Research Article

A Task Allocation Algorithm Based on Score Incentive Mechanism for Wireless Sensor Networks

Feng Wang,1,2 Guangjie Han,1,3 Jinfang Jiang,1 Wei Li,1 and Lei Shu3

1Department of Information & Communication Systems, Hohai University and Changzhou Key Laboratory of Special Robot and Intelligent Technology, Changzhou 213022, China
2School of Information Science & Engineering, Changzhou University, Changzhou 213164, China
3Guangdong Petrochemical Equipment Fault Diagnosis Key Laboratory, Guangdong University of Petrochemical Technology, Maoming 525000, China

Correspondence should be addressed to Guangjie Han; hanguangjie@gmail.com

Received 10 November 2014; Accepted 26 December 2014

Academic Editor: Jiongkun Hu

Copyright © 2015 Feng Wang 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.

A wireless sensor network (WSN) consists of many resource constraint sensor nodes, which are always deployed in unattended and hostile environment[1]. In recent years, WSNs have been widely used in many applications including military, industrial, household, medical, marine, and other fields, especially in natural disaster monitoring, early warning, rescuing, and other emergency situations[2]. During the applications, especially for some applications, for example, image processing, data fusion, and multimedia data compression, the real-time demand is higher and higher. Meanwhile, the task complexity of the applications is higher and higher. It is generally known that a single sensor node with limited resources is unable to complete a complex task. Therefore, it is especially important to find an appropriate task allocation scheme, which can efficiently apply a large amount of complex tasks to several collaborative nodes for distributed processing.

1. Introduction

WSNs consist of many tiny, light, and energy-limited sensor nodes, which are always deployed in unattended and hostile environment[1]. In recent years, WSNs have been widely used in many applications including military, industrial, household, medical, marine, and other fields, especially in natural disaster monitoring, early warning, rescuing, and other emergency situations[2]. During the applications, especially for some applications, for example, image processing, data fusion, and multimedia data compression, the real-time demand is higher and higher. Meanwhile, the task complexity of the applications is higher and higher. It is generally known that a single sensor node with limited resources is unable to complete a complex task. Therefore, it is especially important to find an appropriate task allocation scheme, which can efficiently apply a large amount of complex tasks to several collaborative nodes for distributed processing.

For some complex real-time application, it is requested that all the complex tasks should be successfully completed in time although some sensor nodes fail or are attacked by malicious nodes. This puts forward a higher requirement for task allocation algorithm design in WSNs. On the one hand, the sensor nodes are vulnerable to failure and malicious attacks. These failures will reduce the efficiency of the original task allocation algorithm and eventually interrupt the execution of the algorithm. On the other hand, the sensor nodes always cannot finish a complex task in time due to their limited resource and processing capacity. If one sensor node runs out of its energy, the task allocation algorithm needs to run again from the start, which wastes large amounts of energy and cannot satisfy the real-time requirement of the application. Many task allocation algorithms have proposed for WSNs[3–7]. However, they are not suitable for real-time complex task execution.
Based on the abovementioned problems, we propose a task allocation algorithm based on score incentive mechanism (TASIM) for real-time complex task execution in WSNs. The considered WSN consists of many heterogeneous sensor nodes, which have different initial energy levels and different task process speeds. The proposed TASIM can divide a complex task into several subtasks and allocate them on different sensor nodes for collaborative execution, which can efficiently reduce the energy consumption of a single sensor node and balance the energy consumption of the whole network. In addition, if a sensor node fails or dies, the uncompleted tasks on the sensor node can be migrated to another sensor node as soon as possible based on a task migration algorithm, which ensures that the tasks are completed in time, and accordingly prolong the network life.

The contributions of the paper are listed as follows.

(i) A score incentive mechanism is proposed for task allocation in WSN. A sensor node which successfully completes tasks can be rewarded by some scores, while another sensor node which is unable to finish tasks is punished by deducting some scores. Based on the scores, the uncompleted tasks on failure nodes can be timely migrated to other sensor nodes for further execution, which ensures that the complex tasks can be finished in time.

(ii) The concept of ranking domain is adopted in the process of task allocation. The sensor nodes are divided into different ranking domains based on their different resource levels and service abilities. Cluster members with different ranking domains can communicate with each other to complete the task cooperatively, which efficiently balance the sensor nodes’ energy consumption, shorten the algorithm’s running time, and prolong the network lifetime.

(iii) Heterogeneous network structure is suited for task allocation. Heterogeneous network is a new trend in the development of WSNs, where sensor nodes have different initial energy levels and different abilities of computation, communication, processing, and storing. Therefore, in the process of task allocation, the heterogeneity of sensor nodes should be taken into account.

The rest of this paper is organized as follows. Section 2 introduces related works about task allocation in WSNs. Section 3 gives system models including network model and task model. Section 4 presents the score incentive mechanism and the concept of ranking domain. Section 5 investigates details of TASIM. Section 6 presents simulation results. Finally, Section 7 concludes the paper.

