ISSN (ONLINE): 2456-5717

International Journal of Advanced Research in Basic Engineering Sciences and Technology (IJARBEST)
Vol.3, Issue.11, November 2017



# Enlarging the Network Duration Using Fuzzy Sense Based Clustering Algorithm for WSN

[1]K.J.Anila Anbu Malar, [2]T.Rajesh
[1]PG Scholar, Communication Systems, PSN College of Engineering and Technology, TamilNadu, India
[2]Associate Professor, PSN College of Engineering and Technology, TamilNadu, India

***Abstract* – One of the most vital activities to reduce energy consumption in wireless sensor networks is clustering. In clustering, one node from a group of nodes is selected to be a cluster head, which handles majority of the computation and processing for the nodes in the cluster. This paper proposes an algorithm for fuzzy based dynamic cluster head selection on cloud in wireless sensor networks. The proposed algorithm calculates a Potential value for each node and selects cluster heads with high potential. The proposed algorithm minimizes cluster overlapping by spatial distribution of cluster heads and discards malicious nodes i.e. never allows malicious nodes to be cluster heads.**

***Index Terms* – Wireless Sensor Network (WSN), Selection of Cluster Heads (SCH), Fuzzy logic**

## I. INTRODUCTION

Wireless sensor networks(WSN), sometimes called wireless sensor and actuator networks(WSAN), are spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, also enabling control of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on. The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding. In computer science and telecommunications, wireless sensor networks are an active research area with numerous workshops and conferences arranged each year, for example IPSN, SenSys, and EWSN

## II. Systems Assumption

In the proposed model, sensor nodes are considered to be deployed randomly to monitor the environment continuously.

1) All the sensor nodes are static except the base station
2) The base station is mobile
3) Homogeneous networks have been considered such that all the sensor nodes have initial equal energy.
4) The distance between the node and the base station can be computed based on received signal strength

## III. Systems Model

The proposed clustering method follows the basic principle of LEACH. The cluster is formed in each round. In every clustering round, each node generates a random number between 0 and 1. If the random number for a particular node is bigger than the threshold value T, the node becomes the CH. In basic LEACH [2], the cluster formation algorithm was defined to ensure that the no. of cluster per round is $k$, a system parameter. The optimal value of $k$ *(k optimal)* in LEACH can be determined analytically by computation and communication energy model. For instance, if there are N nodes distributed randomly over M×M region, and $k$ clusters are assumed, then there are N/$k$ nodes per cluster (one CH and (N/$k$)-1) Non Cluster head nodes. Each CH dissipates energy by receiving the signal, aggregates it and sends the average signal to BS.

It is assumed that sensor nodes send the data after detecting an interesting event. CH collects these data, aggregates it and send to the base station. To save some energy, we may think of one SCH among the CHs can send the data to BS to utilize the bandwidth efficiently. Instead of multiple CHs, one SCH can deliver the message to BS that can reduce energy consumption and enhances energy efficiency. The proposed model is depicted in Figure 3. Another assumption we have made the BS mobility that can relax collision avoidance by collecting the data from SCH. Other sensor nodes including CHs and SCH remain static. Base station may adopt many different paths to collect the information from the SCH Further assumption we have made that Remaining battery power, Mobility, and Centrality: the three fuzzy descriptors are suitable to calculate the chance to be the SCH that can deliver the message to the BS. Assuming that in each round energy level of each CH gets reduced, remaining battery power has been considered as a metric. Centrality is considered as another major metric because centrality focuses on the location of SCH how much it is positioned centrally to communicate with other CHs. Mobility implies that when BS moves in a particular direction, the distance between BS and SCH increases or decreases w.r.t to the speed and direction of moving BS.

