

# A Topology Control Algorithm for Constructing Power Efficient Wireless Ad Hoc Networks

Szu-Chi Wang

Dept of Electrical Engineering  
National Taiwan University  
Taipei, Taiwan  
wsc@lion.ee.ntu.edu.tw

David S. L. Wei

Dept of Computer and Information Science  
Fordham University  
Bronx, New York  
wei@dsm.fordham.edu

Sy-Yen Kuo

Dept of Electrical Engineering  
National Taiwan University  
Taipei, Taiwan  
sykuo@cc.ee.ntu.edu.tw

**Abstract** - In this paper, we present a localized algorithm for constructing power efficient topology for wireless ad hoc networks. Each mobile node determines its own transmission power based only on local information. The proposed algorithm first constructs the constrained Gabriel graph from the given unit disk graph and then reduces the total transmission power by allowing each node individually excises some replaceable links. The constructed topology is sparse, has a constant bounded power stretch factor, and the total transmission power is lower than those obtained from other proposed algorithms. In addition, compared with others, our algorithm requires lower time complexity to generate a solution, and can thus further save the energy for each mobile node. We demonstrate the performance improvements of our algorithm through simulations.

## I. INTRODUCTION

A wireless ad hoc network is formed from mobile nodes without using an infrastructure. Each mobile node is responsible for serving not only as a user but also as a router. The connectivity of the network is maintained through the corporation of all nodes of the network. On the one hand, without a need of a fixed infrastructure allows the network to be easily deployed in areas where a communication infrastructure does not exist (e.g. due to an earthquake or flood) or the construction of a fixed infrastructure is inconvenient or impossible (e.g. on a battlefield or in space). On the other hand, the lack of a physical backbone infrastructure poses a strong need of topology control of the network. It is known that the performance of a protocol for an ad hoc network can be enhanced if the protocol is designed based on overlaying a virtual infrastructure on the ad hoc network. Also, due to the finite power supply of a mobile computer, power conservation has been widely used as a primary control parameter in the design of protocols for wireless ad hoc networks. Therefore, the problem of power-efficient topology control has been attracting more and more researchers from the areas of mobile computing and networking. In this paper, we present an algorithm for constructing power-efficient topology for wireless ad hoc networks. A wireless ad hoc network can be modeled by a

weighted directed graph  $G = (V, E)$ , where  $V$  represents the set of all mobile nodes and  $E$  represents the set of possible interconnections between pairs of mobile nodes. For each edge  $(u, v) \in E$ , node  $v$  must be in the transmission range of node  $u$ . We use  $\|uv\|$  to denote the Euclidean distance between node  $u$  and node  $v$ . The weight of the edge  $(u, v)$  is the power consumed for transmitting signal from node  $u$  to node  $v$ . Its value can be formulated as  $\|uv\|^\beta$  in the most widely-used power-attenuation model, where  $\beta$  is a real constant between 2 and 5 depending on the wireless transmission environment. This power consumption is typically called *transmitter power* and is treated as the major part of power consumption to transmit signals. Like most of the previous works and for the purpose of performance comparisons, we do not count the power consumed at the receiver node in this paper. Throughout the paper, we use the terms link and edge interchangeably.

In this paper, we assume that all the mobile nodes are distributed in a two-dimensional plane and each mobile node has a GPS receiver on board for acquiring its own location information. We assume that initially all mobile nodes are operated at full transmission power and have the transmission radius equal to one unit by a proper scaling. Consequently, the resulted graph  $G$  will be a unit-disk graph (denoted as UDG ( $V$ )) and there is an edge between two nodes if and only if their Euclidean distance is at most one. We assume that UDG ( $V$ ) is strongly connected. All of the mobile nodes have unique identifiers (*ID*) numbered from 1 to  $N$ , where  $N = |V|$ . Each mobile node can individually adjust its own transmission power. We assume that omni-directional antennas are used by all mobile nodes to transmit and receive signals. Wireless ad hoc networks are power constrained, and it is thus undesirable to ask each node to transmit with maximum power. Otherwise, the total transmission power will often be unnecessarily high and the transmission interference will occur more frequently. As a result, each mobile node should adjust its transmission power to potentially reduce the total transmission power while still maintain network connectivity.

