International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:7, No:7, 2013

Clustering in WSN Based on Minimum Spanning Tree Using Divide and Conquer Approach

Uttam Vijay and Nitin Gupta

Abstract—Due to heavy energy constraints in WSNs clustering is an efficient way to manage the energy in sensors. There are many methods already proposed in the area of clustering and research is still going on to make clustering more energy efficient. In our paper we are proposing a minimum spanning tree based clustering using divide and conquer approach. The MST based clustering was first proposed in 1970's for large databases. Here we are taking divide and conquer approach and implementing it for wireless sensor networks with the constraints attached to the sensor networks. This Divide and conquer approach is implemented in a way that we don't have to construct the whole MST before clustering but we just find the edge which will be the part of the MST to a corresponding graph and divide the graph in clusters there itself if that edge from the graph can be removed judging on certain constraints and hence saving lot of computation.

Keywords—Algorithm, Clustering, Edge-Weighted Graph, Weighted-LEACH.

I. Introduction

A sensor is a device that can sense specific physical parameters of a system or a region of interest, convert the sensed data into electrical signals, and transmit the signals to a base station (BS) or sink using wireless radio. The collection of sensor to perform a specific function is known as Wireless Sensor Network [1], [2]. These WSNs have some limitations compared to other traditional networks and some of them are heavy processing, storage and power constraints. Generally, sensor node doesn't have power and communication range to directly propagate the message to the base station. Hence, a sensor node not only senses and sends its own data but also acts as router and propagates the data of its neighbours.

Clustering is the process of partitioning data set into subsets known as clusters such that data in each subset share some common properties. Work on clustering has been going on since 1970 [3]. Node clustering algorithms are usually performed in two phases: node -clustering setup and clustering maintenance. In the node - clustering setup phase, cluster heads are chosen among the nodes in the network. After electing the cluster heads, other nodes affiliated with the cluster heads would form the clusters. Nodes that are not cluster heads are called ordinary nodes.

Uttam Vijay is a postgraduate student of National Institute of Technology Hamirpur, Himachal Pradesh, India in computer science and Engineering (e-mail: -contactuttam.uttam@gmail.com).

Nitin Gupta is with National Institute of Technology, Hamirpur, Himachal Pradesh, India as Assistant Professor in computer science and Engineering Department (e-mail:-nitin@nith.ac.in, nitin3041@gmail.com).

Clustering has many advantages. After clustering aggregated data is transferred to data sink and this reduces unnecessary and redundant data transfer. It also reduces number of nodes taking part in transmission and thus helps in useful energy consumption. Furthermore it allows scalability for large number of nodes, reduces communication overhead and allows efficient use of resources in WSNs.

The main emphasis of the paper is to find a clustering method which is energy efficient. We are implementing MST based clustering using divide and conquer approach which was proposed for large databases [4] and see if it works for wireless senor network considering the constraints faced by it such as dense node deployment, heavy energy, memory and processing constraints, changing topology, error prone sensors etc. In the paper we are trying to give clustering algorithm which minimizes the energy dissipation of the network, increases the network life time, give a better balanced clusters and cluster heads that are correctly distributed.

Algorithm basically works on idea that removing an edge from MST (preferably longest) will lead to two clusters. So we need to create a MST and then remove the longest edge from that to create two clusters. Here we are using divide and conquer approach which allows us the advantage that we don't need to create complete MST, we just identify the edge which will be the part of the corresponding MST of that graph created by the sensors for removal and hence dividing the graph in sub graph. We stop our process once the largest cluster has less number of nodes than required, even if subgraph contain cycles which saves a lot of computation.

There are many works already done and many others going in field of clustering for WSNs. Even minimum spanning tree based approach is quite a favourite way among the researchers We will compare our work to some existing MST Based algorithm like MST, MEMD, LMST, MCTC, DRNG algorithms in simulation. In Sections II and III we will discuss the proposed scheme in detail with an example to present our work. In Section IV we will compare our algorithm on certain criteria by simulating on omnet++ and give conclusion.

