Fault-Tolerant Topology Control for Heterogeneous Wireless Sensor Networks

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

Abstract—**This paper addresses fault-tolerant topology control in a heterogeneous wireless sensor network consisting of several resource-rich supernodes used for data relaying and a large number of energy constrained wireless sensor nodes. We introduce the k-degree Anycast Topology Control (k-ATC) problem with the objective of selecting each sensor's transmission range such that each sensor is k-vertex supernode connected and the maximum sensor transmission power is minimized. Such topologies are needed for applications that support sensor data reporting even in the event of failures of up to $k-1$ sensor nodes. We propose two solutions for the k-ATC problem: a greedy centralized algorithm that produces the optimal solution and a distributed and localized algorithm that incrementally adjusts sensors' transmission range such that the k-vertex supernode connectivity requirement is met. Simulation results are presented to verify our approaches.**

Keywords: Energy efficiency, fault-tolerant topology control, heterogeneous wireless sensor networks.

I. INTRODUCTION

In this paper we address topology control in heterogeneous WSNs consisting of two types of wireless devices: resource-constrained wireless sensor nodes deployed randomly in large numbers and a much smaller number of resource-rich supernodes, placed at known locations. The supernodes have two transceivers, one to connect to the wireless sensor network (WSN), and another to connect to the supernode network. The supernode network provides better QoS and is used to quickly forward sensor data packets to the user. With this setting, data gathering in heterogeneous WSNs has two steps: first, sensor nodes transmit and relay measurements on multihop paths towards a supernode (see Figure 1). Once a data packet encounters a supernode, it is forwarded using fast supernode-to-supernode communication toward the user application. Additionally, supernodes could process sensor data before forwarding.
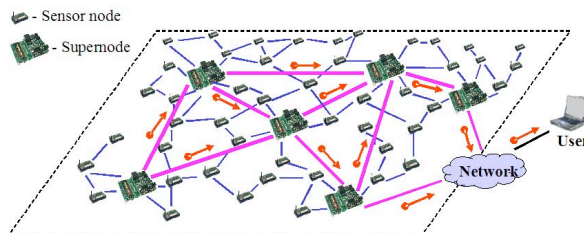
Fig. 1. Heterogeneous WSN.

A study by Intel [9] shows that using a heterogeneous architecture results in improved network performance, such as lower data gathering delays and a longer network lifetime. Hardware components of the heterogeneous WSNs are now commercially available [4].

We model topology control as a range assignment problem for which the communication range of each sensor node must be computed. The objective is to minimize the maximum sensor transmission power while maintaining k-vertex disjoint communication paths from each sensor to the set of supernodes. In this way, the network can tolerate the failure of up to $k-1$ sensor nodes. In contrast with range assignment in ad hoc wireless networks, this problem is not concerned with connectivity between any two nodes. Our problem is specifically tailored to heterogeneous WSNs, where data is forwarded from sensors to supernodes.

The contributions of this paper are the following: (1) we formulate the k-degree Anycast Topology Control (k-ATC) problem for heterogeneous WSNs, (2) we propose two solutions for solving the k-ATC problem: a centralized greedy algorithm that provides an optimal solution and a localized algorithm practical for large networks, and (3) we analyze the algorithms' performance through simulations.

The rest of this paper is organized as follows. In Section II we present related works on fault-tolerant topology control problems. Section III describes the heterogeneous WSN architecture, the network model,

and introduces the k-ATC problem. We continue in Section IV with our solutions for solving the k-ATC problem. Section V presents the simulation results, and Section VI concludes our paper.

II. RELATED WORK

The benefits of using heterogeneous WSNs, containing devices with different capabilities, have been presented recently in literature. In [16], it is reported that, when properly deployed, heterogeneity can triple the average delivery rate and can provide a 5-fold increase in the network lifetime.

The work in [13] introduces another type of heterogeneous WSN called actor networks, consisting of sensor nodes and actor nodes. The role of the actor nodes is to collect sensor data and perform appropriate actions. This paper presents an event-based coordination framework using linear programming and a distributed solution with an adaptive mechanism to trade off energy consumption for delay, when event data has to be delivered within a specific latency bound.

The majority of the existing work in fault-tolerant topology control studies the k-vertex connectivity requirement between any two nodes in the network. Such a requirement is more appropriate for ad hoc wireless networks, where any two nodes can be source and destination. In WSNs, data is transmitted from sensors to the sink(s), so maintaining a specific degree of fault-tolerance between any two sensors is not critical. It is rather important to have fault-tolerant data collection paths between sensors and sink(s) (or supernodes in our case).

A considerable amount of work ([1], [2], [7], [10] and [12]) has been done regarding the fault-tolerant topology control problem with the objective of minimizing the total power consumption while providing k-vertex connectivity between any two vertices. The majority of these algorithms are centralized, and they propose approximation algorithms for various topologies. Calinescu et al. [2] propose an algorithm with a performance ratio of 4 for the 2-connectivity problem. Jia et al. [10] propose a $3k$-approximation algorithm, $k \geq 3$, by first constructing the $(k-1)$th nearest neighbor graph and then augmenting it to k-connectivity by using one of the existing minimum edge weight k-connected algorithms. The Fault-Tolerant Cone-Based Topology Control (CBTC) algorithm proposed by Bahramgiri et al. [1] is a distributed and localized algorithm that achieves k-connectivity by having each vertex increase its transmission power until either the maximum angle between its two consecutive neighbors is at most $\frac{2\pi}{3k}$, or its maximal power is reached.

The works in [11] and [14] address the fault-tolerant topology control with the objective of minimizing the maximum power consumption. Ramanathan et al. [14] propose a centralized greedy algorithm for assuring biconnectivity ($k = 2$) that iteratively merges two biconnected components until only one remains. Li and Hou [11] introduce two algorithms for the k-connectivity problem; one centralized and the other distributed and localized. The algorithms examine edges in increasing order of their weight, and select edges only if the k-connectivity condition is not satisfied. These algorithms minimize the maximal power consumption between all k-vertex connected topologies.