Due to the characteristics of wireless ad hoc networks, it is preferred that the underlying network topology can be

constructed in a localized manner. Stojmenovic *et al.* first give the definition of a localized algorithm [1]. A localized algorithm requires the transmission only over small distances. It also dramatically reduces the interference levels among nodes. In our case, a distributed transmission power control algorithm is called localized if every node  $u$  can decide its transmission power based only on the information of the nodes reachable in a constant number of hops. We assume that there is no node failure and the communication channel is reliable in this paper.

Hereafter we adopt several definitions given in [2]. Let  $f$  be a complete transmission power assignment on  $V$  and  $G_f$  be the associated communication graph. Obviously,  $G_f \subseteq G$ . The total transmission power of  $f$  is defined as  $sc(f) = \sum_{v \in V} f(v)$ , where  $f(v)$  is the minimum transmission power needed to reach all of the neighbors of  $v$  in  $G_f$ . Given an  $h$ -hop (unicast) path  $\Pi(u, v)$  from node  $u$  to node  $v$  in  $G_f$ , the path is expressed as  $\Pi(u, v) = v_0 v_1 \dots v_{h-1} v_h$ , where  $u = v_0$ ,  $v = v_h$ . The total transmission power consumption of this path is defined as:

$$p(\Pi(u, v)) = \sum_{i=1}^h \|v_{i-1} v_i\|^\beta.$$

Given a communication graph  $H$ , the least-power path between node  $u$  and node  $v$ , denoted by  $\Pi_{\min}(u, v)$ , is a path whose total transmission power consumption is the minimum among all of the paths that connect these two nodes in  $H$ . Let  $p_H(u, v)$  stand for  $p(\Pi_{\min}(u, v))$ . The power stretch factor of the graph  $G_f$  with respect to  $G$  is then defined as:

$$\rho_{G_f}(G) = \max_{u, v \in V} (p_{G_f}(u, v) / p_G(u, v)).$$

The two most widely used energy conservation approaches in literature are: (1) reducing the transmission power of each node; and (2) reducing the total power consumed by all nodes involved in one communication session. The later can be achieved by preserving the least-power paths in the underlying network. However, these two approaches may offset each other, and a discussion of it can be found in [3]. The major concern of our work is to develop a localized topology control algorithm where each mobile node makes a decision about its transmission power based on only its local information. These locally made decisions collectively ensure global network connectivity and the network topology controlled by the transmission power of each node must be power efficient. More precisely, the proposed algorithm must achieve the following objectives: (1) a complete transmission power assignment with low total transmission power, (2) a constant bounded power stretch factor, and (3) a sparse network, namely a constant bounded average node degree.

The rest of this paper is organized as follows. Section II briefly introduces the related works. Section III describes the basic ideas and the details of our algorithms. Performance evaluation is performed via simulations in Section V. The extension of the proposed algorithm to mobile network is discussed in Section V. We conclude the paper in Section VI.

