

EOTS: An Energy-consumption Optimization-oriented Task Scheduling Algorithm for Wireless Sensor Networks

<http://dx.doi.org/10.3991/ijoe.v12i07.5904>

Zeyu Sun^{1,2}, Yuanbo Li¹, Chuanfeng Li¹, Yalin Nie¹

¹Luoyang Institute of Science and Technology, Luoyang, China

²Xi'an Jiaotong University, Xi'an, China

Abstract—The mismatch of task scheduling results in rapid network energy consumption during data transmission in wireless sensor networks. To address this issue, the paper proposed an Energy-consumption Optimization-oriented Task Scheduling Algorithm (EOTS algorithm) which formally described the overall power dissipation in the network system. On this basis, a network model was built up such that both the idle energy consumption in sensor nodes and energy consumption during the execution of tasks were taken into account, with which the whole task was effectively decomposed into sub-task sequences. They underwent simulated annealing and iterative refinement, with the intention of improving sensor nodes' utilization rate, reducing local idle energy cost, as well as cutting down the overall energy consumption accordingly. The experiment result shows that under the environment of multi-task operation, from the perspective of energy cost optimization, the proposed scheduling strategy recorded an increase of 21.24% compared with the FIFO algorithm, and an increase of 16.77% in comparison to the EMRSA algorithm; while in light of network lifetimes, the EOTS algorithm surpassed the ECTA algorithm by a gain of 19.21%. Therefore, the effectiveness of the proposed EOTS algorithm is verified.

Index Terms—wireless sensor network; energy consumption model; task scheduling

I. INTRODUCTION

Wireless sensor networks are a new-type network system linked by tens of thousands of self-organized multi-hop sensor nodes. It unifies organically the information world and the physical world, and realizes such operations as data sampling, data computation, data communication, and data storage [1-2]. Along with rapid development of information technology, the application of wireless sensor networks has been broadened to a variety of engineering fields encompassing military and national defense, environmental monitoring, rehabilitation and disaster response, smart home, sanitary and medicine, agricultural production, as well as transportation [3-4].

The fast-pace advancement of science and information technology drives data sampling in wireless sensor networks to be more oriented, specific, and precise, which poses greater challenges to highly-efficient management of energy consumption in the system of wireless sensor networks [5]. In this connection, how to reduce energy cost in the network system has drawn increasing attention. During data transmission, energy-efficient-based schedul-

ing has become the priority of task scheduling in the wireless sensor networks. Compared to conventional task scheduling that pursues minimum makespan, energy-efficient-based scheduling features the main objective of a best-effort reduction of the times and time for data resources to exploit, so that the utilization rate of data resources increases at the same time when overall performance and energy consumption keep balance [8-9]. Without comprehensive consideration of overall data load rate and energy consumption utilization rate, tasks based on traditional load balance scheduling strategy tend to be allocated to overmuch sensor nodes, thus leading to energy dissipation. By cluster reconfiguration and data placement, Maheshwari et al. [10] proposed an algorithm that dynamically reconfigured cluster by scaling up (down) the number of nodes in the sensor based on current workload. Pietri et al [11] established several models of data utilization rate at the state of overall load balance, and used the feedback theory to adjust and compute utility of sensor nodes, thus maintaining overall load balance at the same time when the total energy consumption is lowered.

It is also an efficacy energy-saving approach to reducing energy consumption of sensor nodes based on characteristics of currently operating tasks. For instance, the clustering structure was used in Paper [12] to complete dynamic transformation between clusters, so that the energy cost of sensor nodes was lowered until the overall energy achieved a balance; with virtual technology in wireless sensor networks, Paper [13] integrated numerous clusters into the same network system, and turned off idle nodes for the sake of energy saving. In addition, task load based scheduling strategy is another way to achieve energy optimal management. According to computation of node performance and task attributes, Mashayekhy et al. Paper [14] combined energy cost factors with task scheduling mechanism, and put forward the energy-consumption sensory scheduling model accordingly. Under the constraints of service level agreements, the model allocated the optimal execution node for each task based on greedy algorithm in a way that realized minimization of overall energy cost. Ge et al. Paper [15] proposed a computation model which took into account execution performance and overall energy consumption performance. After acquiring system performance and task parameters, this model used scheduling strategy to achieve balance between overall performance and energy consumption. Guzek et al. [16] transformed the issue of task scheduling in distributed systems into a matter of perfor-

mance optimization under the constraints of multiple objectives.

