A Distributed Topology Control Algorithm in the Presence of Multipath Propagation

Harish Sethu and Thomas Gerety
Department of Electrical and Computer Engineering
Drexel University
Philadelphia, PA 19104-2875
Email: {sethu,tg56}@drexel.edu

Abstract

Each node in a wireless multihop network can adjust the power at which it makes its transmissions and thus change the topology of the network by choosing the neighbors with which it directly communicates. Many previous algorithms for distributed topology control have assumed an ability at each node to deduce some spatial information such as the direction and the distance of its neighbor nodes with respect to itself. However, such a deduction of spatial information cannot be relied upon in a multipath environment where different paths of a signal may have different loss characteristics and none of the paths may be line-of-sight. In this report, we present Step Topology Control (STC), a simple distributed topology control algorithm which reduces power consumption while preserving the connectivity of a heterogeneous sensor network in a multipath environment. The STC algorithm avoids the use of GPS devices and also makes no assumptions about the distance and direction between neighboring nodes. We show that the STC algorithm achieves the same or better order of communication and computational complexity in comparison to other known algorithms that also preserve connectivity and can be similarly adapted to a multipath environment. We also present a detailed simulation-based comparative analysis of the energy savings and interference reduction achieved by the algorithms. The results show that the STC algorithm performs better than other algorithms in uniform wireless environments and especially better in the presence of radio irregularities and multipath propagation. This document also reports a correction to [1].

I. INTRODUCTION

In a multi-hop wireless sensor network, a node communicates with another node across one or more consecutive wireless links with messages possibly passing through intermediate nodes. The topology of such a network can be viewed as a graph with an edge connecting any pair of nodes that can communicate with each other directly without going through any intermediate nodes. Each node in such a network can choose its own neighbors and thus control the topology by changing the power at which it makes its transmissions or, in the case of nodes capable of directional transmissions, by also changing the set of directions in which it will allow transmissions. The goal of such topology control is the design of algorithms that each node can execute in a distributed manner in order to reduce energy consumption, maintain connectivity, and increase network lifetime and/or capacity.

In recent years, a large number of topology control algorithms have been proposed and studied for a diversity of goals [2]. Early work on topology control assumed that accurate location information about its neighbors will be available to the nodes, such as through the use of GPS devices [3]–[7]. This assumption adds to the expense of the nodes and also results in high delays due to the acquiring and tracking of satellite signals. Also, one cannot rely on GPS in many real application environments such as inside buildings or thick forests. Some other topology control protocols that preserve connectivity rely on the more likely ability of a node to estimate the distance and direction to its neighbors. For example, in the cone-based distributed topology control (CBTC) algorithms, a node u transmits with the minimum power $p_{u,\alpha}$ required to ensure that there is some node it can reach within every cone of degree α around u [8]. Assuming a specific loss propagation model, the Euclidean distance to a neighbor can be deduced with knowledge of the power at which a transmission is made by a neighbor and the power at which the signal is received. The direction of a neighbor with respect to itself can be deduced from the angle of arrival of a signal.

Wireless communication, however, is often characterized by the phenomenon of multipath propagation wherein a signal reaches the receiving antenna via two or more paths [9]. In addition, there are several other kinds of radio irregularities that have an impact on the topology control algorithms [10]. The different paths, with differences in delay, attenuation, and phase shift, make it difficult for the receiving node to deduce either its distance from the sender or the direction of the sender. In this report, we focus on the design of topology control algorithms that work in the presence of multipath propagation and therefore, can be employed in conjunction with strategies such as MIMO that can exploit spatial diversity. More specifically, we focus on connectivity-preserving algorithms that make *no* spatial assumptions (i.e., no assumptions about the availability of GPS devices in nodes and also no assumptions on the ability of a receiving node to deduce either the distance or the direction of the sender). Besides accommodating the environmental causes of radio irregularities, we seek to also meet the requirements of a heterogeneous sensor network where (a) different nodes may have different maximum transmission powers, and (b) variations in node/antenna configurations may lead to different reception thresholds in different directions.

A. Problem Statement

Assume that each node $u \in V$ is associated with a certain maximum power P_u with which it is capable of making an omnidirectional transmission (for ease of discussion, we use omni-directional transmissions but our problem statement and the proposed algorithm can be readily adapted for directional transmissions). Note that we allow P_u to be different for different nodes, allowing a heterogeneous sensor network environment as in [11]. Consider the nodes in the network as vertices of a directed graph $G_{\max} = (V, E_{\max})$ in which nodes u and v are connected by a directed edge from u to v if and only if (a) an omni-directional transmission from u at its maximum power P_u can directly reach v, and (b) an omni-directional transmission from v at its maximum power P_v can directly reach u. Note that if $(u, v) \in E_{\max}$, then $(v, u) \in E_{\max}$. Many widely deployed MAC and address resolution protocols in wireless networks not only assume bidirectional links but also assume two-way handshakes and acknowledgments [12]. Therefore, with bidirectional communication assumed between directly communicating nodes, $G_{\max}$ represents a realistic communication topology at maximum node powers. The goal of topology control in this report is to find a new directed graph $G = (V, E)$ such that:

- $E \subset E_{\max}$.
- If $(u, v) \in E$, then $(v, u) \in E$, as is generally expected by MAC layer protocols.
- If there exists a path between u and v in $G_{\max}$, then there also exists a path between u and v in G.
- The power required in an omni-directional transmission by a node u to reach all of its neighbors in G is less than or equal to its maximum power P_u.

The above requirements preserve the connectivity of the network while seeking to reduce the energy consumption and interference.

B. Related Work

Let $C(u, v)$ denote the minimum energy cost of a successful transmission from u to v. A topology control algorithm that minimizes energy consumption will remove an edge (u, v) if and only if there exists a path between u and v through an intermediate set of nodes $n_1, n_2 \dots n_k$ such that $C(u, n_1) + C(n_1, n_2) + \dots C(n_k, v) < C(u, v)$. Accomplishing this requires a significant exchange of information between nodes and in such a case, the topology control algorithm is indistinguishable from a routing algorithm. As a result, a number of distributed topology control algorithms have been proposed where nodes rely on lesser exchange of information between neighbors [2]. In this subsection, we focus on the subset of these protocols that can be adapted for a wireless environment with the phenomenon of multipath propagation.

The KNEIGH protocol is based on determining the number of neighbors that each node should have in order to achieve full connectivity with a high probability [13]. The XTC protocol is a generalization of the KNEIGH protocol in which each node also considers a notion of 'link quality' in its communication to its neighbors [14]. These protocols, however, do not guarantee connectivity even though they do achieve connectivity with a high probability.

The Small Minimum-Energy Communication Network (SMECN) protocol [5] seeks to achieve a lower energy cost while guaranteeing connectivity by removing an edge (u, v) if and only if there exists a path $u \rightarrow n \rightarrow v$ such that $C(u, n) + C(n, v) < C(u, v)$. As proposed in [5], the protocol requires the use of GPS devices but the same results can be accomplished if each node exchanges information with each of its neighbors regarding the energy costs of reaching all of its neighbors. When used with some of the widely available routing algorithm implementations such as AODV or DSR that base their decisions on the number of hops in a path rather than the total energy cost of the path, the SMECN protocol does not necessarily result in a significant reduction

```
Function STC at node (u):
    G ← (V, φ) /* directed graph with no edges */
    Compile outTupleList(u) and inTupleList(u)
    Broadcast outTupleList(u) and inTupleList(u) at maximum power P_u
    Receive outTupleList(n) and inTupleList(n) from each neighbor n in G_max
    Compute fPairOfPaths(n) for each n two or fewer hops away from u
    Compute bPairOfPaths(n) for each n two or fewer hops away from u
    Sort outTupleList(u)
    k ← degree of u in G_max
    do k − 1 times
        t(u, v) ← the largest tuple in outTupleList(u)
        remove t(u, v) from outTupleList(u)
        vSet = {i | t(i, v) ∈ inTupleList(v), t(i, v) < t(u, v)}
        NoForwardPath = True
        NoBackwardPath = True
        for n ∈ vSet
            p = the first path in fPairOfPaths(n) without v
            if max{r|r ∈ p} < t(u, v)
                NoForwardPath = False
                break (out of for loop)
            end if
        end for
        vSet = {i | t(v, i) ∈ outTupleList(v), t(v, i) < t(v, u)}
        for n ∈ vSet
            p = the first path in bPairOfPaths(n) without v
            if max{r|r ∈ p} < t(v, u)
                NoBackwardPath = False
                break (out of for loop)
            end if
        end for
        if NoForwardPath or NoBackwardPath
            add a directed edge (u, v) to G
        end if
    end do
```

Fig. 1. Pseudo-code description of the Step Topology Control algorithm at node u.

in energy consumption. As will be shown in Section IV, the energy savings achieved by SMECN is not significantly greater even if the routing algorithm were to choose a minimum-energy path.

