International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 11 (2018) pp. 8913-8921

An Optimized Approach of Routing Enhancement Using Hybrid ABC and Cuckoo Search and Analysed by FFNN

Mandeep Kaur[1] Balwinder Singh Sohi[2]

[1]Research Scholar, IK Gujral Punjab Technical University, Jalandhar, Punjab, India.
[2]Director, Chandigarh Group of Colleges, Mohali, Punjab, India.

Abstract

A wireless sensor network consists of n numbers of node and the nodes are powered by small cells which cannot be recharged once nodes have been deployed into the network. In such a case, it becomes very important to prevent battery life from any misuse. There are several issues due to which a path can be distorted like any mis-event in the network or any distortion in the path. A network may also bear some intrusion (insider or outsider). A path failure may lead to a high consumption of energy when the network faces any distortion, there must be prevention or alternate way to find another path. The nodes are grouped in a way so that each group has a cluster head. The cluster head communicates with other cluster head and the entire packets are dropped to a sink. This paper presents a path optimization technique in wireless sensor network using cuckoo search and artificial bee colony algorithm.

Keywords: Wireless Sensor Network, Path Optimization, Cuckoo Search, Artificial Bee Colony Algorithm, DAG (Directed Acyclic Graph), Energy Consumption, End to End Delay MATLAB (Matrix Laboratory)

INTRODUCTION

A wireless sensor network [1] is an aligned network which has some height and width and the nodes are deployed randomly in the network as shown in Figure showing the deployment of the nodes in a network with height 'H' and width 'Y'.

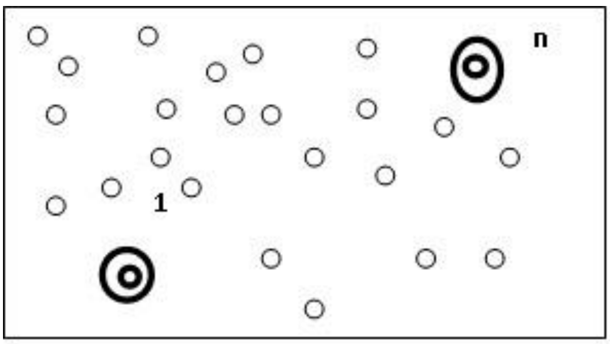

Figure 1. Nodes in the network

A wireless sensor network [2] has N number of nodes which are charged with very tiny or small batteries. A node once put into the network cannot be charged again while the network cannot be charged again while the network is in simulation mode. The main aim of wireless sensor network is to transfer the data from one end to another end. There would be one or more than one source in the network. A source node is one that seeks to transfer the data. A source has to seek for a route as the source cannot directly transfer the data from one end to another [3]. To enhance the data transfer and to reduce the energy consumption, LEACH [4] first introduced the concept of clustering. Each cluster contains a cluster head that has the highest energy out of nodes present in the cluster which node will lie in which cluster that completely depends upon the signal strength which a node receives and transfer. The entire node in a cluster only communicates with the cluster head only. Figure 2 represents the communication of nodes [5].

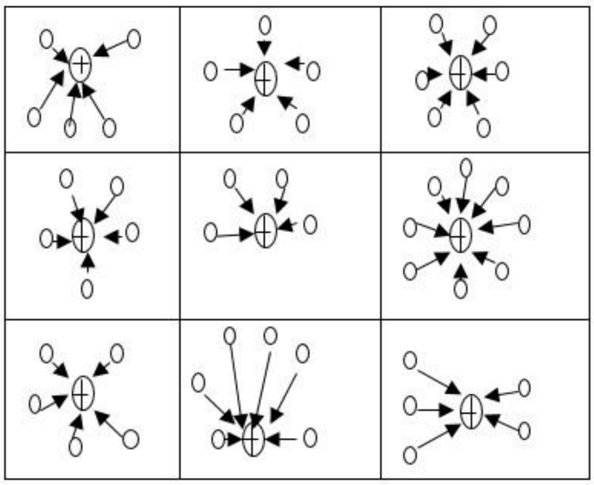

Figure 2. Clustered WSN

Different wireless sensor network clustering algorithms [6, 7] have proposed different types of clustering techniques and algorithms. It is not compulsory to follow any specific clustering algorithm. The selection of clustering algorithms completely depends upon the type of requirement which has to be processed. As for example, some LEACH protocol [8] uses the following architecture.

$$K = 2^{n-1} \quad (1)$$

Where k is the total number of nodes and suffers from delay in route discovery [RD] process. RD refers to finding optimal path from source to destination. A network may face route discovery delay (RDD) problems due to following reasons [9]:

i. Sleep/Awake (SA) status of nodes [10]
ii. Gateway problem [11] [12]

iii. Mis-communication [13]
iv. Overhead in the network [14] [15]

Inadequate computation with communication resources are the only drawbacks in WSN. They have inadequate battery power, partial storage and computation competence, prone for the security attacks and have inadequate bandwidth for communication. To overcome these issues, in this research, a hybridization of ABC (Artificial bee colony) and cuckoo search is considered to optimize the node property and select a better node to transmit the packet data. After that, artificial neural network is used to select a node on the basis of optimized property so it can be said that artificial neural network is taken as a classifier in the proposed work.

