

國立交通大學

網路工程研究所

碩士論文

基於行動電信網路的車速量測之研究

Deriving the Vehicle Speeds from Mobile  
Telecommunications Network

研究生：張裕隆

指導教授：林一平 教授

中華民國 102 年 6 月

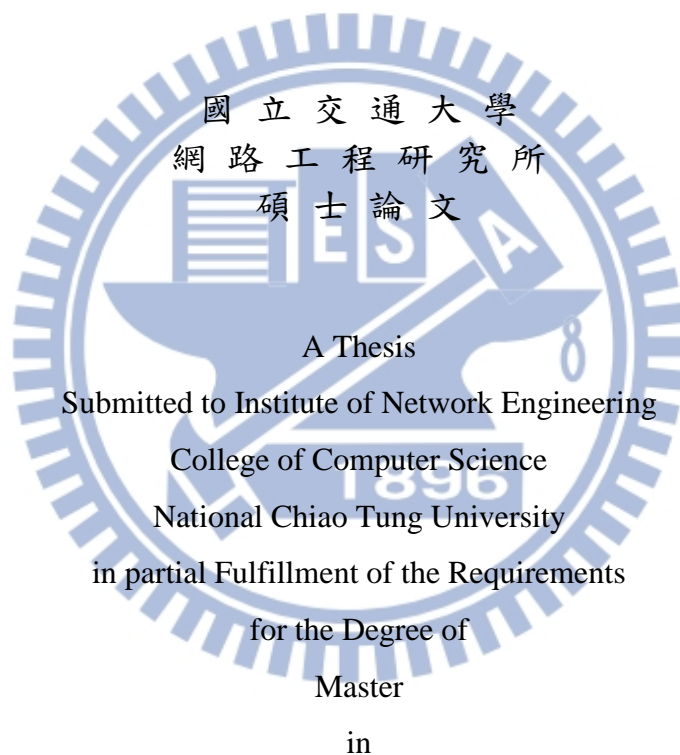
基於行動電信網路的車速量測之研究  
Deriving the Vehicle Speeds from Mobile Telecommunication Network

研究生：張裕隆

Student : Yu-Long Chang

指導教授：林一平

Advisor : Yi-Bing Lin



Computer Science

June 2013

Hsinchu, Taiwan, Republic of China

中華民國 102 年 6 月

# 基於行動電信網路的車速量測之研究

學生：張裕隆

指導教授：林一平博士

國立交通大學網路工程研究所碩士班

## 摘 要

智能運輸系統 (Intelligent Transportation Systems ; ITS) 可以利用一些感測器 (sensors) 或軟體測量道路車速。林、張及皇甫曾提出 *Lin-Chang-Huangfu* (LCH) 方法來計算細胞停留時間。它只需要使用行動交換中心 (Mobile Switching Center ; MSC) 中標準計數器的值，而不需要對電信網路做任何修改。本論文藉由推導 LCH 方法計算出的細胞停留時間 (cell residence times) 的偏差 (bias) 來探討此方法的準確性。我們並藉由一些過濾和補償技術來擴充 LCH 方法應用於車速的估計，並且以台灣桃園縣龍潭鄉國道三號上的車輛偵測器 (Vehicle Detector ; VD) 來驗證我們的方法。我們的研究結果指出 LCH 方法在車速的估計上是有效的。

**關鍵字：**Lin-Chang-Huangfu (LCH) 方法，行動交換中心，電信，車速

# Deriving the Vehicle Speeds from Mobile Telecommunications Network

Student: Yu-Long Chang

Advisor: Dr. Yi-Bing Lin

Institute of Network Engineering  
National Chiao Tung University



## ABSTRACT

Vehicle speeds of roads are often measured by the *Intelligent Transportation Systems* (ITS) through some sensors or software solutions. Our previous work proposed the *Lin-Chang-Huangfu* (LCH) scheme to compute the cell residence times by the standard counter values in the mobile telecommunications switches without any modification to the telecommunications network. In this paper, we investigate the accuracy of the LCH scheme by deriving the bias of the cell residence times computed in this scheme. Then we extend the LCH scheme with some filtering techniques for the vehicle speed estimation, and validate our approach with the vehicle detector measurements at National Highway 3, Longtan Township, Taoyuan County, Taiwan. Our study indicates that the LCH scheme is an effective approach for the vehicle speed estimation.

**Index Terms:** Lin-Chang-Huangfu (LCH) Scheme, Mobile Switching Center (MSC), telecommunication, vehicle speed

## 誌 謝

首先誠摯感謝指導教授林一平博士，沒有老師的專業建議與細心的指導，我無法完成此篇碩士論文。在林一平博士嚴格而不厭其煩的指導中，我學習到了做研究的方法與態度，並從中獲取了極大的助益。接著我要感謝洪慧念博士與彭南夫博士，感謝你們在論文中給我的幫助，讓我的論文更有深度。我也要感謝逢愛君博士與顏在賢博士，你們在碩士班口試中給予許多寶貴的建議。

接著我要感謝實驗室的夥伴，尤其是劉仁煌學長，謝謝你幫助我完成我的碩士論文，給了我很多有用的意見以及協助我修改論文，也感謝你從專題生開始給予我的指導。也謝謝實驗室的夥伴的陪伴，使得我碩士班的生活更加充實。特別感謝鍾易翰在迷霧壟罩的峽谷中帶領我們前進，適時的給予我們協助，領導我們擊敗各種困難，讓我們在這旅途上一路往前，直到終點，謝謝你 Hank。

最後我要感謝我的家人以及朋友，謝謝你們在我碩士期間給我的支持與鼓勵，讓我可以繼續堅持下去完成論文，也要感謝系桌的大家，謝謝你們陪我打球，陪我抒發壓力，還有所有幫助過我的人，謝謝你們。



# 目錄

中文摘要	i
ABSTRACT	ii
誌謝	iii
目錄	iv
圖目錄	v
符號說明	vi
第一章 簡介	1
第二章 Lin-Chang-Huangfu (LCH) 方法	4
第三章 細胞停留時間估計的偏差 (Bias)	6
第四章 改善車速評估準確性之技術	16
第五章 數據分析	20
第六章 結論與未來工作	25
參考文獻	26
附錄 A 三種通話個數公式的模擬流程圖	29
附錄 B $E[\rho_n(\tau)]$ 之推導	31
附錄 C 程式碼	33

# 圖目錄

1.1	交通資訊軟體	1
1.2	簡易電信網路架構	2
3.1	手機移動和通話抵達的時間圖 ( $\Delta$ : 一個手機移動至細胞 $i$ , $\blacktriangle$ : 一個手機離開細胞 $i$ , $\circ$ : 一通電話到達, $\bullet$ : 一通電話結束, $\blacksquare$ : $\Delta\tau$ )	7
4.1	兩種影響車速估計準確性的情境	17
5.1	實驗環境為台灣桃園縣龍潭鄉國道三號 (以 $\times$ 表示 VD, 以 $\circ$ 表示 BS, 以 $\bullet$ 表示天線的方向)	20
5.2	2011 年 9 月 16 日 VD 和 LCH 方法的車速比較 ( $\Delta t = 1$ hour, $x = 1$ km)	22
5.3	2011 年 9 月 25 日 VD 和 LCH 方法的車速比較 ( $\Delta t = 1$ hour, $x = 1$ km)	23
5.4	LCH 和 VD 方法之間的差異 ( $\Delta t = 1$ hour, $x = 1$ km)	24
A.1	三種通話個數公式的流程圖	29

## 符號說明

$\Delta t$ ：時間區段

$\Delta\tau$ ：時間點  $\tau$  所屬的時間區段， $\left(\left\lfloor \frac{\tau}{\Delta t} \right\rfloor \Delta t, \left\lfloor \frac{\tau}{\Delta t} \right\rfloor \Delta t + \Delta t\right)$

$\rho(\tau)$ ：在  $\Delta\tau$  中細胞的運送流量

$t_c$ ：通話保留時間 (call holding time)

$t_m$ ：細胞停留時間 (cell residence time)

$t_m^*(\tau)$ ：由 LCH 方法估計在  $\Delta\tau$  中到達細胞的手機的細胞停留時間

$N(\tau)$ ：在  $\Delta\tau$  期間細胞中的期望使用者個數

$\lambda_u(\tau)$ ：在  $\Delta\tau$  期間使用者進入細胞的到達速率

$t_a$ ：通話到達時間間隔 (inter-call arrival time)

$v(\tau)$ ：目標路段在  $\Delta\tau$  中的平均車速

$b_{t_m}(t_m^*(\tau))$ ：細胞停留時間的偏差

$\alpha(\tau)$ ：在  $\Delta\tau$  中新通話待在細胞中的個數

$\beta(\tau)$ ：在  $\Delta\tau$  中已存在的通話通話待在細胞中的個數

$\gamma(\tau)$ ：在  $\Delta\tau$  中交遞通話待在細胞中的個數

$\lambda$ ：通話到達速率

$\frac{1}{\eta}$ ：細胞停留時間的平均值

$\frac{1}{\mu}$ ：通話保留時間的平均值



$\frac{1}{\delta}$  :  $\Delta\tau$  的平均值

$f_m(t_m)$  : 細胞停留時間的密度函數 (density function)

$F_m(t_m)$  : 細胞停留時間的分佈函數 (distribution function)

$f_m^*(t_m)$  :  $f_m(t_m)$  的拉普拉斯轉換

$\tau_c$  : 剩餘通話保留時間

$\tau_m$  : 剩餘細胞停留時間

$P_i$  : 一新通話交遞  $i$  次後結束通話的機率

$q_1$  :  $t_c > \tau_m$  的機率，即一個新通話發生交遞的機率

$q_2$  :  $\tau_c > t_m$  的機率，即一個通話在曾經交遞的情況下發生交遞的機率

$\rho_n(\tau)$  : 新通話在  $\Delta\tau$  中細胞的通話時間

$\rho_e(\tau)$  : 已存在通話在  $\Delta\tau$  中細胞的通話時間

$\rho_h(\tau)$  : 交遞通話在  $\Delta\tau$  中細胞的通話時間

$\Delta\tau_n$  : 新通話 (new call) 到達的時間點觀察  $\Delta\tau$  的剩餘時間

$\Delta\tau_h$  : 交遞通話 (handover call) 到達的時間點觀察  $\Delta\tau$  的剩餘時間

$t$  : 觀察時間區段

$t^*$  : 在  $t$  期間，一個使用者被通話佔據的時間

$n_1$  : 一通電話的平均交遞次數

$n_2$  : 細胞個數

$t_{c,i}$  : 第  $i$  個通話的通話保留時間

$f_c(t_{c,i})$  : 第  $i$  個通話的通話保留時間的密度函數

$\theta$  : 在  $t$  期間， $n_2$  個細胞所量測到的通話時間

$\Delta\tau_k$  : 在最近第  $k$  週同一天的同一個時間區段

$K_h$  : 計算車速的樣本數門檻值

$V_s$  : 使用近歷史 (near-history) 補償技術的門檻值

$W$  : 權重係數



# 第一章 簡介

大部分的智能運輸系統 (Intelligent Transportation Systems ; ITS) 會測量道路車速來協助駕駛人估計行車時間和避開塞車路段。為了提供此服務，ITS 必須負責收集和計算道路車速。使用者可以透過網路 (例如 Internet) 來獲取這些交通資訊。圖 1.1 展示一個智慧型手機上的交通資訊軟體。



圖 1.1: 交通資訊軟體

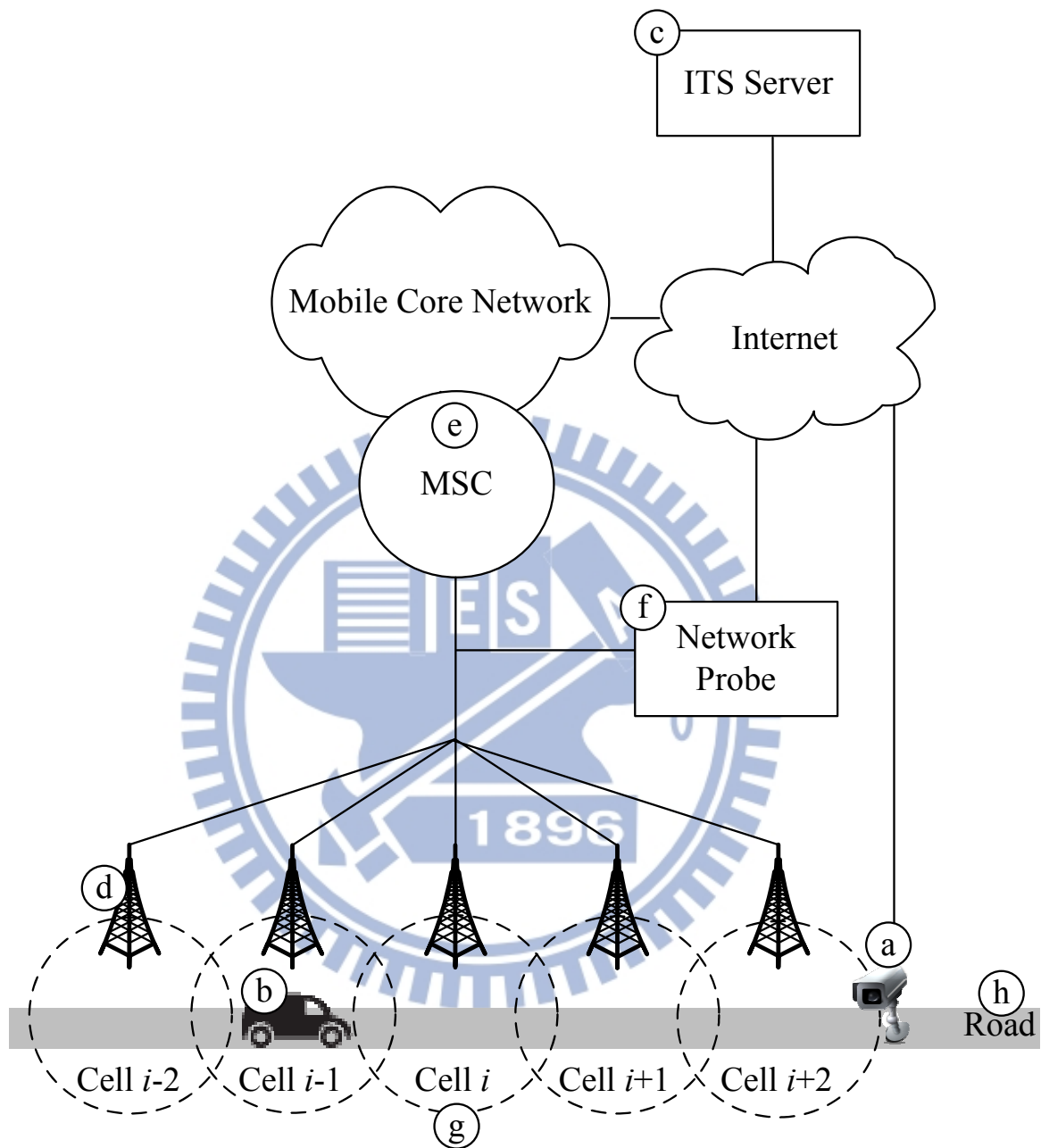


圖 1.2: 簡易電信網路架構

一般測量車速的方法可以分成以下三類:

- **車輛偵測器** (Vehicle Detector ; VD) 方法 [1] 將車輛偵測器 (圖 1.2 ④) 安裝在道路

上測量車輛(圖 1.2 ⑥)的速度，並且藉由有線或無線網路將車速回報給 ITS 伺服器(圖 1.2 ③)。

- **基於全球定位系統的探測車** (Global Positioning System(GPS)-based Vehicle Probe ; GVP) 方法 [2, 3] 將在車輛中的手機 (User Equipments ; UEs) 裝載 GPS 接收器。接收器則會把 GPS 座標和時間等資訊透過行動電信網路傳送，也就是透過基地台 (Base Stations ; BSs ; 圖 1.2 ④) 和行動交換中心 (Mobile Switching Center ; MSC ; 圖 1.2 ⑤) 傳給 ITS 伺服器，伺服器會根據收到的 GPS 座標來計算車速。
- **流動車輛上的手機信令資料** (Cellular Floating Vehicle Data ; CFVD) 方法 [4, 5, 6] 將安裝網路探測器 (Network Probe ; 圖 1.2 ⑦) 來監控 BSs 和 MSC 之間的信令 (signals)，並且把信令傳給 ITS 伺服器。根據手機的通話行為，ITS 伺服器會追蹤手機所在的細胞 (cell；一個 BS 的無線電覆蓋範圍)，估計出手機的細胞停留時間 (cell residence times)，並推導出移動速度。

第一類使用 VD 的方法在偵測器的建置和維護上需要花費較高的成本 [4, 6]。第二類 GVP 方法則需要具有 GPS 接收器的手機，並且需要額外的無線電資源來傳送 GPS 資料。此外，我們的經驗顯示，在一些鄉下道路上並不會有很多具備 GPS 接收器的車輛，這些道路的交通資訊就很難從 GPS 資料來獲得。而第三類 CFVD 的方法需要在電信網路中加上額外的信令連結線路 (signaling links) 和網路探測器，用來監測手機和行動核心網路之間的信令訊息 (signaling messages)。

林、張及皇甫的論文 [7, 8, 9] 提出了 *Lin-Chang-Huangfu* (LCH) 方法，此方法可以增強 CFVD，使 CFVD 不再需要額外的硬體收集資料。LCH 方法可以從 MSC 中標準計數器的值推導出細胞停留時間 (手機在細胞中所待的時間區段)，計數器的值是由 MSC 定期自動去收集的。因此，我們的方法可以快速計算出細胞停留時間。我們用 LCH 方法算出的細胞停留時間來估計車速，並且使用台灣桃園縣龍潭鄉國道三號上的 VD 來驗證 LCH 方法。我們更進一步去探討由 LCH 方法所計算的細胞停留時間的偏差。從這個偏差的研究結果顯示，LCH 是一個適合用來做車速估計的方法。

本論文的章節架構如下。第二章介紹如何推導出 LCH 方法，以及如何計算細胞停留時間。第三章在推導由 LCH 方法計算的細胞停留時間的偏差。第四章介紹我們在改善車速估計的準確性時所使用的技術。第五章用一些數據結果來探討 LCH 方法的表現。第六章為本篇論文之總結，並提出未來工作方向。



## 第二章 Lin-Chang-Huangfu (LCH) 方法

本章根據論文 [7] 中 LCH 方法描述如下。圖 1.2 中的 MSC (圖 1.2 ©) 要負責處理通話和行動管理 [11]。多個 BS 會連接到此 MSC。一個 BS 的無線電覆蓋範圍我們稱為細胞 (虛線的圓圈；圖 1.2 ©)。當手機通話時，細胞中的手機會透過 BS 連接到 MSC。如果手機在通話時從原本細胞移動至其他細胞，手機的通話路徑就會從舊的細胞換到新的細胞中。這個過程稱為交遞 (handover)。

在標準的商業行動電信運營中，MSC 會記錄手機的通話行為 (即手機播打/接收電話或手機在通話過程中，由一個細胞移動至另一個細胞)。MSC 在每個  $\Delta t$  中收集每個細胞通話行為的統計資料，間隔時間通常是 15 分鐘到數個小時。其中兩個統計資料分別為交遞進入細胞和離開細胞的個數，還有細胞中的語音通話時間 (單位為 Erlang)。現在考慮一個時間點  $\tau$ 。我們定義  $\Delta\tau$  為一個時間區段  $\left(\left\lfloor \frac{\tau}{\Delta t} \right\rfloor \Delta t, \left\lfloor \frac{\tau}{\Delta t} + 1 \right\rfloor \Delta t\right)$ 。為了方便敘述，我們定義估計車速的路段為「目標路段」(target road segment；圖 1.2 (h))。細胞  $i$  (圖 1.2 ©) 的平均細胞停留時間推導如下。用  $\rho(\tau)$  表示在  $\Delta\tau$  中細胞  $i$  的運送流量 (carried traffic)。換句話說， $\rho(\tau)$  是在  $\Delta\tau$  到達細胞  $i$  的通話個數乘以期望的運送通話保留時間 (carried call holding times；單位為分鐘)。運送通話保留時間和供應通話保留時間 (offered call holding time) 是不同的。供應通話保留時間是假設所有 BS 的無線電通道數沒有極限時，一通電話的通話時間 (單位為分鐘)，這時所有的新通話和交遞都會成功完成。但在實際情況下，BS 的通道容量是有極限的。因此，通話可能會在一起始就被阻擋，或是在交遞的過程中被強制結束。一通電話話連接後，通話時間會被 MSC 記錄在統計資料  $\rho(\tau)$  中，並且定義為運送通話保留時間  $t_c$ 。MSC 中量測到的是運送通話保留時間，在通話保留時間  $t_c$  中，通話不會在一起始就被阻擋，或是在交遞的過程中被強制結束 (雖然通話可能會在  $t_c$  結束時被強制終止)。讓  $\gamma(\tau)$  表示在  $\Delta\tau$  中交遞進入細胞  $i$  的通話個數。用  $t_m$  表示為細胞停留時間。在論文 [7] 中，作者由 Little's Law [12] 推導在

時間區段  $\Delta\tau$  中到達細胞  $i$  手機的細胞停留時間  $t_m^*(\tau)$ 。Little's Law 可以改寫為

$$N(\tau) = \lambda_u(\tau)t_m^*(\tau) \quad (2.1)$$

其中  $N(\tau)$  為在  $\Delta\tau$  期間細胞  $i$  中的期望使用者個數， $\lambda_u(\tau)$  為在  $\Delta\tau$  期間使用者進入細胞  $i$  的到達速率。用  $t_a$  表示通話到達時間間隔 (inter-call arrival time)，則一個使用者在  $\Delta\tau$  會有  $(\Delta t/t_a)$  個通話到達，並且會貢獻  $t_c(\Delta t/t_a)$  通話時間。因此

$$\rho(\tau) = \frac{N(\tau)t_c\Delta t}{t_a} \text{ 或是 } N(\tau) = \frac{\rho(\tau)t_a}{t_c\Delta t} \quad (2.2)$$

由於  $\lambda_u(\tau)$  為在  $\Delta\tau$  期間使用者進入細胞  $i$  的到達速率。在  $\Delta\tau$  中會有  $\lambda_u(\tau)\Delta t$  個使用者進入細胞  $i$ 。這些使用者中，會有  $(t_c/t_a)$  個正在通話。換句話說，在  $\Delta\tau$  期間會有  $(\lambda_u(\tau)t_c\Delta t/t_a)$  個使用者交遞進入細胞  $i$ 。並且

$$\gamma(\tau) = \frac{\lambda_u(\tau)t_c\Delta t}{t_a} \text{ 或是 } \lambda_u(\tau) = \frac{\gamma(\tau)t_a}{t_c\Delta t} \quad (2.3)$$

經由方程式 (2.1)、(2.2) 和 (2.3) 可知

$$t_m^*(\tau) = \frac{\rho(\tau)}{\gamma(\tau)} \quad (2.4)$$

基於方程式 (2.4)，我們接著推導目標路段的平均車速。假設細胞  $i-1$  中裝載手機的車輛 (圖 1.2 ⑥)，經由細胞  $i$  到達細胞  $i+1$ 。讓  $x$  表示細胞  $i$  覆蓋的目標路段長度。從方程式 (2.4) 可以知道，目標路段在  $\Delta\tau$  中的平均車速  $v(\tau)$  推導為

$$v(\tau) = \frac{x}{t_m^*(\tau)} = \frac{x\gamma(\tau)}{\rho(\tau)} \quad (2.5)$$

在下一章節，本論文推導由 LCH 方法所計算的細胞停留時間的偏差。由偏差可以顯示出 LCH 方法是用來估計車速的好計量器。

### 第三章 細胞停留時間估計的偏差 (Bias)

這一章節以估計 LCH 方法的偏差來評估此方法的準確性。我們將會顯示 LCH 方法相對於一般細胞停留時間估計方法，在車速的估計上有較小的偏差。因此，LCH 是一個適合用來估計車速的方法。

假設  $t_m$  為實際的細胞停留時間，根據方程式 (2.4) 的定義， $t_m^*(\tau)$  是  $t_m$  的估計器。細胞停留時間的偏差就可以定義為

$$b_{t_m}(t_m^*(\tau)) = E[t_m^*(\tau)] - E[t_m] \quad (3.1)$$

須要注意的是，在統計學中偏差和誤差 (error) 是不同的。估計器的偏差指的是估計值和實際值之間的差異 [13]。我們首先推導受  $\Delta\tau$  限制的  $\gamma(\tau)$  和  $\rho(\tau)$ 。接著我們利用推導出來的  $\gamma(\tau)$  和  $\rho(\tau)$  公式來表示細胞停留時間估計器  $t_m^*(\tau)$ 。最後再代入方程式 (3.1) 來計算偏差。

圖 3.1 描述 6 個手機在觀察時段  $[t_0, t_{19}]$  活動的時間圖。以圖 3.1 中 UE 3 的行為做例子。UE 3 在時間點  $t_8$  移動至細胞  $i$  (以符號  $\Delta$  標示)，並且在時間點  $t_{11}$  有一通電話到達 UE 3 (以符號  $\circ$  標示)。UE 3 在時間點  $t_{13}$  離開細胞  $i$  (以符號  $\blacktriangle$  標示)，並且在時間點  $t_{15}$  UE 3 的通話結束 (以符號  $\bullet$  標示)。UE 3 的細胞停留時間為  $t_m = t_{13} - t_8$ 。UE 3 的運送通話保留時間為  $t_c = t_{15} - t_{11}$ 。

考慮時間區段  $\Delta\tau = t_{19} - t_5$  (即圖 3.1 中灰色區塊)。在  $\Delta\tau$  期間從細胞  $i$  觀測到的通話會是下三種型態的一種：

- **新通話 (new call)** 是起始於  $\Delta\tau$  中的通話；例如在圖 3.1 中的時間點  $t_{10}$  到達 UE 6 的通話和時間點  $t_{11}$  到達 UE 3 的通話。
- **已存在的通話 (existing call)** 是在  $\Delta\tau$  開始前就已經連接到細胞  $i$  的通話；例如 UE 1 的通話，此通電話在  $\Delta\tau$  前的時間點  $t_4$  就到達 UE 1，此時 UE 1 已經待在細胞



$i$ ；換句話說，UE 1 在  $t_5$  前就到達細胞  $i$ ，通話在  $t_4 < t_5$  開始，並且仍然進行著。

- **交遞通話 (handover call)** 是指從一個細胞轉換到另一個細胞的通話；例如 UE 2 在時間  $t_6$  發生交遞、UE 4 在時間  $t_9$  發生交遞和 UE 5 在時間  $t_{12}$  發生交遞。