The Directed Relative Neighborhood Graph (DRNG) protocol [11] removes an edge (u, v) if and only if there exists a path $u \to n \to v$ such that $\max\{C(u, n), C(n, v)\} < C(u, v)$. In the Directed Local Spanning Subgraph (DLSS) protocol [11], each node creates a local spanning tree from the subgraph induced by itself and its neighbors. A node u retains the edge (u, v) in the topology-controlled graph if and only if the edge (u, v) exists in the local spanning tree generated at node u. Any edge that is removed by DRNG is also removed by DLSS and therefore, DLSS achieves a sparser graph than DRNG. The fewer edges in DLSS often leads to a lower energy consumption, but it can sometimes lead to longer paths and therefore higher energy consumption than DRNG.

Among other attempts to accommodate the irregularities of a wireless radio environment in topology control are some that allow for uncertainties in whether or not a nearby node is reachable if the distance to that node is above a certain threshold [15], [16]. However, these algorithms assume that each node can know the distances to other nearby nodes, something that cannot be relied upon in a multipath environment. Some other works have also considered realistic wireless models but they have only presented centralized algorithms [17].

C. Contributions

In this report, we present the Step Topology Control (STC) algorithm in which a node u removes an edge (u, v) if and only if there exists:

- a path with three or fewer hops from u to v such that the energy cost across each hop is less than $C(u, v)$.
- a path with three or fewer hops from v to u such that the energy cost across each hop is less than $C(v, u)$.

The STC algorithm relies on each node exchanging information with each of its neighbors regarding the energy costs of communication to all of its neighbors.

The STC algorithm may be seen as an extension of the DRNG algorithm to allow a search for three-hop paths and to ensure bidirectional communication between directly communicating nodes. However, our implementation of the algorithm ensures that the order of communication and the computational complexity do not increase despite a search for three-hop paths. This search for three-hop paths makes only a small impact on performance when the radio environment is uniform across the network. However, it makes a *significant* impact in the presence of multipath propagation or other irregularities in the radio environment. In such environments, therefore, the STC algorithm achieves a much lower power consumption and interference than other topology control algorithms in spite of maintaining the same or better order of communication and computational complexity as other algorithms. We also show that this improved performance of the STC algorithm exists largely independent of the size of the network.

Section II presents the STC algorithm along with a pseudo-code description. Section III presents an analysis of the communication and computational complexity of the STC algorithm and presents a comparative analysis. Section IV presents several simulation-based results that provide a thorough comparison of the energy consumption properties of the STC algorithm and other existing topology control strategies that can be adapted to a multipath environment. In particular, we examine the topology control algorithms in environments with both constant and varied path loss exponents and study their scalability in performance as the number of nodes increases. Section V concludes the report with a summary of its findings and future research directions. A proof of the relationship between the STC algorithm and a CBTC algorithm with all applicable optimizations is presented in the Appendix.

II. Step Topology Control

Denote by $P_{\min}(u, v)$, the minimum power necessary for a transmission from u to reach v, otherwise known as *transmission power threshold*. We allow that $P_{min}(u, v)$ is not necessarily the same as $P_{\min}(v,u)$. The basic idea behind the Step Topology Control (STC) algorithm is to find both forward and backward paths with three or fewer hops between u and v such that each hop requires a lower energy cost than that required for an equivalent direct transmission between u and v. If such multihop paths exist between u and v, node u drops the edge (u,v) from the directed graph it generates and node v similarly drops the edge (v,u).

The STC algorithm relies on being able to uniquely order the energy costs of transmissions across nodes. In practice, since the power levels at which transmissions are made may take on only certain discrete values, we add additional identifiers to permit a unique ordering. We assume that each node u is uniquely identified by an integer ID_u. For each ordered pair of nodes u and v, we associate an ordered tuple $t(u,v)=(t_1,t_2,t_3)$, where $t_1=P_{\min}(u,v)$, $t_2=\mathrm{ID}_u$, and $t_3=\mathrm{ID}_v$. We say that $(t_1,t_2,t_3)<(t_1',t_2',t_3')$ if and only if (1) $t_1<t_1'$, or (2) $t_1=t_1'$ and $t_2<t_2'$, or (3) $t_1=t_1'$, $t_2=t_2'$, and $t_3<t_3'$. For the sake of completeness, we define $t(u,u)<t(x,y)$ for any u and $x\neq y$ since a transmission to itself should cost less energy than a transmission to another node. Note that $t(u,v)$ and $t(v,u)$ are strictly ordered by the above lexicographic rule and not equal even if the minimum power required for transmission between u and v is the same in either direction. We call $t(u,v)$ a transmission tuple. A path p in the graph consists of an ordered sequence of transmission tuples. Denote by *maxTuple*(p) the largest tuple in path p.

Consider any two nodes u and v such that $(u,v)\in E_{\max}$. We assume that u can determine the minimum power necessary for its transmission to reach v as well as the minimum power required for a transmission from v to reach itself. This is accomplished by transmitting beacon messages at increasing powers and noting the power at which each neighbor is first discovered. Each beacon message carries within it the power at which it is transmitted so that the discovered neighbor may also note the minimum power necessary for it to be reached by a neighboring node. Each node u can thus compile two lists of transmission tuples: *outTupleList*(u) containing $t(u,v)$ for all $(u,v)\in E_{\max}$, and *inTupleList*(u) containing $t(v,u)$ for all $(v,u)\in E_{\max}$. This process of exchanging power level information is feasible in practice and is part of many proposed energy-aware MAC layer protocols as well as topology control algorithms such as DLSS.

Figure 1 presents a pseudo-code description of the STC algorithm. Once a node u compiles *outTupleList*(u) and *inTupleList*(u), it begins execution of the algorithm. Each node u first broadcasts both its *inTupleList* and *outTupleList* at its maximum power P_u to reach all of its neighbors (line 04). The node also collects the *inTupleLists* and *outTupleLists* from each of its neighbors (line 05).

With the information about the energy costs compiled as above, node u computes two forward paths (*fPairOfPaths*(n)) and two backward paths (*bPairOfPaths*(n)) for each node n that is reachable by two or fewer hops (lines 06–07). We describe below in greater detail the construction of the forward path data structures for node n (the construction of the backward pair of paths is similar). Let p_1 denote the first of the pair of paths in *fPairOfPaths*(n) and let p_2 denote the second. We choose p_1 and p_2 such that *maxTuple*$(p_1)<$ *maxTuple*$(p_2)<$ *maxTuple*(p) where p is any path of two or fewer hops from u to n other than p_1 and p_2. In lines 16–22, this data structure allows the node to quickly determine if there exists a path s of three or fewer hops between u and another node v such that *maxTuple*$(s)<t(u,v)$. The reason we need a pair of paths instead of just one path is because one of these paths may be through v and one would not choose to replace the edge (u,v) with a path that goes through v.

Node u then orders the tuples in its *outTupleList*(u) (line 08) and considers each of the edges (u, v) in reverse lexicographical order of the associated tuples $t(u, v)$, i.e., the neighbor that requires the largest power to be reached is considered first (line 11). As each neighbor v is processed, the corresponding tuple $t(u, v)$ is removed from *outTupleList*(u) (line 12).

To determine if an edge (u, v) should be removed, a node u looks for a forward path $u \to n_1 \to n_2 \to v$, where the nodes n_1 and n_2 may or may not be distinct (lines 13–22). The condition that the path should satisfy is $\max\{t(u, n_1), t(n_1, n_2), t(n_2, v)\} < t(u, v)$. The node similarly seeks to find a reverse path via nodes n_3 and n_4 (lines 23–33) such that $\max\{t(v, n_3), t(n_3, n_4), t(n_4, u)\} < t(v, u)$. Note that the number of distinct nodes among n_1, n_2, n_3 and n_4 may range anywhere between 1 and 4. If both forward and backward two- or three-hop paths are found satisfying the desired conditions on the tuples, the edge (u, v) is removed from the graph (lines 31–33).

