

國立臺灣大學電機資訊學院資訊工程學系

博士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Doctoral Thesis

M2M通訊網路上即時監控之研究

Real-time Monitoring on Machine-to-Machine

Communications Networks

The seal of National Taiwan University is a circular emblem. It features a central bell (the 'University Bell') flanked by two traditional Chinese lanterns. The seal is surrounded by the university's name in Chinese characters: '國立臺灣大學' at the top and '勵品學安' at the bottom. The seal is rendered in a light gray, semi-transparent watermark style.

傅懷磊

Huai-Lei Fu

指導教授：林風 博士

Advisor: Phone Lin, Ph.D.

中華民國一百零一年七月

July, 2012



Acknowledgement

First of all, I would like to thank my advisor, Dr. Phone Lin, for his guidance, support, encouragements, and patience. His positive thinking and attitude lead me to the right way for doing a good research.

I also would like to thank my co-advisor, Dr. Yuguang Fang, who is a teacher, a researcher and an elder with wisdom. His deep insight brought me a lot of inspirations that were invaluable for all of my life.

I am also grateful for Dr. Ai-Chun Pang, Dr. Wanjiun Liao, Dr. Chu-Sing Yang, and Dr. Yi-Bing Lin for serving on my doctoral committee and giving me valuable feedbacks.

Furthermore, I would like to thank my colleagues, Dr. Chai-Hien Gan, Dr. Shin-Ming Cheng, Dr. Yen Cheng Lai, Chi-Wei Tseng, Chia-Peng Li, Ching-Hsiang Chuang, Kai-Hsiang Hu, Shih-Shi Chen, Chia-Lun Chang, Hou-Chun Chen, Tsung-Che Yu, Shin-Neng Wang, Ting-Yu Wang, Wei-Yen Lin, Zih-Ci Lin, Pai-Chun Chung, Ming-Feng Tsai, Guang-Ming Huang, Kun-Feng Huang, Chen-Hsiang Chen, Yen-Ting Sheu, and En-Hao Ye. We've been through a lot in last five years. I enjoy the accompany of my colleagues.

I also thank my parents who have fully supported me all the time. Their support lets me be able to focus on the research with no worry.

Last, I would like to thank my dear girlfriend, Yi-Tyng Hsu. Without her, I could not go this far. Her love gives me power to face any difficulties.



Chinese Abstract

次世代M2M通訊技術儼然成爲物聯網（Internet of Things）的發展基礎。無須人力介入，龐大數量的機器能夠互相連結用以創造具高度自主性及智能的應用服務。在眾多的M2M應用服務中，最具發展潛力的應用即爲「即時監控」（Real-time Monitoring），其目的是透過區域式的M2M通訊網路來確保即時資料能被持續監控。在即時監控應用中，最重要的研究議題爲傳輸耗能和即時資料正確性的權衡。爲了解決這個議題，本論文提出省電傳輸機制來減緩機器傳輸資料的耗能，同時之間確保即時資料的正確性。本論文總共提出兩個傳輸機制，包含分散式傳輸機制（Energy-efficient Distributed Reporting; EDR）和集中式傳輸機制（Energy-efficient Centralized Reporting; ECR）。此外，本論文也提出分析模型從省電性和資料正確性兩方面來研究傳輸機制之效能。再者，在本論文中，我們實作空氣污染感測系統（Air Pollution Sensing System; APS）。APS系統爲即時監控系統之雛形，用以監控都市區域的空氣品質。此系統已經建置於台灣大學中進行小規模的實測。

關鍵字: M2M通訊; 效能模型分析; 即時監控; 回報機制



English Abstract

Machine-to-Machine (M2M) communications have emerged as a new communication paradigm to support Internet of Things (IoT) applications. With little or even no human intervention, the interconnection among a large amount of machines enables highly autonomous and intelligent applications. Undoubtedly, the most promising M2M application is “real-time monitoring”, the goal of which is to ensure continuous surveillance via region-based M2M communications networks. One major issue for real-time monitoring is the energy-validity tradeoff, i.e., the tradeoff between the energy efficiency for machines and the validity of sensing reports. To deal with the energy-validity tradeoff for real-time monitoring applications, in this thesis, we propose energy-efficient reporting mechanisms, including Energy-efficient Distributed Reporting (EDR) and Energy-efficient Centralized Reporting (ECR), to mitigate energy consumption for machines while keeping the sensing data valid. We propose analytical models and simulation experiments to investigate the performance for the proposed mechanisms in terms of the power saving and the report validity. Furthermore, we develop a prototyping real-time monitoring system, namely the- Air Pollution Sensing (APS) system, to monitor the air quality in the urban environment. The APS system has been deployed in the campus of National Taiwan University for small-scale testing.

Keywords: Energy-validity tradeoff; machine-to-machine communications; performance modeling; real-time monitoring; reporting mechanism



Contents

Acknowledgement	i
Chinese Abstract	iii
English Abstract	v
1 Introduction	1
1.1 Real-time Monitoring Applications	2
1.2 Our Contributions	6
1.3 Organization of the Dissertation	8
2 Problem Definition	9
3 EDR: Energy-efficient Distributed Reporting	13
3.1 Mechanism Operations	13
3.2 Analytical Models	17
3.2.1 Derivation of Power Saving Probability P_{ps}	20



3.2.2	Derivation of Outdated Monitoring Probability P_{om}	29
3.2.3	Calculation of p	36
3.2.4	Simulation Model Validation	38
3.3	Performance Evaluation	39
3.3.1	Effects of α	39
3.3.2	Effects of T_{sleep}	41
3.3.3	Effects of K	42
3.3.4	Effects of θ	44
4	ECR: Energy-efficient Centralized Reporting	47
4.1	Mechanism Operations	48
4.2	Energy Minimization for Centralized Scheduling	50
4.2.1	Assumptions and Notations	50
4.2.2	ILP Formulation	51
4.2.3	Problem Complexity	52
4.2.4	Greedy Scheduling Algorithm	53
4.3	Performance on ILP and Greedy Approach	54
5	APS: Air Pollution Sensing System	57
5.1	System Architecture	57

<i>CONTENTS</i>	ix
5.1.1 The Sensing Device	58
5.1.2 The Back-end Server	60
5.1.3 The User Device	61
5.2 Software Architecture	62
5.2.1 The Software Architecture of The Back-end Server	62
5.2.2 The Software Architecture of The Sensing Device	71
5.3 Message Flows	74
5.3.1 The Sensing Report Transmission Procedure	74
5.3.2 The Sleeping Period Adjustment Procedure	77
5.3.3 The Navigation Service Procedure	79
6 Conclusions and Future Work	83
Bibliography	85
A The Mathematical Proof	93
B The DCF of IEEE 802.11	95
Curriculum Vita	98



List of Figures

1.1	The network architecture of a region-based M2M communications network	4
2.1	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	11
3.1	The timing diagram for the behavior of an arbitrary M2M node in the network	22
3.2	The timing diagram for MPs, receiving and non-receiving periods for the M2M gateway	29
3.3	Effects of α on P_{ps} and P_{om} for BDSA and TDSA, where $\gamma = (1/6.4)\mu$, $\theta = 100/\mu$, $A = 4$, $K = 4$, and $\rho = 0.4$	40
3.4	Effects of A on P_{ps} and P_{om} for BDSA and TDSA, where $\gamma = (1/6.4)\mu$, $\theta = 100/\mu$, $K = 4$, and $\rho = 0.4$	42
3.5	Effects of K on P_{ps} and P_{om} for TDSA, where $\gamma = (1/6.4)\mu$, $\theta = 100/\mu$, $A = 4$, and $\rho = 0.4$	43

3.6	Effects of α on P_{ps} and P_{om} for BDSA and TDSA, where $\gamma = (1/6.4)\mu$, $A = 4$, $K = 4$, and $\rho = 0.4$	44
4.1	An example for a transmission schedule for the M2M nodes n_1, n_2, n_3 . . .	49
5.1	An example of sensing regions	58
5.2	The system architecture for the APS system	59
5.3	A bicycle equipped with a netbook, a GPS receiver, a CO ₂ sensor, and a 3G modem	60
5.4	The software architecture for the back-end server	62
5.5	An example of the user profile stored in the database	64
5.6	An example of the air quality data stored in the database	64
5.7	The pseudo code of the EDR mechanism implemented in the report manager	67
5.8	Monitoring service	68
5.9	An example of road sections	70
5.10	Navigation Service	70
5.11	The software architecture for the sensing device	71
5.12	The message flow for the sensing report transmission procedure	75
5.13	The message flow for the sleeping period adjustment procedure	77
5.14	The message flow for the navigation service procedure	79

B.1 The operation of the DCF in the IEEE 802.11 protocol 96





List of Tables

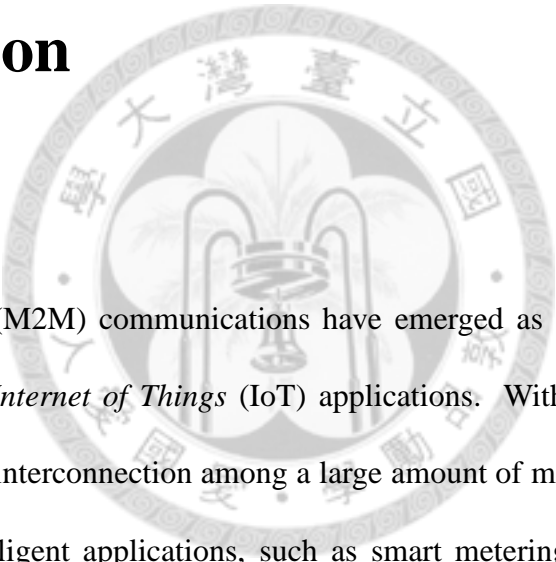
3.1	Validation of the simulation and analysis results for P_{ps} and P_{om}	39
4.1	Comparison between MATLAB ILP solver and greedy algorithm	55





Chapter 1

Introduction



Machine-to-Machine (M2M) communications have emerged as a new communication paradigm to support *Internet of Things* (IoT) applications. With little or even no human intervention, the interconnection among a large amount of machines enables highly autonomous and intelligent applications, such as smart metering, healthcare monitoring, fleet management and tracking, environmental monitoring and industrial automation. Examples of M2M communications networks include vehicular ad-hoc networks (VANETs) [29], underwater sensor networks (UWSNs) [17, 42], and wireless body area networks (WBANs) [22, 28]. The realization of M2M communications has been worked out by the 3rd Generation Partnership Project (3GPP), namely *Machine-Type Communications* (MTC) [15, 16], and by the European Telecommunications Standards Institute (ETSI) [25, 26]. In the near future, we can foresee that there will be a widespread increase in the amount of machines interconnecting through the cellular infrastructure.

1.1 Real-time Monitoring Applications

As pointed out in [36], the most promising M2M application is the *real-time monitoring* application, whose objective is to collect real-time data (e.g., temperature, humidity, air quality and toxic gas) from machines to ensure continuous surveillance for application-oriented purposes. Several real-time monitoring systems have been developed, e.g., CitySense [30, 39], Center for ENvironmental Sensing And Modeling (CENSAM) [37] and Personal Environmental Impact Report (PEIR) [38, 51]. The CitySense project aims to deploy an urban-scale wireless sensor network. In CitySense, a sensor is an embedded PC equipped with IEEE 802.11 a/b/g radios. About 100 sensors are mounted on buildings and street lights across the city of Cambridge for surveillance, and the buildings or streets provides power supply for sensors. The CENSAM project provides pervasive monitoring and control within urban environment in Singapore. The environmental sensors (e.g., airflow sensors on the roadside and chemical sensors on the autonomous underwater vehicles) are deployed to collect data ranging from urban atmospheric conditions to marine water quality. The collected data are analyzed to better understand the regional climate around Singapore and to estimate wind stress at the surface of the South China Sea. The PEIR project uses location information of mobile devices to calculate personalized estimates of environmental impact and exposure. The mobile device uploads its Global Positioning System (GPS) data to the server side. Based on the received GPS data, the PEIR system uses an activity classification model to determine the user transportation mode, e.g., staying in one specific location, walking, or driving. Then, the PEIR system takes the results from the activity classification model as inputs to the emissions factors model to generate estimates of environmental impact and exposure for the user.

In our project, we propose the Air Pollution Sensing system (APS)¹ [27] to monitor air quality for the metropolitan area in Taiwan, where air pollution is mainly caused by traffic jam under which crowded vehicles are with low mobility and their emissions result in high concentration of air pollutants. The air quality changes very rapidly with many factors, such as traffic load, weather and wind velocity. It requires certain control policies to routinely monitor air quality based on real-time surveillance results (e.g., current air quality reports and air pollution warning notifications). In APS, we partition the metropolitan area into multiple regions and install sensors on either motorists' helmets or cars to collect air quality data (e.g., pollutant levels) in each region, such that the urban-scale air quality can be effectively and efficiently monitored. In every monitoring period, the collected air quality data are processed and analyzed, and used for specific purposes at the end of every monitoring period. For example, we can keep pedestrians informed about the polluted areas, and pedestrians can choose less polluted routes. This real-world application scenario can monitor the environment well with small fraction of sampling of sensed data because the sensors (installed on motorists' helmets or cars) keep moving around.

To better articulate the problem we intend to address, we first give a generic example for an M2M communications network comprises a number of *M2M nodes* (i.e., machines) and M2M gateways. As shown in Figure 1.1(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 [32], ZigBee [10],

¹The APS won the championship at the 2008 Microsoft Imagine Cup, Taiwan Area.

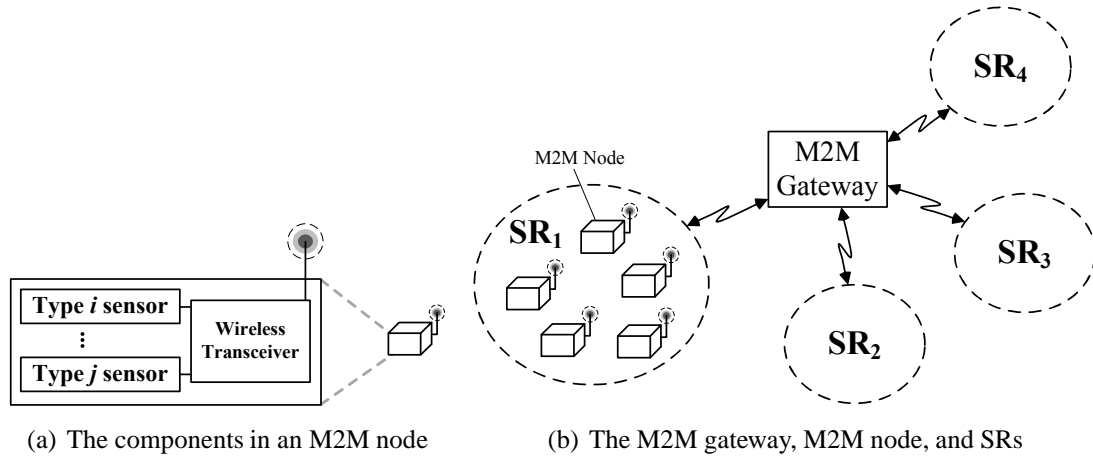


Figure 1.1: The network architecture of a region-based M2M communications network

UMTS [14], LTE [13], or WiMAX [31].

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 computation on the collected sensed data from M2M nodes.

For real-time monitoring applications, 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 1.1(b) illustrates an example of an M2M communications network partitioned into four SRs, namely SR₁, SR₂, SR₃, and SR₄. In this example, there are one M2M gateway and five M2M nodes in SR₁.

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.

In real-time monitoring applications, the time is divided into multiple *monitoring periods* (MPs). By the end of each monitoring period, the collected real-time data are processed and analyzed by the central controller for autonomous and intelligent decision making. For each type of sensed data, there is a corresponding MP as its time constraint. We deem a sensed data to be a *valid* sensed data 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 later. 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 named the *energy-validity* tradeoff. Careful scheduling is required for sensed data transmission to prevent redundant transmissions, save power for M2M nodes, and prolong the lifetime of the M2M communications network. Meanwhile, the M2M gate-

way can maintain the freshness of the sensed data to achieve real-time monitoring.

To address this issue, many previous studies [46, 47, 52, 53, 55, 56] 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 [46, 47, 53] 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 [34, 44] 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.

1.2 Our Contributions

In this dissertation, we investigate the energy-validity tradeoff for “multi-type” real-time monitoring applications on the M2M communications network, in which M2M nodes are mobile. We propose intelligent reporting mechanisms to mitigate energy consumption for M2M nodes while keeping the sensing data valid. Our contributions are threefold:

First, 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 *Dynamic Sleeping Adjustment* (DSA) algorithms for the M2M node to adjust its sleeping period. We design analytical models in the discrete-time domain to investigate the performance of EDR based on the assumption that the communication protocol between the M2M node and gateway is IEEE 802.11 [32]. The analytical models are sophisticated to capture the reporting and mobility behaviors of M2M nodes at the same time. In most previous studies [19, 23], the reporting and mobility behaviors are not jointly considered.

Second, 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.

Third, as mentioned previously, we develop a prototyping real-time monitoring system, the distributed *Air Pollution Sensing* (APS) system, to monitor the air quality in the urban environment. The APS system implements the proposed reporting mechanisms for air quality monitoring. We design complete software architecture and message flows. The APS system has been deployed in the campus of National Taiwan University for small-scale testing.

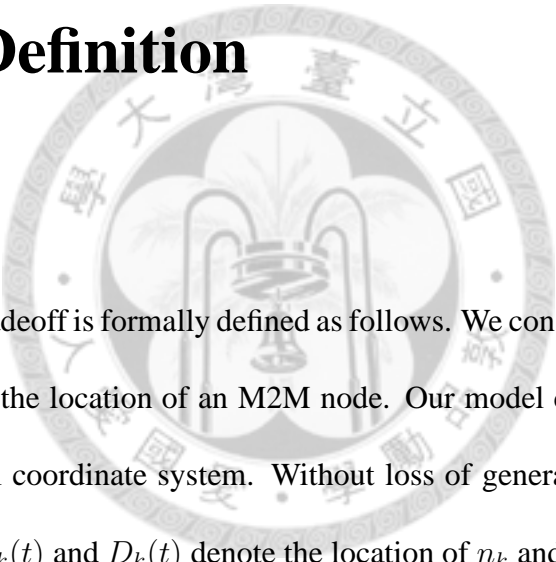
1.3 Organization of the Dissertation

The rest of this dissertation is organized as follows. In Chapter 2, the energy-validity tradeoff is formally defined. The problem definition is comprehensive enough to express many different application settings. In Chapter 3, we present the EDR mechanism and its analytical models. In Chapter 4, we present the ECR mechanism with the corresponding ILP formulation and complexity analysis. In Chapter 5, the details of the APS system is presented. Chapter 6 concludes the dissertation.



Chapter 2

Problem Definition



The energy-validity tradeoff is formally defined as follows. We consider a two-dimensional coordinate system for the location of an M2M node. Our model can be easily extended to a three-dimensional coordinate system. Without loss of generality, let n_k denote the k th M2M node. Let $L_k(t)$ and $D_k(t)$ denote the location of n_k and the data sensed by n_k at time t , respectively. A sensing report generated by n_k at time t consists of $L_k(t)$ and $D_k(t)$, which is denoted as $\mathbb{R}_k(t) = \{L_k(t), D_k(t)\}$.

Suppose that the system starts at time t_0 . Assume that the M2M gateway correctly receives a sensing report $\mathbb{R}_k(t_1) = \{L_k(t_1), D_k(t_1)\}$ (generated by n_k at t_1) at t_2 , where $t_0 < t_1 < t_2$. The service area of a M2M gateway is divided into multiple m SRs, and SR_j denotes the j th SR. An SR is modeled as a rectangle with the length l and the width w . Let $C_j = (\hat{x}_j, \hat{y}_j)$ be the center of SR_j . The M2M gateway determines that $\mathbb{R}_i(t_1)$

belongs to SR_j if $L_i(t_1) = (x_i, y_i)$ satisfies

$$\hat{x}_j - l/2 \leq x_i < \hat{x}_j + l/2, \quad (2.1)$$

and

$$\hat{y}_j - w/2 \leq y_i < \hat{y}_j + w/2. \quad (2.2)$$

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 $\mathbb{S} = \{1, 2, 3, \dots, N\}$. 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. Let t_a (t_s) denote the period when n_k is in the active (sleeping) period. Figure 2.1 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 j th 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

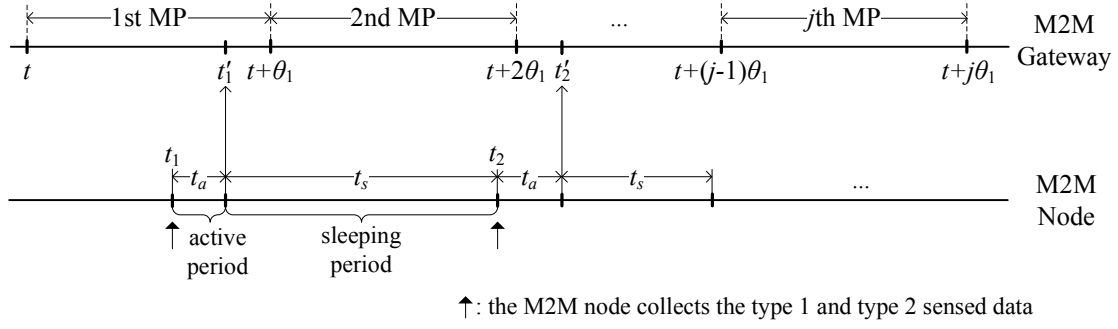


Figure 2.1: An example that the M2M gateway monitors the type 1 sensed data and the M2M node transmits the type 1 and type 2 sensed data

an M2M node at time t_x in the j th MP (i.e., $t + (j - 1)\theta_i \leq t_x \leq t + j\theta_i$), and received by the M2M gateway at t'_x (where $t_x < t'_x$). Then we define that the type i sensed data is *valid* if

$$t + (j - 1)\theta_i \leq t_x < t'_x \leq t + j\theta_i. \quad (2.3)$$

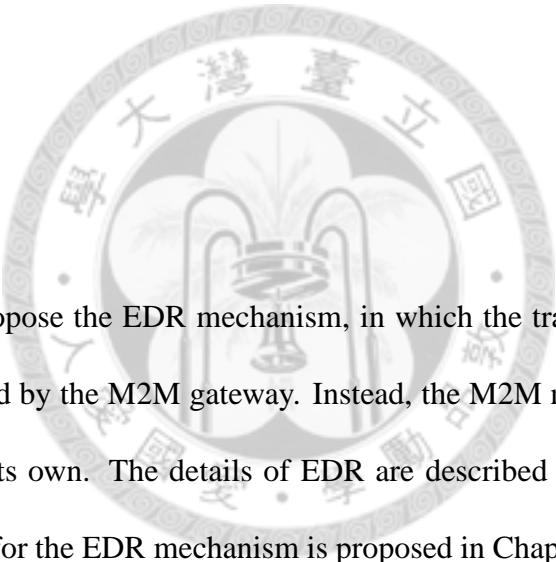
Otherwise (i.e., $t'_x > t + j\theta_i$), the type i sensed data is *invalid*.

