

## *Research Article*

# **An Immune Cooperative Particle Swarm Optimization Algorithm for Fault-Tolerant Routing Optimization in Heterogeneous Wireless Sensor Networks**

**Yifan Hu,<sup>1,2</sup> Yongsheng Ding,<sup>1,2</sup> and Kuangrong Hao<sup>1,2</sup>**

<sup>1</sup> College of Information Sciences and Technology, Donghua University, Shanghai 201620, China

<sup>2</sup> Engineering Research Center of Digitized Textile and Fashion Technology, Ministry of Education, Donghua University, Shanghai 201620, China

Correspondence should be addressed to Yongsheng Ding, ysding@dhu.edu.cn

Received 26 May 2011; Accepted 3 August 2011

Academic Editor: Jinling Liang

Copyright © 2012 Yifan Hu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The fault-tolerant routing problem is important consideration in the design of heterogeneous wireless sensor networks (H-WSNs) applications, and has recently been attracting growing research interests. In order to maintain  $k$  disjoint communication paths from source sensors to the macronodes, we present a hybrid routing scheme and model, in which multiple paths are calculated and maintained in advance, and alternate paths are created once the previous routing is broken. Then, we propose an immune cooperative particle swarm optimization algorithm (ICPSOA) in the model to provide the fast routing recovery and reconstruct the network topology for path failure in H-WSNs. In the ICPSOA, mutation direction of the particle is determined by multi-swarm evolution equation, and its diversity is improved by immune mechanism, which can enhance the capacity of global search and improve the converging rate of the algorithm. Then we validate this theoretical model with simulation results. The results indicate that the ICPSOA-based fault-tolerant routing protocol outperforms several other protocols due to its capability of fast routing recovery mechanism, reliable communications, and prolonging the lifetime of WSNs.

## **1. Introduction**

The complex networks have attracted growing research interests in topology structure and dynamic problems. Many kinds of system can be described with the complex network model, and these models are constructed by several nodes connected with each other, such as the Internet and the wireless sensor networks (WSNs). Due to the ability of collecting data from the environment and reporting it back to the sink without human supervision [1, 2], WSNs, especially heterogeneous ones, have come to pervade every aspect of our lives, such as habitat monitoring, industrial sensing, and traffic control [3–5]. The heterogeneous WSNs always

deploy an appropriate number of heterogeneous wireless sensor nodes (called macronodes), which contain devices with more capabilities, storage space, and energy than ordinary nodes. They cannot only improve the success rate of data transmission of WSNs, but also reduce the energy consumption of transmission, thus can effectively prolong the network lifetime. The benefits of using heterogeneous WSNs (H-WSNs) have been presented in the literature [6–9]. It is reported that when properly deployed, heterogeneity can triple the average delivery rate and provide a fivefold increase in the network lifetime [6].

However, in practical applications, unpredictable events such as environmental impairment, communication link broken, and battery depletion may cause the sensor devices to fail, partitioning the network and disrupting network functions. Therefore, fault tolerance becomes a critical issue for the successful communication of H-WSNs. It is expected that the network topology broken by software or hardware failure of sensor nodes could be automatically reconstructed and self-healed by the fault-tolerant routing technology so as to be recovered from path failure and ensured the performance of the communication tasks.

The objective of this paper is to solve the fault-tolerant routing problem for the H-WSNs while maintaining  $k$  disjoint communication paths from each source sensor to the macronode it belongs to (called  $k$ -disjoint-path routing recovery problem). For this purpose, we propose a swarm intelligence algorithm, immune cooperative particle swarm optimization algorithm (ICPSOA), to provide fast recovery from path failure in the H-WSNs. In this way, the network can tolerate the failure of up to  $k - 1$  sensor paths with reconstructed topology, traditional retransmissions can be decreased, and reliability can be provided with lower energy consumption. Our problem is specifically tailored to the situation that data is forwarded from sensors to macronodes.

The main contributions of this paper are as follows: firstly, we formulate the  $k$ -disjoint-path routing recovery problem for the H-WSNs. Then, in order to maintain  $k$  disjoint communication paths from each source sensor to the set of macronodes, we propose the ICPSOA-based protocol to reconstruct the network topology and provide fast routing recovery from path failure in the H-WSNs. The proposed method can provide simplicity, robustness, and effectiveness for routing recovery problem of the WSNs.

The remainder of this paper is organized as follows: Section 2 overviews the related work on fault-tolerant routing problem, especially routing recovery problem in the H-WSNs. Then, we propose the H-WSNs architecture and fault model and the ICPSOA-based approach for solving the routing recovery problem in Section 3. The simulation results are presented in Section 4. Section 5 provides a conclusion of our paper and discusses a few future directions for further improving the performance of our approach.

## **2. Related Work**

### ***2.1. The Fault-Tolerant Routing Algorithms of WSNs***

Fault-tolerant routing protocols proposed for WSNs can be classified into three groups: (1) proactive routing, called disjoint multipath, in which several paths from source node to sink are calculated, maintained in advance, and stored in a routing table, but greater energy consumption and the requirement to predict the global topology information are the disadvantages [10–12], (2) reactive routing, where all paths are created on demand [7], and (3) hybrid routing, which is a mix of the above two groups [8, 9].

One of the common fault-tolerant routing solutions is to establish disjoint multipath with proactive routing mechanism. Disjoint multipath constructs a number of alternative

paths which are node/links disjoint with the primary path and other alternative paths. Thus, a failure in any or all nodes/links on the primary path does not affect the alternative paths. Using this multipath scheme in a network with  $k$  disjoint paths from source to destination can tolerate at most  $k - 1$  intermediate network component failures. A secure and energy-efficient multipath routing protocol proposed by Nasser and Chen [10] is effectively resistive to some specific attacks, and has the character of pulling all traffic through the malicious nodes by advertising an attractive route to the destination.

A considerable amount of work has also been done on the hybrid routing scheme, which combines multipath scheme and reactive routing scheme. In this scheme, multiple paths are calculated and maintained in advance, and then, alternative paths are created on demand. EARQ (energy-aware routing for real-time and reliable communication) is a hybrid routing scheme proposed by Heo and Hong [9], which selected a path that expended less energy than others, among paths that delivered a packet in time, which enabled even distribution of energy expenditure to sensor nodes. It also provided reliable communication and fast recovery from path failure, because it only sent a redundant packet via an alternative path if the reliability of a path was less than a predefined value. Pandana and Liu [11] sought to propose an algorithm which designed the connectivity weight of each node and established a most reliable path in order to keep the other nodes' connectivity.

Our work differs from the above existing ones [13–16] by considering a different architecture and routing objective. We consider the H-WSNs architecture with a number of macronodes and concern with providing  $k$ -connectivity from each source node to the set of macronodes, and as such, we provide a hybrid routing scheme to maintain the multipath routing. The H-WSNs usually consist of two types of wireless devices [12]: a large number of resource-constrained wireless sensor nodes deployed randomly and a much smaller number of resource-rich macronodes placed at known locations. The macronode network, which provides more energy, transmission bandwidth, computing ability, and storage space, is used to quickly forward sensor data packets to the sink. With this setting, data gathering in the H-WSNs has two steps. Firstly, sensor nodes transmit and relay information on multihop paths toward any macronode. Then, it is forwarded to the sink using fast macronode-to-macronode communication once a packet encounters a macronode.

