Energy-Efficient Reporting Mechanisms for Multi-Type Real-time Monitoring in Machine-to-Machine Communications Networks

Huai-Lei Fu*, Hou-Chun Chen*, Phone Lin*, and Yuguang Fang†

*Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan, R.O.C.
Email: {vicfu, abner}@pcs.csie.ntu.edu.tw, plin@csie.ntu.edu.tw

†Department of Electrical and Computer Engineering, University of Florida, 435 Engineering Building,
P.O. Box 116130, Gainesville, FL 32611, U.S.A.
Email: fang@ece.ufl.edu

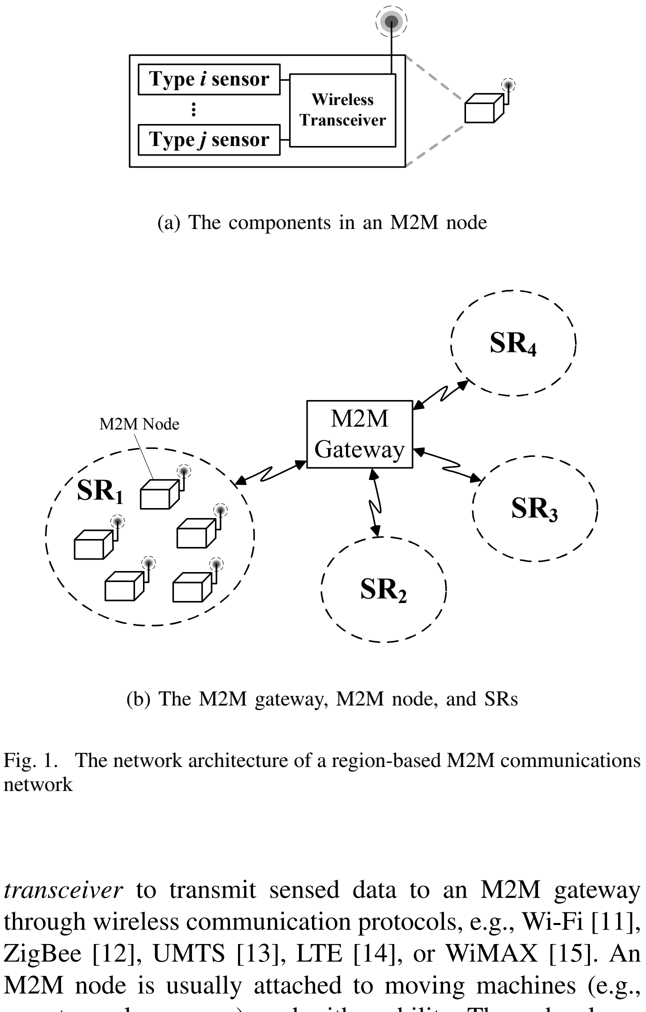
***Abstract*—In Machine-to-Machine (M2M) communications, machines are wirelessly connected to accomplish collaborative tasks without human intervention, and provide ubiquitous solutions for real-time monitoring. The real-time monitoring application is one of the killer applications for M2M communications, where M2M nodes transmit sensed data to an M2M gateway, and then the M2M gateway can have real-time monitoring for each sensing region. In real-time monitoring application, the energy consumption for the M2M nodes to send sensed data to the M2M gateway is an important factor that significantly affects the performance of the system. In this paper, we first consider the energy consumption as well as the validity of sensed data to design either centralized or distributed energy-efficient reporting mechanisms. We then analyze the complexity of the reporting mechanisms. Simulation experiments are conducted to investigate the performance of the proposed mechanisms, and show that the distributed mechanism outperforms the centralized mechanism when the M2M nodes are mobile.**

***Index Terms*—Energy efficiency, Machine-to-machine communications, Reporting mechanism, Sensed data validity**

I. Introduction

Machine-to-Machine (M2M) communications is being standardized in several working groups, e.g., ETSI [1], [2] and *machine-type communications* (MTC) in 3GPP [3], [4]. In M2M communications, machines are wirelessly connected to accomplish collaborative tasks without human intervention. A machine has "intelligence" to make decision autonomously based on collected information from other machines. Examples of M2M communications networks include vehicular ad-hoc networks (VANETs) [5], underwater sensor networks (UWSNs) [6], [7], and wireless body area networks (WBANs) [8], [9].

An M2M communications network comprises a number of *M2M nodes* and *M2M gateways* [10]. As shown in Figure 1(a), an M2M node is equipped with multiple *sensors* to collect different types of sensed data (e.g., temperature, humidity, density of toxic gas and air quality) and a *wireless transceiver* to transmit sensed data to an M2M gateway through wireless communication protocols, e.g., Wi-Fi [11], ZigBee [12], UMTS [13], LTE [14], or WiMAX [15]. An M2M node is usually attached to moving machines (e.g., a motorcycle or a car), and with mobility. Through a location determination system (e.g., global positioning system; GPS), an M2M node can obtain its location information. An M2M gateway, which is usually equipped with permanent power supply, has powerful computation and transmission capability. The main task of an M2M gateway is to have



(a) The components in an M2M node

(b) The M2M gateway, M2M node, and SRs

Fig. 1. The network architecture of a region-based M2M communications network

computation on the collected sensed data from M2M nodes.

As pointed out in [10], the most promising M2M application is *real-time monitoring*, including e-healthcare, smart grids, environmental monitoring, industrial automation, and so on. Take urban environment in Taiwan for example. Air pollution is mainly caused by traffic jam due to that crowded vehicles in traffic jam are with low mobility, and their emission results in high concentration of air pollutant. With real-time monitoring on air quality in urban environment, we can keep pedestrians informed about the polluted areas, so that pedestrians can choose more healthful routes.

In a real-time monitoring application, the monitoring area of an M2M communications network is divided into several *sensing regions* (SRs). In an SR, there may be one or more types of sensed data to be collected, and the values for a specific type from different M2M nodes are more or less the same during a certain time period. Figure 1(b) illustrates an example of an M2M communications network partitioned into four SRs, namely SR_1, SR_2, SR_3, and SR_4. In this example, there are one M2M gateway and five M2M nodes in SR_1.

Each M2M node may have different sensing capabilities since each M2M node may be equipped with different types of sensors to collect different types of sensed data. There are two operating modes for an M2M node: active mode or sleep mode. The period when the M2M node is in the active (sleep) mode is called active (sleeping) period. In the active mode, the M2M node collects the sensed data from its equipped sensors, and then transmits the collected sensed data with time and location information to the M2M gateway. After transmission, the M2M node switches to the sleep mode and stays in the sleep mode for a period of time to save its power consumption.

