Abstract

Topology control algorithms allow each node in a wireless multi-hop network to adjust the power at which it makes its transmissions and choose the set of neighbors with which it communicates directly, while preserving global goals such as connectivity or coverage. This allows each node to conserve energy and contribute to increasing the lifetime of the network. In this paper, in contrast to most previous work, we consider (i) both the energy costs of communication as well as the amount of available energy at each node, (ii) the realistic situation of varying rates of energy consumption at different nodes, and (iii) the fact that co-operation between nodes, where some nodes make a sacrifice by increasing energy consumption to help other nodes reduce their consumption, can be used to extend network lifetime. This paper introduces a new distributed topology control algorithm, called the Cooperative Topology Control with Adaptation (CTCA), based on a game-theoretic approach that maps the problem of maximizing the network’s lifetime into an ordinal potential game. We prove the existence of a Nash equilibrium for the game. Our simulation results indicate that the CTCA algorithm extends the life of a network by more than 50% compared to the best previously-known algorithm. We also study the performance of the distributed CTCA algorithm in comparison to an optimal centralized algorithm as a function of the communication ranges of nodes and node density.
I. Introduction

In wireless ad hoc networks, especially ad hoc sensor networks, the battery life of each node plays a critical role in determining the functional lifetime of the entire network. When a node exhausts its limited energy supply, it may fail to reach nearby nodes leading to a disconnected network and disabling some essential communications. Without energy, the node will also fail to continue the environmental monitoring activities essential to the functional operation of the system. Adding redundant nodes in the network may extend the functional lifetime but it is ultimately a less cost-effective approach. In this paper, we consider the problem of extending the lifetime of a network using a new adaptive game-theoretic approach.

Topology control is among the better-known approaches to conserving energy and prolonging a network’s functional life. In a topology control algorithm, each node adjusts the power at which it makes its transmissions to reduce the energy consumption to only what is needed to ensure topological goals such as connectivity or coverage. Examples of topology control algorithms include Directed Relative Neighborhood Graph (DRNG) [1], Directed Local Spanning Subgraph (DLSS) [1], Step Topology Control (STC) [2] and Routing Assisted Topology Control (RATC) [3]. In most traditional algorithms, the topology of the network is determined at the very beginning of the life of the network where the only consideration for each node is to reduce its transmission power while keeping the graph connected. After the execution of one of these algorithms, each node will transmit at the selected power level until it eventually runs out of energy. However, depending on the location of a node in relation to others, some nodes may end up with a much larger communication radius, and therefore a much larger transmission power, than some others. This uneven distribution of the assigned transmission powers may result in an unbalanced energy consumption at the nodes, leading to some nodes exhausting their energy far sooner than some others. Such a scenario can end the functional life of the network earlier than necessary. This highlights two weaknesses of these algorithms: they are not adaptive to different rates of energy consumption on different nodes and they do not allow cooperation between nodes to extend the network lifetime. Each of these weaknesses is addressed by the algorithm proposed in this paper: Cooperative Topology Control with Adaptation (CTCA).

![Fig. 1: An example illustrating cooperative topology control.](image)

We illustrate the principle of cooperative topology control with a simple toy example shown in Fig. 1. Suppose Fig. 1(a) illustrates the result of a topology control algorithm, where no node can reduce its transmission power unilaterally without disconnecting the graph. In this figure, the presence of an edge from one node, say $A$, to another node, say $B$, implies that $A$ can
transmit at a power level sufficient to reach node $B$. The communication radius of each node is shown by the dashed arcs. Assuming all nodes start out with the same energy supply and make transmissions at the same rate, we note that node $A$ has the largest energy cost and thus has the shortest lifetime. Node $C$, on the other hand, has the smallest transmission power, and therefore, has the longest lifetime. Traditional topology control algorithms discussed above will lead to the situation in Fig. 1(a) ending the functional life of the network when node $A$’s energy is exhausted even though node $C$ would have plenty of remaining energy. Fig. 1(b) illustrates a topology where node $C$ increases its transmission power so that it can now reach node $B$ directly. Now, node $A$ is able to reduce its transmission power to only be directly connected to node $C$, as shown in Fig. 1(c). This involves node $C$ making a sacrifice by increasing the power at which it makes its transmissions in order to allow node $A$ to reduce its transmission power, thus extending the life of node $A$ and of the network.

The actual power consumption for sending and receiving a data packet varies significantly depending on the radio environment of the space where the sensor nodes are located and also the electronics of the devices. The log-distance path loss model based on the path loss as a logarithmic function of the distance $d$ has been confirmed both theoretically and by measurements in a large variety of environments [4]–[6]. In this model, the path loss at distance $d$, $PL(d)$ is expressed as:

$$PL(d) = PL(d_0) + 10\gamma \log_{10}(d/d_0)$$

where the constant $d_0$ is an arbitrary reference distance and $\gamma$ is called the path loss exponent. This implies that the energy consumed to make a transmission across a distance $d$ is proportional to $d^\gamma$. Since $\gamma$ ranges from 2.5 to 6 in most real environments [4], especially over longer distances, a single transmission over distance $d$ often consumes more energy than two transmissions each over distance $d/2$. This motivates the goal of most topology control algorithms to choose multiple smaller hops in place of a single longer hop with the intent to reduce overall energy consumption. While device electronics can sometimes be such that choosing smaller hops—especially at smaller distances—does not always guarantee lower energy consumption, there is another good reason to choose smaller hops: reducing interference in all communications. Therefore, a general goal of a topology control algorithm is to achieve lower transmission powers for all the sensor nodes in order to reduce both energy consumption and interference [7]. In other words, the topology illustrated in Fig. 1(c) is more desirable.

In this paper, we employ game theory to facilitate such topology control that allows cooperation between nodes as illustrated in Fig. 1. Our approach is through developing an ordinal potential game [3], [8], [9] into which our problem can be mapped, so that all nodes pursue a localized strategy that can be expressed through a single global function, or the global potential function. Our approach also allows an adaptive strategy so that a node does not end up with the same power level through its entire lifetime. This is significant to extending the network lifetime because it is almost always the case that different nodes consume energy at different rates. Our approach to allowing adaptation is through incorporating the energy remaining on the nodes in the neighborhood into the decisions made by each node. Since this remaining energy changes over the life of a network, our topology control algorithm adaptively adjusts the power levels at each node. This constantly keeps shifting energy consumption from nodes with less energy reserves to those with more energy reserves, thus extending the life of the network.

3
A. Problem Statement

Given a wireless sensor network, let graph $G(t) = (N, E(t))$ represent its topology at time $t$, where $N_i \in N$ represents a node within the network, and $(N_i, N_j) \in E(t)$ represents the fact that node $N_j$ is within node $N_i$’s communication radius and can hear from $N_i$ directly at time $t$. Assume $G(t)$ is a connected graph at time $t$. Topology control algorithms have traditionally emphasized preserving connectivity as a constraint while pursuing the goal of reduced energy consumption at each node.

However, depending on the type of application for which an ad hoc sensor network is deployed, it is possible that a network is functional even if a certain subset of nodes runs out of energy [10]–[12]. The functional lifetime of a network, therefore, depends on the application in use and consequently, there is some debate on how best to define the functional lifetime of a network. In an ad hoc sensor network with a non-hierarchical topological organization, one may assume an application-dependent parameter, $c$, to define the functional lifetime as the length of time the network topology possesses at least one connected component with $n - c$ or more nodes (where $n$ is the total number of nodes in the network). When $c = 0$, the functional lifetime of the network is the length of time $G(t)$ is a connected graph, which is until any node is disconnected or runs out of energy.

We find that a definition of functional lifetime using $c = 0$ is a more versatile one for two reasons: firstly, on any application, there may be some crucial nodes which, when they die, can disable the functionality of the network; secondly, a definition based on the $c = 0$ case can form the foundation of greedy algorithms designed to extend functional lifetime for $c > 0$.

In this paper, therefore, we consider the functional life of the network to have ended when one of the following two cases occurs:

- Case 1: A node reduces its current transmission power in order to save energy, but becomes unable to reach certain nodes and, consequently, loses connection from part of the network.
- Case 2: A node runs out of energy, thus getting disconnected from the rest of the network.

If Case 1 happens, the communication links whose removal caused the network to become disconnected can be restored back into the network to restore the functional life of the network. On the other hand, if Case 2 happens, the network’s functional lifetime cannot be extended in any way. Therefore, to improve the lifetime of the network, (i) Case 1 should be avoided by always ensuring connectivity in the assignment of power levels to the nodes, and (ii) Case 2 should be pushed as far into the future as possible by reducing the rate of energy consumption at the node that is estimated to have the smallest remaining lifetime. The problem can now be defined as one of periodically reassigning the power at which each node makes its transmissions so that the first occurrence of either Case 1 or Case 2 is pushed as far ahead in time as possible.

B. Contributions and Organization

Section II reviews the related work on approaches that have been employed to increase a wireless sensor network’s lifetime through topology control. Section III analyzes the rationale behind the approach used in this paper and presents a few definitions and lays out the foundational concepts for the game-theoretic approach used in this paper. Section IV proves the existence of a Nash equilibrium for the ordinal potential game used to map our problem. Our proof is based
on showing that the difference in individual payoffs for each node from unilaterally changing its strategy and the difference in values of the global potential function have the same sign.

The pseudo-code for the Cooperative Topology Control with Adaptation (CTCA) algorithm is presented in Section V. A simulation-based evaluation of its performance and a comparative analysis with other topology control algorithms are described in Section VI. Our results show that the CTCA algorithm extends the life of a network by more than 50% compared to the best previously-known algorithm. This section also compares the topology delivered by the distributed CTCA algorithm to the optimal solution obtained using a centralized algorithm. We also study the dependence of the performance of CTCA in relation to the optimal on the communication ranges of nodes and on the node density.

Section VII concludes the paper.

II. RELATED WORK

The task of extending the life of a wireless sensor network can be tackled through multiple complementary ways involving routing protocols, medium access strategies or any of several other protocols that facilitate network operations. In this section, we will discuss only the approaches most related to this paper; that is, approaches based on changing the topology of the network by individual nodes changing the power levels at which they make their transmissions while preserving network connectivity.

Traditional topology control algorithms such as Small Minimum-Energy Communication Network (SMECN) [13], Minimum Spanning Tree (MST) [14], DRNG [1], DLSS [1] and STC [2] usually start the topology control process with each node transmitting at its maximum transmission power to discover all of its neighbors. Local neighborhood and power-level information is next exchanged between neighbors. The minimum transmission power of each node such that the graph is still connected is later computed at each node without further communication between nodes. The Weighted Dynamic Topology Control (WDTC) [15] algorithm improves upon the work of MST, and considers the remaining energy of each node in addition to the energy cost of communication across each pair of nodes. The algorithm, however, forces bidirectional communication between each pair of nodes and, in addition, requires periodic communication by each node at its maximum possible power level. Other related algorithms seek to offer a robust topology where the graph can stand multiple channel failures; for example, a $k$-connected graph is sought in [16], [17] and a two-tiered network in [18].