The similar hybrid routing schemes for the H-WSNs are as follows: CPEQ (cluster-based periodic, event-driven, and query-based protocol) [17] groups sensor nodes to efficiently relay data to the sink by uniformly distributing energy dissipation among the nodes. It can provide fast broken path reconfiguration and high reliability in the delivery of event packets and speed up new subscriptions by using the reverse path. Cardei and Yang [18] proposed  $GATC_k$  and  $DATC_k$  in the H-WSNs, with the objective of minimizing the total energy consumption while providing  $k$  independent paths from each node to macronodes. Such a topology provides the infrastructure for fault-tolerant data-gathering applications robust to the failure of up to  $k - 1$  sensors. Boukerche et al. [8] used a protocol of ICE (intercluster communication-based energy-aware and fault-tolerant protocol) by alternating the nodes responsible for intercluster communication inside one cluster. If one of multiple paths has faulty nodes, the other ones will be used for the event notification's propagation. But the fast routing recovery mechanism for path failure has rarely been considered. Further, as the fault-tolerant optimization problem to find the optimal routing is NP-hard, these heuristic deterministic methods would always get the likely optimal routing result, and is easy to fall into local optimum. So, we employ a swarm intelligence algorithm, the ICPSOA, to improve the performance of solving these problems.

In this paper, we propose an ICPSOA-based fault-tolerant routing algorithm, which reconstructs the network topology of H-WSNs and provides a fast recovery from path failure with alternative path. We also compare the performance of the protocols of EARQ, CPEQ, and ICE with that of our approach. As we know, EARQ is an effective fault-tolerant routing protocol for homogeneous WSNs, while ICE and CPEQ are for H-WSNs to provide routing recovery from path failure. In this way, we can evaluate the fault-tolerant routing recovery mechanism with different network types.

## 2.2. SPSOA, CPSOA, and ICPSOA

The EA-based bionic randomized algorithm has become the important tools for solving complex optimization problems because of its intelligence and widely used and global search ability. But the algorithm dealing with fault-routing problem of WSNs should support the characteristic of energy saving. In general, better fault-tolerant performance always needs more energy consumption. Therefore, we choose light-weight algorithm based on the particle swarm optimization algorithm (PSOA), which has a simple structure and is easy to realize.

The PSOA is a new EA based method to search an optimal solution in the high-dimensional problem space [13], where each particle is a potential solution to the problem under analysis. In updating a population of particles with regard to their internal position and velocity, the PSOA is informed by the experiences of all the particles. It provides an idea to find solutions to complex problems using group advantage without global model and centralized control and can be suitable to apply in a dynamical modeling environment. It has been applied to many optimization problems, such as control problems and protocol design [14]. A remarkable difference between the PSOA and other EA-based algorithms is that the PSOA is very simple and has few parameters to be adjusted. Therefore, in general, it requires less computational complexity.

In the standard PSOA (SPSOA), each particle is a potential solution to the problem. Assume  $N$  particles fly in the  $D$ -dimensional search space, the position of the  $i$ th particle is  $x_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{iD}^t)^T$ , and its velocity is  $v_i^t = (v_{i1}^t, v_{i2}^t, \dots, v_{iD}^t)^T$ .  $p_i = (p_{i1}^t, p_{i2}^t, \dots, p_{iD}^t)$  is the best previous position of the particle, and  $p_g$  is the global best position of the whole particle swarm. Therefore, the velocity and position of each particle will be updated according to [15]

$$\begin{aligned} v_{id}^{t+1} &= wv_{id}^t + c_1 \text{rand}_1 (p_{id}^t - x_{id}^t) + c_2 \text{rand}_2 (p_{gd}^t - x_{id}^t), \\ x_{id}^{t+1} &= x_{id}^t + v_{id}^{t+1}, \end{aligned} \quad (2.1)$$

where  $1 \leq d \leq D$ ,  $c_1$  and  $c_2$  are learning factors, and usually, we make  $c_1 = c_2 = 2$ ;  $w$  is the inertia weight and used to control the tradeoff between the global and local exploration ability of the swarm. Random numbers  $\text{rand}_1$  and  $\text{rand}_2$  are uniformly distributed in  $[0, 1]$ .

The SPSOA also exhibits several disadvantages: it sometimes poses the problem of converging to undesired local optimum, for the diversity of population decreases in the latter iteration of evolution; optimizing stops when reaching a likely optimal solution, and thus the accuracy of the algorithm is limited. Therefore, a cooperative PSOA (CPSOA), which uses cooperative behavior of multiple swarms to improve the SPSOA, is proposed in [16]. In the CPSOA, limitation of an individual can be compensated by a number of other individuals from other symbiotic groups in the interaction. It can avoid misjudgment caused by single

exchange of information [16]. However, it still uses the formula of the SPSOA to evolve. The trajectory of each particle is unable to yield high diversity of particles to increase search space. Therefore, the CPSOA may get a suboptimal solution.

For this reason, we draw on good diversity characteristic of immune mechanism and develop an immune CPSOA (ICPSOA), in which each particle is considered as an antibody. Particle clone is used to generate a new population with offspring. Mutation is used to diversify the search process. Immune restrain is considered to restrain the inferior ones in order to keep the stable population. Immune memory is used to store the feasible solutions [19]. The affinity between antibody and antigen can measure the optimal path, and the affinity between antibodies and antibodies can evaluate the diversity of population [20]. In the ICPSOA, mutation direction of the particle (called antibody) is determined by evolution equation, and its diversity is increased by immune mechanism [21]. Although the addition of the immune mechanism may add more time complexity to the system, the proposed ICPSOA largely improves the capability of jumping out of local optima. The use of the ICPSOA for the fault-tolerant routing problem in H-WSNs has been presented in the following sections.

### **3. Fault-Tolerant Routing Problem in H-WSNs Based on the ICPSOA**

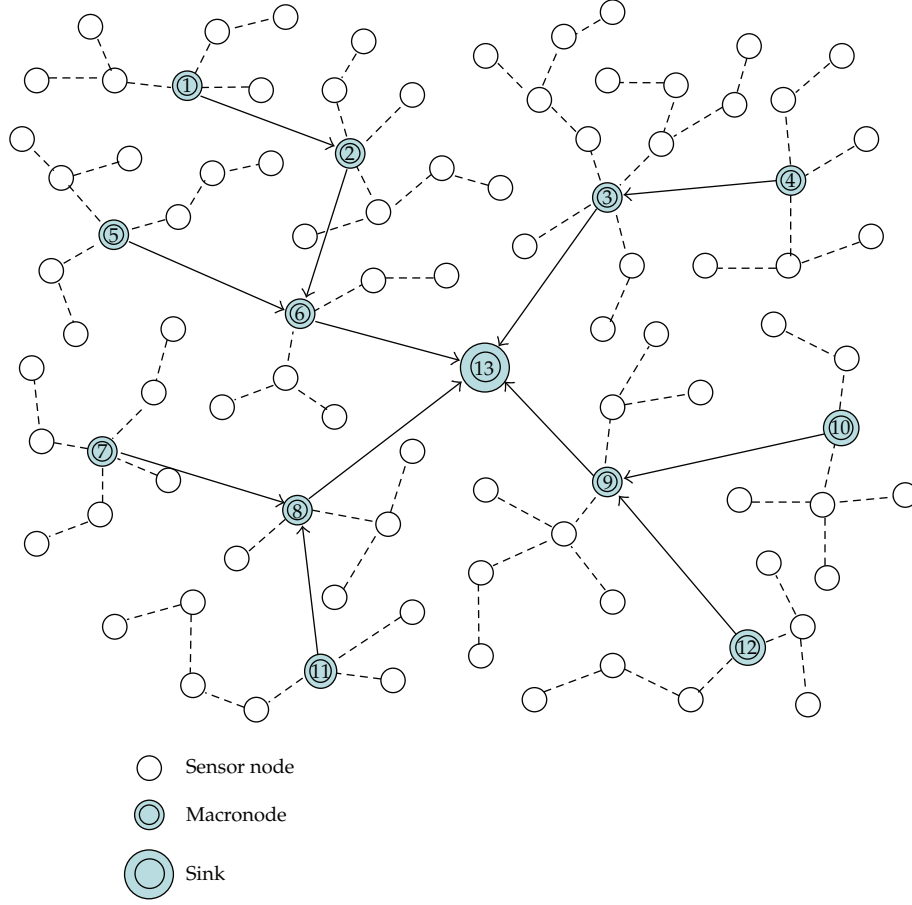
#### **3.1. Model of the Proposed H-WSNs**

##### *3.1.1. The Architecture for the Model of H-WSNs*

The architecture for the model of H-WSNs contains two types of wireless sensor devices as shown in Figure 1. The lower layer is formed by sensor nodes with constrained resource, including small amount of source nodes and other relay nodes. The main tasks performed by the source nodes are sensing, data processing, and data transmission. The tasks performed by the relay nodes are data processing and relaying. The dominant energy consumer is the radio transceiver. The upper layer consists of resource-rich macronodes overlaid on the H-WSNs. Wireless communication links between macronodes have considerably longer ranges and higher data rates, allowing the macronode network to bridge remote regions of the interest area. The tasks performed by a macronode are data aggregation and transmission, complex computations, and decision making. The ICPSOA is also executed by macronodes.

