

Delft University of Technology
Master's Thesis in Embedded Systems

Topology Control in Energy-harvesting Wireless Sensor Networks

Xin Wang



Topology Control in Energy-harvesting Wireless Sensor Networks

Master's Thesis in Embedded Systems

Embedded Software Section
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands

Xin Wang
X.Wang-15@student.tudelft.nl

24th August 2015

Author

Xin Wang (X.Wang-15@student.tudelft.nl)

Title

Topology Control in Energy-harvesting Wireless Sensor Networks

MSc presentation

28th August 2015

Graduation Committee

Prof. dr. K. G. Langendoen Delft University of Technology

Dr. R. R. Venkatesha Prasad Delft University of Technology

Dr. Al-Ars Delft University of Technology

Ir. V. Rao Delft University of Technology

Abstract

Ambient energy-harvesting technology is a promising approach to keep wireless sensor networks (WSNs) operating perennially. 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 focus only on the transmission power while constructing a static topology without taking into account the residual energy of the nodes. Consequently, they cannot handle the situation when nodes have different energy levels, and when the number of active nodes varies with time in EH-WSN. Since the number of nodes alive in EH-WSNs is varying there is no possibility of having a centralized solution. 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 strictly connected. In some cases, to ensure fault tolerance the network may be required to be k -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.

Preface

Energy-harvesting is a promising technology that can be widely applied in smart applications. It could further reduce the energy dependency and use of batteries. Wireless sensor network needs this technology to create a perennial operation of nodes to support various smart applications. Diving deep into this topic, I found interesting research challenges that need to be solved rather than what can be solved. Among the challenges, topology control is a fundamental one, which manages the wireless network infrastructure. This motivated me to find solutions to topology control in the energy-harvesting wireless sensor network. I hope this work provides new ideas for designing topology control algorithms in the energy-harvested sensor networks.

I would like to thank my daily supervisor Vijay Rao for his patient guidance and thoughtful support through this process. He is not only a knowledgeable advisor but also a kind person who really knows what problems I have encountered and helped me a lot. I learned valuable academic approach and attitude from him. I also want to thank Dr. Venkatesha Prasad (VP) who always gave me *big pictures* and insightful advice when I had problems. I also want to acknowledge Prof. Koen Langendoen for hosting me at the Embedded Software group, and Dr. Zaid Al - Ars for being a member of my committee. Furthermore, I want to thank the researchers of the Embedded Software department for their constructive discussions. I will carry the knowledge and attitude learned from you with me for the rest of my career.

Xin Wang

Delft, The Netherlands
24th August 2015

Contents

Preface	v
1 Introduction	1
1.1 Motivation	2
1.2 Topology control	2
1.2.1 Energy conservation	3
1.2.2 Network capacity	4
1.3 Challenges	4
1.4 Problem statement	5
1.5 Contribution	6
1.6 Summary	6
2 Literature study	7
2.1 Works in energy-harvesting wireless sensor networks	7
2.2 Works in wireless sensor networks	8
2.2.1 Fault-tolerant algorithms	8
2.2.2 Neighbor-based algorithms	9
2.2.3 Residual energy-aware algorithms	11
2.3 Summary	11
3 Models and Problem Description	13
3.1 Network model	13
3.2 Communication model	14
3.3 Energy model	14
3.3.1 Realistic model	15
3.3.2 Stochastic model	16
3.4 Problem description	16
3.5 Assumptions	17
3.6 Summary	18
4 Proposed Algorithms	19
4.1 Design guidelines	19
4.2 Fault-tolerance in EH-WSN	19
4.3 EBTC-* overview	20

4.4	Topology construction in EBTC-1	21
4.5	Topology construction in EBTC-2	23
4.5.1	The influence of α	24
4.5.2	Connectivity	25
4.6	Topology maintenance	25
4.6.1	Maintenance strategy	25
4.6.2	Energy bounds	26
4.7	Summary	27
5	Performance Evaluation	29
5.1	Experimental setup	29
5.2	Evaluation metrics	29
5.2.1	Network metrics	30
5.2.2	Energy-harvesting related metrics	31
5.3	Results	32
5.4	Discussion	38
5.4.1	Influence of α in EBTC-2	38
5.4.2	Energy threshold E_T	40
5.5	Testbed	40
5.6	Summary	43
6	Conclusions and Future Work	45
6.1	Conclusions	45
6.2	Further work	46

Chapter 1

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. As a promising technology, WSNs have been widely used in various scenarios, including human tracking and animal monitoring. Applications have been developed for extreme purposes, e.g., event detection of enemy soldiers [1] and ecosystem surveillance [2]. In these applications, sensors are deployed so that they can communicate with each other using wireless channel, collecting the required information.

Typically these sensors are required to last long. However, since the nodes are battery-powered, their lifetime is limited. Many strategies and algorithms have been proposed to enhance energy-efficiency that are expected to extend the lifetimes. However, it is impractical to have battery operated nodes since replenishing them is a laborious task. If nodes are deployed in inaccessible locations, the network has a limited lifetime. Consequently, harvesting energy from ambient sources to power these nodes has attracted attention in recent time; since harvesting can, theoretically, power the nodes perpetually [3].

Energy-harvesting is a technique that harvests or scavenges a variety of untapped ambient energy sources and converts the harvested energy into electrical energy to recharge the batteries [4]. Energy-harvesting technology enables the network to extract energy from surrounding environment, such as solar power [5], mechanical movement [6], heat [7] and fluid flow [8]. Figure 1.1 shows a typical node in an energy-harvesting wireless sensor network (EH-WSN). Energy-harvesting technology provides numerous benefits [4]; we list some of them:

1. Reduce the dependency on battery power: with the harvested energy, nodes eliminate the use of battery power—the harvested ambient energy may be sufficient to eliminate the need for batteries completely.
2. Reduce installation and maintenance cost: self-powered nodes elimin-

ate service visits to replace batteries.

3. Provide long-term solutions: reliable nodes with energy-harvesting devices will function as long as the ambient energy is available, which is perfectly suited for long-term applications.

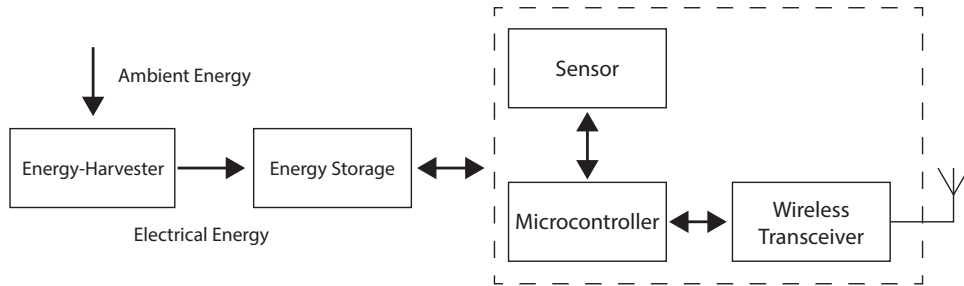


Figure 1.1: A typical node in EH-WSN (picture adapted from [4]).

Applications in EH-WSN have been designed and implemented to overcome the weaknesses in traditional WSN. To give an example, in some environments, sensors are deployed in hard-to-reach areas, which makes it difficult to replace batteries. For instance, Park et al. [9] presented a low-power embedded structural health monitoring sensing system with energy-harvesting technology, which converts mechanical vibration to electrical energy to power the sensors. Buchli et al. [10] demonstrated a novel dynamic power management scheme that enables operations of solar energy-harvesting systems over time periods on the order of multiple years. They showed that the proposed scheme benefits a energy-harvesting wireless sensor system deployed in a remote, high-alpine environment. In this case, since the environment is extreme, this application is hard and expensive to achieve without the energy-harvesting technology.

1.1 Motivation

In an EH-WSN, the network lifetime is no longer restricted by the limited energy supply. Even though the nodes in the network are considered to be perpetual, the network is not guaranteed to be always connected. In order to maintain a connected network while achieving desired network properties, such as fault-tolerance and low transmission power, topology control technique is essential in battery-powered WSNs and EH-WSNs.

1.2 Topology control

The most energy consuming operation on a wireless sensor node is wireless communication—current consumption by the radio is high and is further ag-

gravated by idle-listening and retransmission of packets for each neighboring node [11]. 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 [11]. Topology control algorithms aim to conserve energy and improve the network capacity by choosing the *right* transmission power and neighbors such that the network is connected and has desired properties. The trade-off between energy conservation and the network connectivity can be simplified in Figure 1.2. We will elaborate the issues as follows.

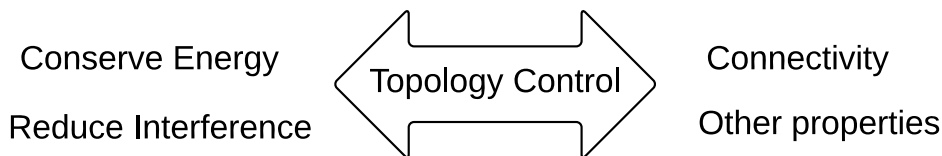


Figure 1.2: The trade-off between node energy conservation and network connectivity in topology control. Conserving too much energy may harm the network connectivity; enhancing the network connectivity may also increase the network interference.

1.2.1 Energy conservation

The efficient use of available energy in a network is always one of the fundamental metrics, especially in a WSN, where nodes have limited power, it is critical for nodes to save energy. Thus, in order to reduce energy consumption, a practical and efficient approach is to reduce the transmission power.

Considering the wireless channel and energy consumption models in WSN, one observation is that instead of using a long, energy-inefficient edge, nodes should choose a multihop path composed of short edges that connects the two endpoints of long edge to communicate [11]. This observation is a fundamental idea in topology control to reduce energy consumption. Therefore, topology control algorithms in battery-powered WSN aim to choose short links between nodes while preserving the network connectivity.

However, in an EH-WSN, with the renewable energy, energy consumption is not always the critical aspect when choosing links. Nodes need to manage energy smartly. Therefore nodes should choose neighbors *wisely* in EH-WSN—node should choose neighbors based on not only the energy consumption but also the energy levels of their neighbors.

1.2.2 Network capacity

In a wireless network, the communications share the same radio channel. This implies the undesired collision and interference during communication are unavoidable, and interference damages network traffic capacity. The problem of reducing interference in the network can be elaborated as follows.

Since transmission power decides an area where the interference may occur, an ideal situation is that the overlapping area of transmission ranges is minimized. A practical approach to decrease interference is to set their transmission power to the desired value, such that the transmission ranges are therefore limited. So it comes back to the statement we have mentioned in Section 1.2.1. However, if the transmission powers are reduced too much, the network has a chance to be disconnected. So when designing a topology control algorithm, it is necessary to keep a balance between the connectivity and network performance.

1.3 Challenges

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. For instance, a node with higher remaining energy can be assigned a role that carries more communication tasks. Moreover, the network topology keeps changing even when all the nodes transmit at the highest possible power because some nodes may be inactive at a time instant. There is a vast body of literature on topology control for battery-powered WSNs that typically do not consider “heterogeneous” levels of energy among nodes [12, 13]. Most of the existing works do not consider nodes to have different roles based on their energy, since energy decreases monotonically over time in battery-powered nodes. Other works that do consider heterogeneous levels of energy take a static set of nodes to have higher energy levels. Furthermore, constructing new topology every time when energy levels change is expensive in terms of energy itself. Consequently, localized topology maintenance algorithms are required to keep the EH-WSNs operating perpetually.