## II. PREVIOUS WORKS

In [4] Rodoplu and Meng described a distributed protocol for constructing a topology that guarantees the least-power path between every pair of nodes that are connected in a given (UDG) graph. Recently, Li *et al.* [5] proposed a protocol based on their results but performs better and is computationally simpler. In [6] Li and Wan proposed a distributed position-based protocol to construct an enclosure graph for conserving power in one communication session. Their protocol is more time and space efficient than [4]. All of these works focus on constructing a subgraph of  $G$  that contains the *minimum-power topology*, namely, a subgraph that includes the union of all of the least-power paths. On the other hand, the problems of finding a complete transmission power assignment with optimization criteria of the maximum/total transmission power assigned has been studied in [7-9]. However, the proposed transmission power assignment approaches in [8, 9] with objective of minimizing the total transmission power are all centralized and cannot be transformed to localized algorithms. The trade-off between sparseness and power efficiency of the topology is discussed in [2]. In [2] Li *et al.* studied the power efficiency property of several well-known proximity graphs, such as the constrained Gabriel graph (denoted by GG( $G$ )), the constrained relative neighborhood graph (denoted by RNG( $G$ )), and the constrained Yao graph (denoted by YG $_k$ ( $G$ )), over a graph  $G$ . They also showed that if  $G = \text{UDG}(V)$  the power stretch factor of GG( $G$ ) is always one, and the power stretch factors of RNG( $G$ ) and YG $_k$ ( $G$ ) could be as large as  $N-1$  and  $1/(1-2\sin(k/\pi))$ , respectively. However, they also showed that the total transmission power of these geometry structures could be arbitrarily larger than the minimum total power needed to maintain the strong connectivity of the network.

## III. OUR LOCALIZED TOPOLOGY CONTROL ALGORITHM

### A. Observations

For simplicity, we first assume that each node in the network is stationary. However, our algorithm can be easily modified to adapt to the mobile environment as discussed in Section V. Considering a complete transmission power assignment  $f$ , there is obviously a tradeoff between  $sc(f)$  and  $\rho_{G_f}(G)$ , *i.e.* over reducing the transmission power of each node may diminish least-power paths or vice versa. The problem of finding a complete transmission power assignment  $f$  whose total transmission power is the minimum among all of the complete assignments is called the min-total assignment problem. However, no localized algorithm for the min-total assignment problem has been given. Moreover, this problem is NP-hard and it is still an open question for the best approximation ratio. Therefore, our algorithm is designed in such a way that we first construct a subgraph  $H = (V, E')$  from  $G$ , where  $E' \subseteq E$ , and  $H$  has a constant bounded power stretch factor. After that, the total transmission power is minimized as much as possible. The basic idea is using the local information of each node to excise some logical links of the subgraph while still keeps the power stretch factor being

bounded by a constant  $c$ . Unlike the algorithms for constructing power efficient topologies proposed in [2], we do not aim to construct a topology that guarantees a constant bounded node degree (though our algorithm may generate a such one) in this paper. The main reason is that, as shown in [2], a geometry structure with a constant bounded node degree may contain very limited least-power paths between any two nodes. Our design goal is to preserve as many of these least-power paths as possible. Moreover, as mentioned in [3], further eliminating edges may result in more congestion and hence worse throughput and fault tolerance in the long run.

### B. The Algorithm

Our algorithm consists of two phases. In the first phase we construct the constrained Gabriel graph  $GG(G)$  over the unit disk graph  $G = UDG(V)$ . We choose the constrained Gabriel graph simply because it can be constructed locally and efficiently and has a power stretch factor of one. In addition, the constrained Gabriel graph was used as a subgraph in several important routing protocols, for example, the FACE routing protocol [10] and the GPSR routing protocol [11]. Nevertheless, any other geometry structure that guarantees a constant bounded power stretch factor can be used in the first phase. Suppose that  $GG(G)$  is associated with a transmission power assignment  $f$ . For each node  $u$ , if a logical link  $(u, v)$  of  $GG(G)$  satisfies the equation  $\|uv\|^\beta = f(u)$ , then  $(u, v)$  is called a *critical link*. The transmission radius associated with  $f(u)$  is denoted by  $radius(u)$ . For each node  $u$ , the amount of transmission power saved by excising all its critical link(s) is denoted by  $ps(u)$ . The priority of node  $u$  is a pair  $pri(u) = \langle ps(u), ID \rangle$ . Let  $pri(v_1) = (ps(v_1), ID_1)$  and  $pri(v_2) = (ps(v_2), ID_2)$ . Then,  $pri(v_1) > pri(v_2)$  if  $ps(v_1) > ps(v_2)$  or  $(ps(v_1) = ps(v_2) \text{ and } ID_1 < ID_2)$ . Each node is only required to know the locations, priorities and the transmission radiuses of the nodes within its two-hop neighborhood of  $UDG(V)$ . The information about the one-hop neighbors can be obtained by using some form of beacon messages that are sent periodically and asynchronously by each node to declare its presence. To obtain the information of two-hop neighbors, a common solution is that each node attaches the list of its own one-hop neighbors, while sending the beacon messages. These two kinds of information are extensively used to facilitate message routing and broadcasting in wireless ad hoc networks and therefore, the cost of maintaining such information can be amortized. In fact, with the knowledge of node position, Calinescu [12] recently proposed a connected-dominating-set based approach for collecting the two-hop neighborhood information with  $O(N)$  communications.

