SCIENCE CHINA
Information Sciences

• RESEARCH PAPERS •

December 2010 Vol. 53 No. 12: 2527–2537
doi: 10.1007/s11432-010-4101-y

Algorithms for k-fault tolerant power assignments in wireless sensor networks

LIU Li[1,2*], LI Lian[1,3] & HU Bin[1,4]

[1]*School of Information Science and Engineering, Lanzhou University, Lanzhou* 730000*, China;*
[2]*LRI, Univ Paris-sud and CNRS, Orsay* F-91405*, France;*
[3]*School of Computer and Information, Hefei University of Technology, Hefei* 230009*, China;*
[4]*Department of Computing, Birmingham City University, Galton Building, Perry Barr, Birmingham* B42 2SU*, UK*

Received December 24, 2008; accepted September 13, 2010

Abstract This paper addresses fault-tolerant many-to-many routing power assignments in heterogeneous wireless sensor networks. We introduce the k-fault tolerant power assignments problem with the objective of assigning each sensor node transmission power such that (1) any pairwise sensor node is k-vertex connected and (2) the total power consumption is minimized. We propose three solutions for this problem: two centralized algorithms, a greedy algorithm and an $O(\sqrt{n/\epsilon})$-approximation algorithm, and an h-hop distributed and localized algorithm. Related theorems and proofs are presented to prove the correctness of our approaches. Furthermore, simulation and experiment results are presented to verify the efficiency of our approaches.

Keywords heterogeneous wireless sensor network, fault tolerant, power consumption, algorithm

Citation Liu L, Li L, Hu B. Algorithms for k-fault tolerant power assignments in wireless sensor networks. Sci China Inf Sci, 2010, 53: 2527–2537, doi: 10.1007/s11432-010-4101-y

1 Introduction

Wireless sensor networks (WSNs) contain hundreds or thousands of sensor nodes with limited sensing, computing, communication abilities and power. These networks compose a maintenance-free, fault-tolerant platform for gathering different kinds of data from the extreme or inhospitable environment, which has a broad range of applications, such as military surveillance, disaster discovery, wildlife protection, etc.

Heterogeneous wireless sensor networks are special WSNs consisting of different kinds of sensor nodes, such as Berkeley motes and Mica motes which are resource-constrained sensor nodes, and Medusa MK-2 and µAMPS which are resource-rich senor nodes. Different kinds of sensor nodes have different properties in their cost, size, functions and performances. One essential difference is their transmission ranges for communication.

Since a number of applications [1] require the sensor nodes sending data to the other sensor nodes, the transmission range for data delivery is important for each sensor node. Due to different transmission ranges of sensor nodes, a sensor node may not be able to send the data to its destination node directly.

*Corresponding author (email: liliu@lzu.edu.cn)

It may need other medial nodes to transfer its data. Therefore multi-hop mode [2] is an efficient strategy for data delivery.

Multi-hop mode requires the sensor nodes on the data routing path active in order to forward the message. However, a sensor node fails due to either power dissipation or communication obstruction by an obstacle. Thus redundant data routing paths should be established to guarantee the message forwarding to the end users.

When a node is k-vertex connected to another node, it means that k-vertex disjoint routing paths exist between the two nodes, namely k-vertex connectivity (or k-fault tolerance). Therefore, when the failure of sensor nodes up to $k-1$ occurs, it is still able to find a routing path to deliver data to the destination node. As a result, the network is more tolerable to node failures.

On the other hand, energy consumption [3] is essential in determining the lifespan of a sensor network. To save the energy, a node can adjust its transmission range by changing its radio power. Consequently, a network lifespan can be prolonged by assigning each sensor node an appropriate transmission range.

A considerable amount of works [4–6] have contributed to maintain a specific degree of fault-tolerance between any sensor node and a fixed node, usually a sink node, rather than between any two nodes. Some researches are based on the assumption of an uniform transmission range for all the sensor nodes.

However, since WSNs contain a large number of different kinds of sensor nodes, clustering is an efficient way to organize the nodes. Sensor nodes transmit their data to the immediate local cluster head. The communication relies highly on the cluster head, and the energy depletion of cluster heads is faster than other nodes. Therefore, protocols like LEACH [7] were proposed to solve this issue by rotation of the roles of various nodes. Sensor nodes are likely to be selected as the cluster heads in turn according to some negotiated rules. Correspondingly, a cluster head becomes a sensor node as its energy is consumed under a threshold. Any sensor node has the chance to be a cluster node. Therefore, maintaining a specific degree of fault-tolerance between any pair of sensor nodes is critical to WSNs.

Maintaining fault-tolerance data routing paths between any pair of nodes is important for the architectures employing node-scheduling schemes or cluster-based protocols. In this paper, we present the transmission range assignments problem that (1) the total transmission power of all sensor nodes is minimized; (2) maintaining k-vertex disjoint paths between any pair of sensor nodes.

The rest of this paper is organized as follows. In section 2, we present related works on fault-tolerant topology control problems, and show their main results and contributions. In section 3, the k-fault tolerant power assignments problem in heterogeneous WSNs is defined as well as the network models. Section 4 presents two centralized algorithms: a centralized algorithm and an approximation algorithm, and one distributed and localized algorithm. We analyze the performance of these algorithms through simulations and experiments in section 5. And section 6 concludes this paper.

