2011 IEEE International Conferences on Internet of Things, and Cyber, Physical and Social Computing

Energy Efficiency based on Quality of Data for Cyber Physical Systems

Luong Thi Thu Phuong, Nguyen Trung Hieu, Jin Wang, Sungyoung Lee, and Young-Koo Lee
Department of Computer Engineering, Kyung Hee University
Suwon, Korea
{luongthuphuong, nthieu, wangjin, sylee}@oslab.khu.ac.kr, yklee@khu.ac.kr

***Abstract*—Cyber-Physical Systems (CPS) are convergence of real physical world and computation world. CPS systems are rapidly becoming an important development strategies with broad application domains. The key requirements for CPS systems are to support multiple applications with many different resources at the same time in timeliness and also ensure energy efficiency. This paper presents the *Cyber-Physical integrated cloud computing* system called *Cyber-Physical Cloud* (CPC) to solve these key requirements. CPC architecture is designed for various applications such as health care, smart cars, traffic control, etc where multiple wireless sensor networks (WSNs) communicate with each other through mobile ad hoc network. Concurrently, CPC architecture can leverage cloud computing to process effectively computation and resource management by virtualization technique. Moreover, CPS systems are built on heterogeneous WSNs to collect and process the physical objects by obtaining data and thereafter processing queries. As a part of CPS systems, the key requirement for WSNs is to sustain a long lifetime because of limited power supplies. We work on real time preemptive query scheduling in WSNs with a focus on *quality of data (QoD)* to analysis *useful energy consumption* and *interference energy consumption*. We propose the *Energy efficiency Aware Quality of Data* (EAQD) method that runs on real time preemptive query scheduling to find the *expected QoD* for each query thus enabling energy consumption minimization. EAQD method also computes the transmitting/receiving slots needed for each query to avoid wasting slots. The simulation results show the our approach can reduce energy consumption up to 5 times as compares to previous approaches.**

***Keywords*-cyber-physical systems; energy efficiency; quality of data; real time query scheduling;**

I. INTRODUCTION

Nowadays, CPS systems have widely emerged as a new technologies generation with a broad scope of application domains. This is the development from simple systems (i.e., stand-alone embedded systems) to complex and heterogeneous networked systems that integrate communication and computation in which many types of embedded devices are used to sense, monitor, control, and track the real world. CPS systems is to enrich human-to-human, human-to-object, and object-to-object interactions in the physical world as well as in the virtual world. It is easy to realize that applications of CPS systems rapidly appear in every aspect of our life and naturally support more and more services such as health monitoring [1], [2], living assistant [3], robotics in underwater environmental monitoring [4], intelligent transportation systems [5], [6], [7], Siemens Industrial Automation Division industrial [8], social networks [9] etc. Based on CPS definition, we can merge the real world physical objects and the cyber computation space efficiently with high predictable ability, real time communication as well as high accuracy.

A CPS would adopt and even nurture the areas of WSNs due to more sensor inputs and richer network connectivity to be needed [10]. Actually, CPS is designed and deployed upon a large number of actuators and sensors which generate, collect, process and exchange the data from physical world, then the system computes, controls and takes intelligent decisions in Cyber world. A CPS is featured by multiple WSNs integration, heterogeneous information flow and smart decisions [10]. Hence, WSNs is an important part of CPS systems that bridge the virtual computing and real data through sensors and actuators [11].

Based on the necessary connections establishment between WSNs and CPS help us identify key requirements for WSNs which may be useful for CPS research and design.

As foundation of CPS, WSNs observe physical systems by acquiring data and processing queries. Thus, real time preemptive query and reply processing and communication as well as energy conservation are fundamental issues [12], [13], [14], [15]. Query processing in real time preemptive query scheduling protocols have to minimize the energy consumption as much as possible. The existing real time query scheduling methods that support effective prioritization between different query classes have not usually considered different QoD requirements from diverse applications and users. In the current real time scheduling schemes, the queries with different quality preferences are equally treated and are scheduled according to their priorities. Some query processing systems [16], [17], [18], [19] considered quality of data requirements, however they did not apply to real time preemptive query scheduling schemes which are very important for real time communication in WSNs. This is a strong motivation for our work. Therefore, our work works on the real time preemptive scheduling with a focus on QoD to analysis *useful energy consumption* and *interference energy consumption*. We propose the novel method, called *energy efficiency aware quality of data* (EAQD) scheme to determine the *expected QoD* for each query so that minimize the energy consumption in real time preemptive query scheduling.

With regard to CPS applications, how to support many applications with different quality requirements and different

978-0-7695-4580-6/11 $26.00 © 2011 IEEE
DOI 10.1109/iThings/CPSCom.2011.61

IEEE computer society

resources at the same time in timeliness as well as how to minimize the energy consumption are challenges today and even in the future. Cloud computing can provide resource as a service via virtualization technology [20]. The power resources provided by Cloud consists of computation resources (CPU capacity, memory size, disk space, and network bandwidth) and software environment (software, libraries, and tools) for CPS applications execution. With huge resource provided by Cloud computing, many complex applications can be completed within deadline time and meet the QoS requirement. That mean reducing the computation time of application, increasing the transmission time, storage capacity, and delivering the real-time services for end users. For example, u-life care applications integrated cloud computing not only improve the quality of health care service [21] but also can deliver various advantages such as access ability from anywhere at anytime, optimization using best resources for processing complex applications and large data storage ability, load balancing, load sharing, cost reducing, execution time minimization, etc [22], [23]. So we believe that WSNs integrated cloud computing is to be a proposed solution in this aspect. Therefore, in this paper we proposed the *cyber-physical system integrated cloud computing*, called *cyber-physical cloud* (CPC) architecture model that includes ubiquitous connectivity and virtualization. This provides dynamic virtual resource management enabling real time services and quality requirements.

In this paper we make the following key contributions:

- We propose the CPC architecture that integrated cyber-physical with cloud computing to support the multiple applications with many different resources at the same time in timeliness and also ensure effective power consumption. CPC architecture is designed with following features: effective virtual resource management, ubiquitous connectivity and multiple domain applications that go towards smart grid, smart living room and smart transportation. With this design, CPC architecture can support real time services, energy efficiency through power transmission and power distribution, as well as quality requirements.
- In WSN part, we propose EAQD method that finds the expected QoD for each query thus enabling energy consumption minimization. EAQD method runs on real time preemptive query scheduling schemes and also uses the aggregated query service [24] and tree network structure with a focus on QoD to analysis the useful energy consumption and interference energy consumption. EAQD method also computes the transmitting/receiving slots needed for each query and slots feedback will be sent to the scheduler in the real time scheduling schemes as to avoid wasting slots. The expected QoD for each query will be re-arrange and send back the planner in the real time scheduling schemes to re-plan.

