Energy Efficient variant of Enhanced Tree Routing Protocol for HSNs

Kanwalinderjit Gagneja
Dept. of Computer Science
NDSU
Fargo, USA
E-mail: kanwalinder.gagneja@ndsu.edu

Dr. Kendall Nygard
Dept. of Computer Science
NDSU
Fargo, USA
E-mail: Kendall.Nygard@ndsu.edu

Abstract

The sensor nodes' functioning is mainly dependent upon battery power. Sensor nodes route data in the network through multiple hops. So to conserve battery power it is very important to optimally configure the nodes in the network so as to minimize the hop count. In a two tier heterogeneous sensor network two different kinds of nodes are installed, the high-end nodes and the low-end nodes. The Voronoi partitioning technique generates a topology where all the nodes in the network are partitioned into non-overlapping and distributed clusters. One cluster consists of one high end node and multiple low end nodes. ETR (Enhanced Tree Routing) technique when applied to this topology to route the data taking care of asymmetric links, the overall performance of ETR algorithm increases. By simulation results, we show that Voronoi cluster based ETR routing performs better than its existing ETR routing variant. Furthermore, simulations performed show that Voronoi cluster based ETR routing uses less energy, shows lesser network delays and the throughput of the network is considerably increased.

Keywords: Voronoi; Heterogeneous sensor networks; simulation; two tier; cluster.

I. INTRODUCTION

Sensor networks always have issues concerning coverage, power, computational capacity and memory, etc. In sensor networks the number of nodes is generally very large, so for taking care of all the issues related to sensor networks, it is mandatory to use divide and conquer technique. First, if we use two tiers heterogeneous sensor network and second if the area of interest is partitioned into smaller regions or clusters then the overall network operations, transportation cost, energy utilization and on the whole computation of the network could be reduced. Meta-heuristics could be used for making smaller regions or clusters, with which the overall cost of the data routing can be minimized by partitioning a bigger set into subsets, especially in those cases where the cost associated with a subset is linear to its size.

We know that the dividing n nodes into k clusters are a partitioning problem and there are roughly k^n ways of assigning nodes to clusters. Clustering problem is solved by grouping similar elements together [7]. To find a global solution for this problem in realistic time, different meta-heuristics are used frequently; one of those is Voronoi diagrams. Suppose there are n nodes in a plane and n=2. So there are only $O(k^2)$ partitions in the plane, instead of 2^k ways of dividing given set of nodes into two clusters. In a d dimension, for a given set of n nodes there are n^{kd} Voronoi clusters. This form is polynomial instead of exponential with respect to n. So we realize that only Voronoi clustering is one which gives non-overlapping and distributed clusters. So the meta-heuristics used in this paper for partitioning is Voronoi clustering.

So first, we are partitioning a bigger region into Voronoi clusters and second using ETR routing technique to minimize the energy utilization of the network, as energy used in hop count is a significant aspect. Efficient routes also give an adversary less time to seize the data, leading to increased security of the network. The network has large number of low end sensor nodes than high end nodes. The transmission power of low end and high end nodes is adjusted to a threshold level so that low end nodes could reach neighbors up to certain distance and high end nodes could reach all the low end nodes in the cluster as well as some neighboring high ends nodes too, so that they can route the data to the base station.. By setting radio range of nodes to some threshold level, a node could communicate to required numbers of neighbors, ensuring reduced breach of information and lesser energy consumption [10]. In our experimentations, the ETR routing algorithm is first executed for intra cluster routing for the low-end nodes within one cluster, then it is executed for inter cluster routing for the high-end nodes. The ETR algorithm is resilient, so if at some time any node gets disabled, the algorithm finds another route.

II. VORONOI CLUSTERING

In computational geometry Voronoi diagram uses Euclidean distance to partition a set of points in x-y plane. In a layman's language a given number of points in a set are known as sites. The Voronoi diagram will partition these points given in x-y plane into subsets of points falling in one of the nearest site.

