SPT-BASED POWER-EFFICIENT TOPOLOGY CONTROL FOR WIRELESS AD HOC NETWORKS

Szu-Chi Wang
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan

and

David S. L. Wei
Department of Computer and Information Sciences
Fordham University
Bronx, New York

and

Sy-Yen Kuo
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan

ABSTRACT

This paper presents a localized Shortest-Path-Tree (SPT) based algorithm to cope with the topology control problem in wireless ad hoc networks. Each mobile node determines its own transmission power based only on its local information. The proposed algorithm first constructs local SPTs from the initial graph, and then the total power consumption is further reduced by allowing each mobile node to search and excise the replaceable links individually. The constructed topology ensures network connectivity, and possesses several desirable energy-efficient features: 1) its power stretch factor is bounded and can be predetermined, 2) the power consumption is evenly distributed among the mobile nodes, and 3) its total power consumption is lower than that obtained from the best known algorithms. The performance improvements of the proposed algorithm are demonstrated through extensive simulations.

I. INTRODUCTION

An ad-hoc network is an infrastructureless wireless network proposed as an alternative to cellular networks for use in areas where the existing 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). In such a network, each node is responsible for serving not only as a user but also as a router. A wireless link between any two nodes can be established if the radio transmission range of each node can cover each other. In case of insufficient radio transmission range between the two nodes, multiple "hops" may be required, whereby intermediate nodes re-broadcast the messages until the destination node is reached. The reliance on wireless multi-hop communications to maintain connectivity among nodes adds new complexity on the design and operation of the network. The lack of a physical backbone infrastructure poses a strong need of topology control of the network. It has been shown 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 node, power conservation has been widely used as a primary control parameter in the design of the 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 wireless networking.

In this paper, we propose an algorithm for constructing energy-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 interconnections between 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), denoted by $w(u, v)$, can be formulated as $t \cdot \|uv\|^{\alpha} + rp(u, v)$ in the most widely-used power-attenuation model, where t is a threshold related to the signal-to-noise ratio at node u, and α is a constant between 2 and 5 depending on the wireless transmission environment. The former part of the equation is typically called *transmitter power* and is the power consumed for transmitting signal from node u to node v. The remaining part is the power consumed at the receiver and is called *receiver power*. We assume that all receivers have the same power threshold for signal detection, and the value of t is thus an appropriate constant. The sum of transmitter power and receiver power is called *transmission power* in the rest of this paper. Throughout the paper, we use the terms link and edge interchangeably.

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. To begin with, all mobile nodes are operated at full transmission power and have the transmission radius equal to one unit by a proper scaling. Consequently, the resulting graph G can be modeled as a *unit-disc 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 also assume that omni-directional antennas are used by all of the mobile nodes to transmit and receive signals. Wireless ad hoc networks are power constrained, and it is thus undesirable to ask each mobile node to always transmit with maximum power. Otherwise, the total power consumption will often be unnecessarily high and the transmission interference will occur more frequently. In fact, it has been shown that mobile nodes expend most of their power in communications [1]. As a result, each mobile node should adjust its transmission power to reduce its power consumption while still maintain network connectivity. Due to the infrastructureless nature of the ad hoc networks, to avoid flooding the network, it is preferred that the network topology can be constructed in a *localized* manner. Stojmenovic et al. first give the definition of a localized algorithm [2]. A distributed power control algorithm is called localized if each node can decide its transmission power based only on the information of the nodes reachable in a small constant number of hops.

Hereafter we adopt several definitions given in [3]. Let f be a complete transmission power assignment on V, and G_f be the associated communication graph. Clearly, $G_f \subseteq G$. The total power consumption of f (denoted by $tpc(f)$) is defined as $\Sigma_{u\in V} f(u)$, where $f(u)$ is the minimum transmission power needed to reach all the neighbors of u in G_f. Given a path $\Pi(u, v)$ from node u to node v in G_f, it can be expressed as $\Pi(u, v) = v_0v_1\ldots v_{h-1}v_h$, where $u = v_0$, $v = v_h$. The path length of $\Pi(u, v)$ (denoted by $|\Pi(u, v)|$) is h. The transmission power of this path is defined as

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

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

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

The two most widely used energy conservation approaches in the 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 latter can be achieved by preserving the minimum-energy paths of the given UDG (V). However, these two approaches may offset each other, and a discussion of it can be found in [4]. 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 only on its local information. These locally made decisions collectively ensure global network connectivity and the network topology controlled by the transmission power of each mobile node must be energy-efficient. More precisely, the proposed algorithm should achieve the following objectives: (1) a complete transmission power assignment with low total power consumption; (2) a constant bounded power stretch factor.

