22

Applied Graph Theory to Improve Topology Control in Wireless Sensor Networks

Paulo Sérgio Sausen, Airam Sausen and Mauricio de Campos
Master's Program in Mathematical Modeling, Regional University of Northwestern Rio Grande do Sul State (UNIJUI)
Brazil

1. Introduction

This chapter presents solutions for computing bounded-distance multi-coverage backbones in wireless networks. The solutions are based on the (k,r)-CDS problem from graph theory for computing backbones in which any regular node is covered by at least k backbone members within distance r, offering a variable degree of redundancy and reliability.

Advances in lower power-consumption processors, sensor devices, embedded systems and wireless communication have made possible the development and deployment of Wireless Sensor Networks (WSN).Such networks have been used, for instance, in safety and military applications for the purpose of monitoring and tracking geographic boundaries. In industrial applications, WSN may be used for automating manufacturing processes, for monitoring building structures, and in environmental systems for monitoring forests, oceans and precision farming Akyildiz, Su, Sankarasubramaniam & Cayirci (2002).

A WSN is usually composed by a large number of small nodes, called *sensor nodes*, each one having a processing unit, a radio transceiver and an antenna for wireless communication, one or more sensor units (e.g., temperature, movement), and a power unit usually equipped with a low capacity battery. Due to its limited power resources, and because batteries cannot be easily replaced, nodes are built out from power saving components.

To save energy,both approaches apply the partial or total turn-off of some node units. The rule of thumb is to keep active only those units (or components) necessary for performing the sensor network tasks. However, it is not an easy task to decide which nodes should sleep and which should be active at any given time, because these decisions strongly depend on the application running on top of the network. It is also undesirable to keep nodes inactive for too long, because it can impact the network *Quality-of-Service* (QoS).

Topology control through the construction of backbones can offer better support for broadcasting and routing of data packets. The total or partial turn off of nodes not comprising the Backbone constitutes the main motivation for employing such structures in WSNs. Reduced power consumption, and longer network lifetime, are potential benefits when applying Backbones to a network of sensors, because only a subset of nodes need to be active at any time to support basic network services.

In contention based medium access protocols using a single channel, the number of collisions of packets increases as we increase the number of competing nodes. By reducing the total number of active nodes, backbones can potentially reduce the end-to-end delay among sensors and sinks, and possibly extend the network lifetime due to the reduced energy consumption. In addition to that, it might be possible to provide better Quality of Service (QoS) for WSNs.

The basic criteria of QoS in WSN (i.e., *area coverage*) can still be guaranteed by not totally turning off the sensor nodes. Given that the radio is the most expensive element in terms of energy consumption Margi et al. (2006), by just turning it off and leaving the other components active we can still manage to save on energy.

In Ad Hoc networks Alzoubi et al. (2002); Bao & Garcia-Luna-Aceves (2003a), and more recently in WSNs Dai & Wu (2005); Paruchuri et al. (2005), the computation of Backbones has been considered through the computation of some variant of Connected Dominating Set (CDS). A set of nodes in the network is considered a Dominating Set (DS) if all nodes in the network belong to this set or, otherwise, are adjacent (i.e., neighbors) to at least one DS node. A DS is said to form a CDS when the graph induced by the DS is connected (i.e., the DS induces a Backbone for the network).

The problem of computing a CDS of minimum cardinality for any arbitrary network topology, a Minimum CDS (MCDS), constitutes an NP-complete problem Garey & Johnson (1978a), and requires knowledge of the entire network topology. Distributed solutions for computing approximations to the MCDS problem have been proposed in the literature Chen & Liestman (2002); Das & Bharghavan (1997). Even though such solutions aim at reducing the total number of dominating nodes, it has been shown that in wireless networks there is a tradeoff between redundancy and reliability Basu & Redi (2004). Too much redundancy is not desirable, but too little may compromise the network connectivity. This is of special concern to WSNs, because nodes are more vulnerable to failures due to the environments in which they operate.

Dai e Wu Dai & Wu (2005) proposed a distributed solution for computing a k-connected k-dominating Backbone for wireless networks. Their approach combines multiple domination and the k-vertex connected property, which guarantees that a CDS remains vertex connected even when removing up to $k-1$ nodes from the backbone. The shortcomings of this approach have to do with the required high degree of redundancy in the network topology.

In this chapter we propose the first centralized and distributed solutions for computing *bounded-distance multi-coverage backbones* in WSNs. This means that any sensor node is covered by multiple backbone members within a bounded-distance. To guarantee these properties, the (k,r)-CDS mechanism is employed for computing the Backbone. The multiple domination parameter, k, defines the minimum number of backbone nodes covering any regular sensor node. The bounded-distance parameter, r, defines the maximum distance to k backbone nodes for any other sensor in the network. The centralized solution provides an approximation to the optimum solution, and it is used as a lower bound when evaluating the performance of the distributed solution. The distributed solution is source-based in the sense that usually the base-station (or sink) is the focus of attention in a WSN.

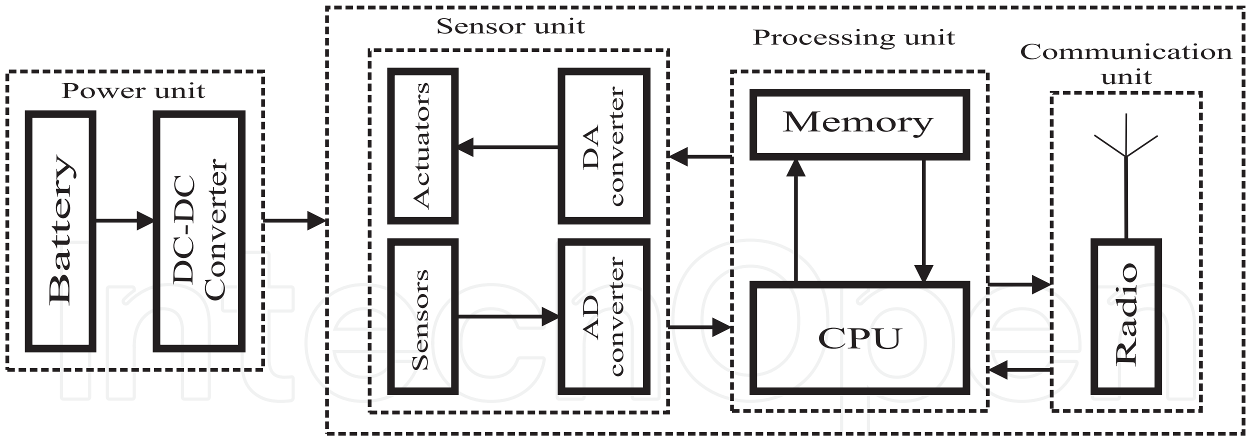


Fig. 1. Sensor node: basic structure

2. Wireless Sensor Networks

A Wireless Sensor Network may well contain hundreds or thousands of small autonomous elements called sensor nodes, and each sensor can feature a large variety of sensors (e.g., temperature, speed, acoustic, seismic). In many cases, nodes are randomly spread over remote areas, making it difficult to perform any maintenance to the nodes. Hence, a node remains live while it has enough battery capacity for its normal operation, and the network lifetime strongly depends on the remaining capacity of the nodes in the network.A sensor node has a few basic components (see Figure 1) Akyildiz, Su, Sankarasubramaniam & Cayirci (2002); Raghunathan et al. (2002) as follows:

- *Power Unit*, usually a battery, which acts as the power source for all node's components;
- *Sensor Unit* that contains a group of sensors and actuators;
- *Processing Unit* which includes a microprocessor or a micro-controller;
- *Communication Unit* which consists of a short range radio for wireless communication.

3. Domination in graph theory

An undirected graph $G = (V, E)$ consists of a set of vertices $V = \{n_1, \ldots n_k\}$, and a set of edges E (an edge is a set $\{n_i, n_j\}$, where $n_i, n_j \in V$ and $n_i \neq n_j$). A set $D \subseteq V$ of vertices in a graph G is called a *dominating set* (DS) if every vertex $n_i \in V$ is either an element of D or is adjacent to an element of D Haynes et al. (1998). If the graph induced by the nodes in D is connected, we have a *connected dominating set* (CDS). The problem of computing the minimum cardinality DS or CDS of any arbitrary graph is known to be NP-complete Garey & Johnson (1978b).

A variety of conditions may be imposed on the dominating set D in a graph $G = (V, E)$. Among them, there are *multiple domination*, and *distance domination* Haynes et al. (1998). *Multiple domination* requires that each vertex in $V - D$ be dominated by at least k vertices in D for a fixed positive integer k. The minimum cardinality of the dominating set D is called the *k-domination number* and is denoted by $\gamma_k(G)$. *Distance domination* requires that each vertex in $V - D$ be within distance r of at least one vertex in D for a fixed positive integer r. In this case, the minimum cardinality of the dominating set D is called the *distance-r domination number*, and is denoted by $\gamma_{\leq r}(G)$.

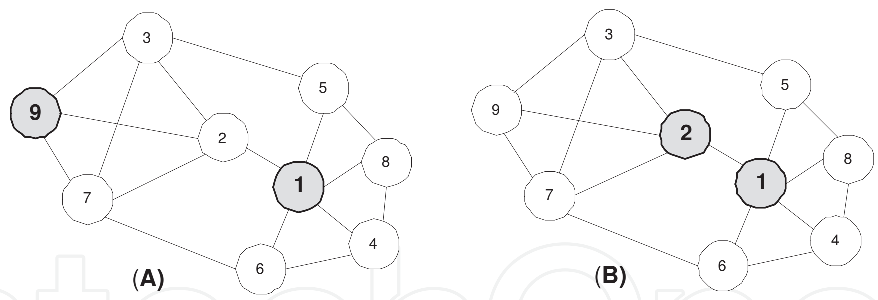


Fig. 2. *Dominating set* examples (gray nodes are *dominating*): (A) Dominating Set (DS); (B) Connected Dominating Set (CDS).

Henning et al. Henning et al. (1996) have presented some bounds on the *distance-r domination number* $\gamma_{\leq r}(G)$. They show that, for an integer $r \geq 1$, if graph G is a connected graph of order $n \geq r+1$, then $\gamma_{\leq r}(G) \leq \frac{n}{r+1}$. An algorithm that computes a *distance-r dominating set* within the established bounds is also presented.

The (k,r)-DS problem is defined Joshi et al. (1993) as the problem of selecting a minimum cardinality vertex set D of a graph $G = (V,E)$, such that every vertex u not in D is at a distance smaller than or equal to r from at least k vertices in D. The problem of computing a (k,r)-DS of minimum cardinality for arbitrary graphs is also NP-complete Joshi et al. (1993). Figure 2 (A) show example the Dominating Set and Figure 2 (B) show the Connect Dominating Set example.

Joshi et al. Joshi et al. (1993) have provided centralized solutions for solving the (k,r)-DS problem in *interval graphs* (IG). A graph G is said to be an interval graph if there is a one-to-one correspondence between a finite set of closed intervals of the real line and the vertex set V, and two vertices u and v are said to be connected if and only if their corresponding intervals have a nonempty intersection. Even though the solutions presented by Joshi et al. Joshi et al. (1993) are optimal, IGs are limited to very simple network topologies.

4. ***Clustering*** and topology control

In wireless sensor networks (WSN), nodes must coordinate among themselves because they cannot assume any fixed infrastructure (e.g., access points). Broadcasting of signaling messages is the underlying mechanism for coordination, and the broadcast can target a portion of the network (e.g., gathering neighborhood information), or the entire network (e.g., discovering routes on demand).

Coordination in WSN includes operations such as neighborhood discovery, organization of nodes (i.e., topology control and clustering), and routing. Examples of organization of nodes include the location of services, computing an efficient backbone for the broadcasting of signals, and routing of data packets.

Organization of nodes can be proactive or on demand. While operations to build such structures require broadcasting of signaling messages, these structures make broadcast operations scale to much larger portions of the network. That is, the hierarchical structure functions as a backbone, on top of which broadcasting can be performed more efficiently.

A virtual hierarchy is also possible. In this case, nodes are organized not depending on their physical location, but based on some other criteria. For example, the virtual topology could be comprised only by nodes speaking a given protocol. In this case, a virtual link exists between any pair of nodes whenever they are talking to each other using the specified protocol.

In some cases, the predefined hierarchical address of each node reflects its position within the hierarchy Tsuchiya (1988). In heterogeneous networks (i.e., networks composed of nodes with different capabilities in terms of energy, bandwidth, processing power, and transmission power), not only the physical location but also the resources of each node can be used as a criteria for deciding the role of each node within its sub-structure.

There are two broad categories of hierarchical architectures in ad hoc networks: *clustering* Chen et al. (2004), and *topology control based on hierarchies* Bao & Garcia-Luna-Aceves (2003b); Li (2003). These architectures can be employed to extend the network lifetime Bandyopadhyay & Coyle (2003); Chen et al. (2001); Younis & Fahmy (2004), achieve load balancing Bao & Garcia-Luna-Aceves (2003b), and to augment network scalability Heinzelman (2000); Li (2003).

