

TASC: Topology Adaptive Spatial Clustering for Sensor Networks

Reino Virrankoski* and Andreas Savvides
 Embedded Networks and Applications Lab (ENALAB)
 Yale University, 51 Prospect Street #212, New Haven, CT 06520
 Tel. +1-203-432-1275, Fax +1-203-432-0593
 reino.virrankoski@hut.fi, andreas.savvides@yale.edu

Abstract—The ability to extract topological regularity out of large randomly deployed sensor networks holds the promise to maximally leverage correlation for data aggregation and also to assist with sensor localization and hierarchy creation. This paper focuses on extracting such regular structures from physical topology through the development of a distributed clustering scheme. The Topology Adaptive Spatial Clustering (TASC) algorithm presented here is a distributed algorithm that partitions the network into a set of locally isotropic, non-overlapping clusters without prior knowledge of the number of clusters, cluster size and node coordinates. This is achieved by deriving a set of weights that encode distance measurements, connectivity and density information within the locality of each node. The derived weights form the terrain for holding a coordinated leader election in which each node selects the node closer to the center of mass of its neighborhood to become its leader. The clustering algorithm also employs a dynamic density reachability criterion that groups nodes according to their neighborhood’s density properties. Our simulation results show that the proposed algorithm can trace locally isotropic structures in non-isotropic network and cluster the network with respect to local density attributes. We also found out that TASC exhibits consistent behavior in the presence of moderate measurement noise levels.

I. INTRODUCTION

The anticipation of large-scale sensor networks and experience from preliminary deployments has demonstrated the need for meaningful decomposition of large distributed sensor networks into a set of smaller sub-networks. Such decomposition should be conducted in a manner that facilitates sensor node coordination and enhances the feasibility of network management and in-network processing and aggregation of sensor data. In this paper, we explore this issue of network decomposition through the development of a specialized

distributed clustering scheme. The scheme we investigate is designed to extract regularity from irregular network topologies by allowing the nodes to organize themselves into groups of locally isotropic (or regular) non-overlapping clusters without requiring the knowledge of node locations. Given the close coupling of sensors to the physical world we advocate that such a classification of sensor nodes according to their spatial attributes would be beneficial from multiple aspects.

Besides the intuitive benefit of improving the ease of network management, the spatial grouping of nodes with respect to regions of close proximity and similar deployment density promotes efficient data aggregation and efficient compression of sensor data. Spatial clustering would also assist transmission power control, since intra-cluster communication requires less transmission power in dense clusters. Moreover, as pointed out by [1] spatial irregularity in sensor sampling can exacerbate the load and cost imbalance between different parts of the network. This is mainly because many of the existing distributed signal processing and compression algorithms assume spatially regular data samples (see Figure 1). This also entails that the spatial grouping of nodes can help reduce the propagation of redundant data inside the network. This argument is further reinforced by the recent results presented in [2].

Despite the fact that clustering has been previously studied both theoretically and in the context of ad-hoc networks [3]–[10], its consideration in the context of sensor networks gives rise to a new problem setup where sensor measurements are used as actual inputs to the problem.

The proposed distributed algorithm does not require node locations but it assumes that nodes are aware of their 2-hop neighborhood. It also assumes that nodes are able to measure distances to their one hop neighbors. We consider both assumptions reasonable. The former is a standard assumption for many neighborhood discovery algorithms whereas the latter is becoming a common

*Visiting Assistant Researcher from the Control Engineering Laboratory, Helsinki University of Technology, Finland.

feature of many sensor network applications, though not all of them. Accurate internode distance measurements in the sensor network domain have been demonstrated using ultrasound in the system described in [13], the MIT Crickets [14] and in the Medusa MK-2 node [15]. In the radio domain, ultra-wide-band ranging systems such as the one offered by Ubisense [16] have already demonstrated accurate distance measurements with small sensor form factors that will be suitable for sensor networks. Moreover camera based schemes such as the one we developed in [24] can accurately measure internode distances without requiring specialized measurement hardware on each node. Finally, we note that the spatial clustering of the network before node localization is actually an advantage for ad-hoc localization. Ad-hoc localization schemes such as [15], [17]–[19] may benefit from the properties of our algorithm to eliminate computation redundancies and geometric error propagation.

The contribution of this paper is the development and characterization of a *Topology Adaptive Spatial Clustering Scheme* (TASC) that operates on combination of node weights and a dynamic density reachability criterion adopted from previous work in the database community [11], [12]. The paper is organized as follows. In the next section we highlight the related work. Section III describes the clustering problem requirements. Section IV provides the details of our weight scheme and density reachability criterion and describes the clustering algorithm. Algorithm evaluation through simulations is presented in Section V. Section VI discusses some additional attributes and Section VII states our conclusions and plans for future work.

II. RELATED WORK

A mathematical framework that has similarities with the network clustering problem is k-means clustering [7]. Some interesting k-means clustering modifications were recently applied to ad hoc clustering in [8], [9], [20]. What makes the setting we investigate different from others is the fact the amount of prior knowledge is smaller than it is in typical k-means applications. Nodes are only able to measure distances to their one hop neighbors, positions are unknown and the network architecture does not offer the centralized knowledge needed for basic k-means algorithm applications.

Basagni in [3] presented a Distributed Clustering Algorithm (DCA) and a Distributed Mobility-Adaptive Clustering (DMAC) algorithm. DCA is suitable for clustering of quasi-static ad hoc networks and DMAC adapts to changes in network topology caused by node mobility. Selection of clusterheads is based on weights. The weights are real numbers that characterize each

node feasibility to become a clusterhead. They are based on node connectivity (number of one hop neighbors) or on node mobility such that weights are inversely proportional to node velocity.

In Max-Min D-Cluster Formation introduced by Amis et al [4] clusterheads are selected such that they form a d -hop dominating set. By definition, if an ad hoc network is modeled as a graph $G = (V, E)$, a set C of vertices is a d -hop dominating set of G if every node in V is at most d ($d > 1$) hops away from a vertex in C . Clusterhead election is based on node id in four logical stages. Since d is an input value to the heuristic, it enables control over the density of clusterheads in the network. Authors also prove that the minimum d -hops dominating set problem is NP-complete.