The rest of paper is constructed as follows. The related work is outlined in section 2. This paper has two main parts: section 3 describes the CPC architecture and section 4 presents the EAQD method. In the section 4 we will explain detail the system models, then we show the overview of EAQD framework and we present the complete energy efficiency maximization method as well as the EAQD algorithm. The simulation results of our work is showed in section 5 and section 6 concludes this paper.

II. RELATED WORK

Firstly, so many CPS architecture models were made as emerged areas of research that are designed to satisfy multiple applications with multiple resources at the same time in timeliness and also energy efficiency guarantee. Based on different types of application, research activities gave different CPS architectures that satisfy various requirements. The mobile health monitoring system call Icare [1] were designed for the elderly. This work used wireless sensor body and smart phones support real time health monitoring and living assistant. In [5], [6], [7], they exploited applications in smart phones [25], for example GPS sensor, accelerometer sensor and WiFi to build the intelligent transport systems. Vtrack [5] provided the location and travel time estimations as well as traffic delay monitoring with high accuracy, reliablity and efficient energy consumption. Cooperative transit tracking [6] detected automatically when the user is riding in a transit vehicle even underground vehicles and determine whether the user in a bus or other vehicles. In [7], the Pothole Patrol system gathered data from sensor equipped vehicles and processing to asses road surface conditions and detect the road segments have potholes. In social network, CenceMe application [9] can automatically retrieve and publish sensing presence to social network using mobile phone. But these works have been met the limitations of resource if many applications involve health care, environment monitoring, tracking and traffic control, motion recognition etc are required at the same time. Further more, most of them did not consider the key feature of CPS is how to support these systems with energy consumption minimum and QoS requirements. Our CPC architecture is built to meet multiple applications, energy conservation and guarantee QoS requirements which can refer to many isolated services by connecting WSNs to cloud computing environments.

Secondly, in WSNs an important issue is to generate query, collect results and process them with a trade off between energy consumption and qualities in in-network query processing. Lots previous works studied this area [16], [17], [19], [26] in in-network query aggregation. Among these, QGEE [16] studies on QoD, this algorithm selects the active nodes to answer the queries in order to reduce the energy consumption of nodes with a negligible drop in quality of data. In [17], Sharaf et al. elaborates the TiNA framework works on top of existing in-network aggregation schemes. TiNA uses temporal coherence tolerances to save energy while maintaining specified quality of data. Similar in this way, Peng et al. brings out quality assessment models for objects and fusion operators in in-network data processing [26]. These above researches give the admission control improvement and the node selection that are very useful for our work. However, they only consider query processing quality more than query and communication in context real time priority query scheduling schemes.

In [18], the authors present the quality contracts which combine the two performance metrics: QoS and QoD. This study builds the QUTS algorithm that maximize total profit by performing the trade off between QoS and QoD. In QAS [19], author proposed the same quality contracts to define the quality awards and system profit. With system profit, QAS scheduling scheme allows user specify their QoD requirements to find the target qualities and execution order of queries thus enabling total system profit maximization. Our work is very close to these works.We adopted the quality and slot feedback method to design our EAQD scheme. Although these works have not considered these studies in real time priority scheduling protocols which cause energy consumption for interference between different query prioritization to evaluate energy efficiency. Our EAQD planned to adopt these limitations from these studies.

There are some researches about real time preemptive query scheduling schemes PQS [15], DCQS [27] that our EAQD works on top of them. These protocols allocate slots such that avoid transmission/recieving conflict, reduce the idle listening periods and energy consumption. But these protocols are not quality aware, thus they are not good for applications that require different qualities of queries. Our EAQD method would overcome this problem by taking quality of queries into account. In addition, we exploit query service in TinyDB and in-network query aggregation [24] that make our EAQD improve the energy efficiency and performance of queries.

III. CYBER-PHYSICAL CLOUD ARCHITECTURE

Firstly, our contribution is proposed CPS integrated cloud computing model called Cyber-Physical Cloud (CPC) that supports for multiple application systems such as u-health care systems, vehicle transportation, social network, etc.

Fig. 1 illustrates a cyber-physical system integrated cloud computing that comprises 2 components: the first part is communication and sensing environments and the second one is multiple cloud computing environments. CPC architecture is featured by mobility management, complex information flow and intelligent decisions. In other words, CPC architecture has ubiquitous connectivity and virtualization which provide which provide the multiple applications at the same time with QoS guarantees and efficient resource management as to minimize the energy consumption for our system.

A CPC is designed for health care monitoring (i.e., smart home, assisted living, body sensor network etc), social network applications (showing the presence of individual, sharing their information via Facebook, etc), smart moving cars (vehicle cloud called v-cloud) or intelligent transportation etc. These applications can communicate with each other, and directly access to resource of many clouds at the same time in timeliness.

A. Communication and sensing environments

In the communication and sensing core, our system involves multiple WSNs that form a wireless mobile ad hoc network. The communication and sensing part exploits absolutely all the characteristics of smart phone (i.e., Iphone 4, PDA, etc) and the smart embedded sensors/actuators to build up the heterogeneous flexible wireless networks that can support for so many spaces and applications.