To illustrate these challenges, we take Figure 1.3 as an example. In this EH-WSN, Node 1 needs to send messages to Node 5, which can be forwarded by one of its neighbors: Node 2, 3 or 4. Considering the remaining energy of Node 2 and 3 in Figure 1.3(a) at time t_1 , Node 2 is closer to Node 1, which makes Node 1 use less transmission energy to reach Node 2. Node 3 has higher remaining energy, which implies Node 3 can take more workload but needs higher transmission energy to reach. The first question for a node is that how to select neighbors such that the network is connected and

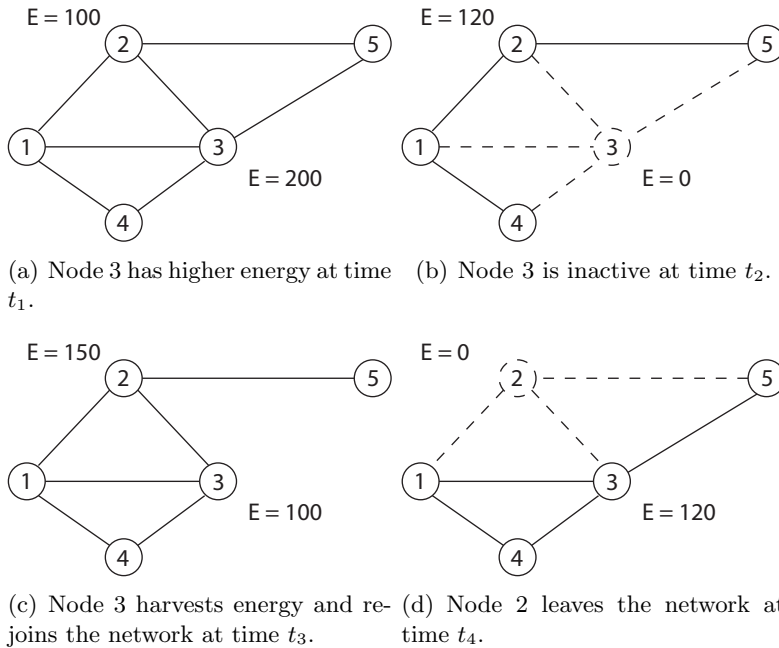


Figure 1.3: The remaining energy of nodes changes over time in EH-WSN

nodes can reduce transmission power while preserving neighbors with high remaining energy.

Furthermore, after some time, as shown in Figure 1.3(b), at time t_2 , Node 3 exhausts energy while Node 2 harvests more energy. Now the network should be aware of this critical energy change at Node 3. Then, as depicted in Figure 1.3(c), Node 3 harvests energy and rejoins the network, but still has limited energy, which indicates nodes should have different roles according to their energy conditions. For instance, Node 3 may now reduce transmission power, dropping the links to farther neighbor like Node 5, to reduce energy consumption. However, as illustrated in Figure 1.3(d), Node 2 leaves the network now. In order to keep the network connected, Node 3 must reconstruct the missing link to Node 5. The second question for a node: in contrary to the static network topology, how to maintain the local topology given the dynamics of energy.

1.4 Problem statement

Topology control technique has been widely used in WSN, and extensively study has been proposed to construct topology with desired properties in WSN. However, as we stated above, new challenges exist in EH-WSN. The problem that we need to address is as follows:

Given an EH-WSN, how to construct and maintain the network topology such that nodes in this topology select neighbors based on the energy levels and achieve desired network properties?

This problem is complicated because the renewable energy in EH-WSN makes it difficult to maintain a static network topology. Therefore topology control algorithms are responsible for updating the topology, which implies we need to design an efficient topology maintenance mechanism. Additionally, building a connected graph based on local information is non-trivial—there does not exist a local graph property which perfectly captures graph connectivity [14]. In addition, due to the dynamic energy distribution among nodes from time to time, we need to find a way to utilize this characteristic.

1.5 Contribution

In this work, we will study the topology control issue in EH-WSN, identifying the challenges in EH-WSN, and proposing solutions to these problems. The main contribution is 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 its one-hop neighbor information to form a globally connected topology. While EBTC-1 guarantees well-connectedness, EBTC-2 is probabilistic and well-connectedness 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. We evaluate the proposed algorithms based on simulations and on a real-world deployment. The results show that the proposed algorithms perform better with regard to network connectivity, fault-tolerance, transmission energy consumption and neighbors' remaining energy.

1.6 Summary

Energy-harvesting technology brings new opportunities for applications that are required to last long. EH-WSN is therefore employed because of its merits. Topology control in EH-WSN is a new topic that needs more study. We presented the challenges and the problem in this chapter. In the next chapter, we will present existing topology control algorithms in EH-WSNs and battery-powered WSNs.

Chapter 2

Literature study

In this chapter, some of the topology control algorithms in EH-WSNs as well as in battery-powered WSNs are presented. We summarize the ideas of these algorithms.

2.1 Works in energy-harvesting wireless sensor networks

Works related to topology control in EH-WSN are limited. The important one is by Tan et al., who presented a game theory-based distributed Energy-Harvesting-Aware (EHA) algorithm [15], which models the behaviors of sensor nodes as a game. This work analyzes the energy consumption rate and energy-harvesting rate of each node at different times. In this game, the high harvesting power nodes cooperate with the low harvesting power nodes to maintain the connectivity of the network. The algorithm first constructs a preliminary topology based on the Directed Local Spanning Sub-graph (DLSS) algorithm [16]. Then each node u tries to find a neighbor that covers the farthest neighbor of node u by adjusting the transmission power step by step. The idea is that a node may make a sacrifice by increasing its transmission power if it can help reduce energy consumption at another node with lower residual energy.

This game theory-based algorithm has drawbacks with respect to implementing them. Since nodes need to send messages to neighbors to negotiate, and the neighbors send responses. This round-trip procedure costs at least two messages exchanged between each pair of nodes, causing high communication overheads. Also, it requires accurate energy-harvesting and energy consumption profiles to predict how nodes behave, which is related to the deployment and environment, and may not always be available. In addition, since it requires the DLSS algorithm to build a topology in the initialization phase, the computational complexity of the EHA algorithm is $O(n^2)$, where n is the number of nodes in the network.

2.2 Works in wireless sensor networks

For battery-powered WSNs, extensive study has been done, and various algorithms were proposed based on different ideas.

2.2.1 Fault-tolerant algorithms

Many works consider building a k -connected topology: a topology is k -vertex connected if the removal of any $k-1$ nodes (and all the related links) does not partition the network. This fault-tolerance is important in WSN, because the network topology is susceptible to battery depletion. However, building a minimum-cost (the cost could be hops, distance, etc.) k -connected subgraph is an *NP-Hard* problem [17]. We elaborate several algorithms that provide fault-tolerance in a WSN.

CBTC(α) is a distributed topology control algorithm [18] that provides fault-tolerance in a network. It is extended from CBTC [19] (Cone-based Topology Control) algorithm. The basic idea is that each node in the network adjusts the transmission power to the minimum value such that it can reach at least one node in “every direction”. Figure 2.1 shows the cones in the CBTC algorithm. In this figure, the cone is set to width $\alpha = \frac{\pi}{2}$, and node u attempts to reach at least one node in every cone. First, node u sets the transmission power to the minimum value such that it can reach three nodes, which are shown inside the dashed circle. To meet the requirement of CBTC, node u must increase its transmission power to reach node v . After this adjustment, now node u keeps one neighbor in every cone.

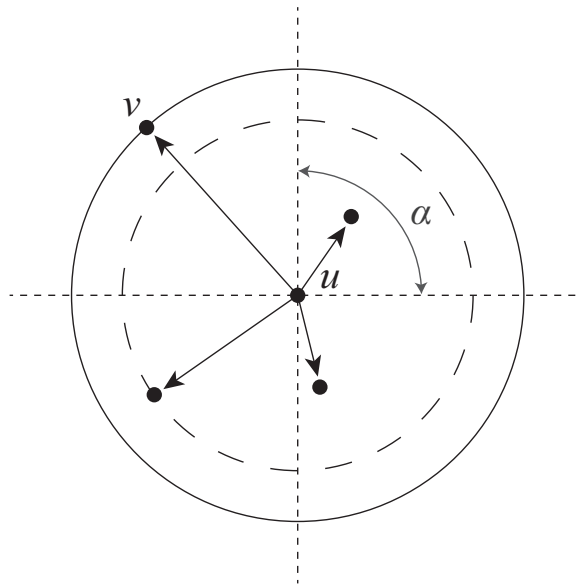


Figure 2.1: Topology control with CBTC algorithm on node u .

In the CBTC(α) algorithm, the topology connectivity depends on the width of the cone, that is, the value of α . Study [19] shows that setting $\alpha \leq \frac{2}{3}\pi$ can preserve the generated topology connected. The CBTC algorithm has two major merits: it is fully distributed, and it preserves the network connectivity. However, the weaknesses are obvious: it requires nodes to provide directional information, and nodes exchange high number of messages to construct the topology.

As aforementioned, the CBTC(α) is a variant of the CBTC algorithm, providing fault-tolerance in a network. Furthermore, Bahramgiri et al. showed that if $\alpha \leq \frac{2\pi}{3k}$, the generated topology is (worst-case) k -connected. However, like the CBTC algorithm, this variant shares the same weak points.

Fault-tolerant Local Spanning Sub-graph (FLSS $_k$) [20] is a distributed algorithm that preserves k -connectivity in a network. It is based on a centralized version—Fault-tolerant Global Spanning Sub-graph (FGSS $_k$).

FLSS $_k$ consists of three phases:

1. Information Collection: each node u collects local information of neighbors by exchanging HELLO messages. Node u then has the set of neighborhood information N_u .
2. Topology Construction: each node u defines, based on the information in N_u , the proper list of neighbors for the final topology.
3. Constructing Bidirectional Links (Optional): each node u adds or removes the links to make sure that all edges are bidirectional.

Since nodes only have local information, in the topology construction phase, each node u builds its local spanning sub-graph S_u over N_u . The sub-graph is constructed by adding edge to the closest neighbor, then checking if the sub-graph is k -connected. If not, then add another edge, repeating this process until the sub-graph is k -connected. Li and Hou proved that FLSS $_k$ preserves k -connectivity, and it minimizes the maximum transmitting range of nodes in the network over the set of all localized algorithms. The drawback of FLSS $_k$ is that whether the sub-graph is k -connected is needed to be tested by using network flow techniques that results in high complexity. Therefore, the algorithm complexity and communication overheads are increased.

2.2.2 Neighbor-based algorithms

Since nodes have the ability to determine the surrounding neighbors that are within the maximum transmitting range, it is possible for each node to select neighbors only based on local information to keep the whole network connected. This is the basic idea of neighbor-based algorithms, and several algorithms are proposed such as the XTC protocol [21] and the K-Neigh [22]. We take the K-Neigh algorithm as an instance.

K-Neigh is a localized and asynchronous protocol that selects k neighbors of each node based on distance. By using distance estimation techniques like Radio Signal Strength Indicator (RSSI), it is possible for a node to estimate the distance to another node. This algorithm works as follows:

1. Every node broadcasts its ID at maximum transmit power;
2. Upon receiving the broadcast messages from other nodes, every node stores the neighbor information;
3. Every node computes its k -closest neighbors, and broadcasts this information at maximum transmit power;
4. By exchanging neighbor information, nodes are able to have lists of symmetric neighbors.

At the end of the protocol execution, nodes set the transmit power to the minimum value that needed to reach the farthest node in the neighbor lists.

Apparently, the connectivity property in the K-Neigh algorithm is not guaranteed, which is denoted by “ k -neighbors connectivity problem”. This problem is described as follows: given a set of N nodes, which is the minimum value of k such that the k -neighbors graph G_k is strongly-connected?

This problem can be easily solved if any of the following conditions is met:

1. Nodes have global knowledge of the network. In this case, the problem is transformed to check if a graph is connected.
2. With only local knowledge, nodes start connecting to the closest neighbor, checking for connectivity and waiting for feedback from other nodes. If it is not connected, nodes increase the number of neighbors, repeating the process.
3. Set $k = n - 1$, where n is the number of nodes in the network.

Reviewing these three conditions, all of them are difficult to meet in realistic scenarios. This is because in a distributed network, global knowledge is hard to achieve, especially considering the energy consumption. Communicating among nodes to acquire global view is expensive with respect to energy consumption. In addition, set $k = n - 1$ is impractical: 1) Nodes must use high transmission power to reach the farthest node; 2) since high transmission power results in high interference in the network, the network traffic capacity therefore is compromised.

Since finding a deterministic solution to the “ k -neighbors connectivity problem” is difficult, researchers have studied this problem in a probabilistic approach: find the critical neighbor number (CNN) in a network such that the network is connected with high probability. However, determining the

CNN is a very difficult problem. Xue and Kumar [23] show that if a node connects to at least $\Theta(\log n)$ nearest neighbors, the graph will be connected.

Back to the K-Neigh algorithm, in order to keep the topology connected, the value of k needs to be investigated. Authors evaluated the minimum value of k that guarantees that the generated topology is connected with probability at least 95%, by performing extensive simulations. The results show that setting $k = 9$ provides connectivity with high probability for a wide range of network size (from 50 to 500 nodes).

2.2.3 Residual energy-aware algorithms

Apart from the works stated above, algorithms that considering the energy conditions of nodes have been proposed. We discuss two representative algorithms.