To determine if there exists a forward path from u to v satisfying the desired conditions, node u first constructs the set *vSet* consisting of nodes i that are neighbors of v such that $t(i, v) < t(u, v)$ (line 13). Now, if there exists a path, p, of two or fewer hops from u to any node in *vSet* such that the path is not through v and *maxTuple*$(p) < t(u, v)$, then the condition for the forward path is satisfied. This is determined in lines 16–22 and the existence of a backward path is similarly determined in lines 24–30.

III. Comparative Analysis

In this section, we discuss the communication and computational complexity of the STC algorithm in comparison to other topology control algorithms that also preserve connectivity while being capable of operation in a multipath environment. We also discuss any provable relationships between STC and other algorithms regarding the set of edges removed from G_{max} by the algorithm.

A. Complexity

We consider the complexity of the following algorithms and present a comparison to the STC algorithm:

- Small Minimum Energy Communication Network (SMECN).
- Directed Relative Neighborhood Graph (DRNG).
- Directed Local Spanning Subgraph (DLSS).

Each of the above algorithms relies on each node first determining the energy cost to each of its neighbors. As explained in Section II, this can be accomplished by transmitting beacon messages at increasing powers and noting the power at which each neighbor is first discovered. When beacon messages carry the power at which they are being transmitted, each node can also learn the energy cost of the communication from each of its neighbors to itself. Depending on the granularity of power levels at which transmissions can be made, a variety of strategies may be employed in these steps to minimize interference and to maximize speed of convergence to the correct power values. The method of compilation of these energy costs is outside the scope of this report and we do not include it in our complexity analysis, especially since this step is common to all of the algorithms under consideration in this section. For the STC algorithm, for example, we consider its complexity at node u after the compilation of *outTupleList*(u) and *inTupleList*(u) is complete.

In the original graph, $G_{\max}$, on which a topology control algorithm is executed, the in-degree of a node is the same as its out-degree (since bidirectional communication is assumed) and therefore, the maximum out-degree (Δ^+) and the maximum in-degree (Δ^-) are identical. We define the node-degree of a node in $G_{\max}$ as the number of its neighbors that it communicates with, i.e., its out-degree or its in-degree. In the following, we denote the maximum node-degree of the original graph by $\Delta = \Delta^+ = \Delta^-$.

Theorem 3.1: The communication complexity of the STC algorithm is $O(\Delta^2)$.

Proof: For any node n, *inTupleList*(n) and the *outTupleList*(n) are each of length $O(\Delta)$. Broadcasting the lists (line 04 in Figure 1), therefore, is $O(\Delta)$ in communication complexity and receiving the lists from each of up to Δ neighbors (line 05) is $O(\Delta^2)$ in communication complexity. The overall communication complexity, therefore, is $O(\Delta^2)$. ■

Theorem 3.2: The computational complexity of the STC algorithm is $O(\Delta^2)$.

Proof: For each neighbor n' of u, one can process the entries in the *outTupleList*(n') to create the pair of paths in *fPairOfPaths*. The determination of whether or not a path to node n should be included in *fPairOfPaths* and whether it should be the first or the second of the pair of paths can be made in $O(1)$ time since it involves no more than two tuple comparisons. Let h denote the number of one-hop or two-hop paths starting from u. Since $h = O(\Delta^2)$, creating the *fPairOfPaths* will require a total time of $O(\Delta^2)$. Creating the *bPairOfPaths* will similarly require a total time of $O(\Delta^2)$.

Sorting of the *outTupleList*(u) in line 08 takes time $O(\Delta \log \Delta)$ since the size of the list is $O(\Delta)$.

We now show that the $2h$ pairs of paths created above can allow the inner `for` loops of lines 16–22 and 24–30 to complete in time $O(\Delta)$. It is readily verified that each iteration of the `for` loops completes in $O(1)$ time (lines 17–21 and 25–29). Since the size of the *vSet* is $O(\Delta)$, these inner `for` loops execute in $O(\Delta)$ time. The only other non-loop component inside the outer `do` loop between lines 10–34 that takes more than $O(1)$ time is the creation of the *vSet* itself, which takes $O(\Delta)$ time. Since the `do` loop iterates $k-1$ times where $k \leq \Delta$, the entire `do` loop completes in time $O(\Delta^2)$.

The overall computational complexity of the STC algorithm, therefore, is $O(\Delta^2)$.

■

Table I presents a comparison of the communication and computational complexities of topology control algorithms that preserve connectivity and which can be used in a multi-path environment. All involve a communication complexity of $O(\Delta^2)$ since they all require that each node collect a list of energy costs from each of its neighbors. In the computational complexity analysis of DLSS, we consider the number of nodes in the local graph as $O(\Delta)$ and the number of edges as $O(\Delta^2)$. The computation of the local minimum spanning tree, assuming Kruskal's algorithm [18], is $O(\Delta^2 \log \Delta)$.

B. Set of edges removed

As regards performance, the STC algorithm removes any edge from the original graph that is also removed by either SMECN or the DRNG algorithm. This is easily argued as follows. The SMECN algorithm removes an edge (u, v) if there exists a two-hop path from u to v through n such that $C(u, n) + C(n, v) < C(u, v)$. Therefore, we have $C(u, n) < C(u, v)$ and $C(n, v) < C(u, v)$ which would be the condition that will lead the DRNG algorithm to also

Algorithm	Communication Complexity	Computational Complexity
SMECN	$O(\Delta^2)$	$O(\Delta^2)$
DRNG	$O(\Delta^2)$	$O(\Delta^2)$
DLSS	$O(\Delta^2)$	$O(\Delta^2 \log \Delta)$
STC	$O(\Delta^2)$	$O(\Delta^2)$

TABLE I
A COMPARISON BETWEEN TOPOLOGY CONTROL ALGORITHMS

remove edge (u, v). The STC algorithm, being a simple extension of the DRNG algorithm, removes all edges that would be removed by DRNG. Thus, the STC algorithm will always yield a lower energy cost per transmission at any given node than either SMECN or DRNG. The STC algorithm does not always remove an edge that would be removed by DLSS. For a comparative analysis with the DLSS algorithm, we rely on the simulation results presented in the next section.

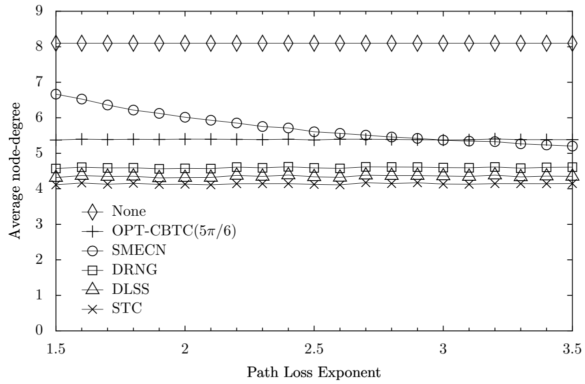
It is of interest to note that the STC algorithm is related to the OPT-CBTC($5\pi/6$) algorithm (which is the CBTC($5\pi/6$) algorithm with all applicable optimizations). The STC algorithm removes all edges that would be removed by OPT-CBTC($5\pi/6$). As a result of this relationship, proved in Appendix A, the STC algorithm exhibits some of the same angular properties as OPT-CBTC($5\pi/6$).

IV. SIMULATION RESULTS

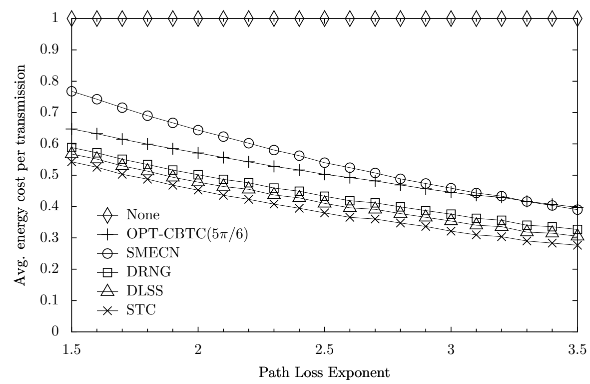
In this section, we present a simulation-based comparative study of the following distributed topology control algorithms:

- Step Topology Control (STC), the algorithm presented in this report.
- OPT-CBTC($5\pi/6$), as a representative CBTC algorithm and because of its relationship to the STC algorithm.
- Small Minimum Energy Communication Network (SMECN).
- Directed Relative Neighborhood Graph (DRNG).
- Directed Local Spanning Subgraph (DLSS).