Further, several previous works have addressed k-connectivity in a rooted graph. Frank and Tardos [6] study k-connectivity from the root to any other node with the objective of minimizing the total weight of the edges. They propose a polynomial time optimal solution using a maximum cost submodular flow problem. Wang et al. [15] propose an approximation algorithm with ratio k for k-connectivity from any node to the root, and an approximation algorithm with ratio $O(n)$ for k-connectivity from the root to any node. However, these algorithms are centralized.

Our work differs from the work in [15] by considering a heterogeneous WSN architecture. In this paper, we propose a centralized algorithm - $GATC_k$ - that minimizes the maximum transmission range, and a distributed and localized algorithm - $DATC_k$ - that is feasible for practical deployment of large scale WSNs.

III. PROBLEM DEFINITION AND NETWORK MODEL

A. Heterogeneous Network Architecture

We consider a heterogeneous WSN consisting of two types of wireless devices: resource-constrained wireless sensor nodes and resource-rich supernodes, as illustrated in Figure 1.

Sensor nodes have low cost, limited battery power, short transmission range, low data rate, and a low duty cycle. The main tasks performed by a sensor node are sensing, data processing, and data transmission/relaying. Supernodes have two radio transceivers, one for communication with sensor nodes and the other which is used to communicate with other supernodes. Supernodes are more expensive, have more power reserves, higher data rates, and better processing and storage capabilities than sensor nodes. The main task performed by a supernode is relaying data from sensor nodes to the user application.

B. Anycast Topology Control Problem

Let us consider a heterogeneous WSN consisting of sensors and supernodes. The supernodes are pre-deployed in the sensing area, they are connected, and their main task is to relay data from sensor nodes to the user application. On the other hand, sensor nodes are deployed randomly in the area of interest. Let R_{max}

be the maximum communication range of a sensor. We assume that when each sensor is using the transmission range R_{max}, there exist at least k paths from any sensor node to the set of supernodes.

Our goal is to provide a reliable data gathering infrastructure. We model this as the objective to establish the transmission range of each sensor such that: 1) there exist k vertex-disjoint communication paths from each sensor to the set of supernodes, and 2) the maximum sensor transmission power is minimized.

The first condition is needed in order to guarantee that data from every sensor reaches at least one supernode when up to $k-1$ sensor nodes fail. The second condition is needed to ensure an energy-efficient design, which is an important requirement in WSNs. We assume that once a packet with data from a sensor reaches a supernode, it will be relayed to the user application using a separate, more capable and less resource-constrained supernode network.

In this paper, instead of assuring the connectivity between any two sensor nodes, we want to provide communication paths from each sensor to the set of supernodes. A sensor can communicate with another sensor or with a supernode if the Euclidean distance between nodes is less than or equal to the sensor's communication range. We consider the path loss communication model where the transmission power of a sensor n_i is $p_i = r_i^\alpha$ for a transmission range r_i, where the constant α is the power attenuation exponent, usually chosen between 2 and 4. Please note that our algorithms can also be used for a more general power model $p_i = r_i^\alpha + c$, where c is a technology-dependent positive constant [8].

The formal definition is given below:

Definition 1 (k-degree Anycast Topology Control (k-ATC) Problem) Given a heterogeneous WSN with M supernodes and N energy-constrained sensors that can adjust their transmission ranges up to a maximum value R_{max}, determine the transmission range r_i of each sensor n_i such that (1) k-vertex supernode connectivity: there exist k-vertex disjoint communication paths from every sensor to the set of supernodes, and (2) the maximum sensor transmission power is minimized, $MAX\{p_i | i = 1 \ldots N\}$ =minimum.

Figure 2 (a) shows an example of a heterogeneous WSN which is 3-vertex supernode connected. This means that each sensor node has 3 vertex-disjoint paths to the set of supernodes. For example, sensor n_3 has the following three vertex-disjoint paths to the set of supernodes: (n_3, n_1, n_5), (n_3, n_4, n_6), and (n_3, n_2, n_6).

Sensor nodes are prone to failure due to physical damage or energy depletion, and thus our goal is to provide a topology that is fault-tolerant to sensor node failures. k-ATC problem applies to heterogeneous WSN applications where each sensor must have k-vertex disjoint
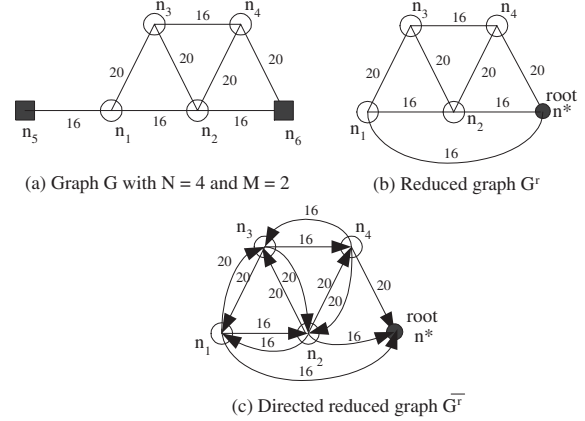

(a) Graph G with N = 4 and M = 2   (b) Reduced graph Gʳ

(c) Directed reduced graph G̅ʳ

Fig. 2. Construction of the reduced graph G^r and its directed version $\overline{G^r}$, where "■" are supernodes, "○" are sensor nodes, and "●" is the root.

data collection paths at all times. An example of such an application is when each sensor must periodically report its measurements and data reporting must be fault-tolerant to the failure of up to $k-1$ sensor nodes.

