Local Construction of Bounded-Degree Network Topologies Using Only Neighborhood Information

Erdem Koyuncu* and Hamid Jafarkhani†

*Department of Electrical and Computer Engineering, University of Illinois at Chicago
†Center for Pervasive Communications and Computing, University of California, Irvine

Abstract—We consider ad-hoc networks consisting of \( n \) wireless nodes that are located on the plane. Any two given nodes are called neighbors if they are located within a certain distance (communication range) from one another. A given node can be directly connected to any one of its neighbors and picks its connections according to a unique topology control algorithm that is available at every node. Given that each node knows only the indices (unique identification numbers) of its one- and two-hop neighbors, we identify an algorithm that preserves connectivity and can operate without the need of any synchronization among nodes. Moreover, the algorithm results in a sparse graph with at most \( 5n \) edges and a maximum node degree of \( 10 \). Existing algorithms with the same promises further require neighbor distance and/or direction information at each node. We also evaluate the performance of our algorithm for random networks. In this case, our algorithm provides an asymptotically connected network with \( n(1 + o(1)) \) edges with a degree less than or equal to \( 6 \) for \( 1 - o(1) \) fraction of the nodes. Numerical results confirm our analytical findings.

I. INTRODUCTION

Topology control is commonly used in ad-hoc wireless networks to reduce interference, provide energy-efficient transmission, enable low-complexity routing, and so on [1]–[3]. In this paper, we study the problem of topology control over plane networks with the disk-connectivity model. Specifically, we consider networks consisting of \( n \) nodes that are indexed (and uniquely identified) by the natural numbers \( 1, \ldots, n \) with locations \( x_1, \ldots, x_n \in \mathbb{R}^2 \). A given node may only be directly connected to any other neighboring node that is within a certain communication range \( R > 0 \) in a bidirectional manner. As an example, a network consisting of 10 nodes with no connections together with the communication range of Node 7 is shown in Fig. 1(a). Node 7 can be directly connected to any one of the nodes in its neighbor set \( \{1, 3, 5, 6\} \).

A special case is when all nodes within communication range are directly connected [4], which results in what we call the Gilbert graph \((\mathcal{V}, g(\mathcal{V}))\) with

\[
g(\mathcal{W}) \triangleq \{(i, j) : i, j \in \mathcal{W}, i < j, |x_i - x_j| \leq R\}, \quad \mathcal{W} \subset \mathcal{V}.
\]

We note that Gilbert graphs are also often called unit-disk graphs whenever \( R = 1 \) or with an appropriate normalization of node locations. As an example, Fig. 1(b) shows the Gilbert graph corresponding to the setup in Fig. 1(a).

The primary goal of a topology control is then to provide a “good” spanning subgraph \((\mathcal{V}, \mathcal{E})\) of the Gilbert graph \((\mathcal{V}, g(\mathcal{V}))\). In this context, it is usually agreed upon that a good topology \((\mathcal{V}, \mathcal{E})\) should satisfy the following properties:

1) **Connectivity:** The network \((\mathcal{V}, \mathcal{E})\) is connected if there is a path between any two distinct nodes in \( \mathcal{V} \). It is clearly desirable to have a connected network so that information from one node may be conveyed to another (possibly through multiple hops) even if these two nodes are not directly connected.

2) **Sparseness:** The network \((\mathcal{V}, \mathcal{E})\) is called a sparse network if \( |\mathcal{E}| \leq cn \) for some constant \( c \geq 0 \). A sparse network is desirable as the computational complexity of routing grows with the number of edges in the network.

3) **Constant Maximum Degree:** The degree of a Node \( i \in \mathcal{V} \) is the number of nodes that are directly connected to Node \( i \). The existence of nodes with high degrees is not desirable in wireless networks due to several practical issues such as radio interference [5]–[7]. In fact, in practice, a given wireless node can be connected to at most a finite number of nodes at any given time, merely due to the fact that there can be at most a finite number of non-interfering frequency bands. In some cases, physical limitations of wireless devices themselves necessitate degree restrictions. Also, several standards
such as Bluetooth have “built-in” node degree constraints [8]. It is thus desirable that the degree of every node in \((V,E)\) is no more than a constant \(d \geq 0\) that is independent of \(n\).