Other topology control algorithms may require communication between nodes throughout the topology control process. One typical example is the work described in [19], which is based on a selfish game on network connectivity to help reduce the transmission power on each node. By offering a utility function which indicates a high profit if the node’s transmission power is small and a low profit if the node’s transmission power is large, each node selfishly reduces its transmission power to maximize its profit. On the other hand, if the node has reduced its transmission power to such an extent that the graph becomes disconnected, the profit of each node becomes 0. This algorithm was later improved in [3], where the requirement of global information (to establish connectivity) is eliminated and a distributed topology control algorithm is proposed. Among the first works on using game theory in topology control problems is [20] which gives tight bounds on worst-case Nash equilibria for a game in which the network is required to preserve connectivity. However, this study only considers selfish nodes which try to
minimize their energy consumption without considering potential sacrifices nodes can make (by expending more energy) to extend a network’s lifetime.

Another class of topology control algorithms is represented by the work reported in [21], where the authors provide a decentralized static complete-information game for power scheduling, considering both frame success rate and connectivity. Yet other approaches to increasing the lifetime of a wireless sensor network include grouping nodes into clusters to create a communication hierarchy in which nodes in a cluster communicate only with their cluster head and only cluster heads are allowed to communicate with other cluster heads or the sink node [22]–[27]. In the work of [28], the authors tried to assign sensor nodes with different initial energy levels so that sensor nodes with high traffic load will be assigned more energy than those with smaller loads. By doing so, with the same amount of overall energy, the network’s lifetime may be extended.

If the network’s lifetime is measured in terms of how many transmissions can be made before the sensor nodes run out of energy, then maximizing the network’s lifetime can be interpreted as maximizing the throughput of the network. In the work of [29], the authors studied the relationship between throughput of the network and its corresponding lifetime under an SINR model. But they focus on a specific network setting where sensor nodes’ neighbors and the communication links are predetermined and the topology of the network remains constant throughout the network’s lifetime.

A survey of topology control algorithms can be found in [30], [31] and a survey of the applications of game theory in wireless sensor networks can be found in [32], [33].

The CTCA algorithm proposed in this paper is the first to use a game-theoretic approach that also adapts to changes in the remaining energy levels of nodes and which allows co-operative behavior amongst nodes. As will be discussed in the following sections, these features allow it to extend the life of a network by more than 50% compared to the best previously-known algorithm.

III. DEFINITIONS AND PRELIMINARIES

In this section, we define terms and concepts that will enable us to specify the localized goals that each node should pursue in order to achieve the global goal of increased lifetime for the network.

Let \( W_i(t) \) denote the amount of energy remaining at node \( N_i \) at time \( t \). Let \( p_i(t) \) denote the power at which node \( N_i \) makes its transmissions at time \( t \). As an estimate of the additional length of time before a node runs out of energy, we define the estimated lifetime of node \( N_i \) at time \( t \), denoted by \( L_i(p_i(t), t) \), as the ratio between the amount of remaining energy on the node at time \( t \) and the power at which it makes its transmissions at time \( t \). That is, \( L_i(p_i(t), t) = W_i(t)/p_i(t) \). Note that the estimated lifetime may or may not accurately capture the actual remaining lifetime of a node (because its transmission powers may change later or its energy reserves may deplete slower/faster than estimated.) When the context is clear, for brevity, we refer to the estimated lifetime as just the lifetime.

In a system in which the rate of energy consumption is largely balanced across the nodes (which is the goal of this paper as a means to improve network lifetime), the node with the smallest estimated lifetime is likely the one that determines the network’s lifetime. We consider the estimated lifetime of a network as the estimated lifetime of the node with the smallest estimated lifetime. If \( N_i \) is the node with the smallest estimated lifetime within the network,
(a) Node $N_3$’s initial state, where node $N_1$ is not capable of reducing its transmission power without disconnecting the graph.

(b) Situation 1: Node $N_3$ reduces its transmission power to its potential transmission power, but $N_1$’s lifetime cannot be improved.

(c) Situation 2: Node $N_3$ increases its transmission power to $p(N_3, N_2)$. Now, node $N_1$ is able to reduce its power without disconnecting the graph.

(d) Node $N_1$ updates its current transmission power to its potential transmission power, and extends its lifetime and of the network.

Fig. 2: An example illustrating how a node should choose its power level so as to increase the network’s lifetime.

then it may be possible to improve the network’s lifetime by improving $N_i$’s estimated lifetime. Fig. 1 shows an example where node $A$ is able to reduce its transmission power with help from node $C$, thus increasing its estimated lifetime and likely the lifetime of the network.

We further illustrate the definitions in this section using the topology shown in Fig. 2. Suppose at time $t$, the topology of the network is as shown in Fig. 2(a). Denote by $p(N_i, N_j)$ the minimum transmission power at which nodes $N_i$ and $N_j$ have to transmit to reach each other. We refer to the set of transmission powers that a node may switch to at time $t$ as its available transmission powers at time $t$. Then, according to the topology given by Fig. 2(a), node $N_3$’s available transmission powers are: $p(N_3, N_1)$, $p(N_3, N_2)$ and $p(N_3, N_4)$, while its current transmission power is $p(N_3, N_1)$ (note that there is no need for node $N_3$ to transmit at any power level other than the ones in this available set).

Let $P$ denote a mapping of nodes in the network to power levels. For example, in Fig. 2(a), the mapping implemented is given by $P = \{ N_1 \rightarrow p(N_1, N_2), N_2 \rightarrow p(N_2, N_3), N_3 \rightarrow p(N_3, N_1), N_4 \rightarrow p(N_4, N_3) \}$. Since node $N_3$ has the potential to transmit at power $p(N_3, N_4)$ while still keeping the graph connected, we refer to power $p(N_3, N_4)$ as node $N_3$’s potential transmission power under this node-to-power mapping $P$.

In general, the potential transmission power of a node is the smallest available transmission power that the node can use such that the graph is still connected while the power levels at all other nodes remain the same. Let $p'_i(P)$ denote the potential transmission power of node $N_i$ under the node-to-power mapping $P$. Note that a node’s potential transmission power is no greater than its current transmission power provided that the network is currently connected. That is, $p'_i(P) \leq p_i(t)$ if $G(t)$ is a connected graph and $P$ is the node-to-power mapping implemented
at time $t$.

Transmitting at the potential transmission power as defined above can increase the lifetime of a node beyond its estimated lifetime and, consequently, of the network. To estimate the best lifetime a node can achieve without changing the transmission powers of other nodes and without disconnecting the network, we define the potential lifetime of a node as the ratio between the node’s current remaining energy and its potential transmission power. Let $L_i' (P,t)$ denote the potential lifetime of node $N_i$ at time $t$ under the node-to-power mapping $P$. Let $p'_i (P)$ denote its potential transmission power under the node-to-power mapping $P$. Then, $L_i' (P,t) = W_i(t)/p'_i (P)$.

If $p'_i (P) \leq p_i (t)$, then $L_i(p_i(t),t) \leq L_i'(P,t)$. Therefore, to increase its estimated lifetime, a node should always try to change its current transmission power to its potential transmission power, if they are not the same. Figs. 2(a) and 2(b) illustrate such a process for node $N_3$, where it changes its current transmission power from $p_i$ to higher power level, if its estimated lifetime $L_i(p_i(t),t)$ is less than that of node $N_1$. Then, $L_i(p_i(t),t) = L_i'(P,t)$.

In Fig. 2(a), suppose node $N_1$ is the node that has the smallest estimated lifetime within the network. Then, $N_1$’s estimated lifetime has to be improved in order to improve the network’s lifetime. In Fig. 2(b), node $N_3$ reduces its transmission power but this does not improve the potential lifetime of node $N_1$. This implies that $N_1$’s estimated lifetime cannot be improved by node $N_3$ reducing its transmission power. On the other hand, if node $N_3$ chooses to transmit at a higher power level, $p(N_3, N_2)$, as illustrated in Fig. 2(c), then node $N_1$’s potential transmission power reduces to $p(N_1, N_3)$. Now, node $N_1$ can reduce its transmission power to the new potential transmission power as illustrated in Fig. 2(d), thus improving the network’s estimated lifetime.

Let $R_i(t)$ denote the set of nodes $N_i$ can reach at time $t$; i.e., $R_i(t) = \{N_j | p(N_i, N_j) \leq p_i(t)\}$. Let $I_i(t)$ denote the set of nodes that can reach node $N_i$ at time $t$, i.e., $I_i(t) = \{N_j | p(N_j, N_i) \leq p_j(t)\}$. Then for any $N_j \in I_i(t)$, we have $N_i \in R_j(t)$. We refer to the nodes in the set $R_i(t)$ as the reachable neighbors of node $N_i$ and the nodes in the set $I_i(t)$ as the reverse-link neighbors of $N_i$. For example, in Fig. 2(a), $N_1$’s reachable neighbors are $N_2$, $N_3$ and $N_4$ while $N_2$ has only one reachable neighbor $N_3$. Also, $N_1$ is a reverse-link neighbor of $N_2$, $N_3$ and $N_4$. Let $H_i(t) = R_i(t) \cup N_i$ and let $O_i(t) = I_i(t) \cup N_i$.

In Fig. 2(a), note that $N_1$’s potential transmission power will not be reduced unless either $N_3$ or $N_4$ has increased its transmission power to be able to reach $N_2$. In general, $N_i$’s reachable neighbors are the only nodes who can help $N_i$ reduce its potential transmission power; and only the nodes who are $N_i$’s reverse-link neighbors may benefit from $N_i$’s increase in its transmission power level.

A question worth answering at this point is about what might be the relationship between improving the network’s lifetime and improving node $N_1$’s estimated and potential lifetime. As we have stated previously, the network’s estimated lifetime is dependent upon the node with the smallest estimated lifetime. If we compare the power mapping illustrated in Fig. 2(a) and Fig. 2(d), we can see that node $N_3$ not only has sacrificed its chance to improve its lifetime but also has sacrificed its own estimated lifetime (increases its current transmission power from $p(N_3, N_1)$ to $p(N_3, N_2)$) in order to help improve node $N_1$’s potential lifetime which is the smallest in Fig. 2(a). When the topology is as illustrated in Fig. 2(d), suppose node $N_3$ is the node with the smallest estimated lifetime. If its estimated lifetime in the topology illustrated in Fig. 2(d) is less than that of node $N_1$’s in the topology illustrated in Fig. 2(a), the network’s estimated lifetime
in fact reduces rather than increases. In such a case, node $N_3$ should not choose to increase its transmission power to help improve $N_1$'s lifetime, but should instead focus on improving its own estimated lifetime.

