

國立臺灣大學電機資訊學院資訊網路與多媒體研究所

碩士論文

Graduate Institute of Networking and Multimedia

College of Electrical Engineering and Computer Science


National Taiwan University

Master Thesis

大規模M2M行動網路之群體位置管理

Group Location Management for Large-scale

Machine-to-Machine Mobile Networking



黃冠銘

Huang Guan-Ming

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

兩年的碩士生涯轉眼間就過去了，又到了畢業的時候。首先我要感謝我的指導老師，林風教授。在兩年的碩士生涯中，林風老師扮演著亦師亦友的角色，讓我學習到做研究的熱情以及態度，並且順利完成了這篇論文。接著要萬分感謝帶著我一起研究的傅懷磊學長，在我迷失方向的時候引領著我走向正確的道路。同時，我也非常感謝林一平教授與楊竹星教授擔任學生的口試委員，並且對本篇論文提供良好的建議，使得這篇論文更加完整。

接著我要感謝與我在這兩年一起奮鬥的實驗室同仁們。啟維學長、家朋學長、有倫學長、亭佑學姊、厚鈞學長、思適學長、家綸學長、宗哲學長、坤豐、百俊、明峰、彥婷、恩豪。特別感謝陪著我一起熬夜做計劃與擔任實驗課助教的振翔。

最後我要感謝我的家人默默的在背後支持著我。

Chinese Abstract

Machine-to-Machine (M2M) 通訊是一種支援Internet of Things (IoT) 應用程式的新通訊架構。爲了使機器(Machine)能夠傳送與接收資料，網路端運行位置管理(Location Management)追蹤機器的位置。然而數以百萬計的機器使得位置管理對Mobile Communication Network (MCN) 產生了大量的信令流量。在這樣大規模的M2M行動網路中，本論文研究3GPP Machine Type Communications (MTC) 所定義的機器移動的群體性。首先，我們定義何謂信令傳輸中的“correlated mobility”，當移動中的機器執行位置更新時。接著我們提出Group Location Management (GLM) 機制解決信令壅塞的問題。在GLM機制中，我們將移動路徑相似的機器們群聚在一起。最後，我們模擬成果顯示GLM機制可以降低機器的註冊信令並且增加佈署M2M通訊在大規模的M2M環境的可行性。

Keywords: 相關移動性 (Correlated Mobility)、位置管理 (Location Management)、M2M通訊 (Machine-to-Machine communications)、MTC通訊 (Machine-type Communications)、信令壅塞 (Signalling congestion)。

English Abstract

Machine-to-Machine (M2M) communications have emerged as a new communication paradigm to support Internet of Things (IoT) applications. To deliver data to and from machines, the network performs Location Management to track machine locations (i.e., Location Area Identity; LAI). Yet this process incurs large signaling traffic to the Mobile Communication Networks (MCN), which is composed of millions to trillions of machines. To address such large-scale M2M mobile networking, this paper investigates the grouping characteristics of machine movement identified in the 3GPP Machine Type Communications (MTC). We first define the “correlated mobility” on the signaling transmissions of moving machines when performing location update. Based on the definition, we propose a Group Location Management (GLM) mechanism to mitigate the signaling congestion problem. In GLM, we group machines based on the similarity of their mobility patterns. Through our performance study, we show how the GLM mechanism can reduce registration signaling from machines and increase the feasibility to deploy M2M communications at a large scale M2M environment.

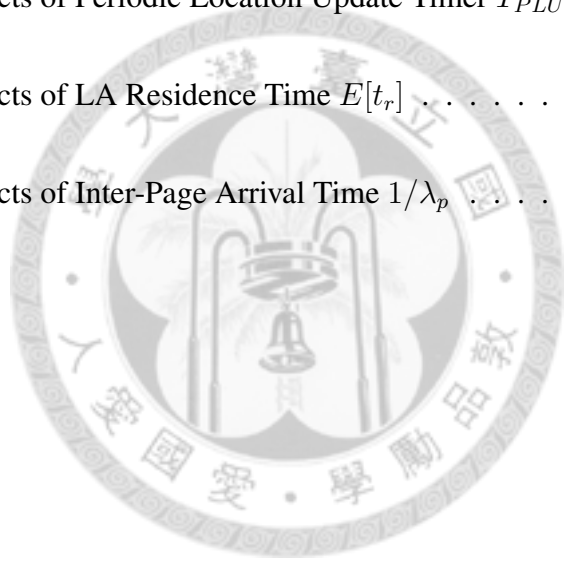
Keywords: Correlated mobility, Location management, Machine-to-Machine communications, Machine-type communications, Signaling congestion

Contents

Acknowledgement	i
Chinese Abstract	ii
English Abstract	iii
1 Introduction	1
2 Group-based Mechanism for Location Management	6
2.1 Correlated Mobility	6
2.2 GLM: Group Location Management	7
2.3 Discussion	13
3 Performance Evaluation	14
3.1 Performance Metrics	14
3.2 Mobility Models	15



<i>CONTENTS</i>	v
3.3 Simulation Models	18
3.3.1 Random Walk Model	20
3.3.2 Bio-inspired Mobility Model	21
3.3.3 Transportation Mobility Model	24
3.4 Simulation Results	28
3.4.1 Effects of K	28
3.4.2 Effects of Periodic Location Update Timer T_{PLU}	29
3.4.3 Effects of LA Residence Time $E[t_r]$	30
3.4.4 Effects of Inter-Page Arrival Time $1/\lambda_p$	30
4 Conclusion	33
Bibliography	35



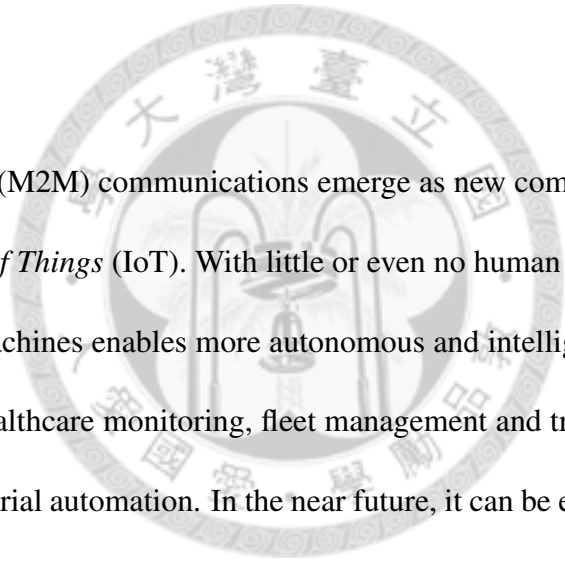
List of Figures

2.1	Flowchart of the LDB part	9
3.1	The 4×4 mesh LA network structure	16
3.2	Flowchart of random walk mobility simulation model.	22
3.3	Flowchart of bio-inspired mobility simulation model.	25
3.4	Flowchart of transportation mobility simulation model.	27
3.5	Effects of K on \mathcal{R} and \mathcal{P} for three mobility models, where $Z_{thres} = 10$, $\epsilon = 2$ minutes, $E[t_r] = 10$ minutes, $1/\lambda_p = 100$ minutes, $T_{PLU} = 60$ minutes.	31
3.6	Effects of T_{PLU} on \mathcal{R} and \mathcal{P} for three mobility models, where $Z_{thres} = 10$, $\epsilon = 2$ minutes, $K = 2$, $E[t_r] = 10$ minutes, $1/\lambda_p = 100$ minutes.	31
3.7	Effects of $E[t_r]$ on \mathcal{R} and \mathcal{P} for three mobility models, where $Z_{thres} = 10$, $\epsilon = 2$ minutes, $K = 2$, $1/\lambda_p = 100$ minutes, $T_{PLU} = 60$ minutes.	32
3.8	Effects of $1/\lambda_p$ on \mathcal{R} and \mathcal{P} for three mobility models, where $Z_{thres} = 10$, $\epsilon = 2$ minutes, $K = 2$, $E[t_r] = 10$ minutes, $T_{PLU} = 60$ minutes.	32



Chapter 1

Introduction



Machine-to-Machine (M2M) communications emerge as new communication paradigms towards the *Internet of Things* (IoT). With little or even no human intervention, the interconnection among machines enables more autonomous and intelligent applications, such as smart metering, healthcare monitoring, fleet management and tracking, environmental monitoring and industrial automation. In the near future, it can be expected that there will be a widespread increase in the amount of machines interconnecting through the cellular infrastructure.

The M2M communications have been realized by the 3GPP working group, namely Machine-Type Communications (MTC) [2, 3]. As defined in 3GPP, there are several unique features for M2M communications, including low mobility, infrequent transmission, and small data transmission [2]. These M2M features are verified through the measurement and characterization of real traffic trace [13], showing that the M2M features are different from human-to-human (H2H) communications. In addition, 3GPP defines

the *group configuration* for M2M communications, i.e., the machines can be grouped together for control and management as long as they share one or more M2M features, and are temporally and/or spatially correlated with each other. Here we illustrate two application scenarios. The first one is the package tracking system, in which the tracking machines for the packages in the same vehicle are highly correlated in terms of time and space. The second one is the intelligent robotic system, swarm-bot [14]. A swarm-bot is a self-assembling and self-organizing artifact, composed of a swarm of mobile robots with the ability to connect to each other. A swarm of mobile robots are required to work as a “team” to finish cooperative tasks, e.g., terrain exploration/inspection. It is with high possibility that the robots are temporally or spatially correlated.