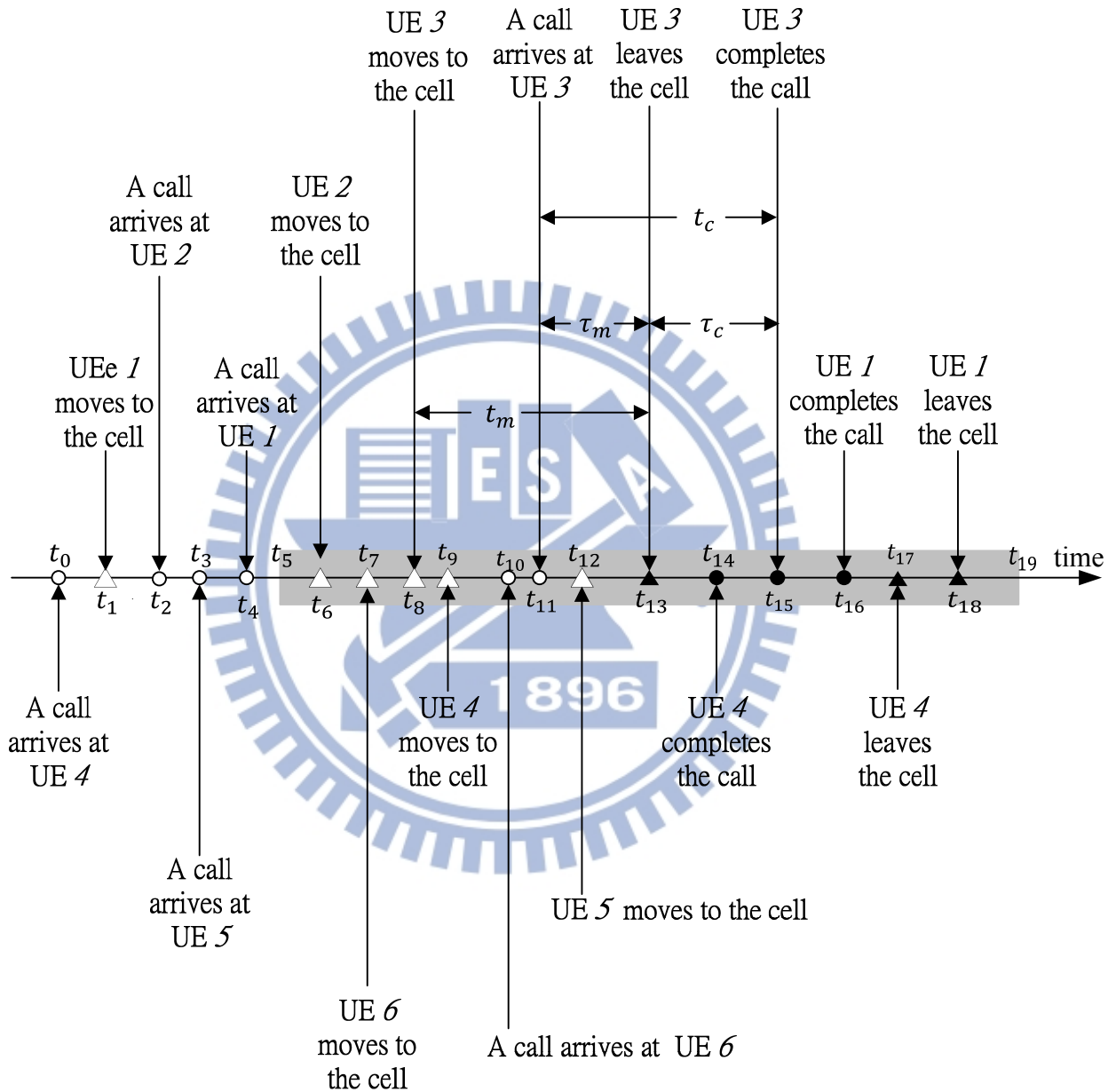


圖 3.1: 手機移動和通話抵達的時間圖 ( $\Delta$ : 一個手機移動至細胞  $i$ ,  $\blacktriangle$ : 一個手機離開細胞  $i$ ,  $\circ$ : 一通電話到達,  $\bullet$ : 一通電話結束,  $\blacksquare$ :  $\Delta\tau$ )

用  $\alpha(\tau)$ 、 $\beta(\tau)$  和  $\gamma(\tau)$  分別表示新通話、已存在的通話和交遞通話在  $\Delta\tau$  中待在細胞  $i$  的個數。在圖 3.1 中， $\alpha(\tau) = 2$  (即是指  $t_{10}$  和  $t_{11}$  到達的電話)， $\beta(\tau) = 1$  (即是指  $t_4$  到

達的電話)， $\gamma(\tau) = 3$  (即是指發生在  $t_6$ 、 $t_9$  和  $t_{12}$  的交遞)。用  $\lambda$  表示通話到達速率 (call arrival rate)。很明顯地知道

$$E[\alpha(\tau)] = \lambda\Delta\tau \quad (3.2)$$

用  $t_m$  表示平均值 (mean) 為  $\frac{1}{\eta}$  的隨機變數， $t_c$  表示平均值為  $\frac{1}{\mu}$  的隨機變數。則我們可以得到以下事實 (facts)。

**Fact 1.** 假設  $t_c$  是指數分佈 (exponential distribution)。假設  $t_m$  有一個任一隨機密度函數 (arbitrary density function)  $f_m(t_m)$ ，其速率為  $\eta$ 。假設時間區段  $\Delta\tau$  是一個任意隨機分佈 (arbitrary distribution)，其平均值為  $\frac{1}{\delta}$ ，則  $E[\beta(\tau)] = \frac{\lambda}{\mu}$ ， $E[\gamma(\tau)] = \frac{\lambda\eta}{\delta\mu}$ 。

**Proof:** 考慮時間區段  $\Delta\tau$  的起始點。用此時間點觀察細胞  $i$  中的通話行為。已存在通話的期望個數  $E[\beta(\tau)]$  會和  $E[\alpha(\tau)]$  及期望通話保留時間  $\frac{1}{\mu}$  成正比例關係，並且和期望時間區段長度  $E[\Delta\tau]$  成反比例關係。因此，我們可以表示  $E[\beta(\tau)]$  如

$$E[\beta(\tau)] = \frac{E[\alpha(\tau)](1/\mu)}{E[\Delta\tau]} = \frac{\lambda}{\mu} \quad (3.3)$$

直觀上，假如  $\frac{1}{\mu} < E[\Delta\tau]$ ，則  $\frac{1}{\mu E[\Delta\tau]}$  是在  $\Delta\tau$  中為已存在通話的機率。於是已存在通話的總數量  $E[\beta(\tau)]$  是  $E[\alpha(\tau)]$  乘以  $\frac{1}{\mu E[\Delta\tau]}$ ，結果如方程式 (3.3)。

明顯地， $E[\gamma(\tau)]$  可以表示為  $E[\alpha(\tau)]$  乘以一通電話的平均交遞次數。由於指數分佈的失憶 (memoryless) 特性，交遞通話的剩餘通話保留時間 (residual call holding time；即是圖 3.1 中的  $\tau_c = t_{15} - t_{13}$ ) 也會是指數分佈，其平均值為  $\frac{1}{\mu}$ 。讓  $t_m$  的分佈函數 (distribution function) 為  $F_m(t_m)$ 。讓剩餘細胞停留時間 (residual cell residence time) 為  $\tau_m$  (即是圖 3.1 中的  $\tau_m = t_{13} - t_{11}$ )。根據剩餘時間理論 (residual life theorem) [14]， $\tau_m$  的密度函數 (density function) 為  $\frac{1 - F_m(t_m)}{(1/\eta)} = [1 - F_m(t_m)]\eta$ 。須要注意的是，量測到的  $t_c$  被稱為運送通話保留時間。因此，假如一個交遞發生，在新的細胞會有一個可用的無線電通道，並且交遞總是成功。定義  $P_i$  為一新通話在交遞  $i$  次後結束通話的機率。定義  $q_1 = \Pr[t_c > \tau_m]$  為一個新通話發生交遞的機率，並且定義  $q_2 = \Pr[\tau_c > t_m]$  為一個通話在曾經交遞的情況下發生交遞的機率。讓  $P_i = q_1 q_2^{i-1} (1 - q_2)$ ，其中  $i \geq 1$ 。於是一通電話的平均交遞次數可以表示為

$$\sum_{i=1}^{\infty} iP_i = \sum_{i=1}^{\infty} i q_1 q_2^{i-1} (1 - q_2) = \frac{q_1}{1 - q_2} \quad (3.4)$$

在方程式 (3.4) 中，機率  $q_1$  的推導如下

$$\begin{aligned} q_1 = \Pr[t_c > \tau_m] &= \int_{\tau_m=0}^{\infty} \int_{t_c=\tau_m}^{\infty} \mu e^{-\mu t_c} [1 - F_m(\tau_m)] \eta dt_c d\tau_m \\ &= \int_{\tau_m=0}^{\infty} e^{-\mu \tau_m} [1 - F_m(\tau_m)] \eta d\tau_m \end{aligned} \quad (3.5)$$

而  $1 - q_2$  可以表示為

$$\begin{aligned} 1 - q_2 = \Pr[\tau_c < t_m] &= \int_{\tau_c=0}^{\infty} \int_{t_m=\tau_c}^{\infty} \mu e^{-\mu \tau_c} f_m(t_m) dt_m d\tau_c \\ &= \int_{\tau_c=0}^{\infty} \mu e^{-\mu \tau_c} [1 - F_m(\tau_c)] d\tau_c \\ &= \left(\frac{\mu}{\eta}\right) q_1 \end{aligned} \quad (3.6)$$

經由方程式 (3.5) 和 (3.6) 可以知道，一通電話的平均交遞次數為  $\frac{q_1}{1 - q_2} = \frac{\eta}{\mu}$ ，則  $E[\gamma(\tau)]$  可以推導如下

$$E[\gamma(\tau)] = E[\alpha(\tau)] \left(\frac{\eta}{\mu}\right) = \frac{\lambda \eta}{\delta \mu} \quad (3.7)$$

方程式 3.2、3.3 和 3.7 可由從附錄 A 的方法來驗證其正確性。並從以上的證明可以得知， $E[\alpha(\tau)]$ 、 $E[\beta(\tau)]$  和  $E[\gamma(\tau)]$  的推導與  $\Delta\tau$  的分佈是獨立的。

**Fact 2.** 假設  $t_c$  和  $\Delta\tau$  都是指數分佈，平均值分別為  $\frac{1}{\mu}$  和  $\frac{1}{\delta}$ ，並且  $t_m$  有一個任意隨機密度函數  $f_m(t_m)$ ，其速率為  $\eta$ 。於是  $E[\rho(\tau)] = \frac{\lambda}{\delta \mu}$ 。

**Proof:** 如同 Fact 1 中的定義，用  $\tau_c$  表示  $t_c$  的剩餘時間， $\tau_m$  表示  $t_m$  的剩餘時間。對於細胞  $i$  的時間區段  $\Delta\tau$ ， $\rho(\tau)$  是由新通話、已存在通話和交遞通話的談話時間所貢獻。考慮圖 3.1 時間區段  $\Delta\tau$  中 UE 3 的新通話。此通話在時間點  $t_{11}$  到達，其通話時間  $t_c = t_{15} - t_{11}$ 。在時間點  $t_{11}$ ，UE 3 的剩餘細胞停留時間為  $\tau_m = t_{13} - t_{11}$ 。由於通話到達的情況為卜松過程 (Poisson Process)，時間點  $t_{11}$  為 UE 3 細胞停留時間的隨機觀察者 (random observer)。時間點  $t_{11}$  也同是  $\Delta\tau$  的隨機觀察者。從剩餘時間理論和指數分佈的失憶特性可知，時間點  $t_{11}$  觀察到的  $\Delta\tau$  剩餘時間  $\Delta\tau_n$ ，也會是指數分佈，其平均值為  $\frac{1}{\delta}$ 。因此，此新通話在  $\Delta\tau$  中的期望通話時間  $E[\rho_n(\tau)]$

為

$$\begin{aligned} E[\rho_n(\tau)] &= E[\min(t_c, \tau_m, \Delta\tau_n)] \\ &= E[\min(\min(t_c, \Delta\tau_n), \tau_m)] \end{aligned} \quad (3.8)$$

明顯地，隨機變數  $x_n = \min(t_c, \Delta\tau_n)$  是指數分佈，其速率為  $\mu + \delta$ 。讓  $f_m(t_m)$  的拉普拉斯轉換為  $f_m^*(t_m)$ 。從附錄 B 中的推導，方程式 (3.8) 可被改寫為

$$\begin{aligned} E[\rho_n(\tau)] &= E[\min(x_n, \tau_m)] \\ &= \frac{1}{\mu + \delta} - \left[ \frac{\eta}{(\mu + \delta)^2} \right] [1 - f_m^*(\mu + \delta)] \end{aligned} \quad (3.9)$$

接著考慮圖 3.1 在  $\Delta\tau$  中 UE 1 的已存在通話。此通話在時間點  $t_4$  到達。在時間點  $t_5$ ，剩餘通話保留時間為  $\tau_c = t_{16} - t_5$ ，剩餘細胞停留時間為  $\tau_m = t_{18} - t_5$ 。由於  $t_5$  可被視為 UE 1 的通話保留時間  $t_c$  的隨機觀察者，因此  $\tau_c$  也為指數分佈，其平均值為  $\frac{1}{\mu}$ 。於是此已存在通話在  $\Delta\tau$  中的期望通話時間為

$$E[\rho_e(\tau)] = E[\min(\tau_c, \tau_m, \Delta\tau)]$$

由於  $\tau_c$  和  $\Delta\tau$  皆是指數分佈，速率分別為  $\mu$  和  $\delta$ ， $E[\rho_e(\tau)]$  可表示為方程式 (3.9)；換句話說， $E[\rho_e(\tau)] = E[\rho_n(\tau)]$ 。

最後考慮圖 3.1 在  $\Delta\tau$  中 UE 4 的交遞通話。此使用者在時間點  $t_9$  移至細胞  $i$ ，其細胞停留時間為  $t_m = t_{17} - t_9$ 。在時間點  $t_9$ ，此通話的剩餘通話保留時間為  $\tau_c = t_{14} - t_9$ 。由於  $t_9$  可被視為 UE 4 通話保留時間  $t_c$  的隨機觀察者， $t_c$  為指數分佈，其平均值為  $\frac{1}{\mu}$ ， $\tau_c$  則和  $t_c$  有一樣的分佈。時間點  $t_9$  也是  $\Delta\tau$  的隨機觀察者，則從時間點  $t_9$  觀察  $\Delta\tau$  的剩餘時間  $\Delta\tau_h$  也會是指數分佈，其平均值為  $\frac{1}{\delta}$ 。因此，此交遞通話在  $\Delta\tau$  中的期望通話時間  $E[\rho_h(\tau)]$  為

$$\begin{aligned} E[\rho_h(\tau)] &= E[\min(\tau_c, t_m, \Delta\tau_h)] \\ &= E[\min(\min(\tau_c, \Delta\tau_h), t_m)] \end{aligned} \quad (3.10)$$