Residual Energy-aware Shortest Path (RESP) [24] is an energy adaptive topology control algorithm, which not only balances the energy consumption of different nodes but also provides fault-tolerance.

In this work, a weight function is employed, involving not only the required transmission power but also the residual energy of nodes. With this weight function, the proposed algorithm preserves the minimum-weight path. The construction of k -connected sub-graph part in this algorithm is similar to the FLSS $_k$ algorithm, but instead of choosing the closest neighbors, it chooses neighbors in the order of sorted results of the weight function, which also causes high communication overhead.

Disjoint Path Vector algorithm (DPV) is proposed by Bagci et al. [25], which is for heterogeneous WSNs consisting of supernodes (with infinite energy) and ordinary nodes. In the settings, route data is collected by sensor nodes to supernodes, proposing the DPV algorithm for topology construction. The authors argue this algorithm works better in the two-tiered heterogeneous network. In this work, supernodes are supposed to have unlimited energy resources, which makes the supernodes and the normal nodes have different tasks in constructing the topology. However, in a typical EH-WSN, where nodes have the same energy capacities, it is difficult to satisfy the requirements of supernodes.

2.3 Summary

In this chapter, we presented several topology control algorithms in EH-WSN as well as in WSN environments. The fundamental ideas are introduced and discussed. We have summarized the related work with their features in Table 2.1. We showed that no work fits our goal: design an implementable topology control algorithm that preserves network connectivity, provides fault-tolerance and selects neighbors with high remaining energy in

EH-WSN. In the next chapters, we will state our system models and propose our solutions.

Table 2.1: 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 supernodes	Nodes with limited energy and supernodes 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

Chapter 3

Models and Problem Description

As we showed the challenges in topology control in EH-WSNs, introduced the related work in previous chapters, we now focus on the new approaches. To gain a better understanding of the system and the network, we describe the network model, communication model, and the energy model along with definitions in an EH-WSN in this chapter. Then we formulate the problem and present assumptions that are used in this work.

3.1 Network model

We consider an EH-WSN network consisting of n nodes with omnidirectional antennas. Nodes can adjust their transmission power levels in steps from the set $\{0, P_1, P_2, \dots, P_{max}\}$, which depends on the radio hardware. For instance, the radio CC2420 [26] has 8 power levels, from a minimum of -25 dBm to a maximum of 0 dBm. Let the network topology be represented by an undirected graph $G = (V(G), E(G))$, where $V(G) = \{v_1, v_2, \dots, v_n\}$ is the set of nodes and $E(G)$, is the set of links in the network. In addition, for every node u in the network, it is assigned a unique identifier, denoted as $id(u)$. The network consists of only stationary nodes.

Since we also need to care about the state of nodes, we define the state of node as follows:

Definition 1. Node's Energy State *If the remaining energy of a node is less than a predefined value E_{min} , we consider its energy state is Inactive; otherwise its energy state is Active. This is denoted by: $state(u) \in \{Active | Inactive\}$.*

Other definitions that we will use in this work are as follows:

Definition 2. Topology *The topology generated by an algorithm is a sub-graph $G' = (V'(G'), E'(G'))$ of the original graph $G = (V(G), E(G))$.*

Definition 3. Neighbor Relation Node u is a neighbor of node v under a topology is denoted by $u \rightarrow v$ if there is an edge (u, v) in the topology.

Definition 4. Neighborhood The neighborhood of node u , denoted by $N(u)$, is the set of nodes that are the neighbors of node u in the topology.

Definition 5. Degree The degree of node u is denoted as $Deg(u)$. This is defined as the neighbors of node u in a topology. Obviously, $Deg(u) = |N(u)|$.

Definition 6. Physical Degree The physical degree of node u is defined as the number of neighbors within the transmission range of node u .

Definition 7. Connectivity For a topology, node u is said to be connected to node v (denoted by $u \Rightarrow v$) if there exists a path $(p_0 = u, p_1, \dots, p_{m-1}, p_m = v)$ such that $p_i \rightarrow p_{i+1}$, $i = 0, 1, \dots, m-1$, where $p_k \in V(G)$, $k = 0, 1, \dots, m$. It follows that $u \Rightarrow v$ if $u \Rightarrow p$ and $p \Rightarrow v$ for some $p \in V(G)$.

Definition 8. k -vertex Connectivity A graph G is k -vertex connected if for any two vertices $v_1, v_2 \in V(G)$, there are k pairwise-vertex-disjoint paths from v_1 to v_2 . Or equivalently, a graph is k -vertex connected if the removal of any $k - 1$ nodes (and all related links) does not partition the network.

Definition 9. Well-connected Graph A graph G is well-connected if for any two vertices $v_1, v_2 \in V(G)$, there is at least one disjoint path from v_1 to v_2 .

Definition 10. Addition and Removal The Addition operation is to add an extra edge (v, u) into G if $(u, v) \in E(G)$ and $(v, u) \notin E(G)$. The Removal operation is to delete any edge $(u, v) \in E(G)$ if $(v, u) \notin E(G)$.

3.2 Communication model

Nodes in the network comply with the IEEE 802.15.4 standard [27], which specifies the physical layer and media access control. IEEE 802.15.4 is proposed for low-rate wireless personal area networks (LR-WPANs), focusing on low cost of deployment, low complexity, and low energy consumption. This standard is designed for wireless sensor applications. We consider only 2.4GHz band.

3.3 Energy model

In EH-WSNs, each node is equipped with an energy storage, which can be a rechargeable battery or a super-capacitor to store energy. We consider the energy-neutral model [28] as follows: suppose the harvested power from the

energy source is $P_s(t)$ at time t , and the energy being consumed at time t is $P_c(t)$. The following equation should be satisfied:

$$\int_0^T P_c(t)dt \leq \int_0^T P_s(t)dt + B_0 \quad \forall T \in [0, \infty) \quad (3.1)$$

where B_0 is the initial energy stored in the idea energy buffer.

3.3.1 Realistic model

We can employ the real data to model the energy profile. Consider a solar energy-harvesting network, we could use the history solar energy resource data to model the behavior of the network. For instance, as shown in Figure 3.1, the solar radiation data is taken from the National Renewable Energy Laboratory ¹, showed how solar radiation changes against time.

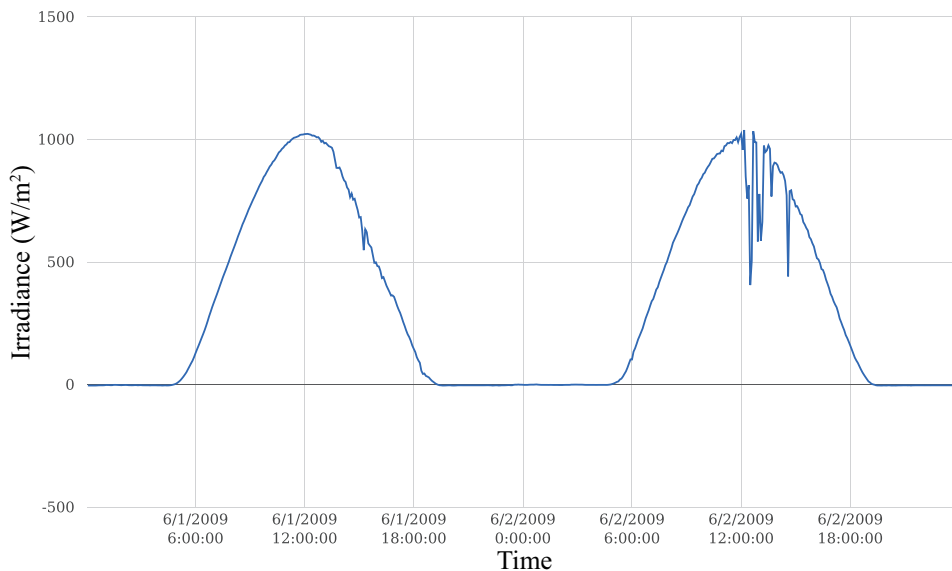


Figure 3.1: Solar radiation data in two days

This approach offers nodes the ability to predict how energy harvests. Though using the history data cannot provide the accurate energy-harvesting profile, it is still useful when an EH-WSN needs to know trends in a long-term period. However, the realistic model highly depends on the environment settings. If the environment is changed or the network needs to be immigrated to other places, this model does not suit the new deployment. To make a more general energy model, we describe the stochastic model as follows.

¹<http://www.nrel.gov/rredc/>

3.3.2 Stochastic model

Apart from the real energy data, we can also model the energy profile using stochastic models. We model an energy-harvesting profile along with an energy consumption profile. Let $\mathcal{E}_h(u, t)$ denote the energy-harvesting rate of node u at time t ; $\mathcal{E}_c(u, t)$ denote the energy consumption rate of node u at time t . We denote the remaining energy of node u at time t as $E_t(u)$. Then assume the remaining energy of node u at time t_0 is $E_{t_0}(u)$. After operating for a time period T , the remaining energy at time $t_0 + T$ is calculated as follows:

$$E_{t_0+T}(u) = E_{t_0}(u) + \int_{t_0}^{t_0+T} \mathcal{E}_h(u, t)dt - \int_{t_0}^{t_0+T} \mathcal{E}_c(u, t)dt \quad (3.2)$$

where, $0 \leq E_t(u) \leq E_{max}$, and E_{max} is the maximum energy nodes can conserve. In addition, to keep nodes operating sustainably, the following constraint should be met:

$$\tilde{\mathcal{E}}_h(u, t) \geq \tilde{\mathcal{E}}_c(u, t) \quad (3.3)$$

where $\tilde{\mathcal{E}}_h(u, t)$ and $\tilde{\mathcal{E}}_c(u, t)$ are the average value of the energy-harvesting rate and the energy consumption rate, respectively.

With these equations, we can model the remaining energy using Bernoulli processes. For example, we consider a network consisting of nodes equipped with a solar panel and a supercapacitor as the power supply. Assume it is a 15 mF supercapacitor, with capacity of 3675 mJ energy. We model a stochastic process showing how the remaining energy changes on a node, as shown in Figure 3.2 with Bernoulli processes.

3.4 Problem description

We described the challenges of topology control in EH-WSNs in Chapter 1, and we showed related work is limited in Chapter 2. The constraints in designing a topology control algorithm are as follows:

1. Since nodes only have local information, building a global-connected topology is difficult;
2. The energy levels of nodes are heterogeneous, which brings new challenges in EH-WSN.

In this work, we try to solve the topology control problem with the constraints. We focus on designing localized topology control algorithms for EH-WSNs with the following properties:

1. Nodes have neighbors with higher energy levels in the generated topology;

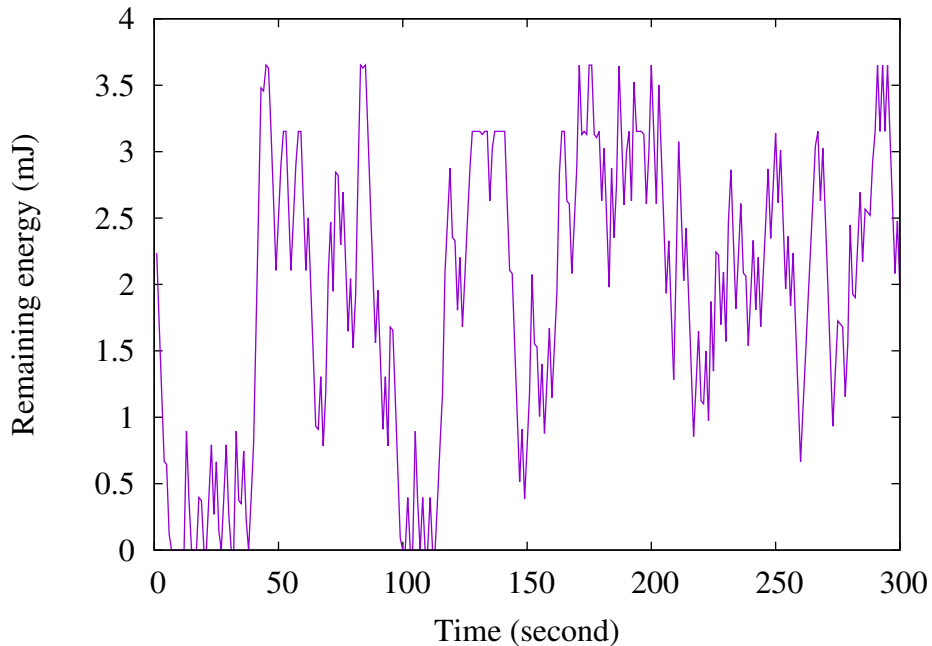


Figure 3.2: Remaining energy on a node modeled with Bernoulli process

2. The algorithm preserves the network connectivity;
3. The adjusted transmission power of nodes should be a factor of the neighbor selection criteria.

