Available online at www.sciencedirect.com

ScienceDirect

Computer Communications 29 (2006) 3092–3103

computer communications

www.elsevier.com/locate/comcom

An SPT-based topology control algorithm for wireless ad hoc networks

Szu-Chi Wang [a], David S.L. Wei [b], Sy-Yen Kuo [a,*]

[a] *Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan*
[b] *Department of Computer and Information Sciences, Fordham University, Bronx, NY, USA*

Abstract

In this paper, we present a localized Shortest-Path-Tree (SPT) based algorithm that copes 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, after which the total power consumption is further reduced by allowing each mobile node to search the replaceable links individually. The constructed topology ensures network connectivity, and possesses the following desirable energy-efficient features: (i) the power stretch factor is bounded and can be predetermined, (ii) the power consumption is evenly distributed among the mobile nodes, and (iii) the total power consumption is lower than that obtained by the best known algorithms. The performance improvements of the proposed algorithm are demonstrated through extensive simulations. We conclude our work with a discussion of future research directions toward more integrated mobile network architectures.

Keywords: Ad hoc networks; Energy efficiency; Localized algorithm; Topology control

---

1. Introduction

Wireless ad hoc networks have been proposed as an alternative to cellular networks for use in areas where the existing communication infrastructure has been disrupted or destroyed (e.g., due to an earthquake or flood), or where construction of a fixed infrastructure would be inconvenient or impossible (e.g., on a battlefield or in space). In such networks each node must serve as a user and as a router. A wireless link between any two nodes can be established if the radio transmission range of each node can cover the other. In the 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 to the design and operation of the network. Furthermore, the lack of a physical backbone poses a strong need for 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 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 protocols for wireless ad hoc networks. Therefore, the problem of power-efficient topology control has attracted increasing attention from researchers in the area of wireless networking.

In this paper, we propose an algorithm for constructing an energy-efficient topology for wireless ad hoc networks. Such networks can be modeled by a weight directed graph $G=(V,E)$, where V represents the set of all mobile nodes and E represents the set of interconnections between the 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 nodes u and 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

---

[*] Corresponding author.
E-mail address: sykuo@cc.ee.ntu.edu.tw (S.-Y. Kuo).

0140-3664/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2005.11.006

at node u, and α is a constant between 2 and 5 depending on the wireless transmission environment. The first part of the equation is typically called the *transmitter power*, which is the power consumed by transmitting a signal from node u to node v. The remaining part is the power consumed by the receiver and is called the *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. Hereafter, the sum of the transmitter power and the receiver power is called the *transmission power*. Also, throughout the paper, we use the terms link and edge interchangeably.

We assume that all mobile nodes are distributed on a two-dimensional plane and each mobile node has a GPS receiver that acquires information about the node's location. 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 resulting graph G can be modeled as a unit-disc graph (denoted as $\mathrm{UDG}(V)$) and there is an edge between two nodes if and only if their Euclidean distance is at most one. We assume that $\mathrm{UDG}(V)$ is strongly connected. All 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 the mobile nodes to transmit and receive signals. Wireless ad hoc networks are power constrained; thus, it is undesirable to require each mobile node to always transmit at maximum power. Otherwise, the total power consumption would often be unnecessarily high and transmission interference would 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 radius to reduce its power consumption, while still maintaining network connectivity. Due to the infrastructureless nature of ad hoc networks, it is preferable that the network topology should be constructed in a localized manner to avoid flooding the network. Stojmenovic and Lin gave the definition of a localized algorithm in [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.

In this paper, 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 $\sum_{u\in V} f(u)$, where $f(u)$ is the minimum transmission power needed to reach all the neighbors of u in G_f. A given path, $\prod(u,v)$, from node u to node v in G_f can be expressed as $\prod(u,v) = v_0v_1 \cdots v_{h-1}v_h$, where $u = v_0$, $v = v_h$. The length of the path $\prod(u,v)$ (denoted by $|\prod(u,v)|$) is h, and its transmission power is defined as

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

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

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

In the literature, the two most widely used energy conservation approaches are: (i) reducing the transmission power of each node; and (ii) 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 $\mathrm{UDG}(V)$. However, these two approaches may offset each other; a relevant discussion can be found in [4]. The major focus of our work is to develop a localized topology control algorithm in which each mobile node makes a decision about its transmission power based solely on its local information. As these locally made decisions collectively ensure global network connectivity, the network topology controlled by each node's transmission power must be energy-efficient. More specifically, the proposed algorithm must achieve the following objectives: (i) it must be capable of adjusting the transmission radii of the nodes with low total power consumption; and (ii) it must have a constant bounded power stretch factor. Fig. 1 illustrates that the strongly connected property could be preserved under the reduction of transmission radii and the excision of links. Moreover, we consider several practical issues, particularly the integration of ad hoc networks into IP-enabled services. IPv6 is designed to satisfy various future networking requirements and a more advanced support of mobility can be achieved accordingly. Therefore, we examine relevant studies and identify some practical challenges that will be important in next-generation networks.

The remainder of this paper is organized as follows. Section 2 briefly introduces related works. Section 3 describes the basic concepts and properties of our algorithm. In Section 4, the superiority of our algorithm is demonstrated via simulations. Also, using several important metrics, we compare the energy-efficiency of our topology with that of other algorithms. In Section 5, we propose an efficient distribution method to deal with node mobility. Finally, in Section 6, we present our conclusions and indicate some future research directions.

