Choose Wisely: Topology Control in Energy-Harvesting Wireless Sensor Networks

Xin Wang, Vijay S. Rao, R. Venkatesha Prasad and Ignas Niemegeers
Delft University of Technology, The Netherlands
Email: X.Wang-15@student.tudelft.nl, {V.Rao, R.R.VenkateshaPrasad, I.G.M.M.Niemegeers}@tudelft.nl

***Abstract*—Ambient energy-harvesting technology is a promising approach to keep wireless sensor networks (WSNs) operating perpetually. Depending on the harvesting source nodes can either be active (alive) or inactive (dead) at any instant in such Energy-Harvesting WSNs (EH-WSNs). Thus, even in a static deployment of EH-WSNs, the network topology is no longer static. A popular method to increase energy-efficiency in WSNs is by employing topology control algorithms. Most of the topology control algorithms in the literature cannot handle the situation when nodes have different energy-levels, and when number of active nodes varies with time in EH-WSN. To address this issue, we present two localized energy based topology control algorithms, *viz.*, EBTC-1 and EBTC-2. EBTC-1 is for convergecast applications of WSNs and EBTC-2 is for a generic scenario where all nodes are required to be connected. While typical topology control algorithms select a particular number of neighbors, the distinguishing feature of both these algorithms is that they select neighbors based on energy-levels, and render the global topology strongly-connected. Simulation results confirm that EBTC-1 and EBTC-2 reduce the transmission power and they let nodes have neighbors with high remaining energy. Results show that our proposed algorithms increase at least 33% in the remaining energy per neighbor. In addition, in terms of energy consumption and fault-tolerance, our proposed algorithms typically achieve 1-connected topology using 74% less energy compared to K-Neigh.**

Keywords*—Energy-harvesting, Topology control, Wireless sensor network

I. Introduction

In the age of Internet of Things (IoT) – which is envisioned to enable many *smart-** applications such as, smart-homes, smart-buildings, and smart-cities – wireless sensor networks (WSNs) have a key role to play. To enable these smart applications, number of sensors in the surroundings is increasing significantly. Typically these sensors are required to last long. However, the nodes being battery-powered, their lifetime is limited. It is also impractical to have battery operated nodes since installing and maintaining them are laborious tasks. Consequently, harvesting energy from ambient sources to power these nodes has attracted attention in recent times [1]. Even though the network using such nodes is considered to be perpetual, it is not guaranteed to be always connected.

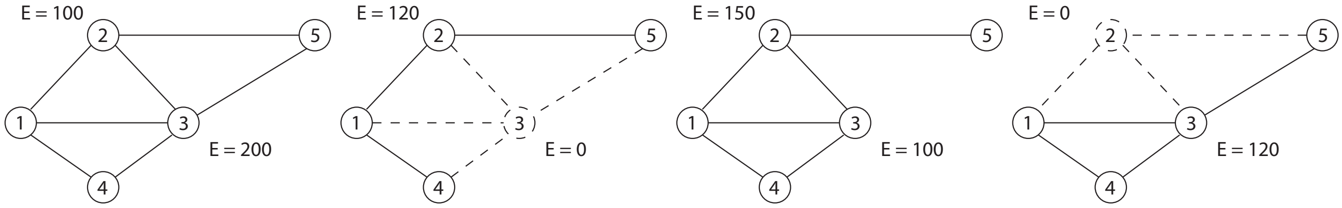
The possibility to harness the harvested energy brings new perspectives and challenges. These are due to variations in the amount of harvested energy across space and time. The requirement in energy-harvesting wireless sensor networks (EH-WSNs) is no longer maximizing lifetime but to exploit the available energy wisely and conservatively. For example, when the harvested power is surplus on a certain node, it can take more load from its "weaker" neighbors. At the same time, the weaker nodes should adopt extremely energy-conservative approaches until they harvest more energy. Though energy-harvesting opportunities for a node is unlimited (over infinite time horizon), available power at any instant is limited, thus necessitating energy-efficient operations.

