A Price-based Adaptive Task Allocation for Wireless Sensor Network

Neda Edalat*, Wendong Xiao†, Chen-Khong Tham† *, Ehsan Keikha*, Lee-Ling Ong*
*Department of Electrical and Computer Engineering, National University of Singapore
Email: Neda.e, eletck, ehsan.keikha, eleoll@nus.edu.sg
† Institute for Infocomm Research, Singapore
Email: wxiao, cktham@i2r.a-star.edu.sg

Abstract

Applications for Wireless Sensor Networks may be decomposed into the deployment of tasks on different sensor nodes in the network. Task allocation algorithms assign these tasks to specific sensor nodes in the network for execution. Given the resource-constrained and distributed nature of Wireless Sensor Networks (WSNs), existing static (offline) task scheduling may not be practical. Therefore there is a need for an adaptive task allocation scheme that accounts for the characteristics of the WSN environment such as unexpected communication delay and node failure. In this paper, we focus on task allocation in WSNs which is performed with the aim of achieving a fair energy balance amongst the sensor nodes while minimizing delay using a market-based architecture. In this architecture, nodes are modeled as sellers communicating a deployment price for a task to the consumer. To address this task allocation problem, proposed price formulation is used as it continuously adapts to changes of the availabilities of resources. This scheme also accommodates for the node failure during task assignment. The Centralized and distributed message exchanged mechanisms between the nodes (sellers) and task allocator (consumer) are proposed to determine the winner among the sellers with the goal of reducing overhead and energy consumption. Simulation results show that, compared with a static scheduling scheme with an objective in energy balancing, the proposed scheme adapts to new environmental changes and uncertain network condition more dynamically and achieves a much better performance on energy balancing.

Index Terms

Energy-Balancing, Task Allocation, Price-Based Scheme, Market Architecture, Wireless Sensor Networks.

1. Introduction

The deployment of advanced applications for wireless sensor networks such as habitat monitoring to healthcare applications [1] may be decomposed into the deployment of tasks on different nodes in the network.

Due to the resource-constrained and distributed nature of these systems, one of the fundamental challenges in WSNs is to achieve a fair energy balance amongst nodes to maximize the overall network lifetime through task allocation [2], [3]. In addition to that situations such as unpredicted propagation delay and node failure may occur *during task assignment*. Static task allocation algorithms do not address these issues. Thus, the design of an adaptive task assignment scheme which considers available resources at each epoch is of essential necessity. Existing work on static task scheduling [2]–[5] achieves the energy balancing objective by regulating the energy consumption via Dynamic Voltage Scaling (DVS) [6]. DVS by decreasing the CPU speed reduces computational energy consumption; however this results in a longer schedule length. In work by Yu and Prassana [3], a energy balance was achieved as given each nodes available resource, each cluster of tasks are assigned to the sensor nodes as *a whole rather than adaptively allocating the individual task at each epoch* by considering resource availability at that epoch.

Pricing schemes for task scheduling has emerged as a promising solution to achieve a fair energy balancing result amongst nodes as this technique adapts to changing conditions [7], [8]. Load balancing and pricing has been recently discussed in the literature [8] for grid computing. However, the application of the pricing schemes to WSNs with consideration for limited resources,is almost unexplored.

In this paper, a price-based online solution for task allocation is proposed, one which places emphasis on a fair energy balance among nodes in order to maximize network lifetime. The task allocation is modeled as a market architecture. The components of the market architecture are explained in this paper. The consumer is modeled as an auctioneer and the sensor nodes represent the sellers in our scheme. When a task is to be allocated, the consumer broadcasts information about the tasks to the sellers. Each seller calculates its cost based on proposed price formulation to achieve the energy balancing objective and delay constrain in a form of "price". Nodes with higher prices are likely to have less remaining energy in future, so the price of node can be adjusted to influence decision making for task assignment to that node. In the case of an unexpected situation such as node failure *during the task assignment*, this scheme would run the