2 Related works

Fault-tolerant topology control algorithms have been proposed to maintain the network connectivity as well as reduce energy consumption. The topology derived is more susceptible to node failures. A considerable amount of work has been done on the fault-tolerant topology control problem with the objective of minimizing the total power consumption while maintaining k-vertex connectivity between any two nodes. The majority of these algorithms are cataloged into two types, centralized algorithms, and localized and distributed algorithms.

Li and Hou [8] presented a centralized greedy algorithm FGSS and a distributed algorithm FLSS, both algorithms preserve k-connectivity. Hajiaghayi et al. [9] presented three approximation algorithms. The first algorithm gives an $O(k \log k)$-approximation. The second algorithm achieves $O(k)$-approximation for general graphs. The last algorithm is a distributed algorithm for the cases of 2-connectivity and 3-connectivity. All these algorithms can be used to minimize power consumption while maintaining k-edge connectivity with guaranteed approximation factors. Jia et al. [10] presented several approximation algorithms for the specific cases of this problem. They proposed a $3k$-approximation algorithm for any $k \geqslant 3$, a $(k+12H(k))$-approximation algorithm for $k(2k-1) \leqslant n$, a $(k+2\lceil (k+1)/2 \rceil)$-approximation algorithm for $2 \leqslant k \leqslant 7$, a 6-approximation algorithm for $k = 3$, and a 9-approximation algorithm for $k = 4$.

Han et al. [11] addressed the problem of deploying relay nodes to provide fault-tolerance with higher network connectivity by providing relay nodes with the same transmission radius. Li et al. [12] addressed the problem of fault-tolerant many-to-one routing in static wireless networks with asymmetric links. Segal and Shpungin [13] developed a general approximation framework for various topology control problems under the k-fault resilience criterion in the plane.

Several works were presented for distributed topology control mechanisms for 3-dimensional settings. Ghosh et al. [14] presented two efficient alternatives. One is a heuristic based on $2-D$ orthographic projections, which provides excellent performance in practice, but is not guaranteed to produce a connected network theoretically. The second is a more rigorous approach based on spherical Delaunay triangulation. Both algorithms they presented have the complexity of $O(d \log d)$.

Moraes et al. [15] showed that the related optimization problems can be classified into four main variants, regarding the topology of the input graph (symmetric or asymmetric) and of the solution (unidirectional or bidirectional). Our work differs from [9–11, 13, 14, 16, 17] as follows:

1. We consider a heterogeneous WSN, which means our work can be used for general wireless sensor network models (completely asymmetric and unidirectional).
2. We address the problem of fault-tolerant many-to-many routing in heterogeneous WSN.
3. We consider a better approximation algorithm for heterogeneous WSN.
4. We consider an h-hop distributed and localized algorithm for heterogeneous WSN.

3 Definition and related theorem

We aim to provide data routing paths to guarantee required connectivity between any pair of nodes. A node can communicate with one another if the Euclidean distance between the two nodes is less than or equal to the node's transmission range. We denote the function of power consumption by $f_p(r)$, where r is the node's transmission range. We take the path loss communication model as the power consumption model. The minimal transmission energy needed for correct reception by neighbor nodes at distance r is proportional to $r^\alpha + c$, where α is the power attenuation exponent and c is an environment-dependent constant [16]. With this model the power consumption is given by $p_i = f_p(r_i) = r_i^\alpha + c$, where p_i is the power consumption in the transmission range r_i of the sensor node n_i.

Definition 3.1. k-Fault tolerant power assignments (FTPA$_k$). Given a heterogeneous WSN consisting of N nodes with the various transmission ranges. For each node n_i, it can adjust the transmission ranges up to its maximum value $R_i^{\max}$. Determine the power p_i of node n_i such that 1) there exist k-vertex disjoint data routing paths between any pair of nodes; 2) the total power consumed over all sensor nodes is minimized, namely $\sum_{i=1}^N p_i$ is minimized.

Network model. A directed weighted graph $G(V, E)$ is represented as the network topology, where $V = \{n_1, n_2, \ldots, n_N\}$ is the set of nodes and $E = \{\langle n_i, n_j\rangle : \mathrm{dist}(n_i, n_j) \leqslant R_i^{\max}\}$ is the set of edges. dist() is the Euclidean distance function. For each edge $\langle u, v\rangle \in E$, there exists a weight $w(u, v)$ associated with it. $w(u, v)$ represents the power consumption needed by u to communicate with v. According to the power consumption model, the weight function is defined as $w(u, v) = \mathrm{dist}(u, v)^\alpha + c$.

FTPA$_k$ aims to construct a minimum-cost k-vertex connected network by finding a set of power assignments for each node. Equivalently, the network is k-vertex connected if any $k-1$ nodes removed without partitioning the network. That is, for every node there exists at least one path to any other node.

Theorem 3.1 shows that there exists a graph that any pair of nodes has at least k disjoint paths. If at least $k+1$ disjoint paths exist from u to v, and one of such paths is the edge $\langle u, v\rangle$ which connects u to v directly, then there are still at least k disjoint paths between any pair of nodes after removing the edge $\langle u, v\rangle$ from the network.

Theorem 3.1. A graph $G(V, E)$ is a k-vertex connected directed graph. If $\langle u, v\rangle \in E$ and there are at least $k+1$ disjoint paths from u to v, namely $\lambda(u, v) \geqslant k+1$, $G \setminus \{\langle u, v\rangle\}$ is a k-vertex connected graph.