The most energy consuming operation on a wireless sensor node is communication – current consumption by the radio is high and is further aggravated by idle-listening and re-transmission of packets for each neighboring node [2]. One popular method to increase energy-efficiency is by restricting the number of communication links, i.e., topology control. Topology control is a technique that conserves energy by reducing transmission power and improves the network capacity by reducing interference [2]. Topology control algorithms aim to achieve these by choosing the *right transmission power* and *neighbors* such that the desired network properties are achieved.

EH-WSNs bring new aspects for topology control: Residual energy-levels in nodes vary over time based on harvesting opportunities. Thus nodes are $Active$ (on) or $Inactive$ (off) making them to often leave or rejoin the network over time. The network is "heterogeneous" in terms of available energy of nodes, which implies nodes can assume different roles. Moreover, the network topology keeps changing even when all the nodes transmit at highest possible power because some nodes at an instant may be inactive. Fig. I illustrates this with an example. Furthermore, constructing new topology every time energy-levels change is expensive in terms of energy itself. Consequently, localized (distributed) topology maintenance algorithms are required to keep the EH-WSNs operating perpetually.

In this paper, we consider two distinct network scenarios for topology construction algorithms: (a) Convergecast, one-to-many or many-to-one scenario, a rooted network with sink as the root; and (b) A generic network, many-to-many scenario, where nodes can exchange data with one or more nodes in the network. We propose two localized Energy Based Topology Control (EBTC) algorithms – EBTC-1 and EBTC-2 corresponding to these two scenarios. In both algorithms, a strongly-connected topology is constructed and maintained by having each node adjust its transmission power level based on the locally collected information. Specifically, our contributions are as follows:

1) We propose localized topology control algorithms for two typical scenarios in EH-WSNs that maximize residual energy in every node, and nodes are assigned load based on their energy-levels. Each algorithm only relies on



(a) At time t_1, Node 3 has higher energy while Node 2 has lower energy. All nodes are connected.

(b) At time t_2, Node 3 exhausts energy. Thus the network topology is changed.

(c) At time t_3, Node 3 harvests energy and rejoins, but limited energy results in Node 2 spending more energy to keep the network connected.

(d) At time t_4, Node 2 leaves the network initiating reconstruction of the link between node 3 & 5.

Fig. 1. Challenges posed by the varying energy-levels in constructing topology in an EH-WSN.

its one-hop neighbor information to form a globally connected topology. While EBTC-1 guarantees strongly-connectedness, EBTC-2 is probabilistic and strongly-connectedness property can be tuned as required.

2) We also propose localized topology maintenance algorithm to handle the dynamic variation in remaining energy-levels at the nodes.
3) To the best of our knowledge, we are the first to propose implementable, low-complexity topology control algorithms for EH-WSNs. We implement our algorithms in Contiki-OS [3], which can be ported on sensor motes without any change and can easily be deployed.

The rest of the paper is organized as follows. We first summarize previous work on topology control in Sec. II. The system model, including network model and energy model is described in Sec. III. We present the proposed algorithms in Sec. IV. After that, we evaluate the performance of the proposed algorithms based on simulations and discuss related issues in Sec. V. Finally, we conclude the paper in Sec. VI.

II. Related Work

Work related to topology control in EH-WSN is limited. The important one is by Tan, *et al.*, who presented a distributed Energy-Harvesting-Aware (EHA) algorithm [4], which models the behavior of sensor nodes as an ordinal potential game where the high-power nodes cooperate with the low-power nodes to maintain the connectivity of the network. This algorithm has drawbacks in implementing it due to high communication overheads and requirement of accurate energy prediction models.

For traditional WSNs, extensive study has been done, and different algorithms were proposed based on various ideas. Many works consider building a *k*-connected topology, wherein *k* disjoint paths exist between every source-destination pair of vertices. This fault-tolerance is important in WSNs specially if the nodes' energy-levels are not taken into account. Building a minimum-cost *k*-connected sub-graph is an *NP-hard* problem [5]. Energy-aware and heterogeneous energy-level algorithms exist, however, their assumptions such as super-nodes having unlimited energy are not applicable to EH-WSNs. We summarize related works in Table I.