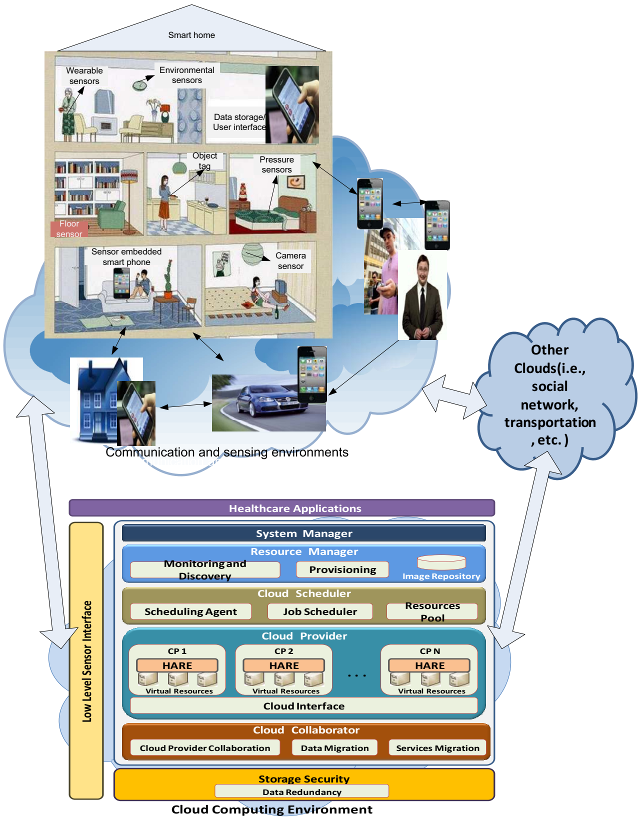

Fig. 1. CPC architecture

Smart living space: Our expectation is to cater the life demand for not only the elderly but also children, we design the smart living space. Our smart living space uses wireless body network with wearable sensors and medical devices such as electroencephalography (EEG), photoplethysmogram (PPG), electrocardiography (EKG), blood pressure sensor, etc to collect the physiological data from the elderly in order to detect emergency. Smart living space also uses environmental sensors such as temperature, light, humidity, etc to monitor the environment changes in which information can automatically store and give users. In addition, smart living space also uses camera, object tag, RFID reader, etc to track and detect the dangerous activities of elders/children in real time. For example, when the children are using knife or come near the flame, our design can detect quickly these cases and give the immediate alarm to their parent through mobile phone in timeliness.

Our smart living space can be consider as a personal health-care system as well as group health-care and obviously our system can offer medical guidance. By using the smart phone such as *Apple iPhone 4* that rapidly becoming the central computer and communication device in people'e lives, we exploit a rich set of embedded sensors (i.e., accelerometer, digital compass, gyroscope, GPS, microphone, and camera)to get the continuous sensor data and deploy personal sensing applications/group sensing applications. For example, distinct patterns within the accelerometer data can be exploited to automatically recognize different activities (i.e., running, walking, eating). This type of system relates these information to personal/group health goals when presenting real time feedback to the user or sharing sensing information, knowledge freely or with privacy protection. It looks like a community sensing applications. The medical guidance can be support in cloud computing that we will explain in next part.

B. Cloud computing environments

In cloud computing, we design the virtual machine (VM) scheduling with two scheduling level: the first one is the application scheduling level, and the second one is the resource scheduling level. In the applications scheduling level, we define the application requirements such as operating system, CPU performance, memory capacity, disk space, necessary software and libraries that provide configuration information for allocation of resources and creation of the VM on the server in the cloud. In the resource scheduling level, we apply the resource allocation algorithms to select the best currently available resources on the cloud for mapping of VMs onto cloud providers in order to minimize the total execution time and to maximize the QoS delivered. Via the two levels scheduling, the application can obtain the required resources for processing and the results are real-time support to the users.

Integrating CPS with cloud computing environment that can provide resource as services via virtualization technology in the form of virtual machine is one of the ways to reduce power consumption. This allows to drop the costs to manager the cloud environment. Additional, the virtual machine scheduling with resource scheduling level is to minimize the number of physical node on cloud provider serving the application, whereas idle physical machine on cloud provider are switched off in order to decrease power consumption.

In the scheduling process, if the application requirements are not met by a cloud provider or the Cloud Provider that are running VMs be die. In this case, the Cloud Provider collaborator with migration technology will be used to guarantee QoS requirements. In this process, the VM will be live migration to another Cloud Provider and continuous run on.

For example health care applications, we choose the health care application called Human Activity Recognize Engine (HARE) that can detect the human activity such as eating, sleeping, walking, ... of patients. In this scenario, the patients' data is real-time collected from sensors and cameras, then being uploaded to the cloud computing environment. Inside the cloud computing, cloud scheduler chooses the best of resources for dynamically creates and deploys VMs following the application requirements. After finishing successful creates and deploys VMs, Cloud Scheduler maps health care applications HARE to VMs, so that HARE can be executed with real-time data as input and the results will be delivered to end users via Software as a Service. Additionally as we said above, it can provide the medical knowledge database for doctors and other users when they access the private/public cloud.

IV. EAQD METHOD

A. Network Model

In this paper, we consider two parts: the CPC model and EAQD method. We have already shown the CPC model above. Here we will go deep into EQD method that makes energy efficiency for WSNs. In EAQD method, we introduce a WSN as a multi-hop network that consists of spatially distributed nodes that are located in a smart home or in a car. Our method works on real time preemptive query scheduling scheme that needs to construct a conflict-free schedule for query execution. In order to facilitate this, the WSN model is according to the IC graph [27], [15]. Fig.2 illustrates IC graph has communication edges and interference edges to determine two transmission can be scheduled concurrently (conflict-free) or not. A node can use RID [28] to determine its adjacent edges. In this WSN model, we assume that clocks are synchronized. These nodes use multi-hop communication to report their data due to the short transmission range and limited transmission power of each node. The communication may be overwhelmed by collisions if nodes transmit packets simultaneously or due to interference.

Our query processing system uses the tree network structure due to simplicity, duplicate-freeness and communication efficiency. In addition, our system assumes the network on which it is running to be stationary. In a tree network, the sink node is the root node. All the other nodes should report their query results to their parents. The parent of a node serves as the router that aggregates the data of the node. A neighbor of a node is one within this node's transmission/recieving range. So the parent of a node is also its neighbor. Finally, a node can shares the same parent with others.

B. Query Model

In our work, we just only consider the aggregation query [24], [12] to improve the energy efficiency and performance of queries. For example the maximum temperature query and the average humidity in a building induce the same workload in the network [24]: each node receives a packet from every child, and then sends a packet to its parent. For maximum query, the outgoing packet includes the maximum value of the data reports from itself and its children. For average query, the packet includes the sums of the values and the number data sources that contributed to the sum. The following query shows an aggregation query, which requires the nodes report the average light. The query partitions motes in the building according to the room where they are located and updates are delivered every 30s.

```
SELECT AVG(light), room FROM sensors
GROUP BY room
SAMPLE PERIOD 30s
```

A common query model in which source nodes generate periodically data reports. Let us assume that a query q is characterized by these parameters: a set of sources that respond to a query, a function for in-network aggregation, the start time, the query period D_{q} and a static priority. A node needs several slots to transmit its data (aggregated) report to its parent.

The query lifetime is the number of epochs from the time of query injection into network to the time when the query stops running.