For each type of sensed data, we define *monitoring period* (MP) as its time constraint. We deem a sensed data to be a *valid* sensed data [16] if the sensed data is collected by an M2M node and received by the M2M gateway in the same MP. The validity for a sensed data will be formally defined in Section II-A. For each type, the M2M gateway must receive at least one valid sensed data in every MP to monitor the sensed data in real-time. However, redundant transmissions occur if more than one valid sensed data is received by the M2M gateway during an MP, which result in more power consumption for M2M nodes. To conserve energy, the M2M node can selectively transmit the sensed data (i.e., it may not transmit all collected sensed data each time) or stay in the sleep mode longer, but the possibility that the M2M gateway cannot receive the valid sensed data every MP may increase. The tradeoff is called the *energy-validity* tradeoff that is also pointed out in our previous work [16]. Careful scheduling of sensed data transmission can prevent redundant transmissions, save power for M2M nodes, and prolong the lifetime of the M2M communications network. Meanwhile, the M2M gateway can maintain the freshness of the sensed data to achieve real-time monitoring.

To address this issue, many previous studies [17–22] propose energy-efficient reporting mechanisms for M2M communications networks. Most of these works focus on stationary nodes with mostly real-time monitoring applications. These works can be divided into two categories: *centralized* and *distributed*. The centralized reporting mechanisms [17], [18], [20] schedule the transmissions for M2M nodes in a centralized fashion. A central controller, e.g., the M2M gateway, coordinates the transmission schedule for M2M nodes. The M2M nodes must keep on updating the schedule sent from the central controller. On the contrary, the distributed reporting mechanisms [23], [24] allow M2M nodes to operate independently without being scheduled by a central controller. An M2M node can decide the transmission schedule on its own. However, the designs of the centralized or distributed reporting mechanisms in the previous works rarely take mobility of M2M nodes and validity of sensed data into consideration. Therefore, these mechanisms do not work well when the M2M nodes are mobile.

In this paper, we propose intelligent reporting mechanisms to deal with the energy-validity tradeoff for "multi-type" real-time monitoring applications for mobile M2M nodes. First, we propose a centralized reporting mechanism, namely *Energy-efficient Centralized Reporting* (ECR), in which the M2M gateway is responsible for scheduling sensed data transmissions for M2M nodes. In ECR, we transform the energy-validity tradeoff to an *energy minimization problem*, and use the integer linear programming (ILP) to formulate the energy minimization problem. Then, we show NP-hardness for the energy minimization problem, and propose a low-complexity greedy algorithm to approximate the solution of the ILP formulation. Second, we propose a distributed reporting mechanism, namely *Energy-efficient Distributed Reporting* (EDR), in which the M2M node decides its own transmission schedule. To deal with the energy-validity tradeoff in a distributed fashion, we design the *threshold-based dynamic sleeping adjustment* (TDSA) algorithm for the M2M node to adjust its sleeping period. Third, we develop the simulation experiments to investigate how mobility affects the performances of ECR and EDR, which are not studied in the previous studies and can be one of our main contributions in this work. Our study shows that EDR outperforms ECR when M2M nodes are moving dynamically.

The rest of this paper is organized as follows. Section II describes the details of the ECR and EDR mechanisms. Section III proposes the energy minimization problem, its ILP formulation, and the NP-hardness proof. In Section IV, the performances of ECR and EDR are investigated through simulation experiments. Finally, Section V concludes this work.

II. Energy-Efficient Reporting Mechanisms

In this section, we first define the validity of sensed data for real-time monitoring applications. Then, we propose

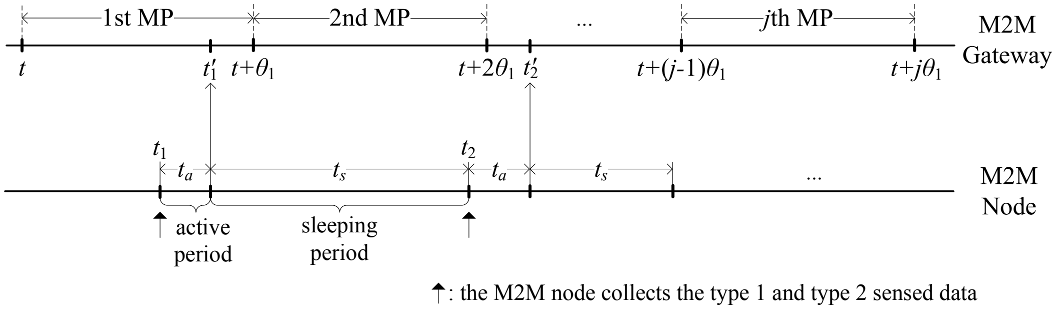


Fig. 2. An example that the M2M gateway monitors the type 1 sensed data and the M2M node transmits the type 1 and type 2 sensed data

two energy-efficient reporting mechanisms ECR and EDR.

A. Validity of Sensed Data

Suppose that there are N types of sensed data, say, type 1, type 2, ..., type N, to be collected. We denote the sensing set by $\mathbb{S} = \{1, 2, 3, ..., N\}$. The type i sensed data is sensed by the type i sensor. Consider an M2M node, n_k, in the SR. Let the set $\mathbb{S}_k \subseteq \mathbb{S}$ denote the types of sensed data that can be collected by n_k, and $\mathbb{S}_k$ is called the sensing set of n_k. As mentioned previously, n_k in the active mode collects the type i sensed data from the type i sensor for all $i \in \mathbb{S}_k$. Compared with the active period or the sleeping period, the time period for n_k to collect the sensed data is small enough to be negligible in this paper. Let t_a (t_s) denote the period when n_k is in the active (sleeping) period. Figure 2 illustrates an example that an M2M node is equipped with the type 1 and type 2 sensors. At time t_1, the M2M node transmits the type 1 and type 2 sensed data. At time $t_1 + t_a$, the M2M node switches to sleep mode, and stays in the sleep mode for a t_s period. After the t_s period, the M2M node switches to active mode at time $t_2 = t_1 + t_a + t_s$, and starts to transmit the collected sensed data.