2. Related Work

In recent years, many task allocation and scheduling algorithms have been proposed for WSNs, which can be classified into three categories: energy efficient, real-time, and hybrid task allocation algorithms. In the energy efficient task allocation algorithms, the limited energy of sensor nodes is taken into consideration to process the tasks with less power, which ultimately aims to save sensor nodes’ energy consumption and prolong network lifetime. In the real-time task allocation algorithms, task processing time is taken into consideration, which aims to process the tasks with less delay, while in the hybrid efficient task allocation algorithms, both energy efficiency and task processing time are taken into account to improve the performance of task allocation.

2.1. Energy Efficient Task Allocation Algorithms. In [8], Xie and Qin proposed a novel balanced energy-aware task allocation (BEATA) algorithm for heterogeneous networked embedded systems. BEATA algorithm aims at making the best trade-off between energy saving and schedule length. The trade-off is achieved by the defined energy-adaptive window (EAW). In EAW scheme, a sensor node with lower energy consumption and earlier finish time is chosen for task processing. However, the residual energy of sensor nodes is not taken into account for choosing processing nodes, which ultimately causes that some nodes are chosen frequently and die early. Therefore, BEATA cannot well balance energy consumption of the whole network.

In [9], an online task scheduling mechanism called CoRAI is proposed to allocate network resources of WSNs. Given the amount of available resources and the task set, CoRAI can determine the sampling frequency for all the tasks so that the frequencies of the tasks on each sensor are optimized subject to the previously evaluated upper-bound execution frequencies. However, CoRAI does not address mapping tasks to sensor nodes. In addition, CoRAI fails to explicitly discuss energy consumptions of sensor nodes.

In [10], a novel task allocation algorithm based on \( A^\star \) algorithm is proposed to allocate tasks. Considering the problem of sensor nodes’ limited energy, a greedy \( A^\star \) algorithm is proposed to optimize the complexity of \( A^\star \) algorithm. However, the sensor nodes conducting task allocation consume much more energy than ordinary sensor nodes. In addition, in task allocation phase, it is not easy to ensure the parallel processing of tasks. In [11], a novel task grouping method is proposed to ensure the parallel processing of tasks. However, how to group the children subtask which has several father tasks is not discussed in [11].

In [12], an energy-constrained task mapping and scheduling scheme called EcoMapS is proposed. First, a channel model is presented for single-hop WSN. Then, based on this channel model, communication and computation are jointly scheduled. Furthermore, a quick recovery algorithm is executed in case of sensor node failures. EcoMapS efficiently minimizes the schedule length subject to the energy consumption constraint in single-hop clustered WSNs. However, EcoMapS does not provide the guarantee of execution deadline; thus it is not suitable for real-time applications.

2.2 Real-Time Task Allocation Algorithms. In [13], Tian and Ekici proposed a task allocation algorithm for multiple heterogeneous WSNs, which is a cross-layer collaborative processing scheme. The algorithm is computing tasks in the application layer, while scheduling communication tasks on the MAC layer and network layer. Computation task and
communication task can be scheduled at the same time, which ensures the real-time performance of task allocation.

In [14], Li et al. proposed a task auction algorithm based on the contract net. Another auction-based strategy for distributed task allocation is proposed in [15]. The real-time distributed task allocation problem is formulated as an incomplete information reverse auction game. Then, based on winner determination protocol (WDP), an energy and delay efficient distributed winner determination protocol (ED-WDP) is proposed to select winner nodes to perform tasks. To the best of our knowledge, this is the first work that considers a completely distributed framework for auction-based task allocation with an effective distributed winner protocol for WSN applications.

In [16], a dynamic task allocation is proposed for multihop WSNs with low node mobility. The proposed algorithm incorporates a fast task reallocation algorithm to quickly recover from network disruptions, such as node or link failures. Since the algorithm runtime incurs considerable time delay while updating task assignments, an adaptive window size is introduced to limit the task processing delay. To the best of our knowledge, this is the first study for mobile multihop WSNs.

2.3. Hybrid Task Allocation Algorithms. In [17], Okhovvat et al. proposed a novel task allocation algorithm based on the queuing theory, which considers both the system energy optimization and the task completion time. The proposed algorithm can efficiently shorten the task processing time. However, in WSNs, all the sensor nodes are energy limited and the energy is hard to supply. The residual energy of sensor nodes decreases along with the increase of number of processing tasks. The task processing performance will be directly affected, which could not remain unchanged, while in [18], Zhu proposed another task allocation algorithm to minimize energy consumption, which may cause serious processing delay.

In [19], a self-adapted task scheduling strategy is proposed based on a discrete particle swarm algorithm. First, a multiagent-based architecture for WSNs is proposed and a mathematical model of dynamic alliance is constructed for task allocation. In the dynamic alliance, the task allocation is conducted based on the energy consumption of sensor nodes, the amount of network loads, and the execution time of tasks to achieve parallel processing of tasks.

