ENERGY BALANCING WITH CLUSTERING IN TREE BASED ROUTING PROTOCOL FOR WIRELESS SENSOR NETWORK

Sonal G. Alghare[1], Mitali Ingle[2] , Dinesh Gawande[3]

[1]*Department of CSE, DBACER, Maharashtra, India*
[2]*Department of CSE, DBACER, Maharashtra, India*
[3]*Department of CSE, DBACER, Maharashtra, India*

ABSTRACT

Wireless sensor network (WSN) is a system composed of a large number of low-cost micro-sensors. This network is used to collect and send various kinds of messages to a base station (BS). WSN consists of low-cost nodes with limited battery power and the battery replacement is not easy for WSN with thousands of physically embedded nodes, which means energy efficient routing protocol should be employed to offer a long-life work time. To achieve the aim, there is a need not only to minimize total energy consumption but also to balance WSN load. Researchers have proposed many protocols such as LEACH, HEED, PEGASIS, TBC and PEDAP. General Self-Organized Tree-Based Energy Balancing routing protocol (GSTEB) builds a routing tree using a process where, for each round, BS assigns a root node and broadcasts this selection to all sensor nodes. Subsequently, each node selects its parent by considering only itself and its neighbor's information, thus making GSTEB a dynamic protocol. Simulation results shows that GSTEB has a better performance than other protocols in balancing energy consumption, thus prolonging the lifetime of WSN.

Keywords—*energy-balance, network life-time, routing protocol, self-organized wireless sensor network.*

1. INTRODUCTION

wireless sensor network is a network capable of wireless communication usually includes spatially distributed thousands or millions of autonomous sensor nodes having micro progression ability and restricted authority, which are arbitarily and broadly set. These small nodes arrange in a little to hundreds vary from refined metropoltian areas to tremendously unfriendly,isolated regions. These nodes coperate with each other to be able to deal with environment in which:

- Sensors work absolutely wireless
- Bring together all the network(self-configuring)
- Vigorously adjust with device collapse and ruin (self-healing)
- Each node sense and then process the data.

1.1 Applications of WSN

Wireless sensor networks (WSN) become trendy these days in many regions:

- Armed surveillance
- Ecological relevance
- Forest fire detection
- Operating indoors for intrusion detection

2. PRESENT WORK

The main objective of this paper is that energy efficient routing protocol developed a centralized clustering for wireless sensor networks. The clustering-based structure is used in a protocol to efficiently organize various nodes and utilize limited resource of system (such as energy).The centralized cluster head selection algorithm of our routing protocol is performed by the base station in order to choose better cluster heads by using the global information on the network. These selected cluster heads ensure the intra-cluster data transmission and data transmission from cluster heads to the base station is energy-efficient. Furthermore, the energy dissipation in selected cluster heads can be balanced while using the cluster head selection algorithm of our protocol. The cluster formation in our protocol is able to balance the load of clusters. Thus, the cluster heads will not drain their energy quickly due to the over-burdened load.

2.1 Network and Radio Model

The work, assumes that the system model has the following properties:

• Sensor nodes are randomly distributed in the square field and there is only one BS deployed far away from the area.

•Sensor nodes are stationary and energy constrained. Once deployed, they will keep operating until their energy is exhausted.

• BS is stationary, but BS is not energy constrained.

• All sensor nodes have power control capabilities; each node can change the power level and communicate with BS directly.

• Each node has its unique identifier (ID).

2.2General Self-Organized Tree-Based Energy-Balance Routing Protocol

GSTEB outperforms LEACH, PEGASIS, TREEPSI and TBC. Reason is GSTEB is a self-organized protocol; it only consumes a small amount of energy in each round to modify the topography for the point of balancing the energy consumption. When lifetime is distinct as the time from the start of the network operation to the death of the first node in the network, this protocol prolongs the lifetime by 100% to 300% compared with PEGASIS. GSTEB is to achieve a longer network life- time for different applications. In each round, BS assigns a root node and broadcasts its ID and its coordinates to all sensor Nodes. Then the network computes the path either by transmitting the path information from BS to sensor nodes or by having the same tree structure being dynamically and individually built by each node. For both cases, GSTEB can change the root and reconstruct the routing tree with short delay and low energy consumption. The operation of GSTEB is divided into Initial Phase, Tree Constructing, Self-Organized Data Collecting and Transmitting Phase, Information Exchanging.