In general, the energy consumption in network systems comprises idle energy dissipation in sensor nodes and energy consumption during the execution of tasks [17-19]. The present studies associated with energy optimal management deal separately with idle energy optimization and energy optimization during execution time, lacking an integral consideration of both, however. To address the issue, a multi-task energy consumption model was built up in combination with characteristics of wireless sensor network application and task scheduling algorithm. The energy-consumption optimization-oriented task scheduling algorithm was proposed accordingly, based on which the whole task was effectively decomposed into sub-task sequences. They underwent simulated annealing and iterative refinement for energy saving, thus cutting down overall energy cost.

II. ENERGY CONSUMPTION MODEL

A. Task formalization description

First of all, taskList and freeList concerning task scheduling in wireless sensor networks were described at any given time t as $(Job_0, Job_1, \dots, Job_k)$ and $(node_0, node_1, \dots, node_n)$, respectively; the task list constituted by a total of K tasks was expressed as $(task_0, task_1, \dots, task_m)$, where the number of idle resources was n , the number of tasks m , and $m \geq n$ by hypothesis. Considering that *frame*, the smallest unit for data application, was independent from each other, which meant that every task corresponded to a *frame*, the aforementioned task list hence could be represented as $(frame_0, frame_1, \dots, frame_m)$, and a *frame* was regarded in uniformity as a *task* in the paper. The three-level scheduling strategy was commonly employed in the cloud data system. Specifically, a stochastic scheduling sequence was selected out first; then, a proper *Job* was chosen from the sequence; finally, a proper *Task* was picked out of the *Job* before it was included in the *taskSchList*, waiting to be allocated to the corresponding data node. As the first two levels exerted no influence on energy consumption of the task, the third level was directly modelled for analysis. In this connection, after scheduling, each of the data nodes maintained an execution list. Assuming that the current state of all the nodes was

$$\begin{aligned} taskList_0 &: [task_0, task_1 \dots task_{r-1}], \\ taskList_1 &: [task_r, task_{r+1} \dots task_{2r-1}], \\ &\quad \text{M} \quad \quad \text{M} \\ taskList_n &: [task_{nr}, task_{nr+1} \dots task_m] \end{aligned} \quad (1)$$

B. Task energy consumption modeling

After scheduling, the $task_i$ in the execution list was different from the one in taskList. Therefore, when the task was submitted and running, the time list for each data node to finish the tasks was obtained as

$$\begin{aligned} T_0 &: [t_0, t_1, \dots, t_{r-1}], \\ T_1 &: [t_r, t_{r+1}, \dots, t_{2r-1}], \\ &\quad \text{M} \quad \quad \text{M} \\ T_n &: [t_{nr}, t_{nr+1}, \dots, t_m] \end{aligned} \quad (2)$$

First, node i was taken as an example to describe energy consumption per node. T_i denoted the total time for node i to operate, as shown in Equation (3).

$$T_i = \sum_{j=ir}^{(i+1)r-1} t_j \quad (3)$$

The time for all nodes to operate could be calculated from Equation (3). Equation (4) was the process of computing T_{max} , where $\max()$ functioned as a way to return to the maximum parameter in the list.

$$T_{max} = \max(T_0, T_1, \dots, T_n) \quad (4)$$

Equation (5) described the operation energy consumption E_i of node i , where E_{work_i} denoted the energy cost for node i to complete all the tasks in *taskList*[$task_{ir}, task_{ir+1}, \dots, task_{(i+1)r-1}$], which was determined by both the power of node i and the task runtime. E_{idle_i} , the idle energy consumption demanded for the node during the scheduling, was affected by the idle power and slack time of the node. The slack time resulting from system instability was ignored here, and it was assumed that node i was at idle when and only when it completed all the tasks in *taskList*. Equation (6) and Equation (7) separately served as the computation of the aforementioned two kinds of energy consumption.

$$E_i = E_{work_i} + E_{idle_i} \quad (5)$$

$$E_{work_i} = \sum_{j=ir}^{(i+1)r-1} nodePower_i * t_j \quad (6)$$

$$E_{idle_i} = nodeIdlePower_i * (T_{max} - T_i) \quad (7)$$

The energy consumption for node i to finish its task was obtained when Equation (6) and Equation (7) were substituted into Equation (5), as shown in Equation (8).

$$E_i = \sum_{j=ir}^{(i+1)r-1} nodePower_i * t_j + nodeIdlePower_i * (T_{max} - T_i) \quad (8)$$

Based on the energy consumption computation model of node i , Equation (9) was the computation for cluster energy consumption in the cluster list $(node_0, node_1, \dots, node_n)$ which incorporated idle energy consumption in sensor nodes and energy consumption during the execution of tasks.

$$E = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} nodePower_i * t_j + \sum_{i=0}^{n-1} nodeIdlePower_i * (T_{max} - T_i) \quad (9)$$

Equation (3) was substituted into Equation (9), and the final energy consumption computation model of cluster nodes was obtained as

$$E = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} (nodePower_i - nodeIdlePower_i) * t_j + \sum_{i=0}^{n-1} nodeIdlePower_i * T_{max} \quad (10)$$

As could be seen from Equation (10), the energy consumption for a set of data tasks to be scheduled and run comprised two parts. The first part concerned node power, idle power, and the runtime of one task in the data nodes. During model analysis, it was assumed that the node power and idle power were constant, which meant that the difference between them was constant. The second part was associated with idle power of a single node and the runtime of all the tasks in the nodes. The time required for the same node to process all the tasks was closely related to task numbers. The task scheduling strategy affected the power dissipation of a task by controlling the target node that runs the task.