II. Proposed Scheme

This clustering algorithm is based on implementing MST using divide and conquer approach where we will pick the edge from a graph which will be the part of corresponding MST without actually completely creating the MST. WSN have constraints of their own and we are trying to implement this approach under the influence of those constraints.

A. We can Summarize the Approach in Four Steps

i. Assign the weight to each available path between the nodes and create an edge-weighed graph.
ii. Apply reverse delete algorithm to form MST with divide and conquer approach and thus creating clusters
iii. We will run the algorithm until some terminating condition is met
iv. Cluster Heads are chosen once cluster is formed which is different approach from all other clustering algorithms used in WSNs

B. Assigning the weight

We use first order radio power model given in [5]. Energy required to transmit data packet of size 'l' from node i to j

$$T_{ij}=l.e_{elec}+ l.e_{amp} * d^2_{ij} \qquad (1)$$

where e_{elec} is energy required to run the transmitter and e_{amp} is energy required to run transmitter amplifier, d_{ij} is the distance between node I and j.

Energy required to receive l bit for any node is given by

$$R_j= e_{elec}.l \qquad (2)$$

So actual weight assignment to provide k bit data is:

$$w_{ij}(k) = \min\{E_i - T_{ij}(k), E_j - R_j(k)\} \qquad (3)$$

where E_i the current energy of node, $T_{ij}(k)$ is the energy required to transmit k bits of packet from node i and j, $R_j(k)$ denotes energy consumed to receive the k bit packet by node j. Here e_{elec} and e_{amp} is constant because its value will be predefined at beginning. So energy required mainly depends on distance between two nodes. So we can assign weight of the edge as the distance between the nodes.

C. Applying the Algorithm

This Clustering Algorithm is based on fact that if we remove an edge i.e. in most cases the longest one from the MST we will have two clusters grouping. In the real world when we consider WSNs due to certain outliers' longest edge cannot be the only edge removing criteria. So we have to define some inconsistency measures and look for the edges satisfying these inconsistency measures. Inconsistency measures are:

i. Threshold value: {mean(w)+standard deviation(σ)}[6]
ii. Cluster size:-Maximum(M) and Minimum (m) cluster size [7]

Algorithm is applied with initial initialization by considering each node as vertex of a graph. Then weight is assigned to each available path. It is not necessary to assign weight to each path but we should have a spanning tree so that we have a path to reach any node of the sensor, then weight is assigned to it according to the paths. Now our main focus is to find the longest edge which can be the part of MST to be created using reverse delete algorithm. So we start with the longest edge and see if removing this edge disconnects the graph or not. If it is not disconnecting the graph we simply delete it and move to next longest edge. Once we found the removing longest edge of tree is disconnecting the graph we declare that edge as inconsistent edge candidate. Now whether to remove this longest edge or not to form the cluster depends on whether this edge is inconsistent or not. We check this edge on our predefined inconsistency measures and if it is found inconsistent we remove it from MST to form the cluster.

D. Algorithm Works in the Following Manner

i. Assign the weights to every edge available in the network
ii. Look for the Inconsistent edge candidate by starting with highest weighted edge.
iii. Take the longest edge and see if removing this edge will disconnect the graph, then it is Inconsistent edge candidate.
iv. Once this edge is confirmed as Inconsistent edge candidate we need to check inconsistency measures to actually remove it to form cluster.

E. Inconsistent Edge Removing and Algorithm Terminating Condition:

i. Calculate the threshold value by adding the mean and standard deviation of the weight of edges and remove those edges which have value larger than threshold.
ii. Estimate the maximum and minimum number of nodes in a cluster and form cluster only if size (resultant clusters)> smallest cluster size
iii. Terminate the algorithm when the size of largest cluster becomes smaller than the estimated maximum number of nodes.