The rest of this paper is organized as follows. Section II briefly introduces the related works. Section III describes the ideas and properties of our algorithm. In Section IV, the superiority of our algorithm is demonstrated via simulations by comparing the energy-efficiency of the constructed topologies of ours with others in terms of several important metrics. We also propose an efficient way to deal with the node mobility in Section V. Finally conclusions are drawn in Section VI.

II. RELATED WORKS

In [5] Rodoplu and Meng described a distributed protocol for constructing a topology that guarantees preserving the minimum-energy path between every pair of nodes that are connected in the original graph G. The concept of *relay region* is first introduced in their paper. Recently, Li and Halpern [6] proposed a protocol based on results in [5] but performs better and is computationally simpler. In [7] 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 efficient in time and space than [5]. All of these works focus on constructing a subgraph of G that includes the union of all of the minimum-energy paths. Also, the problem of finding a complete transmission power assignment using some optimization criteria has been studied in [8, 9, 10, 17]. However, the proposed approaches of transmission power assignment with objective of minimizing the total power consumption in [9, 10, 17] are all centralized and cannot be transformed to localized algorithms. The trade-off between sparseness and energy efficiency of the topology has been discussed in [3]. In [3] Li et al. studied the energy 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 (directed) graph G. They showed that the total power consumption of these geometric structures could be arbitrarily larger than the minimum total power needed to maintain the strong connectivity of the network.

Moreover, most of their works (e.g. [3, 11]) assumed that receiver power is zero, which is not practical according to [12].

Also, in the literature, the concept of visible neighborhood, namely the topology view of each node based on its local information, has been introduced (see, e.g. [13, 14, 16]). Ning Li et al. [13] developed a topology control algorithm in which each node builds its local minimum spanning tree using the one-hop neighborhood information. The network topology derived by their algorithm has the following properties: (1) network connectivity is preserved, (2) node degree is bounded, and (3) all uni-directional links can be removed. It has been validated by simulations that their algorithm has advantages over [5] and [18] in terms of some metrics. In [16], Li proposed a localized algorithm to construct a subgraph of G whose total edge length is bounded by a constant multiplying by that of the minimum spanning tree. Unfortunately, it has been shown that the total power consumption of the constructed subgraph can be $O(|V|^{\alpha})$ larger than the optimal. Each algorithm in [13], [18] and [16] cannot guarantee a constant bounded power stretch factor. The property of the (constant) bounded power stretch factor is desired for the applications sensitive to worst-case behavior of the topology control.

III. Our Topology Control Algorithm

A. Observations

For simplicity, like those previous works, we first assume that each node in the network is stationary. We then adapt our algorithm to the mobile environments in Section V. Considering a complete transmission power assignment f, there is a tradeoff between $tpc(f)$ and $psf_{G_f}(G)$, i.e., over reducing the transmission power of each individual node may diminish minimum-energy paths or vice versa. The problem of finding a complete transmission power assignment f whose total power consumption is the minimum among all of the complete transmission power assignments is usually called the *min-total assignment problem*. A similar problem is to find a solution that contains bi-directional links only. However, to the best of our knowledge, no localized algorithm for the above two problem has been given. Moreover, both problems are NP-Hard [8, 9, 17] and it is still an open question for the best approximation ratio in either case.

Therefore, our algorithm is designed heuristically as follows. First, we construct an edge subgraph G' from G; G' has a power stretch factor of one. The logical link set of node u in G' is denoted as $LL(u)$. We assume that each link is attached a *tag* to describe its attribute in $LL(u)$. Initially $LL(u)$ contains all the edges of node u in UDG (V) and each link is tagged as *regular*. Then, we try to minimize the total power consumption as much as possible. The basic idea is to let each node excise some logical links of the subgraph while still keeps the power stretch factor being bounded by a constant cb. Unlike the algorithms proposed in [3], we do not aim to construct a topology that guarantees a constant bounded node degree (though our algorithm may generate such one) in this paper. The main reason is that, as shown in [3], a geometry structure with a constant bounded node degree may contain very limited minimum-energy paths between any two nodes. Our design goal is to preserve as many of these minimum-energy paths as possible. Moreover, as mentioned in [4], further eliminating edges may result in more congestion and hence worsen network throughput and fault tolerance in the long run. For each node u, since no global knowledge of the network topology is available, each operation can only use the information from the nodes in its vicinity. Our solution is to let u construct a local topology view $LTV(u, k)$ based on the location and logical link information of the nodes within its k-hop neighborhood

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 its own one-hop neighborhood information while sending the beacon messages. These two kinds of information have been extensively used to facilitate message routing and broadcasting in wireless ad hoc networks, and therefore, the cost of maintaining such information can be amortized. Similar idea can be generalized to gather k-hop, $k > 2$, neighborhood information. Note that in practice k should be small compared to network diameter D. The definition of $LTV(u, k)$ is given as follows.