According to EMRSA task scheduling algorithm, for each of the task in the *taskSchList*, the scheduling strategy based on a priori value of energy cost guarantees that the current task is distributed to the node with least energy

dissipated. However, in terms of the overall task, one node may correspond to more than one task to generate the minimum power consumption. Therefore, for a certain node, if the largest energy decrement is generated when the node processes another task instead of the current one, the present seemingly optimal task scheduling strategy impairs the overall optimal energy-saving configuration as a comparison. Given that, in order to achieve energy consumption optimization, it is necessary to upgrade the task scheduling strategy by expanding scheduling scopes.

III. EOTS ALGORITHM BASED ON SIMULATED ANNEALING

A. Initialization

Simulated annealing algorithm [17] is the expansion of local optimization dynamic programming. In light of energy-consumption optimization-oriented task scheduling, the process of scheduling each task is described as a similar sub problem for the sake of local energy consumption optimization. Simulated annealing algorithm was referred to in initializing taskSchList, and the corresponding steps were shown as follows:

1) Solution space in task scheduling

For taskList ($task_0, task_1, \dots, task_{m-1}$) and freeList ($node_0, node_1, \dots, node_{n-1}$), the solution space in task scheduling is equivalent to a mapping scheduling sequence between m tasks and n nodes, namely $taskSchList(task_{0i}, \dots, task_{(m-1)j})$, where $taskSchList$ is the task scheduling sequence, $task_{ij}$ means to execute task i in node j , and $0 \leq i \leq m-1, 0 \leq j \leq n-1$.

2) Set up target function

According to Equation (10), the target function for task scheduling was expressed as

$$\min(E = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \alpha_i * t_j + \sum_{i=0}^{n-1} \beta_i * T_{max})$$

$$\left\{ \begin{array}{l} \alpha_i = nodePower_i - nodeIdlePower_i \\ \beta_i = nodeIdlePower_i \\ T_{max} \geq T_i, i \in \{0, 1, \dots, n-1\} \end{array} \right. \quad (11)$$

3) The principle of new solution generation

The principle of new solution generation exerts influence on the efficiency of pursuing for optimal values, which should hence be as simple as possible in order to reduce calculation amounts. The way to generate new scheduling sequences in the paper is swapping task scheduling nodes within the range of neighborhood factor N .

4) The acceptance criteria for new solutions

In the simulated annealing algorithm, the acceptance criterion for new solutions is: when the value of the target function for a new solution is smaller than the one for the current solution, the new solution is directly accepted; or else the new solution is only accepted under certain conditions. Metropolis acceptance criteria remain unchanged here, and Equation (12) is the conditions under which the new solution is accepted.

$$(\Delta p < 0) \text{ or } (e^{-\Delta p/T} > \text{random}(0,1)) \quad (12)$$

B. Process of task scheduling

According to background description of task scheduling, usually $m \geq n$ (m denotes the number of tasks, and n the number of nodes). Therefore, for each idle resource

node, several rounds of task scheduling are demanded in distributing m tasks. The process of task scheduling is shown as follows:

1. Initialize freeList $node_0 \dots node_{n-1}$, taskList $task_0 \dots task_{m-1}$, simulated annealing parameter T , and neighborhood factor N .
2. Calculate the energy consumption of each task in the current node $node_i$ according to $(nodePower_i - nodeIdlePower_i) * t_j$, and add the minimum calculated value into $taskSchList$.
3. calculate the overall power dissipation for all the tasks in the present taskSchList according to target function (11).
4. Swap the current task for its adjacent task in the $taskSchList$ within the range of neighborhood factor N according to the principle of new solution generation, and generate a new task scheduling sequence.
5. Calculate the overall power dissipation for the newSchList according to target function (11).
6. Calculate the D-value between the overall energy cost in 5) and the one in 3), and compare the D-value with the acceptance criteria of the new solution, On this basis, determine whether the newSchList is qualified to replace the original one or not.
7. Refresh taskList to see whether there is any unscheduled task; if not, the process of task scheduling stops; if there is, the process of task scheduling continues from 1).