Therefore, in-network data transmission can be performed by forming a spanning tree among all the tree nodes. As shown in Figure 1, transmission starts with the leaf nodes (source node) of the tree sending their values to their parent nodes (macronode, nodes 1, 2, ..., 12) in the tree, until the final data is obtained at the root node (sink, node 13). Thus, the overall architecture would have a tree of macronodes and then each macronode can serve as the root of a subtree of ordinary nodes. Here, we are only interested in the fault-tolerant routing between sensor-sensor and sensor-macronode communications.

Assume that the network has the following characters: (1) the H-WSNs is a static network, where the nodes will not move after deployment, (2) every node knows its own position and that of the macronodes and the sink. The location can be obtained by GPS or localization protocols for estimating the location of a node, (3) the wireless transmission energy of macronode can be adjusted based on the distance between the receiver and itself, (4) the adjacent nodes would acquire the state information of their 1-hop neighbors and the links between them through periodically broadcast. The meanings of used symbols is provided in Table 1.



**Figure 1:** The architecture of H-WSNs.

**Table 1:** The main symbols.

$ST_i$	The $i$ th subtree of the tree of the H-WSNs
$N_{ST_i}$	The number of the sensor nodes in $ST_i$
$P_{ST_i}$	The state information of all the nodes in $ST_i$
$n_s$	One of the source nodes in $ST_i$
$n_r$	The root node (macronode) in $ST_i$
$P(s, r)$	The set of all the possible paths between $n_s$ and $n_r$ in $ST_i$
$p_i(s, r)$	The $i$ th path between $n_s$ and $n_r$ in $ST_i$
$n_{fail}$	The failed relay node of $p_i(s, r)$ in $ST_i$
$n_{fail-c}$	The child node of $n_{fail}$ of $p_i(s, r)$ in $ST_i$
$n_{fail-p}$	The parent node of $n_{fail}$ of $p_i(s, r)$ in $ST_i$
$n_i$	The $i$ th sensor node of $p_i(s, r)$ in $ST_i$
$p_b$	The optimal path of $P(s, r)$ with optimal fitness in $ST_i$
$n_{p_b i}$	The $i$ th sensor node of $p_b$ in $ST_i$

### 3.1.2. The $k$ -Disjoint-Path Spanning Graph in the Subtree

The subtree  $ST_i$  of the network is modeled as a directed, connected graph  $G(V, E)$ , where  $V$  is a finite set of subtree nodes and  $E$  is the set of subtree edges representing connection between these nodes, where source node  $n_s \in V$  and macronode (root)  $n_r \in \{V - \{n_s\}\}$ .  $p_i(s, r)$  is a valid path between  $n_s$  and  $n_r$ , and  $P(s, r)$  is the set of all the paths  $p_i(s, r)$ .  $n(n \in p_i(s, r))$  represents a node in  $p_i(s, r)$ , and  $e(e \in p_i(s, r))$  represents direct edge between any two adjacent nodes in  $p_i(s, r)$ . Then, we can get the  $k$ -disjoint-path spanning graph in the subtree  $ST_i$ . The factors affecting the choice of path  $p_i(s, r)$  include (1) the available energy function of each node,  $\text{ene}(n)$ , (2) distance function of the edge between adjacent nodes,  $\text{dist}(e)$ , (3) energy consumption function,  $\text{ene}(e)$ , (4) communication delay function of the node,  $\text{delay}(n)$ . Then, these parameters can determine the fitness function of  $p_i(s, r)$ ,  $\text{fitness}(p_i)$

$$\begin{aligned} \text{fitness}(p_i) &= \frac{\sum_{n \in p_i(s, r)} \text{ene}(n)}{\omega_1 f_1 + \omega_2 f_2 + \omega_3 f_3}, \\ f_1 &= \frac{\sum_{e \in p_i(s, r)} \text{ene}(n)}{\sum_{e \in E} \text{ene}(n)}, \\ f_2 &= \frac{\sum_{n \in p_i(s, r)} \text{delay}(n)}{\sum_{n \in N} \text{delay}(n)}, \\ f_3 &= \frac{\sum_{e \in p_i(s, r)} \text{dist}(e)}{\sum_{e \in E} \text{dist}(e)}, \end{aligned} \quad (3.1)$$

where  $f_1$  is the ratio of the energy consumed by the edges of path  $p_i$  and the energy consumed by all the edges in the subtree,  $f_2$  is the delay of the edges and nodes of path  $p_i$  versus the delay of all the nodes in the subtree, and  $f_3$  is the distance of the edges of path  $p_i$  versus the distance of all the edges in the subtree.  $\omega_1, \omega_2, \omega_3$  are the weight of effective energy, delay and distance constraints in the fitness function, and  $\omega_1 + \omega_2 + \omega_3 = 1$ . We set  $\omega_1 = 0.4, \omega_2 = 0.2, \omega_3 = 0.4$ . The higher fitness value indicates the more suitable path.

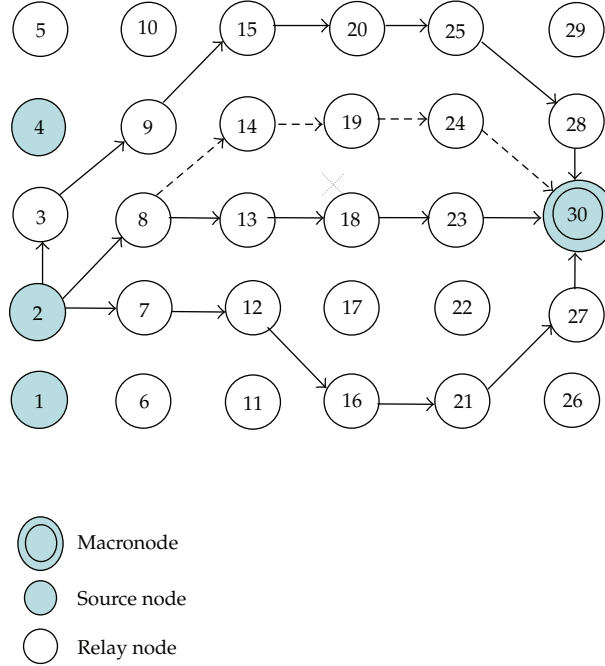
As illustrated in Figure 2, we assume  $k = 3$ , the three disjoint paths between source node (node 2) and root (node 30) are 2-3-9-15-20-25-28-30, 2-8-13-18-23-30, and 2-7-12-16-21-27-30, respectively. The detailed protocol dealing with routing recovery problem is presented in the following sections.

### 3.1.3. The Proposed Fault Model and Energy Model

We use the simple fault model proposed in [22] and identify the node failure in it. The fault-model should be simple enough to analyze, but also sophisticated enough to capture the fault behavior effectively. The probability of sensor nodes failure of subtree is given by  $p_{\text{node}}$ . As we use the more reliable macronodes to sustain the failure during transmission process, the probability of macronode failure is assumed to be  $p_{\text{macro}} \approx 0$ . If any of the sensor nodes fails, our routing recovery approach for node failure in subtree can be implemented.