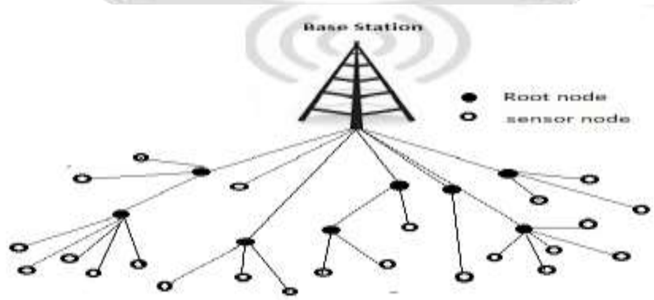

Fig1: Architecture of GSTEB protocol

3. PROPOSED METHODOLOGY

In General Self-Organized Tree based Energy Balance routing protocol (GSTEB) Consider a situation in which the network collects information periodically from a terrain where each node continually senses the environment and sends the data back to BS. General Self-Organized Tree-Based Energy-Balance routing protocol (GSTEB) builds a .routing tree using a process where, for each round, BS assigns a root node and broadcasts this selection to all sensor nodes.

3.1General Self-Organized Energy Balance Clustering Tree based routing protocol Operation

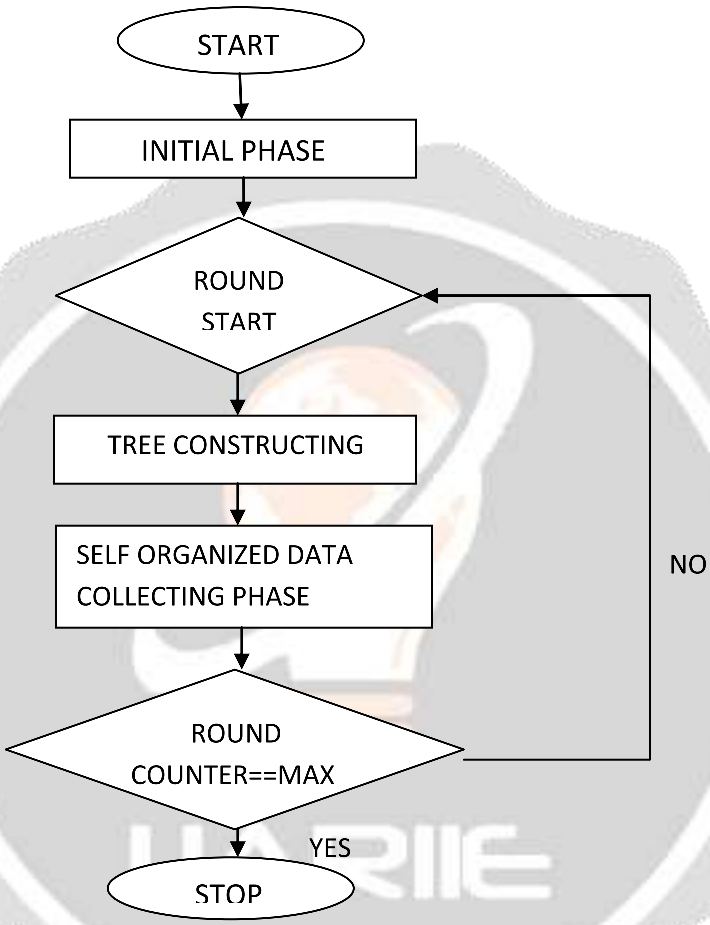

Fig 2: Flowchart of GSTEB

The operation of GSTEB is divided into: Initial Phase, Tree Constructing Phase, Self-Organized Data Collecting and Transmitting Phase, Information Exchanging Phase

3.1.1 Initial Phase

In Initial Phase, the network parameters are initialized. Initial Phase is divided into three steps

Step 1: When Initial Phase begins, BS broadcasts a packet to all the nodes to inform them of beginning time, the length of time slot and the number of nodes N. When all the nodes receive the packet, they will compute their own energy-level (EL) using function:

EL(i) = (Residual Energy(i) / α)

EL is a parameter for load balance, and it is an estimated energy value rather than a true one , i is the ID of each node, and α is a constant which reflects the minimum energy unit and can be changed depending on our demands.

