

# A New Distributed and Power-Efficient Topology Control Algorithm for Wireless Ad-Hoc Networks

Saeid Taghavi Afshord and Bager Zarei  
Computer Engineering Department, Shabestar Branch, Islamic Azad University, Shabestar, Iran  
{taghavi, zarei.bager}@iaushab.ac.ir

Bahman Arasteh  
Computer Engineering Department, Tabriz Branch, Islamic Azad University, Tabriz, Iran  
b\_arasteh@iaut.ac.ir

**Abstract** — To guarantee the performance of ad hoc networks, utilizing a hierarchical architecture model is necessary. An instance of this structure is clustering. In this paper a cluster-based topology control algorithm is proposed which builds an energy-efficient and a low interference topology. It uses low-quality information, exchanges a few messages, and does not need for extra hardware equipment as well. It is also suitable for practical use, because in the implementation of this algorithm is not needed to know the location or the direction information of the network nodes. So it is possible to classify this algorithm as a neighbor-based topology control algorithm. In addition, for the transmitting and the maintained power levels of each node, a modification is applied according to the real hardware platforms. It improves the energy consumption in the ideal conditions.

**Index Terms** — Ad hoc networks, clustering, topology control, energy consumption

## I. INTRODUCTION

In mobile ad-hoc networks (MANETs), all the nodes are cooperating to achieve for a global specified task, such as area monitoring and data gathering. It is important to make an appropriate transmission power selection for each node, to reduce energy consumption and signal interference. This is called topology control (TC), while it is still satisfying the certain global constraints [1]. In the shorter definition, topology control is to determine the transmission power of each node so as to maintain network connectivity and consume the minimum transmission power [2].

At least two issues have motivated the researchers to study the realm of TC techniques: reducing the energy consumption and increasing the network capacity [3]. A TC algorithm tends to manage the logical communication of the network nodes toward the maximum transmitting range. It deals with the optimization of the power

consumption while keeping the network connected. The nodes mobility and updating their information are the most important problems in wireless ad hoc networks; because with moving a single node, the information of all the nodes would be updated. So optimizing the energy usage in these networks is essential for each node and for the whole network, as well [4]. The distributed TC, especially neighbor-based TC protocols are more suitable to the implementation in mobile ad hoc networks [3].

The paper is organized as follows. The basic concepts are introduced in Section 2. The most similar and well-known protocol, *KNEIGH*, for comparison purpose, is briefly presented in Section 3. In Section 4, the proposed protocol along with simulation and experimental results is explained in details. Section 5 concludes the paper and discusses about the achieved properties.

## II. BASIC CONCEPTS

### A. A proposition from graph theory

**Proposition.** Let  $S = \{x_1, x_2, \dots, x_n\}$  be a set of points in the plane such that the distance between any two points is at least one. Show that there are at most  $3n$  pairs of points at distance exactly one [5].

**Proof.** Define graph  $G$  on space  $S$  as follows: the set of points  $x_1, x_2, \dots, x_n$  be as  $V(G)$ , and two arbitrary and distinct vertices  $x_i$  and  $x_j$  from  $V(G)$  is adjacent if and only if the distance between them be exactly 1. Clearly, this kind of edges will construct  $E(G)$ .

It is enough to prove that for each vertex  $x_i$  ( $1 \leq i \leq n$ ),  $d(x_i) = 6$  (that is, the degree of vertex  $x_i$  is equal to 6). Because it means that, each arbitrary vertex  $x_i$  has been linked at most with 3 pairs of  $V(G)$  at distance exactly 1. With considering the all of  $n$  vertices of  $G$ , the assertion will be established.

For that, draw an imaginary circle with radius 1 centered at  $x_i$ . According to the assumption, it is clear that any vertex of  $G$  does not exist inside the circle and according to the definition of  $G$ , any adjacent vertex with  $x_i$  can not be at the outside of the circle, and if a vertex of  $G$  is on this imaginary circle, it should be adjacent with  $x_i$ . We show that at most six vertices of  $G$  can be situated on the circle. Let the vertices  $x_{i1}, x_{i2}, \dots, x_{i6}$  have such properties. This is feasible and the reason is as follows. Suppose that these six vertices have been situated in equal distances and, without loss of generality, they are on the circle at the mentioned order. The central angle of two adjacent vertices (for example  $x_{i1}$  and  $x_{i2}$ ) of this circle will be  $360/6 = 60$  and consequently  $x_{i1}x_{i2}$  will be an 60 degree angle. Therefore the triangle  $x_{i1}x_{i2}x_i$  is equilateral and thus the distance between  $x_{i1}$  and  $x_{i2}$  is exactly equal to 1.