The pseudocode of the simulated annealing based EOTS strategy is shown as follows. The input of the algorithm contains $freeList$, $taskList$, annealing control parameter T , and solution neighborhood space parameter N . T is used to calculate the acceptance probability for the new solutions upon its generation; and N to control the neighborhood space that the new solutions generate. The output is $taskSchList$.

Algorithm

```

1: initialize: taskSchList, newList, searchIndex=0
2: for all freei ∈ freeList do
3:   for all taskj ∈ taskList do
4:     energyij = poweri * tj
5:     energyMatrix.add(energyij)
6:   end for
7: end for
8: while taskList is not empty do
9:   for all freei ∈ freeList do
10:    taskIndex = energyMatrix.getMinValueIndex(freei)
11:    taskj ← getTask (taskList, taskIndex)
12:    taskSchList.add(taskj)
13:    taskEnergy0 = calculateTaskListEnergy(energyMatrix, freeList, taskSchList)
14:    for (searchIndex < N) do
15:      newList ← swapTaskInScheList ()
16:      taskEnergy1 = calculateTaskListEnergy(energyMatrix, freeList, newList)
17:      Δp = taskEnergy1 - taskEnergy0
18:      if (Δp < 0 or e-Δp/T > random(0,1)) then
19:        taskSchList.update(newList)
20:        break
21:      end if

```

```

22:  searchIndex++
23:  end for
24: end for
25: taskList.remove(taskj)
26: return taskSchList
    
```

IV. EXPERIMENT ANALYSIS

To verify the effectiveness of the EOTS strategy, the paper adopted the Inspur TS10000 high-performance computing cluster environment to build up a cloud data system. Blender 2.69, the widely used open-source cross-platform 3D mapping and data software, served as the data engine. Test data was selected from some segments of the movie “Big Buck Bunny”, the second animated movie of copyleft licenses in Blender Foundation. The energy consumption test instrument was Watts’up , with nominal voltage of 250V and nominal current of 15A.

A. Energy consumption performance analysis

The FIFO algorithm and the EMRSA^[14] algorithm were chosen for comparison and analysis of energy consumption performance. To be free from data unit corruption, the Max-Min method and the Z-score method were used to normalize experiment results. Table 1 is the parameter initialization for the experiment in this section and for the EOTS strategy, where *Node Number* denoted node numbers, *Job Number* the number of jobs, *Task Number* the number of tasks, namely the total frame numbers, *N* the search space for the new solution based on the EOTS strategy, and *T* the simulated annealing factor.

Figure 1 is the comparison of energy consumption performance among FIFO, ESERSA, and EOTS. The energy consumption based on each algorithm was normalized separately. The vertical axis represented the corresponding normalization values. The larger the value is, the larger the energy consumption is, and the lower the network performance is. The horizontal axis was task numbers.

As could be seen from Figure 1, FIFO performed worst, and EOTS performed best in that EOTS was 21.24% higher than FIFO, and 16.77% higher than EMRSA when dealing with multiple tasks. The reason for the worst performance of FIFO in tackling different task numbers is that it ignores the energy consumption factors of tasks. The EMRSA algorithm reflects the idea of the greedy algorithm, and minimizes energy cost of tasks on the premise of meeting users’ time requirements. However, the mere consideration of optimal energy conservation for current tasks may easily put this algorithm in the trouble of local optimization. With the help of simulated annealing thought, the EOTS strategy targets at the overall energy optimization instead of the limited local optimization, thus it performs better than the EMRSA algorithm.

B. Analysis of task scale expansion

During the experiment, despite the improvement of energy consumption performance, the EOTS strategy witnesses a prompt decrease of runtime performance at the same time. To effectively develop advantages and remove disadvantages of different algorithms under different constraints, the paper compared and analyzed the relative performance improvement between the EMRSA algorithm and the EOTS strategy under various conditions. Corresponding formulas were shown in Equation (13) and Equation (14), where “energy” denoted the energy con-

sumption based on the current algorithm, and “time” referred to the algorithm’s runtime.

$$energy = \left| \frac{energy_{EMRSA} - energy_{SAEO}}{energy_{EMRSA}} \right| * 100\% \tag{13}$$

$$time = \left| \frac{time_{SAEO} - time_{EMRSA}}{time_{SAEO}} \right| * 100\% \tag{14}$$

As the FIFO algorithm excludes energy consumption factors such that its energy consumption performance falls further behind, it is not necessary to compare it with the other two algorithms, as what the paper does. Figure 2 is the comparison between the EMRSA algorithm and the EOTS strategy in terms of relative performance enhancement.