3.1.2 Node Energy

In this module, Initially energy is assigned to all the 24 nodes as we can see in the above fig 10 .Researcher has also calculated the initial as well as residual energy of node.

3.1.2.1 Algorithm for Finding Residual Energy

```
# Calculation of residual energy

Input:- Number of nodes with Initial Energy

Output:- Nodes with residual energy

Create node =node_id;

            Set routing = AODV;

            Set Channel = 802.11;

            Set Initial Energy = $initial energy;

            Set Residual Energy=$residual energy;

            Set radio range=default;

        If ((node in radio range) && (next hop! =Null)

                  {

Capture data_load (node_all) ;

Create node_Configure (rreq, rrep, tsend, trecv, tdrop, initial energy, residual energy);

                              {

                                        pkt_type;

                                        Time;

                              Tsend, trecv, tdrop, rrep, rreq ;

                              }

                             }

Consume energy[node_id] = initial energy[node_id]-final energy[node_id]

Total energy[node_id] = consume energy[node_id]

if(max energy[node_id] <consume energy[node_id])

{

Max energy = consume energy[node_id]

}

###compute average energy

Average energy=total energy/n
```

In above Node Energy Algorithm, Researcher first

1. create a node,
2. Set channel,
3. Set Initial Energy,
4. Set Residual Energy,
5. Set Radio Range

Here Researcher gave a condition If ((node in radio range) && (next hop!=Null)

If condition is satisfied then Capture all the node data and cluster formation is proceed. Here Researcher calculates Consume Energy of nodes and total energy of nodes using below formulae.

Consume energy[node_id] = Initial energy[node_id]- Final energy[node_id]

Total energy[node_id] = consume energy[node_id]

For example Researcher calculates consume energy of node 7 using formula

Consume energy[6] = final energy[6] – initial energy[6]

= 51.9567-38.0433

= 13.9134 Joule

Total energy[6] = consume energy[6]

= 13.9134 Joule

Step 2: Each node sends its packet in a circle with a certain radius Rc during its own time slot after Step 1. All the other nodes during this time slot will monitor the channel, and if some of them are the neighbors of node i, they can receive this packet and record the information of node i in memory. The nodes which are not in the range of Rc can't monitor the preamble in this time slot, so they can know they are not the neighbors of node i and will turn off their radios, then switch to sleep mode to save energy. After all nodes send their information, each node records a table in their memory which contains the information of all its neighbors.

Step 3: Each node sends a packet which contains all its neighbors' information during its own time slot when Step 2is over. Then its neighbors can receive this packet and record the information in memory. The length of time slots in Steps 2and 3is predefined, thus when time is up, each node has sent its information before Initial Phase ended.

3.1.2.2 Algorithm for Neighbor Selection:

Input:- Number of Nodes with distance between them

Output:- Nodes with neighbors

Create node =node_id;

Set routing = AODV;

Set Channel = 802.11;

Set Initial Energy = $inienergy;

Set Residual Energy=$resenergy;

Set radio range=default;

If ((node in radio range) && (next hop! =Null)

```
{
  Capture data_load (node_all) ;
  Create node_Congifure (rreq, rrep, tsend, trecv, tdrop,,inienergy,residual energy);
      {
          pkt_type;
          Time;
      Tsend, trecv, tdrop, rrep, rreq ;
      }
  }
#calculation of  neighbor
distance=  (sqrt(pow(($nodex2-$nodex1),2)+pow(($nodey2-$nodey1),2)))
if {$d<300}
 {
if ($node2!=$node1)
 {
    Print node with neighbor position
 }
}
```

Example:

Here we consider a radio range of 500m in which we assume that 5 Nodes are present. The node should be in the radio range in order to find its neighbor node. We check this using

If ((node in radio range) && (next hop! =Null)

Then,

The node is configured using

Create node_Congifure (rreq, rrep, tsend, trecv, tdrop,,ini energy,residual energy);

Else,

Next node is considered.

For calculating the neighbor node we use a geometric distance vector

distance= (sqrt(pow(($nodex2-$nodex1),2)+pow(($nodey2-$nodey1),2)))

This is same as

Distance = $\sqrt{(Xnode2\text{-}Xnode1)}$

Where Xnode is the X co-ordinate of a node

Ynode is the Y co-ordinate of a node.