Chen and Liestman [10] present a zonal algorithm to find weakly connected dominating sets. The algorithm consists of three phases. First, an input graph representing the ad hoc network is partitioned into regions of approximately size x . Then, the distributed algorithm for weakly connected sets is run in each region and finally some additional region border vertexes are added.

An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks by Bandyopadhyay and Coyle [5] targets to organize the sensors in clusters such that communication energy consumption is minimized. In single-level clustering, each sensor has same probability p to become a clusterhead. After election each sensor that becomes a clusterhead advertises itself as a clusterhead to all sensors within its radiorange. Advertisement is forwarded to all sensors that are no more than k hops away from the clusterhead. Each sensor joins to the cluster of closest clusterhead. Optimal values of p and k (with respect to communication energy) are computed under the assumption that sensors are distributed as per a homogenous spatial Poisson process.

Younis and Fahmy [6] use hybrid of node residual energy and other parameter, such as node proximity to its neighbors or node connectivity. The clustering goals are network lifetime maximization, scalability and load-balancing. It is assumed that each node has a fixed number of transmission power levels. Transmission power control is further applied to define cluster radius by the transmission power level used for intra-cluster announcements.

Targeting to the clusters that are formed *with respect to existing network topology* is the issue that makes TASC different than any existing sensor network clustering algorithm. If clusterhead selection is based on node id or node connectivity [3], [4] or randomness [5], it does not guarantee that clusterhead location is reasonable in terms of spatial attributes. If one operates with received signal

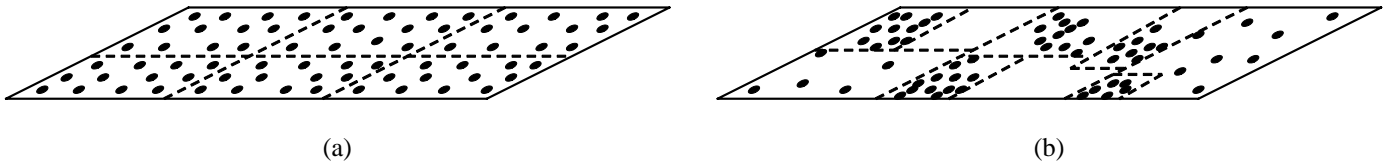


Fig. 1. Difference between uniform network clustering and TASC clustering. Density variations do not exist in figure a), because underlying node deployment is uniform. In that case equal cluster size in terms of number of nodes per cluster and in terms of cluster area guarantees balanced clusters. In figure b), non-uniform node deployment is clustered with respect to existing topology such that clustering targets to *minimize density variations in each cluster*. In non-uniform case, one cannot dictate the number of nodes per cluster or cluster area in advance, because both of them depend on the density variations that exist in the network.

strength [6], the correspondence between geographic distances and received signal strength is often weak in real applications.

In the context of clustering less attention is paid for the problem how to cluster network such that clustering improves data compression. We anticipate that the TASC uniform clustering approach will enable 1) *different data compression rates in each cluster* and 2) *improved overall compression rate in the whole network*, if suitable data compression technique as the one presented in [21] is applied. Furthermore, one is able to achieve savings in data aggregation and communication costs, if as much redundancy as possible can be eliminated in lowest possible hierarchy level in the network and clustering with respect to spatial attributes enables lower transmission power in intra-cluster communication.

III. CLUSTERING OBJECTIVES

Our clustering approach is motivated by the requirements of the sensor network domain. More specifically, a clustering algorithm should partition the network so that the nodes inside each cluster have high correlation in sensor measurements and are evenly spaced in order to maximize gains and reduce errors due to ill geometric positioning as in the case of node localization. In non-uniform network, node density variations are *globally big* but there exist subgroups of nodes such that density variations are *locally small*. We assume that 1) *each node can measure distances to its one hop neighbors* and 2) *each node has knowledge of its 2-hop neighborhood*. We set following main goal to our clustering algorithm:

The main objective of TASC is to cluster non-uniform sensor networks such that relative node density variation in individual clusters is smaller than relative node density variation in the whole network.

The type of non-uniform network clustering that we are targeting is illustrated in Figure 1b and an example of clustering outcome is shown in Figure 2. Density variations are estimated by dividing network area into a set of non-overlapping triangles such that each node locates at least in one triangle vertex like illustrated in

Figure 2b. In such triangulation, density variations are indicated by the triangle edge length standard deviation. *Relative density variation*, that is computed by dividing the edge length standard deviation by the average edge length, describes density variations such that the value is independent on actual distances.

Figure 1 shows that in contrast to uniform deployments, in more random deployments one cannot dictate a fixed number of clusters or use a grid construction since that would diminish the exploitation of correlation properties. Instead, TASC requires only the minimum number of nodes in a cluster in order to avoid the creation of single node clusters.

IV. LEADER ELECTION AND CLUSTER FORMATION

A. Algorithm

In algorithm execution, each node considers its 2-hop neighborhood. Other pre-specified parameters are the *required minimum cluster size* and the *density reachability parameter D_r* , that is explained in detail in subsection C. Leader election and cluster formation takes place in five phases:

- 1) *Each node computes its own weight based on shortest Euclidean paths in its 2-hop environment.*
- 2) *Each node broadcasts its own weight to its 2-hop neighborhood, and receives the weights of its 2-hop neighbors.*
- 3) *Each node nominates the node having biggest weight in the density-reachable subset of its 2-hop neighbors and broadcasts its nominee to its 2-hop neighborhood.*
- 4) *Each node receives all nominees in its 2-hop neighborhood, and elects the closest nominee to its leader.*
- 5) *Each node that ends up in a cluster where the total number of nodes is smaller than pre-specified minimum cluster size joins to closest cluster, where the number of nodes exceeds the required minimum cluster size.*