With *clustering* Chen et al. (2004), the substructures that are condensed in higher levels are called *clusters*. For each cluster there is at least one node representing the cluster, and this node is usually called a *cluster-head*. Cluster-heads act as leaders in their clusters, providing some service to their members. As an example, a cluster-head could be an access point to the outside network, or it could be a *sink* for collecting information from a group of sensors (cluster members) in a wireless sensor network Akyildiz, Weilian, Sankarasubramaniam & Cayirci (2002).

Topology control and clustering are closely related problems. While the former defines a physically connected *backbone* of the network (i.e., the backbone nodes are connected, and they cover all nodes in the network), the latter constructs a *virtual backbone* (i.e., the set of cluster-heads do not necessarily compose a connected component of the network, even though they cover all nodes in the network).

Hierarchical structures are used for broadcasting, transmission of control messages, and routing data packets. The selection of nodes for the backbone will have a direct impact in the overall performance of the protocol. In most cases, dominating nodes are selected using some distributed dominating set algorithm, and if a backbone is desired, auxiliary nodes are selected to connect the dominating nodes.

5. Centralized (k, r)-CDS mechanism

In this section we shows the centralized soluction for computes a (k,r)-CDS for any connected networks. The network can be abstracted as a graph $G = (V, E)$ where the set V includes all nodes (i.e., vertices), and the set E includes all links (i.e., edges). The centralized solution for the (k, r)-CDS problem requires that the entire network topology be known. The centralized solution applies a *greedy* heuristic usually employed to optimization problems when there is the need for defining a group of candidates by optimizing the value of a given metric. The centralized mechanism guarantees that all nodes in the network, with the exception of those that comprise the Backbone, are covered by k members of the Backbone within distance r. It is assumed that all nodes have an unique identifier (i.e. ***id***) and they know the whole network topology.

While carrying out the algorithm, colors are ascribed to nodes to reflect their states during the selection process. To begin with, all nodes are colored **White**. A node is colored **Gray** when it becomes a candidate to the Backbone. Once a node joins the Backbone, the node is colored **Black**.The heuristic chosen for selecting nodes for the (k, r)-CDS is described as follows:

1. Initially, all nodes in the network are colored white.
2. The node with the largest r-hop neighborhood (i.e. $u \in V$ with maximum value for $|N_r^u|$) is colored black. Ties are broken lexicographically by considering the highest ***id***.
3. Next, all nodes adjacent (i.e. neighboring nodes) to the black node are colored gray.
4. The gray node with most white neighbors is then selected and colored black. Ties are broken by choosing the highest ***id***. This procedure continues until the conditions in item 6 are satisfied.
5. In the absence of any white nodes, the node with the most gray neighbors is considered as the next member of the Backbone.
6. The process comes to an end when all regular nodes have been covered by at least k members of the Backbone (black nodes) within distance r; i.e., when all nodes in the network, except those that are members of the Backbone, have met the (k,r)-CDS specification.

5.1 Example

Figure 3 shows the computation of a (2,2)-CDS for a particular network topology. Black nodes are members of the Backbone. For this configuration, nodes that do not belong to the Backbone are covered by at least two nodes of the Backbone at most two hops distant. Initially, all nodes are colored white, and have their degrees calculated (Figure 3(A)). For example, the node with id 3 displays the marking $(7, -, 0)$, which indicates that it has seven neighbors within two hops (i.e., $r = 2$), and it is not covered by any black nodes yet (indicated by the third parameter set to zero).

The node with the largest 2-hop neighborhood (i.e., $u \in V$ with maximum value of $|N_2^u|$) is colored Black (Figure 3(B)). At this point, there is a tie among nodes $0, 2$ and 5. In this case, the node with the largest ID (i.e., node 5) wins. Following that, the neighboring nodes to node 5 are colored gray, while its r-hop neighborhood is accounted for this new Backbone node (i.e., all nodes within distance 2 from node 5 are covered by this node).

Node 0 is the Gray node covering the most White neighbors (nodes 4 and 6), hence it is chosen for the Backbone and colored Black. Therefore, all its neighbors are colored Gray (Figure 3(C)). The remaining gray nodes hold only one White neighbor. Once node 0 has been selected, all its r-hop neighbors (i.e., nodes $2, 3, 4, 6, 7, 8, 9$) have their parameters updated to reflect coverage by node 0 (Black nodes do not need to be covered, for they are part of the Backbone).

The process goes on by selecting the Gray node that has the most White neighbors (Figure 3(D)), and in this case node 9 is selected for untying purpose. After this last selection, the Backbone is complete, and all network nodes are covered by at least two Black nodes within distance 2. The Backbone is then formed by nodes 0, 5, and 9.