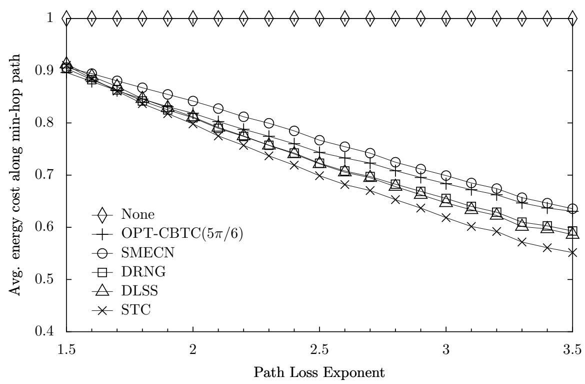
Our study uses three sets of simulation experiments in comparing the effectiveness of different topology control algorithms. In the first set of experiments, we assume that the path loss exponent is the same across the entire network and study the topology control algorithms for different values of the path loss exponent. In the second set of experiments, we assume that path loss exponents vary within a certain range and study the topology control algorithms for different sizes of the range. In the third set of experiments, we study the effectiveness of the algorithms as the number of sensor nodes increases. The networks used in our first two sets of experiments consist of 200 nodes located randomly in a unit square area. In the third set of experiments, we vary the number of nodes from 100 to 500 to study the scalability of the algorithms across a five-fold increase in the number of nodes. Each data point in this report represents an average of one hundred different randomly generated networks.



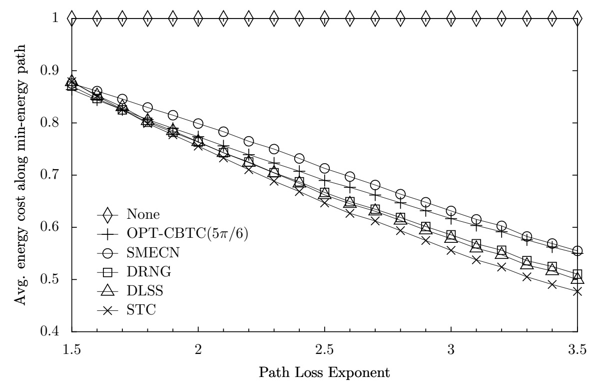
(a) Average node-degree.



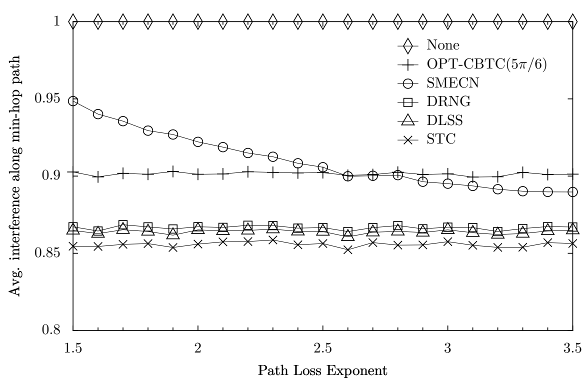
(b) Average $P_{T/H}(u)$ over all u.



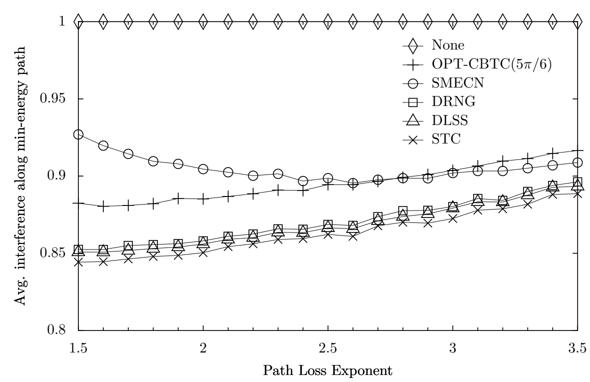
(c) $C_{T/H}(u \to v, \mathrm{H}ops)$ averaged over all pairs of nodes u and v.



(d) $C_{T/H}(u \to v, \mathrm{E}nergy)$ averaged over all pairs of nodes u and v.



(e) $I_{T/H}(u \to v, \mathrm{H}ops)$ averaged over all pairs of nodes u and v.



(f) $I_{T/H}(u \to v, \mathrm{E}nergy)$ averaged over all pairs of nodes u and v.

Fig. 2. Graphs showing the effectiveness of topology control algorithms when the path loss exponent is the same across the entire region of the network (in order to allow comparisons with the CBTC algorithms). The networks are generated with 200 randomly located nodes in a unit square area. Each data point represents an average of one hundred random networks.

A. Metrics

For each network, the baseline for our comparisons is the *initial* graph, H, defined as follows. Assume $P_u = P$ for all nodes u (recall that P_u denotes the maximum power with which a node u can transmit). Consider graph G generated by creating an edge (u, v) from u to v if and only if a transmission from u at power P can reach v and a transmission from v to u at power P can reach u. Let P_H denote the minimum value of P at which G is connected. The subgraph of G generated by each node transmitting at power P_H is the initial graph, H.

Given a graph, T', generated by the execution of the topology control algorithm, we define $P_{T'}(u)$ as the minimum power with which node u should make an omni-directional transmission so as to reach all of its neighbors in graph T'. Let $C_{T'}(u)$ denote the energy cost of a transmission by u at power $P_{T'}(u)$.

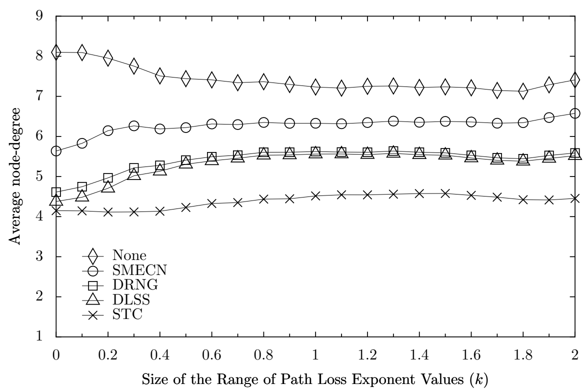
Given the graph T' above, we define a new graph T as follows: a node u is connected by an edge to v in T if and only if $C_{T'}(u) \geq C(u, v)$ or $C_{T'}(v) \geq C(v, u)$. Thus, given omni-directional transmissions, T represents the graph that is actually relevant for performance comparisons and especially, interference comparisons. T, as opposed to T', also enforces the ability for bidirectional transmissions between communicating nodes. We call T the *cover graph* generated by the topology control algorithm. All of our simulation results use the cover graph.

Define $P_T(u)$ as the minimum power with which node u should make an omni-directional transmission so as to reach all of its neighbors in the cover graph T. Let $C_T(u)$ denote the energy cost of a transmission by u at power $P_T(u)$. Note that $P_T(u) \geq P_{T'}(u)$ and $C_T(u) \geq C_{T'}(u)$. In our simulations, we use $P_T(u)$ as the power at which node u makes all its transmissions after the execution of a topology control algorithm generating cover graph T.

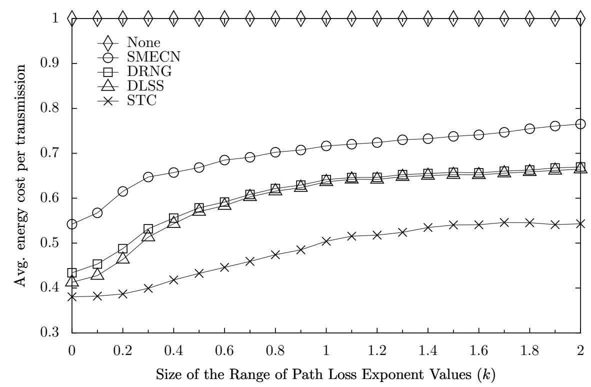
$P_T(u)/P_H$, denoted by $P_{T/H}(u)$, is the ratio of the power at which node u transmits after the execution of the topology control algorithm that generates cover graph T and the power at which it transmits in the initial graph, H. With this normalization to the energy costs in the initial graph, this ratio captures the energy savings per transmission due to the topology control algorithm.

