Appl. Comput. Math. **6** (2007), no.2, pp.162-180

GRAPH THEORETIC CLUSTERING ALGORITHMS IN MOBILE AD HOC NETWORKS AND WIRELESS SENSOR NETWORKS
SURVEY

KAYHAN ERCIYES †, ORHAN DAGDEVIREN ‡, DENIZ COKUSLU ‡, DENIZ OZSOYELLER††, §

Abstract. Clustering in mobile ad hoc networks (MANETs) and wireless sensor networks (WSNs) is an important method to ease topology management and routing in such networks. Once the clusters are formed, the leaders (coordinators) of the clusters may be used to form a backbone for efficient routing and communication purposes. A set of clusters may also provide the underlying physical structure for multicast communication for a higher level group communication module which may effectively be used for fault tolerance and key management for security purposes. We survey graph theoretic approaches for clustering in MANETs and WSNs and show that although there is a wide range of such algorithms, each may be suitable for a different cross-layer design objective.
Keywords : clustering, mobile ad hoc networks, wireless sensor networks, dominating sets, spanning trees, fault tolerant clustering

1. Introduction

Mobile ad hoc networks do not have any fixed infrastructure and consist of wireless mobile nodes that perform various data communication tasks. MANETs have potential applications in rescue operations, mobile conferences, battlefield communications etc. Conserving energy is an important issue for MANETs as the nodes are powered by batteries only. Clustering has become an important approach to manage MANETs. In large, dynamic ad hoc networks, it is very hard to construct an efficient network topology. By clustering the entire network, one can decrease the size of the problem into small sized clusters. Clustering has many advantages in mobile networks. Clustering makes the routing process easier, also, by clustering the network one can build a virtual backbone which makes multicasting faster. However, the overhead of cluster formation and maintenance is not trivial. In a typical clustering scheme, the MANET is firstly partitioned into a number of clusters by a suitable distributed algorithm. A clusterhead is then allocated for each cluster which will perform various task on behalf of the members of the cluster. The performance metrics of a clustering algorithm are the number of clusters and the count of the *neighbor nodes* which are the adjacent nodes between clusters that are formed.

Wireless sensor networks is a key technology for new ways of interaction between computers and the physical environment. However, gathering data is challenging because of the limited and energy constrained computing resources. Since sensor nodes are densely deployed, redundant data may occur. Clustering the network efficiently is an important way of managing large number of sensor nodes.

In this survey, we search various graph theoretic algorithms for clustering in MANETs and WSNs. Section 2 outlines clustering algorithms for MANETs whereas clustering algorithms

†Ege University International Comp. Inst., 35100 Bornova, Izmir, Turkey,
email: kayhan.erciyes@ege.edu.tr
‡Izmir Institute of Technology, Computer Eng. Dept., Urla, Izmir 35430, Turkey,
††Izmir University of Economics, Computer Eng. Dept., Balcova, Izmir, 35330, Turkey
§*Manuscript received 29 June, 2007* .

for WSNs are reviewed in Section 3 and the Conclusions Section provides the overview of the clustering approaches for both types of networks.

2. Clustering in Mobile Ad Hoc Networks

In this section, we review clustering algorithms in MANETs using dominating sets and spanning trees.

2.1. **Clustering Using Dominating Sets.** A dominating set is a subset S of a graph G such that every vertex in G is either in S or adjacent to a vertex in S [66]. Dominating sets are widely used in clustering networks [8]. Dominating sets can be classified into three main classes, Connected Dominating Sets (CDS), Weakly Connected Dominating Sets (WCDS) and Independent Dominating Sets (IDS) [35] as shown in Fig.1.

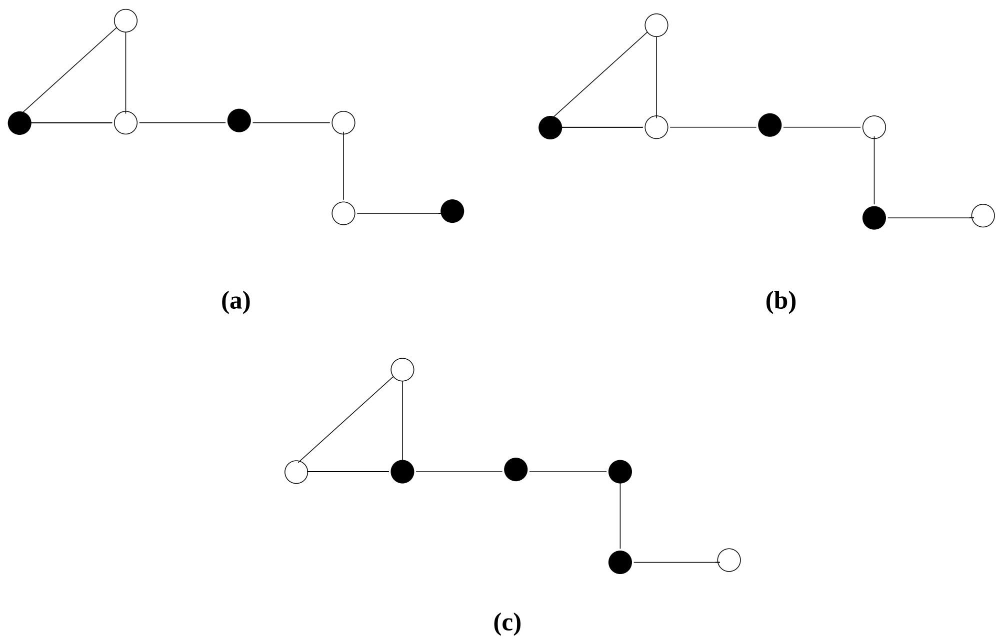

Figure 1. (a) IDS (b) WCDS (c) CDS

- *Independent Dominating Sets:* IDS is a dominating set S of a graph G in which there are no adjacent vertices. Fig.1.a shows a sample independent dominating set where black nodes show clusterheads.
- *Weakly Connected Dominating Sets (WCDS):* A weakly induced subgraph S_w is a subset S of a graph G that contains the vertices of S, their neighbors and all edges of the original graph G with at least one endpoint in S. A subset S is a weakly-connected dominating set, if S is dominating and (S_w is connected [7]. Black nodes in Fig. 1.b show a WCDS example.
- *Connected Dominating Sets:* A connected dominating set (CDS) is a subset S of a graph G such that S forms a dominating set and S is connected. Fig.1.c shows a sample CDS.

Various algorithms exist for clustering using IDS, WCDS and CDS.

2.1.1. *Clustering Using IDS.* By using independent dominating sets, one can guarantee that there are no adjacent clusterheads in the entire graph. This minimizes the number of dummy clusters in the network.

Baker and Ephremides [4] proposed an independent dominating set algorithm called *highest vertex ID*. In this algorithm, each vertex scans its closed neighbor set and chooses the highest id neighbor as a clusterhead. A very similar algorithm to the highest id algorithm is the *lowest id algorithm* by Gerla and Tsai [24] where each vertex with the lowest id within its closed neighborhood is selected as a clusterhead. Gerla and Tsai developed another algorithm to find the independent dominating sets called the *highest degree algorithm*. In this algorithm, each vertex with the highest degree in its closed neighborhood is selected as the clusterhead [24].

Although these algorithms are considered as important algorithms, Chen et al. [9] proposed that these algorithms are not working correctly for some graphs. In some situations, some independent sets cannot form a dominating set. To solve this incorrect operation, Chen et al. developed the *k-distance independent dominating set* algorithm. By this algorithm, Chen et al. adds one more rule to the above algorithms such that in a k-distance dominating set, every clusterhead must be at lest $k+1$ distant from each other [56].

2.1.2. *Clustering Using WCDS.* Although independent dominating sets are suitable for constructing optimum sized dominating sets, they have some deficiencies such as lack of direct communication between clusterheads. In order to obtain the connectivity between clusterheads, WCDSs can be used to construct clusters. The WCDS was first proposed for clustering in ad hoc networks by Chen and Liestman [11]. In this algorithm, the graph is first partitioned into non-overlapping regions. This is done by growing a spanning forest of the graph and at the end of this phase, the subgraph induced by each tree defines a region. Then a greedy approximation algorithm is executed to find a small WCDS of each region. The greedy algorithm is based on Guha and Khuller's second algorithm [25] which is described in the next paragraph. Once small WCDSs are constructed, the union of these WCDSs constructs the dominating set of the entire graph. Some additional vertices from region borders can be added to the dominating set to ensure that the final dominating set of G is weakly-connected. This type of clustering is called *zonal clustering*.