As shown in Figure 2.1, 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 $t \leq t_1 < t'_1 \leq t + \theta_1$, 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 $t + \theta_1 < t_2 \leq t + 2\theta_1$ and $t'_2 > t + 2\theta_1$, the type 1 sensed data is invalid.



Chapter 3

EDR: Energy-efficient Distributed Reporting



In this chapter, we propose the EDR mechanism, in which the transmission schedule is not centrally controlled by the M2M gateway. Instead, the M2M node decides the transmission schedule on its own. The details of EDR are described in Chapter 3.1. Then, the analytical models for the EDR mechanism is proposed in Chapter 3.2, and the performance evaluation is given in Chapter 3.3.

3.1 Mechanism Operations

The EDR mechanism consists of two parts: the M2M gateway and M2M node parts, whose details are given as follows.

M2M Gateway Part: We consider an SR in which there are M types of sensed data to

be collected, and the M2M gateway monitors M types of sensed data for each SR.

The M2M gateway maintains the following information: the set

$$\mathbb{D} = \{D(1), D(2), \dots, D(M)\}$$

for the M 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 (2.3). 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.

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 x th type i MP, the variable $I_i(x)$ is maintained to indicate that the x th 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):

$$I_i(x) = \begin{cases} 1, & \text{if } CR_i = 0; \\ 0, & \text{otherwise.} \end{cases} \quad (3.1)$$

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

$$\Gamma_i(x) = \begin{cases} \Gamma_i(x-1) + I_i(x), & \text{if } x \bmod K \neq 0; \\ \frac{\Gamma_i(x-1) + I_i(x)}{K}, & \text{otherwise.} \end{cases} \quad (3.2)$$

At the end of every K type M MPs, the latest $\Gamma_1, \Gamma_2, \dots, \Gamma_M$ 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 $1 \leq i \leq M$.

M2M Node Part: An M2M node maintains the sleep timer T_{sleep} to control its sleeping period, and the transmission timers $T_{tx,1}, T_{tx,2}, \dots, T_{tx,M}$ 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 $T_{tx,1}, T_{tx,2}, \dots, T_{tx,M}$. The details of the TDSA algorithm are given as follows.

Let $V(j)$ and $V_i(j)$ refer to the j th configuration for the T_{sleep} timer and the $T_{tx,i}$

Algorithm 1: Dynamic Sleeping Adjustment

Input: $\Gamma_1, \Gamma_2, \dots, \Gamma_M$, and $V_1(x), V_2(x), \dots, V_M(x)$
Output: $V(x+1), V_1(x+1), V_2(x+1), \dots, V_M(x+1)$

```

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    $V_i(x+1) \leftarrow \text{Round}(\frac{V'_i(x+1)}{V(x+1)})$ ;
13 end

```

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, $V(1), V_1(1), V_2(1), \dots, V_M(1)$ are all set to θ_1 . Based on the ratio $\Gamma_1, \Gamma_2, \dots, \Gamma_M$ in the received notification message, $V(x+1), V_1(x+1), V_2(x+1), \dots, V_M(x+1)$ are adjusted according to the following algorithm.

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 $\text{Round}(\cdot)$ 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} .

3.2 Analytical Models

In this section, we propose the analytical models for EDR by using Basic Dynamic Sleeping Adjustment (BDSA) for the M2M node, i.e., the configuration for the T_{sleep} timer is randomly generated by the M2M node using the geometric distribution with mean $A\theta$. The analytical models for TDSA are not conducted due to lack of the independence property of sleeping periods. Instead, we evaluate the performance for TDSA through simulation experiments. The reasons to conduct the analytical models for BDSA are to cross-validate the correctness of the simulation experiments and to analyze both the reporting and mobility behaviors for M2M nodes by jointly considering the wireless medium contention and M2M node arrival and departure processes in an SR. Besides, in the analytical models, we only consider single-type EDR to simplify our discussion.

We use two output measures, namely, the power saving probability P_{ps} (for an M2M node) and the outdated monitoring probability P_{om} (for the M2M gateway). We consider the system in the steady state, i.e., for an event occurring at time t with probability $p(t)$,

we consider the steady state probability $\pi = \lim_{t \rightarrow \infty} p(t)$ instead of $p(t)$. Thus, we define P_{ps} and P_{om} , respectively, as

$$P_{ps} = \lim_{t \rightarrow \infty} \Pr[\text{an M2M node is in the sleep mode at time } t],$$

and

$$P_{om} = \Pr[\text{an MP is an outdated MP}].$$

It is clear that larger P_{ps} implies that the M2M node consumes less power, and larger P_{om} indicates that an MP is more likely an outdated MP.

Because the EDR mechanism is based on the DCF in IEEE 802.11 protocol, we follow the studies [21, 49] to model the system in discrete-time. Suppose that the system time is segmented into a sequence of equal time slots, $0, s, 2s, 3s, \dots$, and each time slot lasts for s time units. Without loss of generality, we assume $s = 1$, i.e., the i th time slot starts at time $i - 1$ and ends at time i . Thus the system time t is an integer value.

Owing to the duration of a time slot is short (i.e., $20 \mu s$ for the frequency hopping physical layer of IEEE 802.11 [32]), we use the Bernoulli process to model the M2M node arrival process to an SR. The Bernoulli process is a discrete-time version of the Poisson process and has been used to study the mobile node arrival process in many studies (e.g., [33]). Suppose that the M2M node arrivals to the SR form the Bernoulli process with parameter λ (i.e., an M2M node arrives in a time slot with probability λ , and no M2M node arrival occurs in a time slot with probability $1 - \lambda$). Thus, the inter-node arrival time

period is geometrically distributed with mean $1/\lambda$. With the assumption of Bernoulli, our analytical model can provide mean value analysis. We release this assumption in our simulation experiments and treat the inter-node arrival time as a negative binomial distribution. The second moment analysis is also provided in our study.

As mentioned previously, we consider one-hop communication between the M2M nodes and the M2M gateway, and an M2M node can transmit the sensing report to the M2M gateway directly through the WiFi service. In such case, the area size of the SR affects the propagation delay caused by the wireless transmission distance between the M2M nodes and the M2M gateway. However, the propagation delay is not so significant compared to the delay caused by the contention of the wireless medium, which we mainly concern in our analysis. Therefore, to simplify our discussion, we use the SR residence time for the M2M node to reflect the factor of the SR size. Furthermore, the SR residence time for an M2M node is assumed to be geometrically distributed with mean $1/\eta$. In the simulation experiments, we also release this assumption by supposing that the SR residence time has a negative binomial distribution.

Note that in our study, we consider an SR because in real-world scenarios, the configuration of SRs is irregular, and it is extremely hard to identify the relationships among SRs. The M2M node arrival process for an SR is independent from the M2M node departure process for the same SR, that is, λ and η are independent. In the performance evaluation, we will use the factor $\alpha = \lambda/\eta$ to study the effects of the M2M node mobility behavior on the P_{ps} and the P_{om} performances.

We assume an M2M node accesses the wireless medium by following the *p-persistent* IEEE 802.11 protocol [20] as the random backoff interval. It has been shown that the per-

formance for *p-persistent* IEEE 802.11 protocol closely approximates to the performance for the IEEE 802.11 protocol [20]. Thus the random backoff interval is assumed to be geometrically distributed with the probability γ , i.e., in a time slot, the M2M node has probability γ to transmit the Request To Send (RTS) to contend for the wireless medium.

3.2.1 Derivation of Power Saving Probability P_{ps}

Consider the timing diagram in Figure 3.1, where an M2M node executes the BDSA to set up T_{sleep} . To simplify our discussion, we assume there is no delay between the time instant when the M2M node generates a sensing report and that when the transmission for the sensing report starts. In other words, we do not consider the effects of buffering for sensed data in the M2M node in our analysis because the delay is negligibly short.

Let t_a be the period when the M2M node stays in the active mode, during which the M2M node contends for the wireless medium by executing the DCF with other M2M nodes in the same SR. After transmitting the sensing report successfully, the M2M node switches to the sleep mode for a t_s period. Because the M2M node alternatively stays in the active and sleep mode, from the alternating renewal process [40,45], we have

$$P_{ps} = \frac{E[t_s]}{E[t_a] + E[t_s]}. \quad (3.3)$$

In BDSA, the t_s is generated from the geometric distribution with mean $A\theta$, and we have

$$E[t_s] = A\theta. \quad (3.4)$$

We derive $E[t_a]$ as follows. Let p be the probability that a random observer sees an M2M node performing the DCF for contention of the wireless medium, i.e.,

$$p = \lim_{t \rightarrow \infty} \Pr[\text{an M2M node performs DCF at time } t].$$

The derivation of p depends on $E[t_a]$ and $E[t_s]$ and will be elaborated in Chapter 3.2.3.

Suppose that there are N_a M2M nodes, n_1, n_2, \dots, n_{N_a} , in the active mode that use the DCF to contend for the wireless medium. Observe the behavior of n_x , where $x \in \{1, 2, \dots, N_a\}$. Let $\Pr[N_{a|x} = m]$ denote the probability that there are $N_a = m$ M2M nodes executing the DCF for contention of the wireless medium under the condition that n_x is performing the DCF for contention of the wireless medium. From Appendix A, we have the conditional probability $\Pr[N_{a|x} = m]$ as

$$\Pr[N_{a|x} = m] = \Pr[N_a = m - 1] = \frac{(\alpha p)^{(m-1)} e^{-\alpha p}}{(m-1)!}, \quad (3.5)$$

where $\alpha = \lambda/\eta$.

In Figure 3.1, we assume that n_x 's active period starts at t_0 and ends at $t_0 + t_a$. During the period between t_0 and $t_0 + t_a$, let X denote the number of contentions made by n_x for the wireless medium before n_x seizes the medium and transmit a sensing report at the $X + 1$ st contention. Let $t_{c,i}$ be the i th contention access period ($1 \leq i \leq X + 1$), and $t_{c,1}, t_{c,2}, \dots, t_{c,X+1}$ are i.i.d. random variables. Note that $t_{c,i}$ may not be i.i.d. in the standard IEEE 802.11 protocol. In [18, 19], the authors provided analytical results based on the non-i.i.d. $t_{c,i}$ assumption. However, this work only considered stationary sensors and the number of active sensors was fixed. In our proposed models, we consider the M2M node

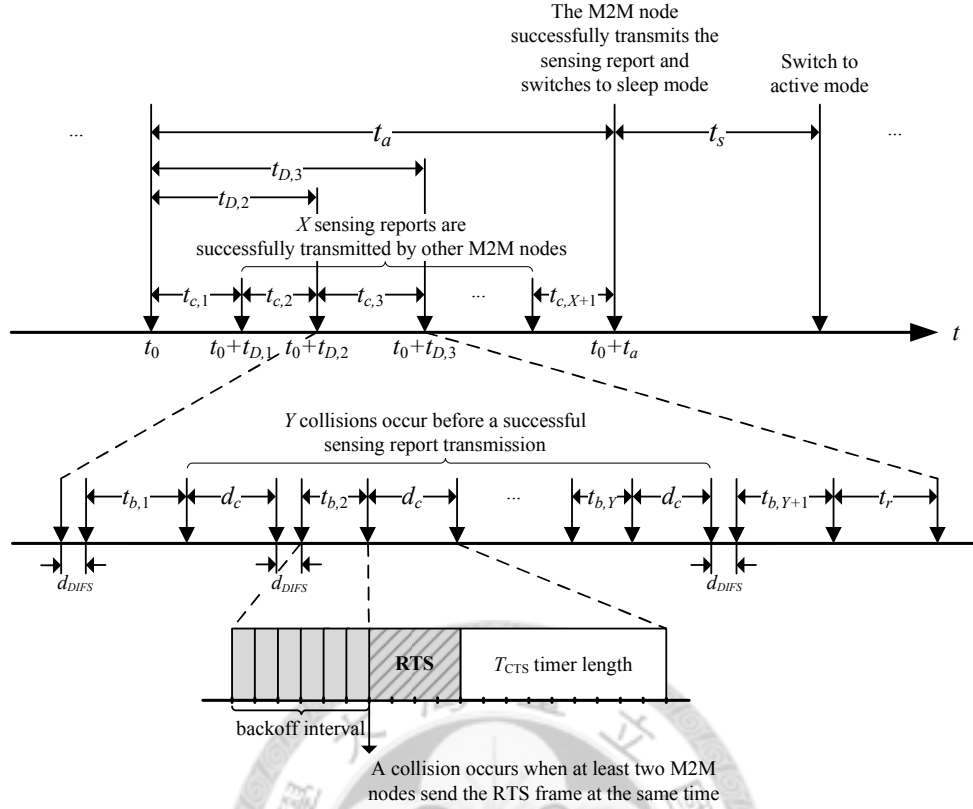


Figure 3.1: The timing diagram for the behavior of an arbitrary M2M node in the network

arrival and departure processes in an SR, which was not addressed in the previous work.

In this paper, the analytical model is based on the *p-persistent* IEEE 802.11 protocol. The difference between the *p-persistent* and standard IEEE 802.11 protocols is the selection of the random backoff interval. In the *p-persistent* IEEE 802.11 protocol, the random backoff interval is sampled from a geometric distribution. In other words, in a time slot, the M2M node has a probability to transmit the RTS to contend for the wireless medium. Based on the operations of the *p-persistent* IEEE 802.11 protocol, we assume $t_{c,i}$ are i.i.d. random variables to simplify our discussion. The performance evaluation in [20] has observed that the performance for the *p-persistent* IEEE 802.11 protocol closely approximates the performance for the standard IEEE 802.11 protocol given that the means of the random backoff intervals in the two protocols are the same.

During $t_{c,i}$ ($i \neq X + 1$), there exists another n_y ($y \neq x$) that contends successfully and transmits a sensing report. In $t_{c,X+1}$, n_x contends successfully and transmits a sensing report. Based on Wald's equation [45], we obtain

$$\begin{aligned}
 E[t_a] &= \sum_{k=0}^{\infty} E \left[\sum_{i=1}^{k+1} t_{c,i} \right] \Pr[X = k] \\
 &= \sum_{k=0}^{\infty} (k + 1) E[t_{c,i}] \Pr[X = k] \\
 &= E[t_{c,i}] (E[X] + 1).
 \end{aligned} \tag{3.6}$$

Derivation of $E[t_{c,i}]$ In Figure 3.1, let $t_{D,i} = t_{c,1} + t_{c,2} + \dots + t_{c,i}$, where $i \geq 1$. By convention, $t_{D,0} = 0$. Consider the i th contention access period $t_{c,i}$ for n_x (i.e., the period between $t_0 + t_{D,i-1}$ and $t_0 + t_{D,i}$). Let d_{DIFS} be the fixed DIFS interval. At $t_0 + t_{D,i-1}$, based on the DCF, n_x determines the wireless medium idle by sensing the wireless medium for d_{DIFS} . After a DIFS interval (i.e., at $t_0 + t_{D,i-1} + d_{DIFS}$), n_x starts to contend for the wireless medium with the other $(N_a - 1)$ M2M nodes by waiting for a random backoff interval, and stops at the time when there is at least one M2M node (among N_a M2M nodes) transmitting the RTS frame.

Define an idle period $t_{b,i}$ as the period between the time when n_x starts to wait for a random backoff interval and the time when there is at least one M2M node (among N_a M2M nodes) transmitting the RTS frame, where the index i denotes the i th idle period. Suppose that among N_a M2M nodes, N_t M2M nodes transmit RTS frames at the same time t in $t_{c,i}$ for n_x , where $0 \leq N_t \leq N_a$. Let $\Pr[N_{t|x} = m] = \Pr[N_t = m | n_x \text{ is performing DCF}]$. The probability $\Pr[N_{t|x} = 0]$ denotes the probability that the wireless medium is sensed idle by n_x . Then we have

$$\begin{aligned}\Pr[N_{t|x} = m] &= \Pr[N_t = m | n_x \text{ is performing DCF}] \\ &= \sum_{n=m}^{\infty} \Pr[N_t = m | N_a = n, n_x \text{ is performing DCF}] \Pr[N_{a|x} = n].\end{aligned}$$

Since the random backoff is geometrically distributed with probability γ , we have

$$\Pr[N_t = m | N_a = n, n_x \text{ is performing DCF}] = \binom{n}{m} \gamma^m (1 - \gamma)^{n-m}. \quad (3.7)$$

Applying (3.7), we have

$$\begin{aligned}\Pr[N_{t|x} = m] &= \sum_{n=m}^{\infty} \binom{n}{m} \gamma^m (1 - \gamma)^{n-m} \left[\frac{(\alpha p)^{(n-1)} e^{-\alpha p}}{(n-1)!} \right] \\ &= \frac{e^{-\alpha \gamma p} (\alpha \gamma p)^{m-1} \gamma (\alpha p - \alpha \gamma p + m)}{m!}.\end{aligned} \quad (3.8)$$

Let $\Pr[t_{b,i} = k]$ be the probability that the wireless medium is sensed idle by n_x for k slots (i.e., the length of $t_{b,i}$ is k slots). Then we have

$$\begin{aligned}\Pr[t_{b,i} = k] &= \Pr[N_{t|x} = 0]^k (1 - \Pr[N_{t|x} = 0]) \\ &= [(1 - \gamma)e^{-\alpha \gamma p}]^k [1 - (1 - \gamma)e^{-\alpha \gamma p}],\end{aligned}$$

and

$$E[t_{b,i}] = \sum_{k=1}^{\infty} k \Pr[t_{b,i} = k] = \frac{(1 - \gamma)e^{-\alpha \gamma p}}{1 - (1 - \gamma)e^{-\alpha \gamma p}}. \quad (3.9)$$

After an idle period $t_{b,i}$, there must be at least one M2M node transmitting the RTS frame, which may result in a collision or a successful transmission for the RTS frame. Since the size of an RTS frame is small (i.e., 20 bytes specified in the IEEE 802.11 standard), we assume the transmission time for the RTS frame is fixed to d_{RTS} . In DCF, if n_x transmits the RTS frame, it triggers the T_{CTS} timer when the RTS frame transmission is finished and expects to receive the Clear to Send (CTS) frame before T_{CTS} expires (the length of T_{CTS} is $d_{CTS-timeout}$). Thus, n_x waits for the time $d_c = d_{RTS} + d_{CTS-timeout}$ to determine whether a collision occurs or not (see Figure 3.1). Otherwise (i.e., n_x does not transmit the RTS), n_x waits for a period equal to d_c . Let Y be the number of collisions during $t_{c,i}$ for n_x . Let P_{coll} be the probability that a collision occurs at t in $t_{c,i}$ for n_x , i.e., at least two M2M nodes transmit the RTS frames at t conditioned on that there is at least one M2M node transmitting the RTS frame. From [20], we have

$$\begin{aligned}
P_{coll} &= \Pr[N_t \geq 2 | N_t \geq 1, n_x \text{ is performing DCF}] \\
&= \frac{\Pr[N_t \geq 2 | n_x \text{ is performing DCF}]}{\Pr[N_t \geq 1 | n_x \text{ is performing DCF}]} \\
&= \frac{\Pr[N_{t|x} \geq 2]}{\Pr[N_{t|x} \geq 1]} \\
&= \frac{1 - \Pr[N_{t|x} = 1] - \Pr[N_{t|x} = 0]}{1 - \Pr[N_{t|x} = 0]}. \tag{3.10}
\end{aligned}$$

Applying (3.8), we have

$$P_{coll} = \frac{1 - \gamma e^{-\alpha\gamma p}(1 + \alpha p - \alpha\gamma p) - (1 - \gamma)e^{-\alpha\gamma p}}{1 - (1 - \gamma)e^{-\alpha\gamma p}}. \tag{3.11}$$

Let P_{succ} be the probability that exactly one M2M node transmits the RTS frame at t conditioned on that there is at least one M2M node transmitting the RTS frame.