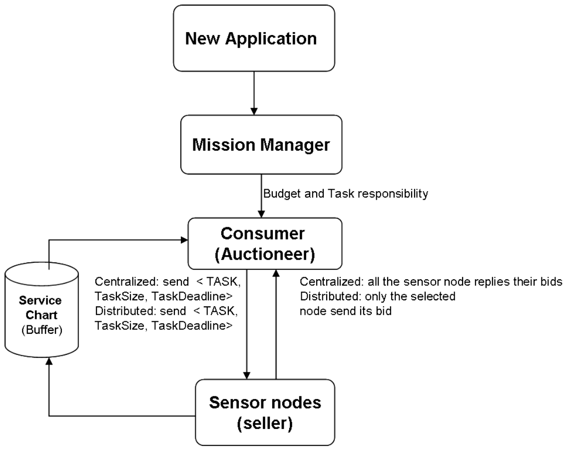

Figure 1. Market Based Architecture for Task Allocation

dynamic recovery phase. while in [9] considers only the case that node failure happens *before task assignment phase* and generated an alternative schedule. In this paper two massage exchange methods for winner determination in each round has been introduced; centralized, distributed scheme. The centralized method requires a larger overhead as each seller sends its bid to the consumer. This issue is addressed with the distributed method where each seller delays transmitting its message in proportion to its bid. Hence, the winning node is the first and only node to send its bid as the losing bids are not communicated.

This proposed scheme is a scalable and adaptive distributed task allocation solution for WSN. This scheme is scalable as it is independent of the number of available nodes and will adapt if the number of nodes change. As allocation is performed real-time, each node would *adaptively* react to the changes in resource availability and utilize new available resources at each time epoch so that it is adaptive.

The remainder of this paper is organized as follows. Section 2 presents the market architecture. The contributions of this paper, which are proposed price-base task assignment and recovery algorithm is introduced in Section 3. Another contribution of this paper, the message exchange mechanisms are discussed in Section 4. Section 5 shows simulation results and performance evaluation. The conclusion of this paper is presented in Section 6.

2. Market-Based Architecture for Task Allocation

In this paper, the adaptive task allocation scheme is modeled as a competitive market.The main goal is to find the suitable resources (cheapest sellers) to do the consumer's arrival task with the goal of maximizing *energy balancing* among the nodes. The market architecture [10] shown in Figure 1, comprises of a Mission Manager, service chart, consumers and sellers. When a new application is instantiated in the network, the input of that is fed into the Mission Manager. The components of this architecture are described:

Mission manager (MM): At the mission manager, the application level tasks are decomposed into the low-level tasks which can be comprehensible for the sensor nodes. The low-level tasks sequences and dependencies are represented by a direct acyclic graph (DAG). In a DAG graph, the vertices represent low-level tasks and the edges represent the precedence relationship between tasks. Another functionality of MM is to list the tasks in the queue based on their Earliest Start Time (EST) and Latest Start Time (LST). Should concurrent tasks exist in the list, higher priority is assigned to tasks with larger number successors on task graph. The manager then allocates the various task responsibilities to the consumers.

Consumer (auctioneer): The consumer acts as an auctioneer. With each task arrival, the consumer communicates the task message as $< Task, TaskSize, TaskDeadline >$ to the sellers. It also assigns the task to the winning seller. Should there be more than one consumer, the mission manager breaks the task graph and allocates the different set of tasks to the consumers.

Seller: The sellers are the sensor nodes. When a task message is received from the consumer, the nodes calculate their cost for accomplishing current task based on their current status of energy availabilities, communication cost, task deadline and resource release time. This winning seller is determined via different decision making schemes which are explained in the section 4.

Service Chart: The service chart acts as a buffer and maintains a history of the previous winning seller's cost information.

3. Task Allocation

This section presents our scheme for real-time task scheduling with objective of energy balancing and delay minimization. Our proposed task scheduling scheme which comprises of three phases; the listing phase, the price-based task assignment phase and recovery phase (in case of node failure).

3.1. Listing Phase

The listing phase is based on work in [9], [11] and computes the task sequence provided by the DAG graph to obtain the earliest start time (EST) and the latest start time (LST) of each task prior to starting the task assignment phase. Given these values, these tasks are queued into a list. The EST and LST for a task v_i can be computed recursively by traversing the DAG downward from entry node and upward from exit node respectively as follows:

$$EST(v_i) \quad = \max_{v_m \in pred(v_i)} EST(v_m + t_i) \tag{1}$$
$$LST(v_i) \quad = \min_{v_m \in succ(v_i)} LST(v_m - t_i) \tag{2}$$

where $pred(v_i)$ and $succ(v_i)$ are the set of immediate predecessors and successors of v_i respectively and t_i is the execution time of the task on sensor nodes. After the listing phase, the task graph is sequentialized into a queue and ready for the price-based task assignment phase. The tasks are queued for assignment to sensor nodes based on the EST. The LST is used as the task deadline for the assignment phase. Unlike work in [9], the EST and LST computed in this stage may be altered in the task assignment phase and dynamically due to packet loss or communication delay. Should there be tasks with concurrent EST, a higher priority is assigned to the tasks with more successors in the task graph.