We introduce the energy model adopted in [18], and the equation of energy model of a sensor node is as follows:

$$\text{ene}(m, d) = \text{ene}_{tx}(m, d) + \text{ene}_{rx}(m) = (a_{11} + a_2 d^m)m + a_{12}m, \quad (3.2)$$



**Figure 2:**  $k$ -disjoint-path spanning graph in subtree.

where  $d$  is the distance from the sensor node to the next-hop node,  $ene_{tx}(m, d)$  and  $ene_{rx}(m)$  are the energy consumption of sending and receiving  $m$  bits of data,  $a_{11}$ ,  $a_2$ , and  $a_{12}$  are energy consumption parameters of sending circuit, sending amplifier, and receiving circuit, and  $n$  is the channel attenuation index. We also define  $ene_{DF}$  as energy consumption of data fusion and  $ene_{RT}$  as energy consumption of updating routing table. For the ICPSOA, we define energy consumption of particle update, immune clone, mutation, particle selection, and restrain per iteration as  $ene_{PU}$ ,  $ene_{IC}$ ,  $ene_{IM}$ ,  $ene_{PS}$ , and  $ene_{PR}$ , respectively. So, the total consumption of the ICPSOA is according to the actual iterated generations per round.

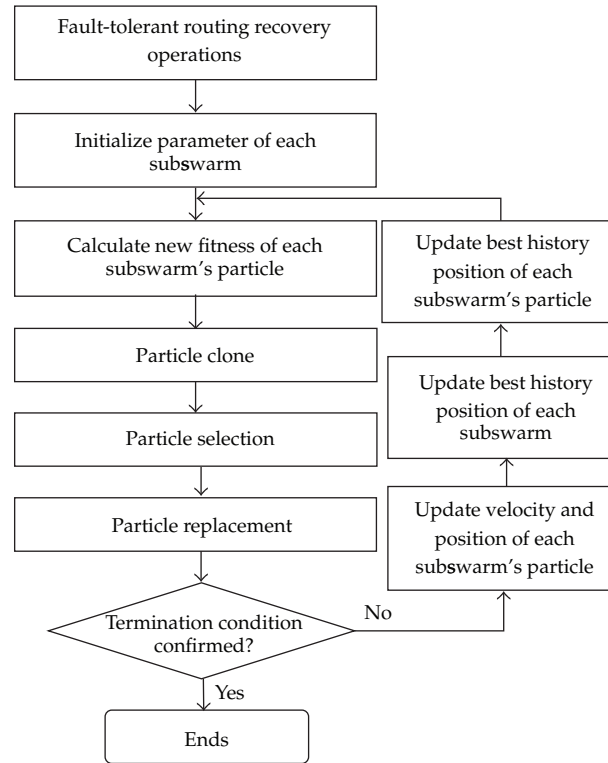
### 3.2. Fault-Tolerant Routing Problem Using the ICPSOA

As described in Section 2.2, the ICPSOA is used to provide a fast recovery mechanism from path failure due to physical damage or energy depletion with an alternative path. It chooses a path with optimal fitness from the optional sensor nodes. The ICPSOA is the kernel of our fault-tolerant routing protocol. Its flowchart is shown in Figure 3, and its framework is shown in Algorithm 1. The detailed procedures are described in the following subsections.

#### (1) Initialization

The principle of the ICPSOA is to search, respectively, in different  $D$ -dimensional target spaces using  $k$  independent particle swarms. To initialize the algorithm, we set the population size of particle  $n$ , the division factor  $k$ , and each particle swarm includes  $n/k$  particles. Then, the  $D$ -dimensional vector (vector of particle's position and velocity) is divided into  $k$  swarms. We define a matrix by  $[D \times 2n]$  to represent the initial particle swarm, in which the former  $n$  columns are the position of particle, and the latter  $n$  columns are the velocity of particle. In





**Figure 3:** The architecture of the ICPSOA for the H-WSNs.

**Input:**  $P_{ST_0}$ : The information parameters of nodes  
 $P_{Gen}$ : The iterated generations for searching process  
**Output:**  $fitness(p_i)$ : The global optimal fitness  
**Process:**  
**Step 1:** Initialization: Generate initial particle swarm parameter.  
**Step 2:** Immunization: Immune clone, mutation, particle selection and restrain.  
**Step 3:** If termination criterion conditions are satisfied, **go to Step 5**; else **go to Step 4**.  
**Step 4:** Update: Update the velocity and position of each sub-swarm and particle.  
**Step 5:** Output: Output the global optimal fitness of the particle swarm. **Ends**.

**Algorithm 1:** The ICPSOA for fault-routing problem.

Algorithm 2,  $b(g)$  is a complete vector function consisting of all subswarms' optimal position vector,  $x_m S_i$  represents position vector of the  $m$ th particle in the  $i$ th swarm,  $p_m S_i$  is the optimal history position vector of the  $m$ th particle in the  $i$ th swarm, and  $p_g S_i$  represents optimal experience position vector of the  $i$ th swarm.

## (2) Immunization

In this step, each particle can be considered as an antibody, resulting in the clonal mutation set  $C$ . The clone number and the fitness of particle are proportional. The clonal number  $N_c$  is

<b>Input:</b>	$P_{ST_0}$ : The information parameters of nodes $n$ : The population size of particle $k$ : The population size of swarm
<b>Output:</b>	$S_i$ : The vector of the $i$ th particle swarm $b(g)$ : Each sub-swarm's optimal position vector function
<b>Process:</b>	Particle's $D$ -dimensional vector is divided into $k$ particle swarms. $b(L, i) = (p_g S_1, \dots, p_g S_{i-1}, L, p_g S_{i+1}, \dots, p_g S_k)$

**Algorithm 2:** Initialization mechanism in the ICPSOA.

usually calculated as follows:

$$N_c = \alpha N, \quad (3.3)$$

where  $\alpha$  is the clone factor and is proportional to particle's fitness value.  $N$  is the number of particles. Mutation rule can be setup according to experience. The particle mutation rule for the function optimization problem is

$$c_i = x_i + \beta \text{rand}, \quad (3.4)$$

where  $c_i$  is the clonal individual,  $x_i$  is the original antibody,  $\beta$  represents the mutation factor, and rand is uniformly distributed in  $[0, 1]$ .

For the particles replacement rule, we need to calculate the antigen stimulus degree of the original particles and select clonal mutation particles. The Euclidean distance between any particle  $C_t$  and antigen  $Y_t$  is

$$d(i, j) = \sqrt{\sum_{i=1}^n (c_{it} - y_{jt})^2}. \quad (3.5)$$

Therefore, the stimulus degree of antibody particle is

$$A(i, j) = \frac{1}{d(i, j)}. \quad (3.6)$$

After that, each particle is compared with stimulus threshold; the higher one will maintain in the subswarm, and the lower one will be replaced (called restrain). The process of this step is as shown in Algorithm 3. Then, go to Step 2 in Algorithm 1.

### (3) Termination Criterion

If the solution is satisfied with the termination criterion, fitness( $p_i$ ) is the optimal fitness or  $P_{\text{Gen}}$  decreases to zero, the optimal path  $p_i$  will be the desired optimal solution, and this procedure ends; otherwise, returns to Step 4 in Algorithm 1. Then, the  $k$ th path is established.