4) Localized Construction: For large networks, it is unreasonable to expect the topology to be generated and imposed upon by a decision center that has global knowledge on the nodes’ physical locations and identities. Instead, the topology should ideally be generated locally in a distributed fashion with every node picking its own connections using as little information from its neighboring nodes as possible.

The design of practical topology control methods that satisfy the properties above have been a major avenue of research in the field of networking. Next, we provide an overview of some of the relevant literature together with our main contribution.

A. Related Work and Main Contribution

Structured graphs have been a source of inspiration for many of the studies on topology control. For example, many algorithms have been inspired by Delaunay triangulations [9]–[13], Gabriel graphs [14]–[16], the minimum spanning tree [17], [18], Yao graphs [14]–[16], [19], relative neighborhood graphs [15], [17], [20], or maximal independent sets [21]. We refer to [1]–[3] for a general detailed treatment of topology control including other algorithms.

The availability of node geographical information (in the form of direction, distance, or both) has been a common assumption in all the above works on topology control. The acquisition and communication of geographical information, especially exact geographical information, are however both non-trivial tasks in practice. Hence, some works have considered scenarios where each node has limited information about its neighbors. A notable algorithm is the XTC algorithm in [22], where each node is assumed to only know its distance to its neighbors as well as a certain ordering of its one- and two-hop neighbors. The XTC algorithm can provide a connected network with constant maximum degree. Another example is the CBTC algorithm in [19] which can operate with neighbor direction information at the nodes.

Ultimately, it is desirable to drop the requirement of geographical information entirely and focus on algorithms that can operate only with neighborhood information. Most of the works in this context have focused on achieving sparse almost sure connectivity instead of preserving connectivity whenever possible; see e.g. the \(k\)-Neigh protocol of [23] that is based on [24], or the random Bluetooth networks in [25]. These works do not consider node degree restrictions. On the other hand, an XTC-like algorithm that does not rely on distance information has been proposed in [26], but it can only preserve connectivity without any guarantees on sparseness or node degrees.

Another approach to position-unaware topology control is the construction and utilization of the connected dominating set (CDS) [27]–[32] of the network. It has been shown in [30] that by using only neighborhood information, one can construct connected sparse topologies via a minimum or close-to-minimum CDS. It is not clear, however, how to obtain a degree-bounded topology using the idea of a CDS with neighborhood information only. For example, [13], [30], [31] require extra position information at each node to obtain a CDS-based topologies with bounded node degrees.

Position-unaware topology control has also been studied for Bluetooth scatternet formation [8] with several proposed algorithms such as BlueStars [33], BlueMesh [34], BlueMIS [35], and BSF-UED [36]. Some of these algorithms can provide degree-bounded topologies, but the degree bound holds for only the master nodes of the network and not for all the nodes of the network. In this context, construction of network topologies with a constant degree bound at every node and without position information at nodes has been described [35] as “an interesting and major open problem in the area.” In this paper, we present a solution to this open problem.

B. Organization

The rest of the paper is organized as follows: In Section I-C, we introduce the notation and conventions that will be used throughout the paper. In Section II, we present our topology control algorithm and formally prove its properties. In Section III, we present an average case analysis of our algorithm. In Section IV, we present a numerical evaluation of our algorithm. In Section V, we draw the main conclusions.

C. Notation and Conventions

Given \(i,j \in V\) with \(i \neq j\), we say that Nodes \(i\) and \(j\) are two neighboring nodes, or simply neighbors if \(|x_i - x_j| \leq R\). Throughout the paper, we will only consider simple graphs (i.e., undirected graphs with no self-loops or multiple edges) of the form \((W,F)\) with \(W \subset V\) and \(F \subset g(W)\). Given any such graph/network \((W,F)\), and any two indices/nodes \(i,j \in W\) with \(i \neq j\), we say that Nodes \(i\) and \(j\) are directly connected in \((W,F)\) if \((i,j) \in F\). Note that \((i,j)\) and \((j,i)\) will always represent the same edge.