2. Related works

In [5], Rodoplu and Meng described a distributed protocol for constructing a topology that guarantees the preservation of the minimum-energy path between every pair of nodes connected on the original graph G. The concept of relay region was first introduced in their paper. Recently, based on the results in [5], Li and Halpern [6] proposed a

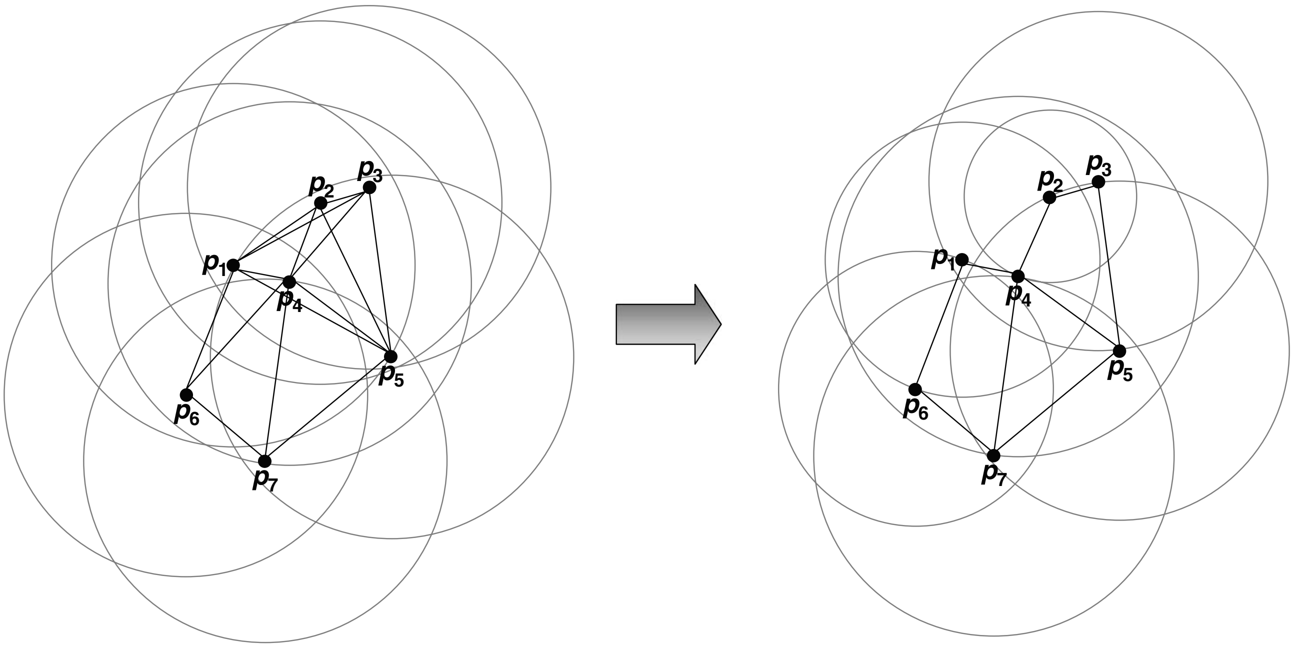

Fig. 1. Topology control via the reduction of transmission radii and the excision of links.

protocol that performs better and is computationally simpler. In [7], Li and Wan presented a distributed position-based protocol to construct an enclosure graph that conserves power in one communication session. Their protocol is more efficient in time and space than that in [5]. All the above works focus on constructing a subgraph of G that includes the union of all of the minimum-energy paths. In addition, the problem of finding a complete transmission power assignment using some optimization criteria has been studied in [8–10,17]. Although the objective of the approaches in [9,10,17] is to minimize total power consumption, they are all centralized and cannot be transformed into localized algorithms. The trade-off between sparseness and energy efficiency in a topology has been extensively investigated. In particular, Li et al. [3] discussed the energy efficiency property of several well-known proximity graphs, such as the constrained Gabriel graph (denoted by $\mathrm{GG}(G)$), the constrained relative neighborhood graph (denoted by $\mathrm{RNG}(G)$), and the constrained Yao graph (denoted by $\mathrm{YG}_k(G)$), over a (directed) graph G. They showed that the total power consumption of these geometric structures can be arbitrarily larger than the minimum total power needed to maintain the strong connectivity of the network. In some of their works [3,11], however, they assume that the receiver power is zero, which is not practical according to [12].

The concept of a visible neighborhood, i.e., the topological view of each node based on its local information, was introduced in [13,14,16] for example. Li et al. [13] developed a topology control algorithm in which each node builds its local minimum spanning tree using one-hop neighborhood information. The network topology derived by their algorithm has the following properties: (i) network connectivity is preserved, (ii) the node degree is bounded, and (iii) all uni-directional links can be removed. It has been validated by simulations that their algorithm has advantages over [5,21] 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 multiplied by that of the minimum spanning tree. Unfortunately, it has also been shown that the total power consumption of the constructed subgraph can be $O(|V|^{\alpha})$ larger than the optimal. The algorithms in [13,21,16] cannot guarantee a constant bounded power stretch factor, which is desirable in applications sensitive to the worst-case behavior of topology control.

3. Our topology control algorithm

3.1. Observations

For simplicity, like other works, we initially assume that each node in a network is stationary. We then adapt our algorithm to the mobile environments described in Section 5. 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 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 problems has been developed. Moreover, both problems are NP-hard [8,9,17] and the best approximation ratio in either case is still an open question.

Therefore, our algorithm is designed heuristically as follows. First, we construct an edge subgraph G' from G; where G' has a power stretch factor of 1. The logical link set of node u in G' is denoted as $LL(u)$. We assume that a tag is attached to each link 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*. We then 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 keeping the power stretch factor bounded by a constant cb. Unlike the algorithms proposed in [3], we do not try to construct a topology that guarantees a constant bounded node degree, though our algorithm may generate one. The reason is that, as shown in [3], a geometrical structure with a constant bounded node degree may contain very few minimum-energy paths between any two nodes, and our design goal is to preserve the maximum number of minimum-energy paths possible. Moreover, as mentioned in [4], further eliminating edges may result in more congestion and hence degrade 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.