3.5 Assumptions

We model the network and nodes based on the following simple assumptions.

1. We assume the network is connected initially. This assumption ensures the topology generated by topology control algorithm can still be connected.
2. We consider the radio channels to be symmetric i.e., if node u can send a message to node v , then v can send a message to u with the same transmission power.
3. We assume that each node knows the distance and minimum transmission power required to reach each of its neighbor. These parameters can be estimated based on RSSI [29].
4. We assume a synchronous MAC in which nodes wakeup around the same time and transfer data. This can be implemented with a scheme such as S-MAC [30] or Contiki-MAC [31].

3.6 Summary

In this chapter, we presented system models, providing the mathematical description of the network model and the energy model. In addition, we summarized the problems in topology control in EH-WSN. Next, we will propose new algorithms in the following chapter.

Chapter 4

Proposed Algorithms

We discussed the system model, definitions, and assumptions in the previous chapter. In this chapter, we first list the design guidelines. Then we discuss the applicability of fault-tolerance as defined in WSNs to EH-WSNs. Next, we describe the proposed algorithms. Finally, we introduce the topology maintenance algorithm.

4.1 Design guidelines

In this work, we consider two typical scenarios:

1. An EH-WSN consists of sensor nodes and a sink, where nodes send their data to the sink. This is typical of WSN deployments.
2. A more generic scenario where the nodes are randomly positioned and nodes must exchange data between some source-destination pairs. In this case, one of the nodes could be a sink as well. This scenario can be envisioned in the realm of machine-to-machine applications.

The algorithms designed must adhere to the following constraints.

- The algorithm should be localized;
- It should have low communication overhead;
- The neighbor selection metric should be a function of the node's energy and neighbor's remaining energy;
- The global topology should be well-connected.

4.2 Fault-tolerance in EH-WSN

As we discussed in Chapter 2, in order to prevent the network from disconnection due to nodes failure, an advocated approach for assuring a fault-

tolerant WSN is to provide global k -connectivity. If a graph is a k -vertex-connected graph, it means by removing $k - 1$ vertices, the network is still connected. Therefore, building a k -connected topology can tolerate failure of k nodes.

One way to achieve a k -connected network is to control the deployment of nodes, placing nodes at desired positions. Thus the network is manually set to keep k -connectivity. However, this approach is neither practical in WSNs nor in EH-WSNs in many applications. So we discard this approach.

Another approach is to apply topology control algorithms to select the links among nodes, building a k -connected topology. However, constructing a minimum-cost k -connected graph is an *NP*-Hard problem. Approximate algorithms have been proposed to construct k -connected topology. For instance, as shown in Section 2.2.1, $FLSS_k$ is a typical distributed algorithm to build such topologies. We would like to discuss the problems of building a k -connected topology in EH-WSN, a distributed topology control algorithm building a k -connected topology typically has the following drawbacks:

1. Since nodes require local sub-graphs, it implies nodes need multiple hops neighbor information. This brings extra communication overheads, causing high transmission energy consumption.
2. Given nodes know the local sub-graph, whether the sub-graph is k -connected needs to be tested by using network flow techniques. This also results in high time and communication complexities. Jorgic et al. present a work on localized detection of k -connectivity [32], which shows that it is impossible for nodes, based on local knowledge, to be accurate with respect to global connectivity properties.

These two major drawbacks indicate that constructing k -connected topology based on local information is difficult, and approximate algorithms result in high communication overheads and high transmission energy consumption. In addition, in an EH-WSN, energy is a critical resource, and managing energy smartly is an objective in topology control. Therefore, building a k -connected topology to provide fault-tolerance is unrealistic in EH-WSNs with respect to the energy consumption. Instead, it is more practical to build well-connected topologies in EH-WSNs, which makes nodes have at least one link to other nodes. In our proposed algorithms, we comply with conclusion, designing topology control algorithms in EH-WSNs that provides fault-tolerance.

4.3 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 various 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.

The first phase of EBTC is topology construction. Nodes first collect their neighbor information, including the remaining energy and the distances between nodes. Then each node selects neighbors according to neighbor selection metric based on local information. Finally, nodes adjust their transmission power to the lowest value that is needed to reach the farthest neighbors. The adjusted transmission power is called *computed transmission power*. There are two major differences between EBTC-1 and EBTC-2: 1) They use different strategies to trigger nodes to initialize the neighbor information collection process; 2) nodes select neighbors based on different criteria.

We discuss and elaborate the EBTC-* as follows, explaining how the algorithms work.

4.4 Topology construction in EBTC-1

First, we consider a convergecast sensor network scenario, where nodes send the collected data towards the sink. EBTC-1 is designed to guarantee that the topology is well-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, in the next

step we begin neighbor selection phase. Here we define the energy threshold E_T , 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 hold neighbors that need low transmission power to reach. This minimizes energy expenditure on the node. Further, by selecting based on energy as the second criteria ensures one of the following: (a) if there are high energy neighbors close to the node, then lesser number of neighbors are selected; (b) if there are 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

```

1 MessageLevel := 0
2 if NodeID = InitializerID then
3   | Broadcast HELLO message with MessageLevel information at
   | maximum transmission power
4 else
5   | When receive HELLO message from node  $v$ 
6   |  $MessageLevel := v.MessageLevel + 1$ 
7   | Send HELLO message with MessageLevel
8   |  $N(u) := N(u) \cup \{v\}$ 
9 end
10 Wait for all nodes to finish neighbor information collection procedure
11 Compute  $N'(u)$  using neighbor selection Algorithm 2
12 Construct bi-directional links by adding missing links

```

As described in Algorithm 1, by sending messages in a level-based order, all nodes in the network are discovered and connected. We prove the correctness of EBTC-1 as follows.

Lemma 1. *All active nodes in a network are discovered using the topology control algorithm EBTC-1.*

Proof. First, all nodes are marked as *undiscovered* at the beginning. Initialized by the initializer node (or sink), the nodes within the transmission range with maximum transmission power of the initializer receive the message and are marked as *discovered*. Then, those *discovered* nodes send messages to their neighbors. Recursively, all nodes receive and send messages. Since all nodes are placed within the transmission range, all nodes are discovered. \square

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

- 1 Sort the nodes in $N(u)$ in ascending order of distance
- 2 $N_E := 0$; $N'(u) := \emptyset$
- 3 **foreach** v in $N(u)$ in this order **do**
- 4 **if** $state(v) = Active$ and $N_E < E_T$ and $v.MessageLevel < u.MessageLevel$ **then**
- 5 $N'(u) := N'(u) \cup \{v\}$
- 6 $N_E := N_E + Energy(v)$
- 7 **end**
- 8 **end**
- 9 Adjust transmission power to the minimum value needed to reach the farthest node in $N'(u)$

Theorem 1. *EBTC-1 algorithm establishes a connected topology where nodes in the topology are active.*

Proof. According to Lemma 1, each active node are discovered and they start the topology construction process in EBTC-1. Conforming to this algorithm, every active node has at least one link (edge) to the parent node. Thus the generated topology is connected. \square

4.5 Topology construction in EBTC-2

EBTC-2 is for the generic case where there is no hierarchy. It is more challenging to construct a well-connected topology with just one-hop information. If incorrect set of neighbors are selected, then the resultant global topology 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 Chapter 5.

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 hopcount information. Once the neighbor information phase is completed, each link is assigned a weight as in Equation 4.1:

$$w(u, v) = \alpha \cdot \frac{E_v}{E_{max}} + (1 - \alpha) \cdot \left(1 - \frac{RSSI_{u,v}}{RSSI_{min}}\right) \quad (4.1)$$

$w(u, v)$ 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

- 1 Broadcast HELLO message at maximum transmission power
 - 2 Upon receiving message from node v :
 - 3 $N(u) := N(u) \cup \{v\}$
 - 4 Wait for all nodes to finish neighbor information collection procedure
 - 5 Compute $N'(u)$ using neighbor selection Algorithm 4
 - 6 Construct bi-directional links by adding missing links
-

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

- 1 Sort the nodes in $N(u)$ in ascending order of weight function
 - 2 $N_E := 0$; $N'(u) := \emptyset$
 - 3 **foreach** v in $N(u)$ in this order **do**
 - 4 **if** $state(v) = Active$ and $N_E < E_T$ **then**
 - 5 $N'(u) := N'(u) \cup \{v\}$
 - 6 $N_E := N_E + Energy(v)$
 - 7 **end**
 - 8 **end**
-

4.5.1 The influence of α

In Equation 4.1, the value of the weight function is proportional to the remaining energy of the neighbor E_v , and is also proportional to the received signal from the neighbor $RSSI_{u,v}$. The factor α is employed to balance the weights of E_v and $RSSI_{u,v}$. Increasing the value of α will give more weight to the energy aspect, while decreasing it will give more weight to the distance (the value of RSSI). Normally, we can keep a balance between

these two factors, setting α to 0.5 is a fair choice. We will revisit this issue in Section 5.4.1.

4.5.2 Connectivity

In EBTC-2, the network connectivity is related to E_T . By choosing a proper value of E_T , the network is connected.

Theorem 2. *In EBTC-2, if the value of E_T is infinity, then the probability that the generated topology containing active nodes is connected approaches 1. This is described as: $E_T \rightarrow \infty \implies Pr(\text{Topology is connected}) \rightarrow 1$.*

Proof. By increasing the value of E_T to infinity, all active nodes select all neighbors that are within the transmission range. Therefore, the generated topology is identical to the original topology. Since the original topology is connected, the generated topology is consequently connected. \square

However, setting E_T to infinity is unrealistic. Instead, we choose a practical value of E_T to show the probability that the topology is connected is high. We will discuss the influence of E_T in Section 5.4.2 based on experiments.

4.6 Topology maintenance

One motivation for those k -connected topology control algorithms is that there is only topology construction phase in traditional WSN. This means the network has no knowledge when a node is going to die because of the depletion of energy. In order to reconstruct the topology whenever it is needed, topology maintenance techniques are proposed.

4.6.1 Maintenance strategy

As discussed in other studies, such as in [33], where authors extended the definition of topology control to include topology construction and topology maintenance, and presented different strategies to perform topology maintenance, including static, dynamic and hybrid techniques, with the global and local scope and using different triggering criteria. In the energy-harvesting environment, maintaining the topology can be done by an energy-triggered mechanism.

In this work, we implement a simple event-triggered (based on energy) to initiate topology maintenance, which is described in Algorithm 5. 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 (e.g., distance based in EBTC-1).

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

- 1 Upon receiving broadcast message $(v, Energy(v))$ from node v :
 - 2 **if** $Energy(v) > UpperBound$ or $N_E < E_T$ **then**
 - 3 | Set $(v, Active)$ in $N(u)$
 - 4 **else if** $Energy(v) < LowerBound$ **then**
 - 5 | Set $(v, Inactive)$ in $N(u)$
 - 6 **end**
 - 7 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. A node sends a message whenever it is *Active* and does not have sufficient number of neighbors i.e., sum of neighbors' is less than the energy threshold E_T .

4.6.2 Energy bounds

Since nodes in EH-WSN harvest and consume energy in different time, we consider a general case, where the remaining energy of a node is a stochastic process. In addition, we need to set up two states, namely *Active* and *Inactive*, to represent the states of nodes in the network.

A naive idea is to set a energy threshold, so that when the remaining energy is below the threshold, we mark it as *Inactive*; when the energy is above the threshold, we mark it as *Active*. When energy condition changes, the node broadcasts the change to its neighbors. However, due to the variation, the drawback of the “single bound” strategy is that nodes may send notification messages many times.

A practical optimization is to employ the “double bounds” strategy, namely an upper bound and a lower bound. As shown in Figure 4.1, the upper bound is depicted as green line, while the lower bound is the red line. In this mode, nodes only send beacon messages when the energy goes up crossing the upper bound or goes down crossing the lower bound. This approach reduces the messages that nodes need to exchange. As a result, the total energy consumption is therefore reduced compared to the “single bound” strategy.

We assume the nodes have only 15 mF supercapacitor as the energy buffer. Based on this and the energy required to transmit a message, we chose the *LowerBound* to be 600 μ J and the *UpperBound* to be 1500 μ J.