In Figure 2, the vertical axis represents the normalized value of performance improvement. The larger the normalized value is, the more obviously the corresponding performance is improved. The horizontal axis represents task numbers. As can be seen from the figure, in comparison with the EMRSA algorithm, the percentage of energy

TABLE I. PARAMETER INITIALIZATION

parameter name	Parameter value
<i>Node Number</i>	40
<i>Job Number</i>	6
<i>Task Number</i>	100-1000
<i>N</i>	2
<i>T</i>	10000

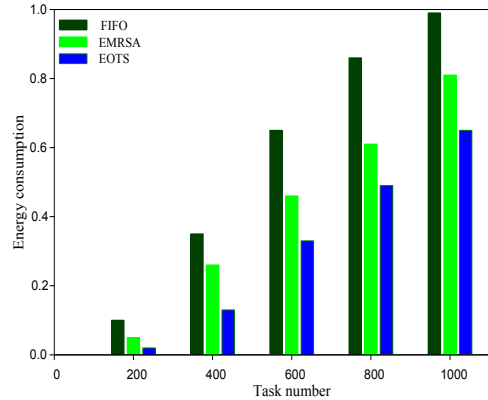


Figure 1. Energy consumption for multiple tasks

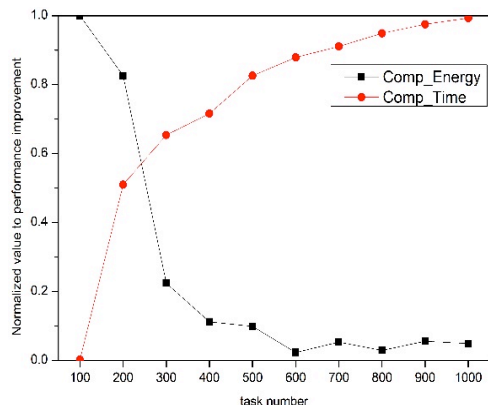


Figure 2. Comparison of the relative performance improvement between the EMRSA algorithm and the EOTS strategy

consumption improvement for the EOTS strategy drops rapidly along with the increasing task numbers, while the proportion of runtime growth gradually rises as the task numbers are enlarged. Two curves intersect at a certain point at between 200 and 300. Thus, when the task number is below 200, the rising amplitude of performance improvement based on the EOTS strategy exceeds the declining amplitude of runtime performance, which means that the EOTS strategy surpasses the EMRSA algorithm on the whole; by contrast, when the task number is above 300, the rising amplitude of performance improvement based on the EOTS strategy is shorter than the declining amplitude of runtime performance, which means that the EOTS strategy generally underperforms the EMRSA algorithm. The above results show that the optimal scheduled task number for the EOTS strategy to improve energy consumption performance is between 200 and 300.

C. Comparison of redundant node numbers and the networks lifetime

On simulation platforms of different scales, the paper compares both the number of redundant nodes and the change of coverage rates among the EOTS algorithm, the SCA algorithm in Paper [20], and the EPDM algorithm in Paper [21], as shown in Figure 3-5.

From figure 3 to figure 5 show separately the comparison between redundant node numbers and the change of coverage rates under the influence of diversified network sizes and different parameters. Figure 3 is the changing curves of sensor node numbers and working node numbers based on the EOTS algorithm, the SCA algorithm, and the EPDM algorithm, respectively, within the 100*100m² simulation domain. As can be seen, the EOTS algorithm requires a relatively small sum of working nodes under the action of disparate parameters; while the EPDM algorithm demands for a large number. The reason for this phenomenon is that the sensor nodes together with their neighboring nodes cover an even larger area by comparison, while the rest of the algorithms realize effective coverage by increasing sensor node numbers. Figure 4 and Figure 5 are the comparison between redundant node numbers under different coverage rates on the simulation platform with a size 300*300m². As can be seen, the more the working node numbers are, the less the sensor nodes are required, but only if meeting the requirements of certain coverage rates.

Figure 6 and Figure 7 are the comparison between network lifetimes and between algorithm runtimes concerning the EOTS algorithm and the ETCA algorithm, respectively. As can be seen from Figure 6, at the initial time, the two algorithms share a basically equal network lifetime; as the sensor node increases in number, all the curves representing network lifetime rise. However, with the adoption of nonlinear consistent coverage mode, the ECTA algorithm witnesses higher energy consumption than the EOTS algorithm during monitoring target nodes. When there are 180 sensor nodes, both the network lifetimes based on the two algorithms level off. The average network lifetime based on the EOTS algorithm was 19.21% higher than the one based on the ECTA algorithm. Figure 7 is the comparison between the EOTS's runtime and the ETCA's runtime with different numbers of sensor nodes. Considering that the chain-table storage is the energy storage mode for sensor nodes based on the ECTA algorithm, the high-energy nodes in the whole chain table

are traversed accordingly in access to higher privilege such that the target nodes are covered. In doing so, the ECTA algorithm is less complex than the EOTS algorithm. Therefore, the EOTS's runtime exceeds the ETCA's runtime.