The work has dealt with the general model of dynamic acyclic graph which consists of set of processors with dependencies [24]. Each process has an individual execution unit shown as node that can have more than one input as well as output. The nodes can be triggered for execution when the inputs become accessible Output. This model is consisted of set of nodes $(n_1, n_2, n_3 \dots \dots n_n)$ that are integrated with the directed edges set and are shown as (n_i, n_j) where n_i can be termed as the Parent node and n_j ca be termed as the child node. A node that has no parent is considered as the entry node, similarly, a node with the parent is considered as the child node. $w\,(n_i)$ is taken as the weight of the node depicting the completion time of the process.

The model has number of fork as well as joins. The joins has predecessors of the nodes whereas the fork has the successors of the nodes. Consider,

$Q_x = [n_1 n_2, \dots \dots n_n]$ And depicts

$$g(Q_x) = min^{r_k}_{k=1,\dots,n} \,/\, max^{e_{k,x}}_{q=1,\dots,n}$$

The model can be defined in terms of granularity as:

$$g(F) = min_{x=1,\dots w}\{\, g(x)\} \tag{2}$$

The coarse grain is defined below:

$$g(L) \geq 1 \tag{3}$$

In an effective form of static task scheduling, an application consists of number of tasks with the data dependencies among them. It is represented by DAG (Directed acyclic Graph) [20]. Each process is an indivisible unit of execution, expressed by node. A node has one or more inputs and can have one or more output to various nodes. When all inputs are available, the node is triggered to execute. After its execution, it generates its output. In this model, a set of node $(n_1, n_2, n_3 \dots \dots n_n)$ are connected by a set of a directed edges, which are represented by (n_i, n_j) where n_i is called the Parent node and n_j is called the child node. A node without parent is called an Entry node and a node with no child is an Exit node. The weight of a node is denoted by $w\,(n_i)$ which represents the process execution time of a process. Since each edge corresponds to a message transfer from one process to another, therefore, the weight of an edge which is equal to the message transmission time from node n_i *to* n_j. Thus, $w\,(n_i, n_j)$ becomes zero when n_i *and* n_j are scheduled to the same processor because intra processor communication time is negligible compared with the inter processor communication time [25].

The granularity of DAG is

$$f(F) = min_{x=1,\dots w}\{\, f(x)\} \tag{4}$$

The DAG coarse grain is

$$f(L) \geq 1 \tag{5}$$

The grain of the task can be written as

$$F_x = \max\{F(k_x)F(L_x)\} \tag{6}$$

And the DAG's Granularity is

$$F(F) = max_{x=1,\dots v}\{F_x\} \tag{7}$$

The DAG's fine grain is introduced as

$$F(F) \leq 1$$

The problem statement of this research work contains all the above mentioned problems. This paper is organized in five sections. Section first is the introduction part which contains the entire relevant information of wireless sensor network. Section 1 also discusses some major concepts like clustering [16] and RD problems [17] and RDD problems [18]. Section two is the proposed work which briefs the entire procedure of working and optimization. Third section of the paper is the result section. The results are evaluated for the optimized load balancer.

MATERIALS AND METHODS

The proposed methodology is based on the following problem. Following diagram represents a scenario in which the source node would be node 1 and the destination would be node 10. There are three power stations nodes which are accessible to every node that are PN_1, PN_2 and PN_3

It's not the first time that a cyclic graph has been implemented in wireless sensor network [27]. The network model has been taken from [26]. Sukhjit Singh has tried to implement DAG for parallel processing and that has inspired us to implement the same in wireless sensor network.

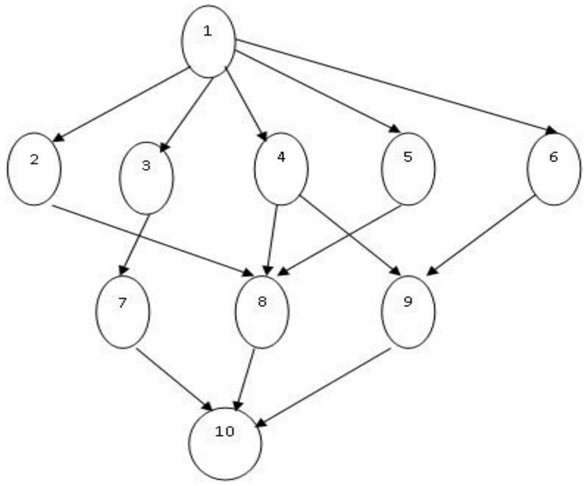

Figure 3. Normal node data communication [25]

The processing cost of every power node is listed. Table 1 illustrates the processing cost of each power node. Hence, each node can only transfer the data through power node as shown in fig. 4.