In [20], an energy efficient real-time dynamic task allocation algorithm is proposed to deal with the failure cluster members, which not only considers the characteristics of the cluster members, but also considers the communication overhead and the task processing time. The algorithm minimizes the energy consumption of the whole network by reducing the number of activated cluster members. But this algorithm completely runs on the cluster head, which requires a higher energy level and processing ability of the cluster head.

In [21], Tian et al. developed an application-independent task mapping and scheduling solution, which not only provides real-time guarantee, but also implements dynamic voltage scaling mechanism to further optimize energy consumption. Using a novel multihop channel model and a communication scheduling algorithm, computation tasks and associated communication events are scheduled simultaneously with a dynamic critical-path scheduling algorithm. However, the proposed solution is not suitable for heterogeneous WSNs, where constituent components are not known a priori.

Another similar task allocation algorithm which concentrates on both energy saving and time constraint is proposed in [22]. The proposed algorithm is a two-phase task allocation technique based on queuing theory. In the first phase, tasks are equally assigned to sensor nodes to measure the service capabilities of them. In the second phase, tasks are specifically allocated to some sensor nodes according to their measured capabilities in such a way to reduce the total completion times of all the tasks in network. However, the residual energy of sensor nodes dynamically changes with the number of processing tasks, which results in dynamic capabilities of sensor nodes. Therefore, how to dynamically update capabilities of sensor nodes needs further research.

Since network dynamicity causes additional complexity in a task allocation system, in [23], a dynamic task allocation and scheduling (DTAS) framework is proposed, in which algorithm complexity, the corresponding runtime, and energy consumption are explicitly taken into account. First, a heuristic minimum hop count algorithm is designed which can effectively reduce problem complexity. Second, a self-learning process (SLP) based on a GA (Genetic Algorithm) is applied, so that multiple design objectives can be met. Intermediate results of SLP can be provided as temporary suboptimal solutions to cope with changing network conditions. The fitness function in SLP initially favors meeting the deadline requirement and, then, gradually leans towards a balanced solution between task execution time and network lifetime. Finally, to deal with sudden node or link failure events and to update the solutions in SLP, a fast task recovery algorithm (FTTRA) is designed to quickly reallocate faulty task assignments.

3. Network Model and Task Model

3.1. Network Model. The WSN considered in this paper is a cluster-based wireless sensor network, where all the sensor nodes are heterogeneous. That is, all the sensor nodes have different initial energy levels and different task process speeds. The adjacent sensor nodes form a cluster, which consists of two kinds of sensor nodes: cluster members and cluster heads. Each cluster head leads and controls several cluster members. The cluster heads can directly communicate with the base station. Also, the cluster heads can communicate with each other through multiple hop, while the cluster members can only directly communicate with their cluster heads or neighbor nodes. Compared with cluster members, the cluster head has a higher initial energy level and task process speed. In addition, the cluster heads are responsible for allocating tasks on their cluster members.

3.2. Task Model. A complicated task consists of several subtasks, which can be denoted by a Directed Acyclic Graph \( G = (T, E, w, l) \). As shown in Figure 1, a DAG consists of a set
of vertices $T$ representing the tasks that need to be executed and a set of directed edges $E$ representing dependency among tasks. Each task has an execution deadline $t_{dl}$. In order to ensure the smooth completion of the complex task, each subtask must be completed before its execution deadline. If there is a directed edge $e_{ij}$ from a vertex $t_i$ to another one, $t_j$, the execution of subtask $t_j$ needs the output result of the execution of subtask $t_i$. Therefore, the subtask $t_j$ needs to be processed after the completion of $t_i$. The weight value $w_i$ of a vertex refers to the amount of computations of the task. The weight value $l_i$ of a directed edge refers to the communication traffic between two tasks. For the sake of simplicity, we assume that the amount of computations approximately equals the amount of transmitted data, which follows a normal distribution.

For an edge $e_{ij}$, $t_i$ is called an immediate predecessor of $t_j$, $t_j$ is called an immediate successor of $t_i$. A task without immediate predecessors is an entry task and a task without immediate successors is an exit task. If one task $t_j$ is scheduled on one node while its immediate predecessor $t_i$ is scheduled on another node, a communication between the two nodes is required. Therefore, $t_j$ cannot start its execution until the communication is completed and the result of $t_i$ is received. If both $t_j$ and $t_i$ are assigned on the same node, the communication latency is considered to be zero. $t_j$ can start execution after $t_i$ is finished. In addition, we assume that a subtask can only be assigned on one sensor node, but each sensor node can process several subtasks.