Definition 1 (*Local Topology View*): The local topology view of node u, denoted by $LTV(u, k) = (V', E')$, is a subgraph of G such that (1) a node $v_i \in V'$ if the hop distance between v_i and u is no more than k; (2) an edge $(v_i, v_j) \in E'$ if $(v_i, v_j) \in LL(v_i)$ and both v_i and v_j belong to V'. The tag of each edge in E' is also recorded.

Suppose that a subgraph of G is associated with a transmission power assignment f. For each node u, if a logical link (u, v) satisfies the equation $w(u, v) = f(u)$, then (u, v) is called a *critical link* of node u. Assume that all critical links of u are excised. If $LL(u) \neq \varnothing$, we define $ps(u)$ as $(f(u) - f'(u)) / e(u)$, where $f'(u)$ is the transmission power needed to maintain the remaining logical link(s) of node u, and $e(u)$ is the number of critical link(s) of node u; otherwise, $ps(u) = 0$. The priority of node u is a pair $pri(u) = \langle ps(u), ID(u) \rangle$. Let $pri(v_1) = (ps(v_1), ID_1)$ and $pr(v_2) = (ps(v_2), ID_2)$. Then, $pri(v_1) > pri(v_2)$ if $ps(v_1) > ps(v_2)$, or $ps(v_1) = ps(v_2)$ and $ID_1 < ID_2$.

B. Two-Phase Localized Algorithm

The proposed algorithm is composed of two phases, namely *local shortest tree construction* and *path search replacement*. These two phases are loosely coupled, i.e., the techniques introduced in each phase can be individually used in the design of any other topology control algorithm with proper modifications.

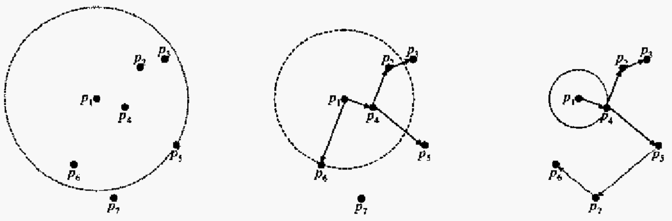
Phase I) Local Shortest Tree Construction: The information needed in this phase by each node is the *ID*s and the locations of its one-hop neighbors, which can be easily gathered since we assume that initially each node sends a beacon message with its maximum transmission power. More precisely, such information is included in $LTV(u, 1)$, and the weight of each edge can thus be derived. Note that the edge weight includes both transmitter power and receiver power between the two end nodes of the edge. Each node u applies Dijkstra's algorithm independently to get the shortest-paths from the source node u to the other nodes in $LTV(u, 1)$. As a result, the local shortest path tree of node u, denoted by $LSPT(u)$, can be obtained. The direct children of node u, $DC(u)$, is defined as $DC(u) = \{v \in V' \mid h(LSPT(u), v) = 1\}$, where $h(LSPT(u), v)$ is the height of a child node v in $LSPT(u)$. Node u then removes the logical link set $\{(u, w) \mid w \notin DC(u)\}$ from $LL(u)$. The topology generated under the above descriptions is denoted as G^{I}.

Since for each node u, only the one-hop neighborhood information is available for constructing $LSPT(u)$, some links in G^{I} may be uni-directional. However, uni-directional links are unfavorable in wireless ad hoc networks. In [13, 15] it has been shown that network topologies free of uni-directional links are much more beneficial to MAC layer control mechanisms, link-level acknowledgments and package transmissions/re-transmissions. Our solution to remove uni-directional links is simple: since at the end of phase I, all nodes are aware of the logical links of its one-hop neighbors, each node deletes its uni-directional links. The resulting topology is denoted as G^{II}. We will prove that G^{II} not only preserves network connectivity but also preserves all minimum-energy paths of G in Section III.D.