A more practical algorithm is K-Neigh [6] protocol. It is an asynchronous, and localized protocol that connects to *k* neighbors of each node based on distance. This guarantees global connectivity to a large extent, but fails in the worst case (around $4\,\%$ of the times). We base our algorithms on K-Neigh due to its simplicity.

Motivation: It is apparent that not many works are present that are applicable to the unique challenges of EH-WSNs. While fault-tolerance has been researched well in WSNs, seeking fault-tolerance is not worthwhile in EH-WSNs. As constructing *k*-connected topology is *NP-Hard*, approximation algorithms have been proposed but they have high communication overhead. In addition, Jorgic *et al.* [7] showed that it is impossible for nodes, based on local knowledge, to be accurate with respect to global connectivity properties. With energy being critical in EH-WSNs, we cannot afford to spend much energy on building *k*-connected topology. Moreover, since nodes may leave and rejoin the network at any time, reconstructing the *k*-connected topology becomes unrealistic.

Consequently, we look to build a strongly-connected topology with low transmission overhead that also handles and utilizes the dynamics of energy in the network. We maximize the residual energy in every node of the network. We aim to make higher energy nodes taking more workload than the lower energy nodes, which reduces the faults in the network. Before we propose our algorithms, we first describe the system in the next section.

III. System Model

We consider an EH-WSN network consisting of n nodes with omni-directional antennas. Nodes can adjust their transmission power levels in steps from the set $\{0, P_1, P_2, \ldots, P_{max}\}$ that depends on the radio hardware. For instance, the radio CC2420 [8] has 8 power levels ranging from -25 dBm to 0 dBm. Let the network topology be represented by an undirected graph $G = (V, E)$, where $V = \{v_1, v_2, ..., v_n\}$ is the set of nodes and E, is the set of links in the network. We consider the radio channels to be symmetric. Every node u is assigned a unique identifier denoted by $id(u)$. The network consists of only stationary nodes. We assume the network is connected initially.

The harvested energy from a source (solar, wind, thermal, vibration, etc.) is stored in devices such as supercapacitors. We assume that the state of charge can be measured. We define the energy state of nodes as $State(u) \in \{Active \mid Inactive\}$, where an $Active$ node has enough remaining energy to participate in the network activity, while an $Inactive$, node does not.

TABLE I. SUMMARY OF RELATED WORK

Work	Type of network	Basic idea	Considering node's energy	Fault tolerance
K-Neigh	WSN	Selects k-closest neighbors	N/A	N/A
CBTC(α)	Nodes with position information	Selects at least one neighbor in each direction	N/A	k-connectivity
FLSS$_k$	k-connected network	Builds the local sub-graph k-connected	N/A	k-connectivity
RESP	Nodes with different energy	Selects neighbors according to a weight function	Residual energy-levels of all nodes	k-connectivity
DPV	Heterogeneous WSN	Selects neighbors such that a node has at least k-vertex-disjoint paths to super-nodes	Nodes with limited energy and super-nodes with unlimited energy	k-connectivity
EHA	EH-WSN	Models the behaviors of nodes based on game theory	High harvesting power nodes cooperate with low harvesting power nodes	N/A

IV. PROPOSED ALGORITHMS

We consider two scenarios for topology construction in EH-WSN: (a) convergecast wherein there is a dedicated sink node from which data originates or to which all the nodes send their data. This is typical of WSN deployments. (b) a more generic scenario where the nodes exchange data. In this case, one of the nodes could be a sink as well. This scenario can be envisioned in the realm of IoT, machine-to-machine applications, etc. The algorithms designed must adhere to these constraints: (i) the algorithm should be localized; (ii) it should have low communication overhead; (iii) the neighbor selection metric should be a function of the node's energy and neighbor's remaining energy; and (iv) the resulting global topology should be strongly-connected.