C. Query Processing in real time preemptive scheduling

Fig. 2 shows a query processing model in-network data aggregation as in [24]: a query is disseminated its parameters to all nodes by a users through a base station or smart phone which is connected to sink node. The other sensor nodes begin to process and schedule the query and generate query results. Each non-leaf node waits to receive the data reports (results) from its children, produces a new data report by aggregating its data with the children's data report to generate a query result, after that sending its query result to its parent.

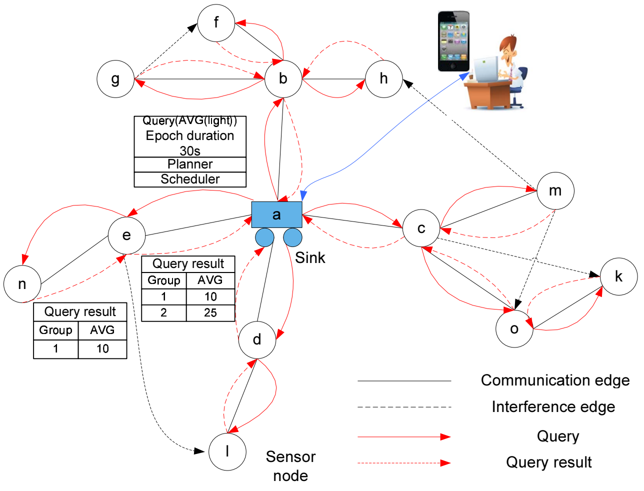

Fig. 2. IC graph and query processing model

To process query in scheduling, there are many research activities have been done [29], [30], [31]. Our work analyzes energy consumption in the real time preemptive query scheduling protocols such as PQS [15], DCQS [27]. These protocols have two components which our approach relies on: a planner and a scheduler. For each query, a planner constructs a transmission plan which plans a ordered sequence of steps based on set of conflict-free transmissions. The length L of a planner is minimized to reduce the query latency. Then, a scheduler dynamically determines which time slots for each step should be executed. The scheduler maintains a record of the start time, period and priority of all admitted queries. In a scheduler, it maintains two queues: *release* queue and *run* queue to guarantee different prioritization of queries while meeting their deadlines. If there is a transmission in the step in which queries are conflict-free with any current queries in run queue, they will be moved from release queue to run queue for transmission. Our work will build energy and quality problems related to this scheduling.

D. EAQD framework overview

We show the EAQD framework overview in Fig. 3 before going to more detail the EAQD method. Firstly, EAQD algorithm gets the data from the scheduling protocol and QoD parameter, then the energy efficiency is established to compute the maximum value, from that calculate the expected QoD of a processed query that should be. Secondly, EAQD will re-assign transmission steps depending on expected QoD of a processed query compare to its high priority queries. Finally, EAQD method provides also the number of transmitting/receiving slots needed for the scheduler to avoid wasting slots. We will explain obviously in latter parts.

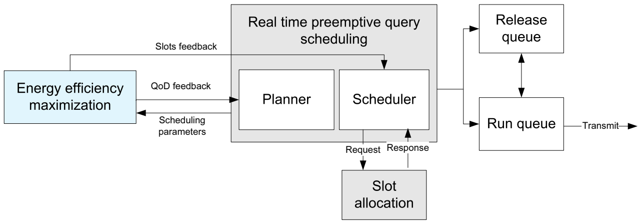

Fig. 3. EAQD framework

E. Energy efficiency maximization based on QoD

As we defined above, for query q the energy consumption has two components: the *useful energy* consumption $E_{\mathrm{u}}(q)$ and the *interference energy* consumption $E_{\mathrm{i}}(q)$. We calculate energy by evaluating consumed time intervals that are calculated with QoD of each processed query. Q is an quality of data defined different values among users for each query.

1) Useful energy consumption: For a current processed query on node n within T length of time, the useful energy consumption is estimated from power and node running time. We should consider energy in the different operations of query execution, the energy consumption is computed as follow:

$$E_{\mathrm{u}} = Ep \cdot \sum T_i \cdot P_i \tag{1}$$

where Ep is total number of epochs that need to process the query [24], T_i is the execution time of each type of i^{th} operation per epoch and P_i is the corresponding power. With T_i, we calculate four execution states of operator include:

- The query evaluation time is calculate as:

  $$T_{\mathrm{eval}} = Q \cdot \sum_{k=1}^{n} T_k - T_0 \tag{2}$$

  where T_k is the execution time of the k^{th} operator, i.e., temperature sampling operator, light sampling operator and non-sensory data sampling operator have different execution times that can be measured offline. T_0 is the overlapping time between the query execution and the communication on node n.

- Time to receive the results from its children:

$$T_{\mathrm{recv}} = Q \cdot N_{\mathrm{ch}} \cdot t_{\mathrm{s}} \tag{3}$$

where t_{s} is a slot time duration and N_{ch} is size of results from children nodes because node *n* aggregates the results of its children. Suppose that a result is transmitted/received in t_{s}.
- Time to transmit the results by node *n* to its parent, called T_{trans} is estimated by subtracting receive time to communication time. In there, the communication time is estimated by communicated results have size is (N_{ch}+1) in-network aggregation because node *n* needs to aggregate a result from each of its children and then send to its parent. Consequently, T_{comm} is calculated as:

$$T_{\mathrm{comm}} = (N_{\mathrm{ch}} + 1) \cdot t_{\mathrm{s}} \tag{4}$$

From that, node *n* needs transmission time as:

$$T_{\mathrm{trans}} = T_{\mathrm{comm}} - T_{\mathrm{recv}} = (N_{\mathrm{ch}} - Q \cdot N_{\mathrm{ch}} + 1) \cdot t_{\mathrm{s}} \tag{5}$$

- Finally, the sleep time is estimated:

$$T_{\mathrm{sl}} = T - (T_{\mathrm{eval}} + T_{\mathrm{recv}} + T_{\mathrm{trans}}) = T - (T_{\mathrm{eval}} + T_{\mathrm{comm}}) \tag{6}$$

The power consumption composes of P_{eval}, P_{trans}, P_{recv} and P_{sl} according to the each time T_i on node *n* for a single query *q*. Firstly, we have the power consumption in evaluation time that is equal power consumption in active mode P_{act} (given as a device parameter). Secondly, the transmit power consumption:

$$P_{\mathrm{trans}} = \frac{1}{D_{\mathrm{q}}} \cdot P_{\mathrm{trans/packet}} \cdot M_{\mathrm{q,n}} \tag{7}$$

And receive power consumption:

$$P_{\mathrm{recv}} = \frac{1}{D_{\mathrm{q}}} \cdot P_{\mathrm{recv/packet}} \cdot \sum_{c \in child(n)} M_{\mathrm{q,c}} \tag{8}$$

where $\frac{1}{D_{\mathrm{q}}}$ is the rate of query *q*, $M_{\mathrm{q,n}}$ is the maximum number of packets transmitted by node *n* to its parent and $M_{\mathrm{q,c}}$ is the maximum number of packets a child *c* transmits to node *n* to satisfy the workload demand of query *q*. $P_{\mathrm{recv/packet}}$ and $P_{\mathrm{trans/packet}}$ are power consumed in receiving and transmitting a packet, respectively and are equal P_{act}.