In the second phase, we reduce the total transmission power by trying to eliminate all the critical link(s) that are replaceable with alternative paths. For each critical link  $(u, v)$ , node  $u$  tries to search another path that reaches node  $v$  based on its local information. We call such path the *replacing path* of  $(u, v)$ . Whenever a node  $u$  found that itself has the highest priority and  $ps(u) > 0$  within its two-hop neighborhood, it

starts a search for the replacing path(s). The searching process is based on Dijkstra's algorithm. However, for each node  $u$ , no global knowledge of the network topology is available. Our solution is to ask  $u$  to construct a local topology view, namely a subgraph  $G'(u) = (V', E')$ , on top of  $GG(G)$  based on the location and transmission radius information of the nodes in its vicinity. A node  $v_i$  belongs to  $V'$  if  $v_i$  is within  $u$ 's two-hop neighborhood of  $UDG(V)$ . An edge  $(v_i, v_j)$  belongs to  $E'$  if  $\|v_i v_j\| \leq radius(v_i)$  and  $(v_i, v_j)$  belongs to the edges of  $GG(G)$ . The edge weight of  $G'$  is defined the same as that of  $GG(G)$ . After the construction of  $G'(u)$ , node  $u$  removes each critical link and applies Dijkstra's algorithm on it to search for the shortest path  $\Pi_{min}(u, v)$ . Node  $u$  saves the result by an *ID* list including all of the intermediate nodes of  $\Pi_{min}(u, v)$ . If no such path exists or  $p(\Pi_{min}(u, v))$  is more than  $c$  times larger than the weight of edge  $(u, v)$  in  $G$ , then the searching procedure is ended. The replacing path of node  $u$  is set as an empty list and  $ps(u)$  is set to 0. Otherwise,  $\Pi_{min}(u, v)$  is the eligible replacing path of  $(u, v)$ , and  $(u, v)$  can be excised. For example, assume that Fig. 1(a) is the local topology view constructed by  $p_1$  based on its local information. The solid lines are the logical links of  $GG(G)$  and are all bi-directional. The area covered by the dashed line is the transmission range of  $p_1$ , which is decided by the directed link  $(p_1, p_5)$ . Note that there should be no other node in the gray area. If  $p_1$  found an alternative path  $\Pi(p_1, p_5) = p_1 p_2 p_3 p_4 p_5$  and  $p(\Pi(p_1, p_5)) < c \cdot \|p_1 p_5\|^\beta$ , where  $c$  is a predetermined constant, then  $p_1$  can excise  $(p_1, p_5)$ . This way, the transmission range (and thus the transmission power) of  $p_1$  can be largely reduced, as shown in Fig. 1(b).

