Topology Control in Ad hoc Wireless Networks using Cooperative Communication *

Mihaela Cardei †, Jie Wu†, and Shuhui Yang†

Abstract

In this paper, we address the Topology control with Cooperative Communication (TCC) problem in ad hoc wireless networks. Cooperative communication is a novel model introduced recently that allows combining partial messages to decode a complete message. The objective of the TCC problem is to obtain a strongly-connected topology with minimum total energy consumption. We show TCC problem is NP-complete and design two distributed and localized algorithms to be used by the nodes to set up their communication ranges. Both algorithms can be applied on top of any symmetric, strongly-connected topology to reduce total power consumption. The first algorithm uses a distributed decision process at each node that makes use of only 2-hop neighborhood information. The second algorithm sets up the transmission ranges of nodes iteratively, over a maximum of six steps, using only 1-hop neighborhood information. We analyze the performance of our approaches through extensive simulation.

Keywords: Ad hoc wireless networks, cooperative communication, energy efficiency, topology control.

*This work was supported in part by NSF grants CCR0329741, ANI 0073736, and EIA 0130806.
†Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431. Email: {mihaela@cse., jie@cse., syang1@}fau.edu.

1 Introduction

Ad hoc wireless networks consist of wireless nodes that can communicate with each other in the absence of a fixed infrastructure. Wireless nodes are battery powered and therefore have a limited operational time. Recently, the optimization of the energy utilization of wireless nodes has received significant attention [9]. Different techniques for power management have been proposed at all layers of the network protocol stack. Power saving techniques can generally be classified into two categories: by scheduling the wireless nodes to alternate between the active and sleep mode, and by adjusting the transmission range of wireless nodes. In this paper, we deal with the second method.

To support peer-to-peer communication in ad hoc wireless networks, the network connectivity must be maintained at any time. This requires that for each node there must be a route to reach any other node in the network. Such a network is called strongly-connected. In this paper, we address the problem of assigning a power level to every node such that the resulting topology is strongly-connected and the total energy expenditure for achieving the strong connectivity is minimized.

In order to reduce the energy consumption, we take advantage of a physical layer design that allows combining partial signals containing the same information to obtain the complete data. Cooperative communication (CC) models have been introduced recently in [11, 15]. By an effective use of the partial signals, a specific topology can be maintained with less transmission power.

In this paper, we first present some theoretical results by showing the NP-completeness of the TCC problem and some relevant bounds. We then propose two distributed and localized algorithms for the TCC problem, that start from a connected topology assumed to be the output of a traditional (without using CC) topology control algorithm. One algorithm uses 2-hop neighborhood information where each node tries to reduce overall energy consumption within its 2-hop neighborhood without losing connectivity under the CC model. The other one is based on a 1-hop neighborhood where each node, starting from a minimum range, iteratively increases its transmission range until all nodes in its 1-hop neighborhood are connected under the CC model. The initial strongly-connected topology is obtained as result of applying a traditional topology control algorithm, such as the distributed MST (DMST) [5] that generates an MST-based topology and the localized MST (LMST) [13] that generates a pseudo MST-based topology.

The rest of this paper is organized as follows. In section 2, we overview topology control protocols. Section 3 describes the CC model and the corresponding network model. Also, we introduce the TCC problem, prove its NP-completeness, and show the performance ratio between TCC and topology control without CC. In section 4, we propose a distributed and localized algorithm that can be applied to any symmetric, strongly-connected topology to reduce the total power consumption. We continue with an iterative approach for setting nodes transmission ranges in section 5. Section 6 presents the simulation results for the proposed algorithms, and section 7 concludes this paper.

2 Related Work

Topology control has been addressed previously in literature in various settings. In general, the energy metric to be optimized (minimized) is the total energy consumption or the maximum energy consumption per node. Sometimes topology control is combined with other objectives, such as to increase the throughput or to meet some specific QoS requirements. The strongly-connected topology problem with a minimum total energy consumption was first defined and proved to be NP-complete in [3], where an approximation algorithm with a performance ratio of 2 for symmetric links is given. In general, topology control protocols can be classified as: (1) centralized and global vs. distributed and localized; and (2) deterministic vs. probabilistic. The localized algorithm is a special distributed algorithm, where the state of a particular node depends only on states of local neighborhood. That is, such an algorithm has no sequential propagation of states. Comprehensive surveys of topology control can be found in [14] and [20].

Most protocols are deterministic. The work in [18] is concerned with the problem of adjusting the node transmission powers so that the resultant topology is connected or bi-connected, while minimizing the maximum power usage per node. Two optimal, centralized algorithms, CONNECT and BICONN-AUGMENT, have been proposed for static networks. They are greedy algorithms, similar to Kruskal's minimum cost spanning tree algorithm. For ad hoc wireless networks, two distributed heuristics have been proposed, LINT and LILT. However, they do not guarantee the network connectivity.

Among distributed and localized protocols, Li et al [12] propose a cone-based algorithm for topology control. The goal is to minimize total energy consumption while preserving connectivity. Each node will transmit with the minimum power needed to reach some node in every cone with degree α. They show that a cone degree $\alpha = 5\pi/6$ will suffice to achieve connectivity. Several optimized solutions of the basic algorithm are also discussed as well as a beaconing-based protocol for topology maintenance.

Li, Hou and Sha [13] devise another distributed and localized algorithm (LMST) for topology control starting from a minimum spanning tree. Each node builds its local MST independently based on the location information of its 1-hop neighbors and only keeps 1-hop nodes within its local MST as neighbors in the final topology. The algorithm produces a connected topology with a maximum node degree of 6. An optional phase is provided where the topology is transformed to one with bidirectional links.

Among probabilistic protocols, the work by Santi et al [19] assumes all nodes operate with the same transmission range. The goal is to determine a uniform minimum transmission range in order to achieve connectivity. They use a probabilistic approach to characterize a transmission range with lower and upper bounds for the probability of connectivity.

Some variants of the topology control problem have been also proposed by optimizing other objectives. Hou and Li in [6] present an analytic model to study the relationship between throughput and adjustable transmission range. The work in [7] puts forward a distributed and localized algorithm to achieve a reliable high throughput topology by adjusting node transmission power. The issue of minimizing energy consumption has not been addressed in these two papers. Jia, Li and Du [8] are concerned with determining a network topology that can meet QoS requirements in terms of end-to-end delay and bandwidth. The optimization criterion is to minimize the maximum power consumption per node. When the traffic is splittable, an optimal solution is proposed using linear programming.

Our work differs from these approaches by using cooperative communication [11, 15]. We explore this model in minimizing total power consumption while achieving a strongly-connected topology. A preliminary work on topology control with hitchhiking model is presented in [2]. In this paper [2], we introduce the Topology Control with Hitchhiking (TCH) problem and design a distributed and localized algorithm (DTCH) that can be applied on top of any symmetric, strongly-connected topology to reduce total power consumption.

3 Model and Problem Definition

In this section, we introduce the cooperative communication model and the corresponding network model. Then, we define the Topology control with Cooperative Communication (TCC) problem, show its hardness, and show a performance ratio between TCC and topology control without cooperative communication.

3.1 Cooperative Communication (CC) Model

Recently, a new class of techniques called *cooperative communication* (CC) (or *cooperation diversity*) has been introduced [11, 15] to allow single antenna devices to take advantage of the benefits of MIMO systems. Transmitting independent copies of the signal from different locations results in having the receiver obtain independently faded versions of the signal, thus reducing the fading effect through multipath propagation. In this communication model, each wireless node is assumed to transmit data and to act as a cooperative agent, relaying data from other users. There are wireless network applications proposed in literature that use the CC model, such as energy efficient broadcasting [1] and constructing a connected dominating set [21].

CC techniques are classified [11] as *amplify-and-forward*, *decode-and-forward*, and *selection relaying*. In the *amplify-and-forward* version, a node that receives a noise version of the signal can amplify and relay this noisy version. The receiver then combines the information sent by the sender and relay nodes. In *decode-and-forward* methods, a relay node must first decode the signal and then retransmits the detected data. Sometimes the detection of a relay node is unsuccessful and cooperative communication can detriment the data reception at the receiver. One method is to have a node decide if it relays its partner's data based on the signal-to-noise ratio (SNR) of the received signal. In *selection relaying*, a node chooses the strategy with the best performance.

The model considered in this paper belongs to the *decode-and-forward* category, where a node makes the relaying decision based on the SNR of the signal received. Such a model requires each node to have a memory that can store several packet amounts of data and a signal processor that can estimate the SNR of each received packet. This model, also referred in literature as the hitchhiking model in [1, 21], takes advantage of the physical layer design that combines partial signals containing the same information to obtain complete information. By effectively using partial signals, a packet can be delivered with less transmission power. The concept of combining partial signals using a maximal ratio combiner [16] has been traditionally used in the physical layer design of wireless systems to increase reliability.

Similar to the model in [1], we consider that messages are packetized. A packet contains a preamble, a header, and a payload. A preamble is a sequence of predefined uncoded symbols assigned to facilitate timing acquisition, a header contains the error-control coded information sequence about the source/destination address and other control flags, and a payload contains the error-control coded message sequence. We assume that the header and the payload of a packet are the outputs of two different channel encoders, and that the two channel codes are used by all the nodes in the system. The separation of a header and a payload in channel coding enables a receiver to retrieve the information in a header without decoding the entire packet. The use of the same channel codes enables a receiver to enhance the SNR at the input to the channel decoder by combining the payloads of multiple packets containing the same encrypted message.