ISSN (ONLINE): 2456-5717

International Journal of Advanced Research in Basic Engineering Sciences and Technology (IJARBEST)
Vol.3, Issue.11, November 2017

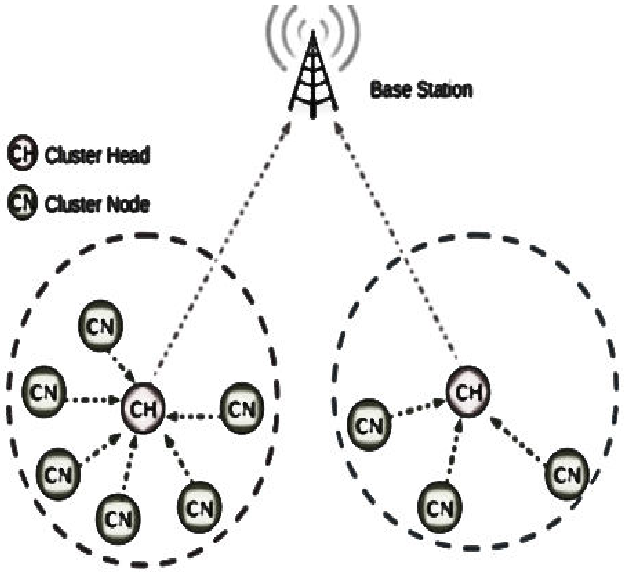


Fig 1. Wireless Sensor Network Model.

.

## IV. FUZZY LOGIC MODEL

The Fuzzy logic model consists of four modules: a fuzzifier, fuzzy inference engine, fuzzy rules and a defuzzifier. The block diagram of the fuzzy Inference system is represented below the explanation. There are four steps required to complete the process.

*Fuzzification*:

In fuzzifier, inputs are given with crisp value and changed into a fuzzy set. This can be achieved by fuzzification.

*Rule Evaluation:*

It stores IF-THEN rule.

*Fuzzy Inference Engine:*

This engine takes both the input values and IF-THEN rules to simulate the reasoning by which it produces a fuzzy inference.

*Defuzzification:*

Defuzzifier transforms the fuzzy set into crisp value

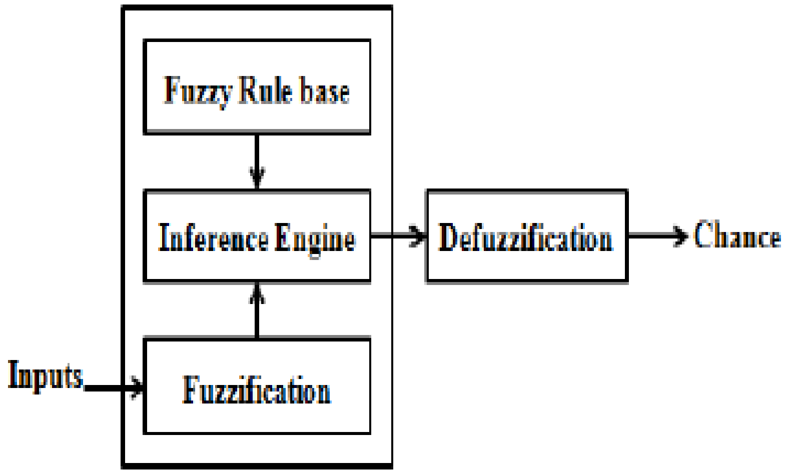


Fig 2. Block Diagram of Fuzzy Interface System.

*a) Fuzzification Module*

In our proposed protocol, Mamdani's Method Fuzzy Inference technique is used to elect the SCH as it is the most frequently used inference technique. The inference techniques and the Fuzzy system used for our proposed model. We have taken three fuzzy input variables to elect the tentative Super Cluster Head. All the three input variables have three membership functions each. The fuzzy set that represents the first input variable i.e. remaining battery power. The linguistic variables for the fuzzy set is less, medium and high. Trapezoidal membership function has been considered for less and high. For medium, triangular membership function has been considered.

The second fuzzy input variable is the Mobility of the base station because the proposed protocol considers the base station as mobile. The linguistic variables for mobility are taken as low, moderate, and frequent. The third fuzzy input variable is the centrality that how much the SCH is central to other clusters. The linguistic variables for centrality are considered as close, adequate, and far. The degree of the membership function is shown by a numerical after each membership function.

TABLE I – MEMBERSHIP FUNCTIONS (1)

| Remaining Battery Power | Mobility | Centrality |
|---|---|---|
| Less (0) | Low (0) | Close (0) |
| Medium (1) | Moderate (1) | Adequate (1) |
| High (2) | Frequent (2) | Far (2) |

*b) Rule Base and Interference Engine*

In our system, we have used 27 rules in the fuzzy inference. The form of the rules is if X, Y, Z then C. X represent remaining battery power, Y represents mobility, Z represents the centrality, and C represents the chance.

