»s *et al.* have shown that recovering the signal x from y can be solved with an *l*1-minimization problem:

$$\min_{x \varepsilon \Re \wedge N} \| x \| l1 \quad s.t \quad y = Ax, \tag{1}$$

Or

$$\min_{x \varepsilon \Re \wedge N} \| \theta \| l1 \quad s.t \quad y = A\Psi\theta,\ x = \Psi\theta \tag{2}$$

The above *l1*-minimaization problem can be solved using linear programming techniques. The recovery of the input signal will be possible with this technique [4].

2.2. Connection of nodes

As simply for random walk algorithm, the path detection is important to find an actual connection of all connected nodes in the networks. These we can do with the use of pictorial graph called of expander graph .this graph gives us random matrix and vice versa. For example, we can take the random matrix which we will get from our simulation for the path detection in following section and we can say this matrix as matrix A. Shown follows.

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Based on this measurement matrix or sensing matrix we can plot the expander graph. As shown in following figure [1]

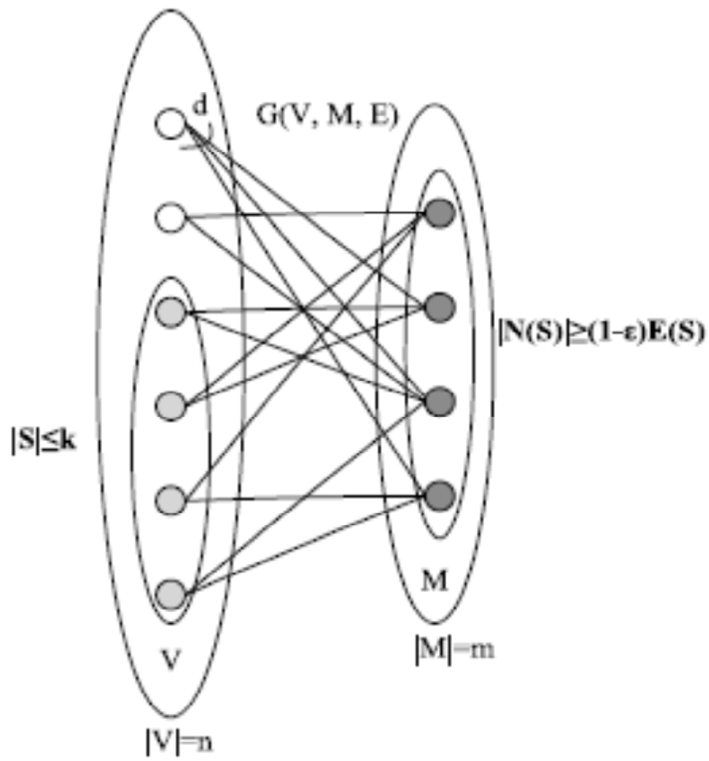

Figure 1: Illustration of expander graph Corresponding to the measurement matrix 'A'[5].

From matrix, we can observe, how we got it from the graph and we can do this process vice versa. Another way is to find a path by simulation which we are going to do. This matrix will give us important factor about the random walk that is the probability of random walk algorithm.

2.3. Theory on Random Walk

[3]From random walk and electric networks, Let's G = (V, E) be a connected graph with n nodes and m edges. If we used random walk for connecting different nodes corresponding to its edges, we would find the following probability

$$Pt(i) = Prob(vt = i):$$

So

$$pij = 1/d(i) \quad ; \text{if } ij \, \varepsilon \, E,$$
$$= 0 \quad ; \text{otherwise}.$$

The random walk and electric networks will give us detail information about random walk in one, two dimension in finite networks

3. PROBLEM STATEMENT

For formation of the network we consider 'm' sensor nodes for initializing 'm' independent Random Walk. As in simulation, we will do the same process we will going to follow as here describe. We simply follow the linear addition of information node to its neighbour node and so on, As explain bellow:

Step (1): at the beginning, '*i*' nodes let's say n*i* which is uniform and randomly selected. This will be our sink node.