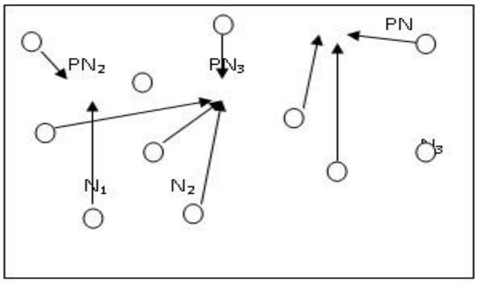

Figure 4. Power node to normal node communication

Table 1: Cost measures

Energy associated with power node	Pre-allocated energy		
	100 Milli Joule	100 Milli Joule	100 Milli Joule
	PN_1	PN_2	PN_3
1	2	4	6
2	4	3	2
3	1	4	2
4	2	1	4
5	3	1	6
6	4	1	3
7	13	11	12
8	5	6	7
9	11	2	6
10	6	1	4

If node will transfer the data through power node 1 then a minimum of 2 milli joules of energy will be consumed. In similar manner, if it is transferred through power nodes2 then 4 milli joules of energy will be consumed.

There are two problems of this paper as described below:

i. Optimize the selection of power node so that least energy is consumed for the transfer of data from source to destination.
ii. Balance the node load in such a manner that energy power node gets equal weight age.

When there is only one node in a LEVEL then it is easy to allocate the power node.

When a node has multiple parent nodes then it becomes difficult to select a power node. As for example node 7 has two parents namely 3 and 4. Node 7 cannot transfer data to node 10 until it doesn't get information from its parents.

The following formulations have been used to evaluate the cost of each power node.

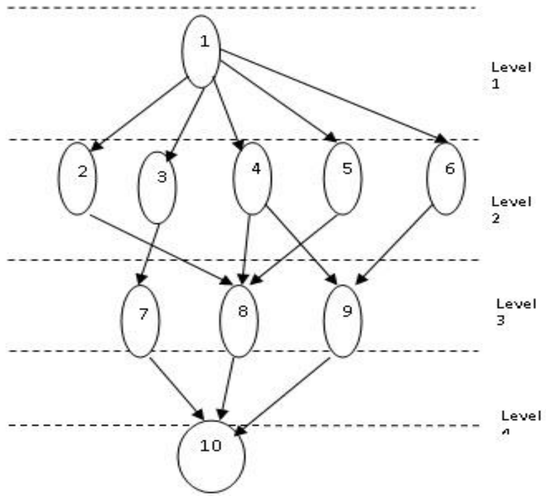

Figure 5. Levels of normal node for data transfer

Algorithm 1: Computation of power consumption when there is only one parent or as no parent:

$$cost = \frac{d}{dt}\,(Previous_{cost}) + cost\ at\ new\ power\ node \quad (8)$$

Now, when fist node has to transfer the data each power node is left with 100 milli joule of energy. Hence, the computation level will be formed as following:

Table 2: Energy consumption of node 1 at 3 power nodes per packet frame in millijoules

PN_1	PN_2	PN_3
Energy Consumption in millijoule		
2	4	6

Here,

$PN_1 \rightarrow$ 0 to 2 milli joules

PN_2→ 0 to 4 milli joules

PN_3→ 0 to 6 milli joules

Here, it is obvious that Node 1 will transfer data through first power node PN_1 as it is consuming least energy for that transfer.

At level two, there are more than one node (2, 3, 4, 5 and 6). Now, which node will transfer the data first, must be decided as per the priority. In such as scenario, cuckoo search will be applied.

THE CUCKOO SEARCH ALGORITHM

The cuckoo search [20] is a swarm intelligence based algorithm in which if one segment doesn't satisfy any of the constraints then the entire segment is dumped. Here, Setting up the priority with cuckoo search is really going to be a challenge [21]. The following algorithm has been designed for the same.