Let $C_T(u \rightarrow v, \mathrm{H}ops)$ denote the sum of $C_T(i)$ for each transmitting node i in the minimum-hop path from u to v in cover graph T. $C_T(u \rightarrow v, \mathrm{H}ops)/C_H(u \rightarrow v, \mathrm{H}ops)$, denoted by $C_{T/H}(u \rightarrow v, \mathrm{H}ops)$, is the ratio of the energy cost along the minimum-hop path from u to v in cover graph T generated by the topology control algorithm and the corresponding cost along the minimum-hop path from u to v in the initial graph, H. This ratio captures the energy savings along a path due to the topology control algorithm. In our simulation experiments, we examine both the minimum-energy paths and the minimum-hop paths. The ratio for the minimum-energy paths is computed similarly as above and is denoted by $C_{T/H}(u \rightarrow v, \mathrm{E}nergy)$.

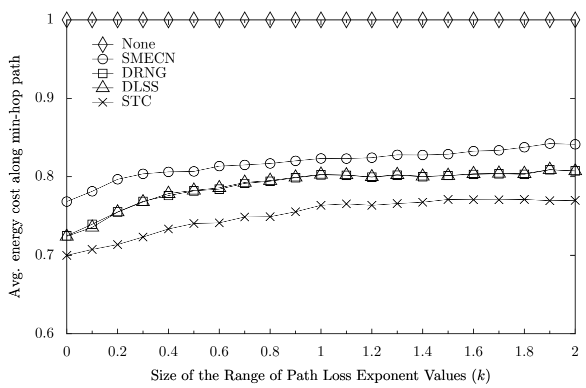
Our measure of interference derives from the definition in [19], which is a refinement of that used in [20]. Define the span, span(e), of an edge $e = (g, h)$ as the number of nodes that are neighbors of at least one of g and h. This represents the number of nodes that would have to remain silent to enable a successful transmission between g and h. We measure interference, $I_T(u \rightarrow v, \mathrm{H}ops)$, along the minimum-hop path from u to v in cover graph T as the sum of span(e) for all e in the path. $I_{T/H}(u \rightarrow v, \mathrm{H}ops)$ denotes the ratio of the interference along the minimum-hop path from u to v in cover graph T generated by the topology control algorithm and the corresponding cost along the minimum-hop path from u to v in the initial graph, H. This ratio represents the reduction in interference achieved due to the topology control algorithm.



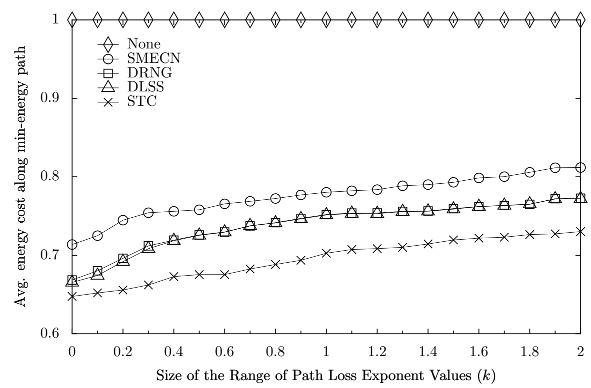
(a) Average node-degree.



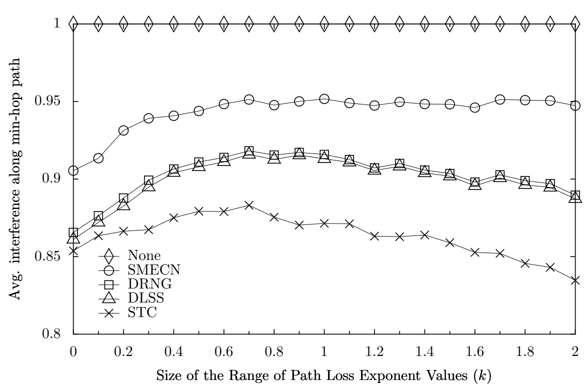
(b) Average $P_{T/H}(u)$ over all u.



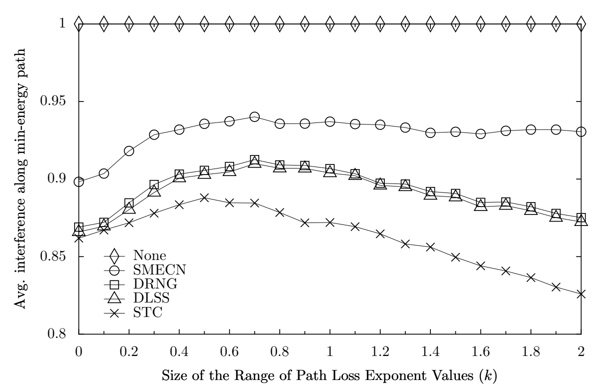
(c) $C_{T/H}(u \to v, \mathrm{H}ops)$ averaged over all pairs of nodes u and v.



(d) $C_{T/H}(u \to v, \mathrm{E}nergy)$ averaged over all pairs of nodes u and v.



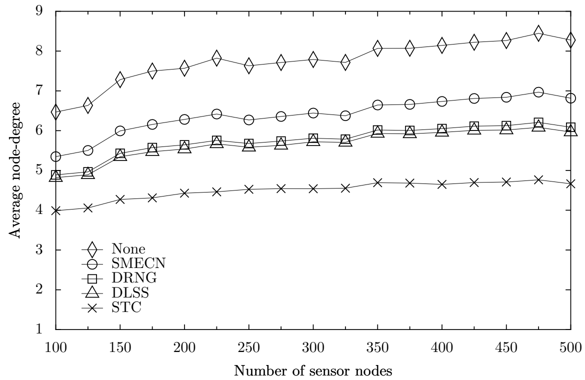
(e) $I_{T/H}(u \to v, \mathrm{H}ops)$ averaged over all pairs of nodes u and v.



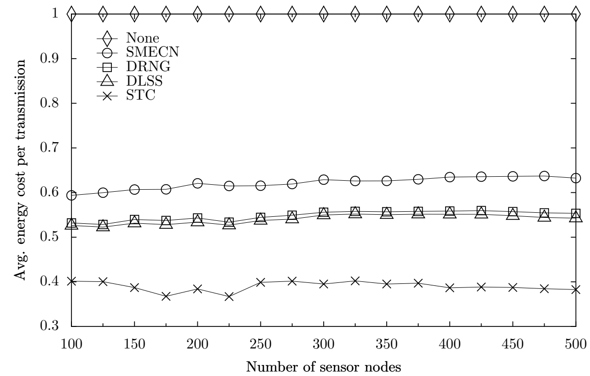
(f) $I_{T/H}(u \to v, \mathrm{E}nergy)$ averaged over all pairs of nodes u and v.

Fig. 3. Graphs showing the effectiveness of topology control algorithms in reducing energy costs when the path loss exponent varies in the range $[2.5 - k/2, 2.5 + k/2]$ for k from 0 to 2. The networks are generated with 200 randomly located nodes in a unit square area.

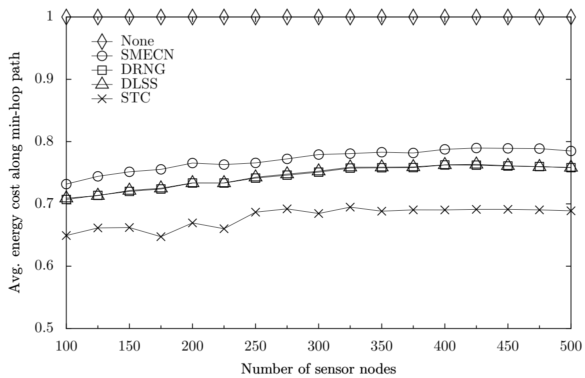
The ratio for the minimum-energy paths is computed similarly as above and is denoted by $I_{T/H}(u \to v, \mathrm{E}nergy)$.