In order to print neighbor nodes the above calculate distance should be less than a default value, 300 in our case

if {$d<300}

{

if ($node2!=$node1)

Moreover, the two nodes should be different, and to check this we use a negation condition. Then,

We display the neighboring nodes

Else,

We check other nodes

For example node1 has a neighbor of node 7. node 1 x-pos[x1(1)=410] is 410 and y-pos[y1(1)=650] and node 7 x-pos [x2(7)=619] is 619 and y-pos[y2(7)=462] is 462. For distance calculation put the values in distance formula.

distance= (sqrt(pow(($node x2-$node x1),2)+pow(($node y2-$node y1),2)))

= (sqrt(pow((619-410),2)+pow((462-650),2)))

= (sqrt(pow((209),2)+pow((-188),2)))

=(sqrt(43681+35344))

= (sqrt(79025))

distance = 281

Node 7 distances is below 300, so it is a neighbor node of 1 as per condition.

After Initial Phase, GSTEB operates in rounds. For GSTEB and all other protocols mentioned, the "round" has the same meaning. In a round, the routing tree may need to be rebuilt and each sensor node generates a DATA_PKT that needs to be sent to BS. When BS receives the data of all sensor nodes, round ended. Round is not a real time measurement unit, but it reflects the ability for transmitting the collected data for sensors, so round is a suitable time measurement unit for WSN lifetime. Each round contains three phases, including Tree Constructing Phase, Self-Organized Data Collecting and Transmitting Phase, and Information Exchanging Phase.

3.1.2 Tree Constructing Phase

Within each round, GSTEB performs the following steps to build a routing tree. In Case there are some differences in the steps of routing tree constructing:

Step 1: BS assigns a node as root and broadcasts root ID and root coordinates to all sensor nodes. Only one node which communicates directly with BS can transmit all the data with the same length as its own, which results in much less energy consumption. In order to balance the network load for Case1, in each round, a node with the largest residual energy is chosen as root. The root collects the data of all sensors and transmits the fused data to BS over long distance

.**Step 2:** Each node tries to select a parent in its neighbors using EL and coordinates which are recorded in Table. The selection criteria for a sensor node, the distance between its parent node and the root should be shorter than that between itself and the root. Each node chooses a neighbor that satisfies criterion 1 and is the nearest to itself as its parent. And if the node can't find a neighbor which satisfies criterion 1, it selects the root as its parent. GSTEB chooses the nodes with more residual energy to transmit.

Step 3: Because every node chooses the parent from its neighbors and every node records its neighbors' information in Table, each node can know all its neighbors' parent nodes by computing, and it can also know all its child nodes. If a node has no child node, it defines itself as a leaf node, from which the data transmitting begins. Each packet sent to the parent nodes will be fused; the minimum energy consumption can be achieved if each node chooses the node nearest to it. But if all nodes choose their nearest neighbors, the network may not be able to build a tree. By this approach, a routing tree is constructed and some nodes still have the possibility of connecting to their nearest neighbors.

3.1.3 Self-Organized Data Collecting and Transmitting Phase

After the routing tree is constructed, each sensor node collects information to generate a DATA_PKT which needs to be transmitted to BS.TDMA and Frequency Hopping Spread Spectrum(FHSS) are both applied. This phase is divided into several TDMA time slots. In a time slot, only the leaf nodes try to send their DATA_PKTs. After a node receives all the data from its child nodes, this node itself serves as a leaf node and tries to send the fused data in the next time slot. Each node knows the ID of its parent node. In each time slot, in order to reduce communication interference, we apply FHSS in which each child node communicates with its parent node using the frequency hopping sequence determined by its parent node ID. Each TDMA time slot is divided into three segments as follows

Segment1: The first segment is used to check if there is communication interference for a parent node. In this segment, each leaf node sends a beacon which contains its ID to its parent node at the same time. Three situations may occur and they divide all the parent nodes into three kinds. For the first situation, if no leaf node needs to transmit data to the parent node in this time slot, it receives nothing. For the second situation, if more than one leaf node needs to transmit data to the parent node, it receives an incorrect beacon. For the third situation, if only one leaf node needs to transmit data to the parent node, it receives a correct beacon.