Han and Jia [34, 36] proposed a WCDS based clustering algorithm. They first construct a maximal independent set by using degrees of the nodes as the decision heuristic. Nodes which have the highest degree among all its one-hop neighbors become root clusterheads. If a node has neighbors with higher degrees than itself , but these neighbors already become the cluster members of other clusters, then the node determines itself as a non root clusterhead. At the end of this phase every node is assigned to an area around its clusterhead. These rules ensure that the area belonging to a root clusterhead is a WCDS. But the combination of the areas does not necessarily generate a WCDS for the entire graph, therefore some nodes at the borders of independent areas are selected as additional clusterheads in order to ensure that the entire graph becomes a WCDS. Chen and Liestman [11] and Alzoubi et al. [1] are other well known WCDS construction algorithms.

2.1.3. *Clustering Using CDS.* CDSs have many advantages in network applications such as ease of broadcasting and constructing virtual backbones [60], however, when we try to obtain a connected dominating set, we may have undesirable number of clusterheads. So, in constructing connected dominating sets, our primary problem is minimum connected dominating set decision problem.

Guha and Khuller [25] proposed two centralized greedy algorithms for finding suboptimal connected dominating sets. In the first algorithm, initially all vertices are white colored. In the first step, the algorithm selects the node with the maximum number of white neighbors as a dominating node. The dominating node becomes black, and its neighbors become grey. Then the algorithm iteratively scans the grey nodes and their white neighbors. In each iteration, the grey node or the pair of nodes with the maximum number of white neighbors is selected as a cluster node. This iteration process continues until no white vertex is left in the graph. In the second algorithm, white vertex with the maximum number of white neighbors is selected as a dominating node. This iteration lasts until no white colored vertex is left in the graph. When the iteration ends, the algorithm re-colors some gray nodes to black so that the dominating set becomes connected. Das and Bharghavan [16, 17] provided distributed implementations of Ghua and Khuller's algorithms [25].

Wu and Li [67, 68, 69] improved Das and Bhraghavan's distributed algorithm as a localized distributed algorithm for finding connected distributed sets in which each node only needs to know its distance-two neighbor [7]. In Wu and Li's algorithm [68], initially each vertex marks itself as F indicating that it is not dominated yet. In the first phase, a vertex marks itself as T if any two of its neighbors are not connected to each other directly. In the second phase, a T marked vertex v changes its mark to F if either of the following conditions is met:

(1) $\exists u \in N(v)$ which is marked $T : N[v] \subseteq N[u]$ and $id(v) < id(u)$
(2) $\exists u, w \in N(v)$which is marked $T : N(v) \subseteq N(u) \bigcup N(w)$ and $id(v) = min\{id(v), id(u), id(w)\}$

Dai and Wu [18, 71] proposed an extended localized algorithm for finding CDS. The algorithm is based on Wu and Li's [68] algorithm with improved pruning rules. Dominant pruning rules with more than two connector hosts were not considered in early studies due to the following two assumptions: 1) testing the coverage of multiple hosts could be costly and 2) only a few neighbor sets need to be covered by three or more other hosts. However, Dai and Wu [18] showed that these assumptions are not always true. According to this algorithm, first phase works same as Wu and Li's algorithm [68], but in phase two, instead of rule 1 and rule 2, Rule k pruning rule is applied to eliminate the dummy clusterheads. According to Rule k, if neighbors of a clusterhead is dominated by more than two directly connected clusterheads, it can be eliminated. With this work, Dai and Wu [18] showed that Rule k can be implemented with local neighborhood information that has the same complexity as Rule 1 and less complexity than Rule 2.

Nanuvala [54] has extended the Wu's CDS Algorithm and added a third pruning rule in order to reduce the size of the resulting CDS. Li et al. [47] proposed an algorithm to construct CDS with bounded diameters, this algorithm first finds a maximal independent set and then selects some other nodes as clusterheads in order to build a CDS at the end. Gao et al. [26] also proposed a CDS algorithm which uses the maximal independent set as the basis of construction of the CDS. In the first phase, a maximal independent set is constructed. In the second phase, a selected node tries to create paths connecting all its two and three hops away clusterheads.

Yan et al. [72] proposed a heuristic algorithm for minimum connected dominating set. The algorithm first calculates a weight for each node indicating node's uptime and its amount of power left. It then uses weight parameter and some rules from Wu and Li's algorithm [68] in selecting the clusterheads. By using this heuristic, Yan et al. [72] make a better estimation on the stability of the backbone topology.

Wan et al. [70] proposed a distributed algorithm for finding a CDS. This algorithm consists of two phases. The first phase constructs a maximal independent set (MIS) using a rooted spanning tree which is constructed at the beginning of the phase. The second phase constructs

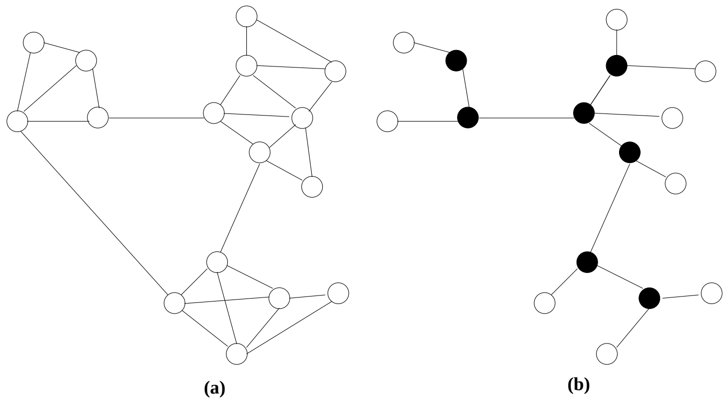

FIGURE 2. (a) A MANET (b) Its Minimum Spanning Tree

a dominating tree from the MIS, whose internal nodes would become a CDS. The algorithm uses $O(n)$ messages and takes $O(n)$ time where n is the number of nodes. Liu et al. [48], improved Wu and Li's [68] algorithm by adding a third phase elimination. In the additional third phase, the algorithm searches redundant clusterheads. A clusterhead is eliminated if it is dominated by two of its clusterhead neighbors. The distributed algorithm has time complexity $O(n^2)$ and message complexity $O(n)$ where n is the number of nodes.

Cokuslu et al. [12] added some extra heuristics to Wu and Li's algorithm [68] and provided more reliable results. First, they determined some situations that a node cannot change its color after the first phase. They also consider the degree of a node when marking it. They added two more pruning rules to the Wu and Li's algorithm by using color $GRAY$ which is used to indicate potential clusterheads. Heuristics shorten the runtime of the algorithm and total of four pruning rules results in a less redundant clusterheads compared to Wu and Li's algorithm.

2.2. **Clustering Using Spanning Tree.** An undirected graph is defined as $G = (V, E)$, where V is a finite nonempty set and $E \subseteq V \times V$. The V is a set of nodes v and the E is a set of edges e. A graph $G_S = (V_S, E_S)$ is a spanning subgraph of $G = (V, E)$ if $V_S = V$. A spanning tree of a graph is an undirected connected acyclic spanning subgraph. Intuitively, a minimum spanning tree (MST) for a graph is a subgraph that has the minimum number of edges for maintaining connectivity [27]. A sample MANET and a minimum spanning tree constructed can be seen in Fig. 2 where any node other than the leaf nodes which are shown by black color depict a connected set of nodes.

Gallagher et. al. [28] proposed a distributed algorithm which determines a minimum-weight spanning tree for an undirected graph that has distinct finite weights for every edge. Aim of this algorithm is to combine small fragments into larger fragments with outgoing edges. A fragment of an MST is a subtree of the MST. An outgoing edge is an edge of a fragment if there is a node connected to the edge in the fragment and one node connected that is not in the fragment. In this algorithm, each node starts with each as a fragment and ends with the MST as a fragment.