It is evident that the distance between two non-adjacent vertices on the circle is greater than 1. So the existing of six vertices on this circle is possible. If the arbitrary vertex  $x_k$  is also situated on the supposed circle, using the simple geometrical calculations, it is obtained that the distance between at least two vertices of these seven vertices will be less than 1, which is a contradiction.

### B. Clustering

Clustering algorithms organize the network into a set of clusters, which are used to facilitate the routing operations between nodes and/or to better balance the energy consumption in the network. Clustering techniques are more often useful in such networks that composed of a large number of nodes.

In a typically clustering protocol, mobile nodes are divided into different virtual groups based on criteria such as geographical distance, communication quality and so on. Under a cluster structure, mobile nodes may be assigned a different status or function, such as cluster head, cluster gateway, or cluster member [6]. Message routing is then performed on the basis of a two-level hierarchy: the message originating at a cluster node is destined to the clusterhead, which decides whether to forward the message to another clusterhead (inter-cluster communication) or to deliver the message directly to the destination (intra-cluster communication). The clusterhead might also perform other tasks such as coordinating sensor node sleeping times, aggregating the sensed data provided by the cluster nodes, and so on.

Although clustering protocols can be seen as a means of controlling the topology of the network by organizing its nodes into a multilevel hierarchy, a clustering algorithm does not fulfill the definition of topology control since typically the transmit power of the nodes is not modified. In other words, a clustering algorithm is concerned with hierarchically organizing the network units assuming the nodes transmitting range is fixed, while a topology control protocol is concerned with how to modify the nodes transmitting ranges in such a way that a communication graph with certain properties is generated [3].

### C. Per-packet and periodical topology control

An important approach in classification of topology control protocols toward communicating each node with its neighbors is distinction between per-packet and periodical topology control. In this paper, we point out the former approach as usual per-packet. In this technique, every node  $u$  maintains for each node  $v$  in its neighbor list, a transmitting power level, which is used in the sending packets to  $v$ . This is typically the minimum power that needed to reach it. Using this technique, a node can send each packet with the minimum possible energy usage, and spatial reuse is also increased. Besides individual transmitting power levels for each neighbor, every node in the network also sets a broadcast power level, which is used to send a message to all its one-hop neighbors simultaneously. Typically, the broadcast power is set to the minimum level that needed to reach the farthest node in the neighbor list [7].

In the periodical approach, the management of the power levels is simplified: a node maintains only the neighbor list and the broadcast power level. Each packet is sent using the same power level, independent of the actual neighbor to which is destined. By setting this common power level to the broadcast power, we are ensured that the messages are correctly received by the interested neighbor.

Usual per-packet TC mostly relies on quite accurate information on node locations, and it is typically applied in combination with location-based or direction-based topology control. While usual per-packet TC is in general more efficient in stationary networks (if certain technological problems can be solved) and actually it is implicitly used in many TC protocols, periodical TC is probably the only feasible choice in mobile networks. But periodical technique has two main weaknesses: it often wastes energy and, it reduces spatial reuse and consequently causes the more interference in the network [3].

### D. Ideal features of a topology control protocol

It is known that the centralized approaches for topology control are doomed to perform poorly in realistic application scenarios. So, only solutions that can be implemented in a *fully distributed* and *asynchronous* fashion have some practical relevance.

Another important feature of a protocol for topology control is locality, which refers to the nodes ability to build their view of the network topology only by using local information. That is, information regarding up to  $h$ -neighbors in the max-power graph (the communication graph that is generated when all the nodes transmit at maximum power), where  $h$  is a small constant (2-3 at most). Localized solutions have several advantages with respect to approaches that require network wide information exchange. Since network nodes can build their local view of the topology by exchanging few messages with neighbor nodes only, localized protocols can be classified as lightweight solutions, which can be implemented in very large networks also; furthermore, locality implies that the network topology can be easily

reconfigured when nodes leave/join the network, or in presence of node mobility.

