Topology Control in Ad hoc Wireless Networks with Hitch-hiking

Mihaela Cardei, Jie Wu, and Shuhui Yang
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431
Email: {mihaela, jie}@cse.fau.edu, syang1@fau.edu

***Abstract*— In this paper, we address the Topology Control with Hitch-hiking (TCH) problem. Hitch-hiking [1] is a novel model introduced recently that allows combining partial messages to decode a complete message. By effective use of partial signals, a specific topology can be obtained with less transmission power. The objective of the TCH problem is to obtain a strongly-connected topology with minimum total energy consumption. We prove the TCH problem to be NP-complete 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. We analyze the performance of our approach through simulation.**

Keywords: Ad hoc wireless networks, energy efficiency, Hitch-hiking model, topology control.

I. 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 energy utilization of wireless node has received significant attention [8]. 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: scheduling the wireless nodes to alternate between the active and sleep mode, and 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 all times. This requires that there exists for each node 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 resultant 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. This model is called Hitch-hiking and has been introduced recently in [1]. By an effective use of the partial signals, a specific topology can be maintained with less transmission power.

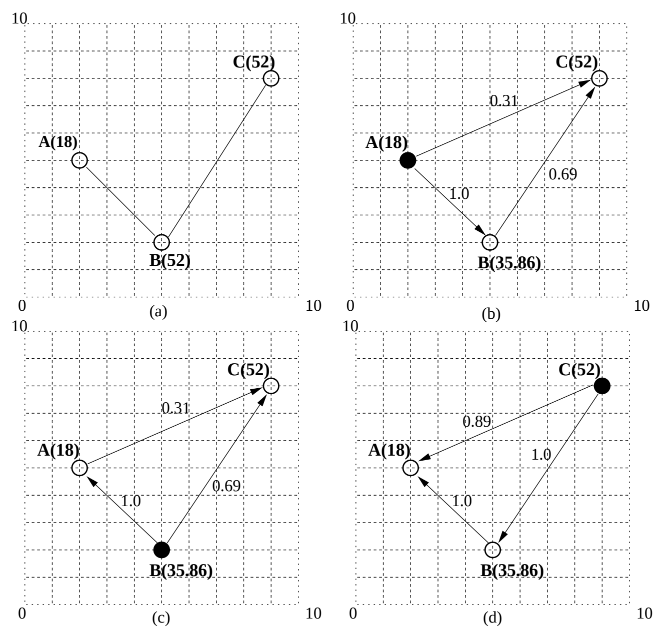


Fig. 1. Three nodes Hitch-hiking example. (a) Initial power consumption based on MST. (b) Power consumption with A as source. (c) B is the source. (d) C is the source.

Figure 1 is a simple example to show the concepts of Hitch-hiking and strong connectivity in a Hitch-hiking model. We assume the power to communicate between two nodes to be the square of the distance between them. The number on each edge represents the coverage provided by that edge to the destination node. In Figure 1 (a), a minimum spanning tree (MST) is formed among the three nodes, where each undirectional link corresponds to two unidirectional links (also symmetric 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. Nodes are strongly-connected if with any one of them as the source node, all the others can get its message directly or by forwarding. In a model with Hitch-hiking, as in Figures 1 (b), (c) and (d), communication power 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.

In this model, only the nodes that have received a complete message can forward it. For example, in (b), node A has a power of 18 to fully cover B ($18 = 3^2 + 3^2$) and to 31% cover C ($31\% = 18/(7^2 + 3^2)$). Since B has received the complete message, it can forward the message to C, providing 69% coverage with power level set to $52 \times 6\% = 35.86$. Thus C gets the complete message. Using the same idea, the other two nodes can be fully covered if we select node B or C as the source node as in (c) and (d). Therefore, the graph is strongly-connected with Hitch-hiking.

Our contributions in this paper are to:

1) define the Topology Control with Hitch-hiking (TCH) problem,
2) prove that TCH is NP-complete and show an upper bound of the performance ratio between the optimal solutions of the TCH problem and the topology problem without Hitch-hiking,
3) design a distributed and localized algorithm that can be applied to any strongly-connected topology to reduce the overall power consumption and study its performances through simulations, and
4) prove that an MST-based topology is an approximation algorithm with ratio bound $2/k$ for the TCH problem, where k is a constant defined in section III.