A path \(p \triangleq (p_1, \ldots, p_{|p|})\) in the graph \((W,F)\) is a vector of distinct elements of \(W\) such that \(|p| \geq 2\) and \((p_i,p_{i+1}) \in F\), \(i \in \{1, \ldots, |P| - 1\}\). We say that Nodes \(i\) and \(j\) are path-connected in \((W,F)\) if there is a path \(p\) in \((W,F)\) with \(p_1 = i\) and \(p_{|p|} = j\). A network \((W,F)\) is called connected if there is a path in \((W,F)\) between any two distinct nodes in \(W\). By definition, any two nodes that are directly connected are also path-connected.

A connected component of \((W,F)\) is a connected subgraph \((W',F')\) of \((W,F)\) such that (i) for any \(i \in W'\) and \(j \in W - W'\), Nodes \(i\) and \(j\) are not path-connected in \((W,F)\), and (ii) for any \(i,j \in W'\), we have \((i,j) \in F \implies (i,j) \in F'\).

II. The Algorithm

In this section, we present our algorithm that preserves connectivity and results in a sparse network with at most \(5n\) edges and a maximum node degree of 10. The setup in which the algorithm operates is as follows: Initially, we consider a network without any connections. The unique algorithm will be available to every node, and when “run,” will directly connect its host node (i.e., the node that is running the algorithm) to a certain subset of its host’s neighboring nodes. All the
connections initiated by the algorithm will be bidirectional. Running the algorithm at every node exactly once will result in the topology with the aforementioned properties. Nodes will be able to run the algorithm in an arbitrary order, or simultaneously in a completely asynchronous fashion.

Let us now present the algorithm itself. A key definition we need is the notion of a lesser neighborhood of a node. For any \( i \in \mathcal{V} \), we define the lesser neighborhood of Node \( i \) as

\[
N_i \triangleq \{ j : j \in \mathcal{V}, j < i, |x_i - x_j| \leq R \}.
\]

To put it into words, the lesser neighborhood of Node \( i \) are neighbors of Node \( i \) with indices that are less than \( i \).

Consider now the Gilbert graph \((\mathcal{N}_i, g(\mathcal{N}_i))\) generated by the lesser neighborhood of Node \( i \). Let \( J_i \) denote the number of connected components of \((\mathcal{N}_i, g(\mathcal{N}_i))\). Since each connected component of a Gilbert graph is necessarily also a Gilbert graph, we can list the connected components of \((\mathcal{N}_i, g(\mathcal{N}_i))\) as \( N_{ij}, g(N_{ij}) \), \( j = 1, \ldots, J_i \), where \( N_{ij} \) are mutually disjoint subsets of \( \mathcal{N}_i \) with \( \bigcup_{j=1}^{J_i} N_{ij} = \mathcal{N}_i \). Our algorithm (at Node \( i \)) is then to

"Connect to all nodes in the set \( \{ \max N_{ij} : 1 \leq j \leq J_i \} \)," where \( \max N_{ij} \) is the largest element of \( N_{ij} \). We have previously mentioned that the nodes may run the algorithm in arbitrary order as long as each node runs the algorithm exactly once. In fact, it is easily observed that the order in which the nodes run the algorithm does not affect the final topology as long as each node runs the algorithm at least once. All the different possibilities in this context will lead to the same final topology that we shall refer to as \((\mathcal{V}, \mathcal{A})\).

We shall shortly see the algorithm in action for an example realization of node locations. Let us first discuss the practicalities associated with the algorithm.