Anyway, the goal of a distributed topology control protocol is to build a ‘reasonably good’ topology. But what are the factors of a ‘reasonably good’ topology? What are the essential features that the generated topology should have? The topology generated by the topology control protocol should rely on bidirectional links only. Furthermore, it is desirable that the topology control protocol preserves connectivity. In other words, if the network is connected when all the nodes communicate with maximum transmit power (i.e. if the max-power graph is connected), then it should preserve this property also after every node in the network has executed the topology control protocol. So, only redundant links should be removed from the network topology.

There is a considerable difference between requiring connectivity preservation in the worst case (i.e. for any node placement, if the max-power communication graph is connected, then the network remains connected after the execution of the topology control protocol) and requiring connectivity with high probability. The former property can be achieved only if the physical node degree in the network is unbounded, while the latter can also be achieved when the numbers of physical neighbors of a node are upper bounded. Building a network topology in which nodes have small physical degree is highly desirable, since this parameter is a measure of the interference generated by a transmitting node: if the physical degree of node  $u$  is small, the number of nodes impacted by  $u$ ’s transmission is relatively small, and spatial reuse is increased. So, it can be stated that at least two of the desired topological properties discussed above are conflicting: ensuring worst-case connectivity and generating a topology with small physical node degree.

A final aspect to consider in the design of a topology control mechanism is the quality of information required by the protocol: since obtaining very accurate information such as node locations is, in general, quite expensive (in terms of additional hardware required on the nodes, or message overhead, or both), it is desirable that the protocol relies on ‘low-quality’ information.

Summarizing, a topology control protocol should have the following properties:

1. be fully distributed and asynchronous
2. be localized
3. generate a topology that preserves the original network connectivity and relies on bidirectional links only
4. generate a topology with small physical node degree
5. rely on ‘low-quality’ information

#### E. Level-based topology control

While investigating topology control problem from theoretical point of view is important but, considering practical features is also useful. In utilizing the practical approaches of TC, inevitably we encounter with some problems. For example most of the TC solutions implicitly assume that the transmitting power level of a node can be set to an arbitrary level, provided it does not

exceed the maximum possible power level. In general, this assumption is not true and most of the wireless cards on the market do not even allow to change the transmit power level. But, fortunately some types of commercially available wireless cards, such as *CISCO Aironet* cards, allow the changing of the transmitting power level. However, this can only be set to a limited number (typically, below 10) of predefined power levels. For instance, *CISCO Aironet 350* can use six different power levels, which is corresponding to a nominal transmit power of 1, 5, 20, 30, 50, and 100 *mW*, respectively.

Motivated by this observation, a set of recently proposed protocols, approach the TC problem by explicitly taking into account this feature of the current wireless transceivers, that is, the availability of only few different transmitting power levels. In [3] these solutions have been called the level-based topology control protocols. However, in attention to the hardware and technological constraints, these protocols mostly have been tested and implemented with combination of the routing protocols [8].

### III. INTRODUCING KNEIGH PROTOCOL

The *KNEIGH* protocol is one of the well-known distributed protocols in the literature. Our proposed protocol is similar to this protocol. But according to the simulation results, it will be shown that our protocol has better properties than *KNEIGH*. Because of the importance and more referring to this protocol in continue, we explain it briefly from the [3, 9].

#### A. Some needed definitions

**Definition 1 (K-neighbors graph)** Let  $N$  be a set of nodes deployed in a certain region  $R$ , with  $|N| = n$ . Given any  $k$ , with  $0 < k \leq n - 1$ , the  $k$ -neighbors graph built on  $N$ , denoted by  $G_k = (N, E_k)$ , is the directed graph obtained connecting each node to its  $k$  closest neighbors. Formally, the directed edge  $(u, v) \in E_k$  if and only if  $\delta(u, v) \leq d_k(u)$ , where  $d_k(u)$  is the distance between node  $u$  and its  $k$ -closest neighbor.

**Definition 2 (Symmetric k-neighbors sub graph)** The symmetric  $k$ -neighbors sub graph on node set  $N$  is defined as the undirected graph  $G_k^- = (N, E_k^-)$ , where the undirected edge  $(u, v) \in E_k^-$  if and only if  $(u, v) \in E_k$  and  $(v, u) \in E_k$ .

In other words, the symmetric sub graph of  $G_k$  is obtained by removing all the unidirectional links in  $G_k$ .