C. Network Model

We consider a heterogeneous WSN consisting of M supernodes and N sensor nodes, with $M \ll N$. We are interested in sensor-sensor and sensor-supernode communications only. That is, we do not model the supernode-to-supernode communication.

We represent the network topology with an undirected weighted graph $G = (V, E, c)$ in the 2-D plane, where $V = \{n_1, n_2, \ldots, n_N, n_{N+1}, \ldots, n_{N+M}\}$ is the set of nodes and E is the set of edges. The first N nodes in V are the sensor nodes and the last M nodes are the supernodes. When we refer in general to a $node\ n_i$, it means n_i can be either a supernode or a sensor node. If we specify the index i such that $1 \le i \le N$ then we refer to a sensor node. If $i > N$ then n_i refers to a supernode. We define the set of edges $E = \{(n_i, n_j) | dist(n_i, n_j) \le R_{max}\}$, where $dist()$ is the Euclidean distance function.

The cost function $c(u, v)$ represents the power requirement for both nodes u and v to establish a bidirectional communication link between u and v. Then the cost function is defined as $c(u, v) = (dist(u, v))^\alpha$.

The directed graph $\overline{G} = (V, \overline{E}, c)$ of G is obtained by replacing each edge (u, v) in E with two directed edges (u, v) and (v, u) in $\overline{E}$. The two directed edges have the same cost as $c(u, v)$ in G.

We assume that each node has a unique id, such as the MAC address, and that each node is able to gather its own location information using one of the localization techniques for wireless networks, such as [3].

Definition 2 (Reachable Neighborhood) The reachable neighborhood $\Gamma(n_i)$ is the set of nodes that n_i can reach by using the maximum transmission range R_{max}, $\Gamma(n_i) = \{n_j \in V | (n_i, n_j) \in E\}$.

Definition 3 (Weight Function) Given two edges (u_1, v_1) and (u_2, v_2) in E, the weight function $w : E \to R$ satisfies $w(u_1, v_1) > w(u_2, v_2)$ if and only if:

- $dist(u_1, v_1) > dist(u_2, v_2)$, or
- $dist(u_1, v_1) = dist(u_2, v_2)$ AND $max\{id(u_1), id(v_1)\} > max\{id(u_2), id(v_2)\}$, or
- $dist(u_1, v_1) = dist(u_2, v_2)$ AND $max\{id(u_1), id(v_1)\} = max\{id(u_2), id(v_2)\}$AND $min\{id(u_1), id(v_1)\} > min\{id(u_2), id(v_2)\}$.

The weight function w guarantees that two edges with different end nodes have different weights. The weight function definition in a directed graph is similar.

Definition 4 (k-vertex Supernode Connectivity) The heterogeneous network is k-vertex supernode connected if for any sensor node $n_i \in V$, there are k pairwise vertex disjoint paths from n_i to the set of supernodes (to one or more supernodes). Or equivalently, the heterogeneous network is k-vertex supernode connected if the removal of any $k-1$ sensor nodes (and all the related links) does not partition the network; that is, for every sensor node n_i there will be a path from n_i to a supernode.

D. Reduced Graph

Given a graph $G(V, E, c)$ corresponding to a heterogeneous WSN and constructed as specified in Section III-C, we construct its *reduced graph* $G^r(V^r, E^r, c^r)$ as follows. We substitute the set of supernodes with only one node called the root. Then $V^r = \{n_1, n_2, ..., n_N, n^*\}$, where the first N nodes are the sensor nodes, and the last node is the root. Edges between sensors remain the same, while an edge between a sensor and a supernode becomes an edge between the sensor and the root. Figure 2 (b) shows an example of the reduced graph G^r for a heterogeneous WSN with 4 sensor nodes and 2 supernodes.

If a sensor is connected to more than one supernode then only one edge is added in G^r with the cost corresponding to the distance to the closest supernode. The pseudocode for constructing the reduced graph is presented in the Algorithm 1.

We define the directed version $\overline{G}^r(V^r, \overline{E}^r, c^r)$ of the reduced graph as follows. Every undirected edge in G^r between two sensors is replaced with two directed edges in $\overline{G}^r$. An edge in G^r between a sensor and the root is replaced in $\overline{G}^r$ with only one directed edge from the sensor to the root. The costs of the edges in $\overline{G}^r$ remain the same as in G^r. Figure 2 (c) shows an example of constructing the directed reduced graph $\overline{G}^r$.

The definitions for the reachable neighborhood and weight function remain unchanged for the reduced

---

Algorithm 1 Construct Reduced Graph (G(V, E, c), N, M)

1: $V^r := \{n_i | n_i \in V \ and \ i \leq N\} \cup \{n^*\}$;
2: $E^r := \phi$;
3: **for** each edge $(n_i, n_j) \in E$ **do**
4:    **if** $(i \leq N)$ AND $(j \leq N)$ **then**
5:       $E^r := E^r \cup (n_i, n_j)$ and $c^r(n_i, n_j) := c(n_i, n_j)$;
6:    **else if** $((i \leq N)$ AND $(j > N))$ OR $((i > N)$ AND $(j \leq N))$ **then**
7:       $u := min(i, j)$ and $v := max(i, j)$;
8:       **if** $(n_u, n^*) \notin E^r$ **then**
9:          $E^r := E^r \cup (n_u, n^*)$ and $c^r(n_u, n^*) := c(n_u, n_v)$;
10:       **else if** $((n_u, n^*) \in E^r)$ AND $(c^r(n_u, n^*) > c(n_u, n_v))$ **then**
11:          $c^r(n_u, n^*) := c(n_u, n_v)$;
12:       **end if**
13:    **end if**
14: **end for**

---

graphs G^r and $\overline{G}^r$. Next, we define the k-vertex connectivity in the reduced graph G^r.