Phase II) Path search replacement: At the beginning of this phase, each node is assumed to perceive the logical link(s) of its one-hop neighbors. As a result, for each node u, $LTV(u, k)$ can be acquired by sending k–1 more beacon messages. Node u then can further reduce its transmission power by trying to eliminate the critical link(s) that are replaceable with alternative paths. That is, for each critical link (u, v), node u tries to search another path that reaches node v based on $LTV(u, k)$. We call such path the *replacing path* of (u, v). The entire replacing path(s) of node u is denoted as $RP(u)$. Whenever a node u finds that it has the highest priority within its k-hop neighborhood and $ps(u) > 0$, it starts a search for $RP(u)$. The searching process is also based on Dijkstra's algorithm. After obtaining $LTV(u, k)$, node u applies Dijkstra's algorithm on it to search for the shortest path $\Pi_{k\text{-min}}(u, v)$. Note that all of the original critical links of each node should be excluded during the searching process, which can be achieved by setting their edge weights to ∞ temporally. Assume that the weights of all shortest paths originated from u are stored in an array $dist[N]$ ordered by the *ID*s of the destination nodes. Note that if a node v is unreachable from u in $LTV(u, k)$, then $dist[ID(v)] = \infty$. Node u saves the result by an *ID* list including all of the intermediate nodes of $\Pi_{k\text{-min}}(u, v)$. Let cb be a predetermined constant. If no such path exists or $p(\Pi_{k\text{-min}}(u, v))$ is more than cb times larger than the weight of edge (u, v), then the search process is ended, $RP(u)$ is set as an empty list, and $ps(u)$ is set to 0. Otherwise, $\Pi_{k\text{-min}}(u, v)$ is the eligible replacing path of (u, v), and (u, v) can be excised. An example is illustrated in Fig. 1. The gray line in Fig. 1(a) is the transmission range of p_1 with full transmission power. After Phase I, the area covered by the dashed line in Fig. 1(b) is the new transmission range of p_1, which is decided by $LSPT(p_1)$. After Phase II, if p_1 found an alternative path $\Pi(p_1, p_6) = p_1p_4p_5p_7p_6$ in $LTV(p_1, 2)$ and $p(\Pi(p_1, p_6)) < cb \cdot w(p_1, p_6)$, then p_1 can excise (p_1, p_6). This way, the transmission range (and thus the transmission power) of p_1 can be largely reduced, as shown in Fig. 1(c).

If all critical link(s) of u can be excised, $ps(u)$ is set to 0 and $f'(u)$ becomes the minimum operational power needed to cover the remaining logical link(s). That is, each node executes the search process at most once in Phase II. Node u then disseminates its replacing path(s) to its k-hop neighborhood by a notification message. The notification message is a tuple $nm = \langle id, LL, RP \rangle$ that records the *ID*, logical links and replacing path(s) of the sender node, respectively. Each constituent link in the replacing path(s) is called as a *replacing link*. For each node u, if it is notified that its link (u, v) is a replacing link, the *tag* of (u, v) is switched to *replacing*. 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 k-hop neighborhood. Otherwise, w updates $LTV(w, k)$ according to nm. If w

found that its priority is the local maximum within its k-hop neighborhood, it starts to search the replacing path(s) of its critical link(s) following the above steps. Finally, after Phase II, the transmission range of each node u can be decided to be the power level it needs to reach its remaining logical links in the constructed topology. The constructed topology after Phase II is denoted as G^{III}.



(a) (b) (c)

Figure 1. Illustrative example.

C. Heuristics to Improve the Performance

It's not hard to see that there is a tradeoff while searching for the relaying path(s), e.g. if some critical links 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 node u excludes all the critical links in $LTV(u, k)$ while applying Dijkstra's algorithm, the possibility of failing to find an eligible replacing path will increase. We use a simple heuristic to further decrease the total power consumption: for each link (v_i, v_j) in $LTV(u, k)$, if $ps(v_i) = 0$ or (v_i, v_j) is not a critical link of node v_i, then we modify the weight of (v_i, v_j) by multiplying it by a *dilation factor* $df(v_i)$, which is defined as $1 + (ps(v_i) / ps(u))$ before applying Dijkstra's algorithm. If not all of the eligible replacing path(s) can be found, Dijkstra's algorithm is applied again on the original $LTV(u, k)$. The remaining steps are the same as above. We can also extend cb to be a real number, e.g. 1.5, instead of using an integer constant such that more flexibility can be introduced while Phase II is applied. In fact, the experiment results proved that using $1 < cb < 2$ can further improve the power stretch factor at the cost of only a slight increase in total power consumption.

D. Properties of the Constructed Topology

We prove some critical properties of the topology generated by our algorithm below. For clarification, we use $w^G(u, v)$ to denote the weight of edge (u, v) in G.

Lemma 1: The minimum-energy path between any two nodes in G is preserved in G^I.

Proof: For every node pair u and v, if $(u, v) \in E$ and is deleted after the construction of $LSPT(u)$, $\exists$ a path $\Pi(u, v)$ in the initial $LTV(u, 1)$ such that $p(\Pi(u, v)) < w^G(u, v)$. Assume a deleted link (u, v) exists in the minimum-energy path Π_m of G. Since $LTV(u, 1)$ is a subset of G, $\Pi(u, v)$ should be included in G. If we replace (u, v) with Π in Π_m, then the transmission power of the new path will be smaller than Π_m, and a contradiction is derived. □

Lemma 2: The minimum-energy path between the two end nodes of each deleted link in G^I is preserved in G^{II}.

