國 立 交 通 大 學

電信工程研究所

碩 士 論 文

Optimal Energy-Efficient Location Update for Mobile Computing

應用最佳節能位置上傳機制之研究於行動計算環境

研究生：張凱評

指導教授：王蒞君 教授

中 華 民 國 一 百 年 八 月

應用最佳節能位置上傳機制之研究於行動計算環境

Optimal Energy-Efficient Location Update for Mobile Computing

研 究 生：張凱評　　　　Student：Kai-Ping Chang

指導教授：王蒞君　　　　Advisor：Li-Chun Wang

國 立 交 通 大 學

電信工程研究所

碩 士 論 文

A Thesis

Submitted to Institute of Communications Engineering

College of Electrical and Computer Engineering

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master of Science

In

Communications Engineering

August 2011

Hsinchu, Taiwan, Republic of China

中華民國 一百 年 八 月

應用最佳節能位置上傳機制之研究於行動計算環境

學生：張凱評　　　　　　　　　　　　　　　　　指導教授：王蒞君

國立交通大學

電機學院電信工程研究所

摘要

提供適地性服務 (Location-Based Services, LBSs) 的最關鍵議題之一，是行動終端裝置因位置的頻繁更新需求所造成的耗電 (Power Consumption) 問題。在本篇論文中，我們提出一個電源感知的位置上傳機制 (Power-Aware Location Update Scheme)，另稱作基於能量考量的位置回傳方法 (Energy-Based Location Reporting)。本論文應用動態規劃 (Dynamic Programming, DP) 於傳統的距離考量位置回傳方法 (Distance-Based Location Reporting)，決定最佳節能位置上傳 (Optimal Energy-Efficient Location Update) 機制，使得在行動計算 (Mobile Computing) 環境中，達成降低耗電的目標。提出基於動態規劃下 (DP-Based) 的位置上傳機制，容許位置回傳錯誤 (Location Reporting Errors) 在令人滿意的範圍內。我們也在不同的能量比率 (Energy Ratio) 與不同的終端移動速度 (Terminal Moving Speed) 下，驗證提出的最佳節能位置上傳機制。

Optimal Energy-Efficient Location Update for Mobile Computing

A THESIS Presented to

The Academic Faculty By

Kai-Ping Chang

In Partial Fulfillment

of the Requirements for the Degree of

Master in Communications Engineering

Institute of Communications Engineering

College of Electrical and Computer Engineering

National Chiao Tung University

2011

Abstract

One of the most critical issues in providing location-based service is the power consumption in mobile terminals due to frequent location queries. In this thesis, we present a power-aware location update scheme, called energy-based location reporting. We apply dynamic programming (DP) to the traditional distance-based location reporting method to determine the optimal location update scheme with an objective of minimizing power consumption in mobile computing. The proposed DP-based energy-efficient location update scheme can also tolerate the location reporting errors within a satisfied level. We also verify the optimal location update scheme by simulations for different energy ratios and different terminal moving speeds.

Acknowledgements

I would like to thank my parents and my older brother. They always give me endless supports. I especially thank Professor Li-Chun Wang who gave me many valuable suggestions in my research during these two years. I would not finish this work without his guidance and comments.

In addition, I am deeply grateful to my laboratory mates, Chu-Jung, Ang-Hsun, I-Cheng, Wei-Ping, Chien-Cheng, Yu-Jung, Tsung-Ting, and junior laboratory mates at Mobile Communications and Cloud Computing Labratory at the Graduate Institute of Communications Engineering in National Chiao-Tung University. They provide me with a lot of assistance and share happiness with me.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Location-based services (LBSs) provide real-time services with user's locations [1–3]. Because current mobile devices have location calculation capability, the LBSs have gained a lot attention over the past years. Some examples of LBSs include providing local information, such as traffic notification, stores, and spots. Another key feature of in recent is related to social networks [4]. A user can obtain the local information based on his current location, and share his local information (e.g. restaurants, shops and tourist spots) with community members.

Even though mobile devices provide the convenience of ubiquitous computations, the use of mobile devices is severely constrained by the limited battery capacity. Thus, it is important to minimize the power consumption of these energy-constrained mobile devices. Figure 1.1 shows a typical network architecture for providing LBSs, consisting of location update mechanisms and wireless communications. Consequently, LBSs consume the energy E_c required for calculating a location, and the energy E_r of reporting a location [5]. According to [6], E_c is usually higher than E_r.

Based on [7], some power consumption methods are proposed to reduce

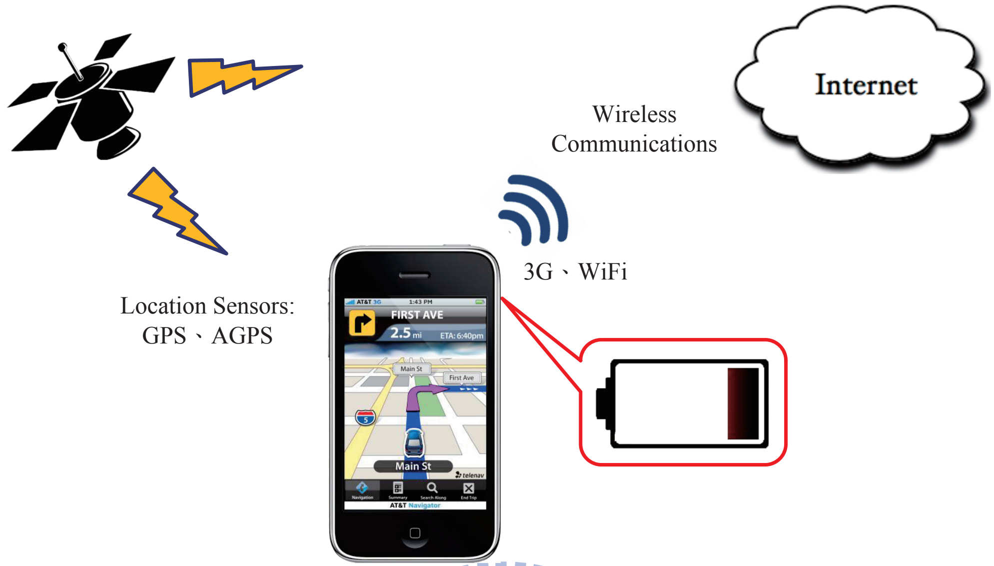


Figure 1.1: A typical network architecture for providing LBSs.

the power consumption on the mobile phones, including (1) minimizing position calculation, (2) selecting position techniques with least power consumption, and (3) minimizing location report frequency.

1. **Minimizing position calculation**: Energy consumption increases with the frequency of location calculation. Thus, reducing redundant location calculation can save power. However, location errors become large when the frequency of location report is reduced. In [8], the cache-based approach was proposed to estimate when the application server needs to obtain a new location from a mobile device. As long as location errors are lower than tolerant value, the latest cached location can be used for LBSs.

2. **Selecting position techniques with least power consumption**: Different location sensors (e.g. GPS, WiFi, and 3G cellular) yields

different power consumption, and have different tolerant location errors in providing LBSs. In [9], it is suggested that the system can choose the appropriate location sensor to save power. For example, if a mobile device stays within the GSM cell, GPS can be switched off.

3. **Minimizing location report frequency**: Reducing the report of location frequency can also save power in mobile devices. Basically, there are two kinds of location reporting mechanisms: (1) the *time-based location reporting* (TBLR) scheme [10–12], and (2) the *distance-based location reporting* (DBLR) scheme [5]. For the TBLR scheme, a moving user sends location information to the server at a fixed time interval. For the DBLR scheme, the moving user sends location information to the server whenever it moves a certain distance. Various location update schemes are suitable for different scenarios. The power consumption of the TBLR and DBLR schemes are affected by different factors. For the TBLR scheme, the energy ratio E_r/E_c affects the power consumption of mobile devices because this scheme requires calculating and reporting locations every fixed period. However, the location calculation periods of mobile locations affect the overall power consumption of the DBLR scheme.

In this thesis, we propose to consider both the factors of location calculation and reporting in the location update mechanism design to minimize the power consumption of mobile devices in LBSs.

On the other hand, location accuracy is also an important factor for LBSs. Figure 1.2 shows that different applications of LBSs require different location accuracy [13]. For example, it usually requires about 80 meters (a minimum city block distance [14]) location accuracy for location-based social networks. Hence, we must consider the location accuracy in our design mechanism. In

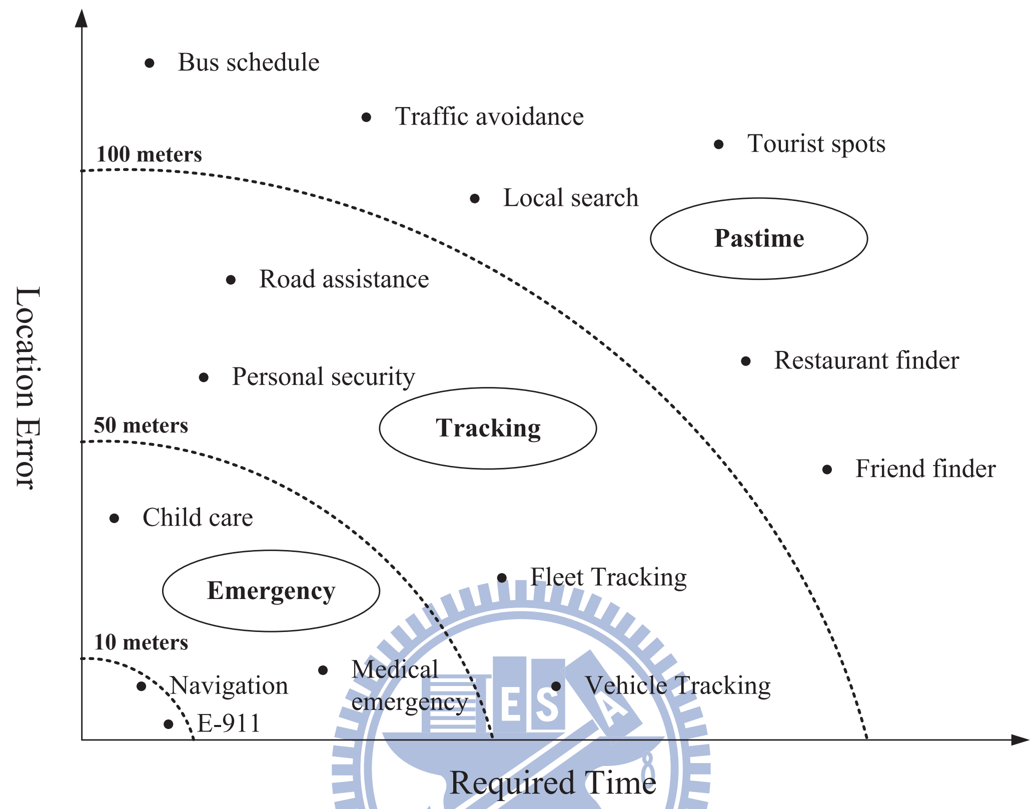


Figure 1.2: Required time and location accuracy for each application of LBSs.

this thesis, we will focus on the LBSs such as location-based social networks to design an energy-efficient location update mechanism.

1.1 Problem and Solution

The objective of this thesis is to develop an energy-efficient location update mechanism for mobile devices to support LBSs. An introduced before, TBLR and DBLR schemes focus on reducing the frequency of reporting mobile location. To further save energy of mobile devices in LBSs, we suggest an energy-efficient location update mechanism needs not only change the frequency of location calculation, but the frequency of location report.

1.1.1 Optimal Energy-Efficient Location Update

In order to minimize the power consumption in mobile terminal for providing LBSs, we investigate how to energy-efficiently change the frequency of location calculation and location report. We present a power-aware location update scheme, called energy-based location reporting method. The proposed energy-based location reporting method applies dynamic programming in traditional distance-based location reporting method to determine the optimal energy-efficient location update. Based on dynamic programming, our proposed location update scheme can not only decide the optimal frequency of location calculation, but the optimal frequency of location report. Because the frequency of location calculation and report are energy-efficiently optimized, the power consumption in mobile devices can be minimized. On the other hand, the dynamic-programming based energy-efficient location update scheme can also tolerate the location reporting errors within a satisfied level.

1.2 Thesis Outline

The rest of this thesis is organized as follows. Chapter 2 introduces the background of our proposed approach. Then, we discuss system models and performance metrics in Chapter 3. Subsequently, we present the proposed dynamic-programming-based energy-efficient location update scheme in Chapter 4. Numerical results are shown in Chapter 5. Finally, we conclude this thesis in Chapter 6.

Chapter 2

Background

In this chapter, we firstly survey related works to location update schemes in mobile computing. Then, the core concept, "dynamic programming," for solving the optimization problem applied in our proposed approach is introduced in depth.

2.1 Literature Survey

Nowadays, tracking schemes for mobile devices in mobile computing are studied. In this part, we introduce two traditional location reporting schemes, including (1) the time-based location reporting (TBLR) scheme and (2) the distance-based location reporting (DBLR) scheme. However, these location update schemes have not simultaneously considered all of the four design features. Table 2.1 classifies the existing location reporting techniques, where the signs "$\circ$" and "$\times$" indicate that the proposed scheme "does" and "does not" consider the corresponding feature, respectively. Furthermore, we illustrate the concepts of both schemes briefly and highlight what literature exploits the above schemes.

Table 2.1: Comparison of Various Schemes for Location Updates.

Scheme Name	Location Accuracy	Energy Efficiency	Minimizing Location Report Frequency	Minimizing Position calculation
TBLR [10–12]	○	×	×	×
DBLR [5]	○	○	○	×
Proposed Scheme	○	○	○	○

2.1.1 Time-Based Location Reporting Scheme

Essentially, the time-based location reporting scheme [10–12] is ubiquitously used for mobile devices in mobile computing, because it is the simplest scheme to implement. In this scheme, each mobile device periodically sends the current location to the application server every τ unit of time. Nevertheless, in one of the discussed problems, the parameter τ dominates system performances, including the location error and the overall power consumption. For example, if a mobile device rapidly moves but with the long period τ, it causes the location error to increase. Consequently, the application server will track the mobile device inaccurately. On the other hand, we are also concerned about the overall power consumption problem for the time-based location reporting scheme. We consider the energy E_r for reporting a location message and the energy E_c for requiring a geographical location from GPS receiver. For the time-based location reporting scheme, each mobile device consumes both E_r and E_c periodically every τ unit of time. The corresponding illustration is shown in Fig. 2.1. Hence, the parameter τ also effects the performance on power consumption.