The authors defined three possible nodes states: *Sleeping*, *Find* and *Found* state. Initially all nodes are in *Sleeping* state and either spontaneously awaken to initiate the overall algorithm or is awakened by a message from another node. A node in *Find* state searches for the minimum-weight outgoing edge to combine with another fragment. Combination rules of fragments are related with levels. A fragment with a single node has the level $L = 0$. Suppose two fragments F at level L and F' at level L';

- If $L < L'$, then fragment F is immediately absorbed as part of fragment F. The expanded fragment is at level L'.
- Else if $L = L'$ and fragments F and F' have the same minimum-weight outgoing edge, then the fragments combine immediately into a new fragment at level L+1
- Else fragment F waits until fragment F' reaches a high enough level for combination.

Under the above rules, the combining edge is called the core of the new fragment. The two nodes adjacent to the core exchange messages on the core branch itself, allowing each of these nodes to determine both the weight of the minimum outgoing edge and the side of the core on which this edge lies. The upper bound for the number of messages exchanged during the execution of the algorithm is 5N$\log_2$N +2E, where N is the number of nodes and E is the number of edges in the graph. A message contains at most one edge weight and $\log_2$8N bits. Worst case time for this algorithm is *O(E+Nlog$_2$N)*.

Awerbuch [2] proposed an algorithm in which each tree will hook itself on edge leading to the neighboring tree of maximum level instead of hooking itself on its minimum weight edge. The algorithm has two stages : Counting Stage and MST Stage. In this algorithm, if the tree waits for a long time, it will be rewarded properly. This is the main idea behind the Counting stage of the algorithm. The MST stage assumes knowledge of V, the total number of nodes, which is provided by the previous Counting stage. This has a lot of similarity with Gallagher et. al.'s algorithm [28]. The only difference is that level increases are originated by many nodes, not only by the root node. The MST stage is performed in two phases. The first phase runs an algorithm identical to Gallagher, Humblet and Spira's algorithm [28], and terminates when all trees reach the size of $\Omega(V/logV)$. The new algorithmic ideas are introduced in the second phase. Algorithm updates the levels in a very accurate fashion, which prevents small trees waiting for big trees and speeds up the algorithm. The algorithm requires *O(E + Vlog V)* messages and *O(V)* time where E is the number of edges.

The algorithms proposed by Gallagher et. al. [28] and Awerbuch [2] uses $Tree - join - tree$ approach. Yao-Nan Lien [46] proposed a distributed minimum spanning tree algorithm that uses $Node - join - tree$ approach. The algorithm is initialized from a single node such that there is no need to wake up all nodes at the beginning as stated in Gallagher, Humblet and Spira 's algorithm [28]. Starting from any node, an MST fragment (M) grows from a single node to complete MST iteratively by drafting nodes into M. In each iteration, each terminal node of M tries to draft more nodes into M by sending a $Follow - me$ message to each of its neighboring nodes except its preceding node. Each neighboring node decides whether or not to hook itself to M as a new terminal node based on its own local information. The new terminal nodes continue the drafting process iteratively until the end of the iteration when there is no node that wants to hook to M. A complete MST is formed if all nodes are included in M. The algorithm needs at most *(2e+n(n-1)/4)* messages in $O(n^2)$ time where e is the number of edges, n is the number of nodes. In the best case, it needs *2e* messages in *O(nlogn)* time.

Ahuja and Zhu [3] proposed a distributed minimum spanning tree algorithm which uses the $Tree - join - tree$ approach as used in the et. al.'s Algorithm [28]. The algorithm works in phases. In phase 1 of the algorithm, each node needs to do the following:

- Sets the minimum adjacent edge as $Branch$ and notifies its decision to the node on the other side of the edge.
- Learns the $nodeID$s at the other side of its adjacent edges.
- Participates in the construction of underlying spanning tree.

By cutting the number of fragments to at least one half in each phase, it needs at most $O(logn)$ phases. In the worst case, the algorithm needs at most $(2m + 2(n-1)log(n/2))$ messages and $(2dlogn)$ time, where d is the diameter of the network and m is the total message number in the first phase. In the best case, it needs only $2m$ messages in $2d$ time. On the average, the algorithm needs only $O(m)$ messages and $O(d)$ time.

Garay et al. [32] provide a modified, controlled version of the Gallagher et. al.'s algorithm [28]. The algorithm is able to achieve the following:

- Upon termination, the number of the fragments is bounded above by $n \,/\, 2^{numberofphases}$.
- Throughout the execution, the diameter of every fragment F satisfies *Diameter (Fragment)* $< 3^{numberofphases}$.

The time complexity of the algorithm is $O(Diam(G){+}n^{0{,}614})$ where n is the number of nodes.

Banerjee and Khuller [5] proposed a protocol based on a spanning tree for hierarchical routing in wireless networks [44, 64]. In their scheme, a cluster is a subset of vertices whose induced graph is connected. These subsets are chosen by considering the cluster size and the maximum number of clusters to which a node can belong. Banerjee and Khuller [5] defined their clustering problem in a graph theoretic framework, and present an efficient distributed solution that meets all the desirable properties. The algorithm proceeds by finding a rooted spanning tree of the graph. The algorithm creates a BFS tree and then visits each vertex in the the tree in post order. The time complexity of the algorithm is $O(|E|)$ where E is the number of edges.

In MANETs, member nodes of same cluster must be close in order to minimize intra-cluster communication. To achieve this, a network can be divided into clusters where two member nodes of same cluster are at most k hops from each other. The algorithmic complexity of k-clustering is known to be NP-Complete for simple undirected graphs [23]. Fernandess and Malkhi proposed a k-clustering framework to divide the network into non-overlapping clusters [23]. They modeled the MANETs as unit disk graphs and introduce a two phase distributed polynomial time and message complexity approximation solution with $O(k)$ worst case ratio over the optimal solution. In the first phase, a spanning tree of the network is constructed. They showed that the tree can be built as a BFS (Breadth First Search), DFS (Depth First Search) or MCDS (Minimum Connected Dominating Set) structure, and compared these structures. In the second phase, the spanning tree is partitioned into subtrees with bounded diameters by a distributed asynchronous algorithm. Detailed analysis of the algorithm is provided but simulation results are not given by the authors.

Gentile et. al. proposed a distributed routing and clustering algorithm to minimize the overhead messages for multi-hop routes with minimum power in a MANET [29]. They assumed a piece-wise linear model for the motion of the nodes. By applying this model, they proposed a kinetic minimum spanning tree construction. After the spanning tree is constructed, each node sets its state to clusterhead initially. To create new clusters, the clusterheads join each other with the control of a single parameter that adjusts clustering. Authors showed that the worst case for the time complexity is $O(n^2)$ where n is the number of nodes. However, in the framework of their mobility model, the simulation results showed that stable clusters are formed.

A topology graph for a mobile ad hoc network [59] can have any arbitrary structure. Srivastava and Ghosh [61] proposed a distributed algorithm for constructing a rooted spanning tree of a dynamic graph with the root being located towards the center of the graph. They described the α *cone* as the origin concerned node and bounded by two rays with an angle α between them. The attribute *color* is given for each node to define their states. The algorithm proposed works in two stages. In the first stage, it finds a spanning forest. In the second stage the trees of the spanning forest are connected together to produce a tree with a single root. The authors proposed a priority-based algorithm for the second stage. This algorithm is mainly based on spanning tree construction also spanning forests can be treated as unconnected clusters. A clustering based analysis and simulation are not provided by the authors.

Johansson and Motyckova [40] proposed a clustering algorithm to limit the maximum number of elements in a cluster and the maximum hop between the member nodes in the same cluster as mentioned in Fernandess and Malkhi's framework. They proposed a simple algorithm based on neighbor list exchange. The nodes choose their cluster or create a new cluster as a clusterhead. The clustering message is forwarded by the member nodes to the clusterhead. It was shown that the message complexity is $\Theta(d^2)$ where d is the network's diameter. Authors did not provide any simulation results.

Dagdeviren et. al. proposed the Merging Clustering Algorithm (MCA) [19, 20] which finds clusters in a MANET by merging the clusters to form higher level clusters as mentioned in Gallagher et. al.'s algorithm [28]. However, they focused on the clustering operation by discarding the minimum spanning tree. This reduces the message complexity from $O(nlogn)$ to $O(n)$. The second contribution is to use upper and lower bound parameters for clustering operation which results in balanced number of nodes in the clusters formed. The lower bound is limited by a parameter which is defined by K and the upper bound is limited by $2K$. The last contribution is the clusterhead (leader) selection method as an alternative to the core of the fragment in [28]. Simulation results showed that the algorithm creates balanced clusters and terminates in reasonable amount of time.