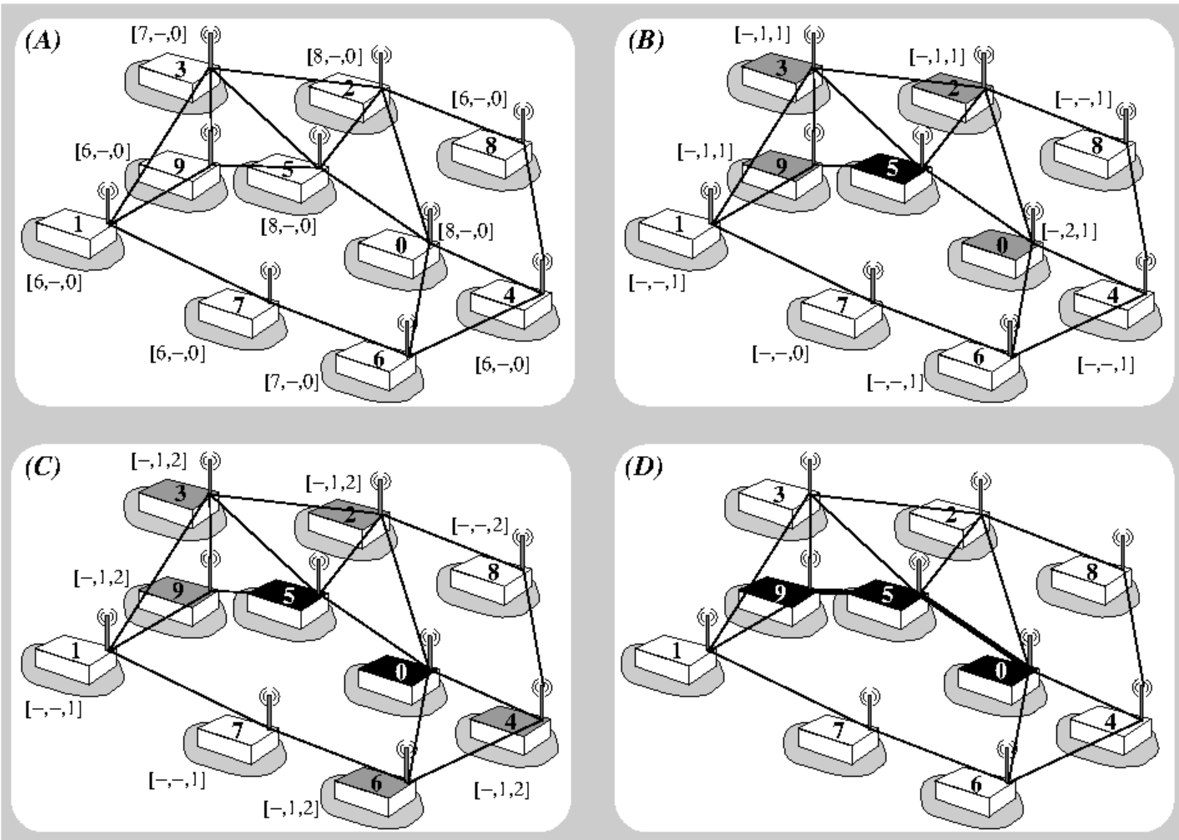


Fig. 3. Exemple (2,2)-CDS applying the centralized solution

5.2 Formal analysis

A WSN is modeled as a graph $G = (V, E)$, where V is the set of sensor nodes, and E is the set of wireless links. Each node in V is associated to its Euclidean coordinates. A link between any pair of nodes u and v exists (i.e., $(u, v) \in E$) if and only if the Euclidean distance between the pair of nodes is smaller than or equal to their transmission range (assumed to be the same for all nodes in V).

A set $V' \subseteq V$ forms a CDS for network G if all nodes in $V - V'$ are adjacent to least one node in V', and the sub-graph $G[V']$ induced by V' is connected.

Lemma 1. *Connectivity is preserved by augmenting a connected component with any node adjacent to the connected component.*

Proof. This can be proved by induction on the cardinality of the connected component. The base case is when the connected component is composed by just one node. Now consider a connected component with cardinality $n - 1$. A node adjacent to any node in the connected component augments its cardinality to n, and keeps the component connected because of the adjacency property. $\square$

Theorem 1. *The centralized mechanism correctly computes a (k, r)-CDS Backbone for any arbitrary connected graph $G = (V, N)$.*

Proof. To account for node's coverage, k sets are used, $D_0, D_1, D_2, ..., D_k$, where D_i represents the nodes covered by at least i nodes from the Backbone within distance of r hops. Initially, all nodes of G are in D_0. The inclusion of a new member to the Backbone can promote one or more nodes from D_i to D_{i+1} for all $i < k$. The nodes elected for the Backbone are immediately inserted into D_k. The first Backbone node, b_0, is the node with the largest r-hop neighborhood in the graph. Whenever a tie occurs, the node with the largest identifier (i.e., **id**) is selected. In each iteration, a new backbone node, u, is selected among the neighboring nodes to the Backbone (i.e., N_B) such that the selected node has the largest number of uncovered neighbors (i.e., $\forall\ n \in N_B$ we have that u maximizes $\sum_{i=0}^{k-1} |D_i \cap N_1^n|$). The connectivity property is guaranteed by Lemma 1. The process ends when all nodes are covered, or all nodes belong to the Backbone (i.e., $D_k = V$). Assuming networks with a finite number of nodes, n (i.e., $n = |V|$), the number of iterations is at most n; therefore, the computation ends within a finite period of time. □

6. Distributed (k, r)-CDS mechanism

Unlike the centralized solution, the distributed (k,r)-CDS solution does not require the information about the whole network topology. All nodes are required to know only their r-hop neighborhood. The (k,r)-CDS extends the (k,r)-DS mechanism Spohn & Garcia-Luna-Aceves (2006) proposed for ad hoc networks. The (k,r)-DS mechanism is used for the construction of bounded distance multi-clusterhead clusters (i.e., each regular cluster member is covered by several clusterheads within a bounded distance), but it does not connect the clusterheads among themselves (i.e., that is why it is a DS).

As for the centralized solution, the distributed mechanism computes a (k,r)-CDS of the network. However, the computation of the Backbone is accomplished distributively. In addition to that, to adhere to the characteristics of sensor networks, the construction of the Backbone stems from a particular node which could be the Base Station (BS) in the network.

We assume that any node in the network could be the BS. Moreover, there could be more than one BS. However, in this work we consider a single BS per network, which is randomly chosen among all nodes in the network. It is also assumed that each node has an unique identifier, and they have knowledge about their r-hop neighborhood. The mechanism is carried out in two phases as described as follows.

In phase 1 the BS initiates the process by sending an Information Message (IM) to their neighbors reporting the Distance to the Base Station, dBS, initially set to zero by the BS itself. On receiving this message, a node updates its distance to the BS and announces its dBS to its neighbors through a new IM message. If a node gets more than one notification message, any message announcing a shorter distance to the BS triggers a new IM message so that any neighbor can eventually learn about a shorter path to the BS. Considering that all transmissions are reliable, after a finite period of time, and possibly after many retransmissions, all nodes in the network learn their shortest distance to the BS.

The second objective of this phase consists of obtaining the r-hop neighborhood topology, which requires r rounds of messages exchange. In each round, the nodes broadcast information about their known neighbors, and their respective distances. After r rounds, all nodes come to know their neighbors within distance r. By piggybacking the information about each node's distance to the BS (i.e., their dBS), at the end of this phase nodes also learn the distance from each r-hop neighbor to the BS.

Phase 2 starts immediately after the completion of phase 1. It is during this phase that nodes elect k nodes from their r-hop neighborhood to become part of the Backbone. After that, the node announces its elected nodes throughout its r-hop neighborhood. The election is based on the information gathered during phase 1.

The BS node actually starts the election process by electing the k nodes closest to itself (i.e. those with the smaller dBS in the neighborhood). Any ties are broken choosing the node with the largest degree, and persisting the tie the node with the largest ID wins. The BS creates and transmits to all its neighbors a message called Election Message (EM) carrying a list called Backbone Members (BM) with the k elected nodes.

Upon receiving an EM message, a node performs - just once - the selection of its Backbone Members. The criteria are similar to those used by the BS with the restriction that only nodes belonging to the Backbone, or those adjacent to it (i.e. neighbors to Backbone members), can be elected. This restriction guarantees the Backbone connectivity property(i.e. the creation of a CDS), and it reduces the total number of Backbone members because nodes already in the Backbone are likely to cover multiple nodes in the neighborhood.