Proof. To prove $G \setminus \{\langle u, v\rangle\}$ is a k-vertex connected graph, we need to prove that after the removal of any set of vertices S, where $\mid S \mid= k-1$, and $u, v \notin S$, at least one path exists between any pair of vertices.

For any pair of two vertices a and b, q_1, q_2, $\ldots$, q_k, are k independent paths between a and b. If $E(q_i) \cap \{\langle u, v\rangle\} = \phi$, $1 \leqslant i \leqslant k$, removal of any set of vertices S where $\mid S \mid= k-1$, does not affect the connectivity between a and b.

Otherwise, let us see the case where $E(q_i) \cap \{\langle u, v\rangle\} \neq \phi$. Since at least $k+1$ independent paths exist between u and v, u and v have at least k independent paths after removal of $\langle u, v\rangle$, namely p_1, p_2, $\ldots$, p_k. Vertex a is still connected to u along path q_i and we call it q'_1, and vertex u is still connected to b along path q_i and we call it q'_2. Path p_k connects vertices u and v. Then $q'_1 + p_k + q'_2$ is the path between a and b.

4 Algorithms

4.1 Greedy power assignments algorithm

Greedy power assignments (GPA$_k$) algorithm is an intuitive algorithm that produces a k-vertex connected spanning subgraph and assigns to each vertex the minimum transmission range needed for reaching all of its neighbors. GPA$_k$ is a centralized algorithm that the whole network topology is known before it starts. A number of broadcast protocols, like [17], can collect the global network topology.

Algorithm: GPA$_k$
Input: $\bar{G}(V, \bar{E})$ and w
Output: assignment p_i for each n_i
1 Sort all edges in E in decreasing order of weight w
2 **for each** edge $\langle u, v\rangle$ in the sorted order **do**
3   **if** $\lambda(u, v) \geqslant k+1$ in $G(V, E)$ **then**
4       $E = E \setminus \{\langle u, v\rangle\}$
5   **end if**
6 **end for**
7 **for** $i = 1$ to N **do**
8    $p_i = \max\{w(n_i, n_j) : \langle n_i, n_j\rangle \in E\}$
9 **end for**

The algorithm starts from sorting the edges in decreasing order of their corresponding weights. Based on Theorem 3.1, we examine all edges in this order iteratively and test whether the graph keeps k-vertex connected if remove an edge $\langle u, v\rangle$. After removing all the possible edges, the remaining subgraph is k-vertex connected. Then the algorithm computes the transmission power for each node so that it can communicate directly with any neighbor nodes in the resultant subgraph.

Testing whether a graph is k-connected can be finished in $O(V+E)$ time for any fixed k by using the network flow techniques. Accordingly, the time complexity of this algorithm is $O(E(V+E))$.

We denote by G_i the resultant graph after ith iteration from lines 2–7. Obviously, $G_0 = G$.

Theorem 4.1 (Correctness)**.** If G is k-vertex connected, GPA$_k$ guarantees a k-vertex connected transmission range assignment.

Proof. We prove that the graph $\langle V, E\rangle$ remains k-vertex connected after removal of edge $\langle u, v\rangle$ in each iteration from lines 2–7. We prove it by recursion. If G_i is k-vertex connected, then G_{i+1} is k-vertex connected. By Theorem 3.1, if u is k-vertex connected to v after removing $\langle u, v\rangle$ in G_i, then $G_{i+1} = G_i \setminus \langle u, v\rangle$ is still a k-connected directed graph. Since G_0 is a k-vertex connected graph, for each G_i it is a k-vertex connected graph.

Theorem 4.2 (Local minimum)**.** G_{i+1} is a k-vertex connected graph with minimum weight reduced from G_i .

Proof. Assume that $\langle u_0, v_0\rangle$ is the first edge in the list of the decreasing order by weight and $\lambda(u_0, v_0) \geqslant k+1$. We denote the total weight of $G_i \setminus \{\langle u_0, v_0\rangle\}$ by $w_0(G_{i+1})$. If we remove $\langle u, v\rangle$ which is not the first edge and satisfies $\lambda(u, v) \geqslant k+1$, we denote the total weight of $G_i \setminus \{\langle u, v\rangle\}$ by $w(G_{i+1})$. It is easy to prove that $w_0(G_{i+1}) \leqslant w(G_{i+1})$ as $w(u_0, v_0) \leqslant w(u, v)$.

4.2 Cheriyan, Vempala and Vetta(CVV)-approximation algorithm

We present an approximation CVV_k algorithm based on a solution of the minimum-cost k-vertex connected spanning subgraph (VCSS_k) problem proposed by Cheriyan et al. [18].

VCSS_k problem is to find a spanning subgraph H of minimum cost such that H is k-vertex connected. In VCSS_k problem, a graph $G(V, E)$ is given with non-negative edge weights $w : E \to R_+$, with the objective to find a spanning subgraph of G such that 1) the subgraph is k-vertex connected; 2) the total weight of the subgraph is minimized.

The main differences between VCSS_k and FTPA_k are that 1) the weight of each edge is uncertain, but the weight in VCSS_k is known; 2) our objective is to find the assignments of the transmission range of each node, rather than the total weight of all edges.