4. Score Incentive Mechanism and Ranking Domain

Many task allocation algorithms have been proposed for cluster-based WSNs [18, 20]. We also proposed a distributed task allocation strategy for collaborative applications (DTAC) in cluster-based WSNs [24]. In order to further improve the efficiency of task allocation, balance the energy consumption of sensor nodes, and prolong the network lifetime, the concepts of score and ranking domain are introduced in the paper. The score is used to quantitatively assess task completion of sensor nodes by reward or punishment. The sensor nodes which successfully complete the task are rewarded by some scores, while some others which fail to complete the task are punished by the deduction of scores. Based on the scores, the sensor nodes are classified into different ranking domains, which are used to help sensor nodes to collaborate with each other.

4.1. Score Incentive Mechanism. In WSNs, there is no fixed or predefined infrastructure; sensor nodes are vulnerable to failure and malicious attacks, which easily run out of energy or lead to communication interrupt. Sensor nodes are divided into three types: stable nodes, unstable nodes, and death nodes. The sensor nodes which have enough resources without failure or malicious attack are stable nodes. Otherwise, the sensor nodes are unstable. It is assumed that (1) all the sensor nodes actively work on tasks to obtain more scores, and (2) unstable node can ask for other nodes' collaboration by exchanging their scores; thus their uncompleted tasks can be migrated on the collaborative nodes for further completion.

Definition 1. Basic score $\text{BasicScore}_{V_j}$, a basic score of a sensor node $V_j$ is the reward score obtained by the energy consumption per 1 joule:

$$\text{BasicScore}_{V_j} = \frac{1}{E_{\text{residual}}} \times \text{Score}_{V_j} \times \lambda (t),$$

$$\lambda (t) = \begin{cases} 1, & t_{idle} < t < t_{deadline}, \\ \frac{(t_{\text{deadline}} - t) + (t_{idle} - t)}{(t_{\text{deadline}} - t)}, & t_{idle} < t < t_{deadline}, \\ \frac{k(t_{\text{deadline}} - t) + (t_{idle} - t)}{(t_{\text{deadline}} - t)}, & t < t_{\text{deadline}} < t_{idle}, \end{cases}$$

where $E_{\text{residual}}$ is the residual energy of the sensor node $V_j$. $\text{Score}_{V_j}$ is the total score of the node $V_j$, $t$ is the task arrival time, $t_{\text{deadline}}$ is the deadline of the task, and $t_{idle}$ is the idle time of the sensor node. $k$ is an adjusting parameter. If the task is allocated by a cluster head, the basic score is calculated by the cluster head. If the task is migrated between cluster members, the basic score is calculated by the cluster member itself.

Definition 2. Energy consumption of task completion $C_{\text{total}}^{V_j}(T_i)$:

$$C_{\text{total}}^{V_j}(T_i) = C_{\text{comp}}^{V_j}(T_i) + C_{\text{comm}}^{V_j}(T_i),$$

Figure 1: A DAG task model.
where $C_{\text{comp}}^{V_j}(T_j)$ is the energy consumption of executing the task $T_j$, $C_{\text{comm}}^{V_j}(T_j)$ is the communication overhead of the sensor node $V_j$.

**Definition 3.** Reward score $\text{Score}_{\text{reward}}^{V_j}(T_j)$: the sensor node $V_j$ which successfully completes the task $T_j$ can be rewarded by some scores:

$$\text{Score}_{\text{reward}}^{V_j}(T_j) = \text{BasicScore}_{\text{rate}}^{V_j} \times C_{\text{total}}^{V_j}(T_j).$$  

**Definition 4.** Penalty score $\text{Score}_{\text{punish}}^{V_j}(T_j)$: the sensor node $V_j$ which fails to complete the task $T_j$ is punished by some scores:

$$\text{Score}_{\text{punish}}^{V_j}(T_j) = \eta \times \text{Score}_{\text{reward}}^{V_j}(T_j).$$

**Definition 5.** Total score $\text{Score}_{\text{total}}^{V_j}$: over a period of time, all the scores that a sensor node obtained are calculated by

$$\text{Score}_{\text{total}}^{V_j} = \sum_{T_j \in V_j(T)} \text{Score}_{\text{reward}}^{V_j}(T_j) - \sum_{T_j \in V_j(T)} \text{Score}_{\text{punish}}^{V_j}(T_j),$$

where $V_j(T)$ is the number of tasks allocated on node $V_j$, $T_j$ is the number of successfully completed tasks. $T_j$ is the number of unsuccessfully completed tasks. The total score of a sensor node is always nonnegative; therefore, if the total score is zero, it is no longer deducted.

### 4.2. Ranking Domain.

In the process of task allocation, the sensor nodes are selected based on their reliability, historical participation, energy consumption, residual energy level, and so forth. Therefore, in this section, the ranking domain is divided into different levels based on their service capabilities. A service capability level is named as a ranking domain. The service capabilities of sensor nodes in the same ranking domain are almost the same.