Mathematically the Voronoi diagram could be defined as, if the area of interest, R, is 2-dimensional, have P low end nodes, and C high end nodes. Where distance between any low end node, $p_{i,}$ and high end nodes, c_j, is given by $d(p_i, c_j)$. Then Voronoi diagram $V(P,C, d(p_i, c_j))$ assigns each low end node p_i to high end node c_j with least distance $d(p_i, c_j)$ [8]. This way low end and high end nodes are partitioned into clusters. Figure 1(a) is showing a 2-dimensional plane with high end nodes and fig. 1(b) shows the Voronoi clusters generated from the fig. 1(a).

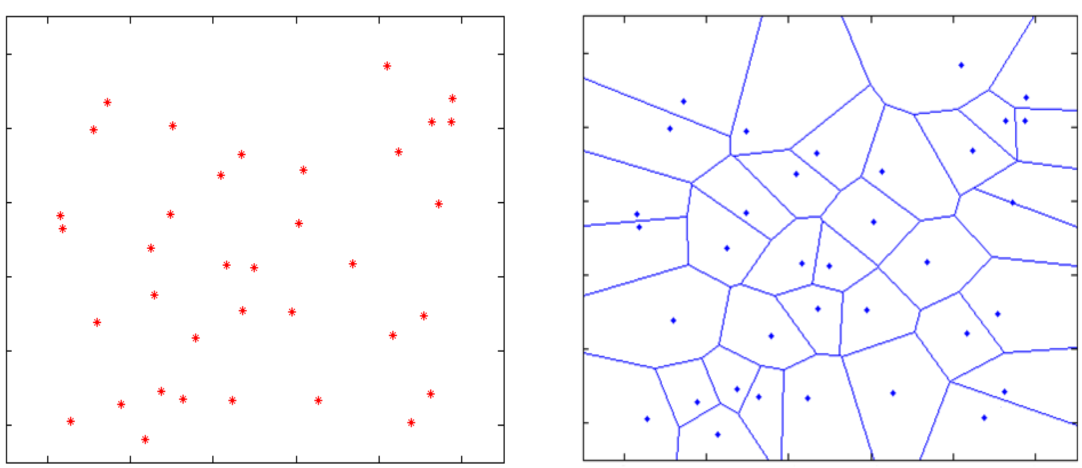

Figure 1 (a) points on 2 dimensional plane (b) Voronoi clusters generated for figure 1(a)

So the Voronoi diagram is a topology for the heterogeneous sensor networks, or is collection of polygons of different order, shapes and sizes or it is storing geometric pattern of vertices, edges, and neighbors with in a radio range. When we are using distance function $d(p_i, c_j)$, all the Voronoi cluster are convex; and polygonal. Every cluster is meeting point of the partial planes. Each cluster has center on the convex hull. The Voronoi edges form the cluster boundaries. Every edge intersects the line joining 2 centers. Every edge encloses all equidistant points from two centers. The end points of edges are Voronoi vertices. A vertex is an endpoint of the three edges and is equidistant from three centers. A vertex makes center of the circle all the way through three centers and that vertex circle is empty means it encloses no additional center.

III. PROBABILITY OF ONE HIGH END NODE PER CLUSTER

It is essential that all the clusters should have one high end sensor for the effective working of our methodology as nearly whole area of interest under HSN would be covered by all high end sensors with their threshold transmission range. In this randomly deployed network, where there are fewer high end nodes and large low end nodes. The network is partitioned into numerous Voronoi clusters governed by one high end node with T threshold transmission range. Suppose the area of interest is divided into k clusters and n high end sensors are in the network. Let a set of all clusters is represented by F. To find the probability with which only one high end node could be present in the cluster is given by the VC-dimension Theorem [5].