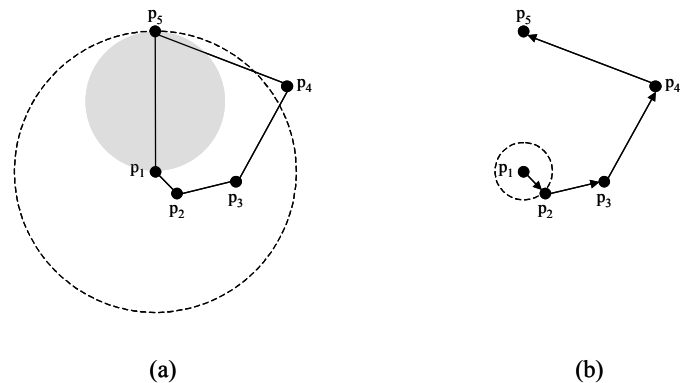


Figure 1. Illustrative example.

If all the critical link(s) can be excised, the priority of  $u$  is recalculated by the same rule, with the critical link(s) decided by a new value of  $f(u)$  that corresponds to the minimum operational power needed to cover the remaining logical links. Node  $u$  then disseminates its new priority and replacing path(s) to its two-hop neighborhood by a notification message. Each constituent link in the replacing path(s) is called as a *replacing link*. The start node of each replacing link adds a *replacing tag* to this link. For each node  $w$  with  $ps$

$(w) > 0$ , if  $w$  receives a notification message, it first checks if any of its critical link(s) is tagged as replacing. If so,  $w$  sets  $ps(w)$  to 0 and disseminates its new priority to its two-hop neighborhood. Otherwise, if node  $w$  found that its priority is the local maximum within its two-hop neighborhood, it starts to search the replacing path(s) of its critical link(s) following the above steps. The idea of using two-hop neighborhood information can be generalized by using  $k$ -hop neighborhood information, where the value of  $k$  can be chosen depending on the value of  $c$ . The algorithm in pseudo code format for each node, say node  $u$ , is given in Fig. 2.  $L$  is the logical links decided by  $u$ .  $LTV(k)$  is the local topology view constructed using the  $k$ -hop neighborhood information of  $u$ . After applying Dijkstra's algorithm on  $LTV(k)$ , the weights of all shortest paths are stored in an array  $dist[N]$  ordered by the  $IDs$  of the destination nodes. Note that if a node  $v$  is unreachable from  $u$  in  $LTV(k)$ , then  $dist[v] = \infty$ . The notification message is a tuple  $nm = \langle id, pri, radius, replacing\_path \rangle$  that records the  $ID$ , priority, transmission radius, and replacing path(s) of the sender node, respectively.

```

For each node  $u$ :
 $L \leftarrow \{ (u, v) \mid (u, v) \in GG(G) \}$ ;
 $iteration \leftarrow 0$ ;
decide  $pri$ ;
send a beacon message to the  $k$ -hop neighborhood;
on receipt of a notification message  $nm$  do
  update the information of node  $nm.id$ ;
  for each link  $(u, v) \in L$  do
    if  $((u, v) \in nm.replacing\_path)$  then  $(u, v).tag \leftarrow replacing$ ;
on update of local information do
  if  $(pri.ps > 0$  and  $pri$  is the max priority in the  $k$ -hop neighborhood and  $iteration < i$ ) then
     $shrink \leftarrow true$ ;
    construct  $LTV(k)$ ;
    apply Dijkstra's algorithm on  $LTV(k)$ ;
    for each critical link  $(u, v) \in L$  do
      if  $((u, v).tag = replacing$  or  $dist[v] > c * ||uv||^{\beta})$  then
         $shrink \leftarrow false$ ;
         $pri.ps \leftarrow 0$ ;
        break;
    if  $(shrink)$  then
       $L \leftarrow L - \{ (u, v) \mid (u, v) \text{ is a critical link} \}$ ;
      update  $pri.ps$ ;
   $iteration \leftarrow iteration + 1$ ;
  send a notification message to the  $k$ -hop neighborhood;

```

Figure 2. Algorithm EGG running at each node.