After a node elects its Backbone members, the BM list in the EM message is updated to reflect any changes to the list, and the message is then transmitted to its neighbors. In addition to that, a Notification Message (NM) is sent to all (if any) new elected nodes (i.e., nodes that were not listed in the original BM list). The election process continues until all nodes in the network have elected their Backbone members. At any stage, a node that has already carried out its election and receives another EM message will just discard it. [1]

6.1 Example

Figure 4 shows the computation of a (2,2)-CDS for a particular WSN. As previously, Black nodes represent Backbone members. Initially, node 0 is selected as the BS, and it is colored Black (Figure 4(A)). Given that $r = 2$, all nodes are familiar with their neighborhood up to 2 hops. The first field in the marking process (shown close to each node in the Figure) identifies the dBS value. The second field indicates whether a node belongs to the Backbone or it neighbors a Backbone member (value 1), otherwise it is set to value 0. The third field indicates the **Degree** of a node.

Using the information obtained during phase 1, the BS selects its two members for the Backbone (Figure 4(B)). Notice that the BS selects itself (because it is a Backbone member by default) and another node from its r-hop neighborhood. The list of elected nodes is transmitted to the BS neighbors through an EM message.

The first elected node embodies the BS itself since $dBS = 0$ and because it starts the whole process. Node 5 is the second elected node for it exhibits the largest id while node 2 ties in Degree and distance (i.e., dBS) with node 5. The election of nodes 0 and 5 reflects on the marking of nodes $2, 3, 4, 6$ and 9, demonstrating that they are either members or neighbors of

[1] To handle periodical elections, one could use an election identifier in the EM message to allow nodes to know when they should re-elect their Backbone members.

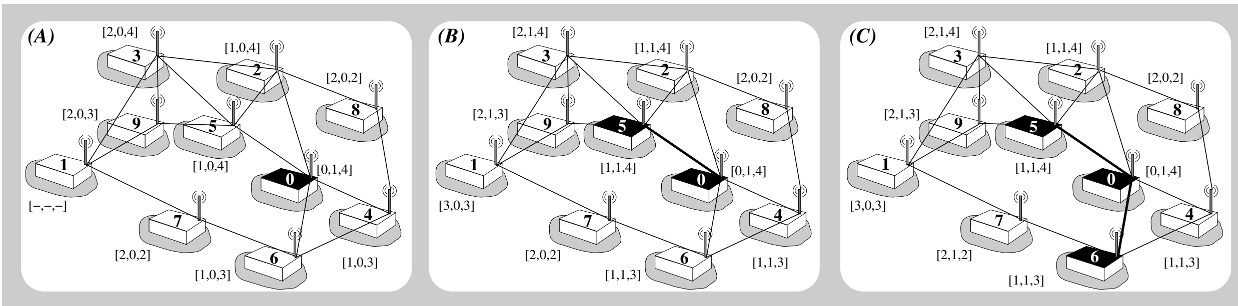


Fig. 4. Exemple (2,2)-CDS applying the distributed solution

a Backbone member. On the sequence, all BS neighbors perform their elections. All nodes with $dBS = 1$ (i.e. the nodes $2, 4, 5$ and 6) elect nodes 0 and 5 as members of the Backbone, triggering a new EM message to be sent to their neighbors.

For nodes two hop distant from the BS (i.e., $dBS = 2$), we have that nodes $3, 8$ and 9 elect nodes 0 and 5, and node 7 elects nodes 0 and 6 (Figure 4(C)). The marking of node 7 is updated demonstrating that this node is a neighbor of a Backbone member. Node 1 which has $dBS = 3$ receives the EM message and elects nodes 5 and 6 as members of the Backbone. This ends the election since all neighbors of node 1 (which is the node farther away from the BS) have already completed their election. The Backbone is then composed by nodes 0, 5 and 6.

6.2 Formal analysis

To prove the correctness of the distributed (k,r)-CDS mechanism, we have to show that it is *safe* (i.e., the algorithm computes a (k,r)-CDS of the network), and that it is *live* (i.e., it completes within a finite period of time).

Lemma 2. *Phase one of the distributed solution has message complexity of $O(n \cdot r)$, where n is the number of nodes in the network and r is the distance parameter.*

Proof. During each round, nodes send messages to all their one-hop neighbors. Phase one takes r rounds. Assuming a network of n nodes, phase one requires $n \times r$ messages to complete. Therefore, the message complexity of phase one is of order $O(n \cdot r)$. □

Lemma 3. *After r rounds of successful transmission of message m, the message is propagated up to r hops away from the originating node.*

Proof. This can be proved by induction on the distance d from the node starting the process, n_0. The base case is when $d = 0$, meaning that the starting node itself knows the message it created to be propagated throughout the network. Now consider a node u at distance $r - 1$ from n_0. Once node u retransmits message m to all its one-hop neighbors, the message eventually reach any neighbor r hops from n_0. □

Theorem 2. *The distributed solution correctly computes a (k, r)-CDS Backbone for any connected graph $G = (V, E)$ during phase two.*

Proof. We assume that any node $i \in V$ knows its r-hop neighbors (i.e., N_r^i), as well as their distances to the base station i_0 (Lemma 3). The base station, i_0, is selected as the first backbone node. On its turn, node i_0 selects for the backbone the $k-1$ nodes closest to itself (i.e., first it tries to select among its neighbors within distance one, then, if necessary, within distance two, and so on until $k-1$ nodes are selected). The selected nodes repeat the process, taking as candidates to backbone members those nodes from their r-hop neighborhood which are closer to the base station, i_0, and also requiring that any candidate must be adjacent to or already a member of the backbone. The latter requirement, guarantees the connectivity of the Backbone (Lemma 1). After node i_0 has selected its $k-1$ backbone members, the base station announces the list of elected nodes to its neighbors, which in turn repeat the process by performing their own election. The election is carried out just once by all nodes in the network (duplicate announcements are just discarded). Considering that messages take a finite period of time to propagate throughout the network, the whole election takes a finite period of time to complete. Given that any node must elect k members among its r-hop neighborhood, and that any elected node must be adjacent to nodes already in the Backbone or members themselves, the connectivity and coverage properties are guaranteed. □

7. Performance analysis

The centralized and distributed (k,r)-CDS mechanisms are compared through extensive simulations using a customized simulator. Network topologies are created based on the ***unit disk graph*** model Clark et al. (1990). According to this model, any pair of nodes, A and B, are said to be connected if the Euclidean Distance between their coordinates is smaller or equal to R, the transmission range of any node. We assume $R = 15m$ for all nodes (i.e., a homogeneous networks). Nodes are randomly placed over the terrain, and only connected topologies are taken into account. For each configuration, we gather results for 30 trials.