Similar to the derivation for P_{coll} , we have

$$\begin{aligned}
 P_{succ} &= \Pr[N_t = 1 | N_t \geq 1 \text{ and } n_x \text{ is performing DCF}] \\
 &= \frac{\Pr[N_{t|x} = 1]}{\Pr[N_{t|x} \geq 1]} \\
 &= \frac{\gamma e^{-\alpha\gamma p} (1 + \alpha p - \alpha\gamma p)}{1 - (1 - \gamma)e^{-\alpha\gamma p}}. \tag{3.12}
 \end{aligned}$$

Then from (3.11) and (3.12), we have

$$\begin{aligned}
 \Pr[Y = k] &= P_{coll}^k P_{succ} \\
 &= \left[\frac{1 - \gamma e^{-\alpha\gamma p} (1 + \alpha p - \alpha\gamma p) - (1 - \gamma)e^{-\alpha\gamma p}}{1 - (1 - \gamma)e^{-\alpha\gamma p}} \right]^k \left[\frac{\gamma e^{-\alpha\gamma p} (1 + \alpha p - \alpha\gamma p)}{1 - (1 - \gamma)e^{-\alpha\gamma p}} \right],
 \end{aligned}$$

and

$$E[Y] = \sum_{k=0}^{\infty} k \Pr[Y = k] = \frac{\alpha\gamma p(\gamma - 1) + e^{\alpha\gamma p} - 1}{\gamma(1 + \alpha p - \alpha\gamma p)}. \tag{3.13}$$

Assume that the transmission times for the CTS and ACK are fixed to d_{CTS} and d_{ACK} .

Let t_{data} be the transmission time for a sensing report (i.e., the data frame), which is assumed to be geometrically distributed with mean $1/\mu$. The SIFS interval is fixed to d_{SIFS} . We do not consider the propagation delay since the delay is far shorter than the transmission times of the RTS, CTS and ACK frames. Let t_r be the total time for an M2M node to transmit a sensing report, including the signaling overhead.

Then, we have

$$\begin{aligned} E[t_r] &= E[d_{RTS} + d_{SIFS} + d_{CTS} + d_{SIFS} + t_{data} + d_{SIFS} + d_{ACK}] \\ &= d_{RTS} + d_{CTS} + \left(\frac{1}{\mu}\right) + d_{ACK} + 3d_{SIFS}. \end{aligned} \quad (3.14)$$

As shown in Figure 3.1, we have

$$\begin{aligned} E[t_{c,i}] &= E\left[\sum_{j=1}^Y (d_{DIFS} + t_{b,j} + d_c)\right] + d_{DIFS} + E[t_{b,Y+1}] + E[t_r] \\ &= E[Y]d_c + (E[Y] + 1)(E[t_{b,i}] + d_{DIFS}) + E[t_r]. \end{aligned}$$

Applying (3.9), (3.13) and (3.14), $E[t_{c,i}]$ is obtained by

$$\begin{aligned} E[t_{c,i}] &= \left[\frac{\alpha\gamma p(\gamma - 1) + e^{\alpha\gamma p} - 1}{\gamma(1 + \alpha p - \alpha\gamma p)}\right] \left[d_c + d_{DIFS} + \frac{(1 - \gamma)e^{-\alpha\gamma p}}{1 - (1 - \gamma)e^{-\alpha\gamma p}}\right] \\ &+ \left[\frac{(1 - \gamma)e^{-\alpha\gamma p}}{1 - (1 - \gamma)e^{-\alpha\gamma p}}\right] + d_{DIFS} + d_{RTS} + d_{CTS} + \left(\frac{1}{\mu}\right) + d_{ACK} + 3d_{SIFS}. \end{aligned} \quad (3.15)$$

Derivation of $E[X]$ Consider the $t_{c,i}$ period. During $t_{c,i}$, there exists exactly one successful RTS transmission. Let $P_{succ,x}$ be the probability that a successful RTS transmission is made by n_x , and we have

$$\begin{aligned} P_{succ,x} &= \Pr[n_x \text{ sends the RTS} | N_t = 1, n_x \text{ is performing DCF}] \\ &= \frac{\Pr[n_x \text{ sends the RTS}, N_t = 1, n_x \text{ is performing DCF}]}{\Pr[N_t = 1, n_x \text{ is performing DCF}]} \\ &= \frac{\sum_{m=1}^{\infty} \Pr[n_x \text{ sends the RTS} | N_a = m, n_x \text{ is performing DCF}] \Pr[N_a|x = m]}{\Pr[N_t|x = 1]} \end{aligned}$$

Applying (3.8) and (3.12), we have

$$\begin{aligned} P_{succ,x} &= \frac{\sum_{m=1}^{\infty} \left(\frac{1}{m}\right) \binom{m}{1} \gamma(1-\gamma)^{m-1} \left[\frac{(\alpha p)^{m-1} e^{-\alpha p}}{(m-1)!}\right]}{\gamma e^{-\alpha p} (1 + \alpha p - \alpha \gamma p)} \\ &= \frac{1}{1 + \alpha p - \alpha \gamma p}. \end{aligned} \quad (3.16)$$

During t_a , in the $X + 1$ contentions, and n_x contends successfully in the $X + 1$ st contention, we have

$$\Pr[X = k] = (1 - P_{succ,x})^k P_{succ,x} = \left[\frac{\alpha p(1-\gamma)}{1 + \alpha p - \alpha \gamma p}\right]^k \left(\frac{1}{1 + \alpha p - \alpha \gamma p}\right),$$

and

$$E[X] = \sum_{k=0}^{\infty} k \Pr[X = k] = \alpha p - \alpha \gamma p. \quad (3.17)$$

With the derivations for $E[t_{c,i}]$ and $E[X]$, we have

$$\begin{aligned} E[t_a] &= \left\{ \left[\frac{\alpha \gamma p(\gamma - 1) + e^{\alpha \gamma p} - 1}{\gamma(1 + \alpha p - \alpha \gamma p)} \right] \left[d_c + d_{DIFS} + \frac{(1-\gamma)e^{-\alpha \gamma p}}{1 - (1-\gamma)e^{-\alpha \gamma p}} \right] \right. \\ &\quad \left. + \left[\frac{(1-\gamma)e^{-\alpha \gamma p}}{1 - (1-\gamma)e^{-\alpha \gamma p}} \right] + d_{DIFS} + d_{RTS} + d_{CTS} + \left(\frac{1}{\mu}\right) + d_{ACK} + 3d_{SIFS} \right\} (1 + \alpha p - \alpha \gamma p). \end{aligned} \quad (3.18)$$

Then from (3.4) and (3.18), P_{ps} is expressed as

$$\begin{aligned} P_{ps} &= A\theta \left/ \left\{ \left[\frac{\alpha \gamma p(\gamma - 1) + e^{\alpha \gamma p} - 1}{\gamma(1 + \alpha p - \alpha \gamma p)} \right] \left(d_c + d_{DIFS} + \frac{(1-\gamma)e^{-\alpha \gamma p}}{1 - (1-\gamma)e^{-\alpha \gamma p}} \right) + \frac{(1-\gamma)e^{-\alpha \gamma p}}{1 - (1-\gamma)e^{-\alpha \gamma p}} \right. \right. \\ &\quad \left. \left. + d_{DIFS} + d_{RTS} + d_{CTS} + \left(\frac{1}{\mu}\right) + d_{ACK} + 3d_{SIFS} \right\} (1 + \alpha p - \alpha \gamma p) + A\theta \right\}. \end{aligned} \quad (3.19)$$

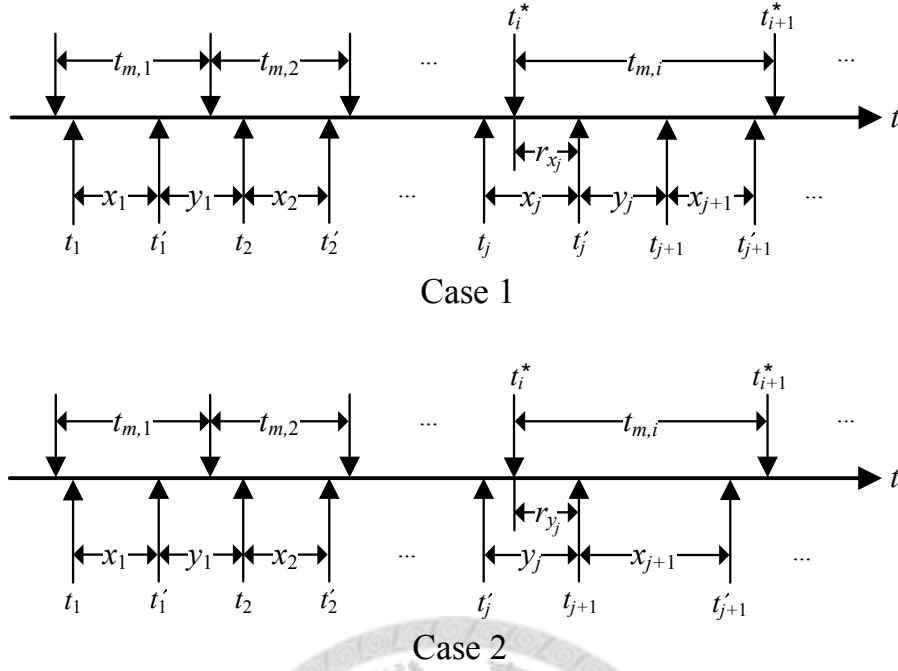


Figure 3.2: The timing diagram for MPs, receiving and non-receiving periods for the M2M gateway

3.2.2 Derivation of Outdated Monitoring Probability P_{om}

As we defined previously, P_{om} is the probability that no valid sensing report is received by the M2M gateway during an MP. Let $\overline{P_{om}}$ be the probability that at least one valid sensing report is received during an MP. Then, we have

$$P_{om} = 1 - \overline{P_{om}}. \quad (3.20)$$

To obtain P_{om} , we derive $\overline{P_{om}}$ as follows. In Figure 3.2, the M2M gateway alternates between the receiving periods (during which a successful sensing report is being transmitted) and the non-receiving periods (during which no sensing report is being transmitted). Define x_j and y_j as the j th receiving period and non-receiving period ($j \geq 1$), respectively. We assume that the receiving periods and non-receiving periods are in sequence

$x_1, y_1, x_2, y_2, \dots, x_j, y_j, x_{j+1}, y_{j+1}, \dots$. Suppose that the j th sensing report is transmitted by n_1 , and the receiving period x_j starts at t_j (when n_1 sends the RTS frame) and ends at t'_j (when n_1 receives the ACK), i.e., $x_j = t'_j - t_j$. In Chapter 3.2, the receiving period for a successful sensing report includes the signaling overheads (i.e., d_{RTS} , d_{CTS} and d_{ACK}), the time to transmit the data frame (i.e., t_{data}), and the period between two frames (i.e., d_{SIFS}). Let $D = d_{RTS} + d_{CTS} + d_{ACK} + 3d_{SIFS}$, then we have

$$x_j = t_{data} + D. \quad (3.21)$$

Since t_{data} is geometrically distributed with mean $1/\mu$, we have

$$\Pr[x_j = n] = \begin{cases} 0, & \text{if } 0 \leq n \leq D; \\ \mu(1-\mu)^{n-D-1}, & \text{if } n > D, \end{cases} \quad (3.22)$$

and

$$E[x_j] = \sum_{n=D+1}^{\infty} n \Pr[x_j = n] = \left(\frac{1}{\mu}\right) + D. \quad (3.23)$$

Suppose that the j th non-receiving period y_j starts at t'_j and ends at t_{j+1} (i.e., $y_j = t_{j+1} - t'_j$). During y_j , there is no successful sensing report transmission. In other words, in every slot of y_j , either no M2M node transmits data, or a collision occurs. In the DCF, a node (either an M2M node or the M2M gateway) determines the wireless medium idle by sensing the wireless medium for d_{DIFS} , and waits for the d_c period to determine that a collision occurs. In other words, when $y_j \geq d_{DIFS}$, and a collision occurs, a node takes $d_c + d_{DIFS}$ to resume the contention for the wireless medium. Therefore, there are at most

$g(y_j) = \left\lfloor \frac{y_j - d_{DIFS}}{d_{DIFS} + d_c} \right\rfloor$ collisions in y_j . Let Z_j be the number of collisions occurring in y_j , where $0 \leq Z_j \leq g(y_j)$. According to the law of total probability, we derive $\Pr[y_j = n]$ by conditioning on $Z_j = k$. Then we have

$$\Pr[y_j = n] = \begin{cases} 0, & \text{if } 0 < n < d_{DIFS}; \\ \sum_{k=0}^{g(n)} \Pr[y_j = n | Z_j = k] \Pr[Z_j = k], & \text{if } n \geq d_{DIFS}. \end{cases} \quad (3.24)$$

During y_j , a time slot is idle if $N_t = 0$. Let I_j be the number of idle time slots in y_j .

Under the condition $Z_j = k$, $y_j = I_j + k(d_{DIFS} + d_c) + d_{DIFS}$. Then we have

$$\begin{aligned} \Pr[y_j = n | Z_j = k] &= \Pr[I_j + k(d_{DIFS} + d_c) + d_{DIFS} = n | Z_j = k] \\ &= \Pr[I_j = n - (k+1)d_{DIFS} - kd_c | Z_j = k] \\ &= \binom{n - (k+1)d_{DIFS} - kd_c + k}{k} \Pr[N_t = 0]^{n - (k+1)d_{DIFS} - kd_c} \\ &\quad \Pr[N_t \geq 1]^{k+1}. \end{aligned} \quad (3.25)$$

Similar to the derivation of $\Pr[Y = k]$, $\Pr[Z_j = k]$ can be obtained by

$$\Pr[Z_j = k] = \Pr[N_t \geq 2 | N_t \geq 1]^k \Pr[N_t = 1 | N_t \geq 1]. \quad (3.26)$$

Applying (3.25) and (3.26), we can rewrite (3.24) for $n \geq d_{DIFS}$ as

$$\begin{aligned} \Pr[y_j = n] &= \sum_{k=0}^{g(n)} \binom{n - (k+1)d_{DIFS} - kd_c + k}{k} \Pr[N_t = 0]^{n - (k+1)d_{DIFS} - kd_c} \\ &\quad (1 - \Pr[N_t = 0] - \Pr[N_t = 1])^k \Pr[N_t = 1]. \end{aligned} \quad (3.27)$$

Since the random backoff is geometrically distributed with mean $1/\gamma$, we have

$$\begin{aligned}
\Pr[N_t = 0] &= \sum_{m=0}^{\infty} \Pr[N_t = 0 | N_a = m] \Pr[N_a = m] \\
&= \sum_{m=0}^{\infty} (1 - \gamma)^m \left[\frac{(\alpha p)^m e^{-\alpha p}}{m!} \right] \\
&= e^{-\alpha \gamma p},
\end{aligned} \tag{3.28}$$

and similarly,

$$\Pr[N_t = 1] = \alpha \gamma p e^{-\alpha \gamma p}. \tag{3.29}$$

Apply (3.28) and (3.29) into (3.27), and we have

$$\begin{aligned}
\Pr[y_j = n] &= \sum_{k=0}^{\lfloor \frac{n-d_{DIFS}}{d_{DIFS}+d_c} \rfloor} \binom{n - (k+1)d_{DIFS} - kd_c + k}{k} \\
&\quad (e^{-\alpha \gamma p})^{n-(k+1)d_{DIFS}-kd_c} [1 - e^{-\alpha \gamma p}(1 + \alpha \gamma p)]^k \alpha \gamma p e^{-\alpha \gamma p}. \\
&= \sum_{k=0}^{g(n)} \binom{n - (k+1)d_{DIFS} - kd_c + k}{k} \alpha \gamma p e^{-\alpha \gamma p [n-(k+1)d_{DIFS}-kd_c+1]} \\
&\quad [1 - e^{-\alpha \gamma p}(1 + \alpha \gamma p)]^k,
\end{aligned} \tag{3.30}$$

where $g(n) = \lfloor \frac{n-d_{DIFS}}{d_{DIFS}+d_c} \rfloor$.

Let $t_{m,i}$ denote the i th MP of the SR, where $i \geq 1$. We assume $t_{m,i}$ is geometrically distributed with mean θ , i.e., in a slot, the current MP ends with probability $1/\theta$. Consider the i th MP in Figure 3.2. Suppose that the i th MP starts at t_i^* and ends at t_{i+1}^* . Since the M2M gateway alternatively changes between the receiving and non-receiving periods, the

i th MP starts either in a receiving or non-receiving period. We consider the following two cases to derive $\overline{P_{om}}$:

Case 1 The i th MP starts during x_j (i.e., $t_j < t_i^* \leq t'_j$; see Figure 3.2 Case 1). In this case, the j th sensing report must be invalid since the transmission for the sensing report starts during the $i - 1$ st MP but the sensing report is received during the i th MP. Let r_{x_j} be the period between t_i^* and t'_j . From the residual life theorem [50], we have

$$\Pr[r_{x_j} = n] = \begin{cases} \frac{\mu}{1 + \mu D}, & \text{if } 0 < n \leq D; \\ \frac{\mu(1 - \mu)^{n-d-1}}{1 + \mu D}, & \text{if } n > D. \end{cases} \quad (3.31)$$

If the next sensing report (i.e., the $j + 1$ st sensing report) is successfully received by the M2M gateway during the i th MP (i.e., $t_i^* + r_{x_j} + y_j + x_{j+1} \leq t_i^* + t_{m,i}$), the i th MP is a real-time MP.

Case 2 The i th MP starts during y_j (i.e., $t'_j < t_i^* \leq t_{j+1}$; see Figure 3.2 Case 2). Let r_{y_j} be the period between t_i^* and t_{j+1} . From the residual life theorem, we have

$$\Pr[r_{y_j} = n] = \begin{cases} \frac{1}{E[y_j]}, & \text{if } 0 < n < d_{DFS}; \\ \frac{1 - \sum_{i=0}^{n-1} \Pr[y_j = i]}{E[y_j]}, & \text{if } n \geq d_{DFS}, \end{cases} \quad (3.32)$$

where $\Pr[y_j = n]$ and $E[y_j]$ can be obtained from (3.30). The i th MP is a real-time MP if the $j + 1$ st sensing report is successfully received by the M2M gateway in the i th MP (i.e., $t_i^* + r_{y_j} + x_{j+1} \leq t_i^* + t_{m,i}$).

Then, $\overline{P_{om}}$ can be calculated as

$$\begin{aligned}\overline{P_{om}} &= \Pr[M_i \geq 1 | \text{Case 1}] \Pr[\text{Case 1}] + \Pr[M_i \geq 1 | \text{Case 2}] \Pr[\text{Case 2}] \\ &= \Pr[r_{x_j} + y_j + x_{j+1} \leq t_{m,i}] \Pr[\text{Case 1}] + \Pr[r_{y_j} + x_{j+1} \leq t_{m,i}] \Pr[\text{Case 2}],\end{aligned}\tag{3.33}$$

where M_i is the number of valid sensing reports received by the M2M gateway during the i th MP. Since the M2M gateway alternates between x_j and y_j , from alternating renewal process [40, 45], we have

$$\Pr[\text{Case 1}] = \frac{E[x_j]}{E[x_j] + E[y_j]}, \text{ and } \Pr[\text{Case 2}] = \frac{E[y_j]}{E[x_j] + E[y_j]}.$$

By the law of total probability, we have

$$\begin{aligned}\Pr[r_{x_j} + y_j + x_{j+1} \leq t_{m,i}] &= \sum_{n=D+1}^{\infty} \sum_{m=d_{DFS}}^{\infty} \sum_{t=1}^{\infty} \Pr[t + m + n \leq t_{m,i} | r_{x_j} = t, y_j = m, x_{j+1} = n] \\ &\Pr[r_{x_j} = t | y_j = m, x_{j+1} = n] \Pr[y_j = m | x_{j+1} = n] \Pr[x_{j+1} = n].\end{aligned}$$

Since r_{x_j} , y_j , x_{j+1} and $t_{m,i}$ are assumed to be independent, we have

$$\begin{aligned}\Pr[r_{x_j} + y_j + x_{j+1} \leq t_{m,i}] &= \sum_{n=D+1}^{\infty} \sum_{m=d_{DFS}}^{\infty} \sum_{t=1}^{\infty} \Pr[t + m + n \leq t_{m,i}] \Pr[r_{x_j} = t] \\ &\Pr[y_j = m] \Pr[x_{j+1} = n].\end{aligned}\tag{3.34}$$

Similar to the derivation of $\Pr[r_{x_j} + y_j + x_{j+1} \leq t_{m,i}]$, we have

$$\Pr[r_{y_j} + x_{j+1} \leq t_{m,i}] = \sum_{n=D+1}^{\infty} \sum_{m=1}^{\infty} \sum_{k=m+n}^{\infty} \Pr[t + m + n \leq t_{m,i}] \Pr[r_{y_j} = m] \Pr[x_{j+1} = n]. \quad (3.35)$$

Applying (3.34), (3.35) and the geometric distribution assumption of $t_{m,i}$ with mean θ ,

we rewrite (3.33) as

$$\begin{aligned} \overline{P_{om}} &= \left(\frac{E[x_j]}{E[x_j] + E[y_j]} \right) \sum_{n=D+1}^{\infty} \sum_{m=d_{DFS}}^{\infty} \sum_{t=1}^{\infty} \sum_{k=t+m+n}^{\infty} \left(\frac{1}{\theta} \right) \left(1 - \frac{1}{\theta} \right)^{k-1} \\ &\quad \Pr[r_{x_j} = t] \Pr[y_j = m] \Pr[x_{j+1} = n] \\ &+ \left(\frac{E[y_j]}{E[x_j] + E[y_j]} \right) \sum_{n=D+1}^{\infty} \sum_{m=1}^{\infty} \sum_{k=m+n}^{\infty} \left(\frac{1}{\theta} \right) \left(1 - \frac{1}{\theta} \right)^{k-1} \Pr[r_{y_j} = m] \Pr[x_{j+1} = n]. \end{aligned} \quad (3.36)$$

Applying (3.22), (3.23) and (3.31) into (3.36), then based on (3.20), we can calculate P_{om}

as follows:

$$\begin{aligned} P_{om} &= 1 - \left(\frac{1/\mu + D}{1/\mu + D + E[y_j]} \right) \sum_{m=d_{DFS}}^{\infty} \left[\frac{u\theta}{1 + \mu(\theta - 1)} \right] \left(\frac{\theta - 1}{\theta} \right)^{D+m} \Pr[y_j = m] \\ &\quad - \left(\frac{E[y_j]}{1/\mu + D + E[y_j]} \right) \sum_{m=1}^{\infty} \left[\frac{\mu(\theta - 1)^{m-1} \theta^{3-m}}{(2\theta - 1)(1 + \mu\theta - \mu)} \right] \Pr[r_{y_j} = m], \end{aligned} \quad (3.37)$$

where $\Pr[y_j = m]$, $E[y_j]$ and $\Pr[r_{y_j} = m]$ can be obtained from (3.30) and (3.32).

