Maximizing System Value among Interested Packets While Satisfying Time and Energy Constraints

Shu Lei, S.Y. Lee, Wu Xiaoling, Yang Jie
Department of Computer Engineering
Kyung Hee University, Korea
{sl8132, sylee, xiaoling, yangjie}@oslab.khu.ac.kr

Abstract: Data filtering is an important approach to reduce energy consumption. Following this idea, Interest is used as a constraint to filter uninterested data in sensor networks. Within these interested data some are more valuable than others. Sometimes among these interested data, we hope to process the more important data first. By using Reward to denote the important level of data, in this paper, we present a packet scheduling algorithm by considering four constraints (Energy, Time, Reward, and Interest) simultaneously. Based on simulation result, we find out that our ETRI-PS packet scheduling algorithm can substantially improve the information quality and reduce energy consumption.

1. Introduction

Conventional research that is proposed to make processor use energy more efficiently, such as Dynamic Voltage Scaling, has been utilized in all kinds of embedded systems. Since wireless communication is the main energy consumer, by extending DVS's concept into communication system, Dynamic Modulation Scaling has been proposed to schedule packet transmission [1]. The key idea is to let radio transmit packets with a lower transmission rate to reduce the energy consumption while still meeting all deadlines. Similar researches [2, 3] also follow this approach by applying lazy scheduling algorithm. These researches focus on minimizing energy consumption of a set of packets by delaying the finish of transmission till the deadline. A common drawback is that they only consider packets that already exist in the buffer, but do not provide threshold or constraint to filter and reduce the coming packets. Another research trend is presented in [4, 5]. Cosmin Rusu, *et al.* consider **Energy**, **Time**, and **Reward** these three constraints simultaneously while Reward denotes the important level of task. They believe that in some overload systems, instead of processing several unimportant tasks that just consume a small amount of energy, it is more meaningful to process one valuable task which will consume more energy. In this **ETR** scheduling algorithm, whenever a new task is processed, it must have the highest ratio (reward value / energy consumption of this task) among all waiting tasks. Later on, in paper [6] Fan Zhang *et al.* extend this ETR algorithm for packet scheduling and present three different transmission algorithms.

Data filtering is also an important approach to reduce energy consumption in sensor networks. Generally, a huge amount of data can be created by a large sensor network. However, in most of the time only the data of some sensor nodes that related to the users' purpose is really valuable. In [7, 8, and 9], data-centric approach is proposed for power efficient data routing, gathering and aggregation. **Interest** is introduced as a constraint which is used to filter and reduce the unnecessary data. In these researches authors simply consider that all these packets have the same important level, but actually among these interested packets, some of them may be more important than others. For example, users are interested in the data of several sensor nodes which are used to monitor one object. The data created by the sensor nodes which are close to the observed object have more valuable information than the data created by the sensor nodes which are far from this object. Therefore, if we can introduce the **Reward** into these interested packets, we are able to select out and process the most important and valuable packet first. In this paper we present a new packet scheduling algorithm, namely, **ETRI-PS (Energy, Time, Reward,** and **Interest)**. Within this algorithm each packet has four parameters as follows: (1) energy consumption; (2) processing time; (3) important level; and (4) interest level.

In section 2 we present problem model. In section 3 we describe ETRI-PS scheduling algorithm. We present simulation work in section 4. And this paper is concluded in section 5.

2. Problem model

We have one cluster in the heterogeneous sensor networks that is deployed as figure 1. Sensor nodes in area A, B, and C are used to monitor three different objects denoted by the triangles. We suppose that a user wants to know the information about the objects in area A and B. After querying and sensing, only the data collected by the sensor nodes which are located in area A and B can be accepted by the cluster head. Data from the area C will be rejected, because the user is not interested in them. If we look inside area A, we can find that the data sensed by the sensor nodes which are relatively closer to the observed object have the higher valuable information. Therefore, we consider these sensor nodes' data more important than others'. Then, whenever the cluster head receives the packets from sensor nodes, it will receive the most valuable packet among several interested packets first.