Proof: We sort all the links that are deleted in the uni-directional link removal procedure by the following rank formula: $(u_1, v_1) > (u_2, v_2)$ if $w^G(u_1, v_1) > w^G(u_2, v_2)$, or $w^G(u_1, v_1) = w^G(u_2, v_2)$ and $ID(u_1) > ID(u_2)$, or $w^G(u_1, v_1) = w^G(u_2, v_2)$ and $ID(u_1) = ID(u_2)$ and $ID(v_1) > ID(v_2)$. Since the order of link deletion does not change the resulting subgraph, without loss of generality, we assume that the links are deleted according to their ranks in an ascending order. We prove by induction that G^{II} preserves the minimum-energy path of each deleted link.

Basis: $k = 1$, Consider the first deleted link (u^1, v^1), clearly for node v^1 there is a path $\Pi(v^1, u^1)$ in $LSPT(v^1)$ such that $p(\Pi(v^1, u^1)) < w^G(v^1, u^1)$. According to our energy model $w^G(v^1, u^1) = w^G(u^1, v^1)$, and each link (u^h, v^h) that comprise $\Pi(v^1, u^1)$ must satisfy $w^G(u^h, v^h) < w^G(u^1, v^1)$. Therefore, each link that belongs to $\Pi(v^1, u^1)$ is bi-directional and is preserved in G^{II}.

Induction: Assume Lemma 2 holds for all deleted links $(u^i, v^i) \mid i = 1, 2, \ldots, k-1$. Now we prove that Lemma 2 also holds for edge (u^k, v^k). Consider link (u^k, v^k), clearly for node v^k there is a path $\Pi(v^k, u^k)$ in $LSPT(v^k)$ that $p(\Pi(v^k, u^k)) < w^G(u^k, v^k)$. Similarly, each edge (u^h, v^h) that comprise $\Pi(v^k, u^k)$ must satisfy $w^G(u^h, v^h) < w^G(u^k, v^k)$, and there are two cases:

Case 1: (u^h, v^h) is bi-directional, as a result, (u^h, v^h) is not deleted.

Case 2: (u^h, v^h) is uni-directional, since $w^G(u^h, v^h) < w^G(u^k, v^k)$, the minimum-energy path between (u^h, v^h) is reserved, namely, there is a path $\Pi(u^h, v^h)$ that is composed of bi-directional links and has the minimum total power consumption. It implies that the deletion of (u^k, v^k) does not affect the minimum-energy path between node u^k and node v^k. □

Lemma 3: G^{II} preserves the network connectivity of G.

Proof: Follows directly from Lemma 1 and Lemma 2, since the minimum-energy path between any two nodes is preserved in G^{II}. □

Lemma 4: $psf_{G^{III}}(G)$ is bounded by cb.

Proof: Clearly, the power stretch factor of G^{II} is one. Consider each constituent link (u, v) of the minimum -energy path, (u, v) can only be substituted by a replacing path $\Pi(u, v)$ such that $p(\Pi(u, v)) \le cb \cdot w^G(u, v)$. Since the end nodes of each replacing link is in the k-hop neighborhood of node u and will not begin their search process until u finishes its Phase II. Moreover, each replacing link of u is marked by a replacing tag and will not be excised in the future. As a result, the power stretch factor of G^{III} is at most cb times larger than G^{II}. □

Theorem 1: G^{III} preserves the network connectivity of G and has a constant bounded power stretch factor cb.

Proof: Follows directly from Lemmas 1, 2, 3, and 4. □

IV. Performance Comparisons

Via simulations, we compared the performance of our algorithms with that of others in terms of total power consumption, power stretch factor, and the node degree of the constructed topologies. The experimental results are summarized in Table I – Table IV. Our algorithm is denoted as ESPT. For the sake of study, the first phase alone of our algorithm is denoted as $ESPT^1$, and the one that consists of both Phase I and Phase II is denoted as $ESPT^2$. The unit disk graph (denoted as UDG) is chosen as a basis for comparison. We chose the algorithm proposed by Li and Halpern [6] (denoted by SMECN) since it performs significantly better than the one proposed by Rodoplu et al. [5] in terms of total power consumption, and has a power stretch factor of one.