The VCSS_k problem is proved NP-hard for $k > 1$. Many approximation algorithms are proposed to resolve this problem. Kortsarz and Nutov [19] designed a more than $k/2$ approximation algorithm.

Cheriyan et al. proposed an approximation algorithm for k-VCSS problem. The algorithm is an $O(\sqrt{n/\epsilon})$ algorithm for any $\epsilon > 0$ and $k \leqslant (1-\epsilon)n$.

Algorithm: CVV_k
Input: $G(V, E)$ (or $\bar{G}(V, \bar{E})$) and w
Output: assignment p_i for each n_i
1 Construct G' of G by assigning each edge the same weight
2 $G'_{CVV} = CVV(G', k)$
3 **for** $i = 1$ to N **do**
4 $\quad p_i = \max\{w(n_i, n_j) : \langle n_i, n_j\rangle \in E(G'_{CVV})\}$
5 **end for**

Theorem 4.3. CVV_k is an $O(\sqrt{n/\epsilon})$-approximation algorithm.

Proof. Let R_{opt} be the optimal solution and R_{sol} be the solution obtained by CVV_k to FTPA_k problem. We have to prove that $R_{\mathrm{sol}} \leqslant O(\sqrt{n/\epsilon}) \times R_{\mathrm{opt}}$.

Let $R^{\mathrm{VCSS}}_{\mathrm{opt}}$ be the optimal solution to k-VCSS problem and $R^{\mathrm{VCSS}}_{\mathrm{sol}}$ be the solution obtained by the algorithms Cheriyan et al. proposed. From the way we construct G', we find that any solution to the k-VCSS problem is also a solution to FTPA_k problem, and vice versa. Since the solution of k-VCSS problem has the approximation ratio $O(\sqrt{n/\epsilon})$, we conclude that $R_{\mathrm{sol}} = R^{\mathrm{VCSS}}_{\mathrm{sol}} \leqslant \pi \times R^{\mathrm{VCSS}}_{\mathrm{opt}} = \pi \times R_{\mathrm{opt}}$, where $\pi = O(\sqrt{n/\epsilon})$.

The complexity of CVV_k is dominated by the complexity of the Cheriyan et al's algorithms, which run in time $O(k^2 n^4 (n + k^{2.5}))$. Thus, CVV_k has the runtime complexity of $O(k^2 n^4 (n + k^{2.5}))$.

4.3 Distributed and localized power assignments algorithm

In this section we present the distributed and localized power assignments (DLPA_k) algorithm. Before we start, a series of notations are described in this section.

Notations.

r_i: the current transmission range of n_i.

N_i: the neighbor nodes of n_i when n_i uses its current transmission range r_i, namely $N_i = \{n_j : \mathrm{dist}(n_i, n_j) \leqslant r_i\}$.

N_i^0: the neighbor nodes of n_i when n_i has its maximal transmission range $R_i^{\max}$, namely $N_i^0 = \{n_j : \mathrm{dist}(n_i, n_j) \leqslant R_i^{\max}\}$.

$G_i(N_i, E_i)$: the localized topology of node n_i with its neighbor nodes of N_i, E_i presents the connected edges among nodes in N_i.

$r_i^{\max}$: the transmission range of n_i needed to reach the farthest neighbor in N_i^0.

$r_i^{\min}$: the minimal transmission range of n_i needed to reach k neighbors in N_i^0.

$\Delta r_i(N)$: the minimal incremental transmission range of n_i needed to reach a node in N.

$N - M$: a node set consist of nodes belong to N, but not belong to M, namely $N - M = \{n : n \in N$ and $n \notin M\}$.

The following code demonstrates the process by which each node readjusts its transmission range to achieve the fault tolerant requirements. Each nodes executes this algorithm respectively to find an adaptable transmission range. The h-hop DLPA$_k$ algorithm executed on a node n_i initially assigns itself the transmission range $r_i^{\min}$ by which n_i is able to reach k neighbors (line 1). Then n_i broadcasts its localized topology $G_i(N_i, E_i)$ (lines 2–4). The broadcasting message can be routed for h hops. After broadcasting, n_i checks whether it has already reached the maximal transmission range $r_i^{\max}$ (line 5). If it has, n_i executes function EXIT which ends this algorithm and returns the result transmission range r_i with its corresponding transmission power $f_p(r_i)$ (line 6). Otherwise, n_i repeats to check whether it has k paths for each of its neighbor nodes in N_i^0 by either receiving the localized topology from other nodes or augmenting its transmission range until it reaches its maximal transmission range (lines 8–26). n_i executes function WAIT(t) to wait for a period of time t to receive other nodes' localized topology (line 9). If n_i receives a broadcasting message from n_r, n_i obtains the localized topology of n_r and updates its own localized topology after adding n_r's localized topology (lines 10–16). Then n_i executes function CHECK to check whether it has k paths for each neighbor node of N_i^0 in its updated localized topology G_i (line 13). If it has, n_i ends with EXIT (line 14). Otherwise, it waits for other nodes' broadcasting messages. Since n_i does not have k paths for each neighbor node as far as the waiting time t expires, n_i augments its transmission range to $r_i + \Delta(r_i(N_i^0 - N_i))$ that n_i can reach one more neighbor node in N_i^0 (lines 17–20). Then n_i broadcasts its updated localized topology and checks whether it has k paths for each neighbor node after augmenting its transmission range (lines 21–24).

