Localized Topology Control Algorithm with no Geometric Information for Ad hoc Sensor Networks

Kamrul Islam, Selim G Akl
School of Computing, Queen's University
Kingston, Ontario, Canada K7L 3N6
islam@cs.queensu.ca, akl@cs.queensu.ca

Abstract

We present a localized algorithm (LTCA) for topology control in wireless sensor networks which has certain desirable properties. First, the algorithm is very simple, strictly local (requires 1-hop information of neighbors) and fast, requiring each node to communicate with its neighbors exactly twice. Second, LTCA does not assume the underlying graph to be a unit disk graph, and in fact works on general graphs. Third, the most important feature of LTCA is that it is completely independent of any location information of nodes in the network graph; it relies only on the connectivity information and the identities (ids) of the neighboring nodes. Assuming sensor nodes are deployed in the plane, it is shown that the resulting subgraph obtained by our algorithm is connected, symmetric and contains few edge crossings. We provide simulation results and show that on random graphs the resulting topology is spanner and the average degree of the subgraph is low. Due to the simplicity of the algorithm, LTCA can easily be implemented in ad hoc sensor networks.

1. Introduction

A wireless sensor network is composed of a set of sensors which communicate with one another over wireless links. Two sensors communicate with each other directly if they are within their transmission ranges otherwise they rely on other sensors to establish an indirect connection between them. Unlike wired networks, sensors in a wireless networks do not rely on a pre-existing communication infrastructure. Due to the limited on-board energy of sensors, algorithms for sensors networks are designed to cleverly utilize this valuable energy to lengthen the lifetime of such networks. In order to reduce the power consumption of the network and extend its lifetime, topology control algorithms, in general, deal with finding a suitable structure (strictly speaking, a spanning subgraph) of the given graph. This resulting subgraph is expected to have certain features (e.g., connectedness, planarity, sparseness, bounded-degree, etc.) which facilitate routing in the network. Formally, we can model a connected sensor network as a graph $G = (V, E)$ where the node set V represents the sensors deployed in the plane and the set of edges E contains all links (u, v) where $u, v \in V$ can directly communicate to each other, topology control algorithms focus on constructing a subgraph $G' = (V, E_{TC}), E_{TC} \subseteq E$. For transmission, longer links in the network consume more power than shorter links since the power spent on a link for a transmission is directly proportional to at least the square of the Euclidean length of that link. That is why, the general approach of topology control algorithms is to remove longer links and retain shorter links which are supposed to save energy in the long run for routing and thus help extend the lifetime of the network. However, the disadvantage of this idea is that if too many longer links are eliminated then some paths may be unacceptably longer in terms of the number of hops. This increases the probability of message loss and takes longer to deliver messages. Care must be taken to design topology control algorithms to meet this challenge.

For any topology control algorithm, the first and foremost requirement is that the subgraph G' be connected. It is expected that G' be symmetric, that is, for any two nodes u and v, u is a neighbor of v if and only if v is a neighbor of u. Among other properties, G' should be planar because certain algorithms (for example, GPSR [2]) utilize the planarity property of graphs for successful routing. Another common requirement is that G' be sparse; that is, the number of edges should be in the order of the number of nodes. Sparsity of a graph helps maintain simple neighborhood for individual nodes. This is because the smaller the neighborhood, the faster the information processing among the nodes in the neighborhood. One of the important properties of G' is that it should be a spanner. This means that the cost of a shortest path between any two nodes in G' must be at most t times the cost of the shortest path between the same two nodes in G. There are a number of cost metrics used in sensor networks. The most often used cost functions are the Euclidean length

of links, the Euclidean length raised to a predefined power, etc. The cost of a path is determined to be the sum of the costs of all the links of the path.

The rest of the paper is organized as follows. In Section 2 we review previous works on topology control in ad hoc sensor networks. In Section 3 we provide definitions and assumptions that are used throughout the paper. Our localized topology control algorithm ($LTCA$) is presented in Section 4. An average-case analysis of the network graphs is presented in Section 5. We conclude in Section 6.