Nodes	Diameter	Degree
200	11,7±0.7	11,8±0.6
400	10,9±0.4	24,2±0.7
600	10,6±0.5	36,6±0.8
800	10,3±0.5	49,0±0.9
1000	10,1±0.3	61,4±0.9

Table 1. Parameters for Scenarios 1

For the simulations we assume an ideal MAC protocol with no collisions, ensuring that all transmissions are successful. The same assumption is made in related work Spohn & Garcia-Luna-Aceves (2006); Wu & Dai (2003). All topologies evaluated in the simulations are assumed to be static (i.e., no mobility). Because there is no known optimum solution for the (k,r)-CDS problem, the proposed centralized solution is used as a lower bound when evaluating the performance of the distributed solution. The simulations encompass a series of experiments changing the distance parameters (i.e., r) and the coverage parameters (i.e., k). All experiments are repeated for 30 trials corresponding to different network topologies.

To gather representative statistical samples, two scenarios are considered. Table 1 presents the average standard values for the **Diameter** (i.e., the largest shortest distance between any pair

Nodes	Diameter	Degree
200	11,9±0.7	12,2±0.6
425	18,0±0.5	11,7±0.5
750	23,2±0.6	12,0±0.0
1185	29,4±0.7	12,0±0.0
1720	35,4±1.0	12,3±0.5

Table 2. Parameters for Scenarios 2

of nodes) and the **Degree** (i.e., average number of one-hop neighbors) over 100 samples for first scenario.

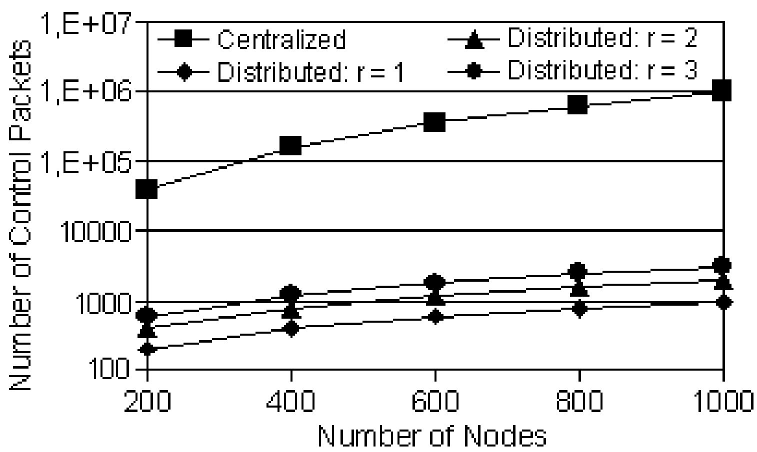


Fig. 5. Control message overhead

In the first scenario, the network diameter stays almost unchanged as the average degree increases with denser networks. As for the second scenario (see Table 2), the network density is kept the same by increasing the terrain size for larger networks. As a consequence, the average degree remains almost the same but the network diameter increases (i.e., the networks are sparser compared to the ones from the first scenario).

To compare the two proposed heuristics, two performance metrics are evaluated: ***control message overhead***, the total number of control messages exchanged for gathering topology information (centralized mechanism), or for the execution of the two phases in the distributed mechanism; ***total number of Backbone nodes***, the total number of nodes that each mechanism selects to form de Backbone.

Figure 5 shows the results for the message overhead regarding the first scenario. Results for the second scenario are omitted because they are similar to those from the first scenario. Considering a network composed of n nodes, the centralized mechanism incurs $O(n^2)$ messages exchange due to the topology dissemination process. In the distributed mechanism, the control overhead depends on the distance parameter r, because the nodes need to obtain information regarding the r-hop neighborhood. Given that all nodes participate in this process, it incurs a $O(n \cdot r)$ message complexity.

As expected, the centralized mechanism presents the best results for the first scenario (Figure 6). As the distance parameter r increases, the performance of the distributed