Information about one-hop neighbors can be obtained by using some form of beacon messages that are sent periodically and asynchronously by each node to announce its presence. To obtain information about two-hop neighbors, a common solution is to allow each node to attach its own one-hop neighborhood information while sending the beacon messages. These two kinds of information are used extensively to facilitate message routing and broadcasting in wireless ad hoc networks; therefore, the cost of maintaining such information can be amortized. A similar concept can be generalized to gather k-hop ($k > 2$) neighborhood information. Note that, in practice, k should be small compared to the network diameter D. The definition of $LTV(u,k)$ is 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: (i) a node $v_i \in V'$ if the hop distance between v_i and u is no more than k; and (ii) an edge $(v_i, v_j) \in E'$ if $(v_i, v_j) \in LL$ 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 $LL(u) \neq \emptyset$, 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 $pri(v_2) = (ps(v_2), ID_2)$. Then, $pri(v_1) > pri(v_2)$ if $ps(v_1) > ps(v_2)$, or $ps(v_1) = ps(v_2)$ and $ID_1 < ID_2$.

3.2. Two-phase localized algorithm

The proposed algorithm is comprised 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 used individually in the design of any other topology control algorithm with proper modifications.

3.2.1. Phase (I) local shortest tree construction

In this phase, each node needs to know the *ID*s and the locations of its one-hop neighbors. This information can be gathered easily, since we assume that each node initially sends a beacon message with its maximum transmission power. More precisely, such information is included in $LTV(u,1)$, so the weight of each edge can thus be derived. Note that the edge weight includes both the transmitter power and the 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, $DC(u)$, are defined as $DC(u) = \{v \in V' | h(LSPT(u), v) = 1\}$, where $h(LSPT(u), v)$ is the height of a child node v in $LSPT(u)$. Node u then removes the logical links $\{(u,w) | w \notin DC(u)\}$ from $LL(u)$. The topology generated under the above descriptions is denoted as G^I.

Since for each node u, only one-hop neighborhood information is available for constructing $LSPT(u)$, some links in G^I may be uni-directional; however, such links are unfavorable in wireless ad hoc networks. In [13,15], it is shown that network topologies free of uni-directional links are much more beneficial for MAC layer control mechanisms, link-level acknowledgments, and package transmissions/re-transmissions. Our method of removing uni-directional links is simple: since at the end of Phase I, all nodes are aware of the logical links of their one-hop neighbors, each node deletes its uni-directional links. The resulting topology is denoted as G^{II}. In Section 3.4, we will prove that G^{II} not only preserves network connectivity, but also preserves all minimum-energy paths of G.

3.2.2. Phase (II) path search replacement

At the beginning of this phase, it is assumed that each node perceives the logical link(s) of its one-hop neighbors. As a result, for each node u, $LTV(u,k)$ can be acquired by sending $k-1$ beacon messages. Node u can then reduce its transmission power further by eliminating the critical link(s) that are replaceable with alternative paths. That is, for each critical link (u,v), node u tries to find another path that reaches node v based on $LTV(u,k)$. We call such a 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 search process is also based on Dijkstra's algorithm. After obtaining $LTV(u,k)$, node u applies Dijkstra's algorithm to it to search for the shortest path $\prod_{k-\min}(u,v)$. Note that all the original critical links of each node should be excluded during the search process, which can be achieved by setting their edge weights to ∞ temporally. Assume that the weights of all the shortest paths that originated in 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 in an *ID* list, including all the intermediate nodes of $\prod_{k-\min}(u,v)$. Let cb be a predetermined constant. If no replacing path exists, or $p(\prod_{k-\min}(u,v))$ is more than cb times larger than the weight of edge (u,v), the search process is terminated, $RP(u)$ is set as an empty list, and $ps(u)$ is set to 0. Otherwise, $\prod_{k-\min}(u,v)$ is the eligible replacing path of (u,v), and (u,v) can be excised. An example is illustrated in Fig. 2. The gray line in Fig. 2(a) is the transmission range of p_1 with full transmission power. After Phase I, the area covered by the dashed line in Fig. 2(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 $\prod(p_1,p_6) = p_1p_4p_5p_7p_6$ in $LTV(p_1,2)$ and $p(\prod(p_1,p_6)) < cb \cdot w(p_1,p_6)$, then p_1 can excise (p_1,p_6). In this way, the transmission range (and thus the transmission power) of p_1 can be substantially reduced, as shown in Fig. 2(c).

If all critical links 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 links. In other words, each node executes the search process at most once in Phase II. Node u then advises its k-hop neighbors about its replacing path(s) by sending a notification message. The message is a tuple $nm = \langle id,\ LL,\ RP\rangle$ that records the *ID*, logical links, and replacing path(s) of the sender node. Each constituent link in the replacing path(s) is called a *replacing link*. If a node u 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 links is tagged as replacing. If such links exist, 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 finds 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 taken as the power level it needs to reach its remaining logical links in the constructed topology. The constructed topology resulting from Phase II is denoted as G^{III}.

We believe that by considering a trade-off between localization and efficiency, choosing $k = 2$ should suffice. We show that it can be very difficult for each node u to independently excise some of its critical link(s) based only on $LTV(u,1)$ while keeping the power stretch factor bounded by a constant $cb > 1$. Consider the topology illustrated in Fig. 3, where $u_1 \cdots u_n$ and $v_1 \cdots v_n$ are equally scattered along two rings, respectively. Assume that the receiver power is negligible and the maximum transmitter radius is R. For $1 \leqslant i \leqslant n$, node u_i, and node v_i are collinear, $\|u_iv_i\| = R$, $0 < \|u_iu_{i+1}\| < R$. Since for each node v_i, $1 \leqslant i \leqslant n$, only $LTV(v_i,1)$ is available, so node v_i does not know whether an alternative path $\prod(v_i,u_i)$ exists. Therefore, in case $\|v_i\ {}_{\mathrm{v}i+1}\|^{\alpha} + \|u_{i+1}u_i\|^{\alpha} > (cb-1)\cdot R^{\alpha}$ and some node v_i decides to excise link (v_i,u_i), the new power stretch factor will exceed cb. Moreover, since the one-hop information of each v_i is identical, the constructed graph can be disconnected if each v_i excises its critical link. On the other hand, it is obvious that if each node decides not to excise its critical link, the total power consumption will be much larger than the optimal if $\|u_i\ u_{i+1}\|/R$ and $\|v_iv_{i+1}\|/R$ are close to 0 and $\|v_i\ v_{i+1}\|^{\alpha} + \|u_{i+1}u_i\|^{\alpha} \ll (cb-1)\cdot R^{\alpha}$. As a result, introducing $LTV(u,k)$ with $k > 1$ is necessary to further reduce the total power consumption.