2. Related work

Topology control in sensor networks has been studied extensively, see for example [1, 8, 9, 13, 15, 17, 10, 18] where authors propose centralized schemes as well as distributed algorithms. Since the sensors can be modeled as a set of points in the plane, many algorithms use some of the fundamental results of computational geometry to produce nice structures from the underlying graph. These include the minimum spanning tree (MST) [13], the Delaunay triangulation [4], a generalized version of the Gabriel graph [14], the relative neighborhood graph (RNG) [7], and so on. A Delaunay triangulation based method to select edges from the underlying graph is described in [4]. Using some heuristics the algorithm selects edges in such a way that they form a regular and uniform structure and the degrees of nodes are upper-boundeded by some constant. In [13], the authors present a centralized method that constructs a spanning tree from the given graph where the goal is to minimize the maximum power of the network. The maximum power of a sensor is defined as the power required to transmit a message to the furthest neighbor of that sensor. However, they also present two distributed algorithms that adjust the transmission power of individual sensors to reduce the power consumption. The problem with their method is that the algorithms do not ensure connectivity in all cases. However, these algorithms [4, 13] may not be effectively applicable to power constrained sensor networks since neither the Delaunay triangulation nor the MST can be computed locally and hence require all the sensors to transmit their positions to the base stations for centralized solutions, which is obviously power-consuming. The Gabriel graph (GG) based solution described in [14] is quite reasonable for sensor networks since GG can be locally computed and this graph is symmetric and energy-spanner. The first distributed topology control algorithm which achieves many of the properties described above is called CBTC (Cone-based Topology Control Algorithm) [18]. The subgraph obtained by the algorithm is an energy-spanner, is planar and sparse and can be distributedly (but not locally) computed. Since then there have been a number of algorithms [5, 6, 16] which have proposed local algorithms to obtain these properties.

A locally constructed spanning subgraph known as the low-weighted modified relative neighborhood graph (RNG) is introduced in [7]. The idea is based on a simple modification of the original RNG where the author shows that the structure obtained by this algorithm is connected and planar. He shows that the sum of all the edge lengths (Euclidean distance) of the subgraph is within a constant factor of that of the MST. This method is local, uses only $O(n)$ messages to build such structures and every node uses only its two-hop neighbor information. Although the total edge length of this structure is within a constant factor of that of the MST, the energy consumption using this structure is not within some constant of the optimum. Wattenhofer and Zollinger [19] propose a simple topology control algorithm, called XTC, that is independent of sensors' specific coordinates in the plane but assumes the distances among neighboring sensors are available. As all the previous algorithms utilize the fact that the location of individual sensors is known, this is the first location-independent algorithm to produce a connected, planar, and degree-bounded topology. XTC also works on general graphs since it is not location dependent. Although the subgraph is not a spanner, the authors provide simulation results and show that the spanner property holds on average-graphs. The only disadvantage of that algorithm is that a small error in the estimated distance information may generate a disconnected subgraph. That is, exact distance information is crucial for their algorithms to produce a connected structure. Later to circumvent them problem, Kevin and Sriram [11] came up with two randomized algorithms which are a generalized version of [19] to produce a topology that is guaranteed to be connected. These algorithms are robust to distance error in the sense that they withstand a certain amount of error in estimating distances between neighbors. Although the resulting subgraph is planar and connected, one of the algorithms cannot guarantee to yield a bounded degree of the resulting topology. In fact, a node in the network topology can have a logarithmic bound on the degree in the original graph. The other randomized algorithm guarantees to produce a subgraph which is bounded degree with high probability.

2.1. Failure due to imprecise location information

Providing each sensor with a global positioning system (GPS) for obtaining exact location information is expensive and prohibitive when thousands of sensors are deployed in the plane. Moreover, these devices consume energy for their operation and thus deplete the valuable on-board energy of sensors. Even the sophisticated GPSs are assumed to have certain measurement errors which make the reported geographic locations of nodes only an approximations of the true location. Algorithms that depend on precise geographic locations may produce disconnected graphs. We show two location-based algorithms used for topology control in sensor

networks which can produce disconnected subgraphs if either the precise location information is not available or there is small error in the distance information between neighbors.

As mentioned before, a very simple and robust algorithm for topology control (XTC) which achieves a number of properties of graphs has been introduced in [19]. When the underlying graph is modeled as a unit disk graph (that is, a link between two sensors exists if the Euclidean distance between them is at most one), the technique [19] assumes exact distance information between neighbors. The algorithm finds a topology relying only on these distances and the ids of the sensors in the graph. However, the approach suffers from the problems of disconnectedness and unbounded degree. For a clear exposition we briefly present the algorithm in [19]. The XTC protocol consists of three steps: (i) Neighbor ordering and neighbor order exchange and (iii) Edge selection. A total order $\prec_u$ is defined and used by each node u to order its neighbors

$v \prec_u w \Leftrightarrow (|(u,v)|, \min\{id(u), id(v)\}, \max\{id(u), id(v)\})$
$< (|(u,w)|, \min\{id(u), id(w)\}, \max\{id(u), id(w)\})$ (1),

where $|(u,v)|$ is the Euclidean distance between u and v. The neighbor orders of the XTC algorithm are based on the lexicographic order of these link weights. u drops v from its neighborhood if there exists w such that $w \prec_u v$ and $w \prec_v u$ and thus u forms an updated neighborhood by removing such v's. Finally, the output of the protocol, that is, the spanning subgraph $G_T = (V, E_T)$ consists of the edge set, E_T where $E_T = \{(u,v) | v \in N(u)\}$ where $N(u)$ is the set of neighbors of u. It is shown in [19] that G_T is symmetric, connected, planar and bounded degree if the underlying graph is connected. As mentioned before, the performance depends on the correct distances of the neighbors and it is sensitive to small perturbances in the distances which produce different orderings. In order for the understanding, we adopt the example mentioned in [11] which shows that XTC becomes disconnected with a small error in the distances of the neighbors. For a unit disk graph shown in Figure 1 (a), assume $0 < \epsilon < 1 - 1/1\sqrt{2}$ and the lengths of the edges are $|(a,b)| = |(d,c)| = (1-\epsilon)/2$ and $|(a,c)| = |(b,d)| = 1/2$. The neighborhood orderings $\prec$ according to (1) are:

$$d \prec_a b \prec_a c,\ c \prec_b a \prec_b d,\ b \prec_c d \prec_c a,\ a \prec_d c \prec_d b.$$