With above analysis, we have the useful energy consumption equation as below:

$$E_{\mathrm{u}} = Ep(T_{\mathrm{eval}}P_{\mathrm{act}} + T_{\mathrm{recv}}P_{\mathrm{recv}} + T_{\mathrm{trans}}P_{\mathrm{trans}} + T_{\mathrm{sl}}P_{\mathrm{sl}}) \tag{9}$$

Or

$$E_{\mathrm{u}} = Ep(x_1 \cdot Q - x_2) \tag{10}$$

where

$$x_1 = (P_{\mathrm{act}} - P_{\mathrm{sl}}) \sum_{k=1}^{n} T_k + N_{\mathrm{ch}} t_{\mathrm{s}} (P_{\mathrm{recv}} - P_{\mathrm{trans}}) \tag{11}$$

and

$$x_2 = (P_{\mathrm{act}} - P_{\mathrm{sl}})T_0 - P_{\mathrm{trans}} t_{\mathrm{s}} (N_{\mathrm{ch}} + 1) - P_{\mathrm{sl}} T + P_{\mathrm{sl}} t_{\mathrm{s}} (N_{\mathrm{ch}} + 1) \tag{12}$$

2) Interference energy consumption: In real time query scheduling, these schemes support preemptive queries such that the current query can be blocked by the higher priority queries. Under that, a query *l* with higher priority than *q* interferes with *q* for at most $\frac{W_{\mathrm{q}}}{D_{\mathrm{l}}} w_{\mathrm{s}}$ slots where W_{q} is the response time of the preemptable part of query *q* and w_{s} is the worst-case interference of an instance of query *l* on an instance of query *q*. By adopting the scheduling policies that our work runs on, we can find the worst-case interference. Thus, the worst-case interference energy consumption of *q* cause by set of its all higher priority queries *H(q)* is estimated as the following equation:

$$E_{\mathrm{i}} = Ep \cdot t_{\mathrm{s}} \cdot P_{\mathrm{act}} \cdot \sum_{l \in H(q)} [\frac{W_{\mathrm{q}}}{D_{\mathrm{l}}}] \cdot w_{\mathrm{s}} \tag{13}$$

or

$$E_{\mathrm{i}} = Ep \cdot x_3 \tag{14}$$

where

$$x_3 = t_{\mathrm{s}} \cdot P_{\mathrm{act}} \cdot \sum_{l \in H(q)} [\frac{W_{\mathrm{q}}}{D_{\mathrm{l}}}] \cdot w_{\mathrm{s}} \tag{15}$$

3) Energy efficiency maximization: To be simple, our work is to minimize the energy consumption in a WSN, so we transform this work into energy efficiency optimization which is the ratio of the useful energy consumption and total energy consumption $R = \frac{E_{\mathrm{u}}}{E_{\mathrm{u}} + E_{\mathrm{i}}}$.

As we mentioned above, QoD is different among users for each query. For example, some users would like to get fresh data slightly late (i.e., prefer high QoD) if not possible to have fresh data fast, however others would like to get the answer very fast, even if they correspond to slightly stale data (i.e., high prefer QoS). Based on the real time preemptive query scheduling scheme, the higher priority queries advance on their dominant rights compared to this current lower priority query to execute in some steps distance (slots). For these lower priority queries, we punish amount of QoD as to attempt the trade off between energy consumption and quality of data. We don't punish QoS of low priority query to guarantee the its response time that satisfied in the real time preemptive scheduling scheme. So based on this, we propose a trade off between QoD and energy consumption as following formulation:

$$R = \frac{Ep(x_1 \cdot Q - x_2)}{Ep(x_1 \cdot Q - x_2) + Ep \cdot x_3 - (\alpha \cdot Q + \beta)} \tag{16}$$

Here, we use $(\alpha \cdot Q + \beta)$ is a linear function of punishment to trade off with total energy consumption. Now, our work is to maximize the energy efficiency to find out the expected quality of data and reduce the energy consumption for each query. Thus, energy efficiency turns into searching for a pair of *(Ep,Q)* such that R is optimized. In essence, this problem is calculating extreme value of two bivariate functions. And we compute the partial derivatives to solve this problem. The solution of our method optimization is close to the *Profit* maximization problem in [19]. The first partial derivatives

($\frac{\partial R}{\partial Q} = 0$) and ($\frac{\partial R}{\partial Ep} = 0$) are necessary conditions for (Q^m,Ep^m) to be extreme value point of R in which Q^m becomes expected quality of data of current processed query. The first spatial derivatives of energy efficiency function are calculated as follows:

$$\frac{\partial R}{\partial Q} = \frac{x_1 x_3 Ep^2 - (\beta x_1 + \alpha x_2)Ep}{(Ep(x_1 Q - x_2) + Epx_3 - (\alpha Q + \beta))^2} \tag{17}$$

$$\frac{\partial R}{\partial Ep} = \frac{-\alpha x_1 Q^2 - \beta x_1 Q + \alpha x_2 Q + \beta x_2}{(Ep(x_1 Q - x_2) + Epx_3 - (\alpha Q + \beta))^2} \tag{18}$$

The second spatial derivatives of energy efficiency are calculated as the following equations:

$$\frac{\partial^2 R}{\partial Q^2} = \frac{-2(x_1 x_3 Ep^2 - \alpha x_2 Ep - \beta x_1 Ep)(x_1 Ep - \alpha)}{(Ep(x_1 Q - x_2) + Epx_3 - (\alpha Q + \beta))^3} \tag{19}$$

$$\begin{aligned} \frac{\partial^2 R}{\partial Q \partial Ep} &= [(2x_1 x_3 Ep - \alpha x_2 - \beta x_1)(x_1 Q \cdot Ep - \\ & x_2 Ep + x_3 Ep - \alpha Q - \beta) - 2(x_1 Q - x_2 + x_3) \\ & (x_1 x_3 Ep^2 - \alpha x_2 Ep - \beta x_1 Ep)] \;/\; (Ep(x_1 Q - x_2) \\ & + Epx_3 - (\alpha Q + \beta))^3 \end{aligned} \tag{20}$$

$$\begin{aligned} \frac{\partial^2 R}{\partial Ep^2} &= -2(-\alpha x_1 Q^2 - \beta x_1 Q + \alpha x_2 Q + \beta x_2) \\ & (x_1 Q - x_2 + x_3) \;/\; (Ep(x_1 Q - x_2) + Epx_3 \\ & - (\alpha Q + \beta))^3 \end{aligned} \tag{21}$$