Definition 5 (k-vertex Connectivity in a Reduced Graph) The reduced graph G^r is k-vertex connected to the root if, for any sensor node $n_i \in V^r$, $i \leq N$, there are k-vertex disjoint paths from n_i to the root n^*. Or equivalently, the reduced graph G^r is k-vertex connected if the removal of any $k-1$ sensor nodes (and all the related links) does not partition the network.

Lemma 1 A heterogeneous WSN is k-vertex supernode connected if and only if the corresponding reduced graph is k-vertex connected to the root.

Proof: Let us consider a sensor node n_i. Assume that the network is k-vertex supernode connected. Then there are k-vertex disjoint paths between n_i and the set of supernodes. By replacing each supernode in the path with the root n^*, we obtain k vertex disjoint paths between n_i and n^* in the reduced graph G^r.

Similarly, if G^r is k-vertex connected, then for any sensor node n_i there are k-vertex disjoint paths between n_i and n^*. Then for any such path $(n_i, n_{i_1}, \ldots, n_{i_j}, n^*)$ we can take an equivalent path in G by replacing n^* with a supernode n_q, $q > N$, such that $(n_{i_j}, n_q) \in E$ and $c(n_{i_j}, n_q) = c^r(n_{i_j}, n^*)$. These paths in G are k-vertex supernode connected. $\square$

Definition 5 and Lemma 1 apply also to the directed reduced graph $\overline{G}^r$.

IV. SOLUTIONS FOR k-ATC PROBLEM

In this section we present two solutions for the k-ATC problem: a centralized algorithm in Section IV-A and a localized algorithm in Section IV-B.

Algorithm 2 Algorithm $GATC_k$

Input: $G(V, E, c)$, a k-vertex supernode connected graph

Output: power assignment p_i for each sensor node n_i

1: Construct the reduced graph $\overline{G}^r(V^r, \overline{E}^r, c^r)$ of G;
2: Let $\overline{G}_k := (V_k, \overline{E}_k, c^r)$ with $V_k := V^r$ and $\overline{E}_k := \overline{E}^r$;
3: Sort all edges in $\overline{E}_k$ in decreasing order of weight (using Definition 3);
4: **for** each edge (u, v) in the sorted order **do**
5:    $\overline{E}'_k := \overline{E}_k \setminus \{(u, v)\}$;
6:    **if** u is k-vertex connected to the root in the graph $(V_k, \overline{E}'_k)$ **then**
7:      $\overline{E}_k := \overline{E}'_k$;
8:    **end if**
9: **end for**
10: **for** $i := 1$ to N **do**
11:    $p_i := max\{c^r(n_i, n_j) | n_j \in V_k \text{ and } (n_i, n_j) \in \overline{E}_k\}$;
12: **end for**

A. $GATC_k$: Fault-tolerant Global Anycast Topology Control

In this section we present a centralized greedy algorithm, $GATC_k$, that builds a k-vertex supernode connected subgraph and then assigns a power to each vertex to cover all 1-hop neighbors.

This algorithm minimizes the maximum transmission power for all the sensor nodes, among all other k-vertex supernode connected subgraphs. This property is important since it will balance the power consumption among all sensor nodes.

The algorithm $GATC_k$ starts from the k-vertex supernode connected graph G, constructs its reduced graph G^r and then transforms it to a directed graph $\overline{G}^r$ as explained in Section III-D. Based on Lemma 1, G^r and $\overline{G}^r$ are k-vertex connected to the root. We examine all edges in $\overline{G}^r$ in decreasing order and remove an edge (u, v) if after its removal, sensor node u remains k-connected to the root. Then the algorithm computes the power p_i for any sensor node n_i such that n_i can directly communicate with any other node joined by an edge in $\overline{E}_k$.

By using network flow techniques [5], a query on whether two vertices are k-connected in a graph (V, E) can be answered in $O(E + V)$-time for any fixed k. Therefore, the complexity of $GATC_k$ is $O(\overline{E}^r(\overline{E}^r + V^r)) = O((\overline{E}^r)^2)$.

Theorem 2 (Correctness). If G is k-vertex supernode connected then the power assigned by $GATC_k$ to each sensor node guarantees a k-vertex supernode connected topology. Thus $GATC_k$ preserves the k-vertex supern-

ode connectivity of G.

Proof: Since G is k-vertex supernode connected, the graphs G^r and $\overline{G}^r$ are k-connected to the root (see Lemma 1). We start with a graph $\overline{G}_k := \overline{G}^r$ and remove edges. We prove that the resultant graph $\overline{G}_k$ remains k-connected at the end of line 9 in algorithm $GATC_k$.

We show this using induction. The base case is the reduced graph $\overline{G}^r$ which is k-vertex connected to the root and is the starting graph before any edge removal. We show that if $\overline{G}_k$ is k-vertex connected to the root before the removal of an edge (u, v), then it remains k-vertex connected to the root after the edge removal as long as u remains k-vertex connected to the root.

To show that $\overline{G}_k$ is k-vertex connected to the root, we show that after the removal of any set C of vertices, $|C| \leq k - 1$, the remaining sensor nodes are still connected to the root. Let us take any sensor node n_i. Before the removal of (u, v), n_i has k-vertex disjoint paths to the root, say $p_1, \ldots, p_k$. If (u, v) is not on any path $p_1, \ldots, p_k$, then the removal of (u, v) does not affect n_i's connectivity. Let us assume now that (u, v) belongs to one of the paths, let us say $(u, v) \in p_k$. If $|C| < k - 1$, then after the removal of C and edge (u, v), n_i is still connected to the root.

Consider now the case $|C| = k - 1$ when any $k - 1$ vertices are removed from the graph. The only critical case is when one vertex is removed from each path $p_1, \ldots, p_{k-1}$ and edge (u, v) is removed from the path p_k. Node n_i is still connected to u along the path p_k and let us call this path p'_1 which is a subpath of p_k. Vertex u is k-vertex connected to the root after the removal of (u, v) so there are k vertex disjoint paths between u and the root. Since $|C| = k - 1$, only $k - 1$ such paths can be broken, so after the removal of C, there will still exist one path between u and the root, let us call it p'_2. Then $p'_1 + p'_2$ will give us a path between n_i and the root.