However, if b and c estimate their distances ($|(b,a)|$ and $|(c,d)|$, respectively) incorrectly,i.e., $|(b,a)| = (1+\epsilon)/2$ and $|(c,d)| = (1-\epsilon)/2$, then we obtain the following orderings:

$$d \prec_a b \prec_a c,\ c \prec_b d \prec_b a,\ b \prec_c a \prec_c d,\ a \prec_d c \prec_d b.$$

With this new orderings, when the XTC algorithm is executed the subgraph becomes disconnected as shown in Figure 1 (b). In the same paper [11], the authors also show instances where nodes in G_T can have unbounded degree if some nodes incorrectly estimate distance between them. For details, the reader is referred to [11].

Here we show that due to a small error in the geographic position of a node, GG can produce disconnected subgraphs. Consider the Figure 1 (c). Suppose the exact positions of nodes a and b are known and a incorrectly estimates the location of c to be c'. The absolute deviation can be made smaller as it is evident from the figure. Assume edges (a,b) and (b,c) belong to the given graph. According to the definition of GG, an edge between two nodes a and b exists if the circle with the diameter ab does not contain any other node inside the circle. However, due to the inaccurate location information of node c, the edge (a,b) does not exist anymore in the GG. But it could exist if c's position were precisely determined. Thus the graph is disconnected and cannot be used for useful purposes, particularly routing.

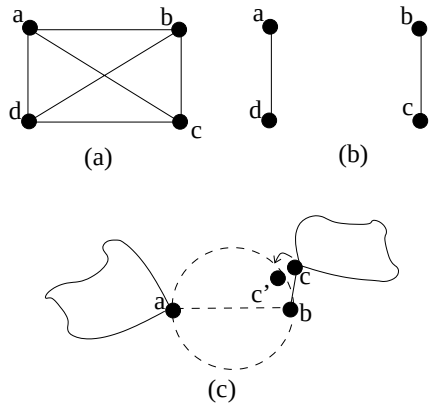

Figure 1. XTC produces a disconnected graph (b) with a small error in the distances among nodes in the unit disk graph in (a). (c) If a incorrectly estimates c's actual position and assumes it is inside the circle as c' then GG can be disconnected because the edge (a,b) gets removed.

3. Preliminaries

A undirected graph consists of a finite set of nodes and a set of edges, where each edge connects a pair of nodes. It is assumed that the graph is connected, that is, there is at least one path between any two nodes. A geometric graph is a graph where the nodes are represented by a set of points in the plane and the edges are line segments joining the nodes. We model a sensor network (sensors are deployed in the plane) by an undirected connected geometric graph $G = (V, E)$ where V denotes the set of sensors and the edge set E represents all links $(u,v) \in E$ between $u, v \in V$ if u

and v can communicate directly to each other. We define the neighbor set $N(u)$ of u as $N(u) = \{v|(u,v) \in E\}$ which consists of nodes one hop away from u. A unit disk graph is a geometric graph containing an edge (u,v) if the Euclidean distance between u and v is at most one. Throughout, we use (u,v) to represent an undirected edge between u and v. Being equipped with a unique id, each node u (the id of u denoted $id(u)$) contains an omnidirectional antenna which is assumed to send identical power in every direction in the plane. The cost of a link (u,v) in the network is defined to be the amount of power required for sensor u to send a message to sensor v or vice versa assuming symmetric links. Thus the cost of (u,v) is $|(u,v)|$ for the Euclidean metric and $|(u,v)|^\alpha$ for the energy metric where $\alpha \geq 2$. A path $p_G(s,t) = (s = u_1, u_2, \cdots, t = u_k)$ in G from node s to node t is a sequence of edges $(u_i u_{i+1})$ and the cost of $p_G(s,t))$ is the sum of the costs of all the edges in $p_G(s,t))$. We consider the spanner ratio of a graph G' as:

$$t \geq |p_{G'}(u,v)^e|/|p_G(u,v)^e|,$$

where t is a constant called the spanner ratio and $|p_G(u,v)^e|$, $e \geq 1$ is the cost of a shortest path between u and v in G in some metric. t is called the Euclidean spanner ratio (resp. energy-spanner ratio) for the Euclidean metric (resp. the energy metric).

