Power-Aware Topology Control for Wirelss Ad-Hoc Networks

Wonseok Baek
Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089, USA
Email: wbaek@usc.edu

David S. L. Wei
Department of Computer and Information Science
Fordham University
Bronx, NY 10458, USA
Email:wei@dsm.fordham.edu

C.-C. Jay Kuo
Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089, USA
Email: cckuo@sipi.usc.edu

I. INTRODUCTION

In this research, we examine the problem of power-aware topology control for wireless ad-hoc networks. The reliance on wireless multi-hop communications to maintain connectivity among nodes adds new complexity on the design and operation of the wireless ad-hoc networks. In particular, the lack of a physical backbone infrastructure poses a strong need of topology control of the network. Generally speaking, if a protocol is designed based on overlaying a virtual infrastructure on the wireless multi-hop network, its performance can be greatly enhanced.

The power-efficient topology control technique that tries to minimize the total power consumption while maintaining global connectivity has attracted more attention in wireless ad-hoc networks. For example, a localized minimum-spanning-tree(LMST)-based power-efficient algorithm was presented in [1]. However, although the power-efficient topology control algorithm can reduce the total power consumption of a network as a whole, it cannot guarantee that power saving is evenly distributed among all network nodes when a routing protocol is adopted on the top of it. Due to uneven power consumption, a few nodes can be depleted individually so as to potentially break the connectivity of the network and paralyze some portion or even the entire network system. A power-aware approach that allows the power consumption to be evenly distributed among network nodes and, thereby, prolongs the network lifetime is highly desirable. The problem of prolonging the network lifetime, where the network lifetime is defined as the time span from the start of the network to the first death of a node, has been tackled through so-called power-aware routing techniques. For example, the optimal maximum lifetime routing problem was solved through Linear Programming, and two heuristic routing algorithms were introduced in [2]. However, the power-aware routing algorithms have its limitation. It typically requires the exact traffic flow information and residual energy level of nodes, and the complexity is relatively high. Thus, it is hard to implement under the situation where traffic flow changes very often.

Our approach to the problem of prolonging the network lifetime is through power-aware topology control. The main reasons for our approach are three folds. First, through the power-aware topology control, we can still adapt to the very simple routing protocol while prolonging the network lifetime. Second, it is expected that frequent changes in traffic flows(source-destination pairs) do not highly affect the energy consumption of each node due to the limited connectivity and the sparseness of the network resulting from the topology control. Namely, unlike the power-aware routing, we can only consider the residual energy level, not the changes in traffic flow. Third, it is unclear what the connectivity of the network will be after the first death of a node. We argue that the network should keep working until the network is partitioned such that we are interested in how long the network can maintain its global connectivity excluding the dead nodes.

A power-aware topology control algorithm that only requires the residual energy level and location information of reachable neighboring node is presented in this work. When a node is making a decision on whether a wireless link between itself and a reachable neighboring node should be preserved in the topology being constructed, the decision is made based on not only the distance from its neighboring nodes but also the residual energy level of itself and its neighboring nodes. Also, the topology is restructured from time to time based on the residual energy level of each node.

II. PROPOSED ALGORITHM

A. Node Classification

Our power-aware topology control algorithm begins with the observation that the energy of some nodes can be saved if those nodes are free from routing job, and it can be achieved through topology control if we limit the number of links of those nodes to strictly to one. Thus, our power-aware topology control algorithm begins with the idea that builds a power-efficient virtual backbone using the nodes with a relatively high residual energy level and at the same time limits the degree of the nodes with relatively low residual energy level to be one. As a result, one main factor that affects a power-aware topology is how to measure the relative residual energy level and how to divide the nodes into two categories based on the measured relative residual energy levels.

There can be many possible ways for that purpose, but our approach to the measurement of the relative residual energy levels is a local and statistical way: To measure the relative residual energy level, each node locally broadcasts its remaining energy level and location information through a beacon message. It is assumed that each node can accurately estimate its location through GPS or other methods. After gathering the information, it calculates the average and standard deviation of the residual energy level of its own and neighbors. Based on the average and standard deviation, and its own residual energy level, each node can be categorized into one of the following three sets.