With above first spatial derivatives, our method has finding ability the extreme point (Q^m,Ep^m), that enables the energy efficiency ratio for each query get the maximum value by checking the extreme condition: if $X = \frac{\partial^2 R}{\partial Q^2} \cdot \frac{\partial^2 R}{\partial Ep^2} - \frac{\partial^2 R}{\partial Q \partial Ep}^2 > 0$, and $\frac{\partial^2 R}{\partial Q^2} < 0$ or $\frac{\partial^2 R}{\partial Ep^2} < 0$ then the energy efficiency to be a maximum one, otherwise the maximum point is not found or exceeds the quality threshold. In no extremum case, (Q^m,Ep^m) will be set as lowest or highest possible quality based on which one makes the better energy efficiency.

F. EAQD Algorithm

Input:

- Set of different priority queries q_i{i=1...n}
- Ordered sequences of transmission steps that execute queries in the planner
- Allocated transmission steps in current slots that execute queries in the scheduler

Output:

- Expected QoD for each query such that energy consumption is minimum
- Re-assign transmission steps into current slots in the run queue and release queue
- Number of needed transmitting/receiving slots for each query

```
1: For $i = 1$ to n do Compute $\frac{\partial R_i}{\partial Q_i} = 0$; $\frac{\partial R_i}{\partial Ep_i} = 0$; $\frac{\partial^2 R}{\partial^2 Q} = X_1$; $\frac{\partial^2 R}{\partial Q \partial Ep} = X_2$; $\frac{\partial^2 R}{\partial^2 Ep} = X_3$;
2: If ($X = X_1 X_3 - X_2^2 > 0$) and $(Q_i, Ep_i) < Threshold$ Then
    3: $WR_i = D_{\text{tr}} \cdot Q_i^m$
    4: $WT_i = D_{\text{re}} \cdot Q_i^m$
5: Else ($Q_i^m$,$Ep_i^m$) is set as lowest/highest possible quality;
End
6: Start/resume($q_i$): if $Q_i^m > Q_{H(q_i)}$ then
    7: run queue = run queue $\bigcup \{q_i\}$;
    8: release queue= release queue - $\{q_i\}$;
9: Else preempt($q_i$):
    10: run queue = run queue - $\{q_i\}$;
    11: release queue = release queue $\bigcup \{q_i\}$;
12: Send $WR_i$ and $WT_i$ of query $q_i$ to the scheduler to allocate transmitting/receive slots needed;
```

In EAQD algorithm, line (1-2) compute first spatial derivatives to find the expected (Q_i^m,Ep_i^m) for query q_i and second spatial derivatives to check the conditions whether (Q_i^m,Ep_i^m) is the optimum point or not. If both of the conditions and the expected QoD satisfy feasible conditions of maximization and threshold of QoD in the range from 0.1 to 1, then continue to compute number of transmitting and receiving slots needed in line (3-4), if not jump to line (5) . Line (3-4) compute WR_i and WT_i are sufficient slots to receive and transmit for query q_i by multiplying the expected quality of data and number of slots would allocate in the scheduler of real time query scheduling protocol previously. Line (5) means that energy efficiency function can not find the maximum point or may be the point exceeds the quality threshold [0.1, 1], then solution for pair (Q_i^m,Ep_i^m) is set as lowest or highest possible quality. In line (6-8), EAQD checks if expected QoD of query q_i becomes higher than QoD of set of its higher priority queries H(q_i) then query q_i will be started/resumed by moving it from release queue to run queue. If not, in line (9-11)query q_i is removed from run queue and is added to release queue. That means q_i is still preempted by its higher priority queries. One of benefit from EAQD algorithm is bandwidth resource reservation when using the needed transmitting/receiving slots should be allocated for each query to avoid the redundant slots. This improvement proposes in line (12). These requested slots are sent back to the scheduler in order to schedule efficiently number of slots for transmission/reception for each result/query. This improvement reduces the expensive communication cost.

V. SIMULATION RESULTS

We evaluated energy consumption based on EAQD method in Matlab. We are interested in high data rate applications such as health-care monitoring and preventive maintenance, so we choose PQS protocol [15] to run the prototype of EAQD. We simulate the performance of EADQ by using the expected QoD for the each processed query in runtime. This simulation is for a part of EAQD scheme at this time and take the experiences for whole performance of EAQD protocol on real sensor motes as future work.

Our settings are 100 sensor nodes (MICA mote)randomly are placed in $500 \times 500m^2$ within 25m communication range of each sensor node. The size of a packet is 2040 bytes, of which is 20 bytes are used for packet headers. In our simulation, the network bandwidth is 2Mbps, thus the slot size is 8.16ms. The query priorities are determined based on their deadlines: the tighter the deadline, the higher priority. The workload is generated by running three queries $q_1 : q_2 : q_3$ have ratios of their rate is 5:2:1, in there q_1, q_2, q_3 have highest priority, middle priority, and low priority, respectively. The tested queries were aggregation queries. In this experience, we consider q_1 is the base rate and we change the workload by changing the base rate.

Here our work investigates energy consumption under heavy workload. Thus, the quality of data function has the coefficients as follows: $\alpha = -167.77$, $\beta = -144.44$. EAQD allows users to specify the quality of data function for each query,then these expected QoD values can make the energy consumption to be minimum one when varying query rates or workload in a arrange total query rate from 1Hz to 5Hz.