#### B. Distance estimation techniques

There are many distance estimation techniques in the literature. Among them we cite the following:

- **Radio signal strength indicator:** Distance is estimated by comparing the transmitted power at the sender (which is piggybacked in the message) with the received power at the receiver of the message. This

technique can be implemented without any additional hardware on the nodes (RSSI registers are a standard feature in many wireless network cards [10], but its accuracy is bonded to the accuracy of the radio channel model used to predict path loss. Since path loss is very difficult to predict in many environments (especially in presence of buildings, obstacles, and so on), it turns out that RSSI-based distance estimation provides reasonable accuracy only in a quite idealized setting (e.g. football field with all the nodes positioned at the ground level) [10];

- **Time of arrival:** Distance is estimated by comparing the time of arrival of different types of signals. Typically, the radio signal is used in combination with acoustic, ultrasound, or infrared signals. Because of the use of different types of signals, ToA-based techniques provide a much better accuracy than RSSI-based mechanisms, and can be implemented at a reasonable hardware cost. For example, the technique proposed in [11] uses a standard PC sound card to generate an acoustic signal, which is received by a cheap microphone. The authors show that this technique provides good accuracy (below 3%) in realistic conditions. However, accuracy drops to only 23% when the line of sight between the nodes is obstructed by heavy obstacles [3].

### C. Protocol description

The *KNEIGH* protocol introduced in [9] is a distributed implementation of the computation of  $G_k^-$  based on distance estimation. In other words, it is assumed that when a node  $u$  receives a message from node  $v$ ,  $u$  is able to estimate (possibly with a certain error) the distance to node  $v$ .

This protocol is very simple. Initially, every node broadcasts its  $ID$  at maximum power (usually it is assumed that all the nodes have the same maximum transmit power  $P_{max}$ , and that the wireless medium is symmetric). Upon receiving broadcast messages from other nodes, every node keeps track of its neighbors, storing for each of them the estimated distance (this can be done by using one of the techniques described in previous section). After all the initial messages have been sent, every node in the network knows its neighbor set and the distance-based ordering of the neighbors. Given this information, every node computes its  $k$ -closest neighbors list  $KN$ , and broadcasts this information at maximum power. By exchanging neighbor lists, nodes are able to determine the set of symmetric neighbors (two nodes are symmetric neighbors if and only if they appear in each other's  $KN$  list) and to exclude the asymmetric neighbors from  $KN$ . At the end of the protocol execution,  $KN(u)$  contains the list of neighbors of node  $u$  in the final topology  $G_k^-$ , and the (broadcast) transmit power of node  $u$  is set to the minimum value needed to reach the farthest node in  $KN(u)$ . Note that this value can be computed given the received signal strength of the messages sent by the farthest node in  $KN(u)$ .

## VI. THE PROPOSED PROTOCOL

In the proposed method, the virtual clusters over the nodes are constructed (phase 1), then the nodes of each cluster are connected together in a creative manner (phase 2) until the final protocol to be obtained. The phase 2 is established in two steps, intra-cluster and inter-cluster connections. Several clustering methods have been proposed in [12] but our method is different than the previous ones.

### A. The Protocol description

We called the proposed algorithm *CLNeigh*. It is summarized in Figure 1 and it is as follows.

Phase 1: initially every node  $u$  sets its unique cluster number,  $cln(u)$ , which is equal to its  $ID$ , then it broadcasts its  $ID$  at the maximum transmitting power (we assume that all the nodes have the same maximum transmitting power  $P_{max}$ , and the wireless medium is symmetric).

Upon receiving the broadcast messages from the other nodes, every node keeps the track of its neighbors (neighbors  $ID$ ), and stores for each of them the estimated distance (this can be done by using one of the techniques which have been described in [3]). After all the initial messages were sent, every node in the network knows its neighbor set and the distance-based ordering of the neighbors. Using this information, every node  $u$  selects the closest neighbor  $v$  and adds  $v$  as a neighbor at the final topology, and then it sends *ADDNeigh(u)* message to  $v$ . Upon receiving this message by  $v$ , it adds  $u$  in the own neighbor list at the final topology, and then it sends the own cluster number,  $cln(v)$  to  $u$  with *Ack* message. When node  $u$  received this message, it replaces the own cluster number with  $cln(v)$ . If the cluster numbers of the other nodes in the network is equal to  $cln(u)$ , by sending consequent messages via neighbors, they also change the own cluster numbers to  $cln(v)$ . After performing this process for all the nodes, the phase 1 will be terminated and consequently the separated clusters will be constructed.