Algorithm 2: Setting up the priority

Function set_priority (nodes,energycosting) { initialize totalenergy=0; Initialize total energy=0; initialize priority_order=[];

for j=1: Node

for i=1: Powernodes.count

totalenergy=totalenergy+energycosting(j,i);

end for

end for

total energy=totalenergy/(node*(powernodes.count));

priority_order=Zeros(1.numel(Nodes)); egg_broken=false;

total_eggs=(Nodes*Power_Nodes);

for 1<=0:total_eggs/PowerNodes

current_egg_value=$\sum_{i=0}^{power\_Nodes\_1} EnergyNodes(k)$/i;

if (current_egg_value*random (

)<totalenergy*random( ))

egg_broken=true;

end if

if egg_broken==true

temp=(Priority_order[last];

Adjsut (Priority_order one position);

else

Priority_order [K]=total-eggs [i];

end if

The above algorithm determined the priority order of the nodes of the given area.

Cuckoo fitness [22] is designed to set the order of the nodes. A cuckoo breaks all its eggs if it finds any egg to be faulty .Here the threshold is randomness in the average cost at energy power node .If a random variation is the total cost is made and the threshold is compared with variation .if the node satisfies the condition then it is set to its position else it is placed last.

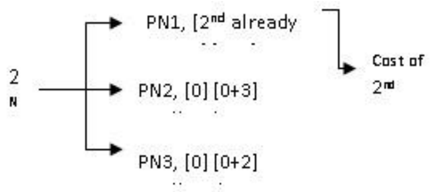

Figure 6. Data transfer points for 2nd node

For the given scenario, the selection process may vary by one or two node positions as the cuckoo search has some randomness to it.

For no. the proceeding order is as follows:

Node-2, 6,4,3,5

As per the precedence order, 2nd node has to transfer the data the options for second node are as follows.

Now, ethically the second node has to be scheduled on process node 3 but there would be some processing cost also which would be applied from PN1 to PN2 & PN3. If some random cost is takes like from 1 to 3 (15). That would also be added to the total cost. Hence the cost will vary like

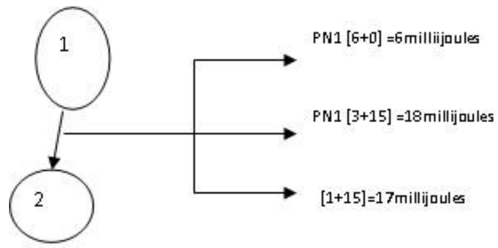

Figure 7. Cost variation for node 2

Again the power Node 1 will be elected. In such scenario, the power Node 1 will be overloaded.

Here, in such scenario, the second optimization algorithm has to put its impact.

Artificial bee colony [23] is a swarm based intelligent optimization algorithm. It is very useful when the area of application or the network model doesn't vary too much there are three kind of bee colony optimization

i. The employed bee: here in the presented scenario, the entire available, power node will act as employed bee.

ii. The outlook bee: the out lookers bee will be the set constraints which will be applied for the selection of the power node.

iii. Scout Bee: it would be that power node, which will go to rest node for a while.

The algorithm architecture is as follows:

Algorithm of ABC

Function selection – power-ABC (power node. Consumption, previous allocations)

Contact = []// it is an array designed to see who can perform the transfer operation

For i=1: power-node

Emp.bee=power-node[i];

Oulooker.threshold=[]; // initializing empty array

Designed for =node-info;

K=$\sum_{j=1}^{n} all\ preivious\ allocations$

If (emp.bee.loadcount<=k)

Count= cancount+1;

Contact[cancount,1]=node info;

Conact[cancount,2]= powernodeinfo;

End if

End for

End function

The objective function of proposed work is given

$$ABC_{ff} = \left\{ \begin{matrix} Bee_{current} & if \quad Bee_{current} > Bee_{onlooker} \\ Bee_{onlooker} & else \end{matrix} \right\} \tag{9}$$

The above format will decide that which power node is already overloaded and cannot be used to transfer data. The "Contact" variable will return that list which would identify the possible power nodes.

There is one last one problem left and that is which power node will be finally selected for the transfer. Here, the energy consumption would be calculated at each possible node and data would be transferred through that power node which would consume least energy.

[Node List; Energy]=find min (Energy consumption. canact);

After the optimization we need to select the better node for the packet transmission and minimization of energy consumption rate so we use the artificial neural network. The algorithm of artificial neural network is given as;

Load Opitmized_Data

Trainingdata = Opitmized_Data

Initialize ANN

Generate group of data = group

Set iteration = 1000

Performance parameter = MSE, Gradient, Mutation & Validation checks

Training Algo = Trainlm

for i = 1: iteration

Weight = Opitmized_Data(i)

Generate Net structure of ANN (net) as a feed forward back propagation technique.