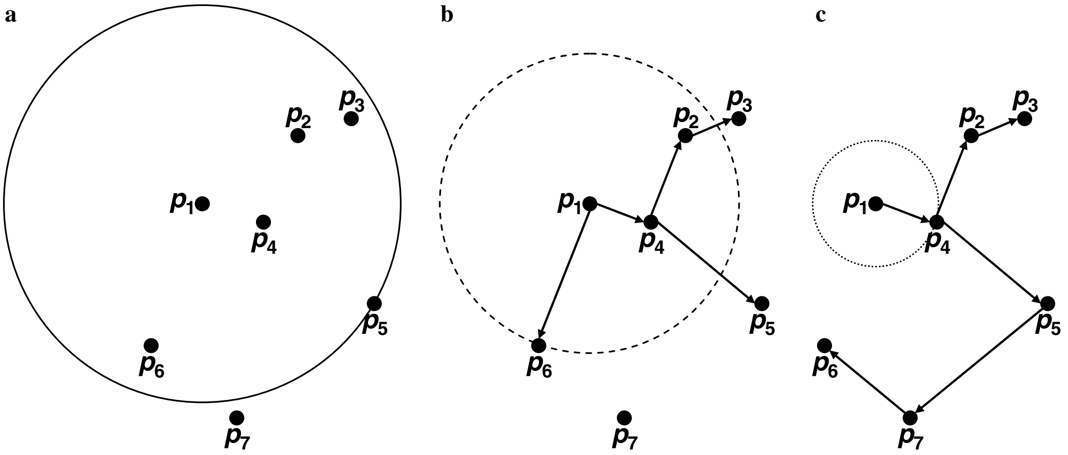

Fig. 2. Illustration of the basic concepts of localized topology control.

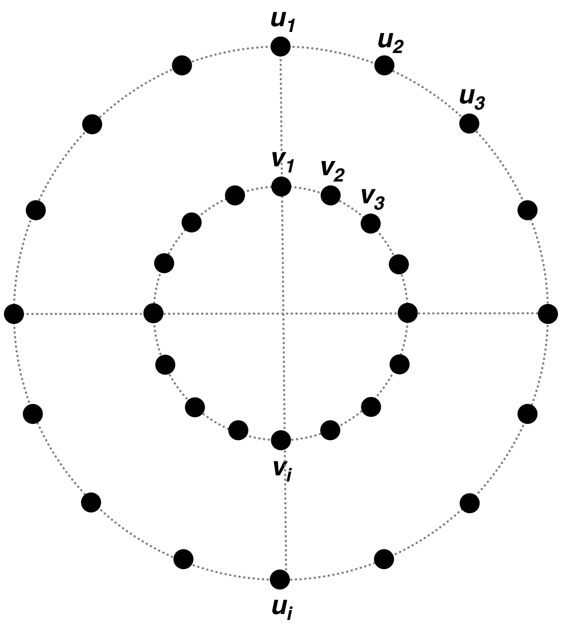

Fig. 3. This topology demonstrates that one-hop neighborhood information is insufficient for reducing the total power consumption while keeping a constrained power stretch factor.

3.3. Heuristics to improve the performance

Clearly, there has to be a tradeoff while searching for the relaying path(s), e.g., if some critical links of low-priority nodes are chosen to construct the replacing paths of the nodes with higher priorities, the transmission power of the 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 reduce 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, we modify the weight of (v_i,v_j) by multiplying it by a *dilation factor* $df(v_i)$, defined as $1+(ps(v_i)/ps(u))$, before applying Dijkstra's algorithm. If all the eligible replacing paths can not be found, Dijkstra's algorithm is applied to the original $LTV(u,k)$, and the above steps are repeated. 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 prove that using $1<cb<2$ can further improve the power stretch factor at the cost of only a slight increase in total power consumption.

3.4. Properties of the constructed topology

We now prove some critical properties of the topology generated by our algorithm. For clarification, we use $wG(u,v)$ 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 (u,v) in the initial $LTV(u,1)$ such that $p(\prod(u,v))<w^G(u,v)$. Assume a deleted link, (u,v), exists on the minimum-energy path $\prod_m$ of G. Since $LTV(u,1)$ is a subset of G, $\prod(u,v)$ should be included in G. If we replace (u,v) with $\prod$ in $\prod_m$, then the transmission power of the new path will be smaller than $\prod_m$, 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 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 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 $\prod(v^1,u^1)$ in $LSPT(v^1)$ such that $p(\prod(v^1,u^1))<w^G(v^1,u^1)$. 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 comprises $p(\prod(v^1,u^1))<w^G(v^1,u^1)$ must satisfy $w^G(u^h,v^h)<w^G(u^1,v^1)$. Therefore, each link that belongs to $\prod(v^1,u^1)$ is bi-directional and is preserved in G^{II}.

Induction: Assume Lemma 2 holds for all deleted links $\{(u^i,v^i)|i=1,2,\dots,k-1\}$. 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 $\prod(v^k,u^k)$ in $LSPT(v^k)$ such that $p(\prod(v^k,u^k))<w^G(u^k,v^k)$. Similarly, each edge (u^h,v^h) that comprises $\prod(v^k,u^k)$ must satisfy $w^G(u^h,v^h)<w^G(u^k,v^k)$. 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. In other words, there is a path, $p(\prod(u^h,v^h))$, composed of bi-directional links that has the minimum total power consumption. This 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 Lemmas 1 and 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, where (u,v) can only be substituted by a replacing path $\prod(u,v)$ such that $p(\prod(u,\ v))\leqslant cb\cdot w^G(u,v)$. Since the

Table 1
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
IJVBT	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

end nodes of each replacing link are in the k-hop neighborhood of node u, they will not begin their search process until u finishes 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 1–4. □

4. Performance comparisons

Via simulations, we compared the performance of our algorithms with that of other approaches in terms of the total power consumption, the power stretch factor, and the node degree of the constructed topologies. The experiment results are summarized in Tables 1–4. Our algorithm is denoted as ESPT. For study purposes, the first part of our algorithm is denoted as ESPT^1, and the part consisting of Phases I and II is denoted as ESPT^2. A unit disk graph (UDG) is used as the basis for comparison. We chose the SMECN algorithm proposed by Li and Halpern [6], since it performs significantly better than the algorithm proposed by Rodoplu and Meng [5] in terms of total power consumption, and it has a power stretch factor of 1.

