

# Placing and Maintaining a Core Node in Wireless Ad Hoc Sensor Networks

Amit Dvir and Michael Segal

Department of Communication Systems Engineering  
Ben Gurion University of the Negev  
Israel  
azdvir, segal@cse.bgu.ac.il

**Abstract.** Wireless Ad hoc sensor networks are characterized by several constraints, such as *bandwidth*, *delay*, *power*, etc. These networks are examined by constructing a tree network. A core node usually chosen to be the *median* or *center* of the multicast tree network with a tend to minimize a performance metric, such as delay or bandwidth. In this paper, we present new efficient strategy for constructing and maintaining a core node in multicast tree for wireless ad hoc sensor networks that undergo dynamic changes based on local information. The new core (*centdian*) function is defined by convex combination that signifies total bandwidth and delay constraints. We provide two bounds of  $O(d)$  and  $O(d + l)$  time for maintaining the centdian using local updates, where  $l$  is the hop count between the new center and the new centdian and  $d$  is the diameter. We also show a  $O(n \log n)$  time solution for finding centdian in the Euclidian complete network using interesting observations. Finally a simulation is presented.<sup>1</sup>

**Key words:** Sensor networks; Wireless Ad hoc Networks; Multicast tree; Core Node

## 1 Introduction

Wireless Ad hoc sensor networks is a network architecture that can be rapidly deployed without relying on pre-existing fixed network infrastructure . Wireless communication is used to deliver information between nodes, which may be mobile and rapidly change the network topology. The wireless connections between the nodes (which later will be referred as links or edges) may suffer from frequent failures and recoveries due to the motion of the nodes and due to additional problems related to the propagation channels (e.g. obstructions, noise) or power limitations. A wireless ad hoc sensor network consists of a number of sensors spread across a geographical area. Each sensor has wireless communication capability and some level of intelligence for signal processing and networking of the data. Recently, wireless sensor networks have been attracting a great deal of commercial and research interest [13, 27, 29]. In particular, practical emergence of wireless ad hoc networks is widely considered revolutionary both in terms of paradigm shift as well as enabler of new applications.

---

<sup>1</sup> This research has been partially supported by INTEL and REMON consortium.

Group communication is the basis for numerous applications in which a single source delivers concurrently identical information to multiple destinations. This is usually obtained with efficient management of the network topology in the form of tree having specific properties. For example, multicast routing refers to the construction of a spanning tree rooted at the source and spanning all destinations [3, 11, 25, 30, 36]. Delivering the information only through edges that belong to the tree generates an efficient form of group communication which uses the smallest possible amount of network resources. In contrast, with unicast routing from the source to each destination, one needs to find a path from the source to each destination and generates an inefficient form of group communication where the same information is carried multiple times on the same network edges and the communication load on the intermediate nodes may significantly increase. We notice that wireless ad hoc sensor networks pose the reliable and efficient communication services necessary for distributed computing [6, 31], while objective functions considered are the most classical that involve the minimization of the average or the maximum distance to service facilities.

Generally, there are two well-known basic approaches to construct multicast trees: the minimal Steiner tree (SMT) and the shortest path tree (SPT). Steiner tree (or group-shared tree) tends to minimize the total cost of a tree spanning all group nodes with possibly additional non group member nodes. The optimal construction of the SMT is known to be a NP-hard problem [14, 22]. Some heuristics that offer efficient solutions to this problem are given in [21, 37]. The best up today solution has been derived by [38] and proved factor of 1.55. In contrary, SPT tends to minimize the cost of each path from the root source to each destination. This can be achieved in polynomial time by using the well-known algorithm by Dossey et al. [12]. The goal of a SPT is to preserve the minimal distances from the root to the nodes without any attempt to minimize the total cost of the tree.

Gupta and Srimani [16] present distributed core selection and migration protocols for multicast tree in MANET with dynamically changing network topology. The proposed core location method is based on the notion of *median* node of the current multicast tree instead of the median node of the entire network. The adaptive distributed core selection and migration method uses the fact that the median of a tree is equivalent to the *centroid* of that tree. Gupta and al. [17] present efficient core migration protocol for MANET that migrates the core until the multicast tree branches reflect the desired QoS requirements of the multicast application where the proposed core location method is based on the notion of *center* node of the current multicast tree. Bing-Hong and al. [9] gave heuristic to the minimum non-leaf multicast tree problem that reduce the number of non-leaves nodes in the multicast tree and their experimental results show that the multicast tree after the execution of their method has smaller number of non-leaves than others in the geometrically distributed network model.