On the other hand, in order to fairly compare the time-based location re-

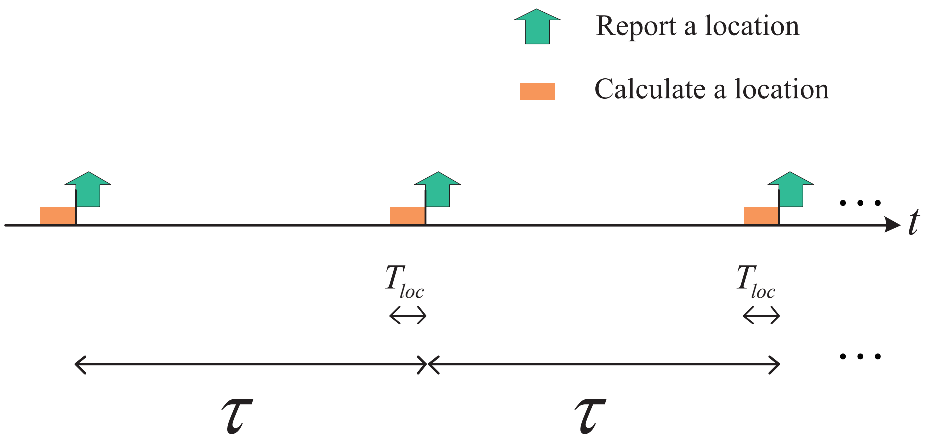


Figure 2.1: An illustration for the time-base location reporting scheme.

porting scheme with our proposed approach, we will use the same observation time to compare two methods. Furthermore, we discuss both of location error and power consumption based on this criterion in the following chapters.

2.1.2 Distance-Based Location Reporting Scheme

In contrast with the time-based location reporting scheme, the distance-based location reporting scheme [5] is definitely different approach. In this scheme, each mobile device tracks the distance d which it moved since last reporting a location message, and it sends a location message again whenever the distance d exceeds a certain parameter d_{up} called distance-based location report condition. However, because of the required location error tolerance from each location-based application server, we must also guarantee that each location calculation is within this tolerance value. The selected location error tolerance determines the report threshold d_{th}, which must not be exceeded by the difference of the two location reports. Figure 2.2 shows how the distance-based location reporting scheme reports the current location. According to the observations, the distance-based location report condition d_{up} dominates system performances, including the location error and the overall power consumption. For example, if a mobile device is with the smaller d_{up}, it causes the overall power consumption to increase but decreases location errors because of frequent location reports. Consequently, the application server will track the mobile device accurately, but the mobile device will drain the battery capacity rapidly. In an existing discussion [5], there are some methods for optimizing value d_{up}.

2.2 Dynamic Programming

In this section, we introduce a mathematical optimization method called "dynamic programming" which will be used in our proposed approach for optimal energy-efficient location update. For optimization problems, dynamic programming is a general mathematical optimization approach for efficiently solving multi-stage optimization problems, or optimal planning problems. It refers to simplifying a complicated problem by breaking it down into simpler subproblems in a recursive manner. In addition, dynamic programming is suitable when subproblems are not independent, which called overlapping subproblems. That is, when subproblems share subsubproblems. This means that dynamic programming can solve each subsubproblem just once and then save its answer in a table, thereby avoiding the work of recomputing the answer every time. This is a benefit of reducing the number of computations.

For developing procedures of dynamic programming, there are three steps, including (1) characterize the overall structure of an optimization problem, (2) define and compute the value of cost function (or reward function) re-

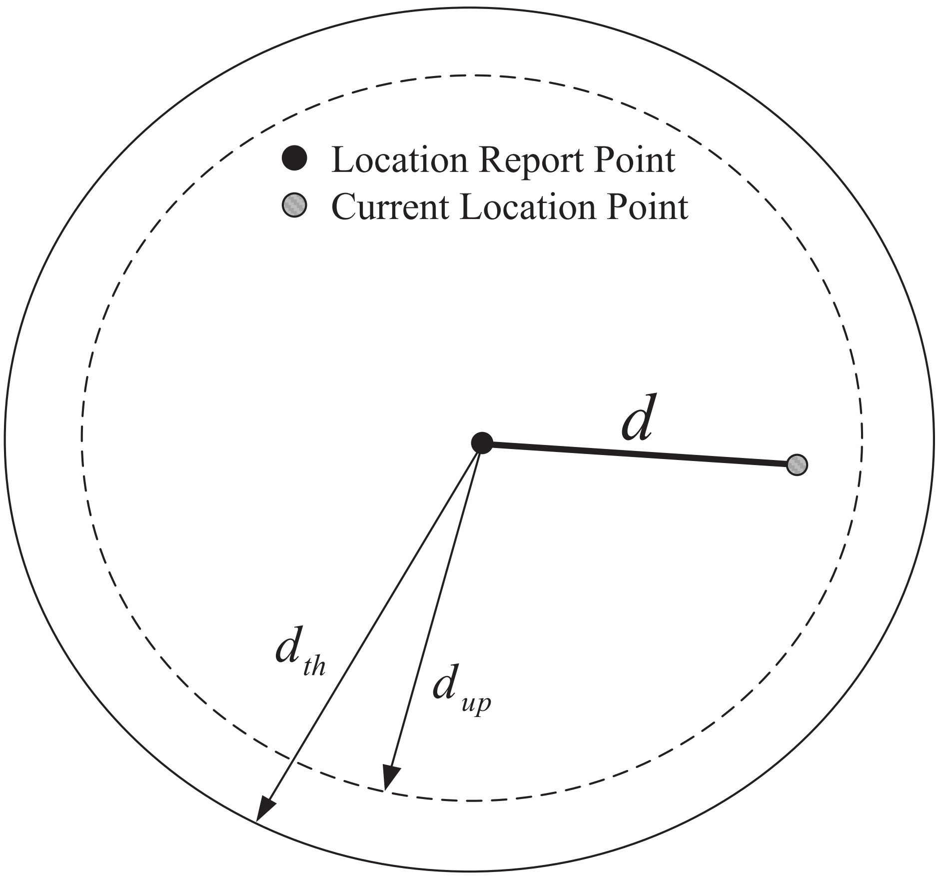


Figure 2.2: An illustration for the distance-base location reporting scheme.

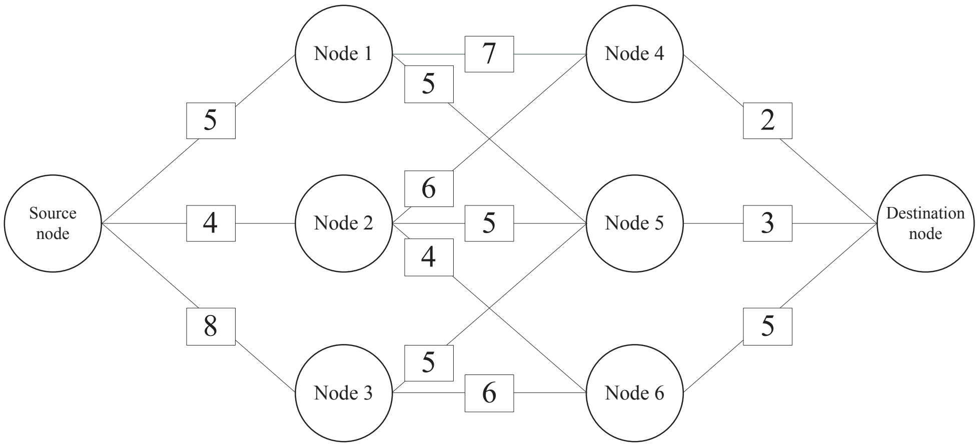


Figure 2.3: A topology for the shortest-path problem.

cursively at each stage, and (3) define and make an optimal solution from all computed information. The general example of a optimization problem with dynamic programming in the deterministic system is the shortest-path problem. Figure 2.3 illustrates a shortest-path topology for the minimum-distance problem.

In general, dynamic programming can solve the multi-stage optimization problem by using two recursive manners. They are shown as follows.

1. *Forward Recursion*: If the recursions proceed from the first stage toward the last stage, the recursive manner is called forward recursion. The forward recursive relationship of the optimal-value function is shown as follows:

$$u_{n+1}(s_{n+1}) = \max_{s_n \in S}\{u_n(s_n) + f_n(s_n, s_{n+1})\} \ . \tag{2.1}$$

In (2.1), n represents the current stage and its range is from 0 to N (the number of stages). $u_n(s_n)$ is the optimal value of all prior decisions.

Note that we are in state s_n and the computations are usually initialized by setting $u_0(s_0) = 0$. $f_n(s_n, s_{n+1})$ is the reward for a particular stage n with the current state s_n and the next state s_{n+1}.

2. *Backward Recursion*: By symmetry, if the recursions proceed from the last stage toward the first stage, the approach is called backward recursion. The backward recursive relationship of the optimal-value function is shown as follows:

$$u_n(s_n) = \max_{d_n \in D_n} \{f_n(d_n, s_n) + u_{n+1}(t_n(d_n, s_n))\} \ . \tag{2.2}$$

   By contrast, in (2.2) $u_n(s_n)$ is the optimal value of all the subsequent decisions. Note that we are in state s_n and the computations are usually initialized by setting $u_N(s_N) = 0$. Furthermore, $f_n(d_n, s_n)$ is the reward for a particular stage n with decision d_n and state s_n. d_n is a permissible decision that may be chosen from the set D_n, and $t_n(d_n, s_n)$ is a transition function which determines the new state at the next stage (i.e. $s_{n+1} = t_n(d_n, s_n)$).

In general, we can adopt one of these two approaches to solve problems. However, there is a primary difference in the "by-products" resulted from these two approaches. In solving a problem by backward recursion, we will obtain by-products which are the optimal values from every state in every stage to the end. Whereas in solving a problem by forward recursion, the corresponding by-products would be the optimal values from the initial state in the first stage to every state in the remaining stages.

For example, in Fig. 2.3, we wish to find the shortest path from the source node to the destination node. The number next to the arcs are distances (or called costs). All corresponding costs are known in advance. A source node has also been added to illustrate that the dynamic-programming solution by using backward recursion finds the shortest path from the destination node to the source node. Essentially, it finds the shortest path from the destination node to all nodes in the topology, thereby solving the minimum-distance problem for any source node. The calculation steps and results with backward recursion are shown in Fig. 2.4. The dynamic-programming solution by using forward recursion finds the shortest path from the source node to the destination node. The calculation steps and results with forward recursion are shown in Fig. 2.5. Although the shortest path will be the same for both methods, forward recursion will not solve the minimum-distance problem for any source node, since all source nodes are not the same. On the other hand, for the problems with uncertain planning horizons, forward recursion has an advantage over backward recursion. Therefore, when using dynamic programming, it is necessary to consider whether forward or backward recursion is the best suitable to the problem you want to solve.

Subsequently, in the real environment, dynamic programming can be used in the deterministic system or the stochastic system. However, in stochastic system the backward recursion is required because of the probabilistic nature of the *markov decision process* (MDP) [15, 16]. In contrast, if the system were deterministic, policies could be evaluated by either forward or backward recursion because expectations need not to be computed.

For our problem, we hope to solve the overall power consumption for location report in any system. We find the a proper optimization approach which is called dynamic programming that can satisfy the above requirement. If we only know the transition probabilities, we have a stochastic dynamic programming problem. Especially, the optimization problem with the

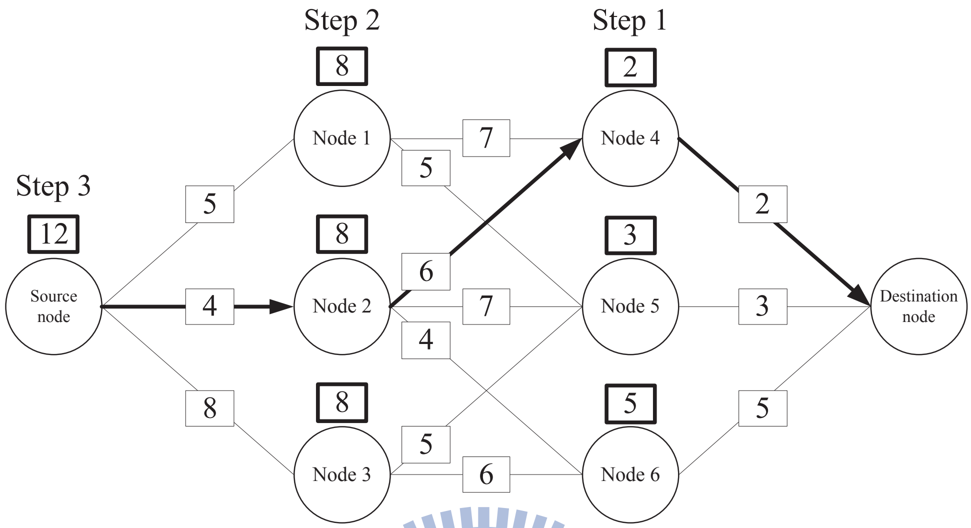


Figure 2.4: Backward recursion for solving the shortest-path problem.