To be able to compute its leader, each node must send two messages to its 2-hop neighborhood and receive 2

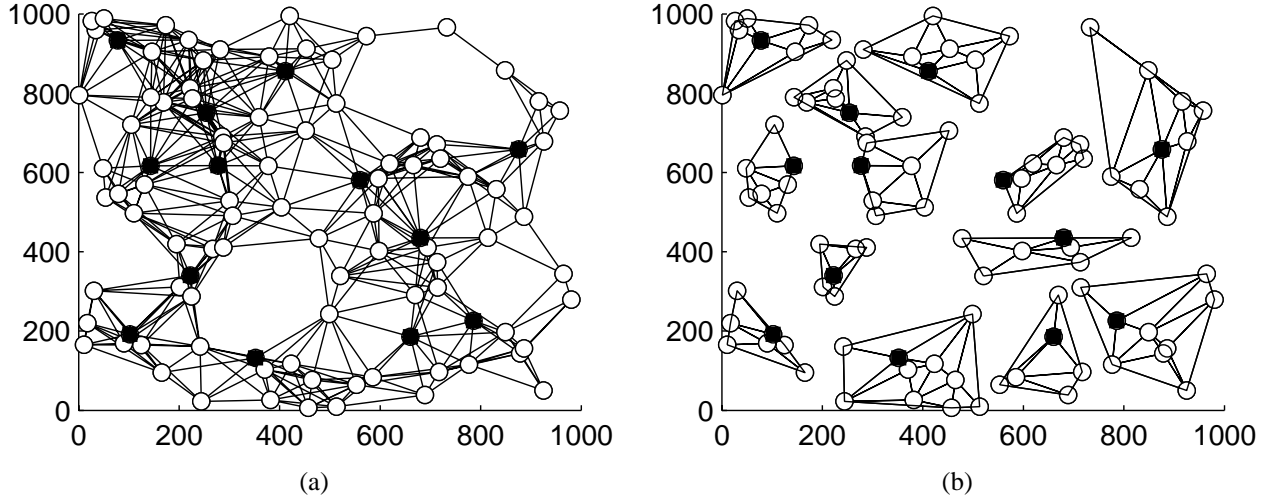


Fig. 2. An example of the clustering outcome when each node considers its 2-hop neighborhood. Figure a) shows node configuration and network connections resulting to clusters shown in figure b). Delaunay triangulation of each cluster is also shown in figure b). Cluster leaders are marked by black square.

messages from each of its two hop neighbors.

B. Weight Computation: Discovering Local Network Structure

The computation of node weights tries to achieve the reverse effect of greedy forwarding in geographic routing [22], [23]. In greedy forwarding, a node found on the path to a packet destination, forwards the packet to its neighboring node with location closest to the location of the destination. Instead of trying to forward traffic to the neighboring node that is closest to the destination, TASC all-pairs-shortest path routing is based on distance measurements to extract information about the network topology. More specifically, node weight is a measurement of two key quantities 1) *the frequency a node is found on the shortest path between pairs of nodes* and 2) *the distance contribution of the edges of that node with respect to the total length of the path*.

Consider the network in Figure 4a. If we define the weights to be the number of times a node is found on the shortest path, then we can compute a weight for each node. Node *A* for instance can be found on the paths *AB*, *AC*, *AD* and *AE* hence it will have a weight of 4. Node *C* is found on eight different paths hence it receives a weight of 8. To construct a proof of this behavior we use the principle of optimality [25]:

If S is the shortest Euclidean path between two nodes, it includes all shortest paths between all pairs of nodes that are located in path S .

Definition 1: Each node in the sensor network gets weight +1 each time the shortest Euclidean path between any pair of nodes in the network crosses that node

Inputs: 2-hop neighborhood, inter-node distance measurements, minimum cluster size, D_r
Output: Leader node

```

weight = ComputeWeight();
BroadcastToNeighborhood(weight);
If all weights received:
    Select heaviest density reachable node
    as nominee;
    BroadcastToNeighborhood(nominee);
EndIf
If all nominations have been received:
    Select the closest nominee as leader;
    BroadcastToNeighborhood(leaderID, nodeID);
EndIf
If this node is leader:
    Wait until election timeout;
    BroadcastToNeighborhood(clustermembers,
    clustersize);
EndIf
If cluster size is received:
    If clustersize < minimum cluster size:
        select the closest neighbor for which
        clustersize ≥ minimum cluster size
        and join its cluster;
    EndIf
    BroadcastToNeighborhood(leaderID, clustersize)
EndIf

```

Fig. 3. Clustering Algorithm. Each node computes its own weight in its own 2-hop neighborhood, and thus function *ComputeWeight()* is called once in each node. D_r is a parameter for density range computation explained in detail in subsection C. In the pseudocode, *clustersize* means cluster size in terms of number of nodes.

or ends at it. Paths are assumed undirected in weight computation.

Theorem 1: Let S be the shortest Euclidean path between two nodes, and let $2n + 1$ be the total number of nodes in path S . When computing all shortest paths between each pair of nodes in path S and assigning weights to each node in S as presented in Definition 1, the node that is from equal hop distance from both endpoints of path S , gets the biggest weight.

Proof: Observe path S having total number of $2n + 1$ nodes, and let c be the node located from equal hop distances from both ends of the path S . Since the total number of nodes in path S is $2n + 1$, there are n nodes on both sides of node c . Based on basic routing theory, S includes shortest paths between all pairs of nodes located in S . Thus, the weight of node c is equal to the total number of shortest paths crossing node c and ending at node c :

$$W_c = n \cdot n + 2n = n^2 + 2n \quad (1)$$

Pick then a node g from the path S so that there are $k < n$ nodes from the other side of that node. In that case there are $n + (n - k)$ nodes on the opposite side. The weight of node g is:

$$\begin{aligned} W_g &= k(n + (n - k)) + k + n + (n - k) \\ &= 2nk + 2n - k^2 \end{aligned} \quad (2)$$

When comparing the weights (1) and (2) we get

$$\begin{aligned} W_g < W_c &\Leftrightarrow 2nk + 2n - k^2 < n^2 + 2n \Leftrightarrow \\ n^2 - 2nk + k^2 &> 0 \Leftrightarrow (n - k)^2 > 0 \end{aligned} \quad (3)$$

That holds always when $0 < k < n$. ■

Corollary 1: If there are $2n$ nodes in the path S discussed in Theorem 1, two nodes in the middle get both equal biggest weight values.