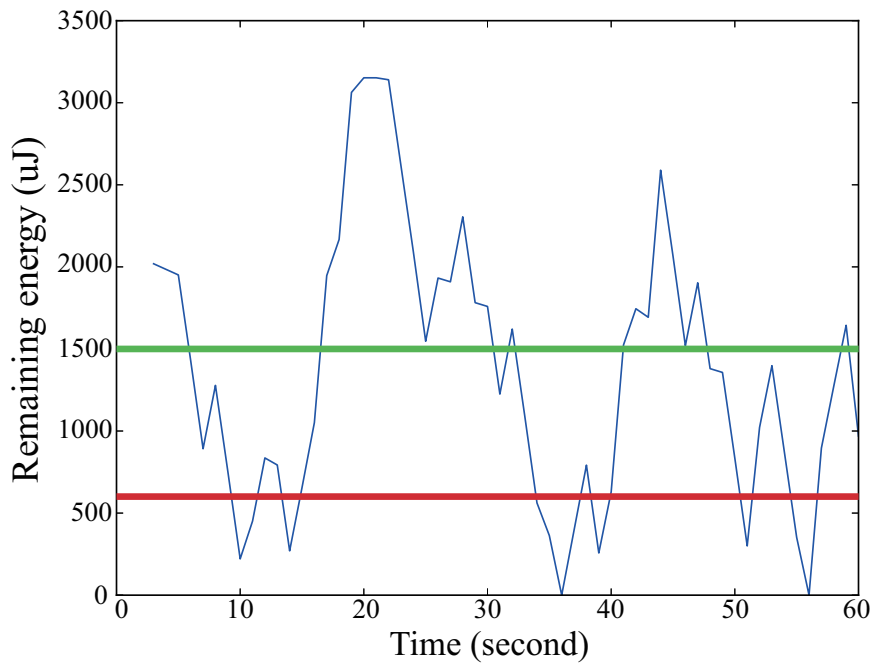


Figure 4.1: Double energy bounds

4.7 Summary

In this chapter, we first showed the motivation and design guidelines for new algorithms in EH-WSN. Then we proposed the EBTC-* algorithms in this chapter. EBTC-1 and EBTC-2 are designed for two scenarios in EH-WSN: convergecast networks with sinks, and generic networks. The two algorithms are proposed to construct a well-connected topology. In the next chapter, we will evaluate the proposed algorithms with different metrics.

Chapter 5

Performance Evaluation

In the previous chapter, we pitched the EBTC-* algorithm, describing how they are designed for EH-WSN. To evaluate the performance of the proposed algorithms, we did extensive study.

5.1 Experimental setup

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 [26] can only operate between 2.7 V-3.3 V, all of the energy in the supercapacitor cannot be used. Therefore, the maximum usable energy $E_{max} = 3675 \text{ mJ}$. We employ the energy model described in Section 3.3. For our simulations, we model this variable as Bernoulli random process with a fixed probability p , because it introduces high dynamicity in energy levels, thereby creating a highly dynamic network. We perform the simulations on Cooja simulator [34] in Contiki-OS 2.7 [35]. The advantage of using Cooja 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 the energy usage. The other simulation parameters are listed in Table 5.1. In addition, we consider collisions and other effects of wireless communications.

5.2 Evaluation metrics

To evaluate the proposed algorithms, we take two categories of metrics to quantify the results: network metrics and energy-harvesting related metrics.

Table 5.1: Simulation parameters

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

5.2.1 Network metrics

We first focus on the network properties of the generated topology. Especially, we evaluate the following metrics.

Node degree: Generally, a node with fewer neighbors means lesser interference in the transmission range. However, without knowing the transmission power of a node, we cannot conclude the relation between node degree and interference. In addition, in an ad hoc network, where nodes can send messages to the surrounding nodes, the number of neighbors suggests the number of choices a node has when sending messages to other nodes. Moreover, higher node degree improves the fault-tolerance property in a network.

Computed transmission power: In Section 1.2, we introduced the motivation of topology control: adjust the transmission power of nodes to achieve an energy-efficient network. The computed transmission power of nodes in a network is important as it affects the network connectivity and interference. Consequently, we would like to investigate how the proposed algorithms reduce the transmission power.

Transmission energy consumption: When constructing the topology, nodes send messages to acquire the neighborhood information. The more messages a node send, the higher transmission energy it consumes. Traditional topology control algorithms in WSN rarely consider the transmission energy consumption. Nonetheless, when it comes to an EH-WSN, where the energy utilization is important, we would like to maximize the energy usage and evaluate the topology control algorithms in terms of this metric. Plus, it also shows the complexity of an algorithm as constructing topology requires exchanging messages among nodes. Therefore, low transmission energy consumption indicates low message complexity in this sense.

Spanner factor: The spanner factor is defined in Equation 5.1. It describes how the lengths of hops in the generated topology are stretched. Since nodes in the generated topology reduce the transmission power, the length of the shortest path to other nodes are consequently increased. This indicates the number of hops among nodes increases. To evaluate this implication, we used this metric.

$$\rho(G_0) = \frac{\sum_{\forall(u,v) \in E(G_0), u \neq v} |ShortestPath(u, v, G_0)|}{\sum_{\forall(u,v) \in E(G_{max}), u \neq v} |ShortestPath(u, v, G_{max})|} \quad (5.1)$$

Interference: As nodes transmit data using the same wireless radio channel, interference occurs when nodes send and receive data at the same time. The interference is related to the transmission power — reducing transmission power decreases the transmission range, therefore, fewer nodes will be in the interference area.

5.2.2 Energy-harvesting related metrics

Apart from the metrics mentioned above, when studying the EH-WSN, some other metrics are also significant. The following metrics we propose are based on the energy aspect in an EH-WSN. The intuition behind them is that with the renewable energy in an EH-WSN, the network is adapted. Consequently, the generated topology should also be dynamic. In addition, considering the heterogeneous property in terms of nodes' residual energy, it is necessary to study the impact of energy.

Remaining energy per neighbor: This metric implies how nodes select neighbors with respect to the energy. In a battery-powered WSN, topology control algorithms do not consider the remaining energy of nodes. So when a node selects neighbors, it treats the neighbors without distinction in terms of the energy levels. However, in our proposed algorithms, they distinguish the energy levels of nodes.

Connectivity over time: In an EH-WSN, when a topology is constructed, the topology does not last for ever. Due to the energy changes over time, topology is updated by the topology maintenance mechanism. We propose this metric to assess how the proposed algorithms maintain the network connectivity.

5.3 Results

We evaluate the performance of EBTC-1 and EBTC-2 against K-Neigh and CBTC with respect to above 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 focuses on selecting neighbors with high remaining energy. Therefore, we would like to show how the proposed algorithms achieve the design goals in EH-WSNs, rather than just comparing the standard topology metrics such as stretch factor or other graph metrics.

In order to give an impression of the topology generated using the proposed algorithms, we show the original topology and the generated topology in Figure 5.1.

Node degree: Firstly, we show that our algorithms reduce interference by reducing the number of edges in Figure 5.2. As compared to the original graph (shown by “None”), the graphs generated by EBTC-* are the lowest even among K-Neigh and CBTC. The K-Neigh algorithm first selects a fixed number k of closest neighbors, in this case, we chose $k = 9$ as recommended [22]. Then nodes exchange the neighbor lists to keep symmetric neighbor lists. Therefore, the resultant number of neighbors is less than k . As for the CBTC algorithm, the node degree is bound by select at least one neighbor in every cone. The cone angle is set to $\frac{2\pi}{3}$ as described in [19].

The node degree of EBTC-* is bound by the energy threshold E_T as mentioned in the algorithm description. Therefore, by adjusting E_T , we can have different results of node degree.

Computed transmission power: Figure 5.4 plots 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.

Transmission energy consumption: Figure 5.5 depicts the average transmission energy consumption of each node in topologies generated by various algorithms. The results show that EBTC-1 and EBTC-2 consume low energy to construct topologies, which is important in terms of building an energy-efficient network topology. To quantify the transmission energy consumption, we know the number of messages needed to broadcast is limited, that is, each node broadcasts two messages when constructing the topology: the HELLO message and the neighbor list message.

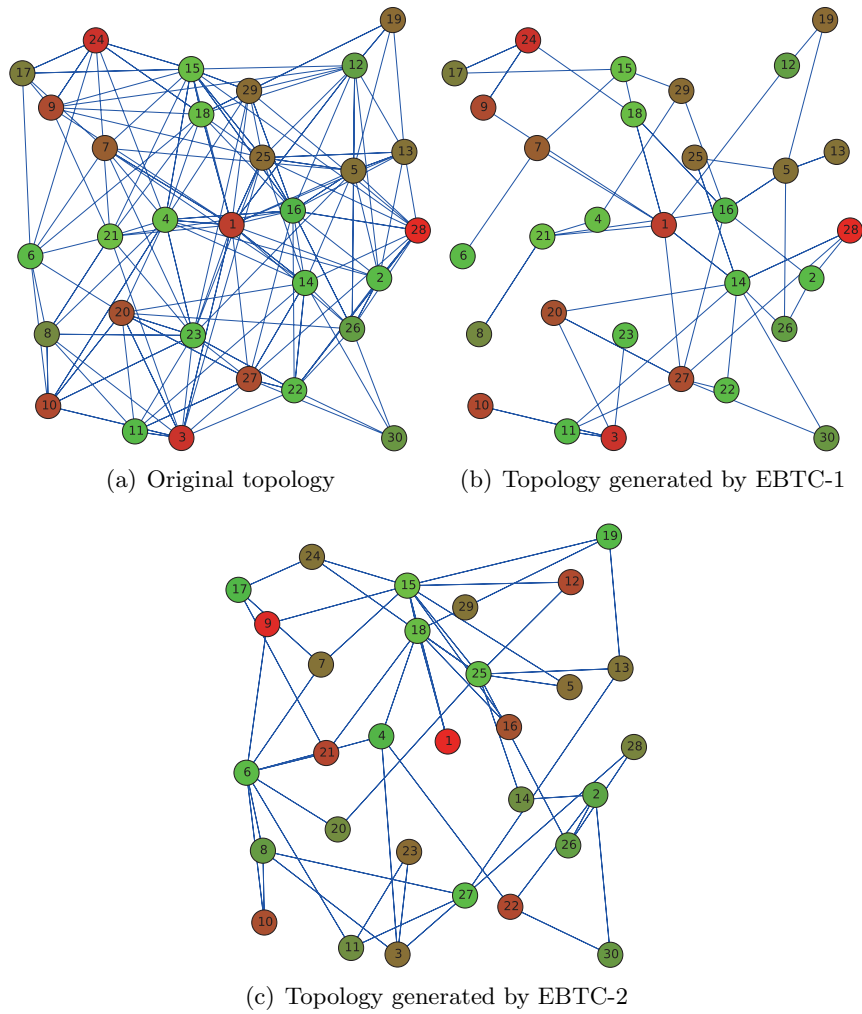


Figure 5.1: A demonstration of the topology generated by EBTC-1. Gradient of colors are used to indicate the energy levels of the nodes. Green implies higher energy while red is for lower energy levels. Node 1 at the center is the sink node.

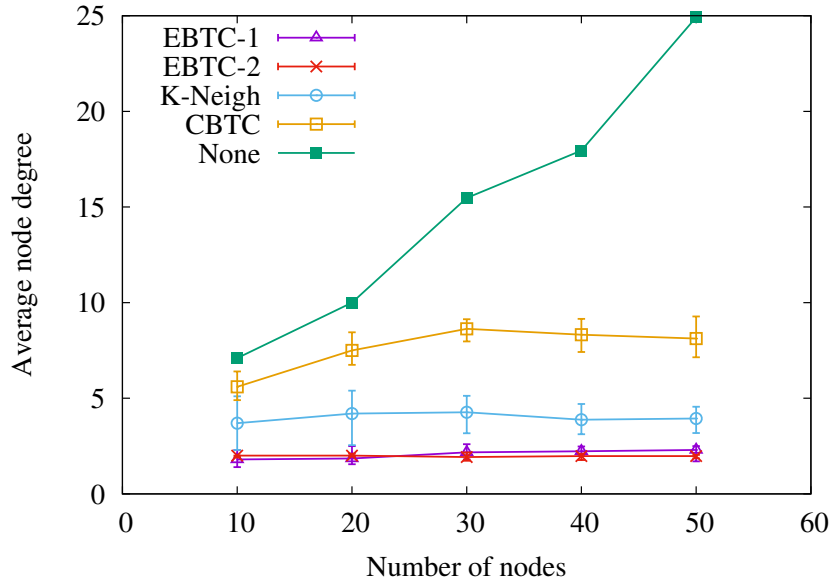


Figure 5.2: Average node degree in the resultant topology of four algorithms. EBTC-* have low average node degree compared to other two algorithms.