The rest of this paper is organized as follows. In section II, we overview topology control protocols. Section III describes the Hitch-hiking model and the corresponding network model. Also, we present the Topology Control with Hitch-hiking (TCH) problem, prove its NP-completeness, and show the performance ratio between TCH and topology control without Hitch-hiking. We propose a distributed and localized algorithm in section IV. Section V presents the simulation results for the DTCH algorithm, and section VI concludes this paper.

II. 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 the 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 [2], where an approximation algorithm with performance ratio of 2 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 state information.

Most protocols are deterministic. The work in [16] is concerned with the problem of adjusting the node transmission powers such that the resultant topology is connected or biconnected, 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 [10] 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 of 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 [11] 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 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 maximum node degree of 6. An optional phase is provided where the topology is transformed to one with bidirectionl links. An extension is given in [12], where the given network contains unidirectional links.

Among probabilistic protocols, the work by Santi, Blough and Vainstein [17] 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 of the probability of connectivity.

Some variants of the topology control problem have been proposed which include optimizing other objectives as well. Hou and Li in [5] present an analytic model to study the relationship between the throughput and adjustable transmission range. The work in [6] puts forward a distributed and localized algorithm to achieve a reliable high throughput topology by adjusting node transmission power. The issue of minimizing the energy consumption has not been addressed in these two papers. Jia, Li and Du [7] are concerned with determining a network topology that can meet the 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 Hitch-hiking [1]. This model allows combining partial signals containing the same information in order to decode the complete message. We explore this feature in minimizing total power consumption while achieving a strongly-connected topology with Hitch-hiking.

III. Model and Problem Definition

In this section, we introduce the Hitch-hiking model and the corresponding network model. Then, we define the Topology

Control with Hitch-hiking (TCH) problem, prove its hardness and present a performance ratio between TCH and topology control without Hitch-hiking.

A. Hitch-hiking Model

Hitch-hiking [1] 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 maximal ratio combiner [14] has been traditionally used in physical layer design of wireless systems to increase reliability. The Hitch-hiking model introduces two parameters related with SNR (signal to noise ratio): γ_p which is the threshold needs for successfully decoding 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, $k = \gamma_{acq}/\gamma_p$. A packet received with a SNR γ is:

- fully received, if $\gamma_p \le \gamma$
- partially received, if $\gamma_{acq} \le \gamma < \gamma_p$
- unsuccessfully received, if $\gamma < \gamma_{acq}$

Consider that 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:

$$c_{ij} = \begin{cases} 1 & \text{for } \beta > 1 \\ \beta & \text{for } k < \beta \le 1 \\ 0 & \text{for } 0 < \beta \le k \end{cases}$$

where $\beta = \gamma/\gamma_p$. A channel gain is often modelled 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 Euclidian distance between two communicating nodes. For example, consider $k = 0.125$ and $\alpha = 2$. 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 Hitch-hiking model is that if the same packet is partially received n times from different neighbors with $\gamma_{acq} \le \gamma_i < \gamma_p$ for $i = 1..n$ such that $\sum_{i=1}^{n} \gamma_i \ge \gamma_p$ then the packet can be combined by a maximal ratio combiner [14] and can be successfully decoded.

B. 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 Hitch-hiking model. 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. A symmetric, strongly-connected graph is a special type of directed graph, where a link ij exists if and only if link ji exists. That is, connections between two nodes i and j are symmetric, although they may have different transmission power levels.

Between any two nodes i and j there will be a link 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 $p_i = r^\alpha$. Associated with each link $ij \in E$ is the coverage provided by node i to node j, defined as follows:

$$c_{ij} = \begin{cases} 1 & \text{for } p_i/d_{ij}^\alpha \ge 1 \\ p_i/d_{ij}^\alpha & \text{for } k \le p_i/d_{ij}^\alpha < 1 \end{cases}$$

The case $p_i/d_{ij}^\alpha < k$ is not included since an edge will exist only when the SNR of the received signal is at least γ_{acq}, that is $p_i/d_{ij}^\alpha \ge k$. In this paper we consider the cases when α equals 2 and 4, and $\gamma_p = 1$.