A. EBTC-* Overview

The basic idea of EBTC-* (namely EBTC-1 and EBTC-2) is that topology control in EH-WSN is not just about selecting links with low costs, but also include selecting neighbors according to the energy-levels of the nodes. Considering the energy issue, we design algorithms based on greedy strategy to maximize the remaining energy of nodes and select neighbors with high residual energy. Consequently, since nodes have "high energy neighbors", their neighbors can receive and transmit more messages, resulting in a more sustainable network. This is the main guideline of the algorithms in this work.

Both variants consist of two phases: topology construction and topology maintenance. The key idea in the construction phase is that nodes select neighbors according to the distances to the neighbors and the remaining energy of its neighbors. In this case, the distance is no longer the only factor in selecting neighbors. Topology maintenance is required in EH-WSN, as a mechanism to update the topology whenever nodes leave or rejoin the network, taking care of the nodes' energy in the heterogeneous network and keeping all active nodes in the topology. There are two major differences between EBTC-1 and EBTC-2: 1) they use different strategies to discover neighbors; and 2) nodes select neighbors based on different criteria.

We discuss and elaborate the two points, explaining how EBTC-1 and EBTC-2 work. If a node has M neighbors, the complexity of both algorithms is $O(M)$ on each node.

B. Topology Construction in EBTC-1

EBTC-1 is designed to guarantee that the topology is strongly-connected with low communication overhead for the convergecast scenario. EBTC-1 has two steps: (i) Neighborhood information collection and (ii) neighbor selection. EBTC-1 is described in Algorithm 1.

The topology construction begins when the sink broadcasts a HELLO message. The message includes its energy-level, state and number of hops from the sink i.e., 0. Nodes that receive the message add the transmitter to its neighbor-list, notes down its energy, number of hops from the sink and the minimum required transmit power. The receiving nodes then broadcast their HELLO messages after medium contention with their energy-level, state and number of hops (incremented by 1).

After the neighbor information collection phase is complete, neighbor selection phase begins. Here an energy threshold E_T is defined, which decides how many neighbors a node should select. Starting from the closest neighbors, a node starts including its neighbors until the sum of neighbors' remaining energy meets the threshold. By using this greedy algorithm, nodes always connect to neighbors that need the lowest transmission powers to reach. This minimizes energy expenditure on the node. Further, by selecting nodes based on energy as the second criteria ensures one of these two: (a) if there are high energy neighbors close to the node, then lesser number of neighbors are selected; and (b) if only low energy neighbors are present, then more number of neighbors are selected. In either case, some kind of fault-tolerance is ensured. Note that one of the neighbors selected in EBTC-1 is mandatory to have a lower hop count to the sink than itself.

Algorithm 1: EBTC-1 on node u

Input: Node u; $InitializerID$ the predefined initializer node's ID
Output: $N'(u)$ computed neighbors of node u
$MessageLevel := 0$
if *NodeID* = *InitializerID* **then**
    Broadcast HELLO message with $MessageLevel$ information at maximum transmission power
else
    When receive HELLO message from node v
    $MessageLevel := v.MessageLevel + 1$
    Send HELLO message with $MessageLevel$
    $N(u) := N(u) \cup \{v\}$
end
Wait for all nodes to finish neighbor information collection procedure
Compute $N'(u)$ using neighbor selection Algorithm 2
Construct bi-directional links by adding missing links

C. Topology Construction in EBTC-2

EBTC-2 is for the generic case where there is no hierarchy. It is more challenging to construct a strongly-connected topology with just one-hop information. If incorrect set of neighbors are selected, then the resultant global topology

Algorithm 2: Neighbor selection of EBTC-1 on node u

Input: $N(u)$ the neighbor list of node u; E_T the predefined energy threshold
Output: $N'(u)$ computed neighbors of node u

Sort the nodes in $N(u)$ in ascending order of distance
$N_E := 0$; $N'(u) := \emptyset$
foreach v *in* $N(u)$ **do**
  if $State(v) = Active$ *and* $N_E < E_T$ *and* $v.MessageLevel < u.MessageLevel$ **then**
    $N'(u) := N'(u) \cup \{v\}$
    $N_E := N_E + Energy(v)$
  end
end
Adjust transmission power to the minimum value needed to reach the farthest node in $N'(u)$

will be disconnected. In EBTC-2, we can set the number of neighbors to be selected, indirectly through the threshold E_T. We shall discuss more about the influence of E_T in Sec. V.