We consider two parameters [1] related with SNR: γ_p, which is the threshold needed to successfully decode the packet payload, and γ_{acq}, which is the threshold required for a successful time acquisition. The system is characterized by $\gamma_{acq} < \gamma_p$. We note with k the ratio of these two thresholds, $k = \gamma_{acq}/\gamma_p$. We assume that the threshold to successfully decode a header is less than or equal to the threshold to successful time acquisition γ_{acq}. A packet received with a SNR γ is: (1) fully received, if $\gamma_p \leq \gamma$, (2) partially received, if $\gamma_{acq} \leq \gamma < \gamma_p$, and (3) unsuccessfully received, if $\gamma < \gamma_{acq}$. Therefore, when a packet is fully or partially received ($\gamma_{acq} \leq \gamma$), the header information is successfully decoded.

Consider that when a wireless node i transmits a packet, the *coverage* of a node j that receives the packet with a SNR per symbol γ is defined as: $c_{ij} = 1$ for $\beta > 1$, $c_{ij} = \beta$ for $k < \beta \leq 1$, and $c_{ij} = 0$ for $0 < \beta \leq k$, where $\beta = \gamma/\gamma_p$. A channel gain is often modeled as a power of the distance, resulting in $\beta = r^\alpha/d_{ij}^\alpha = (r/d_{ij})^\alpha$, where α is a communication medium dependent parameter, r is the communication range of node i, and

d_{ij} is the Euclidean distance between the nodes i and j. For example, consider $k = 0.125$ and $\alpha = 2$. Let us assume node i transmits a packet. For a node j with $r/d_{ij} = 1/2$, the coverage is 0.25, whereas for the case $r/d_{ij} = 1/3$ the coverage is 0. The basic idea in the CC model is that if the same packet is partially received n times from different neighbors with $\gamma_{acq} \leq \gamma_i < \gamma_p$ for $i = 1..n$ such that $\sum_{i=1}^{n} \gamma_i \geq \gamma_p$, then the packet can be combined by a maximal ratio combiner [16] and can be successfully decoded.

3.2 Network Model

We consider an ad hoc wireless network with n nodes equipped with omnidirectional antennas. The nodes in the network are capable of receiving and combining partial received packets in accordance with the CC model introduced in section 3.1. We represent the network by a directed graph $G = (V, E)$, where the vertices set V is the set of nodes corresponding to the wireless devices in the network and the set of edges E corresponds to the communication links between devices. Between any two nodes i and j there will be an edge ij if the transmission from node i is received by the node j with a SNR greater than γ_{acq}.

Every node $i \in V$ has an associated transmission power level $p_i = r^\alpha$. For each edge $ij \in E$, the *coverage* provided by node i to node j is defined as $c_{ij} = 1$ for $p_i/d_{ij}^\alpha \geq \gamma_p$ and $c_{ij} = p_i/(d_{ij}^\alpha \times \gamma_p)$ for $\gamma_{acq} \leq p_i/d_{ij}^\alpha < \gamma_p$. The case $p_i/d_{ij}^\alpha < \gamma_{acq}$ is not included since an edge will exist only when the SNR of the received signal is at least γ_{acq}, that is, $p_i/d_{ij}^\alpha \geq \gamma_{acq}$. In this paper, we consider the cases when α equals 2 and 4, and $\gamma_p = 1$.

3.3 Topology Control with Cooperative Communication (TCC)

In this section, we introduce the Topology control with Cooperative Communication (TCC) problem. The fully received packet is defined as follows. Considering a transmission from a node i to a node j, node j is partially (fully) covered by i if $1 > c_{ij} \geq \gamma_{acq}$ ($c_{ij} = 1$). If,

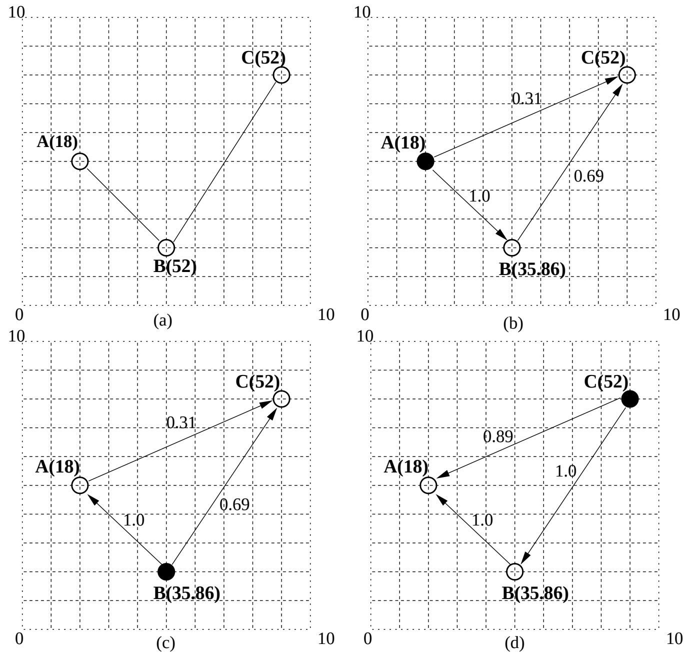


Figure 1: A cooperative communication example. (a) Initial power consumption based on MST. (b) Power consumption with A as source. (c) B is the source. (d) C is the source.

upon combining the packets received from one or more neighbors, say k neighbors, results in a full coverage of node j, i.e. $\Sigma_k p_k / d_{kj}^{\alpha} \geq 1$, then the packet is fully received.

We define *strong connectivity* under the CC model as follows. For any node s sending a packet, there should be a "path" to every other node, that is, the packet should be fully received by all other nodes in the network. The following rules apply: (1) s has the full packet, and (2) only nodes that fully received the packet are able to forward it, including s. Each node that has fully received a packet will forward it only once. Now we can formally define the TCC problem as follows:

TCC Definition. *Given an ad hoc wireless network with n nodes and using the CC model, assign a power level to every node such that: (1) the sum of the power levels in all nodes is minimized $\sum_{i=1}^{n} p_i = MIN$, and (2) the resultant CC-based topology is strongly-connected.*

Figure 1 presents a simple example of strong connectivity using the CC model, where $\gamma_{acq} = 0.2$. We assume the power required to communicate between two nodes to be the square of the distance between them. The number on each edge represents the coverage provided by the source node to the destination node. In Figure 1 (a), a minimum spanning tree (MST) is formed among the three nodes, where each bi-directional link corresponds to two uni-directional links. Each node sets its power to reach its furthest neighbor on the MST. For example, node B must set its power to $4^2+6^2 = 52$ to reach node C. The topology is strongly-connected if, having any node as the source of a message, all the other nodes can get this message directly or by forwarding. In a model with CC as in Figures 1 (b), (c) and (d), communication power of a node can be reduced to partially cover some neighbors as long as several partial messages can be combined for a successful message receipt at those nodes. Figures 1 (b), (c) and (d) show that starting from each node, all other nodes are fully covered, thus the resulting topology is strongly connected. For example, in (b), node A has a power of 18 to fully cover B ($3^2 + 3^2 = 18$), and to 31% cover C ($18/(7^2 + 3^2) = 31\%$). Since B has received the complete message, it can forward the message to C, providing 69% coverage with the power level set to $52 \times 6\% = 35.86$. Thus C gets the complete message. Using the same idea, the two other nodes are fully covered if we select node B or C as the source node. Therefore, the graph is strongly-connected using CC.

3.4 NP-Completeness of the TCC Problem

Kirousis et al [10] gave a formal proof of NP-completeness for the general graph version of the topology control (GTC) problem, without using CC. In order to prove that TCC is NP-complete, we show that TCC belongs to the NP-class and GTC is a special case of TCC.

Theorem 1: The TCC problem is NP-complete.

Proof: It is easy to see that TCC belongs to the NP-class. Having assigned a transmission power for each node in the network, it can be verified in polynomial time whether the resultant topology is strongly-connected using CC and whether the cost of this assignment (sum of the powers of each node) is less than a fixed value.

Next, we show that GTC is a special case of TCC. When $\gamma_{acq} = \gamma_p$, we have no case of partial reception of signals. Thus the TCC problem reduces to the GTC problem, where a signal is either fully received or the reception fails. Hence, the GTC problem is a special case of the TCC problem for $\gamma_{acq} = \gamma_p$.

Because GTC is NP-complete and is a particular case of the TCC problem and because TCC belongs to the NP-class, we conclude that TCC is an NP-complete problem. □

3.5 Performance Ratio Between GTC and TCC Problems

In this section, we prove that the optimal solution of the GTC problem has a performance ratio of $1/k$ with the optimal solution of the TCC problem, where k is defined in section 3.1.

Theorem 2: *The performance ratio between the optimal solution of the GTC problem and the optimal solution of the TCC problem is upper bounded by* $1/k$.

Proof: Let us note the optimal solution of the GTC problem with OPT^{GTC} and the optimal solution of the TCC problem with OPT^{TCC}. It is clear that $OPT^{TCC} \leq OPT^{GTC}$ since the solution set of the TCC problem includes that of the GTC problem. Next, we show that $OPT^{GTC} \leq \frac{1}{k} \cdot OPT^{TCC}$.