In MCNs, the service area is populated with Base Stations (BSs) [11]. The radio coverage of a BS (or a sector of the BS) is called a cell. The cells in the MCN are grouped into Location Areas (LAs), and each LA is assigned with a unique LA ID (LAI). The LAs are used for location management, which consists of “registration” and “call termination”. Let Machine Station (MS) denote the machine that is receiving transmission service from the MCN. Details of registration and call termination can be found in [1].

If two or more MSs share the same mobility behavior (i.e., the MSs visit and leave LAs at the same time), we identify that these MSs have “correlated mobility”. As mentioned previously, correlated mobility is one of the unique features in M2M communications in 3GPP. Definitional details for “correlated mobility” will be elaborated later. MSs with correlated mobility perform registration simultaneously. In the existing location management designed mainly for H2H communications, many MSs have correlated mobility and

perform registration at the same time, which results in a congested RACH. As a result, no MS can obtain a dedicated control channel to perform registration, even when there are still available dedicated control channels.

The RACH congestion delays MS location update, so the MS' LA record in the LDB becomes outdated, increasing the risks of undeliverable incoming sessions. The RACH congestion also blocks the originated data sessions (e.g., a new call connection establishment) because the congested RACH cannot deliver the requests to originate a data session. This phenomenon is named as *signaling congestion*, and it aggravates as MSs with correlated mobility becomes more densely-populated. To summarize, to support large-scale M2M mobile networking, we need a careful design for efficient location management. Using one of the key features of M2M communications, group-based feature, this paper aims to apply group-based control to reduce signaling congestion of MSs.

Previous works [7, 8, 9, 10] proposed different mechanisms to address the signaling congestion problem. In [7], MSs are grouped together, and a group head is chosen in each group to perform the registration on behalf of other members in the same group, thus reducing signaling traffic. However, the group management operations (i.e., creating, joining and leaving operations) are performed through MS' secondary ad-hoc wireless interface (e.g., WiFi interface), causing extra power consumption for each MS. As pointed out in [12], for WiFi communication, 60 percent of energy is consumed in idle listening even though IEEE 802.11 power-saving mode (PSM) is enabled. Considering each MS' limited energy, the mechanism in [7] cannot be fully applied to M2M communications. In [8], a buffering method is proposed to aggregate registrations on the BS when dedicated

control channels are exhausted. The buffering method reduces registration failure probability (i.e., the probability that a registration is rejected), but does not solve the signaling congestion problem caused by the RACH. In [9], based on the GSM architecture, an aggregating scheme is proposed for one-dimensional network (e.g., transportation systems), in which individual registrations from the same LA are aggregated at a virtual intermediate network node, called virtual visitor location register. This mechanism relies on the individual transmissions from MSs to this virtual visitor location register. So, the RACH congestion issue remains in this mechanism. In [10], a Tracking Area (TA) configuration approach is designed to solve the signaling congestion problem for the LTE system by assigning MSs with different and overlapping TA lists based on MS moving patterns. However, this approach relies on measured data on MS moving patterns obtained from network operators. Once the MS moving behavior changes, it introduces large overhead to reconfigure the TA lists, resulting in poor system performance.

In this paper, we propose a Group Location Management (GLM) mechanism and focus on the simulation results of the GLM. The intuition of the GLM mechanism is to group MSs based on the similarity of their mobility patterns, which are reflected in the temporal and spatial correlations among MSs at the LDB. The simulation experiment is based on the discrete event-driven approach, which is widely used in wireless network studies (e.g., [16, 11]). For further study on the influence of different correlations, we use three mobility models called “random walk model”, “bio-inspired mobility model”, and “transportation mobility model”, which represent the low, medium and high correlation.. The simulation results show that the GLM has better performance than original solution.

Therefore, we consider the GLM as a more feasible solution to deploy large-scale M2M communications.

The rest of this paper is organized as follows. In Chapter 2, we first present the details of the GLM mechanism. In Chapter 3, we use the event-driven approach to conduct the simulation experiments to investigate the performance of GLM by considering three different M2M mobility models. Chapter 4 concludes our work.



Chapter 2

Group-based Mechanism for Location Management



In Chapter 2.1, we first define the “correlated mobility” for MSs. In Chapter 2.2, based on the Chapter 2.1, we propose the Group Location Management (GLM) mechanism.

2.1 Correlated Mobility

Suppose that the service area of a CN is partitioned into M LAs, LA_1, LA_2, \dots, LA_M , and there are N MSs, m_1, m_2, \dots, m_N , in the CN. Let $LA_i \xrightarrow{t_x} LA_j$ denote a movement from LA_i to LA_j at the time t_x . Consider the time period $[t, t + T]$. Without loss of generality, during $[t, t + T]$, we suppose that an MS m_i moves from LA_1 to LA_2 at time t_1 , and then following by that m_i has a movement across K_1 LAs: $LA_{i,1} \xrightarrow{t_1} LA_{i,2} \xrightarrow{t_2} LA_{i,3} \dots \xrightarrow{t_{K_1-1}}$ LA_{i,K_1} , where $t < t_1 < t_2 < t_3 < \dots < t_{K_1-1} < t + T$. During $[t, t + T]$, another MS

m_j has a movement across K_2 LAs: $LA_{j,1} \xrightarrow{t'_1} LA_{j,2} \xrightarrow{t'_2} LA_{j,3} \cdots \xrightarrow{t'_{K_2-1}} LA_{j,K_2}$, where $t < t'_1 < t'_2 < t'_3 < \dots < t'_{K_2-1} < t + T$. The correlated mobility between m_i and m_j during $[t, t + T]$ exists if the following three conditions hold:

C1: $K_1 = K_2$.

C2: for $1 \leq k \leq K_1$, $LA_{k,i} = LA_{k,j}$.

C3: for $1 \leq k \leq K_1 - 1$, $|t_k - t'_k| < \epsilon$.

Note that the threshold ϵ is used to determine the similarity between m_i 's and m_j 's mobility in terms of time. In our study, we set $\epsilon = 120$ seconds.

In the standard location management [4], an MS updates its location (i.e., the LAI) in the LDB when one of the following two events occurs:

- The MS moves from one LA to another (i.e., the time when the MS detects that the stored LAI is different from that received from a cell);
- The periodic location update timer T_{PLU} is expired.

2.2 GLM: Group Location Management

In this Chapter, we propose the GLM mechanism. In GLM, to predict the correlated mobility for MSs, the LDB maintains the historical moving path for each MS. Based on that, the similarity between any two moving paths can be calculated to identify the MSs' correlated mobility for grouping. The details of the GLM mechanism are given as follows.

Let $p_i(K)$ denote the moving path for the MS m_i that contains the K most recently visited LAs of m_i , where $K \geq 1$. The moving path $p_i(K)$ is maintained in the LDB by using an order list with size K (e.g., the array data structure), and can be expressed by

$$p_i(K) = \langle p_{i,1}, p_{i,2}, \dots, p_{i,K} \rangle, \quad (2.1)$$

where $p_{i,j}$ is a 2-tuple $(a_{i,j}, t_{i,j})$ that $a_{i,j}$ denotes the last j th LA visited by m_i , and $t_{i,j}$ denotes the time when m_i moves into $a_{i,j}$. For example, suppose that m_i has the movement $\text{LA}_3 \xrightarrow{t_1} \text{LA}_1 \xrightarrow{t_2} \text{LA}_5 \xrightarrow{t_3} \text{LA}_4 \xrightarrow{t_4} \text{LA}_6 \xrightarrow{t_5} \text{LA}_2$. If we set up $K = 3$, we have $p_i(3) = \langle (\text{LA}_2, t_5), (\text{LA}_6, t_4), (\text{LA}_4, t_3) \rangle$.

For $i \neq j$, let $S_{i,j}(K)$ denote the similarity between the moving paths of the MSs m_i and m_j , i.e., $p_i(K)$ and $p_j(K)$. We define

$$S_{i,j}(K) = \prod_{v=1}^K B(p_{i,v}, p_{j,v}), \quad (2.2)$$

where

$$B(p_{i,v}, p_{j,v}) = \begin{cases} 1, & \text{if } a_{i,v} = a_{j,v}, |t_{i,v} - t_{j,v}| < \epsilon, \\ 0, & \text{otherwise.} \end{cases} \quad (2.3)$$

Note that as mentioned previously, the threshold ϵ is used to determine the similarity between m_i 's and m_j 's mobility in terms of time. From (2.2) and (2.3), it is clear that $S_{i,j}(K)$ is a binary function with the output value 1 or 0, and $S_{i,j}(K) = 1$ means that the moving path $p_i(K)$ is similar to $p_j(K)$ for the K most recently visited LAs in terms of time and space. In GLM, as $S_{i,j}(K) = 1$, we determine that m_i and m_j have correlated mobility.