The algorithm is presented in Algorithm 3. Similar to EBTC-1, nodes broadcast HELLO messages, collecting neighbor information. The only difference between this phase of EBTC-1 and EBTC-2 is that there is no need for any hop-count information. Once the neighbor information phase is completed, each link is assigned a weight as in Equation 1:

$$w(u,v) = \alpha \cdot \frac{E_v}{E_{max}} + (1-\alpha) \cdot (1 - \frac{RSSI_{u,v}}{RSSI_{min}}) \quad (1)$$

where, $w(u,v)$ is the weight function of the directed edge (u,v); E_v is the received remaining energy of node v, and E_{max} is the maximum energy capacity of a node. $RSSI_{u,v}$ denotes the RSSI from node v to node u, while $RSSI_{min}$ is the minimum RSSI to ensure connectivity. We also set α, a weight factor, that allows to control the importance level for remaining energy of the neighbor or for the required transmission power to the neighbor.

The next step is to sort neighbor list $N(u)$ of node u in ascending order of their weight and select the neighbors until the neighbors' energy is greater than or equal to E_T. Finally, nodes can add missing edges to construct the symmetric neighbor list, making the graph bi-directional.

Algorithm 3: EBTC-2 on node u

Input: Node u
Output: $N'(u)$ computed neighbors of node u

Broadcast HELLO message at maximum transmission power
Upon receiving message from node v: $N(u) := N(u) \cup \{v\}$
Wait for all nodes to finish neighbor information collection procedure
Compute $N'(u)$ using neighbor selection Algorithm 4
Construct bi-directional links by adding missing links

D. Topology Maintenance

We implement a simple event-triggered (based on energy) procedure to initiate topology maintenance. In EBTC-*, a node sends notification message when its remaining energy drops or increases above pre-defined thresholds. After receiving this notification, nodes re-select their neighbors according to the metric (for e.g, distance based in EBTC-1).

Algorithm 4: Neighbor selection of EBTC-2 on node u

Input: $N(u)$ the neighbor list of node u; E_T the predefined energy threshold
Output: $N'(u)$ computed neighbors of node u

Sort the nodes in $N(u)$ in ascending order of weight function
$N_E := 0$; $N'(u) := \emptyset$
foreach v *in* $N(u)$ *in this order* **do**
  if $State(v) = Active$ *and* $N_E < E_T$ **then**
    $N'(u) := N'(u) \cup \{v\}$
    $N_E := N_E + Energy(v)$
  end
end

We set two energy bounds, namely the $UpperBound$ and the $LowerBound$, where $E_{max} > UpperBound > LowerBound > 0$. A node switches its state only when they cross the thresholds. For example, suppose a node that was $Active$ spent most of its energy below $LowerBound$. In this case, it sends a notification message. Although it may harvest energy above the $LowerBound$ in due course, the node does not enter $Active$ state until the energy-level crosses the $UpperBound$.

Algorithm 5: Topology maintenance on node u

Input: $N(u)$ the neighbor list of node u; E_t the predefined energy threshold
Output: $N'(u)$ computed neighbors

Upon receiving broadcast message $(v, Energy(v))$ from node v:
if $Energy(v) > UpperBound$ *or* $N_E < E_T$ **then**
  Set $(v, Active)$ in $N(u)$
else if $Energy(v) < LowerBound$ **then**
  Set $(v, Inactive)$ in $N(u)$
end
Compute $N'(u)$ using neighbor selection Algorithm 2 or Algorithm 4

Broadcast messages are unacknowledged making them very susceptible to be lost due to collisions or due to lossy wireless channel. This affects our algorithms severely, resulting in disconnected topologies. To overcome these, we exploit the topology maintenance algorithm. When a node is $Active$ but does not have sufficient number of neighbors, i.e., sum of neighbors' is less than the energy threshold E_T, then the node sends a notification message.