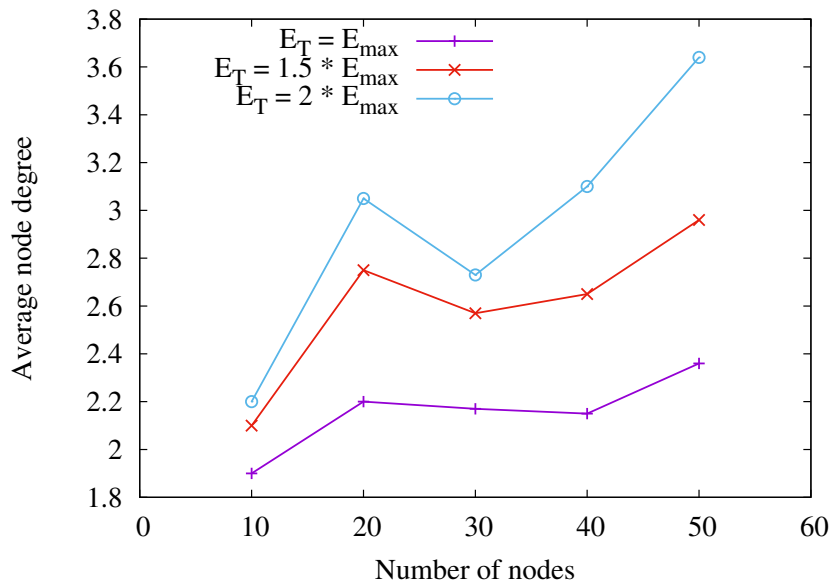


Figure 5.3: Average node degree in the resultant topology of EBTC-2. The topology was well-connected.

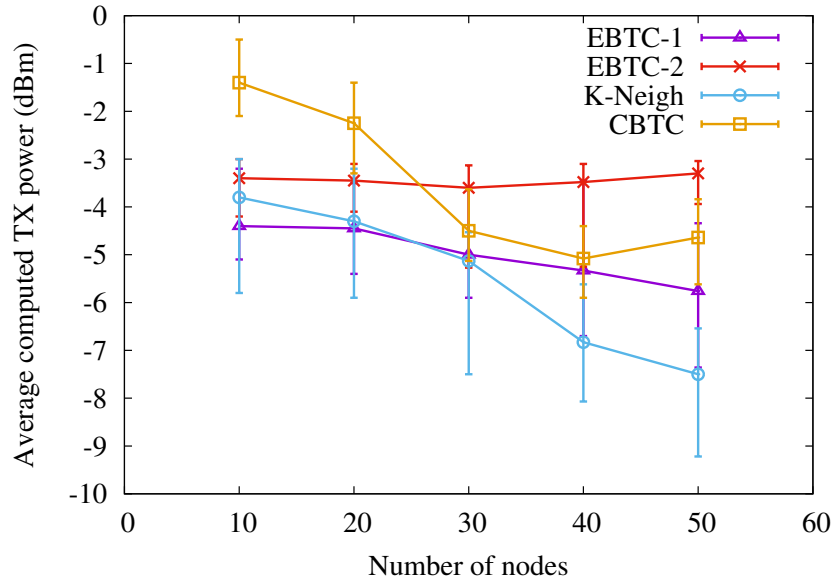


Figure 5.4: All algorithms reduce transmission power. EBTC-1 reduces more transmission power because it selects the closest neighbor; The transmission power is more stable in EBTC-2, which is a consequence of employing the weight function.

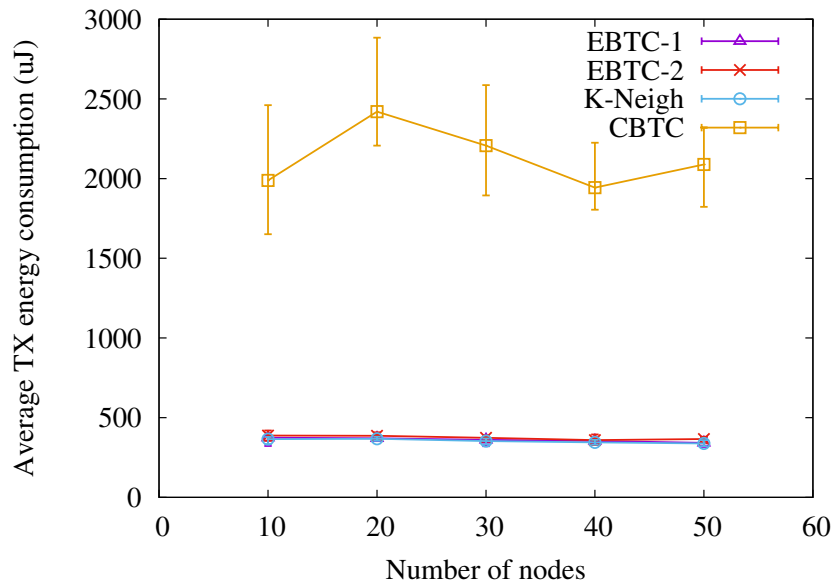


Figure 5.5: EBTC-* and K-Neigh use fewer messages to construct topologies compared to CBTC. Therefore, the transmission energy consumption is low.

Interference: In order to evaluate the interference in the generated topology, we measure the physical node degree, which shows how many nodes are in the transmission range of every node. This is based on the fact that nodes in the transmission range will suffer interference when they transmit data at the same time. Therefore, the lower physical node degree, the lesser interference would occur.

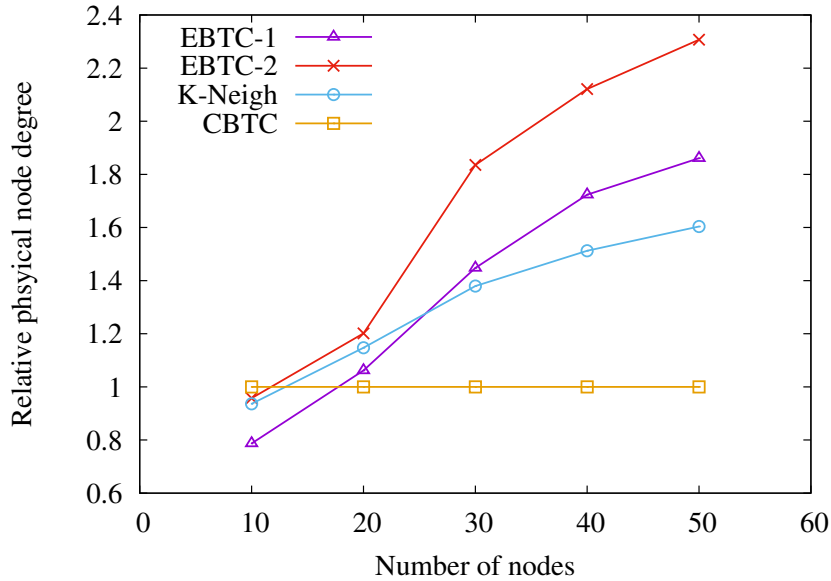


Figure 5.6: Relative physical node degree, where we use the values of CBTC as the baseline.

As shown in Figure 5.6, the average physical node degree of EBTC-* is relatively higher. This is the implication of selecting neighbors based on not only the distance but also the energy levels. We argue the results are acceptable since the topology control algorithms have the trade-off between selecting high energy neighbors and reducing the interference.

Spanner factor: Figure 5.7 shows the spanner factor in the generated topology of various algorithms. Firstly, we observe that the spanner factors in all algorithms are greater than 1, which means the shortest paths in all generated topologies are stretched. Secondly, EBTC-* have higher values of spanner factor. Revisiting the algorithms, we know that the nodes with higher remaining energy are selected frequently, which forms the backbones in the topology. Consequently, nodes with less remaining energy need to communicate with others via the “high energy nodes”. This is the main reason why the shortest paths become longer.

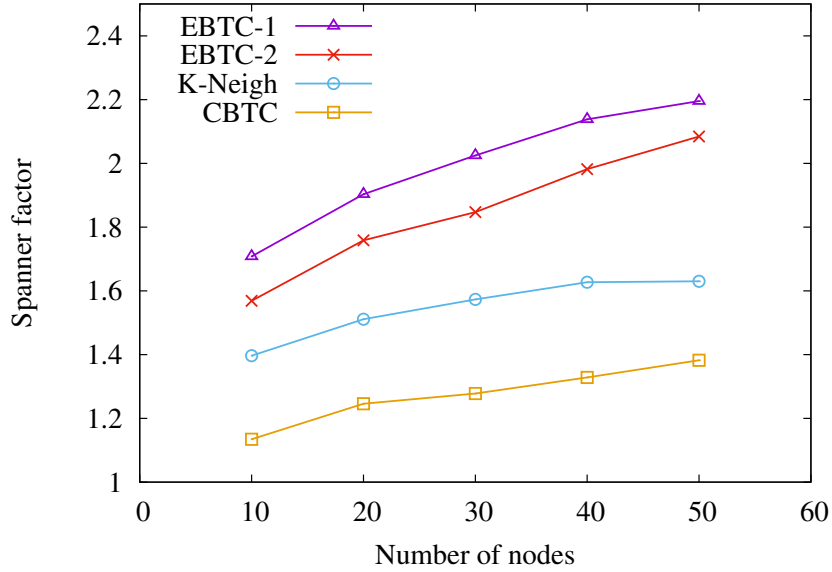


Figure 5.7: Spanner factor of the generated topology. The values of EBTC-* are higher.

Remaining energy per neighbor: Figure 5.8 illustrates the average remaining energy per neighbor of the topologies derived under different algorithms. The average remaining energy per neighbor of EBTC-2 is always higher compared to other algorithm, while the value of EBTC-1 is higher than K-Neigh and CBTC in most cases. This results demonstrate the basic idea of EBTC: nodes select neighbors that are with high remaining energy.

Connectivity over time: Apart from topology construction phase, we also evaluate the network connectivity issue in topology maintenance phase. With the topology maintenance procedure described in Algorithm 5, the network has the ability to maintain a dynamic topology. We consider every time the topology is changes as an *iteration*.

Figure 5.9 shows how the connectivity changes when the topology iterates. The results show that the network may be disconnected for some iterations. The reasons are as follows: 1) the notification message is lost; 2) the fixed deployment of the network makes the node has no active neighbors at certain time. Though these reasons are inevitable, with the help of the topology maintenance mechanism, the network has the ability to recover the connectivity. In addition, Figure 5.10 shows the connectivity over time, where 0-connectivity means the network is disconnected. The first few iterations can be disconnected because the topology construction phase is asynchronous, making some nodes still building their local topology. After

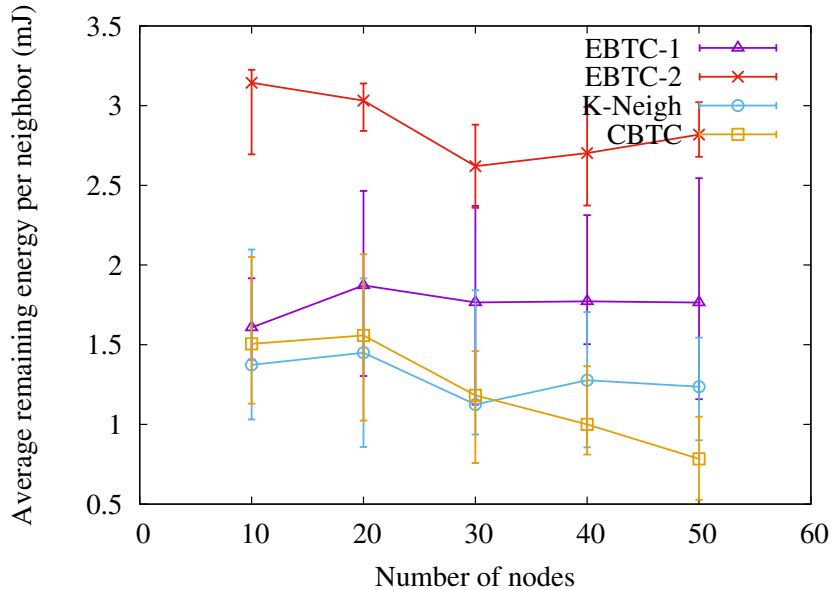


Figure 5.8: EBTC-* select neighbors with higher remaining energy, other algorithms select neighbors without considering the energy levels.

the topology is connected, which implies the topology construction phase is done, and the topology is stable, we can see that the topology is updated over time. The topology is well-connected for most of the iterations, and it also shows that the topology maintenance mechanism is valid and efficient as it can recover the connectivity.