Net_Output = train (net, Trainingdata, group)

end

By using the above described network, we can classify the best node to transmit the packet data and improve the performance parameters of the proposed work. In the proposed work, we have used feed forward back propagation neural network because the capability and the classification rate is dependent on the properties of optimized data. Training of ANN requires is based on certain learning processes so we can say that the feed forward back propagation neural network is used as a feedback module and it can arrange the data according to their group. Initialization of artificial neural network is the primary step which accepts three parameters i.e. training data, group to which each element of data belongs and number of hidden neurons in hidden layer and in the training phase, we have defined the number of iterations which can we used by the system based on feedback process. There are two parameters of validations:

1. The first is the validation checks which are 6 in MATLAB and in proposed algorithm. It may vary according to the versions of MATLAB. Either the validation gets complete or the Gradient becomes equal to the decided gradient of the Neural network would be sufficient enough to stop the training of the artificial neural network.
2. After that data is passed to artificial neural network for training purpose of data which create the category according to the properties of data. Artificial Neural Network Toolbox supports a variety of supervised and unsupervised network architectures. With the toolbox’s modular approach to building networks, custom network architectures can be developed for specific problem.

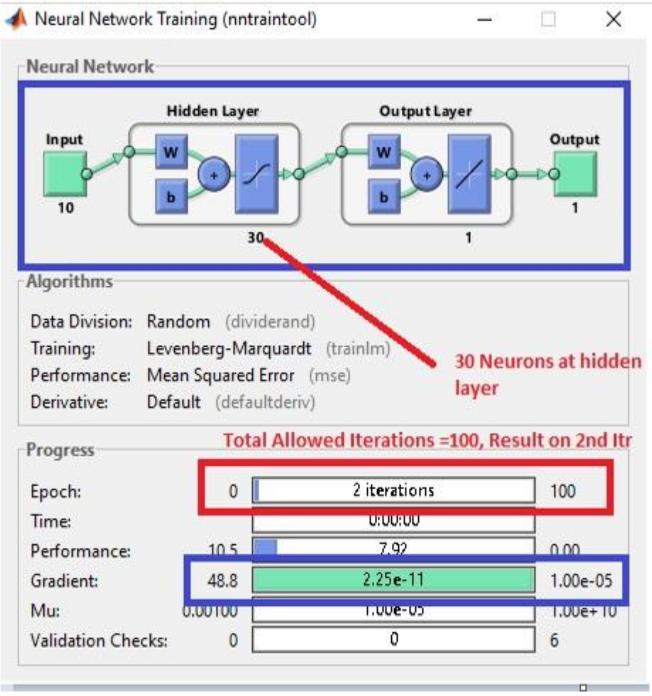

Figure 8. Neural network training architecture

The network architecture can be developed including all inputs, layers, outputs, and interconnections. After that number of iteration are given how much iteration will be sufficient for complete training. The architecture diagram says that the Network first understands the previous setup of the Nodes and the load provided to it. It distributes the load based on the training architecture and defines in iterative selection that how much load should be taken off from each node. It is a feed forward back propagation Neural Network as it traces back the results. Based on the validations, it goes forward and once the validation is complete it traces back based on Mean Square Error.

The following environment has been utilized for the processing

Table 3: Simulation Environment

Area of simulation	1000*1000
Mode of Deployment	Random
Tool	MATLAB
Entry Node	1
Exit Node	1

RESULTS AND DISCUSSION

The following results have been obtained:

i. Energy Consumption

It is the total energy consumed in the transfer of data. It would be the total Sum of the search energy, allocation energy, and execution energy. A total of 10 different node modes with one entry and 1 exit node have been simulated and the following graph.

Table 4: Energy Consumption

Model No	1	2	3	4	5	6	7	8	9	10
Energy Consu-med	47	55	32	41	40	45	26	38	51	52

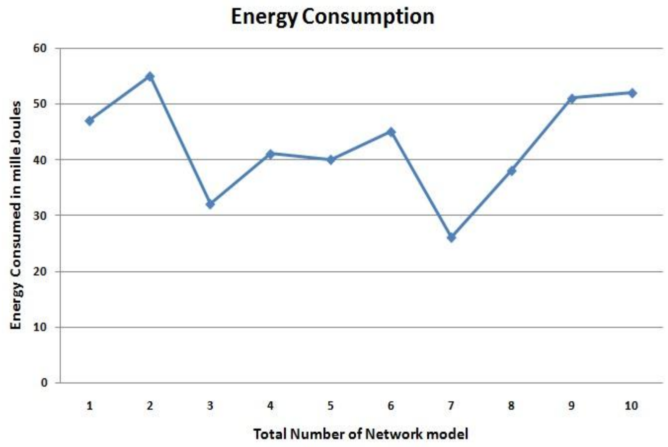

Figure 9. Energy Consumption Vs total number of network model

A maximum of 55 milli joules of energy has been recorded where as a minimum of 32 milli joules has been obtained.

ii. Load per power node

It is the total load in which each power node out of 3 has been processed.