A. Practicalities

For the algorithm to work correctly, Node \( i \) of the network only has to know its lesser neighbors \( N_i \) and the lesser neighbors \( N_j \) of each one of its lesser neighbors \( j \in N_i \). We weaken this statement by saying that each node only has to know its neighbors and the neighbors of its neighbors (i.e. its one- and two-hop neighbors.). One way to provide this information to each node may be via the following simple two-step protocol. Before nodes even start running the algorithm, as a first step, each node may broadcast a "Hello" message (together with the index information) so that each node acquires the knowledge of its neighboring nodes. Each node may then broadcast the indices of its neighbors so that each node can also acquire the indices of each one of its neighboring nodes.

We also note that in practice, a node may not carry any "index information," at least not necessarily in the form of a natural number ranging from 1 to \( n \). Instead, each node may have a unique identification number (or a unique address) that can be used for indexing purposes. These identification numbers can be ordered, for example, lexicographically. Instead of the natural numbers with their standard order, the same algorithm can then operate over the node identification numbers with their lexicographical order. Hence, the (likely) possibility of “unnatural” node indices does not affect the way the algorithm operates or the final results.

B. An Example Run

We now demonstrate how the algorithm operates over the setup in Fig. 1(a). Suppose that initially there are no connections in the network. We illustrate how the algorithm (when it runs at Node 6) determines the direct connections to be initiated by Node 6. The lesser neighborhood of Node 6 is given by \( N_6 = \{1, 2, 3, 4, 5\} \), as shown in Fig. 2(a). Note that Node 6 itself and its “greater” neighbor Node 7 are not members of \( N_6 \). The next step for Node 6 is to calculate the Gilbert graph \((\mathcal{N}_6, g(\mathcal{N}_6))\) induced by \( N_6 \). This graph is as shown in Fig. 2(b) and has \( J_6 = 2 \) connected components \((N_{61}, g(N_{61}))\) and \((N_{62}, g(N_{62}))\) where \( N_{61} = \{1, 2, 5\} \) and \( N_{62} = \{3, 4\} \). Finally, we have \( \max N_{61} = 5 \) and \( \max N_{62} = 4 \), so that Node 6 will initiate a connection to Nodes 4 and 5. The corresponding two undirected edges that will be added to the initial graph will be \((4, 6)\) and \((5, 6)\).

In fact, running the algorithm at each node at least once results in the final network topology that we have previously shown in Fig. 1(c). For example, Node 1, having no lower neighbors \((J_1 = 0)\), will not initiate a connection to any other node. On the other hand, for Node 2, we have \( J_2 = 1 \) with \( N_{21} = \{1\} \), so that Node 2 will initiate a connection to Node 1. Hence, Node 1 in fact gets connected to Node 2, even though it is not Node 1 that initiates this connection.

C. Analysis of the Algorithm

We now analyze the graph \((\mathcal{V}, \mathcal{A})\) generated by the algorithm. We need the following lemma.

Lemma 1. In our algorithm, each node initiates at most 5 connections. In other words, \( J_i \leq 5 \) for any \( i \in \mathcal{V} \) (and for any given realization of node locations.).

Proof. Suppose that a given Node \( i \) initiates connections to both Nodes \( j_1 \) and \( j_2 \); see Fig. 3 for an illustration. The obvious neighborhood conditions \( |x_i - x_{j_1}| \leq R \) and \( |x_i - x_{j_2}| \leq R \) hold. By the design of the algorithm, we also have \( |x_{j_1} - x_{j_2}| > R \) (As otherwise, if \( |x_{j_1} - x_{j_2}| \leq R \), Nodes \( j_1 \) and \( j_2 \) would belong to the same connected component, say \( N_{\ell} \) for some \( \ell \in \{1, \ldots, J_i\} \) of the Gilbert graph generated by \( N_i \). Then, since Node \( i \) initiates a connection to only one of the nodes in \( N_{\ell} \), it would be absurd that it connects to both Nodes...
We have \( j_1, j_2 \). The three inequalities above imply that the edge \( x_{j_1}x_{j_2} \) is the longest edge of the triangle \( x_{j_1}x_{j_2}x_{j_3} \). This leads to the strict inequality \( \theta_1 > 60^\circ \). Using the same arguments, we obtain \( \theta_j > 60^\circ \), \( \forall j \in \{1, \ldots, j_1\} \). Now, assume the contrary to the statement of the lemma and suppose \( j_1 \geq 6 \). We have \( 360^\circ = \theta_1 + \cdots + \theta_{j_1} > J_160^\circ \geq 360^\circ \). This is a contradiction that proves the lemma.

\[
\begin{align*}
\end{align*}
\]

Fig. 3: Figure for the proof of Lemma 1.