Algorithm: DLPA$_k$
Input: h, N_i^0, $R_{\max}$
Output: assignment p_i for n_i
1 $r_i = r_i^{\min}$
2 $N_i = \{n_j : \mathrm{dist}(n_i, n_j) \leqslant r_i\}$
3 $E_i = \{\langle n_i, n_j\rangle : \mathrm{dist}(n_i, n_j) \leqslant r_i\}$
4 BROADCAST(i, G_i)
5 **if** $r_i == r_i^{\max}$ OR $N_i == N_i^0$ **then**
6 EXIT$(f_p(r_i))$
7 **end if**
8 **while** $r_i \leqslant r_i^{\max}$ **do**
9 WAIT(t)
10 **while** RECEIVE$(r, G_r(N_r, E_r))$ before t expires **do**
11 $N_i = N_i \cup N_r$
12 $E_i = E_i \cup \{\langle n_r, n_j\rangle : \langle n_r, n_j\rangle \in E_r$ and $n_j \in N_i^0\}$
13 **if** CHECK$(G_i, N_i^0) == k$ **then**
14 EXIT$(f_p(r_i))$
15 **end if**
16 **end while**
17 **if** t expires **then**
18 $r_i = r_i + \Delta r_i(N_i^0 - N_i)$
19 $N_i = N_i \cup \{n_j : \mathrm{dist}(n_i, n_j) \leqslant r_i\}$
20 $E_i = E_i \cup \{\langle n_i, n_j\rangle : \mathrm{dist}(n_i, n_j) \leqslant r_i\}$
21 BROADCAST(i, G_i)
22 **if** CHECK$(G_i, N_i^0) == k$ **then**
23 EXIT$(f_p(r_i))$
24 **end if**
25 **end if**
26 **end while**

The localized topology information needed by each node n_i is the connectivity information of its visible neighborhood N_i^0. Each node n_i constructs its localized topology based on broadcasting Hello messages sent by neighborhood node n_r with its localized topology G_r. The Hello messages can be forwarded h-hops using a time-to-live equal to h. By exchanging the localized topology information with h-hop neighbors, node n_i can start a distributed process to decide its transmission range by testing whether it is k-vertex connected to each node in N_i.

When h is equal to 1, n_i only obtains its visible neighbors' information to construct a localized topology $G_i(N_i^0, E_i)$, which means n_i is directly connected with the other node in N_i^0, otherwise n_i is k-vertex connected to it in graph $G_i(N_i^0, E_i)$. However, the result only requires n_i is k-vertex connected to other node in graph $G(V, E)$. Hence, the final result produced by 1-hop message exchanging remains redundant edges, which induces more power consumption. By multi-hop message exchanging, the main advantage is that a larger neighborhood is used to search for k disjoint paths. Especially, when h is equal to the diameter of the whole network, each node obtains the global information. However, since h is too large, the messages exchanged through the whole network induce the message block and more power consumption.

DLPA$_k$ starts by assigning each node an initial range that can reach the first k neighbors. This is because a node is k-vertex connected, it must have at least k disjoint neighbors. Each node uses an iterative process to increase its range gradually until it is k-vertex connected with its 1-hop neighbors. Each node maintains a network topology of its h-hop neighbors. The edge set changes as the node receives messages from its h-hop neighbors. The node updates its network topology, and evaluates the connectivity to any of its 1-hop neighbors.

Theorem 4.4. If $G(V, E)$ is a k-vertex connected graph, DLPA$_k$ guarantees a k-vertex connected graph topology after the power assignments.

Proof. For each node n_i, we can say its neighbor nodes with its maximal transmission range, equivalently $N_i = \{n_j : \mathrm{dist}(n_i, n_j) \leqslant R_{\max}\}$, and its edge set $E_i = \{\langle n_i, n_j\rangle : \mathrm{dist}(n_i, n_j) \leqslant R_{\max}\}$. After the DLPA$_k$ finishes, the transmission range r_i of n_i is assigned and its resultant edge set $E_i^r = \{\langle n_i, n_j\rangle : \mathrm{dist}(n_i, n_j) \leqslant r_i\}$. The set of edges $E' = E_i - E_i^r$ are removed. However, for each node in N_i, n_i is k-vertex connected to it. We can observe that DLPA$_k$ is identical to the removal of each edge in E' step by step. Since each edge $\langle n_i, n_j\rangle$ is removed, DLPA$_k$ guarantees that n_i is k-vertex connected to n_j after the removal. According to Theorem 3.1, we can say that a k-vertex connected graph topology is guaranteed after the removals.

DLPA$_k$ runs in at most $| N_i^0 | -k$ iterations. In each iteration, node n_i waits for a period of time t and listens to the broadcast messages sent by its h-hop neighbors. If, during the waiting time, a message is received from a neighbor n_r, then the network topology is updated. When n_i is k-vertex connected to each of its neighbor nodes in N_i^0 after updating, the algorithm terminates. Otherwise, n_i keeps listening until t expires. n_i increases its range with Δr_i to cover at least one more neighbor. Then, n_i broadcast its updated topology information to its h-hop neighbors. The algorithm terminates if k disjoint paths can be found from n_i to other node after the update.