The GLM mechanism consists of the *MS* and *LDB* parts:

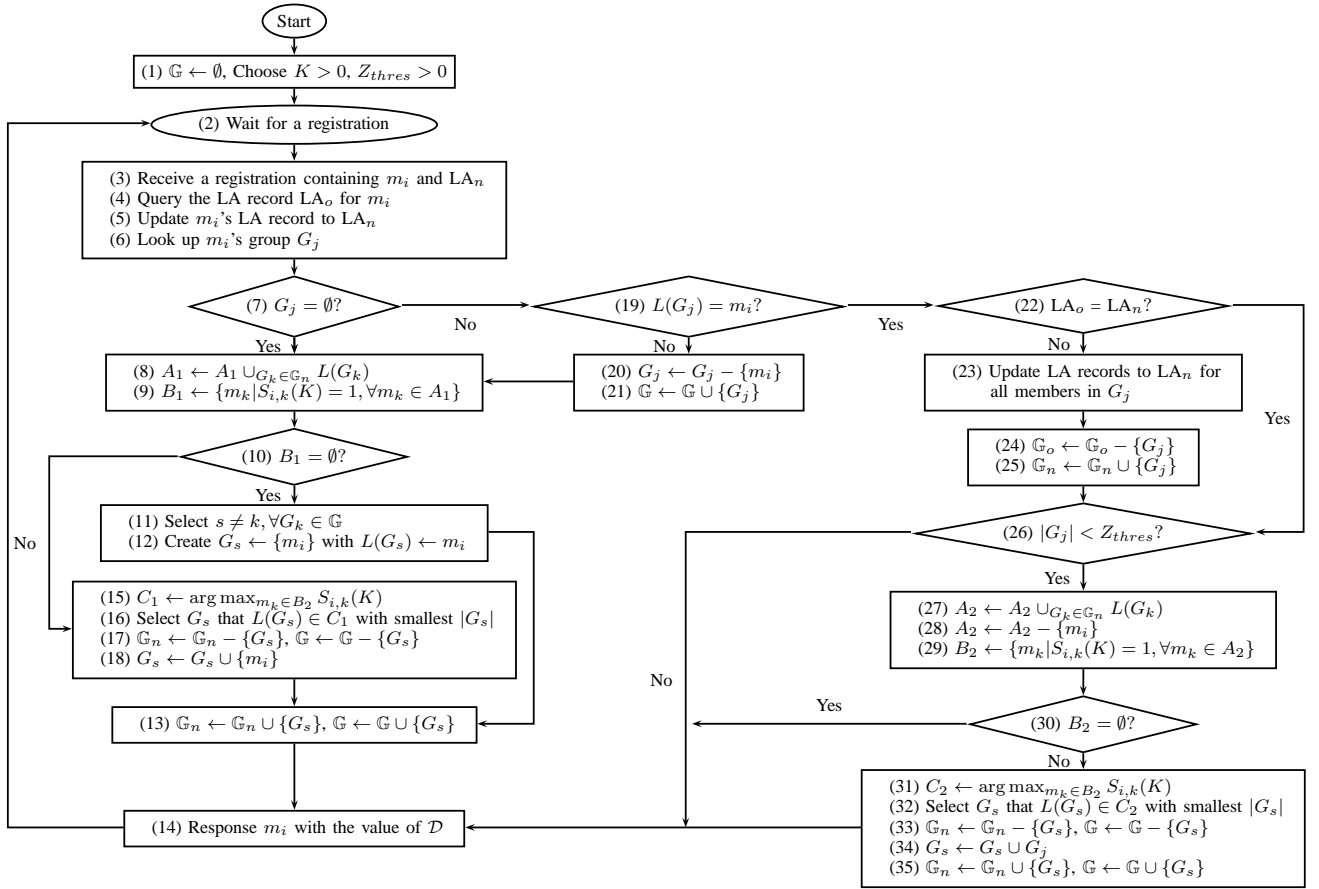


Figure 2.1: Flowchart of the LDB part

MS part: Each MS maintains the T_{PLU} timer and the \mathcal{D} variable, whose usages are as follows.

- The T_{PLU} timer: T_{PLU} is used to determine the timing for periodic location registration (location update). When T_{PLU} expires, the MS performs a registration to report its residing LA.
- The \mathcal{D} variable: \mathcal{D} is used to identify MS's status (i.e., the MS is the leader or a member of a group), where \mathcal{D} is a binary variable with the value "1" or "0" to denote that the MS is the leader or a member of a group, respectively.

When an MS is initially powered on, it performs a registration to report its residing LA to the LDB. After the initial registration, the MS is assigned to a newly created or existing group. The values of T_{PLU} and \mathcal{D} will be obtained from the acknowledgement message (e.g., the TA Update Accept message).

If $\mathcal{D} = 1$ (i.e., the MS is a group leader), the MS is responsible for performing the registration in the following two cases: periodic registration (i.e., on expiration of T_{PLU}) and LA change. Otherwise (i.e., $\mathcal{D} = 0$), the MS only performs the periodic registration. Note that the T_{PLU} and \mathcal{D} values are decided by the LDB after each registration.

LDB part: In GLM, the LDB in the network is responsible for the group maintenance and location management for MSs, including determining the T_{PLU} and \mathcal{S} values for each MS. Figure 2.1 illustrates the flow chart for the LDB part. The notations are introduced as follows. The group i is denoted by the set G_i that contains an amount of MSs, and the group leader of G_i is denoted by $L(G_i)$, where $L(G_i) \in G_i$. For

example, $L(G_1) = m_4$ means that the leader of the group G_1 is the MS m_4 . Let \mathbb{G} denote the group set that contains all groups in the network. Let the subset $\mathbb{G}_i \subseteq \mathbb{G}$ denote the set containing the groups in LA_i , $\forall i \in \{1, 2, 3, \dots, M\}$. It is clear that $\mathbb{G} = \mathbb{G}_1 \cup \mathbb{G}_2 \dots \cup \mathbb{G}_M$.

Initially, we set $\mathbb{G} \leftarrow \emptyset$ and choose the values of K and Z_{thres} (Steps 1 and 2), where K is the length of MS' moving path maintained in the LDB, and Z_{thres} is the group size threshold to determine whether a group should be merged into another one. In GLM, a group can be merged with another group if the two groups of MSs have correlated mobility, and more registrations can be reduced. After the parameter initialization, the LDB waits to receive a location registration (Step 3).

When the LDB receives a location registration containing m_i and LA_n (where m_i is the MS identification and LA_n is the LAI to be registered), it queries the existing LA record for m_i , denoted by LA_o , and updates the LA record of m_i to LA_n . Then the LDB looks up the group information for m_i (Steps 4 to 7). Suppose that m_i belongs to the group G_j . We consider the following three cases.

Case I: $G_j = \emptyset$. In this case, the LDB does not have any group records for the MS m_i , which implies that m_i has just powered on and performs an initial registration. As shown in Steps 8 to 10, the LDB searches the \mathbb{G}_n set. Based on the search results, the LDB executes one of the following two operations:

Group creation: The LDB can not find any group in \mathbb{G}_n whose leader's moving path is similar to the m_i 's moving path (i.e., no group leader has correlated mobility with m_i). The LDB creates a group G_s for m_i and sets

$L(G_s) \leftarrow m_i$ (Steps 11 and 12).

Group assignment: The LDB finds a group G_s in \mathbb{G}_n whose leader's moving path is similar to the m_i 's moving path. The LDB assigns m_i to the selected group G_s (Steps 15 to 18).

Case II: $G_j \neq \emptyset$ and $L(G_j) \neq m_i$. In this case, m_i is a member of the group G_j .

The LDB removes m_i from G_j (Steps 20 and 21). Then the LDB searches the \mathbb{G}_n set and performs one of the two operations mentioned in Case I for m_i , i.e., group creation or assignment.

Case III: $G_j \neq \emptyset$ and $L(G_j) = m_i$. In this case, m_i is the leader of the group

G_j . As $LA_o \neq LA_n$ (i.e., the registration is triggered due to LA change), the LDB updates the LA records for the entire group members in G_j (Step 23), and updates the corresponding \mathbb{G}_o and \mathbb{G}_n sets (Steps 24 and 25). Then, the LDB checks whether the group size $|G_j|$ is smaller than Z_{thres} to determine whether the group G_j should be merged (Step 26). If $|G_j| < Z_{thres}$, the LDB performs the following merge operation:

Group merge: The LDB searches the \mathbb{G}_n set to find a proper group to merge with (except its own group G_j). The selected group G_s satisfies two constraints: the moving path of $L(G_s)$ is similar to the moving path of m_i (Step 29), and the group size of G_s is the smallest (Step 32). Then the merge is performed (Step 34). After group merge, m_i is no longer the leader of G_j and becomes a member of G_s .

After each operation, the LDB updates the group G_s to the \mathbb{G}_n and \mathbb{G} sets (Step 13).

Based on the status of m_i (i.e., leader or member), the LDB responses m_i with the value of \mathcal{D} through the acknowledgement message (Step 14).

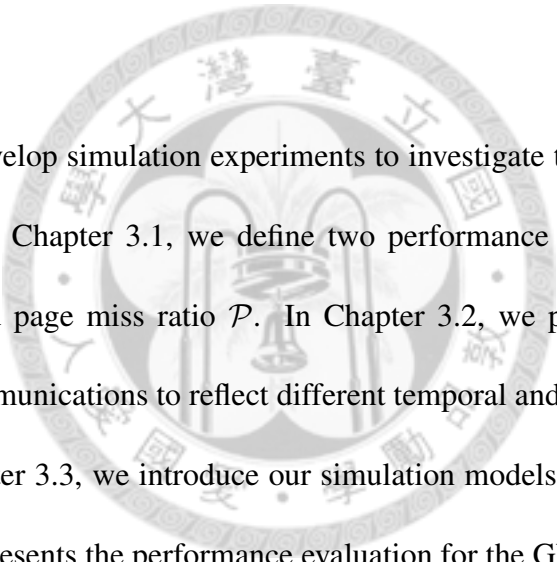
2.3 Discussion

As mentioned in Chapter 1, to route an incoming session to an MS, the CN queries the LDB to obtain the LAI for the MS and instructs all cells in the LA to page the MS. If the MS does not reside in the LA, the CN can not receive the response from the MS in a given time period. Therefore, the CN can not locate the MS to route the incoming session. This phenomenon is called “page miss”.