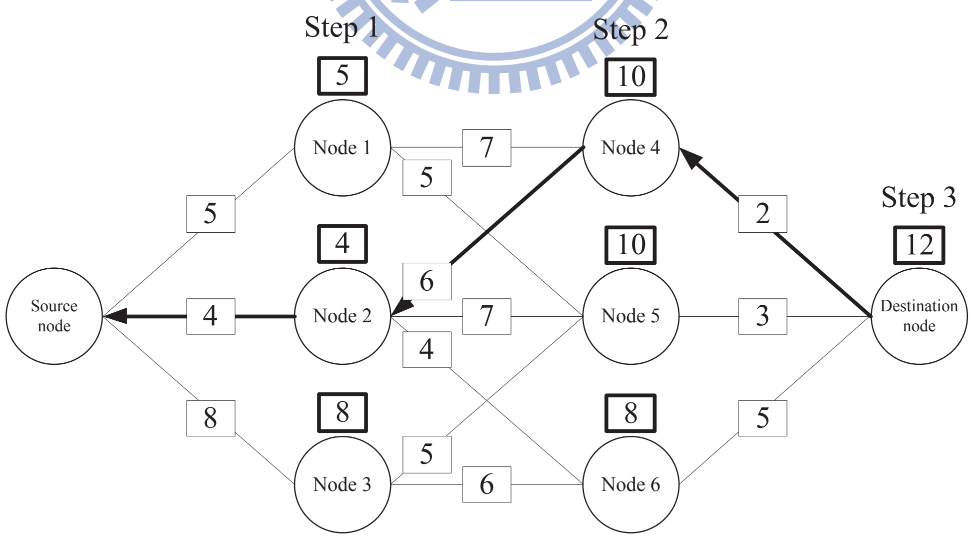


Figure 2.5: Forward recursion for solving the shortest-path problem.

stochastic dynamic programming can be called "markov decision processes (MDPs)" [15,16]. The optimization problem with the deterministic dynamic programming was introduced in the previous section (e.g. the shortest-path problem). Here we are going to especially introduce the fundamental concept of markov decision processes and explain the details for finding the optimal solution.

2.2.1 Markov Decision Processes

The concept in markov decision processes is substantially applied in the optimization problem with the stochastic dynamic programming. In fact, a markov decision process is also a discrete time stochastic process. A markov decision process model consists of five primary elements, including (1) stages n, (2) states s, (3) actions a, (4) transition probabilities p_r, and (5) returns r (rewards or costs).

A markov decision process model is the sequential decision making model shown in Fig. 2.6. At each stage n, the action chooser observes the system state s_n and selects an action a_n based on this system state. The system environment responds at the next stage by randomly changing to a new state s_{n+1}, and giving the action chooser a corresponding return r_n. The corresponding return r_n is based on states s_n, s_{n+1} and action a_n. Hence, r_n can be represented by $r_n(s_{n+1}|s_n, a_n)$. The transition probability that the system state changes to s_{n+1} as a new state is influenced by the chosen action a_n. In particular, it is given by the state transition probability function $p_r(s_{n+1}|s_n, a_n)$. Consequently, the next state s_{n+1} depends on the current state s_n and the chosen action a_n.

For solving the optimization problem with stochastic dynamic programming (or called markov decision processes), the backward recursion is re-

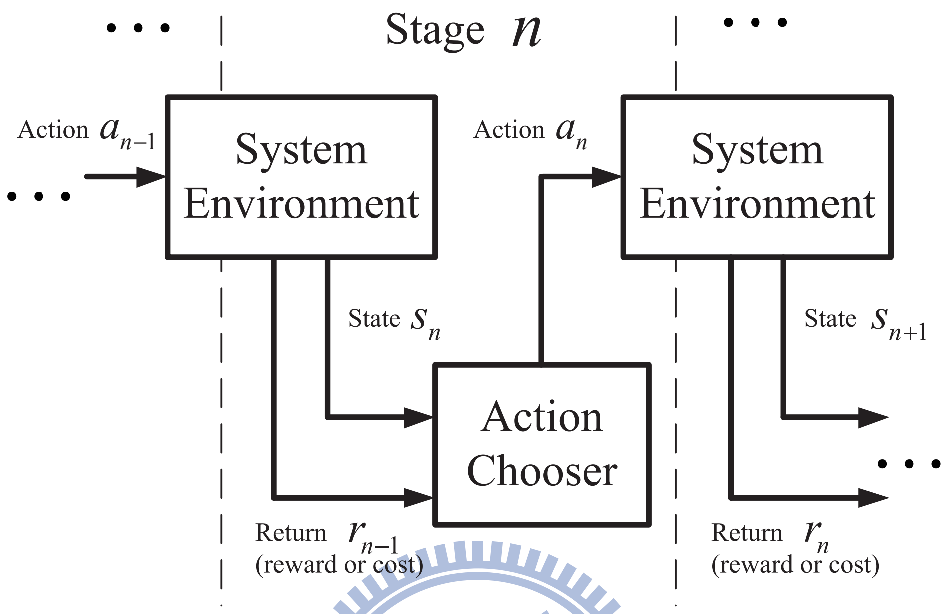


Figure 2.6: The sequential decision making model.

quired because of the probabilistic nature of the markov decision process. In general, if the system environment were deterministic, the optimal sequence actions could be estimated by either forward or backward induction because expectations need not to be computed. However, all actions depend on future behavior, and only the history is known for forward recursion. Therefore, when using dynamic programming, it is necessary to consider if forward or backward recursion is suited to the optimization problem.

In order to illustrate how markov decision process model is applied in the stochastic dynamic programming, Fig. 2.7 shows the two-state system diagram with the current state $s_n = s_{(1)}$. In this diagram, two actions have been allowed in the first state $s_{(1)}$. If the action chooser selects action "1" ($a_n = 1$, the solid line), then the transition from state $s_{(1)}$ to state $s_{(1)}$ will be influenced by the probability $p_r(s_{(1)}|s_{(1)}, 1)$, the transition from state $s_{(1)}$

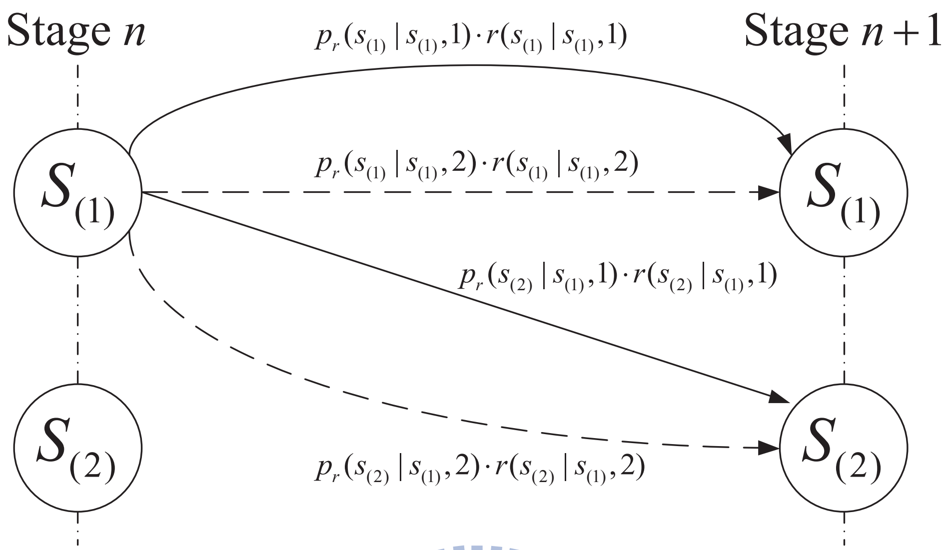


Figure 2.7: The two-state system diagram with the current state $s_n = s_{(1)}$.

to state $s_{(2)}$ will be influenced by the probability $p_r(s_{(2)}|s_{(1)}, 1)$. The returns related to these transitions are $r(s_{(1)}|s_{(1)}, 1)$, $r(s_{(2)}|s_{(1)}, 1)$. If the second action "2" in state $s_{(1)}$ is chosen ($a_n = 2$, the dashed line), then $p_r(s_{(1)}|s_{(1)}, 2)$, $p_r(s_{(2)}|s_{(1)}, 2)$ and $r(s_{(1)}|s_{(1)}, 2)$, $r(s_{(2)}|s_{(1)}, 2)$ would be the relevant transition probabilities and returns, respectively. Using the same concepts, the relevant transition probabilities and returns in the second state $s_{(2)}$ can be defined. The corresponding system diagram with the current state $s_n = s_{(2)}$ is shown in Fig. 2.8.

In order to find the optimal sequence actions, we define $u_n(s_n)$ as the total expected returns at n stage starting from state s_n if an optimal policy is followed. If the returns are rewards, the definition of the total expected return through backward recursion is shown as follows.

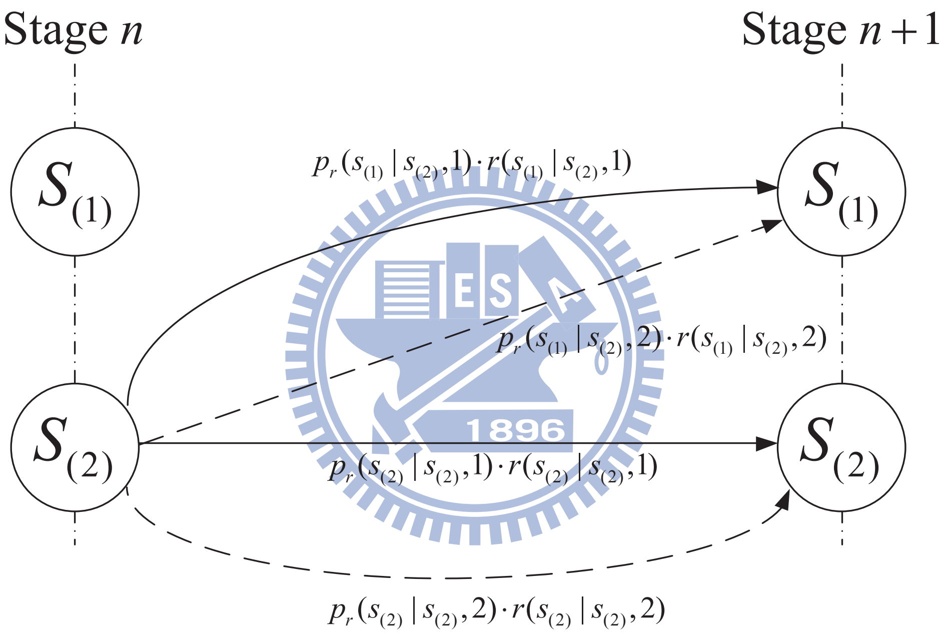


Figure 2.8: The two-state system diagram with the current state $s_n = s_{(2)}$.

$$u_n(s_n) = \max_{a_n \in A}\{ \sum_{s_{n+1} \in S} p_r(s_{n+1}|s_n, a_n) \cdot [r(s_{n+1}|s_n, a_n) + u_{n+1}(s_{n+1})]\} \ , \quad (2.3)$$

where n is from 0 to N (the number of stages) and

$$r(s_n, a_n) = \sum_{s_{n+1} \in S} p_r(s_{n+1}|s_n, a_n) \cdot r(s_{n+1}|s_n, a_n) \ . \quad (2.4)$$

We can use (2.4) to simplify (2.3), then get

$$u_n(s_n) = \max_{a_n \in A}\{r(s_n, a_n) + \sum_{s_{n+1} \in S} p_r(s_{n+1}|s_n, a_n) \cdot u_{n+1}(s_{n+1})\} \ . \quad (2.5)$$

The optimal control policy will be

$$a_n^*(s_n) \in \arg\max_{a_n \in A}\{r(s_n, a_n) + \sum_{s_{n+1} \in S} p_r(s_{n+1}|s_n, a_n) \cdot u_{n+1}(s_{n+1})\} \ , \quad (2.6)$$

where these policies can be represented by vector $a^* = (a_0^*, a_1^*, \ldots, a_{N-1}^*)$. The utilization of the recursive equation (2.6) will inform the action chooser which action to use in each state s_n at each stage n, and make overall system with the maximum expected future rewards at each stage of the process by using policy vector a^*. In Chapter 4, we will define all the elements for our problem based on the definitions of markov decision processes, and investigate how to optimally find the sequence actions with an objective of minimizing the power consumption for location updates by using stochastic dynamic programming (or called markov decision process) and backward recursion.

Chapter 3

System Models

3.1 System Architecture

We consider a system architecture which is based on the distance-based location reporting scheme. Each mobile device moves with the specific movement (e.g. fixed speed or random behavior). As it was mentioned in Chapter 2, the fundamental concept based on the distance-based location reporting scheme is that each mobile device essentially reports locations with a certain distance. The system architecture consists of the following elements.

Location Error Tolerance d_{th}

In general, each location-based application server must have the the required location error tolerance. The selected location error tolerance determines the location report threshold d_{th}, which must not be exceeded by the difference of the two location reports. As a designer of location update scheme, we must also guarantee that each location calculation is within this tolerance value.

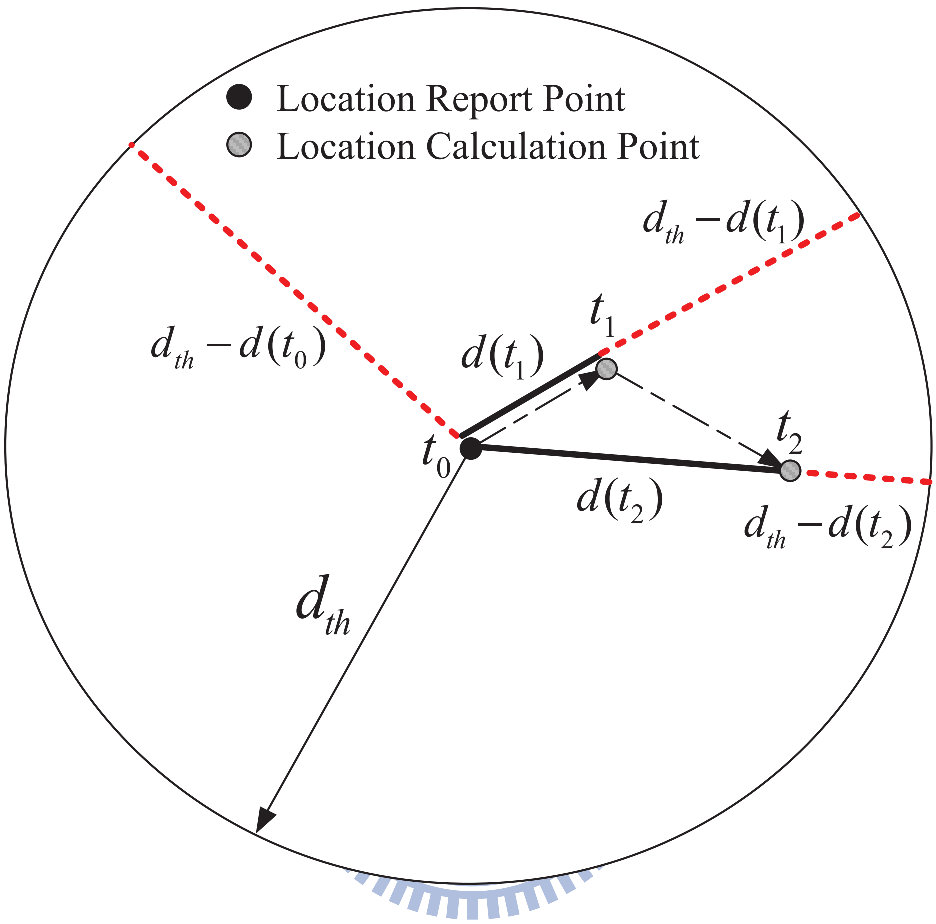


Figure 3.1: The system architecture.