4. Localized topology control algorithm (LTCA)

In this section, we present a deterministic localized topology control algorithm ($LTCA$) for wireless sensor networks. The assumption is that the sensors are deployed in the two dimensional plane and they do not have any information about their position or distances to their neighbors. First we informally describe our algorithm. By transmitting to the maximum transmission power, sensors explore their neighbors, that is, individual sensors learn the corresponding number of neighbors and the ids of these neighbors. Through sending "hello" messages, sensors transmit their ids and also receive ids from their neighbors in the first two steps of $LTCA$. As the algorithm is executed at each node u, u starts removing some of the neighbors from its neighbor list in order to build a sparse graph in the next step. It checks every pair of nodes $k, m \in N(u)$ to see whether they are connected. If they are not connected or if u has only one neighbor, then u retains the neighbors(s) and no removal of neighbors from $N(u)$ takes place. If they are neighbors to each other, i.e., $(k,m) \in E$, then u removes none if $id(u) = \min\{id(k), id(m), id(u)\}$. However, if $(k,m) \in E$ and $id(u) \neq \min\{id(k), id(m), id(u)\}$, then u removes from $N(u)$ the node that has the $\max\{id(k), id(m)\}$. Removal of some neighbor k from $N(u)$ means the corresponding link (u,k) is eliminated from the network. Thus u builds an updated neighbor list $N'(u) \subseteq N(u)$ with the remaining neighbors.

Our topology control algorithm ($LTCA$) which is executed at each node u has the following steps:

Input: For any general connected graph, $G = (V, E)$.

Output: A Spanning subgraph, $G_{TC} = (V, E_{TC} \subseteq E)$.

i) Each node u broadcasts its id(u) to all its neighbors $v \in N(u)$.

ii) For each neighbor $v \in N(u)$, u receives their ids, id(v).

iii) For a node $m \in N(u)$ if there exists some node $k \in N(u)$ such that $id(k) < id(m)$ and $id(k) < id(u)$ then m is dropped from u's list. Otherwise no neighbors are deleted from $N(u)$. The updated neighbor set of u is denoted as $N'(u) \subseteq N(u)$.

After the execution of $LTCA$, we find a subgraph $G_{TC} = (V, E_{TC})$ where $E_{TC} = \{(u,v)|\forall u : v \in N'(u), u \in V\}$.

In the following subsection, we provide an analysis of the graph G_{TC} generated by $LTCA$.

4.1. Analysis of $LTCA$

The foremost property of any topology control algorithm is that the resulting graph must be connected, i.e., there must be at least one path between any two nodes in the subgraph. In this section, we show that the resulting graph $G_{TC} = (V, E_{TC})$ generated by $LTCA$ is connected. Unless otherwise mentioned, the given graph $G = (V, E)$ is assumed to be a general graph and not necessarily a unit disk graph.

Theorem 4.1 *Given a geometric graph $G = (V, E)$, two nodes u and v are connected in $G_{TC} = (V, E_{TC})$ if they are connected in G.*

Proof Suppose there are two nodes u and v connected in G but not connected in G_{TC}. First assume that u are v are neighbors in G, that is, $(u,v) \in E$ in G. If $v \in N'(u)$ and there is no $m \in N'(u)$ such that $m \in N'(v)$ then u and v forms a clique of size two and the algorithm does not discard v from $N(u)$. So $(u,v) \in E_{TC}$. If this is not the case then v is connected to at least one neighbor $m \in N'(u)$ of u and so the neighborhood of u forms a clique of at least three nodes. Consider such a clique C of size p. So we can have $p-2$ cliques of three nodes where u and v are fixed. According to the construction, there are two ways u and v not to be neighbors in G_{TC}, namely $id(u) \neq \min\{id(k) : k \in C\}$ and $id(v) \neq \min\{id(k) : k \in C\}$. It is obvious that if any of u or v has the smallest id among the nodes in C then the adjacent edges to the smallest id node are not removed (removing u

from v's neighbor list implies removing the corresponding edge, (u, v)). Thus if u or v has the minimum id in C then we are done. Suppose none of them has the minimum id. Then it is easy to see that we could reach v from u via the minimum id node in C. So if v is a neighbor of u in G and u forms a clique C with v, then v is at most two hops away from u in G_{TC}.

Suppose there is no edge (u, v) in G but there is a path between them in G since G is connected. We will show that between u and v there exists a path in G_{TC}. We can prove this part through induction. Suppose we have a path $\pi = (u = m_1, m_2, m_3, \cdots, m_{t-1})$ from u to some node m_{t-1} in G_{TC}. The base case which states that $u = m_1$ can reach $v = m_2$ via a path of at most two hops, is already established. We need to consider whether there is a path between m_{t-1} and m_t where $m_t = v$. We distinguish between two cases:

i) The node m_t is such that there is no neighbor $t' \in N'(m_{t-1})$ such that $t' \in N'(m_t)$. Then m_t is not removed from m_{t-1}'s neighbor list because m_{t-1} and m_t form a clique of size two and hence they are directly connected to m_{t-1}.

ii) If m_t is such that there are some neighbors $t' \in N'(m_{t-1})$ such that $t' \in N'(m_t)$. Then it forms a clique of size at least three and we can reach m_t from m_{t-1} with at most two hops.