TABLE II – MEMBERSHIP FUNCTIONS (2)

| Chance |
|---|
| very weak(-1), weak(0), lower medium(1), medium(2), higher medium(3), strong(4), very strong(5) |

We have considered remaining battery power as (Battery Power – 1) because in each round there will be some energy consumption at each node. So, after processing of each round, the remaining energy is considered for the next round. Mobility and Centrality is assumed to be additive factor why because the distance of SCH from base station increases or decreases with respect to the movement of base station. When SCH delivers the message to BS how much it is central to other clusters to deliver the message. As discussed above, the output variable chance of a node for getting elected as a SCH is calculated using remaining battery power, mobility of base station, and centrality of the clusters. The output chance is composed of 7 membership functions very weak, weak, lower medium, medium, higher medium, strong, very strong.

## V. METHODOLOGY

*a) Initialization*

Nodes randomly deployed with initial energy. Find the neighbours and total number of neighbours by using Euclidian distance d(i,j) and communication range R.

*b) First Stage of CH selection-NPC Calculation*

Initialize the network to ensure that the wireless sensor network reaches an initial coverage set by the user. Calculate the perceived probability of all sensor nodes P (si ,m), according to the formula (4); Use the formula to obtain the network public perception (NPP) of each sensor node based on the calculation of P(si , m). Determine a node the redundant node or not, based on the size of NPP (m) in the network. Namely, determine whether its redundant flag RF is 1; The network perception contribution of each sensor network node can be obtained by the results got from step 3 and 4; Select the cluster head node from the redundant nodes. The network perception contribution NP C(si) of the cluster head node is the maximum. Set the sensing intensity of the sleeping sensor node to be 0 and the coverage of the entire network are updated at the same time. When the sensor nodes satisfy NPP(m) ≥ Mmin, set the redundant flag of this sensor to be 1, namely RF(s) = 1. The sensing field of those nodes can be covered

ISSN (ONLINE): 2456-5717

International Journal of Advanced Research in Basic Engineering Sciences and Technology (IJARBEST)
Vol.3, Issue.11, November 2017

by other nodes, then such redundant nodes are figured as the first class of cluster head nodes. A cluster head does not perceive information, which can reduce the energy of perception and data fusion.

$$d(s_i, m) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

The perceptual probability model of $s_i$ to m is:

$$P(s_i, m) = \begin{cases} 1 & d(s_i, m) < R_s - u \\ e^{-\alpha d_i} & R_s - u \leq d(s_i, m) < R_s + u \\ 0 & d(s_i, m) \geq R_s + u \end{cases}$$

$$NPP(m) = \sum_{i=1}^{N} P(s_i, m) = \sum_{i=1}^{N} e^{-\alpha d_i}$$

$$NPC(s_i, m) = \frac{P(s_i, m)}{NPP(m)} = \frac{P(s_i, m)}{\sum_j Ps_i, m}$$

*c) Cluster Forming and routing*

Each selected CH broadcast hello packets to every nodes in the network. Each node join with nearer CH by using the Euclidian distance d(i,ch) for form the cluster. Each node send data to CH and the CH forward the data to base station for particular round (until the CH died).

*d) Second stage CH selection*

When CH died after some rounds of communication second stage CH is elected. To select second stage CH we calculate E(i) residual energy for each nodes. Now consider both NPC and E(i) of nodes to select CH.E(i) = IE(i) – CE(i) ( initial energy – current energy)

*e) Optimized routing*

To optimized in the routing CH select another CH to reach the BS (base station) or Sink. To select another CH, it Consider minimum npc value, maximum E(i) and minimum distance to reach the BS. Each CH node selects forwarder CH to reach BS or directly communicate with BS to optimize in routing.

## VI. SOFTWARE DESCRIPTION

*a) NS2 Structure introduction*

NS2 is an object oriented simulator, written in C++, with a Tcl interpreter as a front-end. The simulator supports a class hierarchy in C++ (also called the compiled hierarchy), and a similar class hierarchy within the Tcl interpreter (also called the interpreted hierarchy).