**Input:**  $n$ : The population size of particles  
 $P_{\text{Gen}}$ : The iterated generations for searching process  
 $k$ : The population size of swarm

**Output:** The allele of the offspring's antibodies (particles)

**Process:**  
**For** each swarm  $i \in [1 \cdots k]$   
**For** each particle  $m \in [1 \cdots n/k]$   
Clone operation:  $N_c = \alpha N$   
Mutation operation:  $c_i = x_i + \beta \text{rand}$   
Replacement (restrain) operation:  

$$d(i, j) = \sqrt{\sum_{i=1}^n (c_{it} - y_{jt})^2}, \quad A(i, j) = 1/d(i, j)$$
If  $A(i, j) > \text{threshold}$ , the particle (antibody) is replaced  
**End For**  
**End For**

**Algorithm 3:** Immune mechanism in the ICPSOA.

#### (4) Update

In this step, the velocity and position of the particle is updated as (2.1). The process is as shown is Algorithm 4. The updating equation of particles' optimal position vector in each subswarm is as follows:

$$b(p_m S_i, i) = \begin{cases} b(x_m S_i, i), & \text{fitness}(b(x_m S_i, i)) \geq \text{fitness}(b(p_m S_i, i)), \\ b(p_m S_i, i), & \text{fitness}(b(x_m S_i, i)) < \text{fitness}(b(p_m S_i, i)), \end{cases} \quad (3.7)$$

where  $1 \leq i \leq k$ . The updating equation of optimal position of each subswarm is

$$b(p_g S_i, i) = \arg \max_{P(p_m S_i, i)} \text{fitness}(b(p_m S_i, i)), \quad 1 \leq m \leq \frac{n}{k}, \quad 1 \leq i \leq k. \quad (3.8)$$

Equation (3.8) indicates that the optimal position of the  $m$ th subswarm will select the personal optimal position with the optimal fitness of particle in the swarm.

Inertia weight  $w$  plays an important role to the convergence of the result among the adjustable parameters. The larger weight can help the particle escape from the local best solution, and the smaller one is better for the convergence, thus the inertia weight can achieve balance between global search and local search. To overcome the limitations of other general strategies, the linear differential decreasing strategy is used [23]. Here, we select  $w_{\text{start}} = 0.85$ ,  $w_{\text{end}} = 0.35$

$$\begin{aligned} \frac{dw(t)}{dt} &= \frac{2(w_{\text{start}} - w_{\text{end}})}{t_{\text{max}}^2} t, \\ w(t) &= w_{\text{start}} - \frac{(w_{\text{start}} - w_{\text{end}})}{t_{\text{max}}^2} t^2. \end{aligned} \quad (3.9)$$

<b>Input:</b>	$P_{ST_0}$ : The information parameters of nodes $S_i$ : The $i$ th particle swarm $b(L, i)$ : Each sub-swarm's optimal position vector function $P_{Gen}$ : The iterated generations for searching process
<b>Output:</b>	$v_{id}^{t+1}$ and $x_{id}^{t+1}$ : The velocity and position of each sub-swarm and its particle $fitness(p_i)$ : The global optimal fitness
<b>Process:</b>	<b>For</b> each swarm $i \in [1 \cdots k]$ <b>For</b> each particle $m \in [1 \cdots n/k]$ Update velocity and position of each sub swarm's particle <b>If</b> $fitness(b(x_m S_i, i)) \geq fitness(b(p_m S_i, i))$ , $b(p_m S_i, i) = b(x_m S_i, i)$ <b>Else If</b> $fitness(b(x_m S_i, i)) < fitness(b(p_m S_i, i))$ , $b(p_m S_i, i) = b(x_m S_i, i)$ Calculate $fitness(p_i)$ of the particle (path) <b>End For</b> $b(p_g S_i, i) = \arg \max_{P(p_m S_i, i)} fitness(b(p_m S_i, i))$ <b>End For</b>

**Algorithm 4:** Update mechanism in the ICPSOA.

The computational complexity is an important issue in designing our optimization algorithms. In the  $n$ th iteration of the ICPSOA, the time to calculate fitness function for immune clone is  $N_c$ , the time to calculate fitness function for particle mutation and selection is  $\beta N_c$ , and the time to calculate fitness function for particle update is  $tN$ . So the total calculating time  $P_n$  in the  $n$ th iteration should be

$$P_n \leq [N_c + N_c \beta] + tN = (1 + \beta)N_c + tN. \quad (3.10)$$

Therefore, the computational complexity of the ICPSOA is  $O(N_c)$ , which indicates that the size of clone group has a direct impact on the search speed of the ICPSOA with the same size of particle.

### 3.3. The ICPSOA-Based Fault-Tolerant Routing Protocol Framework for H-WSNs

The ICPSOA is the kernel of fault-tolerant routing protocol. As shown in Figure 2, once  $n_{fail}$  (node 18) fails, the macronode  $n_r$  (node 30) constructs subgraph  $G'$  ( $G' \subset G$ ) according to the current topology information of nodes and extracts the set of nodes  $N_p$  which can be used to construct an alternative path  $p_i(s, r)$  from  $G'$ . Each node represents a particle, and the population size of particle is  $n$ . Some nodes of  $N_p$  can form a particle sequence  $\{n_{p_i1}, n_{p_i2}, n_{p_i3} \dots, n_{p_im}\}$  ( $m \leq n$ ) according to their order, which can construct a path  $p_i(s, r)$  from source  $n_s$  to  $n_r$ . The algorithm ICPSOA would optimize the particle sequence to obtain the optimal path  $p_b(s, r)$  with optimal fitness  $fitness(p_b)$ , and  $p_b(s, r)$  includes the following nodes  $\{n_s, n_{p_b1}, n_{p_b2}, \dots, n_{p_bm}, n_r\}$ . Each node owns a routing table recording the paths it belongs to and the nodes' information on these paths. We now demonstrate with an example how the routing recovery process is accomplished in our protocol. In the example (Figure 2),  $n_{fail}$ 's child node  $n_{fail-c}$  and parent node  $n_{fail-p}$  are node 23 and 13.

*Step 1.*  $n_{\text{fail}-c}$  reports the failure of  $n_{\text{fail}}$  to  $n_r$ , and  $n_{\text{fail}-p}$  reports the failure to  $n_s$  (node 2), then  $n_s$  starts up another backup path to transmit data.  $n_s$  broadcasts a path request (PR) packet, with routing table including its own available energy and coordinate.

*Step 2.* If an intermediate node  $n_i$  receives PR, it will relay the packet according to its own state of information: if  $n_i$  is on one of the other existed  $k - 1$  paths between  $n_s$  and  $n_r$ , it will ignore the packet; else,  $n_i$  will calculate  $\text{dist}(e)$ ,  $\text{ene}(e)$  and  $\text{delay}(e)$  between  $n_i$  and  $n_{i-1}$  according to the information provided by  $n_{i-1}$ . Then, it continues to relay packet RP, with routing table including above information,  $n_i$  and  $n_{i-1}$ 's ID, and its  $\text{ene}(n)$ .

*Step 3.* Each intermediate node in the subtree repeats Step 2 until  $n_r$  receives the PR. Then  $n_r$  extracts the information, calculates  $\text{fitness}(p_i)$  using the ICPSOA and selects the path  $p_b$  with optimal fitness. Then, it broadcasts packet RP\_ACK, including the IDs of selected nodes  $\{n_{p_b1}, n_{p_b2}, n_{p_b3}, \dots, n_{p_bn}\}$  on  $p_b$  (path 2-8-14-19-24-30 in Figure 2) in routing table.

*Step 4.* If  $n_{p_b i}$  has received PR\_ACK, it checks whether its ID is in the packet's routing table. Then it establishes a connection between child  $n_{p_b i-1}$  and parent  $n_{p_b i+1}$  and delivers PR\_ACK to parent until  $n_s$  receives the packet. Go to Step 5.

*Step 5.*  $n_s$  broadcasts packet PR\_END, and  $n_{p_b i}$  on  $p_b$  delivers it to  $n_r$ . Once  $n_r$  receives PR\_END, the  $k$ th path from source node to its root in the subtree is established, the network topology is reconstructed, and protocol ends.