Multicast is the delivery of the same message from a source node to a number of target nodes which are members of a group. It is provided by a pre-built tree structure. Multicast routing in MANETs is very important in order to support group communication under unreliable conditions and it is a hot research topic [10, 13, 15, 49, 55, 76]. In this paper, we surveyed the multicast algorithms that are targeted for both cluster and spanning tree formation.

Ohta et. al. [55] proposed an algorithm in which, clusterhead forms a spanning tree in the cluster to collect information from all nodes. For clustering and spanning tree formation, the clusterhead periodically broadcasts *clusterMEmber Packet (MEP)* to inform that the clusterhead stays at the current cluster. Nodes which received this message sets its clusterhead, forwards it to the downstream nodes, sends *clusterMember Acknowledge Packet (MAP)* to clusterhead. The clusterhead collects the information and makes a list of all nodes in the cluster. Also it collects the information on neighboring clusters and adjusts the number of nodes in the clusters by division and merging operations. Gateway nodes can listen to the message traffic of different clusters and forward any message to their clusterheads for further information. A theoretical analysis of the algorithm is not given in the paper.

A virtual backbone in a MANET can be built as a connected path between some selected nodes which are generally clusterheads, mainly aimed to provide network wide routing as well as multicasting. Dominating set based algorithms which were introduced in section 2.1 construct a backbone architecture. Ya-Feng et. al [77] focused on the construction of the optimal Virtual Multicast Backbone (VMB) with the fewest forwarding nodes to decrease overhead and cost, due to the scarce resource in ad hoc networks. Instead of conventional Steiner tree model, the optimal shared VMB in ad hoc networks is modeled as Minimum Steiner Dominating Set (MSCDS) in Unit-Disk Graphs (UDG), construction of which is NP-hard. One-hop algorithm and a *d-hop* algorithm is proposed for approximating MSDCS. One-hop algorithm is divided into the steps below:

(1) Find a maximal independent set I in *G(V)*

(2) In G, apply the Steiner tree algorithm in [62] to find a Steiner tree T for the subset I, with all edges having unit weight. The final solution is the set of the nodes of T.

The *One-hop* Algorithm constructs a hierarchical VMB. However, when deployed in sparse UDG, where most multicast nodes are two or more hops apart from each other, it mostly results in trivial single-node multicast clusters and consequently flat VMB. This implies that *One-hop* Algorithm is not suitable for VMB construction in sparse ad hoc networks. To address this issue, an extended *d-hop* Algorithm is proposed with detailed description of distributed implementation, with a constant approximation ratio. The *d-hop* Algorithm finds a *d-MIS* among multicast nodes, which is also a *d-hop* dominating set of the multicast group, and then each node in the *d-MIS* becomes a clusterhead and forms a *d-hop* cluster with all its d-neighbors. In the cluster, some multicast nodes are further chosen to dominate multicast nodes of the cluster. These nodes are connected to the clusterhead with the shortest paths. Among the clusters, a Steiner tree is used to connect all clusterheads. The distributed implementation of *d-hop* Algorithm for constructing an SCDS of multicast nodes has a time complexity $O(dn)$ where d is the graph diameter and a message complexity of $O(nlog(n))$ if d equals to 1, otherwise $O(n^d)$.

Choi et. al. proposed the Implicit Cluster-based Overlay Multicast Protocol Exploiting Tree Division (ICOM-TD) protocol for group communication in MANETs [13]. They proposed the network architecture as a combination of Multicast Members (MMs), Multicast Sources (MSs) and Multicast Receivers (MRs). MMs are the multicast group members as well as multicast cluster nodes. A MS is the clusterhead and a MM which generates and sends multicast data. MR is the receiver of the multicast data and also the cluster member. Multicast trees are created by clusterheads which are created with their geographical information by MSs. The clustering formation phase of this algorithm is similar with Ohta's scheme. Initially, MRs which want to join multicast cluster send a *JoinRequest* to a MS of the multicast cluster through unicasting. This message includes the x, y coordinates and $nodeID$. Whenever a MR wants to join this cluster, it acknowledges with another *JoinRequest* message. Clusterhead collects these member information and updates the member list. Clusterheads seperate the cluster information by using geographic coordinates different than Ohta's algorithm. After the clusterhead constructs clusters, it sends *JoinReply* messages back to the MRs which wait to join clusters. Authors did not provide any analysis related to clustering.

3. Clustering in Wireless Sensor Networks

Clustering is an effective topology control approach in WSNs which can increase network scalability and lifetime. Sensor node clustering is a very important optimization problem. In order to maintain a certain degree of service quality and a reasonable system lifetime, energy needs to be optimized at every stage of the system operation. Clustering scheme can effectively prolong the lifetime of wireless sensor networks by using limited energy resources of the deployed sensor nodes efficiently. In this section we present a survey of published distributed clustering algorithms in wireless sensor networks.

3.1. Energy Efficient Clustering Algorithms.

We review *energy efficient unequal clustering*, *energy-efficient hierarchical clustering* , *energy-efficient multi-level clustering*, *energy-efficient distance based clustering* and *optimal energy aware clustering* in WSNs in this section.

Heinzelman et. al. proposes [37, 39] a distributed clustering algorithm (LEACH) in which the sensor nodes elect themselves as clusterheads with some probability and broadcast their decisions. The remaining nodes join a cluster, of which the clusterhead is closest in terms of the communication energy cost. Then the role of clusterhead is periodically rotated among the nodes to balance energy consumption, since clusterheads have the extra burden of performing a long-range transmission to a distant sink node. Thus, LEACH counteracts the problem of non-uniform energy drainage by role rotation. To summarize, LEACH is a simple and an effective clustering algorithm.

Kim et. al. proposed a new distributed clustering algorithm for ubiquitous sensor network. In the case of single hop with clustering, they analyze the cause of difference in energy consumption between nodes. To balance energy consumption between nodes and increase the amount of data delivery , they choose the remaining energy of nodes, the distance between nodes within a cluster, and the number of nodes in a cluster as clusterhead selection criteria. Their algorithm's initial phase elects clusterhead candidates based on the remaining energy and reelects clusterhead having minimum distance cost within cluster. In the merge phase the algorithm merges clusters having members below the lowerbound. And the partition phase partitions clusters which have members over the upperbound [21].

In [65, 73], the authors propose a distributed clustering approach considering a hybrid of energy and communication cost. They present a protocol called HEED (Hybrid Energy-Efficient Distributed clustering). In this approach, the authors aim to prolong network lifetime by distributing energy consumption. They also aim to terminate the clustering process within a constant number of iterations. The algorithms minimize control overhead and produce well-distributed clusterheads and small-sized clusters. HEED assumes that all nodes are equally significant and energy consumption is not uniform among nodes. The algorithm does not make any assumptions about density of nodes or node capabilities.

In [74], the authors propose a new energy efficient clustering approach (EECS) for single-hop wireless sensor networks, which is more suitable for the periodical data gathering applications. Their approach elects clusterheads with more residual energy in an autonomous manner. This results in good clusterhead distribution an it also introduces a new distance-based method to balance the load among the clusterheads. In the clusterhead election phase, unlike LEACH, the clusterhead is elected by localized competition and its no iteration property makes it differ from HEED. Clusterheads are well distributed with the production of optimal value of competition range. In the cluster formation phase, nodes join clusters both considering its intra-cluster communication cost and its clusterheads' communication cost to the base station. EECS is autonomous and more energy efficient, and simulation results show that it prolongs the network lifetime much more than the other clustering protocols such as LEACH and HEED [74].