Let θ_i be the length of an MP for type i sensed data. Suppose that the real-time monitoring application starts at time t, and the jth MP for type i sensed data starts at $t + (j-1)\theta_i$ and ends at time $t + j\theta_i$. Assume that a type i sensed data is collected by an M2M node at time t_x in the jth MP (i.e., $t + (j-1)\theta_i \le t_x \le t + j\theta_i$), and received by the M2M gateway at t_x' (where $t_x < t_x'$). Then following the definition in [16], the type i sensed data is *valid* if

$$t + (j-1)\theta_i \le t_x < t_x' \le t + j\theta_i. \tag{1}$$

Otherwise (i.e., $t_x' > t + j\theta_i$), the type i sensed data is *invalid*. As shown in Figure 2, the first transmission for the type 1 and type 2 sensed data starts at time t_1 (transmitted by the M2M node) and ends at time t_1' (received by the M2M gateway). Since $t \le t_1 < t_1' \le t + \theta_1$, the type 1 sensed data of the first transmission is valid. On the other hand, the second transmission for the type 1 and type 2 sensed data starts at time t_2 and ends at time t_2'. Since $t + \theta_1 < t_2 \le t + 2\theta_1$ and $t_2' > t + 2\theta_1$, the type 1 sensed data is invalid.

B. Centralized Reporting Mechanism

A reporting mechanism consists of two parts: M2M gateway part and M2M node part. In the ECR mechanism, the M2M gateway calculates the transmission schedule for M2M nodes, and the M2M nodes transmit the sensed data to the M2M gateway according to the transmission schedule received from the M2M gateway. The details of the two parts of ECR are given as follows.

M2M Gateway Part: We consider an SR in which there are N types of sensed data to be collected, and the M2M gateway monitors N types of sensed data for each SR. The M2M gateway maintains the following information: the set $\mathbb{D} = \{D(1), D(2), ..., D(N)\}$ for the N types of sensed data, where $D(i)$ is the latest valid type i sensed data. The MP for the type i sensed data is controlled by the T_i timer, and the value of T_i is set to θ_i. The M2M gateway starts the monitoring of the type i sensed data by triggering the T_i timer. The expiration of T_i denotes the end of an MP for type i sensed data. When the M2M gateway receives a type i sensed data, it checks the validity for the type i sensed data by (1). If the type i sensed data is valid, the $D(i)$ value is updated. Otherwise (i.e., the type i sensed data is invalid), the type i sensed data is ignored.

Without loss of generality, we assume $\theta_1 \le \theta_2 \le, ..., \le \theta_N$, and

$$\theta_i = a_i\theta_1 \qquad \forall i \in \{1, 2, ..., N\}$$

where a_i are integers and $a_1 = 1$. Let θ_1 denote the length of a transmission unit. Define a cycle with length $L = \text{lcm}(a_1, a_2, ..., a_N)\theta_1$, where $\text{lcm}(a_1, a_2, ..., a_N)$ is the least common multiple of $a_1, a_2, ..., a_N$. The M2M gateway maintains the binary variable $x_{i,j,k}$, where $x_{i,j,k} = 1$ denotes that n_k is assigned to transmit the type i sensed data at the jth transmission unit within a cycle, and $\{x_{i,j,k}\}$ is a binary three-dimensional array to denote the transmission schedule of a cycle for M2M nodes. The M2M gateway schedules the sensed data transmission for M2M nodes every cycle. Specifically, the M2M gateway executes a greedy scheduling algorithm (to be elaborated in Section III) at the beginning of

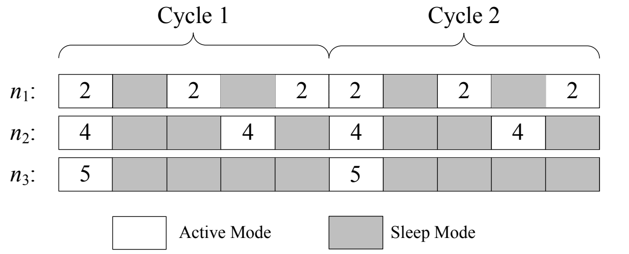


Fig. 3. An example for a transmission schedule for the M2M nodes n_1, n_2, n_3

each cycle to determine the schedule $\{x_{i,j,k}\}$ for the cycle. Then, the schedule is broadcasted by the M2M gateway to the M2M nodes.

M2M Node Part: Consider an M2M node, n_k, in the SR. As mentioned previously, n_k has two operating modes: active mode and sleep mode. After n_k is powered on, n_k stays in the active mode and waits for receiving the transmission schedule broadcasted by the M2M gateway. n_k will not transmit any sensed data until receiving the first schedule, and the time when n_k receives the first schedule is considered as the beginning of the first cycle for n_k. When n_k receives a schedule $\{x_{i,j,k}\}$, n_k checks its transmission assignment by calculating $\mathbb{S}_{k,j} = \{i|\forall x_{i,j,k} = 1\}$, where $\mathbb{S}_{k,j}$ is the set containing the types of sensed data to be transmitted at the jth transmission unit within a cycle. Figure 3 illustrates an example of a transmission schedule $\{x_{i,j,k}\}$ for three M2M nodes, n_1, n_2, n_3. In $\{x_{i,j,k}\}$, $x_{2,1,1}$, $x_{2,3,1}$, $x_{2,5,1}$, $x_{4,1,2}$, $x_{4,4,2}$, $x_{5,1,3}$ are 1, and the others in $\{x_{i,j,k}\}$ are 0. Based on the schedule in Figure 3, n_1 will transmit the type 2 sensed data in the first, the third, and the fifth transmission unit within a cycle. n_2 will transmit the type 4 sensed data in the first and the fourth transmission unit within a cycle. n_3 will transmit the type 5 sensed data in the first transmission unit within a cycle. Otherwise, these three M2M nodes stay in the sleep mode.

C. Distributed Reporting Mechanism

In the ECR mechanism, the M2M gateway needs more information about the sensing sets of M2M nodes, which are collected via communications. This may consume more energy for M2M nodes. Therefore, we propose the EDR mechanism in which the transmission schedule is no longer centrally controlled by the M2M gateway. Instead, the M2M node decides the transmission schedule on its own. The details of the two parts of EDR are given as follows.