變數  $x_h$  可表示為  $x_h = \min(\tau_c, \Delta\tau_h)$ 。假設其為指數分佈，速率為  $\mu + \delta$ 。於是和

方程式 (3.9) 的推導類似，方程式 (3.10) 可被重新改寫為

$$E[\rho_h(\tau)] = E[\min(x_h, t_m)] = \left( \frac{1}{\mu + \delta} \right) [1 - f_m^*(\mu + \delta)] \quad (3.11)$$

經由方程式 (3.2)、(3.9)、(3.11) 和 Fact 1， $E[\rho(\tau)]$  可表示為

$$\begin{aligned} E[\rho(\tau)] &= E[\alpha(\tau)]E[\rho_n(\tau)] + E[\beta(\tau)]E[\rho_e(\tau)] + E[\gamma(\tau)]E[\rho_h(\tau)] \\ &= (E[\alpha(\tau)] + E[\beta(\tau)]) \left\{ \frac{1}{\mu + \delta} - \left[ \frac{\eta}{(\mu + \delta)^2} \right] [1 - f_m^*(\mu + \delta)] \right\} \\ &\quad + E[\gamma(\tau)] \left\{ \left( \frac{1}{\mu + \delta} \right) [1 - f_m^*(\mu + \delta)] \right\} \\ &= \frac{\lambda}{\delta\mu} \end{aligned}$$

**Fact 3.** 假設  $\Delta\tau$  為一固定值  $\frac{1}{\delta}$ 。假如通話保留時間  $t_c$  和細胞停留時間  $t_m$  皆是指數分佈，則  $E[\rho(\tau)] = \frac{\lambda}{\delta\mu}$ 。

**Proof:** 如同 Fact 2 的證明， $\rho(\tau)$  是由新通話、已存在通話和交遞通話所貢獻。新通話在  $\Delta\tau$  中的期望通話時間  $E[\rho_n(\tau)]$  為

$$\begin{aligned} E[\rho_n(\tau)] &= E[\min(y_n, \Delta\tau_n)] \\ &= E[\min(\min(t_c, \tau_m), \Delta\tau_n)] \end{aligned} \quad (3.12)$$

其結果和方程式 (3.8) 一樣，除了  $\Delta\tau_n$  是在  $\left[0, \frac{1}{\delta}\right]$  區間的均勻分佈 (uniform distribution)，其密度函數為  $\delta$ 。由於  $t_c$  和  $\tau_m$  皆為指數分佈，速率分別為  $\mu$  和  $\eta$ ，因此隨機變數  $y_n = \min(t_c, \tau_m)$  是指數分佈，其速率為  $\mu + \eta$ ，而方程式 (3.12) 可被改寫為

$$\begin{aligned} E[\rho_n(\tau)] &= \int_{\Delta\tau_n=0}^{\frac{1}{\delta}} \int_{y_n=0}^{\Delta\tau_n} \delta y_n (\mu + \eta) e^{-(\mu+\eta)y_n} dy_n d\Delta\tau_n \\ &\quad + \int_{\Delta\tau_n=0}^{\frac{1}{\delta}} \int_{y_n=\Delta\tau_n}^{\infty} \delta \Delta\tau_n (\mu + \eta) e^{-(\mu+\eta)y_n} dy_n d\Delta\tau_n \\ &= \left( \frac{1}{\mu + \eta} \right) \left[ 1 - \frac{\delta}{\mu + \eta} + \frac{\delta e^{-\left(\frac{\mu+\eta}{\delta}\right)}}{\mu + \eta} \right] \end{aligned} \quad (3.13)$$



接著考慮在  $\Delta\tau$  中交遞通話的期望通話時間  $E[\rho_h(\tau)]$  為

$$E[\rho_h(\tau)] = E[\min(\tau_c, t_m, \Delta\tau_h)] \quad (3.14)$$

$\tau_c$  和  $t_m$  皆是指數分佈，速率分別為  $\mu$  和  $\eta$ ，而  $\Delta\tau_h$  是在  $\left[0, \frac{1}{\delta}\right]$  區間的均勻分佈。因此方程式 (3.14) 的表示和方程式 (3.12) 相同；換句話說， $E[\rho_h(\tau)] = E[\rho_n(\tau)]$ 。已存在通話在  $\Delta\tau$  中的期望通話時間  $E[\rho_e(\tau)]$  為

$$\begin{aligned} E[\rho_e(\tau)] &= E\left[\min\left(y_e, \frac{1}{\delta}\right)\right] \\ &= E\left[\min\left(\min(\tau_c, \tau_m), \frac{1}{\delta}\right)\right] \end{aligned} \quad (3.15)$$

其中  $y_e = \min(\tau_c, \tau_m)$  為指數分佈，速率為  $\mu + \eta$ ，方程式 (3.15) 可被改寫為

$$\begin{aligned} E[\rho_e(\tau)] &= \int_{y_e=0}^{\frac{1}{\delta}} y_e(\mu + \eta)e^{-(\mu+\eta)y_e} dy_e + \int_{y_e=\frac{1}{\delta}}^{\infty} \left(\frac{1}{\delta}\right) (\mu + \eta)e^{-(\mu+\eta)y_e} dy_e \\ &= \left(\frac{1}{\mu + \eta}\right) [1 - e^{-\left(\frac{\mu+\eta}{\delta}\right)}] \end{aligned} \quad (3.16)$$

經由方程式 (3.2)、(3.13)、(3.16) 和 Fact 1 可知， $E[\rho(\tau)]$  可以表示為

$$\begin{aligned} E[\rho(\tau)] &= E[\alpha(\tau)]E[\rho_n(\tau)] + E[\beta(\tau)]E[\rho_e(\tau)] + E[\gamma(\tau)]E[\rho_h(\tau)] \\ &= (E[\alpha(\tau)] + E[\gamma(\tau)]) \left(\frac{1}{\mu + \eta}\right) \left[1 - \frac{\delta}{\mu + \eta} + \frac{\delta e^{-\left(\frac{\mu+\eta}{\delta}\right)}}{\mu + \eta}\right] \\ &\quad + E[\beta(\tau)] \left(\frac{1}{\mu + \eta}\right) [1 - e^{-\left(\frac{\mu+\eta}{\delta}\right)}] \\ &= \frac{\lambda}{\delta\mu} \end{aligned}$$

**Fact 4.** 假設  $\Delta\tau$ 、 $t_c$  和  $t_m$  都是任意分佈，平均值分別為  $\frac{1}{\delta}$ 、 $\frac{1}{\mu}$  和  $\frac{1}{\eta}$ 。於是  $E[\gamma(\tau)] \approx \frac{\lambda\eta}{\delta\mu}$ 。

**Proof:** 考慮一個很長的觀察時間區段  $t \gg \frac{1}{\delta}$ 、 $\frac{1}{\mu}$ 。讓  $t^*$  為在時間區段  $t$  期間，一個使用者被通話佔據的時間。於是在  $t$  時間中會有  $\left\lfloor \frac{t^*}{(1/\mu)} \right\rfloor = \lceil \mu t^* \rceil$  個通話到達。須要注意的是，使用者在時間區段  $t$  中被期望跨越細胞  $\left\lfloor \frac{t}{(1/\eta)} \right\rfloor$  次，而在這些跨越中，有  $\left\lfloor \frac{t^*}{(1/\eta)} \right\rfloor$  次是發生在通話中。因此，交遞的次數為  $\left\lfloor \frac{t^*}{(1/\eta)} \right\rfloor = \lceil \eta t^* \rceil$ ，並且一通

電話的平均交遞次數為  $E[n_1] = \frac{\lfloor \eta t^* \rfloor}{\lfloor \mu t^* \rfloor}$ 。明顯地， $\lim_{t^* \rightarrow \infty} E[n_1] = \frac{\eta}{\mu}$ ，因此假如  $t$  時間長度夠長，我們可以得到

$$E[\gamma(\tau)] = E[\alpha(\tau)]E[n_1] \approx \frac{\lambda\eta}{\delta\mu} \quad (3.17)$$

**Fact 5.** 假設  $\Delta\tau = \frac{1}{\delta}$  為一定值，並且  $t_c$  和  $t_m$  都是任意分佈，平均值分別為  $\frac{1}{\mu}$  和  $\frac{1}{\eta}$ 。讓  $t$  為觀察區段。於是  $\lim_{t \rightarrow \infty} E[\rho(\tau)] = \frac{\lambda}{\delta\mu}$ 。

**Proof:** 我們假設手機在  $n_2$  個細胞中移動，在時間區段  $t \gg \frac{n_2}{\delta}, \frac{n_2}{\mu}$  中觀察他們的行為。由於這些手機都不會離開這  $n_2$  個細胞，這些通話的交遞和  $t_m$  的分佈皆不會影響通話時間的統計。在時間  $t$  期間會有  $\lceil \lambda t n_2 \rceil$  個通話到達這  $n_2$  個細胞。在這些通話中，有些通話會在時間區段  $t$  前就已經開始，這些都不會算入統計資料（我們稱為初始影響；initial effect）。還有一些被統計的通話，在時間區段  $t$  結束時還不會完成通話（我們稱為終端影響；end effect）。讓  $t_{c,i}$  為第  $i$  個通話的通話保留時間，並且  $f_c(t_{c,i})$  為第  $i$  個通話的通話保留時間的密度函數，其速率為  $\mu$ 。於是在時間  $t$  期間，這  $n_2$  個細胞所量測到的通話時間  $\theta$  近似為

$$\theta = \sum_{i=1}^{\lceil \lambda t n_2 \rceil} \int_{t_{c,i}=0}^{\infty} t_{c,i} f_c(t_{c,i}) dt_{c,i} = \frac{\lceil \lambda t n_2 \rceil}{\mu}$$

總共會有  $n_2 \left\lceil \frac{t}{(1/\delta)} \right\rceil = \lceil n_2 t \delta \rceil$  個時間區段。由於  $E[\rho(\tau)]$  為在  $\Delta\tau$  期間細胞  $i$  的期望通話時間，此值可近似為  $E[\rho(\tau)] \approx \frac{\theta}{\lceil n_2 t \delta \rceil}$ 。假如  $t$  變大，則初始影響及終端影響會變得微不足道，並且

$$\lim_{t \rightarrow \infty} E[\rho(\tau)] = \frac{\lambda}{\delta\mu} \quad (3.18)$$

須要注意的是，Fact 5 的結論會與 Fact 2 和 Fact 3 相同。這些結論和耳朗方程式 (Erlang equation) 一致，此方程式定義每分鐘的流量為  $\frac{\lambda}{\mu}$ 。因此，在時間區段  $\Delta\tau$  的流量為  $\frac{\lambda}{\delta\mu}$ 。

**Fact 6.** 假設  $t_c$  和  $t_m$  皆是任意分佈，平均值分別為  $\frac{1}{\mu}$  和  $\frac{1}{\eta}$ 。假如時間區段  $\Delta\tau = \frac{1}{\delta}$  為一定值，並且我們在一長時間  $t$  觀察通話的行為，於是  $b_{t_m}(t_m^*(\tau)) = \frac{\mu\delta}{\lambda\eta^2}$ 。

**Proof:** 假設我們在一長時間  $t \rightarrow \infty$  觀察通話的行為。由於 LCH 方法的量測都在  $\Delta\tau$  中

進行，從方程式 (3.17) 和 (3.18) 可知，假如  $\frac{1}{\delta}$  趨近於無限大，則  $E[\gamma(\tau)]$  和  $E[\rho(\tau)]$  皆會趨近無限大，並且方程式 (2.4) 就無法用來計算  $t_m^*(\tau)$ 。為了解決此問題，讓  $c_1 = \frac{E[\rho(\tau)]}{(1/\delta)} = \frac{\lambda}{\mu}$  並且  $c_2 = \frac{E[\gamma(\tau)]}{(1/\delta)} = \frac{\lambda\eta}{\mu}$ 。於是我們可以改寫方程式 (2.4) 為函式  $f(r, s)$ ，使得

$$t_m^*(\tau) = f(r, s)|_{r=c_1, s=c_2} = \left(\frac{r}{s}\right)\Big|_{r=c_1, s=c_2} \quad (3.19)$$

為了計算方程式 (3.19)，我們使用泰勒展開式 (Taylor series) 逼近函式  $f(r, s)$  在座標點  $(c_1, c_2)$  的數值，其值為

$$\begin{aligned} f(r, s) &= f(c_1, c_2) + \left[ \frac{\partial f(r, s)}{\partial r} \Big|_{r=c_1, s=c_2} \right] (r - c_1) \\ &\quad + \left[ \frac{\partial f(r, s)}{\partial s} \Big|_{r=c_1, s=c_2} \right] (s - c_2) + \left( \frac{1}{2} \right) \left[ \frac{\partial^2 f(r, s)}{\partial r^2} \Big|_{r=c_1, s=c_2} \right] (r - c_1)^2 \\ &\quad + \left[ \frac{\partial}{\partial s} \left( \frac{\partial f(r, s)}{\partial r} \right) \Big|_{r=c_1, s=c_2} \right] (r - c_1)(s - c_2) \\ &\quad + \left( \frac{1}{2} \right) \left[ \frac{\partial^2 f(r, s)}{\partial s^2} \Big|_{r=c_1, s=c_2} \right] (s - c_2)^2 + \dots \end{aligned} \quad (3.20)$$

由於我們考慮函式  $f(r, s)$  在座標點  $(c_1, c_2)$  的數值，並且期望值  $E[r] = c_1$  和  $E[s] = c_2$ ，我們可以得到

$$E[r - c_1] = 0, E[s - c_2] = 0 \text{ 以及 } E[(r - c_1)(s - c_2)] = 0 \quad (3.21)$$

並且，從方程式 (3.19) 可知

$$\frac{\partial^2 f(r, s)}{\partial r^2} \Big|_{r=c_1, s=c_2} = 0 \text{ 以及 } \frac{\partial^2 f(r, s)}{\partial s^2} \Big|_{r=c_1, s=c_2} = \frac{2c_1}{c_2^3} \quad (3.22)$$