In [50], the authors propose an Energy-Efficient Unequal Clustering (EEUC) mechanism for periodical data gathering applications in wireless sensor networks. The mechanism organizes the network using unequal clustering and multihop routing. Like in the clustering phase of EECS [74], EEUC is a distributed competitive algorithm where the clusterhead is elected by a localized competition unlike LEACH. Since rotation of clusterheads and the metric of residual energy are not sufficient to balance the energy consumption across the network, they introduce an unequal clustering mechanism to balance the energy consumption among clusterheads. The node's competition range decreases as its distance to the base station decreases. The result is that clusters closer to the base station are expected to have smaller cluster sizes, thus they will consume lower energy during the intra-cluster data processing and they can preserve some more energy for the inter-cluster relay traffic. In the proposed multihop routing protocol for inter-cluster communication, a clusterhead chooses a relay node from its adjacent clusterheads according to the node's residual energy and its distance to the base station. Simulation results show that EEUC balances the energy consumption over the network, and achieves a remarkable network lifetime improvement.

Most of probabilistic approaches use the residual energy of each node as the criterion of clusterhead election. Those schemes have defects where unbalanced energy consumption among clusterheads occurrs. To overcome this defect, the authors of [75] consider a distance from the base station to clusterheads. For balanced energy consumption among clusterheads, they also consider the residual energy as the criterion of clusterhead election. Their scheme is fully distributed with the utilization of local information and good energy-efficiency by load balanced clustering scheme. Results of simulation experiments showed that in prolonging the network lifetime, the authors' proposed scheme is more effective than LEACH and EECS [74].

In order to maintain a certain degree of service quality and a reasonable system lifetime, energy needs to be optimized in sensor networks. In [30], the authors study the theoretical aspects of the clustering problem in sensor networks with application to energy optimization. They present an optimal algorithm for clustering the sensor nodes such that each cluster all having a master is balanced and the total distance between sensor nodes and master nodes is minimized. To distribute the load on all master nodes evenly, balanced clusters are needed. Minimizing the total distance helps in reducing the communication overhead and also the energy consumption. This problem (which they call balanced k-clustering) is modeled as a min-cost network flow problem which can be solved optimally using existing techniques [31].

Since limited battery capacity is an important disadvantage of sensor networks, the use of renewable energy sources such as solar power may prolong the lifetime of a sensor network [63]. Thus, letting nodes powered by solar energy to perform the most energy demanding tasks is important in cluster-based networks. Most energy-intensive tasks are performed by clusterheads. Although choosing solar-powered nodes as clusterheads is attractive, it is complicated because of the fact that the energy source is not permanent [63]. In the lab at FU Berlin, the authors of [63] have developed sensor boards equipped with solar cells. This motivated them to investigate if choosing solar-powered nodes as clusterheads is feasible and can provide energy savings and they extend LEACH [37, 39] to become solar aware. In a new version of LEACH [39], the clusterheads are chosen by the base station. The clusterheads selected by the base station remain as clusterheads for a certain time called round. They present two solar-aware extensions to LEACH. And they also compare the increased sensor network lifetime with the centralized LEACH protocol. Both solar-aware approaches provide significant benefits in many scenarios. Because solar-driven nodes have a higher probability to become clusterheads, they can perform the energy-intensive duties without consuming battery power. Keeping the clusterhead selected by the base station for an entire round is insufficient in some cases, so the authors present a mechanism that allows a clusterhead to choose another, solar-powered node, as a new clusterhead [63].They show that this mechanism provides additional benefits in these scenarios.

Qing et. al. propose a new distributed energy-efficient clustering scheme called DEEC for heterogeneous sensor networks. Using the ideas in LEACH, DEEC allows distribution of energy uniformly by rotating the clusterhead role among all nodes. In this algorithm, the probability of clusterhead election is based on the ratio between the energy of each node and the average energy of all nodes. The low-energy nodes will have less chance to become the clusterheads than the nodes with high initial and residual energy. Simulations show that DEEC prolongs network lifetime and achieves more effective messages than other classical clustering algorithms in two-level heterogeneous environments. Furthermore, DEEC performs well for the multilevel heterogeneous [57].

In [41], the authors propose an energy-efficient multi-level clustering algorithm called Multi-Level Clustering Algorithm (EEMC), which aims at minimum energy consumption in sensor networks. EEMC also covers the clusterhead election scheme. In EEMC, the data collection operation is broken up into rounds, where each round begins with a cluster set-up phase, which means that the nodes execute EEMC algorithm to form a multi-level clustering topology independently, and continues with a data transmission phase, which means the nodes transmit the sensed data packets to the sink node under such a clustering topology. In order to minimize algorithm overhead, the data transmission phase is compared to the cluster set-up phase [41]. The algorithm terminates in *O(log logN)* iterations given N nodes and achieves minimum latency, when the path loss exponent is 2. Assuming that sink node is remotely located and sensor nodes are stationary, simulation results show that their proposed algorithm is effective in prolonging the network lifetime of a large-scale network. They also show that the algorithm has low latency and moderate overhead across the network [41].

Liu et. al. consider the wireless sensor networks where all nodes in the network are homogenous and energy-constrained. Each node can change its transmission range to communicate directly with any node in the network. They show that a re-clustering scheme can be obtained to improve existing clustering methods. To increase the lifespan of the network, a modified redirection scheme based on the proposal in [51] is introduced in such a power-limited, cluster-based environment. More precisely, the former is designed to dynamically change the role of the coordinator in a cluster, and thus balance the power consumption under the whole network ground. To be able to conserve more power, member nodes in a cluster without backlogs are allowed to be *sleeping*. A mechanism that considers the minimum power consumption in a delivering path and a method that tries to maximize the residual power in nodes of the sensor network are combined. Simulation results show that, based on a practical energy model, the improved clustering method can achieve nearly three times the lifetime as compared with the conventional method [51].

Cluster based routing protocols attract researchers because of their advantages. To construct clusters, the cluster-based routing protocols need information on the locations of the sensor nodes in the network. However, finding out the locations of all sensor nodes in the network is very difficult because of the incurring costs. In [30] the authors propose a base station centralized simple clustering protocol (BCSP). Instead of location information of the sensor nodes, BCSP utilizes information on the remaining energy of each sensor node and the number of clusterheads. The base station selects clusterhead nodes by considering the remaining energy of sensor nodes, and broadcasts the list of the new clusterhead nodes to the sensor network in the cluster construction phase. Sensor nodes transmit their energy information together with their own sensing data in the data communication phase. From performance experiments, BCSP shows better performance than LEACH [30].

3.2. **Distributed Spatial Clustering Algorithms.** Spatial clustering aims to prolong network lifetime. Instead of collecting data from every node in the cluster, samples of a set of cluster representatives need to be taken, considering their spatio-temporal correlations. This results in reduction in data acquisition and transmission costs [22, 33] in a sensor network. Communication costs and the power consumption for communication are much more higher, if every node transmits its data to the central base station. For power efficiency, the authors propose in-network clustering. They regress time-series data at each node to build models. The contributions of Meka and Singh prove that clustering is both NP-complete and APX-hard. Their algorithm generates high quality clusterings in $O(\sqrt{N} logN)$ time and in *O(N)* message complexity where N denotes the network size. They use the spatial clusters to reduce both range queries and path queries. Results of their experiments show that ELink's clustering quality is superior to other distributed alternative techniques on both real world and synthetic data sets. ELink performs better than the centralized algorithm and other distributed alternative techniques in communication costs [52].

Research in sensor network localization is challenging. The goal of localization is to assign geographic coordinates to each node in the sensor network. In [14], the authors present an approach for localization that can satisfy working with inexpensive off-the-shelf hardware, scaling to large networks, and also achieving good accuracy in the presence of irregularities and obstacles in the deployment area. Recent developments in sensor network clustering algorithms have resulted in distributed algorithms that produce highly regular clusters. To inform their localization algorithm, they use this regularity. Their protocol requires only three randomly placed nodes that know their geographic coordinates, and does not require any ranging or positioning equipment (i.e., no signal strength measurement, ultrasound ranging, or directional antennas are needed) which is a significant advantage.

3.3. **Fault-Tolerant Clustering Algorithms.** In [45], the authors study distributed approximation algorithms for fault-tolerant clustering in wireless ad hoc and sensor networks. They study the problem in two network models. They present distributed approximation algorithms. For an arbitrary parameter t, distributed algorithm running time is $O(t^2)$ and it achieves an approximation ratio of $O(t\Delta^{2/t} \log\Delta)$ in general graphs, where n and Δ denote the network size and the maximal degree, respectively. Probabilistic algorithm runs in $O(\log \log n)$ time and achieves an $O(1)$ approximation when the network is modeled as a unit disk graph. Both algorithms require $O(\log n)$ bits size messages [45].