Proof: The result follows from equations (1)-(3), when total number of $2n$ nodes are used. ■

Theorem 2: When weights in network graph are computed as presented in Definition 1, the node or nodes closest to the network center achieve the biggest weights.

Proof: The proof is a generalization of the discussion presented in equations (1)-(3). Observe M shortest paths that are crossing each other in one node, and mark $N = 2M$. In the symmetric case, the number of nodes in each path is $2n + 1$ and all paths are crossing each other in the midmost node c . When all shortest paths between pairs of nodes located in paths M are taken into account in weight computation, the weight of the node c is

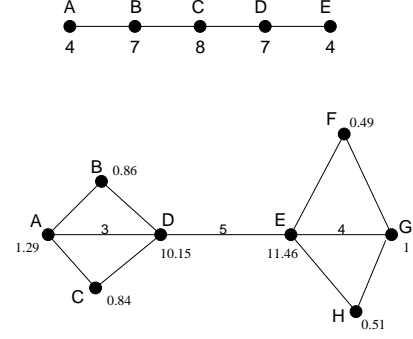


Fig. 4. Weights example.

$$\begin{aligned} W_c &= n \cdot (N - 1) \cdot n + n \cdot (N - 2) \cdot n + \dots \\ &+ n \cdot n + N \cdot n = n^2 \sum_{i=1}^{N-1} i + Nn \end{aligned} \quad (4)$$

Observe next an asymmetric case, where one of the paths has $n + k + 1$ nodes, where $1 \leq k < n$, when the rest of paths have still $2n + 1$ nodes, and paths are crossing each other in node g so that in path M_j there are n nodes on the one side and k nodes on the opposite side of node g . For other paths $M_{i, i \neq j}$, g is still the midmost node. In that case the weight of node g is:

$$\begin{aligned} W_g &= n \cdot (N - 2) \cdot n + kn + n \cdot (N - 3) \cdot n + kn + \dots \\ &+ n \cdot n + kn + kn + (N - 1)n + k \\ &= n^2 \sum_{i=1}^{N-2} i + (N - 1)kn + (N - 1)n + k \end{aligned} \quad (5)$$

When comparing weights W_c and W_g , we get

$$W_g < W_c \Leftrightarrow$$

$$\begin{aligned} n^2 \sum_{i=1}^{N-2} i + (N - 1)kn + (N - 1)n + k &< n^2 \sum_{i=1}^{N-1} i + Nn \Leftrightarrow \\ (N - 1)kn + k &< (N - 1)n^2 + n \end{aligned} \quad (6)$$

which is true under assumption $1 \leq k < n$. ■

This is enough to show that in non-uniform deployment, the node that tends to be the midmost related to all shortest communication paths (in terms of hops) gets the biggest weight. The result of Corollary 1 generalizes this result so that if some of the paths have even number of nodes, there can be several nodes with equal biggest weights in the middle.

1) *Including Distances in Weight Computation:* Although this method of computing weights would decide the central node, it does not give enough information in the cases where the paths are asymmetric such as the paths in the example network shown in Figure

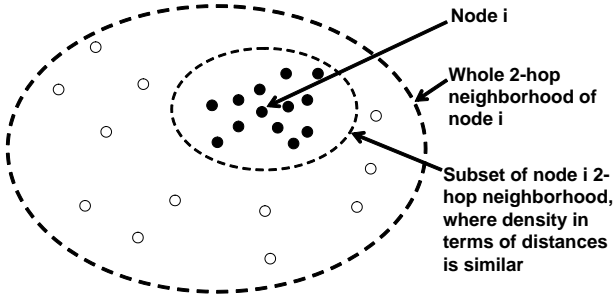


Fig. 5. The effect of density reachability. Node i figures out the subset of its 2-hop neighborhood, where density in terms of distances is similar or higher.

4b. To handle this problem, we augment the weight computation to incorporate distance information. Instead of incrementing the weight by one each time a node is used in a path, we increment the weight as a function of the distance a node contributes to the path. If a node k is found on the path from node i to node j in between nodes a and b , then the weight increment of node k is given by equation (7) where $l_{a,k}$ and $l_{k,b}$ are the lengths of the edges between nodes a and b and node k respectively and $l_{i,j}$ is the length of the whole path from node i to node j .

$$w_{ij} = \frac{l_{a,k} + l_{k,b}}{l_{i,j}} \quad (7)$$

An example of this weight computation is shown in Figure 4b.

C. Density Reachability: Grouping Similar Densities

While the information from node weights can be used to identify local centers, we would still like to construct clusters by grouping nodes in regions with similar density attributes. To achieve this goal, in addition to considering weights we need to consider additional means of pulling areas with high node densities towards the center of a cluster. To be able to do so using only distance measurements, each node must find the subgroup of its 2-hop environment, where node density in terms of distances is similar or higher to the node under consideration. The problem is illustrated in Figure 5. Each node seeks around it such subgroup of nodes, where density variations are smaller than density variations in its whole 2-hop environment. A modified version of density reachability, that is traditionally applied in data clustering to cluster spatial data in the presence of obstacles [11], [12], is applied.

The definition of density reachability is based on distance metric called *density range*, that defines the

upper bound of density variations in terms of distances such that density is considered higher or equal. One can define the *resolution* in which accuracy density reachability differentiates between more and less dense by modifying density range. To make TASC adapt to local density variations, we propose following dynamic density range definition:

Definition 2: The density range r_i of node i with respect to the given density reachability parameter Dr is the smallest disk centered at i that covers $Dr - 1$ other nodes in the vicinity of i .