C. Topology Control with Hitch-hiking (TCH)

In this section, we address the topology control problem using the Hitch-hiking model. The fully received packet is defined as follows: considering a transmission from a node i to a node j, node j is partially or fully covered by i if $1 > c_{ij} \ge \gamma_{acq}$ and $c_{ij} = 1$, respectively. If, upon combining packets received from one or more neighbors, say, k neighbors, results in a full coverage of node j, i.e. $\Sigma_k p_k/d_{kj}^\alpha \ge 1$, then the packet is fully received.

We then define *strong connectivity* under the Hitch-hiking model. Basically, 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 eventually. 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 the packet will forward it only once. Now we can formally define the Topology Control with Hitch-hiking (TCH) problem as follows:

TCH Definition. *Given an ad hoc wireless network with n nodes and using the Hitch-hiking 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 Hitch-hiking based topology is strongly-connected.*

Figure 1 shows the concept of strong connectivity under the Hitch-hiking model, where $\gamma_{acq} = 0.2$. Figure 1 (a) shows the power level assigned to each node. Figures 1 (b), (c) and (d) respectively show that starting from each node, all other nodes are fully covered.

D. NP-Completeness of the TCH Problem

In [9], Kirousis et al give a formal proof of NP-completeness of general graph version of the topology control problem (GTC), without Hitch-hiking. In order to prove that TCH is NP-complete, we will show that TCH belongs to the NP-class and GTC is a special case of TCH.

Theorem 1: The TCH problem is NP-complete.

Proof: It is easy to see TCH 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 with Hitch-hiking 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 TCH. Recall our previous description of γ_{acq} and γ_p in the subsection III-A. When $\gamma_{acq} = \gamma_p$, we will have no case of partial reception of signals. Thus the TCH problem will be reduced to the GTC problem, where a signal is either fully received or the reception fails. Hence, we say that the GTC problem is a special case of the TCH problem, for $\gamma_{acq} = \gamma_p$.

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

E. Performance Ratio Between GTC and TCH 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 TCH problem, where k was defined in section III-A.

Theorem 2: *The performance ratio between the optimal solution of the GTC problem and the optimal solution of the TCH 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 TCH problem with OPT^{TCH}. It is clear that $OPT^{TCH} \leq OPT^{GTC}$ since the solution set of the TCH problem includes that of the GTC problem. Next, we show that $OPT^{GTC} \leq \frac{1}{k} \cdot OPT^{TCH}$.

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

$$\begin{aligned} SOL &= \frac{1}{k} \cdot OPT^{TCH} \\ &= (r_1 \cdot k^{-\frac{1}{\alpha}})^{\alpha} + ... + (r_n \cdot k^{-\frac{1}{\alpha}})^{\alpha} \\ &= {r'_1}^{\alpha} + {r'_2}^{\alpha} + ... + {r'_n}^{\alpha} \end{aligned} \quad (1)$$

For any node $i = 1..n$ and for any node $j \in N_i^{TCH}$, we have $(\frac{r'_i}{d_{ij}})^{\alpha} = (\frac{r_i \cdot k^{-\frac{1}{\alpha}}}{d_{ij}})^{\alpha} = \frac{1}{k} \cdot (\frac{r_i}{d_{ij}})^{\alpha} \geq 1$. Therefore, all nodes that were previously partially covered in the TCH 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 $OPT^{GTC} \leq \frac{1}{k} \cdot OPT^{TCH}$.

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

IV. Distributed Topology Control with Hitch-hiking (DTCH) Algorithm

In this section, we propose a distributed topology control with Hitch-hiking (DTCH) 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 will be able to easily adapt the algorithm to a dynamic and scalable architecture. In describing the algorithm, we use the notations in Table I.