3.2. Price-Based Task Assignment Phase

This assignment phase is performed in real-time. All the tasks that were enqueued at previous phase is consequently dequeued and allocated to the appropriate nodes based on our priced-based scheme. The design objective of the price-based adaptive task scheduling scheme is to allocate the task in real-time with current availability of resources. The pricing scheme would continuously adapt to changes of availability of resources. Moreover, with this method only the price value and not the available resource amount is communicated to the consumer, increasing the privacy of nodes. On the other hand, the computation overhead of the manager is reduced by distribution of some overheads to nodes.

3.2.1. Parameters for Price Formulation. Our proposed pricing scheme is parameterized by six variables; task size, energy price, base price, communication cost, task deadline and processor release time, some of which were used in [12].

- **Task Size (**S**)** which refers to expected energy required to compute or communicate the task.
- **Energy Price (**EP**)** is where each node generates a *so-called* Energy Price per unit task based on its level of remaining energy and is defined as:
$$EP_j = \frac{a}{1 - e^{-E_j/b}} \tag{3}$$
where E is the remaining energy of the node, a is the scaling parameters and b is the preferred coefficient which can appropriately represent the markup of energy price as the energy is consumed. This price is inversely proportional to the energy level allowing nodes with higher energy (lower price) to be selected.
- **Base Price (**BP**)** is defined as computational cost for doing task j by node i which can be calculated as: $BP_{ij} = S_j \times EP_i$
- **Communications Cost (**$CommCost$**)** is the cost of migrating the output of one task on one node to another task on an alternate node and is a function of the distance between the nodes and the size of the data packet Our pricing scheme determines accounts for the communication cost when assigning tasks.
- **Task Deadline (**TD**)** which is the latest start time (LST) defined in the listing phase. When two tasks are required to be scheduling concurrently, priority is given to the task with a closer deadline.
- **Processor Release Time (**RT**)** is the time at which the task execution at node would finish.

3.2.2. Energy Balancing Price Formulation. One contribution of this paper is a scheme that places emphasis on a fair energy balance amongst nodes constrained by the application's schedule length. Should a situation arise a computationally expensive task is released while only nodes with low energy are available, our scheme may wait for another node with relatively higher amount of energy to complete its task and assign this new task to that node to achieve a longer network lifetime. The price for assigning Task j to node i is:

$$P_{ij} = (CommCost + BP_{ij}) \left[1 + \exp^{\left[\frac{\lambda(t, DL_j)}{\gamma(t, RT_i)}\right]} \right] \tag{4}$$

where DL is the arriving time of the task and RT is the release time of the processor and where

$$\lambda(t, DL_j) = \begin{cases} k(t - DL_j), & \text{for } t > DL_j \quad (5) \\ \varepsilon, & \text{for } t < DL_j \quad (6) \end{cases}$$

and

$$\gamma(t, RT_i) = \begin{cases} (t - Rt_i), & \text{for } t > RT_i \quad (7) \\ \varepsilon, & \text{for } t < RT_i \quad (8) \end{cases}$$

A fairer energy balancing amongst nodes is achieved with this price formulation. When the current time is close to the release time of a energy processor RT_i with high energy availability, a low value would be set, increasing the selection possibility by the task manager. The deadline of this task to be assigned is also considered, as tasks with urgent deadlines would be allocated to a node that is available at a closer released time (or a node that readily available) at the expense of an unfair energy balance.

3.3. Recovery Phase

The proposed method enables to recover from node failure during the online task assignment phase. Previous work in [9] only considered node failure prior to task assignment by not selecting the node that had failed. Our recovery phase is to first recover the tasks that had been assigned to the failed node from its successors deployed on other nodes. We first determine if there are any tasks on that failed node that need to be deployed again based on its successors. If these tasks have no successors or no undeployed successors, then redeployment is unnecessary. Redeployment is also unrequired if the output of this task exists on another node. This situation occurs when this output data had been previously communicated to a successor task on another node.