V. Performance Evaluation

We consider an EH-WSN in which each node is powered through a solar panel and stores the harvested energy in a supercapacitor of size 15 mF. Since a typical low-power sensor node [8] can only operate between 3.3 V-2.7 V, all of the energy in the supercapacitor cannot be used. Therefore, the maximum usable energy $E_{max} = 3675mJ$. For our simulations, we model this variable as Bernoulli random process because Bernoulli process introduces highly varying energy-levels, thereby creating a dynamic network. We perform the simulations on Cooja simulator [9] in Contiki-OS 2.7. The

TABLE II. SIMULATION PARAMETERS

Parameter	Value
Deployment area	500m $\times$ 500m
Number of nodes	10, 20, 30, 40 and 50
Node distribution	Random
Radio	CC2420
EBTC energy threshold	$E_T = E_{max}$
EBTC-2 weight	$\alpha = 0.5$

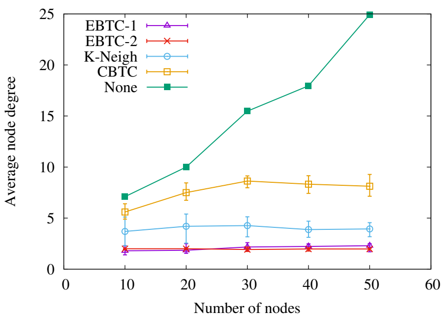


Fig. 2. Average node degree in the resulting topology of four algorithms. EBTC-* have low average node degree compared to other two algorithms

advantage of using Contiki is that the same code can be directly programmed onto a sensor node. We modify the simulator to perform EH-WSN simulations. Furthermore, we consider multipath radio model, collisions and other physical phenomena of wireless communications in our simulations as supported by Cooja. We use the ENERGEST module in Contiki to monitor energy usage. The other simulation parameters are listed in Table II.

We evaluate the performance of EBTC-1 and EBTC-2 against K-Neigh and CBTC with respect to several metrics. The reasons that we compare our proposed algorithms with the classic topology control algorithms are as follows: 1) the number of existing topology control algorithms designed for EH-WSNs is limited; 2) our proposed algorithms is the first work, as far as we know, which focus on selecting neighbors with high remaining energy. We show how the proposed algorithms achieve design goals of EH-WSNs, rather than just comparing against standard topology metrics such as hop and energy stretch factor.

We first show our topology control algorithms benefits the network by reducing interference. Fig. 2 shows that the average node degree in the graphs generated by EBTC-* are the lowest when compared among the original graph (shown by "None"), K-Neigh and CBTC. While K-Neigh chooses a constant number of neighbors, EBTC-* selects based on energy, which on an average tends to be a constant. Fig. 3 illustrates the average remaining energy per neighbor of the topologies derived under different algorithms. The average remaining energy per neighbor in EBTC-2 is always higher compared to other algorithms, while the value of EBTC-1 is higher than K-Neigh and CBTC in most cases. These results demonstrate the basic idea of EBTC: nodes select neighbors that have higher remaining energy.

Since the node selection in our algorithms is based on energy threshold E_T, the obvious question is how to guarantee connectivity. This is especially important in the case of EBTC-

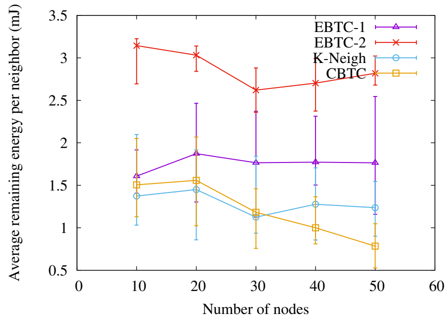
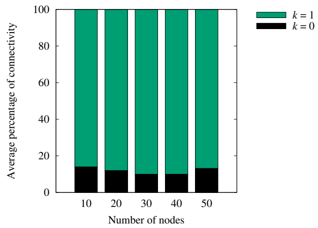
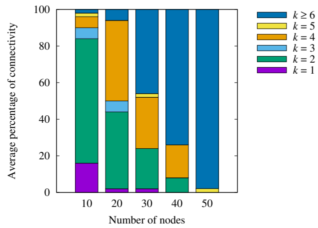


Fig. 3. EBTC-* selects neighbors with higher remaining energy, other algorithms select neighbors without considering the energy-levels.