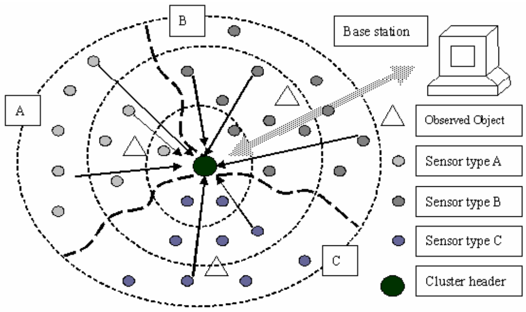


Fig. 1. Different sensor nodes send different packets to cluster head simultaneously

We define the interested areas as A $\subseteq \{A_1, A_2, \ldots, A_M\}$. From each interested area A_x the cluster head can accept a subset of packets $P_x \subseteq \{ P_{x,1}, P_{x,2}, \ldots, P_{x,N} \}$.The processing time of the packet $P_{x,y}$ is denoted by $T_{x,y}$. Associated with each packet $P_{x,y}$, there is an Interest value $I_{x,y}$ and a Reward value $R_{x,y}$. Interest value is used to distinguish the interested packets from different areas. Reward value is used to denote the important level of this packet. The larger reward value means the higher important level. These four constraints of algorithm are defined as follows:

- ❑ The *energy constraint* imposed by the total energy E_{max} available in the cluster head. The total energy consumed by accepted packets should not exceed the available energy E_{max}. Whenever cluster head accept one packet, the energy consumption $E_{x,y}$ of this packet should not exceed the remaining energy RE.

❑ The *time constraint* imposed by the global deadline D. The common deadline of this user's data query is D. Each packet that is accepted and processed must finish before D.

❑ The *interest constraint* imposed by the interest value threshold IT. Each packet that is accepted must satisfy the interest value threshold $IT_{min} \leq I_{x,y} \leq IT_{max}$.

❑ The *reward constraint* imposed by the *value ratio* $V_{x,y}$ ($V_{x,y} = R_{x,y} / E_{x,y}$) between reward value $R_{x,y}$ and energy consumption of packet $E_{x,y}$. The larger $V_{x,y}$, the packet has, the more valuable the packet is.

The ultimate goal of ETRI-PS is to find out a set of packets $P = P_1 \cup P_2 \cup \ldots \cup P_M$ among interested packets to maximize the *system value,* which is defined as the sum of selected packets' *value ratio* $V_{x,y}$. Therefore, the problem is to

maximize $$\sum\nolimits_{x \in A, y \in P} V_{x,y} \quad (1)$$

subject to $$\sum\nolimits_{x \in A, y \in P} E_{x,y} \leq E_{\max} \quad (2)$$

$$\sum\nolimits_{x \in A, y \in P} T_{x,y} \leq D \quad (3)$$

$$IT_{min} \leq I_{x,y} \leq IT_{max} \quad (4)$$

$$x \in \mathrm{A} \quad (5)$$

$$\mathrm{A} \subseteq \{A_1, A_2, \ldots, A_M\} \quad (6)$$

$$y \in P_x \quad (7)$$

$$P_x \subseteq \{1, 2, \ldots, \mathrm{N}\} \quad (8)$$

Because of the $P = P_1 \cup P_2 \cup \ldots \cup P_M$, we can have the following formula as:

$$\sum\nolimits_{x \in A, y \in P} V_{x,y} = \sum\nolimits_{A_1, y \in P_1} V_{A_1, y} + \sum\nolimits_{A_2, y \in P_2} V_{A_2, y} + \ldots + \sum\nolimits_{A_M, y \in P_M} V_{A_M, y} \quad . \quad (9)$$