It's not hard to see that there is a tradeoff while searching for the relaying paths: if some critical link(s) of the nodes with lower priorities are chosen for composing the replacing paths of the nodes with higher priorities, the transmission power of those low-priority nodes cannot be reduced. However, if we exclude all the critical link(s) while applying Dijkstra's algorithm, the possibility of failing to find an eligible replacing path will increase. We use a simple heuristic for further decreasing the total transmission power: applying Dijkstra's algorithm on a subgraph of  $LTV(k)$  with all of the critical link(s) removed. If not all of the eligible replacing path(s) can be found, then Dijkstra's algorithm is applied again on  $LTV(k)$ . The remaining steps are the same as above.

### C. Correctness

It is easy to see that the constructed topology has the following properties: (1) strongly connected; (2) sparse; (3) having a power stretch factor bounded by  $c$ ; (4) planar. Due to the limited space, we only offer a sketch of the proof. First, since our algorithm just excises the logical links of  $GG(G)$ , the constructed topology is a subgraph of  $GG(G)$  and is thus a sparse and planar graph. Second, the strong connectivity is guaranteed since the excision occurs only when all of the eligible replacing path(s) of the critical link(s) can be found. Third, the power stretch factor is bounded by  $c$  since the power stretch factor of  $GG(G)$  is one and each constituent link  $l$  of the least-power path can only be replaced by a path whose total transmission power consumption is at most  $c$  times larger than  $l$ 's link weight. Each link comprised in the replacing path is marked by a replacing tag, and will not be excised in the future.

### D. Complexity

The cost of Dijkstra's algorithm is  $O(V \log V + E)$  provided that the priority queue is implemented in a Fibonacci heap. Since each node uses a subgraph of  $GG$  as the input to Dijkstra's algorithm, and  $GG(G)$  is sparse and can be constructed by an  $O(V \log V)$  time algorithm, the time complexity for each node to search all of the replacing path(s) is thus  $O(V \log V)$ . If the number of times of executing the searching procedure for each node is bounded by a constant  $i$ , the total time complexity of each node is  $O(V \log V)$ . Therefore, our algorithm outperforms those proposed in [4, 5] in terms of time complexity. Time complexity should be also a major concern in designing a localized power-efficient topology control algorithm, as the algorithm has to be frequently executed by each individual node in order to maintain the connectivity of the network in a power-efficient manner. The computations spent in running of the algorithm certainly consume power from the node.

## IV. SIMULATION RESULTS

In this section, we measure the performance of our algorithm via simulations. As mentioned in Section I, we were most interested in the power stretch factor and total transmission power of the topologies constructed by the localized algorithms with similar design goals. We chose the algorithm proposed by Li and Halpern [5] (denoted by SMECN) since it performs significantly better than the algorithm proposed by Rodoplu and Meng [4] in terms of total transmission power. Likewise, the constrained Gabriel graph (denoted by GG) outperforms all other topologies described in [2] in terms of the above two criteria. It is thus chosen for comparison here. Our algorithm was operated with  $c = 2$  and  $i = 1$ . We experimented it with the two strategies described in Section III, *i.e.* applying Dijkstra's algorithm once (denoted by EGG-1) and twice (denoted by EGG-2). We measured the power efficiency of each constructed topology  $H$  by the following metrics: (1) total transmission power

(denoted by  $tpc$ ), and (2) power stretch mean (denoted by  $psm$ ), which is defined as:

$$psm = \left( \sum_{u,v \in V} \frac{p_H(u,v)}{P_G(u,v)} \right) / N^2.$$

$psm$  is a modification of the power stretch factor since all the above topologies clearly have a power stretch factor bounded by a very small constant. This metric is more suitable to further evaluate the energy conservation for a communication session. The power attenuation constant  $\beta$  was set to 2 and 4. For simplicity, we simulated only a static network, that is, we assume that nodes do not move. The transmitter range  $R$  is fixed at 500 meters. The map sizes are equal to  $s \times R$  by  $s \times R$ , for  $s = 1, 3, 5$ . The  $x$  and  $y$  coordinates of each node are selected at random in the interval  $[0, m]$ , where  $m$  is the map size. The experimentation was performed for  $N = 100$ . The experimental results are summarized in Table I and Table II. Each value in the two tables is the average of 50 independent experimental results.  $tpc$  is normalized to lie in between 0 and 1.