Hence, all pairs of nodes are connected in G_{TC}. ∎

In the following, we prove that the edges in G_{TC} are symmetric, that is, if node u includes v as its neighbor in G_{TC}, then v includes $u \in N'(v)$. Having the subgraph symmetric is an important characteristic, because asymmetric communication graphs make simple assumptions complicated and are generally hard to realize in practical situations as investigated in [12].

Theorem 4.2 *Given a geometric graph $G = (V, E)$, node u includes v in $N'(u)$ if and only if v includes u in $N'(v)$ in G_{TC}.*

Proof First it is obvious from the last step of $LTCA$ that for any clique C of size $|C| \in \{2, 3\}$ in the neighborhood of u, if $id(u) = \min\{id(v) : v \in C\}$ then the edge (u, v) is never removed because for any $k \in C$, $k \neq v$ $id(k) > id(u)$ (if such k exists). Now for the sake of contradiction assume u includes v in $N'(u)$ but v does not include u in $N'(v)$. As mentioned above, if u and v form a clique of size two then we are done because according to step (iii) of $LTCA$, neither of them removes the other. If this is not the situation, then node u includes v in $N'(u)$ since for any clique C of three nodes $u, v, k \in C$, $id(k) > \min\{id(u), id(v)\}$. Without loss of generality, assume $id(v) > id(u)$, then $id(k) > id(u)$. Now consider the same neighborhood and the clique C around v. Node v did not include u in $N'(v)$ since $id(u) > \min\{id(v), id(k)\}$ which is a contradiction because $id(u) = \min\{id(v), id(k), id(u)\}$. Therefore, v must include u in $N'(v)$. ∎

We show that the spanning subgraph generated by $LTCA$ contains few intersecting edges. We show that, in general, all the intersections of a clique of any size can be eliminated. This fact is obvious from the following observation. In a clique C of size p, without loss of generality, let the node with the smallest id be u. According to $LTCA$ (step (iii)), any two nodes $w, x \in C$, will remove the edge between them since they both have ids greater than the id of u. This is true for any two nodes in $C - \{u\}$. Thus all the nodes in $C - u$ are directly connected to u and there will be no edge between any two nodes in $C - \{u\}$. Therefore, in u's neighborhood, we have reduced the number of intersections from $O(p^2)$ to 0. However, some nodes in $c \in C - \{u\}$ which are connected to nodes $d \in V - C - \{u\}$ can intersect some edges (a, b) where $a, b \in C$. Elimination of such intersections is impossible with only connectivity information and no geographic location information or even lengths of edges among neighbors. Situations like the one shown in Figure 2 where the graphs 2(b) and 2(c) are indistinguishable from each other (labels indicate the corresponding ids of the nodes) and hence crossings cannot be removed with only connectivity information. However, as illustrated in Figure 2(a) (right) we can obtain a planar subgraph from the clique shown of Figure 2(a)(left) assuming u has the lowest id in the clique. Thus we have the following lemma:

Lemma 4.3 *Given a geometric graph $G = (V, E)$, a clique C of any size in the subgraph $G_{TC} = (V, E_{TC})$ generated by $LTCA$ induces planarity in C unless there are edges (c, d), $c \in C$ and $d \in V - C$ crossing some edges adjacent to the nodes in C.*

Theorem 4.4 *Given a geometric graph $G = (V, E)$, any cycle in the the subgraph $G_{TC} = (V, E_{TC})$ generated by $LTCA$ has length at least 4.*

Proof For the sake of contradiction, assume there is a cycle C through three nodes u,v and w, that is, edges (u, v), (u, w) and (v, w) are in $G_{TC} = (V, E_{TC})$. Since the ids of the nodes are unique there will be a node in the cycle which has the smallest id of the three nodes in C. The algorithm removes the edge between the two nodes which have larger ids among the nodes in C. So the edge with the two larger ids will not be in C which contradicts to the assumption that there is a cycle of length 3. ∎

5. Analysis for random graphs

In this section, we study and provide experimental results regarding the sparseness and spanner properties of the topology G_{TC}. For simulation, we assume the underlying graph is a unit disk graph in which the spanner and sparseness properties of G_{TC} are analyzed. For this purpose, we generate unit disk graphs by placing nodes uniformly and randomly