However, if there exists no back-up of the output of these tasks then reassignment of these tasks onto another processor is required. If the deadline where this output is valid has exceeded, redeployment of this task would not be performed. Task assignment resumes as normal with the rest of the tasks after the recovery phase. The result of implementing a recovery phase which checks for previously communicated data from tasks on a failed node to avoid unnecessary task deployment is the savings in energy and time.

Although the recovery method proposed results in a slight increase in schedule length, simulation results shows the significant improvement in schedule length in comparison to rescheduling considered for static scheduling in addition to energy consumption and balance improvements.

4. Proposed Message Exchange Methods

Another contribution in this paper are techniques to communicate the messages between the sellers and consumer to determine the winner among the sellers at each round. As shown in Figure 1, the consumer sends a Task Message to the nodes. Each node then calculates its cost for doing the task based on the price formulation introduced in Section 3.2.2. The goal of the message exchange protocol is to determine the winning seller to assign the task to. Two schemes are discussed; centralized and distributed.

4.1. Centralized Message Exchange Method

When a new task arrives, the consumer broadcasts a message containing the $\langle$TASK, TaskSize, TaskDeadline$\rangle$ to all nodes. Each node calculates its cost (bid), and sends their bid to the consumer (bidder). The consumer selects the minimum bid from the bids received from all the nodes and the task would allocate to that node. The main disadvantage of this method is the overhead and the cost (energy consumption) for sending all the prices to the consumer when there are a lot of nodes, given that only one node will be selected. Moreover in this method the probability that collisions happen during message exchange is quite high.

4.2. Distributed Message Exchange Method

To reduce the communication overhead and energy consumption for message exchanging, a distributed winner determination scheme is proposed. As with the centralized scheme, the consumer broadcasts a message $\langle$TASK, TaskSize, TaskDeadline$\rangle$ to all nodes. However, instead of communicating its cost for accomplishing the current task immediately, each node sets a waiting time T_w proportional to its calculated price P_{ij} and goes to a LISTEN mode.

$$T_w = \ell \times P_{ij} \tag{9}$$

where ℓ is a linear coefficient. When the waiting time is completed, the node would then send its cost to the consumer. Then node with the lowest price will broadcast its bid first and be selected. Upon reception of a broadcast from a winning seller node, the remaining nodes (which are in a LISTEN mode) would leave the competition and avoid communicating their bids. This scheme will reduce the amount of overhead and energy consumption for sending non-winning messages to the consumer. Another advantage of distributed method is that if the number of available nodes (sellers) increases it won't have effect on the massage overhead and the performance of winner determination method since eventually only one node would reply.

5. Simulations

Simulations has been carried out to evaluate the performance of our schemes. Our proposed pricing scheme with message exchange methods are compared to a static task scheduling method. In order to have fair comparison the same objective which is the energy balancing has been applied for offline scheduling. Hence, the Critical Node Path Tree (CNPT) algorithm [11] was modified to Energy Balanced-CNPT (EB-CNPT) to schedule tasks offline with energy balancing objective based on the information on the available remaining energy of each node *prior to* task assignment.

Simulations are performed to investigate the following aspect:

- Performances of our scheme on energy balancing and energy utilization.
- The effect of our adaptive scheme on scheduling length and energy consumption over a different number of node.
- The effect of our scheme on situations of node failure.

In this simulation, a task graph of 35 tasks, where each task has a maximum number of 3 predecessors ($numPred = 3$) is assigned to 15 nodes. Each node has the following initial energies $[3.4, 2.4, 2, 3.3, 2, 2.3, 2.7, 3.2, 3.1, 2, 1.9, 2.9, 2.7, 3, 3]$Joule. Energy consumption for transmitting is based on the MICAz mote datasheet; $E_{Sending} = .017mJ$, $E_{Recieving(Listenmode)} = .031mJ$.