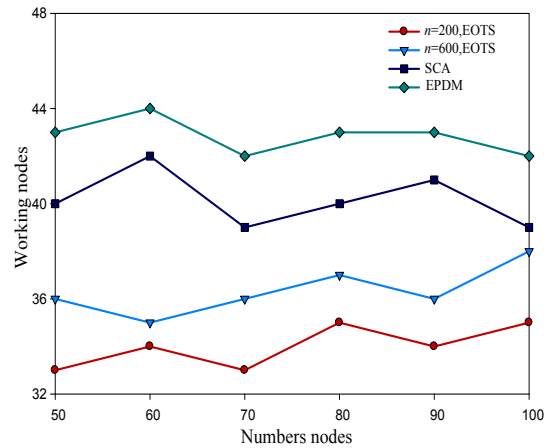


Figure 3. 100*100m², Comparison between the number of sensor nodes and working nodes based on different algorithms

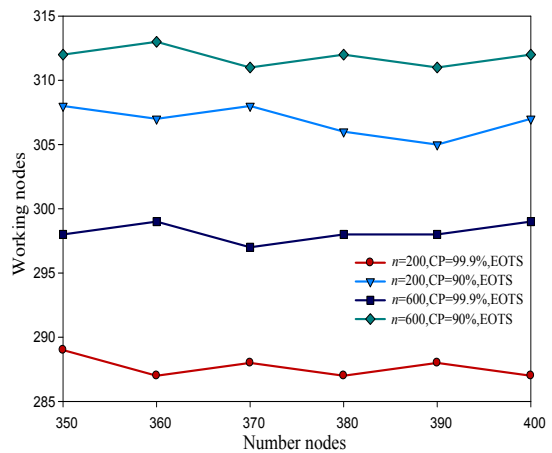


Figure 4. 300*300m², Comparison between the number of sensor nodes and working nodes based on the EOTS algorithm with different coverage rates

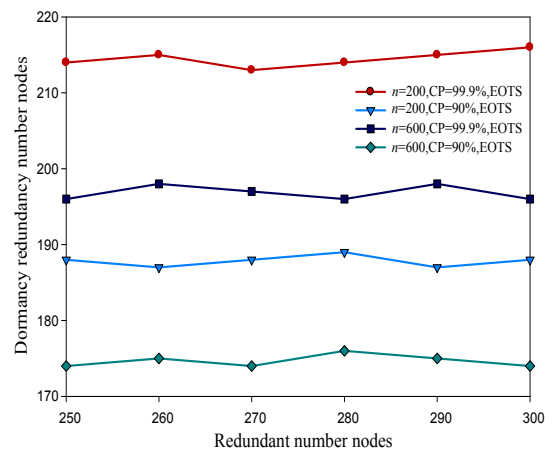


Figure 5. 300*300m², Comparison between dormancy redundant node numbers and redundant node numbers based on the EOTS algorithm with different coverage rates

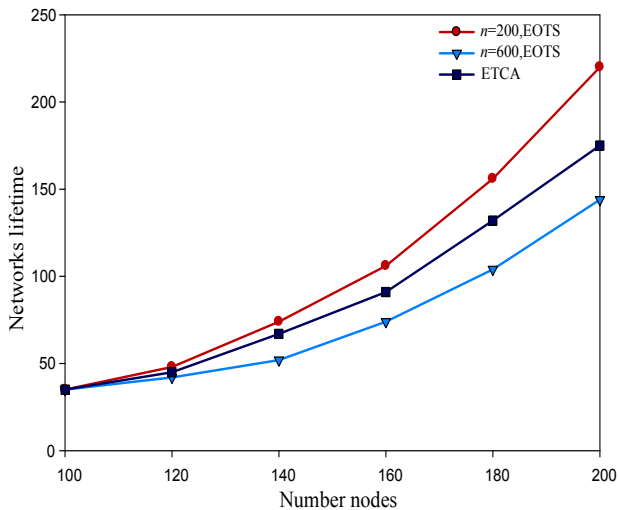


Figure 6. 200*200m², Comparison between network lifetimes based on the EOTS algorithm and the ETCA algorithm