Let us assume there are n nodes in the network, noted with $1, 2, ..., n$. Let us note with $r_1, r_2, ..., r_n$ the node transmission ranges associated with OPT^{TCC} . Then $OPT^{TCC} = r_1^\alpha + r_2^\alpha + ... + r_n^\alpha$. For a node i, we note with N_i^{TCC} the set of nodes partially or totally covered by i. Then $\forall j \in N_i^{TCC}$, $(\frac{r_i}{d_{ij}})^\alpha \geq k$, where d_{ij} is the distance between nodes i and j. Let us consider now the case when each transmission range is increased $k^{-\frac{1}{\alpha}}$ times. This corresponds to a solution SOL with node transmission ranges $r'_1, r'_2, ..., r'_n$:

$$SOL = \frac{1}{k} \cdot OPT^{TCC} = (r_1 \cdot k^{-\frac{1}{\alpha}})^\alpha + ... + (r_n \cdot k^{-\frac{1}{\alpha}})^\alpha = r_1'^\alpha + r_2'^\alpha + ... + r_n'^\alpha$$

For any node $i = 1..n$ and for any node $j \in N_i^{TCC}$, we have $(\frac{r'_i}{d_{ij}})^\alpha = (\frac{r_i \cdot k^{-\frac{1}{\alpha}}}{d_{ij}})^\alpha = \frac{1}{k} \cdot (\frac{r_i}{d_{ij}})^\alpha \geq 1$.

Table 1: DTCC notations.

G	Symmetric, strongly-connected starting topology
f_i	1 if node i decided its final power, otherwise 0
p_i	Transmission power level of node i
$N(i)$	Set of 1-hop neighbors of node i in G
$P(i)$	Set of transmission power levels of node i
$g_i(p)$	Gain of node i at power level p
d_{ij}	Distance between nodes i and j

Therefore, all nodes that were previously partially covered in the TCC solution are now fully covered and the strong connectivity is preserved. Therefore, SOL is also a solution of the GTC problem, with $OPT^{GTC} \leq SOL$. This results in $OPT^{GTC} \leq \frac{1}{k} \cdot OPT^{TCC}$.

To summarize, we have proved that $OPT^{TCC} \leq OPT^{GTC} \leq \frac{1}{k} \cdot OPT^{TCC}$, therefore, $\frac{OPT^{GTC}}{OPT^{TCC}} \leq 1/k$. □

4 Distributed Topology Control using the Cooperative Communication (DTCC) Algorithm

In this section, we propose the distributed topology control using the cooperative communication (DTCC) algorithm that can be applied to any symmetric, strongly-connected topology to reduce the total power consumption. Any node decides its final power based only on local information from its 2-hop neighborhood. To be distributed and localized are important characteristics of an algorithm in ad hoc wireless networks, since it adapts better to a dynamic and scalable architecture.

4.1 Basic Ideas

In describing the algorithm, we use the notations in Table 1. Each node independently "locks" its 1-hop neighborhood to perform power adjustment to save energy. We take node i as the current node for the example in Figure 2. All the nodes on the inner dashed circle including j are i's 1-hop neighbors. The nodes on the outer dashed circle, such as k and l, are i's 2-hop neighbors. The main idea of DTCC is to increase i's power level to "contribute" to the coverage of its 2-hop neighbors so the range of i's 1-hop neighbors can be reduced, and at the same time the overall power consumption can be reduced. To ensure connectivity, 1-hop neighbors should still be able to reach i directly. Such a process is the *2-hop power reduction process*. In fact, in the 2-hop power reduction process, i and its 1-hop neighbors are involved in an "atomic action". To implement such an atomic action, two approaches can be used:

1. *Back-off scheme*. After node i has selected a new power level, it backs off a period of time inversely proportional to its calculated gain. The gain of node i represents the maximum decrease in the total power obtained by adjusting the power of node i to one of the predefined values in $P(i)$. This will give priority to the nodes with higher gain to set up their final power first. If node i receives an update during this interval, then it recomputes its power level and back-off again. If the timer expires without any updates, then node i considers this power level as its final power, and announces this power level together with its neighbors' new power levels to the nodes within its 2-hop neighborhood.

2. *Locking scheme*. Node i needs to securely lock of all its neighbors (in addition to its own lock). Once i completes its power reduction process, it releases its lock and the locks of its neighbors, and announces the power levels of itself and its neighbors to the nodes within its 2-hop neighborhood. Unlike the back-off scheme that may exhibit occasional mis-coordination, the locking scheme guarantees that nodes execute the 2-hop power

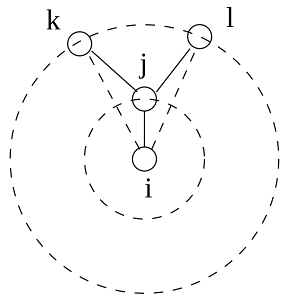


Figure 2: Illustration of 2-hop neighbor set of i.

reduction process without conflict. However, it is more expensive.

4.2 Detailed DTCC Algorithm

The DTCC algorithm starts from a symmetric (bi-directional links), connected topology G, assumed to be the output of a traditional topology control algorithm. Two such algorithms, DMST and LMST, are addressed later in this section. Initially, each node i sets its power p_i to the value p_i^0 needed to reach its furthest 1-hop neighbor in G.

We assume that each node i has all the distance information within its 2-hop neighborhood and the p_j values of all 1-hop neighbors. Note that this kind of information is usually available after the traditional topology control algorithm completes. Node i maintains p_j values for all its 1-hop neighbors. Whenever p_j for a node j changes, node j broadcasts this change to its neighbors.

The goal of the DTCC algorithm, by starting from an initial power p_i^0, is to decide the final power assignment by using the CC model such as to minimize the total power. Next, we describe the mechanism used by each node in order to decide its final power level.

The gain of node i is computed in $ComputeGain(i)$. The gain $g_i(p)$ is defined as the maximum decrease in the total power, obtained by increasing node i's transmission power level to $p \in P(i)$, in exchange for a decrease of the power levels of some of the node i's neighbors. This is because when the power level of node i is increased, i provides partial or full coverage to more nodes in the network. For example, if k is a 1-hop neighbor of node j, where $j \in N(i)$ (see Figure 2), then an increase in the partial or full coverage of node k may facilitate reduction of the power level of node j that can provide less coverage to node k.

Each node i maintains a variable f_i initially set to 0, meaning that this node has not yet decided its final power level. In order to decide its final power, node i computes the gain for various power levels and selects the power level for which the gain is maximum. The power levels in $P(i)$ are those power levels for which node i could reduce the power level of a neighbor j to d_{ij}^{α}, by providing the additional coverage needed for a full coverage of all the neighbors of j.

The process of computing the gain is performed for each power level $p \in P(i)$. Once the gains for all power levels in $P(i)$ are determined, the node selects the power level that produces a maximum gain, noted with p_i^{new}. If there is no power level p such that $g_i(p) > 0$, then p_i will not change. When node i announces its new power level through $Broadcast()$, all its neighbors j with $f_j \neq 1$ will invoke *Reduce*() to decrease their power levels and broadcast the change, as a result of the additional coverage provided by node i.

The pseudocode presented next uses a back-off scheme (see section 4.1) in order to implement the 2-hop power reduction process as an atomic action. Each node i backs-off a time inversely proportional to its calculated gain before deciding its final power. If, during the back-off interval, node i receives a broadcast from a neighbor j, then node i first update its power p_i and then continues the back-off scheme.

Algorithm DTCC(i)

$p_i \leftarrow p_i^0$
$f_i \leftarrow 0$
while $f_i = 0$ **do**
    compute $P(i)$
    ComputeGain(i)
    $p_i^{new} \leftarrow$ power level for which gain is maximum

  start a timer $t \leftarrow \frac{1}{g_i(p_i^{new})}$
  if broadcast message received from j before t expires **then**
    $p_i \leftarrow$ Reduce(j, p_j, i)
  else
    $p_i \leftarrow p_i^{new}$
    $f_i \leftarrow 1$
  end if
  Broadcast(i, p_i, f_i)
end while

ComputeGain (i)

*/*Find gain for all power levels in $P(i)$*/*
for all $p \in P(i)$ **do**
  for all $j \in N(i)$ **do**
    $p_j^{red} \leftarrow$ Reduce (i, p, j)
  end for
  $g_i(p) \leftarrow \sum_{j \in N(i)}(p_j - p_j^{red}) - (p - p_i)$
end for

Reduce (i, p, j)

*/*Reduce the power of node j on the basis of partial coverage provided by node i with power p */*
if $f_j = 1$ **then**
  return p_j
end if
for all $k \in N(j)$ **do**
  $p_j(k) \leftarrow (1 - c_{ik}) \times d_{jk}^{\alpha}$
end for
return $\max\{d_{ij}^{\alpha}, \max_{k \in N(j)} p_j(k)\}$

4.3 Properties

The complexity of the DTCC algorithm run by each node i is polynomial in the total number of nodes n. The complexity of the *ComputeGain(i)* procedure takes $O(|P(i)| \times |\Delta|^2)$ time, where Δ is the maximal node degree. This is because for each neighbor $j \in N(i)$, the $i's$ coverage on each 2-hop neighbor $k \in N(j)$ needs to be computed. This process has to be done for each power level in $P(i)$. When $|P(i)| = O(\Delta)$, it is $O(\Delta^3)$. Therefore, the complexity of the algorithm DTCC run on each node is $O(\Delta^4)$ with another loop.