M2M Gateway Part: Similar to ECR, in EDR, the M2M gateway also maintains the set $\mathbb{D} = \{D(1), D(2), ..., D(N)\}$ for each SR, and the timers T_1, T_2, ..., T_N to denote the MPs for each type, respectively. In the M2M gateway, we maintain the CR_i counter to count the number of valid type i sensed data received by the M2M gateway during a type i MP. At the beginning of a type i MP, CR_i is reset to zero. At the end of the xth type i MP, the variable $I_i(x)$ is maintained to indicate that the xth type i MP is a real-time MP (during which the M2M gateway receives at least one valid type i sensed data) or an outdated MP (during which the M2M gateway receives no valid type i sensed data):

$$I_i(x) = \begin{cases} 1, & \text{if } CR_i = 0; \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Let $\Gamma_i(0) = 0$ for $1 \leq i \leq N$. For $x > 0$, at the end of every type i MP, the M2M gateway computes the ratio $\Gamma_i(x)$:

$$\Gamma_i(x) = \begin{cases} \Gamma_i(x-1) + I_i(x), & \text{if } x \bmod K \neq 0; \\ \dfrac{\Gamma_i(x-1) + I_i(x)}{K}, & \text{otherwise.} \end{cases} \tag{3}$$

At the end of every K type N MPs, the latest Γ_1, Γ_2, ..., Γ_N are broadcasted by the M2M gateway through a notification message to notify the M2M nodes to dynamically adjust their sleeping periods. After the notification message is broadcasted, $\Gamma_i(x)$ is reset to zero for $1 \leq i \leq N$.

M2M Node Part: An M2M node maintains the sleep timer T_{sleep} to control its sleeping period, and the transmission timers $T_{tx,1}$, $T_{tx,2}$, ..., $T_{tx,N}$ to control which types of sensed data to be transmitted. Initially, the M2M node n_k stays in the active mode. When the $T_{tx,i}$ timer expires, the type i sensed data will be transmitted. Note that if multiple transmission timers expire at the same time, the M2M node will transmit multiple types of sensed data in one transmission. After n_k transmits sensed data, it switches to sleep mode and triggers the T_{sleep} timer. When the T_{sleep} timer expires, the n_k switches back to the active mode. In the active mode, n_k also listens to the notification message for the dynamic adjustment of its sleeping period. When n_k receives a notification message from the M2M gateway, n_k executes the TDSA algorithm (see Algorithm 1) to adjust T_{sleep}, and $T_{tx,1}$, $T_{tx,2}$, ..., $T_{tx,N}$. The details of the TDSA algorithm are given as follows.

Let $V(j)$ and $V_i(j)$ refer to the jth configuration for the T_{sleep} timer and the $T_{tx,i}$ timer, respectively. When the SR is well monitored, i.e., every MP is a real-time MP, TDSA attempts to prolong the M2M node's sleeping period to conserve energy consumption. On the other hand, i.e., the SR is not well monitored, TDSA attempts to reduce the M2M node's sleeping period in order to make the M2M gateway receive sensed data. Initially, $V(1)$, $V_1(1)$, $V_2(1)$, ..., $V_N(1)$ are all set to θ_1. Based on the ratio Γ_1, Γ_2, ..., Γ_N in the received notification message, $V(x+1)$, $V_1(x+1)$, $V_2(x+1)$, ..., $V_N(x+1)$ are adjusted according to the following algorithm.

Algorithm 1: Dynamic Sleeping Adjustment

```
   Input: $\Gamma_1$, $\Gamma_2$, ..., $\Gamma_N$, and $V_1(x)$, $V_2(x)$, ..., $V_N(x)$
   Output: $V(x+1)$, $V_1(x+1)$, $V_2(x+1)$, ...,
           $V_N(x+1)$
1  foreach $i : i \in \mathbb{S}_k$ do
2      if $\Gamma_i = 0$ then
3          $V'_i(x+1) \leftarrow V_i(x) + A\theta_i$;
4      else if $\Gamma_i < \rho$ then
5          $V'_i(x+1) \leftarrow V_i(x)$;
6      else
7          $V'_i(x+1) \leftarrow \max\{V_i(x) - A\theta_i, 0\}$;
8      end
9  end
10 $V(x+1) \leftarrow \min\{V'_i(x+1) | i \in \mathbb{S}_k\}$;
11 foreach $i : i \in \mathbb{S}_k$ do
12     $V_i(x+1) \leftarrow$ Round($\frac{V'_i(x+1)}{V(x+1)}$);
13 end
```

The notation ρ ($0 \leq \rho \leq 1$) is an adjustment threshold for the M2M node to determine when to adjust each transmission timer (see Lines 1-9), and the notation A is an adjustable variable. The minimum transmission timer is chosen as the sleep timer (see Line 10) to satisfy the time constraints for all types of sensed data that can be collected by the M2M node. The function `Round()` ensures all the transmission timers are integer multiples of the sleep timer so that the expiration time of transmission timers will match the expiration time of the sleep timer (see Line 12). Therefore, the integer numbers of the transmission timers stand for the number of times the sleep timer expires. For example, we assume that T_{sleep} is set to θ_1, $T_{tx,1}$ (i.e., the smallest transmission timer) is set to 1, and $T_{tx,2}$ is set to 3. Every time after it transmits sensed data to the M2M gateway, the M2M node switches to sleep mode and triggers T_{sleep} to sleep for θ_1. Then, $T_{tx,2}$ expires every three times T_{sleep} expires. That is, the type 2 sensed data is transmitted once every three times of type 1 sensed data transmissions since $T_{tx,2}$ is three times longer than T_{sleep}.

III. Energy Minimization for Centralized Scheduling

In this section, the determination of a transmission schedule for each M2M node during a cycle is formulated as an optimization problem, which is called *energy minimization* problem. The energy minimization problem arranges the transmission time and types of sensed data for each M2M node so that the energy consumption can be minimized and the time constraints for all types of sensed data are satisfied. In the following, we first describe our assumptions and reintroduce the notations. Section III-B provides the ILP formulation. Section III-C proves that the energy minimization problem is NP-hard. Section III-D proposes a greedy scheduling algorithm to approximate the solution of the ILP.

A. Assumptions and Notations

For the sake of simplicity, we assume the same sizes for sensed data regardless of their types, and the fixed transmission rates for M2M nodes. Thus, the minimization of the energy consumption for M2M nodes is transformed to the minimization of the amount of transmitted data, and the amount of transmitted data can be minimized through minimizing the number of transmission units assigned in a transmission schedule.

For the convenience of discussion, we reintroduce the notations as follows:

- N: the number of types of sensed data to be collected in the SR.
- L: the number of transmission units in a cycle.
- M: the number of M2M nodes in the SR.
- i: the index of types, where $1 \leq i \leq N$.
- j: the index of transmission units within a cycle, where $1 \leq j \leq L$.
- k: the index of M2M nodes, where $1 \leq k \leq M$.
- $x_{i,j,k}$: the binary variable defined by

$$x_{i,j,k} = \begin{cases} 1, & \text{if } n_k \text{ transmits type } i \text{ sensed data at} \\ & \text{the } j\text{th transmission unit within a cycle;} \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

B. ILP Formulation

We use the ILP to model the energy minimization problem. The ILP formulation can find the optimal solution for the energy minimization problem, i.e., the optimal transmission schedule for each M2M node during a cycle. Our ILP formulation is described as follows.