The VC-dimension Theorem: Let X_i represents space, and F represents a set of finite VC-dimension, VCd(F), where there are clusters which are independent and identically distributed having randomly deployed high end nodes with probability measure of P_x on space X_i . Let A be an element of F, such that $A \in F$, then for each positive value of δ, ε, the probability with which a high end node could be present in the cluster is given by

$$\lim_{k\to n} P_x \left(\sup_{A\in F} \left|1/n\left(\sum_{i=1}^{n} I(.X_i\in F)\right)- P_x(A)\right| \le \varepsilon\right) = P \qquad (1)$$

Where $P > 1-\delta$ and n is greater than max $\{ (8x\ VCd(F)x\log(16e/\varepsilon))/\varepsilon, (4/\varepsilon)x\log(2/\delta)\}$

The $P_x(A)$ is the average density of high end nodes per cluster = n/k, and we have $\delta(n)=\varepsilon(n)= 20x\log(n)/n$. So the probability

$$P_x(A)=n/k \qquad (2)$$

Replacing (2) in (1) will give us

$$\lim_{k\to n} P_x \left(\sup_{A\in F} \left|(1/n)\left(\sum_{i=1}^{n} I(.X_i\in F)\right)- n/k\right| \le \varepsilon\right) = P \quad \Rightarrow$$

$$\lim_{k\to n} P_x \left(\sup_{A\in F} \left|(1/n)(\text{Number of High end nodes in A})- n/k\right| \le \varepsilon\right) = P \quad \Rightarrow$$

$$P_x (\text{Number of High end nodes in A} \ge n((n/k)- \varepsilon)) = P > 1-\delta$$

In fig. 2 we plot the curve for, P, probability of number of High end nodes in one cluster= n((n/k)- ε)) = n(2-20xlog (n)/n) when n=k, means when there is one high end node per cluster or a cluster has only one cluster head. From the curve we can analyze that the network should have a minimum of 11 clusters to have at least one cluster head. When network size is enough large, that is, n is quite large or n>11, then the network has very good probability, (1-20xlog (n)/n), that each cluster will have only one cluster head and this probability approaches 1 if n approaches to infinity.

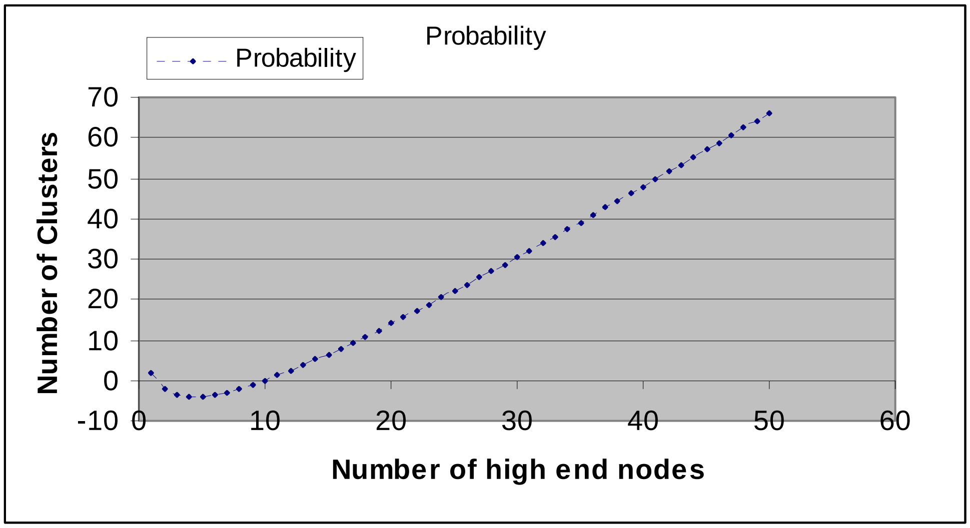

Figure 2: The Probability P for number of high end nodes per cluster

Figure 3 shows the curve for 1-δ(n) and from the curve we analyze that when n is greater than 30 then 1-δ(n) is positive. As the value of n is increased the value of 1-δ(n) approaches to 1 because 1-δ(n)= (1-20xlog(n)/n) and as n approaches to infinity (1-20xlog (n)/n) also approaches to infinity. Hence the high probability of having at least one high end node per cluster ensures the good coverage and performance of our methodology.