3.2.3 Calculation of p

This subsection derives the probability p defined in Chapter 3.2. In Figure 3.1, the M2M gateway receives the $X + 1$ st sensing report from n_x during n_x 's active period, and the $X + 1$ st sensing report is transmitted by the n_x . The period t_{DCF} when n_x performs the DCF for contention of the wireless medium is between t_0 and $t_0 + t_{D,x} + t_{c,X+1} - t_r$ (i.e., $t_0 + t_a - t_r$; when n_x sends the RTS frame for the $X + 1$ st sensing report), and $t_{DCF} = t_a - t_r$. From alternating renewal process, we have

$$p = \frac{E[t_{DCF}]}{E[t_a] + E[t_s]} = \frac{E[t_a - t_r]}{E[t_a] + E[t_s]} = \frac{E[t_a] - E[t_r]}{E[t_a] + E[t_s]}, \quad (3.38)$$

where $E[t_s]$, $E[t_r]$ and $E[t_a]$ can be obtained from (3.4), (3.14) and (3.18), respectively.

We calculate (3.38) by using the following **Iterative Algorithm**:

Step 1 Select an initial value for p .

Step 2 Compute $E[t_s]$ and $E[t_r]$ by using (3.4) and (3.14), respectively.

Step 3 Compute $E[t_a]$ by using (3.18).

Step 4 $p_{old} \leftarrow p$.

Step 5 Compute p by using (3.38).

Step 6 If $|p - p_{old}| \geq \delta p$, then go to Step 3. Otherwise, exit. Note that δ is a pre-defined value set to 10^{-7} to ensure the convergence of p .

We show that the iterative algorithm for the p probability converges to a unique solution. The proof consists of two parts. In the first part, we prove the existence and uniqueness of a solution for p (i.e., equation (3.38)) in Lemma 1. In the second part, we let p_n denote the n th iteration obtained from the iterative algorithm. In Lemma 2, we show that $\{p_n\}$ is a monotonic sequence. From Lemma 1 and Lemma 2, the $\{p_n\}$ sequence is obviously bounded and always converges to the unique equilibrium point.

Lemma 1. *The solution of p is unique in $(0, 1)$.*

Proof. Let

$$g(p) = \frac{E[t_a] - E[t_r]}{E[t_a] + E[t_s]}.$$

Then from (8), (18), and (22), we have

$$g(p) = \left\{ 1 + \frac{\gamma\mu(D+\theta)}{\mu[1+(d_{DIFS}+d_c)(e^{\alpha\gamma p}-1)]+\gamma[1+\mu(d_{DIFS}-1)+(1-r)\alpha p(1+D\mu-d_c\mu)]} \right\}^{-1},$$

where $D = d_{RTS} + d_{CTS} + d_{ACK} + 3d_{SIFS}$.

Let $h(p) = p - g(p)$. Then we have

$$h(0) = -g(0) = -\frac{\gamma + \mu + \gamma\mu(d_{DIFS} - 1)}{\gamma + \mu + \gamma\mu(d_{DIFS} - 1 + D + \theta)} < 0,$$

and

$$\begin{aligned} h(1) &= 1 - g(1) \\ &= 1 - \left\{ 1 + \frac{\gamma\mu(D+\theta)}{\mu[1+(d_{DIFS}+d_c)(e^{\alpha\gamma}-1)]+\gamma[1+\mu(d_{DIFS}-1)+(1-r)\alpha(1+D\mu-d_c\mu)]} \right\}^{-1} \\ &> 0 \end{aligned}$$

because $d_{DIFS} > 1$ and $D > d_c$. From the mean value theorem [41], there is a solution for the equation (3.38). In other words, there exists a p^* in $(0, 1)$ such that $h(p^*) = 0$. Let $g^{(1)}(p)$ and $g^{(2)}(p)$ denote the first and second derivatives of $g(p)$. Because $g^{(1)}(p) > 0$ and $g^{(2)}(p) < 0$, we conclude that $g(p)$ is a concave function and an increasing function

in $[0, 1]$. Since $g(0) > 0$ and $g(1) < 0$, there is only one point in $(0, 1)$ at which $p = g(p)$, and the solution of $p = g(p)$ is unique in $(0, 1)$. \square

Lemma 2. *The sequence $\{p_n\}$ is a monotonic sequence.*

Proof. Let p_n denote the n th iteration obtained from the iterative algorithm. Then we have $p_{n+1} = g(p_n)$ and

$$\begin{aligned}
 p_{n+1} - p_n &= g(p_n) - g(p_{n-1}) \\
 &= g^{(1)}(r_n)(p_n - p_{n-1}) \\
 &= \dots \\
 &= [g^{(1)}(r_n)] \cdots [g^{(1)}(r_1)](p_1 - p_0),
 \end{aligned} \tag{3.39}$$

where r_1, r_2, \dots, r_n are the intermediate values from the mean value theorem. For any value in $(0, 1]$, from (3.39), $g^{(1)}(p) > 0$. If $p_1 > p_0$, p_n is nondecreasing. Otherwise (i.e., $p_1 \leq p_0$), p_n is nonincreasing. Therefore the sequence $\{p_n\}$ is a monotonic sequence and is obviously bounded. \square

3.2.4 Simulation Model Validation

We conduct simulations for BDSA and TDSA of EDR by applying the discrete-event driven approach that has been widely used in wireless networks studies [35, 54]. The simulation experiments are validated against the analytical models for BDSA. Table 3.2.4 shows the comparison between simulation and analytical results (parameter setups will be elaborated later). Most errors are within 3%. Hence the simulation results are consistent

Table 3.1: Validation of the simulation and analysis results for P_{ps} and P_{om}

$\eta = 10^{-4}\mu, \gamma = (1/25.6)\mu, \theta = 100/\mu, \text{ and } A = 4$					
λ (unit: $10^{-3}\mu$)	1	2	3	4	5
P_{ps} (Analytical)	0.934173	0.931780	0.929219	0.926474	0.923526
P_{ps} (Simulation)	0.934444	0.931780	0.929151	0.925986	0.922562
Error	0.000290%	0%	0.000073%	0.000527%	0.001044%
P_{om} (Analytical)	0.303117	0.178502	0.127007	0.098552	0.080756
P_{om} (Simulation)	0.311270	0.182287	0.130291	0.101198	0.081496
Error	2.619269%	2.076396%	2.520511%	2.614182%	0.908870%

with the analytical results.

3.3 Performance Evaluation

The performance of BDSA and TDSA is evaluated in terms of P_{ps} and P_{om} . We set $d_{SIFS} = 1$ and $d_{DIFS} = 2$. The transmission rate is 11 Mbps, and we calculate the corresponding transmission times for the RTS, CTS and ACK frames. For convenience, we set $d_{RTS} = d_{CTS} = d_{ACK} = 2$, and then $d_{CTS-timeout} = 3$. The input parameters λ , η , γ and θ are normalized by μ . For example, if the expected data transmission time is $1/\mu = 100$ time slots, $\eta = 4 \times 10^{-5}\mu$ means that the expected SR residence time of an M2M node is 2.5×10^6 time slots.

The effects of input parameters, α , T_{sleep} , K , and θ on P_{ps} and P_{om} are studied in the following subsections.

3.3.1 Effects of α

In this section, we use the factor $\alpha = \lambda/\eta$ to study the effects of the node mobility behavior on the P_{ps} and the P_{om} performances. A smaller α implies a lower traffic intensity in

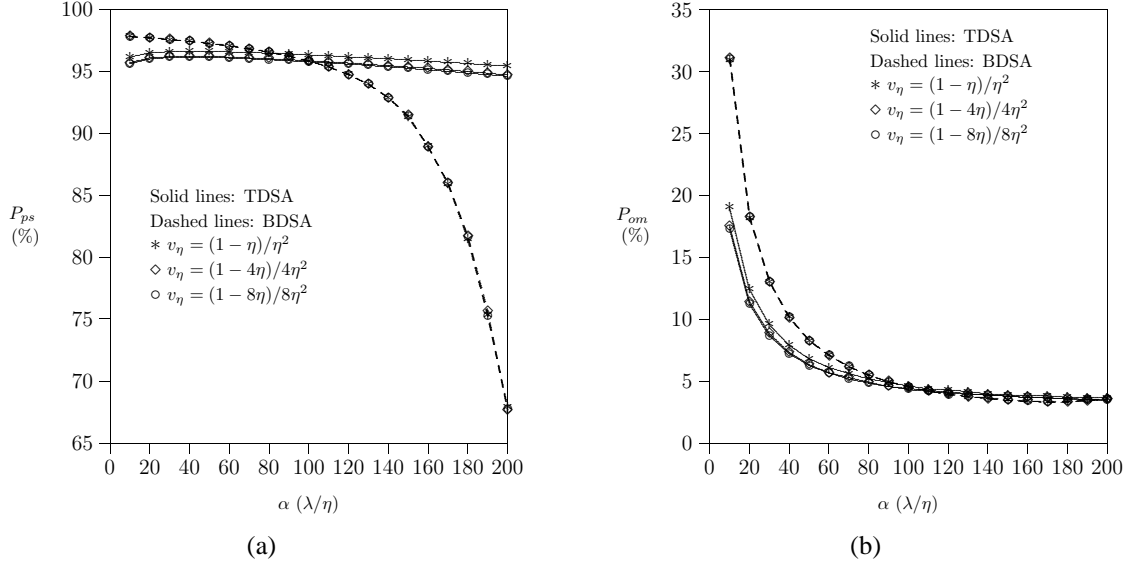


Figure 3.3: Effects of α on P_{ps} and P_{om} for BDSA and TDSA, where $\gamma = (1/6.4)\mu$, $\theta = 100/\mu$, $A = 4$, $K = 4$, and $\rho = 0.4$.

an SR. On the other hand, a larger α implies a higher traffic intensity in an SR, i.e., more M2M nodes are in an SR.

In Figure 3.3 (a), we observe that P_{ps} for both BDSA and TDSA decrease as α increases. P_{ps} for BDSA drops more quickly than that for TDSA when α is larger. In most cases, TDSA outperforms BDSA in terms of P_{ps} . In TDSA, the M2M nodes' sleeping periods are adjusted based on the measurement Γ_j , which reduces the possibility for M2M nodes to transmit the sensing reports simultaneously and prolongs an M2M node to stay in the sleep mode.

Figure 3.3 (b) shows that P_{om} for both BDSA and TDSA decrease as α increases. P_{om} drops quickly when $\alpha \leq 30$, but drops slowly when $\alpha > 30$. This phenomenon implies that the number of M2M nodes in an SR becomes saturated at $\alpha = 30$, and more M2M nodes (i.e., $\alpha > 30$) in an SR do not improve the performance of P_{om} significantly. Furthermore, by adjusting the sleeping period dynamically, TDSA outperforms BDSA in

terms of the P_{om} performance in most of the cases.

In Figure 3.3, we also study how the variance of the SR residence time affects P_{ps} and P_{om} for BDSA and TDSA by assuming that the SR residence time has a negative binomial distribution with mean $1/\eta$ and variance $v_\eta = (1 - r\eta)/r\eta^2$. As v_η increases, it is more likely to observe an M2M node with both short and long SR residence time in an SR, which illustrates a more “dynamic” node mobility behavior. We observe that P_{ps} and P_{om} for BDSA and TDSA are both insensitive to v_η . For the effects of the variance of the inter M2M node arrival time (where we assume that the inter-arrival period has a negative binomial distribution with mean $1/\lambda$ and variance $v_\lambda = (1 - r\lambda)/r\lambda^2$), we observe the same results so the figures are not included in this paper.

3.3.2 Effects of T_{sleep}

In Figure 3.4, we investigate the effects of T_{sleep} (i.e., the A setup) on P_{ps} and P_{om} , where we consider low and high density M2M nodes in an SR, i.e., $\alpha = \lambda/\eta = 10$ and $\alpha = 200$.

A larger A implies that an M2M node stays in the sleep mode longer, potentially saving more power. Therefore, as shown in Figure 3.4 (a), we observe that P_{ps} for BDSA and TDSA both increase when A increases.

In Figure 3.4 (b), for BDSA, when α is small (e.g., $\alpha = 10$), P_{om} increases as A increases. When α is large (e.g., $\alpha = 200$), P_{om} decreases as A increases. Smaller α implies less M2M nodes in an SR, and less chance for collision. In this case, in BDSA, a smaller A setup increases the chance for M2M nodes to successfully send the sensing

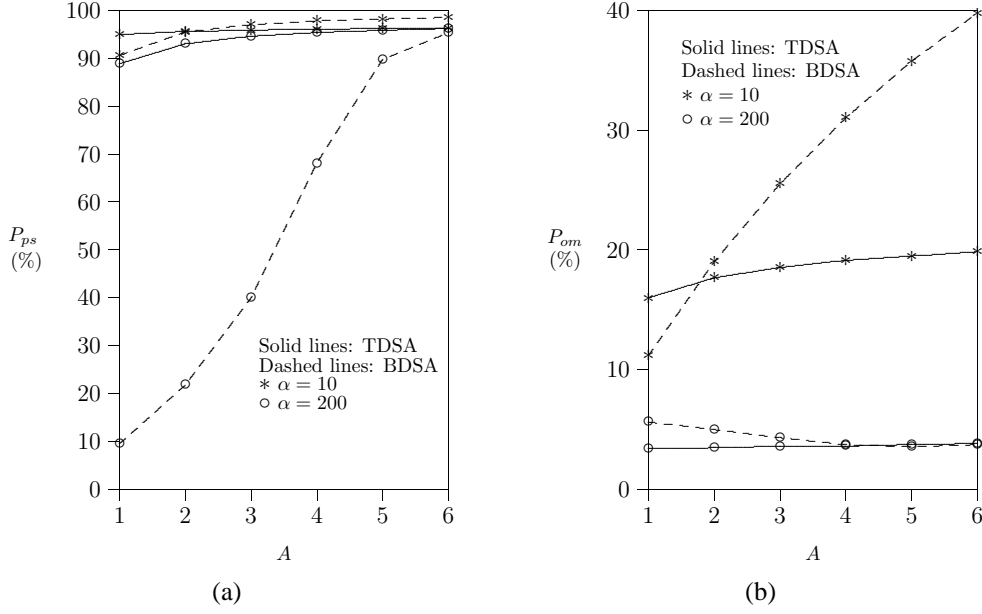


Figure 3.4: Effects of A on P_{ps} and P_{om} for BDSA and TDSA, where $\gamma = (1/6.4)\mu$, $\theta = 100/\mu$, $K = 4$, and $\rho = 0.4$.

reports to the M2M gateway. On the other hand, when α is large, more M2M nodes are in an SR, resulting into higher collision possibility. The collision can be reduced by setting up longer sleeping periods. Hence, we observe the phenomena of P_{om} for BDSA.

In Figure 3.4 (b), we observe P_{om} for TDSA slightly increases as A increases. The effects of configuration of A on P_{om} for TDSA are minor. To conclude, for BDSA, we prefer to set A larger (smaller) for better P_{ps} and P_{om} when α is larger (smaller). For TDSA, the larger A setup is suggested to have the better P_{ps} and P_{om} .

3.3.3 Effects of K

In E^2 DCR, the periodicity of broadcasting is controlled by the K value. A larger K value implies that it is more likely that the sleeping period adjusting process is slow. When there are too many outdated MPs for SR_j , in E^2 DCR, the M2M gateway can decrease K to speed up the sleeping period adjustment process. In this section, we investigate the

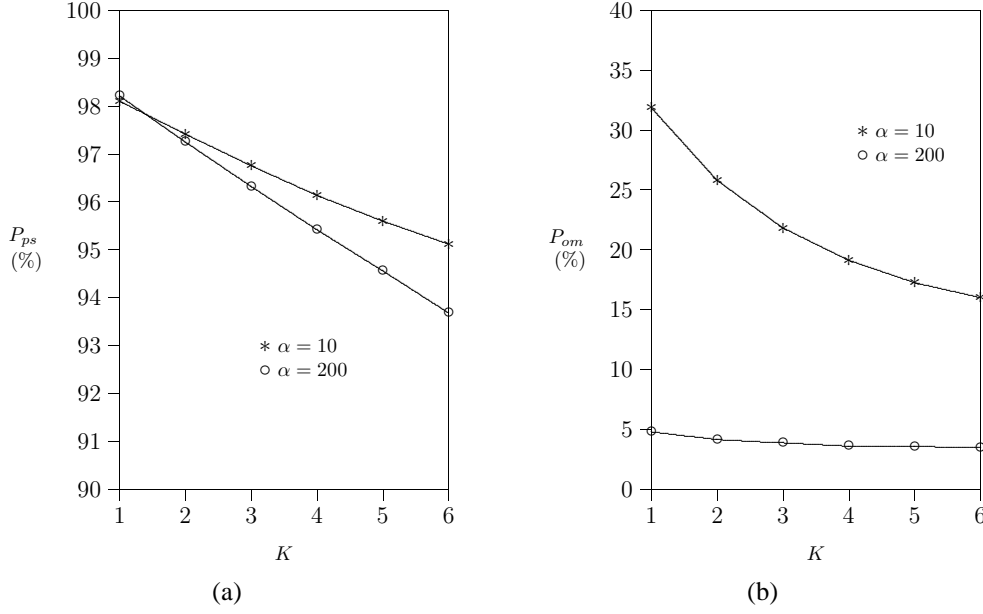


Figure 3.5: Effects of K on P_{ps} and P_{om} for TDSA, where $\gamma = (1/6.4)\mu$, $\theta = 100/\mu$, $A = 4$, and $\rho = 0.4$.

effects of K on P_{ps} and P_{om} for TDSA in Figure 3.5, where we consider low and high density M2M nodes in an SR, i.e., $\alpha = \lambda/\eta = 10$ and $\alpha = 200$.

As shown in Figure 3.5 (a), for smaller K , the M2M node adjusts its sleeping period more frequently to match the monitoring condition in an SR, and higher P_{ps} is observed (i.e., lower energy consumption).

In Figure 3.5 (b), as K increases, P_{om} decreases. This phenomenon is explained as follows. In TDSA, with smaller K , the M2M node has less chance to reconfigure the sleep timer T_{sleep} . Because the sleep timer T_{sleep} is reset to θ when an M2M node enters an SR, before receiving the first notification message from the M2M gateway node, the M2M node sends a sensing report for every θ .

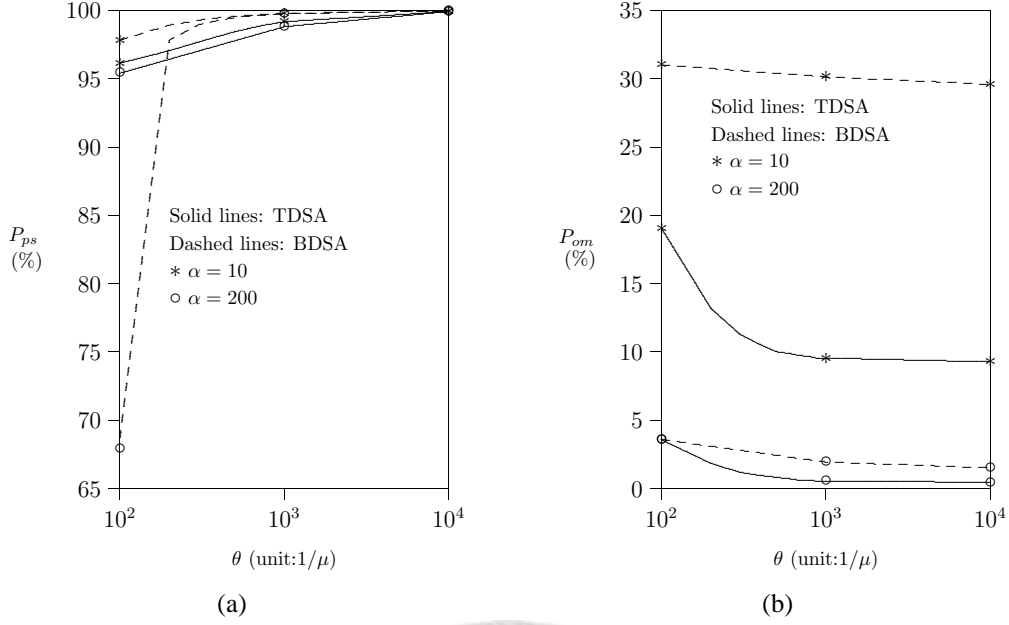


Figure 3.6: Effects of α on P_{ps} and P_{om} for BDSA and TDSA, where $\gamma = (1/6.4)\mu$, $A = 4$, $K = 4$, and $\rho = 0.4$.

