

# A Distributed Energy-efficient Topology Control Routing for Mobile Wireless Sensor Networks

Yan Ren, Bo Wang, Sidong Zhang, and Hongke Zhang

School of Electronics and Information Engineering, Beijing Jiaotong University,  
100044 Beijing, China  
{yren, bwang, sdzhang, hkzhang}@center.njtu.edu.cn

**Abstract.** One of the fundamental issues in *wireless sensor networks* (WSNs) is the *topology control* (TC) problem, which reflects how well the energy consumption is reduced and the network capacity is enhanced. In this paper, by using computational geometry theoretic, we present a fully distributed routing protocol, *Cooperative Energy-efficient Topology Control* (Co-ETC), whose goal is to achieve energy efficiency in *mobile wireless sensor networks*. Based on underlying routing graph, the proposed scheme allows each node (with or without mobility) to locally select communication neighbors and dynamically adjust its transmission radius accordingly, such that all nodes together self-form a energy-efficient topology. The simulation results indicated that the proposed scheme was feasible. Compared with existing state-of-the-art algorithms and protocols, Co-ETC has better energy-efficiency. Moreover, it can adapt to mobile environment well.

**Keywords:** topology control, energy efficiency, routing protocol, wireless sensor networks.

## 1 Introduction

In recent years, extensive research has been conducted on wireless sensor networks (WSNs), considered one of the top research topics [1]. Such environments may have a large number of small sensor nodes, each capable of collecting, storing, processing observations and communicating over short-range wireless interfaces and multiple hops to central locations called *sinks*. Since sensors may be spread in an arbitrary manner, one of the fundamental issues that arise naturally in WSNs is *topology control* (TC) problem.

Topology control technique is to let each wireless node locally adjust its transmission range and select certain neighbors for communication, while maintaining a structure that can support energy efficient routing and improve the overall network performance [2]. In general, it can be considered as a measure of the quality of service of a WSN. For instance, one may ask how well the network can maintain some global graph property (e.g., connectivity). Furthermore, TC formulations can reduce energy consumption and enhance network capacity.

Due to mobility, sensor nodes (even sink) may change their locations after initial deployment in many scenarios. Mobility can result from environmental influences (e.g., wind or water), available mobile platforms in the deployment area (e.g., robots in object tracking, soldiers in battlefield surveillance) or mobile devices can be incorporated into the design of the WSN architecture (e.g., airborne and vehicles) [3]. The impact of mobility on the effectiveness of TC is twofold: Increased message overhead (especially in the case of high mobility scenarios) and nonuniform node spatial distribution. Considering the impact above, deriving results regarding mobile WSNs is even more challenging.

Several TC algorithms or routing protocols [3], [4] which use different underlying routing graphs with several good properties have been proposed in last few years. However, to our knowledge none of them has been defined to explicitly deal with all four key properties for unicast routing graph on WSNs simultaneously: power spanner, sparse, localized and degree-bounded. By using computational geometry theoretic techniques, we present a cooperative topology control routing protocol with following properties:

1. **Energy-efficient.** It ensures that the routes calculated on given routing graph are at most a constant factor (defined as *power stretch factor* that will be specified later in this paper) away from the power-optimal routes.
2. **Sparse.** There is linear number of edges in given network. It eases the task of finding path and maintaining the route path in the presence of node mobility, and it can reduce the communication overhead.
3. **Bounded node degree.** Bounded node degree can reduce bottleneck and neighbor signal interference in network.
4. **Running in distributed fashion.** Every node compute the scalable routing graph cooperatively using the information only provided by its neighbor.
5. **Adaptive to mobile environment.** This ensures that topology requires little maintenance in the presence of mobility that could change routing graph to some extent.

The remainder of this article is organized as follows: In the next section, we first give some preliminaries and define the network model. The basic design of Co-ETC protocol is presented in Section 3. We then discuss several extensions in Section 4. Specifically, we consider how to find an energy-efficient-path and how to avoid “bottleneck” node in communication graph. Finally, we evaluate the performance of Co-ETC in Section 5, conclude the paper and discuss possible future research directions in Section 6.