In the definition, Dr is a constant number of nodes given a priori. When a bigger number is given, density range becomes longer. If density ranges are longer, each node has a bigger upper bound to the distance variations that it accepts to its density reachable set. Thus, when value Dr increases, two changes are happening in the set of density reachable nodes: 1) the set includes more nodes and 2) the set includes bigger density variations. The upper bound is the whole 2-hop neighborhood of node i , and the effect of density reachability diminishes as density range approaches the 2-hop radius. Based on the dynamic definition of the density range, we define a node to be density reachable as follows:

Definition 3: A node j is density reachable from i if there is a path from i to j where the length of every hop l satisfies the constraint: $l \leq r_i$.

Figure 6 shows an example of density reachable set definition. Node i considers its 2-hop neighborhood, and $Dr = 4$. Nodes j , k and the black nodes are density reachable from node i since there exists Euclidean path from node i to these nodes such that the length of each hop in the path is smaller or equal than r_i . Note that for the purposes of our clustering algorithm, density reachability can only expand within the 2-hop neighborhood of each node.

By applying density reachability, each node further limits the number of nodes that it can potentially nominate by considering only density reachable nodes as nomination candidates (see Figure 5). This effect pulls cluster leaders towards most dense groups in the cluster, but nomination among density reachable candidates is still based on weights.

V. EVALUATION OF CLUSTER PROPERTIES

To characterize the properties of the clustering algorithm, we run a set of simulations on a suite of 100 random scenarios. In each scenario, 100 nodes are deployed on a square deployment field of size 1000 by 1000. The simulation also assumes that the distance measurement range of the node is equal to the communication

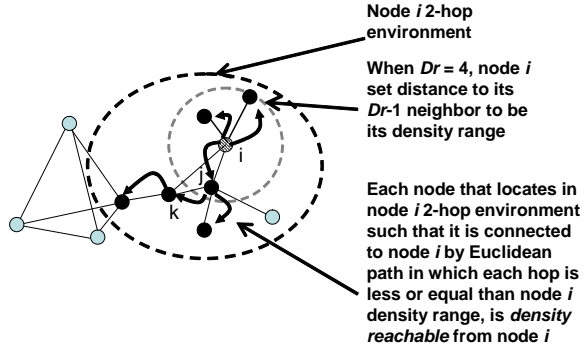


Fig. 6. Example of density reachable nodes selection. Node i select its density reachable nodes in its 2-hop neighborhood when $Dr = 4$. Black nodes indicate the subset of node i 2-hop neighbors, that is density reachable from node i .

range. In practice, we expect that the communication range is greater than the measurement range, so this assumption does not violate the fundamental properties of our clustering algorithm. Each scenario is used five times over different connectivity levels. Each time the connectivity is varied by varying maximum measurement range from 200 to 400 in steps of 50. Respective average node connectivity in each case is 10.31, 15.35, 21.09, 27.32 and 33.80. We note that even though the average connectivity is relatively high, density variations in our simulation scenarios are so high that if maximum communication range (in our simulations equal to maximum measurement range) is less than 200, all nodes are not connected to the network. For most cases, the required minimum number of nodes per cluster is set to 4. To keep communication cost and computational complexity low, each node considers its 2-hop environment. Our simulations are implemented with an in-house version of NeslSim [26], which is implemented in PARSEC. The main role of the NeslSim environment in our work is the enforcing of a distributed implementation of our clustering algorithm. The computation of shortest paths is done using the Floyd-Warshall algorithm running at each node.

1) *Cluster Evaluation Metrics*: As mentioned in section III, node density variation is given by triangle edge length standard deviation, if cluster area is divided into a set of non-overlapping triangles such that nodes locate in triangle vertices. In addition to node density variation and number of nodes per cluster, we are computing *density per cluster* in terms of nodes/m^2 . To be able to compute density per cluster, we must define *cluster area*. We do so using *Delaunay triangulation*.

Delaunay triangulation is a standard triangulation method that we found well suitable for cluster trian-

gulation. By definition, a Delaunay triangulation of a finite set of points in the plane is a triangulation that minimizes the standard deviations of the angles of the triangles, using 60 degrees as the mean. Thus, Delaunay triangulation gives an *optimal planar subdivision in terms of spatial uniformity*. The Delaunay triangulation is related to Voronoi tessellation such that the circle circumscribed about a Delaunay triangle has its center at the vertex of a Voronoi polygon.

We tie the definitions of cluster area, cluster density and node density variation into Delaunay triangulation:

Definition 4: *Cluster area* is a sum of cluster Delaunay triangle areas. The sum of Delaunay triangle areas is equal to the area of the polygon that is defined by outermost Delaunay triangle edges.

Definition 5: *Cluster density* (nodes/m^2) is the number of nodes in the cluster divided by cluster area.

Node density variation is characterized by relative standard deviation of Delaunay triangle edges:

Definition 6: *Relative node density variation* is Delaunay triangle edge length standard deviation in a cluster divided by average Delaunay triangle edge length in that same cluster.

It follows from the definition that a smaller relative node density variation indicates higher degree of uniformity. Cluster shape can be characterized by computing the *distance ratio*, that is minimum distance from polygon center point to node in polygon vertex per maximum distance from polygon center point to node in polygon vertex. Compared to the axial ratio in ellipse fitting, distance ratio gives *worst case* ratio. Three examples of cluster Delaunay triangulation are illustrated in Figure 7.