- The core node set. If its residual energy level is above the average value minus standard deviation, the node declares itself as a core node.
- The non-core node set. If its residual energy level is below the average value minus standard deviation, the node becomes non-core node. The non-core nodes are further categorized into one of the following two subsets.
    - The active node set. If its residual energy level is below the average value minus standard deviation and at least one core node is within its transmission range, the node declares itself as an active node.

  - The passive node set. If its residual energy level is below the average value minus standard deviation and no core node is within its transmission range, the node declares itself as a passive node.

B. Core Node Connectivity

The core nodes construct a virtual backbone and play the role of a router. Each core node performs the LMST (Local Minimum Spanning Tree) algorithm [1] to build the virtual backbone. After constructing the virtual backbone, core nodes locally broadcast the connectivity information (*i.e.* with the one-hop away neighboring information) through a beacon message. It may be noted that the virtual backbone composed of core nodes may not be connected, and indeed there is no need to have a connected virtual backbone in our algorithm.

1) Property of Core Node Connectivity: It was proved in [1] that, for any node pair $[u, v]$ where $u, v \in V$ and V is the set of nodes performing the LMST algorithm, if $d(u, v) \leq d_{max}$ then there exists a path between u and v. Note that $d(u, v)$ is the distance between u and v and d_{max} is the maximum transmission range of a node. This property can be directly applied to the core set such that if the distance between any core node pair is less than the maximum transmission range, these two core nodes are connected. However, the two core nodes can still be connected through multiple hops even though the distance between them is greater than the maximum transmission range.

C. Non-core node connectivity

The non-core nodes also employ LMST algorithm. Initially, the active nodes and the passive nodes only applies LMST algorithm to the active nodes and the passive nodes, respectively. After constructing local minimum spanning tree and making connection to one-hop neighbors, each non-core node performs the following algorithm.

1) Active Node Connectivity:

a. Each active node applies the LMST algorithm to core nodes only. In other words, it performs the LMST algorithm to construct a local minimum spanning tree only with core nodes excluding other active nodes and passive nodes.
b. Active nodes locally broadcast the connectivity information through the beacon message. The connectivity information contains the list of its one-hop away neighboring core nodes and the location information.
c. After gathering the information of its neighboring active nodes, any active node that has more than one active node as its one-hop away neighboring node performs the following pruning procedure.
   (a) An active node compares the list of its one-hop away neighboring core nodes with that of its one-hop away neighboring active node.
   (b) If there is at least one common core node, it removes the link to the one-hop away neighboring active node.
   (c) If there is no common core node, the active node calculates $d(x, y)$ for all core node pairs $[x, y]$, where node x belongs to the list of its core nodes and node y belongs to the list of its neighboring active node.
   (d) If $\min_{x,y} d(x, y) > d_{max}$, the active node keeps the link to the one-hop away neighboring active node. Otherwise, the active node removes the link.
   (e) Repeat Steps (a)-(d) until a decision is made on every link to its one-hop away neighboring active node.
d. Each active node removes the redundant links to the core nodes through the following pruning procedure.
   (a) Among the links to the core nodes that are obtained in Step a. , the shortest one is always kept.
   (b) For the remaining links, an active node calculates the distance between the closest one-hop away neighboring core node and other one-hop away neighboring core nodes resulted from the constructed MST tree. If the distance is less than the maximum transmission range, d_{max}, its links to other core nodes are removed.
   (c) Following a similar way, an active node keeps the next shortest link among links that are not yet removed. Then, go to Step (b) again.
   (d) Repeat Steps (b) and (c) until a decision is made on every link.
   (e) Finally, each active node uses the connectivity information from its one-hop neighboring core nodes to check if there is a path between core nodes obtained in Steps (b)-(d). If there is a path, the longer link is removed.
e. Finally, each active node locally broadcasts its connectivity information that contains the list of connected neighboring active nodes.