The objective function of the ILP formulation is given as

$$\min \sum_{k=1}^{M} \sum_{j=1}^{L} \sum_{i=1}^{N} x_{i,j,k} \tag{5}$$

subject to:

$$\sum_{k:1 \in \mathbb{S}_k} x_{1,j,k} \geq 1, \forall j \in \{1, 2, ..., L\} \tag{6.1}$$

$$\sum_{k:2 \in \mathbb{S}_k} \sum_{l=0}^{a_2-1} x_{2,j+l,k} \geq 1,$$
$$\forall j \in \{1, 1+a_2, 1+2a_2, \cdots, L-a_2+1\} \tag{6.2}$$

$$\sum_{k:3 \in \mathbb{S}_k} \sum_{l=0}^{a_3-1} x_{3,j+l,k} \geq 1,$$
$$\forall j \in \{1, 1+a_3, 1+2a_3, \cdots, L-a_3+1\} \tag{6.3}$$

$$\vdots$$

$$\sum_{k:N \in \mathbb{S}_k} \sum_{l=0}^{a_N-1} x_{N,j+l,k} \geq 1,$$

$$\forall j \in \{1, 1+a_N, 1+2a_N, \cdots, L-a_N+1\} \qquad (6.\text{N})$$

$$x_{i,j,k} \in \{0,1\} \qquad (7)$$

In the objective function (5), $\sum_{j=1}^{L}\sum_{i=1}^{N} x_{i,j,k}$ means the number of transmission units assigned to the M2M node n_k in a cycle, and $\sum_{k=1}^{M}\sum_{j=1}^{L}\sum_{i=1}^{N} x_{i,j,k}$ is the total transmission units assigned to all M2M nodes during a cycle. The objective function (5), which minimizes the total transmission units, can minimize the total energy consumption due to that the transmission rate and the sensed data size are all fixed. The inequality constraints (6.1), (6.2), (6.3), ..., (6.N) ensure that at least one valid sensed data is transmitted to the M2M gateway for each type of MP. For example, the constraint (6.1) ensures that for each MP for type 1 whose length is one transmission unit, at least one valid type 1 sensed data is received by the M2M gateway from those M2M nodes which have the type 1 sensors. Furthermore, the ILP formulation has complexity $O(2^{M+N+L})$ which is hard to computed in a short period of time.

C. Problem Complexity

Theorem 1. *The energy minimization problem is NP-hard.*

Proof: We prove that the energy minimization problem is NP-hard by reduction from the *set-covering* problem, which is a well-known NP-hard problem [25]. The set-covering problem is described as follows. Let $\mathbb{F}$ denote a set of finite elements and $\mathbb{H} = \{\mathbb{H}_1, \mathbb{H}_2, ..., \mathbb{H}_n\}$, where $\mathbb{H}_i$ $(1 \leq i \leq n)$ is a set containing finite elements in $\mathbb{F}$. The union of $\mathbb{H}_1, \mathbb{H}_2, ..., \mathbb{H}_n$ forms $\mathbb{F}$. From [25], the set-covering problem is to find a subset $\mathbb{H}' \subseteq \mathbb{H}$ covering all elements in $\mathbb{F}$ with the minimum set size (i.e., $|\mathbb{H}'|$ is minimum).

Consider an energy minimization problem instance $(\mathbb{M}, \mathbb{S})$ consisting of a finite set of M2M nodes $\mathbb{M} = \{n_1, n_2, ..., n_M\}$ and a sensing set $\mathbb{S} = \{1, 2, 3, ..., N\}$. We replace every $n_k \in \mathbb{M}$ with its sensing set $\mathbb{S}_k$ and let the set $\tilde{\mathbb{S}} = \{\mathbb{S}_1, \mathbb{S}_2, ..., \mathbb{S}_M\}$, where a sensing set $\mathbb{S}_k$ $(1 \leq k \leq M)$ contains the the types of sensing data that the M2M node n_k can collect. Thus, $\mathbb{S} = \bigcup_{k=1}^{M} \mathbb{S}_k$. We then consider $(\mathbb{S}, \tilde{\mathbb{S}})$ as an instance of the set-covering problem. Therefore, the energy minimization problem is also NP-hard. ■

D. Greedy Scheduling Algorithm

To approximate the solution to the ILP formulation, we here propose a greedy scheduling algorithm with polynomial time complexity as shown in Algorithm 2. The greedy scheduling algorithm is executed by the M2M gateway at the beginning of every cycle to generate the transmission schedule for M2M nodes. The M2M gateway transmits this schedule immediately, so all M2M nodes must wake up at the beginning of each cycle. In a cycle, the sensed data of each single type will not be transmitted by more than two M2M nodes. In other words, for a single type, the M2M gateway assigns transmission units to a single M2M node according to the corresponding MP.

In Algorithm 2, the input is the set $\tilde{\mathbb{S}} = \{\mathbb{S}_1, \mathbb{S}_2, ..., \mathbb{S}_M\}$. The algorithm picks an M2M node n_k whose $\mathbb{S}_k$ covers the most uncovered types (see Line 7), and assigns a set $\mathbb{S}'_k \subseteq \mathbb{S}_k$ for n_k (see Line 8), where $\mathbb{S}'_k$ is the set containing the types of sensed data assigned to n_k during a cycle. Then, it adds $\mathbb{S}'_k$ into the solution set $\tilde{\mathbb{S}}'$ (see Line 10). Finally, when all types in $\mathbb{S}$ is covered by the union of the sets in $\tilde{\mathbb{S}}'$, the algorithm terminates and outputs the solution set $\tilde{\mathbb{S}}'$.