Phase 2: as it was mentioned above, each node already has obtained its own neighbor list at the maximum transmitting power. So, at the beginning of phase 2, every node  $u$  broadcasts its cluster number again at the maximum transmitting power. Upon receiving the broadcast message by the other nodes, each of them updates its own  $N(u)$ .

Constructing the final topology is quite simple. Every node  $u$  in its ordered neighbor list  $N(u)$  begins from the second neighbor (the first neighbor as a closest neighbor already has been added to the neighbors list at the final topology) and it selects five neighbors, respectively. They are selected in sequence, from distinct clusters for the final topology (of course, if it has had the enough number of neighbors). At the end of the protocol construction,  $FN(u)$  contains the list of neighbors of node  $u$  at the final topology.

Similar to the other neighbor-based TC protocols, the (broadcast) transmitting power of node  $u$  is set to the minimum value which is needed to reach to the farthest

node in  $FN(u)$ . An instance of the established protocol by  $CLNeigh$  is shown in Figure 2.

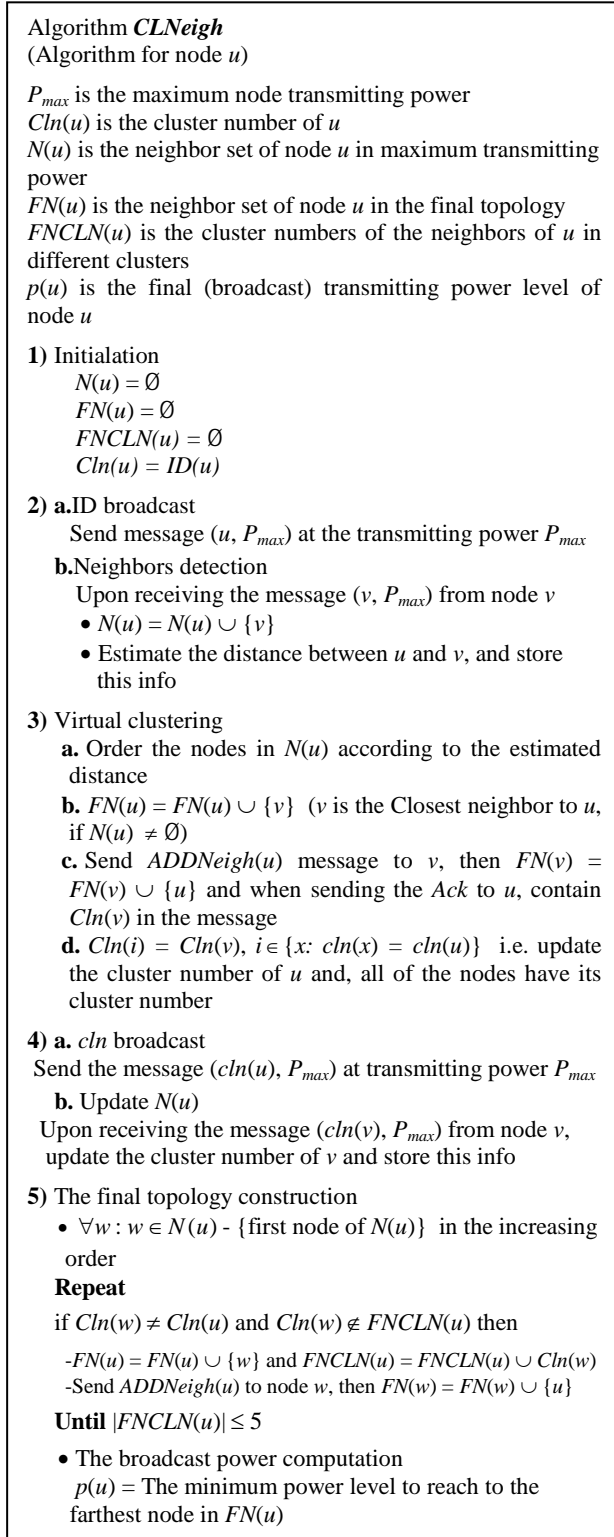


Figure 1.  $CLNeigh$ , the proposed algorithm

### B. Discussion on $CLNeigh$ properties

The idea behind choosing  $six$  neighbors for each node in  $CLNeigh$  is as follows. If the nodes distributed uniformly at random in a certain area, each node (without

having location and direction information of the neighbors) would select the minimum number of neighbors, such that they ensure the connectivity with high probability. Under this criterion and according to the proposition which was discussed at section 2.A, the number  $six$  was selected. Simulation results demonstrate the correctness of this selection as well.