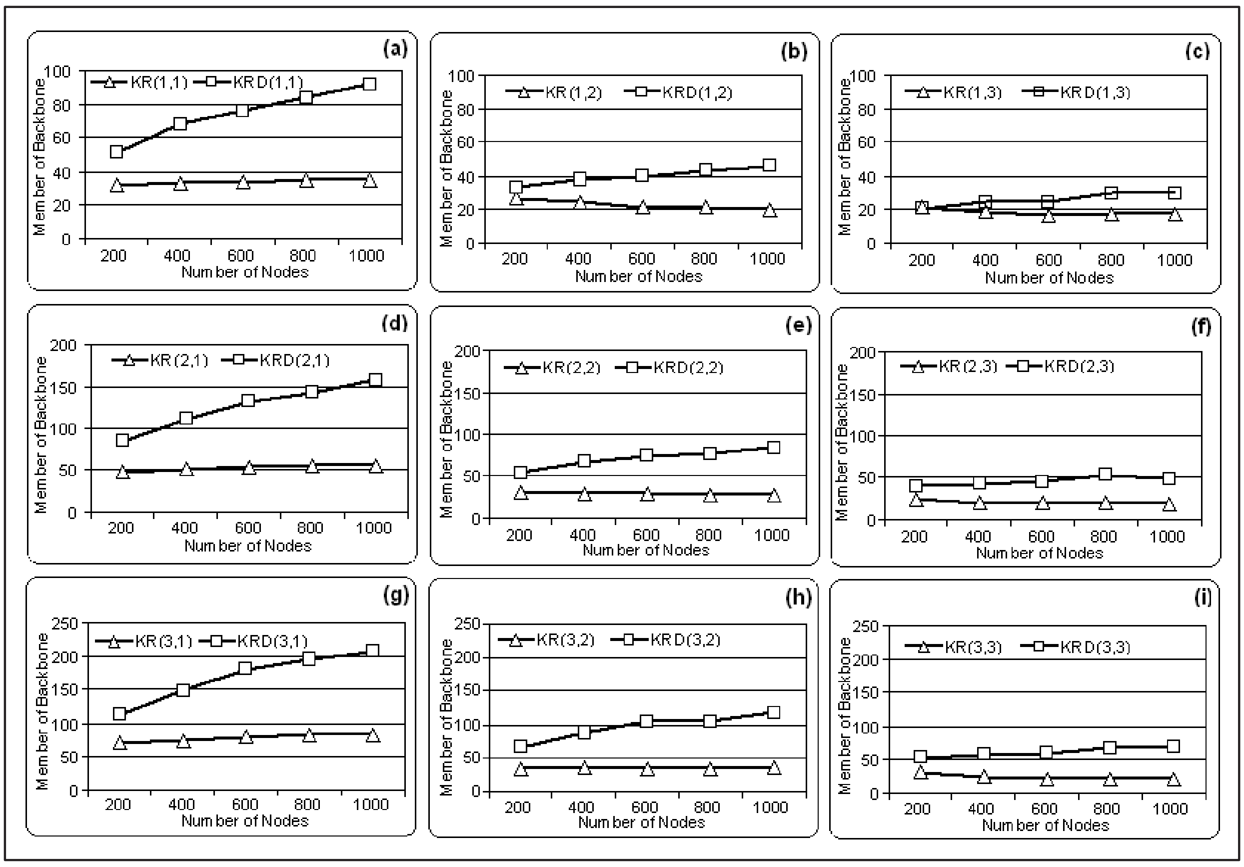


Fig. 6. (k,r)-CDS results for scenario 1

mechanism gets closer to the performance of the centralized mechanism (Figures 6(c), (f), and (i)). The best results are achieved for networks with 200 nodes (Figure 6(c)), when the centralized mechanism elects 20.8 nodes in average, whereas the distributed mechanism elects 21.4 nodes in average. The difference between the two approaches becomes more prominent for larger networks (e.g., networks with 1000 sensor nodes). As the network gets denser, the distributed mechanism elects more nodes. Because nodes elect k nodes among their r-hop neighborhood it is likely that more candidates satisfy the election requirement (i.e., nodes already in the backbone or adjacent to any backbone node), incurring on more redundancy when compared to sparser networks. For the future work we intend to apply some pruning mechanism to reduce redundant backbone members.

For the second scenario, we have chosen to keep node density steady while varying the network diameter. The simulation results (Figure 7) show an increase in the total number of Backbone nodes for both mechanisms. This is due to a better distribution of sensor nodes (i.e., sparser networks). As expected, the centralized mechanism presents better results for most experiments. However, the two heuristics compare to each other in the $(1,2)$-CDS and $(1,3)$-CDS configurations (Figure 7 (b) and (c)). As noticed in the first scenario, as the distance parameter r increases, the two approaches present similar results. On the other hand, the difference between the performance for both solutions reduces compared to the first scenario.

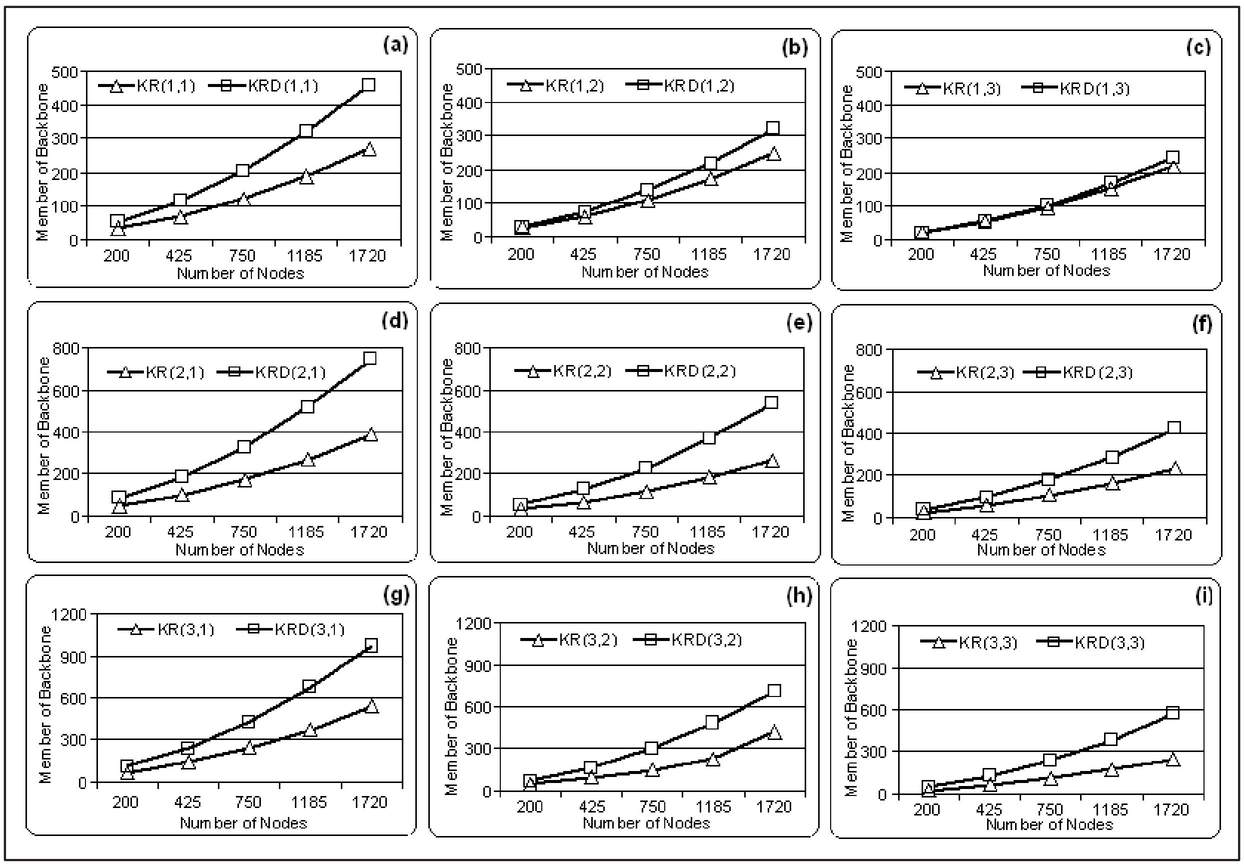


Fig. 7. (k,r)-CDS results for scenario 2

8. Conclusion

We presented the first centralized and distributed solutions for computing *Bounded-Distance Multi-Coverage Backbones* in Wireless Sensor Networks (WSNs). Backbones have the (k,r)-CDS properties, which include the minimum number, k, of nodes in the backbone covering any regular node, as well as a bounded-distance, r, to the covering nodes.

Given the characteristics of WSNs, redundancy is desirable. Therefore, any mechanism requiring redundancy (e.g., data gathering, routing) could take advantage of topologies with the (k,r)-CDS properties. By limiting the distance to the backbone, it could well translate to bounded-delays when accessing the backbone. In any situation, the solutions presented in this paper provide new basis for designing protocols for WSNs.