The following theorem is then the main result of this paper.

**Theorem 1.** The graph \((V, A)\) is connected if and only if the Gilbert graph \((V, g(V))\) is connected. Moreover, \(|A| \leq 5n\) and the degree of each node in \((V, A)\) is no more than 10.

**Proof.** For the statement regarding connectivity, we only need to prove the “if” part with the “only if” part being trivial. Suppose \((V, g(V))\) is connected. Then, for any given two nodes in \(V\), there is a path in \((V, g(V))\) that connects these two nodes with each edge in the path consisting of two neighboring nodes. To show that \((V, A)\) is connected, it is thus sufficient to show that any two neighboring Nodes \(i\) and \(j\) are path-connected in \((V, A)\). To prove this, we may assume that \(i < j\) without loss of generality. First, note that if \(i = j - 1\), then, by design, Node \(j\) will initiate a connection to Node \(i\) and Nodes \(i\) and \(j\) will be path-connected. Otherwise, \(\exists k \in V\) with \(i < k < j\) such that (i) Node \(j\) initiates a connection to Node \(k\), and (ii) there is a path \(p \in (V, g(V))\) connecting Node \(k\) to Node \(i\) such that the index of each node in \(p\) is no more than \(k \leq j - 1\). It is then sufficient to show that any two distinct neighboring nodes that appear in \(p\) are path-connected in \((V, A)\). On the other hand, to prove this latter claim, it is sufficient to show that any two neighboring Nodes \(i'\) and \(j'\) with indices \(i' < j' \leq j - 1\) are path connected in \((V, A)\).

We have now established the following statement: Any two neighboring Nodes \(i\) and \(j\) with \(i < j\) are path-connected in \((V, A)\) if either \(i = j - 1\) or any two neighboring Nodes \(i'\) and \(j'\) with \(i' < j' \leq j - 1\) are path-connected in \((V, A)\). This statement describes a finite descent that immediately leads to the path-connectedness of Nodes \(i\) and \(j\). In fact, applying the statement on itself, any two neighboring Nodes \(i\) and \(j\) with \(i < j\) are path-connected in \((V, A)\) if either \(i = j - 1\), or \(i = j - 2\), or any two neighboring Nodes \(i'\) and \(j'\) with \(i' < j' \leq j - 2\) are path-connected in \((V, A)\). Hence, any two neighboring Nodes \(i\) and \(j\) with \(i < j\) are path-connected in \((V, A)\) if \(i = j - k\) for some natural number \(k\), which is clearly true. This concludes the proof of the claim on connectivity.

We now prove the rest of the claims. The claim \(|A| \leq 5n\) follows immediately as each node initiates at most 5 connections by Lemma 1. We now prove the degree bound. Let \(i \in V\). By design, a node with a lower index (< \(i\)) cannot initiate a connection to Node \(i\). On the other hand, Node \(i\) itself initiates at most 5 connections. To show a maximum node degree of 10, it is thus sufficient to show that there are at most 5 nodes with a higher index (> \(i\)) initiating a connection to Node \(i\). Assume the contrary and suppose there are 6 or more such nodes. Two of these nodes, say Nodes \(j\) and \(k\) (with \(j < k\) without loss of generality) should then be within communication range as well as being within range of Node \(i\). This implies \(\{i, j\} \subset N_{k\ell}\) for some \(\ell \in \{1, \ldots, 5\}\) with \(i \notin N_{k\ell}\) and \(j \notin N_{k\ell}\) for \(\ell' \neq \ell\). Since \(\max C_{k\ell} \geq \max \{i, j\} = j > i\), and \(i \notin N_{k\ell}\) for \(\ell' \neq \ell\), we have, in fact, \(\max N_{k\ell} \neq i\) for every \(\ell\). This contradicts the fact that Node \(k\) initiates a connection to Node \(i\) and thus proves the degree bound.