In [78], Gupta and Younis search the dependability of sensor networks considering the faulty gateways. They propose a run-time recovery mechanism based on healthy gateways to detect and handle faults in one faulty gateway. To limit the performance impacts caused by a gateway failure, a two-phased detection and recovery mechanism are presented. They use a simulation-based fault injection method that assumes occurrence of errors according to a predetermined distribution. The sensors assigned to the faulty gateway are reorganized without shutting down the system. The recovery information for recovery process is created during clustering. Various communication fault scenarios are considered and handled during recovery. Their approach enables fault tolerance in the system by performing periodic checks on the status of the gateways, provides considerable improvement in the stability of the system and reduces the overhead of re-clustering and system reconfigurations.

3.4. **Hierarchical Clustering Algorithms.** *Attribute based hierarchical clustering* is an important hierarchical clustering scheme in WSNs. Delivering data over a network in which the nodes may not have globally unique identifiers, satisfying energy saving requirements and being highly scalable and fault tolerant are the difficulties of data routing in large sensor networks [42]. In addition, if the sensor network becomes a resource shared by members of a large user community , then the routing scheme must also be energy efficient when handling requests that may: arrive at high rates, need different types of data in the response need response from subsets of the deployed sensors that satisfy certain attributes. In [42], the authors performed a simple cost effectiveness analysis of flooding based mechanisms for triggering data collection versus mechanisms which actively maintain a structure to deliver inquiries. They showed that sensor networks employing pure flooding systems are less efficient under heavy utilization and propose an attribute based hierarchical clustering mechanism that requires only one network wide flooding to establish all clusters across all hierarchy levels. The algorithm is also robust with respect to clusterhead failure, and by rotating the clusterhead functionality among cluster members, it implements load balancing [42].

In [6], the authors propose a distributed, randomized clustering algorithm to organize the sensors in a wireless sensor network into clusters. They extend this algorithm to generate a hierarchy of clusterheads and observe that the energy savings increase with the number of levels in the hierarchy.Each sensor in the network becomes a clusterhead with probability p and it becomes a clusterhead to the sensors within its radio range. Clusterhead information is forwarded to sensors that are less than k hops away from the clusterhead. Any sensor joins the cluster of the closest clusterhead if it receives clusterhead information and is not itself a clusterhead. If a sensor is not a clusterhead and has joined any cluster itself, it will become a clusterhead. If a sensor does not receive a clusterhead information within time duration t (where t units is the time required for data to reach the clusterhead from any sensor k hops away), it can deduce that it is not within k hops of any volunteer clusterhead and becomes a clusterhead. Because all the sensors within a cluster are at most k hops away from the clusterhead, after every t units of time, the clusterhead can transmit the collected information to the processing center. This limitation on the hops number allows the clusterheads to schedule their transmissions [6].

Because of the limited bandwidth and energy resources, separating network into connected clusters is a challenging task in self-organizing sensor networks. In [43], the authors make contributions towards improving the efficiency of self-organization in wireless sensor networks. They use their new approach for message-efficient clustering by introducing two algorithms. In authors' proposed algorithms, nodes allocate local *growth budgets* to neighbors. Authors present a new randomized methodology for timer design of cluster initiators which guarantees that initiators will not interfere with each other. They derive a logarithmic upper bound on the expected time for network decomposition. They present a variant that allows more concurrency among initiators and reduces the network decomposition time. However, this method produces slightly more clusters than the first. Simulations results showed that their algorithms produce clusters of bounded size and low diameter, using fewer messages than Ring approach and shows that their proposed methodology scales to large networks [43].

3.5. **Connectivity Based k-Hop Clustering Algorithms.** In [53], the authors present new clustering algorithms for nodes in a mobile ad hoc network. The combination of connectivity and lower ID criterions into a single clustering algorithm which can be used for selecting clusterheads is proposed by the authors. Their aim is to minimize the number of clusters, which results in dominating sets of smaller sizes. They describe algorithms for modifying the cluster structure considering topological changes. They generalize the cluster definition so that a cluster contains all nodes that are at distance of at most k hops from the clusterhead. They measure the average number of created clusters, the number of border nodes, and the cluster size in random unit graphs to test the efficiency of four clustering algorithms (k-lowestID, k-CONID, $k = 1$ and $k = 2$). Results show that the border nodes and ratio of the sum of the clusterheads are stable. They present a unified framework for the new clustering algorithm where the only difference in the same algorithm is a properly defined weight at each node. The authors also propose a framework for generating random unit graphs with obstacles.

3.6. **Probabilistic Clustering Algorithms.** In [38], Wu and Huang describe a constant time clustering algorithm that can be applied on sensor networks. They extend the Younis and Fahmy's probabilistic method in HEED [73]. The simulation results show that in relatively few rounds, especially in sparse networks, the extension can generate a small number of clusterheads. Authors [73] propose an extended version of HEED. Their extension includes three phases. These are performing the core algorithm in [75] to eliminate some nodes, applying the original HEED algorithm and instead of electing all uncovered nodes as clusterheads, they apply the core algorithm again.

3.7. **Density Based Clustering Algorithms.** In [58], the authors propose a hierarchical architecture of sensor network with cluster formation and clusterhead selection algorithm. They use various parameter metrics related to sensor node density. In their proposed network model, they indicate the deployment density variation of sensor nodes by the edge or link lengths standard deviations. Wireless link is used to connect each node to its neighbor nodes and the inter node distance can be found by the average link length. Proposed algorithm identifies intra-cluster link and inter-cluster link. Inter-cluster link is used in the identification of the network discontinuity. They define the clusters assuming inter-cluster links are larger than the intra-cluster links.

4. Conclusions

We surveyed graph theoretic algorithm for clustering in MANETs and WSNs leaving out other clustering approaches which are fundamentally not graph theoretic.

We showed that dominating set based clustering is a fundamental approach whereas spanning tree based clustering is also an important approach in MANETs. There are numerous algorithms on this topic, namely, graph theoretic algorithms for clustering in MANETs and we have outlined only the fundamental and the recent ones with the aim of providing a starting point for any further studies in this topic. For WSNs, clustering algorithms again play an important role to provide efficient communication infrastructure in such networks. There is no single favorable algorithm for clustering in MANETs or WSNs. Each algorithm has certain advantages and drawbacks when compared with the others. Our experience [12] has shown that clustering using dominating sets in a MANET was favorable in a medium size and medium speed network in terms of cluster quality, time and message complexities. Spanning tree based approach we have used [20] again provided favorable results for a medium size MANET with better run time performance than the dominating set based algorithm. We are currently working on a clustering algorithm based on spanning trees for WSNs.

References

[1] Alzoubi, K. M., Wan, P. J., Frieder, O. Maximal Independent Set, Weakly Connected Dominating Set, and Induced Spanners for Mobile Ad Hoc Networks, International Journal of Foundations of Computer Science, vol. 14(2), (2003), pp. 287-303.

[2] Awerbuch ,B. , Optimal Distributed Algorithms for Minimum Weight Spanning Tree, Counting, Leader Election and related problems, Proceedings of the 9th Annual ACM Symposium on Theory of Computing, (1987), pp. 230-240.

[3] Ahuja, M., Zhu Y., A Distributed Algorithm for Minimum Weight Spanning Trees Based on Echo Algorithms, Proceedings of the 9th International Conference on Distributed Computing Systems, (1989).

[4] Baker, D., Ephremides, A., The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm, Communications, IEEE Transactions, vol. 29-11, (1981), pp. 1694-1701.

[5] Banerjee, S., Khuller, S. , A clustering scheme for hierarchical routing in wireless networks, Tech. Report CS-TR-4103, University of Maryland, College Park, (2000).

[6] Bandyopadhyay, S., Coyle, E.J., An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks, INFOCOM 2003, vol. 3, (2003), pp. 1713-1723.

[7] Chen, Y., P., Liestman, A., L., Liu, J. ,Clustering Algorithms for Ad Hoc Wireless Networks, Nova Science Publisher, (2004).