Next Location Calculation Period

As it was mentioned in the introduction for location error tolerance, we must also guarantee that each location calculation is within the required location error tolerance d_{th} from each location-based application server. We refer to the DBLR scheme [5] to decide the next location calculation period based on the current location in order to satisfy the above requirement.

Figure 3.1 shows the system architecture and how the DBLR scheme decides the next location calculation period based on the current location. At every time point t_j (the time point of the j-th location calculating a location since last reporting a location) for calculating a location, we can get the current location $loc(t_j)$ and calculate the distance $d(t_j)$ from the last reporting a location (i.e. $d(t_j) = |loc(t_j) - loc(t_0)|$). Hence, we can know how far it is from the current location to the location report threshold (i.e. d_{th}- $d(t_j)$). Subsequently, if we get the above parameters and the maximum moving speed v_{max} of each device, we can decide the next location calculation period to guarantee that each location calculation is within the location error tolerance. Referring to [5], the formula of the next location calculation period $T^{(j)}$ (the j-th location calculation period after last reporting a location) is shown as follows:

$$T^{(j)} = \frac{d_{th} - d(t_{j-1})}{v_{max}} \ . \tag{3.1}$$

Location Report Condition d_{up}

At each mobile device, calculating a new location from the location sensor always requires an amount of time T_{loc} before the location is obtained. Due to the requirement for guaranteeing location error tolerance d_{th}, the DBLR scheme [5] report condition must be at least

$$T^{(j)} < T_{loc} \ . \tag{3.2}$$

However, by using equation (3.1), the equation (3.2) can be simplified to

$$\begin{aligned} \frac{d_{th} - d(t_{j-1})}{v_{max}} &< T_{loc} \ , \\ d(t_{j-1}) &> d_{th} - v_{max} \cdot T_{loc} \ . \end{aligned} \tag{3.3}$$

Consequently, the location report condition d_{up} for the distance-based location reporting scheme is

$$d(t_{j-1}) > d_{up} \ ,$$

where

$$d_{up} = d_{th} - v_{max} \cdot T_{loc} \ . \tag{3.4}$$

3.2 Hovering Around Border Effects

In the previous section, we guaranteed location error tolerance d_{th} by using equation (3.4). Its fundamental concept is that using the next proper location calculation period which is defined in (3.1) to guarantee that the corresponding distance $d(t_j)$ for each calculating location t_j is within location error tolerance d_{th}. However, the main drawback of (3.1) is that when the mobile device comes to the location report condition d_{up}, the frequency of location calculation increases because the next location calculation period becomes shorter. Consequently, when the mobile device approaches to the location report condition, the power consumption may increase rapidly. We call this phenomenon *the power consumption problem of hovering around border*. The phenomenon is also mentioned in [5]. This effect causes the mobile device to drain the battery rapidly. In [5], a proposed method, called "early distance-based reporting" scheme, solves this problem by using some optimization solutions. Nevertheless, this approach can not accurately solve the problem in the stochastic system (or called prediction-based system). In the following chapters, we will introduce how our proposed approach solves the power consumption problem of hovering around border in both the deterministic system and the stochastic system.

3.3 Performance Metrics

In this part, we define three primary performance metrics for three different location update schemes (TBLR, DBLR and EBLR). In our problem, we are concerned about the overall power consumption and location errors for location update schemes. Their definitions are shown as follows.

3.3.1 Average Power Consumption $\overline{P} = \overline{P_c} + \overline{P_r}$

We are concerned to calculate the average power consumption as the overall power consumption for different location update schemes. We use the average power consumption (3.5) rather than the definition in [5] because the observation concept in our proposed approach is the "stages" instead of the "time". Therefore, the average power consumption in our definition is the average value of each immediate power consumption at each stage. The corresponding definition is shown as follows.

$$\overline{P} \triangleq \frac{The\ sum\ of\ power\ consumption}{The\ number\ of\ stages} = \frac{\sum_{n=1}^{N} c(n)}{N}\ , \tag{3.5}$$

where $\overline{P_c}$ is the average power consumption of location calculation, and $\overline{P_r}$ is the average power consumption of location report. Furthermore, N is the number of stages, and $c(n)$ is the cost function at n-th state transition. In other words, the average power consumption is the average value of each immediate cost function.

On the other hand, the computation power is the other design issue for mobile devices. However, in this thesis we do not consider the computation power of our algorithm. We assume that required energy for energy complexity [17] [18] of markov decision process in one-time execution is much smaller than location updates.

3.3.2 Average Location Error $\overline{L_e}$

Besides the power consumption, we consider the magnitude of location errors. The first consideration is the magnitude of the average location error. This performance metric represents the perspective of average value on location errors for different location update schemes. We observe all location errors within the specified observation stage N. Hence, the average location error can be calculated from all location errors. The corresponding definition is shown as follows.

$$\overline{L_e} \triangleq \frac{The\ sum\ of\ location\ errors}{The\ number\ of\ location\ errors} = \frac{\sum_{i=1}^{M} L_e(i)}{M} \ , \tag{3.6}$$

where

$$L_e(i) = |loc_{at\_server}(i) - loc_{real}(i)| \ , \tag{3.7}$$

note that $loc_{at\_server}(i)$ is the location at application server and $loc_{real}(i)$ is the real location of the mobile device. M is the number of location errors. Each location error is obtained by sampling both $loc_{at\_server}(i)$ and $loc_{real}(i)$ every specific time (we use 0.1 seconds).

3.3.3 Maximum Location Error L_{e_max}

The other consideration is the magnitude of the maximum location error. This performance metric represents the perspective of maximum value on location errors for different location update schemes. It is also an index on location errors for the worst value. Based on these samples of location error, we can obtain the magnitude of the maximum location error for each trial. The corresponding definition is shown as follows.

$$L_{e\_max} = \max_{i=1,2,\ldots,M} \{L_e(i)\} \ . \tag{3.8}$$

Based on the above definitions for performance metrics, we will apply them in the following chapters and investigate the corresponding results by using simulations.

Chapter 4

Proposed Energy-Efficient Location Update Mechanism

In this chapter, we introduce a power-aware location update scheme, called energy-based location reporting (EBLR), which applies dynamic programming to the traditional distance-based location reporting technique to determine the optimal location update scheme with an objective of minimizing power consumption in mobile computing. Through dynamic programming, the EBLR can energy-efficiently change the frequency of location calculation and location report at every decision time point (stage) in any system. The corresponding contents of our proposed approach are presented in the following sections.

4.1 Assumptions

For stochastic dynamic programming, the deterministic dynamic programming can be said by the specific case of markov decision processes. Consequently, in this thesis, we only introduce general case (i.e. the stochastic case) for our proposed approach. In the stochastic environment, location-based applications only know the statistics of the system. The statistics of the system may be the probability distributions of the terminal moving directions or moving speeds. By using the probability distributions of statistics, the corresponding state transition probabilities may be obtained. Our stochastic system is based on the above conditions. We assume that the location-based application can obtain the probability distributions of statistics. Furthermore, we also assume that the cost function and the transition probability are stationary (In a word, their value or distribution are time-invariant). A simple stochastic system is described in Fig. 4.1. The terminal device moves with the fixed speed v but with random forward angle θ.

In the next section, we will use a simple example to illustrate the following definitions for optimal energy-efficient location update scheme.

4.2 Definitions

We define the state presentations, actions, transition probabilities, cost functions, and the principle of optimality for our problem.

4.2.1 System States

In the DBLR scheme, we know that the next location calculation period is related to the distance $d(t_j)$ ($d(t_j) = |loc(t_j) - loc(t_0)|$). Hence, we use the distance $d(t_j)$ to represent each system state in the EBLR scheme because our proposed approach, the EBLR scheme, refers to the next location calculation period in the DBLR scheme. However, it is difficult to definitely separate discrete states because of the continuity of distance. Therefore, we use the multiples of the shortest location calculation period T_{loc} (or called the

$$d' = [(d + v \cdot T_{next} \cdot \cos\theta)^2 + (v \cdot T_{next} \cdot \sin\theta)^2]^{\frac{1}{2}}$$

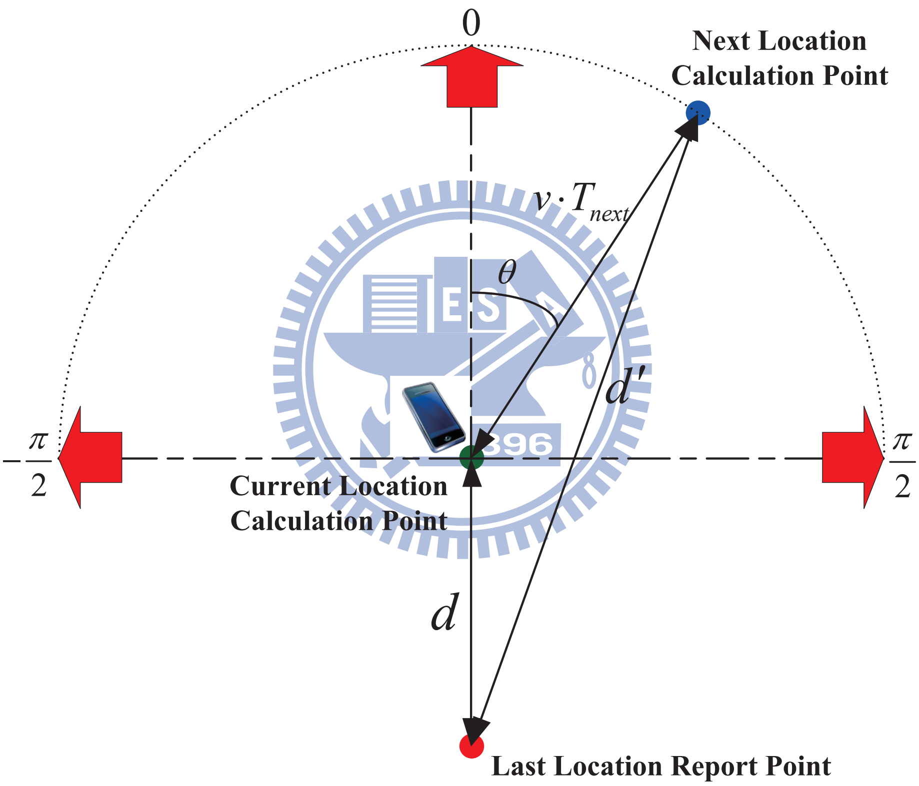


Figure 4.1: An illustrative example of stochastic system.

Table 4.1: The System Parameters for Illustration.

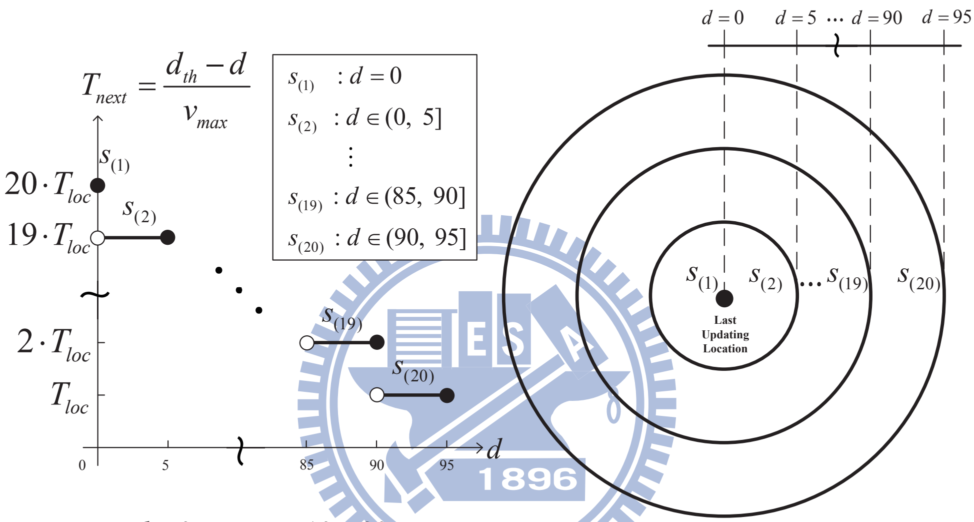
Parameter	Value
Required Time for Calculating a Location, T_{loc}	**0.5 seconds**
Maximum Speed, v_{max}	**10 m/s**
Report Threshold, d_{th}	**100 meters**

required time of calculating a location) as the discrete states. It is beneficial that the continuous distance can be definitely separated into discrete states. Our approach is to use this concept to represent each state.