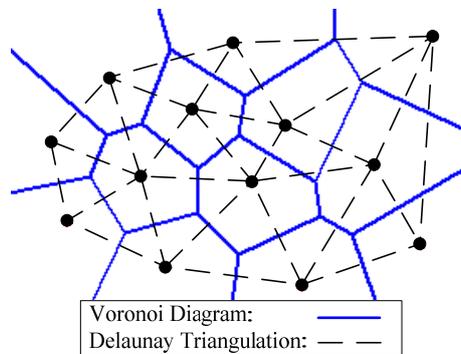
## 2 Preliminaries and Network Model

In our subsequent discussions, network topology is represented by an undirected simple graph  $G = (V, E)$  when all the nodes transmit at maximum power, where  $V = \{v_1, v_2, \dots, v_n\}$  is the set of  $n$  wireless sensor nodes and  $E$  is the set of links in the WSN. For sake of simplicity, we will assume that every node has the same maximum transmission range  $R_{max}$ . A node can reach all nodes (called *neighbors*) inside its

transmission region. We also assume that each node is assigned a unique identifier,  $ID$  and knows its location.

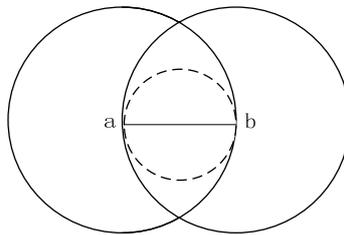
Before we design the Co-ETC protocol, we first give some computational geometry concepts, which will be used as the foundations of geometry for Co-ETC:

**Definition 1 (Voronoi Diagram and Delaunay Triangulation).** The Voronoi diagram, denoted by  $VD(V)$  (real lines in Fig. 1), of a set of discrete nodes partitions the plane into a set of convex polygons (called *voronoi regions*) such that all sites inside a voronoi region are closest to only one node. This construction effectively produces polygons with edges that are equidistant from neighboring nodes. The Delaunay triangulation, denoted by  $DT(V)$  (dotted lines in Fig. 1), is the dual graph of the Voronoi Diagram. It is the unique triangulation such that the circumcircle of every triangle contains no nodes of  $V$  in its interior.



**Fig. 1.** Voronoi Diagram and Delaunay Triangulation

**Definition 2 (Gabriel Graph).** The Gabriel Graph, denoted by  $GG(V)$ , consists of all edges  $uv$  such that the  $disk(a, b)$  does not contain any node from  $V$ , where  $disk(a, b)$  is the closed disk with diameter  $ab$  (see Fig. 2).



**Fig. 2.** Gabriel Graph

**Definition 3 (Power Stretch Factor).** The power stretch factor, denoted by  $\rho_p$ , respect to  $G$  is the maximum over all possible node pairs of the ratio between the cost of the minimum-power path in  $G'$  (Any arbitrary subgraph of  $G$ , e.g.,  $DT(V)$  and  $GG(V)$ ) and in  $G$ , where is represented by  $p_{G'}(u, v)$  and  $p_G(u, v)$  respectively. In other words,

$$\rho_p = \max_{u,v \in V} \frac{p_{G'}(u,v)}{p_G(u,v)} \quad (1)$$

Notice that, generally, we would like to use a subgraph  $G'$  (also called a routing graph) which has a low-power stretch factor and which is sparser than the original graph  $G$ . Such a routing graph can be used to compute routes with guarantee that the energy needed to communicate along is almost minimal. In addition, computing optimal routes in  $G'$  is easier and has less communication overhead than in  $G$ . Moreover, such a sparse routing graph requires little maintenance in the presence of node mobility. The power stretch factor and maximum node degree of the graphs defined previously have been analyzed in [5], and are reported in Table I.

**Table 1.** Power Stretch Factor and Maximum Node Degree of  $DT$  and  $GG$ .

	Power Stretch Factor	Maximum Node Degree
DT	$\left(\frac{1 + \sqrt{5}}{2} \pi\right)^\alpha$	$\Theta(n)$
GG	1	$n-1$

As shown, the Gabriel Graph is energy-optimal since it has a power stretch factor of 1. And neither  $DT(V)$  nor  $GG(V)$  has constant maximum node degree. We will utilize these attractive routing graphs for extensions of our Co-ETC protocol in Section 4.

### 3 Design of Co-ETC Protocol

In this section, we consider base design of the Co-ETC protocol, which enables each node to maintain the relevant part of the Voronoi routing graph efficiently and consistently, with the knowledge only provided by its neighbors, and manage topology changes due to mobility, node joining/leaving. The motivation of the proposed protocol stems from the commonality encountered in the mobile WSNs. Co-ETC is composed of three main procedures: route graph setup, node joining and leaving, and node movement.