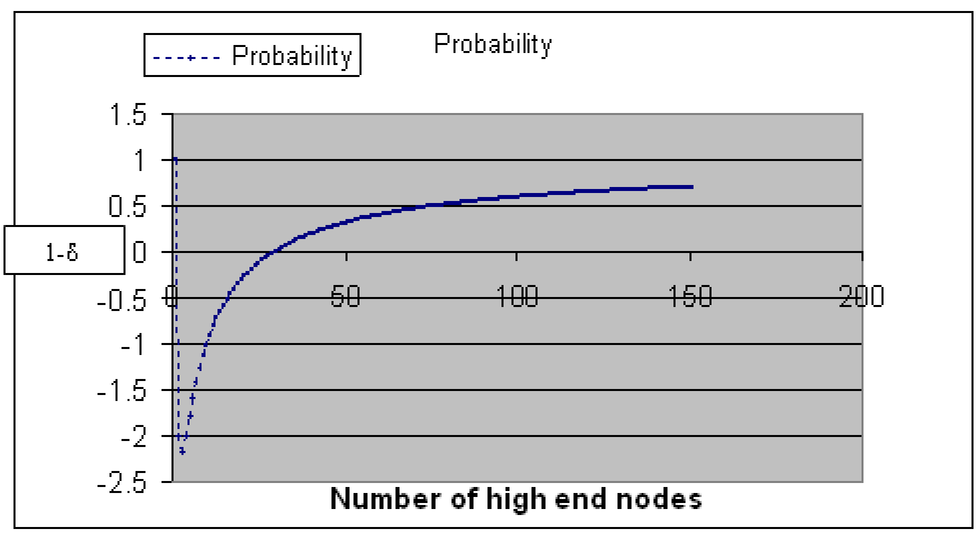

Figure 3: Showing the Probability 1-δ

In HSNs clustering, area covered, uniformity, time and distance traveled by data directly affects the performance of the network. The coverage could be defined as the ratio of the area covered by all high end nodes to the area of interest [6]. So the high end node will cover the area contained only in its sensing radius and correspondingly will make clusters, where cluster function could be defined as

$$C = \sum_{i=1}^{n} C_i$$

$\{C_1 = \{ p_j \ ; j \in$ low end nodes in cluster 1$\}$,
$C_2 = \{ p_j \ ; j \in$ low end nodes in cluster 2$\}$,
…………………… $C_n = \{ p_j \ ; j \in$ low end nodes in cluster n$\}\}$

And covered area will be given as $C = (C_{i=1,2,\ldots,n} A_i)/A$, where A_i represents the covered area by ith high end node, n is the total number of high end nodes and A is the area of interest, r is the radius covered by one node so one node will cover an area of πr^2.

When the network is properly distributed and covered and nodes are more connected to each other then the network utilize less energy, all nodes use their resources effectively with reduced power usage and increased throughput. So the Voronoi clustering is increasing the expected system lifetime.

IV. ROUTING METHODOLOGY

The ETR (Enhanced Tree Routing) [4] methodology reduces the number of hops required to reach to the base station. A special indexed tree is formed by limiting the number of offspring to three only. The idea is to maintain efficient routing by utilizing alternative hops with reduced hop count.

When some node fails, the shortest path tree topology can not forward data packets on their path. But our approach use neighbor table to forward the data in such situation. Under ideal conditions too this approach minimizes the number of hop counts. In general, this approach is an alternative to shortest path tree routing.

When sensors are deployed in a network then nodes records data in their neighbor table about their neighbors those are within its radio range. This neighbor table has data about parent node, child nodes, and all neighbor nodes. Each node is assigned an address before setting up the network. The neighbor table helps in generating parent child link. If the node finds that alternative path is shorter, then node uses the alternative path for data forwarding.