[8] Chen, Y., P., Liestman, A. L., Approximating Minimum Size Weakly-Connected Dominating Sets for Clustering Mobile Ad Hoc Networks, 3rd ACM International Symposium on Mobile Ad Hoc Networks and Computing, (2002), pp. 165-72.

[9] Chen, G., Nocetti, F.G., Gonzalez, J.S., Stojmenovic, I., Connectivity based k-hop clustering in wireless networks, System Sciences, HICSS, Proceedings of the 35th Annual Hawaii International Conference, (2002), 2450-2459.

[10] Choi, Y., Kim, B., Jung, K., Cho, K., Kim, S.H., An Overlay Multicast Mechanism Using Single-hop Clustering and Tree Division for Mobile Ad hoc Networks , VTC 06, vol. 2, (2006), pp. 921-926.

[11] Chen, Y. P., Liestman, A.L., A Zonal Algorithm for Clustering Ad Hoc Networks, International Journal of Foundations of Computer Science, vol. 14(2), (2003), pp. 305-322.

[12] Cokuslu, D., Erciyes, K., Dagdeviren, O., A Dominating Set Based Clustering Algorithm for Mobile Ad hoc Networks, vol. 3991, (2006), pp. 571-578.deniz cokuslu references son Orhan Dagdeviren references

[13] Choi, Y., Lee, E., Lee, D., Park, S., Yu, F., Kim, S.-H. , An Implicit Cluster-based Overlay Multicast Protocol for Mobile Ad hoc Networks , ICOIN 07, (2007), pp. 3507-3512.

[14] Chan, H., Luk, M., Perrig, A. , Using Clustering Information for Sensor Network Localization, DCOSS 2005, (2005).

[15] Canourgues, L., Lephay, J., Soyer, L., Beylot, A.L., STAMP: Shared-Tree Ad Hoc Multicast Protocol , MILCOM 06, (2006), pp. 1-7.

[16] Das, B., Sivakumar, R., Bhargavan, V., Routing in Ad Hoc Networks Using a Spine, Proc. Sixth IEEE International Conference of Computers Communication and Networks, (1997), pp. 1-20.

[17] Das, B., Bharghavan, V., Routing in ad-hoc networks using minimum connected dominating sets, Communications, ICC97, (1997), pp. 376-380.

[18] Dai, F., Wu, J., An Extended Localized Algorithm for Connected Dominating Set Formation in Ad Hoc Wireless Networks, IEEE Transactions On Parallel and Distributed Systems, (2004).

[19] Dagdeviren, O., Erciyes, K., Cokuslu, D. , Merging Clustering Algorithms in Mobile Ad hoc Networks, , ICDCIT 2005, Springer Verlag LNCS, vol. 3816, pp. 56-61.

[20] Dagdeviren, O., Erciyes, K., Cokuslu, D., A Merging Clustering Algorithm for Mobile Ad hoc Networks , ICCSA 2006, Springer Verlag vol. 3981, (2006), pp. 681-690.

[21] Daehwan, K., SangHak, L., We Duke, C., Energy-Efficient Clustering Algorithm in Wireless Sensor Networks, EUC 2006, LNCS 4096, (2006), pp. 1078-1088.

[22] Deshpande, A., Guestrin, C., Hong, W., Madden, S., Exploiting correlated attributes in acquisitonal query processing, ICDE, (2005).

[23] Fernandess, Y., Malkhi, D., K-clustering in wireless ad hoc networks, ACM Workshop On Principles Of Mobile Computing archive Proceedings of the second ACM international workshop on Principles of mobile computing, (2002), pp. 31-37.

[24] Gerla, M, Tsai, J. T., Multicluster, mobile, multimedia radio network, Wireless Networks, vol. 1(3), (1995), pp. 255-265.

[25] Guha, S., Khuller, S., Approximation Algorithms for Connected Dominating Sets, Springer-Verlag New York, (1998), LLC, ISSN: 0178-4617.

[26] Gao, B., Yang, Y., Ma, D., A new distributed approximation algorithm for constructing minimum connected dominating set in wireless ad hoc networks, International Journal of Communication Systems, vol. 18, (2005), pp. 743-762.

[27] Grimaldi, R. P. , Discrete and Combinatorial Mathematics, An Applied Introduction, Addison Wesley Longman, Inc., (1999).

[28] Gallagher, R. G., Humblet, P. A., Spira, P. M. A, Distributed Algorithm for Minimum-Weight Spanning Trees, ACM Transactions on Programming Languages and Systems (1983), pp. 66-77.

[29] Gentile, C., Haerri, J., Vandyck, R. E., Kinetic minimum-power routing and clustering in mobile ad-hoc networks, Proc. Vehicular Tech. Conf. , (2002).

[30] Giljae, L., Minsun, L., Woojin, S., Junguk, K., Okhwan, B., A., Base Station Centralized Simple Clustering Protocol for Sensor Networks, EUC 2006, LNCS 4096, (2006), pp. 682-691.

[31] Ghiasi, S., Srivastava, A., Yang, X., Sarrafzadeh, M. , Optimal Energy Aware Clustering in Sensor Networks, MDPI, (2002).

[32] Garay, J.A., Kutten S., Peleg D., Zhu Y. , A sub-linear time distributed algorithm for minimum-weight spanning trees, Proceedings 34th Annual Symposium on Foundations of Computer Science, (1993), pp. 659-668.

[33] Guestrin, C., Bodik, P., Thibaux, R., Paskin, M., Madden, S. , Distributed regression: An efficient framework for modeling sensor network data, IPSN, (2004).

[34] Han, B., Jia, W., Efficient Construction of Weakly-Connected Dominating Set for Clustering Wireless Ad Hoc Networks, IEEE Globecom, (2006), pp. 1-5.

[35] Haynes, T., W., Hedetniemi, S., T., Slater, P., J., Domination in graphs, Advanced Topics, Marcel Dekker, Inc., (1998).

[36] Han, B., Jia, W., Clustering Wireless Ad Hoc Networks with Weakly-Connected Dominating Set, Journal of Parallel and Distributed Computing, vol. 67(6), (2007), pp. 727-737.

[37] Heinzelman, W., Chandrakasan, A., Balakrishnan, H., Energy-Efficient Communication Protocol for Wireless Microsensor Networks, HICSS 2000, vol. 2, (2000).

[38] Huang, H., Wu, J., A Probabilistic Clustering Algorithm in Wireless Sensor Networks, VTC 2005, vol. 3, (2005), pp. 1796-1798.

[39] Heinzelman, W., Chandrakasan, A., Balakrishnan, H. , An Application-Specific Protocol Architecture for Wireless Microsensor Networks, IEEE Transactions on Wireless Communications, vol.1(4) , (2002), pp. 660 - 670.

[40] Johansson, T., Carr-Motyckova, L., On Clustering in Ad Hoc Networks, Proc. Vehicular Tech. Conf. Fall , Swedish National Computer Networking Workshop 2003, (2003).

[41] Jin. Y., Wang, L., Kim, Y., Yang, X., EEMC: an energy-efficient multi-level clustering algorithm for large-scale wireless sensor Networks, Computer Networks, (2007).

[42] Ke, W., Basu, P., Abu Ayyash, S., Little, T.D.C. , Attribute Based Hierarchical Clustering in Wireless Sensor Networks, MCL Technical Report, (2003).