TABLE I
DTCH 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

A. Basic Ideas

In DTCH, each node independently "locks" its 1-hop neighborhood to perform power adjustment to save energy. We take node i as the current node for example (see 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 DTCH is to increase i's power level to "contribute" the coverage of its 2-hop neighbors so the range of i's 1-hop neighbors can be reduced, and the overall power consumption can also be reduced. To ensure connectivity, 1-hop neighbors, say j, should still be able to reach i directly. Such a process is the *2-hop power reduction process*. Each node performs this process once and gets its final power level. 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. 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 backs-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 their corresponding 1-hop neighborhood. The 1-hop neighbors of i may have new power levels during i's 2-hop power reduction process, but will not finalize their power levels until themselves perform this reduction process.
2) *Locking scheme*. Node i needs to secure locks of all its neighbors (in addition to its own lock). Once i completes its power reduction process, it announces the final power level of itself and new power levels of its neighbors to their corresponding 1-hop neighborhood, and releases its lock and the locks of its neighbors. Unlike the back-off

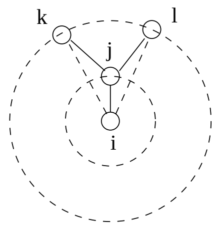


Fig. 2. Illustration of 2-hop neighbor set of i.

scheme that may exhibit occasional mis-coordination, the locking scheme guarantees that nodes execute the 2-hop power reduction process without conflict. However, the locking scheme is more expensive to implement.

B. Detailed Algorithm

The TCH algorithm starts from a symmetric, strongly-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.

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

The goal of the DTCH algorithm is, by starting from an initial power p_i^0 needed to reach its furthest 1-hop neighbor for each node i, to decide the final power assignment by using the Hitch-hiking advantage such that the total power is minimized. 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 decrease in the total power, obtained by increasing node i's transmission power level to p, in exchange for a decrease of the power levels of some other nodes. 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 which is 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 any of its neighbor j to d_{ij}^{α}, by providing the additional coverage needed to a full coverage of the neighbors of j. This can be done in procedure $ComputeP(i)$.

The process of computing the gain is performed for each power level $p \in P(i)$. Once the gain for all power levels in $P(i)$ is determined, the node selects the power 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.

Algorithm DTCH(i)
$p_i \leftarrow p_i^0$
$f_i \leftarrow 0$
while $f_i = 0$
  ComputeP(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 $p_i \leftarrow p_i^{new}$
      $f_i \leftarrow 1$
  Broadcast(i, p_i, f_i)
End_DTCH.

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

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
for all $k \in N(j)$
  $p_j^{red}(k) \leftarrow (1 - c_{ik}) \times d_{jk}^{\alpha}$
return $\max\{d_{ij}^{\alpha}, \max_{k \in N(j)} p_j^{red}(k)\}$
End_Reduce.

C. Properties

Next, we show that the complexity of the DTCH algorithm run by each node i is polynomial in the total number of nodes n. The complexity of the $Gain(i)$ procedure takes $O(|P(i)| \times |\Delta|^2)$, 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 DTCH run on each node is $O(\Delta^4)$ with another loop.

Theorem 3: *The power level assignment provided by the DTCH algorithm guarantees a strongly-connected topology with the Hitch-hiking 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.

First, we note that there are two cases when a node's power level may change in the DTCH algorithm: (a) in line 10, but here the value is increased, so this will not affect connectivity, and (b) in line 9, when a node's power level may be reduced.

Let us assume by contradiction that after applying the DTCH 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, and let us note with $i_0 = i$, i_1, ..., $i_m = j$ a path 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 DTCH, 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 DTCH. □

D. Two Special Cases

We have applied the DTCH algorithm on two starting topologies output by two distributed algorithms: DMST (Distributed MST) and LMST (Localized MST). Again, a localized algorithm is a special distributed algorithm without sequential propagation. We note with DMST the Gallegar's distributed algorithm [4] for constructing an MST, and with DMST-based DTCH, the DTCH algorithm starting from a topology G generated by DMST.