### 3.1 Routing Graph Setup

Initially, when a node wants to communicate with another one, no prior knowledge of the routing graph is available. For sake of simplicity, we will assume that every node has the same initial transmission range  $R$ . This ensures routing graph is consistently for adjacency neighbors. To avoid message collision, every node waits for a random back-off time period  $T_i$ , then simply flood a query with its own  $ID$  and coordinates to neighbors. After every node has all its neighbors' coordinates according to the definition of  $VD$  in Section 2, each node generate the local  $VD$  itself, and the union of local graphs corresponds to all the sensors in  $V$  (see Fig. 3). Notice that, each node has to have enough  $R$  to reach its neighbors at the beginning. Generally, we assign it equal to the maximum transmission range  $R_{max}$ .

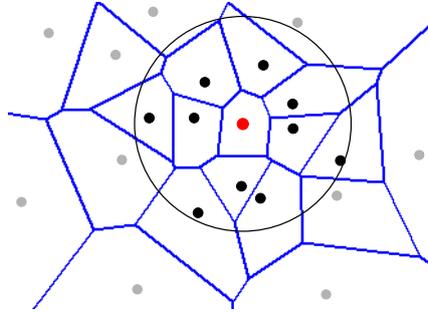


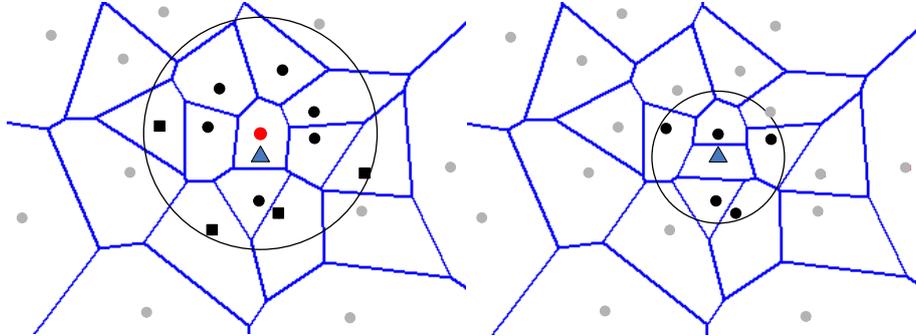
Fig. 3. Voronoi Diagram in WSN

In above routing graph, each node maintains a Voronoi diagram of all neighbors and directly connects to them with minimal latency. As only a few neighbors are kept, the cost to maintain such a routing graph at each node is low.

### 3.2 Node Joining and Leaving

A joining node (mark as  $\blacktriangle$ ) first sends a *Query* message with its joining coordinates and  $ID$  to neighbors. Any acceptors (the node whose region contains the joining node's coordinates) will respond by sending a list of joining node's *adjacency neighbors* (the node whose Voronoi regions border the given node's, e.g., we mark such nodes as  $\bullet$  in Fig. 4). Notice that, there must be only one acceptor except it on the Voronoi lines. The joining node first computes and organizes its local Voronoi diagram, and then connects to each adjacency neighbor of itself (affected by joining). Related neighbors also update their Voronoi diagram to account for the joining node.

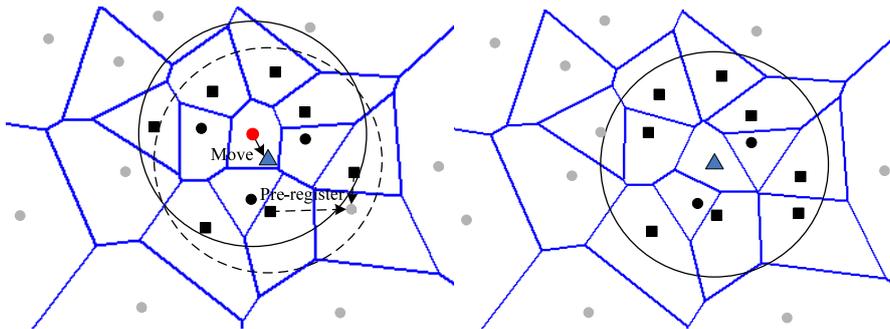
When a node will leave from WSN due to node failures or power duty cycling reasons etc., the leaving node simply disconnects. Its adjacency neighbors will update their Voronoi diagrams, replacements are learned via other still-connected adjacency neighbors. The procedure can be seen as the converse procedure of Fig. 4 (node  $\blacktriangle$  leave the network).