With the GLM mechanism, we can reduce significantly the registration signaling overhead and mitigate the signaling congestion problem. However, the GLM mechanism may result in higher page miss probability. Consider four MSs m_1, m_2, \dots, m_4 in the LA_1 , and the four MSs belong to the same group with the leader m_1 . Assume that the leader m_1 moves to LA_2 at time t_1 , the members m_2 and m_3 move together with m_1 at t_1 , and the member m_4 moves to LA_2 at t_2 , where $t_1 < t_2$. At t_1 , a registration due to LA change is executed by m_1 to update the LA record from LA_1 to LA_2 in the LDB. Therefore, at t_1 , the LA records for m_1, m_2, \dots, m_4 are updated to LA_2 . Yet the member m_4 still resides in LA_1 until t_2 . Therefore, during the time period $[t_1, t_2]$, the m_4 's LA record in the LDB is not correct, and the pages to the member m_4 are missed. During $[t_1, t_2]$, if m_4 performs the periodic registration (i.e., on the expiration of the T_{PLU} timer), the m_4 's LA record in the LDB will be corrected, and the LDB will divide m_4 from the group.

Chapter 3

Performance Evaluation



In this chapter, we develop simulation experiments to investigate the performance of the GLM mechanism. In Chapter 3.1, we define two performance metrics, the signaling reduction ratio \mathcal{R} and page miss ratio \mathcal{P} . In Chapter 3.2, we propose three mobility models for M2M communications to reflect different temporal and/or spatial correlations among MSs. In Chapter 3.3, we introduce our simulation models and parameter setups. Finally, Chapter 3.4 presents the performance evaluation for the GLM mechanism.

3.1 Performance Metrics

Observing the N MSs for a given time period T , we evaluate the GLM mechanism based on the two performance metrics, the signaling reduction ratio \mathcal{R} and page miss ratio \mathcal{P} , with the following definitions.

Signaling reduction ratio \mathcal{R} : During the observation period T , let $N_{r,GLM}$ denote the

number of the total registrations executed by the N MSs for the GLM mechanism.

Let N_r denote the number of the total registrations executed by the N MSs for the standard location management mechanism (as mentioned in Chapter 2), where $N_{r,GLM} \leq N_r$. The signaling reduction ratio \mathcal{R} can be expressed as

$$\mathcal{R} = \frac{E[N_r] - E[N_{r,GLM}]}{E[N_r]},$$

where $0 \leq \mathcal{R} \leq 1$. A larger \mathcal{R} ratio implies larger reduction of registration signaling overhead.

Page miss ratio \mathcal{P} : During T , let N_p denote the number of pages issued by the CN for the N MSs, and $N_{p,miss}$ denote the number of page misses in N_p pages, where $N_{p,miss} \leq N_p$. The page miss ratio \mathcal{P} is defined as

$$\mathcal{P} = \frac{E[N_{p,miss}]}{E[N_p]},$$

where $0 \leq \mathcal{P} \leq 1$. A higher \mathcal{P} ratio implies that the CN has higher possibility to have a page miss, resulting in a higher risk of undeliverable incoming sessions.

3.2 Mobility Models

In this paper, we use three MS mobility models to reflect different levels of temporal and/or spatial correlation among the MSs. We consider the N MSs moving around in the M LAs. Without loss of generality, we assume the mesh LA structure and $M = w^2$. In the mesh LA structure, the size of an LA is square shaped. An LA is indexed by (x, y) ,

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2)	(4,3)	(4,4)

Figure 3.1: The 4×4 mesh LA network structure

where $x, y \in \{1, 2, 3, \dots, w\}$. The LA (x, y) has c neighboring LAs, where

$$c = \begin{cases} 2, & \text{if } (x, y) \in \{(1, 1), (1, w), (w, 1), (w, w)\}, \\ 3, & \text{if } (x, y) \in \{(x, y) | x = 1 \text{ and } 2 \leq y \leq w - 1, \\ & x = w \text{ and } 2 \leq y \leq w - 1, 2 \leq x \leq w - 1 \\ & \text{and } y = 1, \text{ or } 2 \leq x \leq w - 1 \text{ and } y = w\}, \\ 4, & \text{otherwise.} \end{cases}$$

Figure 3.1 illustrates the mesh LA network structure with 16 LAs. In our mobility models, the size of LAs are indirectly reflected by the LA residence time. If the LA size is large, the mean LA residence time is relatively long, and vice versa.

Initially, the locations of the N MSs are uniformly distributed to M LAs. To simulate different levels of correlation in terms of time and space, we consider three kinds of mobility models for M2M communications, including the random walk, bio-inspired mobility and transportation mobility models. The three mobility models are described as follows:

Random walk: An MS decides its own moving direction and LA residence time independently and randomly. Specifically, suppose that the MS resides in the LA (x, y) for a time period and the MS moves to one of (x, y) 's neighboring LAs with the equal probability. In this mobility model, there is no temporal and spatial correlation among MSs.

Bio-inspired mobility: This mobility model is designed based on the concept of the bird-flocking behavior [15], which is described as follows:

The MSs distributed to the same LA form a “cluster”¹, and then we have M clusters in the network. There is a cluster head in each cluster. Consider an arbitrary cluster that contains X MSs. In the beginning, the cluster head and the other $X - 1$ MSs are in the same LA, supposing they reside in (x, y) at time t . Each MS has its own LA residence time for (x, y) . Assume that the cluster head resides in (x, y) for the period t_{ch} , and the other $X - 1$ MSs reside in (x, y) for the periods $t_1, t_2, t_3, \dots, t_{X-1}$, respectively. At $t + t_{ch}$, the cluster head moves to one of (x, y) 's neighboring LAs with the equal probability. Consider one of the $X - 1$ MSs leaving (x, y) at $t + t_i$, where $i = 1, 2, 3, \dots, X - 1$. If $t_i < t_{ch}$ (i.e., the MS leaves (x, y) earlier than the cluster head), the MS “disperses” its cluster and moves to one of (x, y) 's neighboring LAs with the equal probability. Otherwise (i.e., $t_i > t_{ch}$), the MS “follows” the moving direction of the cluster head.

¹Note that the term “cluster” is different from the term “group”. The term cluster is used to describe the MS mobility behavior (i.e., the MSs in the same cluster have correlated mobility), and the term group is used in the GLM mechanism for location management as mentioned in Chapter 2.

For the MS dispersed from its cluster (i.e., the MS moves to an LA different from the LA in which the cluster head resides), it moves toward the head's residing LA (i.e., the MS "returns" to the cluster head's LA), and the moving path is calculated by using the shortest path algorithm. Note that if there exist multiple shortest paths, we randomly choose one.

In this mobility model, the MSs have certain levels of temporal and spatial correlations.

Transportation mobility: This mobility model simulates the scenario where the MSs are in the same vehicle. Similar to the bio-inspired mobility model, the MSs distributed to the same LA form a cluster and we have M clusters in the network. In each cluster, there is a cluster head, and the cluster head decides the moving direction and LA residence time for the other MSs in the same cluster. The moving direction is decided randomly. In this mobility model, there exists very high correlation among MSs in terms of time and space.

3.3 Simulation Models

We develop the simulation experiments for the GLM mechanism based on the discrete event-driven approach. In the simulation experiments, we consider a 4×4 LA mesh structure (i.e., $M = 16$ as shown in Figure 3.1). We observe the $N = 640$ MSs moving around in the 4×4 LA mesh structure for a time period $T = 3000$ minutes. We assume that the page arrivals to an MS form the Poisson process with rate λ_p , and the mean

inter-page arrival time $1/\lambda_p$ is set to 100 minutes. In our simulation experiments, the LA residence time t_r is exponentially distributed. The three proposed mobility models (as mentioned in Chapter 3.2) are implemented in this chapter.

We first define four types of events listed as follows:

- The **CLUSTERCROSSBOUND** event represents that a cluster crosses boundary of LA. It is used only in transportation mobility model.
- The **CROSSBOUND** event represents that an MS crosses boundary of LA.
- The **PAGING** event represents that a paging arrival of an MS.
- The **PLU** event represents that an MS performs periodic location update, i.e., the periodic location update timer of the MS is expired.

The following counters are used in our simulation to calculate the output measures \mathcal{R} and

\mathcal{P} :

- N_{LU} counts the total number of **CROSSBOUND** events.
- N_{LLU} counts the total number of **CROSSBOUND** events which performed by group leaders.
- N_P counts the total number of **PAGING** events.
- N_{PM} counts the total number of page miss.
- N_{PLU} counts the total number of **PLU** events.

We repeat the simulation runs until t_s exceeds the time period T , which is a predefined positive number, to ensure the stability of the simulation results. Two output measures are investigated in our models, including the signaling reduction ratio \mathcal{R} and the page miss ratio \mathcal{P} (as mentioned in Chapter 3.1).

3.3.1 Random Walk Model

Figure 3.4 illustrates the flowchart of random walk mobility simulation model. Step 1 sets up the input parameters (including N , K , λ_c , λ_p , Z_{thres} , and T_{PLU}). In addition, the counters (e.g., N_{LU} , N_{LLU} , N_p , N_{PM} , and N_{PLU}) are set to zero, and the event queue is set to empty. Step 2 sets a initial GTable which has the functionality of GLM. Step 3 generates **CROSSBOUND**, **PAGING**, and **PLU** events for each MS. The timestamp of each event is set to zero. Step 4 removes the next event e from the event queue, and set the value of t_s to $e.timestamp$. Step 5 checks the type of event e .