The degree bound of 10 is tight, i.e. there are certain realizations of node locations for which the resulting topology has a node with degree 10. A minimal example is with 11 nodes, \(x_6 = [0, 0]\), and \(x_7 = R[\cos \frac{i\pi}{11}, \sin \frac{i\pi}{11}]\), \(i \in \{1, \ldots, 11\} - \{6\}\). It does not seem to be as trivial, however, to find node locations that result in as much as 5\(n\) edges. In fact, as we show in the next section, the number of edges in most connected topologies that are generated by the algorithm is closer to \(n\) than 5\(n\). We will also show that the maximum node degree in most networks generated by the algorithm is 6.

As the final remark of this section, we note that Theorem 1 can be applied and extended to networks in higher (or lower) dimensions, i.e. for networks in \(\mathbb{R}^d\) for any \(d \geq 1\) with the same disk-connectivity model. In fact, let \(\mu_d\) denote the maximum number of points that can be packed in the unit ball in \(\mathbb{R}^d\) such that any two given distinct points are more than one unit apart. We have \(\mu_1 = 2\), \(\mu_2 = 5\) (as shown in Lemma 1), and it is not difficult to show that \(\mu_d\) is finite for any \(d \geq 3\). The exact same algorithm generates a connectivity-preserving topology with at most \(\mu_d n\) edges with a maximum node-degree of 2\(\mu_d\). In fact, similar results can be proved for connectivity models different than the disk model provided that the model admits a similar packing property.

**III. AVERAGE CASE EVALUATION**

Our algorithm, in worst cases, results in a topology with 5\(n\) edges and a maximum node degree of 10. However, numerical results suggest that for most realizations of node locations, the resulting topology is in fact much sparser and most nodes have a degree less than or equal to 6. We present an analytical justification of this phenomenon using random graphs.

We let the node locations \(x_1, \ldots, x_n\) be independent and uniformly distributed on \([0, 1]^2\) (instead of being arbitrary fixed points in \(\mathbb{R}^2\) as has been the case in previous sections). For any given fixed realization of node locations, we may simply run our algorithm to obtain one fixed topology corresponding to the given locations. The random nature of the node locations however means that the resulting topology \((V, A)\) will also be random. We are interested in the properties of the now-random graph \((V, A)\) (We use the same notation for fixed and random graphs as the difference will be obvious from the context).
For the random Gilbert graph \((\mathcal{V}, g(\mathcal{V}))\), Penrose [37] has shown that if \(R^2 = \frac{\log n + \alpha}{\pi n}\), then \(\Pr((\mathcal{V}, g(\mathcal{V})) \text{ is connected}) \rightarrow e^{-e^{-\alpha}}\) as \(n \rightarrow \infty\), where \(\Pr(\cdot)\) represents the probability of an event, \(\log(\cdot)\) is the natural logarithm, and \(e\) is the base of the natural logarithm. In particular, \(\Pr((\mathcal{V}, g(\mathcal{V})) \text{ is connected}) \rightarrow 1\) if and only if \(\alpha \rightarrow \infty\). We consider here random networks with communication radii just asymptotically above the connectivity threshold obtained by Penrose. Our main result is the following theorem, whose proof can be found in [38].

**Theorem 2.** Suppose \(R^2 / (\log n + \alpha) \rightarrow \infty\), and consider the random network \((\mathcal{V}, \mathcal{A})\). Then,

\[
\forall \epsilon > 0, \Pr(|\mathcal{A}| \geq (1 + \epsilon)n) \rightarrow 0.
\]

Moreover, let \(d_{\leq 6}\) denote the fraction of vertices in \((\mathcal{V}, \mathcal{A})\) with degree no more than 6. We have

\[
\forall \epsilon > 0, \Pr(d_{\leq 6} \geq 1 - \epsilon) \rightarrow 1.
\]

Hence, on average, our algorithm provides an extremely sparse connected network with \(n(1 + o(1))\) edges with a degree less than or equal to 6 for \(1 - o(1)\) fraction of the nodes.