The two hierarchies are closely related to each other; from the user's perspective, there is a one-to-one correspondence between a class in the interpreted hierarchy and one in the compiled hierarchy.

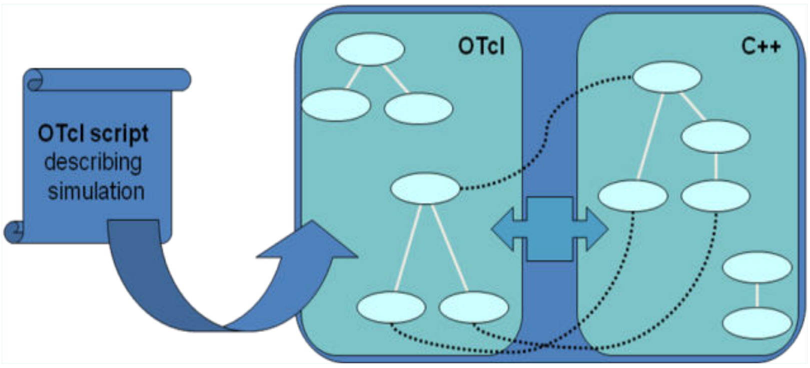


Fig 3. Representation of OTcl and C++ Hierarchies.

NS2 uses two languages because it has two different kinds of things it needs to do: Detailed simulations of protocols require a systems programming language which can efficiently manipulate bytes, packet headers, and implement algorithms that run over large data sets.

For these tasks run-time is important and turn-around time (run simulation, find bug, fix bug, recompile, re-run) is less important. C++ is fast to run but slower to change, making it suitable for detailed protocol implementation

A large part of network research involves slightly varying parameters or configurations, or quickly exploring a number of scenarios. In these cases, iteration time (change the model and re-run) is more important. Since configuration runs once (at the beginning of the simulation), run-time of this part of the task is less important.

Tcl runs slower than C++ but can be changed very quickly (and interactively), making it ideal for simulation configuration.

Users create new simulator objects through the Tcl interpreter. These objects are instantiated within the interpreter, and are closely mirrored by a corresponding object in the compiled hierarchy. Class Tcl Object is the base class for most of the other classes in the interpreted and compiled hierarchies.

Every object in the class Tcl Object is created by the user from within the interpreter. An equivalent shadow object is created in the compiled hierarchy. The two objects are closely associated with each other.

The interpreted class hierarchy is automatically established through methods defined in the class TclClass. User instantiated objects are mirrored through methods defined in the class TclObject.

*b) Tcl / C++ Variable Binding*

Class InstVar defines the methods and mechanisms to bind a C++ member variable in the compiled shadow object to a specified Tcl instance variable in the equivalent interpreted object. The binding is set up such that the value of the variable can be set or accessed either from within the interpreter, or from within the compiled code at all times.

Whenever the variable is read through the interpreter, the trap routine is invoked just prior to the occurrence of the read. The routine invokes the appropriate get function that returns the current value of the variable.

This value is then used to set the value of the interpreted variable that is then read by the interpreter. Likewise, whenever the variable is set through the interpreter, the trap routine is invoked just after to the write is completed.

The routine gets the current value set by the interpreter, and invokes the appropriate set function that sets the value of the compiled member to the current value set within the interpreter.

*c) The basic primitive for creating a node*

*set ns [new Simulator]*

*$ns node*

The instance procedure node constructs a node out of simpler classifier objects (to be discussed later). The Node itself is a standalone class in Tcl. However, most of the components of the node are themselves Tcl Objects.