The Voronoi clustering and Enhanced Tree Routing algorithms have been implemented in the NS-2 network simulator. Basically, two agents have been written, one is VoronoiClusteringAgent for setting up the Voronoi Cluster Topology and another is ETRAgent, implementing the Enhanced Tree Routing protocol. The output of VoronoiClusteringAgent becomes the input to ETRAgent. A paper [4] has already been published on how do the ETRAgent works. The addressing scheme used is such that it minimizes the processing cost and the energy of the network is saved. Initially, ETRAgent counts hops from source to destination using a neighbor instead of shortest path tree, if hop counts are less with this method then neighbor is used to forward the data, else shortest path tree is used to route the information.

If source node, S, has no association of grandparent or sibling or grandchild with the destination node, D, and there could be one neighbor node, n, which could be either parent or child of destination node D [4]. Where ETRAgent assumes node n being neighbor of source node, S. Suppose source node, S, have data to transmit to destination node D, and node S makes a temporary link through n for data transmission as shown in fig 4. The shortest tree route is S->c->b->a->D, having hop count of 4, denoted as $H_{SPT} = 4$. With our approach the route is S->n->D, having hop count of 2, denoted as $H_{ETR} = 2$.

If $H_{SPT} - H_{ETR} > 0 \Rightarrow H_{ETR} < H_{SPT}$ and
the symmetric difference between these two hop counts is given as

$$H_{SPT} - H_{ETR} = \Delta$$

Which implies that there does exist one neighbor node, which is not a sibling and not even a grandparent/grandchild of the source node and is selected for routing the data. [4]

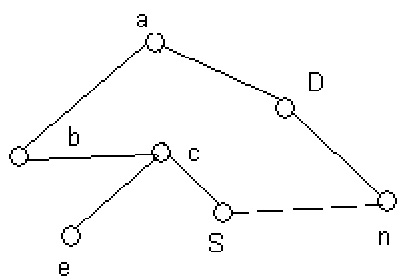

Figure 4: Node n is a neighbor node of S as S is neither a sibling nor grandparent/grandchild of D [4]

If source, S, and destination, D, nodes have any link as depicted in fig. 5 of grandparent-grandchild or parent-child or sibling link then having a neighbor node to route information will not decrease the number of hops. Therefore this routing approach will work only if source and destination node do not share any parent-child link [9]. Moreover if shortest path tree gives more hop counts than with ETRAgent then n, which is one hop neighbor to the source is used to route the data.

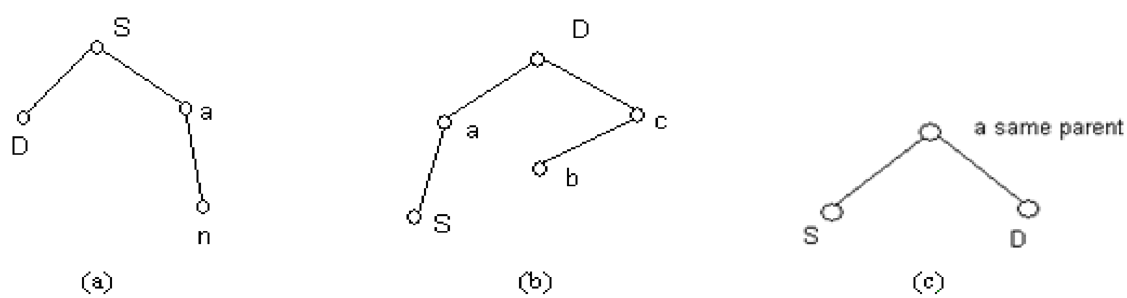

Figure 5: Source and destination nodes sharing a) parent-child link b) grandparent-grandchild link c) sibling link [4]

Although Clustered-ETR is an improved form of ETR [4], the steps followed in **Clustered-ETR are as follows:**

Step 1: Initially divide the given (x,y) plane into Voronoi clusters. Where each cluster will have one high end node and many low end nodes, and high end node will become the cluster head.

Step 2: All nodes will initialize their neighbor tables and start detecting if some event occurred.