III. Example

We take an example of WSN with 4*4=16 nodes with some random distance between the nodes:

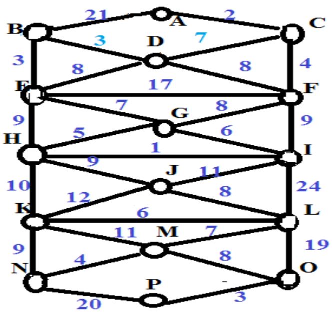

Fig. 1 Graphical Representation of Sensor Network

We need maximum cluster size (M) and minimum cluster size (m). Ideal cluster size given is integer closer to $\sqrt[3]{2 * N}$ [8] in network of N*N nodes. But we need both maximum and minimum cluster size.

So we take:

$$m= \sqrt[3]{2 * 4} \text{ (ideal cluster size)} =2, \quad M=2*m=4$$

because if we cluster M sized network we need to have at least two m sized clusters.

When we consider WSN which in most cases is very closely connected, we have proposed a threshold value which is sum of mean and standard deviation of the edge weight and removing and not considering all the edges having the value more than this threshold because in a densely connected graph there is very less chance of that removing a edge with weight more than the proposed threshold will disconnect the graph. Even if in some cases any edge having value greater than threshold is disconnecting the graph we will ignore it and won't consider it as inconsistent edge candidate. This is because there is a minimal chance of this happening and also if it happens the nodes disconnected will become the part of any cluster on further applying the scheme proposed. Hence we can prevent lot of computation by not considering that edges with the weight over threshold as our inconsistent edge candidate in the MST. For our graph

Mean(w)=(21+2+7+3+3+4+8+8+17+6+8+9+9+5+7+1+9+11 +10+24+12+8+6+11+7+9+18+4+8+20+3)/31=279/31=9

StandardDeviation=

$$\sqrt{(21-9)^2+(2-9)^2+(7-9)^2\ldots\ldots+(20-9)^2/31}$$

$$=\sqrt{975/_{31}}=\sqrt{31.45}=5.6$$

Threshold=9+5.6=14.6. So, W_e>threshold will be removed. So AB, EF, IL, LO, NP are removed then we have new graph:

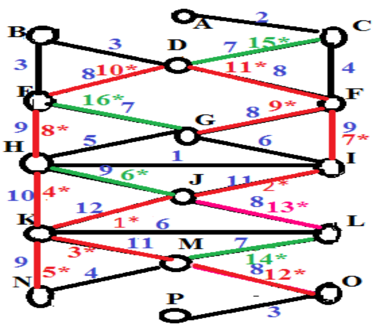

Fig. 2 Sensor Network after considering Threshold

- Step1:**KJ** (not disconnecting graph so removed)
- Step2:**JI**(not disconnecting graph so removed)
- Step3:**KM** (not disconnecting graph so removed)
- Step4:**HK**(not disconnecting graph so removed)
- Step5:**KN** (not disconnecting graph so removed)
- Step6:**HJ** (Inconsistent edge so Clustering done)
- Step7:**FI** (not disconnecting graph so removed)
- Step8:**EH** (not disconnecting graph so removed)
- Step9:**GF** (not disconnecting graph so removed)
- Step10:**DE** (not disconnecting graph so removed)
- Step11:**DF** (not disconnecting graph so removed)
- Step12:**OM** (Inconsistent edge so cluster formed)
- Step13:**JL** (inconsistent edge Candidate but ignored as size(smaller cluster)<m)
- Step14:**LM** (inconsistent edge so cluster formed)
- Step15:**CD** (Inconsistent edge so cluster formed)
- Step16:**EG** (Inconsistent edge so cluster formed)