For any node $N_i$ within the graph, its estimated lifetime will not improve unless it switches to its potential transmission power. That is to say, no node can help node $N_i$ with its estimated lifetime except for node $N_i$ itself; but as illustrated in Fig. 2, another node may help with $N_i$'s potential lifetime. In our example, node $N_3$ helps node $N_1$ improve its potential lifetime which eventually allows $N_1$ to increase its estimated lifetime by changing its transmission power to the potential transmission power.

The above discussion leads to the following primary and secondary goals for each node in order to improve the network’s lifetime while also conserving energy as much as possible:

- **Primary goal:** Let $m(i)$ denote the node with the smallest potential lifetime amongst the reverse-link neighbors of node $N_i$. Let $q$ denote the potential lifetime of node $m(i)$. The primary goal of node $N_i$ should be to increase the potential lifetime of node $m(i)$ above $q$ while making sure that its own estimated lifetime does not reduce below $q$.

- **Secondary goal:** The secondary goal of node $N_i$ should be to increase its own estimated lifetime.

The secondary goal is achieved once a node adopts its potential transmission power as its current transmission power (in this situation, its potential lifetime then becomes its estimated lifetime). Note that, if the primary and the secondary goals conflict, a node should always choose to meet the primary goal. This leads us into the design of the cooperative game that each node can play with its reachable neighbors and its reverse-link neighbors.

One may argue that since the ultimate goal is to extend functional lifetime (the primary goal), the secondary goal should be unnecessary. However, there are three reasons why we include the secondary goal in our design of the algorithm.

Firstly, for any realistic algorithm, it is not possible to precisely predict the network lifetime since it is not possible to predict with precision the future traffic load experienced by any given sensor node. As a result, an algorithm can only seek to maximize the estimated functional lifetime of a node and not the actual functional lifetime. Therefore, the secondary goal helps each sensor node save energy as much as possible so that when the estimated functional lifetime deviates from the actual functional lifetime, the node that actually determines the network’s functional lifetime (the one that dies earliest) does not waste any energy by transmitting at a power larger than what was necessary to stay connected. Secondly, as discussed in Section [A], we define the functional lifetime assuming $c = 0$. However, if the functional lifetime for some application is defined assuming that the network function survives until $c > 0$ nodes are disconnected, a good heuristic for extending the functional lifetime emerges if we implement the secondary goal in conjunction with the primary goal assuming $c = 0$ on the surviving largest connected component every time a node dies. Thirdly, reducing the transmission power of a sensor node may help reduce the interference among transmissions, reducing retransmissions and also helping extend the functional lifetime of the network.

In the above discussion, without loss of generality, we assume that there is only one node $m(i)$ with the smallest potential lifetime amongst reverse-link neighbors of $N_i$. If there is a tie with two nodes having identical potential lifetimes, one can always break the tie in the algorithm using a consistently applied second criterion such as the node id.
TABLE I: A glossary of selected terms. If the time \( t \) specified within each notation is clear within the context, it is omitted for purposes of brevity. In the glossary, however, we list each notation in both forms (with and without \( t \)).

| Notation | Definition |
|----------|------------|
| \( N \)  | The set of \( n \) nodes within the network. \( N = \{N_1, N_2, \ldots, N_n\} \). |
| \( G(t) \) | Graph representing the network topology at time \( t \). \( G(t) = (N, E(t)) \). |
| \( p_i(t) \) or \( p_i \) | Node \( N_i \)'s transmission power at time \( t \). |
| \( P \) | A mapping, \( P : N \rightarrow \mathbb{R} \), of the nodes in the network to transmission power levels. For example, the mapping actually implemented at time \( t \) is \( \{N_1 \rightarrow p_1(t), N_2 \rightarrow p_2(t), \ldots\} \). |
| \( P_{-,i} \) | A mapping of all the nodes except \( N_i \) to transmission power levels. |
| \( p(N_i, N_j) \) | The minimum transmission power needed for node \( N_i \)'s transmissions to reach node \( N_j \). We assume \( p(N_i, N_j) = p(N_j, N_i) \). |
| \( R_i(t) \) or \( R_i \) | The set of nodes reachable by \( N_i \)'s transmissions at time \( t \). \( R_i(t) = \{N_j \mid (N_i, N_j) \in E(t)\} \). This set of reachable neighbors is also called the reachable neighborhood of \( N_i \). |
| \( H_i(t) \) or \( H_i \) | The set including \( N_i \) and its reachable neighbors. \( H_i(t) = R_i(t) \cup N_i \). |
| \( I_i(t) \) or \( I_i \) | The set of nodes at time \( t \) which can directly reach \( N_i \) with their transmissions. \( I_i(t) = \{N_j \mid (N_i, N_j) \in E(t)\} \). This set of reverse-link neighbors of \( N_i \) is also called reverse-link neighborhood of \( N_i \). |
| \( O_i(t) \) or \( O_i \) | The set including \( N_i \) and its reverse-link neighbors. \( O_i(t) = I_i(t) \cup N_i \). |
| \( P_i \) | A mapping, \( P_i : H_i \rightarrow \mathbb{R} \), of \( N_i \) and its reachable neighbors to transmission power levels. |
| \( P_{i,-j} \) | A mapping of \( N_i \) and its reachable neighbors except \( N_j \) to transmission power levels. |
| \( p'_i(P_i) \) | \( N_i \)'s potential transmission power when its and its reachable neighbors’ power levels are given by the mapping \( P_i \). |
| \( A_i \) | \( N_i \)'s possible transmission power choices. |
| \( W_i(t) \) or \( W_i \) | \( N_i \)'s remaining energy at time \( t \). |
| \( L_i(a_i, t) \) | \( N_i \)'s estimated lifetime (remaining) at time \( t \) if set to transmit at power \( a_i \). \( L_i(a_i, t) = W_i(t)/a_i \). |
| \( L'_i(P_i, t) \) | \( N_i \)'s potential lifetime at time \( t \) when its and its reachable neighbors’ transmission powers are given by the mapping \( P_i \). \( L'_i(P_i, t) = W_i(t)/p'_i(P_i) \). |
| \( m(i, t) \) or \( m(i) \) | The node with the smallest potential lifetime among \( N_i \)'s reverse-link neighbors at time \( t \). \( m(i, t) = \arg \min_{N_j \in I_i(t)} (L'_j(P_j, t)) \). |

IV. THE ORDINAL POTENTIAL GAME

In this section, we present the notation and the utility function governing the ordinal potential game into which we map the problem of extending the lifetime of the network.

A. Notation

Table [IV-A] presents a glossary of terms used in this section. In this paper, for brevity and clarity, we sometimes omit the time index \( t \) whenever the corresponding instant of time is clear from the context.

10
Suppose that at time \( t = 0 \), each node is transmitting at its maximum transmission power \( p_{\text{max}} \). Then, the set of nodes that includes \( N_i \)’s reachable neighbors is \( R_i(0) = \{ N_j \mid p(N_i, N_j) \leq p_{\text{max}} \} \). Let \( n_i \) denote the size of the set \( R_i(0) \). Therefore, we define the available transmission powers for node \( N_i \) as \( A_i = \{ p^i_1, p^i_2, \ldots, p^i_{n_i} \} \) where, for any \( p^i_k \in A_i \), there exists a node \( N_j \in R_i(0) \) such that \( p^i_k \) is the minimum transmission power required for \( N_i \) to reach \( N_j \). Note that a node does not need to transmit at power levels other than the ones needed to reach other nodes within its maximum range.

Let \( P \) denote a mapping of the nodes in the network to transmission power levels. The mapping implemented at time \( t \) is \( P = \{ N_i \rightarrow p_i(t), N_2 \rightarrow p_2(t), \ldots \} \), where \( p_i(t) \) is the power at which node \( N_i \) is set to make transmissions at time \( t \). We write \( P = \{ f_i, P_{-i} \} \) where \( f_i \) is a mapping of node \( N_i \) to a certain power level and \( P_{-i} \) is a mapping of all other nodes in the network to power levels.

Let \( P_i \) denote a mapping, \( P_i : H_i \rightarrow \mathbb{R} \), of \( N_i \) and its reachable neighbors to power levels. We write \( P_i = \{ f_j, P_{i,-j} \} \) where \( f_j \) is a mapping of node \( N_j \in H_i \) to a certain power level and \( P_{i,-j} \) is a mapping of all other nodes in \( H_i \) to power levels.

Let \( L_i(a_i, t) \) denote the estimated lifetime of node \( N_i \) at time \( t \) if set to transmit at power level \( a_i \). Per the definition of estimated lifetime, \( L_i(a_i, t) = W_i(t)/a_i \). If \( a_i = p'_i(P_i) \), then \( L_i(a_i, t) = L'_i(P_i, t) \).

Denote by \( m(i, t) \), or \( m(i) \) for brevity, the node in \( I_i(t) \) with the smallest potential lifetime. We define the potential lifetime of \( N_i \)’s reverse-link neighborhood at time \( t \) as the potential lifetime of node \( m(i) \) at time \( t \), i.e., \( L'_m(i)(P_{m(i)}, t) \).

### B. The utility function

In the following, we present and justify the utility function governing the ordinal potential game upon which our topology control algorithm is based.

As stated in Section III, the primary goal for each sensor node is to improve the potential lifetime of its reverse-link neighborhood without also causing a reduction in the network’s estimated lifetime. While prioritizing the primary goal, the secondary goal of the sensor node is to improve its own estimated lifetime. Both of these goals are captured in the utility function presented in this section.

Let \( a_i \in A_i \) denote a power level that is available to node \( N_i \) at time \( t \). Define the primary utility function (corresponding to the primary goal described in Section III) for node \( N_i \) with power level \( a_i \) at time \( t \) as:

\[
    u^X_i(a_i, t) = \min \left( \min_{N_j \in I_i(t)} L'_j(\{ N_i \rightarrow a_i, P_{j,-i} \}, t), L_i(a_i, t) \right) = \min \left( L'_m(i)(\{ N_i \rightarrow a_i, P_{m(i), -i} \}, t), L_i(a_i, t) \right)
\]

This is the minimum of the estimated lifetime of node \( N_i \) at power level \( a_i \) and the potential lifetime of the node whose value is the minimum amongst its reverse-link neighbors. Maximizing this is the primary goal as explained in Section III.

Define the secondary utility function for node \( N_i \) with power level \( a_i \) at time \( t \) as:

\[
    u^Y_i(a_i, t) = L_i(a_i, t)
\]
This is the estimated lifetime of node $N_i$ when transmitting at power level $a_i$ at time $t$. Maximizing this is the secondary goal as also explained in Section III.