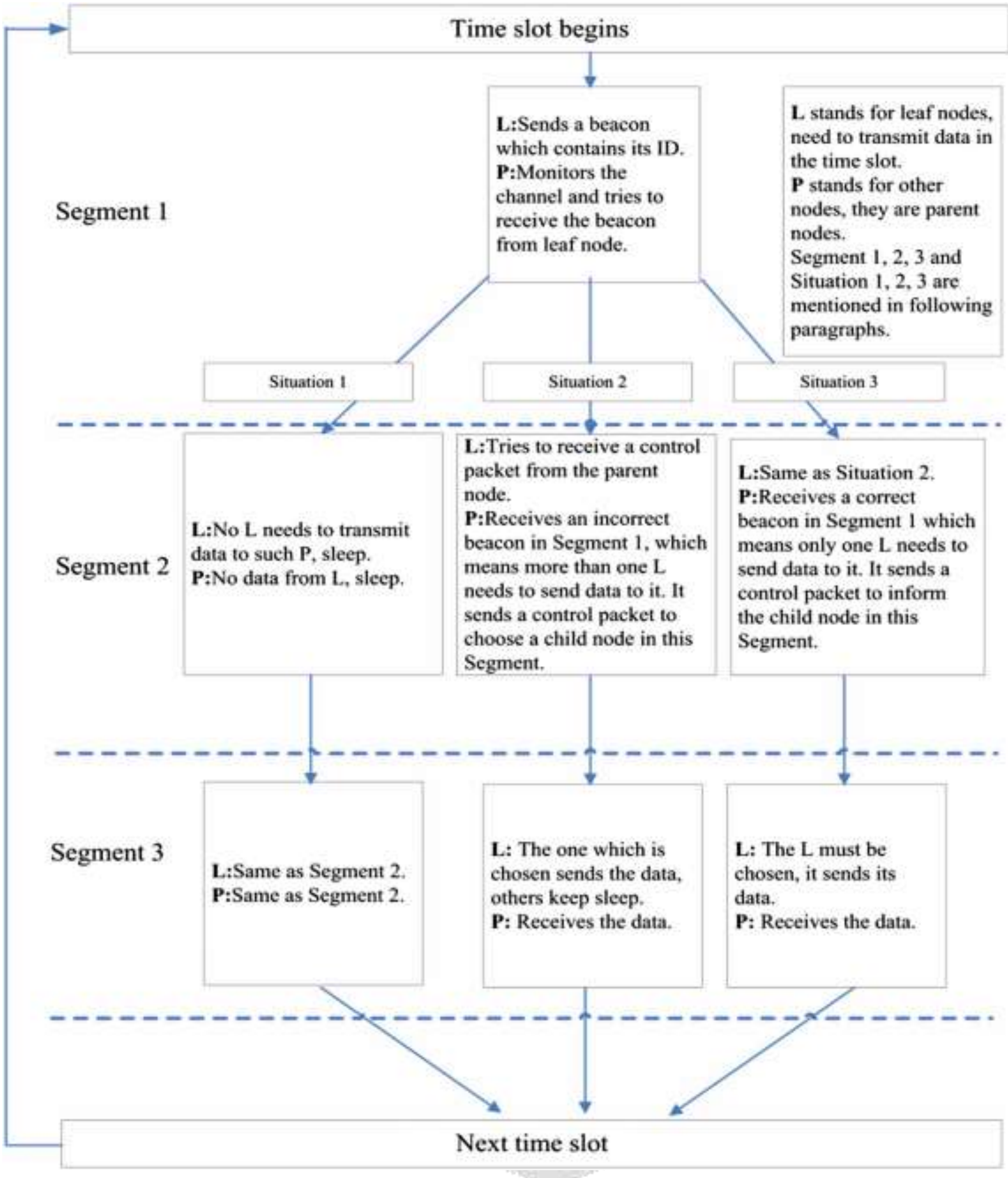

Fig 3: Process of a time slot in Self-Organized Data Collecting and Transmitting Phase L stands for leaf nodes, which need to transmit data P stands for other nodes, they are parent nodes

Segment 2: During the second segment, the leaf nodes which can transmit their data are confirmed. For the first situation, the parent node turns to sleep mode until next time slot starts. For the second situation, the parent node sends a control packet to all its child nodes. This control packet chooses one of its child nodes to transmit data in the next segment. For the third situation, the parent node sends a control packet to this leaf node.

This control packet tells this leaf node to transmit data in the next segment.