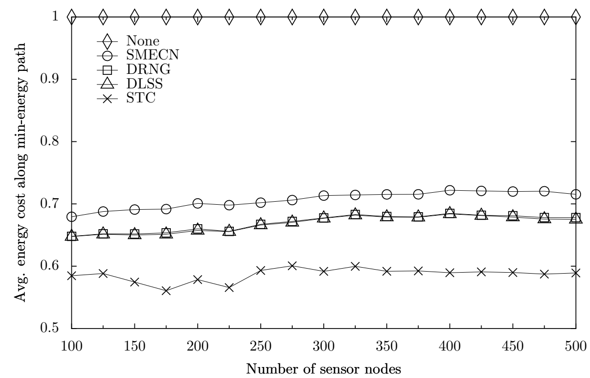
(a) Average node-degree.



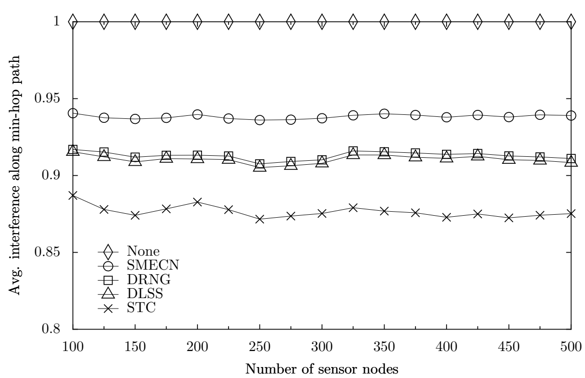
(b) Average $P_{T/H}(u)$ over all u.



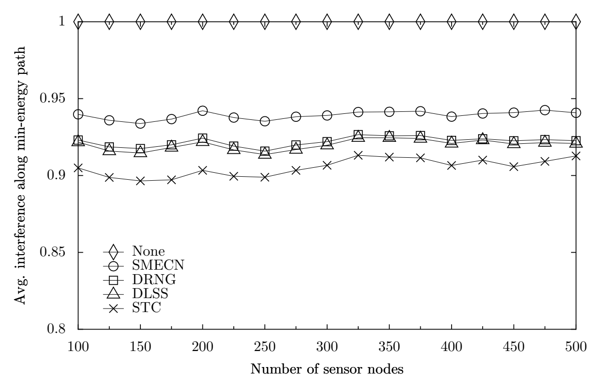
(c) $C_{T/H}(u \to v, \mathrm{H}ops)$ averaged over all pairs of nodes u and v.



(d) $C_{T/H}(u \to v, \mathrm{E}nergy)$ averaged over all pairs of nodes u and v.



(e) $I_{T/H}(u \to v, \mathrm{H}ops)$ averaged over all pairs of nodes u and v.



(f) $I_{T/H}(u \to v, \mathrm{E}nergy)$ averaged over all pairs of nodes u and v.

Fig. 4. Graphs showing the effectiveness of topology control algorithms in reducing energy costs when the path loss exponent varies in the range between 2.7 and 3.5 (urban area). The graphs show that the effectiveness of the STC algorithm increases with the number of nodes.

B. Experiment 1

We include the CBTC algorithms in our first set of simulation experiments. The CBTC algorithms, however, assume that the loss propagation characteristics of the medium are uniform across the entire region. In fairness to the CBTC algorithms, therefore, we conduct this simulation experiment with identical path loss exponents between any pair of directly communicating nodes in the network. For path loss exponents ranging from 1.5 to 3.5, Figure 2(a) plots the average node-degree of a node in the cover graphs generated by the topology control algorithms. SMECN is the only algorithm for which the average node degree varies substantially with a change in the path loss exponent. This is because SMECN makes a comparison between the energy cost along an edge with the *sum* of the energy costs along other edges. The result of such a comparison changes with the assumed value of the path loss exponent and therefore, SMECN results in different topologies under different values of the path loss exponent. All other algorithms are based on *one-to-one* comparison of edges as regards their energy costs, and the result of such comparisons is the same independent of the path loss exponent as long as the same path loss exponent characterizes the entire network.

For the same path loss exponents in the range from 1.5 to 3.5, Figure 2(b) presents the ratio $P_{T/H}(u)$ averaged over all nodes. Figures 2(c) and 2(d) plot the ratios $C_{T/H}(u \rightarrow v, \mathrm{H}ops)$ and $C_{T/H}(u \rightarrow v, \mathrm{E}nergy)$ averaged over all pairs of nodes u and v, respectively. These plots show the energy savings achieved by the algorithms along minimum-hop paths and along minimum-energy paths. Figures 2(e) and 2(f) similarly present results on the average interference encountered in a path. These results show that STC performs better—though only slightly better—than the other topology control algorithms when the path loss exponents are constant across the network. These results also show that even while the STC algorithm generates a sparser graph than other existing algorithms, it manages to keep the paths from lengthening unnecessarily and thus achieves an overall reduction in energy consumption.

C. Experiment 2

In our second set of experiments, we allow a non-uniform value of the path loss exponent in the region of interest. For each pair of directly communicating nodes, we choose a random value of the path loss component from a given range. We consider ranges of path loss exponents centered at 2.5 and gradually increasing to the range $[1.5, 3.5]$. Since the CBTC algorithms work under the assumption that the loss propagation characteristics of the medium are uniform across the entire region, we do not include CBTC algorithms in this set of experiments. If the energy cost of transmission from u to v is not the same as that from v to u, under DRNG or DLSS it is sometimes possible for a node u to keep a directed edge to node v, but for node v to drop the directed edge to u. However, since many MAC layer protocols expect bidirectional communication between directly communicating nodes [12], a topology control algorithm should ideally generate a graph in which a directed edge from u to v exists if and only if a directed edge from v to u exists. Therefore, in order to ensure a fair comparison when we simulate DRNG or DLSS in our studies, we assume that the energy cost or the path loss component between two directly communicating nodes is the same in both directions. While not always true, it is the most common case even in real multipath environments.

Figure 3(a) plots the average node degree in the cover graphs generated by the various topology control algorithms. It is of interest to note that even though the STC algorithm does not improve performance by much when the path loss exponents are uniform across the network, it does make a significant difference when the path loss exponents vary within a range. Figure 3(b) plots the ratio $P_{T/H}(u)$ averaged over all nodes for each range of path loss exponents. For the same path loss exponent ranges, Figures 3(c) and 3(d) plot the ratios $C_{T/H}(u \to v, \mathrm{H}ops)$ and $C_{T/H}(u \to v, \mathrm{E}nergy)$ averaged over all pairs of nodes u and v. Figures 3(e) and 3(f) similarly plot the average interference along the minimum-hop and the minimum-energy paths. These plots again indicate that even though the STC algorithm generates a sparser graph, it does not result in paths that are so much longer that the energy consumed or the interference along a path actually increases. In fact, we find that the STC algorithm reduces the energy consumed and, especially, the interference along a path significantly in comparison to other topology control algorithms applicable in a multipath environment. In particular, the effectiveness of the STC algorithm in cutting down the interference along a path improves slightly as the possible range of path loss exponents increases.

Since the DLSS algorithm is the closest in performance to the STC algorithm, it is worthwhile discussing the reasons behind the significant difference between their performances in the presence of multipath. Even though both DLSS and STC algorithms incur the same order of initial information exchange overhead, only in the STC algorithm does a node u use the information about a node that is not directly reachable by u but is a common neighbor of two or more neighbors of u. In DLSS, the local subgraph at node u used for generating a localized spanning tree in DLSS is one that is induced by the neighbors of u and does not contain a node that is not directly reachable by u. STC performs better than DLSS because, in an irregular radio environment, it is more likely that a node, say n, is unreachable by a node u even if it is reachable by more than one neighbor of u. Thus, DLSS ignores node n in the creation of the local spanning tree while the STC algorithm will consider it as long as n is reachable by some neighbor of u.