經由方程式 (3.21) 和 (3.22)，方程式 (3.20) 的期望值可以描述為

$$\begin{aligned} E[f(r, s)] &= f(c_1, c_2) + \left( \frac{1}{2} \right) \left[ \frac{\partial^2 f(r, s)}{\partial s^2} \Big|_{r=c_1, s=c_2} \right] E[(s - c_2)^2] \\ &= \frac{c_1}{c_2} + \left( \frac{c_1}{c_2^3} \right) E[(s - c_2)^2] \\ &= \frac{1}{\eta} + \left( \frac{\mu^2}{\lambda^2 \eta^3} \right) E[(s - c_2)^2] \end{aligned} \quad (3.23)$$



在方程式 (3.23) 中， $E[(s - c_2)^2] = Var \left[ \frac{\gamma(\tau)}{(1/\delta)} \right] = \delta^2 Var[\gamma(\tau)]$ 。從方程式 (3.7) 可知， $\gamma(\tau)$  為卜松隨機變數 (Poisson random variable)，其平均值為  $E[\gamma(\tau)] = \frac{\lambda\eta}{\delta\mu}$  並且其變異數 (variance) 為  $Var[\gamma(\tau)] = \frac{\lambda\eta}{\delta\mu}$ 。因此，方程式 (3.23) 可被改寫為

$$\begin{aligned} E[f(r, s)] &= \frac{1}{\eta} + \left( \frac{\mu^2}{\lambda^2\eta^3} \right) Var \left[ \frac{\gamma(\tau)}{\left( \frac{1}{\delta} \right)} \right] \\ &= \frac{1}{\eta} + \frac{\mu\delta}{\lambda\eta^2} \end{aligned} \quad (3.24)$$

由於  $t_m$  其平均值為  $\frac{1}{\eta}$ ，從方程式 (3.1) 和 (3.24) 可知， $b_{t_m}(t_m^*(\tau))$  為

$$b_{t_m}(t_m^*(\tau)) = E[t_m^*(\tau)] - E[t_m] = \left[ \frac{1}{\eta} + \frac{\mu\delta}{\lambda\eta^2} \right] - \frac{1}{\eta} = \frac{\mu\delta}{\lambda\eta^2} \quad (3.25)$$

方程式 (3.25) 指出 LCH 方法在通話保留時間長 (即  $\mu$  值小)、 $\Delta\tau$  時間長 (即  $\delta$  值小)、通話到達速率  $\lambda$  大，或是細胞停留時間短 (即  $\eta$  值大) 時有比較好的準確性。因為車輛通常比行人有較短的細胞停留時間 (即較快的速度)，方程式 (3.25) 指出 LCH 方法在車速估計上，比一般包含行人的細胞停留時間估計方法有較高的準確性。

## 第四章 改善車速評估準確性之技術

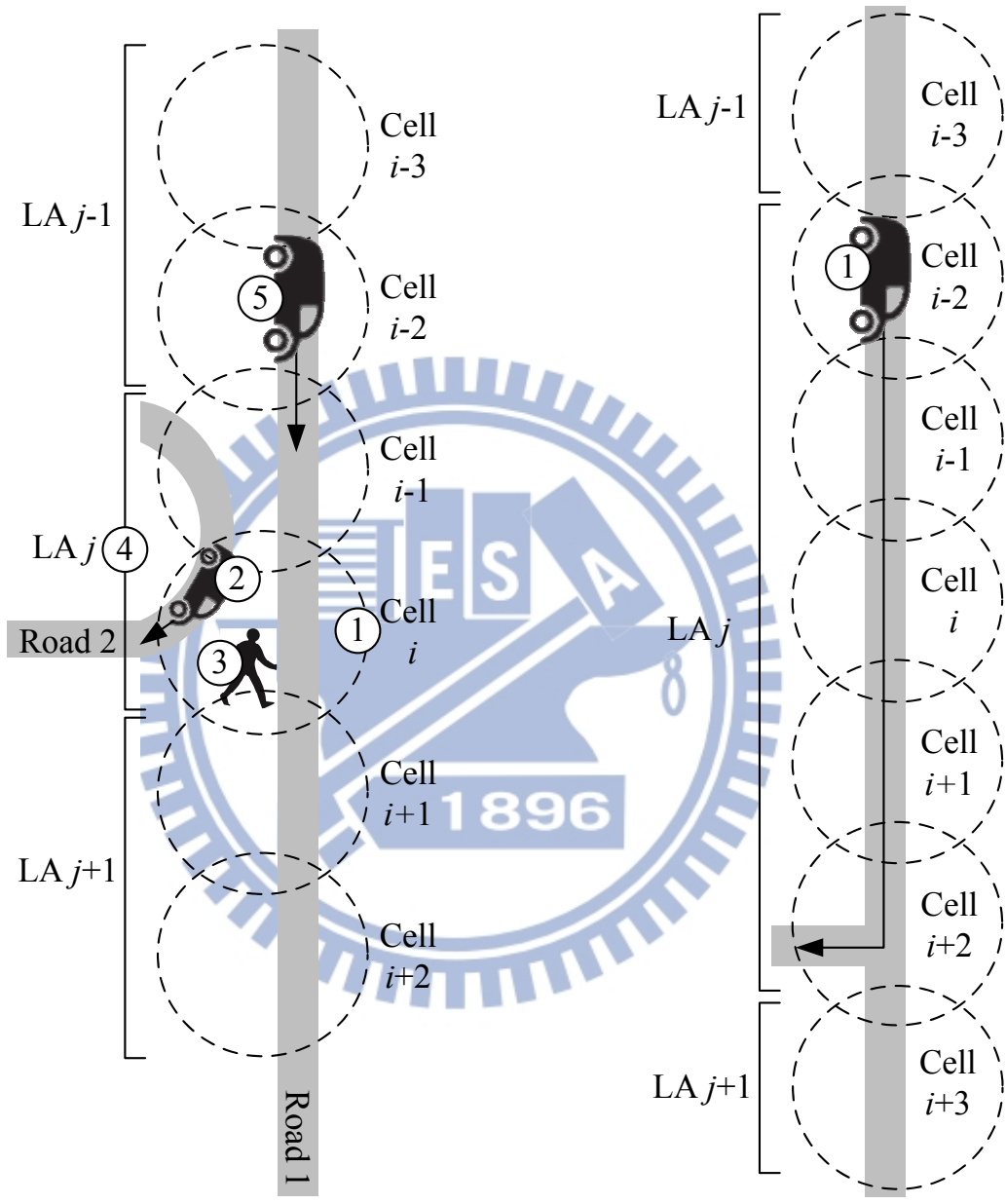
此章節描述幾個進一步改善方程式 (2.5) 車速估計準確性的技術。我們首先介紹兩個篩選技術，用來排除不在目標道路上的手機。接著我們描述兩個補償技術，以漏桶算法 (leaky-bucket) 整合策略 [16] 來增加觀察的通話個數。

假如細胞  $i$  (圖 4.1(a) 中的①) 會覆蓋目標道路之外的區域 (如圖 4.1(a) 中的② 和 ③)，這些不在目標道路上的手機，他們的通話行為也會包含在  $\gamma(\tau)$  和  $\rho(\tau)$  中。這些通話行為會降低方程式 (2.5) 的準確性。為了解決此問題，我們首先介紹行動電信網路中的標準位置更新 (location update) 程序，接著展示我們如何從細胞  $i$  中分辨出目標道路上的手機。在行動電信網路中，一群細胞會組成位置區域 (Location Areas; LAs; 例如 LA  $j$  包含細胞  $i-1$  和細胞  $i$ ; 即是圖 4.1(a) 中的④)。當一個手機 (圖 4.1(a) 中的⑤) 從原來的 LA 移動至另一個 LA，手機會執行位置更新程序，告知 MSC 新的 LA [11, 15]。位置更新訊息會從 BS 透過 MSC 傳遞到行動資料庫 (特別是 Visitor Location Register; VLR)。基於位置更新程序，我們提出以下技術來分辨出目標道路上的手機：

**過濾技術 1.** 對於每個在細胞  $i$  有通話行為的手機，當這些手機進入細胞  $i$  所屬的 LA 時，讓細胞  $A$  為手機執行位置更新的地方，並且當這些手機離開細胞  $i$  所屬的 LA 時，讓細胞  $B$  為手機執行另一個位置更新的地方。假設細胞  $A$  和細胞  $B$  皆覆蓋目標道路，則這些手機就會被辨認為在目標道路上移動。

舉例來說，當手機從 LA  $j-1$  經由 LA  $j$  移動至 LA  $j+1$  (如圖 4.1(a) 中⑤)，一個位置更新會發生在 LA  $j$  中的細胞  $i-1$  (即是細胞  $A$ )，另一個位置更新會發生在 LA  $j+1$  中的細胞  $i+1$  (即是細胞  $B$ )。因為細胞  $i-1$  和  $i+1$  皆會覆蓋在目標道路上，手機就可以被分辨為在目標道路行駛的車輛。

根據終點的不同，有些車輛會在細胞  $i$  所屬的 LA 中停止，或者從目標道路轉到其他道路 (如圖 4.1(b) 中①)。這些手機就不會被過濾條件 1 分辨出來，但是可以被以下的技術偵測出來：



(a) 一個細胞也許會覆蓋多條道路和行人

(b) 一台車輛也許會移動到另一條道路，但沒有離開細胞所屬的位置區域 (Location Area ; LA)

圖 4.1: 兩種影響車速估計準確性的情境

**過濾技術 2.** 對於每個在細胞  $i$  有通話行為的手機，假設手機的細胞交遞順序包含了至少三個覆蓋在目標道路的細胞，手機就可以被辨認為在目標道路上移動。

在圖 4.1(b)中，假設手機的細胞交遞順序包含細胞  $\{i-2, i-1, i\}$ 、細胞  $\{i-1, i, i+1\}$  或細胞  $\{i, i+1, i+2\}$ ，則手機就可以被辨認為在目標道路上。

我們使用被過濾技術 1 或 2 辨認出來的手機在細胞  $i$  的交遞資訊和通話保留時間計算目標道路的  $\gamma(\tau)$  和  $\rho(\tau)$ 。於是目標道路的平均速度可由方程式 (2.5) 計算出來。

雖然這兩個過濾技術可有效的排除不在目標道路上的手機，但是也會降低觀察通話的數量。明顯地，假如在  $\Delta\tau$  中觀察到的通話很少，則樣本就不能反映出實際目標道路的車速。為了解決此問題，我們考慮基於漏桶算法整合策略的遠歷史 (far-history) 和近歷史 (near-history) 補償技術。遠歷史補償技術使用過去數週在同一天同一個  $\Delta\tau$  的統計資料，用來補償觀察通話的數量。在過去數週同一天的交通模式通常是類似的，例如這星期一的交通模式和上星期一是一類似的。對於時間區段  $\Delta\tau$ ，讓  $\Delta\tau_k$  為在最近第  $k$  週同一天的同一個時間區段，並且讓  $\Delta\tau_0 = \Delta\tau$ 。舉例來說，假如  $\Delta\tau$  為這星期一的時間區段，則  $\Delta\tau_1$  為上星期一的同一時間區段。遠歷史補償技術會保證至少用  $K_h$  個交遞樣本計算  $v(\tau)$ 。詳細內容描述如下

**遠歷史補償技術.** 在門檻值為  $K_h$  的目標道路平均速度可計算如下

$$v(\tau) = \frac{x \left[ \sum_{k=0}^K \gamma(\tau_k) \right]}{\sum_{k=0}^K \rho(\tau_k)} \text{ where } K = \min \left\{ N : \sum_{n=0}^N \gamma(\tau_n) \geq K_h \right\} \quad (4.1)$$

值得注意的是，假如  $\gamma(\tau)$  沒有少於  $K_h$ ，方程式 (4.1) 的表示和方程式 (2.5) 相同 (即是沒有使用歷史資料)。

因為 MSC 在每個  $\Delta t$  中收集  $\rho(\tau)$  和  $\gamma(\tau)$ ，有些通話會跨過時間區段 (例如已存在通話)。這些跨越時間區段的通話會在連續的時間區段造成不平滑影響。假如連續時間區段的交通模式沒有顯著的改變，漏桶算法策略可以平滑結果。儘管如此，假如連續時間區段中車速有顯著的變化 (例如兩個時間區段的車速差距大於門檻值  $V_s$ )，則此技術就不該使用。詳細內容描述如下。

近歷史補償技術。目標道路平均速度可計算如下

$$v(\tau) \leftarrow \begin{cases} Wv(\tau) + (1 - W)v(\tau - \Delta t), & \text{for } |v(\tau) - v(\tau - \Delta t)| < V_s \\ v(\tau), & \text{for } |v(\tau) - v(\tau - \Delta t)| \geq V_s \end{cases} \quad (4.2)$$

其中  $0 \leq W \leq 1$  為權重係數 (weighting factor)， $v(\tau - \Delta t)$  為前一個時間區段  $\Delta t$  的車速，並且  $V_s$  為偵測交通模式顯著改變的門檻值。

在方程式 (4.2) 中，一個較大的  $W$  數值表示過去的車速  $v(\tau - \Delta t)$  對於現在的車速  $v(\tau)$  有較小的影響。



## 第五章 數據分析

在本章節中，我們把 LCH 方法計算的車速與 VD 方法量測的車速做比較。我們已經取得台灣桃園縣龍潭鄉國道三號(圖 5.1 中的 (a)) 的車速資料。這些車速資料是台灣交通部公佈，由高速公路 66 km 處的 VD (圖 5.1 中的 (b)) 量測而來。經由龍潭鄉的 BS (大約在高速公路 66.8 km 處，距離 VD 約 800 公尺；圖 5.1 中的 (c))，我們利用 LCH 方法、過濾技術 1 和 2 以及遠歷史和近歷史補償技術來估計車速，此細胞的長度為  $x = 1$  km，並且  $\Delta t = 1$  hour。 $x$  的長度可由 [17, 18] 測量。我們考慮  $K_h = 10$ 、 $V_s = 40$  km/hour，並且  $W = 0.5$ 。