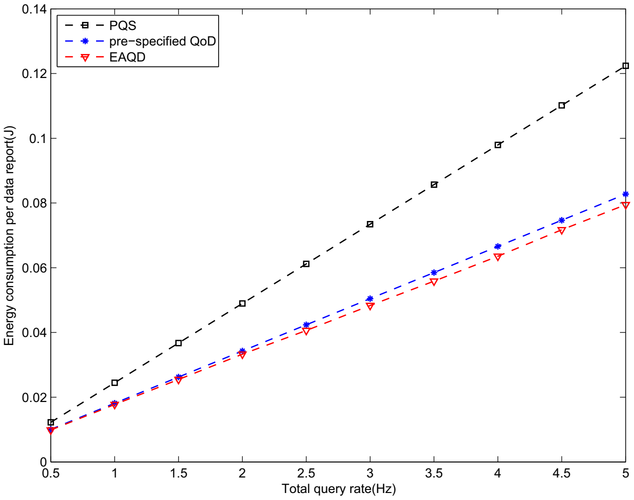

Fig. 4. Energy consumption per data report for three queries are executed and their base rate is varied

The figure 4 shows the energy consumption per data report. We observe that energy consumption in EAQD method improves better than PQS (a query scheduling protocol without any query quality optimization) and better than pre-specified QoD value since total query rate increases from 1Hz to 5Hz. In contrast to PQS, EAQD provides significant improvement of energy consumption because of PQS does not consider the expected QOD adjustment for each query into energy efficiency. For instance, when the total query rate is 2Hz, PQS consumes energy approximately 1.5 times greater than EAQD does, when the total query rate comes up to 5Hz, energy consumption in PQS increases faster than one in EAQD. The pre-specified QoD values for all queries refers to its revenue is proportional to the cost of query and it is invariable for each query during runtime. This figure also shows that if we keep pre-specified quality of data for all queries when running in PQS protocol, it still consumes energy more than EAQD. As our simulation assumption, the data fidelity is 100%, energy consumption per data report in PQS and default QOD will be constant.

We continue to compare EAQD against the query processing method without any query optimization. In figure 5, our simulation last 200s. Network randomly generates the workload with varying total query rate in range [0.5, 5]Hz. Fig. 5 shows EAQD can save energy 5 times greater on processing same workload.

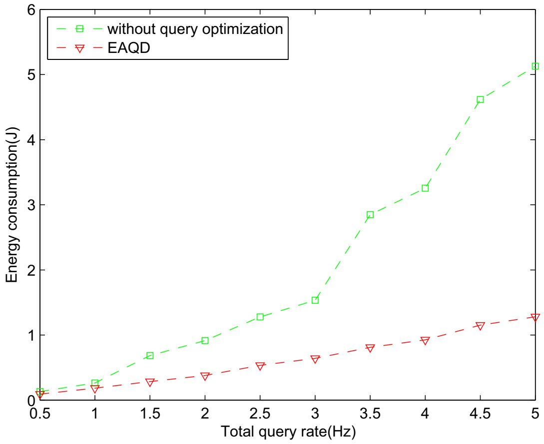

Fig. 5. Energy consumption for three queries and their base rate is varied

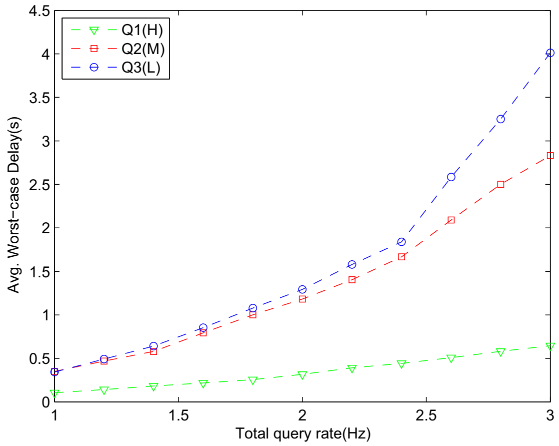

Fig. 6. Average worst- case delay for three queries are executed and their base rate is varied

Figure 6 shows the measured average worst-case delay when the base rate is increased from 1Hz to 3Hz. Our EAQD method provides prioritization between three queries Q1(H), Q2(M), Q3(L) with high , medium and low priorities, respectively. The results are the average of 10 runs with data fidelity is 90% confidence. Average worst-delay is close to 0 for rate 1Hz-1.5Hz due to EAQD can achieve near 100% fidelity in low rate. For higher rates, quality request can be too high and simultaneously the workload increases faster due to the resources may not enough to satisfy the user requests and query processors may fail since high resources limitation is an important problem in WSNs. This causes the fidelity to drop and therefore the average worst-case delay increases. For instance, when total query rate is 2.6Hz, EAQD provides an average worst-case delay of 2.59s for Q(L) and only 0.5139s for Q(H). For Q(H), the average worst-case delay does not change much that is around 0.5s because of its highest priority. We evaluated the performance of EAQD that is acceptable in real time scheduling schemes.

VI. CONCLUSION

In this paper, we presented the CPC architecture which can support simultaneously a lot of applications with effective energy consumption and satisfied quality of service requirements such as health care, living assistant, smart cars, transportation, etc. The CPC architecture is combination of multiple WSNs and cloud computing environments to enable convergence of multiple applications and resources. The CPC architecture processes effectively computation and resource management by virtualization technique. With the CPC architecture, our system also ensures the energy efficiency.

In addition, we presented the EAQD method framework. EAQD takes interference energy consumption into account when giving the trade off between QoD and total energy consumption. EAQD method runs on existing real time preemptive query scheduling protocols of WSNs to find the target QoD for each query thus enabling the minimum energy consumption for our system, thus enabling the energy consumption minimization and delay tolerance for our system. EAQD also solves the bandwidth waste problem in runtime with slots feedback mechanism to real time scheduling protocol. As shown in simulation results, EAQD improves energy conservation in the existing real time preemptive query scheduling protocols that it runs on. And EAQD outperforms than the method without quality query optimization.

ACKNOWLEDGMENT

This research was supported by MKE (Ministry of Knowledge Economy), Korea, under the IT/SW Create research program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2010-(C1810-1005-0001)). Professor Sungyoung Lee is the corresponding author. Finally, the authors very thank you to the anonymous reviewers for their constructive comments and suggestions which greatly improve the quality of this paper.

REFERENCES

[1] Z. Lv, F. Xia, G. Wu, L. Yao, and Z. Chen, "Icare: A mobile health monitoring system for the ederly," in *IEEE-ACM Int'l Conf. Green Computing and Communications and Int'l Conf. Cyber, Physical and Social Computing*, Los Alamitos, CA, USA, 2010, pp. 699–705.

[2] Y. M. Huang, M. Y. Hsieh, H. C. Chao, S. H. Hung, and J. H. Park, "Pervasive, secure access to a hierarchical sensor-based healthcare monitoring architecture in wireless heterogeneous networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 7, May 2009.