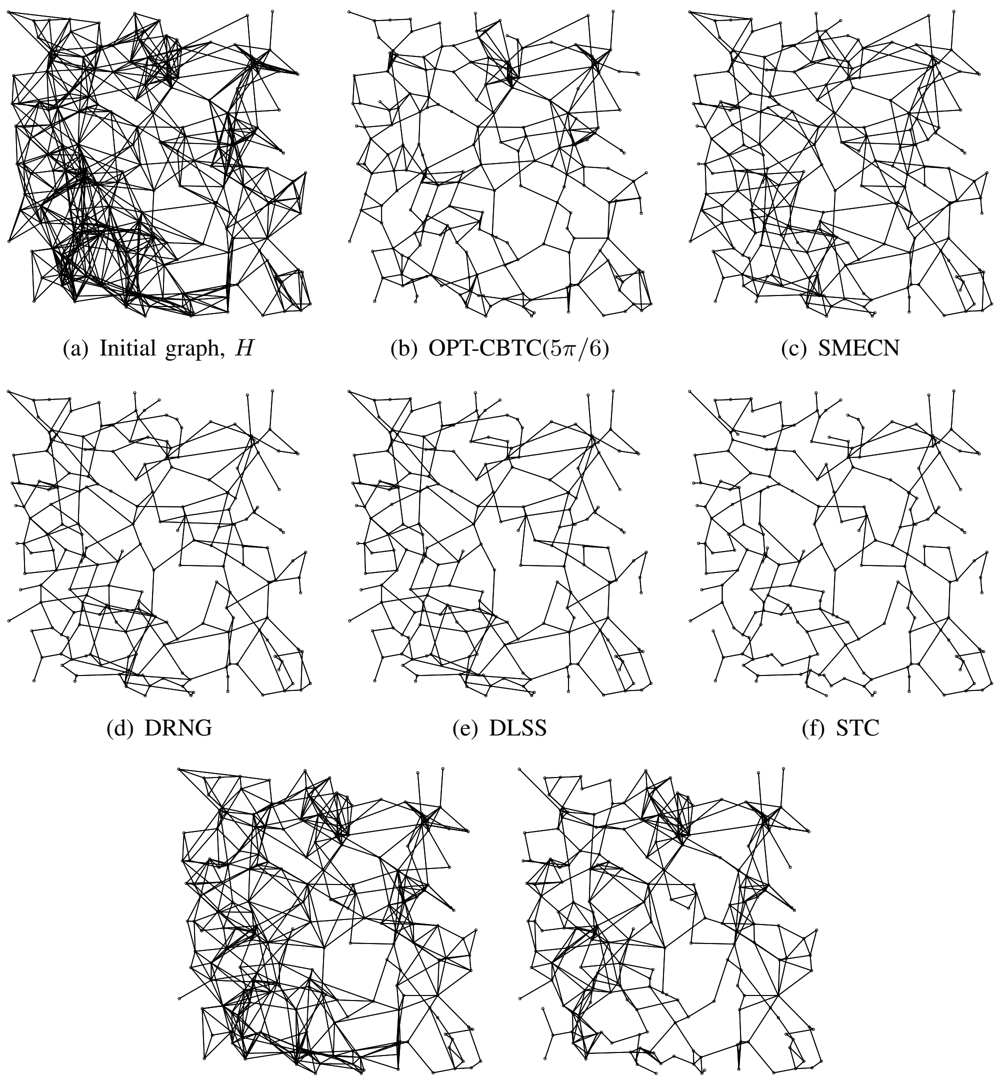
D. Experiment 3

In this set of experiments, we study the effectiveness of the algorithms across a five-fold increase in the number of nodes from 100 to 500. In these experiments, we use the range $[2.7, 3.5]$ of path loss exponents to simulate an urban area [21]. Figures 4(a) and 4(b) show that the STC algorithm performs better than other algorithms independent of the number of nodes. In fact, all of the topology control algorithms in our study scale similarly since they are all localized algorithms where no control information propagates beyond more than two hops.[1]

Figures 4(c) and 4(d) plot the ratios $C_{T/H}(u \to v, \mathrm{H}ops)$ and $C_{T/H}(u \to v, \mathrm{E}nergy)$ averaged over all pairs of nodes u and v for minimum-hop paths and for minimum-energy paths, respectively. Figures 4(e) and 4(f) similarly plot the results for interference experienced along a path. These plots all show that STC achieves lower energy costs and interference than other algorithms independent of the number of nodes in the network.

Finally, Figures 5(a)–5(f) present a pictorial representation of the graphs generated by the topology control algorithms. For these figures, we use the range $[2.7, 3.5]$ of path loss exponents to simulate an urban area [21] except for the case of OPT-CBTC($5\pi/6$) for which we used the average value of 3.1 (because CBTC algorithms assume that the path loss exponents are the

[1]The minimum energy cost of a transmission is generally assumed proportional to d^{α} where d is the distance the transmission has to reach and α is the path loss exponent. When values of d lie both above and below unity and when α is not a constant across the network, energy costs as measured by d^{α} are not directly comparable. To ensure a valid comparison, one needs to normalize the values of d used so that d is always above unity. The results presented in [1] did not use this normalization. This report corrects the error.



(a) Initial graph, H (b) OPT-CBTC($5\pi/6$) (c) SMECN

(d) DRNG (e) DLSS (f) STC

(g) DLSS (cover graph) (h) STC (cover graph)

Fig. 5. Graphs generated by the topology control algorithms when the path loss components range between 2.7 and 3.5 (urban area). For the OPT-CBTC($5\pi/6$) algorithm which assumes a uniform path loss exponent in the entire region, the plot shows the graph generated when the path loss exponent is 3.1 (average of 2.7 and 3.5). Figures (a)–(f) depict the graph generated by the respective topology control algorithms while Figures (g) and (h) depict the cover graphs. Each network consists of 200 randomly located nodes in a unit square area.

same across the entire region of the network). Figures 5(a)–5(f) are not the cover graphs since it is harder to observe the distinct edges and the sparseness achieved by the algorithms in the more dense cover graphs. Figures 5(g) and 5(h) depict the cover graphs generated by the DLSS and STC, the two algorithms that generate the most sparse cover graphs.

V. Concluding Remarks

Real wireless environments are characterized by a variety of radio irregularities and by the phenomenon of multipath propagation. Without topology control algorithms in these environments with large variations in loss characteristics, the nodes in a network may be forced to use very high power levels in their transmissions to ensure communication and network connectivity. Most topology control algorithms do not accommodate for the unique requirements of real wireless environments and often assume the ability of a node to deduce spatial information about its neighbors. The Step Topology Control (STC) algorithm presented in this report makes no spatial assumptions and achieves a reduction in the power levels of transmissions without the use of GPS devices.

We have presented simulation results studying the energy consumption properties of the STC algorithm in comparison to other algorithms that can be adapted to a multipath environment. While the STC algorithm certainly performs better than other algorithms when the loss characteristics are uniform in the region of the network, it performs significantly better when there exists a variation in these loss characteristics. This makes the STC algorithm especially desirable in the presence of multipath propagation and other causes of irregularities in the radio environment.

Most interestingly, the performance advantages of the STC algorithm come without an increase in the order of communication or computational complexity. In fact, DLSS, the topology control algorithm that comes closest to the STC algorithm in performance, has a higher computational complexity than the STC algorithm.

We show that the STC algorithm is related to the OPT-CBTC($5\pi/6$) algorithm and retains some of the angular properties of the CBTC algorithms. When the path loss characteristics are known to a node within its neighborhood, these properties may be employed in an optimization of the discovery process to determine when the local neighborhood is fully discovered for topology control purposes.

Acknowledgment

This work was supported in part by NSF Awards CNS-0322797 and CNS-0626548.

Appendix

Relationship to Cone-Based Topology Control

In the cone-based topology control (CBTC) algorithm, a node u determines the minimum power $p_{u,\alpha}$ at which it can make an omni-directional broadcast and successfully reach at least one neighbor node in each cone/sector of angle α. An edge (u, v) is removed from the network topology in an execution of CBTC(α) if u cannot reach v with power $p_{u,\alpha}$ *and* v cannot reach u at power $p_{v,\alpha}$. It is proved in [8] that when $\alpha \leq 5\pi/6$, the network connectivity is preserved. Additional optimizations to further reduce the power consumption at each node and remove more edges are possible and these include:

Shrink-Back Operation: If after the execution of CBTC(α) at a node u not all cones of angle α contain a neighbor node (i.e., there is an α-gap in the cone coverage as in the case of nodes at the boundary of the network), node u may unnecessarily transmit at its maximum power. With this optimization, a node transmits at the power at which further increasing its transmission power does not increase the cone coverage.