During this process,  $n_r$  will broadcast one packet and receive three packets,  $n_s$  will broadcast two packets and receive two packets, a part of relay nodes  $n_i$  in the subtree will deliver four packets. We assume the number of hops of  $p_i(s, r)$  ( $i \in (1, k - 1)$ ) is  $N_{p_i(s, r)_i}$ , then the number of the packets received by  $n_r$  before running the ICPSOA is  $N = N_{\text{ST}_i} - \sum_{i=1}^{k-1} N_{p_i(s, r)_i}$ . In this way the energy consumption of packet receiving and broadcasting of  $n_r$  can be calculated.

## 4. Simulation Results

### 4.1. Simulation Model

To evaluate the performance of the ICPSOA, we design a corresponding simulation scenario upon Matlab. The simulation experiment is constructed on Windows XP with Intel Pentium 4 processor (2.4 GHz) and 2 GB RAM. The goal of the simulation is to show that the ICPSOA can provide a more stable transport environment in an error-prone network. The results are also compared to the protocols of ICE, CPEQ, and EARQ.

In Table 2, we present the parameters configured for the conducted simulation experiments. The sensor nodes are randomly deployed on area  $A$ , and the macronodes are located at known coordinates. 500 rounds are taken and five packets are delivered in each round. The size of the network is the same for different algorithms and the fitness function is then measured. The parameters used for the ICPSOA are function dimension  $D = 30$ , iterated generations  $P_{\text{Gen}} = 1000$ , division factor  $k = 5$ , clone factor  $\alpha = 4$ , mutation factor  $\beta = 0.5$ . According to the description of the ICPSOA in Section 3.1.3, we set energy consumption  $\text{ene}_{\text{PU}} = 80 \text{ pJ}$ ,  $\text{ene}_{\text{IC}} = 5 \text{ pJ}$ ,  $\text{ene}_{\text{IM}} = 10 \text{ pJ}$ ,  $\text{ene}_{\text{PS}} = 15 \text{ pJ}$ ,  $\text{ene}_{\text{PR}} = 10 \text{ pJ}$  per iteration. The metrics that we use in our experiments are average number of alive nodes per round, average energy depletion ratio per round (measured as the energy dissipation versus

**Table 2:** Simulation parameters.

Simulation parameter	Value
Network area $A$	10000 m $\times$ 10000 m
Number of sensor node $n_{\text{node}}$	50–500
Number of macronode $n_{\text{macro}}$	10
Available energy on sensors $E_{\text{node}}$	120 J
Sensing radius of sensor node $R_{\text{node-s}}$	300 m
Communication radius of sensor node $R_{\text{node-c}}$	600 m
Communication radius of macronode $R_{\text{macro-c}}$	3500 m
Bandwidth of sensor node $B_{\text{node}}$	250 kb/s
Number of disjoint multipath $k$	3
Number of simulation rounds	500
Number of packet in each round	1
Size of packet in each round $m$	400 bits
Probability of node failure $p_{\text{node}}$	0.02
Energy consumption of sending circuit $a_{11}$	40 nJ/bit
Energy consumption of receiving circuit $a_{12}$	80 nJ/bit
Energy consumption of sending amplifier $a_2$	200 pJ/bit/m <sup>2</sup>
Channel attenuation index $n$	2
Energy consumption of data fusion $e_{\text{neDF}}$	4 nJ/bit
Energy consumption of updating routing table $e_{\text{neRT}}$	2 nJ/bit

the initial energy), average delay of packet delivery, and packet delivery ratio (measured as the number of successfully delivered packet versus required packet).

#### **4.2. Evaluation of the Simulation Results**

To illustrate the effect of the proposed protocol, we take a snapshot during a simulation. Figure 4 shows a small area (2000 m  $\times$  2000 m), which illustrates a subtree with the existed three paths between source node (node 29) and macronode (node 18). We can see when an intermediate node (node 24) fails, source node immediately establishes an alternative path to connect the macronode in order to replace the previous 3rd path.

The simulation ends after 1000 rounds. We compare the number of alive nodes per round for these four protocols. As shown in Figure 5 for different network sizes, the number of nodes died in the ICPSOA, ICE, and CPEQ is less than EARQ over the same number of rounds. This is because comparing with heterogenous WSNs, all the nodes only need to transmit data to its root (macronode) of the subtree in the H-WSNs, which indirectly shortens the transmission distance between sensor nodes to the sink, and prolongs their lifetime. The fast routing recovery mechanism of the ICPSOA also makes its number of alive nodes 5% ~ 10% more than that of ICE, and CPEQ in the same rounds. Then, we would only compare the ICPSOA, ICE, and CPEQ with the same network style (H-WSNs) in the remaining simulation process.

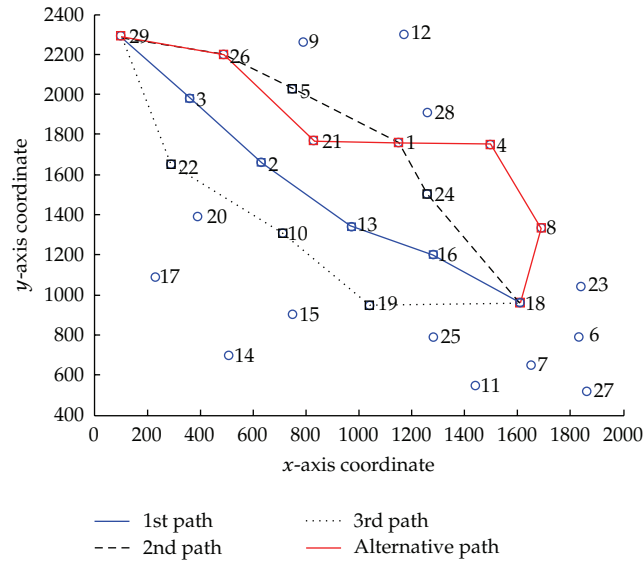


Figure 4: Snapshot of establishing the alternative path using ICPSOA.