Algorithm 2: Greedy

```
Input: A set of sensing sets $\tilde{\mathbb{S}} = \{\mathbb{S}_1, \mathbb{S}_2, ..., \mathbb{S}_M\}$
Output: A subset $\tilde{\mathbb{S}}'$ of $\tilde{\mathbb{S}}$ whose elements cover the
        sensing set $\mathbb{S}$
1  $\tilde{\mathbb{S}}' \leftarrow \emptyset$;
2  $\tilde{\mathbb{S}}_c \leftarrow \emptyset$;
3  for $k = 1$ to $M$ do
4      $\mathbb{S}'_k \leftarrow \emptyset$;
5  end
6  while $\tilde{\mathbb{S}}_c \neq \mathbb{S}$ do
7      select an $\mathbb{S}_k \in \tilde{\mathbb{S}}$ to maximize $|\mathbb{S}_k \cap (\mathbb{S} - \tilde{\mathbb{S}}_c)|$;
8      $\mathbb{S}'_k \leftarrow \mathbb{S}_k \cap (\mathbb{S} - \tilde{\mathbb{S}}_c)$;
9      $\tilde{\mathbb{S}}_c \leftarrow \tilde{\mathbb{S}}_c \cup \mathbb{S}'_k$;
10     $\tilde{\mathbb{S}}' \leftarrow \tilde{\mathbb{S}}' \cup \{\mathbb{S}'_k\}$;
11 end
```

IV. Performance Evaluation

In this section, we compare the performances for the ILP and greedy approaches in terms of time cost and sizes of solution sets. Then, we investigate the effects of M2M node mobility on the ECR and EDR mechanisms through discrete-event driven simulation (that has been widely used in wireless networks studies [26], [27]).

A. Performance on ILP and Greedy Approach

We use a binary matrix with M rows and N columns to denote the problem instance mentioned in Section III-C, where M and N are the number of M2M nodes in an SR and the number of types of sensed data to be collected, respectively. We do not take the mobility of M2M nodes into consideration in this subsection and will discuss it in the next subsection. The entry row k, column i in the matrix is set to "1" if the M2M node n_k can collect the type i sensed data. Otherwise, the entry is set to "0". In other words, row k in the matrix denotes the sensing set of n_k for $1 \leq k \leq M$.

Considering each combination of the values of M and N, we randomly generate 100 independent binary matrices

M	100			
N	10	20	50	100
M_{ILP}	1.87	2.00	2.98	3.76
M_{greedy}	1.87	2.05	2.99	3.96
T_{ILP}	0.883000	1.230700	50.009700	144.87
T_{greedy}	0.000135	0.000194	0.000453	0.001

(a) 100 M2M nodes

M	200			
N	10	20	50	100
M_{ILP}	1.88	2.00	2.97	3.18
M_{greedy}	1.88	2.00	3.00	3.86
T_{ILP}	3.877200	5.681700	442.9498	3379.8
T_{greedy}	0.000159	0.000202	0.000497	0.0011

(b) 200 M2M nodes

TABLE I
COMPARISON BETWEEN MATLAB ILP SOLVER AND GREEDY ALGORITHM

(i.e., problem instances) by a discrete uniform distribution. Then we solve these problem instances by MATLAB ILP solver and the greedy algorithm on a desktop with a 3.0GHz quad-core processor, respectively. The results are shown in Table I. Let M_{ILP} and M_{greedy} be the average number of M2M nodes in the solution set by MATLAB ILP solver and the greedy algorithm, respectively. Let T_{ILP} and T_{greedy} be the average elapsed time for solving the problem instance by MATLAB ILP solver and the greedy algorithm, respectively. As shown in Table I, the relative errors increase with M and N. Furthermore, the elapsed time for solving the ILP drastically increases with N. For example, the time for MATLAB ILP solver to solve the problem instance with $M = 200$ and $N = 100$ is 3379.8 seconds, which is much larger than 0.0011 seconds, i.e., the time for the greedy algorithm to solve the problem instance.

B. Simulation Experiments

In this section, we develop simulation experiments to investigate mobility effects of M2M nodes on the proposed reporting mechanisms. We apply the arrivals and departures of M2M nodes in an SR to simulate movements of M2M nodes.

We assume that the inter-arrival time for M2M nodes is Gamma distributed with mean $1/\lambda$, and the SR residence time is Gamma distributed with mean $1/\eta$. The Gamma distribution is selected because it can be used to approximate many other distributions [28]. We use different variances of the Gamma distribution to represent the dynamic of M2M nodes' moving patterns. The variances of inter-arrival time and SR residence time are denoted by v_λ and v_η, respectively. In our simulation experiment, the smallest time unit is a transmission unit.

We generate two problem instances with 20 M2M nodes and 5 types of sensed data, which are binary matrices with 20 rows and 5 columns. In problem instance 1, each M2M node has the same capability, i.e., each M2M node can sense all the five types of sensed data. In problem instance 2, the binary matrix is generated randomly by the uniform distribution. We assume $\theta_1 = 1$, $\theta_2 = 2$, $\theta_3 = 3$, $\theta_4 = 4$, and $\theta_5 = 5$. Thus, the length of a cycle is 60 transmission units.

Two output measures, the real-time monitoring probability P_r and the power saving probability P_s, are defined as

$$P_r = \frac{\sum_{i=1}^{N} P_{r,i}}{N},$$

where

$$P_{r,i} = \Pr[\text{the M2M gateway receives at least one valid type } i \text{ sensed data in an MP}],$$

and

$$P_s = \lim_{t\to\infty} \Pr[\text{an M2M node is in sleep mode at time } t].$$

We study P_r and P_s on different input parameters (i.e., λ and η) as follows.

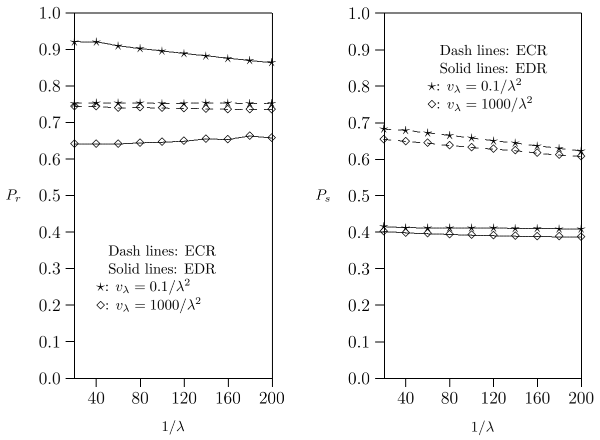
1) Effects of M2M Node Arrival Rate λ: Figure 4 and Figure 5 study the effects of λ on P_r and P_s for problem instance 1 and problem instance 2, respectively. In Figure 4(a), we can observe that P_r for EDR does not decrease too much when $1/\lambda$ increases since there are always M2M nodes to transmit sensed data. However, there are more M2M nodes stay in active mode in the EDR than that in the ECR. In Figure 4(b), it can be observed that P_s for EDR is lower than P_s for ECR since the EDR improves P_r by slightly raising the number of redundant transmissions.