Therefore, we conclude that $\overline{G}_k$ remains k-vertex connected to the root after the removal of (u, v) as long as u remains k-vertex connected to the root. $\square$

Theorem 3. The maximum transmission range (or equivalently power) among all the sensor nodes is minimized by $GATC_k$.

Proof: We show this property by contradiction. Let (u, v) be the first edge that is not removed from $\overline{E}_k$ as we examine the list of decreasingly ordered edges by weight. Then u will have the maximum range between all the senor nodes in $\overline{G}_k$.

Assume by contradiction that there exists a topology $\widetilde{G}$ that has the maximum transmission range from all the sensor nodes less than $c^r(u, v)$. Then the induced topology $\widetilde{G}$ does not contain any edge with cost greater than or equal to $c^r(u, v)$. Since $GATC_k$ could not remove the edge (u, v) from $\overline{E}_k$, it results that without the edge (u, v), u is not k-connected to the root, thus

f_i	1 if sensor node n_i decided its final power, otherwise 0
r_i	Current transmission range of sensor node n_i
p_i	Current transmission power level of sensor node n_i, $p_i = r_i^\alpha$
$\Gamma(n_i)$	$\{n_j | dist(n_i, n_j) \leq R_{max}\}$
p_i^{max}	Transmission power of node n_i needed to reach the farthest neighbor in $\Gamma(n_i)$
p_i^{min}	Transmission power of node n_i needed to reach the closest k neighbors in $\Gamma(n_i)$
$\overline{G}_{n_i}$	n_i's localized topology view; directed graph $\overline{G}_{n_i} = (V_{n_i}, \overline{E}_{n_i})$ where $V_{n_i} = \Gamma(n_i)$ and $\overline{E}_{n_i} = \{(n_u, n_v) | n_u, n_v \in V_{n_i}$ AND $dist(n_u, n_v) \leq r_u\}$
$\Gamma'(n_i)$	$\{n_j | dist(n_i, n_j) \leq r_i\} \cup \{n_j | (r_i < dist(n_i, n_j) \leq R_{max})$ AND (n_i is k-vertex connected to n_j in $\overline{G}_{n_i}$) $\}$

violating the connectivity correctness of $\widetilde{G}$. □

B. $DATC_k$: Fault-tolerant Distributed Anycast Topology Control

$DATC_k$ is a distributed and localized algorithm that efficiently assigns the power level of each sensor node such that k-vertex supernode connectivity is preserved. The main algorithm notations are introduced in Table I.

Each node n_i starts by constructing its localized neighborhood $\Gamma(n_i)$ based on *Hello* messages exchanged between neighbors with communication range R_{max}. Each sensor node n_i starts a distributed process to decide its final transmission power p_i, as presented next in the $DATC_k$(i) algorithm.

Sensor node n_i computes p_i^{max} and p_i^{min}, which are the power needed to reach the farthest neighbor in $\Gamma(n_i)$ and the first k neighbors in $\Gamma(n_i)$, respectively. Each sensor node n_i uses an iterative process to establish its final power, starting from p_i^{min}. The final power p_i selected by node n_i will be between p_i^{min} and p_i^{max}. In order for a node to be k-vertex connected, it must have at least k disjoint neighbors. Therefore, its transmission power must cover the k closest neighbors resulting in $p_i \geq p_i^{min}$.

The goal of the algorithm is to find a minimum transmission power p_i of node n_i, $p_i \in [p_i^{min}, p_i^{max}]$, such that each node n_j in $\Gamma(n_i)$ is either within communication range r_i of node n_i or there exist k-vertex disjoint paths between n_i and n_j. When this condition is met, node n_i declares its current power estimate as its final power assignment by setting f_i to 1.

Every node n_i maintains p_j values for each neighbor $n_j \in \Gamma(n_i)$. We assume that a node n_i has a complete topological view of its 1-hop neighborhood, and this is a directed, asymmetric graph $\overline{G}_{n_i}$ where nodes have different communication ranges. The edge set of this

---

Algorithm 3 Algorithm $DATC_k$(i)

1: $p_i := p_i^{min}$;
2: **if** $p_i^{min} = p_i^{max}$ **then**
3:     $f_i := 1$;
4: **else**
5:     $f_i := 0$;
6: **end if**
7: Broadcast(i, p_i, f_i);
8: **while** $f_i = 0$ **do**
9:     compute Δp_i, the minimum incremental power needed to cover at least one neighbor in $\Gamma(n_i) - \Gamma'(n_i)$;
10:     start timer t;
11:     **if** broadcast message received from a neighbor n_j before t expires **then**
12:         update $\Gamma'(n_i)$ and Δp_i
13:         **if** $\Gamma'(n_i) = \Gamma(n_i)$ **then**
14:             $f_i := 1$;
15:             Broadcast(i, p_i, f_i);
16:             Return;
17:         **end if**
18:     **end if**
19:     **if** timer t expires **then**
20:         $p_i := p_i + \Delta p_i$;
21:         update $\Gamma'(n_i)$;
22:         **if** $\Gamma'(n_i) = \Gamma(n_i)$ **then**
23:             $f_i := 1$;
24:         **end if**
25:         Broadcast(i, p_i, f_i);
26:     **end if**
27: **end while**
28: Return;

---

topology changes over time (new edges are added) as n_i receives advertisements from its neighbors. A node n_i can compute the connectivity between any two 1-hop neighbors if nodes broadcast their location or their 1-hop neighbors in the *Hello* messages.