Step (2): after selecting our first node we will use the standard random walk algorithm to connect next node. After detecting the next node, it will connect to that node and at same time it will add the information

nj(1)= ni(0) + nj(0)

At the same time, the node v*i* decrements the length t.

Step (3): Repeat the above steps up to t =0 i.e. length become 0 and at the end the final constructed signal equation will be:

np(t') = nk(t'-1) + np(t'-1).

For y last node and x previous node in the random walk. This way we can form the overall network, that final signal will collect by sink and after getting all information matrix, we will use compressive sensing for signal compression purpose.

4. NUMERICAL SIMULATION

[6,7,8]In this section, we will show the probability of random walk. For basic concept we will take help of Lemma 1:

Lemma 1. *Let Bt(v) be the event that the random walk W starting at u visits v by time t(t >=T). Then the probability* ***Pr*** *(Bt(v)) will be between* $\frac{(1-\mu)t}{2(\eta+2)cn}$ *and* $\frac{(1+\mu)}{cn}$ *where* $\eta = \frac{1}{(1-\mu)k}$ *and c is a constant*

The above lemma shows that the upper bound and the lower bound of the probability that a random walk will visit their nodes.

This lemma will give us an idea about the probability. We will find the overall performance for our proposed technique through simulation. We use MATLAB tools for performing simulation we consider 'n' nodes which are connected randomly. We took 100 nodes with in the square area and perform random walk with the sparse signal.

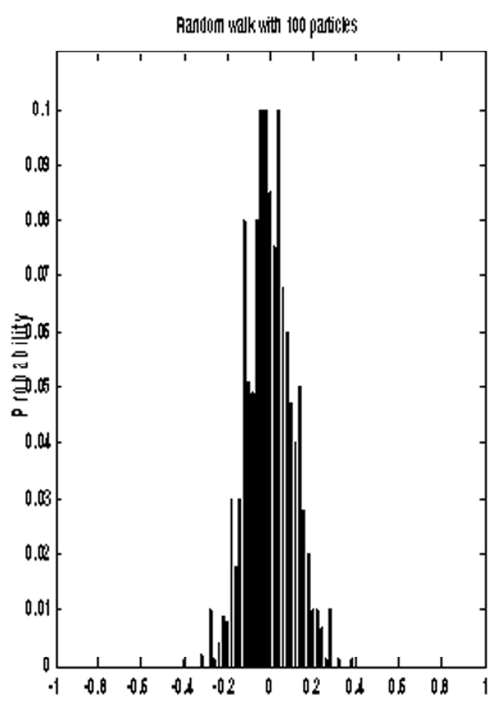

Figure 2: Probability of the random walk for 100 nodes covers 1 unit square area

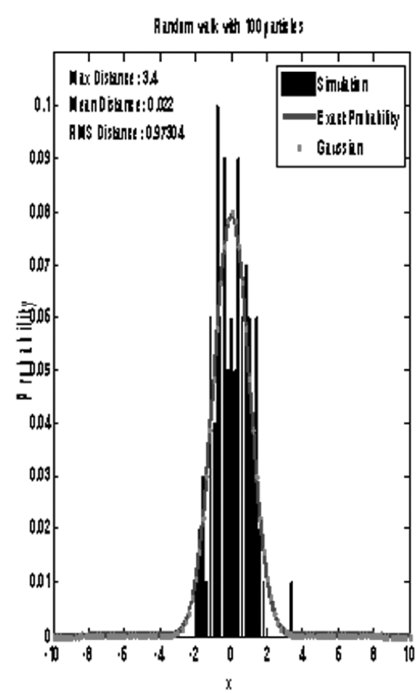

Figure 3: Exact probability with both Gaussian and binomial distributions

In simulation, we specify the length of 'T' which will be time step and 'N' total number of nodes used along with this value we specify the length of space step and length of time step for plotting the graph by following the random walk flowchart approach. We will able to find the exact probability of random walk. Both the Gaussian and binomial function used to find the exact probability of random walk.