The execution time complexity of DLPA$_k$ executed by each node n_i is a polynomial. We denote $\triangle$ as the maximum node degree, namely $\triangle = \max\{| N_i^0 |: 1 \leqslant i \leqslant N\}$. For each node n_i, it runs at most $O(\triangle)$ iteration. For each iteration, it receives at most $O(\triangle^k)$ messages from h-hop neighbors. The time to update the topology is dominated by the time to test the graph connectivity which is $O(V + E)$. Because there are at most $O(\triangle^k)$ nodes from which n_i can receive the messages, the time to update the topology is at most $O(\triangle^{1+2k})$. The complexity of DLPA$_k$ is $O(\triangle^{2+3k})$. The memory space complexity of DLPA$_k$ is mainly dominated by storing the topology information, and the topology information for each node is at most $O(\triangle^{2k})$. This is because each node needs to memorize $O(\triangle^k)$ h-hop neighbor nodes. The message complexity can be summarized as follows: A sensor node receives at most $O(\triangle^k)$ messages for each iteration, and the length of each message is at most $O(\triangle^2)$. Therefore, during the waiting time, the total load of message transmission to one sensor node is at most $O(\triangle^{2+k})$. The whole network load is at most $O(N\triangle^{2+k})$ at a time.

5 Simulation and experiment results

In this section we present our simulation and experiment results. We analyze and compare the performance of the algorithms proposed under various parameters.

We compare these algorithms both in a small scale network in an experiment where the number of sensor nodes does not exceed 50, and in a large scale network where the number of sensor nodes is more than 100 by simulation. We use three types of sensor nodes, MICA2, MICAz and Jennic in the real experiment for the small scale network. The following parameters are considered:

1. The square and sensor nodes deployment. In a small scale network, the sensors are deployed in a 15 m×15 m area. And in a large scale network, the sensors are deployed in a 20 m×20 m area. The sensor nodes are randomly scattered within the square.

2. The number of sensor nodes N. We vary N to examine the scalability. In the small scale network, N is varied from 10 to 50 with incremental of 5. In the large scale network, it is in the range of 100 to 500 with incremental of 100.

3. The fault tolerance degree k. We vary k from 2 to 4 in this simulation.

4. The maximum sensor transmission range $R_i^{\max}$ for each sensor node n_i. To guarantee that the network is k-connected, we set $R_i^{\max}$ to be randomly 1–5 m both in the small scale network and in the large scale network.

5. The power attenuation exponent α and constant c in the large scale network simulation. We set α to be 2 and c to be 0.

6. The number of hops h in the distributed algorithm. We vary h from 1 to 3.

For each number N, the place of deployment for sensor nodes is the same for different tunable parameters. We consider the following performance metrics in our evaluation:

1. The total power consumption. This is the summation of power consumption of each sensor node, namely $p_{\text{total}} = \sum_{i=1}^{N} p_i$.

2. The power reduced rate. This is the ratio of the total power consumption and maximum power consumption. We compute the power reduced rate of the total power consumption as $r = 1 - \frac{p_{\text{total}}}{p_{\max} \times N}$. We use $R_{\max}$ to calculate $p_{\max}$.

Figure 1 shows the comparison of two centralized algorithms, GPA_k and CVV_k in small-scale network. In Figure 1(a) we compare the performance of total power consumption between GPA_k and CVV_k. When we increase the required connectivity k from 2 to 4 gradually, more power is needed for both algorithms. With the increase in the number of sensor nodes, the total power does not increase gradually. Especially, when the number is 10, its total power is larger than any other's. This is because in a fixed territory with less sensor nodes, a sensor node needs more power to reach other nodes. However, with more sensors, the total power tends to increase, but the power consumption for each sensor is reduced. So when the number of nodes varies from 25 to 50, the total power increases vibratingly in the small scale network. When the connectivity is fixed, GPA_k has larger power consumption than CVV_k. When the number of nodes is small, the difference of power consumption between GPA_k and CVV_k is slight. When the number of nodes is increased, the gap is enlarged. Figure 1(b) shows the reduced rate of the total power consumption. CVV_k has a larger reduced rate than GPA_k. All of the reduced rates increase when the number of sensor nodes rises. As the number of sensor nodes increases, all of the reduced rates reach up to 0.9. When connectivity is smaller, the reduced rates are larger in both GPA_k and CVV_k.

Figure 2 is the comparison of GPA_k and CVV_k in a large scale network. From Figure 2(a), we can see that CVV_k has better performance than GPA_k in respect of the total power consumption. The power consumption increases nearly linearly with the growth of the number of sensor nodes. When the connectivity increases, the total power consumption increases in both GPA_k and CVV_k. Figure 2(b) is the reduced rate comparison. All of the reduced rates of GPA_k and CVV_k with various connectivity and number of nodes are up to 0.88. The reduced rates of CVV_k are higher than the reduced rates of GPA_k with all kinds of connectivity. When the number of sensor nodes is 100, the reduced rates almost reach as high as 1 in both GPA_k and CVV_k. However, when the number of sensor nodes reaches 200, the reduced rates decrease. As the number of sensor nodes increases from 200, the reduced rates change slightly to