Table 5: Allocated Nodes

Model No	1	2	3	4	5	6	7	8	9	10
Nodes to P1	3	2	4	4	4	5	4	3	5	1
Nodes to P2	2	5	4	4	2	1	3	4	4	4
Nodes to P3	5	3	2	2	4	4	3	3	1	5

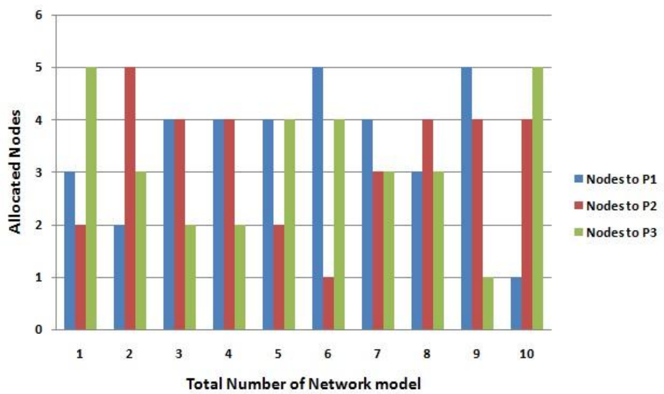

Figure 10. Allocated nodes Vs total number of job model

It has been observed that the proposed algorithm has efficiently chosen that power node which consumes least energy and also the graphs show that the nodes are well balanced.

Apart from the allocation, delay has also been evaluated and the following tables have been recorded.

Table 6: Total Delay in ms

Model No	1	2	3	4	5	6	7	8	9	10
Total Delay in ms	4	5	3.8	6	6.2	5.8	5.2	6.3	6.1	4.5

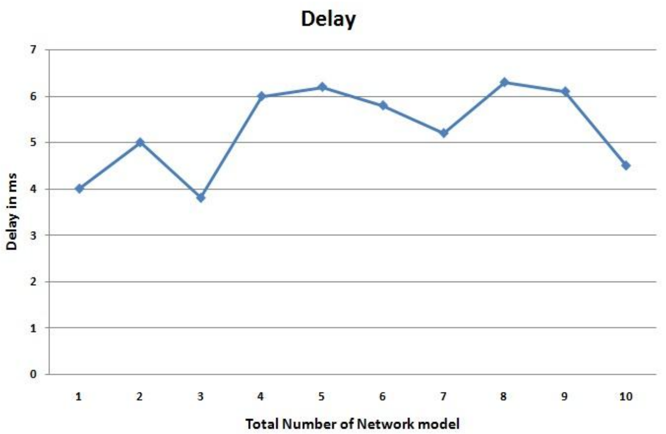

Figure 11. Delay vs Total number of Job models

Fig. 11 shows variation in delay with the change in job models. The variations are uncertain due to the uncertainty in the job model. A maximum delay of 6.3 has been observed where as the minimum delay has been recorded as 3.8.

Table 7: Throughput in percentage

Model No	1	2	3	4	5	6	7	8	9	10
Throughput (%)	84.53	85.97	93.8	86.96	86.2	85.8	85.2	86.3	91.2	92.5

The delay has been calculated as the total time taken for the allocation of the nodes to the transferring node and the total time for which a request waits in the queue of the power node.

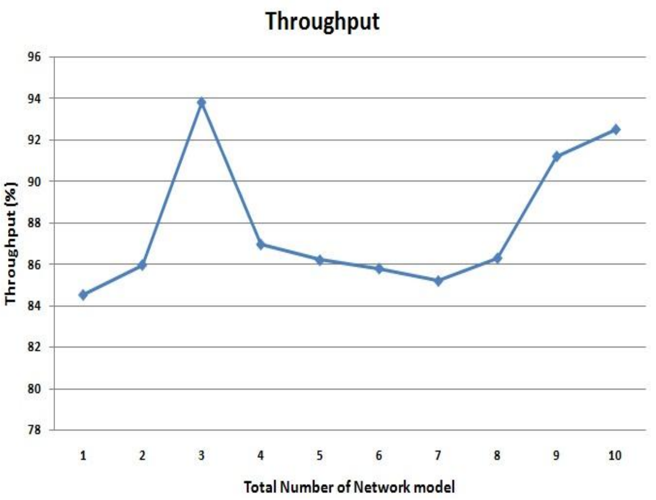

Figure 12. Throughput vs Total number of Job models

Above table 7 and fig.12 shows variation in throughput according to the change in job models.

Form the figure we observed that the average throughput is more than 87% with respect to the job models and the throughput is acceptable

Table 8: Packet loss in percentage

Model No	1	2	3	4	5	6	7	8	9	10
Packet Loss (%)	4.83	8.58	3.85	6.74	4.24	8.58	5.22	8.03	1.25	9.2

Table 8 and fig. 13 shows variation in packet loss during the transmission of packets according to the change in job models.