Similarly, the constrained Gabriel graph (GG) outperforms those described in [3] in terms of total power consumption and the power stretch factor. It is thus chosen for comparison here. The algorithm proposed by Li [16] (AMST) is based on the constrained relative neighborhood graph. However, Li's paper only offers a theoretical analysis without the experimental performance, and is of interest to us. The MST-based topology control algorithm (LMST) proposed in [13] is also used for comparison, because its total power consumption is lower than that of SMECN (though its power stretch factor is larger). Our algorithm operates with $\alpha = 4.0$, $k = 2$, and $cb = 1.5$ and 2.0, respectively. In order to make a fair comparison, it is assumed that each node in SMECN broadcasts 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 do not incorporate the removal of uni-directional links.

We observe the following metrics of each constructed topology H: (i) the total power consumption (*tpc*) associated with H; (ii) the power stretch mean (*psm*), defined as $psf = (\sum_{u,v \in V} \frac{p_H(u,v)}{p_G(u,v)})/N^2$;[1] (iii) the maximum power stretch factor (*max psf*) is observed throughout the experiment;[2] (iv) the variance of transmission power (*var tp*);[3] (v) the average node degree (*avg nd*); and (vi) the maximum node degree (*max nd*) is observed throughout the experiment. The last two metrics are chosen to facilitate a better understanding of the topology's characteristics. The values of *tpc*, *psm*, *var tp*, and *avg nd* are the average of 50 independent experimental results. Note that *tpc* is normalized to lie 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 m. The map sizes are equal to $s \times R$ by $s \times R$ for $s = 1, 3, 5, 7$, where s is directly related to the network's 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 Tables 1–4, we observe that the topology constructed by our algorithm (ESPT^1/ESPT^2) has a much lower *tpc* than those of GG and SMECN in all cases, and the increase in the *psm* of ESPT^2 is almost negligible (less than 0.04). This means that the total power consumption of our algorithm can be reduced significantly, while preserving most of the minimum-energy paths. For comparison with LMST, we first consider denser networks (see Tables 1 and 2). The *tpc* values of our algorithm are slightly higher than those of LMST. However, since the differences are relatively small with respect to UDG, we believe that our algorithm has advantages over 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 may have a very large max psf (about 9 in this case), while our algorithm guarantees

---

[1] *psm* is a modification of the 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 of 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, for convenience, we set $t = 1$ when counting the transmission power.

Table 2
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 3
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 4
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

a bounded *max psf* (1.5 and 2.0 in this case). For sparser networks (see Tables 3 and 4), ESPT^2 outperforms LMST in both *tpc* and *psm*. Although the *max psf* of LMST decreases as s is increases in our experiments, LMST does not provide a theoretical upper bound on the *max psf*. Both ESPT^1 and ESPT^2 have smaller *tpc* values compared to AMST in all cases. Note that although the topology constructed by AMST tends to have a small *psm* (which is still larger than our algorithm), its power stretch factor can be as large as $N - 1$ [16]. Clearly, if the receiver power is a non-zero constant, the power stretch factor of both LMST and AMST becomes even larger.

Both Kirousis et al. [9] and Cheng et al. [17] propose an MST-based centralized solution that approximates the optimal total power consumption with a factor of 2. Since there is no known localized topology control algorithm that guarantees a better approximation factor, we compare the ratio of the *tpc* of ESPT^2 (with $cb = 1.5$ and 2.0) to that of their algorithm in Fig. 4. The ratio remains small and stable as s changes, which shows that our localized algorithm performs almost as well as the centralized algorithm. Finally, we also observe that our algorithm performs well in *var tp*, especially in the case of ESPT^2 with $s > 1$. Since, overall, our algorithm performs better than the other approaches in both the summation and the variance of transmission power, it has a better chance of avoiding node power depletion. The above experimental results also indicate that, if the source and destination nodes are chosen randomly, our algorithm effectively curtails both the 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 basis for power efficient communications, and thus prolong lifetime of the nodes and the network. The topology constructed by each algorithm compared in this section is shown in Fig. 5.

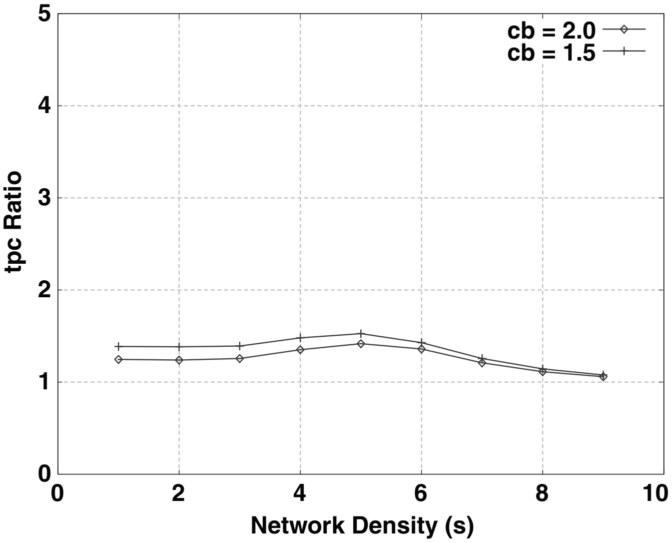

Fig. 4. Comparison of the average total power consumption of ESPT^2 with that of a well-known 2-approximation MST-based centralized algorithm.