This simple structure consists of two Tcl Objects: an address classifier (classifier_) and a port classifier (dmux). The function of these classifiers is to distribute incoming packets to the correct agent or to correct outgoing link

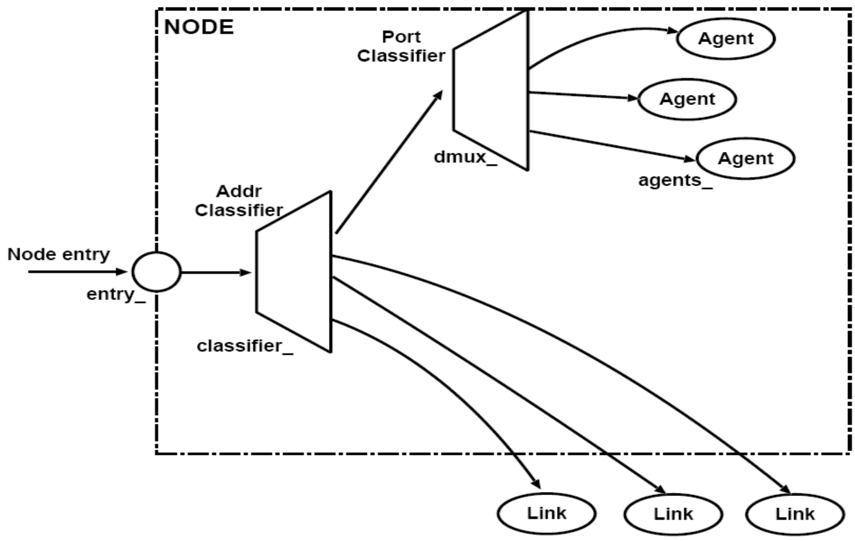


Fig 4. Distribution of incoming packets to the corresponding agent link

*d) Trace and Monitoring Support*

There are a number of ways of collecting output or trace data on a simulation. Generally, trace data is either displayed directly during execution of the simulation, or (more commonly) stored in a file to be post-processed and analysed. There are two primary but distinct types of monitoring capabilities currently supported by the simulator.

The first, called traces, record each individual packet as it arrives, departs, or is dropped at a link or queue. Trace objects are configured into a simulation as nodes in the network topology, usually with a Tcl "Channel" object hooked to them, representing the destination of collected data (typically a trace file in the current directory). The other types of objects, called monitors, record counts of various interesting quantities such as packet and byte arrivals, departures, etc.

## VII. SIMULATION

To check the validity of the proposed protocol, NS-2 simulator (2.34) has been used as the tool to compare the performance metrics of our interest with LEACH protocol which ensures extended lifetime of the WSN for the proposed protocol.

*a) Experimental Set-Up*

In this experiment, we have considered 40 nodes randomly deployed over the area between (x=0, y=0) and (x=100, y=100) with BS location (x=50, y=50). We assume four no. of clusters. Each round duration is 20s. The bandwidth of the channel is 1 Mbps. Each data message is 500 bytes long; packet header length is 25 bytes. We have used a simple energy model. The communication parameters and the required parameters of interest. We run the simulation for 20000s. After running the simulation extensively, it is concluded that the proposed approach performs better than LEACH.

*b) Results and Discussion*

The experimental results obtained from the simulations to evaluate the proposed algorithm. Along with the simulation setup. This is defines the node density as the number of nodes position in 200*200m square area. 100 randomly connected topologies for each node. The node transmission range is set to 50m. MAC type for proposed method is IEEE 802.11. The performance of proposed scheme is assessed by using network simulator (NS).Detect malicious nodes and never allows them to be the cluster head. Node Replication Attack Detection Protocol is used to detect malicious node. Clones are generated some malicious nodes. Node Replication Attack Detection Protocol detect clones & Replay attacks detected using replay attack detection protocol. Replay attack detection protocol is based on timestamps.