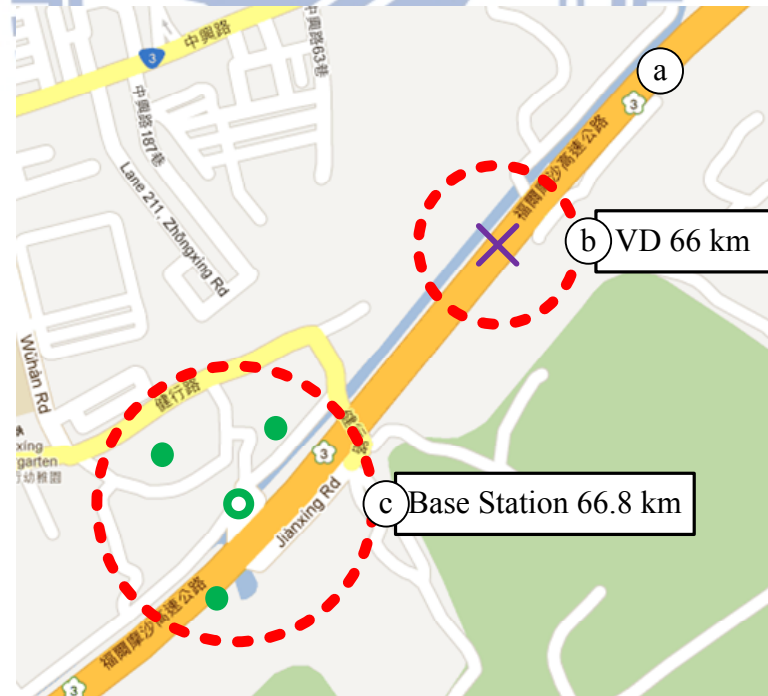


圖 5.1: 實驗環境為台灣桃園縣龍潭鄉國道三號 (以  $\times$  表示 VD，以  $\circ$  表示 BS，以  $\bullet$  表示天線的方向)



用  $v_s$  分別表示由 VD 量測的車速 ( $s = VD$ ) 和由 LCH 方法計算出來的車速 ( $s = LCH$ )。圖 5.2(a) 和 5.2(b) 分別描繪道路北向和南向在 2011 年 9 月 16 日，從 8:00 到 20:00 的  $v_s$ 。在這兩張圖中，VD 和 LCH 方法皆顯示交通狀況在 8:00 到 20:00 期間沒有顯著的變化 (例如  $v_s > 75$  km/hour)。圖 5.3(a) 和 5.3(b) 分別描繪道路北向和南向在 2011 年 9 月 25 日，從 8:00 到 20:00 的  $v_s$ 。在圖 5.3(a) 中，兩種方法皆會偵測到在 18:00 到 20:00 期間的交通堵塞 (例如  $v_s < 60$  km/hour)。在圖 5.3(b) 中，兩種方法皆顯示交通狀況在 8:00 到 20:00 期間沒有顯著的變化 (例如  $v_s > 75$  km/hour)。圖 5.2 和 5.3 顯示在兩個車行方向，LCH 方法的速度趨勢會和 VD 方法一致。我們進一步定義差異 (discrepancy)  $\epsilon$  如下

$$\epsilon = \left| \frac{v_{LCH} - v_{VD}}{v_{VD}} \right| \quad (5.1)$$

基於圖 5.2 和 5.3，圖 5.4 描繪在觀察時間區段的  $\epsilon$  曲線。此圖顯示  $\epsilon$  在大部分的例子都會低於 20%。當交通堵塞發生時 (在圖 5.4(b) 中，19:00 到 20:00 的  $\circ$  曲線)， $\epsilon$  會大於 20%。理由是當交通堵塞發生時，車輛的細胞停留時間通常會比較長，並且方程式 (3.25) 已經暗示 LCH 方法在較長的細胞停留時間的情況下，有較低的準確性。

我們也從行動電信網路收集了 49 天 (2011 年 9 月 13 日到 2011 年 10 月 31 日)，每天 8:00 到 20:00 期間的  $\gamma(\tau)$  和  $\rho(\tau)$  統計資料。基於這些觀察資料，我們的研究顯示，假如不使用第四章提出的技術， $E[\epsilon]$  為 14.46%，假如使用過濾技術 1 和 2， $E[\epsilon]$  會降低到 12.7%，假如過濾技術和補償技術皆使用， $E[\epsilon]$  會更進一步降低到 7.51%。

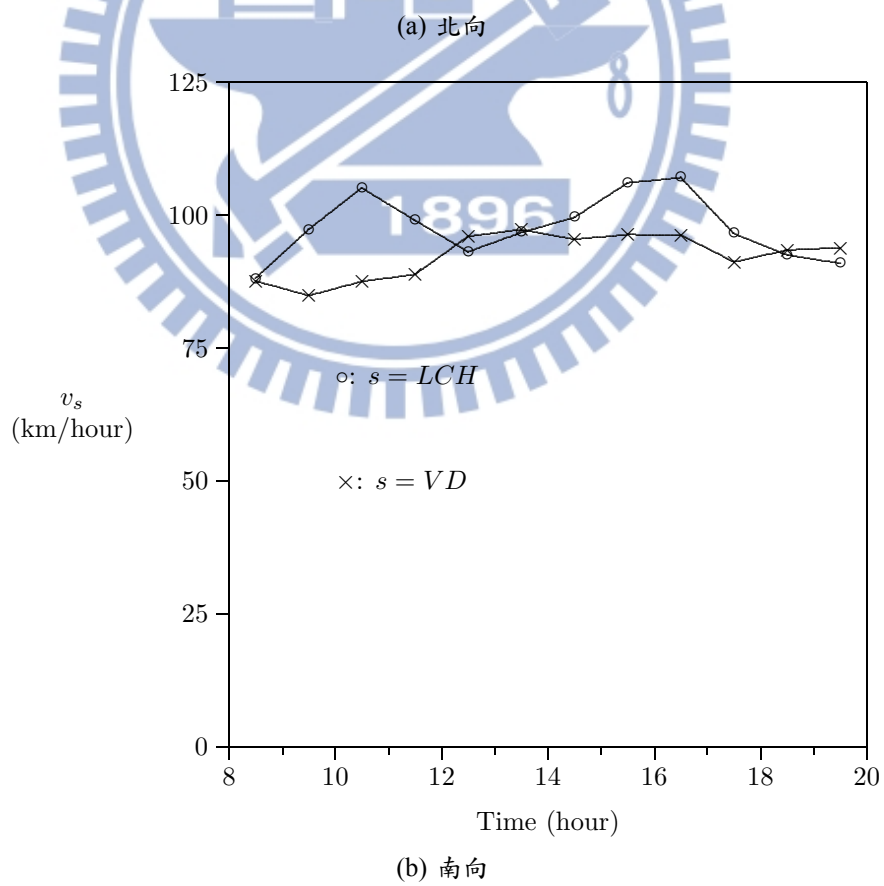
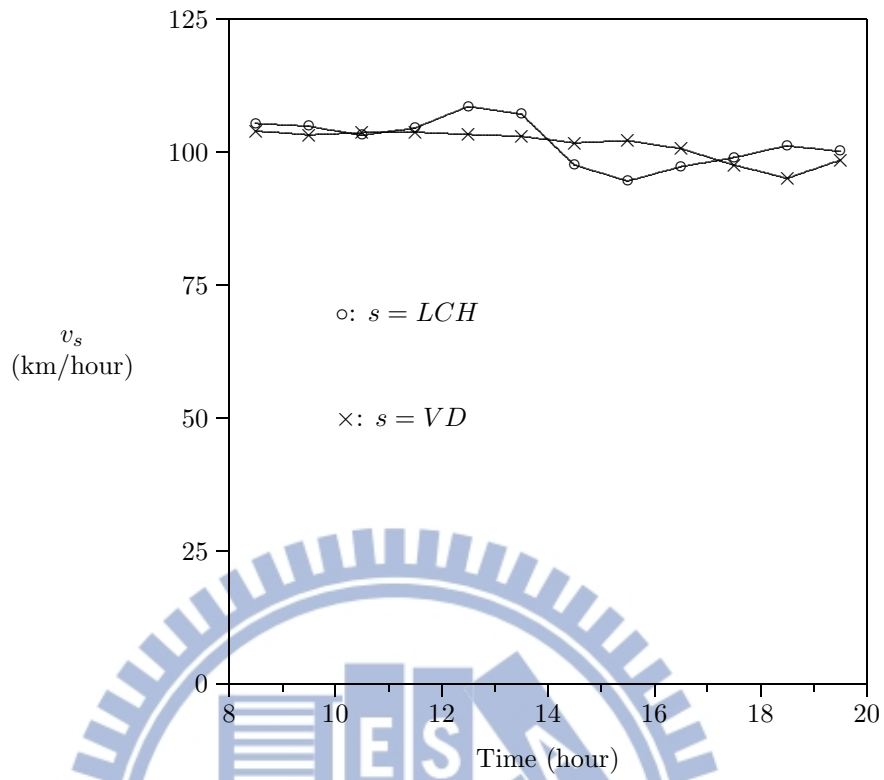
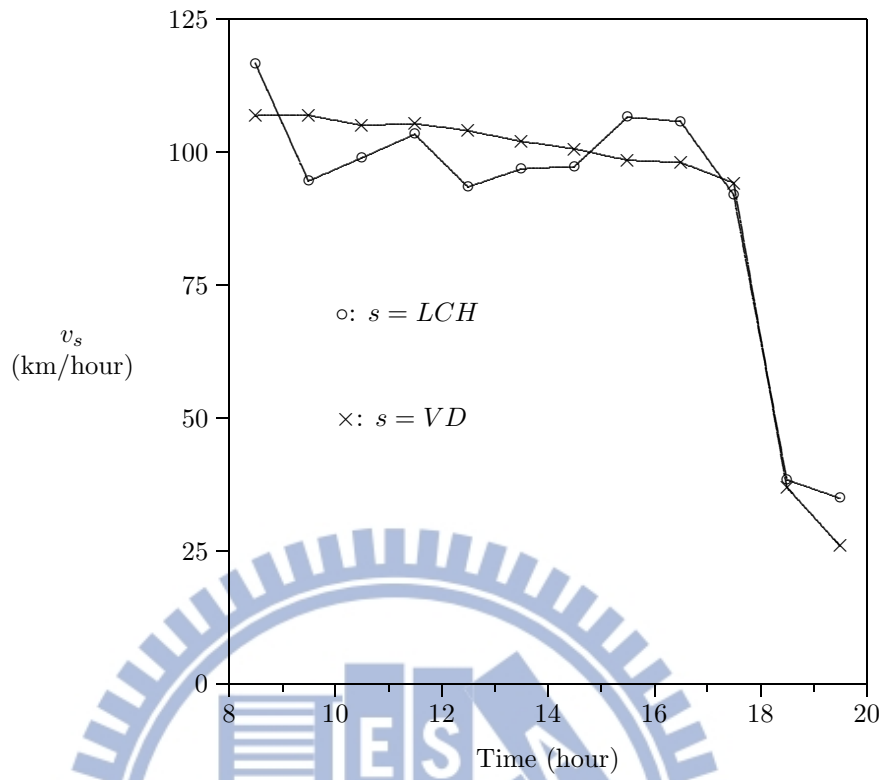
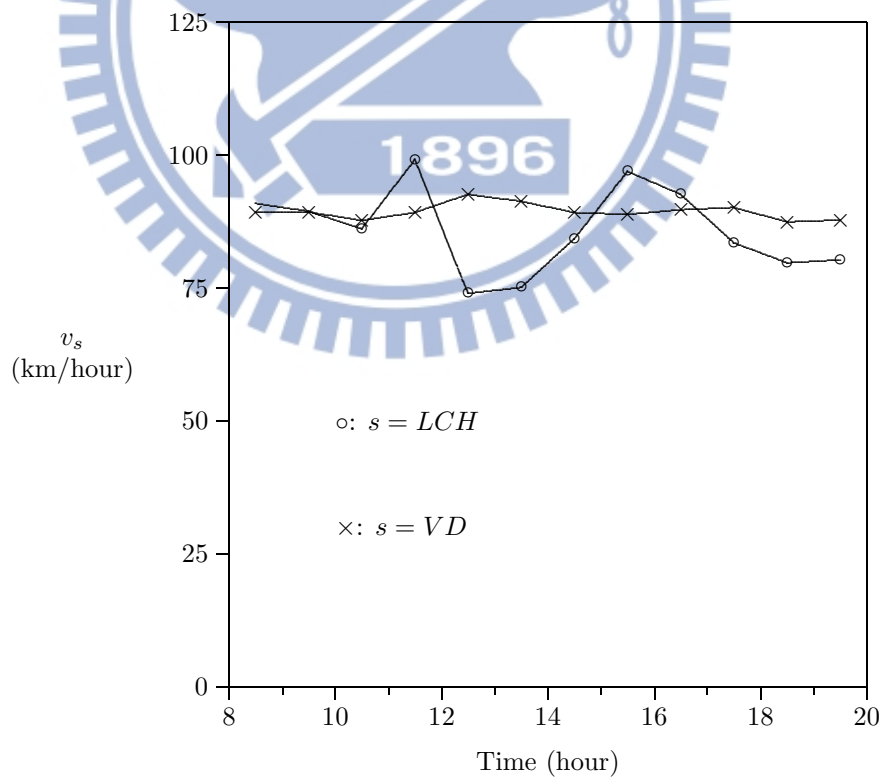


圖 5.2: 2011 年 9 月 16 日 VD 和 LCH 方法的車速比較 ( $\Delta t = 1$  hour,  $x = 1$  km)





(a) 北向



(b) 南向

圖 5.3: 2011 年 9 月 25 日 VD 和 LCH 方法的車速比較 ( $\Delta t = 1$  hour,  $x = 1$  km)