**Definition 6.** Trust of a sensor node $D_{V_j}$:

$$D_{V_j} = \frac{\text{Score}_{\text{total}}^{V_j}}{\sum_{T_j \in V_j(T)} \text{Score}_{\text{reward}}^{V_j}(T_j)}, \quad D_{V_j} \in [0, 1],$$

where $\sum_{T_j \in V_j(T)} \text{Score}_{\text{reward}}^{V_j}(T_j)$ is the sum of the reward scores of node $V_j$.

**Definition 7.** Participation of a sensor node $A_{V_j}$:

$$A_{V_j} = \frac{N_{V_j}}{N_{\text{CHead}}}, \quad A_{V_j} \in [0, 1],$$

where $N_{\text{CHead}}$ is the total number of tasks which are released by a cluster head. $N_{V_j}$ is the number of tasks allocated on the sensor node $V_j$.

**Definition 8.** Rate of residual energy $R_{V_j}$:

$$R_{V_j} = \frac{E_{\text{residual}}^{V_j}}{E_{\text{initial}}^{V_j}}, \quad R_j \in [0, 1],$$

where $E_{\text{initial}}^{V_j}$ and $E_{\text{residual}}^{V_j}$ are the initial and residual energy of node $V_j$, respectively.

**Definition 9.** Service capability of a sensor node $P_{V_j}$:

$$P_{V_j} = \mu D_{V_j} + \beta A_{V_j} + \gamma R_{V_j},$$

where $\mu + \beta + \gamma = 1$, $\mu \geq 0$, $\beta \geq 0$, $\gamma \geq 0$, $P_{V_j} \in [0, 1]$.

**Definition 10.** Ranking domain: sensor nodes can be divided into different levels based on their service capabilities. A service capability level is named as a ranking domain. The service capabilities of sensor nodes in the same ranking domain are almost the same.

In this paper, the sensor nodes can be divided into three ranking domains ($G_1, G_2, G_3$). First, a threshold Th is defined for the rate of residual energy. If the residual energy of a sensor node $R_{V_j}$ is lower than the threshold ($R_{V_j} < \text{Th}$), the sensor node is considered to be energy insufficient to complete the task. Therefore, the sensor node is divided in the third ranking domain $G_3$. Otherwise, the sensor node is considered to have enough service capacities for task completion. Based on specific environmental parameters, the rate of sensor nodes in the first ranking domain $\alpha$ can be obtained. If the number of the sensor nodes which have enough service capacities is less than 2, they all are divided in the first ranking domain.

The cluster head is responsible for dividing ranking domains of cluster members. For each sensor node, there is a node list, in which the node ID, the residual energy level, the total scores, the number of successfully completed tasks, the number of tasks waiting for completion, and the related ranking domain are all recorded. After completing each task, the residual energy level, the total scores, the number of successfully completed tasks, and the number of tasks waiting for completion of the execution node are different from before. Therefore, the calculation of ranking domains is dynamic, which is updated according to the dynamic task allocation.
5. A Task Allocation Algorithm Based on Score Incentive Mechanism

TASIM consists of three phases: (1) task allocation on cluster head, (2) task migration on unstable node, and (3) task reallocation on death node.

5.1. Task Allocation on Cluster Head. Task allocation on cluster head is a kind of centralized allocation scheme. The allocation principle is firstly selecting the sensor nodes in the first ranking domain. Based on the deadline and the energy needed to complete a task, the sensor nodes in the first ranking domain are first selectively chosen as candidate nodes. If there are not enough sensor nodes in the first ranking domain, the sensor nodes in the second ranking domain will be selected. Then, the basic scores for the candidate nodes are calculated. The candidate node with the least basic score is chosen as working node for task execution. The details of task allocation are listed as follows.

**Step 1.** Sort all the subtasks based on their dependency among each other. The sorted subtasks are listed in a task allocation queue. The subtask in the front of task allocation queue can be preferentially allocated. In addition, all the subtasks in the allocation queue can be allocated and executed at the time.

**Step 2.** Allocate the first subtask in the task allocation queue. Cluster head first estimates the energy needed to complete the task. Then, the sensor nodes with enough residual energy for task completion are selected as candidate nodes, which are denoted by \( V_{\text{candidate}} \).

**Step 3.** Select working nodes from candidate nodes. First, judge whether the candidate nodes can complete the task within the deadline of the task. The sensor nodes which are not able to complete the task within the deadline are removed from the candidate nodes. Then, the candidate node with the least basic score is chosen as working node.

**Step 4.** If there are not enough sensor nodes in the first ranking domain, the sensor nodes in the second ranking domain will be further selected based on the abovementioned three steps.

**Step 5.** After selecting the working node, the cluster head will send the subtasks and the related basic scores to the working node.

**Step 6.** Task execution: the working node executes the allocated subtasks and waits for the allocation of the next subtasks. The rest of subtasks will be allocated one by one until all the subtasks are successfully allocated.

**Step 7.** If the working node successfully completes the subtask, it will be rewarded by the scores \( \text{Score}_i(T) \). Then, the information of the total scores and the residual energy will be sent back to the cluster head with the result of task execution.