Pairwise Edge Removal: If there is an edge from u to v_1 and from u to v_2, then this operation removes the longer edge if $\angle v_1 u v_2 < \pi/3$, even if there is no edge between v_1 and v_2. It is proved in [8] that this preserves the connectivity.

OPT-CBTC($5\pi/6$) is the cone-based topology control algorithm that uses CBTC($5\pi/6$) along with the applicable optimizations of the shrink-back operation and pairwise edge removal. The CBTC algorithms assume uniform loss characteristics across the entire region of the network and also assume that the maximum power P_u of each node u is the same. We will make the same assumptions in the following to prove the relationship between STC and OPT-CBTC($5\pi/6$).

Let $G_{\mathrm{CBTC}(5\pi/6)} = (V, E_{\mathrm{CBTC}(5\pi/6)})$ denote the graph obtained by CBTC($5\pi/6$). Denote by $G_{\mathrm{OPT-CBTC}(5\pi/6)} = (V, E_{\mathrm{OPT-CBTC}(5\pi/6)})$ the graph obtained by OPT-CBTC($5\pi/6$). Let $G = (V, E)$ denote the graph obtained by the STC algorithm. Let $d(u, v)$ denote the distance between two nodes u and v. We first restate the lemma from [8] that helps in drawing the relationship we seek.

Lemma A.1: Any edge $(u, v) \in E_{max}$ either belongs to $E_{\mathrm{CBTC}(5\pi/6)}$ or there exist u', $v' \in V$ such that (a) $d(u', v') < d(u, v)$, (b) either $u' = u$ or $(u, u') \in E_{\mathrm{CBTC}(5\pi/6)}$, and (c) either $v' = v$ or $(v, v') \in E_{\mathrm{CBTC}(5\pi/6)}$.

The above lemma is proved in [8]. We now proceed to prove the relationship between STC and OPT-CBTC($5\pi/6$) by first proving the relationship between STC and CBTC($5\pi/6$).

Lemma A.2: If an edge $(u, v) \notin E_{\mathrm{CBTC}(5\pi/6)}$, then $(u, v) \notin E$.

Proof: Assume that edge $(u, v) \notin E_{\mathrm{CBTC}(5\pi/6)}$. When an edge $(u, n) \in E_{\mathrm{CBTC}(5\pi/6)}$ and $(u, v) \notin E_{\mathrm{CBTC}(5\pi/6)}$, we know that $d(u, v) > d(u, n)$ since n is discovered by u at a certain power level but v is not discovered by u. Therefore, using Lemma A.1, we know that there exist nodes u' and v' such that $d(u', v') < d(u, v)$ and in addition, one of the following three conditions is satisfied:

Case 1: $u' = u$, $v' \neq v$, $d(v, v') < d(u, v)$. In this case, we have a two-hop path between u and v through v' such that both hops are of distance less than $d(u, v)$.

Case 2: $u' \neq u$, $v' = v$, $d(u, u') < d(u, v)$. In this case also, we have a two-hop path between u and v through u' such that both hops are of distance less than $d(u, v)$.

Case 3: $u' \neq u$, $v' \neq v$, $d(u, u') < d(u, v)$, and $d(v, v') < d(u, v)$. We now have a three-hop path between u and v through u' and v' such that all three hops are of distance less than $d(u, v)$.

Since there exist nodes such that either a two-hop or a three-hop path exists between u and v with each hop corresponding to a distance smaller than $d(u, v)$, $(u, v) \notin E$.

■

Theorem A.3: If $(u, v) \notin E_{\mathrm{OPT-CBTC}(5\pi/6)}$, then $(u, v) \notin E$.

Proof: Since we know from Lemma A.2 that STC removes all the edges that are removed by CBTC($5\pi/6$), we only have to prove that STC also removes the edges removed by the optimizations of the shrink-back operation and pairwise edge removal.

When an edge (u, v) is removed as part of the Shrink-Back operation, the cone coverage around u does not change. Therefore, there exist nodes u' and v' such that $d(u', v') < d(u, v)$ and in addition, one of the three cases listed in the proof of Lemma A.2 applies. Exactly as in the proof of Lemma A.2, this implies that $(u, v) \notin E$.

When the Pairwise Edge Removal operation removes an edge (u, v), it implies there is another neighbor node (u, n) such that $d(u, n) < d(u, v)$ and $\angle nuv < \pi/3$. Since $\angle nuv < \pi/3$, edge (n, v) is not the longest edge in the triangle nuv. Since $d(u, n)$ is also smaller than $d(u, v)$, we have a two-hop path between u and v through n where the distance across each hop is less than $d(u, v)$. Therefore, edge (u, v) is removed by STC as well. ■

REFERENCES

[1] H. Sethu and T. Gerety, "A distributed topology control algorithm in the presence of multipath propagation," in *Proceedings of the 2nd International Workshop on Advances in Wireless Sensor Networks*, 2007.

[2] P. Santi, *Topology Control in Wireless Ad Hoc and Sensor Networks*. John Wiley and Sons, 2006.

[3] N. Li, J. C. Hou, and L. Sha, "Design and analysis of an MST-based topology control algorithm," *IEEE Transactions on Wireless Communications*, vol. 4, no. 3, pp. 1195–1206, May 2005.

[4] V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1333–1344, August 1999.

[5] L. Li and J. Y. Halpern, "Minimum energy mobile wireless networks revisited," in *Proceedings of the IEEE International Conference on Communications (ICC)*. IEEE, 2001, pp. 278–283.

[6] L. Jia, R. Rajaraman, and C. Scheideler, "On local algorithms for topology control and routing in ad hoc networks," in *Proceedings of the ACM symposium on Parallel Algorithms and Architectures (SPAA)*. ACM Press, 2003, pp. 220–229.

[7] X.-Y. Li, W.-Z. Song, and Y. Wang, "Efficient topology control for ad-hoc wireless networks with non-uniform transmission ranges," *Wireless Networks*, vol. 11, no. 3, pp. 255–264, 2005.

[8] L. E. Li, J. Y. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer, "A cone-based distributed topology control algorithm for wireless multi-hop networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 1, pp. 147–159, February 2005.

[9] D. Puccinelli and M. Haenggi, "Multipath fading in wireless sensor networks: measurements and interpretation," in *Proceedings of the 2006 International Conference on Communications and Mobile Computing*, 2006, pp. 1039–1044.

[10] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Models and solutions for radio irregularity in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 2, pp. 221–262, 2006.

[11] N. Li and J. C. Hou, "Localized topology control algorithms for heterogeneous wireless networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 6, pp. 1313–1324, December 2005.

[12] LAN MAN Standards Committee of the IEEE Computer Society, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," 2003. [Online]. Available: http://standards.ieee.org

[13] D. M. Blough, M. Leoncini, G. Resta, and P. Santi, "The k-Neigh protocol for symmetric topology control in ad hoc networks," in *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (ACM MobiHoc)*. ACM Press, 2003, pp. 141–152.

[14] R. Wattenhofer and A. Zollinger, "XTC: A practical topology control algorithm for ad hoc networks," in *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 2004, pp. 216–223.

[15] M. Damian, S. Pandit, and S. Pemmaraju, "Local approximation schemes for topology control," in *Proceedings of the Annual ACM Symposium on Principles of Distributed Computing (PODC)*. ACM Press, 2006, pp. 208–217.

[16] F. Kuhn and A. Zollinger, "Ad-hoc networks beyond unit disk graphs," in *Proceedings of the Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*. ACM Press, 2003, pp. 69–78.

[17] D. M. Blough, M. Leoncini, G. Resta, and P. Santi, "Topology control with better radio models: Implications for energy and multi-hop interference," in *Proceedings of the 8th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*. ACM Press, 2005, pp. 260–268.

[18] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical Society*, vol. 7, no. 1, pp. 48–50, February 1956.

[19] T. Johansson and L. Carr-Motyčková, "Reducing interference in ad hoc networks through topology control," in *Proceedings of the Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*. ACM Press, 2005, pp. 17–23.

[20] M. Burkhart, P. von Rickenbach, R. Wattenhofer, and A. Zollinger, "Does topology control reduce interference?" in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*. ACM Press, 2004, pp. 9–19.

[21] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Prentice Hall, 2002.