If event e is a **CROSSBOUND** event at Step 5, Step 6 increases N_{LU} by one. Step 7 randomly selects a nLA which is the neighbor LA of the current LA of the MS. Step 8 sets $e.LA$ to nLA. Step 9 stores the nLA to the moving path of MS. Step 10 checks the state \mathcal{D} of MS. If $MS.\mathcal{D}$ equals to 1 that means the MS is a leader of group, the Step 11 increases the N_{LLU} by one. Step 12 updates the moving path of the MS to GTable and execute the GLM. The progress of GLM has shown in Fig. 2.1. Step 13 checks the return value \mathcal{D} of GLM. If the \mathcal{D} equals to 1, Step 14 sets the $MS.\mathcal{D}$ to 1. Otherwise, step 15 sets the $MS.\mathcal{D}$ to 0. Step 16 generates the next **CROSSBOUND** event of the MS and inserts it to the event queue. This simulation goes to Step 17.

If event e is a **PAGING** event at Step 5, Step 18 increases N_p by one. Step 19 sets cLA to the current LA which the GTable stored of the MS. Step 20 checks whether the cLA equals to nLA. If not (e.g., paging miss), Step 21 increases N_{PM} by one. Step 22 generate the next **PAGING** event of the MS and inserts it to the event queue. This simulation goes back to Step 17.

If event e is a **PLU** event at Step 5, Step 23 increases N_{PLU} by one. Step 24 updates the moving path of the MS to GTable and executes the GLM. Step 25 checks whether the return value \mathcal{D} of GLM is equals to 1. If so, Step 26 sets MS. \mathcal{D} to 1. Otherwise, Step 27 sets the MS. \mathcal{D} to 0. Step 28 generates the next **PLU** event of the MS and inserts it to the event queue. This simulation goes back to Step 17.

If the timestamp t_s is less than 3,000 at Step 17, the simulation goes back to the Step 4. Otherwise, the simulation will be terminated and calculate the output measures \mathcal{P} and \mathcal{R} at Steps 29 and 30.

3.3.2 Bio-inspired Mobility Model

Figure 3.3 illustrates the flowchart of bio-inspired mobility simulation model. Step 1 sets up the input parameters (including N , K , λ_c , λ_p , Z_{thres} , and T_{PLU}). In addition, the counters (e.g., N_{LU} , N_{LLU} , N_p , N_{PM} , and N_{PLU} are set to zero, and the event queue is set to empty. Step 2 sets a initial GTable which has the functionality of GLM. Step 3 initializes N_c clusters and randomly distributes them into M network. Step 4 initializes N MSs and randomly distributes them into N_c clusters. Step 5 randomly selects a MS to be the head of cluster for each cluster. Step 6 generates **CROSSBOUND**, **PAGING**, and

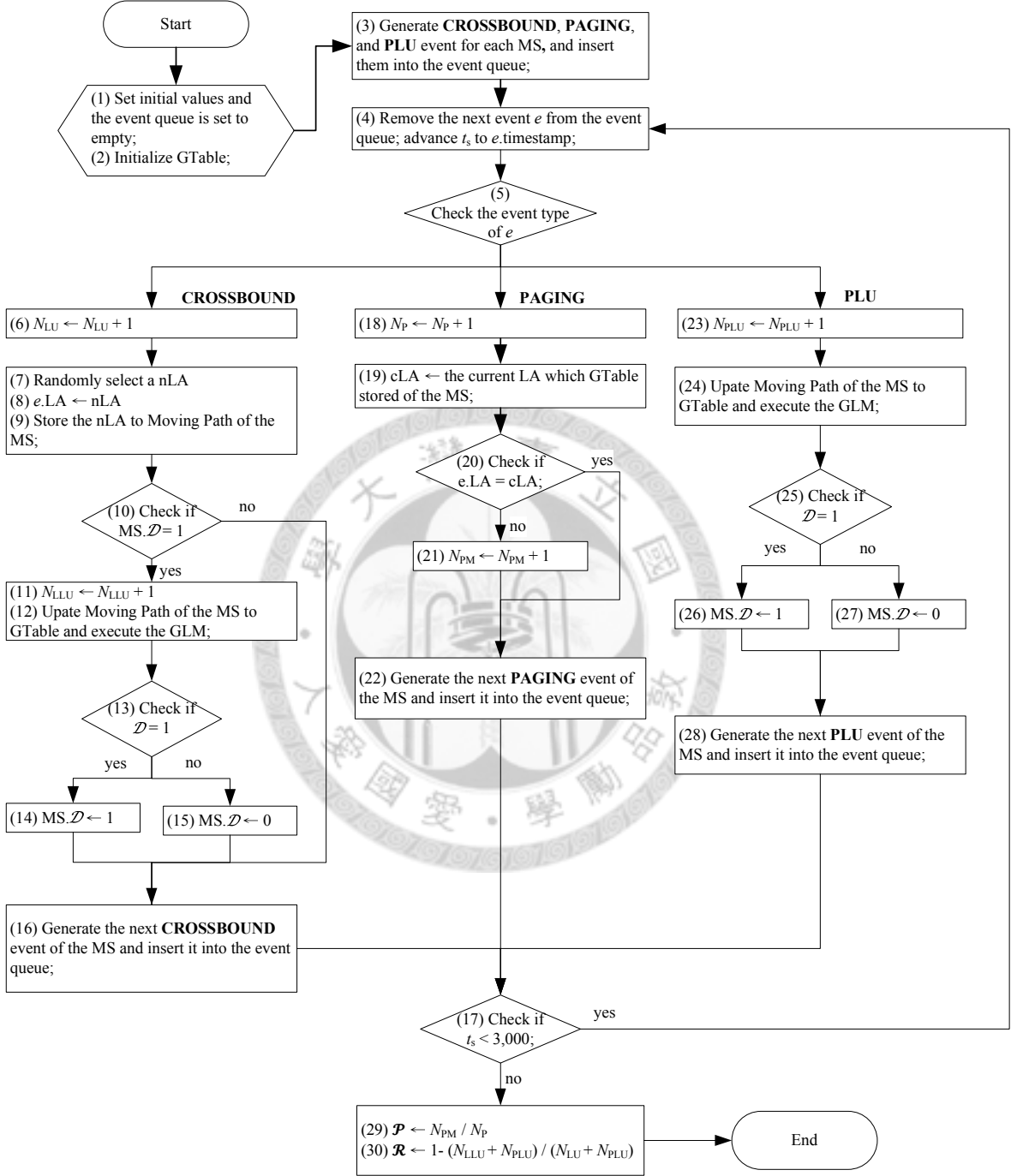


Figure 3.2: Flowchart of random walk mobility simulation model.

PLU events for each MS and inserts them into the event queue. The timestamp of each event is set to zero. Step 7 removes the next event e from the event queue, and set the value of t_s to e .timestamp. Step 8 checks the type of event e .

If event e is a **CROSSBOUND** event at Step 8, Step 9 increases N_{LU} by one. Step 10 checks whether the MS is a head of cluster. If so, Step 11 randomly selects a nLA. Otherwise, Step 12 selects a nLA which is near to head's residing LA. Step 13 sets e .LA to the selected nLA. Step 14 stores the nLA to the moving path of the MS. Step 15 checks if the state \mathcal{D} of the MS is equals to 1. If so, Step 16 increases N_{LLU} by one. Step 17 updates the moving path of the MS to GTable and executes the GLM. The progress of GLM has shown in Fig. 2.1. Step 18 checks the return value \mathcal{D} of GLM is equals to 1. If so, Step 19 the sets $MS.\mathcal{D}$ to 1. Otherwise, Step 20 sets $MS.\mathcal{D}$ to 0. Step 21 generates the next **CROSSBOUND** event of the MS and inserts it to the event queue. This simulation goes to Step 22.

If event e is a **PAGING** event at Step 8, Step 23 increases N_p by one. Step 24 sets cLA to the current LA which the GTable stored of the MS. Step 25 checks whether the cLA equals to nLA. If not (e.g., paging miss), Step 26 increases N_{PM} by one. Step 27 generate the next **PAGING** event of the MS and inserts it to the event queue. This simulation goes back to Step 22.

If event e is a **PLU** event at Step 8, Step 28 increases N_{PLU} by one. Step 29 updates the moving path of the MS to GTable and executes the GLM. Step 30 checks whether the return value \mathcal{D} of GLM is equals to 1. If so, Step 31 sets $MS.\mathcal{D}$ to 1. Otherwise, Step 32 sets the $MS.\mathcal{D}$ to 0. Step 33 generates the next **PLU** event of the MS and inserts it to the

event queue. This simulation goes back to Step 22.

If the timestamp t_s is less than 3,000 at Step 22, the simulation goes back to the Step 7. Otherwise, the simulation will be terminated and calculate the output measures \mathcal{P} and \mathcal{R} at Steps 34 and 35.

3.3.3 Transportation Mobility Model

Figure 3.2 illustrates the flowchart of bio-inspired mobility simulation model. Step 1 sets up the input parameters (including $N, K, \lambda_c, \lambda_p, Z_{thres}$, and T_{PLU}). In addition, the counters (e.g., $N_{LU}, N_{LLU}, N_p, N_{PM}$, and N_{PLU}) are set to zero, and the event queue is set to empty. Step 2 sets a initial GTable which has the functionality of GLM. Step 3 initializes N_c clusters and randomly distributes them into M network. Step 4 initializes N MSs and randomly distributes them into N_c clusters. Step 5 generates **CLUSTERCROSSBOUND** events for each cluster and inserts them into the event queue. Step 6 generates **CROSSBOUND, PAGING, and PLU** events for each MS and inserts them into the event queue. The timestamp of each event is set to zero. Step 7 removes the next event e from the event queue, and set the value of t_s to e .timestamp. Step 8 checks the type of event e .