5.4 Discussion

As described in the algorithms, we provide several flexible parameters that can be tuned and discussed.

5.4.1 Influence of α in EBTC-2

EBTC-2 employs a metric to quantify and select neighbors, which is affected by the weight of α . α can give weight to either energy of the neighbor or the distance, making the weight function generic. Here, we study the influence of α .

Figure 5.11 shows the average remaining energy per neighbor in terms of different values of α . We notice that giving more weight to remaining energy of neighbors leads to higher remaining energy per neighbor. Plus, as shown in Figure 5.12, 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

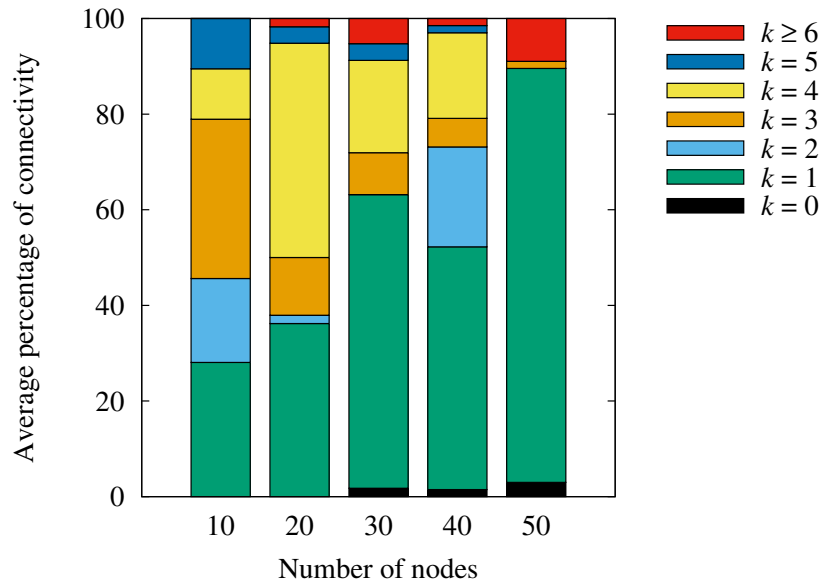


Figure 5.9: The k -connectivity property in topology maintenance phase with EBTC-2. Though the network is disconnected sometime, the connectivity recovers after next iteration.

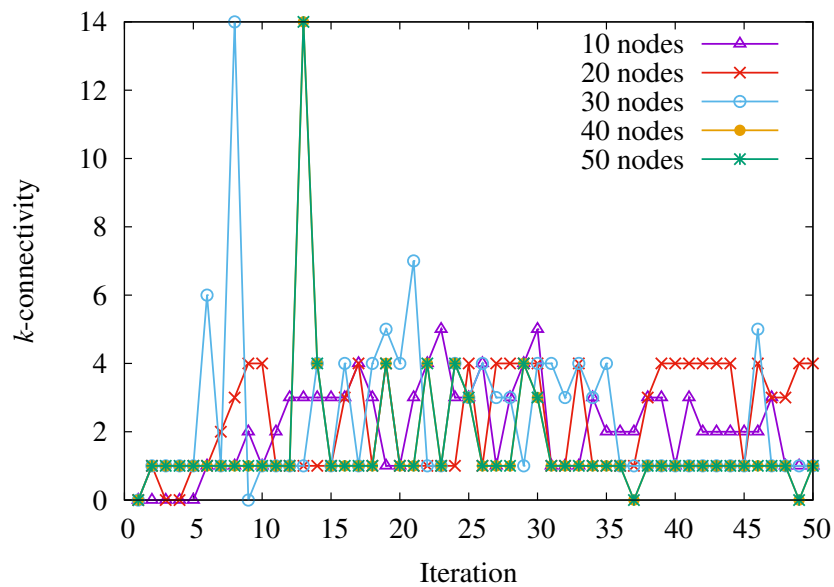


Figure 5.10: The connectivity in a network over time with EBTC-2. Though the network is disconnected sometime, the connectivity recovers after next iteration.

neighbors (be well-connected) or choose higher energy neighbors who can route packets for the nodes.

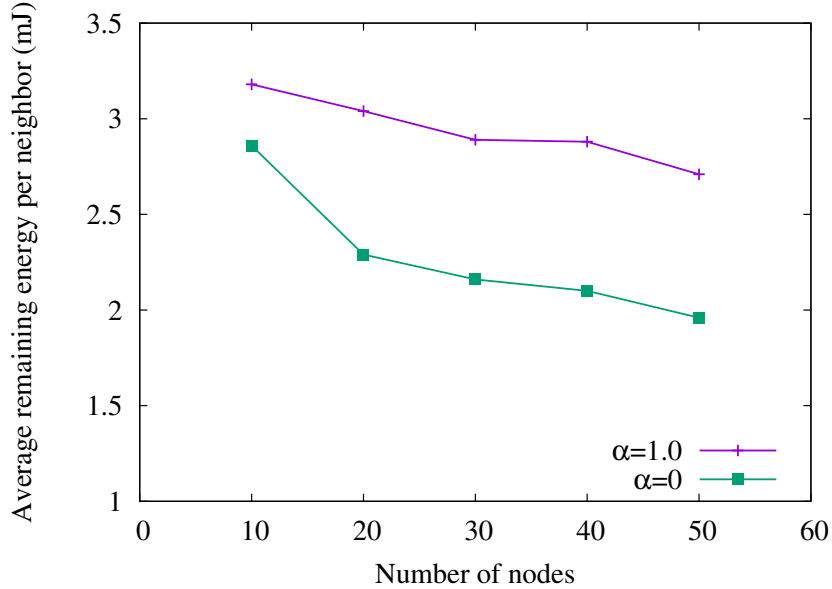


Figure 5.11: The average remaining energy per neighbor is related to the weight function in EBTC-2. α is higher shows EBTC-2 gives more important to the neighbor’s energy.

5.4.2 Energy threshold E_T

Since the node degree in our algorithms is based on energy threshold E_T , the obvious question is how guaranteed is the connectivity. This is more important in the case of EBTC-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, as shown in [23], if a node connects to $\Theta(\log n)$ nearest neighbors, the graph will be connected. Based on this, we evaluate the topology construction for various E_T .

As shown in Figure 5.3, average node degree increases with higher E_T . According to the simulations, we found that value of $E_T \geq E_{max}$ rendered the graph well-connected for number of n . The distribution of network connectivity is shown in Figure 5.13.

5.5 Testbed

We conducted experiments on Indriya [36], a WSN testbed that consists of 96 available nodes, where nodes in this testbed are deployed across three floors.

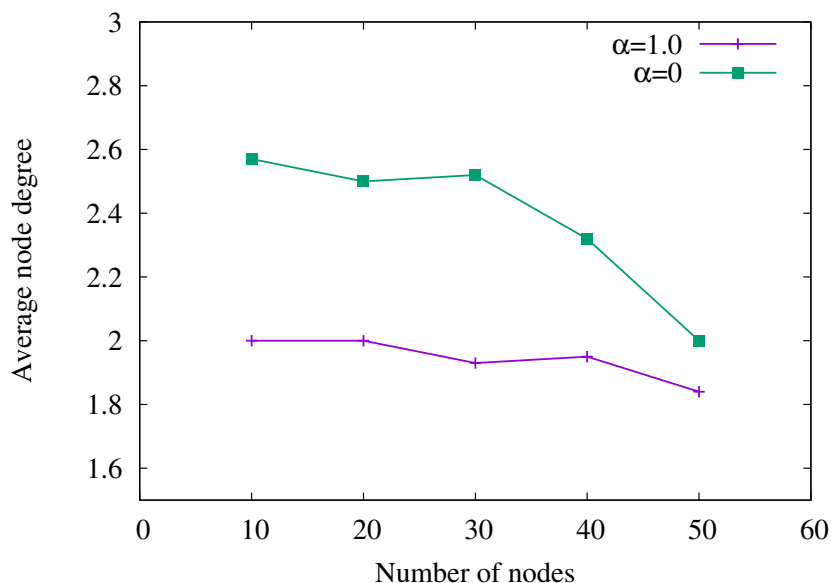


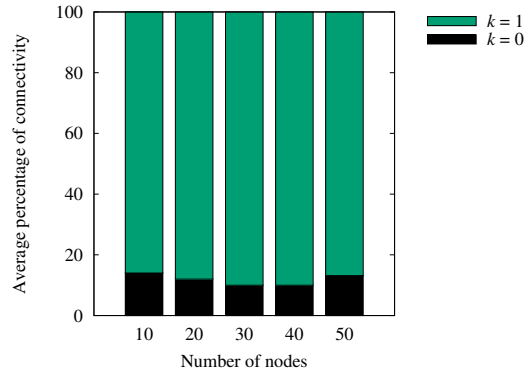
Figure 5.12: As a consequence in changing α value in EBTC-2, the average node degree varies. When the value of α is higher, EBTC-2 focuses more on neighbor’s energy. Consequently, it only needs to select fewer neighbors to meet the neighbor selection criteria.

The type of nodes is TelosB, which built of TI-MSP430 microcontroller and equipped with CC2420 radio.

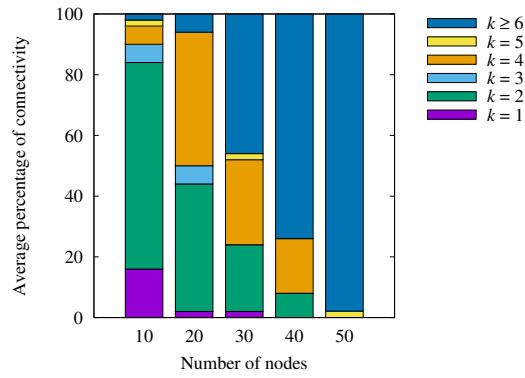
When testing the proposed algorithms on the testbed, the following limitations exist: 1) As the deployment is fixed and some nodes in the network are unavailable, the connectivity issue is more important compared to other metrics; 2) due to the complexity of the indoor environment, the wireless channel is lossier compared to the simulation setting. Therefore, we tested the EBTC-* algorithms mainly to evaluate the connectivity issue in the generated topology.

We observed that broadcast messages were easier to get lost in the realistic deployment compared to the simulated radio channel. Thus, the generated topology was not always connected. Nonetheless, since our proposed algorithms enable the network to be fault-tolerant, the connectivity can be recovered over iterations. In addition, we know that the fault-tolerance is related to the value of E_T , as E_T decides how many neighbors to select.

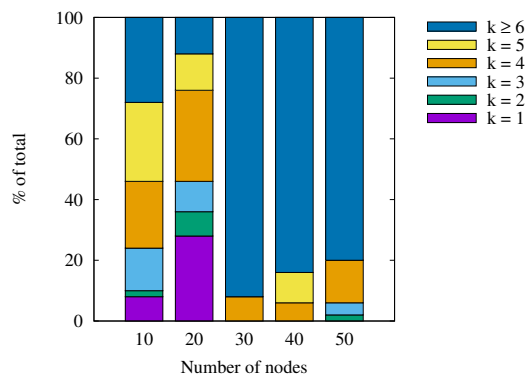
We evaluated the results of the generated topology in the EBTC-1 and the EBTC-2 algorithms. Table 5.2 shows the several evaluation metrics based on the testbed results. In this table, average node degree, average computed transmission (TX) power and spanner factors are similar to the results that are based on simulations. Furthermore, % of *connected* means



(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 well-connected.



(c) The k -connectivity in a network with $E_T = 2E_{max}$. The generated topology is a well-connected.

Figure 5.13: Network connectivity in EBTC-2 with different values of E_T . The topology generated by EBTC-2 is well-connected when $E_T = E_{max}$.

Table 5.2: Experiment results on the testbed.