**IV. Numerical Results**

We have run our algorithm on a network with \(n = 1000\) nodes and initially no connections. Nodes are located independently and uniformly on \([0,1]^2\). We have considered the choices \(\pi R^2 = \frac{N}{n}\) for \(N \in \{10, 20, 30\}\). The parameter \(N\) can be thought as a quantification of “node density” as any given node of the network then has roughly (ignoring the edge effects) \(N\) neighbors on average. Regarding our specific choices for \(N\), we note that the probabilities of connectivity for the associated Gilbert graphs are approximately 0.588, 0.994, and 0.999 for the choices \(N = 10\), \(N = 20\), and \(N = 30\), respectively. These values are obtained numerically. Hence, the three different choices for \(N\) represent the three different scenarios of “mostly-disconnected,” “usually connected,” and “almost-always connected” networks.

In Fig. 4, we show the cumulative distribution functions (CDFs) of the normalized number of edges \(\frac{|\mathcal{A}|}{n}\) for different \(N\). The beginning and the end of each graph correspond to the cases where the simulated CDF is exactly equal to 0 and 1. Thus, in all the node location realizations, we have simulated, the number of edges of the network never exceeded \(1.14n\) or went below \(1.02n\) for any choice of \(N\). These observations are in agreement with the inequality \(|\mathcal{A}| \leq 5n\) as proved by Theorem 1. In addition, the fact that the number of edges are very close to \(n\) for any \(N\) is in agreement with Theorem 2, where we proved that the algorithm usually generates topologies with \(n(1 + o(1))\) edges, especially when \(N\) is large. In fact, if \(N\) were infinite, all the nodes would be within the line topology with only the \(n\) edges \((1,2), (2,3), \ldots, (n-1,n)\).

In Fig. 5, we show the probability mass functions (PMFs) corresponding to the degree of a given node of the network for different \(N\). More specifically, let \(I\) have a uniform PMF on the set \(\{1, \ldots, n\}\). For a given \(N\), the corresponding PMF at a given degree \(d\) in Fig. 5 is then the probability that Node 1 has degree \(d\) in the final network topology. Also, Fig. 6 shows the PMFs of the maximum node degree of the network. According to Theorem 1, both PMFs should only take values on the set \(\{0, \ldots, 10\}\). In fact, for every \(N\), we have not observed a single realization of node locations where the maximum node degree is 8. Such realizations obviously exist (see Section II); the results of Figs. 5 and 6 rather suggest that they correspond to very rare events. We can also observe from Fig. 5 that almost all the nodes in the network have a degree of 6 or less, which is in agreement with Theorem 2. Also, the fraction of nodes with degree 2 increases as \(N\) increases, and we expect it to approach to 1 as \(N \rightarrow \infty\) as a result of the aforementioned convergence to line topology.

**V. Conclusions**

We have studied the problem of topology control in wireless ad-hoc networks consisting of \(n\) nodes that are located on the plane. We have considered the disk-connectivity model with the following fundamental question: How to generate networking topologies with the practically-relevant graph-theoretical properties such as connectivity or degree-boundedness?

We have observed that previous work require detailed geographical/locational information at each node to achieve these
desired properties. We have shown that, in fact, a sufficient condition to achieve degree-bounded connectivity is just for each node to know the identification numbers of its one- and two-hop neighbors - no distance/directional information is needed whatsoever. Our corresponding local topology control algorithm guarantees a connected network with $5n$ edges and a maximum node degree of 10. We have shown that for most networks, these numbers are in fact much lower.

ACKNOWLEDGMENT

This work was supported in part by the DARPA GRAPHS program Award N66001-14-1-4061.

REFERENCES