**Step 8.** If the working node fails or is attacked by malicious nodes and it is unable to complete the subtasks within deadline, the sensor nodes will be punished by deducting some scores \( \text{Score}_i^\text{failure}(T) \) \( \text{Score}_i(T) = \eta \times \text{Score}_i^\text{failure}(T) \). If the uncompleted subtask can be successfully migrated to another node, we obtain \( \eta = 0.5 \). Otherwise, \( \eta = 1 \).

5.2. Task Migration on Unstable Node. Since WSNs are always deployed in hostile environment, sensor nodes are easily attacked and vulnerable to fail. When a sensor node monitors its failure or is being attacked by a malicious node when executing tasks, it considered itself unstable. The unstable node cannot continually complete the tasks; thus they ask for other nodes’ collaborations to help with the tasks’ further completion. In this paper, we propose task migration; that is, the uncompleted tasks can be migrated on other cluster members as a working node becoming unstable. In the task migration scheme, cluster members consult each other to complete the task migration, which does not need the cluster head’s participation, thus effectively reducing the cluster head’s working load and energy consumption.

There are many kinds of methods to achieve the collaboration between sensor nodes. In this section, an auction mechanism is proposed for sensor nodes’ collaboration and task migration. The details of the task migration scheme are listed as follows.

**Step 1.** Invitation for bids: when a sensor node monitors its failure or is being attacked by a malicious node and cannot continually complete the allocated tasks, it will initiate an auction to find a cluster member for task migration. The unstable node \( V_{\text{failure}} \) first finds out all the completed tasks \( T_{\text{failure}} \). Then, it sends a Request for Proposals (RfP) to the cluster members in the same cluster. In the RfP, the uncompleted tasks are described in detail, which include the deadlines of the tasks and the biggest budget for completing the tasks \( \text{Score}_{\text{budget}} \) \( \text{Score}_{\text{budget}} = \text{Score}_i^\text{failure} \).

**Step 2.** Selective bidding: when the cluster member receives the RfP, the selective bidding scheme is employed. That is, according to the task descriptions in the RfP, each cluster member judges whether it is able to complete the tasks. If it cannot complete the tasks, it will quit the bidding. Otherwise, it will return a quote to the tender nodes. The details of the selective bidding are listed as follows.

**Step 2.1.** In the process of selective bidding, the deadlines of the tasks are firstly considered. If the expected task completion time is larger than the deadline, the bidding node will quit the bidding.

**Step 2.2.** Otherwise, the rewarded scores \( \text{Score}_i(T_{\text{failure}}) = \text{BasicScore}_i \times C_{\text{total}}(T_{\text{failure}}) \) for completing the tasks are calculated, where \( \text{BasicScore}_i \) is the basic score and \( C_{\text{total}}(T_{\text{failure}}) \) is the total energy consumption for completing all the bidding tasks. If the reward score is larger than the expected budget in the \( \text{Score}_i(T_{\text{failure}}) > \text{Score}_{\text{budget}} \), the sensor node will also quit the bidding.
Step 2.3. If \( \text{Score}_{\text{reward}}(T_{\text{failure}}) < \text{Score}_{\text{budget}} \), the sensor node will accept the bidding and provide the bid price \( \text{Score}_{\text{reward}}(T_{\text{failure}}) \).

Step 3. Time-interleaved bidding: based on the different ranking domains, assign different bid times for the bidding nodes. That is, a bidding node in the first ranking domain can tender at \( t_1 \) time, while another bidding node in the second ranking domain can tender at \( t_2 \) time. We set \( t_2 > t_1 \). That is to say, the sensor nodes with higher ranking domain have a bigger bid on priority.

Step 4. The first bidder winning: in order to shorten the tender time, the first bidder winning scheme is adopted in TASIM. That is, the bidding node which firstly delivers the tender wins the bidding.

Step 5. Task migration: the tender node \( V_{\text{failure}} \) then migrates the uncompleted tasks \( T_{\text{failure}} \) to the winning node. The scores of the tender node are deducted by \( \text{Score}_{\text{reward}}(T_{\text{failure}}) + \text{Score}_{\text{punish}}(T_{\text{failure}}) \). Then, the updated scores, the residual energy, the winning node, and the successful migrated tasks are reported to the cluster head.

Step 6. Information update: after receiving the reported information, the cluster head will update the total scores and the residual energy of the unstable node as soon as possible.

Step 7. Task execution: the winning node executes the migrated tasks. After the successful completion, it reports the results to the cluster head and can be rewarded by the scores \( \text{Score}_{\text{reward}}(T_{\text{failure}}) \).

Step 8. Task migration failure and task reallocation: If we cannot find a winning node with time \( t_2 \), all the uncompleted tasks will be reported to the cluster head and wait for reallocation.

5.3. Task Reallocation on Death Node. If a cluster member dies, the uncompleted tasks on the death node will be reallocated by the cluster head. The details of task reallocation are listed as follows.