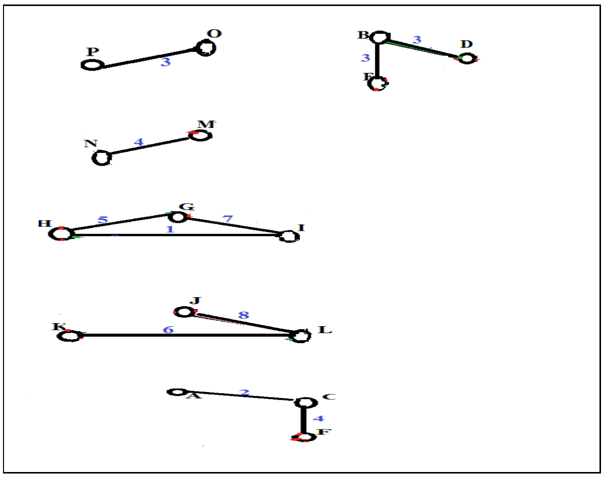

Fig. 3 Clustered Sensor Network

We can stop the process here because of size of largest cluster is 3<M so need for further clustering. There is no need for forming complete MST, even if there is cycle in one cluster, it doesn't matter. Process is completed in only 16 steps but in any other MST based clustering, it requires more steps because first complete MST will be formed and only then clustering starts.

Pseudocode for Algorithm:

```
/*Data structures and variables*/
S= set of nodes
E=Edges in the graph
C=Cluster formed
We=Weight of the edge E
w=Mean of the weight of the edges
σ=Standard Deviation of the weights of the  edges
M=Maximum size of cluster
m=minimum size of cluster

/*Algorithm*/
Sort Edges in descending order
index i->0;
  While i<Size(E);
if We[i]>σ+w;
{
Delete E[i]
  }
else
  {
define T<-E[i];
delete E[i];
if (isConnected(T.v1,T.v2);==Fail)
{
E[i]<-T;
}
bool a= checkInconsistent(E[i]);
```

```
if(a==true)
createClusters(E[i]);
checkExit(E[i]);
i->i+1;
}
```

```
/*Pseudo code for checking whether the edge is Inconsistent or not:*/
checkInconsistent(E[i])
{
define index j->0;
  C[j]=All nodes connected to E[i].v1;
C[j+1]=All nodes connected E[i].v2;
if (size(C[j] or size(C[j+1])<m)
/*here we are looking for size of smaller cluster*/
return (false);
else
return(true);
}
```

```
/*Pseudo code for checking terminating condition:*/

checkExit(E[i]);
{
define index j->0;
  C[j]=all nodes connected to E[i].v1;
C[j+1]=all nodes connected E[i].v2;
if (size(C[j] and size(C[j+1])<M)
/*here we are looking for size of larger cluster*/
stop the algorithm:
else
continue:
}
```

```
/*Pseudo code for checking connectivity of graph*/
isConnected(V1,V2)
{
  S->φ;
   D-> φ;
   S->  S union V1;
   While (S is not empty);
   {
      u- >extract(S);
      D-> D union {u};
for each vertex v Є adj[u];
if(v==V2)
return (success)
If(v not belong to D)
{
D - >D union {v}
}
}
return (fail);
}
```

A. Cluster Head Selection

Cluster Heads cannot be chosen before the clusters are formed as we have to consider all nodes to be a part of graph and only then can apply our approach. Algorithm is applied on whole network with every node considered as ordinary node at start.

Once cluster are formed cluster heads can be chosen as in LEACH but with some improvement by Weighted Spanning Tree LEACH (WST-Leach) [9] which is an improvement for LEACH. In LEACH cluster head selection is completely random but in the paper it is proposed to consider three factors. If a node has more energy left it should be preferred as CH, if a node has more neighbour nodes near it, it should also be given preference and if a node is near to base station, it must also be a factor for cluster head selection. So considering these three factors we will select our CH as

T(n)=0   if node n does not belong to G,

where G is a collection of non-CHs in this round.