From formula (9), we can find that the real problem of ETRI-PS is to find out the subset of $P_x \subseteq \{1, 2, \ldots, \mathrm{N}\}$ to maximize the *system value* for each interested area A_x. Thus, the problem is to

maximize $$\sum\nolimits_{A_x, y \in P_x} V_{x,y} \quad (10)$$

subject to , $$\sum\nolimits_{y \in P} T_{x,y} \leq D \quad (11)$$

$$IT_{min} \leq I_{x,y} \leq IT_{max} \quad (12)$$

$$E_{x,y} \leq RE \quad (13)$$

$$x \in \mathrm{A} \quad (14)$$

$$\mathrm{A} \subseteq \{A_1, A_2, \ldots, A_M\} \quad (15)$$

$$y \in P_x \quad (16)$$

$$P_x \subseteq \{1, 2, \ldots, \mathrm{N}\} \quad (17)$$

Inequality (11) guarantees that the *time constraint* is satisfied. Inequality (12) guarantees that only the interested packets are accepted, and inequality (13) guarantees that the energy budget is not exceeded. In order to solve the problem that is presented by (10)-(17), we give the following ETRI-PS algorithm.

3. ETRI-PS packet-scheduling algorithm

Before sending the real data of a packet, sensor node can send its packet's parameters to cluster head by including them in a small packet, which just consumes very limited energy. We give a name to this kind of small packet as *Parameter Packet (PP).* There is a physical buffer that exists inside cluster head to store these *PPs*. After receiving these *PPs*, cluster head can decide which packet to be accepted based on these sent parameters. In this case, in sensor networks, except the cluster head, all the other sensor nodes which are going to send packets to the cluster head can logically be considered as a buffer, since all of these packets are waiting for the processing of cluster head. We regard this buffer as the First Tier Buffer (FTB). Actually the FTB is a logical concept for cluster head. The Second Tier Buffer (STB) is the buffer that physically exists inside cluster head. Therefore, we propose the Two Tiers Buffer model for wireless sensor networks as figure 2 shows. In terms of this TTB model, basically, we can define our ETRI-PS algorithm into these following steps:

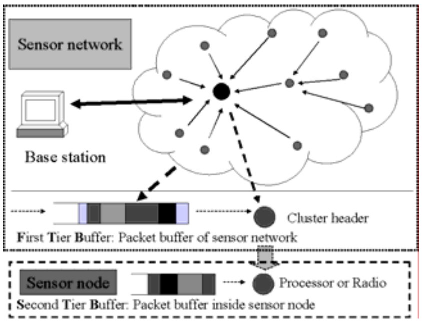

Sensor network
Base station
Cluster header
First Tier Buffer: Packet buffer of sensor network
Sensor node
Processor or Radio
Second Tier Buffer: Packet buffer inside sensor node

Fig. 2. Two Tiers Buffer (TTB)

Step 1: Initialization. After receiving PP $\subseteq$ {PP_1, PP_2, …, PP_N}, we assume that tables exist inside the cluster head for storing parameters of every packet *i* ($i \in$ PP): *energy consumption* $E_{x,y}$, *processing time* $T_{x,y}$, *reward value* $R_{x,y}$, and *interest value* $I_{x,y}$. For each PP_i, there are energy consumption for checking CE_i and a period of time for checking CT_i. We also use two structure arrays, *considered(i)* and *selected(i)* of size N, to store the information for all received *PPs*. Initially, we start with an empty schedule (*selected(i).status = false*) and no *PP* is considered (*considered(i).status = false*). The set of selected *PPs* (initially empty) is defined as S = {*(i)* | *selected(i).status = true*}. After selecting the *PPs*, cluster head accepts packets that are corresponded to these selected *PPs*. Therefore, packet's parameters can be expressed as *considered(i).*$E_{x,y}$, *considered(i).*$T_{x,y}$, *considered(i).*$R_{x,y}$, *considered(i).*$I_{x,y}$, *selected(i).*$E_{x,y}$, *selected(i).*$T_{x,y}$, *selected(i).*$R_{x,y}$, and *selected(i).*$I_{x,y}$. We define four variables: 1) *checking energy* ($\sum_{i \in PP} CE_i$) is used to store total energy consumption for checked PPs; 2) *checking time* ($\sum_{i \in PP} CT_i$) is used to store total processing time for checked PPs; 3) *processing energy* ($\sum_{i \in PP}$ selected(i).$E_{x,y}$) is used to store total energy consumption for processed packets; and 4) *processing time* ($\sum_{i \in PP}$ selected(i).$T_{x,y}$) is used to store total processing time for processed packets.