The *bandwidth* of a tree is defined as the total distance of packet transmissions required to deliver packet from core node  $v$  to all others nodes. The maximum *delay* of the tree is the maximum distance to traversed by any packet in traveling from core node  $v$  to other node. The *transport* of a node is defined as the total distance of the node to all others nodes in the tree. The corresponding solution concepts have been considered in literature as median and center [26, 28, 39]. Since the median approach is based

on averaging, it often provides a solution in which remote and low-population density areas are discriminated against in terms of accessibility to public facilities, as compared with centrally situated and high-population density areas. For this reason, an alternative approach, involving the maximum distance between any customer and closest facility can be applied. This approach is referred to as the center solution concept [4]. The min-max objective primarily addresses geographical equity issues, which are of particular importance in spatial organization of emergency service systems. On the other hand, locating a facility at the center may cause a large increase in the total distance, thus generating a substantial loss in spatial efficiency. The problems of using only center or median as a core lead to search for a compromised solution concept called *centdian*, where centdian function presents some kind of trade-off between the center and the median functions ([19]). The centdian function for node  $v$  in the network is defined by  $D_v = \lambda \cdot \text{sum}(v) + (1 - \lambda) \cdot \text{dist}(v)$ ,  $0 \leq \lambda \leq 1$  where  $\text{dist}(v)$  is the maximum distance from node  $v$  to other nodes in the network,  $\text{sum}(v)$  is the sum of distances from node  $v$  to all other nodes in the networks.

Halpern [18] introduced the centdian model and studied the properties of the centdian in a tree. In a subsequent work, Carrizosa et al. [10] presented an axiomatic approach justifying the use of the centdian criterion. Tamir et al. [41] present the first polynomial time algorithm for the  $p$ -centdian problem on a tree with  $O(pn^6)$  complexity where  $p$  is the number of facilities. For more results about centdian problem, see [2, 7, 20, 32, 33, 40]. Other related notion of *ordered median* of a tree ([5, 23, 34, 35]) generalizes the most common criteria mentioned above, e.g., median, center and centdian. If there are  $n$  demand points in a tree  $T$ , this function is characterized by a sequence of reals,  $\kappa = (\kappa_1, \dots, \kappa_n)$ , satisfying  $\kappa_1 \geq \kappa_2 \dots \geq \kappa_n \geq 0$ . For a given subtree  $S \in T$ , let  $X(S) = \{x_1, \dots, x_n\}$  be the set of weighted distances of the  $n$  points to  $S$ . The value of the ordered median objective at  $S$  is obtained as follows: Sort the  $n$  elements in  $X(S)$  in non-increasing order, then compute the scalar product of the sorted list with the sequence  $\kappa$ . It is easy to see that when  $\kappa_i = 1, i = 1, \dots, n$ , we get the median objective and when  $\kappa_1 = 1$  and  $\kappa_i = 0, i = 2, \dots, n$ , we obtain the center objective. For the case  $\kappa_1 = 1$  and  $\kappa_i = \lambda, i = 2, \dots, n$  we get the centdian objective. Unfortunately, constructing and maintaining cores by use of ordered median technique is not suitable for wireless ad hoc sensor networks, since this technique requires keeping some global information about nodes of network which is completely inconceivable in the case of wireless ad hoc sensor networks.

Most protocols for constructing core node are not suitable for wireless ad hoc sensor networks, since these algorithms are not based on local updates. In this paper, we present new efficient strategy for constructing and maintaining a core node under centdian criteria in multicast tree for wireless ad hoc sensor networks with dynamic changes in the network topology. The new core node is defined by convex combination of the sum of the weighted distance paths (sum of the weighted edges in the path) of all the nodes in the tree network to the core node and the maximum weighted distance from the core node to the farthest node in the tree network satisfied center and median core functions. We also provide two bounds of  $O(d)$  and  $O(d + l)$  time for maintaining the centdian after a change (add/remove edge/node) in the topology of the tree network, where  $l$  is the hop count between the new center and the new centdian of the multicast

tree and  $d$  is the diameter of the tree. We show an  $O(n \log n)$  time algorithm for finding a centdian node in the Euclidian complete network bases on observation in [8]. Finally, we present a simulation that compare our new core solution with well known cores' strategies to exhibit the advantages and the efficiency of our algorithms.

This paper is organized as follows: Section 2 presents a new algorithm that finds and maintains a centdian core in a multicast tree. In Section 2.2 we show a solution for a static Euclidian network. Next we show our simulation results and finally, we conclude with several ideas for future work.