Table I shows that the topology constructed by our algorithm has a  $tpc$  of about 30% less than that of GG and SMECN in most cases. One the other hand, the increase in its  $psm$  is less than 5%. This means that the total transmission power can be decreased significantly while conserving most of the least-power paths. This trend is more obvious when  $\beta = 4$ . As shown in Table II, if we compare EGG-1/EGG-2 with GG, the  $tpc$  of GG is much larger than that of EGG-1/EGG-2, but their difference in  $psm$  is almost negligible. The  $tpc$  of SMECM is also higher than that of EGG-1/EGG-2 in this case. Notice that the time complexity of SMECM is  $O(V^2)$ , but ours is only  $O(V \log V)$ . One can also see that by applying Dijkstra's algorithm twice indeed can further reduce  $tpc$  at a cost of a slight increase in  $psm$ . The experimental results indicate that our algorithm can prolong both the node and the network lifetime and thus improve the energy efficiency in wireless ad hoc networks.

## V. DEALING WITH MOBILITY

The above discussions are all performed on the stationary networks, that is, every node is assumed to be static. While taking node movement into consideration, each node must be able to adjust its transmission power dynamically. The basic idea for achieving network reconfiguration is that all the mobile nodes send beacon messages with full transmission power and run the proposed algorithm periodically. However, it is too costly if many nodes move at high speed. As mentioned in many previous works (e.g., [6]), node movement can be viewed as two events, namely node addition and node deletion. Therefore, what we should solve turns out to be finding an efficient way to add a new node to and remove a node from the network. Our algorithm uses the  $k$ -hop neighborhood information in a way that network reconfiguration can be solved more efficiently. For node addition, the new node  $v$  broadcasts its position information with full transmission power. Each node  $u$  that received this

TABLE I. The performance measurements of different topologies with  $\beta = 2$

	$s = 1$		$s = 3$		$s = 5$	
	$tpc$	$psm$	$tpc$	$psm$	$tpc$	$psm$
GG	1.0	1.0	1.0	1.0	1.0	1.0
SMECN	0.99992	1.0	0.99995	1.0	0.99994	1.0
EGG-1, $k=2$	0.666025	1.02792	0.669387	1.02764	0.724976	1.03409
EGG-2, $k=2$	0.641633	1.03955	0.644607	1.03934	0.708392	1.04131
EGG-1, $k=3$	0.666025	1.02792	0.669387	1.02764	0.724745	1.03411
EGG-2, $k=3$	0.641633	1.03955	0.644534	1.03953	0.707691	1.04135

TABLE II. The performance measurements of different topologies with  $\beta = 4$

	$s = 1$		$s = 3$		$s = 5$	
	$tpc$	$psm$	$tpc$	$psm$	$tpc$	$psm$
GG	1.0	1.0	1.0	1.0	1.0	1.0
SMECN	0.49970	1.0	0.50571	1.0	0.65060	1.0
EGG-1, $k=2$	0.405776	1.00945	0.408402	1.00963	0.53725	1.01206
EGG-2, $k=2$	0.393086	1.02337	0.39657	1.02172	0.523476	1.02237
EGG-1, $k=3$	0.405776	1.00945	0.408072	1.00962	0.519015	1.01158
EGG-2, $k=3$	0.393086	1.02337	0.395919	1.02174	0.503721	1.02264