Step 2: In FTB, **we filter and accept packets based on the ETRI constraints**. A packet that can be accepted should satisfy all the following criteria:

❑ This packet's *PP* is not considered before (*considered (i).status = false*).

❑ The current schedule is feasible (*checking time* + *processing time*) ≤ *D*.

❑ By accepting this packet to current schedule, the energy budget is not exceeded (*checking energy* + *processing energy* + *considered(i).*$E_{x,y} \le E_{max}$).

❑ This packet is intentionally queried by user ($IT_{min} \le$ *considered(i).*$I_{x,y} \le IT_{max}$).

❑ Among all the *PPs* that satisfy the above criteria, select the one that has the largest *considered(i).*$V_{x,y}$ = *considered(i).*$R_{x,y}$ / *considered(i).*$E_{x,y}$.

After choosing the *PP*, cluster head can send Acknowledge back to accept new packet. In addition, for those packets which user is not interested in, their corresponded sensor nodes will discard them. In this case, we refuse and discard the unnecessary data; consequently, we can reduce the energy consumption.

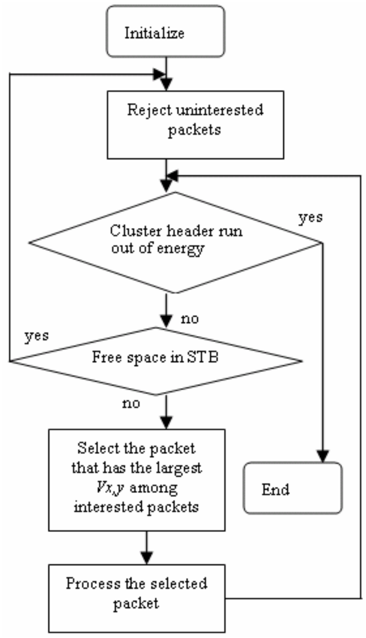
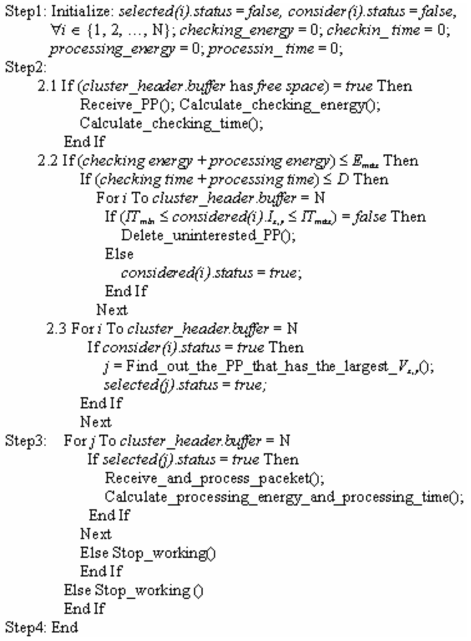

Initialize
Reject uninterested packets
Cluster header run out of energy
yes
no
yes
Free space in STB
no
Select the packet that has the largest Vx,y among interested packets
End
Process the selected packet

Fig. 3. Flowchart of ETRI-PS



```
Step1: Initialize: selected(i).status = false, consider(i).status = false,
       ∀i ∈ {1, 2, …, N}; checking_energy = 0; checkin_ time = 0;
       processing_energy = 0; processin_ time = 0;
Step2:
    2.1 If (cluster_header.buffer has free space) = true Then
            Receive_PP(); Calculate_checking_energy();
            Calculate_checking_time();
        End If
    2.2 If (checking energy + processing energy) ≤ $E_{max}$ Then
            If (checking time + processing time) ≤ D Then
              For i To cluster_header.buffer = N
                If ($IT_{min}$ ≤ considered(i).$I_{x,y}$ ≤ $IT_{max}$) = false Then
                  Delete_uninterested_PP();
                Else
                  considered(i).status = true;
                End If
              Next
     2.3 For i To cluster_header.buffer = N
           If consider(i).status = true Then
             j = Find_out_the_PP_that_has_the_largest_$V_{x,y}$();
             selected(j).status = true;
            End If
            Next
Step3:  For j To cluster_header.buffer = N
           If selected(j).status = true Then
             Receive_and_process_paceket();
             Calculate_processing_energy_and_processing_time();
            End If
            Next
            Else Stop_working()
            End If
        Else Stop_working ()
        End If
Step4: End
```