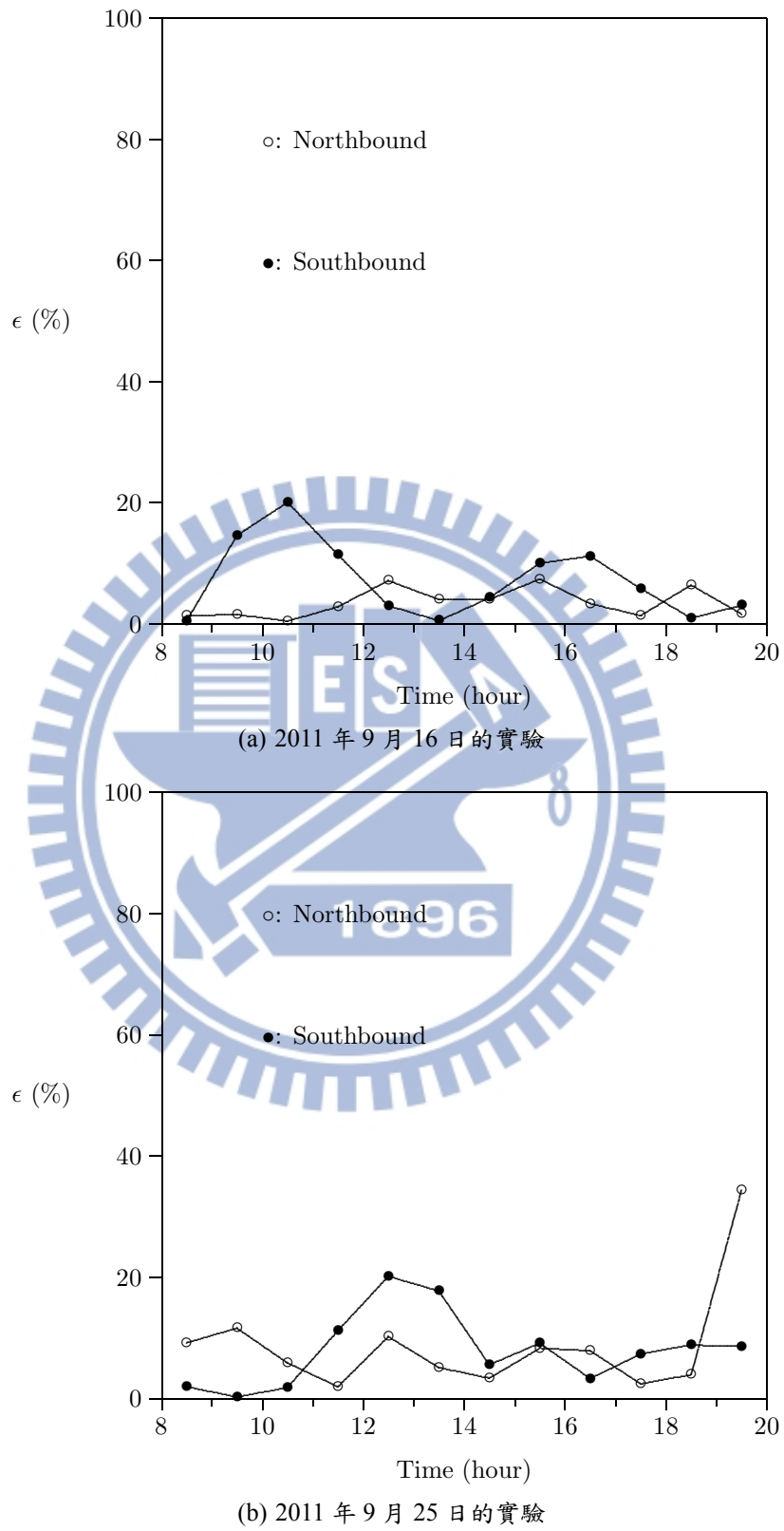


圖 5.4: LCH 和 VD 方法之間的差異 ( $\Delta t = 1$  hour,  $x = 1$  km)

## 第六章 結論與未來工作

本論文延伸林、張及皇甫曾提出的 LCH 方法 [7, 8] 來推導車速。偏差的分析結果顯示出 LCH 方法對於車速估計是一個合適的方法 (和行人的速度估計相比)。我們更進一步利用幾個技術來改善 LCH 方法的準確性，並且把由改善後的 LCH 方法計算之車速與政府安裝的 VD 量測到之車速做比較。比較的結果顯示，由 LCH 方法計算的車速和 VD 方法量測的車速是一致的，並且在大部分例子下，兩種方法之間的差異 (discrepancies) 會低於 20%。使用過濾技術 1 和 2 後，平均差異  $E[\epsilon]$  會由 14.46% 降低到 12.7%，假如再加上補償技術， $E[\epsilon]$  會更進一步降低到 7.51%。明顯地，使用這四個技術可以有效地排除不在目標道路上的手機，並且解決樣本數不夠和不平滑影響等問題，大幅的提升車速預估的準確性。我們的研究指出 LCH 方法可以適當的偵測道路的車速，並且可以避免已存在方法所需的昂貴部署花費 (例如 VD 方法的感測器部署)。最後，車速估計提出許多研究議題。在未來，我們會調查在分散式 ITS 網路的車速預測 [19]，並且發展更多的技術來提升車速預估的準確性。

## 參考文獻

- [1] Sang Jin Park, Tae Yong Kim, Sung Min Kang, and Kyung Heon Koo , "A novel signal processing technique for vehicle detection radar," in *2003 IEEE MTT-S International Microwave Symposium Digest*, 2003, pp. 8-13.
- [2] Anurak Poolsawat, Wasan Pattara-Atikom, and Boonchai Ngamwongwattana, "Acquiring road traffic information through mobile phones," in *8th International Conference on ITS Telecommunications (ITST)*, 2008, pp. 170-174.
- [3] Ruey Long Cheu, Chi Xie, and Der-Horng Lee, "Probe vehicle population and sample size for arterial speed estimation," *Computer-Aided Civil and Infrastructure Engineering*, vol. 17, no. 1, pp. 53–60, Jan. 2002.
- [4] Chih-Yi Chiang, Ju-Yin Chuang, Jian-Kai Chen, Chia-Chen Hung, Wei-Hui Chen, and Kuen-Rong Lo, "Estimating instant traffic information by identifying handover patterns of UMTS signals," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2011, pp. 390-395.
- [5] Kuen-Rong Lo, Chih-Yi Chiang, Ju-Yin Chuang, Jian-Kai Chen, Chia-Chen Hung, and Wei-Hui Chen , "Feasibility analysis of UMTS handover logs for traffic state estimation," in *2011 11th International Conference on ITS Telecommunications (ITST)*, 2011, pp. 684-690.
- [6] Bon-Yeh Lin, Chi-Hua Chen, and Chi-Chun Lo, "A traffic information estimation model using periodic location update events from cellular network", *Communications in Computer and Information Science*, 2011, vol. 135, pp. 72-77.
- [7] Yi-Bing Lin, Ming-Feng Chang, and Chien-Chun Huang-Fu , "Derivation of cell residence times from the counters of mobile telecommunications switches," *IEEE Transactions on*

*Wireless Communications*, 2011, vol. 10, no. 12, pp. 4048-4051.

- [8] Yi-Bing Lin, Chien-Chun Huang-Fu, and Nabil Alrajeh, "Predicting human movement based on telecom's handoff in mobile networks," *IEEE Transactions on Mobile Computing*, in press.
- [9] Chien-Chun Huang-Fu and Yi-Bing Lin, "Deriving Vehicle Speeds From Standard Statistics of Mobile Telecom Switches," *IEEE Transactions on Vehicular Technology*, 2012, vol. 61, no. 7, pp. 3337-3341.
- [10] Ren-Huang Liou, Yi-Bing Lin, Yu-Long Chang, and Ming-Feng Chang, "Deriving the vehicle speeds from mobile telecommunications network," *the 12th International Conference on ITS Telecommunications (ITST)*, November 2012.
- [11] Yi-Bing Lin and Ai-Chun Pang, *Wireless and Mobile All-IP Networks*. John Wiley & Sons, Inc., 2005.
- [12] Edward D. Lazowska, John Zahorjan, G. Scott Graham, and Kenneth C. Sevcik, *Quantitative System Performance Computer System Analysis Using Queueing Network Models*. Prentice-Hall, Inc., 1984.
- [13] Jay L. Devore, *Probability and Statistics for Engineering and the Sciences*. Duxbury Press, 2011.
- [14] Leonard Kleinrock, *Queueing Systems, Vol. I, Theory*, Wiley, New York, 1976.
- [15] Yi-Bing Lin, "Performance modeling for mobile telephone networks", *IEEE Network*, vol. 11, pp. 63–68, Nov./Dec. 1997.
- [16] Yi-Bing Lin and Imrich Chlamtac, *Wireless and Mobile Network Architectures*. John Wiley & Sons, Inc., 2000.
- [17] David Gundlegard and Johan M. Karlsson, "Handover Location Accuracy for Travel Time Estimation in GSM and UMTS," *IET Intelligent Transport Systems*, 2009, vol. 3, no. 1, pp. 87-94.
- [18] Hillel Bar-Gera, "Evaluation of a Cellular Phone-Based System for Measurements of Traffic Speeds and Travel Times: A Case Study from Israel," *Transpiration. Research Part C:*

*Emerging Technologies*, 2007, vol. 15, no. 6, pp. 380-391.

- [19] Chai-Hien Gan and Yi-Bing Lin, "Push-to-talk service for intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, 2007, vol. 8, no. 3, pp. 391-399.



## 附錄 A 三種通話個數公式的模擬流程圖

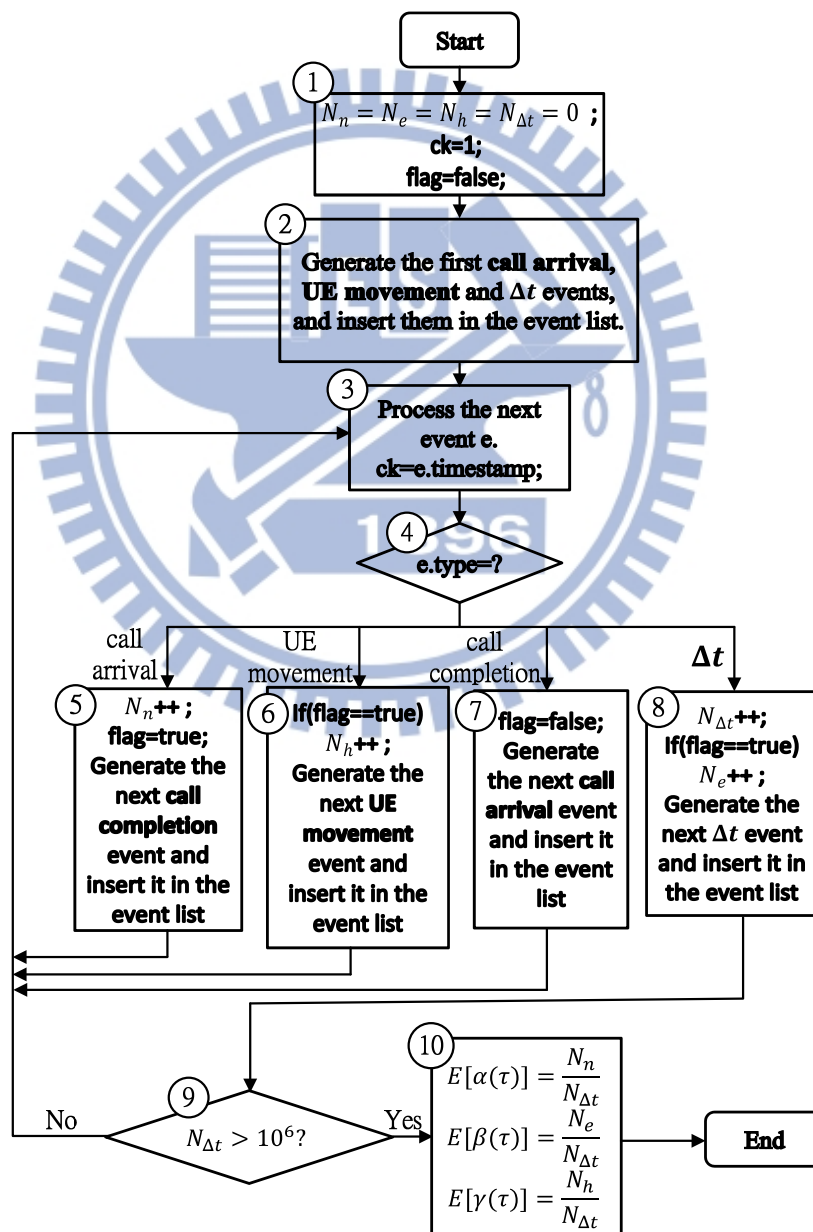


圖 A.1: 三種通話個數公式的流程圖



圖 A.1是在描述驗證三種通話個數公式的流程圖，也就是方程式 3.2、3.3和 3.7。步驟 1 初始化新通話、已存在通話、交遞通話和時間區間的計數器  $N_n$ 、 $N_e$ 、 $N_h$  和  $N_{\Delta t}$  為 0，以及時間郵戳 (timestamp)  $ck$  和通話狀態標誌  $flag$  分別為 1 和  $false$ 。步驟 2 產生第一個到達的通話、手機移動和第一個時間區段這三個事件，並且放進事件串列 (event list) 中，此事件串列會依照事件的時間做排序，越晚發生的事件會放在越後面。步驟 3 會取出事件串列中的第一個事件，並且讓時間郵戳改為此事件的時間。步驟 4 會檢查此事件的類別，如果是通話到達 (call arrival) 的事件會執行步驟 5，如果是手機移動 (UE movement) 的事件會執行步驟 6，如果是通話結束 (call completion) 事件會執行步驟 7，如果是時間區段 ( $\Delta t$ ) 的事件則會去執行步驟 8。

步驟 5 會將新通話計數器  $N_n$  加 1，把通話狀態  $flag$  設為  $true$ ，並且產生通話結束的事件放入事件串列中，接著執行步驟 3。步驟 6 會先檢查通話狀態  $flag$  是否為  $true$ ，如果是則將交遞通話計數器  $N_h$  加 1，不是就不執行。並且產生下一個手機移動的事件放入事件串列中，接著執行步驟 3。步驟 7 會將通話狀態  $flag$  設為  $false$ ，並且產生下一個通話到達的事件放入事件串列中，接著執行步驟 3。步驟 8 會將時間區段計數器  $N_{\Delta t}$  加 1，並且檢查通話狀態  $flag$  是否為  $true$ ，如果是則將已存在通話計數器  $N_e$  加 1，不是就不執行。接著產生下一個時間區段事件放入事件串列中，並且執行步驟 9。