Algorithm	E_T	Avg. node degree	Avg. TX power	% of connected	Spanner factor
EBTC-1	E_{max}	2.03	-19.07 dBm	61.40%	2.66
	$1.5 E_{max}$	2.54	-18.22 dBm	90.42%	2.54
	$2 E_{max}$	3.04	-15.08 dBm	96.29%	1.78
EBTC-2	E_{max}	2.17	-7.60 dBm	85.29%	1.54
	$1.5 E_{max}$	2.75	-6.07 dBm	91.43%	1.47
	$2 E_{max}$	3.80	-4.97 dBm	96.80%	1.26

network connectivity over iterations. The results from the table show that though the network experiences disconnection, the network has the ability to recover connectivity. Particularly, by tuning the value of E_T , the probability that the network is connected is increased. By setting $E_T = 2 E_{max}$, the connectivity over iterations is greater than 96% on the given testbed deployment.

5.6 Summary

In this chapter, we evaluated our proposed algorithms based on extensive simulations. We also tested our algorithms in a real-world deployment — the Indriya testbed. We used various evaluation metrics, including node degree, transmission energy consumption, computed transmission power, spanner factor, interference, remaining energy per neighbor. According to the results, it shows that when selecting neighbors, EBTC-* take neighbors' remaining energy into considerations. In addition, the results show that EBTC-* reduce transmission power, and uses low transmission energy to constructing the topology. We conclude that our proposed algorithms works in EH-WSN and meet the design goal.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

We investigated topology control—a topic that has been studied intensively in battery-powered WSNs but has not been fully explored in EH-WSN. We investigated the new challenges and constraints in topology control in EH-WSN setting. Since the nodes harvest energy from the ambience, number of nodes that are alive at a given instant keeps varying. This poses a bigger challenge in re-using the topology control algorithms proposed hitherto. Thus we proposed two localized topology control algorithms, namely EBTC-1 and EBTC-2. We evaluated the proposed algorithms based on simulations and on a real-world deployment—the Indriya testbed. As for the complexity, EBTC-* algorithms take only around 80 lines of code to implement.

The main contribution of this work is as follows: 1) We show that existing topology control algorithms that are designed for battery-powered WSN are not suitable for EH-WSN; 2) we propose two localized topology control algorithms for two typical scenarios in EH-WSN that maximize residual energy in every node, and nodes are assigned loads based on their energy levels; 3) the proposed algorithms are low-complexity and they can handle the dynamic variation in remaining energy levels at the node; 4) we conclude that, by choosing proper value of E_T , EBTC-2 algorithm can guarantee network connectivity.

With respect to the performance, simulation 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 terms of energy consumption and fault-tolerance, our proposed algorithms typically achieve 1-connected topology using 74% less energy compared to K-Neigh. The average spanner factors of EBTC-1 and EBTC-2 are 1.99 and 1.84 respectively, which shows that the average

lengths of shortest paths among nodes are increased. As for interference, the number of physical node degree is increased by at most 30% in EBTC-* algorithms compared to K-Neigh.

The increase in spanner factors and the interference are acceptable trade-offs for selecting neighbors based on distances and energy levels. With regard to topology maintenance, results show that EBTC-* are adaptable to the changes in energy levels, and it preserves a connected network in at least 97% of iterations over time. Testbed results demonstrate that in a real-world and complicated indoor environment, by increasing the value of E_T , our proposed algorithms still can keep the network connected over time, which proves that EBTC-* are flexible and implementable algorithms that meets the requirements.

Considering the connection to different layers of the network, our proposed algorithms can support other layers. For example, for MAC layer, EBTC-* can reduce contention as it lowers the interference in the network by reducing transmission power. In addition, EBTC-* can be integrated into routing protocols, providing the energy information of neighbors for further use. Therefore, routing protocols can utilize the energy information to create energy-efficient paths in EH-WSN.

In a nutshell, we can conclude our work suggests: 1) Nodes in EH-WSN should be assigned with different roles based on their energy levels; 2) topology control algorithms in EH-WSNs should select neighbors based on nodes' energy levels to keep the network connected; 3) as for fault-tolerance, by having the EBTC-* algorithms, instead of targeting at achieving a k -connected topology, the network is able to be well-connected and with high value of k with respect to k -connectivity property; 4) the definition of topology control needs to be extended when considering more aspects, such as the energy levels, in an EH-WSN other than only the impact of distance. These four should be new additions to the literature.

6.2 Further work

Many research topics in topology control in EH-WSN are still unexplored. EBTC-* algorithms are only the beginning and it has opened up many problems. Here we list some suggestions for further work as follows.

1. Designing a harvesting-aware system with a predictable energy model. In this case, the topology control algorithm coupled with prediction should utilize the energy model to build a more energy-efficient topology;
2. Reducing interference based on EBTC-* algorithms. Since EBTC-* algorithms introduce relatively higher interference in the network, one

possible research topic is to reduce the interference and yet achieve connectivity and reduce the energy usage;

3. Integrate EBTC-* algorithms with other protocols layer in EH-WSN. As the proposed algorithms are simple and practical, they can be implemented in real-world applications. For instance, the proposed algorithms can be integrated with data collection protocols, like the Collection Tree Protocol [37], to create energy-efficient paths.
4. Constructive interference (CI) is being used in many WSN applications. Topology control for a better beneficial CI may be the next level for both EH-WSN topology control as well as CI.

Publication

- [1] Xin Wang, Vijay S. Rao, R. Venkatesha Prasad, and Ignas Niemegeers. Choose wisely: Topology control in energy-harvesting wireless sensor networks. Submitted in *Consumer Communications and Networking Conference (CCNC), 2016 IEEE*.

Bibliography

- [1] P. Naz, S. Hengy, and P. Hamery, “Soldier detection using unattended acoustic and seismic sensors,” in *SPIE Defense, Security, and Sensing*, pp. 83890T–83890T, International Society for Optics and Photonics, 2012.
- [2] L. Mo, Y. He, Y. Liu, J. Zhao, S.-J. Tang, X.-Y. Li, and G. Dai, “Canopy closure estimates with greenorbs: sustainable sensing in the forest,” in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pp. 99–112, ACM, 2009.
- [3] R. V. Prasad, S. Devasenapathy, V. S. Rao, and J. Vazifehdan, “Reincarnation in the ambiance: Devices and networks with energy harvesting,” *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 1, pp. 195–213, 2014.
- [4] W. K. Seah, Y. Tan, and A. T. Chan, *Research in energy harvesting wireless sensor networks and the challenges ahead*. Springer, 2013.
- [5] D. Dondi, A. Bertacchini, D. Brunelli, L. Larcher, and L. Benini, “Modeling and optimization of a solar energy harvester system for self-powered wireless sensor networks,” *Industrial Electronics, IEEE Transactions on*, vol. 55, no. 7, pp. 2759–2766, 2008.
- [6] P. D. Mitcheson, E. M. Yeatman, G. K. Rao, A. S. Holmes, and T. C. Green, “Energy harvesting from human and machine motion for wireless electronic devices,” *Proceedings of the IEEE*, vol. 96, no. 9, pp. 1457–1486, 2008.
- [7] X. Lu and S.-H. Yang, “Thermal energy harvesting for wsns,” in *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pp. 3045–3052, IEEE, 2010.
- [8] H. D. Akaydin, N. Elvin, and Y. Andreopoulos, “Energy harvesting from highly unsteady fluid flows using piezoelectric materials,” *Journal of Intelligent Material Systems and Structures*, vol. 21, no. 13, pp. 1263–1278, 2010.

- [9] G. Park, T. Rosing, M. D. Todd, C. R. Farrar, and W. Hodgkiss, “Energy harvesting for structural health monitoring sensor networks,” *Journal of Infrastructure Systems*, vol. 14, no. 1, pp. 64–79, 2008.
- [10] B. Buchli, F. Sutton, J. Beutel, and L. Thiele, “Dynamic power management for long-term energy neutral operation of solar energy harvesting systems,” in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, pp. 31–45, ACM, 2014.
- [11] P. Santi, “Topology control in wireless ad hoc and sensor networks,” *ACM computing surveys (CSUR)*, vol. 37, no. 2, pp. 164–194, 2005.
- [12] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. Kumar, “Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the compow protocol,” in *European wireless conference*, vol. 2002, pp. 156–162, Florence, Italy, 2002.
- [13] N. Li, J. C. Hou, and L. Sha, “Design and analysis of an mst-based topology control algorithm,” *Wireless Communications, IEEE Transactions on*, vol. 4, no. 3, pp. 1195–1206, 2005.
- [14] A. Cornejo and N. A. Lynch, “Reliably detecting connectivity using local graph traits,” in *OPODIS*, pp. 87–102, Springer, 2010.
- [15] Q. Tan, W. An, Y. Han, Y. Liu, S. Ci, F.-M. Shao, and H. Tang, “Energy harvesting aware topology control with power adaptation in wireless sensor networks,” *Ad Hoc Networks*, vol. 27, pp. 44–56, 2015.
- [16] N. Li and J. C. Hou, “Localized topology control algorithms for heterogeneous wireless networks,” *IEEE/ACM Transactions on Networking (TON)*, vol. 13, no. 6, pp. 1313–1324, 2005.
- [17] S. Khuller, “Approximation algorithms for finding highly connected subgraphs,” *Vertex*, vol. 2, p. 2, 1997.
- [18] M. Bahramgiri, M. Hajiaghayi, and V. S. Mirrokni, “Fault-tolerant and 3-dimensional distributed topology control algorithms in wireless multi-hop networks,” *Wireless Networks*, vol. 12, no. 2, pp. 179–188, 2006.
- [19] 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,” in *Proceedings of the twentieth annual ACM symposium on Principles of distributed computing*, pp. 264–273, ACM, 2001.
- [20] N. Li and J. C. Hou, “Localized fault-tolerant topology control in wireless ad hoc networks,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 17, no. 4, pp. 307–320, 2006.

- [21] R. Wattenhofer and A. Zollinger, “Xtc: A practical topology control algorithm for ad-hoc networks,” in *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, p. 216, IEEE, 2004.
- [22] D. M. Blough, M. Leoncini, G. Resta, and P. Santi, “The k-neigh protocol for symmetric topology control in ad hoc networks,” in *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pp. 141–152, ACM, 2003.
- [23] 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.
- [24] X. Wang, M. Sheng, M. Liu, D. Zhai, and Y. Zhang, “Resp: A k-connected residual energy-aware topology control algorithm for ad hoc networks,” in *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*, pp. 1009–1014, IEEE, 2013.
- [25] H. Bagci, I. Korpeoglu, and A. Yazıcı, “A distributed fault-tolerant topology control algorithm for heterogeneous wireless sensor networks,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 26, no. 4, pp. 914–923, 2015.
- [26] T. Instruments, “Cc2420: 2.4 ghz ieee 802.15. 4/zigbee-ready rf transceiver,” *Available at Available at <http://www.ti.com/lit/gpn/cc2420>*, p. 53, 2006.
- [27] I. Howitt, J. Gutierrez, *et al.*, “Ieee 802.15. 4 low rate-wireless personal area network coexistence issues,” in *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, vol. 3, pp. 1481–1486, IEEE, 2003.
- [28] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, “Power management in energy harvesting sensor networks,” *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 6, no. 4, p. 32, 2007.
- [29] P. Kumar, L. Reddy, and S. Varma, “Distance measurement and error estimation scheme for rssi based localization in wireless sensor networks,” in *Wireless Communication and Sensor Networks (WCSN), 2009 Fifth IEEE Conference on*, pp. 1–4, IEEE, 2009.
- [30] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient mac protocol for wireless sensor networks,” in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1567–1576, IEEE, 2002.
- [31] A. Dunkels, “The contikimac radio duty cycling protocol,” 2011.

- [32] 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 *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on*, pp. 33–38, IEEE, 2007.
- [33] P. M. Wightman and M. A. Labrador, “Topology maintenance: Extending the lifetime of wireless sensor networks,” *Latin America Transactions, IEEE (Revista IEEE America Latina)*, vol. 8, no. 4, pp. 469–475, 2010.
- [34] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, “Cross-level sensor network simulation with cooja,” in *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*, pp. 641–648, IEEE, 2006.
- [35] A. Dunkels, B. Grönvall, and T. Voigt, “Contiki-a lightweight and flexible operating system for tiny networked sensors,” in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pp. 455–462, IEEE, 2004.
- [36] M. Doddavenkatappa, M. C. Chan, and A. L. Ananda, “Indriya: A low-cost, 3d wireless sensor network testbed,” in *Testbeds and Research Infrastructure. Development of Networks and Communities - 7th International ICST Conference, TridentCom 2011, Shanghai, China, April 17-19, 2011, Revised Selected Papers*, pp. 302–316, 2011.
- [37] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, “Collection tree protocol,” in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pp. 1–14, ACM, 2009.