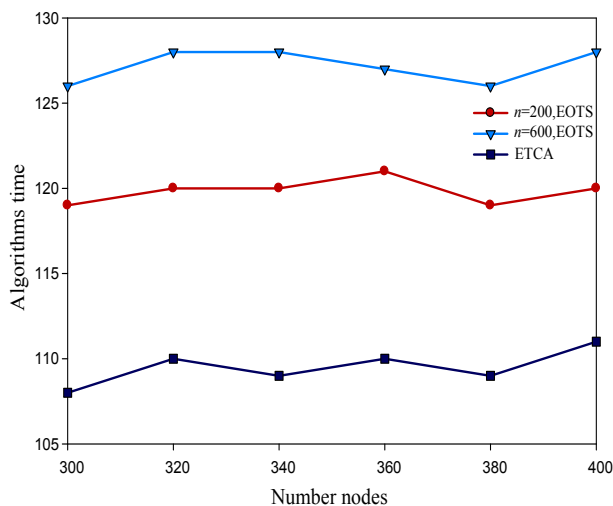


Figure 7. Comparison between the EOTS's runtime and the ETCA's runtime

V. CONCLUSION

In addressing the issue of energy consumption optimization for data systems in wireless sensor networks, the paper took advantage of the parallel characteristic between frames in data application, and comprehensively considered energy consumption optimization during slack time and execution time. On this basis, the paper established an energy consumption model, based on which the EOTS algorithm was proposed with the thought of simulation annealing. Through local energy consumption optimization, the goal of overall energy consumption optimization was achieved. The paper validated the strategy experimentally, where the environment of multiple tasks was provided. The result shows that from the perspective of energy cost optimization, the proposed scheduling strategy records an increase of 21.24% compared with the FIFO algorithm, and an increase of 16.77% in comparison to the EMRSA algorithm; while in light of network lifetimes, the EOTS algorithm surpasses the ECTA algorithm by a gain of 19.21%. In this way, the proposed EOTS algorithm is proved with favorable stability and expandability.

REFERENCES

- [1] Tseng YC, Chen PY, Chen W. "k-angle object coverage problem in a wireless sensor networks". *IEEE Sensor Journal*, 2012, vol.12, no.12, pp.3408-3416. <http://dx.doi.org/10.1109/JSEN.2012.2198054>
- [2] Habib A, Sajal DK. "Centralized and clustered k-coverage protocols for wireless sensor networks". *IEEE Transactions on Computers*, 2012, vol.61, no.1, pp. 118-132. <http://dx.doi.org/10.1109/TC.2011.82>
- [3] Seok JH, Lee JY, Lee JJ. "A Bipopulation-based evolutionary algorithm for solving full area coverage problems". *IEEE Sensor Journal*, 2014, vol.13, no.3, pp.796-807.
- [4] Mihaela C, JIE W. "Energy-efficient coverage problems in wireless ad-hoc sensor networks". *Computer Communications*, 2005, vol.29, no.4, pp.413-420.
- [5] Sandra S, Jaime L, Miguel G, José F. "Toledo. power saving and energy optimization techniques for wireless sensor networks". *Journal of Communications*. 2011, vol.6, no.6, pp.439-459.
- [6] Li K. "Energy efficient scheduling of parallel tasks on multiprocessor computers". *The Journal of Supercomputing*, 2012, vol.60, no.2, pp. 223-247. <http://dx.doi.org/10.1007/s11227-010-0416-0>
- [7] Mohammed E, Amman J, Ala AF, Muhammad A. "A Precise Indoor Localization Approach based on Particle Filter and Dynamic Exclusion Techniques". *Network Protocols and Algorithms*, 2013, vol.5, no.2, pp.50-71. <http://dx.doi.org/10.5296/npa.v5i2.3717>
- [8] Lee YC, Zomaya AY. "Energy efficient utilization of resources in cloud computing system". *The Journal of Supercomputing*, 2012, vol. 60, no.2, pp. 268-280. <http://dx.doi.org/10.1007/s11227-010-0421-3>
- [9] Zong Z, Manzanares A, Ruan X, Qin X. "EAD and PEBD: two energy-aware duplication scheduling algorithms for parallel tasks on homogeneous clusters". *IEEE Transactions on Computers*, 2011, vol.60, no.3, pp. 360-374. <http://dx.doi.org/10.1109/TC.2010.216>
- [10] Maheshwari N, Nanduri R, Varma V. "Dynamic energy efficient data placement and cluster reconfiguration algorithm for mapreduce framework". *Future Generation Computer System*, 2012, vol.28, no.9, pp.119-127. <http://dx.doi.org/10.1016/j.future.2011.07.001>
- [11] Sun ZY, Li H, Chen H. "Optimization coverage of wireless sensor networks based on energy saving". *International Journal of Future Generation Communication and Networking*, 2014, vol.7, no.4, pp.35-48. <http://dx.doi.org/10.14257/ijfgcn.2014.7.4.04>
- [12] Jaime L, Jesus T, Miguel G, Alejandro C. "Hybrid Stochastic Approach for Self-Location of Wireless Sensors in Indoor Environments". *Sensors*. 2009, vol.9, no.5, pp.3695-3712. <http://dx.doi.org/10.3390/s90503695>
- [13] Tan Y, Zeng GS, Wang W. "Policy of Energy Optimal Management for Cloud Computing Platform with Stochastic Tasks". *Journal of Software*, 2012, vol.23, no.2, pp. 266-278. <http://dx.doi.org/10.3724/SP.J.1001.2012.04143>
- [14] Zhao CJ, Wu HR, Liu Q, Zhu L. "Optimization strategy on coverage control in wireless sensor network based on Voronoi". *Journal on Communications*, 2013, vol.34, no.9, pp.115-122.
- [15] Du JZ, Wang K, Liu H. "Maximizing the lifetime of K-discrete barrier coverage using mobile sensor". *IEEE Sensors Journal*, 2013, vol.13, no.12, pp. 4690-4701. <http://dx.doi.org/10.1109/JSEN.2013.2270555>
- [16] Guzek M, Pecero JE, Dorronsoro, B, Bouvry P. "Multi-objective evolutionary algorithms for energy-aware scheduling on distributed computing systems". *Applied Soft Computing*, 2014, vol.24, no.5, pp. 432-446. <http://dx.doi.org/10.1016/j.asoc.2014.07.010>
- [17] Sousa T, Morais H, Vale Z, Faria P, and Soares J. "Intelligent energy resource management considering vehicle-to-grid: A simulated annealing approach". *IEEE Transactions on Smart Grid*, 2012, vol.3, no.1, pp.535-542. <http://dx.doi.org/10.1109/TSG.2011.2165303>
- [18] Talmai O, Madhanmohan RJ, Dharma PA. "Accurate Distance Estimation Using Fuzzy based combined RSSI/LQI Values in an Indoor Scenario: Experimental Verification". *Network Protocols and Algorithms*, 2012, vol.4, no.4, pp.174-199.