MST has been considered before as a reference point in designing topology control mechanisms in the general model (without Hitch-hiking) because of its important properties and good performances. MST has the minimum longest link among all the spanning trees [3], 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 important property of the MST-based topology in the general case (without Hitch hiking) is that it provides an approximation algorithm with performance ratio of 2 [9].

Next, we prove that an MST-based topology has a performance ratio of $2/k$ for the TCH problem. We refer to the 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 as MST-based topology.

Theorem 4: *An MST-based topology is an approximation algorithm with ratio bound of $2/k$ for the TCH 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 TCH problem with OPT^{TCH}, and the MST-based solution with MST.

It is proved in [9] 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 prove that $OPT^{GTC} \leq \frac{1}{k} \cdot OPT^{TCH}$, therefore, $MST \leq \frac{2}{k} \cdot OPT^{TCH}$. Since $OPT^{TCH} \leq MST$, we obtain that $OPT^{TCH} \leq MST \leq \frac{2}{k} \cdot OPT^{TCH}$ and the theorem holds. □

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

Secondly, we apply the DTCH algorithm to a symmetric strongly-connected topology [1] produced by the LMST algorithm, and refer to this case as LMST-based DTCH. LMST is a localized algorithm introduced by Li et al [11] as discussed in Section II. As DTCH is also localized, the resultant LMST-based DTCH algorithm is localized and distributed. We present the simulation results for LMST-based DTCH in section V-B. Note that if DTCH is applied on DMST or LMST, the complexity is $O(1)$. This is because in LMST and DMST, the degree of any node in the resultant topology is bounded by 6. Therefore, the number of power level of node i, $|P(i)|$, in DTCH is constant. The complexity of DTCH in the general case is $O(|P(i)| \times |N(i)|^2)$, which is $O(1)$ here.

Let us now present an example with a topology consisting of six nodes, distributed as in Figure 3. The number on each node indicates the power level used by that node in maintaining the topology, based on DMST in (a) and LMST in (b). To simplify the picture, we use undirectional links when the coverage in both directions is 1, which refers to full coverage, 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 Hitch-hiking. The power level assigned to each node is the power needed to reach the furthest neighbor in DMST. In this case, we obtain a total cost of 186. In Figure 3 (b), we show the topology obtained after using the LMST algorithm [11], with a total cost of 287. LMST uses a localized way to generate the MST; every node decides its 1-hop neighbors independently. Therefore, in a global view, the MST might be a graph.

In Figure 3 (c), we show the topology and power assignment after running the DMST-based DTCH algorithm. We assume $\gamma_{acq} = 0.01$. First, each node computes its $gain$. 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)),

[1]which means all unidirectional links can be removed without impairing the network connectivity.

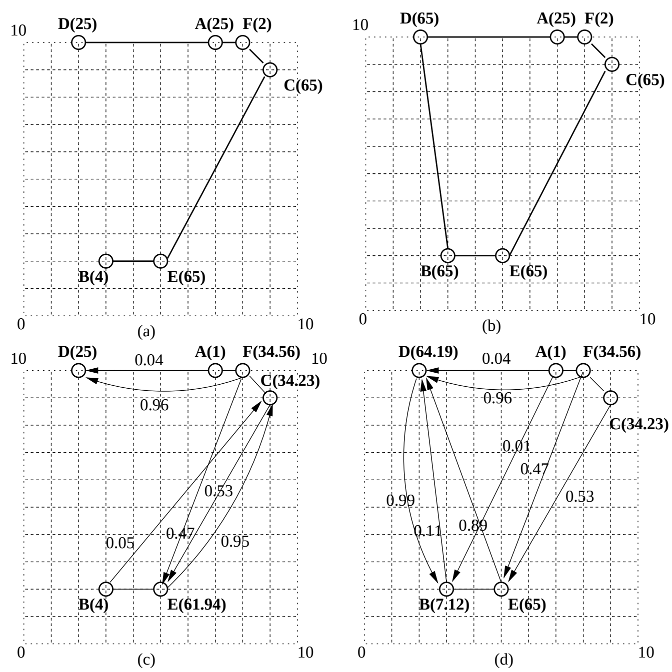


Fig. 3. Example for topology control with and without Hitch-hiking. (a) DMST and power consumption. (b) LMST and power consumption. (c) DMST-based DTCH. (d) LMST-based DTCH.

while preserving the strong connectivity. 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) illustrates the execution of the LMST-based DTCH algorithm. We obtain a total cost of 206.1 and a reduction ratio of 28.19% compared with the output of the LMST algorithm (in Figure 3 (b)), while preserving the strong connectivity.