Bearing in mind the two goals, primary and secondary, for each node, we define the utility function $u_i$ for node $N_i$ with power level $a_i$ at time $t$ as:

$$u_i(a_i, t) = c_i(a_i, t) \left[ u^X_i(a_i, t) + \ell_i(a_i, t) u^Y_i(a_i, t) \right]$$

$$= c_i(a_i, t) \min \left( L_{m(i)}' \left( \{ N_i \rightarrow a_i, \mathbf{P}_{m(i) - i} \} \right), L_i(a_i, t) \right) + c_i(a_i, t) \ell_i(a_i, t) L_i(a_i, t)$$

(3)

where $c_i(a_i, t)$ and $\ell_i(a_i, t)$ are defined in the following paragraphs.

The term $c_i(a_i, t)$ in Eqn. (3) is a binary function indicating whether node $N_i$, when set to transmit at power $a_i$, is connected to every node $N_j \in R_i(t)$. More specifically,

$$c_i(a_i, t) = \begin{cases} 1, & \text{if a path exists from } N_i \text{ to each } N_j \in R_i(t) \\ 0, & \text{otherwise} \end{cases}$$

If node $N_i$ has lost connectivity with a certain node $N_j$ by transmitting at power $a_i$ at time $t$, i.e.,

$c_i(a_i, t) = 0$, then, the network’s connectivity is lost and by Case 1 in Section I-A, the network’s life has ended. This should be reflected in node $N_i$’s own utility function, and thus, $u_i(a_i, t) = 0$ when $c_i(a_i, t) = 0$. Note that checking for the existence of a path to every $N_j \in R_i(t)$ is a localized function and does not require global or centralized knowledge.

The term $\ell_i(a_i, t)$ in Eqn. (3) is a binary function indicating whether the node’s own estimated lifetime should be considered when calculating its utility at power level $a_i$. In the following, we will describe the conditions under which $\ell_i(a_i, t)$ takes on the values of either 0 or 1.

As discussed in Section III, improving its own lifetime is only the secondary goal for every sensor node. When the primary and the secondary goals of a node are in conflict, the secondary goal of improving its own lifetime should yield to the primary goal. Therefore, for $a_i$ that leads to this situation, $\ell_i(a_i, t) = 0$ indicating that the secondary goal of node $N_i$ yields to the primary goal. On the other hand, for the power level $a_i$ at which node $N_i$ is able to achieve its primary goal without conflict with the secondary goal, $\ell_i(a_i, t)$ should take on the value of 1 and node $N_i$ should now focus on its secondary goal as well. In cases where node $N_i$ is not able to achieve the primary goal at whichever power it is transmitting, it should then focus on improving its own estimated lifetime and therefore, the function $\ell_i(a_i, t)$ then takes on the value of 1 for every $a_i$ selected. The following paragraphs present a formal definition of function $\ell_i(a_i, t)$.

Suppose at power level $a_i$, node $N_i$ is able to help node $m(i, t)$ reduce its potential transmission power, and its own estimated lifetime at power $a_i$ is larger than its reverse-link neighborhood’s previous potential lifetime. Under such a circumstance, we refer to power level $a_i$ as a preferred power level of node $N_i$. Note that, for a node-to-power mapping $\mathbf{P}$, there may exist several preferred power levels for node $N_i$. We denote the set of preferred power levels for node $N_i$ under the node-to-power mapping $\mathbf{P}$ as $K_i(\mathbf{P}_{-i})$. For any power level $a_i \in K_i(\mathbf{P}_{-i})$, node $N_i$’s reverse-link neighborhood’s potential lifetime is extended by node $N_i$ transmitting at power $a_i$ and node $N_i$’s lifetime at power level $a_i$ exceeds its previous reverse-link neighborhood’s potential lifetime. Therefore, at a power level $a_i \in K_i(\mathbf{P}_{-i})$, the primary goal for node $N_i$ is met and node $N_i$ should focus on optimizing for its own estimated lifetime (the secondary goal) through the utility function. Therefore, we can conclude that $\ell_i(a_i, t) = 1$ in such a case.
If there exists no preferred power level \( a_i \) at which \( N_i \) can transmit to increase its primary utility function (indicated by \( K_i(P_{-i}) = \emptyset \)), then also node \( N_i \) should focus on improving its own estimated lifetime through the utility function. In this case, \( N_i \’s lifetime should be still be relevant in the local utility function \( u_i(a_i, t) \) and so, \( \ell_i(a_i, t) = 1 \) for any value of \( a_i \).

If neither of the above two cases is valid, then transmitting at power level \( a_i \) causes a conflict between \( N_i \’s primary and secondary goals. In this case, node \( N_i \) should focus on its primary goal only and therefore, \( \ell_i(a_i, t) = 0. \)

Based on the above reasoning, \( \ell_i(a_i, t) \) is defined as:
\[
\ell_i(a_i, t) = \begin{cases} 
1, & \text{if } K_i(P_{-i}) = \emptyset, \text{ or } a_i \in K_i(P_{-i}) \\
0, & \text{otherwise}
\end{cases}
\]

C. The ordinal potential game

We are now ready to describe the strategic game \( \Gamma = \langle N, A, U \rangle \) as having the following three components:
- Player set \( N \): \( N_i \in N = \{N_1, N_2, \ldots, N_n\} \) where \( n \) is the number of nodes in the network.
- Action set \( A \): \( a \in A = \Pi_{i=1}^n A_i \) is the space of all action vectors, where each component \( A_i \) represents the set of available power levels at which \( N_i \) may transmit.
- Utility function set \( U \): For each player \( N_i \), utility function \( u_i : A \rightarrow \mathbb{R} \) as given by Eqn. \( (3) \) which models the node’s preferences for its available power level choices. The vector of these utility functions is \( U : A \rightarrow \mathbb{R}^n \).

**Theorem 4.1:** The game \( \Gamma = \langle N, A, U \rangle \) is an ordinal potential game and its ordinal potential function is given by
\[
\Phi(P, t) = C(P) \min_{N_i \in N} L'_i(P_i, t)
\]
where \( C(P) \) is the binary connectivity function indicating whether the graph is connected with node-to-power mapping \( P \), i.e,
\[
C(P) = \begin{cases} 
1, & \text{if the graph is connected} \\
0, & \text{otherwise}
\end{cases}
\]

**Proof:** We will prove this by applying the definition of an ordinal potential game and proving that, at any time instant \( t \), the difference in individual utilities for each node from unilaterally changing its strategy and the difference in values of the global potential function have the same sign \([9], [34]\). Denote the mapping of nodes to power levels as follows: when \( N_i \) is transmitting at power level \( a_i \) as \( P = \{N_i \rightarrow a_i, P_{-i}\} \) and when \( N_i \) is transmitting at power level \( a'_i \) as \( P' = \{N_i \rightarrow a'_i, P_{-i}\} \). First, for the difference in an individual node’s utilities, we have:
\[
\Delta u_i(t) = u_i(P, t) - u_i(P', t)
\]
Omitting \( t \) for brevity, we can rewrite this equation as:
\[
\Delta u_i = u_i(P) - u_i(P')
\]
\[
= c_i(a_i)L_i(a_i)\ell_i(a_i) + c_i(a_i) \min \{L'_m(i)(P), L_i(a_i)\}
\]
\[
- c_i(a'_i)L_i(a'_i)\ell_i(a'_i) - c_i(a'_i) \min \{L'_m(i)(P'), L_i(a'_i)\}
\]
Note that, with \( N_i \)'s power level being either \( a_i \) or \( a_i' \), the power levels for the rest of the nodes within the network remain the same. Since at any time instant \( t \), \( N_j \in O_i(t) \) are the only nodes whose potential lifetime may be affected by \( N_i \)'s power level, we can thus conclude that for node \( N_j \notin O_i(t) \), \( L_j'(P, t) = L_j'(P', t) \).

Now, the difference in the values of the global potential function, \( \Delta \Phi(t) \), is:

\[
\Delta \Phi(t) = C(P) \min_{N_i \in N} L_i'(P, t) - C(P') \min_{N_i \in N} L_i'(P', t)
\]

Since this equation holds for any value of \( t \), we can omit \( t \) to simplify the notation.

\[
\Delta \Phi = C(P) \min_{N_i \in N} L_i'(P) - C(P') \min_{N_i \in N} L_i'(P')
\]

\[
= C(P) \min \left\{ \min_{N_j \in O_i} L_j'(P), \min_{N_k \notin O_i} L_k'(P) \right\} - C(P') \min \left\{ \min_{N_j \in O_i} L_j'(P'), \min_{N_k \notin O_i} L_k'(P') \right\}
\]

\[
= C(P) \min \{ T_i(P), T_{-i}(P) \} - C(P') \min \{ T_i(P'), T_{-i}(P') \}
\]

where \( T_i(P) = \min_{N_j \in O_i} L_j'(P) \) is the smallest potential lifetime amongst node \( N_i \) and its reverse-link neighborhood when the node-to-power mapping is \( P \). Recall that \( O_i = N_i \cup I_i \) and therefore:

\[
T_i(P) = \min_{N_j \in I_i} \{ L_i'(P), \min_{N_j \in I_i} L_j'(P) \}
\]

\[
= \min \{ L_i'(P), L_{m(i)}'(P) \} \quad (5)
\]

\( T_{-i}(P') \) is similarly defined. \( T_{-i}(P) \) and \( T_{-i}(P') \) are also defined similarly as \( \min_{N_j \notin O_i} L_k'(P) \) and \( \min_{N_j \notin O_i} L_k'(P') \), respectively. Since nodes within \( O_i \) are the only nodes whose potential lifetime may be influenced by node \( N_i \)'s change in its transmission power, we can therefore conclude that \( T_{-i}(P) = T_{-i}(P') \).

Without loss of generality, we assume that \( a_i > a_i' \), indicating that if \( C(P') = 1 \), then \( C(P) = 1 \). We can also conclude that \( L_i(a_i) < L_i(a_i') \). According to the definition of \( C(P) \), if \( a_i = 0 \), then \( C(P) = 0 \). The possible cases of \( c_i(a_i, t) \) and \( c_i(a_i', t) \) are (omitting \( t \) for brevity):

- Case 1: \( c_i(a_i) = c_i(a_i') = 0 \) \( \Rightarrow \) \( C(P) = C(P') = 0 \)
- Case 2: \( c_i(a_i) = 1, c_i(a_i') = 0 \) \( \Rightarrow \) \( C(P') = 0 \)
- Case 3: \( c_i(a_i) = c_i(a_i') = 1 \)

In Cases 1 and 2, the network is not connected with \( a_i \) or \( a_i' \) or both. In these cases, it is easy to prove that \( \Delta u_i \) and \( \Delta \Phi \) have the same sign. We consider Case 3 in detail in the following.