We use a simple example to illustrate how we define our system states. The system parameters are shown in Table 4.1 for illustration. For example, in Fig. 4.2, the range of distance is divided up into twenty states ($s_{(1)}$, $s_{(2)}$,. . ., $s_{(19)}$ and $s_{(20)}$) based on the system parameters in Table 4.1. In addition, each state has an individual range of distance. For any distance in the same state, all the next location calculation periods will be the same by using the minimum location calculation period (e.g. the minimum location calculation period is $19 \cdot T_{loc}$ in the state $s_{(2)}$, then $T_{next} = 19 \cdot T_{loc}$ for any distance in the state $s_{(2)}$). The corresponding next location calculation period T_{next} which can be obtained by using equation (3.1). They are shown in Table 4.2.

From the above calculations, we can also define the state transitions in this stochastic system example. The state transition diagrams are shown in Fig. 4.3.

In Fig. 4.3, when the decision maker chooses action a at the current stage n, the corresponding immediate cost function (4.8) is $c(s_n, a)$ and the corresponding state transition probability is $p_{s_n s_{n+1}}(a)$. In the following section, we will define the corresponding state transition probability for optimal



$$d = 0 \ : \ T_{next} = 10 = 20 \cdot T_{loc}$$
$$d = 5 \ : \ T_{next} = 9.5 = 19 \cdot T_{loc}$$
$$\vdots$$
$$d = 90 : \ T_{next} = 1 = 2 \cdot T_{loc}$$
$$d = 95 : \ T_{next} = 0.5 = T_{loc}$$

Figure 4.2: The state illustration based on system parameters.

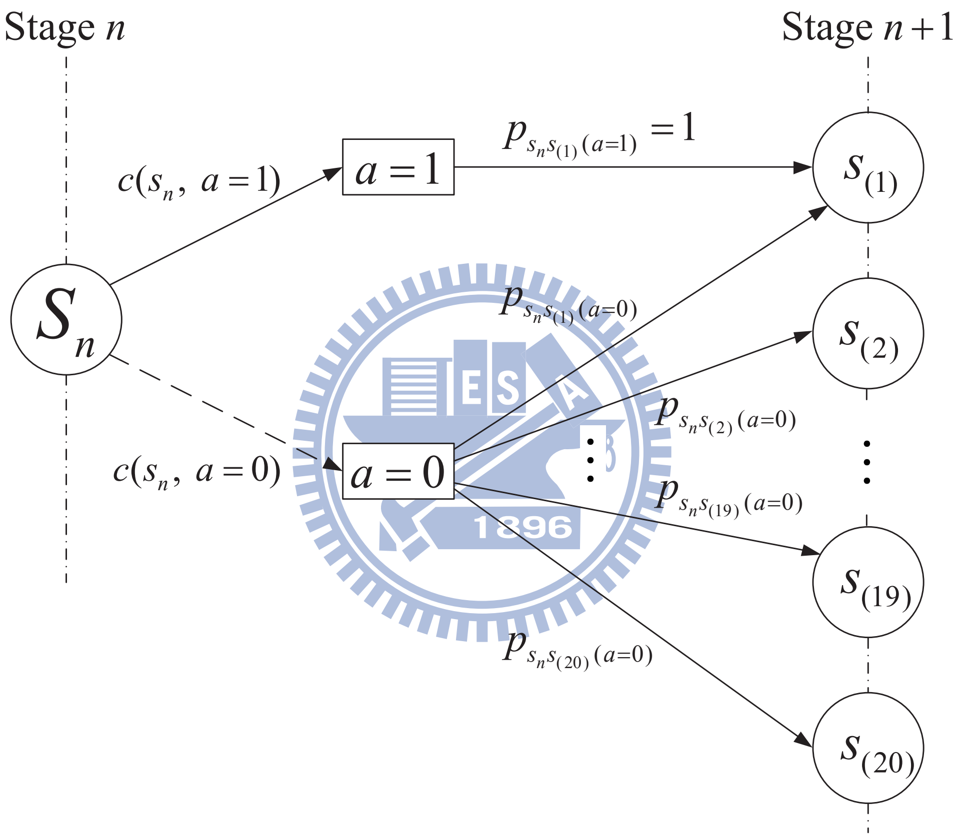


Figure 4.3: The state transition diagrams.

Table 4.2: All States.

State	The Range of Distance (m)	Next Location Calculation Period
s	d	$T_{next}(s)$
$s_{(1)}$	$d = 0$	$20 \cdot T_{loc} = 10$ s
$s_{(2)}$	$d \in (0, 5]$	$19 \cdot T_{loc} = 9.5$ s
$\vdots$	$\vdots$	$\vdots$
$s_{(19)}$	$d \in (80, 85]$	$2 \cdot T_{loc} = 1$ s
$s_{(20)}$	$d \in (90, 95]$	$T_{loc} = 0.5$ s

energy-efficient location update.

4.2.2 Action Representations

In our scenario, we only have two actions to choose at each decision time point (stage). One is the action which represents that the mobile device will only calculate the current location at the next location calculation time point (stage), denoted by symbol "0". The other is the action which represents that the mobile device will both calculate and report the current location at the next location calculation time point (stage), denoted by symbol "1".

4.2.3 Transition Probabilities

For the stochastic system, $p(j|i, a)$ denotes the probability that the system is in state j at the next stage when the decision maker chooses action a in state i at the current stage. Here, we define the corresponding transition probabilities $p(j|i, a)$ for the stochastic system shown in Fig. 4.1. They are shown as follows:

1. When the chosen action $a = 1$ (calculating and reporting the current

location)

$$p(j|i, a=1) = \begin{cases} 1 & , \quad j = s_{(1)} \text{ and } i \in S \\ 0 & , \quad \text{otherwise} \end{cases} \quad . \tag{4.1}$$

2. When the chosen action $a = 0$ (only calculating the current location)

$$p(j|i, a=0) = p(B_{lower}(j) < d^{'} \leq B_{upper}(j)) \quad , \tag{4.2}$$

where

$$d^{'} = \sqrt{[d + v \cdot T_{next}(i) \cdot Cos\theta]^2 + [v \cdot T_{next}(i) \cdot Sin\theta]^2} \quad , \tag{4.3}$$

and

$$d \sim U(B_{lower}(i), B_{upper}(i)) \quad , \quad \theta \sim U(-\pi/2, \pi/2) \quad . \tag{4.4}$$

$T_{next}(s)$ is the next location calculation period when the state is s. $B_{lower}(s)$ is the lower boundary of distance range when the state is s. By contrast, $B_{upper}(s)$ is the upper boundary of distance range when the state is s. For example, in Table 4.2, $T_{next}(s_{(20)})$ is 0.5 (s), $B_{lower}(s_{(20)})$ is 90 (m), and $B_{upper}(s_{(20)})$ is 95 (m).

In equation (4.1), we define the transition probabilities for action $a = 1$. Note that as long as the chosen action a is "1" for any state s_n at the current stage n, the state s_{n+1} at the next stage $n+1$ will always be the first state $s_{(1)}$ which represents the point of reporting the current location (i.e. $d = 0$). Otherwise, other transition probabilities are zero.

On the other hand, in equation (4.2), we define the transition probabilities for action $a = 0$. All transition probabilities can be obtained by using (4.2) (4.3) (4.4) and integral operations. Figure 4.4 illustrates how the transition probability can be obtained. For example, if we are going to obtain the transition probability $p(j = s_{(4)}|i = s_{(2)}, a = 0)$ based on all state definitions in Table 4.2, then the transition probability can be simplified to

$$\begin{aligned} p(j = s_{(4)}|i = s_{(2)}, a = 0) &= p(B_{lower}(s_{(4)}) < d' \leq B_{upper}(s_{(4)})) \\ &= p(10 < d' \leq 15) \ , \end{aligned} \tag{4.5}$$

where

$$d' = \sqrt{[d + v \cdot T_{next}(s_{(2)}) \cdot Cos\theta]^2 + [v \cdot T_{next}(s_{(2)}) \cdot Sin\theta]^2} \ , \tag{4.6}$$

and

$$d \sim U(B_{lower}(s_{(2)}), B_{upper}(s_{(2)})) \ , \ \theta \sim U(-\pi/2, \pi/2) \ . \tag{4.7}$$

By using (4.5) (4.6) (4.7) and integral operations, we can obtain the corresponding transition probability.

4.2.4 Cost Functions

Because the main objective is to minimize the power consumption of calculating and reporting locations at the mobile device, we define that the cost function is the power consumption between each decision time point (stage). As we know that the power consumption between each decision time point (stage) is related to each location calculation period and action. The formula is shown as follows:

$$c(i, a) \triangleq \frac{Immediate\ energy\ consumption}{Next\ location\ calculation\ period} = \frac{\epsilon(a)}{\tau(i)} \ . \tag{4.8}$$

$c(i, a)$ is the immediate cost function when the decision maker chooses action a in state i at the current stage. Furthermore, $\epsilon(a)$ is the energy function of action a and the next location calculation period $\tau(i)$ is the

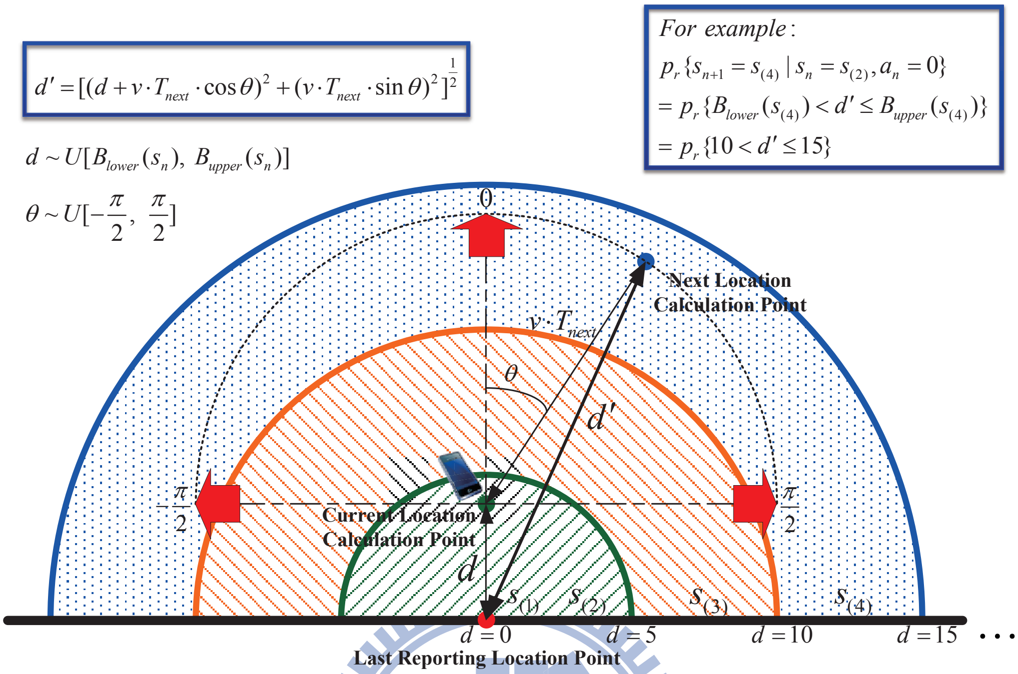


Figure 4.4: An illustrative example of calculating the transition probability.

function of state i. Based on the definition for the action, $\epsilon(a)$ is defined as follows.

$$\epsilon(a) = \begin{cases} E_c & , \quad a = 0 \\ E_c + E_r & , \quad a = 1 \end{cases} . \tag{4.9}$$

Subsequently, $\tau(i)$ is defined as follows.

$$\tau(i) = T_{next}(i), \quad i \in S \ , \tag{4.10}$$

where S is a set for system states. For example, if system states are based on Table 4.2, the corresponding $\tau(i)$ is defined as

$$\tau(i) = T_{next}(i), \quad i \in s_{(1)}, s_{(2)}, \ldots, s_{(19)}, s_{(20)} \ . \tag{4.11}$$

In the concept of our approach, we use the power consumption value at each stage as the immediate cost function. That is why we call our approach the *energy-based location reporting* scheme.

4.3 Optimal Energy-Efficient Location Update Algorithm

In this section, we are going to introduce an optimal energy-efficient location update algorithm. The principle of optimality with DP and the procedure of power consumption minimization are shown as follows.

4.3.1 Principle of Optimality

Recall that we consider all system elements (states, actions, transition probabilities and cost functions), the principle of optimality for the stochastic location update in our proposed approach, the energy-based location reporting scheme, can be identified by using the above system elements.

1. **Recursion Manner (Bellman's Equation)**: Because of the stochastic system, here we can use the backward recursion manner to find the optimal value function $u_n(s_n)$, and the computations are initialized by setting $u_N(s_N) = 0$. Furthermore, a_n is a permissible decision that may be chosen from the set A=0,1, and $n = 0, \ldots, N-1$.

$$u_n(s_n) = \min_{a_n \in A}\{c(s_n, a_n) + \sum_{s_{n+1} \in S} p_r(s_{n+1}|s_n, a_n) \cdot u_{n+1}(s_{n+1})\} \ . \tag{4.12}$$

2. **Optimal Policy**: Any element of *Markov* policy vector a^* satisfies the following equation

$$a_n^*(s_n) \in \arg\min_{a_n \in A}\{c(s_n, a_n) + \sum_{s_{n+1} \in S} p_r(s_{n+1}|s_n, a_n) \cdot u_{n+1}(s_{n+1})\} \ . \tag{4.13}$$

This optimal control policy is called the stochastic *Markov* policy. Through backward recursion, we can find the optimal solution for scheduling the sequence actions (i.e. vector a^*) of each mobile device in the stochastic system.

4.3.2 Procedures of Power Consumption Minimization

In this section, we present the procedure of power consumption minimization by using the optimal energy-efficient location update algorithm with three steps. The algorithm solves optimality equation (4.12) subject to boundary condition $u_N(s_N) = c(s_N)$. The steps for the optimal energy-efficient location update algorithm are shown as follows.

- **Step 1 :** Set $n = N$ and

$$u_N(s_N) = c(s_N), \quad \text{for each } s_N \in S \ ,$$

where $c(s_N)$ is usually set to zero.