(A) Random walk simulation in two dimension area

Here we will simulate the path finding in random walk algorithm. The position coordinates in 2- dimensions will specify while simulation and this dimension will be considered as the dimension of the starting node. As per the algorithm function we stated in section Iv we must add the node information to the next node to which it is connecting and goes increasing with the connection to the nodes in WSNs networks. We assume our node start at the co-ordinates of (1,1). The path will have four directions to go with coordinates (1,1)as +x-direction, (1,0) as –x-direction, (0,1) as +y- direction and last (0,0) as –y-direction . This can be changed as per the understanding of the application and its use. We are specifying the separate matrix to store the information of the node as it goes finding the respective nodes. We are taking N=100 nodes and specifying the sparcity k=20. For selecting the neighbouring node, we will check the Euclidian distance between those nodes. For our simulation we take this distance less than 0.5.

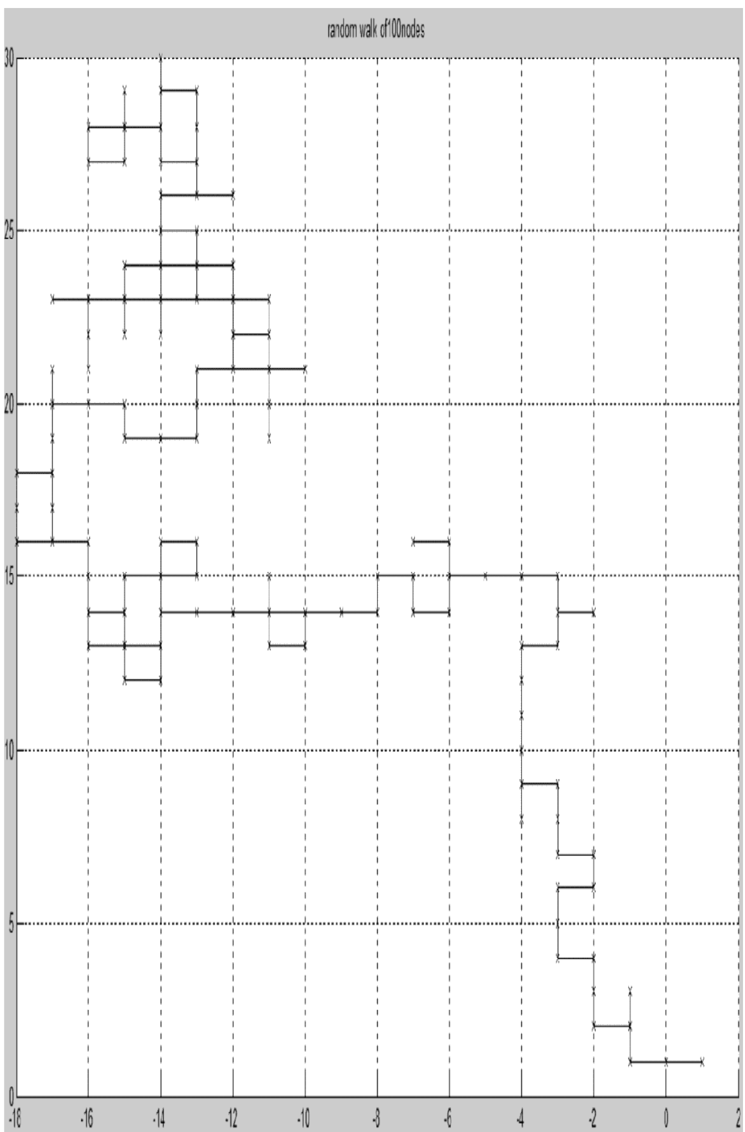

Figure 4: Random walks the path with 100 nodes and having 0.5 distances to connect next nodes.