2) *Examining Cluster Uniformity*: The first experiment was to evaluate the node density variation in clusters, when distance measurements are assumed noiseless. The density reachability parameter Dr and the required minimum cluster sizes are both set to 4. Figure 8 shows that TASC outcome remains consistent when the network connectivity (the average number of neighbors/node) varies between 10 and 35. This consistency is expected, because connectivity is varied by varying maximum measurement range, but nodes are not moving. The average of Delaunay triangle edges standard deviation per cluster (percentage of the average Delaunay triangle edge length per cluster) computed from 6697 clusters outcome is 0.5211 (52.11%), and the respective standard deviation is 0.123. Comparison between underlying network node density variation and the node density variation in its clusters is illustrated in Figure 9. For each node configuration (Scenario #), the standard deviation of the Delaunay triangle edges of the whole network and the average of that particular network clusters Delaunay

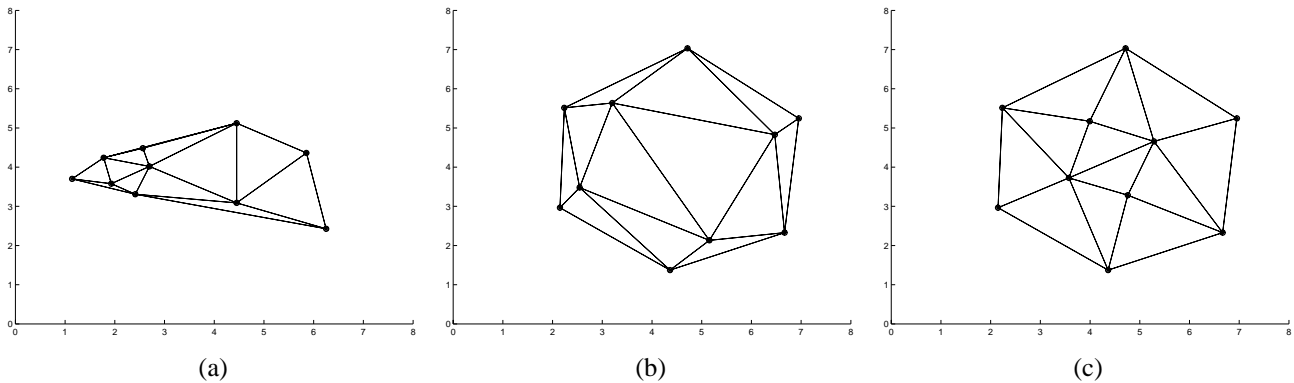


Fig. 7. Three examples of cluster Delaunay triangulation. Relative standard deviation of Delaunay triangle edge lengths is a) 0.559, b) 0.385 and c) 0.248. Cluster distance ratio is a) 0.462, b) 0.912 and c) 0.912. Since node locations in the convex hull are exactly the same in clusters b) and c), the values of cluster area distance ratio are same in both clusters, but difference in relative triangle edge length standard deviation indicates that nodes are more evenly spaced in cluster c).

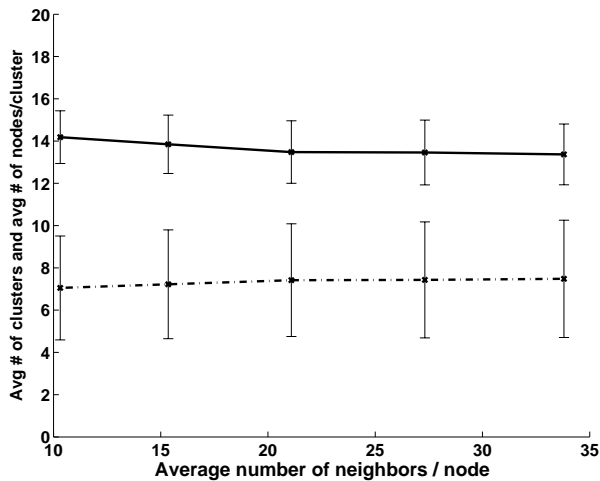


Fig. 8. Average number of clusters (upper solid line) and average number of nodes per cluster (lower dashed line). Standard deviation is shown by errorbars.

triangle edges standard deviation, is shown. The result shows obvious improvement in the degree of uniformity thus verifying that TASC is able to cluster globally non-uniform network into smaller uniform node configurations that exist in the network. Since a non-uniform network includes large density variations and TASC groups nearby nodes together, the cluster size in terms of number of nodes and in terms of cluster area is inversely proportional to cluster density like illustrated in Figure 1b. Our simulation verifies the existence of that trend, and it is shown in Figure 10. The overall average cluster distance ratio computed from 6697 clusters is 0.4966 and respective standard deviation is 0.1619. Those values are enough to show that we are not ending up with flat node chain type of clusters.

3) *Cluster Sizes and Density Reachability*: Intuitively, one would expect the average cluster size increase with increasing measurement range since the area of the 2-

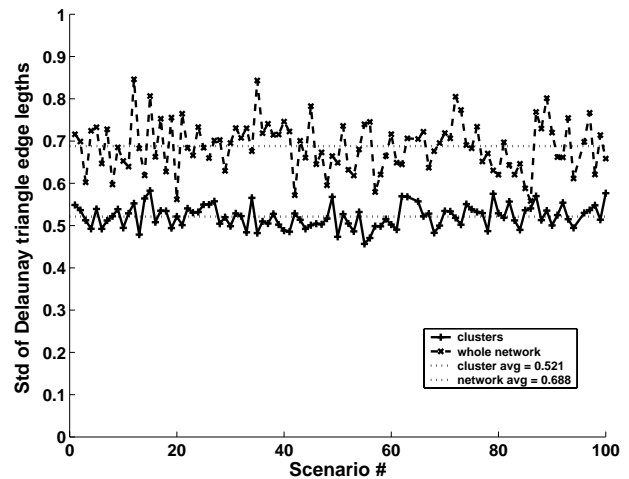


Fig. 9. Comparison between relative node density variation in the network and the average of the relative node density variation in network clusters. Delaunay triangle edge length standard deviation is represented as a percentage of the average edge length per cluster.

hop neighborhood increases. Instead, the average cluster size remains constant between 7 and 9 nodes in each of the tested cases, as illustrated in Figure 8. This is enforced by the density reachability. For each node i , the subgroup of node i 2-hop neighbors (see Figure 5) depends on node density range. Each node computes its density range based on constant parameter Dr that is given a priori. If the value of Dr is kept constant, changes in maximum measurement range do not change the density reachable subsets (see illustration in Figure 5), because the underlying node configuration remains the same. With no control on the eventual cluster density properties, the cluster sizes increase with measurement range.