3.3.4 Effects of θ

The MP interval θ is the time constraint for the real-time monitoring applications. Different kinds of real-time monitoring applications have different requirements for θ . The configuration of θ varies according to the applications, and θ can be obtained from the measured data. To study the effects of θ on the P_{ps} and P_{om} performance, we run more simulation experiments by considering low and high density M2M nodes in an SR, i.e., $\alpha = \lambda/\eta = 10$ and $\alpha = 200$ as shown in Figure 3.6. We observe the following phenomena:

In Figure 3.6 (a), P_{ps} for both BDSA and TDSA increases as θ increases. A larger θ implies that the length of an MP is larger. In BDSA or TDSA, the M2M node adjusts the sleeping period based on the θ value. Therefore, as θ increases, an M2M node has a longer sleep timer setup, resulting in more power saving. In Figure 3.6 (b), P_{om} for both BDSA

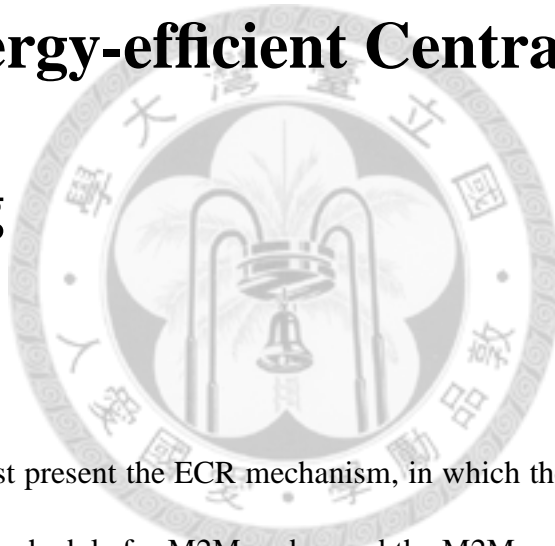
and TDSA decreases as θ increases. A larger θ also implies that the time constraint for the sensing report becomes looser. Thus, it is less likely for the M2M gateway to receive an invalid sensing report. We observe the similar performance trends for $\alpha = 10$ and $\alpha = 200$.





Chapter 4

ECR: Energy-efficient Centralized Reporting



In this chapter, we first present the ECR mechanism, in which 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 ECR are described in Chapter 4.1. Then, in Chapter 4.2, the determination of a transmission schedule for each M2M node during a cycle is formulated as an optimization problem, which is called the *energy minimization* problem. The energy minimization problem is formulated to an ILP problem and proved NP-hard. To approximate the solution of the ILP formulation, we propose a greedy algorithm with polynomial time complexity. Furthermore, we compare the performance for the ILP and greedy approaches in terms of time cost and sizes of solution sets in Chapter 4.3.

4.1 Mechanism Operations

The ECR mechanism consists of two parts: the M2M gateway and M2M node parts, whose details are given as follows.

M2M Gateway Part: Similar to EDR, in ECR, the M2M gateway also maintains the set $\mathbb{D} = \{D(1), D(2), \dots, D(M)\}$ for each SR, and the timers T_1, T_2, \dots, T_M to denote the MPs for each type, respectively. Without loss of generality, we assume $\theta_1 \leq \theta_2 \leq \dots \leq \theta_M$, and

$$\theta_i = a_i \theta_1 \quad \forall i \in \{1, 2, \dots, M\}$$

where a_i are integers and $a_1 = 1$. Let θ_1 denote the length of a transmission unit. Define a cycle with length $L = \text{lcm}(a_1, a_2, \dots, a_M) \theta_1$, where $\text{lcm}(a_1, a_2, \dots, a_M)$ is the least common multiple of a_1, a_2, \dots, a_M . 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 j th 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. More specifically, the M2M gateway executes a greedy scheduling algorithm (to be elaborated later) at the beginning of 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

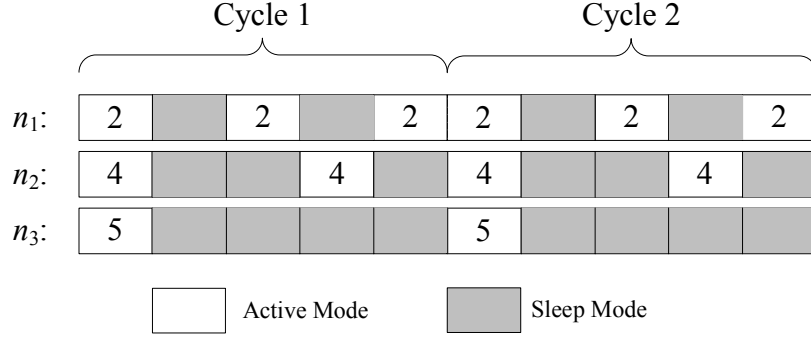


Figure 4.1: An example for a transmission schedule for the M2M nodes n_1, n_2, n_3

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 j th transmission unit within a cycle. Figure 4.1 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 4.1, n_1 will transmit the type 2 sensed data in the first, the third, and the fifth transmission unit within a cycle. The M2M node n_2 transmits the type 4 sensed data in the first and the fourth transmission unit within a cycle. The M2M node n_3 transmits the type 5 sensed data in the first transmission unit within a cycle. Otherwise, these three M2M nodes stay in the sleep mode.

4.2 Energy Minimization for Centralized Scheduling

In the energy minimization problem, the M2M gateway 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. Chapter 4.2.2 provides the ILP formulation for the energy minimization problem. Chapter 4.2.3 proves that the energy minimization problem is NP-hard. Chapter 4.2.4 proposes a greedy scheduling algorithm to approximate the solution of the ILP.

4.2.1 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:

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

- $x_{i,j,k}$: the binary variable defined by

$$x_{i,j,k} = \begin{cases} 1, & \text{if } n_k \text{ transmits type } i \text{ sensed data at} \\ & \text{the } j\text{th transmission unit within a cycle;} \\ 0, & \text{otherwise.} \end{cases} \quad (4.1)$$

4.2.2 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.

The objective function of the ILP formulation is given as

$$\min \sum_{k=1}^N \sum_{j=1}^L \sum_{i=1}^M x_{i,j,k} \quad (4.2)$$

subject to:

$$\sum_{k:1 \in \mathbb{S}_k} x_{1,j,k} \geq 1, \forall j \in \{1, 2, \dots, L\} \quad (4.3.1)$$

$$\sum_{k:2 \in \mathbb{S}_k} \sum_{l=0}^{a_2-1} x_{2,j+l,k} \geq 1, \forall j \in \{1, 1+a_2, 1+2a_2, \dots, L-a_2+1\} \quad (4.3.2)$$

$$\sum_{k:3 \in \mathbb{S}_k} \sum_{l=0}^{a_3-1} x_{3,j+l,k} \geq 1, \forall j \in \{1, 1+a_3, 1+2a_3, \dots, L-a_3+1\} \quad (4.3.3)$$

⋮

$$\sum_{k:M \in \mathbb{S}_k} \sum_{l=0}^{a_M-1} x_{M,j+l,k} \geq 1, \forall j \in \{1, 1+a_M, 1+2a_M, \dots, L-a_M+1\} \quad (4.3.N)$$

$$x_{i,j,k} \in \{0, 1\} \quad (4.4)$$

In the objective function (4.2), $\sum_{j=1}^L \sum_{i=1}^M x_{i,j,k}$ means the number of transmission units assigned to the M2M node n_k in a cycle, and $\sum_{k=1}^N \sum_{j=1}^L \sum_{i=1}^M x_{i,j,k}$ is the total transmission units assigned to all M2M nodes during a cycle. The objective function (4.2), 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 (4.3.1), (4.3.2), (4.3.3), ..., (4.3.N) ensure that at least one valid sensed data is transmitted to the M2M gateway for each type of MP. For example, the constraint (4.3.1) 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.

4.2.3 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 [24]. The set-covering problem is described as follows. Let \mathbb{F} denote a set of finite elements and $\mathbb{H} = \{\mathbb{H}_1, \mathbb{H}_2, \dots, \mathbb{H}_x\}$, where \mathbb{H}_i ($1 \leq i \leq x$) is a set containing finite elements in \mathbb{F} . The union of $\mathbb{H}_1, \mathbb{H}_2, \dots, \mathbb{H}_x$ forms \mathbb{F} . From [24], 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{N}, \mathbb{S}) consisting of a finite set of

M2M nodes $\mathbb{N} = \{n_1, n_2, \dots, n_N\}$ and a sensing set $\mathbb{S} = \{1, 2, 3, \dots, M\}$. We replace every $n_k \in \mathbb{N}$ with its sensing set \mathbb{S}_k and let the set $\tilde{\mathbb{S}} = \{\mathbb{S}_1, \mathbb{S}_2, \dots, \mathbb{S}_N\}$, where a sensing set \mathbb{S}_k ($1 \leq k \leq N$) contains the types of sensing data that the M2M node n_k can collect. Thus, $\mathbb{S} = \bigcup_{k=1}^N \mathbb{S}_k$. 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. \square

4.2.4 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 $\tilde{\mathbb{S}} = \{\mathbb{S}_1, \mathbb{S}_2, \dots, \mathbb{S}_N\}$. 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 $\tilde{\mathbb{S}} = \{\mathbb{S}_1, \mathbb{S}_2, \dots, \mathbb{S}_N\}$
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 N **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**

4.3 Performance on ILP and Greedy Approach

We use a binary matrix with N rows and M columns to denote the problem instance mentioned in Chapter 4.2.3, where N and M 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 here and will discuss it later. 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 $1 \leq k \leq N$.

Considering each combination of the values of N and M , we randomly generate 100 independent binary matrices (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 N_{ILP} and N_{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

(a) 100 M2M nodes				
N	100			
M	10	20	50	100
N_{ILP}	1.87	2.00	2.98	3.76
N_{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

(b) 200 M2M nodes				
N	200			
M	10	20	50	100
N_{ILP}	1.88	2.00	2.97	3.18
N_{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

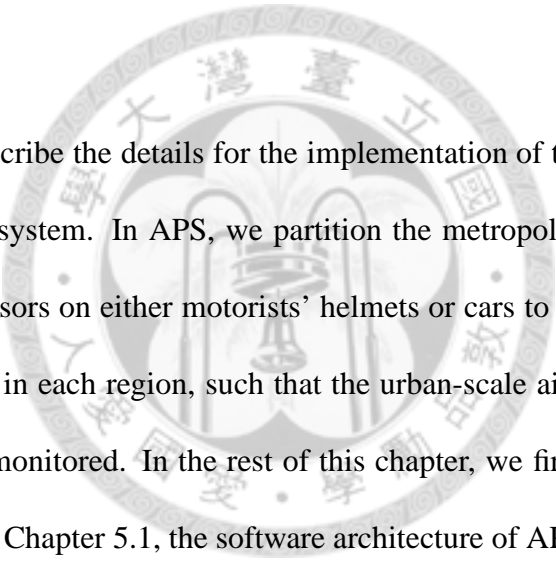
Table 4.1: Comparison between MATLAB ILP solver and greedy algorithm

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 N and M . Furthermore, the elapsed time for solving the ILP drastically increases with M . For example, the time for MATLAB ILP solver to solve the problem instance with $N = 200$ and $M = 100$ 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.



Chapter 5

APS: Air Pollution Sensing System



In this chapter, we describe the details for the implementation of the distributed *Air Pollution Sensing* (APS) system. In APS, we partition the metropolitan area into multiple regions and install sensors on either motorists' helmets or cars to collect air quality data (e.g., pollutant levels) in each region, such that the urban-scale air quality can be effectively and efficiently monitored. In the rest of this chapter, we first describe the system architecture of APS in Chapter 5.1, the software architecture of APS in Chapter 5.2.

5.1 System Architecture

In the APS system, the urban environment is divided into several SRs by latitude and longitude. Each region is associated with a unique Region ID (RID). Figure 5.1 illustrates an example where there are 16 regions with RIDs R_1, R_2, \dots, R_{16} ranging from latitude 25.03° to 25.034° at an interval of 0.001° and from longitude 121.53° to 121.534° at an interval of 0.001° .

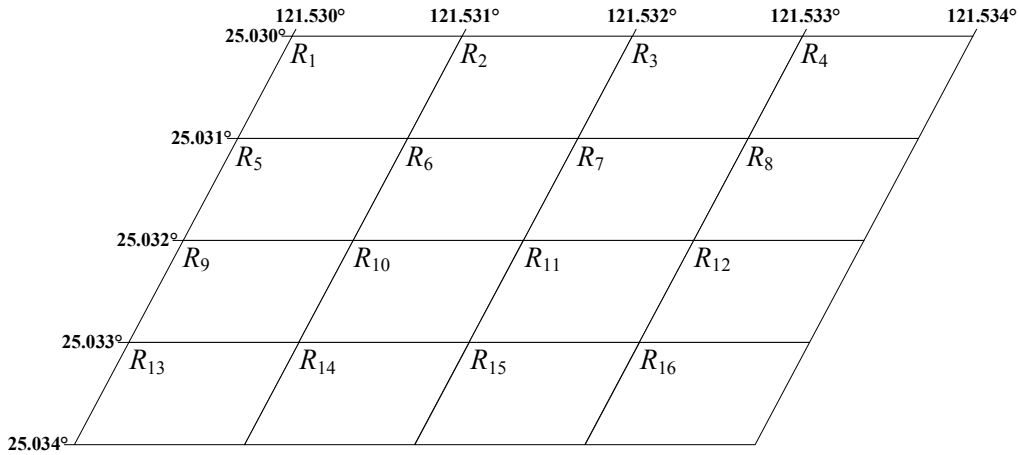


Figure 5.1: An example of sensing regions

Since air quality is time-sensitive, the collected air quality data are only valid for a certain period. In APS, the time is divided into multiple MPs. As mentioned previously, during an MP, a sensing report transmission is valid if the sensing report is generated and received in the same MP. Otherwise (i.e., the sensing report is generated before the sensing report is received), the sensing report transmission is invalid. The detailed definition of data validity is given in (2.3).

Figure 5.2 illustrates the system architecture of the APS system comprising three kinds of nodes: the *sensing device* (see Figure 5.2 (a)), the *back-end server* (see Figure 5.2 (b)) and *user device* (see Figure 5.2 (c)), whose functionalities will be elaborated in the following subsections.

5.1.1 The Sensing Device

In the APS system, the sensing device is defined as the device equipped with the air quality sensor, the GPS receiver and the wireless transmitter. There are two kinds of deployments for the sensing devices.

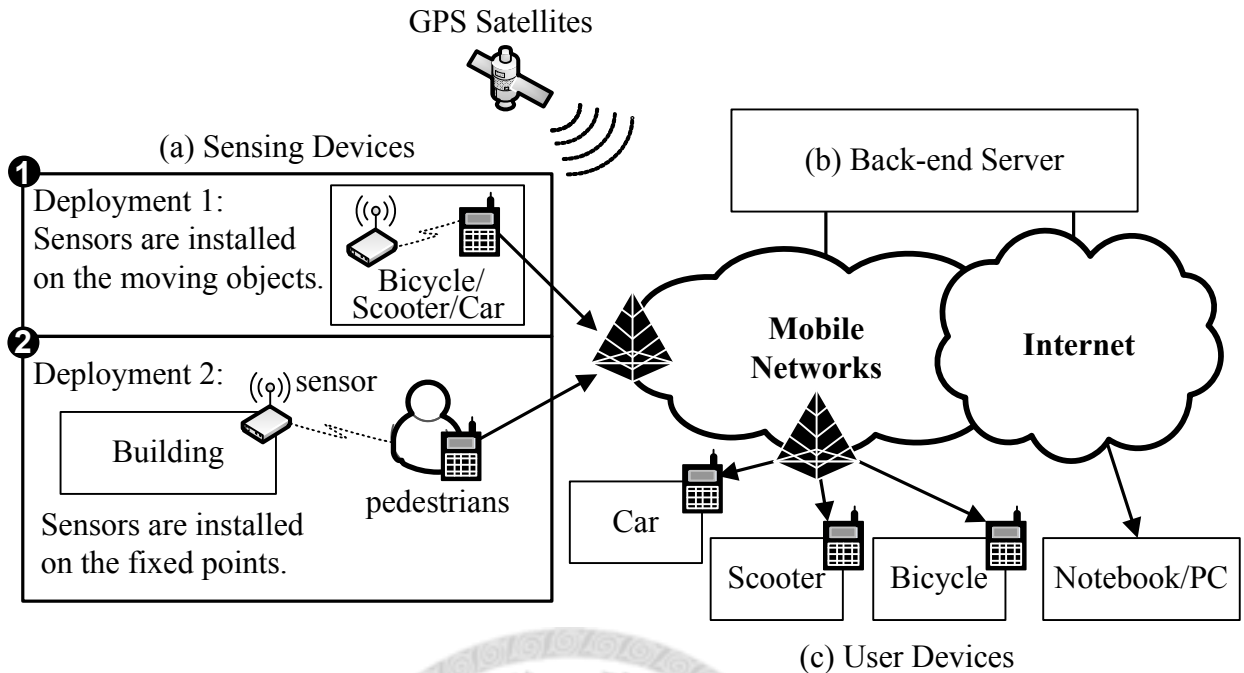


Figure 5.2: The system architecture for the APS system

Deployment 1: See Figure 5.2 (1). The sensing devices are installed on mobile nodes (e.g., cars, scooters, bicycles, or motorists' helmets). With Deployment 1, the APS system can provide broad-coverage of monitoring area since the system can monitor any sensing region where the mobile vehicles reside in the urban environment.

Deployment 2: See Figure 5.2 (2). The sensing devices are installed on fixed points in the urban environment (e.g., buildings and trees).

After collecting the air quality data (by the air quality sensor) and receiving the GPS and time information (by the GPS receiver), the sensing device generates a sensing report (containing air quality data, GPS and time information), and transmits the sensing report to the back-end server through its wireless transmitter.

In our prototype of APS, the sensing device is a netbook (i.e., ASUS Eee PC T91 [2]) equipped with a CO₂ sensor (i.e., CO₂ Meter K-30 Probe [4]), a GPS receiver (i.e., Glob-



Figure 5.3: A bicycle equipped with a netbook, a GPS receiver, a CO₂ sensor, and a 3G modem

alSat ND-100 [5]) and a 3G modem (i.e., ZTE MF 627 [11]). As shown in Figure 5.3, the netbook is installed on the bicycle, where the CO₂ sensor, the GPS receiver and the 3G modem are all connected to the netbook through the USB ports.

To save the energy during the sensing report transmissions, we implement the EDR mechanism in the sensing device. In our design, the sensing device is in one of the two modes: active mode and sleep mode. In the active mode, the sensing device can generate the sensing report and then transmit the sensing report to the back-end server. In the sleep mode, the sensing device turns off the CO₂ sensor, the GPS receiver and the 3G modem for power saving. More details will be elaborated in Chapter 5.2.

5.1.2 The Back-end Server

The back-end server is equipped with the transmission modules (i.e., the Ethernet network card and the Nokia Card Phone in our prototype) to support the data transmission through the Internet Protocol (IP) on Internet and Short Message Service (SMS) on mobile networks, respectively. The back-end server is assigned an IP address and a phone number

(i.e., an MSISDN). The major tasks of the back-end server are to collect the sensing reports from the sensing devices, to control the sensing report transmissions for the sensing devices and to provide services to the user devices.

The APS system produces air quality information for the urban environment, and provides users with several kinds of services. In our prototype, we implement the monitoring service and the navigation service as examples. The monitoring service provides user devices with maps layered with the air quality information. The user can check the air quality of some specific regions in the urban environment. By utilizing the monitoring service, we design the “good” navigation service for users in the urban environment. Here, “good” means cost-effective, time-saving, and healthful. The good navigation service can dynamically calculate healthful routes for users (especially for bikers and motorcyclers) based on air quality information. In other words, with the good navigation, users not only save time and fuel to arrive their destinations, but via healthful routes. The details of these three services will be elaborated in Chapter 5.2.1, and the interactions among the three kinds of nodes will be elaborated in Chapter 5.3.

Furthermore, we implement the EDR mechanism in the back-end server to adjust the sleeping periods of sensing devices. The details will be elaborated in Chapter 5.2.1.

5.1.3 The User Device

In the APS system, the user devices are the mobile devices (e.g., personal computer, notebook and smartphone). All services are web-based. The users access these services through the web browsers (e.g., Internet Explorer, Firefox and Google Chrome) on het-

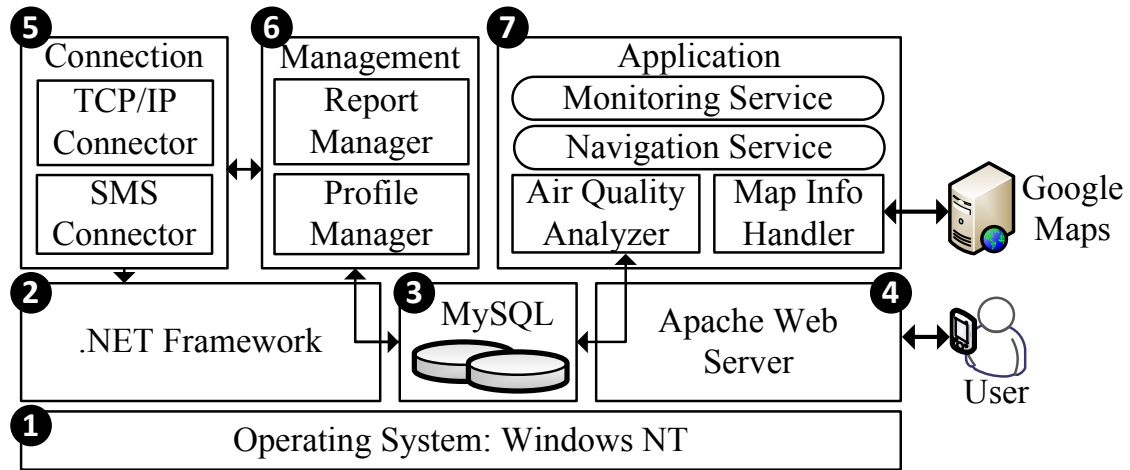


Figure 5.4: The software architecture for the back-end server

erogeneous mobile devices. The software implementation overhead of APS on the user devices is considered light.