**Fig. 4.** Node Joining Procedure. (Left) Before node joining. (Right) After node joining.

### 3.3 Node Movement

When a node moves, its position updates and transmission radius are sent to all connected neighbors. If the recipient is a *periphery neighbors* (the node whose adjacency neighbors may partially lie outside transmission range, we mark such nodes as ■. Notice that, some adjacency neighbors may be periphery neighbors at the same time.) it will check whether any of its adjacency neighbors tend to be visible to this moving node and send a *Pre-register* message with position update. In this way, new neighbors can estimate the connecting time and enhance the resilience of routing graph. When a new neighbor is detected, notifications are sent to the moving node for initiating communications. The moving node also disconnects any periphery neighbors that have left its transmission range. This procedure can be explained in Fig. 5.



**Fig. 5.** Node Movement. (Left) Before node movement. (Right) After node movement.

The three main procedures of Co-ETC protocol above illustrate that nodes communicate with each other efficiently and the changes to routing graph are localized. Besides, protocol is scalable and adaptive to mobile environment.

## 4 Extensions

In this section, we consider two extensions of the Co-ETC protocol and present efficient distributed algorithms: 1) find an energy-efficient-path between nodes  $u$  and  $v$  in  $V$ ; 2) adjust node degree of existing routing graph dynamically to avoid “bottleneck”.

From section 2, we have known that the Gabriel Graph is energy-optimal since  $\rho_p(GG) = 1$ . However, we cannot use this graph directly in our extension because: 1) the Gabriel Graph may have long edges, while we are only allowed to connect points within limited transmission range and 2) the empty-disk rule is a global rule and is not suitable for local computation. To deal with those two problems, we consider the following algorithm to find an energy-efficient-path constructed by edges of Gabriel Graph according to existing Voronoi routing graph:

### Algorithm 1: Energy-Efficient-Path ( $V, u, v$ )

1. From the source point  $u$ , node  $u$  adds an edge  $uv_1$  if and only if the  $disk(u, v_1)$  does not contain any node from  $V$ .
2. Continually add all such edges into new routing graph until there is no edge which is belonged to the Gabriel Graph  $GG(V)$ . Notice that, because the original graph is a subgraph of  $G$ , so not every edge of  $GG(V)$  can be added into new routing graph.
3. Assign each constructed edge  $v_i v_j$  the weight equal to  $\|d_{ij}\|^\alpha$ .
4. Run the Bellman-Ford distributed shortest path algorithm [6] to compute the shortest path connecting  $u$  and  $v$ , which has the minimum weight among all paths between  $u$  and  $v$ .

The correctness of the algorithm is based on the following observation: If there is a sensor node  $v_k$  inside  $disk(v_i, v_j)$ , then  $\|d_{ik}\|^\alpha \leq \|d_{ij}\|^\alpha$  and  $\|d_{jk}\|^\alpha \leq \|d_{ij}\|^\alpha$ . It is obvious that the path  $v_i v_k v_j$  is in the  $GG(V)$ . Thus, the path by substituting edge  $v_i v_j$  with edges  $v_i v_k$  and  $v_k v_j$  consumes less energy, which is a contradiction. Consequently, edge  $v_i v_j$  must be a Gabriel edge. For this reason, those extra long edges that are not belong to old Delaunay Triangulation also will not be added into our new routing graph.

According to the analyses in Section 2, neither  $DT(V)$  nor  $GG(V)$  has constant maximum node degree. It means that there will be bottleneck nodes in our routing graph, and these nodes are forced to connect beyond their capacities. So we adjust node degree of existing routing graph dynamically as following:

### Algorithm 2: Dynamic-Node-Degree-Adjustments ( $V$ )

1. For all nodes whose neighbors exceed the upper bound value  $N_{max}$ , decreases its transmission power by  $\Delta P$  once a step, until it connects  $N_{max}$  neighbors. At the same time, original node sends *Hello* message and attaches to it with current link neighborhood list and current transmission power, denoted by  $P_{original-node}$ .
2. Whenever another node, which so far does belong to the neighborhood list, hears the *Hello* message of the original node for the first time. For avoiding one-directed link, it first compares  $P_{original-node}$  and  $P_{itself}$ .
  - If  $P_{original-node} > P_{itself}$ , it leaves its transmission power as before, whichever is smaller.