Step 3: When some event occurs, verify if neighbor node is the sink/destination node, if yes, route the information to the destination and exit.

Step 4: Test whether sink/destination node is a grandparent/sibling, if yes, send the information all the way through the existing path and exit.

Step 5: Test whether sink/destination node is a grandchild, if yes, use shortest path tree to send the information and exit.

Step 6: Test whether sink/destination is not a sibling/grandparent/grandchild of the nodes selected for routing the information, instead have a neighbor node within its radio range. If the selected neighbor node satisfies the property, $H_{SPT} - H_{ETR} > 0$, select it to route the information and exit.

V. EVALUATION

We present the performance evaluation in this section. The NS-2 networks simulator [3, 11] is used to evaluate the efficiency and effectiveness of our cluster based enhanced tree routing algorithm (Clustered-ETR). Clustered-ETR algorithm is compared with ETR routing algorithm [4]. In Clustered-ETR technique the network is divided into Voronoi cluster and ETR runs on the undivided network. Being ETRAgent [4] already written by us in NS-2, and we knew what more attributes could be added to it to improve its performance, so ETR is chosen to evaluate against the Clustered-ETR algorithm.

A. Simulation Environment

Both the techniques, that is, Clustered-ETR and ETR are executed on randomly deployed nodes in square field of 670m x 670m network. There is only one base station also known as sink node and has address 0, all other nodes are source nodes. There are 16 high end nodes and 250 low end nodes. At MAC layer IEEE 802.11 protocol is used. ETR being routing algorithm, runs at the network layer level. When some event occurs, source gets ready for data transmission. The size of packets is 32 bytes. Every connection starts at 0.1 simulated second. The whole simulation executes for 150 simulated seconds. The antennas in use are omni-directional. For the base station, data sharing interface is set as Direct Sequenced Spread Spectrum (DSSS) Wave LAN. This setting is proposed by Lucent Technologies and it run on 914MHz radio frequency. The base station has infinite resources. Whereas high end nodes have 100.0J of initial energy and low end nodes have 10.0J of initial energy. Capture Threshold frequency for high end node is 10.0MHz and for low end nodes is 3.0MHz. Carrier sense Threshold value for high end node is 220m and for low end nodes is 65m. The radio frequency for high end nodes is 914 KHz and for low end nodes is 9140Hz. Each result is averaged over five random network topologies. To make comparison possible, the energy models used are same for both techniques, i.e., ETR and Clustered-ETR.

B. Performance Analysis

The NS-2 network simulator is used for performance evaluation. The following parameters are chosen for the comparison:

1. **Delivery Ratio**: It is also known as throughput and is a measure of the number of bytes received in one second at the destination. The network throughput of Clustered-

ETR is almost 1.6 times better than ETR as is shown in fig. 6. This happened because in Clustered-ETR the network is divided into regions, which makes network more structured and covered. Being organized network can give more data packets in less time than any random organization of network.