In Case 3, the local graph within \( N_i \)'s range is connected whether \( N_i \)'s power level is \( a_i \) or \( a_i' \). Since all other nodes except \( N_i \)'s power levels remain the same at time \( t \), we can conclude that \( C(P) = C(P') \). This leads us to two situations: in one, \( C(P) = C(P') = 0 \), i.e., the full graph \( G \) is not connected because of some node located outside of \( N_i \)'s range, and in the other, \( C(P) = C(P') = 1 \), i.e., the full graph \( G \) is connected. In the case the graph is not connected, \( C(P) = C(P') = 0 \) and, therefore, \( \Delta \Phi_i = 0 \). Thus, we can conclude that \( \Delta u_i \) and \( \Delta \Phi_i \) have the same sign. In the following, we now focus on the situation in which the full graph \( G \) is connected.

The Case 3 situation in which the graph is connected, i.e., \( C(P) = C(P') = 1 \), can be further categorized into four sub-cases:
• Sub-case (3a):
  \( \min\{T_i(P), T_{-i}(P)\} = T_i(P) \), and \( \min\{T_i(P'), T_{-i}(P')\} = T_i(P') \).

• Sub-case (3b):
  \( \min\{T_i(P), T_{-i}(P)\} = T_{-i}(P) \), and \( \min\{T_i(P'), T_{-i}(P')\} = T_i(P') \).

• Sub-case (3c):
  \( \min\{T_i(P), T_{-i}(P)\} = T_i(P) \), and \( \min\{T_i(P'), T_{-i}(P')\} = T_{-i}(P') \).

• Sub-case (3d):
  \( \min\{T_i(P), T_{-i}(P)\} = T_{-i}(P) \), and \( \min\{T_i(P'), T_{-i}(P')\} = T_{-i}(P') \).

Case (3a): In this case, whether \( N_i \)'s power level is \( a_i \) or \( a'_i \), the node with the smallest potential lifetime lies either within \( N_i \)'s reverse-link neighborhood or is node \( N_i \) itself. Since \( T_i(P) = \min\{L_{m(i)}'(P), L_i'(P)\} \) and \( T_i(P') = \min\{L_{m(i)}'(P'), L_i'(P')\} \), we can conclude that:

\[
\Delta \Phi = T_i(P) - T_i(P') = \min\{L_{m(i)}'(P), L_i'(P)\} - \min\{L_{m(i)}'(P'), L_i'(P')\} \tag{6}
\]

\[
\Delta u_i = L_i(a_i) - L_i(a_i') - L_i(a_i') + L_i(a_i) + \min\{L_{m(i)}'(P), L_i(a_i)\} - \min\{L_{m(i)}'(P'), L_i(a_i')\} \tag{7}
\]

Since a node’s potential transmission power is no larger than its current transmission power, we can conclude that \( p_i(P) \leq a_i \) and \( p_i(P') \leq a_i' \). Also, \( a_i \) is at least one power level larger than \( a_i' \) and, therefore, \( p_i(P') \leq a_i' \leq p_i(P) \leq a_i \). We conclude:

\[
L_i'(P') \geq L_i(a_i') \geq L_i'(P) \geq L_i(a_i) \tag{8}
\]

Now, there are four sub-sub-cases based on the values of \( \ell_i(a_i) \) and \( \ell_i(a_i') \):

• Case (3a-i): \( \ell_i(a_i) = \ell_i(a_i') = 1 \)
• Case (3a-ii): \( \ell_i(a_i) = \ell_i(a_i') = 0 \)
• Case (3a-iii): \( \ell_i(a_i) = 1 \), and \( \ell_i(a_i') = 0 \)
• Case (3a-iv): \( \ell_i(a_i) = 0 \), and \( \ell_i(a_i') = 1 \)

In the following, we consider each of the above sub-sub-cases.

Case (3a-i): According to the definition of \( \ell_i(a, t) \), either both power levels \( a_i \) and \( a_i' \) can help improve node \( N_i \)'s reverse-link neighborhood’s potential lifetime or neither of them can. Therefore, we have \( L_{m(i)}'(P) = L_{m(i)}'(P') \). Note that, \( L_i'(P) \leq L_i'(P') \). Now, we can rewrite Eqn. (6) as:

\[
\Delta \Phi = \min\{L_{m(i)}'(P), L_i'(P)\} - \min\{L_{m(i)}'(P'), L_i'(P')\} \\
= \min\{L_{m(i)}'(P'), L_i'(P)\} - \min\{L_{m(i)}'(P'), L_i'(P')\} \\
\leq \min\{L_{m(i)}'(P'), L_i'(P')\} - \min\{L_{m(i)}'(P'), L_i'(P')\} \\
= 0
\]

As for \( \Delta u_i \), since \( L_i(a_i) < L_i(a_i') \), and \( L_{m(i)}'(P) = L_{m(i)}'(P') \), we can rewrite Eqn. (7) as:

\[
\Delta u_i = L_i(a_i) - L_i(a_i') + L_i(a_i') - L_i(a_i) + \min\{L_{m(i)}'(P), L_i(a_i)\} - \min\{L_{m(i)}'(P'), L_i(a_i')\} \\
< \min\{L_{m(i)}'(P'), L_i(a_i)\} - \min\{L_{m(i)}'(P'), L_i(a_i')\} \\
\leq \min\{L_{m(i)}'(P'), L_i(a_i')\} - \min\{L_{m(i)}'(P'), L_i(a_i)\} \\
= 0
\]
Therefore, we have $\Delta \Phi \leq 0$ and $\Delta u_i < 0$. Thus, as for Case (3a-i), $\Delta \Phi$ and $\Delta u_i$ share the same sign.  

**Case (3a-ii):** Since $\ell_i(a_i) = \ell_i(a_i') = 0$, it indicates that node $N_i$’s reverse-link neighborhood’s potential lifetime cannot be extended when node $N_i$ is transmitting at either power level $a_i$ or $a_i'$. Thus, we can conclude that $L'_m(i)(P) = L'_m(i)(P')$. Following a similar line of deduction as in Case (3a-i), we can conclude that $\Delta \Phi \leq 0$.

As for $\Delta u_i$, also following the same line of deduction as in Case (3a-i), we have:

$$\Delta u_i = \min \{L'_m(i)(P), L_i(a_i)\} - \min \{L'_m(i)(P'), L_i(a_i')\}$$

$$\leq \min \{L'_m(i)(P'), L_i(a_i)\} - \min \{L'_m(i)(P'), L_i(a_i')\}$$

$$= 0$$

This implies that both $\Delta \Phi$ and $\Delta u_i$ are no larger than 0 and, therefore, share the same sign.

**Case (3a-iii):** The fact that $\ell_i(a_i) = 1$ and $\ell_i(a_i') = 0$ indicates that node $N_i$’s preferred power set is not empty and power level $a_i$ is one of the preferred power levels while power level $a_i'$ is not. Therefore, by node $N_i$ transmitting at power $a_i$, its reverse-link neighborhood’s potential lifetime can be extended. Denote node $N_i$’s reverse-link neighborhood’s previous potential lifetime by $L'_\text{pre}$. We know that $L'_m(i)(P) > L'_\text{pre}$. On the other hand, since $\ell_i(a_i') = 0$ and $a_i > a_i'$, we can conclude that by node $N_i$ transmitting at power level $a_i'$, its reverse-link neighborhood’s potential lifetime cannot be improved. Thus, we have $L'_m(i)(P') = L'_\text{pre} < L'_m(i)(P)$. Also, according to the definition of $\ell_i(a_i)$, we can conclude $L_i(a_i) > L'_\text{pre}$. Together with Eqn. (8), we can conclude that $L'_i(P) \geq L_i(a_i) > L'_m(i)(P')$. Thus, we can rewrite Eqns. (6) and (7) as:

$$\Delta \Phi = \min \{L'_m(i)(P), L'_i(P)\} - \min \{L'_m(i)(P'), L'_i(P')\}$$

$$\geq \min \{L'_m(i)(P'), L'_i(P)\} - L'_m(i)(P')$$

$$> \min \{L'_m(i)(P'), L'_m(i)(P')\} - L'_m(i)(P') = 0$$

$$\Delta u_i = L_i(a_i) - \min \{L'_m(i)(P'), L_i(a_i')\} + \min \{L'_m(i)(P), L_i(a_i)\}$$

$$\geq L_i(a_i) - L'_m(i)(P') + \min \{L'_m(i)(P'), L_i(a_i)\}$$

$$> \min \{L'_m(i)(P'), L_i(a_i)\} > 0$$

Therefore, we have proved that, in Case (3a-iii), both $\Delta \Phi$ and $\Delta u_i$ are positive numbers, and therefore, share the same sign.

**Case (3a-iv):** The fact that $\ell_i(a_i) = 0$ indicates that the preferred power level set $K_i(P_{-i})$ is not empty and power level $a_i$ is not within $K_i(P_{-i})$. On the other hand, since $\ell_i(a_i') = 1$, and $P_{-i} = P_{-i}'$, we can conclude that $a_i'$ is a preferred power level and $L'_\text{pre} < L_i(a_i')$. This also indicates that when transmitting at power level $a_i'$, node $N_i$ serves as a relay node for node $m(i)$ enabling it to reduce its transmission power without disconnecting the network. Therefore, by transmitting at power $a_i > a_i'$, node $N_i$ should also be able to serve as the bridge node for node $m(i)$. Thus, we can conclude that $L'_m(i)(P) = L'_m(i)(P')$. Following similar lines of deduction as in Cases (3a-i) and (3a-ii), we conclude that $\Delta \Phi \leq 0$.

Now, since node $m(i)$’s potential lifetime can be improved by node $N_i$ transmitting at power level $a_i$, therefore, the only reason why $\ell_i(a_i) = 0$ is that by transmitting at this power, node
\( N_i \)’s lifetime at power level \( a_i \) is less than node \( m(i) \)’s previous potential lifetime, i.e., \( L_i(a_i) < L'_i < L_i(a'_i) \). We therefore can rewrite Eqn. 7 as:

\[
\Delta u_i = \min \{ L'_m(i)(P), L_i(a_i) \} - \min \{ L'_m(i)(P'), L_i(a'_i) \} - L_i(a'_i) \\
\leq \min \{ L'_m(i)(P'), L_i(a'_i) \} - \min \{ L'_m(i)(P'), L_i(a'_i) \} - L_i(a'_i) \\
= 0 - L_i(a'_i) < 0
\]

Therefore, we have proved that, in Case (3a-iv), \( \Delta \Phi \) and \( \Delta u_i \) share the same sign.

From the above arguments, we have proved that \( \Delta \Phi_i \) and \( \Delta u_i \) hold the same sign for all possible sub-cases in Case (3a).