The algorithm executes in at most $|\Gamma(n_i)| - k$ rounds (or iterations). In each round, power level p_i is minimally incremented with Δp_i such that at least one node in $\Gamma(n_i) - \Gamma'(n_i)$ is added to $\Gamma'(n_i)$. Δp_i can easily be computed since node n_i maintains the distance and location information for all nodes in $\Gamma(n_i)$. The algorithm is completed when $\Gamma(n_i) = \Gamma'(n_i)$.

All broadcast messages sent to advertise new power level updates are sent with power level $p_{max} = R_{max}^\alpha$. If, during the back-off interval, a broadcast message is received from a neighbor in $\Gamma(n_i)$, then $\Gamma'(n_i)$ and Δp_i are updated before continuing the back-off waiting. When node n_i decides to broadcast its advertisement, it updates its power level p_i and neighboring set $\Gamma'(n_i)$ in lines $20 - 21$ of algorithm $DATC_k$.

The rounds should be designed to have each node advertise its new power estimate once in the case that the node did not establish its final power yet. Ideally, nodes send the broadcast without colliding with their neighbors' advertisement. To avoid simultaneous updates among neighbors, a back-off scheme is used. Each node backs-off for a time that is inversely proportional to its calculated gain before sending a broadcast. The gain can be computed for example as $p_{max} - (p_i + \Delta p_i)$. In this case, nodes with a smaller power level will advertise earlier, thus helping the nodes with larger transmission powers. This approach could help to balance power consumption among sensor nodes.

The complexity of the $DATC_k$ algorithm run by each node n_i is polynomial in the total number of nodes $N + M$. Let us denote the maximum node degree as Δ, that is $\Delta = max_{i=1..N} |\Gamma(n_i)|$. The complexity of $DATC_k$ is $O(\Delta^5)$. This is because for a node n_i there are at most $O(\Delta)$ rounds, the time to update Δp_i is at most $O(\Delta^3)$, and during the back-off at most Δ neighbor updates can be received.

Theorem 4 (Correctness). If G is k-vertex supernode connected then the power level assignment provided by the $DATC_k$ algorithm guarantees a k-vertex supernode connected topology.

Proof: For simplicity of our discussion, let us consider G's reduced graph G^r and its directed version $\overline{G}^r$, both being k-connected to the root.

Our proof is by induction. The starting graph $\overline{G}^r$ is the base case, corresponding to a transmission power p_i^{max} for any sensor n_i. We remove edges from this graph when we set the power of a node n_i to a value less than p_i^{max}. For the inductive step, let us assume that the current graph is k-connected to the root and that an edge (n_i, n_j) is removed, or equivalently n_i's final range assignment $r_i < dist(n_i, n_j)$. In conformity with the $DATC_k$ algorithm, this happens when n_i remains k-vertex connected to n_j after the removal of (n_i, n_j).

We show that any sensor node n_u maintains its k-vertex connectivity to the root, after the removal of (n_i, n_j). For this, we show that the removal of any set C of vertices, $|C| \leq k - 1$ and $n_u \notin C$, does not affect the connectivity of n_u.

Before the removal of (n_i, n_j), n_u has k-vertex disjoint paths to the root, let us say $p_1, \ldots, p_k$. If (n_i, n_j) is not on any path $p_1, \ldots, p_k$, then n_u's connectivity is not affected. Assume now that (n_i, n_j) belongs to one of the paths, let us say $(n_i, n_j) \in p_k$. If $|C| < k - 1$, then after the removal of C and edge (n_i, n_j), n_u is still connected to the root.

Let us now consider $|C| = k - 1$. The only critical case is when one vertex is removed from each path $p_1, \ldots, p_{k-1}$ and edge (n_i, n_j) is removed from the path p_k. Node n_u is still connected to n_i along the path p_k.

Let us call this path (which is subpath of p_k) p'_1. Node n_j is still connected to the root along the path p_k; we call this path (which is subpath of p_k) p'_3. Vertex n_i is k-vertex connected to the node n_j so after the removal of C only $k - 1$ such paths can be broken, it follows that n_i is still connected to n_j and let us call this path p'_2. Then $p'_1 + p'_2 + p'_3$ will give us a path between n_u and the root.

Therefore, we conclude that the $DATC_k$ algorithm assigns power levels to nodes in such a way that guarantees a k-vertex supernode connected topology. □

V. SIMULATION

In this section we present the results of our simulation. We analyze and compare the performance of $GATC_k$ and $DATC_k$ with various parameters. We developed a C++ custom simulator to test the two algorithms.

A. Simulation environment and settings

The sensors are deployed in a $100m \times 100m$ area. The supernodes are uniformly deployed in this area. The following parameters and their trade-offs are considered in the simulation:

1) The network size N. We vary N to examine the scalability of the proposed algorithms. In the small scale network, the network size is varied from 10 to 50. In the large scale network, it is in the range 100 to 500.

2) The number of supernodes M. We set M to 1 and 3 for small scale networks and between 2 and 10 in large scale networks.

3) The value of k. We use 2 and 4 as the value of k in the simulation.

4) The power attenuation exponent α. We use 2 and 4 as its value in the simulation.

5) The initial sensor transmission range R_{max}. In order to guarantee that the WSN is k-vertex supernode connected, we set the initial sensor transmission range in a small scale network to be $50m$, and in a large scale network to $20m$ for $k = 2$, and $40m$ for $k = 4$.

A sample network is discarded if it is not k-vertex supernode connected with its initial settings. For each tunable parameter, the simulation is repeated 100 times. The performance metrics are as follows:

1) The maximum transmission power among all the sensors.

2) The total power consumption. This is the summation of power consumption of each sensor (according to its final transmission range).