If event e is a **CLUSTERCROSSBOUND** event at Step 8, Step 9 randomly selects a nLA. Step 10 sets the e .LA to nLA. Step 11 generates the next **CROSSBOUND** events of the MSs which belong to this cluster and inserts them into event queue. The timestamp of the **CROSSBOUND** events is set to t_s . Step 12 generates the next **CLUSTERCROSSBOUND** event of the cluster and inserts it into the event queue. This simulation goes to Step 13.

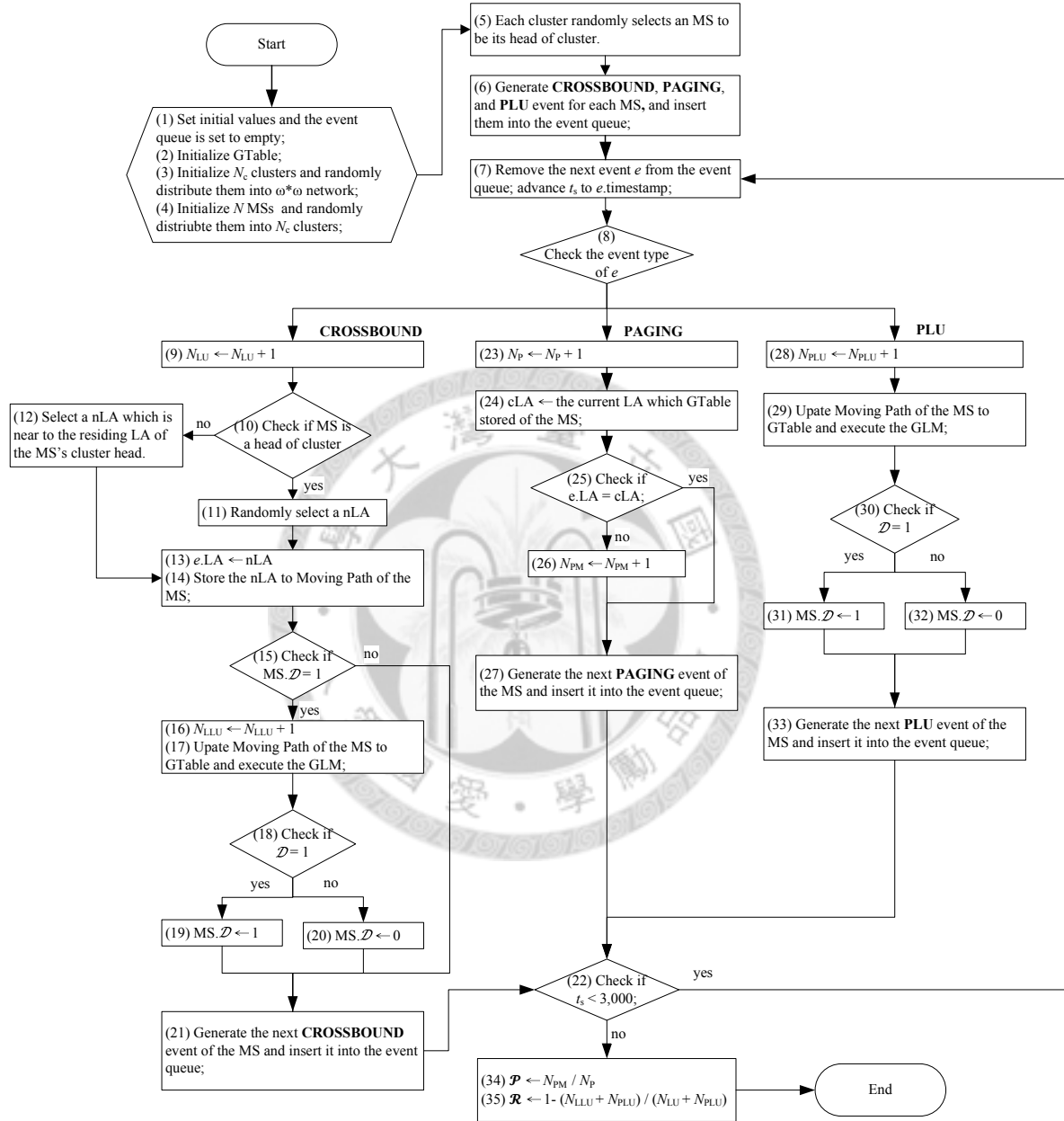


Figure 3.3: Flowchart of bio-inspired mobility simulation model.

If event e is a **CROSSBOUND** event at Step 8, Step 14 increase the N_{LU} by one. Step 15 sets nLA to $e.LA$. Step 16 stores the nLA to the moving path of the MS. Step 17 checks if the $MS.D$ is equals to 1. If so, Step 18 increases the N_{LLU} by one. Step 19 updates the moving path of the MS to GTable and execute the GLM. The progress of GLM has shown in Fig. 2.1. Step 20 checks the return value \mathcal{D} of GLM. If the \mathcal{D} equals to 1, Step 21 sets the $MS.D$ to 1. Otherwise, step 22 sets the $MS.D$ to 0. This simulation goes back to Step 13.

If event e is a **PAGING** event at Step 8, Step 23 increases N_p by one. Step 24 sets cLA to the current LA which the GTable stored of the MS. Step 25 checks whether the cLA equals to nLA . If not (e.g., paging miss), Step 26 increases N_{PM} by one. Step 27 generate the next **PAGING** event of the MS and inserts it to the event queue. This simulation goes back to Step 13.

If event e is a **PLU** event at Step 8, Step 28 increases N_{PLU} by one. Step 29 updates the moving path of the MS to GTable and executes the GLM. Step 30 checks whether the return value \mathcal{D} of GLM is equals to 1. If so, Step 31 sets $MS.D$ to 1. Otherwise, Step 32 sets the $MS.D$ to 0. Step 33 generates the next **PLU** event of the MS and inserts it to the event queue. This simulation goes back to Step 13.

If the timestamp t_s is less than 3,000 at Step 13, the simulation goes back to the Step 7. Otherwise, the simulation will be terminated and calculate the output measures \mathcal{P} and \mathcal{R} at Steps 34 and 35.

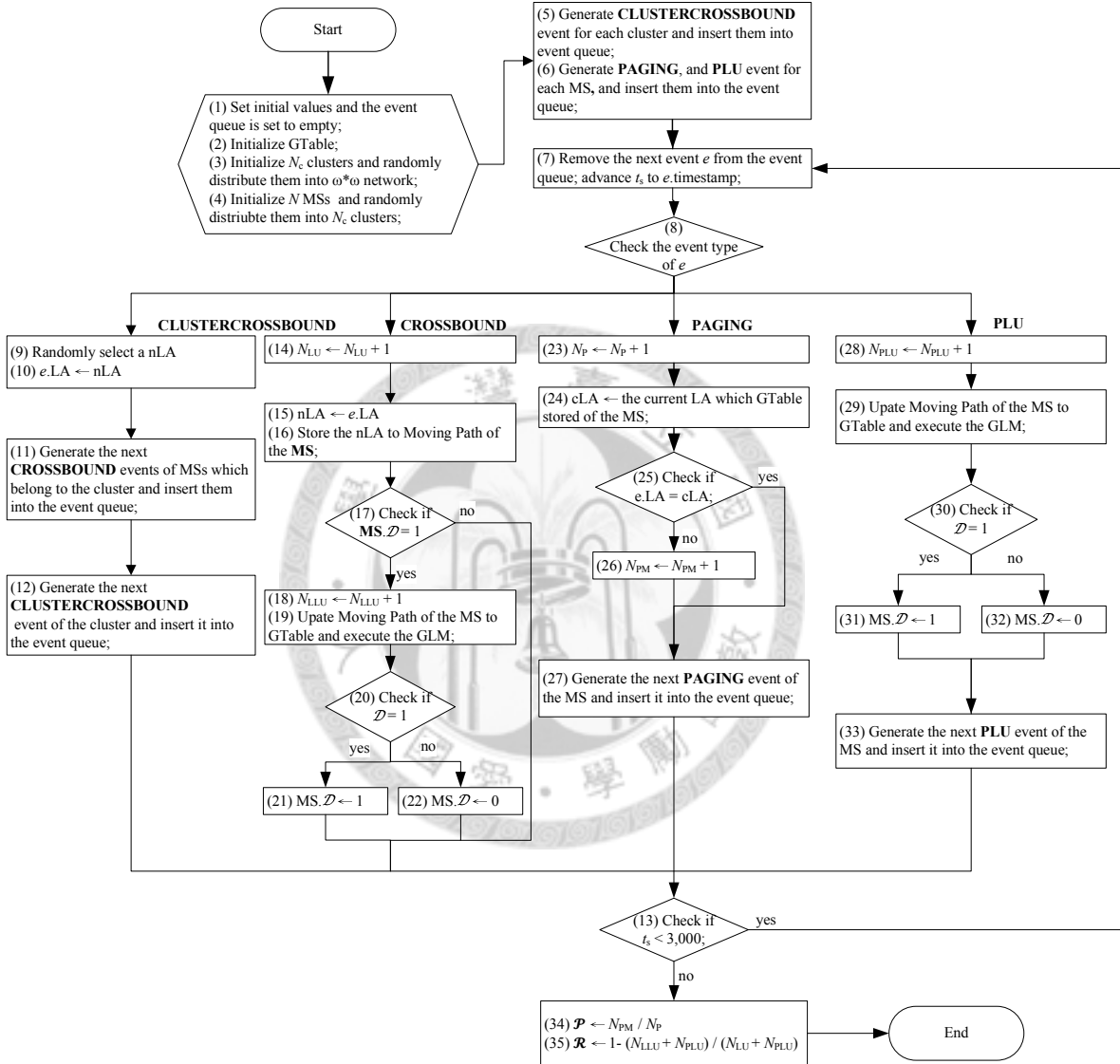


Figure 3.4: Flowchart of transportation mobility simulation model.