**Case (3b):** We have \( T_i(P') < T_{-i}(P') = T_{-i}(P) < T_i(P) \). This indicates that by node \( N_i \) transmitting at power \( a_i \), its primary goal has been met. Therefore, we have \( \ell_i(a_i) = 1 \), and \( \ell_i(a'_i) = 0 \). Then, we have \( \Delta \Phi_i = T_{-i}(P) - T_i(P') > 0 \) and \( \Delta u_i = L_i(a_i) + T_i(P) - T_i(P') > L_i(a'_i) > 0 \). Therefore, in Case (3b), \( \Delta \Phi_i \) and \( \Delta u_i \) hold the same sign.

**Case (3c):** In this case, \( T_i(P) < T_{-i}(P) = T_{-i}(P') < T_i(P') \). Therefore, we have \( \Delta \Phi_i = T_i(P) - T_{-i}(P') < 0 \). Exactly as in Case (3a), there are four sub-cases depending on the values of \( \ell_i(a_i) \) and \( \ell_i(a'_i) \). For Cases (3c-i), (3c-ii) and (3c-iv), we can follow similar lines of deduction as in Cases (3a-i), (3a-ii) and (3a-iv) to prove that \( \Delta u_i \leq 0 \) and, therefore, \( \Delta \Phi_i \) and \( \Delta u_i \) hold the same sign.

As for Case (3c-iii), it can be shown that it is impossible. Following the logic discussed in Case (3c-iii), we have \( \Phi_i = T_i(P) - T_{-i}(P') > T_i(P) - T_i(P') > 0 \), which is in contradiction to the assumption in Case (3c-iii) that \( T_i(P) < T_i(P') \).

So, in all sub-cases of Case (3c), \( \Delta \Phi_i \) and \( \Delta u_i \) have the same sign.

**Case (3d):** In this case, we can conclude that \( \Delta \Phi_i = T_{-i}(P) - T_i(P') = 0 \). Therefore, no matter what the sign of \( \Delta u_i \), we can conclude that \( \Delta \Phi_i \) and \( \Delta u_i \) have the same sign.

This concludes the consideration of all possible cases and sub-cases, in all of which we have shown that \( \Delta \Phi(P) \) and \( \Delta u_i(P) \) have the same sign. This proves that \( \Phi(P, t) \) is an ordinal potential function of \( u_i(P, t) \), and \( \Gamma \) is an ordinal potential game.  

Since this is an ordinal potential game, seeking the optimal global potential function yields a Nash equilibrium \([9], [34]\). In the next section, we propose a distributed localized algorithm that adaptively seeks to optimize the global potential function through each node seeking to optimize its own utility function defined in Eqn. (5).

V. **THE CTCA ALGORITHM**

This section presents the Cooperative Topology Control with Adaption (CTCA) algorithm in which each node plays the ordinal potential game, discussed in the previous section, with the goal of increasing network lifetime.

A. **Pseudo-code and rationale**

We use the same terminology as in the previous section, but for brevity, we omit the time \( t \) in our notation. The pseudo-code of the CTCA algorithm is presented in Fig. 1. The initialization phase of the algorithm (lines 01–13) enables each node to rapidly reduce its transmission power...
ALGORITHM 1: CTCA algorithm executed at node \( N_i \)

**Input:** Maximum transmission power \( P_{\text{max}} \)

**Output:** \( G'_i = (V_i, E'_i) \), the local topology of node \( N_i \)

**Initialization phase:**

1. Broadcast a Hello message with power \( p_{\text{max}} \);
2. Compile \( R_i \);
3. \( k \leftarrow \) number of reachable neighbors in \( R_i \);
4. Compile \( A_i = \{p(N_i, N_j) \mid N_j \in R_i\} \);
5. Sort \( A_i \) such that \( A_i[1] < A_i[2] < \cdots < A_i[k] \);
6. Broadcast neighbor info \((N_j, p(N_i, N_j))\) for \( N_j \in R_i \) with power \( A_i[k] \);
7. Receive the information sent by neighbor \( N_j \in R_i \);
8. Run DLSS algorithm, determine \( p_i \), compile \( R_i \);
9. Broadcast \( p_i \) with power level \( A_i[k] \);
10. Receive \( p_j \) from \( N_j \in R_i \), and compile \( I_i \);
11. \( S_i \leftarrow \text{AbleToReducePower}(N_i, p_i) \);
12. Broadcast \( S_i \) with power \( p_i \);
13. Receive \( S_j \) from \( N_j \in I_i \);

**Power adjustment phase:**

14. \( \text{EnergyInfoShared} \leftarrow \text{False} \);
15. while \( W_i > 0 \) do
16. \( q \leftarrow 0 \);
17. if not \( \text{EnergyInfoShared} \) then
18. \( \text{Broadcast} W_i \) with power \( p_i \);
19. \( \text{EnergyInfoShared} \leftarrow \text{True} \);
20. end
21. Send remaining energy request for \( N_j \in I_i \);
22. Receive \( W_j \) from \( N_j \in I_i \);
23. Wait for a random time \( t \in [0, T_1] \);
24. \( p_i, q \leftarrow \text{NAPA}(p_i, q) \);
25. Wait for \( T_2 \) time;
26. \( \text{EnergyInfoShared} \leftarrow \text{False} \);
27. Wait for \( T_3 - T_2 \) time;
28. end
29. return \( G_i \)

(Using the DLSS algorithm executed in line 08), compile \( A_i \) (the list of power levels that \( N_i \) can switch to) and prepare for the power adjustment phase illustrated in lines 14–29.

To offer a dynamic environment where each node updates its transmission power periodically, the algorithm operates in rounds. At the beginning of each round, each node broadcasts its current remaining energy if it has not been broadcasted before, which is indicated by the \( \text{EnergyInfoShared} \) flag. This process is described in lines 17–19. It will also send out a request for its reverse-link neighbors’ current remaining energy levels and update \( W_j \) for \( N_j \in I_i \) based on the received data as described in lines 21–22. The node will then wait for a random period of time \( t \) ranging from 0 to \( T_1 \) before executing the \textit{Neighbor-Assisted Power Adjust (NAPA)} function to adjust its transmission power. \text{NAPA} is the game-theoretic component of the CTCA algorithm.

The random time interval of \( t \in [0, T_1] \) is used to introduce randomness in the order in which
sensor nodes perform their power adjustment routines. Time $T_2$ in line 25 is needed because the energy level on a node is constantly changing and it helps to insert this waiting period to modulate how frequently a node requests energy level information from its reverse-link neighbors and how frequently it broadcasts its current energy level in response to requests from its reachable neighbors. Therefore, to ensure that node $N_i$’s reachable neighbors have a relatively up-to-date information on node $N_i$’s energy level, node $N_i$ changes its EnergyInfoShared flag to False so that once its reachable neighbors request its information, it will send back the latest energy level. On the other hand, node $N_i$ should reduce the number of times that its information is sent due to its own energy concerns. Therefore, if node $N_i$’s energy level has not changed noticeably so as to affect its reachable neighbors’ actions, it will keep its EnergyInfoShared as True until $T_2$ time has passed. Time $T_3$ in line 27 is needed to ensure that another round of the topology control process will not begin until the ongoing topology control process has finished.

The detailed NAPA function is illustrated in Algorithm 2. As we have explained in Section III, each node should always try to meet its primary goal unless it cannot be accomplished. This process for helping improving its primary goal is illustrated in lines 16–26 in Algorithm 2. If $N_i$ is to increase its transmission power to help improve its reverse-link neighborhood’s potential lifetime (as illustrated by node $N_3$ in Fig. 2(a)), several conditions have to be met:

1. The node with the minimum potential lifetime within $N_i$’s reverse-link neighborhood (node $m(i)$) cannot improve its potential lifetime on its own ($S_m$ is False), as in node $N_1$’s case illustrated in Fig. 2(a).
2. Node $m(i)$ is not transmitting at its minimum transmission power ($p_m > \min(A_m)$).
3. $m(i)$’s potential lifetime can be improved with $N_i$ transmitting at a certain larger power $a_i$. In this case, $N_{c(m)} \in H_i$, indicating that node $N_i$ should be transmitting at power $p(N_i, N_{c(m)})$.
4. $N_i$’s lifetime when transmitting at power $p(N_i, N_{c(m)})$ is larger than its reverse-link neighborhood’s potential lifetime, i.e., $W_i/p(N_i, N_{c(m)}) > L_{m(i)}'(P_i)$.

Conditions (1) and (2) are implemented in line 16 of Algorithm 2 and conditions (3) and (4) are implemented in line 18 in Algorithm 2. If all of the conditions listed above have been met, then node $N_i$ will choose to increase its transmission power so as to help improve its reverse-link neighborhood’s potential lifetime. On the other hand, if the node cannot help to improve its reverse-link neighborhood’s potential lifetime (indicated by CanHelp being False in line 28), then it will try to meet its secondary goal of improving its own estimated lifetime. This process is indicated by lines 29–36. In cases where a node can still improve its estimated lifetime, it will schedule to perform the NAPA function again after a random period of time (lines 39–40).

Function AbleToReducePower in Algorithm 3 illustrates the procedure implemented by each node to calculate its potential transmission power. It is also the function that helps a node $N_i$ determine its local connectivity function $c_i(a_i, t)$ at power level $a_i$. If there exists a reachable neighboring node $N_j$ of $N_i$ such that $N_j$ can communicate with the node that determines $N_i$’s current transmission power (denoted by $N_x$), then $N_i$’s potential transmission power, denoted by $p_i'(P_i)$, is one level below its current transmission power, and $S_i$ is True. In other words, there exists a path between node $N_i$ and node $N_x$ when node $N_i$ is transmitting at power level $p_i'(P_i)$, and thus, $c_i(p_i'(P_i), t) = 1$. For any power level $a_i > p_i'(P_i)$, we have $c_i(a_i, t) = 1$. On the other hand, if such node $N_j$ could not be found, we have $p_i'(P_i) = p_i$, and $S_i$ is False. At this point, for any power level $a_i < p_i$,
ALGORITHM 2: The Neighbor-Assisted Power Adjust (NAPA) function executed at node $N_i$

