Energy Efficient Connectivity Maintenance in Wireless Sensor Networks

Yanxiang He and Yuanyuan Zeng

School of Computer, Wuhan University, 430072, Hubei, P.R. China
yxhe@whu.edu.cn, zyywhu@gmail.com

Abstract. Connectivity maintenance in energy stringent wireless sensor networks is a very important problem. Constructing a connected dominating set (CDS) has been widely used as a connectivity topology strategy to reduce the network communication overhead. In the paper, a novel energy efficient distributed backbone construction algorithm based on connected dominating set is presented to make the network connected and further prolong the network lifetime, balance energy consumption. The algorithm is with $O(n)$ time complexity and $O(n)$ message complexity. The results show that our algorithm outperforms several existing algorithms in terms of network lifetime and backbone performance.

1 Introduction

The research on wireless sensor networks has been fueled up by many applications in various areas [1, 2]. An important problem of sensor networks is the stringent power budget of wireless sensor nodes. Research has shown that a great amount of energy of sensor nodes is consumed for communications. Reducing the communication cost is an important way to save energy of sensor nodes and to prolong the life time of sensor. Minimizing energy consumption and maximizing the system lifetime has been a major design goal for make connectivity maintenance in wireless sensor networks.

A dominating set (DS) of a graph is a subset of nodes such that each node in the graph is either in the subset or adjacent to at least one node in that subset. A CDS is a DS which induces a connected sub graph. A connected dominating set (CDS) is a good candidate of a virtual backbone for connectivity maintenance in wireless networks [3], because any node in the network is less than 1-hop away from a CDS node. One objective for constructing the backbone is to minimize the size of a backbone. We assume that every node has the same transmission range so that we can model the network topology using unit disk graphs, UDG in short. Unfortunately computing a minimal CDS (denoted by MCDS) of a UDG graph has been proved to be NP-hard [4].

In this paper, we propose a novel efficient distributed algorithm that computes a sub-optimal MCDS in polynomial time to maintain the network connectivity. The backbone nodes have heavy communication load and their energy consumption is high. To balance the energy consumption load, each time when we compute the MCDS, we also compute out the length of time that this MCDS should work as a

D.-S. Huang, K. Li, and G.W. Irwin (Eds.): ICIC 2006, LNCIS 344, pp. 95 – 105, 2006.

backbone. The length of time is dependent on the residual energy of the MCDS nodes. When this length of time expires, the sensor nodes coordinate with each other again and compute the next MCDS as the backbone for the next period of time. By doing so, the energy consumption of all nodes in the network is balanced and the lifetime of the network is extended.

The remainder of this paper is organized as follows. Section 2 briefly introduces the related work in the literature. Section 3 discusses out distributed algorithm for constructing the CDS as backbone to make connectivity. Section 4 is the performance analysis and simulations. Section 5 is the conclusion and future work.

2 Related Work

In the last few years, researchers actively explored advanced power conservation topology control approaches for wireless sensor networks. Extensive work has been done on the connectivity maintenance issue. Research in [4] focused on energy conservation by controlling sensor transmission power in order to maintain network connectivity. It demonstrated that the network connectivity can be maintained if each sensor has at least one neighbor in every cone of $2\pi/3$. Xu et al. [5] proposed two algorithms that can conserve energy by identifying redundant nodes of connectivity. In GAF [6], nodes use geographic location information to divide the world into fixed square grids. Nodes within a grid switch between sleeping and listening, with the guarantee that one node in each grid stays up to route packets. SPAN [7] is another protocol that achieves energy efficiency for wireless sensor networks by introducing off-duty and on-duty cycles for sensor nodes. Dominating set based topology control leads to a virtual backbone for the deployed ad hoc and sensor networks. The virtual backbone is formed by representing the connected routing nodes as a connected dominating set (CDS). Since the minimal CDS problem is NP-hard, most previous work has focused on finding distributed heuristics for reducing the size of CDS.