[43] Krishnan, R., Starobinski, D., Efficient clustering algorithms for self-organizing wireless sensor Networks, Elsevier, (2004).
[44] Kleinrock, L., Faroukh, K., Hierarchical Routing for Large Networks, Computer Networks, (1997).
[45] Kuhn, F., Moscibroda, T., Wattenhofer, R., Fault-Tolerant Clustering in Ad Hoc and Sensor Networks, ICDCS 2006, (2006), pp. 68-78.
[46] Lien, Y.N., A New Node-Join-Tree Distributed Algorithm for Minimum Weight Spanning Trees, Proceedings of the 8th International Conference on Distributed Computing System, (1988), pp. 334-340.
[47] Li, Y., Kim, D., Zou, F., Du, D-Z., Constructing Connected Dominating Sets with Bounded Diameters in Wireless Networks, International Conference on Wireless Algorithms, Systems and Applications, (2007), pp. 89-94.
[48] Liu, H., Pan, Y., Jiannong, C., An Improved Distributed Algorithm for Connected Dominating Sets in Wireless Ad Hoc Networks, Proc. of Parallel and Distributed Processing and Applications: Second International Symposium, ISPA 2004, (2004), pp. 340-348.
[49] Li, Y., Peng, S., Chu, W. , K-MCore for Multicasting on Mobile Ad Hoc Networks , PDCAT 06, (2006), pp. 109-114.
[50] Li, C., Ye, M., Chen, G., Wu, J. , An Energy-Efficient Unequal Clustering Mechanism for Wireless Sensor Networks, vol. 16(1), (2005), pp. 3-17.
[51] Liu, J.S., Lin, C.R., Energy-efficiency clustering protocol in wireless sensor networks, Elsevier Computer Communications, vol. 29(12) (2003), pp. 2230-2237.
[52] Meka, A., Singh, A. , Distributed Spatial Clustering in Sensor Networks, EDBT 06, LNCS 3896, (2006), pp. 980-1000.
[53] Nocetti, F.G., Gonzalez, J.S., Connectivity Based k-Hop Clustering in Wireless Networks Telecommunication Systems, vol. 22(1-4), (2003), pp. 205-220.
[54] Nanuvala, N., An Enhanced Algorithm to Find Dominating Set Nodes in Ad Hoc Wireless Networks, Master of Science Thesis in the College of Arts and Science, Georgia State University, (2006).
[55] Ohta, T., Kawaguchi, T., Kakuda, Y., An Autonomous Clustering-Based Hierarchical Multicast Routing for Mobile Ad Hoc Networks, IEICE Transactions on Communications, E88-B(12), pp. 4451-4461, 2005.
[56]]Ohta, T., Inoue, S., Kakuda, Y., An Adaptive Multihop Clustering Scheme for Highly Mobile Ad Hoc Networks, 6th ISADS'03, (2003).
[57] Qing, L., Zhu, Q., Wang, M., Design of a distributedenergy-efficient clustering algorithm for heterogeneous wireless sensor Networks, Elsevier, vol. 29(12), (2006), pp. 2230-2237.
[58] Rahman, O., Choi, B.G., Monowar, M., Hong, C.S., A Density Based Clustering for Node Management in Wireless Sensor Network, Springer Verlag LNCS, vol. 4773, (2007), pp. 527-530. Deniz Ozsoyeller refler son
[59] Royer, E.M., Toh, C.K., A review of current routing protocols for ad-hoc mobile wireless networks, IEEE Magazine Personal Commun., (1999), pp. 46-55.
[60] Stojmenovic, I., Seddigh, M., Zunic, J. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks, IEEE Transactions on Parallel and Distributed Systems, (2002), pp. 14-25.
[61] Srivastava, G., Distributed Algorithms for Finding and Maintaining a k-tree core in a Dynamic Network.
[62] Singh, G., Vellanki, K., A distributed protocol for constructing multicast trees , Proceedings of the International Conference on Principles of Distribured Systems, (1998). Orhan Dagdeviren references son Deniz Ozsoyeller refler.
[63] Voigt, T., Dunkels, A., Alonso, J., Solar-aware Clustering in Wireless Sensor Networks, ISCC 2004, vol. 1, (2004), pp. 238-243.
[64] Xu, Z., Dai, S., Garcia-Luna-Aceves, J. J., Hierarchical Routing Using Link Vectors, Proceedings of Infocom, (1998).
[65] Younis, O., Fahmy, S., Distributed Clustering for Scalable, Long-Lived Sensor Networks, MOBICOM 03, (2003).
[66]]West, D., Introduction to Graph Theory, Second edition, Prentice Hall, Upper Saddle River, N.J., (2001).
[67] Wu, J., Li, H., On Calculating Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Networks, Proc. Third International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, (1999), pp. 7-14.
[68] Wu, J., Li, H., A Dominating-Set-Based Routing Scheme in Ad Hoc Wireless Networks, Springer Science+Business Media B.V., Formerly Kluwer Academic Publishers B.V. ISSN: 1018-4864, (2001).
[69] Wu, J., Fei, D., An extended localized algorithm for connected dominating set formation in ad hoc wireless networks, IEEE Transactions, Parallel and Distributed Systems, (2004).
[70] Wan, P.J., Alzoubi, K. M., Frieder, O., Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks, Springer Science and Business Media B.V., Formerly Kluwer Academic Publishers, (2002), pp. 141-149.
[71] Wu, J., Extended Dominating-Set-Based Routing in Ad Hoc Wireless Networks with Unidirectional Links, IEEE Trans. Parallel and Distributed Systems, (2002), pp. 189-200.

[72] Yan, X., Sun, Y., Wang, Y., A Heuristic Algorithm for Minimum Connected Dominating Set with Maximal Weight in Ad Hoc Networks, Grid and Cooperative Computing, (2003), pp. 719-722.

[73] Younis, O., Fahmy, S. , HEED A Hybrid, Energy-Efficient Approach, IEEE Transactions on Mobile Computing, INFOCOM 2004, (2004), pp. 366- 379.

[74] Ye, M., Li, C., Chen, G., Wu, J. , An Energy Efficient Clustering Scheme in Wireless Sensor Networks, Ad Hoc and Sensor Wireless Networks, Vol. 3, (2006), pp. 99-119.

[75] Young-Ju, H., Seon-Ho, Park., Jung-Ho, E., Tai-Myoung, C., Gervasi, O., and Gavrilova, M., Energy-Efficient Distance Based Clustering Routing Scheme for Wireless Sensor Networks, Springer-Verlag, ICCSA 2007, LNCS 4706, (2007), pp. 195-206.

[76] Yu, F., Choi, Y., Kim, B., Jung, K., Cho, H., Kim, S.H., Stable On-demand Overlay Multicast in Mobile Ad Hoc Networks , APCC 06, (2006), pp. 1-5.

[77] Ya-feng, W., Yin-long, X.,Guo-liang, C., Kun, W., On the construction of virtual multicast backbone for wireless ad hoc networks , MASS 04, (2004), pp. 25-27.

[78] Younis, M., Gupta, G. , Fault-Tolerant Clustering of Wireless Sensor Networks, WCNC 2003, vol. 3, (2003), pp. 1579-1584.

Kayhan Erciyes - received a BSc. degree in Electrical Eng. and Electronics from the University of Manchester, MSc. degree in Electronic Control Eng. from the University of Salford and a Ph.D. degree in Computer Engineering from Ege (Aegean) University. He was a visiting scholar at Edinburgh University Computer Science Dept. during his Ph.D. studies. Dr. Erciyes worked as visiting and tenure track faculty at Oregon State University, University of California Davis and California State University San Marcos, all in the U.S.A. He also worked in the research and development departments of Alcatel Turkey, Alcatel Portugal and Alcatel SEL of Germany. His research interests are broadly in parallel and distributed systems and computer networks. More precisely, he works on distributed algorithms for synchronization in mobile ad hoc networks, wireless sensor networks and the Grid. Dr. Erciyes is a faculty member at Ege University International Computer Institute, Izmir, Turkey.

Orhan Dagdeviren - received the MSc degree in 2006 and BSc degree in 2004 from the Izmir Institute of Technology. He is a PhD candidate in Computer Engineering at International Computing Institute at Izmir, working under the supervision of Professor Kayhan Erciyes. His interests lie broadly in the computer networking and distributed systems areas. His recent focus has been on graph theoric middleware protocol design for wireless sensor and mobile ad hoc networks. Also he is interested in group communication for grid computing.

Deniz Cokuslu - was born in 1981. He got his BS degree from Izmir Institute of Technology Department of Computer Engineering in 2004, and his MS degree in 2007. He had worked for Yasar Holding, Astron Project Development Office between 2003 and 2004 as a software developer. He has contributed to the Project Sardes, INRIA Rhone-Alpes in 2006. His research interests are distributed computing, mobile and sensor networks. He is currently a PhD student in Ege University, International Computer Institute.

Deniz Ozsoyeller - received a BSc degree in Computer Eng. from Izmir University of Economics. Currently, she is a MSc. candidate in Computer Engineering at International Computing Institute at Izmir, working under the supervision of Professor Kayhan Erciyes. Her interests lie in the computer networking and distributed systems areas. Her recent focus has been on spanning tree formation in wireless sensor networks.