- **Step 2 :** Substitute $n - 1$ for n and then compute $u_n(s_n)$ for each $s_n \in S$ by

$$u_n(s_n) = \min_{a_n \in A}\{c(s_n, a_n) + \sum_{s_{n+1} \in S} p_r(s_{n+1}|s_n, a_n) \cdot u_{n+1}(s_{n+1})\} \ .$$

Set

$$a_n^*(s_n) \in \arg\min_{a_n \in A}\{c(s_n, a_n) + \sum_{s_{n+1} \in S} p_r(s_{n+1}|s_n, a_n) \cdot u_{n+1}(s_{n+1})\} \ .$$

- **Step 3 :** If $n = 0$, stop. Otherwise return to step 2.

Figure 4.5 shows the flow chart of optimal energy-efficient location update algorithm. The minimum power consumption can be achieve in this algorithm.

In the following chapter, we will use this algorithm and compare our proposed energy-based location reporting method with other approaches (DBLR and TBLR) in depth by using simulations.

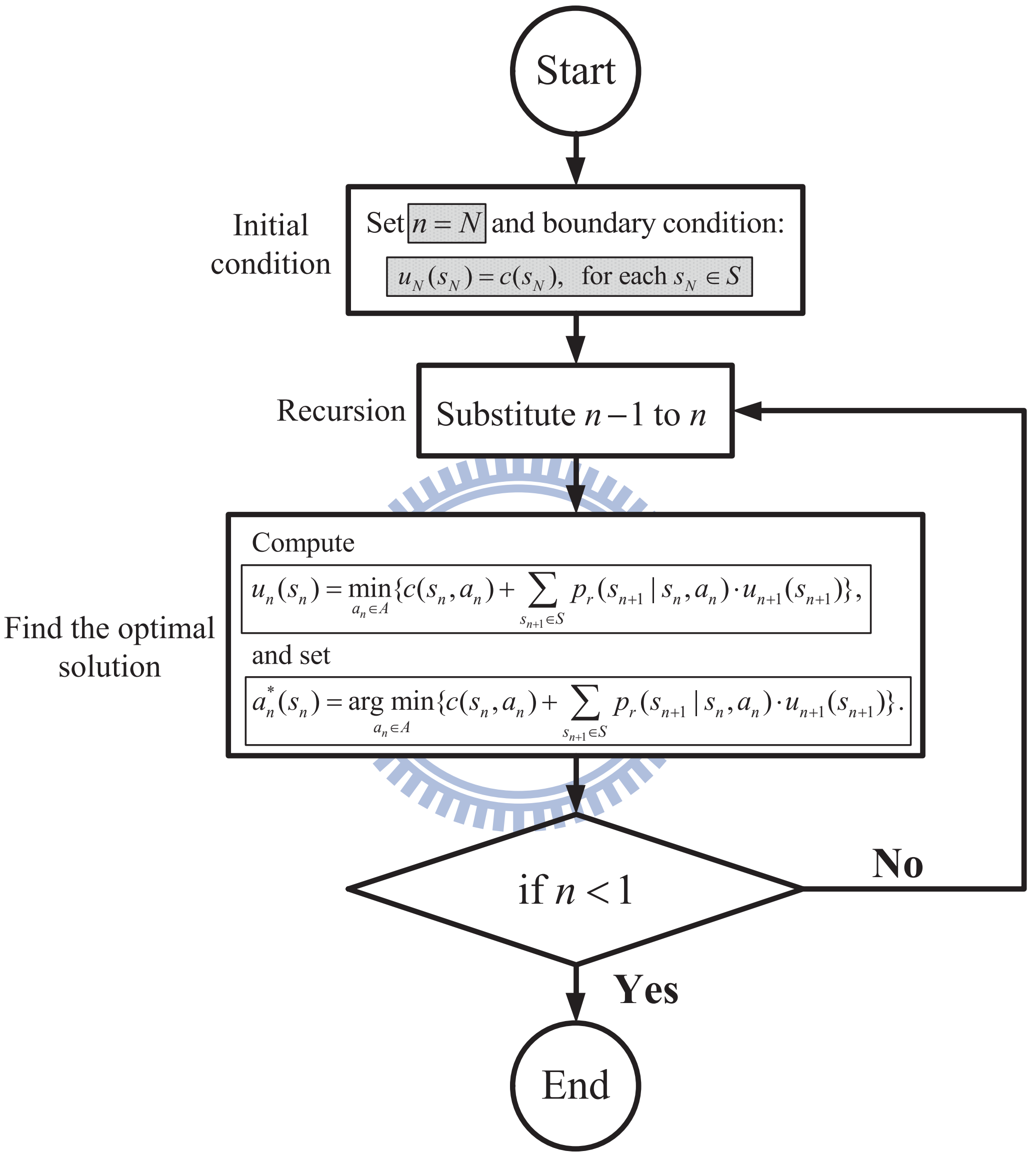


Figure 4.5: The flow chart of optimal energy-efficient location update algorithm.

Chapter 5

Numerical Results

In this chapter, we show numerical results to reveal the importance of the two key impact factors on performance metrics for different location update schemes in the stochastic system. The impact factors consist of (1) the energy ratio E_r/E_c, and (2) the terminal moving speed v. Furthermore, we verify that with our proposed approach the effects of the above factors on the overall system power consumption can be decreased by implementing optimal energy-efficient location update algorithm.

5.1 Simulation Setup

For the stochastic system, we simulate the terminal moving environment with the given speed v and the random forward angle θ. The random forward angle θ is a random variable with the uniform distribution from $-pi/2$ to $pi/2$. Moreover, the terminal moves with the fixed moving speed but goes forwards with the random angle. The terminal moving environment is shown in Fig. 4.1. We consider three location update schemes, including (1) the time-based location reporting scheme (TBLR), (2) the distance-based location

Table 5.1: The System Parameters.

Parameter	Value
Required Time for Calculating a Location, T_{loc}	**0.5 seconds**
Maximum Speed, v_{max}	**10 m/s**
Speed, v	**Fixed (1 m/s to 10 m/s)**
Moving Direction	**Random angle** $\theta \sim U(-\pi/2, \pi/2)$
Report Threshold, d_{th}	**100 meters**
Report Condition, d_{up}	**95 meters**
Energy for Calculating a Location, E_c	**75 mJoules**

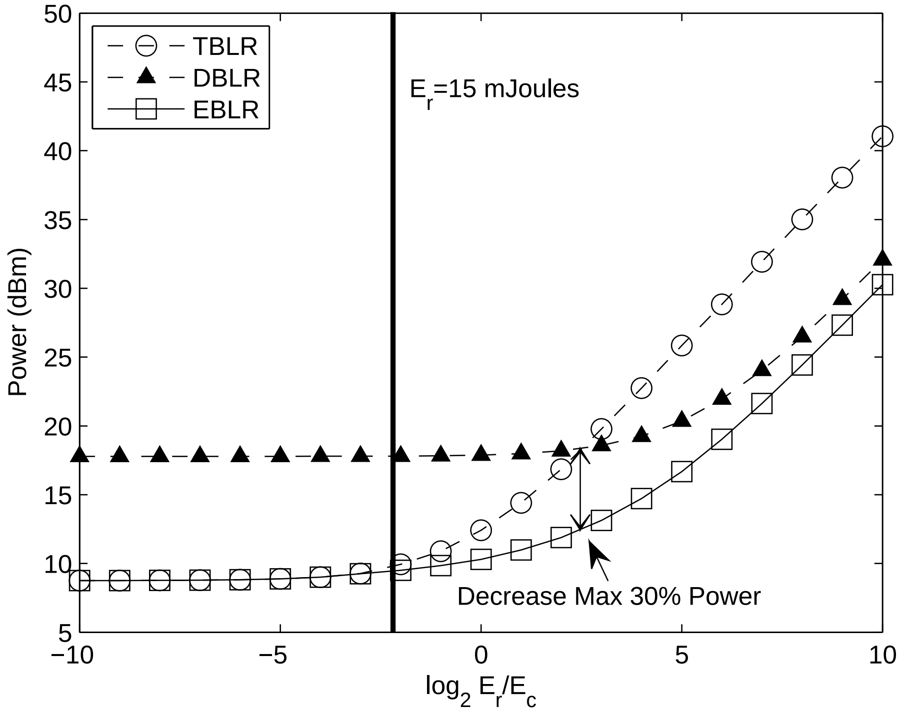
reporting scheme (DBLR), and our proposed approach (3) the energy-based location reporting scheme (EBLR). Furthermore, we use the system parameters in Table 5.1 except that the energy E_r of reporting locations and the terminal moving speed v are set in different values for different conditions. The trial times for simulation are set to be bigger than 100 times. On the other hand, the number of stages N is set to be bigger than 100.

5.2 Effects of Energy Ratio on Power Consumption for Various Location Update Schemes with Different Terminal Moving Speeds

Firstly, in the stochastic system we investigate the effects of the energy ratio E_r/E_c on power consumption for various location update schemes with different terminal moving speeds v. In general, different energy ratios satisfy different amounts of reporting information. For example, if the mobile device sends the location data which is like the format [+24.789462,+121.000113], it needs to at least consume about 15 mJoules energy for transmitting 182 bits location data [19]. Hence, different energy ratios represent that they are applied in different location-based applications. In order to evaluate the performance on power consumption for various energy ratios and different terminal moving speeds, we consider that the value range of the energy ratio is 2^{-10} to 2^{10} and the value range of the terminal moving speed is 1 (m/s) to 10 (m/s).

Figure 5.1 shows how the terminal moving speed and the energy ratio affect the overall system power consumption in the stochastic system. We use the general energy ratio value (i.e. The line for $E_r = 15$ mJoules) to observe the effect on each curve. We observe that in each subfigure the overall system power consumption increases with the energy ratio E_r/E_c for location updates when the energy ratio is bigger than the general energy ratio value. Moreover, each subfigure represents the effect of the energy ratio on power consumption for various location update schemes with a certain terminal moving speed. Figure 5.1 also shows that the energy E_r of reporting location significantly affects the power consumption for various location update schemes. As the energy ratio increases, the power consumption increases due to the increasing energy of location report.

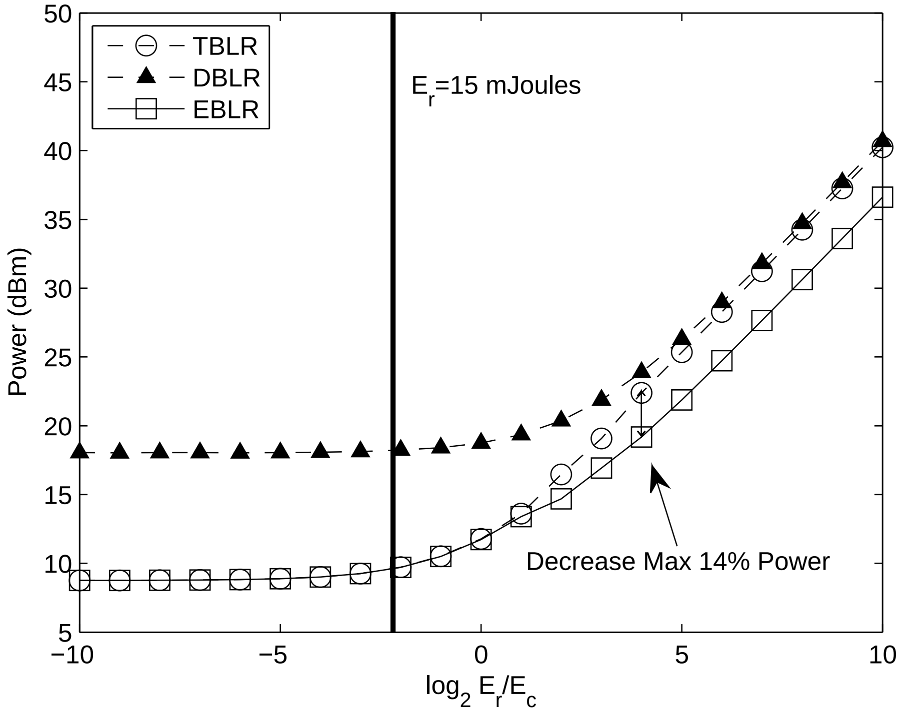
Nevertheless, the energy ratio E_r/E_c effects location update schemes differently on power consumption with various terminal moving speeds. In Fig. 5.1(a), the DBLR scheme consumes less power than the TBLR scheme does when the energy of location report increases. It is because that the frequency of location report with the DBLR scheme decreases when the terminal moving speed decreases. As the terminal moving speed increases, in Figs. 5.1(b) and 5.1(c), the DBLR scheme consumes more power than other



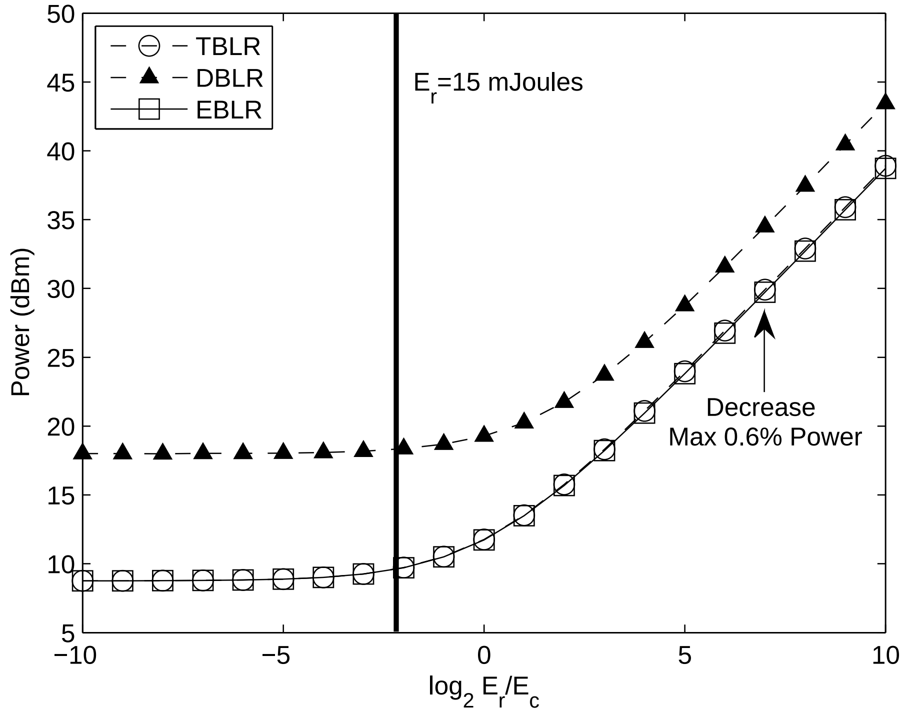
(a) The terminal moving speed $v = 1$.

schemes because the frequency of both reporting and calculating location increases. However, as the terminal moving speed achieves maximum speed v_{max}, in Fig. 5.1(d), all location update schemes use the same period of reporting and calculating location.