(B) Number of Random Walks depend on sparcity (K)

In this section, we have proved that if we take O(*K log(n/k))*)independent random walks with length in *t=O(n/k)* a random geometric network, we can recover k-sparse signals from m random projections generated from random walks. To recover k-sparse signals, the approached which employ dense Gaussian or Bernoulli random matrices need to take m *=O(k log(n/k))* random projections with each other from a linear combination of O(*n*) measurements. In addition, the scheme using uniform sparse random matrices such as expander based compressive sensing need to take *m=O(k log(n/k)))* random projections with each uniformly and randomly selecting *O(n/k)* nodes. However, such a scheme needs a precise routing to generate random projections. Therefore, our scheme has the significant advantage over the conventional approaches regarding communication cost. On the other hand, it should be noted that the length of each random walk t should be larger than the mixing time T. This is because a vertex can be visited by a random walk with a certain probability when the path reaches its stationary distribution. It has been shown that the mixing time of a random geometric graph is *O(n/logn)*. Hence, it implies that the length of each random walk t should be larger than O(n*/log n*) even though k *>=O(log n)*. It is also interesting to

We can also plot the theoretical curve for a number of the random walk as *m=2klog(n/k)*. Form graph we can say that the both values for a number of the random walks are same within the range of difference. This shows that the number of steps that random walk need to take for successful recovery is much less than the actual node n. Here we take 'n' values i.e. the number of random walk for n= 500, 1000 and 2000 respectively for simulation.

(C) Performance of compressive sensing

[9,10]As fig 6 shown, the path for random walk algorithm, it will give us the random information matrix 'A' which is our randomly generated matrix for the compressive sensing and we can recover the original signal with the use of this matrix and original signal matrix. We assume that the signal matrix has 256 elements,