Likewise, the constrained Gabriel graph (denoted by GG) outperforms those described in [3] in terms of both total power consumption and power stretch factor. It is thus chosen for comparison here. The algorithm proposed by Li [16] (denoted as AMST) is based on the constrained relative neighborhood graph. However, Li's paper offers only theoretical analysis without the experimental performance, and is of interest to us. The MST-based topology control algorithm proposed in [13] (denoted as LMST) is also chosen to compare because its total power consumption is lower than that of SMECN (though its power stretch factor is larger). Our algorithm operated with $\alpha = 4$, $k = 2$, and $cb = 1.5$ and 2.0, respectively. So as to have fair comparison, in SMECN each node was assumed to broadcast its neighbor discovery message (NDM) with the maximum operational power while gathering necessary local information. As for LMST, to balance the trade-off between *tpc* and *psm*, we did not incorporate the removal of uni-directional links.

We observe the following metrics of each constructed topology H: (1) total power consumption associated with H (denoted by *tpc*), (2) power stretch mean (denoted by *psm*), which is defined as $psm = (\sum_{u,v \in V} \frac{p_H(u,v)}{p_G(u,v)}) / N^2$,[1] (3) the maximum power stretch factor (denoted by *max psf*) observed throughout the experiment,[2] (4) the variance of transmission power (denoted by *var tp*),[3] (5) average node degree (denoted by *avg nd*), and (6) the maximum node degree observed throughout the experiment (denoted by *max nd*). The last two metrics are chosen mainly for a better understanding of topology characteristics. The values of *tpc*, *psm*, *var tp* and *avg nd* are the average of 50 independent experimental results. Be advised that *tpc* is normalized to lie in between 0.0 and 1.0 by dividing its values by the total power consumption of UDG. The transmitter range R is fixed at 500 meters. The map sizes are equal to $s \times R$ by $s \times R$, for $s = 1, 3, 5, 7$; s directly relates to network density. The x and y coordinates of each node are selected at random in the interval $[0, m]$, where m is the map size. The experiment was performed for $N = 100$.

From Table I – IV we observe that the topology constructed by our algorithm ($ESPT^1$/$ESPT^2$) has a *tpc* much less than that of GG and SMECN in all cases, and the increase in the *psm* of $ESPT^2$ is almost negligible, say less than 0.04. This means that the total power consumption can be decreased significantly while preserving most of the minimum-energy paths. As for the comparison with LMST, first we consider denser networks (see Table I – II). The *tpc* values of our algorithm are slightly more than those of LMST. However, since the differences are relatively very small with respect to UDG, we believe that our algorithm has advantages while considering other energy-efficiency metrics, such as *psm* and *max psf*. Note that since LMST does not take the power stretch factor into consideration, the subgraph generated by LMST might have very large *max psf* (about 9 in this case), while our algorithm guarantees a bounded *max psf* (1.5 and 2.0 in this case). For sparser networks (see Table III – IV), $ESPT^2$ can outperform LMST in both *tpc* and *psm*. Albeit the *max psf* of LMST is decreasing when s is increasing in our experiments, LMST does not provide a theoretical upper bound on the *max psf*. Both $ESPT^1$ and $ESPT^2$ have smaller *tpc* compared to AMST in all cases. Note that although the topology constructed by AMST tends to have a small *psm* (but still larger than that of ours), its power stretch factor can be as large as N-1 [16]. Clearly,

1. *psm* is a modification of power stretch factor since most of 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.

2. *max psf* shapes the worst-case behavior of applications that operate over the constructed topology.

3. Due to the uniform power threshold in our model, we set $t = 1$ when counting transmission power for convenience.

if the receiver power is a non-zero constant, the power stretch factor of LMST and AMST becomes even larger.

Finally, it can also be observed that our algorithm performs well in *var tp*, especially in the case of ESPT^2 with $s > 1$. Since, overall, our algorithm has edge on others in both summation and variance of transmission power, our algorithm can provide a better opportunity of avoiding node power depletion. The above experimental results also indicate that, if the source and destination nodes are randomly chosen, our algorithm effectively curtails both total power consumption and the variance of transmission power. In addition, with its small *psm* and constrained *max psf*, we believe that the topology constructed by our algorithm can provide a base for power efficient communication operations, and thus can prolong the node and network lifetime.

V. Dealing with Mobility

The above discussions are all performed under the assumption that every node is static. While taking node mobility into consideration, each node must be able to adjust its transmission power dynamically. An intuitive idea to achieve network reconfiguration is that all of the mobile nodes send beacon messages with full transmission power and run the proposed algorithm periodically. In this paper, we consider the case of modest movement of the nodes. With high mobility, basically a smaller k should be adopted in Phase II and the interval to send beacon messages should be very short. In fact, it will be extremely difficult for a topology control algorithm to even effectively guarantee network connectivity if the topology changes too fast.