Next, we show the correctness of the DTCC algorithm:

Theorem 3: *The power level assignment provided by the DTCC algorithm guarantees a strongly-connected topology with the CC model.*

Proof: Initially, each node is assigned the power level needed to reach the furthest 1-hop neighbor in G. The starting topology G is strongly-connected, that is, between any two nodes there exists a path. We note that there are two cases when a node's power level may change in the DTCC algorithm: (a) in line 11, but here the value is increased, so this will not affect connectivity, and (b) in line 6 of the procedure *Reduce()*, when a node's power level may be reduced.

Let us assume by contradiction that after applying the DTCC algorithm, the strong connectivity is not preserved. Then, there exist two nodes i and j such that when the node i is sending a packet, this packet is not fully received by j. The nodes i and j are connected in G, so there exists a path $i_0 = i$, i_1, ..., $i_m = j$ between i and j. We show by induction that i_m fully receives the packet sent by i_0.

First, i_0 has the full packet. If i_0 did not change its power or has increased the power level, then i_1 is fully covered by i_0 and therefore receives the full packet from i_0. Let us consider the case when i_0 has reduced its power level. Then, in conformity with DTCC, the current power of i_0 was updated when one of its neighbors, say k, has set up its final power. In that case, i_0 fully covers k and i_0 together with k fully cover all i_0's neighbors, including i_1. So i_1 also fully receives the packet. Applying the same mechanism, we can show that any node on the path fully receives the packet sent by its predecessor, even if it is not fully covered by its predecessor. Thus, node i_m fully receives the packet, contradicting our initial assumption that strong connectivity is not maintained after running DTCC. □

4.4 Two Special Cases

We have applied the DTCC algorithm on two starting topologies output by two distributed algorithms: DMST (Distributed MST) and LMST (Localized MST). We note with DMST the Gallegar's distributed algorithm [5] for constructing an MST, and with *DMST-based DTCC* the DTCC algorithm that starts from a topology G generated by DMST. Also, we note with LMST the algorithm proposed by Li et al [13] for constructing a pesudo MST, and with *LMST-based DTCC* the DTCC algorithm that starts from a topology G generated by LMST.

MST has been considered before as a reference point in designing topology control mechanisms in the general model (without CC) because of its important properties and good performance. MST has the minimum longest edge among all the spanning trees [4], therefore, if every node has assigned a power level needed to reach the furthest neighbor then the maximum power assigned per node is minimized for the MST compared with other spanning trees. This property results in maximizing the time until the first node will deplete its power resources. Another property of the MST-based topology in the general case (without CC) is that it provides an approximation algorithm with a performance ratio of 2 [10].

Next, we prove that an MST-based topology has a performance ratio of $2/k$ for the TCC problem. An *MST-based topology* is a mechanism that builds an MST over all n nodes in the network and then assigns to any node the power needed to reach the furthest neighbor in the MST.

Theorem 4: *An MST-based topology is an approximation algorithm with ratio bound of $2/k$ for the TCC problem, where $k = \gamma_{acq}/\gamma_p$ is a constant $k \in (0, 1]$, and represents a characteristic of the wireless communication medium.*

Proof: Let us note the optimal solution of the GTC problem with OPT^{GTC}, the optimal solution of the TCC problem with OPT^{TCC}, and the MST-based solution with MST.

It is proved in [10] that an MST-based topology has a performance ratio of 2 for the GTC problem, therefore $MST \leq 2 \cdot OPT^{GTC}$. In Theorem 2, we proved that $OPT^{GTC} \leq \frac{1}{k} \cdot OPT^{TCC}$, therefore, $MST \leq \frac{2}{k} \cdot OPT^{TCC}$. Since $OPT^{TCC} \leq MST$, we obtain that $OPT^{TCC} \leq MST \leq \frac{2}{k} \cdot OPT^{TCC}$ and thus the theorem holds. □

Since DMST-based DTCC starts from an MST-based topology and improves it, using the CC advantage, DMST-based DTCC will also have a performance ratio of $2/k$ for the TCC problem.

As DTCC and LMST are localized, the resultant LMST-based DTCC is localized. However, LMST-based DTCC does not guarantee a performance ratio since LMST is not strictly MST-based topology. We present the simulation results for LMST-based DTCC in section 6. Note that if the DTCC is applied on LMST, the complexity is $O(1)$. This is because in LMST, the degree of any node in the resulting topology is bounded by 6 [13]. Therefore, the power level of node i, $|P(i)|$, is constant in DTCC. The complexity of DTCC in general case is $O(|P(i)| \times |N(i)|^2)$, which is $O(1)$ here.

Figure 3 shows an example of a six nodes topology. The number on each node indicates the power level used by that node in maintaining the topology based on (a) DMST and (b) LMST. We use unidirectional links to represent full coverage in both directions, whereas directional links with values less than 1 indicate the amount of partial coverage.

In Figure 3 (a) we present a DMST-based topology without CC. The power level assigned to each node is the power needed to reach the furthest neighbor in DMST. The total cost is 186. In Figure 3 (b), we show the topology obtained after using the LMST algorithm [13], with a total cost of 287. LMST uses a localized way to generate the MST where every node decides its 1-hop neighbors independently. Therefore, in a global view, the resulting topology might be a graph with cycles.

Figure 3 (c) shows the topology and power assignment after running the DMST-based DTCC algorithm. We assume $\gamma_{acq} = 0.01$ and $\alpha = 2$. First, each node computes its *gain*.

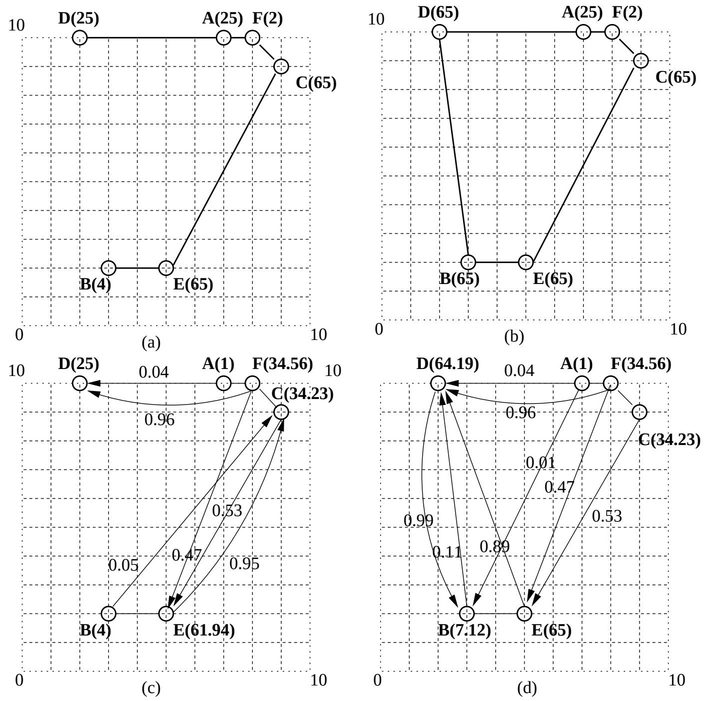


Figure 3: Example of DTCC ($\gamma_{acq} = 0.01$, $\alpha = 2$). (a) DMST and power consumption. (b) LMST and power consumption. (c) DMST-based DTCC. (d) LMST-based DTCC.

As node F has the largest *gain*, it increases its power to 34.56, and thus nodes A and C decrease their power to 1 and 34.23, respectively. In the second round, node B sets its power to 4 and node E decreases its power to 61.94. We obtain a total cost of 160.73, and a 13.59% power reduction compared with the output of the DMST algorithm in Figure 3 (a). Strong connectivity is also preserved. For example, node A reduces its power to 1, which partially covers its neighbor D with 0.04, while node T provides the additional 0.96 coverage. Thus, a message sent from A is fully received by F, and then A and F can together cover D.

Figure 3 (d) shows the execution of the LMST-based DTCC algorithm with a total cost of 206.1 and a reduction ratio of 28.19% compared with LMST algorithm in Figure 3 (b).

Table 2: ITCC notations.

G	Symmetric, strongly-connected starting topology
f_i	1 if node i decided its final power, otherwise 0
p_i	Transmission power level of node i
p_i^{max}	Transmission power of node i needed to reach furthest neighbor in $N(i)$
p_i^{min}	Transmission power of node i needed to reach closest neighbor in $N(i)$
$N(i)$	Set of 1-hop neighbors of node i in G
$N'(i)$	Set of 1-hop reachable neighbors using CC

5 Incremental Topology Control using Cooperative Communication (ITCC) Algorithm

In this section, we propose a distributed and localized algorithm that uses a different approach to set up nodes' transmission power. The Incremental Topology control using Cooperative Communication (ITCC) algorithm is based on 1-hop neighborhood information. Each node, starting from a minimum power, iteratively increases its transmission power until all the nodes in its 1-hop neighborhood are fully covered under the CC model.

5.1 Basic Ideas

The main algorithm notations are introduced in the Table 2. ITCC algorithm starts from a symmetric, connected topology G, assumed to be the output of a traditional topology control algorithm such as DMST and LMST. Each node i computes p_i^{max} and p_i^{min}, the transmission powers needed to reach the furthest and the closest neighbor in $N(i)$, corresponding to G. The final power selected by node i is a value between p_i^{min} and p_i^{max}. The goal of this algorithm is to find a minimum transmission power for node i in $[p_i^{min}, p_i^{max}]$, such that all the nodes in $N(i)$ are fully covered by node i using CC. In the CC model, if a node v fully receives a message transmitted by a node u (directly or using CC), then v will resend the message once using its current power level.