Segment 3: The permitted leaf nodes send their data to their parent nodes, while other leaf nodes turn to sleep mode. If all the leaf nodes try to transmit their data at the same time, the data messages sent to the same parent node may interfere with each other. By applying Frequency Division Multiple Access (FDMA) or Code Division Multiple Access (CDMA), the schedule generated under competition is able to avoid collisions. By using the control of BS, the energy waste can be reduced and thus the process may be much simpler. At the beginning of each round, the operation is also divided into several time slots. In the time slot, the node whose ID is I turns on its radio and receives the message from BS. BS uses the same approach to construct the routing tree in each round, and then BS tells sensor nodes when to send or receive the data. When BS receives all the data, the network will start the next phase.

3.1.4 Information Exchanging Phase

Each node needs to generate and transmit a DATA_PKT in each round, it may exhaust its energy and die. The dying of any sensor node can influence the topography. So the nodes that are going to die need to inform others. The process is also divided into time slots. In each time slot, the nodes whose energy is going to be exhausted will compute a random delay which makes only one node broadcast in this time slot. When the delay is ended, these nodes are trying to broadcast a packet to the whole network. While all other nodes are monitoring the channel, they will receive this packet and perform an ID check. Then they modify their tables. If no such packet is received in the time slot, the network will start the next round.

3.2 Clustering techniques

Dividing the sensor networks into small manageable units is called as clustering. Though the main reason behind the implementation of the clustering scheme is to improve the scalability of the network, it is an important factor in achieving energy efficient routing of data within the network. Apart from achieving scalability of the network it has more advantages like conserving communication bandwidth within the clusters, avoiding redundant message transfer between the sensor nodes, localizing energy efficient route setup within the clusters.

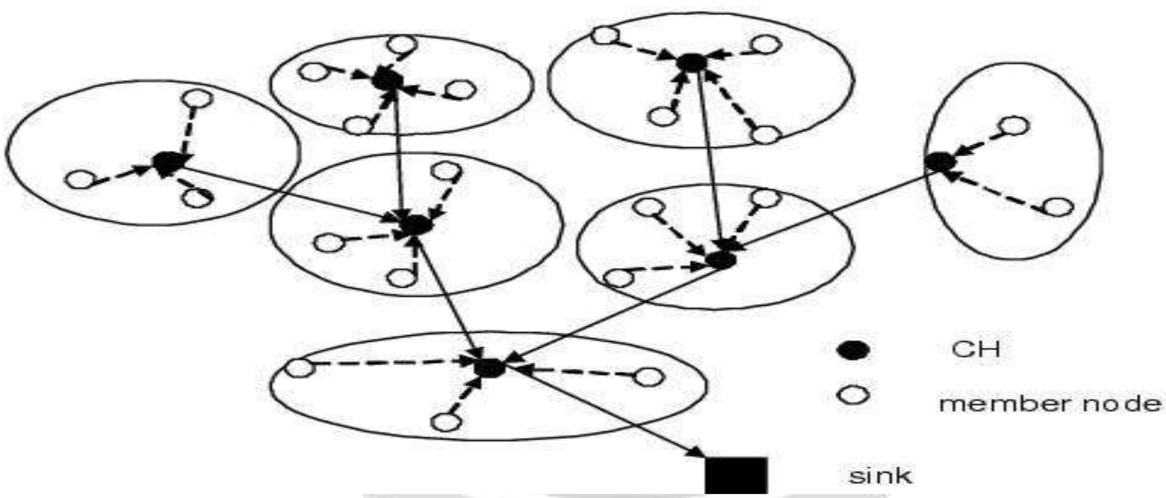

Fig 4: Clustering in WSN's

Step 1: Initialize network

Step 2: Deploy network randomly in predefined sensor field.

Step 3: Apply GSTEB to evaluate levels

Step 4: Apply clustering to develop cluster heads.

Step 5: Associate nodes with CHs.

Step 6: Evaluate and update energy consumption.

Step 7: Check whether all nodes become dead, if yes then show network life time and Return else continue to step 3.

3.2.1Cluster Head selection Algorithm:

In the cluster-based protocol, the nodes are organized into local clusters. Each cluster consists of one cluster head and no. of member nodes (the non-cluster-head nodes belong to the same cluster). All non-cluster-head nodes must transmit their data to the cluster head, while the cluster head must receive data from all the cluster members, perform aggregation on the data and transmit processed data to the remote base station. In the scenario where all nodes are energy-limited, it is necessary to choose appropriate cluster heads for the protocol.