V. Simulation Results

We present the results of our simulation based on the size of the network. Subsection V-A models the TCH problem as a minimization constrained problem and presents results of several "toy" examples for small scale topologies, with up to 8 nodes. Subsection V-B shows results for larger scale topologies, when the number of nodes varies between 100 and 1000.

A. Small Scale Network Topologies

In this section, we formulate the TCH problem as a constrained minimization problem that is solved and implemented using the optimization toolbox in Matlab [13]. Then, we compare the results obtained by running DMST-based DTCH and DMST with the results obtained using Matlab, for small scale topologies. Results obtained using Matlab are optimal solutions, so this experiment will be an indication of how DMST and DMST-based DTCH perform.

The parameters includes set V of n nodes and their locations, and p_{max} the maximum power level that can be assigned to a node. We also assume $\gamma_{acq} > 0$, $\gamma_{acq} \to 0$ and $\gamma_p = 1$. Assigning a very small value to γ_{acq} results in having any node participating in the coverage of any other node. Let p_i for $i = 1..n$, $p_i \in \Re$, represent the power level of every node i. f_{ij}^m for $m, i, j = 1..n$, are binary variables. $f_{ij}^m = 1$ if any packet sent from node m will be fully received by node i after j steps; otherwise, $f_{ij}^m = 0$. In order to achieve strong connectivity with Hitch-hiking we need to have a "path" from any node to any other node. Therefore, a packet sent by any node must be fully received by any other node after a number of steps. The maximum number of steps is $n-1$, as we will argue later. Next, we present TCH as a constrained minimization problem:

$$\begin{array}{ll} \text{minimize} & p_1 + p_2 + ... + p_n \\ \text{subject to} & (1) \quad X_m, (m = 1..n) \\ & (2) \quad p_{max} \geq p_i > 0, (p_i \in \Re) \end{array}$$

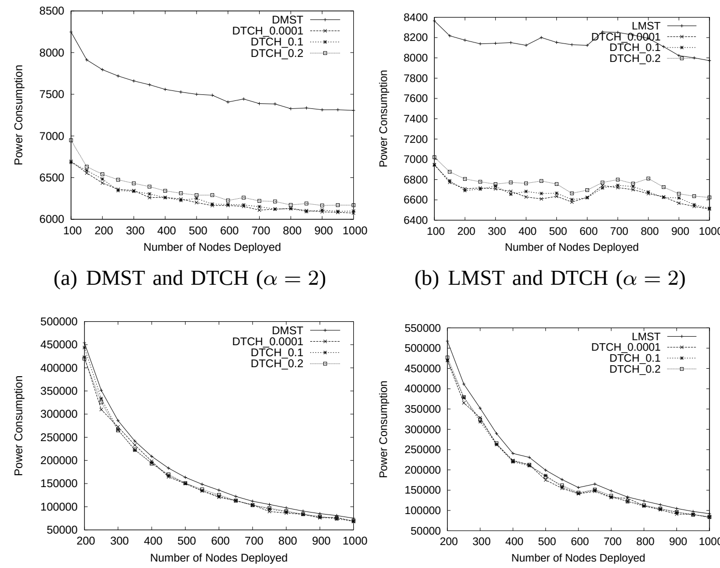
where X_m is a set of conditions, defined as follows:

$$\begin{array}{ll} (3) & f_{1n}^m = f_{2n}^m = ... = f_{nn}^m = 1 \\ (4) & f_{i1}^m = 0, (i = 1..n, i \neq m) \\ (5) & f_{m1}^m = 1 \\ (6) & f_{i(j-1)}^m \leq f_{ij}^m \leq f_{i(j-1)}^m + \sum_{\substack{k=1..n \\ k \neq i}} f_{k(j-1)}^m \cdot \frac{p_k}{d_{ki}^\alpha}, \\ & (i = 1...n, j = 2...n) \\ (7) & f_{ij}^m = 0 \ or \ 1 \end{array}$$