Current MCDS approximation algorithms include centralized and distributed algorithms. Following the increased interest in wireless ad hoc and sensor networks, many distributed approaches have been proposed because of no requirements for global network topology knowledge. These algorithms contain two types. One type is to find a CDS first, then prune some redundant nodes to attain MCDS. Wu and Li proposed in [11] a distributed algorithm with $\Theta(m)$ message complexity and $O(\Delta^3)$ time complexity, the approximation factor at most n/2. Butenko et al [12] constructs a CDS starting with a feasible solution, and recursively removes nodes from the solution until a MCDS is found. The other type is to form a maximal independent set (MIS) at first, and then find some connectors to make the independent nodes connected together. P. J .Wan et. in [9] proposes a distributed algorithm with performance ratio of 8. Min et al in [11] propose an improved algorithm by employing a Steiner tree in the second step to connect the nodes in the MIS with performance ratio of 6.8.

3 Distributed Algorithm for Energy Efficient MCDS

When all sensor nodes have the same transmission range, the network topology is modeled as a UDG G= (V, E). An edge represents that nodes *u* and *v* are within each other's transmission range. Each node u is associated with a unique ID, denoted by *id(u)* (this can be, for instance, IP or MAC address).

The aim of our algorithm is to compute a sub-optimal MCDS as a backbone for wireless sensor networks. Our algorithm consists of two phases. In the first phase, we compute a maximal independent set (MIS) of the network graph. An independent set of a graph is a subset of V that no two nodes in the subset have an edge. An MIS of a graph is an independent set that cannot include any more node in V. An MIS is a DS of a graph. Note that this DS (obtained as the MIS) may not be connected. The second phase of the algorithm is to choose the minimal number of nodes (called connectors) to make the DS connected, i.e., a CDS. Each time when constructing a CDS, the length of operating time of this CDS is determined according to the residual energy of the CDS nodes. When this operating time expires, the next CDS is computed. To extend the lifetime of the network, we always give higher priority to the nodes with higher residual energy to be as backbone nodes. Thus, nodes will be usually acting as backbone nodes in turn and the energy consumption of nodes is well balanced. For each node *u*, we define weight as: *w*(*u*)={*energy(u)*, *degree(u)*}.

The higher significant part of *w*(*u*) is the residual energy of *u*. When two nodes have the same energy, the node with a higher degree has a higher priority. This policy would make the size of the CDS smaller (under the condition of energy balance).

3.1 MIS Construction

Since any two nodes in MIS cannot have an edge, that is, when a node is in MIS, any other node that has an edge incident to this node cannot be included in the MIS. We use colors to indicate if a node is in MIS or not. The algorithm always starts from a node that initiates (invokes) the execution of the algorithm. We call this node initiator. We use black to indicate the nodes in MIS and grey to indicate non-MIS nodes.

Each node is in one of the four states: white, black, grey and transition. Initially, all nodes are in white, and at the completion of the algorithm all nodes in the network must be either in black (MIS nodes) or in grey (non-MIS nodes). The state transition of a node is done in response to the message it receives. There are three types of messages: 1) BLACK message sent out when a node becomes a black node; 2) GREY message, sent out when a node becomes a grey node; 3) INQUIRY message, sent out when a node inquires the weights and states of its neighbors. Every message contains node state, id and weight in format.