## 2 Algorithm to find centdian of multicast tree in wireless ad hoc sensor networks

We model the topology of wireless ad hoc sensor networks by weighted undirected graph  $G(V, E, W_e)$ , where  $V$  is the set of nodes,  $E$  is the set of edges between neighboring nodes and  $W_e$  is an edge weight function, e.g. squared distance between the endpoints of edges. Note that the edges represent logical connectivity between nodes, i.e., there is an edge between two nodes  $u$  and  $v$  if they can hear each other's local broadcast. Since the nodes are mobile, the network topology graph stochastically changes. Let us define by  $T(V, E')$  a weighted multicast tree of  $G$ . For a node  $v \in T$  we define by  $K_v$  the number of nodes in the connected component containing  $v$  (created by removing  $e(v, x)$ ); by  $W_v$  the total sum of weighted distances from the nodes in the connected component containing  $v$  (created by removing  $e(v, x)$ ) to node  $v$ . *Center* of a tree  $T$  is a node  $c_1 \in T$  such that the maximal distance from  $c_1$  to any other node in  $T$  is minimized, i.e.  $dist(c_1, T) = \min_{v \in T} dist(v, T)$ . In order to find a center of tree  $T$  we can use the distributed algorithm described in [26] that requires  $r(I) + (d(T)/2)$  time, where  $r(I)$  is the the maximal weighted distance from the initiator node  $I$  to any other node in  $T$  and  $d(T)$  is the weighted diameter of the tree. This algorithm finds the center node by starting form an arbitrary node  $I$  and goes from the internal nodes towards the leaves and back to the new center using the information from the leaves about the weighted distance path and the knowledge that the center of the tree lies on the diameter of the tree. *Median* of a tree  $T$  is a node  $c_2 \in T$  such that the sum of the weighted distances from  $c_2$  to any other node in  $T$  is minimized, i.e.  $sum(c_2, T) = \min_{v \in T} sum(v, T)$ . In order to find a median of a tree  $T$  we can use the distributed algorithm in [26] that requires  $\max_{x \in T} (r(I) + d(x, c_2))$  time, where  $d(x, c_2)$  is the weighted distance between node  $x$  and the new median. This algorithm finds the median node by starting form an arbitrary node  $I$  and goes from the internal nodes towards the leaves. Each leave propagates the weight of its edge and each internal node propagates the sum of values obtain from its descendants plus the weight of the edge connecting him to its predecessor in the tree.

Next, we show a simple algorithm to find the number of nodes in each one of node  $v$  branches. We define by  $K_{v_i}$ ,  $i = 1 \dots b$ , to be the number of nodes in the  $i$ th branch of node  $v$ , with  $b$  standing for the number of branches of node  $v$ . By convergecast process from the leaves towards the center of the tree we can find the total number of nodes in the tree. By knowing this number, we start a new process from the leaves to find for each node  $v$  its values  $K_{v_i}$ . Each leaf sends to its father  $w$  in a rooted tree  $T$  a  $num(I)$

message. Each internal node  $w$  gets from his sons their  $num$  messages and sums all the values in the messages. The process converges towards  $c_1$ .

It is a well-known fact that a centdian is located on the path connecting center  $c_1$  and median  $c_2$  ([18]). The following lemma presents an efficient way of calculating centdian in a multicast tree based on knowledge about location of the center and median.

**Lemma 1.**  $D_v > D_x$  iff  $\lambda((K_x - K_v) + 1) < 1$ .

**Proof:** A centdian node  $x$  must minimize expression  $D_x = \lambda \cdot sum(x, T') + (1 - \lambda) \cdot dist(x, T')$ . Denote by  $D_v = \lambda \cdot sum(v, T') + (1 - \lambda) \cdot dist(v, T')$  the cost of node  $v$  who is the neighbor of current centdian  $x$  with the minimum value of  $D$  from  $x$  neighbors. We should move the centdian towards  $v$  only if  $D_x > D_v$ . Notice that  $dist(x, T') = \frac{D_x - \lambda sum(x, T')}{(1 - \lambda)}$ . We conclude that  $sum(x, T') = W_x + (K_x + 1) \cdot d(x, v) + W_v$  and  $sum(v, T') = W_v + (K_x + 1) \cdot d(x, v) + W_x$ . Therefore,  $D_v = \lambda(W_v + (K_x + 1) \cdot d(x, v) + W_x) + (1 - \lambda)dist(v, T')$ . It easy to see that there are 5 different cases (out of 9) that we should deal when  $D_x > D_v$ ,

- 1)  $sum(v, T') < sum(x, T')$  and  $dist(v, T') < dist(x, T')$ .
- 2)  $sum(v, T') > sum(x, T')$  and  $dist(v, T') < dist(x, T')$ .
- 3)  $sum(v, T') = sum(x, T')$  and  $dist(v, T') < dist(x, T')$ .
- 4)  $sum(v, T') < sum(x, T')$  and  $dist(v, T') = dist(x, T')$ .
- 5)  $sum(v, T') < sum(x, T')$  and  $dist(v, T') > dist(x, T')$ .

We present an analysis only for cases 1-3 that are relevant (case 4 is trivial ( $sum(v, T') = sum(x, T')$ )) and case 5 is equivalent to cases 1-3. In cases 1-3  $|dist(x, T') - dist(v, T')| \leq d(x, v)$ , therefore