The ITCC algorithm adopts an iterative process where each node gradually increases its power (initially p_i^{min}). To avoid simultaneous updates among neighbors, either a back-off or a locking scheme can be used (see section 4.1).

5.2 Detailed ITCC Algorithm

We assume that each node i has the distance and location information for its 1-hop neighborhood $N(i)$, information usually available after running the traditional topology control algorithm. Each node i maintains its current power estimate, p_i and the p_j value for each node $j \in N(i)$. When a node decides its final power value, it sets f_i to 1.

The goal of the ITCC algorithm is, by starting from an initial power p_i^{min} needed to reach the closest 1-hop neighbor for each node i, to iteratively increment the power until all nodes in $N(i)$ are fully covered using the CC model. When this condition is met, node i declares its current power estimate as its final power assignment. Next, we describe the mechanism used by each node i to decide its final power level.

Each node i maintains a variable f_i which is initially set to 0, meaning that this node has not yet decided its final power level. The algorithm executes in at most $|N(i)|$ rounds (or iterations). In each round, power level p_i is minimally incremented with Δp_i such that at least one node in $N(i) - N'(i)$ is added to $N'(i)$. Δp_i can easily be computed since node i maintains the distance and location information for all nodes in $N(i)$. The algorithm finishes when $N(i) = N'(i)$, that is using the current power estimate p_i, node i covers all nodes in $N(i)$ using the CC model.

All broadcast messages sent to advertise new power level updates are sent with power level p_i^{max}. If, during the back-off interval, a broadcast message is received from a neighbor in $N(i)$, then $N'(i)$ and Δp_i are updated before continuing the back-off waiting. It might happen that the value Δp_i decreases, but this is safe since node i did not advertise the new power level yet. When the time comes for node i to broadcast its advertisement, it updates its power level $p_i \leftarrow p_i + \Delta p_i$ and the reachable neighborhood set $N'(i)$. If $N(i) = N'(i)$, then the current power level is the final power level of node i.

The rounds should be designed to have each node advertise its new power estimate once. Ideally, the nodes will send the broadcast without colliding with their neighbors' advertising. To avoid collisions, we could use a 1-hop neighborhood locking scheme or a back-off mechanism (see section 4.1). The pseudocode presented next uses a back-off scheme, where each node backs-off a time inversely proportional to its calculated gain before sending a broadcast. The gain can be computed for example as $p_i^{max} - (p_i + \Delta p_i)$. In this case, nodes with a smaller power level will advertise earlier, thus helping through CC the nodes with a higher transmission power. This scheme could help to balance power consumption. If, during the back-off time interval, node i receives an advertisement from a neighbor $j \in N(i)$, then node i does first the update and then continues the back-off scheme.

Algorithm ITCC(i)

$p_i \leftarrow p_i^{min}$
$f_i \leftarrow 0$
Broadcast(i, p_i, f_i)
while $f_i = 0$ **do**
    compute Δp_i, the minimum incremental power needed to cover at least one neighbor in $N(i) - N'(i)$
    start timer t
    if broadcast message received from j before t expires **then**
        update $N'(i)$, Δp_i
        if $N(i) = N'(i)$ **then**
            $f_i \leftarrow 1$
            Broadcast(i, p_i, f_i)
            return
        end if
    end if
    if timer t expires **then**
        $p_i \leftarrow p_i + \Delta p_i$
        update $N'(i)$
        if $N(i) = N'(i)$ **then**

```
      f_i ← 1
    end if
    Broadcast(i, p_i, f_i)
  end if
end while
```

5.3 Properties

The complexity of the DTCC algorithm run by each node i is polynomial in the total number of nodes n. Let us note Δ the maximal node degree in the graph G, that is, $\Delta = \max_{i=1 \ldots n} |N_i|$. The complexity of DTCC is $O(\Delta^4)$. This is because for a node i, there are at most Δ rounds, the time to update Δp_i is at most Δ^2, and during the back-off at most Δ neighbor updates can be received.

When a node i finishes executing ITCC algorithm, it decides its final transmission range p_i. Using this transmission range, the algorithm assures that node i fully covers all the nodes in $N(i)$, using the CC model. The coverage relationship is transitive. For any three nodes p, q and r, if p fully covers q and q fully covers r, then p fully covers r as well. Next, we show the correctness of the ITCC algorithm:

Theorem 5: *The power level assignment provided by the ITCC algorithm guarantees a strongly-connected topology with the CC model.*

Proof: Let us assume by contradiction that the resulting topology is not strongly connected, that is, there exist two nodes i and j such that a message sent by node i is not fully received by the node j, using CC.

Note that G is strongly connected, that means there is a path in G from i to j, $i_0 = i, i_1, i_2, \ldots, i_m = j$, such that $i_{k+1} \in N(k)$ for any $k = 0 \ldots m-1$. When algorithm ITCC ends, each node i fully covers all nodes in $N(i)$ using the CC model. Therefore, each node i_k on the path fully covers the successor node i_{k+1}, for $k = 0 \ldots m-1$. Since the coverage relationship is transitive, it follows that $i = i_0$ fully covers $j = i_m$ using the CC model. Thus, our assumption is false, and the topology resulted after applying ITCC algorithm is strongly connected. □

ITCC algorithm differs from the DTCC algorithm (see section 4) in the following aspects:

- DTCC uses 2-hop neighborhood information, while ITCC uses 1-hop neighborhood information.
- DTCC starts from the power needed to reach the furthest 1-hop neighbor and increases this value in order to reduce the power needed by its children. ITCC starts from the power needed to reach the closest 1-hop neighbor and increases this value incrementally until its 1-hop neighborhood is fully covered.
- DTCC is executed in one round, while ITCC executes over at most Δ rounds.

5.4 Two Special Cases

We have applied the ITCC algorithm to two starting topologies, DMST (Distributed MST) and LMST (Localized MST).

First, we apply the ITCC algorithm to the topology G generated by DMST and note this algorithm with *DMST-based ITCC*. Since DMST-based ITCC starts from a MST-based topology and improves it, using the CC model, DMST-based ITCC has a performance ratio of $2/k$ for the TCC problem (see Theorem 4 in section 4.4).

Then, we apply the ITCC algorithm to the topology G generated by $LMST$ and name this algorithm *LMST-based ITCC*. LMST-based ITCC is a distributed and localized algorithm since both LMST and ITCC are distributed and localized. Another important observation is that the degree of any node in the resulting topology G is bounded by 6 [13]. Therefore, each node i has $|N_i| \leq 6$, and thus $\Delta \leq 6$. The complexity of the LMST-based ITCC is

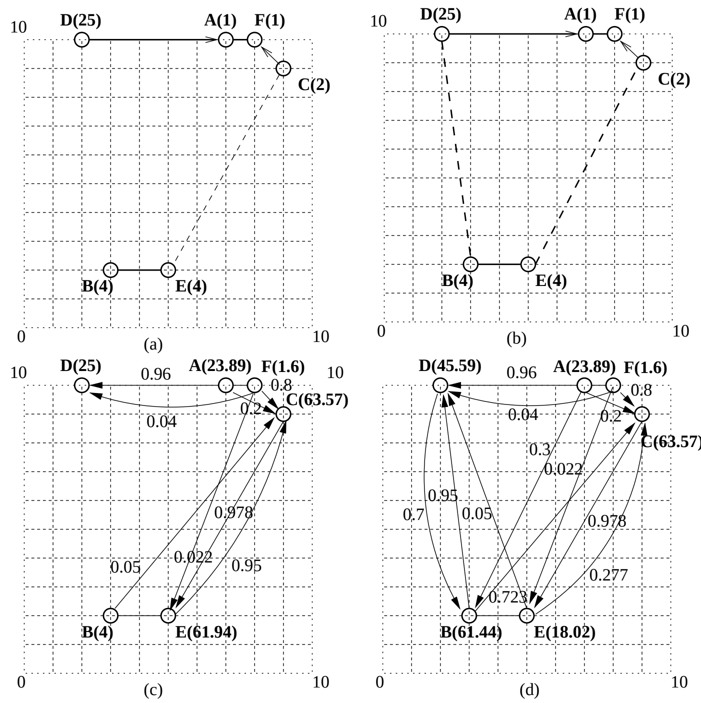


Figure 4: Example of ITCC ($\gamma_{acq} = 0.01$, $\alpha = 2$). (a) DMST and power consumption. (b) LMST and power consumption. (c) DMST-based ITCC. (d) LMST-based ITCC.

therefore $O(1)$.

We use the same example as in Figure 3 to show how ITCC algorithm works. Figure 4 (a) is the initial power assignment of DMST-based ITCC. The graph is disconnected with this power assignment (shown in solid lines), since each node can only reach its closest neighbor. Each node then increases its power until every neighbor is covered. Figure 4 (c) is the result. For example, initially, node F 100% covers its neighbor A and 50% covers neighbor C. It then increases its power to 1.6 to 80% cover C, because the fully covered neighbor A contributes additional 20% coverage. The final total power obtained is 180.