We discuss the time complexity of each phase of our algorithm separately. We use N_i to denote the number of nodes in the i-hop neighborhood, and E_i to denote the number of edges in the i-hop neighborhood. First of all, it is well known that the cost of Dijkstra's algorithm is $O(|V|lg|V| + E)$, provided that the priority queue is implemented in a Fibonacci heap. Consequently, each node needs $O(N_1 lgN_1 + E_1)$ time to complete Phase I. In the worst case, the time complexity is $O(|V|^2)$. In Phase II, since each node performs the search process at most once, the computation cost of each node is $O(N_k lgN_k + E_k)$. As in Phase I, the worst-case time complexity of Phase II is also $O(|V|^2)$. Therefore, the time complexity for constructing both G^{II} and G^{III} is $O(|V|^2)$.

5. Dealing with mobility

5.1. Limitations and potentialities

When considering node mobility, each node must be able to adjust its transmission power dynamically. An intuitive idea to achieve network reconfiguration is that all mobile nodes send beacon messages with full transmission power and run the proposed algorithm periodically. In this chapter, 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 time period for sending beacon messages should be very short. In fact, if the topology changes too fast, it is extremely difficult for a topology control algorithm to effectively guarantee network connectivity. On the other hand, the above idea may be too costly in the case of low node mobility. As mentioned in previous works (e.g., [7]), node movement can be viewed as two events, namely node addition and node deletion. Therefore, our task is to find an efficient way to add a new node to and remove a node from a 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 recalculates $LSPT(u)$ and checks if (u, v) is a logical link of this new $LSPT(u)$. If it is, 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 recalculates $LSPT(u)$ and re-executes the search process; node u also checks if (u, v) is a replacing link of some node. If node u becomes aware that the movement of node v will break the replacing path of some node w, u informs w to re-execute the search process and disseminates the 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.

5.2. Practical considerations

Through the simulations of ESPT^2 we observed that, in most cases, the average number of nodes affected by removal of a relaying link (denoted as *anm*) is close to one, with *cb* set to either 2.0 or 1.5 (see Fig. 6). In other words, 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 the transmission radius after deciding the new logical links. If there is a change, node u updates its critical link(s). If node u finds that its *ps* value becomes larger than 0, it disseminates its new priority and re-executes the search process described in Section 5.2. No recalculation is needed for node u if no nodes are added to its one-hop neighborhood and no nodes are deleted from its replacing path. Furthermore, the communications and computations of the reconfiguration process can be performed locally, which is preferred in the context of mobile ad hoc networks.

Another important issue is to provide address autoconfiguration for IP-enabled appliances for future ubiquitous computing, particularly in the case of IPv6 [18]. In fact, various related RFCs have been presented in recent years (e.g., [19,20]). IP-address autoconfiguration protocols should avoid central administration and be robust in order to cope with the dynamic environment. It would be appropriate, therefore, to integrate them with the proposed localized topology control scheme.

6. Concluding remarks and future directions

In this paper, we have demonstrated 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 to construct and maintain a topology on a given unit disk graph. The concept of k-redundant edges has been proposed by Li

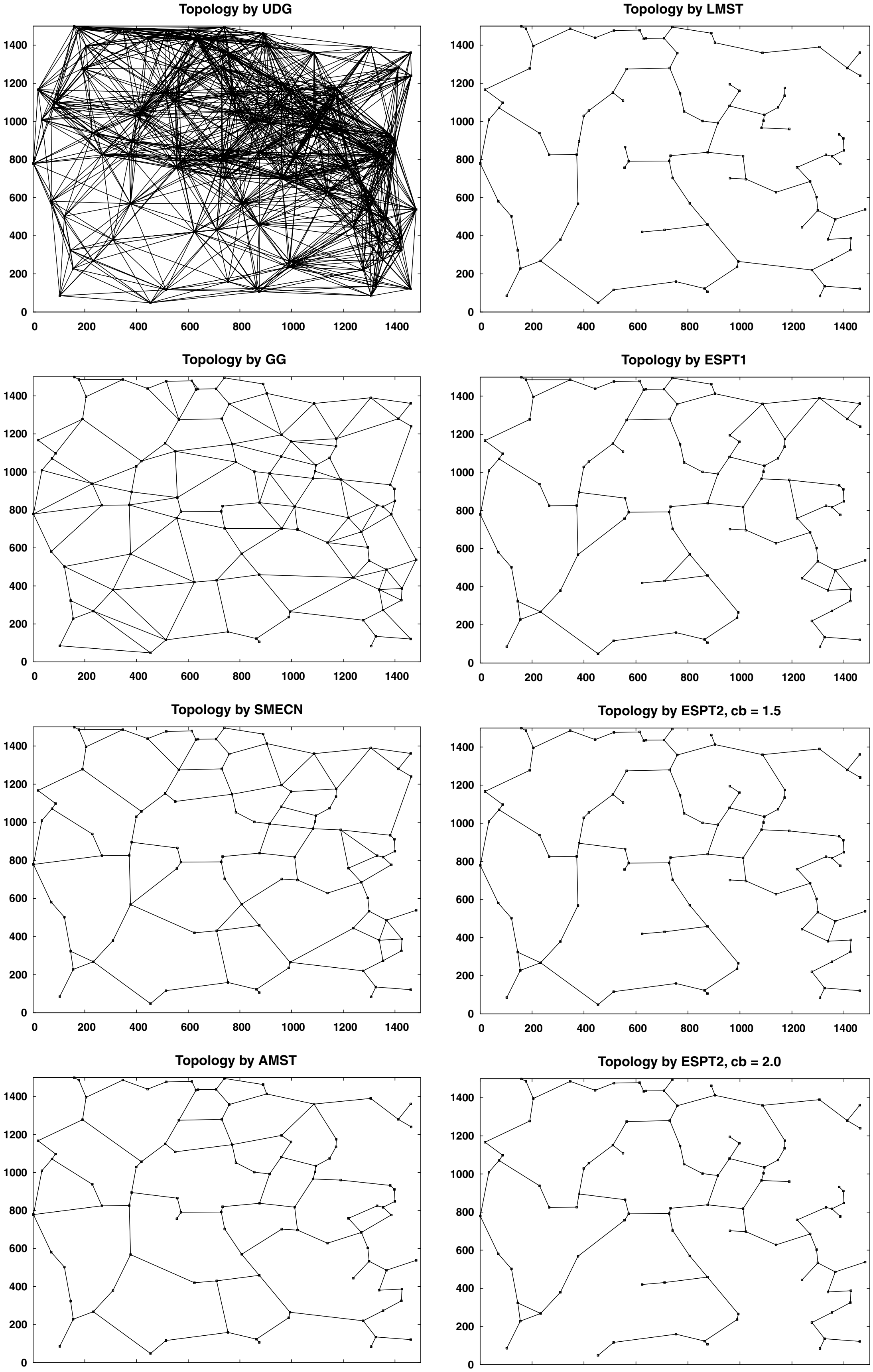

Fig. 5. Network topologies constructed by the above algorithms.