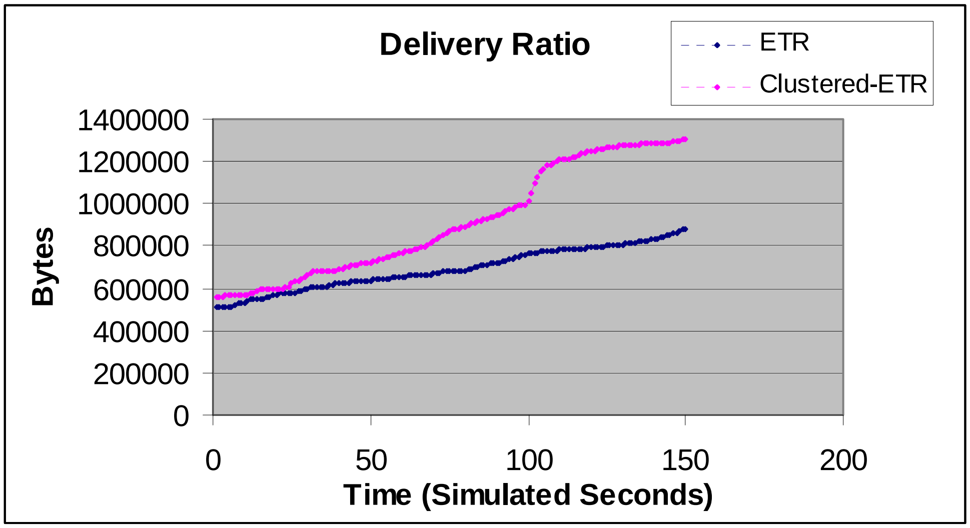

Figure 6: Delivery Ratio (throughput)

2. **Energy Utilization**: The total energy consumption grows for both protocols with the passage of time are demonstrated in fig. 7. Clustered-ETR is using almost 15% less energy than ETR. The reason is ETR nodes have more neighbor nodes to overhear than clustered-ETR, as it is divided into clusters and nodes listen to only those nodes which are in same cluster. So with ETR, more energy is dissipated for overhearing every neighbor node.

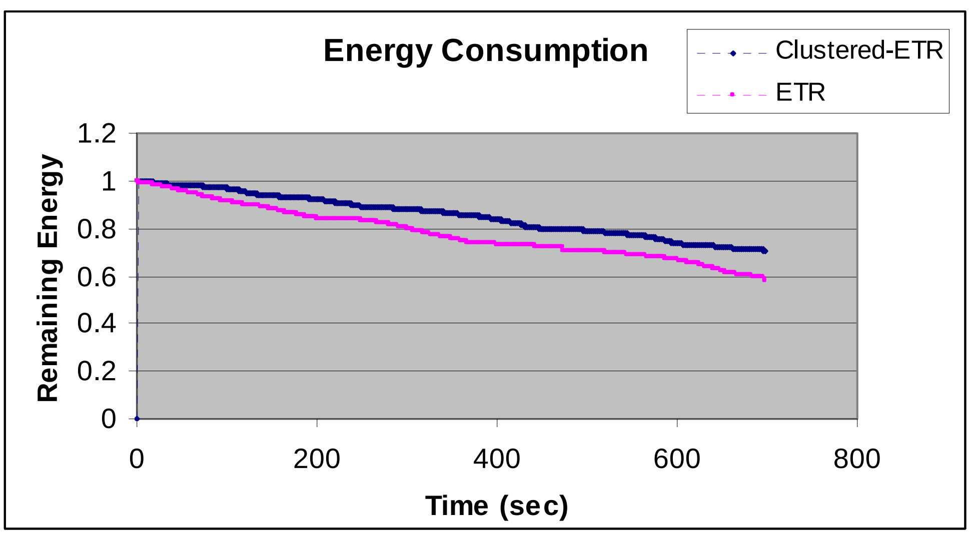

Figure 7: Energy utilization

3. **Network Delay**: The network delay is measured as the average time taken by data to reach to the destination from the source. Moreover, propagation delay and the

queuing delay also get added to the network delay. In fig. 8 it is shown that Clustered-ETR technique has almost half the network delay as that of ETR. The network delay in both protocols is also increasing as the network size is increasing. The reason is with the increase in size of network, routing path expands, which lead to more intermediate nodes and increased delays.