Step 1. Cluster members periodically report their status information to the cluster head, including their residual energy and their scores. The cluster head periodically updates the ranking domains.

Step 2. If the cluster head cannot receive any information about one cluster member within the prescribed period of time, the cluster member is considered to be dead.

Step 3. The cluster head checks whether there are uncompleted tasks on the death node. If there are not, the death node’s information will be directly deleted. Otherwise, the uncompleted tasks will be reallocated to other cluster members.

6. Simulation Results and Analysis

Our experiments are performed using Matlab. Two different sets of simulations are implemented. First, the performance of the TASIM based on different simulation parameters, for example, the number of cluster members and the different values of \( \alpha \), is evaluated. Then, the network lifetime, the success task execution rate, and the node failure rate of TASIM, EcoMapS [12], and DTAS [23] are compared. The deployment area is set to 500 m \( \times \) 500 m. The communication range of sensor nodes is set to 75 m. Other simulation parameters are shown in Table 1.

6.1. Performance of TASIM

6.1.1. Network Equilibrium and the Average Number of Cluster Members. Generally, the proposed TASIM can well balance the energy consumption of the whole network. As shown in Figure 2, the network equilibrium does not change with different number of cluster members. That is because of the following. (1) In the score incentive mechanism, the residual energy levels of sensor nodes are taken into account for task allocation. The sensor node which successfully completes more tasks can obtain more score but have less residual energy. In this case, the basic score of the sensor node is relatively small, which leads to a smaller possibility to be selected for task execution. Thus, the sensor nodes with more residual energy are more likely to be selected for task execution. (2) The ranking domain of a sensor node is dynamically updated. Therefore, the task allocation can be timely updated based on sensor nodes’ status information. (3) The proposed TASIM is centralized algorithm. The cluster head knows the information of all the cluster members; therefore, the cluster head can efficiently allocate tasks to balance the energy consumption of sensor nodes.

6.1.2. Energy Consumption and the Average Number of Cluster Members. As shown in Figure 3, the more tasks are executed, the more energy is consumed. However, the energy...
consumption is almost irrelevant to the average number of cluster members. This is because of the following. (1) The energy consumption of network depends on the size and the number of completed tasks. Thus, the more tasks are completed, the more energy is consumed. (2) The proposed TASIM chooses the most suitable sensor nodes for task allocation. The sensor nodes with higher residual energy and better service abilities are more likely chosen as working nodes, which is not related with the number of sensor nodes. Therefore, the average energy consumption of the network does not change with the average number of cluster members.

6.1.3. Residual Energy and the Average Number of Cluster Members. As concluded in Figure 3, the more tasks are executed, the more energy is consumed. Figure 4 shows that the average residual energy of sensor nodes decreases with the increase of number of tasks, that is because much more energy is exhausted. In addition, we find that, under the same number of tasks, the more sensor nodes can obtain the higher average remaining energy. This is because the number of working nodes remains the same with the constant number of tasks. In this case, the more number of sensor nodes introduces more spare nodes. All the spare nodes have much more residual energy than the working node; thus the average residual energy is relatively higher.

6.1.4. Algorithm Running Time and the Average Number of Cluster Members. As shown in Figure 5, the running time of TASIM increases with the increasing number of tasks and cluster members. First, TASIM is a kind of centralized algorithm. When the cluster head allocates each task, it needs to filter the sensor nodes as the best working node. If there is not a suitable node in the first ranking domain, the filter will be continued in the second ranking domain. Each task allocation will introduce a filter process. Therefore, more tasks lead to more filter processes, and the running time of task allocation will be accordingly increased. In addition, when a sensor node fails or is attacked, the uncompleted tasks on the sensor node will be migrated to other cluster members. If there are more cluster members, the number of sensor nodes which participate in the auction will increase. The time of task migration increases and the running time of TASIM becomes longer.

6.1.5. Network Equilibrium and $\alpha$. As shown in Figure 6, the node ratio $\alpha$ in the first ranking domain is changed from 0.1 to 0.7. The parameter $\alpha$ does not have obvious influence on the network equilibrium, because $\alpha$ only decides the number of iterations.
of sensor nodes in the first grade domains but does not have influence on selecting sensor node for task allocation.

6.1.6. Energy Consumption and $\alpha$. As shown in Figure 7, the node ratio $\alpha$ in the first ranking domain is set as 0.1, 0.3, 0.5, and 0.7, respectively. The different values of $\alpha$ also do not have obvious influence on the energy consumption of the network, because the energy consumption directly depends on the sizes and the number of tasks. Therefore, the energy consumption of the network does not change with the different values of $\alpha$.

6.1.7. Residual Energy and $\alpha$. As concluded in Figure 8, the different values of $\alpha$ also do not have obvious influence on the energy consumption of the network. Therefore, the residual energy of sensor nodes does not distinctly change with the different values of $\alpha$.