Fig. 4. Source code of ETRI-PS

Step 3: In STB, we transmit accepted packets to base station by using Dynamic Modulating Scaling. As the algorithm that has been presented in [1], let radio transmit packets with a lower transmission rate to reduce the energy consumption while still meeting all deadlines. Thus, in this step, we use the Dynamic Modulation Scaling to maximize the cluster head's lifetime.

Another aspect: Replace or drop a packet in the STB. A new packet is always accepted if possible. When receiving new *PP* from sensor node, if the STB is full, we can replace or drop a packet based on the following criteria:

❑ This packet's *PP* is selected (*selected(i).status = true*).

❑ Among all selected packet's *PPs*, find out the one that has the smallest *selected(i).*$V_{x,y}$ = *selected(i).*$R_{x,y}$ / *selected(i).*$E_{x,y}$.

❑ If this found one is not the new packet that is going to be accepted, we use this new packet to replace this found one, otherwise, we drop this new packet.

4. Simulation and discussion

In our simulation, we randomly deploy nine sensor nodes. And we randomly initialize these nodes with: *total energy* (scope: from 111 to 888), *buffer size* (scope: from 6 to 9). In addition, we design 8 different packets that are randomly initialized with the following four parameters: energy consumption (scope: from 3 to 10), processing time, reward value (scope: from 3 to 10) and interest value (scope: from 3 to 10). Eight of these nine sensor nodes are chosen to be the packet generators which randomly create eight different packets and send to the remaining one. The remaining one works as the cluster head. For this cluster head, we design four parameters: *total energy = 666*, *buffer size = 6*, and *interest threshold = 5*. The meaning of threshold is that we just accept the packets when their interest values are belonging to the top 5 among these 8 packets. These eight sensor nodes are organized into three groups based on their packets' interest values. The packets that have the interest values belong to {8, 9, 10} are considered as group A, the packets that have the interest values belong to {6, 7} are considered as group B, and the packet that have interest values belong to {3, 4, 5} are considered as group C. Therefore, the cluster head just accepts the packets from area A and B. In terms of energy consumption, we mainly consider the following two parts that have strong relationship with our proposed ETRI-PS, which are *processing energy* $\{E_{\text{(Returning ACK)}} + E_{\text{(Receiving packet)}} + E_{\text{(Processing)}} + E_{\text{(Broadcasting event)}} + E_{\text{(Listening)}} + E_{\text{(Accepting ACK)}} + E_{\text{(Sending packet)}}\}$ and *checking energy* $\{E_{\text{(Accepting event)}} + E_{\text{(Deciding)}}\}$. And the checking energy is designed to be 0.3, which is 10% of the minimum packet consumption 3. Besides ETRI-PS, we provide two different existing packet scheduling algorithms to run on cluster head for comparison:

1) Compared Algorithm one (CA 1) [1]:

a) In FTB: No *interest constraint* and *reward constraint*

b) In STB: Maximizing system lifetime (*Dynamic Modulation Scaling*)

The cluster head doesn't set any threshold to reduce the incoming packets, but just simply receives packets and relays them. Once it gets a packet, it will always process this packet just meeting its deadline.

2) Compared Algorithm three (CA 2) [6]:

a) In FTB: Maximizing reward value, but no *interest constraint*

b) In STB: Maximizing system lifetime (*Dynamic Modulation Scaling*)

The cluster head always accepts the packet that has the largest *value ratio* among several checked packets. Once it gets a packet, it will always process this packet just meeting its deadline.