**Input**: current power level $p_i$, current execution number $q$

**Output**: new power level $p_i$, execution number $q$

1. if $q < Q$ then /* $Q$ is the maximum number of times a node may execute this function per round */
   2. $q \leftarrow q + 1$;
   3. for $N_j \in I_i$ do
      4. if $S_j$ then
         5. $k \leftarrow$ index of $p_j$ in $A_j$;
         6. $L'_j(P_i) \leftarrow A_j[k - 1]$;
      else
         7. $L'_j(P_i) \leftarrow p_j$;
      end
   end
   10. Compute node $m(i) \leftarrow \arg \min(L'_j(P_i) \mid N_j \in I_i)$;
   11. $S_m \leftarrow$ received $S_m$ from node $m(i)$;
   12. $p_m \leftarrow$ received node $m(i)$’s current transmission power;
   13. $S_i \leftarrow \text{AbleToReducePower}(N_i, p_i)$;
   14. $CanHelp \leftarrow \text{False}$;
   15. if not $S_m$ and $p_m > \min(A_m)$ then
      16. $N_{c(m)} \leftarrow \{ N_x \mid p(m(i), N_x) = p_m \}$;
      17. if $N_{c(m)} \in H_i$ and $W_i / p(N_i, N_{c(m)}) > L'_{m(i)}(P_i)$ then
         18. $p_i \leftarrow p(N_i, N_{c(m)})$;
         19. Broadcast NeighborInfo Request with power $p_i$;
         20. Receive response from newly added neighbors;
         21. Update $R_i$;
         22. $S_i \leftarrow \text{AbleToReducePower}(N_i, p_i)$;
         23. Inform $N_j \in \{R_i \cup I_i\}$ of $(p_i, S_i)$;
         24. $CanHelp \leftarrow \text{True}$;
      end
   end
   28. if not $CanHelp$ then
      29. if $p_i > p'_i(P_i)$ then
         30. $p_{\text{temp}} \leftarrow p_i$;
         31. $p_i \leftarrow p'_i(P_i)$;
         32. Update $R_i$;
         33. $S_i \leftarrow \text{AbleToReducePower}(N_i, p_i)$;
         34. Broadcast $(p_i, S_i)$ with power $p_{\text{temp}}$;
         35. Inform $N_j \in I_i$ of current $(p_i, S_i)$;
      end
   end
   38. if $S_i$ then
      39. Wait for a random time $t \in [0, T_1]$;
      40. $p_i, q \leftarrow \text{NAPA}(p_i, q)$;
   else
      42. return $p_i, q$;
   end
end
ALGORITHM 3: The AbleToReducePower function executed at node $N_i$

**Input:** current power level $p_i$

**Output:** Able to reduce power flag $S_i$

1. $N_{c(i)} \leftarrow N_x \mid p(N_i, N_x) = p_i$;
2. $W_{c(i)} \leftarrow$ node $N_{c(i)}$’s current power level;
3. $S_i \leftarrow$ False;
4. $p'_i(P_i) \leftarrow p_i$;
5. if $p_i > \min(A_i)$ then
6.     for $N_j \in R_i$ do
7.         if $N_{c(i)} \in R_j$ and $L_i(p_i) < \frac{W_{c(i)}}{p(N_{c(i)}, N_j)}$ then
8.             $S_i \leftarrow$ True;
9.             $k \leftarrow$ index of $p_i$ in $A_i$;
10.            $p'_i(P_i) \leftarrow A_i[k - 1]$;
11.            break;
12.        end
13.     end
14. return $S_i$;

node $N_i$ is transmitting at a power level lower than what is necessary to stay connected with node $N_x$, and therefore, loses its local connectivity. Note that, in either of the two cases, no communication among sensor nodes has to be conducted in order to calculate node $N_i$’s local connectivity function $c_i(a_i, t)$.

To ensure up-to-date information sharing amongst a node’s reachable neighborhood and its reverse-link neighborhood, the communication routines that are executed by each node are illustrated in Algorithm 4. These routines ensure that once a node has changed its current status (such as current transmission power, potential transmission power and current remaining energy), nodes whose status may be affected are informed.

As has been proved in the previous section, game $\Gamma = \langle N, A, U \rangle$ is an ordinal potential game, and seeking the optimal global potential function yields a Nash equilibrium. Therefore, given enough time, the NAPA procedure converges to an equilibrium. In our observations, we find that $Q = 4$ is adequate to ensure good performance. Therefore, in our implementations, we allow only four executions of the NAPA function per round per node.

The initialization stage of the CTCA algorithm as illustrated in Fig. 1 introduces the same order of computational and communication complexity as the DLSS algorithm, which is $O(\Delta^2)$. The communication and computation complexity of the CTCA algorithm at each round is $O(\Delta)$.

VI. SIMULATION RESULTS

A. Simulation and Energy Consumption Model

The energy model used in our simulation is identical to that used in the research literature on topology control [24], [35]. This model incorporates energy consumption due to transmission, reception, and for radio electronics in both free space and over a multi-path channel above a
ALGORITHM 4: Functions executed at $N_i$ upon receiving control/request messages.

Upon receiving transmission power updates from $N_j$
1. if $N_j \in I_i$ then
2.   if $p_j < p(N_j, N_i)$ then
3.     Remove $N_j$ from $I_i$;
4.   else
5.     Update $N_j$’s entry in $I_i$ with $(p_j, S_j)$;
6. end
7. if $N_j \in R_i$ then
8.   Update $N_j$’s entry in $R_i$ with $(p_j, S_j)$;
9. end
10. $S_{\text{temp}} \leftarrow S_i$;
11. $S_i \leftarrow \text{AbleToReducePower}(N_i, p_i)$;
12. if $S_i \neq S_{\text{temp}}$ then
13.   Broadcast $S_i$ with power $p_i$;
14.   Wait for a random time $t \in [0, T_1]$;
15. end
16. end
17.
18. Upon receiving NeighborInfo Request from $N_j$
19. if $N_j \notin I_i$ then
20.   Inform $N_j$ of $(W_i, p_i, S_i)$;
21.   Add $N_j$ to $I_i$;
22. end
23.
24. Upon receiving remaining energy request
25. if not EnergyInfoShared then
26.   Broadcast $W_i$ with power level $p_i$;
27.   EnergyInfoShared $\leftarrow$ True;
28. end

\[ E_{T_x}(d) = E_{\text{elec}} \times k + \begin{cases} 
\varepsilon_f \times d^2 \times k & \text{if } d < d_0 \\
\varepsilon_m \times d^4 \times k & \text{if } d \geq d_0 
\end{cases} \]
\[ E_{R_x} = E_{\text{elec}} \times k \]  
(9)

where $E_{T_x}(d)$ is the energy consumed in transmitting the signal to an area of radius $d$ and $E_{\text{elec}}$ is the energy consumed for the radio electronics. $\varepsilon_f$ is the transmitter’s amplifier coefficient in free space and $\varepsilon_m$ is the transmitter’s amplifier coefficient in the multi-path channel. $d_0$ is the distance threshold beyond which the channel is considered as multi-path. $E_{R_x}$ is the energy consumed in receiving the signal, and $k$ is the number of bits in the packet. Radio parameters are set as $E_{\text{elec}} = 50nJ/bit$, $\varepsilon_f = 10pJ/bit/m^2$, $\varepsilon_m = 0.0013pJ/bit/m^4$, and $d_0 = 87.8m$.

Our simulation is conducted for a square 10km×10km region within which 200 nodes are placed in random locations. Each node is equipped with 40kJ of energy and has a maximum transmission power $p_{\text{max}}$, which corresponds to a transmission radius of 20% of the width of the square region. The constant $T_3$ in the CTCA algorithm is chosen to be 1000 times larger than $T_1$,
and $T_2$ is chosen to be half of $T_3$. Each data point in the results reported here is the average of 200 randomly generated graphs. Using the batch means method to estimate confidence intervals, we have determined that the 95% confidence interval is within ±2% for each of the data points reported in our results. In our simulation model, we employ the TinyOS standard \cite{36} for sensor node data transmission, including its packet formats.

B. Comparative analysis against other algorithms

In this section, we compare the performance of CTCA algorithm against some of the other algorithms. Among the well-cited algorithms, our criteria for including them in this comparative analysis are the following:

- The algorithm applies to or allows application in which communication is uni-directional, where if node $N_i$ is within the communication radius of node $N_j$, node $N_j$ is not required to be within the communication radius of node $N_i$. It is the same assumption that we make in this paper.
- The communication and computational complexity for an adaptive algorithm is $O(\Delta)$ or lower each round.

Based on the above criteria, we have selected Directed Relative Neighborhood Graph (DRNG) \cite{1}, Directed Local Spanning Subgraph (DLSS) \cite{1}, Step Topology Control (STC) \cite{2}, and Routing Assisted Topology Control (RATC) \cite{3}. In the case of the RATC algorithm, it was reported in \cite{3} that when sensor nodes operate under a given level of 3-hop knowledge, the algorithm yields the best performance. Thus, we also allow up to 3-hop level of information to be exchanged among sensor nodes in the RATC algorithm.

In our experiments, every round, each node will send a designated data packet to every other node within the network, i.e., a node will send out $n - 1$ packets each round. Data packets are routed through the minimum energy consumption path. In case of the CTCA algorithm, at the beginning of each round, each node adjusts its transmission power according to the energy
Fig. 4: A comparative performance analysis of different algorithms.

situation in its local area; as for all other algorithms, each node will send out a hello message to check their neighbors’ availability.

In our simulations, we include the full energy costs of the overhead (such as hello messages) associated with executing each of the algorithms considered in the comparative analysis.

Fig. 3 reports the network lifetime achieved by the different algorithms. For each point in the graph, its x-axis value indicates the number of rounds that has passed. The y-axis value indicates the percentage of graphs (of the 200 randomly generated graphs used as a starting point in the experiments) that are still connected. As shown in the figure, a significantly larger fraction of graphs stay connected when using the CTCA compared to other previously-known algorithms.
On average, we find that the life of a network is extended by more than 50% compared to other algorithms.

Fig. 4 reports the network’s parameters (the average transmission power per node and the average energy cost along the minimum energy path) achieved by the different algorithms. For static algorithms such as DLSS, DRNG, STC, and RATC, the topology of the network is determined at the very beginning of the network’s lifetime. Therefore, their network’s parameters remain the same throughout the network’s lifetime, as indicated by straight lines in Figs. 4(a) and 4(b). The CTCA algorithm, on the other hand, changes the topology of the network with time, and thus, produces different parameters each round. In Figs. 4(a) and 4(b), we have reported each algorithm’s performance until 50% of the random graphs that we have generated become disconnected. DLSS is the only algorithm that achieves average transmission power or energy cost per path comparable to the CTCA algorithm. However, as time progresses, all algorithms except CTCA retain the same average transmission power per node until the network’s functional life ends, but the CTCA algorithm adapts accordingly and preserves connectivity for much longer. It is worth noting that, in the case of the CTCA algorithm, between the first round when a graph is connected to the 14th round when it is only 50% likely that it is connected, the average transmission power per node in the CTCA algorithm increases by only about 20%. The same observation can be made for the average energy cost along the minimum energy path.