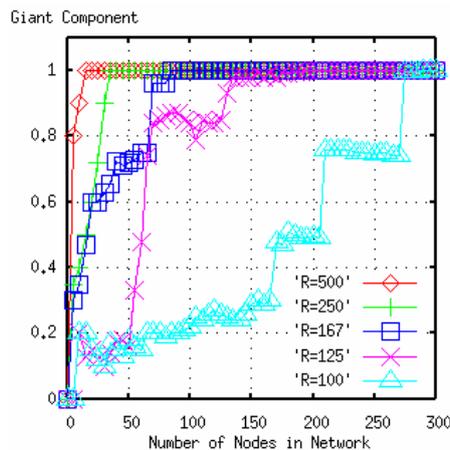
- Or else it set its transmission power to the original node equal to  $P_{original-node}$  (adopts directional antennas for transmission) and then begins the same procedure like original node.
3. Node answers *Hello* message with *Reply* message and attaches to it only with current transmission power.
  4. Node will restores to the preferred transmission power if the number of neighbors falls below the lower bound  $N_{min}$ .

## 5 Evaluation and Simulation Results

To demonstrate the feasibility and effectiveness of our Co-ETC protocol, we have done an across-the-board performance evaluation of this protocol according to several vital criteria including network connectivity, network-lifetime, and robustness to mobility. We implement our Co-ETC protocol on the NS-2 simulator using an idealized MAC layer with fixed link error probability. The simulation environment is generated by using MATLAB. In simulation, the network nodes are uniformly distributed in a square area  $A$  of size  $1000*1000$ .

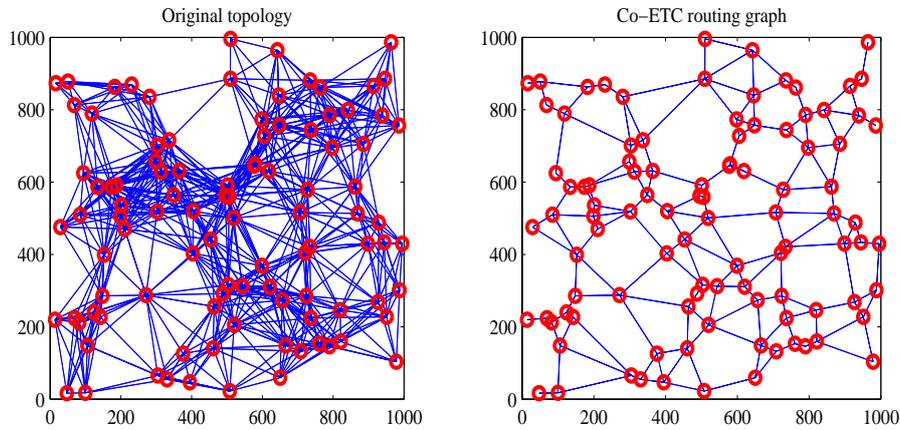
### 5.1 Network Connectivity

Perhaps, the most basic requirement of a topology is that it be connected. More precisely, to ensuring routing graph efficiently constructed, nodes in network must have enough initial transmission range  $R$  to reach its neighbors at the Route-Graph-Setup stage (Section 3.1). In simulation, we assign it equal to the maximum transmission range  $R_{max}$ . For every value of  $R$  considered in the simulations, we generated 100 random placements and, for every placement, we evaluated the percentage of nodes belong to the largest connected component. With different  $R$ , the average percentage of nodes which are in the largest strongly connected component related to the number of nodes is shown in Fig.6.



**Fig. 6.** Percentage of nodes that belong to the largest connected component

As shown in Fig. 6, there are critical thresholds under different transmission range  $R$ . The sharpness of the threshold depends on the  $R$ . Based on these results; we can choose appropriate  $R$  for the initial scheme of different numbers of nodes. Fig.7 has shows the actual topologies for one simulated network with 100 nodes with maximum transmission range  $R_{max}=250$ . Fig.7(a) and (b) are the topologies resulting from original graph  $G$  and our Co-ETC routing graph. As can be seen, Co-ETC maintains network connectivity in its routing graph.