The problem tries to minimize total power in the network. Constraint (1) is further expanded in conditions (3) through (7) and basically requires that from any node m there should be a route to any other node in the network. Therefore, a packet transmitted by m should be fully received by all other nodes in the network after at most $n-1$ steps. For a variable f_{ij}^m, m represents the source node currently considered, i represents a destination and j is for the step number. A packet sent by a node k is received by node i with fraction p_k/d_{ki}^α.

Let us assume now that a node m transmits a packet. Then, for a strongly-connected topology, any other node should be able to fully receive this packet in at most $n-1$ steps. Also, only the nodes that fully received a packet are able to forward the packet. We also assume that partial messages are stored by the receiver node. $f_{ij}^m = 0$ means that node i does not receive the packet by step j. $f_{ij}^m = 1$ means that node j fully received the packet at step j, and from condition (6) this will result in $f_{i(j+1)}^m = ... = f_{in}^m = 1$. As we can see in the inequality (6), and because variables f are binary, only nodes that fully received a frame will contribute in other nodes partial frame receipts.

Condition (3) asks that all nodes fully received the packet after step n, by asking $f_{in}^m = 1$, for any $i = 1..n$. Conditions (4) and (5) states that initially (step 1, when $j = 1$) only the source node has the full packet. Another observation is that if all other nodes will receive the packet, then this will happen within at most $n-1$ steps. This is because there are $n-1$ nodes that have to receive the packet, and at every step at least



(a) DMST and DTCH ($\alpha = 2$) (b) LMST and DTCH ($\alpha = 2$)

(c) DMST and DTCH ($\alpha = 4$) (d) LMST and DTCH ($\alpha = 4$)

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

Fig. 4. Results for topologies with 3 to 8 nodes.

one more node fully receives the packet, otherwise, there will exist one or more nodes that will not fully receive this packet.

In the simulations, we consider 8 nodes randomly distributed into an 1×1 Km^2 area, as illustrated in Figure 4 (a), using $\gamma_{acq} = 0.0001$, $\gamma_p = 1$ and $\alpha = 2$. In Figure 4 (b), we represent the total energy consumed for topologies between 3 and 8 nodes by using the first 3 nodes of the 8 nodes, then the first 4, the first 5, and so on. When the number of nodes is between 3 and 7, Matlab converges to the optimal solution, whereas for the 8 node topology we show the result after 643 iterations. Considering this node distribution, DMST-based DTCH results are within 15% of the optimal solution and provide an overall reduction in energy consumption of up to 17.5% compared with the DMST-based solutions.

B. Large Scale Network Topologies

In this section, we evaluate the DMST-based DTCH algorithm and LMST-based DTCH algorithm for large scale topologies, up to 1000 nodes. We set up our simulation in a 100×100 m^2 area. Nodes are randomly distributed in the field initially and will 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 the localized algorithm lacks global information, the topology obtained when running LMST will be less efficient than DMST. Therefore, the power consumption with LMST will be greater than that of DMST theoretically. 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.
3) The parameter γ_{acq}, which depends on actual wireless communication. We use 0.0001, 0.1 and 0.2 as its value in the simulation.