(a) The k-connectivity in a network with $E_T = 0.5E_{max}$. The network has a chance to be disconnected.



(b) The k-connectivity in a network with $E_T = E_{max}$. The generated topology is a strong-connected.

Fig. 4. Network connectivity in EBTC-2 with different values of E_T. This concludes the topology generated by EBTC-2 as strongly-connected when $E_T = E_{max}$.

2, since in EBTC-1 a link to one of its parents is always added. A simple solution is to choose a high E_T value. However, that could create too many links. To choose the right E_T, we refer to [10], which shows that if a node connects to $\Theta(\log n)$ nearest neighbors, the graph will be globally connected. Based on this, we evaluate the topology construction for various E_T. Fig. 4(a) shows that if the threshold is $0.5E_{max}$, then around 15% of the times, the graphs were disconnected ($k = 0$) for 10 nodes. However, when the threshold is E_{max}, the graphs are always at least connected (see Fig. 4(b)). Furthermore, with that threshold, the generated topologies are also fault-tolerant, i.e., the k-connectivity property is exhibited. As density increases, this fault-tolerance also increases. In Fig. 5, we see that the average node degree is not very high to achieve high fault-tolerance in the network.

Fig. 6 shows the average adjusted transmission power in the topologies generated by aforementioned algorithms. The results show that EBTC-1 and EBTC-2 reduce the transmission power. Specifically, EBTC-1 reduces more transmission power compared to EBTC-2. This is because EBTC-1 gives more priority to distance, while EBTC-2 balances distance as well as neighbors' energy condition.

Influence of α in EBTC-2: EBTC-2 employs a metric to quantify and select neighbors, which is affected by the value of α. α can give priority either to the energy of the neighbor or to the distance, making the weight function generic. Here, we study the influence of α. Fig. 7 shows the average remaining energy per neighbor in terms of different values of α. We notice that giving more weight to the remaining energy of neighbors leads to higher remaining energy per neighbor.

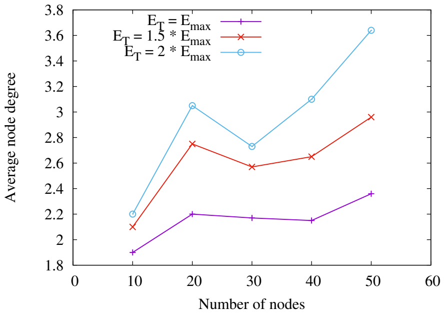


Fig. 5. Average node degree in the resultant topology of EBTC-2. The topology was strongly-connected.

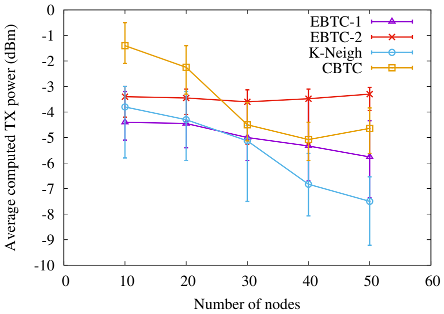


Fig. 6. EBTC-1 reduces more transmission power because it selects the closest neighbors; the transmission power is more stable in EBTC-2, which is a consequence of employing the weight function.

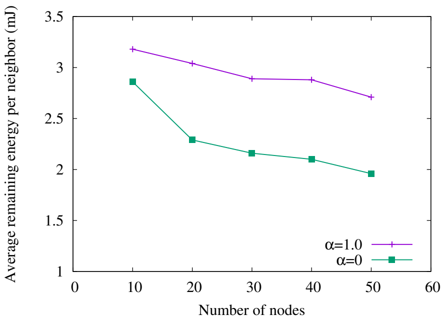


Fig. 7. The average remaining energy per neighbor is related to the weight function in EBTC-2. α is higher shows EBTC-2 gives more important to the neighbor's energy.