As the start of the algorithm, the initiator colors itself in black. A node that colors itself in black will broadcast a BLACK message to its neighbors (to indicate itself as an MIS node). A neighbor that receives a BLACK becomes a grey node (a non-MIS node), and it broadcasts a GREY message to its neighbors. A node that receives a GREY message needs to compete to become a black node. So it broadcasts an INQUIRY message to its neighbors to inquire their states and weights. It sets a timeout to wait for the replies of the INQUIRY message. The node is in the transition state during this timeout period, because it cannot determine whether it would become black or grey. If it finds it has the highest weight among all its transition state neighbors based on the replies from all its neighbors, its color is changed to black, and it does the same as the other black nodes do. If this node is still in the transition state when the timeout expires, its color changes to white. The algorithm is fully distributed and all nodes execute the same algorithm concurrently. Any node whose neighbors are all colored in black or grey terminates. The MIS construction procedure ends when every node terminates.

```
MIS construction algorithm:

initiator () {
    Color itself black;
    Broadcast a BLACK msg;
}
Each node i, responses to the msg it receives:
MIS-algorithm {
Receive a msg;
If it is black/grey then
    Ignore the msg;
    If its neighbors have no white neighbors then
         Return;
    end if
else
    Switch on message-type {
         Black:
             Color itself grey;
              Broadcast a GRAY msg;
          Grey:
              Broadcast an INQUIRY msg;
              Enter transition state;
              Set a timeout waiting for replies;
              If w(i) is the highest then
                  Color itself black;
                  Broadcast a BLACK msg;
              end if
              If in transition after timeout then
                  Color itself white;
              end if
           Inquiry:
              Reply its own color and w(i);
}
    end Switch
end if }
```

Theorem 1: The set of black nodes represented as B that computed by the first section algorithm is an MIS of the network graph.

Proof: The algorithm colors the nodes of the graph layer by layer, and propagates out from the initiator to reach all nodes in the network, with one layer of black and the next layer as grey. At each layer, black nodes are selected by gray nodes of previous layer and are marked black. The construction incrementally enlarges the black node set by adding black nodes 2 hops away from the previous black nodes set. Also the newly colored black nodes could not be adjacent to each other, for the interleaving coloring layer of black and grey nodes. Hence every black node is disjoint from other black nodes. This implies that B forms an independent set. Further, the algorithm will end up with black or grey nodes only. Each grey node must have at least one black neighbor, so if coloring any grey node black, B will not be disjoint anymore. Hence B is a maximal independent set.

Theorem 2: Considering the propagation layer of MIS, Let B_i and G_i be the set of nodes marked black and grey at i^{th} layer. For a MIS node in B_i, there always exists that it has a neighbor in G_i connecting at least another MIS node in B_{i+1} with it.

Proof: For any node $g \in G_i$ is a non-MIS node formed at the i^{th} layer. In the construction algorithm, it must be selected to be marked grey from white state on receiving a Black message from its black neighbor in B_i. Next, after determining its state, the grey node g sends out a Grey message to all its neighbors in the $i+1^{th}$ layer. The neighbor finds itself with the highest weight among all its transition neighbors will become a black node in B_{i+1}. This implies that there always exists a non-MIS neighbor node $g \in G_i$ has at least two MIS nodes in B_i and B_{i+1} respectively. So for a MIS node in B_i, there always exists that it has a neighbor in G_i connecting at least another MIS node in B_{i+1} with it.

3.2 Connected Dominating Set Construction

In this section, we make interlacing selection of interconnecting nodes (called connectors) into the formation of connected DS based on previous MIS construction, i.e. connectors of black nodes are established in an interlaced fashion during the construction of MIS. When all grey neighbors of a black node terminate the MIS procedure, this implies the first section algorithm for this node and its grey neighbors terminates. Then the black node will enter the second CDS construction to find connectors. Apparently, the CDS section algorithm starts from MIS initiator too because of propagation order of the MIS procedure algorithm.

Our main idea is to employ a Steiner tree in this subsection to connect nodes in MIS. In a graph, a Steiner tree for a given subset of nodes, called terminals. Every node other than the terminals in the Steiner tree is called a Steiner node. The constructed MIS nodes are terminals, and the selected connectors from non-MIS nodes are Steiner nodes. The internal nodes in the Steiner tree become a CDS. We expect to select a small number of Steiner nodes from non-MIS nodes with higher power in order to obtain good efficiency of CDS. We use a greedy approximation algorithm that every black node selects the grey node with maximal black neighbor number as a connector. If two grey nodes have the same black neighbor number, then the one with higher energy level has higher priority.