We applied level-based TC to the neighbor-based TC, to reduce the energy consumption in the implemented protocols. Our method is similar to the usual per-packet approach. In the usual per-packet approach, the transmitting power level for each node to reach to the specified neighbor is continuously computed. But according to the level-based approach, we choose the minimum power level such that, it can access to this neighbor. In other words, for each neighbor at the final topology, we keep distinct transmitting power levels according to those power levels available on the wireless medium. Furthermore, for each node among the pre-selected power levels for the neighbors, we store the maximum power level as a (broadcast) transmitting power. We refer to this approach as the modified method.

### C. Simulation

We designed and developed a software tool to generate many topologies by using different protocols. For comparison purpose, the well-known protocols in the literature, which are closer to our proposed protocol, were also implemented. *KNEIGH* (with two phases) [9], *Homogeneous* (no energy control and each node is transmitting at the maximum power level) and  $CLNeigh$  are our implemented algorithms. This application is used to measure the energy consumption of the protocols.

We simulated a system of  $N$  nodes which are uniformly distributed at random in a rectangular region of  $400 \times 400$  meters. All the experiments were done for some values of  $N$  ranged between 50 and 2000. For each  $N$ , we generated randomly 500 nodes replacement, and after running the specified TC protocol for each distribution of the nodes, we selected two nodes among all of them at random. Then the shortest path between these two nodes was calculated, and simultaneously the energy consumption over the path was measured. This manner was repeated 10000 times. After that, the average of the obtained values for each replacement was calculated and the average of the recorded averages in each step was computed and kept. An instance of the constructed topologies by using different protocols, and a routing between two arbitrary nodes on  $CLNeigh$  is shown for  $N = 500$  in Figure 2.

In all the experiments, we calculated the energy consumption in three ways:

- The usual per-packet approach, as the optimal evaluation criterion
- The modified approach
- The original approach of the implemented protocols to compare the amount of the achieved improvement

We noticed that, the transmitting power levels are chosen according to *CISCO Aironet 350* cards.