it is too costly if many nodes move at high speed. As mentioned in many previous works (e.g., [6]), node movement information checks if  $(u, v)$  is a new logical link of GG ( $G$ ). If so,  $u$  updates its logical links and informs  $v$  to add link  $(v, u)$ . For node deletion, if a removed node  $v$  changes the transmission radius of some node  $u$ , then  $u$  updates the critical link(s). If some node found that its  $ps$  value becomes larger than 0, then it updates its priority and re-executes the searching procedure if there is a need. In addition, though the moving distance of a moving node  $v$  comprised in the relaying path could affect the power stretch factor of the generated topology, it is not hard to see that node  $v$  only affects the logical links of the nodes within its  $k$ -hop neighborhood. If some node  $w$  is aware of that the movement of node  $v$  will break the replacing path of node  $u$ , then node  $w$  informs node  $u$  to re-execute the searching procedure and disseminates its new replacing path if there is one. Obviously, the communications and computations of the reconfiguration process can be performed locally, which is preferred for wireless ad hoc networks.

## VI. CONCLUSIONS

In this paper, we show how to construct and maintain a power-efficient topology in wireless ad hoc networks in a distributed and localized manner. We develop a distributed algorithm that requires only local information for constructing and maintaining a logical topology on the unit disk graph. The proposed algorithm first constructs the constrained Gabriel graph from the given unit disk graph. After that, the total transmission power can be reduced via searching for the replacing path of each critical link. The topology constructed by our algorithm has several desired features such as sparse and constant bounded power stretch factor. The simulation results show that our algorithm outperforms others in terms of total transmission power. In addition, our algorithm requires lower time complexity and

can thus further save the energy for each mobile node, as computations also consume power from a mobile computer. Furthermore, the topology constructed by our algorithm helps other programs save energy in two ways, as stated in [5]. First, it suffices for each node to use the transmission power computed by our algorithm to send periodic beacon messages. Second, the routing algorithm can be restricted to using the edges of this subgraph to find an energy-efficient path for a communication session.

#### REFERENCES

- [1] I. Stojmenovic and X. Lin, "Power-aware Localized Routing in Wireless Networks," *IEEE International Parallel and Distributed Processing Symposium*, 2000.
- [2] X.-Y. Li, P.-J. Wan, Y. Wang, and O. Frieder, "Sparse Power Efficient Topology for Wireless Networks," *HICSS*, Hawaii, 2002.
- [3] R. Wattenhofer, L. Li, P. Bahl, and Y. M. Wang, "Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks," *IEEE INFOCOM 2001*, pp. 1388-1397, April 2001.
- [4] V. Rodoplu and T. H. Meng, "Minimum Energy Mobile Wireless Networks," *IEEE J. Selected Areas in Communications*, 17(8):1333-1344, August 1999.
- [5] L. Li and J. Halpern, "Minimum Energy Mobile Wireless Networks Revisited," *IEEE International Conference on Communications (ICC)*, June 2001.
- [6] X.-Y. Li and P.-J. Wan, "Constructing Minimum Energy Mobile Wireless Networks," *ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Long Beach, California, Oct 4-5, 2001.
- [7] R. Ramanathan and R. Rosales-Hain, "Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment," *IEEE INFOCOM*, 2000.
- [8] L. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc, "Power Consumption in Packet Radio Networks," *Symposium on Theoretical Aspects of Computer Science (STACS '97)*, 1997.
- [9] G. Calinescu, I. Mandoiu, and A. Zelikovsky, "Symmetric Connectivity with Minimum Power Consumption in Radio Networks," *IFIP-TCS*, 2002.
- [10] I. Stojmenovic and S. Datta, "Power and Cost Aware Localized Routing with Guaranteed Delivery in Wireless Networks," *Proc. Seventh IEEE Symposium on Computers and Communications (ISCC)*, 2002.
- [11] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, 2000.
- [12] G. Calinescu, "Computing 2-hop Neighborhoods in Ad Hoc Wireless Networks," *2<sup>nd</sup> International Conference on AD-HOC Networks and Wireless (ADHOC-NOW '03)*, October 2003.