- [19] Seyed MJ, Karim F, Mehdi D. "Multiobjective Optimization for Topology and Coverage Control in Wireless Sensor Networks". *International Journal of Distributed Sensor Networks*. 2015, vol.2015,no.1,pp.1-11.
- [20] Sun ZY, Wu WG, Wang HZ, Chen H, Xing XF. "A Novel Coverage Algorithm Based on Event- Probability-Driven Mechanism in Wireless Sensor Network". *EURSIP Journal on Wireless Communications and Networking*, 2014,vol.2014, no.1,pp.1-17. <http://dx.doi.org/10.1186/1687-1499-2014-58>
- [21] Meng FZ, Wang HZ, He H. "Connected coverage protocol using cooperative sensing model for wireless sensor networks". *Acta Electronica Sinica*, 2011,vol.39,no.4,pp.772-779.
- [22] Xing XF, Wang GJ, Li J. "Polytype target coverage scheme for heterogeneous wireless sensor networks using linear programming". *Wireless Communications and Mobile Computing*, 2014,vol.14,no.8,pp.1397-1408. <http://dx.doi.org/10.1002/wcm.2269>
- [23] Yourim Y, Yong HK. "An efficient Genetic Algorithm for Maximum Coverage Deployment in wireless sensor network". *IEEE Transactions on Cybernetics*. 2013, vol.45,no.5,pp.1473-1483. <http://dx.doi.org/10.1109/TCYB.2013.2250955>

AUTHORS

Zeyu Sun (corresponding author) is an associate professor in School of Computer and Information Engineering, Luoyang Institute of Science and Technology, Luoyang, 471023, China. His research interests include wire-

less sensor networks, Internet of Things, (email: lylgszy@163.com)

Yuanbo Li is a lecture in School of Computer and Information Engineering, Luoyang Institute of Science and Technology, Luoyang, 471023, China. His research interests include wireless sensor networks, Internet of Things, (email: 408891898@163.com)

Chuanfeng Li is an associate professor in School of Computer and Information Engineering, Luoyang Institute of Science and Technology, Luoyang, 471023, China. His research interests include wireless sensor networks, Internet of Things, (email: lichuanfeng@sina.com)

Yalin Nie is a lecture in School of Computer and Information Engineering, Luoyang Institute of Science and Technology, Luoyang, 471023, China. Her research interests include wireless sensor networks, Internet of Things, (email: nieyalin111@163.com)

Projects (61503174, U1304603) supported by the National Natural Science Foundation of China; Project (14B520099) supported by Henan Province Education Department Natural Science Foundation; Project (142102210471,162102210113) supported by Natural Science and Technology Research of Foundation Project of Henan Province Department of Science. Submitted 02 June 2016. Published as resubmitted by the authors 03 July 2016.