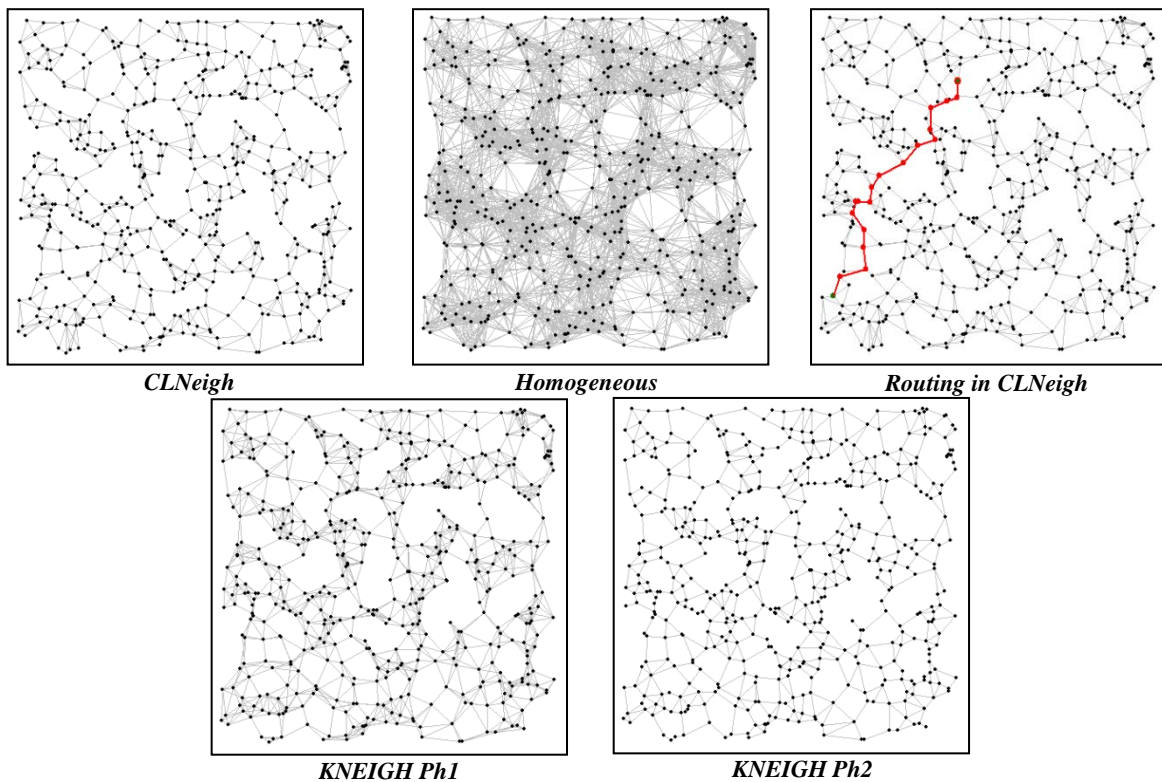


Figure 2. An instance of the constructed topologies using the different protocols for  $N=500$  and an example of a routing in *CLNeigh*

#### D. Analysis of the experimental results

The simulation results for the improvement of the energy consumption at *CLNeigh*, *KNEIGH Ph1* and *KNEIGH Ph2* protocols are represented in Figure 3. The density of the nodes is increased by growing of  $N$ , because the nodes placement environment for all the values of  $N$  is the same and it has been fixed. So, for the smaller values of  $N$ , the nodes connections can be established only by using the longer edges, and accordingly the energy usage will be increased. In the contrary, when the density is being increased, the size of edges will be shorter and the energy consumption will be decreased. Anyway, after some values of  $N$ , the energy consumption will be increased slowly, which is caused by participation of the further nodes in the path. As it is observed from Figure 3, decrement and continuous movement of the optimal approach for each protocol confirms this result.

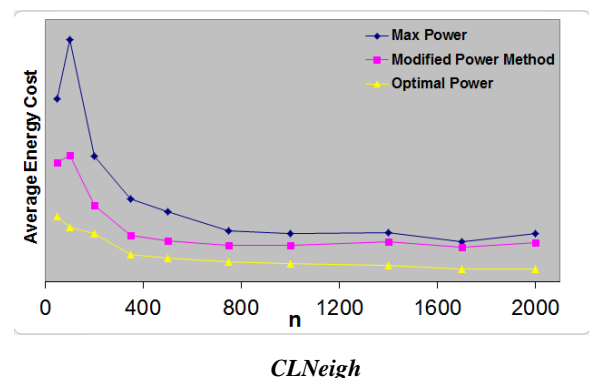
#### V. CONCLUSION AND DISCUSSION

In this paper a novel algorithm for topology control of the ad-hoc networks was proposed. It is called *CLNeigh* and it has the most properties of a good protocol in this field. Because, it relies on the low-quality information, do not need to the additional and the expensive hardware, fully localized and distributed, do not require to access to the whole network information and it generates a topology with small node degree. These properties are exactly desirable to

construct a robust topology in the wireless ad-hoc networks.

*KNEIGH* has an appropriate behavior, but it wastes the power resources. For that, an energy efficient method was also suggested. Using this improvement, it is more suitable for the practical implementation. Since a topology control protocol in a mobile network should be run periodically, minimizing the energy consumption during the protocol execution is more important than the other parameters such as quality of the constructed topology. So we believe that, our method has also significant influence in this situation.

Finally, *CLNeigh* similar to *KNEIGH* has not any guarantee to preserve worst-case connectivity. But according to the simulation results, with choosing an appropriate power level and in case where the nodes distributed uniformly, beside all of the useful properties of this algorithm, the network connectivity will be kept as well.



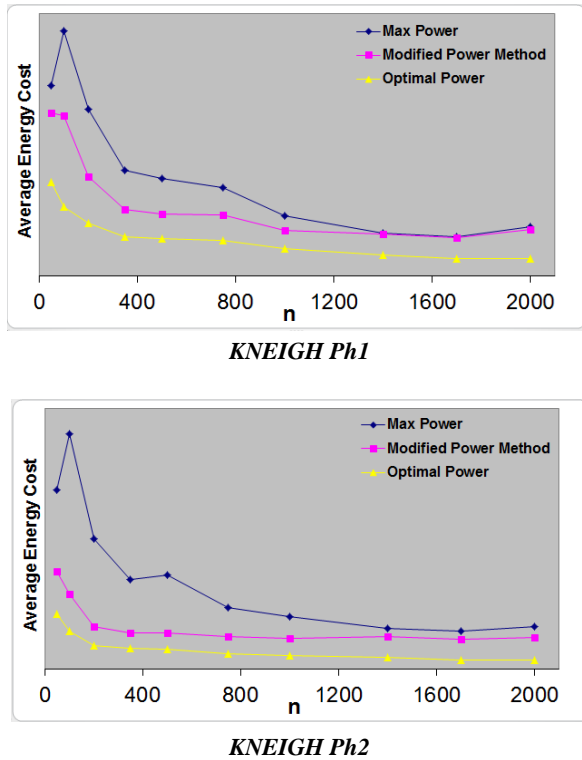


Figure 3. The obtained values of the simulation to compare the energy consumption using three approaches