Comparisons on the energy balancing performance are made between our schemes and the modified static scheduling scheme (EB-CNPT). Figure 2 shows the remaining energy at the nodes after scheduling one round of tasks,

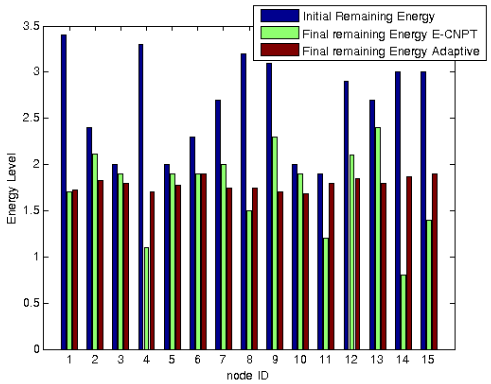

Figure 2. Energy Level vs Node ID

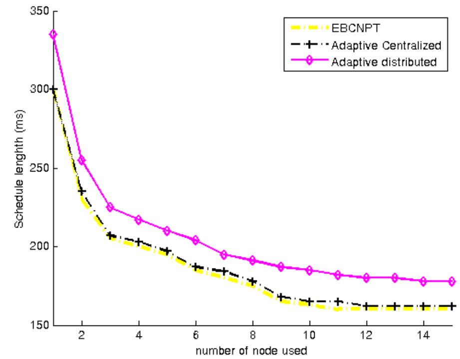

Figure 3. Scheduling Length vs Number of Nodes used

given the initial energy provided. The remaining energy for each node appears more balanced when the pricing scheme was used as compared to static scheduling. The reason of this performance is that our adaptive scheme would continuously adapt to the changes of available resource of each node, as the price that is set for task allocation changes after each task assignment. However, for static scheduling (EB-CNPT) the resources availability at each node has been considered only once, prior to task assignment with the static case.

Comparisons are also made on the scheduling length and energy consumption of our adaptive schemes with two message exchange methods and the EB-CNPT over an increasing number of available nodes. For our scheme, the scheduling length includes the time spent on communicating messages between the consumer and seller nodes. Figure 3 shows that the scheduling length would decrease by increasing the number of available node as the tasks may be allocated over a larger number of nodes. The scheduling length for real-time scheduling is higher or equivalent to any static task scheduling due to the time taken to exchange messages. Among the different message exchange mechanisms, the centralized scheme results in the lowest scheduling length as the only additional latency is due to the communication of messages to the consumer. The distributed scheme results in the highest scheduling length due to the linear waiting "back-off" time applied prior to a seller communicating the winning bid.

As shown in Figure 4, the energy consumption of nodes increases by increasing the available nodes due to increasing the communication cost for migrating data among the nods .The energy consumption of the adaptive schemes are slightly lower than static task allocation method since in our price formulation the computation cost is considered as well as the communication cost. Among the message exchange methods applied for adaptive scheme, the centralized method results in the higher communication overhead and hence the higher energy consumption; however the communication overhead has been considerably reduced in distributed

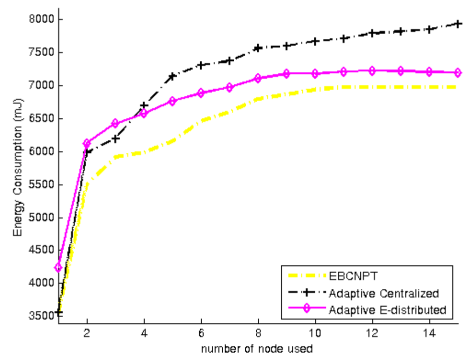

Figure 4. Energy Consumption vs Number of Nodes used

winner determination method and it results in lower energy consumption.

Node failure during the online task assignment phase is simulated. For the static case, rescheduling of the whole task graph is performed. This is compared with the adaptive recovery phase. Node failure has been simulated to occur at different times from the start time of task assignment. Figure 5 shows the performance of our adaptive scheme compared to the static case in terms of scheduling length. Results exhibit a significantly lower schedule length and therefore reduced energy consumption as shown in Figure 6. When node failure occurs during the early stages of task assignment, the schedule length is almost constant as not many tasks have been completed. When node failure occurs during the middle phases of the task assignment, the schedule length increases due to the number of uncommunicated dependencies resulting in the rescheduling and redeployment of many tasks. However, node failure occurs during the later times, many of the tasks have been completed and their dependencies have been communicated and therefore do not need to be redeployed.