$$\begin{aligned}
D_v &= \lambda(W_v + (K_x + 1) \cdot d(x, v) + W_x) + (1 - \lambda)dist(v, T') = \lambda(W_v + \\
&+ (K_x + 1) \cdot d(x, v) + W_x) + (1 - \lambda)(dist(x, T') - dist(x, v)) = \lambda(W_v + \\
&+ (K_x + 1) \cdot d(x, v) + W_x) + (1 - \lambda)\left(\frac{D_x - \lambda sum(x, T')}{(1 - \lambda)} - dist(x, v)\right) = \\
&= \lambda(W_v + (K_x + 1) \cdot d(x, v) + W_x) + D_x - \lambda sum(x, T') - dist(x, v)(1 - \lambda) = \\
&= \lambda W_v + \lambda(K_x + 1) \cdot d(x, v) + \lambda W_x + D_x - \lambda sum(x, T') - dist(x, v)(1 - \lambda) = \\
&= \lambda W_v + \lambda(K_x + 1) \cdot d(x, v) + \lambda W_x + D_x - \lambda(W_x + (K_v + 1) \cdot d(x, v) + W_v) - \\
&- dist(x, v)(1 - \lambda) = \lambda W_v + \lambda(K_x + 1) \cdot d(x, v) + \lambda W_x + D_x - \lambda W_x - \lambda(K_v + 1) \cdot \\
&\cdot d(x, v) - \lambda W_v - dist(x, v)(1 - \lambda) = D_x + \lambda d(x, v)(K_x - K_v) - dist(x, v)(1 - \lambda).
\end{aligned}$$

As we stated above, we move the centdian node only if  $D_1 > D_2$ . This happens when  $D_x - (D_x + \lambda d(x, v)(K_x - K_v) - dist(x, v)(1 - \lambda)) > 0$  or in other words  $\lambda(K_x - K_v + 1) < 1$ . In case 4  $dist(x, T') = dist(v, T')$ , thus  $D_v = \lambda(W_v + (K_x + 1) \cdot d(x, v) + W_x) + (1 - \lambda)dist(x, T')$ . Analysis similar to the previous one shows that  $D_x > D_v$  only if  $\lambda d(x, v)(K_x - K_v) < 0$ . In case 5 we get that  $D_x > D_v$  only if  $\lambda((K_v - K_x) - 1) < -1$ . It follows that the inequality  $D_x > D_v$  holds when  $\lambda(K_x - K_v + 1) < 1$   $\square$

Based on this lemma, we can locate the centdian in the multicast tree, starting either at center or the median of the multicast tree and going over the path between them, locally improving from one neighboring node to another. Thus, the centdian in the tree can be found in  $O(l)$  time when the location of the center and the median are known, with  $l$  standing for the number of nodes in the path connecting center and median.

### 2.1 Maintaining a centdian in a multicast tree

When we have some change in the multicast tree, each node needs to update the number of nodes in each of its branches. By using  $K_x$  values of the centdian node, our maintaining algorithms running on the subtree with the highest number of the multicast group members. In what follows we show two different approaches to maintain centdian in a multicast tree. Both approaches use the fact that the centdian function is convex and therefore has only one minimum point.

The first approach is to maintain the center of the tree in  $O(d)$  time using the algorithm in [24]. The new centdian lies on the path between the center and the median of the updated tree. Therefore, starting from the new center and finding the node that locally improves the centdian function value, point us the direction towards the new centdian of the tree. The centdian, thus, can be maintained in worst-case  $O(d+l)$  time, where  $l$  is the hop count between the new center and the new centdian of the multicast tree and  $d$  is the diameter of the tree.

The second approach uses the fact that the neighbor of the old centdian  $x$  that the most improves the centdian function value lies on the path between the old centdian and the new centdian. Therefore, the centdian can be maintained in worst-case  $O(d)$  time.

Since we want only multicast group members to be assigned the responsibility of core node, the second approach needs to be modified. If the new centdian node is a multicast member it becomes the actual new centdian of the tree. If not, we seek the path towards the old centdian in order to find a node that belongs to multicast group and declare this node to be the new actual centdian of the multicast tree.

### 2.2 Algorithm to find centdian in Euclidean plane

We model the topology of planar network as explained above by having the edge weight function defined as the squared distance between the nodes. The motivation to choose this function is the common method that power transmitting behaves quadratically to the distance between transmitting and receiving node. Using the observation in Bespamyatnikh et al. [8] we are able to solve the centdian problem in Euclidean plane in  $O(n \log n)$  time. The farthest point Voronoi diagram of a collection of points  $S$  in the plane is a partition of plane into cells, each of which consists of the points further to one particular points than to any others. This diagram can be constructed in  $O(n \log n)$  time supporting a query requests in  $O(\log n)$  time. For a given point  $p$ , a query asks about the farthest neighbor of  $p$  in  $S$ . Thus, in  $O(n \log n)$  time we can find, for each point, its farthest neighbor performing total  $n$  queries. In other words, for every node  $v$  in the network, we find  $dist(v)$  in total  $O(n \log n)$  time.