5.2 Software Architecture

In this section, we describe the software architecture for the back-end server and the sensing device.

5.2.1 The Software Architecture of The Back-end Server

Figure 5.4 illustrates the software architecture of the back-end server, which is build up on the Windows NTTM operating system (see Figure 5.4 (1)). We develop the software of the back-end server by using .NET framework [9] (see Figure 5.4 (2)), MySQL database [8] (see Figure 5.4 (3)) and Apache web server [1] (see Figure 5.4 (4)). The software of the back-end server consists of three major components: the connection component, the management component and the application component, whose details are given below.

The Connection Component

As shown in Figure 5.4 (5), the connection component is responsible for data transmission between the sensing device and the back-end server, and between the user device and the back-end server. The connection component has two connectors: the Transmission Control Protocol/Internet Protocol (TCP/IP) connector and the Short Message Service (SMS) connector. The TCP/IP connector invokes the `Socket.Send()` and `Socket.Receive()` methods of .NET framework to send and receive data via TCP/IP. For efficient data transmission, we implement the Internet Group Management Protocol (IGMP) in the TCP/IP connector to support multicasting on Internet. In the SMS connector, we use the Internet and SMS integration platform iSMS¹ [43] to implement Short Message Service Cell Broadcast (SMSCB) [12] to broadcast the SMS to multiple devices in a specified area.

The Management Component

As shown in Figure 5.4 (6), the management component consists of the profile manager and the report manager. We design two classes `ProfileManager` and `ReportManager` for these two components, both are responsible for querying and instructing commands to MySQL database by invoking the `MySQLConnection.Open()` method and the `MySQLCommand()` constructor of .NET framework. MySQL database stores two kinds of data, the user profile and air quality information. The user profile contains the user ID and password as shown in Figure 5.5. There are three kinds of air quality data, i.e., CO₂,

¹iSMS is a platform that integrates IP networks with the SMS in mobile telephone systems. Through iSMS, the back-end server can send and receive the SMS.


ID	Password
vicfu	12345
hcchen	54321
	

Figure 5.5: An example of the user profile stored in the database

Air Quality Data						
RID	Type	Value	Type	Value	Type	Value
R_1	CO ₂	568	PM10	51	PM2.5	23
R_2	CO ₂	450	PM10	43	PM2.5	25
⋮						
R_{10}	CO ₂	787	PM10	56	PM2.5	20
⋮						

Figure 5.6: An example of the air quality data stored in the database

Particulate Matter 10 (PM10) and PM2.5. The profile manager provides the `Access()`, `Modify()`, `Create()` and `Delete()` member methods to access, modify, create and delete the user profile stored in MySQL database, respectively. Figure 5.6 shows an example for the air quality information.

In the report manager, we implement the EDR mechanism. For the sensing region R_j , the report manager maintains the T_j timer, the C_j counter, the index i , the $I_j(i)$ variable and the ratio $\Gamma_j(i)$. The threshold of the T_j timer is θ . The expiration of the T_j timer denotes the end of an MP. The C_j counter counts the number of valid sensing reports from (R_j) received by the back-end server during an MP. In the beginning of an MP, C_j is set to 0. The index i is used to indicate that the i th MP. In the end of the i th MP, the binary variable $I_j(i)$ is maintained to indicate whether the back-end server receives any valid sensing report during the i th MP. $I_j(i) = 1$ denotes that the back-end server does not receive any valid sensing report, and $I_j(i) = 0$ denotes that the back-end server receives

at least one valid sensing report.

In the end of every MP, the report manager computes the ratio $\Gamma_j(i)$, where $\Gamma_j(0) = 0$.

$$\Gamma_j(i) = \begin{cases} \Gamma_j(i-1) + I_j(i) & \text{if } i \bmod K \neq 0 \\ \frac{\Gamma_j(i-1) + I_j(i)}{K} & \text{otherwise (i.e., } i \bmod K = 0) \end{cases} \quad (5.1)$$

In the end of every K MPs, the ratio $\Gamma_j(i)$ is transmitted by the back-end server to the sensing devices in R_j to dynamically adjust their sleeping periods through the adjustment message (where the sleeping period for a sensing device is the period when the sensing device stays in the sleep mode). The adjustment message transmission will be elaborated later in this section.

In the report manager, based on equation (2.3), we implement the `CheckValidity()` member method (which takes the time information carried in a sensing report as input parameter) to check whether the sensing report is valid or not. The `IdentifyRID()` member method determines the RID for a sensing report based on the GPS information carried in the sensing report. The `SaveAirQualityData()` member method (where the air quality data and RID are input parameters) saves the air quality information into MySQL database.

Figure 5.7 shows the pseudo code for EDR mechanism, which is implemented as the `EDR()` member method. In EDR, we define two events:

- **E1**: the back-end server receives a sensing report for R_j .
- **E2**: the T_j timer expires.

When the **E1** event occurs, the report manager executes Lines 7-13 to save the air quality data of the valid sensing report to MySQL database by invoking `CheckValidity()`,

IdentifyRID() and SaveAirQualityData(). When the *E2* event occurs (i.e., the end of an MP), the report manger executes Lines 15-25 to update the Γ_j ratio. At the end of every *K* MPs, an adjustment message (containing Γ_j , the latitude and longitude of the center, the length and width of R_j) is transmitted by the TCP/IP connector or the SMS connector through invoking the SendAdjustment() member method (see Line 21). Based on the carried information in the adjustment message, the sensing device in R_j can determine whether it should adjust its sleeping period or not. The details of the sleeping period adjustment for the sensing device will be elaborated in Chapter 5.2.2.

The Application Component

As shown in Figure 5.4 (7), the application component consists of the map info handler, the air quality analyzer, the monitoring service and the good navigation service, all run on the Apache web server and are implemented by using the object-oriented scripting languages, JavaScript and Hypertext Preprocessor (PHP). In the map info handler, we provide APIs to obtain the map-information from Google Maps [7] such as map, longitude and latitude, available route and road intersection. The map info handler takes longitude and latitude as the input parameters of the GMap2.setCenter() method in Google Maps API [6] to get the corresponding map. Note that Google Maps can be replaced by other existing map services such as Microsoft Bing Maps [3].

In the air quality analyzer, we design the QueryAirQuality() method to query the air quality data stored in the MySQL database through the Apache web server by using using the mysql_connect() method and the mysql_query() method in the PHP library. The QueryAirQuality() method takes the RID as its input parameter and re-

Pseudo Code

```

1  $i \leftarrow 1$ ;
2  $C_j \leftarrow 0$ ;
3  $T_j \leftarrow \theta$ ;
4 while an event occurs do
5     switch the event do
6         case E1
7             invoke CheckValidity() by taking time information as input;
8             if the sensing report is valid then
9                 invoke IdentifyRID() by taking GPS information as input;
10                invoke SaveAirQualityData() by taking air quality data and RID as
                inputs;
11                 $C_j \leftarrow C_j + 1$ ;
12            else
13                ignore the sensing report;
14        case E2
15            if  $C_j = 0$  then
16                 $I_j(i) \leftarrow 1$ ;
17            else
18                 $I_j(i) \leftarrow 0$ ;
19            if  $(i \bmod K) = 0$  then
20                 $\Gamma_j(i) \leftarrow \Gamma_j(i - 1) + I_j(i)/K$ ;
21                invoke SendAdjustment() to transmit the adjustment message;
22                 $\Gamma_j(i) \leftarrow 0$ ;
23            else
24                 $\Gamma_j(i) \leftarrow \Gamma_j(i) + I_j(i)$ ;
25             $i \leftarrow i + 1$ ;

```

Figure 5.7: The pseudo code of the EDR mechanism implemented in the report manager

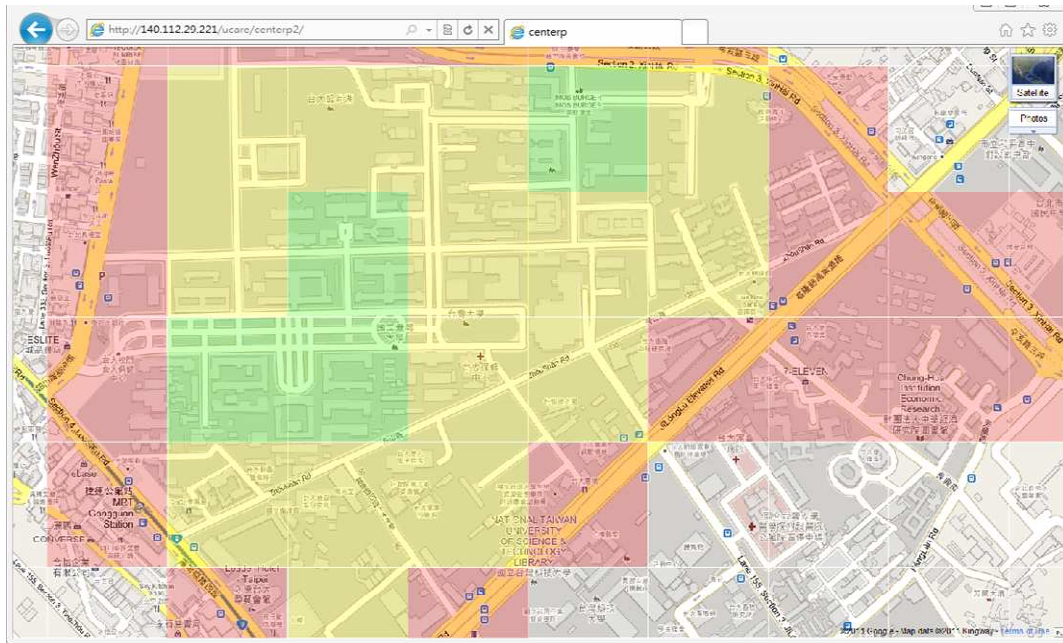


Figure 5.8: Monitoring service

turns the corresponding air quality data for that region. The `EvaluateAirQuality()` method is implemented to evaluate air quality for a sensing region and determine three degrees of air quality: good, moderate and unhealthy. The details of these two services are described as follows:

Monitoring Service: Through the map info handler, the back-end server invokes the `GMap2.addOverlay()` method of the Google Maps API to mark the air quality degree for each sensing region on Google Maps by different colors, where green, yellow and red colors denote good, moderate and unhealthy effects on health, respectively. When the air quality data for a sensing region stored in the MySQL database is changed, the color of the sensing region is changed to maintain the real-time air quality monitoring. In our prototype, we implement the monitoring service in the National Taiwan University (NTU) campus. The air quality information for the NTU campus is integrated with map information shown in Figure 5.8.

Navigation Service: In the urban environment, most air pollution comes from exhaust gas emitted by vehicles. We use air quality information to estimate traffic condition. To determine the air quality for a road, we divide the road into several road sections according to the intersections by invoking the `DivideRoadSection()` method. Figure 5.9 illustrates an example where a road is divided into two road sections (denoted by the dash-line rectangles), i.e., road sections a and b. The size of each road section depends on the length and width of the road. We implement the `EvaluateRoadSection()` method to assess air quality for a road section. Suppose that there are x sensing regions in a road section. The air quality of the road section is estimated by taking the mean of the air quality weighted values of the x sensing regions. The weights are the proportions of the area of the road section in each region. Take Figure 5.9 as an example. Road section b is covered by two regions, R_1 and R_2 , and 5% of road section b is in R_1 and 95% is in R_2 . Let V_1 and V_2 be the air quality values of R_1 and R_2 , respectively. Then, the air quality of road section b is $0.05 \cdot V_1 + 0.95 \cdot V_2$. We can calculate the air quality for a road by invoking the `EvaluateRoadSection()` method. We implement the `PlanRoute()` method by applying and modifying the existing route planning algorithms (such as [48]). The `PlanRoute()` method takes the air quality as one of the navigation factors to dynamically calculate a good route. The user benefits from faster, less congested and more healthful routes in the urban environment. Figure 5.10 illustrates an example of our navigation service in which the green arrow on the top right corner guides the moving direction for the mobile user.

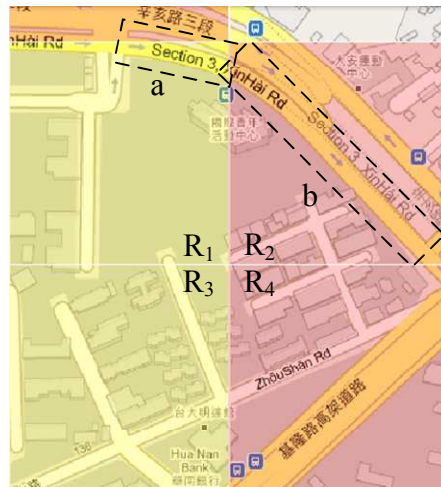


Figure 5.9: An example of road sections

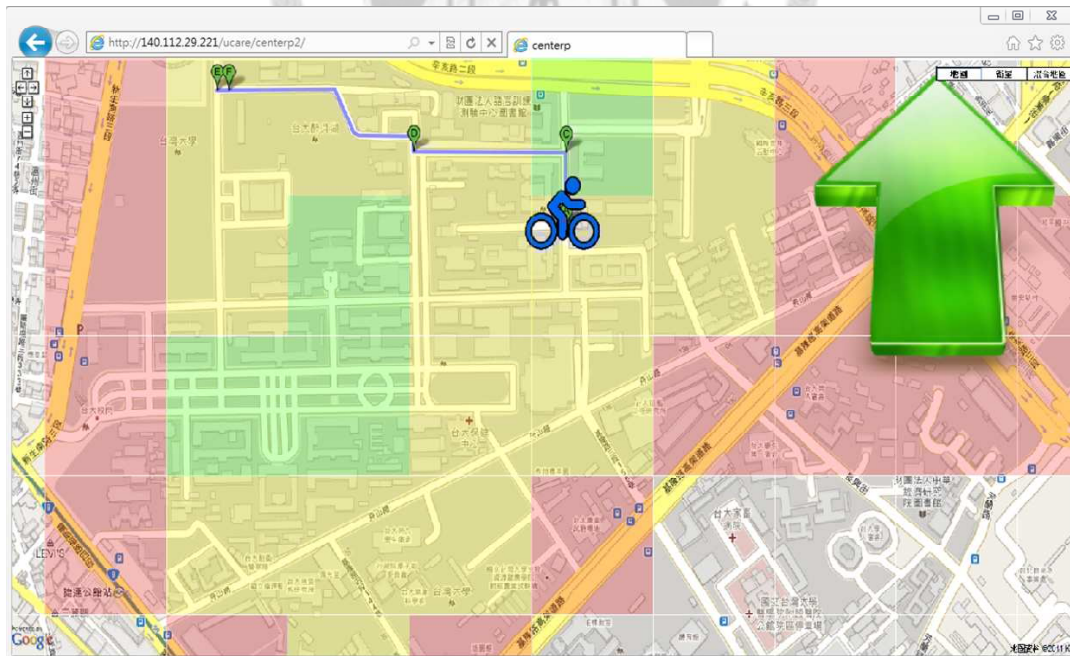


Figure 5.10: Navigation Service

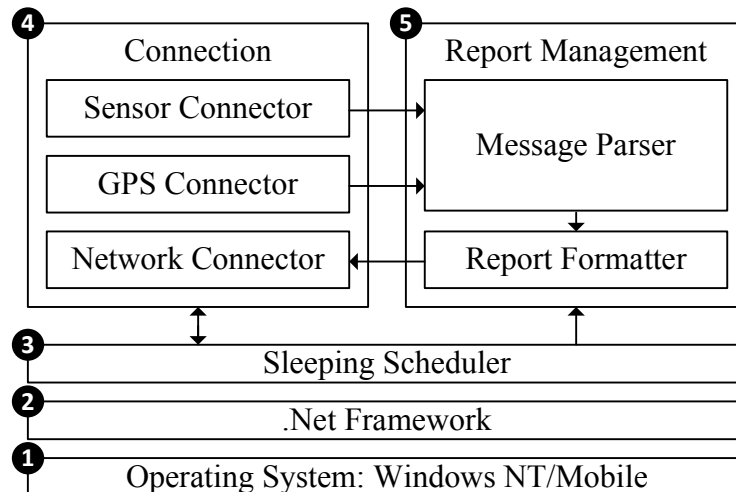


Figure 5.11: The software architecture for the sensing device

5.2.2 The Software Architecture of The Sensing Device

Figure 5.11 illustrates the software architecture of the sensing device, including the sleeping scheduler, the connection component and the report management component, whose details are given below.

The Connection Component

See Figure 5.11 (4). There are three connectors in this component, including the sensor connector, the GPS connector and the network connector. We utilize two classes `SerialPort` and `Socket` in the .Net framework to implement these three components. The sensor connector and the GPS connector are responsible for establishing the connection between the air quality sensor and the sensing device, and the connection between the GPS receiver and the sensing device, respectively. The sensing device can receive the air quality data and the GPS information through the sensor connector and the GPS connector by invoking the method `SerialPort.ReadLine()` in the .Net

framework, respectively. The network connector can establish the connection between the sensing device and the back-end server through the mobile networks by invoking the method `Socket.Connect()` in the .Net framework. Through the methods in the network connector, `Socket.Send()` and `Socket.Receive()` in the .Net framework, the sensing device can send/receive data to/from the back-end server.

The Sleeping Scheduler

See Figure 5.11 (3). In the sleeping scheduler, we implement the EDR mechanism to control the sleeping period for the sensing device. In other words, the sleeping scheduler determines when the sensing device should switch to the active mode. The sleeping scheduler maintains a sleep timer T_{sleep} for the sensing device. After transmitting a sensing report, the sensing device switches to the sleep mode, and the sleeping scheduler starts the T_{sleep} timer. When the T_{sleep} timer expires, the sensing device switches to the active mode.

The value of the T_{sleep} timer (i.e., the sleeping period of the sensing device) can be dynamically adjusted by the sleeping scheduler. Let Lat and Lng denote the latitude and longitude of the sensing device. When the sensing device receives an adjustment message for R_j (containing the latitude Lat_j and the longitude Lng_j of the center of R_j , the length L_j of R_j , and the width W_j of R_j), the sleeping scheduler uses the following equations to check whether the sensing device is in R_j .

$$Lat_j - 0.5 \cdot L_j \leq Lat \leq Lat_j + 0.5 \cdot L_j, \quad (5.2)$$

and

$$Lng_j - 0.5 \cdot W_j \leq Lng \leq Lng_j + 0.5 \cdot W_j. \quad (5.3)$$

If equation (5.2) or equation (5.3) does not hold (i.e., the sensing device is not in R_j), the sleeping scheduler ignores the received adjustment message. Otherwise (i.e, equations (5.2) and (5.3) both hold), the sleeping scheduler invokes the `AdjustPeriod()` method (which takes the ratio Γ_j in the adjustment message as input parameter) to adjust the T_{sleep} timer. Let $V(k)$ denote the k th configuration for the T_{sleep} timer. Initially, $V(1)$ is set to θ . $V(k + 1)$ is adjusted to

$$V(k + 1) = \begin{cases} V(k) + A\theta, & \text{if } \Gamma_j = 0 \\ V(k), & \text{if } \Gamma_j < \rho \\ \max\{V(k) - A\theta, \theta\}, & \text{otherwise (i.e., } \Gamma_j \geq \rho), \end{cases}$$

where ρ ($0 \leq \rho \leq 1$) and A are pre-defined thresholds.

The Report Management Component

See Figure 5.11 (5). The report management consists of the message parser and the report formatter. We design two classes for these two components which are `MessageParser` and `ReportFormatter`. The message parser parses the raw data received from the sensor connector and the GPS connector to get the air quality data (i.e., the CO_2 value) by invoking the `ParseSensorData()`, latitude and longitude information by invoking the `ParserGPSData()` member methods, respectively. Based on the parsed data,

the report formatter generates the sensing report by invoking the `GenerateReport()` member method. The sensing report includes two 64-bit floating points for latitude and longitude, a 64-bit floating point for the CO₂ value and a 64-bit integer for time. The total size of a sensing report is 256 bits. The report formatter invokes the `SendReport()` member method to instruct the network connector to transmit the sensing report.

5.3 Message Flows

In the APS system, we design four procedures, the sensing report transmission procedure (to transmit the sensing report from the sensing device to the back-end server), the sleeping period adjustment procedure (to adjust the sleeping period of the sensing device), the monitoring service procedure (to enable the user device to run the monitoring service), and the navigation procedure (to enable the user device to run the navigation service). This section describes the message flows for three procedures: the sensing report transmission procedure, the sleeping period adjustment procedure, and the navigation service procedure. The message flow for the monitoring service, whose details are omitted, is similar to that for the navigation service procedure.

5.3.1 The Sensing Report Transmission Procedure

Figure 5.12 illustrates the message flow for the sensing report transmission procedure including following steps.