As the density range begins to approach the maximum measurement range of the node, the effect of density reachability decays to the point where it cannot differ-

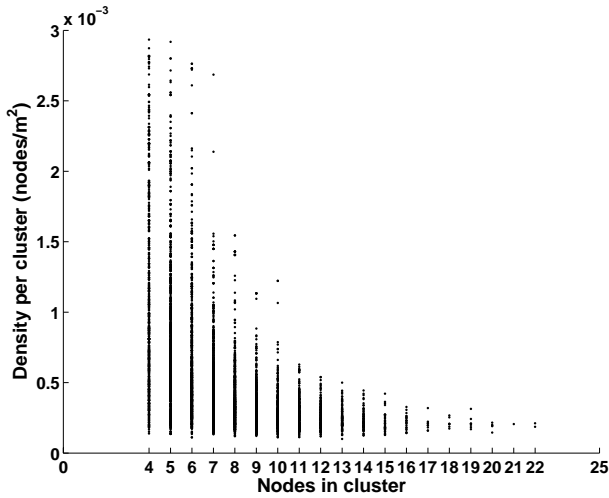


Fig. 10. TASC cluster nodes such that density variations in each cluster are smaller than density variations in the whole network. As a consequence cluster size is smaller in dense areas, but becomes bigger in sparse areas (compare to Figure 1b). Each dot in the figure shows the number of nodes per cluster and respective density per cluster. The overall shape verifies inverse proportionality between area per cluster and density per cluster.

entiate among density variations in the vicinity of the node. The size of the hop environment (how many hops) where each node executes the algorithm controls the upper bound of the cluster size. Within the chosen hop environment, that is 2 hops in our simulations, density reachability further limits the cluster size since each node density range is the upper bound of distance variations, that node accepts into its density reachable set. Thus bigger value of Dr increases node density ranges. When the node density range approaches the maximum measurement range, the set of density reachable nodes approaches the entire 2-hop neighborhood of the node. As a consequence, cluster size increases and the resolution in which accuracy TASC cluster the network with respect to local uniformity, becomes weaker.

A. Clustering in the Presence of Measurement Noise

The measurement noise is modeled as additive noise following a white Gaussian distribution that the standard deviation of which is entered as a percentage of the measured distance. The effects of measurement noise on cluster size and cluster uniformity are shown in Figure 11. We are able to obtain consistent cluster sizes with up to such noise level, where additive noise standard deviation is 30% of measured distance. A dramatic change in the cluster size consistency occurs when the noise standard deviation is increased up to 40% of measured distance. Then cluster size variation in terms of number of nodes per cluster becomes huge indicating that algorithm is not able to find density variations with

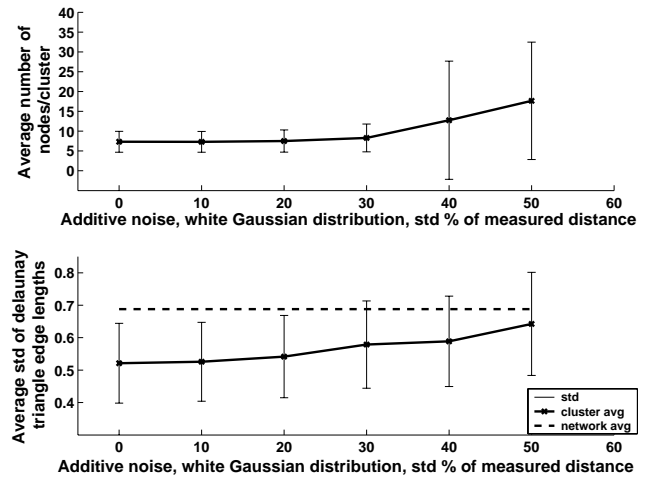


Fig. 11. Measurement noise effect to cluster size and cluster uniformity. Upper figure shows cluster size and cluster size standard deviation, lower figure average Delaunay triangle edge standard deviation (percentage of average Delaunay triangle edge length in each cluster) and average of Delaunay triangle standard deviations over all network scenarios (compare to Figure 9).

reasonable accuracy. The change is shown by errorbars that indicate cluster size standard deviation in Figure 11. Density variation in clusters stays below whole network average (compare to Figure 9), but Figure 11 shows that the relative density variation in clusters approaches network average such that relative density variation in clusters is not remarkably smaller than it is in the whole network, if noise standard deviation exceeds 30% of measured distance. When the noise standard deviation was increased up to 50% of measured distance, TASC failed to produce separate clusters in 10% of simulated network scenarios.

VI. DISCUSSION

Despite the encouraging results on the behavior of TASC, we acknowledge that there are multiple issues to consider in realistic deployments. First, the parameters of TASC should be adapted to fit the particular application needs. The option of a node running multiple instances of TASC with different parameters is worth of exploring. Second, the timing parameters of the algorithm should be more rigorously defined to comply with an actual deployment. For some systems where incremental deployment makes sense the leader election mechanisms need to be adapted to support the addition and subtraction of nodes from the network. Based on our experience from the simulation behavior and our efforts to build a scalable sensor network testbed, we believe that these changes are possible. In addition to the features described here, weight computation in TASC can reveal important properties of a network topology that should be further investigated. Even though we made the assumption that each

node considers its 2-hop neighborhood, it is possible to generalize the algorithm such that each node considers its n -hop neighborhood for any choice of n . However, the generalization to n -hop environment requires possible changes to density reachability criteria, and if the size of the hop environment increases, also computation and communication costs in each node will increase.