Figure 4 (b) is the initial power assignment of LMST-based ITCC and Figure 4 (d) is the resultant power assignment. The final total cost obtained is 214.11.

6 Simulation Results

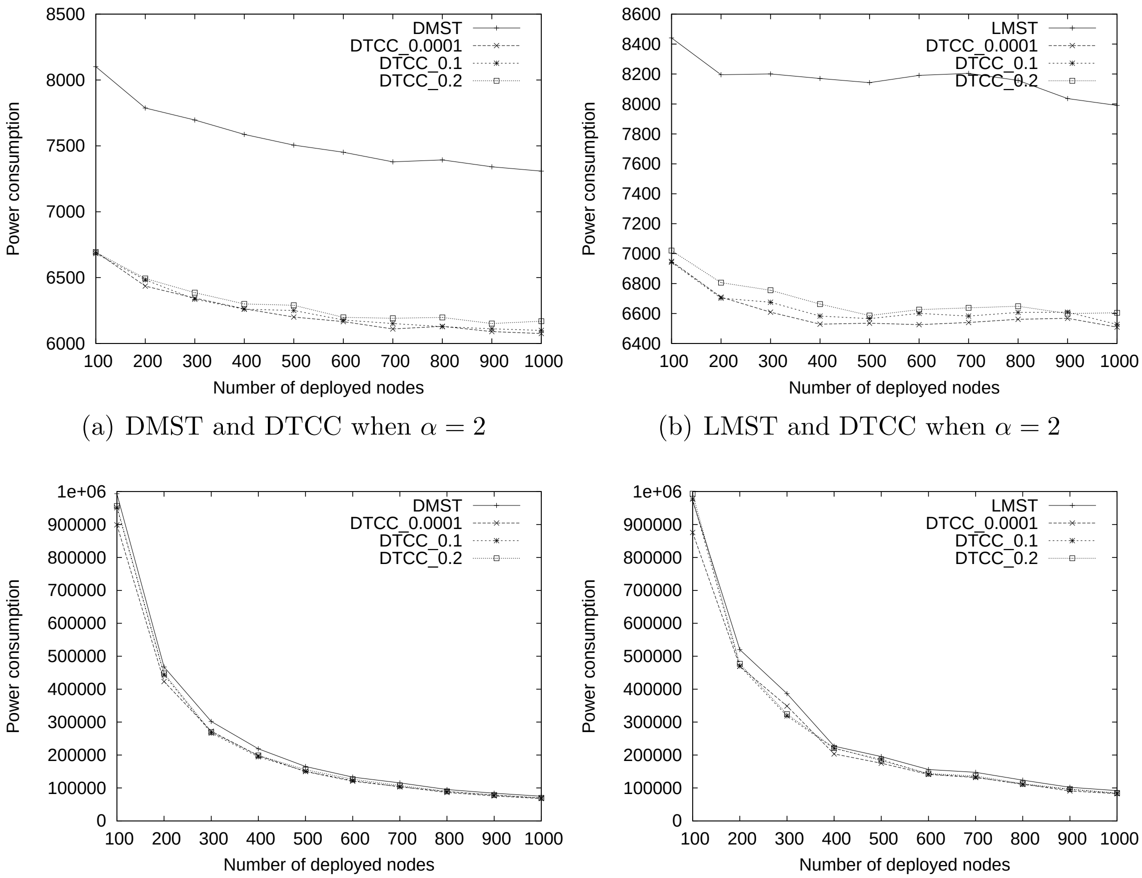
In this section, we evaluate the DMST-based DTCC, LMST-based DTCC, DMST-based ITCC, and LMST-based ITCC algorithms for topologies up to 1000 nodes. We set up our simulation in a $100 \times 100 m^2$ area. The nodes are randomly distributed in the field and remain stationary once deployed. We use both DMST and LMST algorithms in the simulation to generate the starting topologies and to calculate the initial power assignment. Since a localized algorithm lacks global information, the topology obtained when running LMST will be less efficient than DMST, that is the power consumption with LMST will be greater than that using DMST. In the simulation, we consider the following tunable parameters:

1. The node density. We change the number of deployed nodes from 100 to 1000 to check the effect of node density on the performance.

2. The index exponent α, which shows the relation between distance and power consumption. We use the values 2 and 4.

3. The parameter γ_{acq}, which depends on actual wireless communication. In the simulation we use the values 0.0001, 0.1, and 0.2.

Figures 5 (a) and (b) show power consumption depending on the number of nodes, when α is 2. Figure 5 (a) illustrates DMST and DMST-based DTCC, and (b) LMST and LMST-based DTCC. We observe that the overall power consumption can be greatly reduced by using the DTCC algorithm. The smaller the γ_{acq}, the better the performance. Power consumed by DMST is less than that consumed by LMST. The node density does not have much effect on the power consumption, especially when there are more than 200 nodes. This is because when there are more nodes, the average distance between nodes is smaller, and so is the average communication power. Therefore, the overall power consumption changes slightly.



(a) DMST and DTCC when $\alpha = 2$

(b) LMST and DTCC when $\alpha = 2$

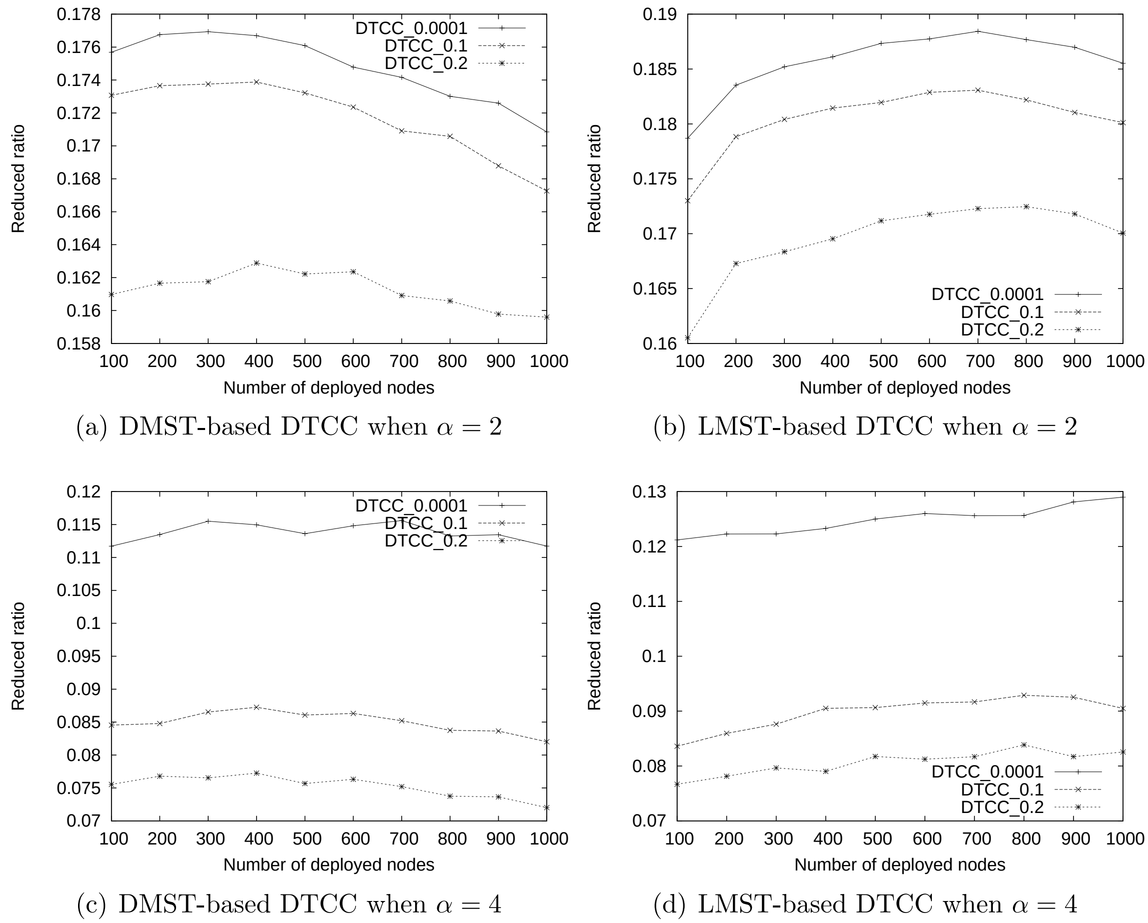
(c) DMST and DTCC when $\alpha = 4$

(d) LMST and DTCC when $\alpha = 4$

Figure 5: Power consumption of DTCC with DMST and LMST ($\gamma_{acq} \in \{0.0001, 0.1, 0.2\}$).

Figures 5 (c) and (d) show the power consumption depending on the number of nodes when α is 4. We can see that the advantage in power efficiency when using DTCC still holds. The difference between power consumption of these two algorithms is less distinctive.

Figure 6 shows the reduced ratio of the consumed power. Figure 6 (a) shows DMST-based DTCC for $\alpha = 2$, and (c) when $\alpha = 4$. Figure 6 (b) represents LMST-based DTCC for $\alpha = 2$, and (d) when $\alpha = 4$. We observe that LMST-based DTCC with an α of 2 achieves the highest reduction in the power consumption, which can be up to 18.6%, while DMST-based DTCC with an α of 4 has the least power reduction.