Steps 1 and 2. In the sensing device, the sensor connector and the GPS connector receive

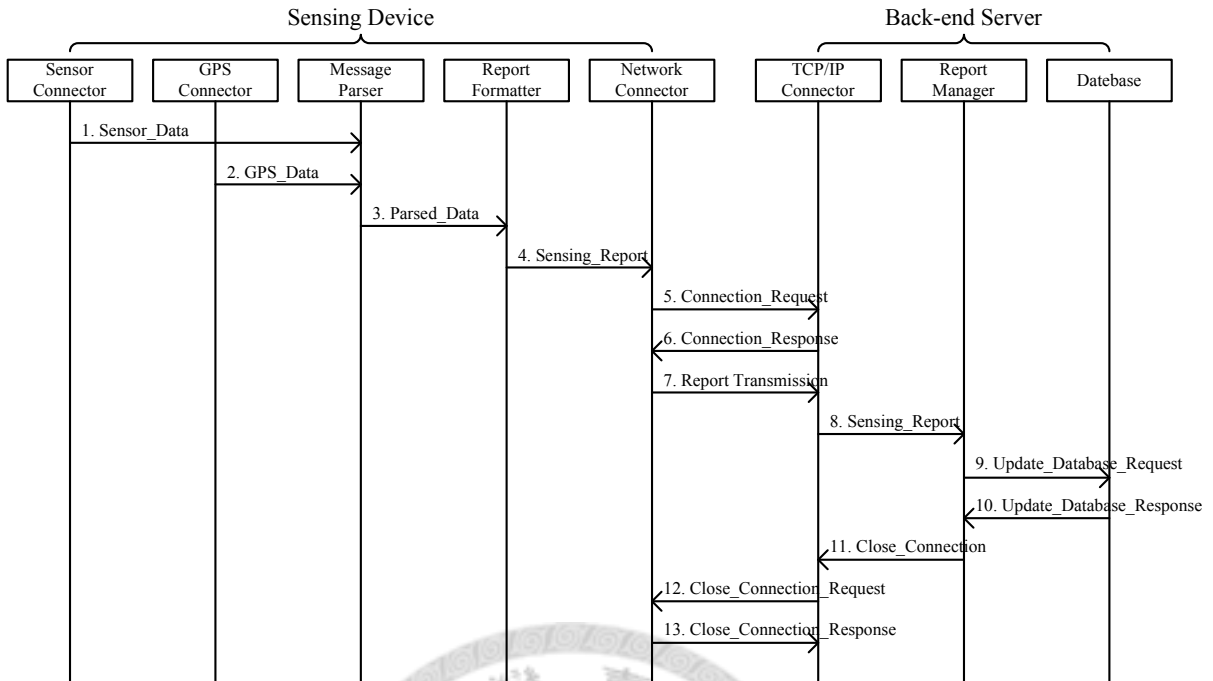


Figure 5.12: The message flow for the sensing report transmission procedure

the air quality data and GPS information (i.e., `Sensor_Data` from the air quality sensor and `GPS_Data` from the GPS receiver) by invoking `SerialPort.ReadLine()`, respectively. The sensor connector and the GPS connector pass their received data to the message parser.

Step 3. After receiving the `Sensor_Data` and `GPS_Data`, the message parser in the sensing device invokes the `ParseSensorData()` and `ParseGPSData()` member methods get the `Parsed_Data` (including the CO_2 value, latitude, longitude and time). Then, the message parser forwards the `Parsed_Data` to the report formatter.

Step 4. The report formatter in the sensing device invokes the `GenerateReport()` member method to generate the `Sensing_Report` based on the `Parsed_Data` and passes the `Sensing_Report` to the network connector by calling the `SendReport()`

member method.

Steps 5 and 6. Upon receipt of the `Sensing_Report`, the network connector in the sensing device invokes the `Socket.Connect()` method to exchange the `Connection_Request` message and the `Connection_Response` message with the TCP/IP connector to establish the connection between the sensing device and the back-end server.

Step 7. The network connector starts to transmit the `Sensing_Report` to the TCP/IP connector by invoking the `Socket.Send()` method.

Step 8. Upon receipt of the `Sensing_Report`, the TCP/IP connector in the back-end server forwards the `Sensing_Report` to the report manager.

Steps 9 and 10. The report manager invokes the `CheckValidity()` member method to check the validity of the `Sensing_Report` (whose details were mentioned in Chapter 5.2.1). If the `Sensing_Report` is invalid, it will be ignored by the report manager. Otherwise, the report manager invokes the `IdentifyRID()` member method to determine the RID for the `Sensing_Report`, and sends the `Update_Database_Request` message (containing the information carried in the `Sensing_Report`) to the database to update the corresponding entries through the `SaveAirQualityData()` member method. After that, the database responds the report manager with the `Update_Database_Response` message.

Step 11. Upon receipt of the `Update_Database_Response`, the report manager sends the `Close_Connection` message to the TCP/IP connector.

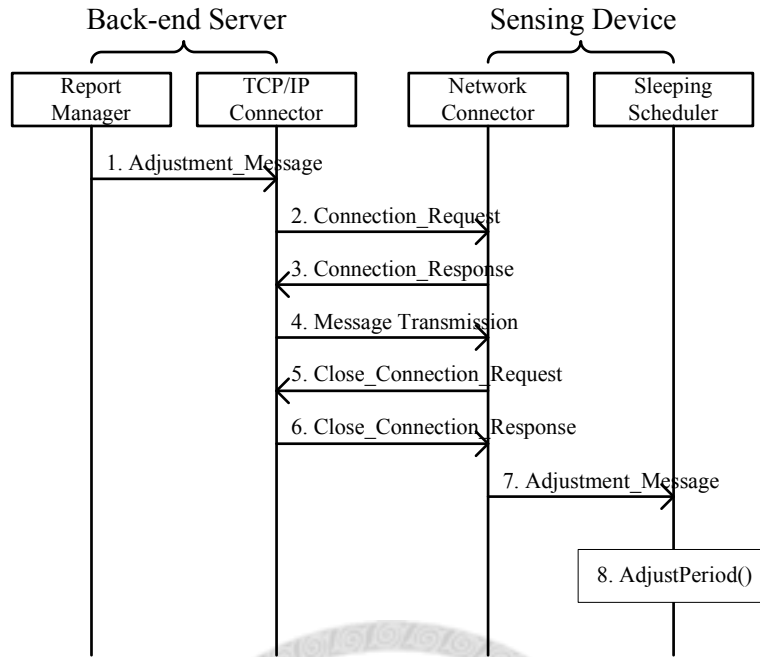


Figure 5.13: The message flow for the sleeping period adjustment procedure

Steps 12 and 13. The TCP/IP connector invokes the `Socket.Close()` to close the connection between the back-end server and the sensing device by exchanging the `Close_Connection_Request` message and the `Close_Connection_Response` message with the network connector.

5.3.2 The Sleeping Period Adjustment Procedure

Through this procedure, the sensing device can adjust its sleeping period based on the received ratio Γ_j , which is calculated and transmitted by the back-end server. Figure 5.13 illustrates the message flow between the back-end server and the sensing device for the sleeping period adjustment procedure, which is described in the following steps.

Step 1. The report manager invokes the `SendAdjustment()` member method to encapsulate the calculated ratio Γ_j (for sensing region R_j) into the `Adjustment_Message`

message and sends the message to the sensing device through its TCP/IP connector.

Steps 2 and 3. Upon receipt of the `Adjustment_Message` message, the TCP/IP connector invokes the `Socket.Connect()` method to exchange the `Connection_Request` message and the `Connection_Response` message with the network connector to establish the connection between the back-end server and the sensing device.

Step 4. The TCP/IP connector starts to transmits the `Adjustment_Message` message through the `Socket.Send()` method.

Steps 5 and 6. Upon receipt of the `Adjustment_Message` message, the network connector and the TCP/IP connector invokes the `Socket.Close()` method to exchange the `Close_Connection_Request` message and the `Close_Connection_Response` message to close the connection between the back-end server and the sensing device.

Step 7. The network connector forwards the `Adjustment_Message` message to the sleeping scheduler.

Step 8. The sleeping scheduler invokes the `AdjustPeriod()` method to adjust its sleeping period based on the Γ_j ratio carried in the `Adjustment_Message` message (whose details we described in Chapter 5.2.2).

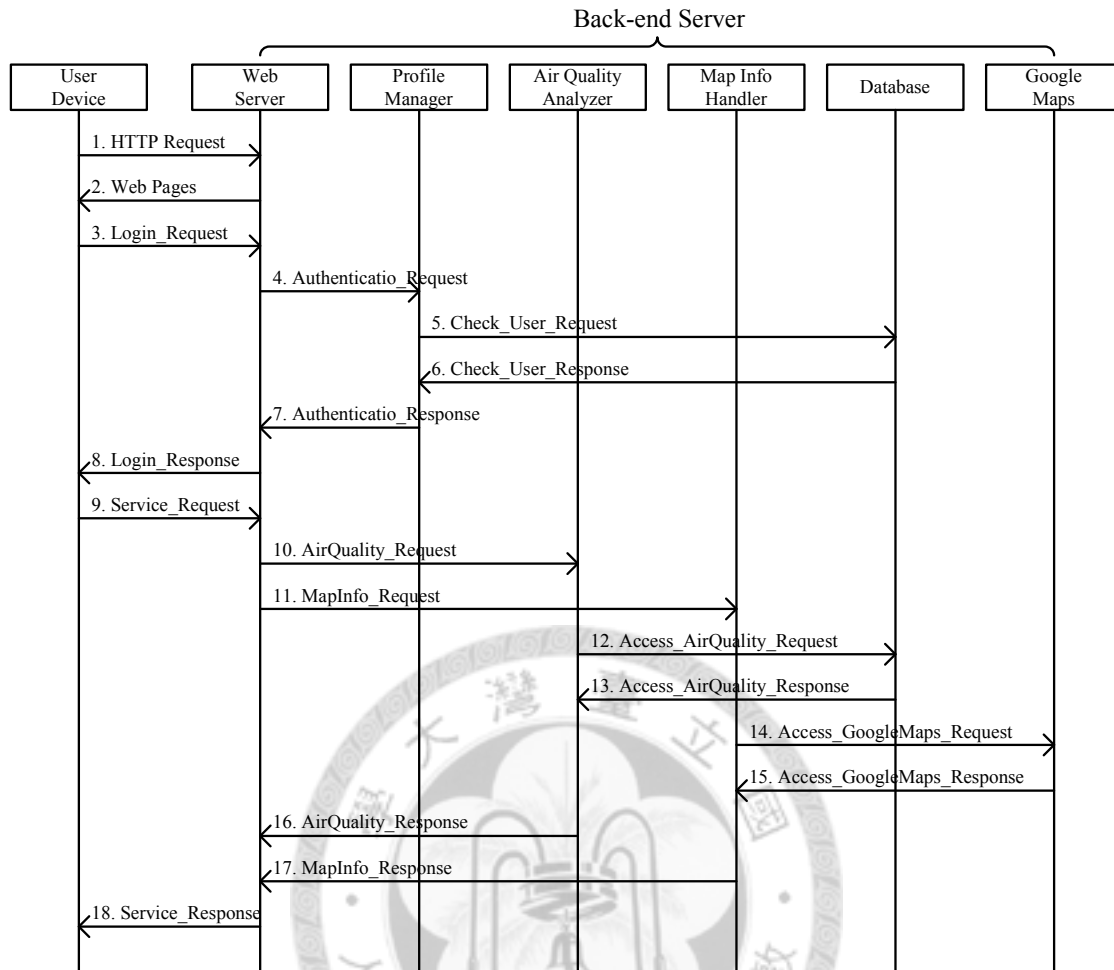


Figure 5.14: The message flow for the navigation service procedure

5.3.3 The Navigation Service Procedure

Figure 5.14 illustrates the message flow between the user device and the back-end server for the navigation service procedure, which is described in the following steps.

Steps 1 and 2. The user starts the navigation service by web browsing. Once the web server on the back-end server receives an HTTP request, it replies the web pages to the user device.

Step 3. The user logs in on the web server by sending the `Login_Request` message (containing the user ID and password) to the server.

Step 4. Upon receipt of the `Login_Request` message, the web server sends the `Authentication_Request` message (containing the user ID and password) to the profile manager to authenticate the user.

Steps 5 and 6. The profile manager then invokes the `Access()` member method to query the user profile by sending the `Check_User_Request` message to the database. The database returns the `Check_User_Response` message which contains the user ID and password to the profile manager.

Step 7. The profile manager compares the user ID and password in the `Authentication_Request` message with the user ID and password in the `Check_User_Response` message. The profile manager notifies the web server whether the authentication is successful by sending the `Authentication_Response` message.

Step 8. The web server sends the `Login_Response` message back to the user device. If the authentication is successful, the user then can use the navigation service.

Step 9. The user device sends the `Service_Request` message to the web server to access the navigation service.

Steps 10 and 11. When the web server receives the `Service_Request` message from the user device, the web server triggers the air quality analyzer and the map info handler by sending them the `AirQuality_Request` message and the `MapInfo_Request` message, respectively.

Steps 12 and 13. Upon receipt of the `AirQuality_Request` message, the air quality analyzer invokes the `QueryAirQuality()` method to query the air qual-

ity data by exchanging the `Access_AirQuality_Request` message and the `Access_AirQuality_Response` message with the database. Then, the air quality analyzer invokes the `PlanRoute()` methods to plan routes for users according to the results of the `EvaluateRoadSection()` method.

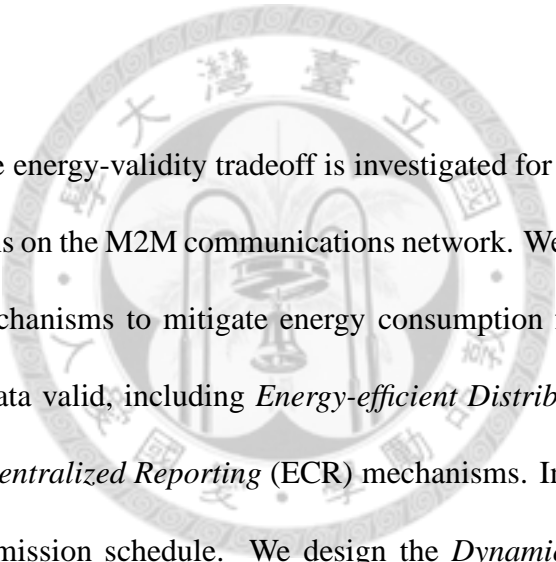
Steps 14 and 15. After receiving the `MapInfo_Request` message, the map info handler exchanges the `Access_GoogleMaps_Request` message and the `Access_GoogleMaps_Response` message with the Google Maps server to get map information.

Steps 16-18. The air quality analyzer and the map info handler send the `AirQuality_Response` message and the `MapInfo_Response` message to the web server, respectively. Then, the web server integrates the planned routes (i.e., the directions and the path) carried in the `AirQuality_Response` message with the map information carried in the `MapInfo_Response` message. Then, the web server sends the `Service_Response` message (containing the planned routes and the map information) message to the user device.



Chapter 6

Conclusions and Future Work



In this dissertation, the energy-validity tradeoff is investigated for “multi-type” real-time monitoring applications on the M2M communications network. We proposed two energy-efficient reporting mechanisms to mitigate energy consumption for M2M nodes while keeping the sensing data valid, including *Energy-efficient Distributed Reporting* (EDR) and *Energy-efficient Centralized Reporting* (ECR) mechanisms. In EDR, the M2M node decides its own transmission schedule. We design the *Dynamic Sleeping Adjustment* (DSA) algorithms for the M2M node to adjust its sleeping period. The analytical models are proposed to investigate the performance for EDR by capturing the reporting and mobility behaviors of M2M nodes at the same time. In ECR, the M2M gateway schedules sensed data transmissions for M2M nodes. The energy-validity tradeoff is transformed to an *energy minimization* problem, and formulated to the energy minimization problem by the integer linear programming (ILP). We showed NP-hardness for the energy minimization problem, and proposed a low-complexity greedy algorithm to approximate the solution of the ILP formulation. Furthermore, a prototyping real-time monitoring system,

Air Pollution Sensing (APS) system, is developed to monitor the air quality in the urban environment, which has been deployed in the campus of National Taiwan University for small-scale testing.



Bibliography

- [1] *Apache HTTP Server*. <http://httpd.apache.org>.
- [2] *ASUS EeePC*. <http://www.asus.com>.
- [3] *Bing Maps*. <http://www.bing.com/maps>.
- [4] *CO2 Meter*. <http://www.co2meter.com>.
- [5] *GlobalSat GPS Receiver*. <http://www.globalsat.com.tw>.
- [6] *Google Maps API*. <http://code.google.com/apis/maps/index.html>.
- [7] *Google Maps*. <http://maps.google.com>.
- [8] *MySQL*. <http://www.mysql.com>.
- [9] *.NET Framework*. <http://www.microsoft.com/net>.
- [10] *Zigbee*. <http://www.zigbee.org>.
- [11] *ZTE*. <http://www.zte.com.cn/en>.
- [12] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Terminals; Technical realization of Cell Broadcast Service (CBS); (Release 1998). Technical Report GSM TS 03.41, 3GPP, September 2000.

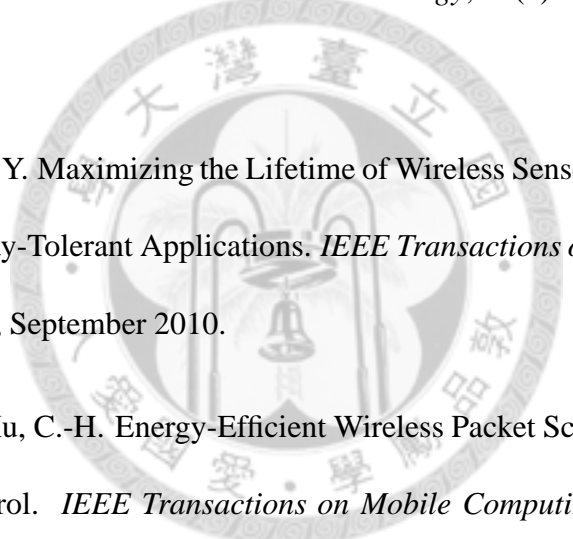
- [13] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2 (Release 10). Technical Report 3G TS 36.300, 3GPP, September 2010.
- [14] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; UTRAN overall description (Release 10). Technical Report 3G TS 25.401, 3GPP, March 2011.
- [15] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Service requirements for Machine-Type Communications (MTC); Stage 1 (Release 11). Technical Report 3G TS 22.368, 3GPP, March 2011.
- [16] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; System Improvements for Machine-Type Communications; (Release 11). Technical Report 3G TS 23.888, 3GPP, June 2011.
- [17] Akyildiz, I.F. Pompili, D., Melodia, T. Underwater Acoustic Sensor Networks: Research Challenges. *Ad Hoc Networks (Elsevier) Journal*, 3(3):257–279, March 2005.
- [18] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, Mar 2000.
- [19] F. Bouabdallah, N. Bouabdallah, and R. Boutaba. Toward reliable and efficient reporting in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 7(8):978–994, August 2008.

- [20] F. Cali, M. Conti, and E. Gregori. Dynamic tuning of the iee 802.11 protocol to achieve a theoretical throughput limit. *IEEE/ACM Transactions on Networking*, 8(6):785–799, Dec 2000.
- [21] M.L. Chaudhry. Transient solution for the number of busy servers for the discrete-time queue $\text{geom}^X/g/\infty$. *The Indian Journal of Statistics, Series B*, 62(2):290–302, Aug 2000.
- [22] Chen, M., Gonzalez, S., Vasilakos, A., Cao, H., and Leung, V.C.M. Body Area Networks: A Survey. *ACM/Springer Mobile Networks and Applications*, 16(2):171–193, August 2010.
- [23] C.F. Chiasserini and M. Garetto. An analytical model for wireless sensor networks with sleeping nodes. *IEEE Transactions on Mobile Computing*, 5(12):1706–1718, December 2006.
- [24] Cormen, T.H., Leiserson, C.E., Rivest, R.L., and Stein C. *Introduction to Algorithms*. MIT Press and McGrawHill, second edition, 2001.
- [25] ETSI. Machine to Machine Communications (M2M); M2M Service Requirements V1.1.1. Technical report, August 2010.
- [26] ETSI. Machine to Machine Communications (M2M); M2M Functional Architecture V1.1.1. Technical report, October 2011.
- [27] H.-L. Fu, H.-C. Chen, and P. Lin. APS: Distributed Air Pollution Sensing System on Wireless Sensor and Robot Networks. *Comput. Commun.*, 35(9):1141–1150, May 2012.

- [28] 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*, 42(1):58–65, January 2009.
- [29] Hartenstein, H. and Laberteaux, K.P. A Tutorial Survey on Vehicular Ad Hoc Networks. *IEEE Communications Magazine*, 46(6):164–171, June 2008.
- [30] CitySense - An Open, Urban-Scale Sensor Network Testbed, 2008. <http://www.citysense.net>.
- [31] IEEE. IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems. Technical Report IEEE Std 802.16e, IEEE, February 2006.
- [32] IEEE. IEEE Standard for Information Technology - Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks - Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Technical report, March 2012.
- [33] M. Khabazian and M. Ali. A performance modeling of connectivity in vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 57(4):2440–2450, 2008.
- [34] 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*, 10(5):505–518, April 2011.