Bespamyatnikh et al. [8] observed that "squared" Euclidean metric is separable, i.e. the distance between two points is the sum of their squared  $x$  and  $y$ -coordinates'

differences. We follow the notations from [8]. We sort our points according to their  $x$  and  $y$ -coordinates. Let  $\{p_1, \dots, p_n\}$  be the sorted points. For every point  $p_i \in S$  we can compute the sum  $\Sigma_i^x$  of the  $x$ -distances from  $p_i$  to the rest of the points in  $S$ . This is performed efficiently as follows. For the point  $p_1$  we compute  $\Sigma_1^x$  by computing and summing up each of the  $n - 1$  distances. For  $1 < i \leq n$  we define  $\Sigma_i^x$  recursively: assume the  $x$ -distance between  $p_{i-1}$  and  $p_i$  is  $\delta$ , then  $\Sigma_i^x = \Sigma_{i-1}^x + \delta \cdot (i - 1) - \delta \cdot (n - i + 1)$ . The sums  $\Sigma_i^x$  (for  $i = 1, \dots, n$ ) can be computed in linear time when the points are sorted. The value of  $\sum_i^y$  is computed similarly. Next, let  $sum_i^x = \sum_{j=1}^n (x_j - x_i)^2$ . The recursion formula for computing all the squared  $x$ -distances is easily computed to be  $sum_i^x = sum_{i-1}^x - 2\delta \sum_{i-1}^x - n\delta^2$ , where the  $x$ -distance between  $p_{i-1}$  and  $p_i$  is  $\delta$ .

Assume the point  $p \in S$  is  $i^{th}$  in the  $x$  order and  $j^{th}$  in the  $y$  order. The sum of squared Euclidean distances from  $p$  to the points in  $S$  is  $sum(p) = sum_i^x + sum_j^y$ .

It remains to compute, for every node  $v$  the value of centdian function based on values of computed already  $sum(v)$  and  $dist(v)$  values. This is done in linear time. Thus, we can conclude,

**Theorem 1.** *Given a set  $S$  of  $n$  nodes in Euclidean complete graph with a cost of every edge that equals the squared Euclidean distance between nodes, we can find the centdian node in this graph in  $O(n \log n)$  time.*

### 3 Simulation

This section describes the medium-scale experiment in details. The objectives of the experiment were to test whether the suggested maintaining algorithm actually works, and to compare its results to the performance of other core algorithms. For this simulation we choose to implement the second approach. As we performed our simulation we made an interesting observation about the runtime bound of the first approach of maintaining the centdian node.

#### 3.1 Environment

The following assumptions have been made:

- For each node, the transmission and reception range are equal; however different nodes can have different ranges. The radius value refers to the transmission range.
- All the nodes are equal in their functionalities and abilities.
- The movement of each node is based on mobility model of random walk based on random directions and speeds ([15]).<sup>2</sup>
- There is no dependence between the nodes and the boundary of the network is predefined.

In our simulation we used 5 different types of cores: center, median, continuous median, centdian and continuous centdian (with different values of  $\lambda$ ). The difference between continuous and non continuous core is that non continuous core can "jump" from one node to another while continuous core keep continuous track of the path of previous core towards the newly computed core.

<sup>2</sup> In this model each node moves from its current location to a new location by randomly choosing speed and direction in which to travel.

### 3.2 Results

The main goal of our simulation is to examine the influence of the multicast group on the cores' behavior. One of the parameters we wish to exam is the period of time that cores are co-located at the same node (defined by collision). From the obtained results we can learn that the behavior of the cores in the multicast tree is sometimes similar to the behavior of the cores in regular network tree, for example:

- When  $\lambda = 1$  the collision between the centdian core and center core is 100%.
- When  $\lambda \leq 0.5$  the collision between the centdian core and median core is 100% as has been proved in [18].

In some cases, the collision value between the median/center core and centdian core should be 1, but in our simulation the collision value in those cases are less then 1. The reason for that is the well known fact that in a tree two centers/medians may possibly exist. In the simulation we choose them in arbitrarily fashion. We simulate ad hoc sensor network with 100 nodes and 50 multicast nodes with a variety range of radius and  $\lambda$  values in network boundary of 600x600 meters. Figures 1–7 show the influence of the radius on the constraints' values of the network. In particular, Figures 1–2 show the transport and delay values of the tree network as an unimodal linear function with a break point being a maximal value. The reason for that is as long as the radius is growing the network becomes more connected and more nodes are participating in the network. Starting at some point of time the network becomes to be connected and the pathes from nodes to the core contain small amount of hops. Figure 3 shows the connection between the radius and the life span of the cores, with life span being the period of time/rounds that the core does not change its location. It easy to see that as we increase the radius the life span also grows up.

The collision between the cores with various values of  $\lambda$  and radius is depicted in Figures 4–7. In Figure 4 we focus on the collision between the centdian and median, while in Figures 5–6 we have examined the collision between the continuous centdian and the well known cores. Figure 7 shows the collision between the new centdian and the new center. From Figure 8 we can learn that for most radius' values, the value  $l$  (number of hops between center and centdian) is small. The continuous centdian core achieves improved convergence to delay performance than median core and better transportation performance than center core. The continuous centdian core achieves these properties in a well connected networks, as well as in sparse networks too.