On the other hand, the above-suggested idea may be too costly in case of low mobility. As mentioned in previous works (e.g. [7]), 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. We assume that a moving node, say node v, broadcasts a node deletion notification before its movement and broadcasts its new position information to notify node addition at the end of each beacon interval, both with full transmission power. For node addition, if node u received the new node information from node v, it re-decides *LSPT* (u) and checks if (u, v) is a logical link of this new *LSPT* (u). If yes, node u re-executes the search process and updates its logical link(s); node u informs node v to add link (v, u) if necessary. For node deletion, consideration should be taken only if (u, v) is a logical link of node u. In this case, node u re-decides *LSPT* (u) and re-executes the search process; node u also checks if (u, v) is also a replacing link of some node; if node u is aware of that the movement of node v will break the replacing path of some node w, then node u informs node w to re-execute the search process and disseminates its new replacing path if there is one. It is not hard to see that node v only affects the logical link(s) of the nodes within its k-hop neighborhood.

Table I The Performance Measurements with $s = 1$

	$s = 1$					
	tpc	*psm*	*max psf*	*var tp*	*avg nd*	*max nd*
UDG	1.0	1.0	1.0	0.0	96.5312	99
GG	0.000666	1.0	1.0	4.68E+15	3.5992	8
SMECN	0.000328	1.0	1.0	1.20E+15	2.6996	6
AMST	0.000253	1.01491	2.51846	8.76E+14	2.3386	5
LMST	0.000099	1.1967	9.70118	9.30E+13	1.98	4
ESPT^1	0.000178	1.0	1.0	2.43E+14	2.4252	5
ESPT^2, *cb*=2.0	0.000123	1.03482	1.99171	1.27E+14	2.1356	5
ESPT^2, *cb*=1.5	0.000137	1.016	1.49991	1.52E+14	2.2036	4

Table II The Performance Measurements with $s = 3$

	$s = 3$					
	tpc	*psm*	*max psf*	*var tp*	*avg nd*	*max nd*
UDG	1.0	1.0	1.0	0.0	25.5408	53
GG	0.05339	1.0	1.0	2.82E+19	3.59	8
SMECN	0.026874	1.0	1.0	8.10E+18	2.7016	6
AMST	0.020503	1.01413	2.46733	5.71E+18	2.3388	4
LMST	0.010074	1.08511	6.31301	2.01E+18	2.0258	4
ESPT^1	0.01495	1.0	1.0	2.60E+18	2.4284	5
ESPT^2, *cb*=2.0	0.010102	1.03485	1.99906	8.56E+17	2.1372	5
ESPT^2, *cb*=1.5	0.011188	1.01516	1.49855	9.95E+17	2.2044	4

Table III The Performance Measurements with $s = 5$

	$s = 5$					
	tpc	*psm*	*max psf*	*var tp*	*avg nd*	*max nd*
UDG	1.0	1.0	1.0	0.0	10.4232	27
GG	0.245597	1.0	1.0	2.29E+20	3.4496	8
SMECN	0.159539	1.0	1.0	1.47E+20	2.6596	5
AMST	0.128633	1.01305	2.49099	1.15E+20	2.3204	4
LMST	0.10403	1.03698	4.38695	9.96E+19	2.135	4
ESPT^1	0.122629	1.0	1.0	1.08E+20	2.45	5
ESPT^2, *cb*=2.0	0.086227	1.03671	1.9971	6.51E+19	2.1564	5
ESPT^2, *cb*=1.5	0.09287	1.01372	1.49999	6.87E+19	2.2236	5

Table IV The Performance Measurements with $s = 7$

	$s = 7$					
	tpc	*psm*	*max psf*	*var tp*	*avg nd*	*max nd*
UDG	1.0	1.0	1.0	0.0	5.6392	14
GG	0.380877	1.0	1.0	3.46E+20	2.9548	8
SMECN	0.299838	1.0	1.0	3.01E+20	2.44	6
AMST	0.261474	1.01752	2.8558	2.66E+20	2.1856	4
LMST	0.254784	1.0228	3.14811	2.71E+20	2.1376	4
ESPT^1	0.27593	1.0	1.0	2.86E+20	2.3468	5
ESPT^2, *cb*=2.0	0.228194	1.03382	1.99931	2.32E+20	2.1096	5
ESPT^2, *cb*=1.5	0.237019	1.0135	1.4938	2.40E+20	2.158	5

Through our simulations, we observed that, in most cases, the average number of nodes affected by removing of a relaying link is close to one, that is, the links that comprised the replacing paths were sporadic and evenly distributed. Thus, at the beginning of each beacon interval, node u checks if there is a change in transmission radius after deciding the new logical links. If yes, node u updates its critical link(s). If node u found that its ps value becomes larger than 0, then it disseminates its new priority and re-executes the search process. No recalculation is needed for node u if there is no node addition in its one-hop neighborhood and no node deletion in its replacing path. The communications and computations of the reconfiguration process can thus be performed locally, which is preferred for wireless ad hoc networks.