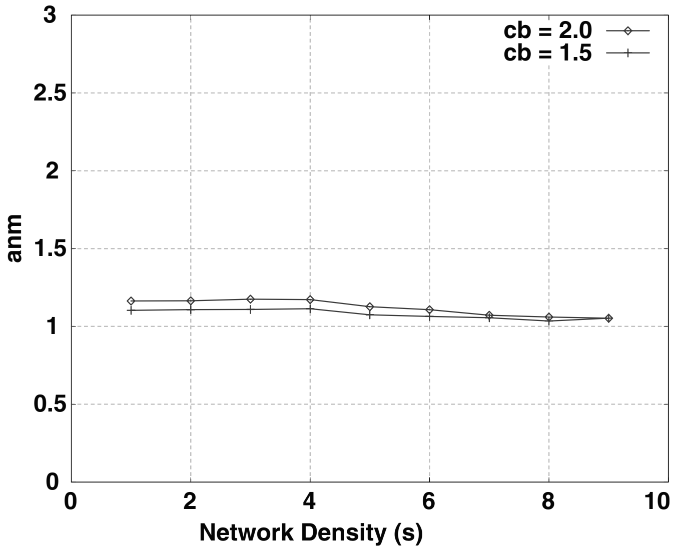

Fig. 6. The *anm* values in different network densities.

and Halpern [6]. However, the algorithm in [6] only deals with 2-redundant edges. The contributions of our algorithm are manifold. First, the approaches in Phase I determine k-redundant edges for $k \geqslant 2$. In other words, our approach can produce a minimum-power topology with lower total power consumption that is totally different to the topologies in [5,6]. Our algorithm is also simpler in computation than those in [5,6]. Second, the topology after Phase II has several desirable features, such as a bounded power stretch factor, low total power consumption and small variance in transmission power. The simulation results show that our algorithm outperforms other approaches in terms of various important metrics. Third, our algorithm provides the flexibility of pre-determining the power stretch factor of the derived subgraph.

In addition, the topology constructed by our algorithm helps other programs save energy in the two ways suggested in [6]: (i) the transmission power computed by our algorithm is sufficient for each node to send beacon messages; and (ii) the routing algorithm can be restricted to using the edges of the current subgraph to find an energy-efficient path for a communication session. In summary, the proposed algorithm effectively and efficiently constructs a virtual backbone that can support energy-aware communication applications over wireless ad hoc networks. Nonetheless, an important question arises: If an algorithm can only guarantee better total power consumption, but has a poor power stretch factor (or vice versa), how can we judge if it guarantees a longer node and network lifetime? Furthermore, it has been pointed out that a suitable definition of network lifetime is application-dependent (e.g., [21,22]). A main concern of our future work is to explore, with empirical evaluation, an efficient solution of network reconfiguration that can effectively handle various asynchronous communication settings in a mobile ad hoc environment.

Finally, IPv4 cannot satisfy the potential requirements of various emerging mobile networks, including wireless ad hoc networks [23]. Therefore, other solutions, such as Mobile IPv6 (MIPv6) are attracting more attention, because they can augment the Internet with mobility management capacity [20,24]. Consequently, extending our work to provide viable next-generation network architectures that would merge ad hoc networks with the Internet and incorporate certain favorable features of IPv6 is of great interest. Unfortunately, integration of this kind poses many technical challenges, such as IP address handoff and seamless message routing (see, e.g., [18,24,25]). Therefore, another future extension of our work will be to investigate techniques that could gracefully integrate the proposed topology control algorithm with IPv6-based networks.

References

[1] L.M. Feeney, M. Nilsson, Investigating the energy consumption of a wireless network interface in an ad hoc networking environment, in: Proceedings IEEE INFOCOM, April 2001.

[2] I. Stojmenovic, X. Lin, Power-aware localized routing in wireless networks, in: Proceedings IEEE International Parallel and Distributed Processing Symposium, 2000.

[3] X.-Y. Li, P.-J. Wan, Y. Wang, O. Frieder, Sparse power efficient topology for wireless networks, HICSS, Hawaii, January 2002.

[4] R. Wattenhofer, L. Li, P. Bahl, Y.M. Wang, Distributed topology control for power efficient operation in multihop wireless ad hoc networks, in: Proceedings IEEE INFOCOM, April 2001.

[5] V. Rodoplu, T.H. Meng, Minimum energy mobile wireless networks, IEEE Journals on Selected Areas in Communications 17 (8) (1999) 1333–1344.

[6] L. Li, J. Halpern, Minimum energy mobile wireless networks revised, in: Proceedings IEEE ICC, June 2001.

[7] X.-Y. Li, P.-J. Wan, Constructing minimum energy mobile wireless networks, in: Proceedings ACM MobiHoc, October 2001.

[8] A.E.F. Clementi, P. Penna, R. Silvestri, On the power assignment problem in radio networks, in: Electronic Colloquium on Computational Complexity (ECCC), 2000.

[9] L. Kirousis, E. Kranakis, D. Krizanc, A. Pelc, Power consumption in packet radio networks, in: Symposium on Theoretical Aspects of Computer Science (STACS), 1997.