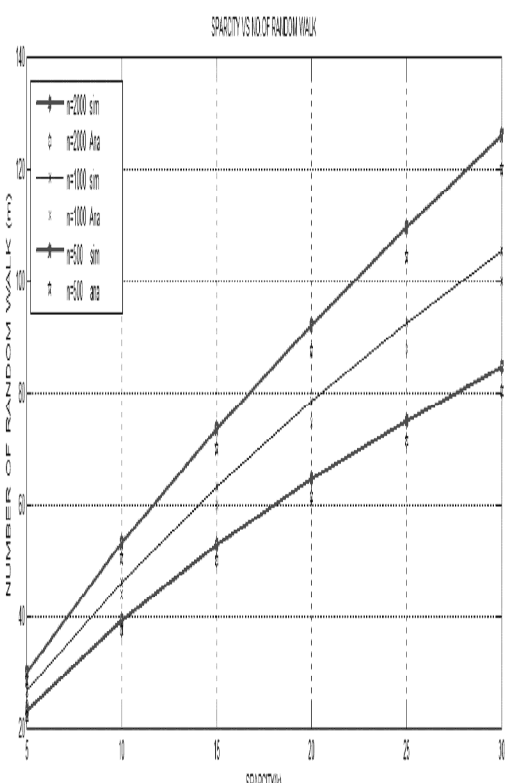

Figure 5: Sparcity K versus numbers of random walks in random walk algorithm

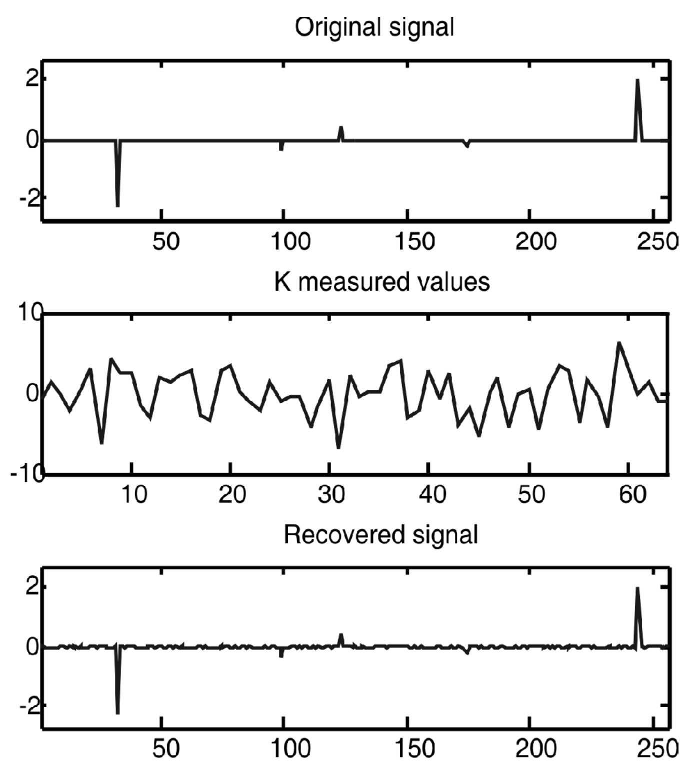

Figure 6: Compressive sensing for input signal with mixing with random k value and its recovery with l1 minimization theorem

and the random matrix has only 64 elements. With the use of compressive sensing, acquisition of the signal can be easily done and we can recover the original signal by using the l1 minimization theorem fig 9 shows that the only 64 no. of the signal are sufficient to recover the signal with 256 elements. This gives us the compression ratio of 75% which is more than Sufficient for our recovery for an input signal or information signal.

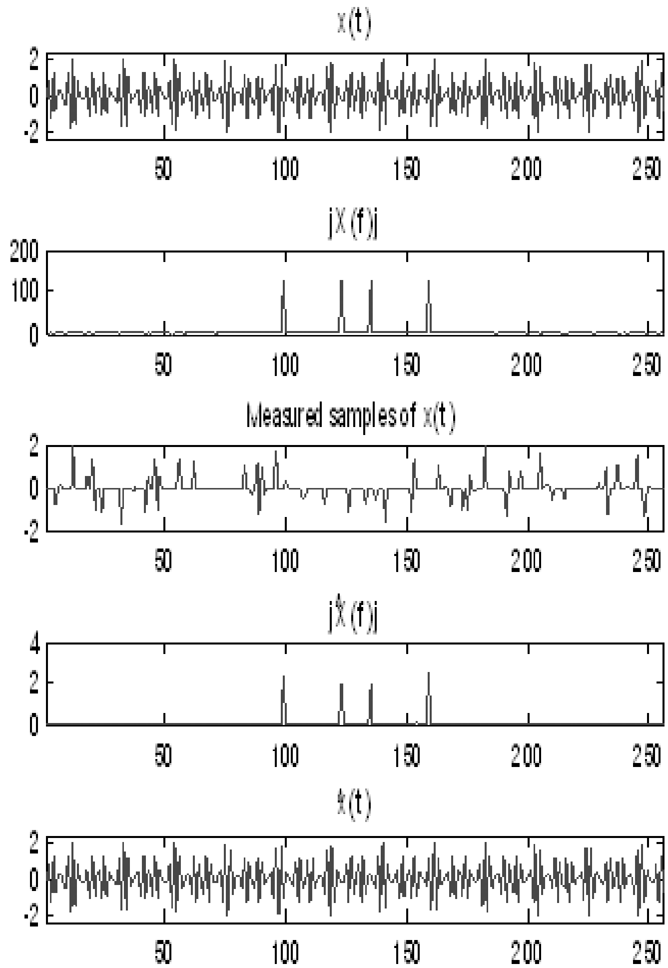

Figure 7: Compressive sensing in the frequency domain for the same input but with FFT and IFFT function.