We design the simulation parameters as follows: 1) *lifetime* of Cluster Head (CH), 2) *checking energy* of cluster head, 3) *processing energy* of cluster head, 4) *energy utilization* of cluster head (*energy utilization = processing energy / (checking energy + processing energy)*), 5) *processed packets number* by cluster head, 6) *total created packets* from sensor nodes, 7) *discarded packets* in sensor nodes, 8) *average interest value* per packet, 9) *average reward value* per packet.

From figure 5, we can find that for a given amount of energy, by using the *Dynamic Modulation Scaling*, the *lifetimes* of three different algorithms are almost same. As the result of the figure 6, the *checking energy* of ETRI-PS is much more than the *checking* *energy* of others. The reason is that we add the *interest constraint* in this ETRI-PS algorithm. Naturally, the energy that can be used to process packets is lower than others (*checking energy* + *processing energy* = E_{max}). This consequently causes relatively low *energy utilization* of ETRI-PS, as showed in figure 7. Even though the *energy utilization* of ETRI-PS is relatively lower than others, by using our ETRI-PS packet scheduling algorithm, we can still substantially reduce the energy consumption of whole sensor networks. The saved energy comes from the normal sensor nodes but not from the cluster head. By analyzing the figure 8, we can find that the *processing ratio* (*processing ratio = processed packets / total created packets*) of ETRI-PS is much lower than others; inversely, the *discarding ratio* (*discarding ratio = discarded packets / total created packets*) is much higher than others. The lower *discarding ratio* the sensor nodes have, the more uninterested packets the sensor nodes send. Thus, the more unnecessary energy is consumed. In conclusion, by using the ETRI-PS, the sensor nodes can reduce the unnecessary transmission of uninterested data to reduce the energy consumption. As we design the *interest threshold* to just accept the packets that have the larger interest values, therefore, the desired *average interest value* should be larger than that of other algorithms. Figure 9 shows that the *average interest*

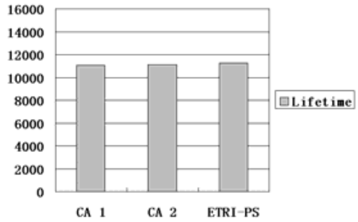


Fig. 5. Lifetime of cluster head

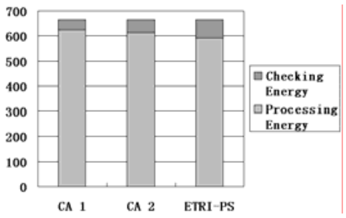


Fig. 6. Checking energy and processing energy

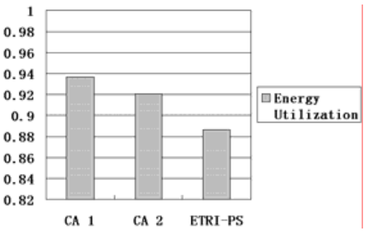


Fig. 7. Energy utilization of cluster head

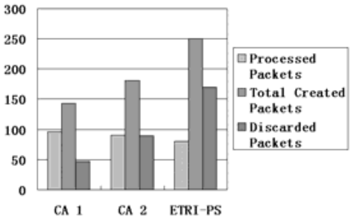


Fig. 8. Total created packets = processed packets + discarded packets

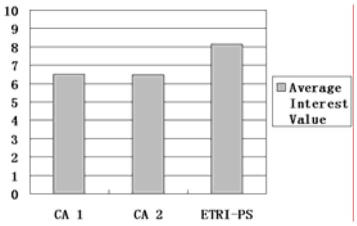


Fig. 9. Average interest value per packet

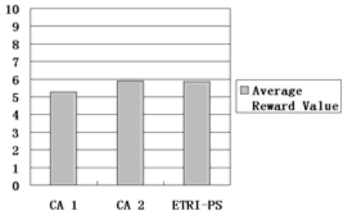


Fig. 10. Average reward value per packet

value of ETRI-PS is much larger than others, that means the ETRI-PS can exactly process the user interested packets well. Figure 10 shows the comparison among three algorithms' *average reward values*. In the algorithm CA 1, because we do not intentionally maximize the *value ratio* ($V_{x,y} = R_{x,y} / E_{x,y}$), as a result, the *average reward value* of CA 1 is relatively smaller than others. Compared with CA 2, even though we add the *interest constraint* to CA 2, the *average reward value*s of two algorithms are almost same. This means the ETRI-PS can inherit the original purpose of ETR well.