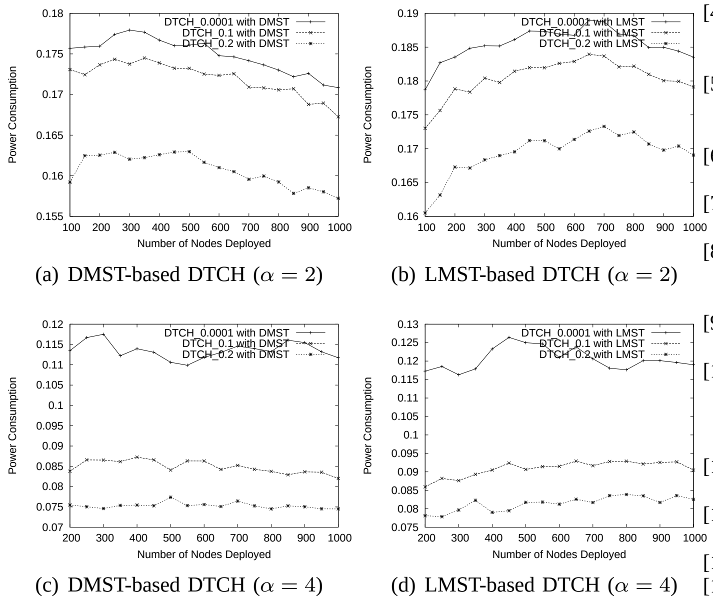
Figures 5 (a) and (b) show the power consumption depending on the number of nodes, when α is 2. Figure 5 (a) illustrates DMST and DMST-based DTCH, and (b) LMST and LMST-based DTCH. We observe that the overall power consumption can be greatly reduced by using the DTCH algorithm. The smaller the γ_{acq},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

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 DTCH still holds. The difference between these two algorithms' power consumption is less distinctive.

Figure 6 shows the reduced ratio of the consumption power. Figure 6 (a) shows DMST-based DTCH for $\alpha = 2$, and (c) when $\alpha = 4$. Figure 6 (b) represents LMST-based DTCH for $\alpha = 2$, and (d) when $\alpha = 4$. We observe that LMST-based DTCH with α being 2 achieves the highest reduction in the power consumption, which can be up to 18.5%, while DMST-based DTCH with α being 4 has the least power reduction.

Simulation results can be summarized as follows:

- Using Hitch-hiking, the proposed DTCH algorithm reduces the nodes' energy consumption in topology control by 7% to 19%. The LMST-based DTCH has greater energy reduction than DMST-based DTCH.
- With $\alpha = 2$, DTCH achieves better performance than $\alpha = 4$. The former is around 17%, and the latter around 9%.
- 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 value of γ_{acq}, the energy reduction ratio will reduce slightly.



(a) DMST-based DTCH ($\alpha = 2$) (b) LMST-based DTCH ($\alpha = 2$)

(c) DMST-based DTCH ($\alpha = 4$) (d) LMST-based DTCH ($\alpha = 4$)

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

VI. Conclusions

In this paper, we have addressed the Topology Control with Hitch-hiking (TCH) problem in an ad hoc wireless network with an objective of minimizing the total energy consumption while obtaining a strongly-connected topology. Power control impacts energy usage in wireless communication with effect on battery lifetime, which is a limited resource in many wireless applications. We have proved that TCH is NP-complete and proposed a distributed and localized algorithm that can be applied to any symmetric, strongly-connected topology in order to reduce the total power consumption. Our algorithm uses a distribution decision process at each node that makes use of only 2-hop neighbor information. We have analyzed the performance of our algorithm through simulations. Our future work are, to do some further analysis on other effect of DTCH, such as delay and throughput; by starting from DTCH algorithm, to design an efficient topology maintenance mechanism that effectively adapts to a dynamic and mobile wireless environment.

Acknowledgement

This work was supported in part by NSF grants CCR0329741, ANI 0073736, and EIA 0130806.

References

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

[2] 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.

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

[4] R. Gallager, P. Humblet and P. Spira, "A Distributed Algorithm for Minimum-Weight Spanning Tree", *ACM Transactions on Programming Languages and Systems*, pp 66-77, 1983.

[5] 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.

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

[7] X. Jia, D. Li and D.-Z. Du, "QoS Topology Control in Ad Hoc Wireless Networks", *IEEE Infocom'04*.

[8] 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.

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

[10] 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.

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

[12] N. Li and J. Hou, "Topology Control in Heterogeneous Wireless Networks: Problems and Solutions", *IEEE Infocom'04*.

[13] Matlab, http://www.mathworks.com/

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

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

[16] R. Ramanathan and R. Rosales-Hain, "Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment", *IEEE Infocom'00*, pp 404-413, 2000.

[17] 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.