[10] G. Calinescu, I. Mandoiu, A. Zelikovsky, Symmetric connectivity with minimum power consumption in radio networks, in: IFIP-TCS, 2002.

[11] X.-Y. Li, P.-J. Wan, W. Yu, Power efficient and sparse spanner for wireless ad hoc networks, in: Proceedings IEEE ICCCN, October 2001.

[12] J.A. Stine, G. de Veciana, A comprehensive energy conservation solution for mobile ad hoc networks, in: Proceedings IEEE ICC, April 2002.

[13] N. Li, J. Hou, L. Sha, Design and analysis of an MST-based distributed topology control algorithm, in: Proceedings IEEE INFOCOM, 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, in: Proceedings IEEE GLOBECOM, December 2003.

[15] S. Narayanaswamy, V. Kawadia, R.S. Sreenivas, P.R. Kumar, Power control in ad hoc networks: theory, architecture, algorithm and implementation of the COMPOW protocol, in: EW2002, Next Generation Wireless Networks: Technologies, Protocols, Services and Applications, February 2002.

[16] X.-Y. Li, Approximate MST for UDG locally, in: Proceedings International Computing and Combinatorics Conference (COCOON), July 2003.

[17] X. Cheng, B. Narahari, R. Simha, M.X. Cheng, D. Liu, Strong minimum energy topology in wireless sensor networks: NP-completeness and heuristics, IEEE Transactions on Mobile Computing 2 (3) (2003) 248–256.
[18] Z. Fan, S. Subramani, An address autoconfiguration protocol for IPv6 hosts in a mobile ad hoc network, Computer Communications 28 (4) (2005) 339–350.
[19] S. Thomson, T. Narten, IPv6 Stateless Address Autoconfiguration, RFC 2462, 1998.
[20] D. Johnson, C. Perkins, J. Arkko, Mobility support in IPv6, RFC 3775, 2004.
[21] L. Li, J.Y. Halpern, V. Bahl, Y.M. Wang, R. Wattenhofer, Analysis of a cone-based distributed topology control algorithms for wireless multi-hop networks, in: Proceedings ACM PODC, 2001.
[22] J.E. Wieselthier, G.D. Nguyen, A. Ephremides, Energy-aware wireless networking with directional antennas: the case of session-based broadcasting and multicasting, IEEE Transactions on Mobile Computing 1 (3) (2002) 176–191.
[23] M. Tatipamula, P. Grossetete, H. Esaki, IPv6 integration and coexistence strategies for next-generation networks, IEEE Communications Magazine (2004).
[24] L. Lamont, M. Wang, L. Villasenor, T. Randhawa, S. Hardy, Integrating WLANs and MANETs to the IPv6 based Internet, in: Proceedings IEEE ICC, May 2003.
[25] J. Eriksson, M. Faloutsos, S.V. Krishnamurthy, Scalable ad hoc routing: the case for dynamic addressing, in: Proceedings IEEE INFOCOM, March, 2004.

Sy-Yen Kuo is the Dean of the College of Electrical Engineering and Computer Science, National Taiwan Ocean University, Keelung, Taiwan. He is also a Professor at the Department of Electrical Engineering, National Taiwan University where he is currently on leave and was the Chairman at the same department from 2001 to 2004. He received the BS (1979) in Electrical Engineering from National Taiwan University, the MS (1982) in Electrical & Computer Engineering from the University of California at Santa Barbara, and the Ph.D. (1987) in Computer Science from the University of Illinois at Urbana-Champaign. He spent his sabbatical years as a Visiting Professor at the Computer Science and Engineering Department, the Chinese University of Hong Kong from 2004 to 2005 and as a visiting researcher at AT&T Labs-Research, New Jersey from 1999 to 2000, respectively. He was the Chairman of the Department of Computer Science and Information Engineering, National Dong Hwa University, Taiwan from 1995 to 1998, a faculty member in the Department of Electrical and Computer Engineering at the University of Arizona from 1988 to 1991, and an engineer at Fairchild Semiconductor and Silvar-Lisco, both in California, from 1982 to 1984. In 1989, he also worked as a summer faculty fellow at Jet Propulsion Laboratory of California Institute of Technology. His current research interests include dependable systems and networks, software reliability engineering, mobile computing, and reliable sensor networks.

Professor Kuo is an IEEE Fellow. He has published more than 240 papers in journals and conferences. He received the distinguished research award (1997-2005) from the National Science Council, Taiwan. He was also a recipient of the Best Paper Award in the 1996 International Symposium on Software Reliability Engineering, the Best Paper Award in the simulation and test category at the 1986 IEEE/ACM Design Automation Conference (DAC), the National Science Foundation's Research Initiation Award in 1989, and the IEEE/ACM Design Automation Scholarship in 1990 and 1991.

Szu-Chi Wang received the BS degree in Computer Science and Information Engineering and the Ph.D. degree in Electrical Engineering from National Taiwan University in 1995 and 2005, respectively. His research interests lie in the areas of designing, implementing and analyzing distributed algorithms for wireless ad hoc networks.

David S.L. Wei received the Ph.D. degree in Computer and Information Science from the University of Pennsylvania, Philadelphia, in 1991. He is currently a Professor of Computer and Information Science Department at Fordham University. From May 1993 to August 1997 he was on the Faculty of Computer Science and Engineering at the University of Aizu, Japan (as an Associate Professor and then a Professor). Dr. Wei has authored and co-authored more than 70 technical papers in the areas of distributed and parallel processing, wireless networks and mobile computing, and optical networks in various archival journals and conference proceedings. He served on the program committee and was a session chair for several reputed international conferences. He served as a co-chair of Power Aware Communication and Software, Minitrack in the Software Track at the 34th Hawaii International Conference on Systems Sciences (HICSS-34). He is a lead guest editor of IEEE Journal on Selected Areas in Communications for the special issue on Mobile Computing and Networking, and a guest editor of IEEE Journal on Selected Areas in Communications for the special issue on Peer-to-Peer Communications and Applications. Currently, Dr. Wei focuses his research effort on wireless networks, mobile computing, and peer-to-peer communications.