Plus, as shown in Fig. 8, low values of α result in high average node degree. Each node can choose its own α giving it the flexibility to either have more neighbors (be strongly-connected) or choose higher energy neighbors who can route packets for the nodes.

VI. Conclusion

Ambient energy-harvesting wireless sensor networks (EH-WSNs) are gaining importance in the growing market of Internet of Things, since EH-WSNs have thousand fold higher operational times, if not perpetual. However, the utility of the network can be constrained as harvested energy varies drastically over space and time. Therefore, algorithms are needed to construct and maintain the network with less overheads and high fault-tolerance. Given the energy dynamics, nodes energy

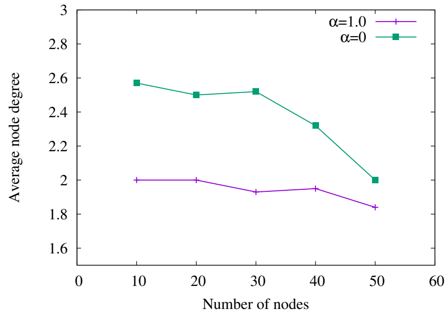


Fig. 8. As a consequence in changing α value in EBTC-2, the average node degree varies. When the value of α is higher, EBTC-2 focuses more on neighbor's energy. Consequently, it only needs to select fewer neighbors to meet the neighbor selection criteria.

must also be considered in creating topologies, without which the network may be disconnected.

In this paper, we proposed two localized topology control algorithms, namely EBTC-1 and EBTC-2, that satisfy the above mentioned design goals. The proposed algorithms consist of topology construction and maintenance phases, iterating the topology to the changes of energy in EH-WSN. EBTC-1 is for convergecast applications of WSNs and EBTC-2 is for more generic applications. Contrary to selecting a number of neighbors to form a connected network, we select based on remaining energy of the neighbors. Results show that compared to classic algorithms that do not take neighbor's remaining energy into consideration, our proposed algorithms increase at least 33% in the remaining energy per neighbor. In addition, in terms of energy consumption and fault-tolerance, our proposed algorithms typically achieve 1-connected topology using 74% less energy compared to K-Neigh.

References

[1] R. Prasad, S. Devasenapathy, V. Rao, and J. Vazifehdan, "Reincarnation in the ambiance: Devices and networks with energy harvesting," *Commn. Surveys Tutorials, IEEE*, vol. 16, no. 1, pp. 195–213, 2014.

[2] P. Santi, "Topology control in wireless ad hoc and sensor networks," *ACM computing surveys (CSUR)*, vol. 37, no. 2, pp. 164–194, 2005.

[3] A. Dunkels, B. Grönvall, and T. Voigt, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *IEEE LCN*, 2004, pp. 455–462.

[4] Q. Tan, Y. Liu, Y. Han, W. An, S. Ci, and H. Tang, "Energy harvesting aware topology control with power adaptation in wireless sensor networks," in *IEEE WCNC*, April 2014, pp. 2722–2727.

[5] S. Khuller, "Approximation algorithms for finding highly connected subgraphs," *Vertex*, vol. 2, p. 2, 1997.

[6] D. Blough, M. Leoncini, G. Resta, and P. Santi, "The k-neighbors approach to interference bounded and symmetric topology control in ad hoc networks," *Mobile Computing, IEEE Transactions on*, vol. 5, no. 9, pp. 1267–1282, Sept 2006.

[7] M. Jorgic, N. Goel, K. Kalaichevan, A. Nayak, and I. Stojmenovic, "Localized detection of k-connectivity in wireless ad hoc, actuator and sensor networks," in *ICCCN 2007*. IEEE, 2007, pp. 33–38.

[8] "Texas Instruments CC2420." [Online]. Available: http://www.ti.com/product/cc2420

[9] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with cooja," in *IEEE LCN*, 2006, pp. 641–648.

[10] F. Xue and P. R. Kumar, "The number of neighbors needed for connectivity of wireless networks," *Wireless networks*, vol. 10, no. 2, pp. 169–181, 2004.