P_r and P_s for ECR in Figure 4 are slightly better than those in Figure 5 since M2M nodes in problem instance 1 can sense all the types of sensed data. In both figures, v_λ just slightly affects P_r and P_s.

In Figure 4(a) and Figure 5(a), we observe that P_r for ECR stays stable, i.e., 75%, since the M2M gateway can always find some M2M nodes to send the sensed data. The reason that P_r for ECR cannot exceed 80% is the cycle length. In this experiment, the cycle length is set to 60 transmission units, the least common multiple of lengths of all MPs, which is close to the region residence time, $1/\eta$. In such situation, the assigned M2M node may leave the region before a cycle ends that leads to the M2M gateway does not receive valid sensed data in some MPs.

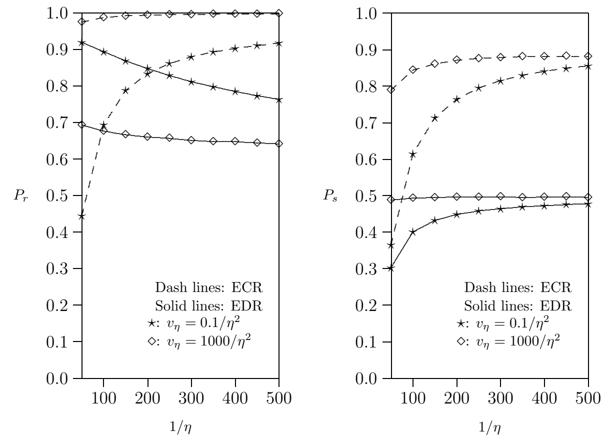
On the other hand, Figure 4(b) and Figure 5(b) show the effects of $1/\lambda$ and v_λ on P_s for ECR. v_λ and $1/\lambda$ do not affect P_s for ECR too much. There is only slightly degradation when $1/\lambda$ and v_λ increase.

2) Effects of SR Residence Time $1/\eta$: Figure 6 and Figure 7 illustrate the effects of η on P_r and P_s for ECR for problem instance 1 and problem instance 2, respectively. Figure 6 shows the effects of η on P_r and P_s for ECR and EDR, where $1/\lambda = 100$, $K = 4$, $A = 1$, and $\rho = 0.4$.



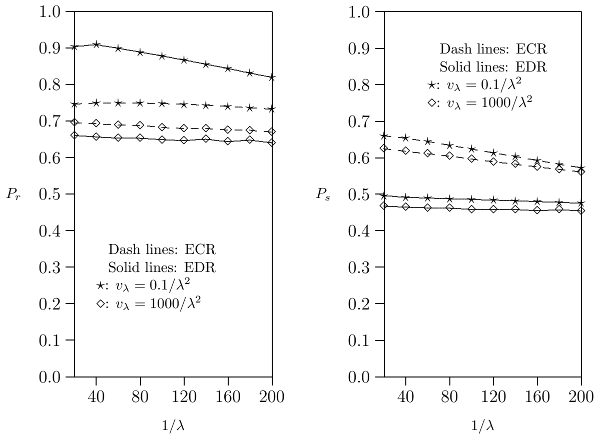
(a) (b)

Fig. 4. Effects of λ on P_r and P_s for ECR and EDR in problem instance 1, where $1/\eta = 100$, $K = 4$, $A = 1$, and $\rho = 0.4$.



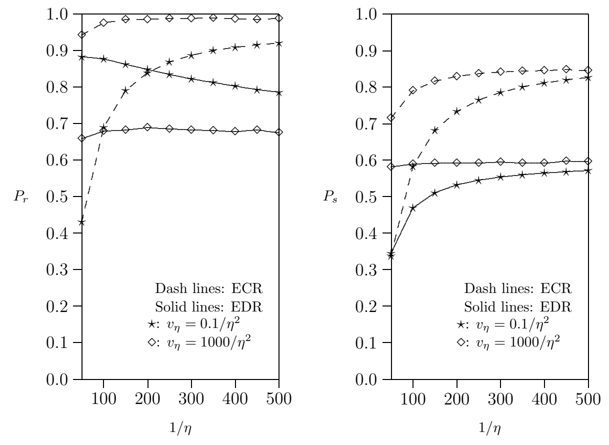
(a) (b)

Fig. 6. Effects of η on P_r and P_s for ECR and EDR in problem instance 1, where $1/\lambda = 100$, $K = 4$, $A = 1$, and $\rho = 0.4$.



(a) (b)

Fig. 5. Effects of λ on P_r and P_s for ECR and EDR in problem instance 2, where $1/\eta = 100$, $K = 4$, $A = 1$, and $\rho = 0.4$.



(a) (b)

Fig. 7. Effects of η on P_r and P_s for ECR and EDR in problem instance 2, where $1/\lambda = 100$, $K = 4$, $A = 1$, and $\rho = 0.4$.

In Figure 6(a), the effects of η on P_r for EDR are much smaller than that for ECR since the situation that assigned M2M nodes leave the SR does not happen in the distributed scheduling. On the other hand, P_s for EDR is lower than that for ECR since the number of M2M nodes stay in the active mode in for ECR is always smaller.

It can be observed that P_r and P_s for ECR increase with $1/\eta$ since the network is more like a static sensor network when the M2M nodes stay longer in an SR. When v_η becomes larger, the possibility that M2M nodes stay in an SR for a longer time is higher. Thus, P_r for ECR approaches to 100% since each M2M node can sense all the types of sensed data. However, when v_η is small, the residence time an M2M node spends in an SR is close to $1/\eta$ which is much smaller than the residence time for larger v_η.

V. Conclusions

To deal with the energy-validity tradeoff for the real-time monitoring applications in M2M communications networks, we proposed the *Energy-efficient Centralized Reporting* (ECR) mechanism, and the *Energy-efficient Distributed Reporting* (EDR) mechanism. In ECR, the energy-validity tradeoff is transformed to the energy minimization problem, and formulated by the ILP. We prove that the energy minimization problem is NP-hard. Then, we propose a low-complexity greedy scheduling algorithm to solve the ILP approximately. In EDR, the *threshold-based dynamic sleeping adjustment* (TDSA) algorithm is proposed for the M2M node to adjust its sleeping period in a distributed fashion. The M2M node can transmit the valid sensed data more efficiently while mitigating the power consumption. Simulation experiments were developed to investigate the P_r and P_s for two proposed mechanisms, ECR and EDR. Our study shows that ECR and EDR both have good P_s performance (i.e., both ECR and EDR can save energy well), and the P_r performance of EDR outperforms that of ECR when M2M nodes are moving dynamically.