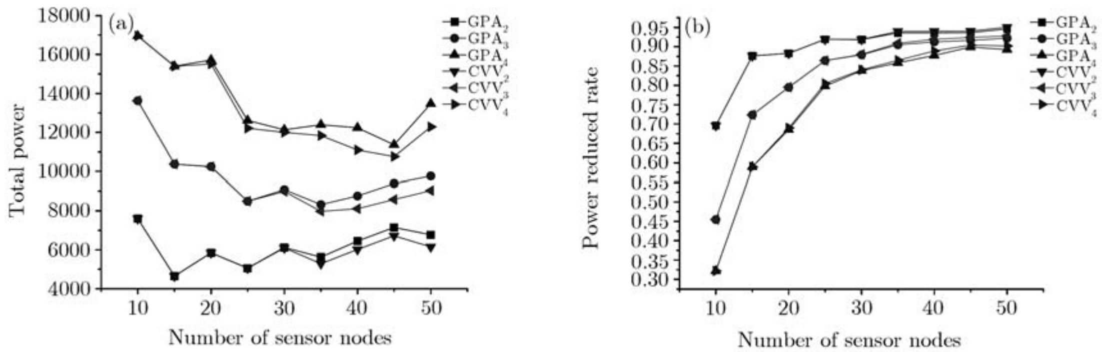

Figure 1 Comparison of GPA_k and CVV_k in small scale networks. (a) Total power; (b) power reduced rate.

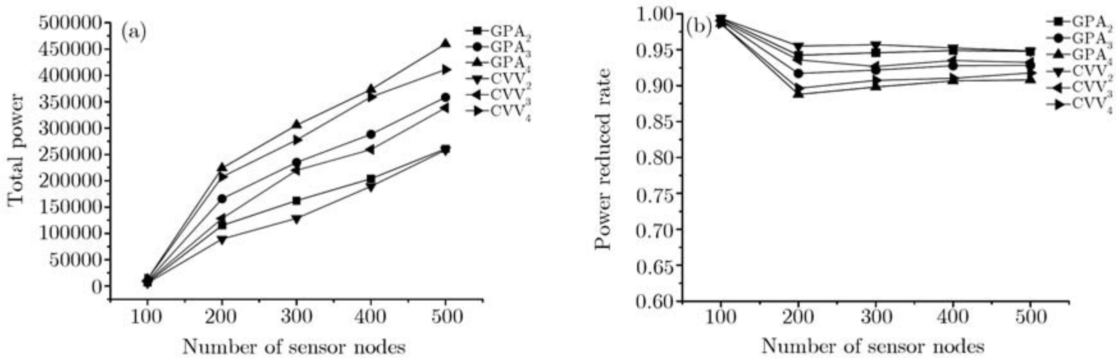

Figure 2 Comparison of GPA_k and CVV_k in large scale networks. (a) Total power; (b) power reduced rate.

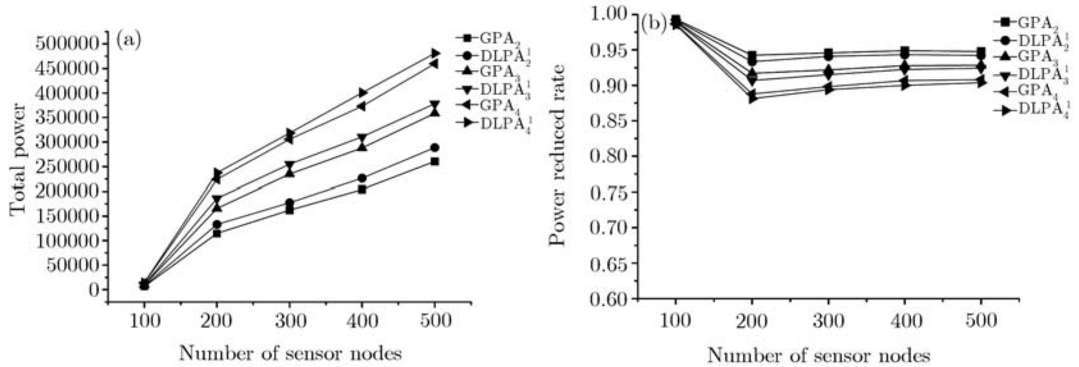

Figure 3 Comparison of GPA_k and DLPA_k. (a) Total power in large scale networks; (b) power reduced rate in large scale networks.

upswing or downswing. This is because of the indetermination of the relationship between the increased number of sensor nodes and the reduced power of each sensor nodes. However, the most important thing from the figure is that the change of power consumption is small when the number of sensor nodes is large.

Figure 3 is the comparison of GPA_k and DLPA_k in a large scale network. In this figure, we show the results by using DLPA_k when $h = 1$. The results are quite the same as the comparison of GPA_k and DLPA_k. Figure 3(a) shows the comparison of the total power consumption in large scale network. The total power consumption increases linearly with the growth of the number of sensor nodes in both GPA_k and DLPA_k. GPA_k has better performance than DLPA_k. Figure 3(b) shows its corresponding reduced rates of total power. The increase of power consumption in both GPA_k and DLPA_k is small while the number of sensor nodes is large. The small scale network experiment also shows that the total power consumption of GPA_k is smaller than the total power consumption of DLPA_k when k is 2, 3, 4 respectively. The total power consumption increases vibratingly with the growth of the number of sensor nodes. The increase of power consumption in both GPA_k and DLPA_k is small with the growth of the number of sensor nodes, while the initial power consumption increases linearly.

We also did an experiment to show the performance of DLPA_k with different values of h. We find that with the increase of h, the power consumption decreases. This is because with more hops of neighborhood information, it is more possible for a node to reach k-connectivity without increasing its power. Especially, when a node knows the global information, DLPA_k has the same performance as GPA_k. A larger value of h has greater reduced rate of power consumption.