5. Conclusion

Packet scheduling algorithm for communication subsystem is a potential approach to reduce energy consumption of sensor networks. ETRI-PS provides us a prioritized transmission scheduling algorithm according to the transmitted data's important level. By using ETRI-PS packet scheduling algorithm, we can achieve the following contributions: (1) Use interest constraint as the threshold to filter the uninterested incoming packets to reduce the energy consumption; (2) Use reward constraint to choose the high quality information and minimize the queried packet number to minimize the energy consumption but still satisfy the minimum information requirement. As the simulation result shows, by using the Two Ties Buffer model and ETRI-PS packet scheduling algorithms, we can easily reduce energy consumption of sensor nodes and enhance the quality of queried information.

References

[1] C. Schurgers, V. Raghunathan, M.B. Srivastava, "Modulation Scaling for Real-time Energy Aware Packet Scheduling", *Global Communications Conference (GlobeCom'01)*, San Antonio, Texas 2001.

[2] B. Prabhakar, E.U. Biyikoglu and A.El Gamal, "Energy-efficient Transmission over a Wireless Link via Lazy Packet Scheduling", IEEE/ACM Transactions On Networking, Vol. 10, No. 4, August 2002.

[3] R. Rao, K. and Alex C. Snoeren, "Practical Lazy Scheduling in Sensor Networks", ACM, SenSys'03, November, Los Angeles, California, USA, 2003

[4] C. Rusu, R. Melhem, D. Mosse, "Maximizing Rewards for Real-Time Applications with Energy Constraints", ACM TECS, vol 2, no 4, 2003.

[5] C. Rusu, R. Melhem, D. Mosse, "Multi-version Scheduling in Rechargeable Energy-aware Real-time Systems", to appear in Journal of Embedded Computing, 2004.

[6] F. Zhang, S.T. Chanson, "Throughput and Value Maximization in Wireless Packet Scheduling under Energy and Time Constraints", 24th IEEE International Real-Time Systems Symposium**,** 2003.

[7] B. Krishnamachari, D. Estrin, S. Wicker, "Modeling Data-Centric Routing in Wireless Sensor Networks", proceedings of 6th international workshop on Modeling analysis and simulation of wireless and mobile systems, 2003.

[8] J. Beaver, M. A. Sharaf, A. Labrinidis, P.K. Chrysanthis, "Power-Aware In-Network Query Processing for Sensor Data", In the Proceedings of the 2nd Hellenic Data Management Symposium, September 2003.

[9] H.O. Tan and I. Korpeoglu, "Power Efficient Data Gathering and Aggregation in Wireless Sensor Networks", SIGMOD/PODS volume 32, number 4, December 2003.

[10] K. Kalpakis, K. Dasgupta, and P. Namjoshi, "Maximum Lifetime Data Gathering and Aggregation in Wireless Sensor Networks", To Appear in Proceedings of IEEE Networks'02 Conference, 2002.

[11] A. Kansal, D. Potter, M.B. Srivastava, "Performance Aware Tasking for Environmentally Powered Sensor Networks", ACM SIGMETRI-PSCS Performance Evaluation Review, 2004.

[12] J.F. Liu, P.H. Chou, N. Bagherzadeh, F. Kurdahi, "A Constraint-based Application Model and Scheduling Techniques for Power-aware Systems", Proceedings of the 9th International Symposium on Hardware/Software Codesign, pages 153-158, Copenhagen, Denmark, April 2001

This work is partially supported by the Korean Ministry of Commerce, Industry & Energy.