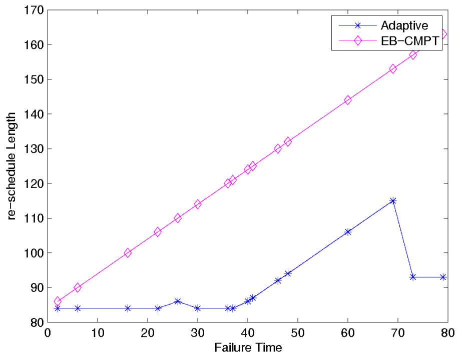

Figure 5. Scheduling length vs Failure time

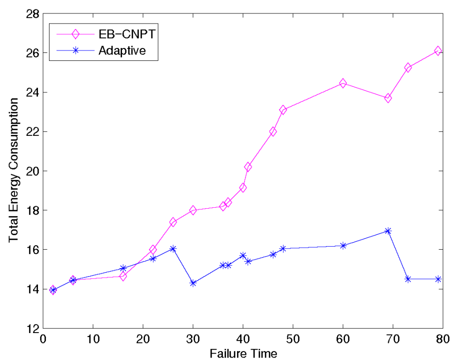

Figure 6. Energy Consumption vs Failure time

6. Conclusion

This paper has proposed online task allocation schemes in WSNs that improves the energy balance amongst nodes and maximizes network lifetime. The price formulation used continuously adapts to changes of the availabilities of resources and this scheme accommodates for the node failure during task assignment via a recovery phase. Two message exchanged mechanisms between the nodes and task allocator have been proposed with the goal of reducing energy consumption and overhead. Simulation results have shown the performance of the proposed online schemes to deal with uncertain situation of WSN.

Acknowledgment

This work was supported in part by the Agency of Science, Technology and Research in Singapore, under SERC Grant No.R-263-000-403-305.

References

[1] A. Tolstikov and J. Biswas and C.-K. Tham and P. Yap, *Eating activity primitives detection - a step towards ADL recognition*, 10th International Conference on e-health Networking, Applications and Services, 2008 (HealthCom 2008), 7-9 July 2008, Page(s):35 - 41

[2] L.K. Goh and V. Bharadwaj, *An energy-balanced task scheduling heuristic for heterogeneous wireless sensor networks*,Proceedings of 15th International Conference of High Performance Computing (HiPC 2008)

[3] Y. Yu and V.K. Prasanna, *Energy-balanced task allocation for collaborative processing in wireless sensor networks*, Source: Mobile Networks and Applications, Version 10, No. 1-2, Pages 115-131, Feb.-April 2005.

[4] Y. Tian, Y. Gu andE. Ekici and F. Ozguner, *Dynamic critical-path task mapping and scheduling for collaborative in-network processing in multi-hop wireless sensor networks*, 2006 International Conference on Parallel Processing Workshops, p 8 pp., 2006.

[5] R. Poornachandran and H. Ahmad and H. Cam, *Energy-Efficient Task scheduling for Wireless Sensor Nodes with Multiple Sensing Unit*, Conference Proceedings of the 2005 IEEE International Performance, Computing and Communications Conference (IEEE Cat. No.05CH37653), p 409-14, 2005

[6] F. Yao and A. Demers and S. Shenker, *A scheduling model for reduced CPU energy*, IEEE Annual Foundations of Computer Science, Pages 374-382, 1995.

[7] L. Anand and D. Ghose and V. Mani, *ELISA: an estimated load information scheduling algorithm for distributed computing systems*, Computers and Mathematics with Applications, Version 37, No. 8, Pages 57-85, April 1999

[8] Q. Zheng and C.-K. Tham and V. Bharadwaj, *Dynamic load balancing and pricing in grid computing with communication delay*,Grid Economics and Business Models, September 2008.

[9] Y. Tian, E. Ekici and F. Ozguner, *Energy-constrained task mapping and scheduling in wireless sensor networks* IEEE International Conference on Mobile Ad-hoc and Sensor Systems, 2005

[10] A. Viswanath and T. Mullen and D. Hall and A. Garga, *A. MASM: a market architecture for sensor management in distributed sensor networks* Proceedings of the SPIE - The International Society for Optical Engineering, v 5813, n 1, p 281-9, 28 March 2005

[11] T. Hagres and J. Janecek, *A high performance, low complexity algorithm for compile-time job scheduling in homogeneous computing environments*, Proceedings 2003 International Conference on Parallel Processing Workshops,2003

[12] Z. Wang and Y. Yan and P. Jia and S. Wang, *Market-based adaptive task scheduling for sensor networks*, 2006 International Conference on Wireless Communications, Networking and Mobile Computing, 2006