VI. Concluding Remarks

In this paper, we show how to construct and maintain an energy-efficient topology in wireless ad hoc networks in a distributed and localized manner. Our algorithm requires only local information for constructing and maintaining a topology on the given unit disk graph. The concept of k-redundant edges has been proposed by Li and Halpern [6]. However, the algorithm in [6] deals with 2-redundant edges only. The contributions of our algorithm are multi-fold. First, the approaches in Phase I figure out k-redundant edges for $k \geq 2$. That is, we can get a minimum-power topology with less total power consumption by a new approach that is totally different from those given in [5, 6]. Our algorithm is also simpler in computation than theirs. Second, the topology constructed after Phase II has several desirable features such as bounded power stretch factor, low total power consumption and small variance of transmission power. The simulation results show that our algorithm outperforms others in terms of various important metrics. Third, our algorithm provides the flexibility of pre-determining the power stretch factor of the derived subgraph. In summary, the proposed algorithm effectively and efficiently constructs a virtual backbone that can support energy-aware communication applications over wireless ad hoc networks.

References

[1] L. M. Feeney, M. Nilsson, "Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment," *IEEE INFOCOM 2001*, pp. 1548–1557, 2001.

[2] I. Stojmenovic and X. Lin, "Power-aware Localized Routing in Wireless Networks," *IEEE International Parallel and Distributed Processing Symposium*, 2000.

[3] X.-Y. Li, P.-J. Wan, Y. Wang, and O. Frieder, "Sparse Power Efficient Topology for Wireless Networks," *HICSS*, Hawaii, January 2002.

[4] 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.

[5] V. Rodoplu and T. H. Meng, "Minimum Energy Mobile Wireless Networks," *IEEE Journals on Selected Areas in Communications*, 17(8):1333–1344, August 1999.

[6] L. Li and J. Halpern, "Minimum Energy Mobile Wireless Networks Revised," *IEEE International Conference on Communications* (*ICC 2001*), June 2001.

[7] 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, October 2001.

[8] A. E. F. Clementi, P. Penna, and R. Silvestri, "On the Power Assignment Problem in Radio Networks," *Electronic Colloquium on Computational Complexity* (*ECCC*), 2000.

[9] L. Kirousis, E. Kranakis, D. krizanc, and A. Pelc, "Power Consumption in Packet Radio Networks," *Symposium on Theoretical Aspects of Computer Science* (*STACS*), 1997.

[10] G. Calinescu, I. Mandoiu, and A. Zelikovsky, "Symmetric Connectivity with Minimum Power Consumption in Radio Networks," *IFIP-TCS*, 2002.

[11] X.-Y. Li, P.-J. Wan, W. Yu, "Power Efficient and Sparse Spanner for Wireless Ad Hoc Networks," *IEEE International Conference on Computer Communications and Networks* (*ICCCN01*), Scottsdale, Arizona, October 2001.

[12] J. A. Stine, G. de Veciana, "A Comprehensive Energy Conservation Solution for Mobile Ad Hoc Networks," *IEEE International Conference on Communications* (*ICC 2002*), New York, April 2002.

[13] N. Li, J. Hou, and L. Sha, "Design and Analysis of an MST-Based Distributed Topology Control Algorithm," *IEEE INFOCOM 2003*, June 2003.

[14] S. C. Wang, D. S. L. Wei, S. Y. Kuo, "A Topology Control Algorithm for Constructing Power Efficient Wireless Ad Hoc Networks," *IEEE GLOBECOM 2003*, December 2003.

[15] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar, "Power Control in Ad-Hoc Networks: Theory, Architecture, Algorithm and Implementation of the COMPOW Protocol," *Proc. of European Wireless2002, Next Generation Wireless Networks: Technologies, Protocols, Services and Applications*, pp. 156–162, Florence, Italy, February 2002.

[16] X.-Y. Li, "Approximate MST for UDG Locally," *The Ninth International Computing and Combinatorics Conference* (*COCOON 2003*), July 25–28, 2003, Big Sky, MT, USA.

[17] X. Cheng, B. Narahari, R. Simha, M. X. Cheng, and D. Liu, "Strong Minimum Energy Topology in Wireless Sensor Networks: NP-Completeness and Heuristics," *IEEE Transactions on Mobile Computing*, 2(3):248–256, July-September 2003.

[18] L. Li, J.Y. Halpern, V. Bahl, Y.M. Wang and R. Wattenhofer, "Analysis of a Cone-Based Distributed Topology Control Algorithms for Wireless Multi-hop Networks", *Proc. ACM Symposium on Principle of Distributed Computing (PODC)*, 2001.