**Fig. 7.** Topology graph. (Left) Original topology. (Right) Co-ETC routing graph.

## 5.2 Network-Lifetime

We compared the Co-ETC protocol (after adding extensions) presented in the previous sections to K-NEIGH protocol [3] and CBTC algorithm [4]. Notice that, the CBTC algorithm (with  $\alpha=2/3\pi$ ) and K-NEIGH protocol were all with pruning optimization in simulation. Since CBTC algorithm does not provide routing by itself, we used the AODV [7] routing protocol for it in simulation. The same network setup was used to compare the implementations of routing protocols. 100 sensor nodes were randomly placed in area with  $R=250$ . Each node had initial energy of 0.5 Joule. Five of them were source nodes, which produce CBR data traffic. The length of a data packet was 64 bytes. One node was assign as *sink* node. Nodes moved randomly with constant speeds  $10/\text{time}$ . Node bounced back once hitting the boundary.

In WSNs, the metric of actual interest is not the energy-efficient of per packet, but the whole operational lifetime of WSN. So we use network lifetime as the metric to evaluate the energy-efficient performance of protocols. We consider the network to be down until there is no data packet can be received by sink node.

Fig.8 illustrates the normalized network lifetime simulation results of Co-ETC, and the reference K-NEIGH and CBTC+AODV ( $\alpha=2/3\pi$ ) under different network loads in

both static and mobile scenarios. It is shown that Co-ETC extends the network-lifetime significantly in the mobile scenario. Lifetime at most a factor of 2.2 those of K-NEIGH and CBTC+AODV could be reached. Although Co-ETC is designed to be efficient in dynamic WSNs, it also prolongs the lifetime 10-45 percent in static scenario. Contrary to K-NEIGH and CBTC, our Co-ETC protocol is almost independent of the choice of scenario: static or mobile. This is due to the fact that the constructed routing graph of Co-ETC has good property adapt to mobile environment.

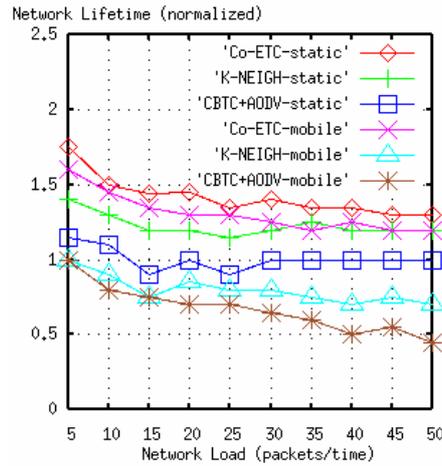


Fig. 8. Network lifetimes of different schemes

## 6 Conclusions

In this paper, we consider energy-efficient topology control routing in mobile wireless sensor networks. We present a distributed routing protocol, Co-ETC, and indicate that base on underlying routing graph, it has better energy-efficiency. Moreover, it can adapt to mobile environment well.

Although Co-ETC outperforms other existing state-of-the-art algorithms and protocols in terms of network lifetime, and has desirable property including mobility, sparse, and localized. However, we have assumed that all nodes have circular communications ranges. In practice, this is often violated due to fading and multipath effects. It would be interesting to extend results of this paper to more realistic scenarios. Furthermore, we will further do research on using more complex mobility models such as group mobility.

**Acknowledgments.** This work is supported by National Science Foundation of China. (No. 60473001, 60572037, 60573001), and the Innovation Foundation of Science and Technology for Excellent Doctorial Candidates of Beijing Jiaotong University (No. 48013)

## References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. *Computer Networks*, Vol. 38. (2002) 393-422
2. Rajaraman, R.: Topology control and routing in ad hoc network: A survey. *SIGACT News*, Vol. 33. (2002) 60-73
3. Blough, D.M., Leoncini, M., Resta, G., Santi, P.: The k-neighbors protocol for symmetric topology control in ad hoc networks. *Proc. ACM Mobihoc'03*, (2003) 141-152
4. Li, L., Halpern, J.Y., Bahl, P., Wang, Y.M., Wattenhofer, R.: A cone-based distributed topology-control algorithm for wireless multi-hop networks. *IEEE/ACM Transactions on Networking (TON)*, Vol. 13. (2005) 147-159
5. Wang, W., Li, X., Moaveninejad, K., Wang, Y., Song, W., The spanning ratio of  $\beta$ -skeletons. *Proc. CCCG*, (2003) 35-38
6. Cormen, T.J., Leiserson C.E., Rivest, R.L.: *Introduction to Algorithms*. Massachusetts: MIT Press/ New York: McGraw-Hill (1990)
7. Perkins, C.E., Royer, E.M.: Ad-hoc on-demand distance vector routing. *Proc. IEEE WMCSA'99*, (1999) 90-100