## 4 Conclusion and Future Work

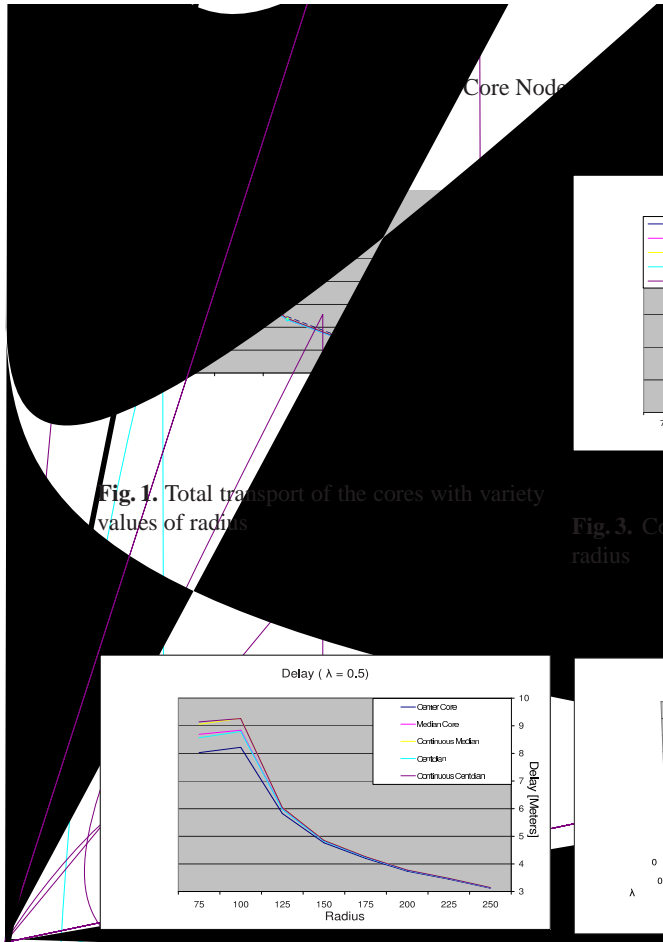
We have developed a new distributed algorithm for finding and maintaining centdian core in ad hoc sensor network that is based on processing local information of the network. Analytic analysis to bound value  $l$  seems to be very interesting. One interesting future direction is by adapting self-stabilizing algorithm to core selection problem in ad hoc sensor network when it gets partitioned and partitions get connected. The analysis of the model where one assumes some distribution for the velocities of the nodes seems also attractive.



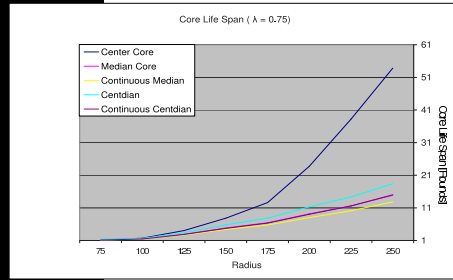
## References

1. S. Alstrup, J. Holm and M. Thorup, "Maintaining median and center in dynamic trees", In 7th Scandinavian Workshop on Algorithm Theory,(2000), pp. 46-56.
2. I. Averbakh and O. Berman, "Algorithms for path medi-centers of a tree", *Computers Operation Research*, Vol. 26, (1999), pp. 1395-1409.
3. T. Ballardie, P. Francis, and J. Crowcroft, "Core Based Trees (CBT): An Architecture for Scalable Inter Domain Multicast Routing" *Proc. ACM SIGCOMM*, (1993), pp. 85-95.
4. J. Bar-Ilan, G. Kortsarz and D. Peleg, "How to allocate network centers", *Journal of Algorithms*, Vol. 15, (1993), pp. 385-415.
5. R. Benkoczi, B. Bhattacharya and A. Tamir, "Collection Depots Facility Location Problems in Trees", submitted, <http://www.tau.ac.il/~atamir/tamirp.html>.
6. Y. Ben-Shimol, A. Dvir and M. Segal, "SPLAST: A novel approach for multicasting in mobile ad hoc networks", *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Vol. 2, (2004), pp. 1011-1015.
7. O. Berman and E. K. Yang, "Medi-centre location problems", *Journal Operation Research Society*, Vol. 42, (1991), pp. 313-322.
8. S. Bespamyatnikh, K. Kedem, M. Segal and A. Tamir "Optimal facility location under various distance functions", *Int. Journal on Comp. Geometry and Appls.*, 10(5), (2000), pp. 523-534.
9. L. Bing-Hong, K. Wei-Chieh and T. Ming-Jer, "Distributed formation of core-based forwarding multicast trees in mobile ad hoc networks", *Telecommun System*, Vol. 32, (2006), pp 263-281.
10. E. J. Carrizosa, E. Conde, F. R. Fernandez and J. Puerto, "An axiomatic approach to the centdian criterion", *Location Science*, Vol. 2, (1994), pp. 165-171.
11. S.E. Deering, D. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu, and L. Wei, "The PIM Architecture for Wide-Area Multicast Routing", *IEEE/ACM Trans. Networking*, Vol. 4, Num. 2, (1996), pp. 153-162.
12. J. Dossey, A. Otto, L. Spence, C. Eynden, *Discrete Mathematics*, Harper Collins College Publishers, (1993).
13. S. Dulman, M. Rossi, P. Havinga and M. Zorzi, "On the hop count statistics for randomly deployed wireless sensor networks", *International Journal of Sensor Networks*, Vol. 1, (2006), pp. 89-102.
14. M. R. Garey, R.L. Graham, D.S. Johnson, "The complexity of computing Steiner minimal trees", *SIAM J. Appl. Math.*, Vol 32, (1977), pp 835-859.
15. L. M. Gavrilovska and V. M. Atanasovski, "Ad Hoc Networking Toward 4G: Challenges and QoS Solutions", *International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services*, (2005).
16. S. K. S. Gupta, P. K. Srimani, "Adaptive core selection and migration method for multicast routing in mobile ad hoc networks", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 14, (2003), pp. 27-38.
17. S. K. S. Gupta, L. Schwiebert, C. Jiao and M. Kochhal, "An Efficient Core Migration Protocol for QoS in Mobile Ad Hoc Networks" *IEEE Int'l Performance Computing, and Communications Conference*, (2002), pp. 387-391.
18. J. Halpern, "Finding minimal center-median convex combination (cent-dian) of a graph", *Management Science*, Vol. 16, (1978), pp. 534-544.
19. J. Halpern, "The location of a centdian convex combination on an undirected tree", *Journal Regional Science*, Vol. 16, (1976), pp. 237-245.
20. G. Y. Handler, "Medi-centers of a tree", *Transportation Science*, Vol. 19, (1985), pp. 246-260.