5. CONCLUSION

This result shows the after considering all the problems in WSNs network our proposed algorithm is working properly. While doing simulation we consider those problems discuss above and we consider this all constrain while implementing random walk algorithm for an Actual formation of the network. The figure [3] shown exact probability up to 90% which will be well as consider to another algorithm gives information about the network form by random walk algorithm will be good as this leads to low energy requirement which solves the major problem in WSNs networks. The RMS values between nodes also shown and mean value to give extra information about the probability of random walk algorithm.

The simulation of path selection in figure [4] give us the sensing matrix as discuss earlier which will be important to find the expander graph to show the connection of nodes. Along with this, simulation also shows the connection for 100 nodes leads to an actual number of the random walk for our algorithm which is considerably 1/4th of total nodes. This gives the better result with less connection in networks which is an advantage as compare to another algorithm for WSNs networks.

The sparcity (K) factor also plays an important role in network formation and its cost. The result shows that, readings are bounded with simulation readings and analytical readings which is also an advantage for our algorithm as all the result satisfying and improving the quality of the network. This directly leads to reduction of communication cost. Result of the compressive sensing also for both time domain and frequency domain input and showing the recovery as we getting exactly same signal as input which will show the network is lossless and gives low recovery error as for our simulation it is almost null and low for increment of signal number.

REFERENCES

[1] Nikhil S.Hage "Efficient Data with Compressive Sensing in Wireless Sensor" Volume 3 Issue 1, January 2016.

[2] L. LOVÁSZ, Paul ErdÓs "Random Walks on Graphs: A Survey"Eighty (Volume 2) Keszthely (Hungary), 1993, pp. 1–46.

[3] Peter G. Doyle J. Laurie Snell "*Random walks and electric networks*" Version dated 5 July 2006.

[4] E. Candès and J. Romberg "*Sparsity and incoherence in compressive sampling*," Inverse Prob., vol. 23, no. 3, pp. 969–985, 2007.

[5] Emmanuel J. Candès and Michael B. Wakin "Introduction to compressive sensing" by, IEEE Trans. Inf. Theory, vol. 52, no. 12, pp. 5406-5425, Dec. 2006.

[6] Xuan gou Wu, Yan Xiong, MingxiLiy "*Distributed Spatial-Temporal Compressive Data Gathering for Large-scale WSNs*" School of Computer Science and Technology, University of Science and Technology of China, Hefei, China State Key Laboratories of Transducer, Technology, Shanghai, Email:wxgou@mail.ustc.edu.cn, yxiongs@ustc.edu.cn, keeper@gmail.com, wchuang@ustc.edu.co, *IEEE, 2013*.

[7] Xiaohan Yu and Seung Jun Baek "*Compressive data aggregation in wireless sensor networks using sub-Gaussian random matrices*" College of Information and Communication, Korea University Anamdong, Seongbukgu, Seoul, Korea, 136713 E-mail:*{yuxiaohan188,sjbaek}@korea.ac.kr,IEEE, 2013*.

[8] Haifeng Zheng, Feng Yang, Xiaohua Tian, Member, IEEE, Xiaoying Gan, Xinbing Wang, Senior Member, IEEE, and Shilin Xiao, Member, IEEE "*Data Gathering with Compressive Sensing in Wireless Sensor Networks*", Jan-2015.

[9] Zhuang Xiaoyan "*Wireless Sensor Networks based on Compressed Sensing*" School of Automation Engineering University of Electronic Science and Technology of China Chengdu, China, zhuangxyan@gmail.com, *IEEE, 2010*.

[10] Riccardo Masiero, Giorgio Quer, Michele Rossi and Michele Zorzi "*A Bayesian Analysis of Compressive Sensing Data Recovery in Wireless Sensor Networks*" Department of Information Engineering, University of Padova, 6/B35131, Padova, Italy, E-mail:*{riccardo.masiero,giorgio.quer,rossi,zorzi}@dei.unipd.it, IEEE, 2009*.