Each MIS node is in one of the three states: black, transition and blue. Each non-MIS node is in one of the three states: grey, compete and blue. The black and grey are initial state of CDS procedure (after finishing the first MIS procedure), and blue state is final state to indicate the node is in CDS or not. The transition and compete state is the unsure state when a node can't decide itself as a CDS node. There are three types of messages: 1) INQUIRY message, sent out when a black node inquires its grey neighbors about their state and number of black neighbors. 2) INVITE message, sent out to invite a grey neighbor to be a connector. 3) BLUE message, sent out when a node changes blue. At the start of the algorithm, the MIS initiator colors itself in blue. A node that colors itself in blue will broadcast a Blue message to its neighbors (to indicate itself as a CDS node). Next, the black and grey nodes (after finishing MIS procedure) will execute corresponding state transition mechanism.

When a node is black initially, if the node and its grey neighbors have finished the MIS procedure, the black node will broadcast an INQUIRY message and enters transition state. It sets a timeout to wait for the replies of the INQUIRY message. The node is in the transition state during this timeout period, because it cannot determine which node should be selected to behave as a connector. If a node in transition state receives a BLUE message will enter blue state. This implies that it already has a grey neighbor as a connector. Otherwise, the node still has no neighbor as a connector will try to select one. The selection of connectors is based on replies of INQUIRY message, which include the information of black neighbor number and energy level of its grey neighbors. The connector selection rules are: 1) the selected neighbor should be adjacent to at least a blue node. 2) The selected grey node is with maximal black neighbor number. If multiple nodes are found, then we use node energy level as a tie breaking mechanism (higher energy node wins). The rule 1 protects the constructed CDS is a complete component of Steiner tree merged together. The rule 2 protects the CDS with smaller size and energy efficiency. The intuition of transition state is to wait for replies of INQUIRY from its neighbors, and make decision to select a neighbor as a connector. When the node selects a grey neighbor matching the above two condition, it sends out an INVITE message to neighbor, and changes itself to blue state. When enters in blue, the node broadcasts a BLUE message to indicate itself CDS node.

When grey initially, a node response the received messages. A grey node that receives a BLUE message will update its information about number of black neighbors. A grey node that receives an INQUIRY message replies the sender with the number of black neighbors and its energy, and then enters compete state. The intuition behind compete state is to probe the network to see if itself suits as a connector. If a grey node in compete state receives an INVITE message, it is invited as a connector and colored in blue. When enters blue state, the node broadcasts a BLUE message to neighbors. The CDS construction algorithm continues until: 1) Any MIS node colored blue and no white neighbors terminates the procedure. 2) Any non-MIS node terminates the procedure when all its neighbors are colored blue or grey. The same operation continues until every node terminates. The CDS algorithm ends when every node terminates.

```
CDS construction:
Initiator() {
     Color itself in blue;
     Broadcast a BLUE msg;}
Each node i, execute operation according to its state:
CDS-algorithm{
Switch on state-type{
      Black:
           If all neighbors terminate then
              Broadcast an INQUIRY msg;
              Enter the transition state;
```

```
            end if
        Grey:
            Receive a msg;
            If receive an INQUIRY msg then
                Reply its black neighbor number;
                Reply its energy(i);
                Enter the compete state;
            end if
            If receives a BLUE msg then
                Update its black neighbor number;
                Update its energy(i);
            If all neighbors color in blue or grey then
                Return;
         Transition:
            Set a timeout waiting for replies;
            Receive a msg;
            If receive a BLUE msg then
                Color itself in blue;
             else
                Find a neighbor as a connector;
                Send out an INVITE msg to the node;
                Color itself in blue;
            end if
          Compete:
             Receive a msg;
             If receive an INVITE msg then
                 Color itself in blue;
         Blue:
             Broadcast a BLUE msg;
             If all neighbors color in blue or grey then
                 Return ;}
  end Switch}
```

Theorem 3: The set of blue nodes computed by the algorithm is a CDS of the network graph.