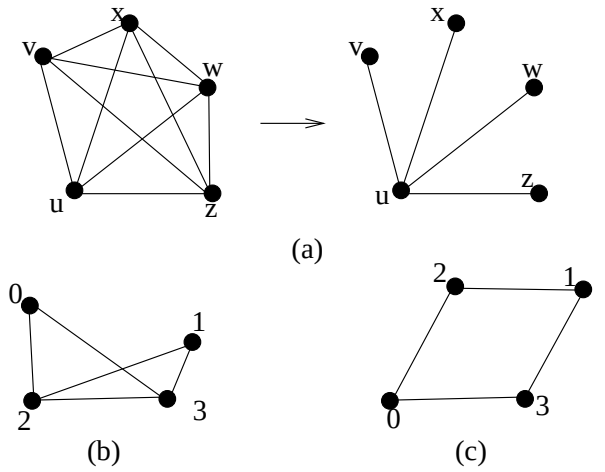

Figure 2. Situations like the one shown in (b) and (c) are indistinguishable from each other and hence crossings cannot be removed with only connectivity information: labels indicate the corresponding ids of the nodes. As illustrated in (a) (right) we can obtain a planar subgraph from a clique (left), assuming u has the lowest id in the clique.

on a given fixed square field and demonstrate that the spanner property for G_{TC} holds on these average graphs. For the sparseness property of G_{TC}, we show that the average degree of the nodes of this graph is low for each randomly generated unit disk graph, although the maximum degree of G_{TC} is not bounded. This low degree indicates that G_{TC} is sparse in the average-case. In Figure 3 we show the unit disk graph, the Gabriel graph and G_{TC} of 600 nodes placed randomly and uniformly where the minimum distance between any two nodes is at least 0.4 unit.

To evaluate the different properties of G_{TC}, we compare it with the Gabriel graph (GG). An edge (u, v) between two nodes u and v in the GG exists if the circle with (u, v) as the diameter does not contain any other nodes inside it. We choose GG since it is one of the most prominent topology control structures which is connected, planar, energy-spanner and can be locally computed. Thus it is a good candidate for comparison, specially with respect to energy-spanner property. Although GG is not spanner in the Euclidean metric (the cost of an edge (u, v) is the Euclidean distance $|(u, v)|$), it is an optimal energy-spanner when the metric is an energy metric (the cost of an edge (u, v) is some power of distance $|(u, v)|^{\alpha}$ where α is in between 2 and 6).

5.1. Sparseness and spanner properties of G_{TC}

First we discuss the spanner property of G_{TC}. In order to study the spanner property of G_{TC} on randomly generated graphs, we consider both the Euclidean spanner ratio and the energy-spanner ratio of G_{TC} and GG:

$$t \geq |p_{G_{TC}}(u, v)|^e / |p_G(u, v)|^e.$$

As mentioned before, t is a constant called the spanner ratio (Euclidean spanner ratio for Euclidean metric and energy-spanner ratio for energy metric) $|p_G(u, v)|$ is the cost of a shortest path between u and v in G in some metric and $e \geq 1$ and the cost of a path under some metric is the sum of the costs of all the edges between u and v. The smaller the value of t, the better the spanner a subgraph is.

Network density, i.e., the number of nodes in a unit disk is an important parameter which influences many of the properties [3] including the spanner property of the average case network graphs. We consider a number of different density values starting from a low density where there are 4 nodes/unit disk, 6 nodes/unit disk, 12 nodes/unit disk, 25 nodes/unit disk. We consider both the Euclidean and energy-spanner with these different network densities for GG and G_{TC}. For each network density mentioned above, we randomly generate 1000 different network graphs of the same size and for each such graph we randomly choose a pair of nodes to compute the Euclidean and the energy-spanner ratios in both GG and G_{TC}. Then we compute the maximum and the average spanner ratios of all the 1000 graphs for each network density. Figure 4(left) depicts the results of the spanner ratios of GG and G_{TC} with respect to the Euclidean metrics. The solid and the dashed lines represent the spanner ratios of GG and G_{TC} respectively. Mean values are plotted in black and max values in gray. It can be seen from the figure that GG has a mean spanner ratio which is slightly above 1.1 and the maximum value is around 1.35. In the same Euclidean metric G_{TC} has a steady mean value around 1.2. However, the maximum value is less stable than the GG but stays below 7. We observe that the lower the network density the better the spanner ratio in G_{TC}.

The energy-spanner ratio is shown in Figure 4 (right) where the solid and dashed lines denote the energy-spanner ratios for GG and G_{TC} respectively. Since GG contains an energy-minimal path between any pair of nodes its energy-spanner ratio is exactly one (mean and max values are the same). G_{TC} has a maximum energy-spanner ratio as high as 13, but it maintains an average energy-spanner ratio below 1.5 in all considered network densities. In total, we ran our simulations on 10000 randomly generated graphs with different density values to come up with the above performance results.

We also perform simulations to study the average-case

behavior of node degree in both GG and G_{TC}. Figure 5 shows that the G_{TC} has a better average degree than that of GG. For each of the random graphs with different number of nodes the average degree of GG is always high (at least 3.5 and above) whereas the average degree of G_{TC} is always below 3.4. Although they are not stable, the gap between them is significant.