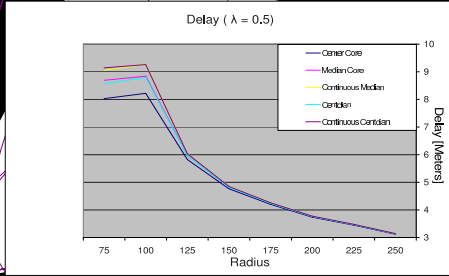
21. F. Hwang F, D. Richards, "Steiner Tree Problems", *Networks*, Vol 22, Num 1, (1992), pp.55-89.
22. F. K. Hwang, D. S. Richards, P. Winter. *The Steiner tree problem*, North-Holland, (1992), pp. 93-202.
23. J. Kalcsics, S. Nickel, J. Puerto and A. Tamir, "Algorithmic results for ordered median problems defined on networks and the plane", *Operations Research Letters*, Vol. 30, (2002), pp. 149-158.
24. M. H. Karaata, S. V. Pemmaraju, S. C. Bruell and S. Ghosh, "Self-stabilizing algorithms for finding centers and medians of trees", 13th Annual ACM Symposium on Principles of Distributed Computing, (1994), pp. 374-395.
25. Y. B. Ko and N. H. Vaidya. "Geocasting in Mobile Ad hoc Networks: Location-Based Multicast Algorithms". In *Second IEEE Workshop on Mobile Computing Systems and Applications*, (1999), pp. 101100.
26. E. Korach, D. Rotem, N. Santoro, "Distributed Algorithms for Finding Centers and Medians in Networks", *ACM Trans. Program. Lang. Syst.*, Vol 6, (1984), pp. 380-401.
27. J. Lansford and P. Bahl, "The Design And Implementation Of HomeRF: A Radio Frequency Wireless Networking Standard For The Connected Home" *Proceedings of the IEEE*, Vol. 88, (2000), pp. 1662-1676.
28. J. H. Lin and J. S. Vitter, "Approximation algorithms for geometric median problems", *Information Processing Letters*, Vol. 44, (1992), pp. 245-249.
29. S. Meguerdichian, F. Koushanfar, G. Qu and M. Potkonjak, "Exposure In Wireless Ad-Hoc Sensor Networks", *Proc. ACM MobiCom*, (2001).
30. *Mobile Ad hoc Networks (MANET) Charter*. MANET Homepage. <http://www.ietf.org/html.charters/manetcharter.html>.
31. E. Pagani and G. P. Rossi, "Reliable broadcast in mobile multi-hop packet networks", *IEEE international conference on Mobile computing and networking*, (1997), pp. 34-42.
32. D. Perez-Brito, J. A. Moreno-Perez and I. Rodriguez-Martin, "The finite dominating set for the  $p$ -facility centdian network location problem", *Location Science*, Vol. 11, (1997), pp. 27-40.
33. D. Perez-Brito, J. A. Moreno-Perez and I. Rodriguez-Martin, "The 2-facility centdian network problem", *Location Science*, Vol. 6, (1998), pp. 369-381.
34. J. Puerto, A. M. Rodriguez-Chia, A. Tamir and D. Perez-Brito, "The bi-criteria doubly weighted center-median path problem on a tree", *Networks*, Volume 47, Issue 4, (2006), pp. 237-247.
35. J. Puerto and A. Tamir, "Locating tree-shaped facilities using the ordered median objective", *Mathematical Programming*, Vol. 102, (2005), pp. 313-338.
36. T. Pusateri. *Distance Vector Multicast Routing Protocol*. Internet Draft draft-ietf-idmr-dvmrp-v3-09.txt, Internet Engineering Task Force, (1999).
37. R. Ravi, "Steiner trees and beyond: Approximation algorithms for network design", *Ph.D. Dissertation*, Brown University, (1993).
38. G. Robins, A. Zelikovsky. "Improved Steiner tree approximation in graphs", *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, (2000), pp. 770-779.
39. A. Tamir, "An  $O(pn)$  algorithm for  $p$ -median and related problems on tree graphs", *Operations Research Letters*, Vol. 19, (1999), pp. 59-64.
40. A. Tamir, "Fully polynomial approximation schemes for locating a tree-shaped facility: A generalization of the knapsack problem", *Discrete Applied Mathematics*, Vol. 87, (1998), pp. 229-243.
41. A. Tamir, D. Perez-Brito and J. A. Moreno-Perez, "A polynomial algorithm for the  $p$ -centdian problem on a tree", *Networks*, Vol. 32, Num. 4, (1998), pp. 255-262.