The active node connectivity has two important properties. First, any active node is just one-hop away from a core node. This property is a direct consequence of the constructing algorithm. Second, some active nodes may make connection to more than one core node if there exist more than one core node and the distance between them is greater than d_{max}. Some of these additional links to the one-hop away neighboring core nodes are to connect disjointed sets of core nodes. In the mean time, other additional links to the one-hop away neighboring core nodes may be redundant.

2) Passive Node Connectivity: The algorithm for passive node is more or less the same as that of active node: the passive nodes applies the same algorithm to the active node. Similar to the active node connectivity property, any passive node is just two-hops away from a core node. Also, some passive nodes may have more than one link to connect disjointed sets of core nodes. Thus, some links to neighboring active nodes of the passive node are to connect disjointed sets of core nodes while other links to neighboring active nodes may be redundant.

III. PERFORMANCE EVALUATION

A. Proof for Global Connectivity

Let $G = (V, E)$ be the graph for the original topology. V is the set of all nodes and E is the set of all edges. We assume the original topology is connected. Also, let $G_1 = (V, E_1)$ be the graph for the topology generated by our algorithm but before the pruning procedure where E_1 is the subset of E, and let $G_2 = (V, E_2)$ be the graph for the topology generated by our algorithm after the pruning procedure where E_2 is the subset of E_1.

Lemma 1: G_1 is a connected graph.

Proof: We define CN as the set of all core nodes and NCN as the set of all non-core nodes. Note that CN and NCN are subset of V such that $V = CN \bigcup NCN$. According to our algorithm, each node that belongs to CN performs LMST algorithm. It may be noted that the nodes belonging to CN may not be connected. We define CCN_i as the ith subset of CN such that for any two node u and v that belongs to CCN_i, there exists a path. Also, note that $CN = \bigcup CCN_i$. In similar way, $CNCN_i$ is defined as the ith subset of NCN such that for any two node u and v that belongs to $CNCN_i$, there exists a path and $NCN = \bigcup CNCN_i$.

Note that G_1 is the graph for the topology before each active and passive node perform pruning procedure. We prove the global connectivity by proving that once each active node and passive node in G_1 performs LMST algorithm and makes connections to core nodes and active nodes, there exists a path between any two nodes.

First, we consider each set CCN_i and $CNCN_j$ as a single virtual node and denote CCN_i by vnode_ccn_i and $CNCN_j$ by vnode_cncn_j. Then, the distance between two virtual nodes is defined as $\min_{u,v} d(u,v)$ where real node u belongs to the one virtual node and real node v belongs to the other virtual node. Note that the distance between any two vnode_ccn is greater than d_{max} and the distance between any two vnode_cncn is greater than d_{max}.

Now, consider the algorithm that the passive nodes make connection to the active nodes. This is same as putting edges between vnode_cncn_v and vnode_cncn_w, if the distance between vnode_cncn_v and vnode_cncn_w is less than or equal to d_{max}. We define $CNCN2_i$ as the new subset of NCN resulting from the algorithm that the passive nodes make connection to the active nodes, and denote $CNCN2_i$ by vnode_cncn2_i. Note that these new vnode_cncn2 contains at least one active node.

Next, consider the algorithm that the active nodes make connection to the core nodes. This is the same as putting edges between vnode_ccn_x and vnode_cncn2_y, if distance between vnode_ccn_x and vnode_cncn2_y is less than or equal to d_{max}. Then the resulting graph is connected since every vnode_cncn2 contains at least one active node and every vnode_cncn2 is one-hop away from at least one vnode_ccn (distance is less than d_{max}). If the resulting graph is not connected, it violates the assumption that the original topology is connected. ■

Theorem 1: G_2 is a connected graph.

Proof: The proof for the global connectivity after pruning procedure of active (passive) nodes is straight forward.