Proof: The set of blue nodes are contained by MIS and connectors. MIS is a dominating set, so we only need to proof the connectivity. Let $\{b_0,b_1\ldots b_n\}$ be the independent set, which elements are arranged one by one in the construction order. H_i be the graph over $\{b_0,b_1\ldots b_i\}(1\leqslant i<n)$ in which pairs of nodes are interconnected by connectors. We prove connectivity by induction on j that H_j is connected. Since H_1 consists of a single node, it is connected trivially. Assume that H_{j-1} is connected for some $j\geqslant 2$. Considering message propagation layer in our algorithm, let B_{i-1} and G_{i-1} be the set of nodes marked black and grey at the i-1th layer, respectively. The gray node in G_{i-1} with maximal number of black neighbor and adjacent to a blue node is selected as connecters. According to theorem 2, it's enough to find grey nodes which interconnect B_{i-1} nodes at i-1th layer with B_i nodes in the ith layer. As H_{j-1} is connected, so must be H_j. So the nodes in MIS and connectors set are connected together, and they also form a dominating set. Therefore the set of blue nodes computed by the algorithm is a CDS.

4 Performance Evaluation and Simulations

The message complexity and time complexity of our distributed algorithm are analyzed at first. Since each node sends out a constant number of messages, the total number of message is *O(n).* The use of linear message takes at most linear time.

Theorem 4: Our distributed algorithm has *O(n)* time complexity, and *O(n)* message complexity.

Next, we analyze the size of energy efficient CDS. The following important property of independent sets is that:

Lemma 1: In a unit disk graph, every node is adjacent to most five independent nodes.

Lemma 2: In any unit disk graph, the size of every maximal independent set is upper-bounded by 3.8opt+1.2 where opt is the size of minimum connected dominating set in this unit disk graph.

Theorem 5: In the CDS construction phase, the number of the connectors will not exceed 3.8opt, where opt is the size of MCDS.

Proof: Let B be the independent set and S be the connectors set of a graph. From lemma2: $|B| \leqslant 3.8opt+1.2$. From theorem2 and lemma1, it can be deduced that connectors has black neighbor number ranged from 2 to 5. The worst case occurs when all nodes are distributed in a line. By analyzing utmost situation, the number of gray connecting nodes must be less than the number of MIS nodes (details omitted). $|S| \leqslant |B|-1 \leqslant 3.8opt$. The number of output connecting node will not exceed 3.8opt.

Theorem 6: The approximation factor of our algorithm is not exceeding 7.6.

Proof: Our distributed algorithm includes two phases. One is the MIS construction, and the other the forming of CDS by Steiner nodes. From Lemma2, the performance ratio in the first phase is 3.8. From Theorem 5, the performance ratio is 3.8 in the second phase, so the resulting CDS will have size bounded by 7.6.

In our algorithm, the node with higher power will have bigger chance to become a CDS node. The reconstruction mechanism makes the balance of energy consumption in networks as energy level changes. Our algorithm guarantees that the CDS nodes have good energy efficiency and extend the network lifetime.

The simulation network size is 100-300 numbers of nodes in increments of 50 nodes respectively, which are randomly placed in a 160X160 square area to generate connected graphs. Radio transmission range is 30 or 50m. Each node is assigned initial energy level 1 Joule (J). A simple radio model is used: E_{elec} is energy of actuation, sensing and signal emission/reception. E_{amp} is energy for communication, varies according to the distance d between a sender and a receiver. $E_{amp}= \xi_{fs}$, when $d<d_{crossover}$, and $E_{amp}= \xi_{mp}$ when $d \geqslant d_{crossover}$. The transceiver energy model: mimics a "sensor radio" with E_{elec}50nJ/bit, ξ_{fs} 10pJ/bit/m^2, ξ_{mp} 0.0013pJ/bit/m^4. Data fusion is omitted. We assume there are one event occurs in average 0.5s. The broadcast packet size is 25 bytes, data packet size 100 bytes, and packet header size is 25 bytes.