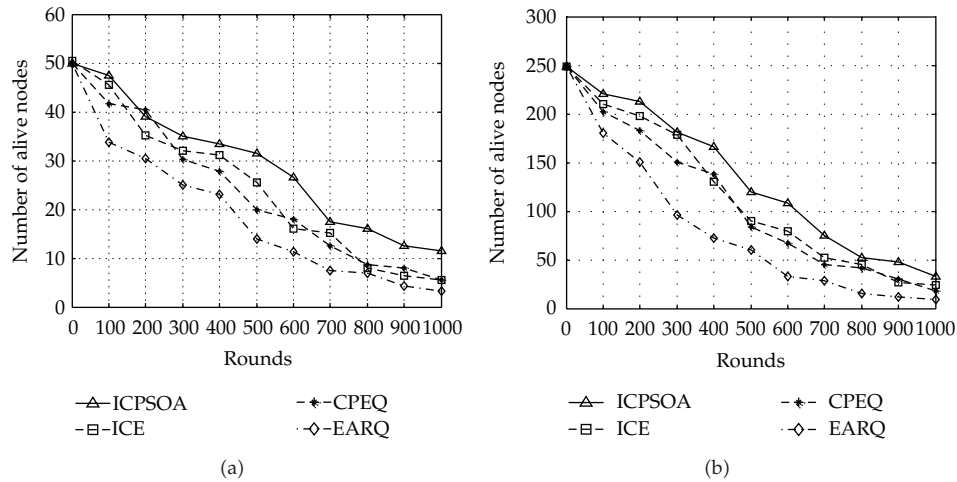
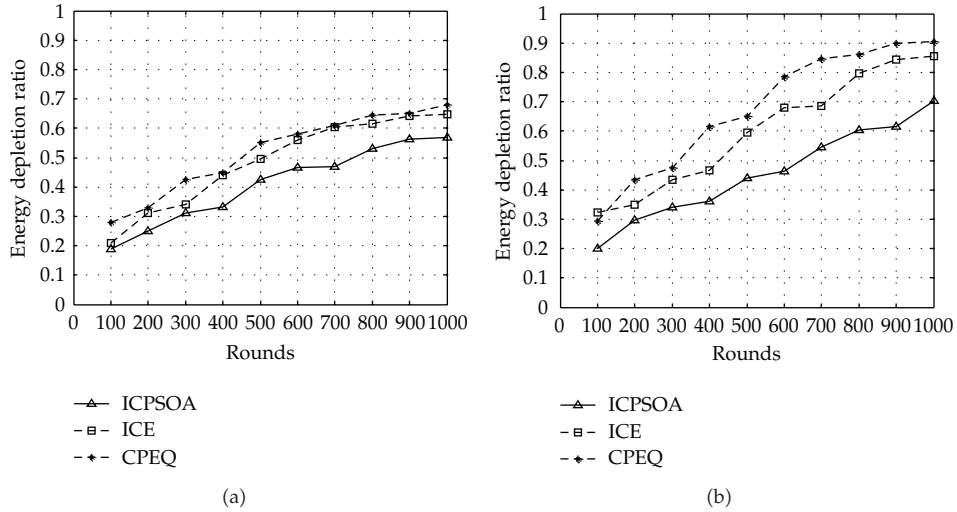
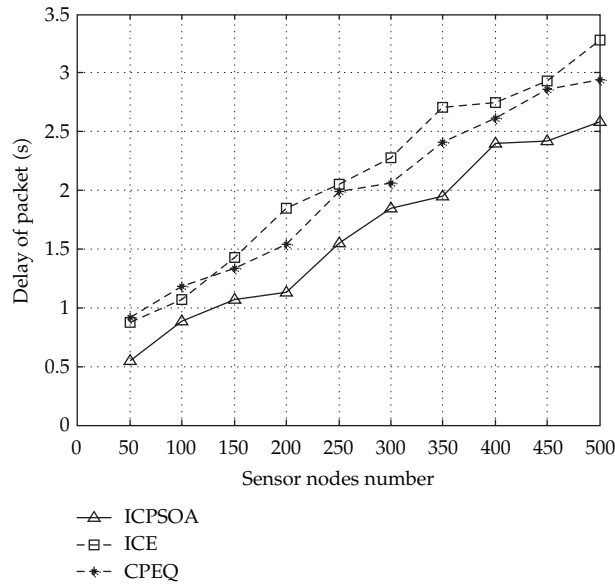


Figure 5: Experimental results for number of alive nodes per round with variant scale of network. (a) Size of 50 nodes. (b) Size of 250 nodes.

As shown in Figure 6 for different network sizes, energy depletion ratio of the ICPSOA based protocol is 5% ~ 15% smaller than that of ICE and CPEQ. And the dispersion between them is more obvious as the size of the cluster increases. That is because firstly, each source node has  $k$  paths to the macronode, and the total energy consumed is minimized; secondly, the ICPSOA can select the nodes with better QoS parameters (such as more available energy and less distance of path) to establish alternative path and construct a more reliable transmission environment to reduce the retransmission caused by unstable paths, therefore, prolong the network lifetime as compared to ICE and CPEQ.



**Figure 6:** Experimental results for energy depletion ratio per round with variant scale of networks. (a) Size of 50 nodes. (b) Size of 250 nodes.

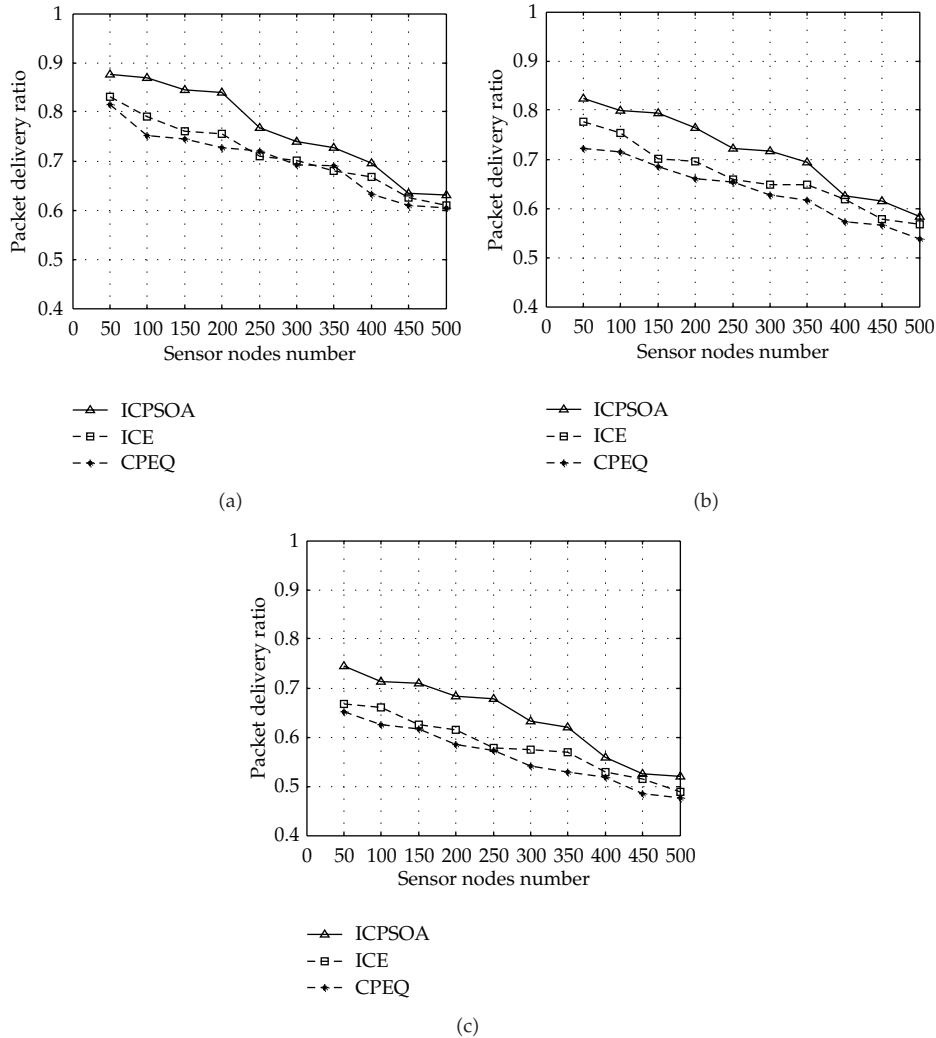


**Figure 7:** Experimental results for average delay of packet delivery.

Figure 7 shows the average delay of packet delivery (average delay of each packet delivered from source node to the sink). We can observe that the ICPSOA outperforms ICE and CPEQ in terms of average delay for the same networks. The ICPSOA has demonstrated a lower delay when network size grows. A low delay of packets can be explained by the multipath property and shortest alternative path selection of the proposed ICPSOA-based protocol for fault tolerance.

Figure 8(a) shows that the H-WSNs with the ICPSOA can deliver more packets to the sink than the network with EARQ and ICE with the value of 0.02 of  $p_{node}$ . In most cases,





**Figure 8:** Experimental results for average successful packet delivery ratio with variant failure probability. (a)  $p_{node} = 0.02$ . (b)  $p_{node} = 0.04$ . (c)  $p_{node} = 0.08$ .

the ICPSOA can send 5% ~ 15% more packets to the sink. A bigger value of packet delivery ratio indicates a lower packet dropout probability [24, 25] and a better network capability of delivering useful information. This result can be explained by the fact that the ICPSOA provides a fast recovery from path failure with an optimal alternative path, which improves the success rate of data transmission. Note that the packets in the ICPSOA experience a higher delivery ratio as the size of the network grows, which indicates the ICPSOA-based protocol of the H-WSNs is more feasible for practical deployment of large-scale WSNs than ICE and CPEQ.