VII. CONCLUSIONS AND FUTURE WORK

Our evaluation has shown that by using the novel combination of weights and density reachability, TASC achieves the desired behavior: *It can decompose large non-uniform networks into smaller locally uniform clusters*. Simulations with noise indicate that TASC tolerates noisy distance measurements up to level, where the standard deviation of Gaussian noise is 30% of measured distance. In addition to the previously mentioned applications, the distribution of weights inside a network can also be used as an indicator for spatial regularity in a specific deployment. One possible research avenue would be to develop an algorithm for making localized decisions on how nodes should reposition themselves to improve sampling uniformity. Another possibility is to repeat the weight-based election process to construct hierarchies. The initial results are encouraging and suggest the more rigorous evaluation of TASC needs in more realistic deployment settings. As part of our future work, we plan to test TASC in the context of our 3-D testbed. The two immediate uses of TASC in our 40-node testbed is to assist with ad-hoc node localization and in radio frequency allocation through the meaningful, spatial decomposition of a dense Zigbee network.

ACKNOWLEDGMENT

This work was partially funded by the National Science Foundation award #0448082 and by scholarships from Nokia Foundation and Emil Aaltonen Foundation.

REFERENCES

- [1] D. Ganesan, S. Ratnasamy, H. Wang and D. Estrin, *Coping with irregular spatio-temporal sampling in sensor networks*, in Proceedings of Second Workshop on Hot Topics in Networks (HotNets-II), November 2003
- [2] S. Patten, B. Krishnamachari and R. Govindan, *The Impact of Spatial Correlation on Routing with Compression in Wireless Sensor Networks*, Proceedings of the Third International Symposium on Information Processing in Sensor Networks (IPSN'04), April 26 - 27, 2004, Berkeley, California, USA
- [3] S. Basagni, *Distributed Clustering for Ad Hoc Networks*, International Symposium of Parallel Architectures, Algorithms and Networks (I-SPAN'99), Fremantle, Australia, June 23-25, 1999.
- [4] A. D. Amis, R. Prakash, T. H. P. Vuong, D. T. Huynh, *Max-Min D-Cluster Formation in Wireless Ad Hoc Networks*, Proceedings of IEEE INFOCOM 2000.
- [5] S. Bandyopadhyay, E. J. Coyle, *An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks*, Proceedings of IEEE INFOCOM 2003.
- [6] O. Younis, S. Fahmy, *Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach*, Proceedings of IEEE INFOCOM 2004.
- [7] McQueen, J. B., *Some Methods for Classification and Analysis of Multivariate Observations*, Proceedings of the Fifth Symposium on Math, Statistics and Probability (pp. 281-297), 1967.
- [8] Kanugo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., Wu, A. Y., *A Local Search Approximation Algorithm for k-Means Clustering*, Proc. of the 18th Annual ACM Symp. on Computational Geometry, 2002, 10-18.
- [9] Ghiasi, S., Srivastava, A., Yang, X., Sarrafzadeh, M., *Optimal Energy Aware Clustering in Sensor Networks*, Sensors 2002, 2, 258-269.
- [10] Chen, Y. P., Liestman, A. L., *A Zonal Algorithm for Clustering Ad Hoc Networks*, International Journal of Foundations of Computer Science, 14(2):305-322, April 2003.
- [11] Ester, M., Krieger, H-P., Sander, J., Xu, X., *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases With Noise*, 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96), Portland, Oregon, 1996.
- [12] Zaane, O. R., Lee, C-H., *Clustering Spatial Data in the Presence of Obstacles: a Density-Based Approach*, Sixth International Database Engineering and Applications Symposium (IDEAS 2002), Edmonton, Alberta, Canada, July 17-19, 2002.
- [13] A. Harter and A. Hopper, *A New Location Technique for the Active Office*, IEEE Personal Communications, vol. 4, No. 5, October 1997, pp.42 - 47.
- [14] N. B. Priyantha, A. Chakraborty, H. Balakrishnan, *The Cricket Location-Support System*, Proceedings of 6th ACM Mobicom, Boston, MA, August 2000.
- [15] A. Savvides, H. Park and M. B. Srivastava, *The n-hop Multilateration Primitive for Node Localization Problems*, Proceedings of Mobile Networks and Applications, 8, 443-451, 2003
- [16] Ubisense website, <http://www.ubisense.net>
- [17] N. B. Priyantha, H. Balakrishnan, E. Demaine, S. Teller, *Anchor-Free Distributed Localization in Sensor Networks*, LCS Tech. Report 892.
- [18] Y. Shang, W. Ruml, *Improved MDS-Based Localization*, Proceedings of IEEE INFOCOMM, Hong Kong, March 7-11, 2004
- [19] Ji, X., Hongyuan, Z., *Sensor Positioning in Wireless Ad-hoc Sensor Networks Using Multidimensional Scaling*, IEEE Infocom, March 7-11, 2004.
- [20] Klein, D., Kamvar, S. D., Manning, C. D., *From Instance-level Constraints to Space-level Constraints: Making the Most of Prior Knowledge in Data Clustering*, The Nineteenth International Conference on Machine Learning (ICML-2002), Sydney, Australia, July 8-12, 2002.
- [21] J. Chou, D. Petrovic, K. Ramchandran, *A Distributed and Adaptive Signal Processing Approach to Reduce Energy Consumption in Sensor Networks*, Proceedings of IEEE INFOCOM 2003.
- [22] GPSR B. Karp and H.T. Kung, *GPSR: Greedy Perimeter Stateless Routing for Wireless Networks*, Proceedings of Mobicom 2000
- [23] LAR Y.B Ko and N. Vaidya, *Location Aided Routing (LAR) in Mobile Networks*, Proceedings of ACM/IEEE MobiCom, pp. 66-75, October 1998
- [24] D. Lymberopoulos, A. Barton-Sweeny, A. Savvides, *Sensor Localization and Camera Calibration in Networks of Low-Power Imagers*, Yale ENALAB Technical Report 080501, August 2005
- [25] D. Bertsekas, *Dynamic Programming and Optimal Control*, vol 1, Athena Scientific, 2000
- [26] NeslSim Website <http://www.ee.ucla.edu/saurabh/NESLsim/>