- [35] Lin, Y.-B., and Yang, S.-R. A Mobility Management Strategy for GPRS. *IEEE Transactions on Wireless Communications*, 2(6):1178–1188, November 2003.
- [36] 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*, 49(4):28–35, April 2011.
- [37] Center for Environmental Sensing and Modeling, 2008. <http://censam.mit.edu>.
- [38] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda. PEIR, the Personal Environmental Impact Report, as a Platform for Participatory Sensing Systems Research. *Proceedings of MobiSys'09*, pages 22–25, June 2009.
- [39] R.N. Murty, G. Mainland, I. Rose, A.R. Chowdhury, A. Gosain, J. Bers, and M. Welsh. CitySense: An Urban-Scale Wireless Sensor Network and Testbed. *Proceedings of IEEE HST Conference*, pages 583–588, May 2008.
- [40] R. Nelson. *Probability, Stochastic Processes, and Queueing Theory*. Springer-Verlag, 1995.
- [41] A.-C. Pang, Y.-B. Lin, and Y. Fang. Implicit deregistration with forced registration for pcs mobility management. *Wireless Network*, 7(1):99–104, 2001.
- [42] Park, M.K. and Rodoplu, V. UWAN-MAC: An Energy-Efficient MAC Protocol for Underwater Acoustic Wireless Sensor Networks. *IEEE Journal of Oceanic Engineering*, 32(3):710–720, July 2007.

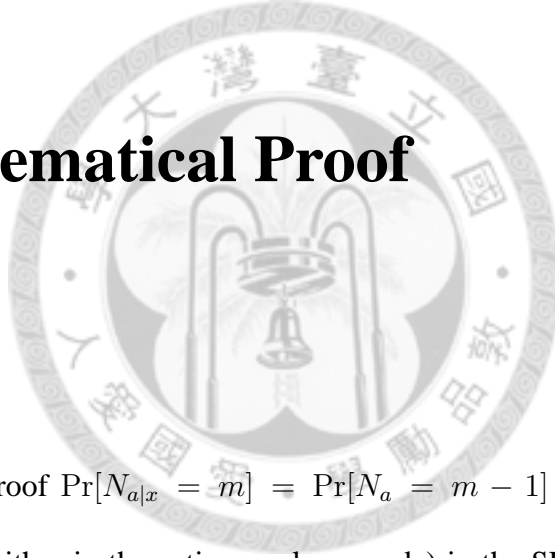
- [43] Rao, H. C.-H., Chang, D.-F., and Lin, Y.-B. iSMS: An Integration Platform for Short Message Service and IP Networks. *IEEE Network*, 15(2):48–55, March/April 2001.
- [44] Rhee, I., Warrier, A., Min, J., and Xu, L. DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 8(10):1384–1396, October 2009.
- [45] S.M. Ross. *Stochastic Processes, second ed.* John Wiley & Sons, 1996.
- [46] 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*, 9(7):927–940, July 2010.
- [47] Srivastava, R. and Koksai, C.E. Energy Optimal Transmission Scheduling in Wireless Sensor Networks. *IEEE Transactions on Wireless Communications*, 9(5):1650–1660, May 2010.
- [48] Szczerba, R.J., Galkowski, P., Glicktein, I.S., and Ternullo, N. Robust Algorithm for Real-time Route Planning. *IEEE Transactions on Aerospace and Electronic Systems*, 36(3):869–878, July 2000.
- [49] H. Takagi. *Queueing Analysis - A Foundation of Performance Evaluation, Volume 3: Discrete-Time Systems.* North-Holland, Amsterdam, 1993.
- [50] H.M. Taylor and S. Karlin. *An Introduction to Stochastic Modeling.* Academic Press, 1998.
- [51] Personal Environmental Impact Report, 2008.
<http://urban.cens.ucla.edu/projects/peir>.

- [52] Wang, H., Agoulmine, N., and Jin, Y. Network Lifetime Optimization in Wireless Sensor Networks. *IEEE Journal on Selected Areas in Communications*, 28(7):1127–1137, September 2010.
- [53] 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*, 9(11):1622–1622, November 2010.
- [54] Yang, S.-R., and Lin, Y.-B. Performance Evaluation of Location Management in UMTS. *IEEE Transactions on Vehicular Technology*, 52(6):1603–1615, November 2003.
- [55] Yun, Y. and Xia, Y. Maximizing the Lifetime of Wireless Sensor Networks with Mobile Sink in Delay-Tolerant Applications. *IEEE Transactions on Mobile Computing*, 9(9):1308–1318, September 2010.
- [56] Zhong, H. and Xu, C.-H. Energy-Efficient Wireless Packet Scheduling with Quality of Service Control. *IEEE Transactions on Mobile Computing*, 6(10):1158–1170, August 2007.
- 
- The image contains a large, faint watermark of the National Sun Yat-sen University seal. The seal is circular and features the university's name in Chinese characters: '國立臺灣大學' (National Sun Yat-sen University) at the top and '安學品教' (Peace, Learning, Quality, Education) at the bottom. In the center of the seal is a traditional Chinese symbol, likely representing a scale of justice or a similar emblem.



Appendix A

The Mathematical Proof



In this chapter, we proof $\Pr[N_{a|x} = m] = \Pr[N_a = m - 1]$ as follows. Let N be the number of MSs (either in the active or sleep mode) in the SR. Let $\alpha = \lambda/\eta$. Then from [21], we have

$$\Pr[N = n] = \frac{\alpha^n e^{-\alpha}}{n!}. \quad (\text{A.1})$$

Since a random observer has the probability p to see an M2M node performing the DCF to contend the wireless medium, we have

$$\Pr[N_a = m | N = n] = \binom{n}{m} p^m (1 - p)^{n-m}. \quad (\text{A.2})$$

Applying (A.1) and (A.2), we obtain

$$\begin{aligned}
 \Pr[N_a = m] &= \sum_{n=m}^{\infty} \Pr[N_a = m|N = n] \Pr[N = n] \\
 &= \sum_{n=m}^{\infty} \binom{n}{m} p^m (1-p)^{n-m} \left(\frac{\alpha^n e^{-\alpha}}{n!} \right) \\
 &= \frac{(\alpha p)^m e^{-\alpha p}}{m!}.
 \end{aligned}$$

As defined in Chapter 3.2, we have

$$\begin{aligned}
 \Pr[N_{a|x} = m] &= \Pr[N_a = m | n_x \text{ is performing DCF}] \\
 &= \sum_{n=m-1}^{\infty} \Pr[N_a = m | N = n, n_x \text{ is performing DCF}] \\
 &\quad \Pr[N = n | n_x \text{ is performing DCF}].
 \end{aligned}$$

Since N is independent of that whether n_x is performing the DCF or not, we have

$$\begin{aligned}
 \Pr[N_{a|x} = m] &= \sum_{n=m-1}^{\infty} \Pr[N_a = m | N = n, n_x \text{ is performing DCF}] \Pr[N = n] \\
 &= \sum_{n=m-1}^{\infty} \binom{n}{m-1} p^{m-1} (1-p)^{n-m+1} \left(\frac{\alpha^n e^{-\alpha}}{n!} \right) \\
 &= \frac{(\alpha p)^{m-1} e^{-\alpha p}}{(m-1)!} \\
 &= \Pr[N_a = m-1].
 \end{aligned}$$

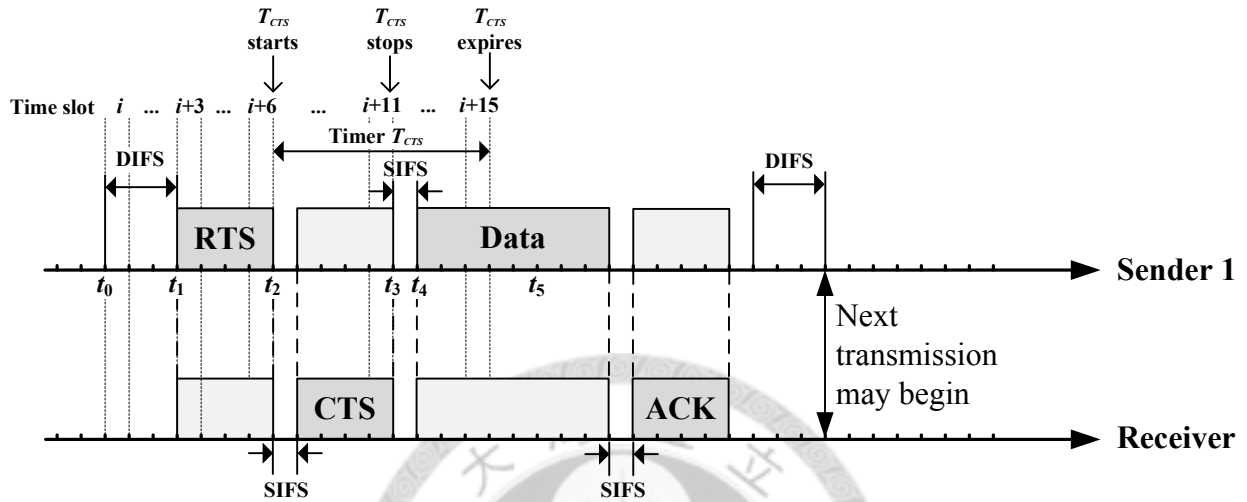
Appendix B

The DCF of IEEE 802.11

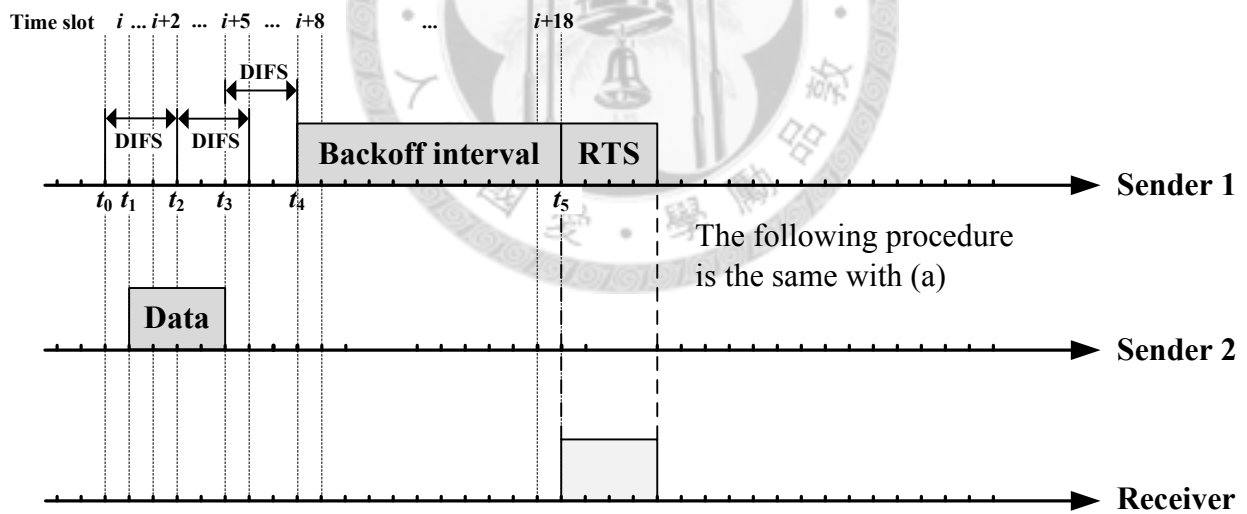
The operation of the IEEE 802.11 protocol [32] is briefly described in this appendix. The IEEE 802.11 protocol defines the Distributed Coordination Function (DCF) to contend for the wireless medium for transmission. The DCF is based on a discrete-time backoff scale [32], that is, the backoff interval is slotted. The transmission starts at the beginning of a time slot. Figure B.1 shows an example to illustrate the execution of DCF.

Suppose that sender 1 has data to transmit to the receiver at t_0 (that is the beginning of time slot i). sender 1 senses the wireless medium by turning on the receiver for a time period, namely, DCF InterFrame Space (DIFS) interval. In this figure, we suppose that the length of a DIFS interval is three time slots. We consider the following two cases for the sensing result by sender 1 during a DIFS interval:

Case 1: See Figure B.1 (a). There is no transmission from other nodes (either MSs or the receiver) during the DIFS interval (i.e., sender 1 senses the wireless medium as IDLE). At the beginning of time slot $i + 3$ (i.e., t_1), sender 1 starts the transmission



(a) The wireless medium is sensed IDLE during the DIFS interval



(b) The wireless medium is sensed BUSY during the DIFS interval

Figure B.1: The operation of the DCF in the IEEE 802.11 protocol

of an Request-To-Send (RTS) frame to the receiver. Suppose that the transmission length of an RTS frame is four time slots. After transmission of the RTS frame at t_2 (i.e., the end of time slot $i + 6$), sender 1 starts the timer T_{CTS} , and expects to receive the Clear-To-Send (CTS) frame (where we assume that the length of a CTS frame is four time slots) from the receiver before T_{CTS} expires.

If the CTS frame is received by sender 1 at t_3 before T_{CTS} expires (as shown in this figure), sender 1 waits for a time interval, namely, Short InterFrame Space (SIFS), and then starts the transmission of the data frame at t_4 , where we assume the length of the SIFS is one time slot, and the length of the data frame is eight time slots.

Otherwise (i.e., sender 1 does not receive the CTS frame before T_{CTS} expires), sender 1 executes the Backoff procedure (details can be found in [32]) to calculate a random backoff timer T_{backoff} at t_5 (when T_{CTS} expires). sender 1 waits for the DIFS interval and counts down T_{backoff} . After the T_{backoff} timer expires, and then resends the RTS frame.

Case 2: See Figure B.1 (b). The wireless medium is occupied by the transmission of other nodes (either sender or receiver; in this figure, sender 2 is transmitting the data frame) during the DIFS interval (i.e., sender 1 senses the wireless medium BUSY at t_1). After DIFS ends at t_2 (where $t_2 > t_1$), sender 1 starts to sense the wireless medium for another DIFS interval. During this DIFS, if sender 1 senses the wireless medium IDLE at t_3 , it resets DIFS. At the end of the DIFS interval (i.e., t_4), sender 1 executes the Backoff procedure to calculate the T_{backoff} timer. After the T_{backoff} timer expires at t_5 (i.e., the end of time slot $i + 18$), sender 1 sends the RTS frame. The sender 1 then performs Case 1 to transmit the data frame.

Curriculum Vita

Huai-Lei Fu received his bachelor, master and doctoral degrees in Computer Science and Information Engineering (CSIE) from Tatung University (TTU), Yuan Ze University (YZU) and National Taiwan University (NTU) in 2005, 2007 and 2012, respectively. His research interests include machine-to-machine communications network, multicast and broadcast service, mobility management, performance modeling, and wireless sensor network. He has received the YZU Academic Silver Medal Award in 2006, the YZU School Work Silver Medal Award in 2006, the Y. Z. Hsu Scholarship Award in 2006 and 2007, and the NTU Outstanding Student Award (Academic Category) in 2009. He is an IEEE student member.

Honors and Awards

Programming Contest Awards

- **Championship, Microsoft Imagine Cup Software Design Competition, Taiwan Finals, 2008**
 - By the project “APS: Air Pollution Sensing System”
 - Served as the team leader
 - Exhibit the project in the Muse’e du Louvre
 - Exhibit the project in the 2008 Taipei International Invention Show and Technomart, Sept., 2008

- **Superior Class, Digital Life Creativity Contest (eLife), 2006**

- By the project “iCATV”
- Served as the team leader and application developer

Academic Awards

- **National Taiwan University Outstanding Student Award in Academic, 2008**

- This award is aimed at rewarding and recognizing students for their academic or professional excellence.

- **Y. Z. Hsu Scholarship Award, 2007**

- **Y. Z. Hsu Scholarship Award, 2006**

- **YZU School Work Silver Medal Award, 2006**

- **YZU Academic Silver Medal Award, 2006**



Publications

Submitted Papers

- **Fu, H.-L.**, Wang, T.-Y., Lin, P., and Fang, Y. Tradeoff between Energy Efficiency and Report Validity for Mobile Sensor Networks. Major Revised to **ACM Transactions on Sensor Networks**.
- **Fu, H.-L.**, Lin, P., Yue, H., and Huang, G.-M. Group Location Management for Large-scale Machine-to-Machine Mobile Networking. Submitted to Proceedings of **IEEE International Conference on Computer Communication (INFOCOM'13)**.
- Yue, H., **Fu, H.-L.**, Guo, L., Fang, Y., and Lin, P. An Efficient Prediction-based Routing Protocol in Delay Tolerant Networks. Submitted to Proceedings of **IEEE International Conference on Computer Communication (INFOCOM'13)**.

Refereed Journal Papers

- **Fu, H.-L.**, Lin, P., and Lin, Y.-B. Reducing Registration Traffic for Femtocell/Macrocell Network. Accepted to appear in **IEEE Transactions on Mobile Computing**.
- **Fu, H.-L.**, Chen, H.-C., and Lin, P. APS: Distributed Air Pollution Sensing System on Wireless Sensor and Robot Networks. **Computer Communications**, 35(9): 1141-1150, May 2012.
- Hu, K.-H., **Fu, H.-L.**, and Lin, P. MBS Zone Configuration Schemes for Wireless Multicast and Broadcast Service. **Wireless Communications and Mobile Computing (WCMC)**, 10(12):1589-1604, December 2010.

- Chung, T.-Y., Fu, H.-L., and Lee, H.-C. DCLS: A Fair and Efficient Scheduling with Bandwidth Guarantee in WiMAX Mesh Networks. **Journal of Information Science and Engineering**, 26(4):1429-1442, July 2009.
- Guizani, M., Lin, P., Cheng, S.-M., Huang, D.-W., and Fu, H.-L. Performance Evaluation for Minislot Allocation for Wireless Mesh Network. **IEEE Transactions on Vehicular Technology**, 57(6): 3732-3745, November 2008.

Conference Papers

- Fu, H.-L., Chen, H.-C., Lin, P., and Fang, Y. Energy-Efficient Reporting Mechanisms for Multi-Type Real-time Monitoring in Machine-to-Machine Communications Networks. Proceedings of **IEEE International Conference on Computer Communication (INFOCOM'12)**, Orlando, FL, March 25-30, 2012. (Acceptance ratio: $278/1547 = 17.97\%$)
- Lin, Z.-C., Fu, H.-L., and Lin, P. Dynamic Channel Allocation for Wireless Zone-Based Multicast and Broadcast Service. Proceedings of **IEEE International Conference on Computer Communication (INFOCOM'11)**, Shanghai, China, April 10-15, 2011. (Acceptance ratio: $291/1823 = 15.96\%$)
- Chen, H.-C., Fu, H.-L., Lin, P., and Hsu, C.-H. Energy-aware Transmission Scheduling in Mobile Sensor Networks. Proceedings of **IEEE Global Communications Conference (GLOBECOM'11)**, Houston, TX, December 5-9, 2011.
- Wang, S.-N., Lin, P., Gan, C.-H., and Fu, H.-L. A Study for Location Update Cost in a Femtocell Network. Proceedings of **IEEE Vehicular Technology Conference**

(VTC2010-Fall), Ottawa, Canada, September 6-9, 2010.

- **Fu, H.-L.**, Wang, T.-Y., Lin, P., and Fang, Y. A Region-based Reporting Scheme for Mobile Sensor Networks. Proceedings of **IEEE Vehicular Technology Conference (VTC2010-Spring)**, Taipei, Taiwan, May 16-19, 2010.
- Hu, K.-H., **Fu, H.-L.**, and Lin, P. Design of Zone Configuration Scheme for Wireless Zone-based Multicast and Broadcast Service. Proceedings of **The International Wireless Communications and Mobile Computing Conference (IWCMC 2010)**, Caen, France, June 28, 2010 - July 2, 2010.
- Wang, C.-Y., Lin, P., Shih, C.-S., **Fu, H.-L.**, and Jeng, J.-Y. A Middleware Approach for Migration of Legacy Telecom Operational Support Systems into NGOSS-Compliant. Proceedings of **The Asia-Pacific Network Operations and Management Symposium (APNOMS 2011)**, Taipei, Taiwan, September 21-23, 2011.
- Hung, C.-K., **Fu, H.-L.**, Chen, Y.-K., Chu, C.-T., Huang, K.-M., Shih, C.-S., Lin, P., and Jeng, J.-Y. Seamless On-line Service Upgrade for Telecommunication Web-Services. Proceedings of **The Asia-Pacific Network Operations and Management Symposium (APNOMS 2011)**, Taipei, Taiwan, September 21-23, 2011.
- Chung, T.-Y., Chang, Y.-H., Chiu, S.-S., Yang, C.-J., Chen, Y.-M., and **Fu, H.-L.** SOP: SIU-based Overlay Proxy Service for Peer-to-Peer Streaming. Proceedings of **International Computer Symposium (ICS)**, vol. 2, pp. 783-789, Taipei, Taiwan, December 4-6, 2006.
- Chung, T.-Y., **Fu, H.-L.**, Wang, C.-C., and Chen, Y.-M. OSSP: Overlay Streaming Service Platform. Proceedings of **The Joint Conference on Information (JCIS)**,

vol. 1, pp. 796-799, Kaohsiung, Taiwan, October 8-11, 2006.

Pending Patents

- **Fu, H.-L.**, Hu, K.-H., Wang, T.-Y., Lin, Z.-C., Lin, P., and Fang, Y. System and Method of Detecting Air Pollution, Route-Planning Method Applied to Said Detection System, and Warning Method of Air Pollution, R.O.C. and U.S.A. Patent.
- Wang, T.-Y., **Fu, H.-L.** and Lin, P. Reporting Method for Data Transmission Device and Computer Readable Medium Thereof, R.O.C. and U.S.A. Patent.