#### REFERENCES

- [1] J. Wu, and F. Dai, Mobility-Sensitive Topology Control in Mobile Ad Hoc Networks. *IEEE Trans. on Parallel and Distributed Systems*, vol 17, (6), 2006, pp. 522-535.
- [2] J. Yu, H. Roh, W. Lee, S. Pack, and D. Du, Topology Control in Cooperative Wireless Ad-Hoc Networks, *Selected Areas in Communications*, *IEEE Journal on*, vol. 30, no. 9, pp. 1771-1779, October 2012.
- [3] P. Santi, *Topology Control in Wireless Ad Hoc and Sensor Networks*, John Wiley & Sons Ltd, England, 2005.
- [4] X. Xie, and H. Zhang, Topology Algorithm Research Based on Energy and Power Control for TopDisc Algorithm. In *Proceeding of the 2nd International Conference on Computer Modeling and Simulation*, Sanya, China, 2010, pp. 37-40.
- [5] J. A. Bondy, and U. S. R. Murty, *Graph theory with applications*, Elsevier Science Publishing Co., Inc., 1976.
- [6] J. Y. Yu, and P. H. J. Chong, A Survey of Clustering Schemes for Mobile Ad Hoc Networks. *IEEE Communications Surveys and Tutorials*, vol 7(1), 2005, pp. 32-48.
- [7] S. Narayanaswamy, V. Kawadia, and et al., Power Control in Ad Hoc Networks: Theory, Architecture, Algorithm and Implementation of the COMPOW Protocol. In *Proceeding of the European Wireless Conference*, Florence, Italy, 2002, pp. 156-162.
- [8] V. Kawadia, and P. Kumar, Power Control and Clustering in Ad Hoc Networks. In *Proceeding of IEEE Infocom*, San Francisco, USA, 2003, pp. 459-469.
- [9] D. Blough, M. Leoncini, and et al., The k-Neighbors Protocol for Symmetric Topology Control in Ad Hoc Networks. In *Proceeding of ACM MobiHoc*, Annapolis, MD., 2003, pp. 41-152.
- [10] A. Savvides, C. Han, and M. Srivastava, Dynamic fine-grained localization in ad hoc networks of sensors. In *Proceeding of the ACM Mobicom 01*, Rome, Italy, 2001, pp. 166-179.
- [11] L. Girod, and D. Estrin, Robust Range Estimation Using Acoustic and Multimodal Sensing, In *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001)*, Maui, Hawaii, USA, pp. 1-9, October 2001.
- [12] M. Chatterjee, K. S. Das, and et al., WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks. *Cluster computing Journal*, vol 5(2), 2002, pp. 193-204.

**Saeid Taghavi Afshord** received his BSc degree in applied mathematics and MSc degree in computer engineering from the Islamic Azad University, Tabriz and Qazvin branches in 2003 and 2006, Iran respectively. He joined the Islamic Azad University, Shabestar branch, Iran, as a faculty member in 2008. Currently he is a PhD student in computer engineering at the United Institute of Informatics Problems of the NAS of Belarus, from March 2011. His research areas are Energy Saving in Ad Hoc and Sensor Networks, and Methods for Boolean functions Decomposition.

**Bager Zarei** received his BSc degree and MSc degree in computer engineering from the Islamic Azad University, Shabestar and Qazvin branches in 2004 and 2006, Iran respectively. He joined the Islamic Azad University, Shabestar branch, Iran, as a faculty member in 2007. His research areas are Meta-Heuristics and non-deterministic algorithms for solving optimization problems.

**Bahman Arasteh** received his master degree from Islamic Azad University of Arak, Iran, in 2006. He is currently working toward the Ph.D. degree in Islamic Azad University of Iran, Science and Reserch branch. His research interests include software-level fault Tolerance, reliability of programming languages, compilers and distributed applications.