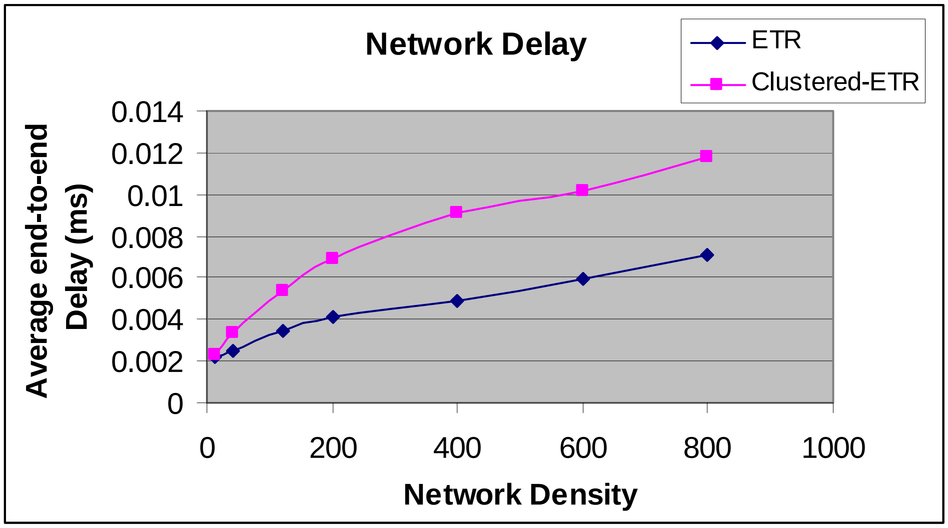

Figure 8: Network Delay

VI. CONCLUSION

In this paper, with empirical results we have shown that Clustered-ETR has about 1.6 time higher delivery ratio, half of the time less network delays and 15% less energy utilization than ETR. This has become possible because in Clustered-ETR the network is partitioned into cluster, which leads to a more structured and covered network. When the network is properly distributed and covered, and nodes are more connected to each other, then the network utilizes less energy, and all nodes use their resources effectively with reduced power usage and increased throughput. So in this organized network, nodes are using fewer amounts of basic resources, which have improved the performance of the network.

References:

1. Alejandro Navarro, and Hugh Rudnick, "Large-Scale Distribution Planning—Part II: Macro-Optimization with Voronoi's Diagram And Tabu Search", IEEE Transactions on Power Systems, Vol. 24, No. 2, pages 752-758, May 2009.
2. Chunyu Ai, Hailong Hou, Yingshu Li, and Raheem Beyah, "Authentic delay bounded event detection in heterogeneous wireless sensor networks", Ad Hoc Networks, Elsevier, pages 599–613, July 2008.
3. Fall K. and Varadhan K., ns notes and documentation, available from http://www-mash.cs.berkeley.edu/ns/.
4. Jit Kaur K., Xiaojiang Du, and Kendall Nygard, "Enhanced routing in Heterogeneous Sensor Networks", DOI 10.1109/computationworld.2009.43

5. Laskowski M.C., "Vapnik-Chervonenkis classes of definable sets," J. London Mathematical Society, vol. 45, no. 2, pages 377–384, 1992.
6. Nojeong Heo and Pramod K. Varshney, "Energy-Efficient Deployment of Intelligent Mobile Sensor Networks", IEEE Transactions On Systems, Man, And Cybernetics—Part A: Systems And Humans, Vol. 35, No. 1, pages 78-92, Jan. 2005.
7. Ossama Younis, Marwan Krunz, and Srinivasan Ramasubramanian, "Node Clustering in Wireless Sensor Networks: Recent Developments and Deployment Challenges", IEEE Network, pages 20-25, May/June 2006.
8. Rajko Mahkovic and Tomaˇz Slivnik, "Constructing the Generalized Local Voronoi Diagram from Laser Range Scanner Data", IEEE Transactions On Systems, Man, And Cybernetics—Part A: Systems And Humans, Vol. 30, No. 6, pages 710-719, Nov. 2000.
9. Wanzhi Qiu, Efstratios Skafidas, and Peng Hao, "Enhanced tree routing for wireless sensor networks", Ad Hoc Networks, Elsevier, pages 638–650, July, 2008.
10. Wei-Peng Chen, Jennifer C. Hou, and Lui Sha, "Dynamic Clustering for Acoustic Target Tracking in Wireless Sensor Networks", IEEE Transactions On Mobile Computing, Vol. 3, No. 3, pages 258-271, July-September 2004.
11. http://nsnam.isi.edu/nsnam/index.php/User_Information#Documentation, 2008.