.**3.2.2 Cluster Formation**

The cluster formation of the centralized protocol aims at balancing the energy load of cluster heads. Once the nodes have been selected as a cluster heads, they will invite other non-cluster head nodes in the network to join the clusters. Each cluster head broadcasts an invitation message using a non-persistent carrier-sense multiple access (CSMA) MAC protocol. The cluster head selection algorithm ensures that the selected cluster heads can cover all the nodes in the network with limited transmission range.

3.2.2.1 Algorithm for Cluster Formation:

Input:- Nodes with residual energy ,initial energy, distance

Output:- Nodes with cluster heads

```
# calculation of cluster heads

Create node =node_id;

        Set routing = AODV;

        Set Channel = 802.11;

        Set Initial Energy = $initial energy;

        Set Residual Energy=$residual energy;

        Set radio range=default;

     If ((node in radio range) && (next hop! =Null)

              {

                     Capture data_load (node_all) ;

Create node_Congifure (rreq, rrep, tsend, trecv, tdrop, initial energy, residual energy);

                     {

                            pkt_type;

                            Time;

                     Tsend, trecv, tdrop, rrep, rreq ;

                     }

              }

                     for (i=0;i<n;i++)

consume energy[i] = initial energy[i]-final energy[i]//computer consume energy
```

```
total energy[i] = consume energy[i]
if(max energy < consume energy[i])
        {
            Max energy = consume energy[i]
            node_id=i;
        }
    if (dist<300 && max energy[node_id] >energy neighbour[node_id])
        {
            Max energy node is cluster head ;
        }
```

In above Cluster Head Formation Algorithm, Researcher first

1. Create a node,
2. Set channel,
3. Set Initial Energy,
4. Set Residual Energy,
5. Set Radio Range

Here Researcher give a condition If ((node in radio range) && (next hop!=Null)

If condition is satisfied then Capture all the node data and cluster formation is proceed. Here Researcher calculates Consume Energy of nodes and total energy of nodes using below formulae.

Consume energy[i] = final energy[i] –initial energy[i]

Total energy[i] = consume energy[i]

then Researcher find out maximum energy of node using below condition.

```
if(max energy[i] < consume energy[i])
{
Max energy = consume energy[i]
  node_id = i;
}
```

At last Researcher find out CH of cluster using below condition

```
if (dist<300 && max energy[node_id]> energy neighbour[node_id])
{
Max energy node is cluster head ;
```

}

For example node 0 has a neighbor of node 19. node 0 x-pos[x1(0)=549] is 549 and y-pos[y1(0)=125] and node 19 x-pos [x2(19)=622] is 622and y-pos[y2(19)=180] is 180. For distance calculation put the values in distance formula.

distance= (sqrt(pow(($node x2-$node x1),2)+pow(($node y2-$node y1),2)))

= (sqrt(pow((622-549),2)+pow((180-125),2)))

= (sqrt(pow((73),2)+pow((55),2)))

=(sqrt(5329+3025))

= (sqrt(8354))

distance = 91

Node 19 distances is below 300, so it is a neighbor node of 0 as per condition.

Researcher calculates consume energy of node 6 using formula

Consume energy[6] = final energy[6] – initial energy[6]

= 51.9567-38.0433

= 13.9134 Joule

Total energy [6] = consume energy [6]

= 13.9134 Joule

As per discussion the node 0 energy is highest energy in a cluster, so node 0 is a Cluster Head of that cluster. In order to reduce the interference of transmission, all cluster heads load a random time delayt1, after which they broadcast the invitation messages. The random time delay t1 should be set appropriately to ensure there is enough interval time between broadcasting an invitation by two random cluster heads without incurring considerable time delay of the system.

4. CONCLUSION

In a project, two definitions of network lifetime and two extreme cases of data fusion are proposed. The simulations show that when the data collected by sensors is strongly correlative, GSTEB outperforms LEACH, PEGASIS,TREEPSI and TBC. Because GSTEB is a self-organized protocol, it only consumes a small amount of energy in each round to change the topography for the purpose of balancing the energy consumption. All the leaf nodes can transmit data in the same TDMA time slot so that the transmitting delay is short. When lifetime is defined as the time from the start of the network operation to the death of the first node in the network, GSTEB prolongs the lifetime by 100% to 300% compared with PEGASIS. When the data collected by sensors cannot be fused, GSTEB offers another simple approach to balancing the network load. In fact, it is difficult to distribute the load evenly on all nodes in such a case. Even though GSTEB needs BS to compute the topography, which leads to an increase in energy waste and a longer delay.