(a) DMST-based DTCC when $\alpha = 2$

(b) LMST-based DTCC when $\alpha = 2$

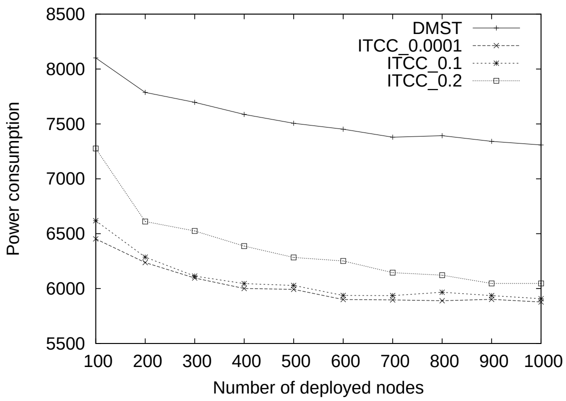
(c) DMST-based DTCC when $\alpha = 4$

(d) LMST-based DTCC when $\alpha = 4$

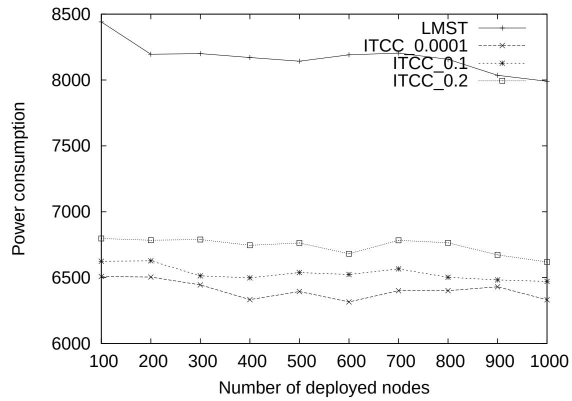
Figure 6: Reduced ratio of DTCC with DMST and LMST ($\gamma_{acq} \in \{0.0001, 0.1, 0.2\}$).

Figures 7 and 8 are the simulation results of ITCC. Figure 7 shows the analysis of power consumption of DMST-based ITCC, LMST-based ITCC, with different α. We can see that this figure is quite the same with Figure 5, except that when α is 2, the effect of parameter γ_{acq} is more significant. Figure 8 shows the reduced ratio of power consumption in ITCC with different γ_{acq}. When α is 2, the LMST-based ITCC can save more than 21.5% of its energy.

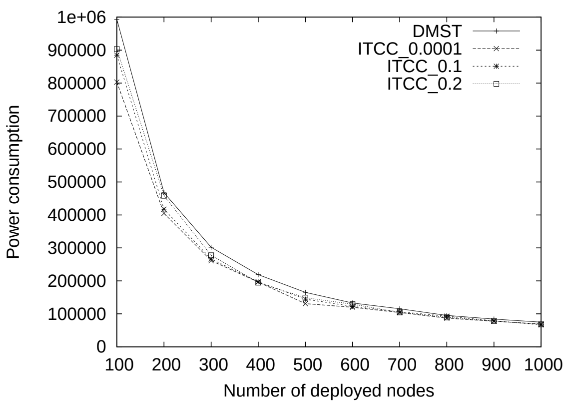
Figure 9 compares the power reduction ratio between DTCC and ITCC. When $\alpha = 2$ and γ_{acq} is relatively small (say smaller than 0.1), ITCC outperforms DTCC. Otherwise, DTCC achieves more power reduction than ITCC. In general, DTCC achieves more energy savings than ITCC since in DTCC the nodes increase their transmission range only once



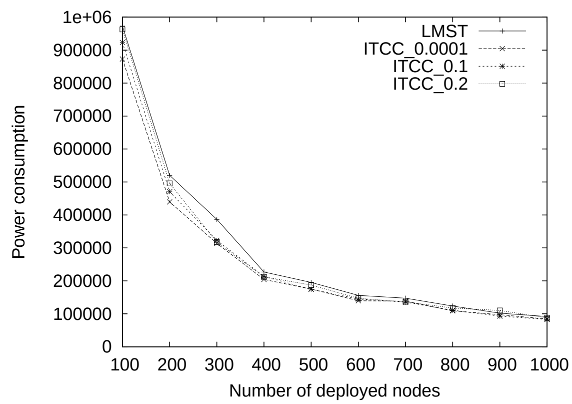
(a) DMST and ITCC when $\alpha = 2$



(b) LMST and ITCC when $\alpha = 2$



(c) DMST and ITCC when $\alpha = 4$

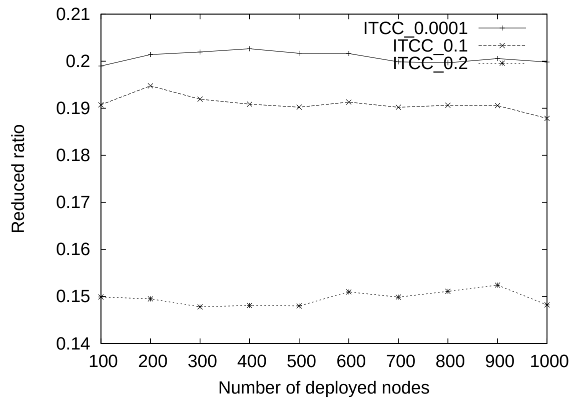


(d) LMST and ITCC when $\alpha = 4$

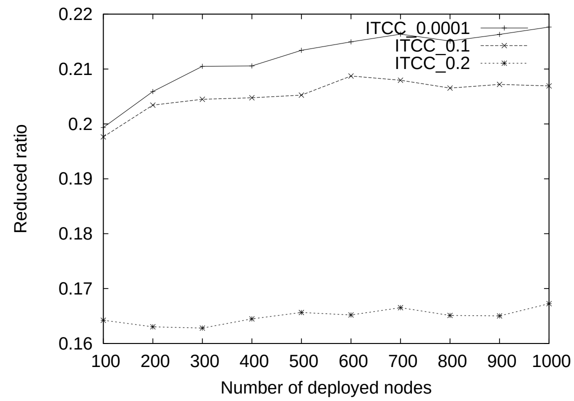
Figure 7: Power consumption of ITCC with DMST and LMST ($\gamma_{acq} \in \{0.0001, 0.1, 0.2\}$).

with a large increment, and therefore the CC contribution on their neighbors is higher. But the difference between these two algorithms is slight.

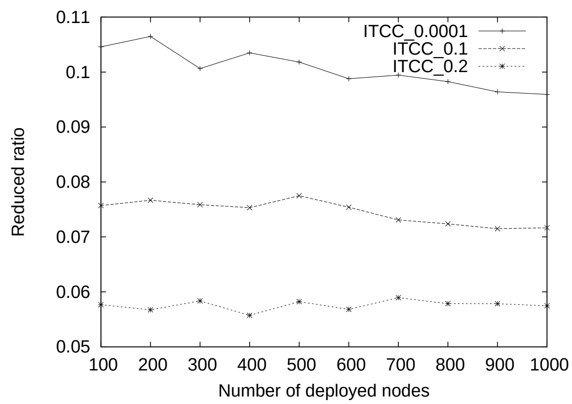
Maximum energy consumption among all the nodes is an important performance metric. It shows whether the energy consumption among all the nodes is balanced or not. Table 3 shows the reduction ratio of ITCC and DTCC in maximum transmission power taken over all the nodes in the network. We can see that the greater the parameter α, the smaller the ratio, and the smaller the γ_{acq}, the greater the ratio. The difference between DTCC and ITCC is slight, but ITCC has a relatively greater reduction. The maximum energy in ITCC is always smaller or equal to the one in the original DMST/LMST topology, while the maximum energy in DTCC can be greater than the original one. This is because ITCC



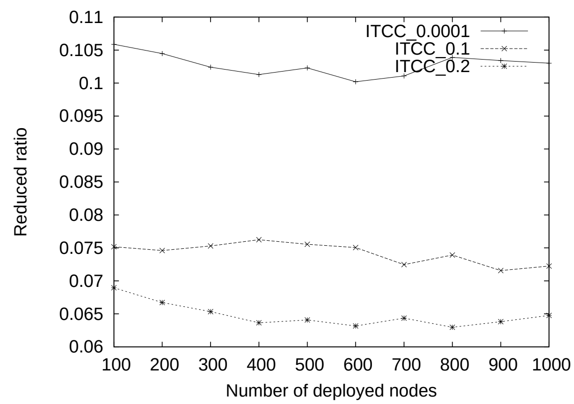
(a) DMST-based ITCC when $\alpha = 2$



(b) LMST-based ITCC when $\alpha = 2$



(c) DMST-based ITCC when $\alpha = 4$



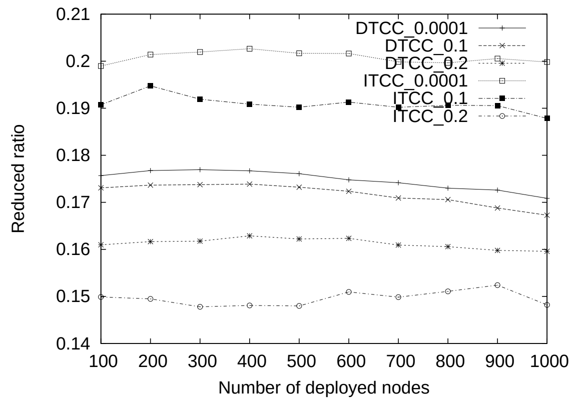
(d) LMST-based ITCC when $\alpha = 4$

Figure 8: Reduced ratio of ITCC with DMST and LMST ($\gamma_{acq} \in \{0.0001, 0.1, 0.2\}$).