The data routing takes flooding protocol. We take parameter *timeout* in MIS construction algorithm as 0.5s.

The simulation makes average solutions over 30 iterations of random generating scenes. Fig. 1, 2 shows the size of the dominating set with the increasing number of nodes in the network for a certain transmission radius. ECDS has a good performance with smaller CDS size when comparing with WAA and WLA as the network size increases. Fig. 3 shows the average CDS residual energy as the network size increases for working 150s (one event every 0.5s) with r=50m. ECDS achieves better energy efficiency with much higher residual energy comparing with WAA and WLA. And WLA has the worst energy efficiency for its big size of dominating set. Fig. 4 shows the network lifetime (length of working time until can't construct a backbone for the network) as the network size increases from 100 to 300 nodes when r=50m. ECDS has much better energy performance comparing with WAA and WLA. It can work with longest time until can't construct a backbone any more. Apparently, ECDS has better network lifetime when compared with the other two algorithms.

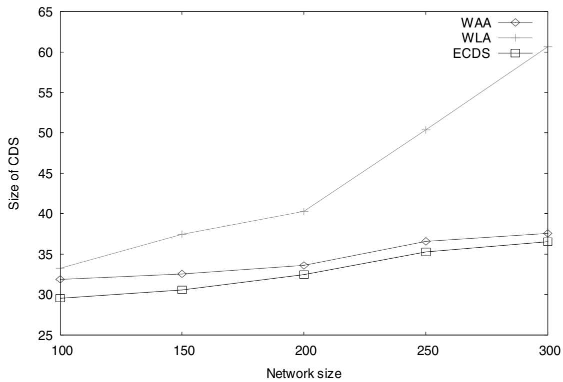

Fig. 1. Size of CDS as the network size increases when r=30m

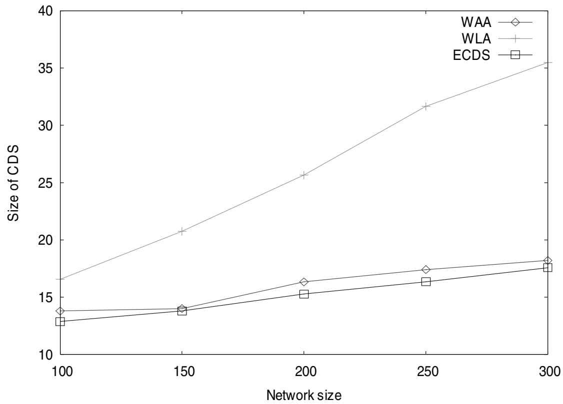

Fig. 2. Size of CDS as the network size increases when r=50m

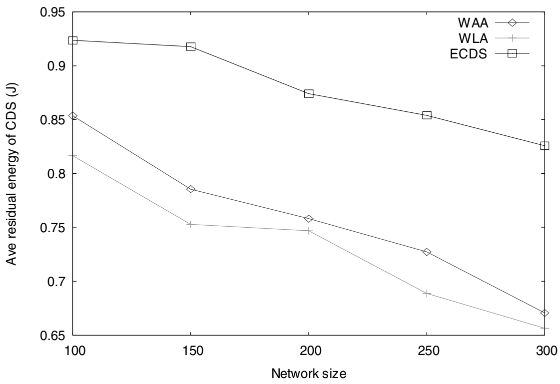

Fig. 3. Average CDS residual energy as the network size increases when r=50m

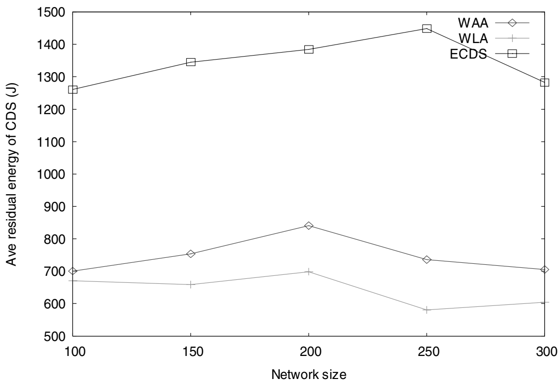

Fig. 4. Network lifetime as the network size increases when r=50m

5 Conclusion and Future Work

A distributed energy efficient backbone based on connected dominating set algorithm for connectivity maintenance of wireless sensor networks is presented. The nodes with higher weight have more chance to be selected as backbone nodes to efficiently manage the network. The algorithm makes energy consumption balanced by computing a new CDS when nods residual energy of network cut down to a certain level. The algorithm time complexity and message complexity of this algorithm are both *O(n)*. The performance ratio is 7.6. Moreover, the algorithm is fully distributed, only uses simple local node behavior to achieve a desired global objective. The simulation results show that the algorithm can efficiently prolong network lifetime and balance node energy consumption with a smaller backbone size, comparing with existing classic algorithms. The future work will focus on simulations under various settings and MCDS improvement with QoS consideration.