3.4 Simulation Results

In this Chapter, we investigate the performance of GLM by studying the effects of the parameters K , T_{PLU} and $E[t_r]$ on \mathcal{R} and \mathcal{P} . The details are given as follows.

3.4.1 Effects of K

In Figure 3.5, we investigate the effects of K (i.e., the length of the MS' moving path maintained in the LDB) on \mathcal{R} and \mathcal{P} , where we set $Z_{thres} = 10$, $\epsilon = 2$ minutes, $E[t_r] = 10$ minutes, $1/\lambda_p = 100$ minutes, and $T_{PLU} = 60$ minutes.

As shown in Figure 3.5(a), as K increases, it is less likely for the GLM mechanism to form groups for MSs, so less registration signaling overhead can be reduced. We observe that the \mathcal{R} performance decreases as K increases. On the other hand, in Figure 3.5(b), as K increases, the MSs with correlated mobility are more likely to be grouped by the GLM mechanism. Thus, better \mathcal{P} performance is observed when K increases. It is also worth noticing that as $K > 5$, the GLM mechanism can not reduce the registration signaling (i.e., the \mathcal{R} performance is zero) for the random walk and bio-inspired mobility models.

In Figure 3.5, we also investigate the \mathcal{R} and \mathcal{P} performance for the three mobility models. The GLM mechanism has very good \mathcal{R} and \mathcal{P} performance for the transportation mobility model due to the high correlation among MSs. In Figure 3.5(a), as $K > 5$, we observe that \mathcal{R} for the bio-inspired mobility is higher than \mathcal{R} for the random walk (i.e., more registration signaling can be saved in the bio-inspired mobility). On the other hand, in Figure 3.5(b), as $K > 5$, \mathcal{P} for the bio-inspired mobility is only higher than \mathcal{P} for the

random walk slightly.

To summarize, when the temporal and spatial correlation among MSs becomes higher (i.e., the MSs have higher chance to have correlated mobility behaviors), the GLM mechanism can reduce larger registration signaling overhead.

3.4.2 Effects of Periodic Location Update Timer T_{PLU}

In Figure 3.6, we investigate the effects of T_{PLU} on \mathcal{R} and \mathcal{P} , where we set $Z_{thres} = 10$, $\epsilon = 2$ minutes, $K = 2$, $E[t_r] = 10$ minutes, and $1/\lambda_p = 100$ minutes.

As mentioned in Chapter 2.1, the registration is triggered by LA change or T_{PLU} expiration. In the GLM mechanism, the registrations due to the expiration of T_{PLU} can not be reduced. It turns out that as T_{PLU} increases, the number of the total registrations executed by an MS decreases, which magnifies the ratio for the reduced registrations due to LA change. Thus, we observe that the \mathcal{R} performance increases as T_{PLU} increases.

On the other hand, as mentioned in Chapter 2.3, the expiration of the T_{PLU} timer can help GLM correct the wrong LA record in the LDB (that results in page miss). Therefore, in Figure 3.6(b), as T_{PLU} increases (i.e., it takes longer time for an MS to correct the wrong LA record), the \mathcal{P} performance increases (i.e., higher page miss probability).

To summarize, only when the MSs have correlated mobility (see the transportation mobility model in Figure 3.6) do we prefer to use a larger T_{PLU} timer. When the MSs do not have strong correlation between them (see the random walk or bio-inspired mobility models in Figure 3.6), to avoid high page miss probability, a small T_{PLU} is suggested.

3.4.3 Effects of LA Residence Time $E[t_r]$

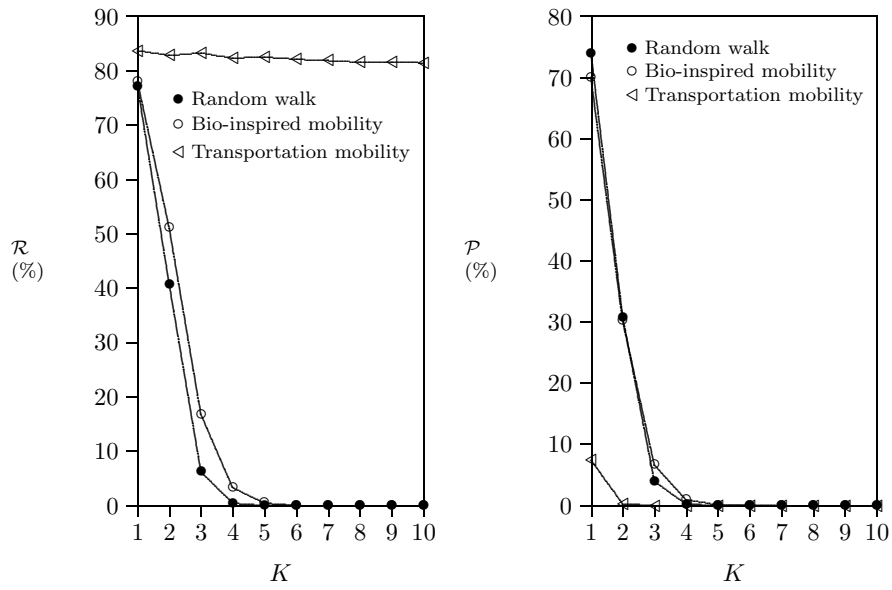
In Figure 3.7, we investigate the effects of $E[t_r]$ (i.e., the mean LA residence time) on \mathcal{R} and \mathcal{P} , where we set $Z_{thres} = 10$, $\epsilon = 2$ minutes, $K = 2$, $1/\lambda_p = 100$ minutes, and $T_{PLU} = 60$ minutes.

As shown in Figure 3.7(a), as $E[t_r]$ increases (i.e., an MS stays in an LA longer), the MS has less LA boundary crossings during T , so fewer registrations due to LA change can be reduced. Therefore, we observe that the \mathcal{R} performance decreases as $E[t_r]$ increases. On the other hand, in Figure 3.7(a), as $E[t_r]$ increases, the \mathcal{P} performance decreases because it is less likely that the LA record is not correct in the LDB.

3.4.4 Effects of Inter-Page Arrival Time $1/\lambda_p$

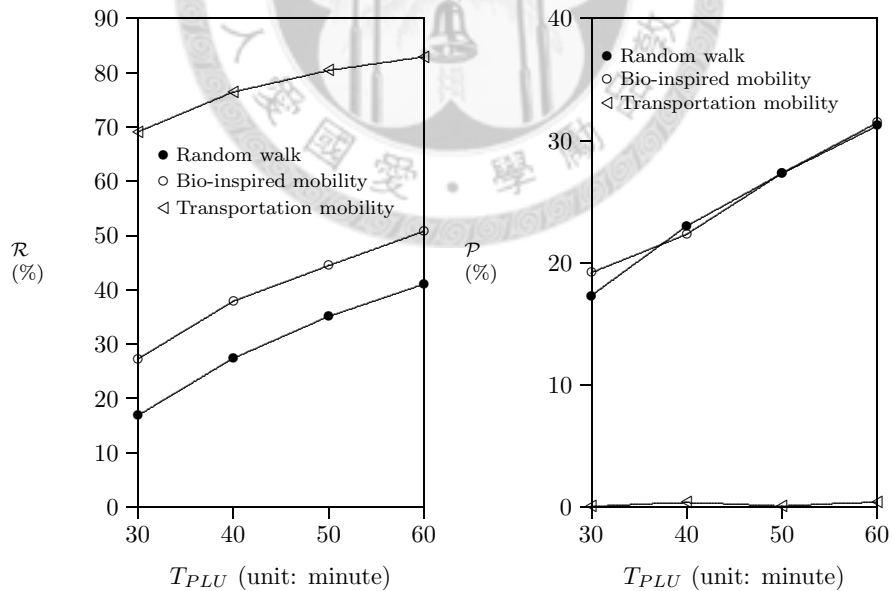
In Figure 3.8, we investigate the effects of $1/\lambda_p$ (i.e., the mean inter-page arrival time) on \mathcal{R} and \mathcal{P} , where we set $Z_{thres} = 10$, $\epsilon = 2$ minutes, $K = 2$, $E[t_r] = 10$ minutes, and $T_{PLU} = 60$ minutes.

In Figure 3.8(a) and Figure 3.8(b), we observe that both \mathcal{R} and \mathcal{P} performance are insensitive to the mean inter-page arrival time for the three kinds of mobility models.



(a) Signaling reduction ratio (b) Page miss ratio

Figure 3.5: Effects of K on \mathcal{R} and \mathcal{P} for three mobility models, where $Z_{thres} = 10$, $\epsilon = 2$ minutes, $E[t_r] = 10$ minutes, $1/\lambda_p = 100$ minutes, $T_{PLU} = 60$ minutes.



(a) Signaling reduction ratio (b) Page miss ratio

Figure 3.6: Effects of T_{PLU} on \mathcal{R} and \mathcal{P} for three mobility models, where $Z_{thres} = 10$, $\epsilon = 2$ minutes, $K = 2$, $E[t_r] = 10$ minutes, $1/\lambda_p = 100$ minutes.