ACKNOWLEDGEMENT

The work of P. Lin was sponsored in part by the National Science Council, ROC, under contract numbers NSC 100-2219-E-002-015, NSC 100-2219-E-007-010, NSC 101-2219-E-002-002, NSC 100-2221-E-002-176, and NSC 100-2221-E-002-184. This work was also supported by National Science Council, National Taiwan University and Intel Corporation under Grants NSC 99-2911-I-002-201, NSC 100-2911-I-002-001, and 10R70501.

The work of Y. Fang was partially supported by U.S. National Science Foundation under grant CNS-0916391.

REFERENCES

[1] ETSI, "Machine to Machine Communications (M2M); M2M Service Requirements V1.1.1," *ETSI TS 102 689*, August 2010.

[2] ETSI, "Machine to Machine Communications (M2M); M2M Functional Architecture V1.1.1," *ETSI TS 102 690*, October 2011.

[3] 3GPP, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Service requirements for Machine-Type Communications (MTC); Stage 1 (Release 11)," Tech. Rep. 3G TS 22.368, 3GPP, March 2011.

[4] 3GPP, "3rdGeneration Partnership Project; Technical Specification Group Services and System Aspects; System Improvements for Machine-Type Communications; (Release 11)," Tech. Rep. 3G TS 23.888, 3GPP, June 2011.

[5] Hartenstein, H. and Laberteaux, K.P., "A Tutorial Survey on Vehicular Ad Hoc Networks," *IEEE Communications Magazine*, vol. 46, pp. 164–171, June 2008.

[6] Akyildiz, I.F. Pompili, D., Melodia, T., "Underwater Acoustic Sensor Networks: Research Challenges," *Ad Hoc Networks (Elsevier) Journal*, vol. 3, pp. 257–279, March 2005.

[7] Park, M.K. and Rodoplu, V., "UWAN-MAC: An Energy-Efficient MAC Protocol for Underwater Acoustic Wireless Sensor Networks," *IEEE Journal of Oceanic Engineering*, vol. 32, pp. 710–720, July 2007.

[8] Chen, M., Gonzalez, S., Vasilakos, A., Cao, H., and Leung, V.C.M., "Body Area Networks: A Survey," *ACM/Springer Mobile Networks and Applications*, vol. 16, pp. 171–193, August 2010.

[9] Hanson, M.A., Powell, H.C., Barth, A.T., Ringgenberg, K., Calhoun, B.H., Aylor, J.H., and Lach, J., "Body Area Sensor Networks: Challenges and Opportunities," *Computer*, vol. 42, pp. 58–65, January 2009.

[10] Lu, R., Li, X., Liang, X., Shen, X., and Lin, X., "GRS: The Green, Reliability, and Security of Emerging Machine to Machine Communications," *IEEE Communications Magazine*, vol. 49, pp. 28–35, April 2011.

[11] IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-speed Physical Layer in the 5 GHz Band," *IEEE Std 802.11a*, December 1999.

[12] *ZigBee. http://www.zigbee.org*.

[13] 3GPP, "3rd Generation Partnership Project; Technical Specification Group Radio Access Network; UTRAN overall description (Release 10)," Tech. Rep. 3G TS 25.401, 3GPP, March 2011.

[14] 3GPP, "3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2 (Release 10)," Tech. Rep. 3G TS 36.300, 3GPP, September 2010.

[15] IEEE, "IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems," Tech. Rep. IEEE Std 802.16e, IEEE, February 2006.

[16] Fu, H.-L., Wang, T.-Y., Lin, P., and Fang, Y., "A Region-based Reporting Scheme for Mobile Sensor Networks," in *Proceedings of IEEE VTC2010-Spring Conference*, pp. 1–5, May 2010.

[17] Shi, L. and Fapojuwo, A.O., "TDMA Scheduling with Optimized Energy Efficiency and Minimum Delay in Clustered Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 9, pp. 927–940, July 2010.

[18] Srivastava, R. and Koksal, C.E., "Energy Optimal Transmission Scheduleing in Wireless Sensor Networks," *IEEE Transactions on Wireless Communications*, vol. 9, pp. 1650–1660, May 2010.

[19] Wang, H., Agoulmine, N., and Jin, Y., "Network Lifetime Optimization in Wireless Sensor Networks," *IEEE Journal on Selected Areas in Communications*, vol. 28, pp. 1127–1137, September 2010.

[20] Wang, J., Li, D., Xing, G., and Du, H., "Cross-Layer Sleep Scheduling Design in Service-Oriented Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 9, pp. 1622–1622, November 2010.

[21] Yun, Y. and Xia, Y., "Maximizing the Lifetime of Wireless Sensor Networks with Mobile Sink in Delay-Tolerant Applications," *IEEE Transactions on Mobile Computing*, vol. 9, pp. 1308–1318, September 2010.

[22] Zhong, H. and Xu, C.-H., "Energy-Efficient Wireless Packet Scheduling with Quality of Service Control," *IEEE Transactions on Mobile Computing*, vol. 6, pp. 1158–1170, August 2007.

[23] Lin, C.-K., Zadorozhny, V.I., Krishnamurthy, P.V., Park, H.-H., and Lee, C.-G., "A Distributed and Scalable Time Slot Allocation Protocol for Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 10, pp. 505–518, April 2011.

[24] Rhee, I., Warrier, A., Min, J., and Xu, L., "DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 8, pp. 1384–1396, October 2009.

[25] Cormen, T.H., Leiserson, C.E., Rivest, R.L., and Stein C., *Introduction to Algorithms*. MIT Press and McGrawHill, second ed., 2001.

[26] Lin, Y.-B., and Yang, S.-R., "A Mobility Management Strategy for GPRS," *IEEE Transactions on Wireless Communications*, vol. 2, pp. 1178–1188, November 2003.

[27] Yang, S.-R., and Lin, Y.-B., "Performance Evaluation of Location Management in UMTS," *IEEE Transactions on Vehicular Technology*, vol. 52, pp. 1603–1615, November 2003.

[28] Park, K. Il, and Lin, Y.-B., "Reducing Registration Traffic for Multitier Personal Communications Services," *IEEE Transactions on Vehicular Technology*, vol. 46, pp. 597–602, August 1997.