The pruning procedure of active (passive) nodes is equivalent to checking the existence of a path for the link considered. The existence of a path is guaranteed by LMST algorithm itself. Thus, whenever active (passive) nodes remove a link, it is guaranteed that there exists a path that replaces the removed link. ■

B. Simulation Results

To compare the LMST topology control algorithm with our power-aware LMST-based topology control algorithm, we consider two performance metrics: network lifetime and network partition time. The network lifetime is defined in the same way as defined in many other literatures, and network partition time is defined as the time span from the start time of the network until the network is partitioned.

The simulations are set up for the randomly distributed 50 nodes with uniform distribution in 750(m)×750(m) grid. The maximum transmission range is set to 250(m). The transmission time of a packet is assumed to be 5 unit times, and beacon message exchange is set to take place at every 100 unit times. The path loss exponent is set to 4 (usually between 3 and 4). It is assumed that the initial energy level of each node is the same for all nodes. We assume random traffic that follows the Poisson arrival process with rate 0.001.

The MAC layer is assumed to be slightly modified 802.11b MAC layer. In this simulation, each node transmit RTC-CTS messages with adjusted power such that it is transmitted with the power to reach the outmost one-hop neighboring node. The routing protocol is assumed to be the slightly modified link-state routing protocol: the required power for transmission between two nodes is used for link cost. In addition to that, each core node looks for alternative path for the packet that is forwarded through non-core node. Whenever it finds the alternative path that consists of core nodes only, it reroute the packet through the new path.

TABLE I

PERFORMANCE COMPARISONS (IN UNIT TIME)

	LMST TC	power-aware TC
average network lifetime	16892.0	32610.4
average network partition time	46410.8	39184.2

See the Table I for the comparison of average network lifetime and average network partition time. As expected, the average network lifetime is improved in huge amount, almost 100%. However, interestingly, LMST topology control algorithm shows better performance in terms of the network partition time. To get some insight about this phenomenon, see the Figure 1 where red circle mark represents LMST topology control and blue star mark represents power-aware LMST-based topology control. In Figure 1, it is shown that how many node is alive as time goes by for a typical topology and random traffic. It is observed that similar results are obtained for different random seeds for the random topology and random traffic. The reason for this phenomenon can be explained as the following. In the power-aware scheme, the energy consumption is evenly distributed over all nodes, and thus most of the nodes have very close residual power level. As such, when a node start to die, most of other nodes with close residual power level are also dying. In addition, to save the energy of some nodes we have to use the energy of other nodes. As a result, after some time, nodes are dying more quickly in a power-aware scheme than in a non-power-aware scheme. (In the ideal situation where the energy consumption rate is same for all nodes, all nodes should die all at once.)

REFERENCES

[1] N. Li, JC Hou, and L. Sha, "Design and analysis of an MSTbased topology control algorithm," *Proc. IEEE INFOCOM 2003*, San Francisco, CA, USA.

[2] J.-H. Chang and L. Tassiulas,"Energy Conserving Routing in Wireless Adhoc Networks" *Proc. IEEE INFOCOM 2000*, pp.22-31, Tel Aviv, Israel.

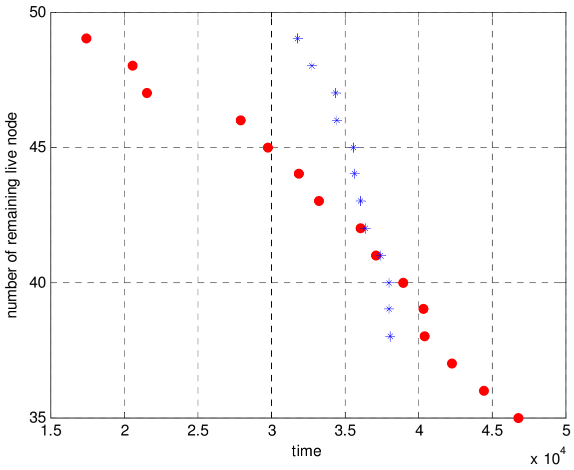

Fig. 1. Number of Remaining Live Node vs Time