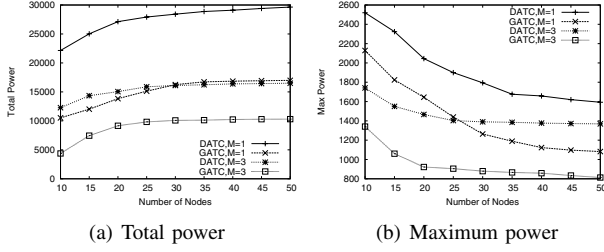3) The reduction ratio of both maximum power consumption and total power consumption. We use

(a) Total power (b) Maximum power

Fig. 3. $GATC_k$ and $DATC_k$ in the small scale network.

(a) Total power (b) Maximum power

(c) Reduced rate of total power (d) Reduced rate of maximum power

Fig. 4. Comparison of $GATC_k$ and $DATC_k$ in the large scale network.

the initial sensor transmission range to calculate the original power consumption.

B. Simulation results

Figure 3 shows the comparison of $GATC_k$ and $DATC_k$ in a small scale network, where N varies from 10 to 50, M is 1 or 3, k is 2, and α is 2. In (a) we compare the performance of $GATC_k$ and $DATC_k$. We observe that compared to $GATC_k$, $DATC_k$ has its total power doubled in general. When M is 3, less power is needed than when it is 1. Thus more supernodes scattered in the network help to preserve the k-vertex supernode connectivity.

With the increase in the number of sensors, the total power increases. However, as shown in (a) the rate of power increase is lower than that of sensors. This is because with more sensors, the total power tends to increase, but the power consumption for each sensor is reduced. (b) is the maximum power comparison. With the increase in the number of sensors, the maximum power decreases for all approaches. $GATC_k$ has the smallest maximum power and $DATC_k$ has the largest one for both $M = 1$ and $M = 3$. When M is larger, the maximum power is smaller for all approaches. These simulations verify our theoretical result that $GATC_k$ minimizes the maximum transmission range between all sensors.

Figure 4 is the comparison of $GATC_k$ and $DATC_k$ in large scale network, where N varies from 100 to 500, M is 3, α is 2, and k is 2 or 4. (a) is the total power consumption comparison. $GATC_k$ has better performance than $DATC_k$, and the power consumption is small when k is 2. When k is 2, the power consumption increases with the number of sensors. However, when k is 4, the power consumption even decreases slightly. This is because when k is large, the increased number of sensors not only brings some extra power consumption but also helps each sensor reduce its transmission power, and the latter effect is more significant than the former one. (b) is the maximum power comparison. As the number of sensors grows, the maximum power decreases for both approaches. $GATC_k$ has smaller maximum power than
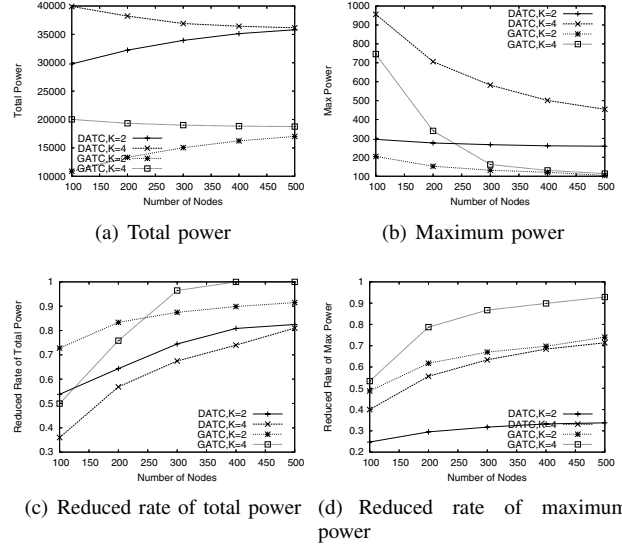
$DATC_k$. When k is 4, a larger maximum power is needed.

Figures 4 (c) and (d) are the corresponding reduced rate of the total power consumption and maximum power consumption. We compute the reduced rate of the total power consumption as $1 - (p_1 + p_2 + \ldots + p_N)/(p_{max} \times N)$, and the reduced rate of the maximum power as $1 - p_i/p_{max}$, where $p_{max} = R_{max}^2$ and p_i is the maximum final power among all sensors. $GATC_k$ has larger reduction rate than $DATC_k$ in terms of both total power and maximum power. All of the reduction rates increase with the number of sensors. For the total power graph, the increase of power consumption in both $GATC_k$ and $DATC_k$ is small (see Figure 4 (a)) with the growth of the number of sensors, while the initial power consumption increases linearly. As for the maximum power, $GATC_k$ and $DATC_k$ have smaller maximum power when N is larger (see Figure 4 (b)), while the initial maximum power remains unchanged.

Figure 5 is the analysis of $GATC_k$ and $DATC_k$ with different values for the parameters M and k. (a) and (b) show the resultant power consumption when α is 4 in large scale network. We set $M = 3$ and $k = 2, 4$. We can see that these two figures are similar to Figures 4 (a) and (b) except that the difference among all the curves is more significant.

Figures 5 (c) and (d) show the variation of the total power and the maximum power with the number of supernodes when $N = 200$, $\alpha = 2$, and $k = 2, 4$. We can see that with the increase of M, the power consumption is decreased. This is consistent with the results shown in Figures 3 (a) and (b). Again, when k is 4, more power

(a) Total power, $\alpha = 4$ (b) Maximum power, $\alpha = 4$

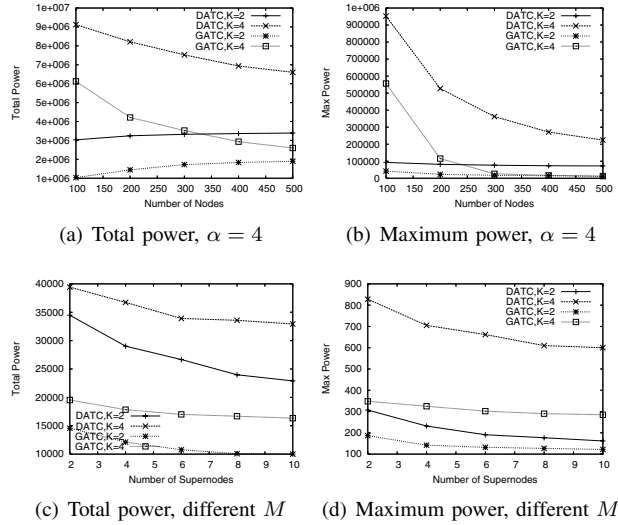(c) Total power, different M (d) Maximum power, different M