We summarize the results as follows:

1. CVV_k has the best performance in terms of total power consumption.
2. Larger k requires larger power consumption.
3. When the number of sensor nodes increases, the total power consumption increases vibratingly in a small scale network, and increases linearly in a large scale network.
4. The increase (sometimes decrease) ratio of power consumption is small when the number of sensor nodes is large.
5. When h increases in DLPA_k, the total power decreases. A small value of h is sufficient to provide performance.

6 Conclusions

In this paper we proposed that the k-fault tolerant many-to-many routing power assignments in heterogeneous WSNs with the objective of providing k-vertex disjoint paths between any two sensor nodes while minimizing the total energy consumption.

We proposed three solutions to the FTPA_k problem, two centralized approaches, GPA_k and CVV_k, and one distributed and localized algorithm, DLPA_k. GPA_k is a greedy algorithm that minimizes the maximum power regarding all sensor nodes. CVV_k is an approximation algorithm with performance ratio $O(\sqrt{n/\epsilon})$. According to the theoretical analysis and simulation results, CVV_k has the best performance in terms of total power consumption. The approximation ratio is the best up to our knowledge. However, it is hard to use in practice when the number of sensor nodes is too large, because of its huge time complexity. GPA_k is a medial approach that has good performance in terms of total power consumption and power reduced rate. DLPA_k performs not as good as GPA_k or CVV_k. It consumes the most power. However, DLPA_k is a distributed and localized algorithm which is practical in heterogeneous WSNs, even in a large scale network.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 61003240), and Gansu Provincial Science & Technology Department (Grant No. 1007RJYA010).

References

1 Liu L, Hu B, Li L. Algorithms for energy efficient mobile object tracking in wireless sensor networks. Cluster Comput, 2010, 13: 181–197
2 Li L Y, Li C L. QoS multicast routing protocol in hierarchical wireless MANET. Sci China Ser F-Inf Sci, 2008, 51: 196–212
3 Tang Q L, Yang L Q, Qin T F, et al. Energy-saving PPM schemes for WSNs. Sci China Ser F-Inf Sci, 2008, 51: 571–585
4 Cardei M, Yang S, Wu J. Algorithms for fault-tolerant topology in heterogeneous wireless sensor networks. IEEE Trans Parall Distrib Syst, 2008, 19: 545–558
5 Liu L, Hu B, Li L. Energy conservation algorithms for maintaining coverage and connectivity in wireless sensor networks. IET Commun, 2010, 4: 786–800
6 Wang F, Thai M T, Li Y, et al. Fault tolerant topology control for all-to-one and one-to-all communication in wireless networks. University of Minnesota, Technical Report TR 05-038, 2005

7 Heizelman W R, Chandrakasan A, Balakrishnan H. Energy-efficient communication protocol for wireless micro sensor networks. In: IEEE Proceedings of the Hawaii International Conference on System Sciences, Maui, Hawaii, USA, 2000. 2: 3005–3014

8 Li N, Hou J C. FLSS: A fault-tolerant topology control algorithm for wireless networks. In: 10th Annual International Conference on Mobile Computing and Networking, Philadelphia, PA, USA, 2004. 275–286

9 Hajiaghayi M, Immorlica N, Mirrokni V S. Power optimization in fault-tolerant topology control algorithms for wireless multi-hop networks. IEEE/ACM Trans Network, 2007, 15: 1345–1358

10 Jia X, Kim D, Wan P, et al. Power assignment for k-connectivity in wireless ad hoc networks. J Combin Opt, 2005, 9: 213–222

11 Han X F, Cao X, Lloyd E L, et al. Fault-tolerant relay node placement in heterogeneous wireless sensor networks. IEEE Trans Mobile Comput, 2010, 9: 643–656

12 Li D Y, Wang Y X, Zhu Q H, et al. Fault-tolerant routing: k-inconnected many-to-one routing in wireless networks. Theoret Comput Sci, 2010, doi:10.1016/j.tcs.2009.06.035

13 Segal M, Shpungin H. On construction of minimum energy k-fault resistant topologies. Ad Hoc Networks, 2009, 7: 363–373

14 Ghosh A, Wang Y, Krishnamachari B. Efficient distributed topology control in 3-dimensional wireless networks. In: The 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), San Diego, California, USA, 2007. 91–100

15 Moraes R, Ribeiro C C, Duhamel C. Optimal solutions for fault-tolerant topology control in wireless ad hoc networks. IEEE Trans Wirel Commun, 2009, 8: 5970–5981

16 Ingelrest F, Simplot-Ryl D, Stojmenovic I. Optimal transmission radius for energy efficient broadcasting protocol in ad hoc and sensor networks. IEEE Trans Parall Distrib Syst, 2006, 17: 536–547

17 Perkins C E, Belding-Royer E M, Das S. Ad hoc on demand distance vector (AODV) routing. RFC3561, The Internet Society, 2003

18 Cheriyan J, Vempala S, Vetta A. Approximation algorithms for minimum-cost k-vertex connected subgraphs. In: 34th Annual ACM Symposium on the Theory of Computing. New York: ACM, 2002. 306–312

19 Kortsarz G, Nutov Z. Approximating node connectivity problems via set covers. In: 3rd International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX), Lecture Notes in Computer Science, Berlin: Springer, 2000, 1913/2000: 3–21