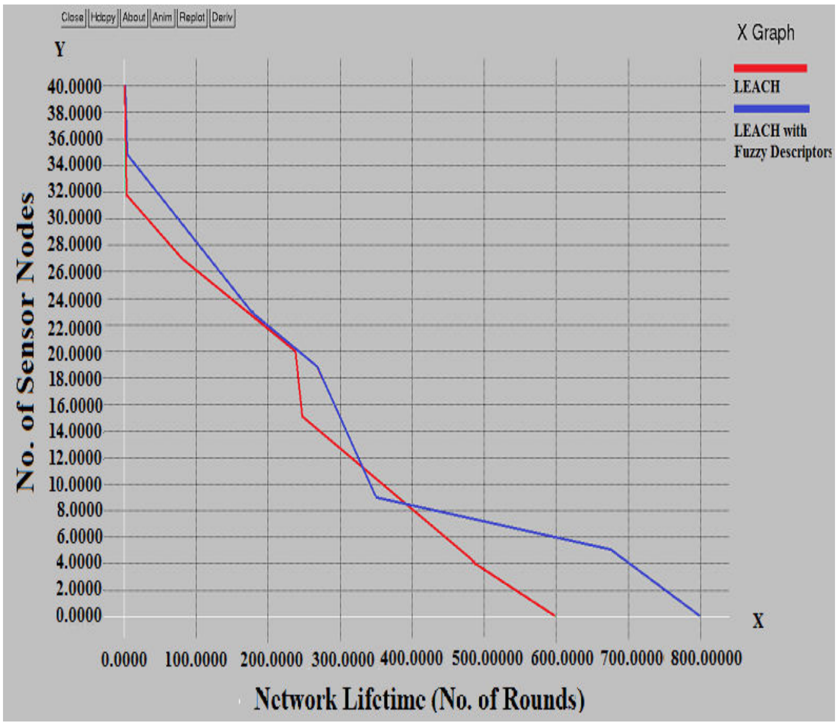


Fig 5. Network Lifetime Vs. No. of nodes.

## VIII. CONCLUSION

Protocols for Wireless Sensor Networks need to be energy efficient to prolong network lifetime. To achieve enhancement in network lifetime and reliable and efficient communication in WSNs, deploying a good clustering technique is essential. The proposed algorithm (DCHFC) for clustering offers spatially distributed cluster heads selection, which reduces cluster overlapping. DCHFC detects malicious nodes and never allows them to be the cluster head; thereby DCHFC increases the reliability and makes sensor networks fault tolerant. Cluster head selection by DCHFC is based on potential value calculated using fuzzy. The spatial distribution of clusters prevents overlapping. Having all these features DCHFC proves to be the better choice for energy consumption and enhancement of network lifetime.

## REFERENCES

[1] Shah R C, Rabaey J M. Energy aware routing for low energy ad hoc sensor networks[C]. Procs. 3rd IEEE Wireless Communications and Networking Conf.(WCNC),Orlando,2013:151-165.

[2] Fotouhi H ,Alves, M, Zamalloa MZ. Reliable and Fast Hand-Offs in Low-Power Wireless Networks [J]. IEEE TRANSACTIONS ON MOBILE COMPUTING, 2014, 13(11): 2621-2633

[3] Javaid N, Waseem M, Khan Z A. ACH: Away Cluster Heads Scheme for Energy Efficient Clustering Protocols in WSNs[C]. 2013 Saudi International Electronics, Communications and Photonics Conference, Piscataway: IEEE.2013:364-7.

[4] Javaid N, Waseem M, Khan Z A. ACH: Away Cluster Heads Scheme for Energy Efficient Clustering Protocols in WSNs[C]. 2013 Saudi International Electronics, Communications and Photonics Conference, Piscataway: IEEE.2013:364-

[5] Hosseinirad S M, Ali M M, Basu S K. LEACH routing algorithm optimization through imperialist approach [J]. International Journal of Engineering, Transactions A: Basics.2014,27(1):39-50

[6] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, Energy-efficient Communication Protocol for Wireless Microsensor Networks., in IEEE Computer Society Proceedings of the Thirty Third Hawaii International Conference on System Sciences (HICSS '00), Washington, DC: IEEE, 2000, pp. 1-10

[7] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, "An application-specific protocol architecture for wireless Microsensor networks," in IEEE Transactions on Wireless Communications, 1(4), pp. 660 – 670, Oct. 2002

[8] Gupta, D. Riordan and S. Sampalli, "Cluster -head Election using Fuzzy Logic for Wireless Sensor Networks", Communication Networks and Services Research Conference, pp. 255-260, May 2005

[9] Jong-Myoung Kim, Seon-Ho Park, Young-Ju Han, TaiMyoung Chung, "CHEF: Cluster Head Election mechanism using Fuzzy logic in Wireless Sensor Networks" ICACT, PP. 654-659, Feb. 2008.

[10] Abhijeet Alkesh, Ashutosh Kumar Singh, N.Purohit, "A moving base station strategy using fuzzy logic for life time enhancement in wireless