References

1. Pottie, G.J., Kaiser, W.J.: Wireless Integrated Network Sensors, Communications of ACM. 43 (2000) 51-58
2. Akyildiz, I.F, Su, W., Sankarasubramaniam, Y., Cayirci, E.: A Survey on Sensor Networks, IEEE Communications Magazine, 40 (2002) 102-114
3. Ephremides, A., Wiselthier, J., Baker, D.: A Design Concept for Reliable Mobile Radio Networks With Frequency Hoping Signaling, Proc. IEEE, 75
4. Wattenhofer, R., Li, L., Bahl, P., Wang, Y.: Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks, Proc. of InforCom (2001)
5. Xu, Y., Bien, S., Mori, Y., Heidemann, J., Estrin, D.: Topology Control Protocols to Conserve Energy in Wireless Ad Hoc Networks, Technical Report 6, University of California, Los Angeles (2003)
6. Xu, Y., Heidemann, J., Estrin, D.: Geography-Informed Energy Conservation For Ad Hoc Routing, In: proc. of 7th Annual Int'l Conf on Mobile Computing and Networking (MobiCom), Rome, Italy. (2001) 70-84
7. Chen, B., Jamieson, K., Balakrishnan, H., Morris, R.: Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks, Proc. MobiCom. (2001) 85-96
8. Clark, B.N., Coloburn, C.J., Bhargavan V.: Unit Disk Graphs, Discrete Mathematic, 86 (1990) 165-177
9. Wan, P.J., Alzoubi, K., Frieder, O.: Distributed Well Connected Dominating Set in Wireless ad hoc networks, in proc. of INFOCOM (2002)
10. Alzoubi, K.M., Wan, P.J.: New Distributed Algorithm For Connected Dominating Set In Wireless Ad Hoc Networks. Proc. 35th Hawaii Int'1 Conf, System Sciences (2002) 3881-3887
11. Min, M., Huang, C.X., Huang, S.C.-H., Wu, W., Du, H., Jia, X.: Improving Construction for Connected Dominating Set with Steiner Tree in Wireless Sensor Networks, to appear in Journal of Global Optimization (2004)
12. Wu, W., Du, H., Jia, X., Li, Y., Huang, C.-H., Du D-Z.: Minimum Connected Dominating Sets And Maximal Independent Sets In Unit Disk Graphs, Technical Report 04-047, Department of Computer Science and Engineering, University of Minnesota (2004)