[3] M. Y. Nam, A. Sabbagh, J. E. Kim, M. K. Yoon, C. G. Lee, and E. Y. Ha, "A real-time ubiquitous system for assisted living: Combined scheduling of sensing and communication for ream-time tracking," *IEEE Transactions on Computers*, vol. 57, no. 6, pp. 795–808, 2008.

[4] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke, "Data collection, storage, and retrieval with an underwater sensor network," in *Proceedings of the 3rd international conference on Embedded networked sensor systems*, New York, NY, USA, 2005, pp. 154–165.

[5] A. Thiagarajan, L. Ravindranath, K. Lacurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "Vtrack: accurate, energy aware road traffic delay estimation using mobile phones."

[6] A. Thiagarajan, J. Biagioni, T. Gerlich, and J. Eriksson, "Cooperative transit tracking using smart-phones," in *Proc. 8th ACM Conf. Embedded Networked Sensor Systems*, Zurich, Switzerland, 2010, pp. 85–98.

[7] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: using a mobile sensor network for road surface monitoring," in *Proc. 6th Int'l Conf. Mobile Systems, Applications, and Services*, Breckenridge, CO, USA, 2008, pp. 29–39.

[8] F. Chen, T. Talanis, R. German, and F. Dressler, "Real-time enabled ieee 802.15.4 sensor networks in industrial automation," in *Proc. IEEE Int'l Symposium on Industrial Embedded Systems, 2009*, Lausanne, 2009, pp. 136–139.

[9] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. Campbell, "Sensing meets mobile social networks: the design,implementation and eveluation of the cenceme application," in *Proc. 6th ACM Conf. Embedded Networked Sensor Systems*, New York, NY, USA, 2008, pp. 337–50.

[10] F. J. Wu, Y. F. Kao, and Y. C. Tseng, "From wireless sensor networks towards cyber physical systems," *Pervasive and Mobile Computing*, pp. 1–13, April 2011.

[11] F. Xia, A. Vinel, R. Gao, L. Wang, and T. Qiu, "Evaluating ieee 802.15.4 for cyber-physical systems," *EURASIP J. Wireless Commun. and Networking*, vol. 2011, pp. 1–15, 2011.

[12] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tinydb: an acquisition query processing system for sensor networks," *ACM Trans. Database Syst.*, vol. 30, no. 1, pp. 122–173, March 2005.

[13] J. Gehrke and S. Madden, "Query processing in sensor networks," in *IEEE Pervasive Computing*, Los Alamitos, CA, USA, 2004, pp. 46–55.

[14] Y. Yao and J. Gehrke, "Query processing for sensor networks," in *Proc. 2003 CIDR Conf.*, 2003.

[15] O. Chipara, C. Lu, and G. C. Roman, "Real-time query scheduling for wireless sensor networks," in *Proc. 28th IEEE Int'l Real-Time Systems Symposium, RTSS'07*, Tucson, AZ, 2007, pp. 389–399.

[16] Q. Ren and Q. Liang, "Energy and quality aware query processing in wireless sensor database systems," *Inf. Sci.*, vol. 177, no. 10, pp. 2188–2205, May 2007.

[17] A. Sharaf, J. Beaver, A. Labrinidis, and K. Chrysanthis, "Balancing energy efficiency and quality of aggregate data in sensor networks," *The VLDB Journal*, vol. 13, no. 4, pp. 384–403, December 2004.

[18] H. Qu and A. Labrinidis, "Preference-aware query and update scheduling in web-databases," in *In Proc. IEEE 23rd Int'l Conf. Data Engineering 2007 (ICDE)*, Istanbul, 2007, pp. 356–365.

[19] H. Wu, Q. Luo, J. Li, and A. Labrinidis, "Quality aware query scheduling in wireless sensor networks."

[20] S. Bhardwaj, L. Jain, and S. Jain, "Cloud computing: a study of infrastructure as a service (iaas)," *International Journal of Engineering and Information Technology*, vol. 2, no. 1, pp. 60–63, 2010.

[21] L. Wang, G. von Laszewski, M. Kunze, and J. Tao, "Schedule distributed virtual machines in a service oriented environment," in *Int'l Conf. Advanced Information Networking and Applications*, Los Alamitos, CA, USA, 2010, pp. 230–236.

[22] H. Jinhua, G. Jianhua, S. Guofei, and Z. Tianhai, "A scheduling strategy on load balancing of virtual machine resources in cloud computing

environment," in *Proc. 3rd Int'l Symposium on Parallel Architectures, Algorithms and Programming (PAAP) 2010*, Dalian, 2010, pp. 89–96.

[23] S. Chaisiri, B. Lee, and D. Niyato, "Optimization of resource provisioning cost in cloud computing," *IEEE Transactions on Services Computing*, vol. PP, no. 99, pp. 1–32, 2011.

[24] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tag: a tiny aggregation service for ad hoc sensor networks," *SIGOPS Oper. Syst. Rev.*, vol. 36, pp. 131–146, Dec 2002.

[25] N. D. Lane, E. Miluzzo, L. Hong, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, Sept 2010.

[26] L. Peng and K. S. Candan, "Data quality guided load shedding for expensive in-network data processing," in *In Proc. IEEE 23rd Int'l Conf. Data Engineering 2007 (ICDE)*, Istanbul, 2007, pp. 1325–1328.

[27] O. Chipara, C. Lu, and J. Stankovic, "Dynamic confilct-free query scheduling for wireless sensor networks," in *Proc. 2006 IEEE Int'l Conf. Network Protocols*.

[28] G. Zhou, T. He, J. A. Stankovic, and T. F. Abdelzaher, "Rid: radio interfence detection in wireless sensor networks," in *Proc. 24th Annual Joint Conf. IEEE Computer and Communications Societies, INFOCOM'05*, 2005, pp. 891–901.

[29] B. Hohlt, L. Doherty, and E. Brewer, "Flexible power scheduling for sensor networks," in *Proc. 3rd Int'l symposium on Information processing in sensor networks*, Berkeley, California, USA, 2004, pp. 205–214.

[30] G. Lu, N. Sadagopan, B. Krishnamachari, and A. Goel, "Delay efficient sleep scheduling in wireless sensor networks," in *Proc. 24th Annual Joint Conf. IEEE Computer and Communications Societies, INFOCOM'05*, 2005, pp. 2470–2481.

[31] H. Wu, Q. Luo, and W. Xue, "Distributed cross-layer scheduling for in-network sensor query processing," in *In PERCOM'06 Proceedings of 4th Annual IEEE on Pervasive Computing and Communications*, 2006.