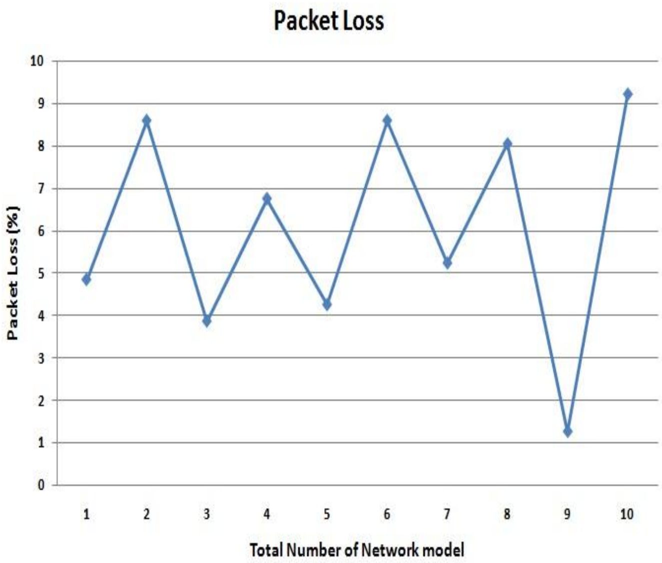

Figure 13. Packet loss vs Total number of Job models

From the figure, we observed that the average packet loss I s less than 6% with respect to the job models and it is necessary for all models the rate of packet loss would be minimum. In the proposed work, by hybridization of optimization technique the probability of packet losses is reduced.

CONCLUSION

A network in which the nodes have to transfer data through other power nodes is not easy when it comes to the management. The network proceedings have always suffered from transmission losses and overload for such scenarios it becomes quite difficult to save the network for path failure. When each node is a part of a path then precedence also plays an important role. The proposed solution in the paper has resolved the problem of precedence using cuckoo search. Furthermore, each node may have 'n' number of options for the transfer of data. Here, advanced artificial bee colony has been applied to select final power node for the data transfer. The system is being evaluated on 1000*1000 area of simulation. The entire system has been evaluated on MATLAB 2012 and four evaluation parameters have been computed. The evaluation of the work is on dependent on different parameters, like Energy Consumption, Total delay, Throughput and packet loss. It is being seen that energy consumption being obtained is 55 mill joules; delay is maximum 6.3 and 3.8 as minimum. The delay variability is with respect to job models. Maximum throughput obtained is 87 % by means of job models. The observed packet loss is 6 % with respect to job models which is being reduced by means of the optimization technique.

REFERENCES

[1] Kobo H. I, A. M. Abu-Mahfouz and G. P. Hancke. A Survey on Software-Defined Wireless Sensor Networks: Challenges and Design Requirements. *in IEEE Access*, vol. 5, no. , pp. 1872-1899, 2017.

[2] M. Asif, S. Khan, R. Ahmad, M. Sohail and D. Singh. Quality of Service of Routing Protocols in Wireless Sensor Networks: A Review. *in IEEE Access,* vol. 5, no. , pp. 1846-1871, 2017.

[3] D. Djenouri and M. Bagaa. Synchronization Protocols and Implementation Issues in Wireless Sensor Networks: A Review. *in IEEE Systems Journal*, vol. 10, no. 2, pp. 617-627, June 2016.

[4] A. and H. Y. Kong. Energy efficient cooperative LEACH protocol for wireless sensor networks. *in Journal of Communications and Networks*. vol. 12, no. 4, pp. 358-365, Aug. 2010.

[5] Xu, J., Wang, K., Wang, C., Hu, F., Zhang, Z., Xu, S., & Wu, J. Byzantine fault-tolerant routing for large-scale wireless sensor networks based on fast ECDSA. *in Tsinghua Science and Technology*, vol. 20, no. 6, pp. 627-633, December 2015.

[6] W. Zhang, L. Li, G. Han and L. Zhang. E2HRC: An Energy-Efficient Heterogeneous Ring Clustering Routing Protocol for Wireless Sensor Networks. *in IEEE Access*, vol. 5, no. , pp. 1702-1713, 2017.

[7] Y. Zhou, N. Wang and W. Xiang. Clustering Hierarchy Protocol in Wireless Sensor Networks Using an Improved PSO Algorithm. *in IEEE Access,* vol. 5, no. , pp. 2241-2253, 2017.

[8] J. S. Lee and T. Y. Kao. An Improved Three-Layer Low-Energy Adaptive Clustering Hierarchy for Wireless Sensor Networks. *in IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 951-958, 2016.