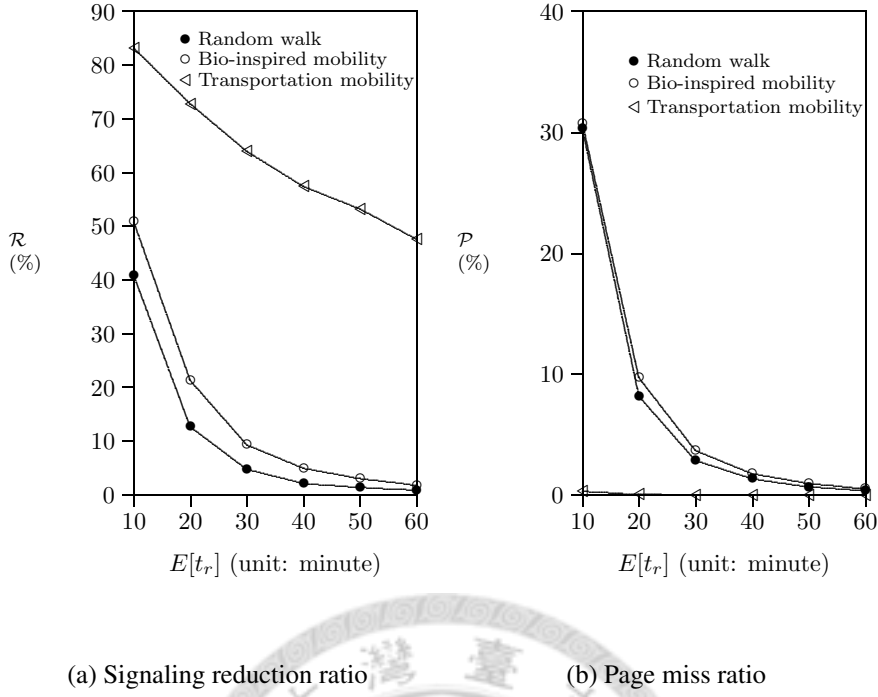


Figure 3.7: Effects of $E[t_r]$ on \mathcal{R} and \mathcal{P} for three mobility models, where $Z_{thres} = 10$, $\epsilon = 2$ minutes, $K = 2$, $1/\lambda_p = 100$ minutes, $T_{PLU} = 60$ minutes.

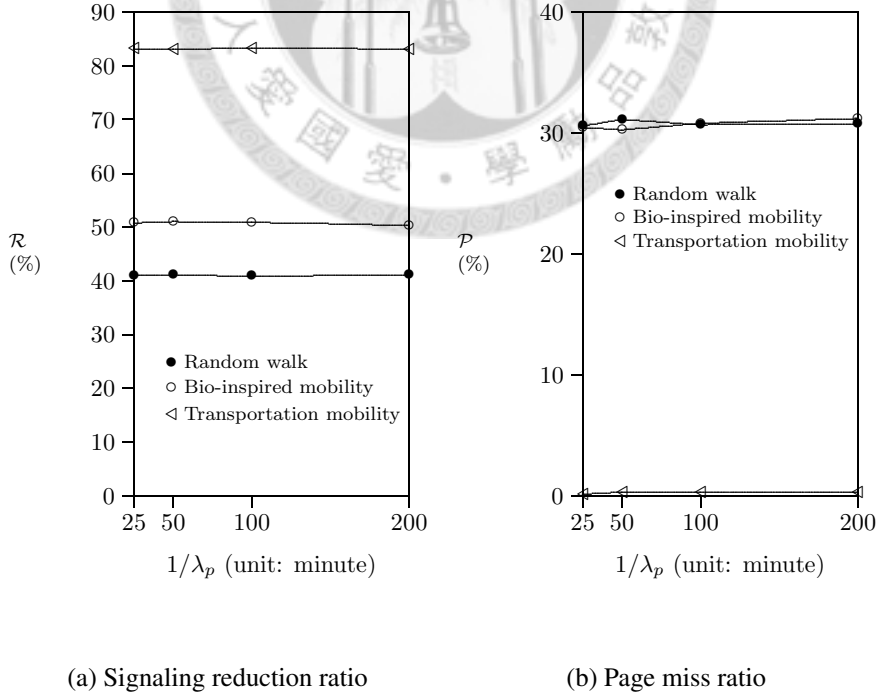
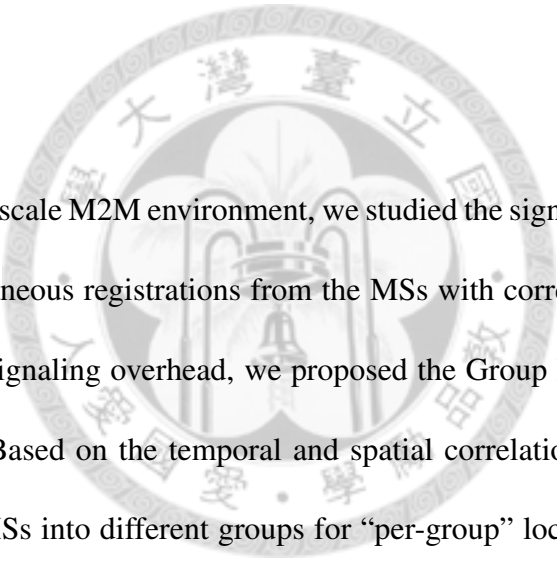


Figure 3.8: Effects of $1/\lambda_p$ on \mathcal{R} and \mathcal{P} for three mobility models, where $Z_{thres} = 10$, $\epsilon = 2$ minutes, $K = 2$, $E[t_r] = 10$ minutes, $T_{PLU} = 60$ minutes.

Chapter 4

Conclusion



In this paper, for large-scale M2M environment, we studied the signaling congestion problem caused by simultaneous registrations from the MSs with correlated mobility. To reduce the registration signaling overhead, we proposed the Group Location Management (GLM) mechanism. Based on the temporal and spatial correlations among MSs at the LDB, GLM divides MSs into different groups for “per-group” location management. In each group, the registration aggregation is done by selecting an MS to be the group leader and to perform the registration on behalf of the other members in the group. In GLM, the group management is performed by the LDB in the CN without involving the MSs. There is no extra energy consumption and protocol modification for the MSs. However, the GLM mechanism may cause higher page miss probability, resulting in higher risk of undeliverable incoming session. To study the signaling reduction and page miss ratios, we conducted simulation experiments and considered three kinds of mobility models for M2M communications, including random walk, bio-inspired mobility and transportation

mobility. Our study shows that the GLM mechanism can significantly reduce the registration signaling overhead for MSs with correlated mobility.



Bibliography

- [1] 3GPP, 3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Location Management Procedures (Release 8), 3G TS 23.012, September 2009.
- [2] 3GPP, 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Service requirements for Machine-Type Communications (MTC); Stage 1 (Release 11), 3G TS 22.368, March 2011.
- [3] 3GPP, 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; System Improvements for Machine-Type Communications; (Release 11), 3G TS 23.888, June 2011.
- [4] 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 11), 3G TS 36.300, March 2012.
- [5] Thompson, J. M., Gross, D., Shortle, J. F., and Harris, C. M., *Fundamentals of Queueing Theory*, John Wiley & Sons, 2008.

- [6] Lin, Y.-B., Reducing Location Update Cost in a PCS Network, *IEEE Transactions on Networking*, vol. 5, no. 1, pp. 25-33, February 1997.
- [7] Wang, F., Tu, L., Zhang, F., and Huang, Z., Group Location Update Scheme and Performance Analysis for Location Management in Mobile Network, *Proceedings of IEEE Vehicular Technology Conference (VTC)*, pp. 2429-2433, May 2005.
- [8] Zhang, Y. and Fujise, M., Location Management Congestion Problem in Wireless Networks, *IEEE Transactions on Vehicular Technology*, vol. 56, no. 2, pp. 942-954, March 2007.
- [9] Han, I. and Cho, D.-H., Group Location Management for Mobile Subscribers on Transportation Systems in Mobile Communication Networks, *IEEE Transactions on Vehicular Technology*, vol. 53, no. 1, pp. 181-191, January 2004.
- [10] Razavi, S. M. and Yuan, D., Mitigating Mobility Signaling Congestion in LTE by Overlapping Tracking Area Lists, *Proceedings of ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, pp. 285-292, October 2011.
- [11] Fu, H.-L., Lin, P., and Lin, Y.-B., Reducing Signaling Overhead for Femto-cell/Macrocell Networks, To Appear in *IEEE Transactions on Mobile Computing*.
- [12] Zhang, X. and Shin, K. G., E-MiLi: Energy-Minimizing Idle Listening in Wireless Networks, *Proceedings of ACM International Conference on Mobile Computing and Networking (MobiCom)*, pp. 205-216, September 2011.

- [13] Shafiq, M. Z., Ji, L., Liu, A. X., Pang, J., and Wang, J., A First Look at Cellular Machine-to-Machine Traffic - Large Scale Measurement and Characterization, *Proceedings of the International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, June 2012.
- [14] Mondada, F., Pettinaro, G.C., Kwee, I., Guignard, A., Gambardella, L., Floreano, D., Nolfi, S., Deneubourg, J.-L., Dorigo, M., SWARM-BOT: A Swarm of Autonomous Mobile Robots with Self-Assembling Capabilities, *Proceedings of the International Workshop on Self-Organisation and Evolution of Social Behaviour*, pp. 11-22, September 2002.
- [15] Misra, S. and Agarwal, P., Bio-inspired Group Mobility Model for Mobile Ad Hoc Networks based on Bird-flocking Behavior, *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 16, no. 3, pp. 437-450, 2012.
- [16] Lin, Y.-B. and Yang, S.-R., A Mobility Management Strategy for GPRS, *IEEE Transactions on Wireless Communications*, vol. 2, no. 6, pp. 1178-1188, November 2003.