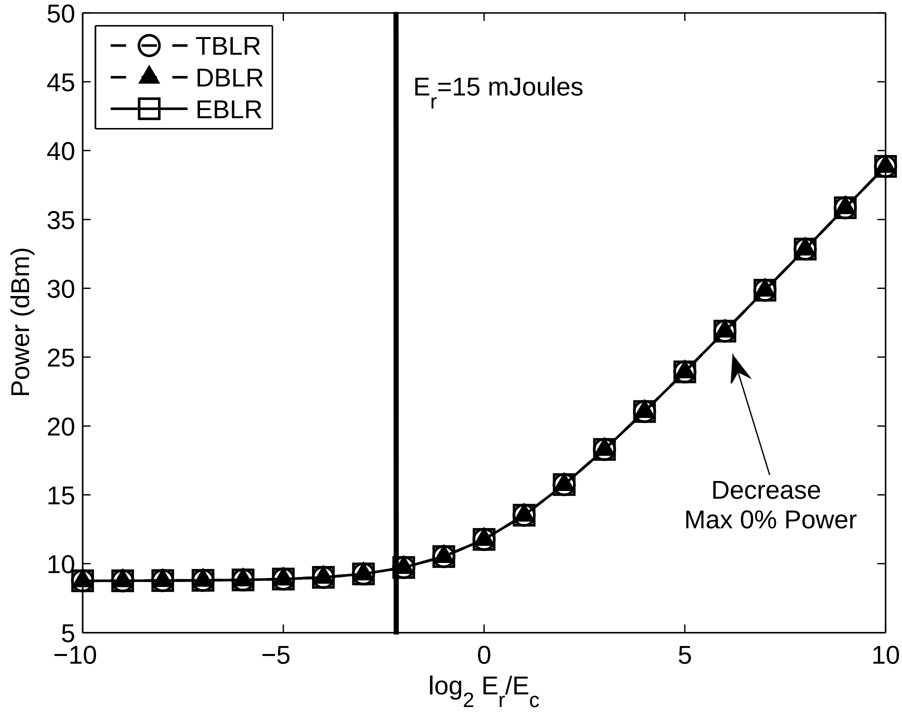
Due to the use of stochastic dynamic programming (4.12) (4.13), in the case of any terminal moving speeds the EBLR scheme can dynamically and optimally select actions which consume less power by using the statistics of system (i.e. state transition probabilities). That is the reason why the EBLR scheme which uses the stochastic dynamic programming is always the best solution when the energy ratio varies for location updates. Moreover, in this simulation we prove that in EBLR the maximum saving power can



(b) The terminal moving speed $v = 4$.



(c) The terminal moving speed $v = 7$.



(d) The terminal moving speed $v = 10$.

Figure 5.1: Effects of the energy ratio on the overall system power consumption for various location update schemes with different terminal moving speeds.

be about 30 percent which is compared to the other better scheme. We also observe that the EBLR is suitable for solving the power consumption problem of hovering around border (i.e. the terminal moving speed is lower). In other words, the EBLR does not have obvious advantages when the terminal moving speed is higher (the maximum saving power is close to 0 percent).

5.3 Effects of Terminal Moving Speeds on Location Errors for Various Location Update Schemes with Different Energy Ratios

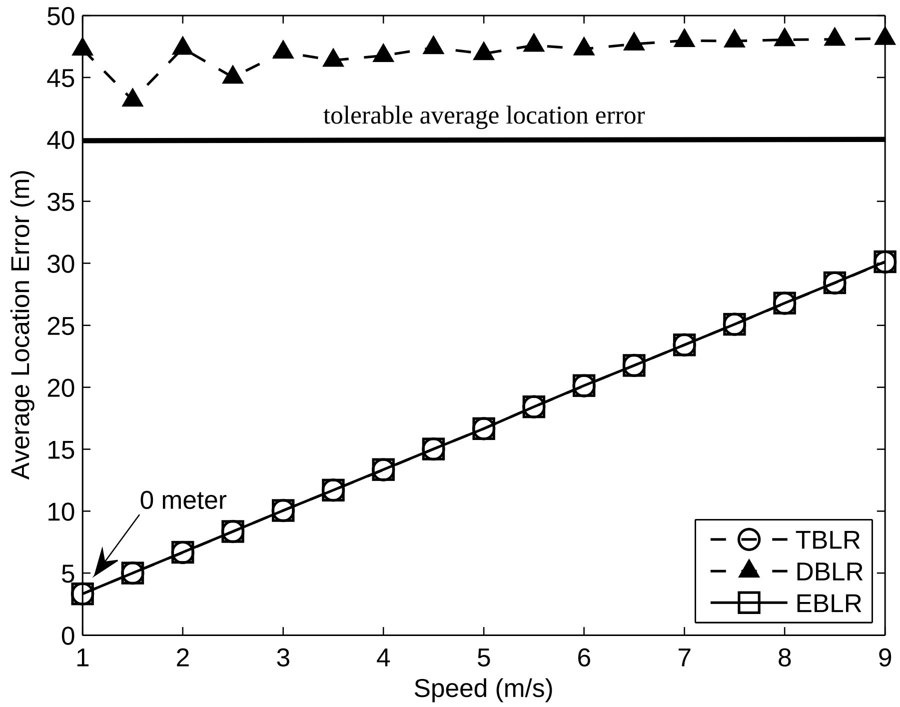
In this part, we are going to evaluate other performance metrics, location errors, including (1) average location errors and (2) maximum location errors for location updates. Likewise, we also consider three different location update schemes, including TBLR, DBLR, and EBLR, then investigate the effects of the terminal moving speeds v on power consumption for various location update schemes with different energy ratio E_r/E_c. Moreover, as it was mentioned in Chapter 1, we use the minimum city block distance (i.e. it is 80 meters) as the maximum tolerable location error for our application, location-based social network. On the other hand, we straightforwardly use a half of the minimum city block distance (i.e. it is 40 meters) as the average tolerable location error.

Average Location Errors

Figure 5.2 illustrates the average location errors related to terminal moving speeds with different energy ratios for location updates. The figure shows some phenomena for various location update schemes. For the DBLR scheme, the average location errors have less variation because the location report condition is based on terminal moving distance d. Even though the terminal moving speed increases, it still do not report a location when the terminal moving distance d does not exceed the location report condition d_{up}. Consequently, the average location errors with the DBLR scheme are always larger than other schemes. We observe that in the DBLR the average location errors are larger than the average tolerable location error (40 meters) for any case. In contrast, the TBLR scheme always reports locations with a fixed period. This effect leads to a result that the average location errors increase when the terminal moving speed increases.

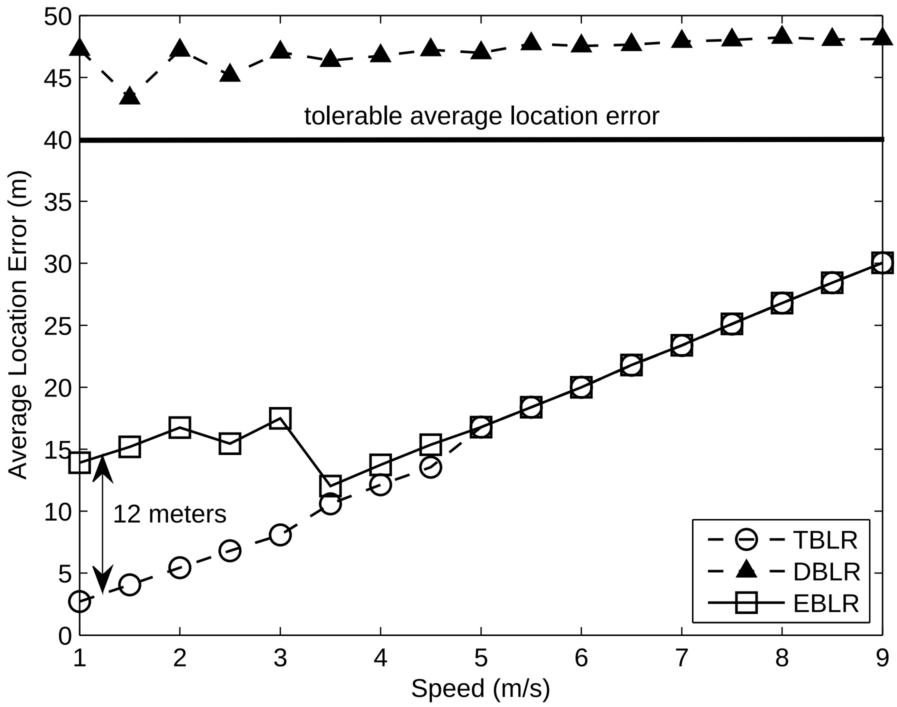
Nevertheless, the EBLR scheme reports locations by using power-aware decision (4.12) (4.13) based on the stochastic dynamic programming. The power-aware approach leads to a result that the optimal decision must select the result of potentially making the longer period for calculating and reporting locations when the terminal moving speed increases. This trends to choose the action "1" which represents that the mobile device will both calculate and report the current location. It implied that the location report condition with the EBLR scheme is similar to the TBLR scheme whenever the terminal moving speed increases. That is the reason why the average location errors with the EBLR scheme are always better than the DBLR scheme but usually worse than the TBLR scheme does in trial samples.

On the other hand, in Figs. 5.2(b) and 5.2(c), for location updates we also observe that with the lower terminal moving speeds, the EBLR scheme has larger average location errors when the energy ratio increases. The reason is that the EBLR scheme trend to optimally choose the sequence actions in order to minimize the expectation of overall system power consumption, but

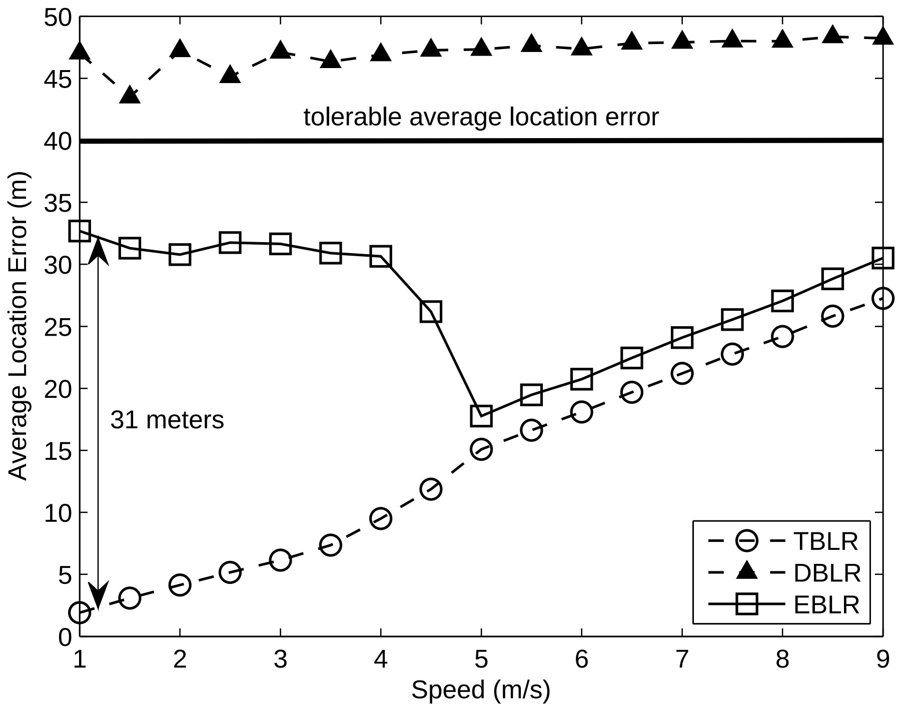


Average Location Error (m)
Speed (m/s)
tolerable average location error
0 meter
TBLR
DBLR
EBLR

(a) $log_2 E_r/E_c = -5$.


50
45
40
35
30
25
20
15
10
5
0
1
2
3
4
5
6
7
8
9
tolerable average location error
Average Location Error (m)
12 meters
TBLR
DBLR
EBLR
Speed (m/s)

(b) $log_2 E_r/E_c = 0$.



(c) $log_2 E_r/E_c = 5$.