Fig. 5. Comparison of $GATC_k$ and $DATC_k$ when $\alpha = 4$ and with increasing M.

is necessary and $GATC_k$ has better performance than $DATC_k$. We can also observe that the decrease of power in $DATC_k$ is more significant than that of $GATC_k$.

The simulation results can be summarized as follows:

1) $GATC_k$ has better performance than $DATC_k$ in terms of maximum power consumption and total power consumption. The tradeoff is that $GATC_k$ is centralized while $DATC_k$ is localized.

2) More supernodes help to reduce the power consumption of each sensor. Larger k demands larger power consumption in all approaches.

3) When the number of sensors N increases, the total power consumption increases slightly for both $GATC_k$ and $DATC_k$ if k is 2; it even decreases slightly if k is 4. The maximum power consumption decreases with the growth of N.

4) The reduction rate in terms of both total power and maximum power increases with the growth of N.

5) When α increases from 2 to 4, the difference between $GATC_k$ and $DATC_k$ is more significant.

VI. CONCLUSIONS

In this paper we addressed the k-degree Anycast Topology Control problem in heterogeneous WSNs with the objective of minimizing the maximum sensor transmission power while providing k vertex independent paths from each sensor to the set of supernodes. Such a topology provides the infrastructure for fault-tolerant data gathering applications that are robust to the failure of up to $k - 1$ sensors.

We proposed two solutions to this problem: a centralized approach $GATC_k$ and a localized algorithm $DATC_k$. $GATC_k$ is an optimal solution that minimizes

the maximum power between all sensor nodes. Simulation results show that the centralized algorithm $GATC_k$ has better performance in terms of power consumption. $DATC_k$ consumes more power, sometimes as much as two times that of $GATC_k$. However, $DATC_k$ is a distributed and localized algorithm, and this is an important property in WSNs showing that this algorithm is scalable and practical for large networks.

For future work, we plan to extend our work for applications that require a fault-tolerant bidirectional topology that provides communication paths both from sensors-to-supernodes and from supernodes-to-sensors.

REFERENCES

[1] M. Bahramgiri, M.T. Hajiaghayi, V.S. Mirrokni, Fault-tolerant and 3-Dimensional Distributed Topology Control Algorithms in Wireless Multi-hop Networks, *IEEE Int'l. Conf. on Computer Communications and Networks (ICCCN'02)*, 2002.

[2] G. Calinescu and P. -J. Wan, Range Assignment for High Connectivity in Wireless Ad Hoc Networks, *2nd Int'l. Conf. on Ad-Hoc Networks and Wireless*, 2003.

[3] X. Cheng, A. Thaeler, G. Xue, and D. Chen, TPS: A Time-Based Positioning Scheme for Outdoor Wireless Sensor Networks, *IEEE INFOCOM*, 2004.

[4] Crossbow motes and Stargate-Xscale, http://www.xbow.com

[5] S. Even and R. E. Tarjan, Network Flow and Testing Graph Connectivity, *SIAM Journal on Computing*, Vol. 4, pp. 507-518, 1975.

[6] A. Frank and E. Tardos, An Application of Submodular Flows, *Linear Algebra and its Applications*, Vol. 114-115, pp. 329-348, 1989.

[7] M. Hajiaghayi, N. Immorlica, and V. S. Mirrokni, Power Optimization in Fault-Tolerant Topology Control Algorithms for Wireless Multi-Hop Networks, *9th Annual Int'l. Conf. on Mobile Computing and Networking*, 2003.

[8] F. Ingelrest, D. Simplot-Ryl, and I. Stojmenovic, Optimal Transmission Radius for Energy Efficient Broadcasting Protocol in Ad Hoc and Sensor Networks, *IEEE TPDS*, 17(6), 2006.

[9] Heterogeneous Networks with Intel XScale, http://www.intel.com/research/exploratory/heterogeneous.htm

[10] X. Jia, D. Kim, P. Wan, and C. Yi, Power Assignment for k-Connectivity in Wireless Ad Hoc Networks, *INFOCOM*, 2005.

[11] N. Li and J. C. Hou, FLSS: A Fault-Tolerant Topology control Algorithm for Wireless Networks, *10th Annual Int'l. Conf. on Mobile Computing and Networking*, 2004.

[12] E. L. Lloyd, R. Liu, M. V. Marathe, R. Ramanathan, and S. S. Ravi, Algorithmic Aspects of Topology Control Problems for Ad Hoc Networks, *MOBIHOC*, 2002.

[13] T. Melodia, D. Pompili, V. C. Gungor, and I. Akyildiz, A Distributed Coordination Framework for Wireless Sensor and Actor Networks, *MobiHoc'05*, 2005.

[14] R. Ramanathan and R. Rosales-Hain, Topology Control of Multihop Wireless Networks using Transmit Power Adjustment, *IEEE INFOCOM*, 2000.

[15] F. Wang, M. T. Thai, Y. Li, X. Cheng, and D. -Z. Du, Fault Tolerant Topology Control for All-to-One and One-to-All Communication in Wireless Networks, *IEEE Transaction on Mobile Computing*, under review.

[16] M. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu, S. Singh, Exploiting Heterogeneity in Sensor Networks, *IEEE INFOCOM*, 2005.