We compared the centralized and distributed mechanisms through extensive simulations using a customized simulator. Given that the centralized solution is unsuitable for WSNs, because of the incurred control overhead, it is used as a lower bound when evaluating the performance of the distributed solution. It is shown that even though the distributed solution builds larger backbones, it does not incur on much control overhead.

As future work, we are planning applying a pruning mechanism to the distributed solution during the election process. In face of that, one can expect Backbones of smaller cardinality compared to the original solution. We also plan to include other coverage metrics (e.g., area coverage) within the selection process, and load balancing for members of the backbone (i.e., nodes should alternate roles as backbone members in order to prolong the network lifetime).

9. References

Akyildiz, I. F., Weilian, S., Sankarasubramaniam, Y. & Cayirci, E. (2002). A survey on sensor networks, *IEEE Communications Magazine* 40(8): 102–114.

Akyildiz, I., Su, W., Sankarasubramaniam, Y. & Cayirci, E. (2002). A survey on sensor networks, *IEEE Communications Magazine* 40(8): 102–114.

Alzoubi, K. M., Wan, P.-J. & Frieder, O. (2002). Distributed heuristics for connected dominating sets in wireless ad hoc networks, *Journal of Communications and Networks* 4(1): 22–29.

Bandyopadhyay, S. & Coyle, E. (2003). An energy efficient hierarchical clustering algorithm for wireless sensor networks, *IEEE INFOCOM*, pp. 1713–1723.

Bao, L. & Garcia-Luna-Aceves, J. J. (2003a). Topology management in ad hoc networks, *MobiHoc*, ACM Press, New York, NY, USA, pp. 129–140.

Bao, L. & Garcia-Luna-Aceves, J. J. (2003b). Topology management in ad hoc networks, *MobiHoc '03: Proceedings of the 4th ACM international symposium on mobile ad hoc networking & computing*, ACM Press, pp. 129–140.

Basu, P. & Redi, J. (2004). Movement Control Algorithms for Realization of fault Tolerant Ad Hoc Robot Networks, *In IEEE Network* 18: 36–44.

Chen, B., Jamieson, K., Balakrishnan, H. & Morris, R. (2001). Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks, *MobiCom '01: Proceedings of the 7th annual international conference on mobile computing and networking*, ACM Press, pp. 85–96.

Chen, Y. P. & Liestman, A. L. (2002). Approximating minimum size weakly-connected dominating sets for clustering mobile ad hoc networks, *MobiHoc*, ACM Press, New York, NY, USA, pp. 165–172.

Chen, Y. P., Liestman, A. L. & Liu, J. (2004). *Ad Hoc and Sensor Networks*, Nova Science Publisher, chapter Clustering algorithms for ad hoc wireless networks.

Clark, B., Colbourn, C. & Johnson, D. (1990). Unit disk graphs, *Discrete Math* 86: 165–177.

Dai, F. & Wu, J. (2005). On constructing k-connected k-dominating set in wireless networks., *IPDPS*, IEEE Computer Society, Denver, CA, USA.

Das, B. & Bharghavan, V. (1997). Routing in Ad-Hoc Networks Using Minimum Connected Dominating Sets, *ICC* pp. 376–380.

Garey, M. & Johnson, D. (1978a). *Computers and Intractability: A Guide to NP-Completeness*, Freeman, San Francisco, California.

Garey, M. R. & Johnson, D. S. (1978b). *Computers and Intractability*, Freeman, San Francisco.

Haynes, T. W., Hedetniemi, S. T. & Slater, P. J. (eds) (1998). *Fundamentals of Domination in Graphs*, Marcel Dekker, Inc.

Heinzelman, W. (2000). *Application-Specific Protocol Architectures for Wireless Networks*, PhD thesis, Massachusetts Institute of Technology.

Henning, M. A., Oellermann, O. R. & Swart, H. C. (1996). The diversity of domination, *Discrete Mathematics* 161(1-3): 161–173.

Joshi, D., Radhakrishnan, S. & Narayanan, C. (1993). A fast algorithm for generalized network location problems, *Proceedings of the ACM/SIGAPP symposium on applied computing*, pp. 701–8.

Li, X.-Y. (2003). *Ad Hoc Networking*, IEEE Press, chapter Topology Control in Wireless Ad Hoc Networks.

Margi, C. B., Petkov, V., Obraczka, K. & Manduchi, R. (2006). Characterizing energy consumption in a visual sensor network testbed, *in Proceedings of the 2nd IEEE TridentCom* .

Paruchuri, V., Durresi, A., Durresi, M. & Barolli, L. (2005). Routing through backbone structures in sensor networks., *ICPADS*, Vol. 2, Fuduoka, Japani, pp. 397–401.

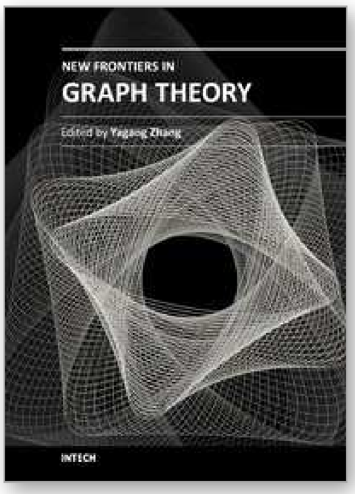
Raghunathan, V., Schurgers, C., Park, S. & Srivastava, M. B. (2002). Energy-aware wireless microsensor networks, *IEEE Signal Processing Magazine*, 19(2): 40–50.

Spohn, M. A. & Garcia-Luna-Aceves, J. J. (2006). Bounded-distance multi-clusterhead formation in wireless ad hoc networks, *Ad Hoc Networks (Elsevier)* . http://dx.doi.org/10.1016/j.adhoc.2006.01.005.

Tsuchiya, P. F. (1988). The landmark hierarchy: a new hierarchy for routing in very large networks, *SIGCOMM '88: Symposium proceedings on communications architectures and protocols*, ACM Press, pp. 35–42.

Wu, J. & Dai, F. (2003). Broadcasting in ad hoc networks based on self-pruning, *IEEE INFOCOM* pp. 2240–2250.

Younis, O. & Fahmy, S. (2004). Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks, *Transactions on Mobile Computing* 3(4): 366–379.



New Frontiers in Graph Theory

Edited by Dr. Yagang Zhang

ISBN 978-953-51-0115-4
Hard cover, 526 pages
Publisher InTech
Published online 02, March, 2012
Published in print edition March, 2012