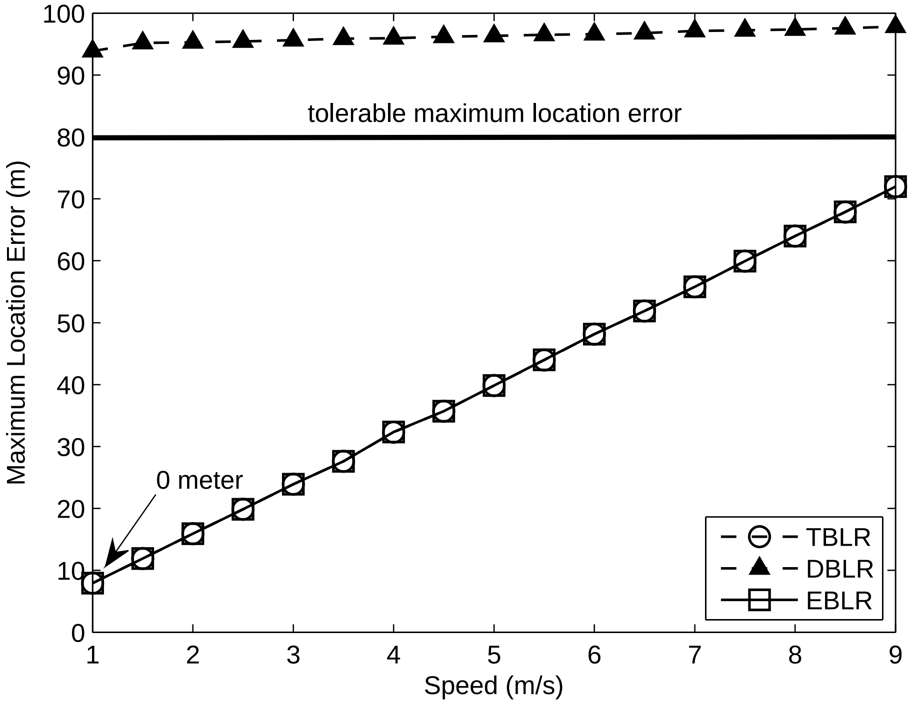
Figure 5.2: Effects of the terminal moving speed on average location errors for various location update schemes with different energy ratios.

sacrifices the better performance on the average location errors. However, in the EBLR we also observe that even through the average location errors will increase when the terminal moving speed is lower, they are still in satisfied level (i.e. they are all smaller than the average tolerable location error, 40 meters) for providing our location-based services. This proves that EBLR can provide a better balance between power consumption and location error than other schemes.

Maximum Location Errors

Figure 5.3 illustrates the maximum location errors related to terminal moving speeds with different energy ratios. For the effects of the terminal moving speed, the simulation results in each location update scheme on the maximum location errors are similar to the average location errors. Likewise, for the DBLR scheme, the maximum location errors have less variation because the location report condition is based on terminal moving distance d. Even though the terminal moving speed increases, it still do not report a location when the terminal moving distance d does not exceed the location report condition d_{up}. Consequently, the maximum location errors with the DBLR scheme are always larger than other schemes. We observe that in the DBLR the maximum location errors are larger than the maximum tolerable location error (80 meters) for any case. In contrast, the TBLR scheme always reports locations with a fixed period. This effect leads to a result that the maximum location errors increase when the terminal moving speed increases.

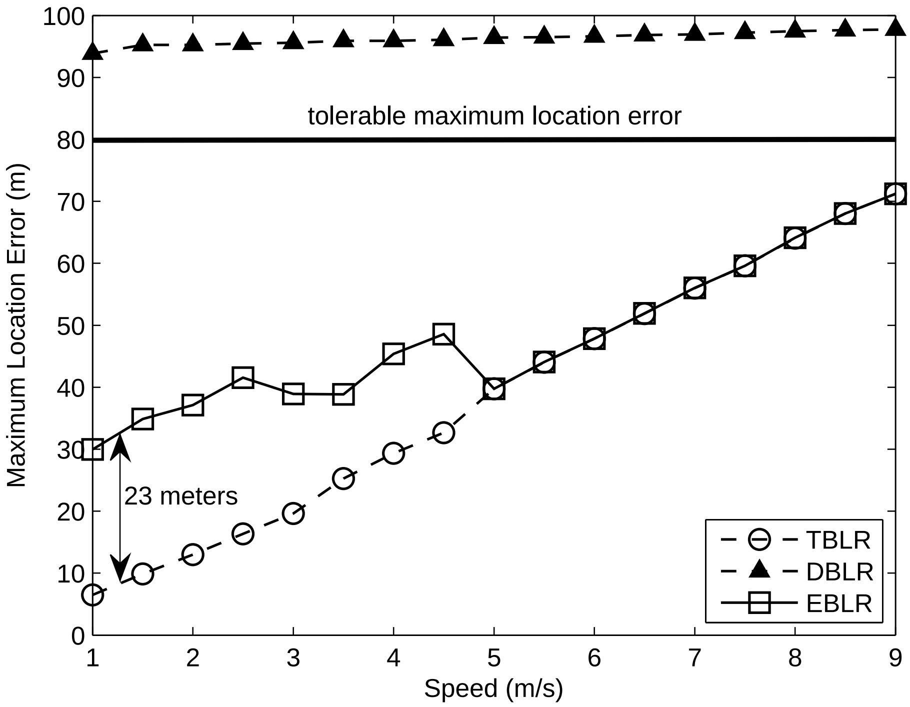
Nevertheless, for the same reason, the EBLR scheme reports locations by using power-aware decision and distance-based location reporting scheme. The power-aware approach leads to a result that the optimal decision must select the longer period for calculating and reporting locations when the



(a) $log_2 E_r/E_c = -5$.

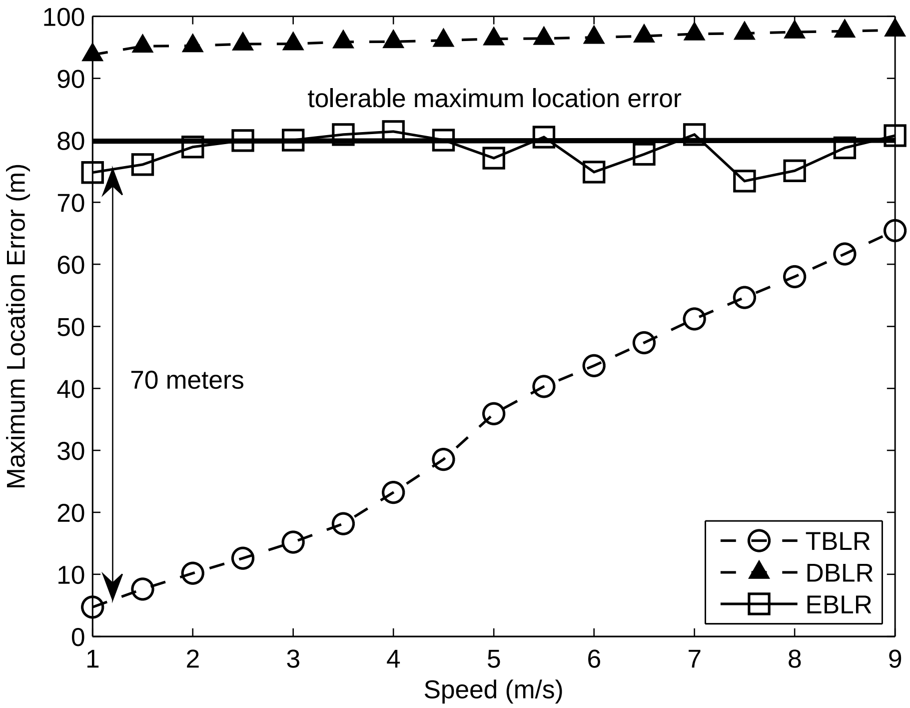
terminal moving speed increases. This trends to choose the action "1" which represents that the mobile device will both calculate and report the current location. It implied that the location report condition with the EBLR scheme is similar to the TBLR scheme whenever the terminal moving speed increases. That is the reason why the maximum location errors with the EBLR scheme are always better than the DBLR scheme but usually worse than the TBLR scheme.

Furthermore, the main difference in simulation results between the average and maximum location errors is the effect of the energy ratio. In Figs. 5.3(b) and 5.3(c), we observe that with the lower terminal moving speeds, the EBLR scheme's maximum location errors are more severely af-



100
90
80
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
9
Maximum Location Error (m)
Speed (m/s)
tolerable maximum location error
23 meters
TBLR
DBLR
EBLR

(b) $log_2 E_r / E_c = 0$.



(c) $log_2 E_r/E_c = 5$.

Figure 5.3: Effects of the terminal moving speed on maximum location errors for various location update schemes with different energy ratios.

fected when the energy ratio increases compared to the average location errors. The maximum location errors increase more severely when the energy ratio increases. It is because that the maximum location errors is the worst value of location errors. However, like the average location errors, in the EBLR we also observe that even through the maximum location errors will increase when the terminal moving speed is lower, they are still in satisfied level (i.e. they are all smaller than the maximum tolerable location error, 80 meters) for providing our location-based services.

Chapter 6

Conclusions

6.1 Energy-Efficient Location Update Mechanism

In this thesis, we present a power-aware location update scheme, called energy-based location reporting (EBLR), which applies dynamic programming (DP) to the traditional distance-based location reporting technique to determine the optimal location update scheme with an objective of minimizing power consumption in mobile computing. In addition, our proposed approach can solve the optimization problem in general system, including deterministic system and stochastic system for power consumption minimization by using dynamic programming. Subsequently, an optimal energy-efficient location update algorithm is proposed. We also develop the procedure of power consumption minimization with the optimal energy-efficient location update algorithm. Through our proposed approach and algorithm, we can address the power consumption problem of hovering around border in the traditional distance-based location reporting technique. Furthermore, we improve performances on both the power consumption and location errors. Numerical results show that the energy-based location reporting scheme has the minimum power consumption.

In numerical results, we observed that the DBLR scheme decreases the frequency of reporting locations, and thus the power consumption of reporting location can be lowered. In contrast, the TBLR scheme uses the frequent period of reporting locations so that the power consumption of reporting locations increases. However, the DBLR scheme has a major drawback - the power consumption problem of hovering around border. This phenomenon leads to a result that the power consumption of calculating locations with the DBLR scheme is larger than other schemes. This influence dominates the total power consumption with the DBLR scheme. In contract with the above schemes, the optimal sequence decisions are chosen with dynamic programming by implementing optimal energy-efficient location update algorithm, the EBLR scheme. Consequently, the total power consumption can be as small as possible. Numerical results show that the EBLR scheme can save about 30 percent power at most compared to the other better scheme. Besides, the value of average location error and maximum location error with the EBLR scheme can be close to the best value which the TBLR scheme has in most of cases. Even through in the EBLR both of the average location error and the maximum location error will increase when the terminal moving speed is lower, they are still in satisfied level (i.e. they are all smaller than the tolerable location error) for providing our location-based services. This proves that EBLR can provide a better balance between power consumption and location error than other schemes.

6.2 Future Research

For the future research of the thesis, we provide the following suggestions to extend our work:

- The complexity of algorithm can be considered, including energy complexity and time complexity.
- Consider the different stochastic environment and observe how it effect the performance metrics.
- Optimize the mode of state separation and find out the corresponding minimum power consumption in stochastic environment for providing location-based services.

Bibliography

[1] T. D'Roza and G. Bilchev, "An Overview of Location-Based Services," *Springer Netherlands BT Technology Journal*, vol. 21, pp. 20–27, 2003.

[2] S. Vaughan-Nichols, "Will Mobile Computing's Future Be Location, Location, Location?" *Computer*, vol. 42, pp. 14 –17, Feb. 2009.

[3] P. Bellavista, A. Kupper, and S. Helal, "Location-Based Services: Back to the Future," *IEEE Pervasive Computing*, vol. 7, pp. 85–89, Apr. 2008.

[4] N. Li and G. Chen, "Sharing Location in Online Social Networks," *IEEE Network*, vol. 24, pp. 20–25, Sept. 2010.

[5] T. Farrell, R. Lange, and K. Rothermel, "Energy-Efficient Tracking of Mobile Objects with Early Distance-Based Reporting," *IEEE International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous)*, pp. 1–8, Aug. 2007.

[6] A. Anand, C. Manikopoulos, Q. Jones, and C. Borcea, "A Quantitative Analysis of Power Consumption for Location-Aware Applications on Smart Phones," pp. 1986 –1991, June. 2007.

[7] M. B. Kjaergaard, "Minimizing the Power Consumption of Location-Based Services on Mobile Phones," *IEEE Pervasive Computing*, vol. 99, p. 1, Apr. 2010.

[8] A. Leonhardi and K. Rothermel, "A Comparison of Protocols for Updating Location Information," *Cluster Computing*, vol. 4, pp. 355–367, Oct. 2001.

[9] N. Deblauwe and P. Ruppel, "Combining GPS and GSM Cell-ID Positioning for Proactive Location-Based Services," *IEEE International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous)*, pp. 1–7, Aug. 2007.

[10] A. Bar-Noy, I. Kessler, and M. Sidi, "Mobile Users: To Update or not to Update?" *IEEE International Conference on Computer Communications (INFOCOM)*, vol. 2, pp. 570–576, June. 1994.

[11] G. Y. Lee and Y. Lee, "Numerical Analysis of Optimum Timer Value for Time-Based Location Registration Scheme," *IEEE Communications Letters*, vol. 6, pp. 431 – 433, Oct. 2002.

[12] Y. Lee and G. Y. Lee, "Optimal Time-Interval for Time-Based Location Update in Mobile Communications," *IEEE Global Communications Conference (GLOBECOM)*, vol. 3, pp. 2079–2083, Nov. 2002.

[13] Inclusion of Uplink TDOA Location Method. [Online]. Available: http://www.3gpp.org/ftp/tsg_ran/tsg_ran/TSGR_14/Docs/PDFs/RP-010822.pdf

[14] City Block. [Online]. Available: http://en.wikipedia.org/wiki/City_block

[15] Martin L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming, 1st.* John Wiley & Sons, Inc., 1994.

[16] Sheldon M. Ross, *Introduction to Stochastic Dynamic Programming: Probability and Mathematical*. Academic Press, Inc., 1983.

[17] R. Jain, D. Molnar, and Z. Ramzan, "Towards a Model of Energy Complexity for Algorithms," *IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 3, pp. 1884–1890, Mar. 2005.

[18] J. N. Tsitsiklis, "Computational Complexity in Markov Decision Theory," *Hellenic European Research on Computer Mathematics and its Applications (HERCMA)*, Sept. 2007.

[19] B. Gedik and L. Liu, "MobiEyes: A Distributed Location Monitoring Service Using Moving Location Queries," *IEEE Transactions on Mobile Computing*, vol. 5, pp. 1384–1402, Oct. 2006.

Vita

Kai-Ping Chang

He was born in Taiwan, R. O. C. in 1987. He received a B.S. in Electronic and Engineering from Chang Gaung University of Technology in 2009. From July 2009 to August 2011, he worked his Master degree in the Mobile Communications and Cloud Computing Lab in the Department of Communication Engineering at National Chiao-Tung University. His research interests are in the field of wireless communications and mobile computing.