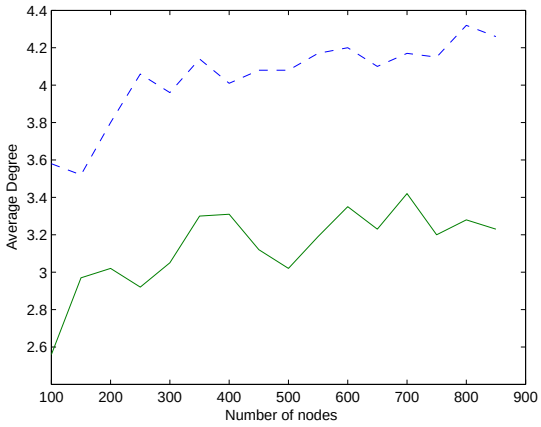

Figure 5. Average degree of GG and G_{TC} in random graphs of different sizes. The dashed and the solid lines denote the sparseness of GG and G_{TC} respectively. As can be seen from the figure, the average degree of G_{TC} is always smaller than that of GG in all the instances of different sizes of network graphs.

6. Conclusion

In this paper, we have presented a simple algorithm $LTCA$ for topology control in wireless sensor networks which has certain properties. This one-hop localized algorithm is very simple, requires the ids of the neighbors and assumes no geometric information at all. Each node communicates with its neighbors only twice for sending and receiving ids and the algorithm terminates after two rounds. Unlike most other techniques, $LTCA$ does not assume the graph to be a unit disk graph. In fact it works for all connected graphs since it does not use any geometric information. One fundamental property is that it always produces a connected spanning subgraph from the underlying graph. This property is robust because many existing well-known topology control protocols (such as the Gabriel graph, the Relative neighborhood graph, the XTC-algorithm) fail to produce connected subgraph if the exact location information of the nodes is not known or there are errors in the estimation of distances between nodes. In practical situations, providing precise location information is expensive and even challenging and a slight error in the geographic location of nodes results in a disconnected subgraph. We show that the subgraph generated by $LTCA$ is symmetric and has few crossings. We provide extensive simulation results for the sparseness and spanner property in random network graphs. Comparing with the well-known topology, Gabriel graph, it is shown that we can obtain good results in terms of sparseness and spanner ratios in average graphs.

References

[1] M. Burkhart, P. Rickenbach, R. Wattenhofer, and A. Zollinger. Does topology control reduce interference. *In Proc. Mobihoc*, 2004.

[2] B. Carp. Geographic routing for sensor networks. *Ph.D thesis, Harvard University*, 2000.

[3] O. Dousse, P. Thiran, and M. Hasler. Connectivity in ad-hoc and hybrid networks. *In Proc. Infocom*, 2002.

[4] L. Hu. Topoogy control for multihop packet radio networks. *IEEE Transaction on Communication*, 10(41), October 1993.

[5] L. Jia, R. Rajaraman, and C. Scheidler. On local algorithms for topology control and routing in ad hoc networks. *In Proc. SPAA*, 2003.

[6] L. Li, H. Halpern, V. Bahl, Y. Wang, and R. Wattenhofer. Analysis of a cone-based distributed topology control algorithms for wireless multihop networks. *In Proc. PODC*, 2001.

[7] X. Li. Approximate mst for udg locally. *In Proc. COCOON*, 2003.

[8] X. Li, G. Calinescu, and P. Wan. Distributed construction of planar spanner and routing for ad hoc wireless networks. *In Proc. Infocom*, 3, 2002.

[9] X. Li, P. Wan, Y. Yang, and O. Frieder. Sparse power effi cient topology for wireless networks. *In Proc. ICCC*, 2001.

[10] X. Li, P. Wan, Y. Yang, and O. Frieder. Localized low weight graph and its applications in wireless ad hoc networks. *In Proc. Infocom*, 2004.

[11] K. Lillis and S. Pemmaraju. Topology control with limited geometric information. *In Proc. OPODIS*, 2005.

[12] R. Prakash. Undiectional links prove costly in wireless ad-hoc networks. *In Proc. DIAL-M*, 1999.

[13] R. Ramanathan and R. Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. *In Proc. Infocom*, 2000.

[14] V. Rodoplu and T. Meng. Minimum energy mobile wireless networks. *IEEE J. Selected Areas in Communication*, 8(17), August 1999.

[15] Y. Wang and X. Li. Geometric spanners for wireless ad hoc networks. *In Proc. ICDCS*, 2002.

[16] Y. Wang and X. Li. Localized construction of bounded degree planar spanner. *In Proc. DIALM-POMC*, 2003.

[17] Y. Wang, X. Li, and O. Frieder. Distributed spanner with bounded degree for wireless networks. *International Journal of Computer Foundations of Computer Science*, 2(14), 2003.