Otherwise

$$T(n)= \{\frac{P}{1-P*(r \bmod (1/_P))} * \{W1 * \frac{S(n).E}{E0} + w2 * \frac{S(n).Nb}{p*N} + w3 * \frac{1}{S(n).T0\ Bs}\} \quad (4)$$

where S (n).E is the remaining energy of node n; *E*0 is the initial energy; N is the total number of nodes; S(n).Nb is the neighbour numbers of node n within a radius R; S (n).ToBs is the distance between node n and the BS; w1,w2,w3 are coefficients respectively

IV. Simulation and Future Work

Algorithm has been theoretically compared with other similar algorithms like MST, DRNG, LMST [10], MCTC [11], MEMD [12] and Original Topology on the following basis proposed in [10]

- → Node Degree:-It is the statistical link status of the node. Clustering with smaller average node degree will produce better performance due to reduction of contention and inference.
- → Node Power Radius ->It is the transmission range of the node. Smaller the range better will be the performance because it implies less power consumption.
- → Average Link Length-> Average link length is defined as the ratio between the summations of all links, *i.e.*, the distance between sensor node and sensor node, and link counts after topology Control.
- → Network Life Time Span->Network is active until the all node have energy left. Once the residual energy of last node is zero networks is dead.

It is been derived by us theoretically considering lesser number of nodes that our algorithm may not produce smaller node degree compared to other algorithms because in many cases we are not completely eliminating the cycles but this inefficiency is more than covered by producing smaller node power radius and average link length and larger network life time span. We have done it theoretically and in our future work we are ready to apply the algorithm with network Simulator Omnet++.. We will take 400 sensor nodes uniformly distributed in area of 1000*1000 m. Maximal transmission range will be 100 units for all sensors .Base station will be located in the centre (500,500). Every sensor node has 1 j initial power will be given as e_{elec} =50nj/bit and e_{amp}=100pj/bit.m^2 and compare all the algorithms on all four factors and give a complete experimental results with proper graphs.

REFERENCES

[1] F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci. "A Survey on Sensor Networks" IEEE communication Magazine, Aug 2002:102-114

[2] Wireless Sensor Network(A Networking Perspective) Jun Jheng and Abbas Jamalipour [Book]

[3] C.T. Zahn, "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters," IEEE Trans. Computers, vol. 20, no. 1, pp. 68-86, Jan. 1971

[4] A Divide-and-Conquer Approach for Minimum Spanning Tree-Based Clustering Xiaochun Wang, Xiali Wang, and D. Mitchell Wilkes, Member, IEEE Transactions On Knowledge And Data Engineering, Vol. 21, No. 7, July 2009

[5] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "Energy Efficient Communication Protocol for Wireless Microsensor Networks," in Proceedings of 33rd Hawaii Int'l. Conf. Sys. Sci., 2000.

[6] M. Laszlo and S. Mukherjee, "Minimum Spanning Tree Partitioning Algorithm for Micro aggregation," IEEE Trans. Knowledge and Data Eng., vol. 17, no. 7, pp. 902-911, July 2005

[7] O. Grygorash, Y. Zhou, and Z. Jorgensen, "Minimum Spanning Tree-Based Clustering Algorithms," Proc. IEEE Int'l Conf. Tool with Artificial Intelligence, pp. 73-81, 2006

[8] Wang, D.: An Energy-Efficient Cluster Head Assignment Scheme for Hierarchical Wireless Sensor Networks. Int. Journal of Wireless Inf. Networks 15(2) (2008) 61-71

[9] Weighted Spanning Tree Clustering Routing Algorithm based on LEACH Huazhong Zhang, PeipeiChe2nd International Conference on Future Computer and Communication 2010 volume 2 page 223-227

[10] N. Li, J. C. Hou, and L. Sha. Design and Analysis of an MST-based Topology control algorithm. IEEE Infocom' 2003, San Francisco, USA, 2003, 1702–1712

[11] MST-Based Clustering Topology Control Algorithm ForWireless Sensor Networks1Cai Wenyu Zhang Meiyan (School of Electronics and Information, Hangzhou Dianzi University, Hangzhou 310018, China)Vol.27 No.3 Journal Of Electronics (China) May 2010

[12] An Energy Effective Adaptive Spatial Sampling Algorithm for Wireless Sensor Networks Zhang Meiyan+, ZhengXiaodan 2012 International Conference on Computer Technology and Science (ICCTS 2012) IPCSIT vol. 47 (2012) © (2012) IACSIT Press, Singapore Doi: 10.7763/Ipcsit.2012.V47.61