[9] P. Ho and H. Tan. A Coherent Multi-User Frequency Shift Keying Receiver for Wireless Sensor Networks Based on Clustering Algorithms. *in IEEE Wireless Communications Letters*, vol. 5, no. 6, pp. 580-583,2016.

[10] C. Zhu, V. C. M. Leung, L. T. Yang and L. Shu. Collaborative Location-Based Sleep Scheduling for Wireless Sensor Networks Integratedwith Mobile Cloud Computing. *in IEEE Transactions on Computers*, vol. 64, no. 7, pp. 1844-1856, 2015.

[11] Y. Miao, S. Vural, Z. Sun and N. Wang. A Unified Solution for Gateway and In-Network Traffic Load Balancing in Multihop Data Collection Scenarios. *in IEEE Systems Journal*, vol. 10, no. 3, pp. 1251-1262, 2016.

[12] S. J. Habib and P. N. Marimuthu. Data aggregation at the gateways through sensors' tasks scheduling in wireless sensor networks. *in IET Wireless Sensor Systems*, vol. 1, no. 3, pp. 171-178, 2011.

[13] R. Misra and C. Mandal. Rotation of CDS via Connected Domatic Partition in Ad Hoc Sensor Networks. *in IEEE Transactions on Mobile Computing*, vol. 8, no. 4, pp. 488-499, 2009.

[14] Y. Wei, Y. Mao, S. Leng and W. Huang. A low-overhead energy-efficient ARQ protocol for wireless sensor networks. *in China Communications,* vol. 11, no. 10, pp. 74-87, 2014.

[15] S. Rührup and I. Stojmenovic. Optimizing Communication Overhead while Reducing Path Length in Beaconless Georouting with Guaranteed Delivery for Wireless Sensor Networks. *in IEEE Transactions on Computers*, vol. 62, no. 12, pp. 2440-2453, 2013.

[16] Z. Xu, L. Chen, C. Chen and X. Guan. Joint Clustering and Routing Design for Reliable and Efficient Data Collection in Large-Scale Wireless Sensor Networks. *in IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 520-532, 2016.

[17] X. Liu, Q. Huang and Y. Zhang. Balancing Push and Pull for Efficient Information Discovery in Large-Scale Sensor Networks. *in IEEE Transactions on Mobile Computing,* vol. 6, no. 3, pp. 241-251, 2007.

[18] M. Al-Jemeli and F. A. Hussin. An Energy Efficient Cross-Layer Network Operation Model for IEEE 802.15.4-Based Mobile Wireless Sensor Networks. *in IEEE Sensors Journal*, vol. 15, no. 2, pp. 684-692, 2015.

[19] Z. Chen, X. Li, B. Lv and M. Jia. A Self-Adaptive Wireless Sensor Network Coverage Method for Intrusion Tolerance Based on Particle Swarm Optimization and Cuckoo Search. *IEEE Trustcom/BigDataSE/ISPA, Helsinki*,vol.1, pp. 1298-1305,2015.

[20] H. Xuan, R. Zhang and S. Shi. An Efficient Cuckoo Search Algorithm for System-Level Fault Diagnosis. *in Chinese Journal of Electronics*, vol. 25, no. 6, pp. 999-1004, 2016.

[21] Q. Liao, S. Zhou, H. Shi and W. Shi. Parameter Estimation of Nonlinear Systems by Dynamic Cuckoo Search. *in Neural Computation*, vol. 29, no. 4, pp. 1103-1123,2017.

[22] H. Wang, Y. Chen and S. Dong. Research on efficient-efficient routing protocol for WSNs based on improved artificial bee colony algorithm, *in IET*

Wireless Sensor Systems, vol. 7, no. 1, pp. 15-20, 2 2017.

[23] S. J. Habib and P. N. Marimuthu, Data aggregation at the gateways through sensors' tasks scheduling in wireless sensor networks. in *IET Wireless Sensor Systems*, vol. 1, no. 3, pp. 171-178, 2011.

[24] Rizos Sakellariou and Henan Zhao. A hybrid heuristic for dag scheduling on heterogeneous systems. *in 18th International Symposium on Parallel and Distributed Processing. IEEE*, pp. 111-124,2004.

[25] Singh, Sukhjit, and Nirmal Kaur. A Heterogeneous Static Hierarchical Expected Completion Time Based Scheduling Algorithm in Multiprocessor System. *International Research Journal of Engineering and Technology (IRJET*) Vol 3 , pp951-956,2016.

[26] Petersen, S., Doyle, P., Vatland, S., Aasland, C.S., Andersen, T.M. and Sjong, D.Requirements, drivers and analysis of wireless sensor network solutions for the Oil & Gas industry. *Emerging Technologies and Factory Automation, ETFA. IEEE Conference on. IEEE*, pp-219-226, 2007.