Note that, the CTCA algorithm is an algorithm that determines the topology of the network. Therefore, in our simulation, to capture how the CTCA algorithm is able to help extend the network’s functional lifetime, we only employ the simplest routing algorithm—routing the packet through the minimum energy path. The performance of the CTCA algorithm may vary depending upon the routing algorithm used to transmit data packets. An efficient routing algorithm may help extend the network’s functional lifetime even further if the energy dissipation can be distributed more evenly. However, whether or not the routing algorithm is an efficient one, the CTCA algorithm accommodates the impact of the routing algorithm because it dynamically adapts to the current energy level at each node.

C. Comparison against the optimal solution

In topology control algorithms, the weight of an edge usually reflects the cost of transmission through that particular link. Alternatively, if we assign the weight of an edge \((N_i, N_j)\) at time \(t\) as:

\[
w_{i,j}(t) = \frac{p(N_i, N_j)}{W_i(t)}
\]

then, this weight function captures the amount of estimated lifetime consumed by the sender node \(N_i\) if a transmission is made through link \(N_i \rightarrow N_j\). A centralized topology control algorithm to minimize the maximum weight of an edge while preserving connectivity is trivial (e.g., based on removing edges from the graph in order of decreasing weight until removing an edge would destroy connectivity). Let \(T_{Opt,r}\) denote the maximum possible estimated lifetime of the network achieved using this optimal algorithm on the input graph in round \(r\).

Let \(T_{CTCA,r}\) denote the estimated lifetime of the network achieved using the CTCA algorithm on the input graph in round \(r\). We define the average price paid by the CTCA algorithm as:

\[
\text{Average price paid} = \text{avg} \left( \frac{T_{Opt,r}}{T_{CTCA,r}} \right)
\]
In our performance analysis, we compute the above average price paid by taking the average over multiple runs each with a different random graph as the input. We use the term *price* in line with the traditional terminology in game theory used in metrics comparing Nash equilibrium solutions against the social optimum (e.g., the price of anarchy [37]). In this section, we will use this metric (the average price paid by CTCA) as a measure of the quality of the solution reached by the CTCA algorithm (note that this only measures the quality of the solution and not the energy expended to reach the solution, addressed in the previous subsection in computing actual lifetimes).

In the following set of experiments, we study the quality of the solution delivered by the distributed CTCA algorithm in comparison to the centralized optimal solution. In our experiments, we begin with sensor nodes all with the same amount of starting energy (10J) and randomly deployed in a 1000m × 1000m square region. While the energy levels of nodes at the beginning of the 1st round is the same, the uneven distribution of the energy consumption on the sensor nodes leads to unevenness in the energy levels of the nodes in subsequent rounds. Accommodating this unevenness being an important goal of this paper, we trace the performance of the CTCA algorithm in different rounds (1st, 2nd and 6th). We choose the first round because it is when energy levels are all the same. We choose the 2nd round because this is the first round at the beginning of which energy levels may be different on different nodes. We choose the 6th round because, as shown in Fig. 3, the network may have passed nearly 50% of its lifetime after this many rounds.

1) The influence of communication radius of nodes: In our first set of experiments, we study the influence of the sensor node’s maximum communication radius on the performance of the CTCA algorithm. We use 200 sensor nodes deployed in the region. We conduct 500 independent simulations and report the results in Fig. 5. Fig. 5(a) reports the average price paid by the CTCA algorithm with different values of the communication radius of nodes in each of rounds 1, 2 and 6. Fig. 5(b) reports the percentage of times that the CTCA algorithm is able to find the optimal solution for different communication radii in those same rounds.

Fig. 5(a) shows that when the communication radius is very small (e.g., at 100 meters or 10% of the length of each side in the square area), the performance of the CTCA algorithm is very close to that of the optimal algorithm (with average price paid close to 1.0). This is because, given the same sensor node density, the topology graph is already very sparse when the communication range of the nodes is relatively small. The CTCA algorithm as well as the optimal algorithm cannot do much to improve the lifetime in this situation. On the other hand, as the communication range of the sensor nodes is increased, the number of choices available to each sensor node when trying to adjust its transmission power increases. A centralized algorithm is better able to exploit these choices because of more information available to it as compared to the localized information available to each node in the distributed CTCA algorithm. As the communication range increases even more, each node is able to gain sufficient information about the region around itself even in a distributed algorithm like CTCA. Therefore, at larger communication ranges, the disparity in the performance of the distributed CTCA algorithm and the centralized optimal algorithm reduces again with the average price paid by the CTCA algorithm reaching closer to 1.0.

The above phenomenon also explains why, as shown in Fig. 5(b), the CTCA algorithm is more likely to find the optimal solution when the communication range of nodes is very low.
(a) The average price paid by CTCA (average ratio of the optimal lifetime and that achieved by CTCA).

(b) The percentage of times that the CTCA algorithm finds the optimal solution.

Fig. 5: The performance of the distributed CTCA algorithm in comparison to the centralized optimal algorithm in different rounds plotted against the communication radius of the nodes.

compared to when the range is intermediate. The figure also shows that, as expected, the CTCA algorithm finds the optimal solution with high likelihood when the communication range of the nodes is high. Note that the weaker performance at intermediate ranges is an inherent limitation of a distributed algorithm which works with only localized information and not necessarily of the CTCA algorithm (which performs better than other distributed algorithms as shown in the previous subsection).

It is of interest to observe that, in round 6, the disparity in the energy levels remaining on
(a) The average price paid by CTCA (average ratio of the optimal lifetime and that achieved by CTCA).

(b) The percentage of times that the CTCA algorithm finds the optimal solution.

Fig. 6: The performance of the distributed CTCA algorithm in comparison to the centralized optimal algorithm in different rounds plotted against sensor node density, expressed in number of nodes per square kilometer.

different nodes becomes larger and which, in turn, triggers more frequent cooperation between nodes and more changes in the topology. Depending on the order in which different nodes cause these changes, the potential solution space available to the CTCA algorithm can be large. As a result, even though the average performance of the CTCA algorithm in the 6th round is similar to that in other rounds, it is much less likely to reach the perfect optimal solution in the 6th round.
2) The influence of sensor node density: In our second set of experiments, we limit the maximum communication radius of each sensor node to 200 meters (when the CTCA algorithm performs close to the worst in comparison to the optimal centralized algorithm). To study the impact of the sensor node density on the performance of the CTCA algorithm, we conducted an experiment with sensor node densities ranging from 100 to 300 per square kilometer (i.e., 100 to 300 nodes in the region in our simulation experiments). The results are reported in Fig. 6. Fig. 6(a) reports the average price paid by the CTCA algorithm for different densities in different rounds. Fig. 6(b) reports the percentage of times that the CTCA algorithm is able to find the optimal solution.

In Fig. 6, we observe a similar trend as in Fig. 5 with a dip in the performance at intermediate communication ranges in the former figure and a dip in the performance at intermediate densities in the latter figure. The trend is explained by the same phenomena described earlier in the context of changes in performance with changes in the communication radius of the nodes.

VII. CONCLUSION

In this paper, we proposed a game-theoretic approach for nodes in a sensor network to cooperatively change their transmission powers to help extend the network lifetime. We have proved the existence of a Nash equilibrium for our game and provided an algorithm, Cooperative Topology Control with Adaptation (CTCA), which achieves such an equilibrium. Our simulation results show that the CTCA algorithm is able to improve the lifetime of a wireless sensor network by more than 50% compared to the best previously-known algorithms.

To better assess the performance of the CTCA algorithm, we also compare the quality of the topology delivered by the CTCA algorithm to the optimal solution obtained using a centralized algorithm. Our results show that with increased information available to each node about its region (such as when the communication range is large), the CTCA algorithm performs closer to the optimal one. Also, the more topological options available to each node (such as when the node density is high), the more likely that the average performance of the CTCA algorithm is closer to the optimal one.

While a distributed algorithm like the CTCA is able to perform well with more information or options available at each node, we find that there is a significant gap between the average performance of the CTCA algorithm and the optimal centralized one. Even though the CTCA algorithm performs better than other distributed algorithms, this paper suggests that there may yet be more room for new research on better distributed algorithms.

While our work has used a game-theoretic approach under the constraint that the network remain connected, our algorithm can be adapted to other criteria that describe the functional life of a network (such as whether or not each portion of a certain region is covered by a sensor node within a pre-defined distance). The connectivity is captured in the term $C(P)$ in Equation (4) and can be replaced by a different criterion such as coverage. Our ongoing research is focused on developing and describing a generalized version of this approach.

REFERENCES

[1] N. Li and J. C. Hou, “Localized topology control algorithms for heterogeneous wireless networks,” IEEE/ACM Transactions on Networking, vol. 13, no. 6, pp. 1313–1324, December 2005.
[28] A. Xenakis, I. Katsavounidis, and G. Stamoulis, “Investigating wireless sensor network lifetime under static routing with unequal energy distribution,” in Proc. Asia-Pacific Signal Information Processing Association Annual Summit and Conference. New York City: IEEE, 2012, pp. 1–7.

[29] J. Luo, A. Iyery, and C. Rosenberg, “Throughput-lifetime tradeoffs in multihop wireless networks under an SINR-based interference model,” IEEE Transactions on Mobile Computing, vol. 10, no. 3, pp. 419–433, 2011.

[30] S. Mahfoudh and P. Minet, “Survey of energy efficient strategies in wireless ad hoc and sensor networks,” in Proc. Seventh International Conference on Networking. New York City: IEEE, 2008, pp. 1–7.

[31] P. Santi, “Topology control in wireless ad hoc and sensor networks,” ACM Computing Surveys, vol. 37, no. 2, pp. 164–194, June 2005.

[32] R. Machado and S. Tekinay, “A survey of game-theoretic approaches in wireless sensor networks,” ELSEVIER Computer Networks, vol. 52, no. 16, pp. 3047–3061, August 2008.

[33] A. A. Aziz, Y. A. Sekercioglu, P. Fitzpatrick, and M. Ivanovich, “A survey on distributed topology control techniques for extending the lifetime of battery powered wireless sensor networks,” IEEE Communications Surveys Tutorials, vol. 15, no. 1, pp. 121–144, 2013.

[34] A. B. MacKenzie, L. DaSilva, and L. A. DaSilva, Game Theory for Wireless Engineers. San Rafael, CA, USA: Morgan and Claypool, 2006.

[35] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” IEEE Transactions on Wireless Communications, vol. 4, pp. 660–670, October 2002.

[36] TinyOS, “TinyOS 2.0.2 documentation,” 2008, http://tinyos.net/tinyos-2.x/doc/txt/tep113.txt.

[37] E. Koutsoupias and C. Papadimitriou, “Worst-case equilibria,” in Proc. International Symposium on Theoretical Aspects of Computer Science. New York City: Springer, 1999, pp. 404–413.