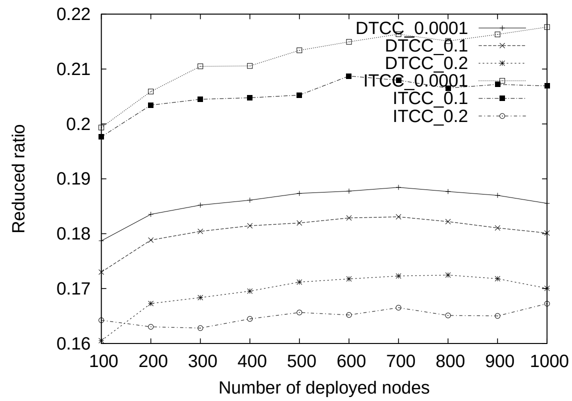
increases the node transmission range gradually and the upper bound of its power is to reach its furthest neighbor. However, in DTCC, a node may increase its power greatly if this can lead to greater reduction of the power of its neighbors. Thus, using ITCC provides a more balanced energy consumption per node, resulting in a longer network lifetime. In general, LMST-based DTCC/ITCC has greater reduction ratio than DMST-based ones.

Simulation results can be summarized as follows:

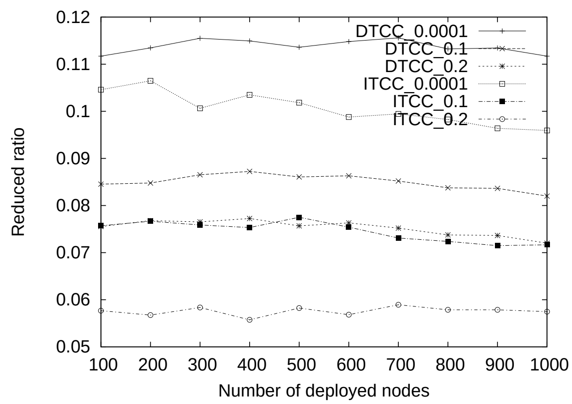
1. Using the CC model, the proposed DTCC and ITCC algorithms reduce the nodes' energy consumption in topology control by 7% to 21%. The LMST-based DTCC or ITCC has greater energy reduction than DMST-based ones.



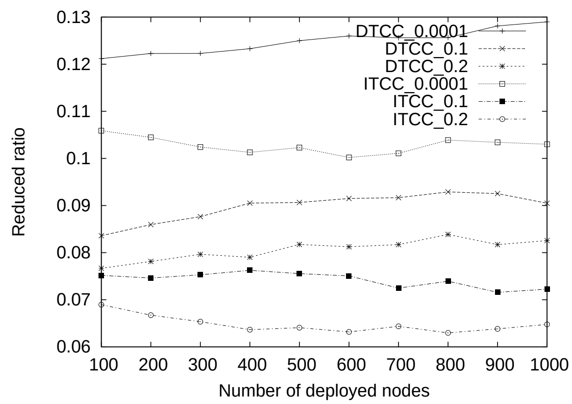
(a) DMST-based DTCC & ITCC when $\alpha = 2$



(b) LMST-based DTCC & ITCC when $\alpha = 2$



(c) DMST-based DTCC & ITCC when $\alpha = 4$



(d) LMST-based DTCC & ITCC when $\alpha = 4$

Figure 9: Reduced ratio comparison of DTCC and ITCC with DMST and LMST ($\gamma_{acq} \in \{0.0001, 0.1, 0.2\}$).

2. With $\alpha = 2$, DTCC and ITCC achieve better performance than with $\alpha = 4$. The former is around 17%, and the latter around 9%.

3. The energy reduction ratio is not sensitive to the parameter γ_{acq} when γ_{acq} is very small; there is no difference between 0 and 0.0001 of γ_{acq}'s value. With increasing values of γ_{acq}, the energy reduction ratio will reduce slightly.

4. The energy savings produced by DTCC and ITCC are comparable, with DTCC producing slightly better results in general. But ITCC has a smaller maximum node power which is good for balanced energy consumption.

Table 3: Reduction ratio of maximum transmission power among all nodes.

DMST-based	$\alpha = 2$			$\alpha = 4$		
γ_{acq}	0.0001	0.1	0.2	0.0001	0.1	0.2
DTCC	0.004	0.003	0.001	0.0005	0.0003	0.000006
ITCC	0.024	0.015	0.009	0.001	0.0004	0.00002
LMST-based	$\alpha = 2$			$\alpha = 4$		
γ_{acq}	0.0001	0.1	0.2	0.0001	0.1	0.2
DTCC	0.042	0.031	0.006	0.008	0.002	0.00003
ITCC	0.064	0.045	0.011	0.009	0.004	0.0002

7 Conclusions

In this paper, we have addressed the NP-complete problem on Topology Control with Cooperative Communication (TCC) in ad hoc wireless networks, with the objective of minimizing the total energy consumption while obtaining a strongly-connected topology. Power control impacts energy usage in wireless communication with an effect on battery lifetime, which is a limited resource in many wireless applications. We have proposed two distributed and localized algorithms that can be applied to any symmetric, strongly-connected topology in order to reduce the total power consumption. The first one uses a distributed decision process at each node that makes use of only 2-hop neighborhood information. The second uses the cooperative communication of nodes within a 1-hop neighborhood in order to set nodes' transmission ranges iteratively, in at most six rounds. We have analyzed the performance of our algorithms through simulations. Our future work is, by starting from DTCC or ITCC algorithm, to design an efficient topology maintenance mechanism that effectively adapts to a dynamic and mobile wireless environment.

References

[1] M. Agarwal, J. H. Cho, L. Gao, and J. Wu, Energy Efficient Broadcast in Wireless Ad hoc Networks with Hitch-hiking, *Proc. of IEEE INFOCOM*, 2004.

[2] M. Cardei, J. Wu, and S. Yang, Topology Control in Ad hoc Wireless Networks with Hitch-hiking, *The First IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON04)*, Oct. 2004.

[3] W.-T. Chen and N.-F. Huang, The strongly connecting problem on multihop packet radio networks, *IEEE Transactions on Communications*, 37(3), pp 293-295, Mar. 1989.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, 2nd edition, *McGraw-Hill*, 2002.

[5] R. Gallager, P. Humblet, and P. Spira, A Distributed Algorithm for Minimum-Weight Spanning Tree, *ACM Transactions on Programming Languages and Systems*, 1983.

[6] T.-C. Hou and V. O. K. Li, Transmission Range Control in Multihop Packet Radio Networks, *IEEE Transactions on Communications*, 34(1), pp 38-44, Jan. 1986.

[7] L. Hu, Topology Control for Multihop Packet Radio Networks, *IEEE Transactions on Communications*, 41(10), pp 1474-1481, 1993.

[8] X. Jia, D. Li, and D.-Z. Du, QoS Topology Control in Ad Hoc Wireless Networks, *Proc. of IEEE INFOCOM*, 2004.

[9] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen, A Survey of Energy Efficient Network Protocols for Wireless Networks, *Wireless Networks*, 7(4), pp 343-358, Aug. 2001.

[10] L. M. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc, Power consumption in packet radio networks, *STACS 1997*, LNCS 1200, pp 363-374.

[11] N. Laneman, D. Tse, and G. Wornell, Cooperative Diversity in Wireless Networks: Efficient Protocols and Outage Behavior, *IEEE Transactions on Info. Theory*, 2003.

[12] L. Li, J. Y. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer, Analysis of a Cone-Based Distributed Topology Control Algorithm for Wireless Multi-Hop Networks, *Proc. ACM Symposium on Principles of Distributed Computing*, pp 264-273, Aug. 2001.

[13] N. Li, J. Hou, and L. Sha, Design and Analysis of an MST-based Topology Control Algorithm, *IEEE Infocom'03*.

[14] X.-Y. Li, Topology Control in Wireless Ad Hoc Networks, Ad Hoc Networking, edited by S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, *IEEE Press*, 2003.

[15] A. Nosratinia, T. E. Hunter, and A. Hedayat, Cooperative Communication in Wireless Networks, *IEEE Communications Magazine*, Vol. 42, No. 10, Oct. 2004, pp 74-80.

[16] J. G. Proakis, Digital Communications, 4th edition, *McGraw Hill*, 2001.

[17] T. S. Rappaport, Wireless communications, 2nd edition, *Prentice-Hall*, 2002.

[18] R. Ramanathan and R. Rosales-Hain, Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment, *Proc. of IEEE INFOCOM*, pp 404-413, 2000.

[19] P. Santi, D. M. Blough, and F. Vainstein, A Probabilistic Analysis for the Range Assignment Problem in Ad Hoc Networks, *ACM Mobihoc*, pp 212-220, Aug. 2000.

[20] C.-C. Shen, and Z. Huang, Topology Control for Ad Hoc Networks: Present Solutions and Open Issues, Handbook of Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks, edited by J. Wu, *CRC Press*, 2005.

[21] J. Wu, M. Cardei, F. Dai, and S. Yang, Extended Dominating Set and Its Applications in Ad Hoc Networks Using Cooperative Communication, accepted to appear in *IEEE Transactions on Parallel and Distributed Systems.*