[18] R. Wattenhofer, L. Li, P. Bahl, and Y. Wang. Distributed topology control for power effi cient operation in multihop wireless networks. *In Proc. Infocom*, 2001.

[19] R. Wattenhofer and A. Zollinger. Xtc: A practical topology control algorithm for ad hoc networks. *In Proc. WMAN*, 2004.

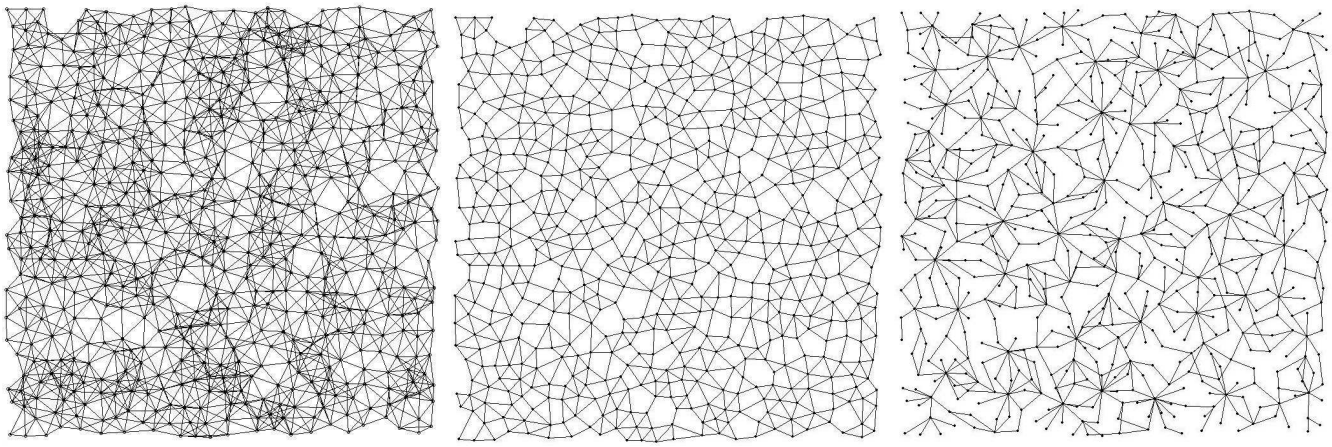

Figure 3. The unit disk graph (left), the Gabriel graph (center) and G_{TC} of 600 nodes placed randomly and uniformly are shown where the minimum distance between any two nodes is at least 0.4 unit.

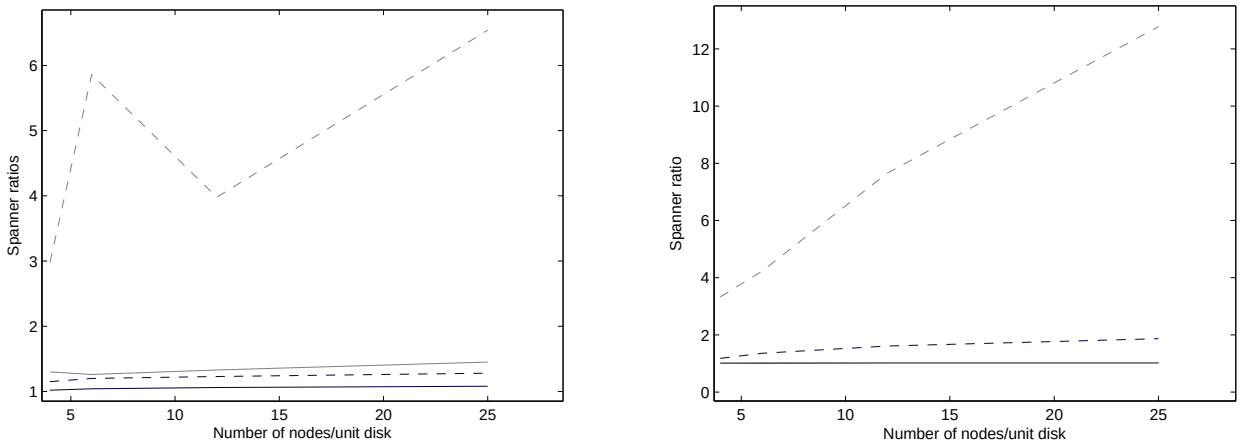

Figure 4. Spanner ratios of GG and G_{TC} w.r.t the Euclidean metric (left). The solid (resp. the dashed) line represents the spanner ratio of GG (resp. G_{TC}). Mean values are plotted in black and max values in gray. Spanner ratios of GG and G_{TC} w.r.t the energy metric is shown (right). The solid (resp. the dashed) line represents the energy-spanner ratio of GG (resp. G_{TC}). Mean values are plotted in black and max values in gray. Since GG contains an energy-minimal path between any pair of nodes its energy-spanner ratio is one (as shown in solid line). So its max and mean values coincide.