We should also compare the performance trend of the three algorithms with different probability of node failure  $p_{node}$ . We plot the packet delivery ratio against the number of nodes in Figure 8 for various  $p_{node}$  (value of 0.02, 0.05, and 0.08). As shown in Figure 8(a), 8(b), and 8(c), the observed packet delivery ratio of the proposed schemes degrades as  $p_{node}$

ascends, which means the performance of proposed scheme is reduced as the percentage of failed nodes increases. But the ICPSOA can still deliver more packets than ICE and CPEQ for different size of the sensor network.

## 5. Conclusions

We propose the ICPSOA-based fault-tolerant routing protocol for H-WSNs, which focuses on a solution to the problem of energy depletion and packet delivery of nodes, by trying to reconstruct the topology structure and recover the routing for the path failure and achieve energy conservation by avoiding unnecessary retransmission. The conserved energy can be used to increase the quantity of information received by the sink. The experiment presents the promising ability of the ICPSOA, and better solutions of fault tolerance and prolonging the network lifetime can be obtained by the ICPSOA-based protocol than the protocols of EARQ, ICE and CPEQ. The results have illustrated the advantage of H-WSNs and backup disjoint multipath, which can reduce the risk of data delivery loss and energy consumption on the path exploring. It also aims at shortening delay of packet delivery, evening energy dissipation among the nodes by constructing the optimal alternative paths in the H-WSNs with the swarm intelligence algorithm. The strength of the ICPSOA is its simplicity, robustness and effectiveness for fast routing recovery compared to other approaches and makes the ICPSOA a potential solution to meet the requirements of critical conditions monitoring applications.

As for future studies, the following directions are under the way: firstly, the proposed protocol ignores the fault-tolerant routing between macronode-macronode communications, which could be considered to form a more complete protocol architecture; secondly, we should further reduce the computational complexity of the proposed ICPSOA such that it converges faster to a better solution, providing robustness against failure in the network.

## Acknowledgments

This work was supported in part by the Key Project of the National Nature Science Foundation of China (no. 61134009), the National Nature Science Foundation of China (no. 60975059), Specialized Research Fund for the Doctoral Program of Higher Education from Ministry of Education of China (no. 20090075110002), Specialized Research Fund for Shanghai Leading Talents, Project of the Shanghai Committee of Science and Technology (nos. 11XD1400100, 11JC1400200, 10JC1400200, 10DZ0506500).

## References

- [1] D. Wang, J. Zheng, Y. Zhou, and J. Li, "A scalable support vector machine for distributed classification in ad hoc sensor networks," *Neurocomputing*, vol. 74, no. 1–3, pp. 394–400, 2010.
- [2] J. Liang, Z. Wang, and X. Liu, "Distributed state estimation for discrete-time sensor networks with randomly varying nonlinearities and missing measurements," *IEEE Transactions on Neural Networks*, vol. 22, no. 3, pp. 486–496, 2011.
- [3] B. Shen, Z. Wang, and X. Liu, "A stochastic sampled-data approach to distributed H-infinity filtering in sensor networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 9, pp. 2237–2246, 2011.
- [4] B. Shen, Z. Wang, and Y. S. Hung, "Distributed consensus H-infinity filtering in sensor networks with multiple missing measurements: the finite-horizon case," *Automatica*, vol. 46, no. 10, pp. 1682–1688, 2010.

- [5] B. Shen, Z. Wang, Y. S. Hung, and G. Chesi, "Distributed H-infinity filtering for polynomial nonlinear stochastic systems in sensor networks," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 5, pp. 1971–1979, 2011.
- [6] C. Laukik, D. Alin, and S. Ranka, "Fault tolerant aggregation in heterogeneous sensor networks," *Journal of Parallel and Distributed Computing*, vol. 69, no. 2, pp. 210–219, 2009.
- [7] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.
- [8] A. Boukerche, A. Martirosyan, and R. Pazzi, "An inter-cluster communication based energy aware and fault tolerant protocol for wireless sensor networks," *Mobile Networks and Applications*, vol. 13, no. 6, pp. 614–626, 2008.
- [9] J. Heo and J. Hong, "EARQ: energy aware routing for real-time and reliable communication in wireless industrial sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 5, no. 1, pp. 3–11, 2009.
- [10] N. Nasser and Y. F. Chen, "SEEM: secure and energy-efficient multipath routing protocol for wireless sensor networks," *Computer Communications*, vol. 30, no. 11-12, pp. 2401–2412, 2007.
- [11] C. Pandana and K. J. R. Liu, "Robust connectivity-aware energy-efficient routing for wireless sensor networks," *IEEE Transaction on Wireless Communications*, vol. 10, no. 7, pp. 3904–3916, 2008.
- [12] X. Han and X. Cao, "Fault-tolerant relay node placement in heterogeneous wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 5, pp. 643–656, 2010.
- [13] C. H. Lin, J. L. Chen, and Z. L. Gaing, "Combining biometric fractal pattern and particle swarm optimization-based classifier for fingerprint recognition," *Mathematical Problems in Engineering*, vol. 2010, Article ID 328676, pp. 1–30, 2010.
- [14] T. Wimalajeewa and S. K. Jayaweera, "Optimal power scheduling for correlated data fusion in wireless sensor networks via constrained PSO," *IEEE Transactions on Wireless Communications*, vol. 7, no. 9, pp. 3608–3618, 2008.
- [15] T. S. Zhan and C. C. Kao, "Modified PSO method for robust control of 3RPS parallel manipulators," *Mathematical Problems in Engineering*, vol. 2010, Article ID 302430, 25 pages, 2010.
- [16] C. J. Lin, C. H. Chen, and C. T. Lin, "A hybrid of cooperative particle swarm optimization and cultural algorithm for neural fuzzy networks and its prediction applications," *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 39, no. 1, pp. 55–68, 2009.
- [17] A. Boukerche, R. W. N. Pazzi, and R. B. Araujo, "Fault-tolerant wireless sensor network routing protocols for the supervision of context-aware physical environments," *Journal of Parallel and Distributed Computing*, vol. 66, no. 4, pp. 586–599, 2006.
- [18] M. Cardei and S. H. Yang, "Algorithms for fault-tolerant topology in heterogeneous wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 4, pp. 545–558, 2008.
- [19] Y. S. Ding, Z. H. Hu, and W. B. Zhang, "Multi-criteria decision making approach based on immune co-evolutionary algorithm with application to garment matching problem," *Expert Systems with Applications*, vol. 38, no. 8, pp. 10377–10383, 2011.
- [20] Y. S. Ding, X. J. Lu, K. R. Hao, L. F. Li, and Y. F. Hu, "Target coverage optimization of wireless sensor networks using multi-objective immune co-evolutionary algorithms," *International Journal of Systems Science*, vol. 42, no. 9, pp. 1531–1541, 2011.
- [21] Z. Hu and Y. S. Ding, "Immune co-evolutionary algorithm based partition balancing optimization for tobacco distribution system," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5248–5255, 2009.
- [22] L. Chitnis and A. Dobra, "Aggregation methods for large-scale sensor networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 2, pp. 1–36, 2008.
- [23] S. Q. Ren and J. Park, "Fault-tolerance data aggregation for clustering wireless sensor network," *Wireless Personal Communications*, vol. 51, no. 1, pp. 179–192, 2009.
- [24] H. Dong, Z. Wang, D. W. C. Ho, and H. Gao, "Robust H-infinity fuzzy output-feedback control with multiple probabilistic delays and multiple missing measurements," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 4, pp. 712–725, 2010.
- [25] B. Shen, Z. Wang, H. Shu, and G. Wei, "Robust H-infinity finite-horizon filtering with randomly occurred nonlinearities and quantization effects," *Automatica*, vol. 46, no. 11, pp. 1743–1751, 2010.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