5. REFERENCES

[1] W. B. Heinzelman, A. Chandrakasan, and H. Balakrishanan, “An application- specific protocol architecture for wireless micro sensor networks, ”IEEE Trans. Wireless Commun, vol. 1, no. 4, pp. 660670,Oct.2002.

[2] O. Younis and S. Fahmy, “HEED: A hybrid, energy-efficient, distributed M clustering approach for ad hoc sensor networks,” IEEE Trans. Mobile Computing, vol. 3, no. 4, pp. 660–669, 2004.

[3] S. Lindsey and C. Raghavendra, “Pegasis: Power-efficient gathering in sensor information systems,” in Proc. IEEE Aerospace Conf., 2002,vol. 3, pp. 1125–1130.

[4] H. O. Tan and I. Korpeoglu, “Power efficient data gathering and aggregation in wireless sensor networks,” SIGMOD Rec., vol. 32, no. 4, pp.66–71, 2003.

[5] S. S. Satapathy and N. Sarma, “TREEPSI: Tree based energy efficient protocol for sensor information,” in Proc. IFIP Int. Conf., Apr. 2006,pp. 11–13.

[6] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: A survey,” Computer Netw.s, vol. 38, no. 4, pp.393–422, 2002.

[7] R. Szewczyk, J. Polastre, A. Mainwaring, and D. Culler, “Lessons from sensor network expedition,” in Proc. 1st European Workshop on Wireless Sensor Networks EWSN ‘04, Germany, Jan. 19-21, 2004.

[8] W. Liang and Y. Liu, “Online data gathering for maximizing network lifetime in sensor networks,” IEEE Trans Mobile Computing, vol. 6,no. 1, pp. 2–11, 2007.

[9] J. H. Chang and L. Tassiulas, “Energy conserving routing in wireless ad hoc networks,” in Proc. IEEE INFOCOM, 2000, vol. 1, pp. 22–31.

[10] G. Mankar and S. T. Bodkhe, “Traffic aware energy efficient routing protocol,” in Proc. 3rdICECT, 2011, vol. 6, pp. 316–320.

[11] N. Tabassum, Q. E. K. Mamun, and Y. Urano, “COSEN: A chain oriented sensor network for efficient data collection,” in Proc. IEEE ITCC, Apr. 2006, pp. 262–267.

[12] M. Liu, J. Cao, G. Chen, and X. Wang, “An energy-aware routing protocol in wireless sensor networks,” Sensors, vol. 9, pp. 445–462, 2009.

[13] K. T . Kim and H. Y. Youn, “Tree-Based Clustering(TBC) for energy efficient wireless sensor networks,” in Proc. AINA 2010, 2010, pp.680–685.

[14] I.F. Akyildiz, W. Su, A power aware enhanced routing(PAER) protocol for sensor networks, Georgia Tech Report, January 2002, submitted for publication.

[15] S. Al-Omari, W. Shi, and C. J. Miller. Sesame: A sensor system accessing and monitoring environment. Technical Report MIST-TR-2004-018, Wayne State University, Nov. 2004.

[16] W. Steven Conner, Jasmeet Chhabra, Mark Yarvis, and Lakshman Krishnamurthy. Experimental evaluation of synchronization and topology control for in-building sensor network applications. In WSNA '03: Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications, pages 38-49,NY, USA,2003. ACM Press.

[17] Loren Schwiebert, Sandeep K.S. Gupta, and Jennifer Weinmann. Research challenges in wireless networks of biomedical sensors. In Mobi Com '01: Proceedings of the 7thannual international conference on Mobile computing and networking, pages 151-165, New York, NY, USA, 2001. ACM Press.

[18] W. Shi and C. Miller. Waste containment system monitoring using wireless sensor networks. Technical Report MIST-TR-2004-009, Wayne State University, March2004.

[19] Li Li, Joseph Y. Halpern, Paramvir Bahl, Yi-Min Wang, and Roger Watten hofer. Acone based distributed topology-control algorithm for wireless multi-hop networks. IEEE/ACM Trans. Netw., 13(1):147-159, 2005.