6.1.8. Algorithm Running Time and $\alpha$. As shown in Figure 9, it can be concluded that the running time of the algorithm increases with the higher value of $\alpha$. The higher $\alpha$ is, the more sensor nodes are divided into the first ranking domain. Therefore, much more is spent for working nodes’ selection, which ultimately introduces longer running time of TASIM.

6.1.9. The Success Rate of Completion Task and $\alpha$. As shown in Figure 10, it can be concluded that the success rate of completion task increases with the higher value of $\alpha$. This because the higher $\alpha$ is, the more sensor nodes are divided into the first ranking domain. When a sensor node fails
or is attacked by malicious nodes, the sensor nodes in the first ranking domain are firstly chosen as candidate node for task migration. The sensor nodes in the first ranking domain are more likely to successfully complete the migrated task than the sensor nodes in the first ranking domain. Therefore, if there are more sensor nodes in the first ranking domain, the migration tasks are more likely to be successfully finished, which ultimately introduces higher success rate of completion task.

6.2. Comparison of TASIM, EcoMapS, and DTAS

6.2.1. Network Lifetime and the Average Number of Cluster Members. In the simulation, the node failure rate is set between 0.01 and 0.11. \( \alpha \) is set to 0.7. As shown in Figure 11, it can be concluded that more sensor nodes can prolong the network lifetime. That is because more sensor nodes can execute more complex tasks and also the total energy of the network is increased; therefore the network can work longer. In addition, it is found that, with the same number of sensor nodes, TASIM outperforms EcoMapS and DTAS in terms of prolonging network lifetime. In TASIM, task allocation is conducted by considering both node failure and death. The uncompleted tasks on the failure or death nodes can be successfully migrated to cluster member for further execution. However, in EcoMapS and DTAS, they do not consider how to deal with the uncompleted tasks on the failure or death nodes.

6.2.2. The Success Rate of Completion Task and the Average Number of Cluster Members. Figure 12 shows that more sensor nodes can efficiently improve the success rate of completion task. That is because more sensor nodes introduce more candidate nodes in the first ranking domain. In this case, it is more likely to choose other cluster members to continually execute the uncompleted tasks on the unstable nodes. As shown in Figure 11, the number of tasks is set to 100. We only need 25 sensor nodes to finish all of them. In TASIM, a working node can obtain the corresponding scores for rewarding successful completion of task. The sensor node with higher score is considered to be much more trusty. In addition, although the chosen node fails or is attacked, the tasks can be successfully migrated to the surrounding node with high credibility. This mechanism ensures that the task can be completed by fewer and more reliable nodes.

6.2.3. Network Lifetime and the Node Failure Rate. Figure 13 shows that higher rate of failure nodes introduces shorter network lifetime. The robustness of TASIM against the node failure rate is better than that of EcoMapS and DTAS. However, when the node failure rate increases to 0.07, the three algorithms cannot work well. Taking our algorithm,
TASIM, for example, it is known that if the node failure rate is relatively higher, the chosen node for task migration or reallocation may be an unstable node. In this case, the uncompleted tasks ask for further task migration or reallocation. Repeated task migration or task reallocation can cause the subtasks timeout, and the whole task cannot be completed in time. In addition, the repeated task migration or task reallocation can exhaust the network resources, which ultimately shortens the network lifetime.

6.2.4. Algorithm Running Time and the Average Number of Cluster Members. The compared three algorithms are all centralized algorithms. Figure 14 shows that more sensor nodes need longer time for task allocation. In addition, DTAS outperforms our proposed TASIM, because in DTAS the uncompleted tasks on unstable nodes can be directly migrated to the other working nodes with the maximal idle time, while in TASIM, task migration is achieved by auction, which introduces longer auction time. Therefore, TASIM prolongs the execution time of the algorithm.

7. Conclusions

Nowadays, task allocation is needed in many applications such as complex task execution, reliable image processing, and multimedia data compression. In this paper, a task allocation algorithm based on score incentive mechanism (TASIM) is proposed for complex task execution in WSNs. TASIM divides sensor nodes into different ranking domains based on their residual resources and services capacities, which narrows the scope of working nodes and balances the energy consumption of them. In addition, a score incentive mechanism is proposed to assess sensor nodes’ qualities for completing tasks. Based on the score, the uncompleted tasks on unstable node can be migrated to other nodes for further completion, which ensures that all the tasks can be finished in time. Furthermore, simulation results demonstrate that the performance of the TASIM is clearly better than that of conventional task allocation algorithms, for example, EcoMapS and DTAS, in terms of both network load balance and energy consumption.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The work is supported by Qing Lan project, Natural Science Foundation of JiangSu Province of China, no. BK20140248,
the National Natural Science Foundation of China under Grant nos. 61401107 and 61401147, the Foundation of Changzhou Key Laboratory of Special Robot and Intelligent Technology, no. CZSR2014004, China, and Jiangsu Province Ordinary University Graduate Innovation project, no. CXZZ13_02.

References