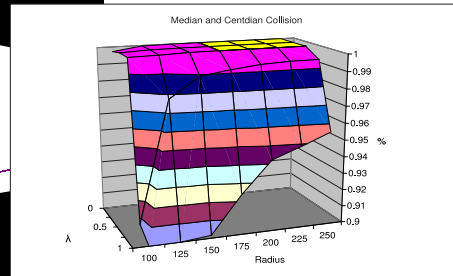
**Fig. 1.** Total transport of the cores with variety values of radius



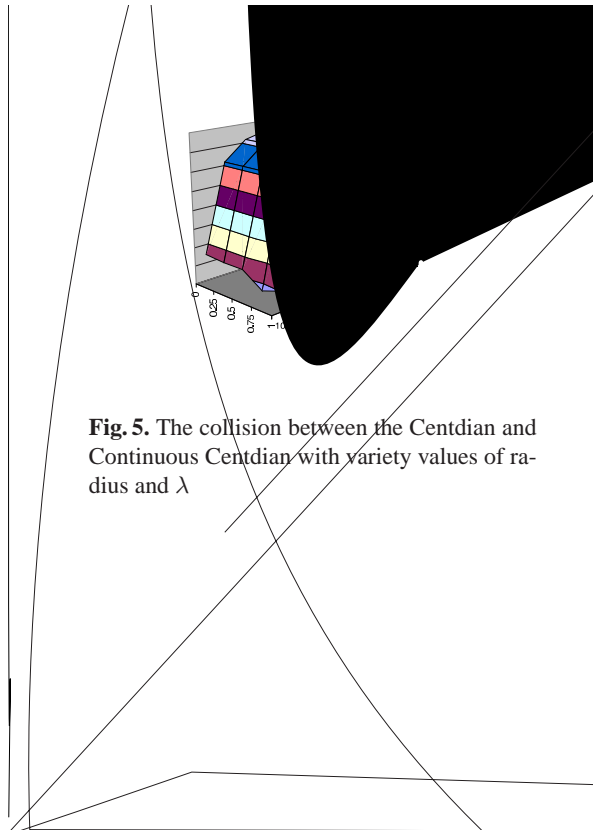
**Fig. 3.** Cores life span with variety values of radius



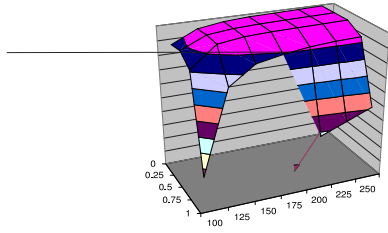
**Fig. 2.** Total delay of the cores with variety values of radius



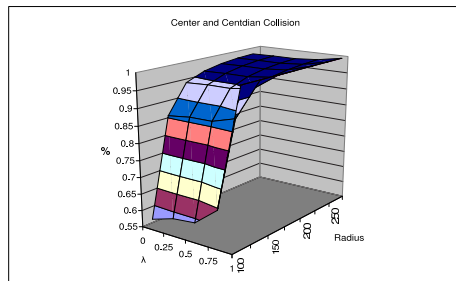
**Fig. 4.** The collision between the Centdian and Median with variety values of radius and  $\lambda$



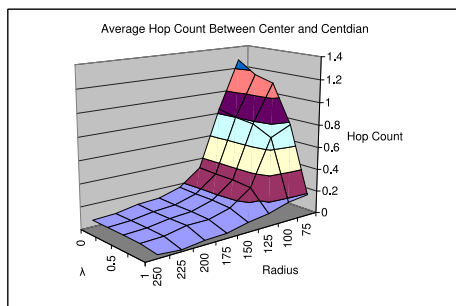
**Fig. 5.** The collision between the Centdian and Continuous Centdian with variety values of radius and  $\lambda$



**Fig. 6.** The collision between the Continuous Centdian and Continuous Median with variety values of radius and  $\lambda$



**Fig. 7.** The collision between the Centdian and Center with variety values of radius and  $\lambda$



**Fig. 8.** The Average Hop Count between Centdian and Center with variety values of radius and  $\lambda$