步驟 9 會檢查時間區段計數器  $N_{\Delta t}$  是否超過  $10^6$ ，如果是就執行步驟 10，不是就會執行步驟 3。步驟 10 會將新通話、已存在通話和交遞通話的計數器  $N_n$ 、 $N_e$  和  $N_h$  都除以時間區段計數器  $N_{\Delta t}$  來算出  $\alpha(\tau)$ 、 $\beta(\tau)$  和  $\gamma(\tau)$ 。並將結果與公式算出的值做比較，接著程式結束。



## 附錄 B $E[\rho_n(\tau)]$ 之推導

本附錄根據論文 [15] 推導在  $\Delta\tau$  中新通話的期望通話時間  $E[\rho_n(\tau)]$ 。假設剩餘細胞停留時間  $\tau_m$  有密度函數  $r_m(t)$ 、分佈函數  $R_m(t)$ ，以及拉普拉斯轉換  $r_m^*(s)$ 。從剩餘時間理論 [14] 可知， $r_m(t)$  可以由細胞停留時間  $t_m$  的分佈函式  $F_m(t)$  表示如下

$$r_m(t) = \eta[1 - F_m(t)] \quad (\text{B.1})$$

經由方程式 (B.1) 和  $t_m$  的拉普拉斯轉換  $f_m^*(s)$ ， $r_m^*(s)$  的推導如下

$$\begin{aligned} r_m^*(s) &= \int_{t=0}^{\infty} \eta[1 - F_m(t)]e^{-st} dt \\ &= \left(\frac{\eta}{s}\right) [1 - f_m^*(s)] \end{aligned} \quad (\text{B.2})$$

分別用  $F_{\rho_n(\tau)}(t)$  和  $F_{x_n}(t)$  表示  $\rho_n(\tau)$  和  $x_n = \min(t_c, \Delta t_n)$  的分佈函式。由於  $\rho_n(\tau) = \min(x_n, \tau_m)$ ，我們可知

$$F_{\rho_n(\tau)}(t) = F_{x_n}(t) + R_m(t)[1 - F_{x_n}(t)] \quad (\text{B.3})$$

因為  $\rho_n(\tau)$  是一個非負的隨機變數，我們可知

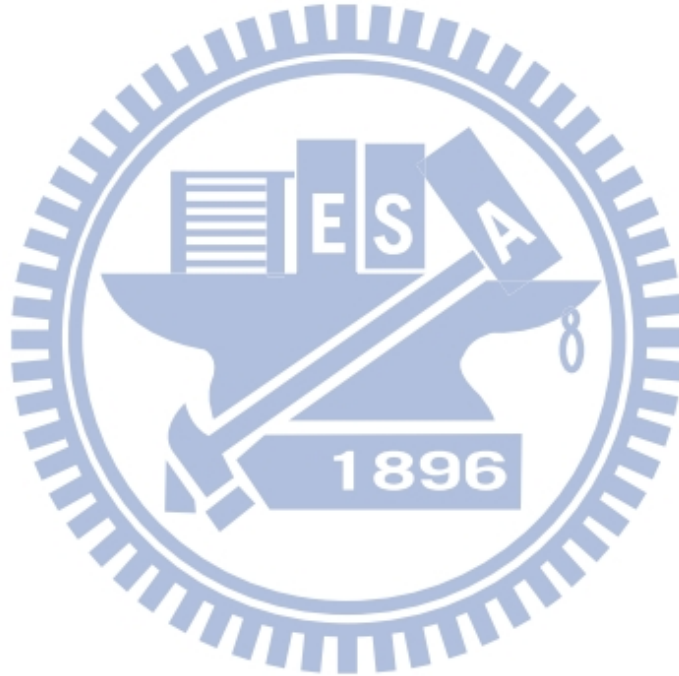
$$E[\rho_n(\tau)] = \int_{t=0}^{\infty} [1 - F_{\rho_n(\tau)}(t)] dt \quad (\text{B.4})$$

從方程式 (B.3) 可知，方程式 (B.4) 可被改寫為

$$E[\rho_n(\tau)] = \int_{t=0}^{\infty} [1 - F_{x_n}(t)] dt - \int_{t=0}^{\infty} R_m(t)[1 - F_{x_n}(t)] dt \quad (\text{B.5})$$

因為  $x_n$  是指數分佈，其速率為  $\mu + \delta$ ，從方程式 (B.2) 可知，方程式 (B.5) 可被改寫為

$$\begin{aligned} E[\rho_n(\tau)] &= E[x_n] - \int_{t=0}^{\infty} R_m(t) e^{-(\mu+\delta)t} dt \\ &= \frac{1}{\mu + \delta} - \frac{r_m^*(\mu + \delta)}{\mu + \delta} \\ &= \frac{1}{\mu + \delta} - \left[ \frac{\eta}{(\mu + \delta)^2} \right] [1 - f_m^*(\mu + \delta)] \end{aligned} \quad (\text{B.6})$$



## 附錄 C 程式碼

實作過濾方法 1 和過濾方法 2 的程式碼

```
1 ifstream in1(RNC321.c_str()) ;
2 ifstream in2(RNC311.c_str()) ;
3 ifstream in3(RNC353.c_str()) ;
4 string tmp ;
5 int IMSI ;
6 string Time ;
7
8 while(in1 >> IMSI >> Time)
9 {
10     string type = "" ;
11     string subType = "" ;
12     int cellId ;
13     int sessionId ;
14
15     in1 >> tmp ;
16     type += tmp ;
17     while(in1 >> tmp)
18     {
19         if(tmp[0]>=65 && tmp[0]<=90)
20             break ;
21         type += " " ;
22         type += tmp ;
23     }
24     subType += tmp ;
25     while(in1 >> tmp )
```

```

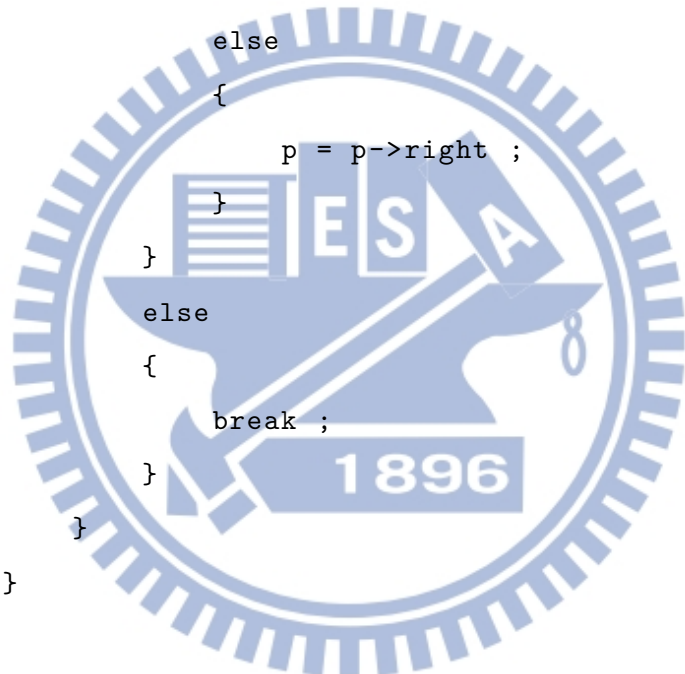
26     {
27         if(tmp[0]>=48 && tmp[0]<=57)
28             break ;
29         subType += " " ;
30         subType += tmp ;
31     }
32     in1 >> cellId >> sessionId >> tmp >> tmp >> tmp >> tmp;
33     hour = atoi(Time.substr(0,2).c_str()) ;
34     minute = atoi(Time.substr(3,2).c_str()) ;
35     second = atoi(Time.substr(6,2).c_str()) ;
36     int nowTime = hour*3600+minute*60+second ;
37
38     if(subType == "Normal loc updating")
39     {
40         if(cellId%10000 == 3483 || cellId%10000 == 3488 || cellId
41            %10000 == 9929) // North
42         {
43             if(Nhead == NULL)
44             {
45                 Nhead = new treenode(IMSI,nowTime) ;
46             }
47             else
48             {
49                 treenode *p = Nhead ;
50                 while(p != NULL)
51                 {
52                     if(p->IMSI > IMSI)
53                     {
54                         if(p->left == NULL)
55                         {
56                             p->left = new treenode(IMSI,nowTime) ;
57                             break ;
58                         }
59                     }
60                     else

```

```

59         {
60             p = p->left ;
61         }
62     }
63     else if(p->IMSI < IMSI)
64     {
65         if(p->right == NULL)
66         {
67             p->right = new treenode(IMSI,nowTime) ;
68             break ;
69         }
70     else
71     {
72         p = p->right ;
73     }
74 }
75 else
76 {
77     break ;
78 }
79 }
80 }
81 }
82 else if(cellId%10000 == 3417 || cellId%10000 == 8494 ||
83     cellId%10000 == 3286) // South
84 {
85     if(Shead == NULL)
86     {
87         Shead = new treenode(IMSI,nowTime) ;
88     }
89     else
90     {
91         treenode *p = Shead ;
92         while(p != NULL)

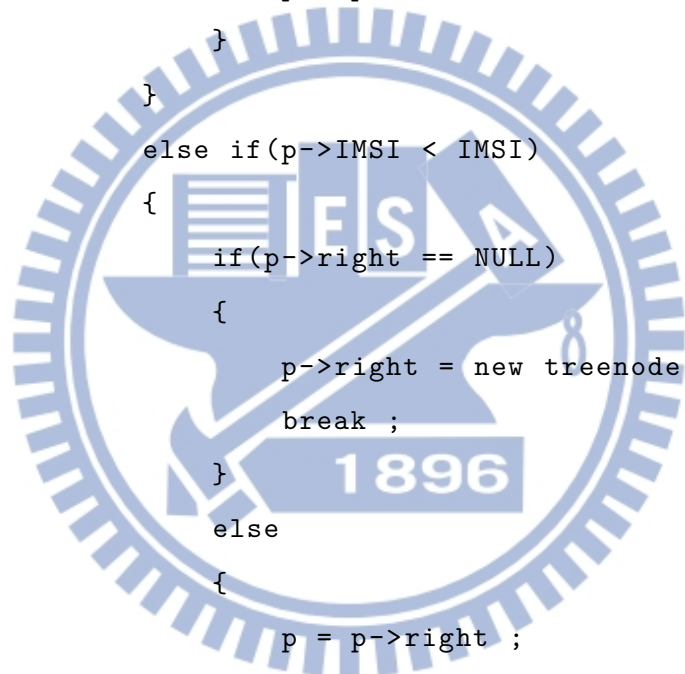
```



```

92     {
93         if(p->IMSI > IMSI)
94         {
95             if(p->left == NULL)
96             {
97                 p->left = new treenode(IMSI,nowTime) ;
98                 break ;
99             }
100            else
101            {
102                p = p->left ;
103            }
104        }
105        else if(p->IMSI < IMSI)
106        {
107            if(p->right == NULL)
108            {
109                p->right = new treenode(IMSI,nowTime) ;
110                break ;
111            }
112            else
113            {
114                p = p->right ;
115            }
116        }
117        else
118        {
119            break ;
120        }
121    }
122 }
123 }
124 }
125 if(type == "Voice call")

```





```

126 {
127     int x = -1 ;
128     if(cellId%10000 == pattern1)
129     {
130         x = 0 ;
131     }
132     else if(cellId%10000 == pattern2)
133     {
134         x = 1 ;
135     }
136     else if(cellId%10000 == pattern3)
137     {
138         x = 2 ;
139     }
140     else if(cellId%10000 == pattern4)
141     {
142         x = 3 ;
143     }
144     else if(cellId%10000 == pattern5)
145     {
146         x = 4 ;
147     }
148     if(x>=0)
149     {
150         if(extrahead == NULL)
151         {
152             extrahead = new treenode(IMSI,nowTime) ;
153             extrahead->sessionId = sessionId ;
154             extrahead->check[x] = 1 ;
155             extrahead->count++ ;
156             extrahead->cell = x ;
157         }
158         else
159         {

```

```

160     treenode *p = extrahead ;
161     while(p != NULL)
162     {
163         if(p->IMSI > IMSI)
164         {
165             if(p->left == NULL)
166             {
167                 p->left = new treenode(IMSI,nowTime) ;
168                 p->left->sessionId = sessionId ;
169                 p->left->check[x] = 1 ;
170                 p->left->count++ ;
171                 p->left->cell = x ;
172                 break ;
173             }
174             else
175             {
176                 p = p->left ;
177             }
178         }
179         else if(p->IMSI < IMSI)
180         {
181             if(p->right == NULL)
182             {
183                 p->right = new treenode(IMSI,nowTime) ;
184                 p->right->sessionId = sessionId ;
185                 p->right->check[x] = 1 ;
186                 p->right->count++ ;
187                 p->right->cell = x ;
188                 break ;
189             }
190             else
191             {
192                 p = p->right ;
193             }

```

```

194     }
195     else
196     {
197         p->count++ ;
198         if(p->sessionId == sessionId)
199         {
200             p->check[x] = 1 ;
201             if(x-p->cell == 0 && p->count<=3)
202             {
203                 p->count-- ;
204             }
205             else if(p->count == 2)
206             {
207                 if(x-p->cell>0)
208                 {
209                     p->dir=2 ;
210                 }
211                 else if(x-p->cell<0)
212                 {
213                     p->dir=1 ;
214                 }
215                 p->cell = x ;
216             }
217             else if(p->count == 3)
218             {
219                 if((p->dir == 1 && x>p->cell) || (p->
220                     dir == 2 && x<p->cell))
221                 {
222                     p->check[x] = 0 ;
223                     p->count-- ;
224                 }
225             }
226             else

```

```

227         {
228             for(int i=0;i<patternNum;i++)
229                 p->check[i] = 0 ;
230             p->sessionId = sessionId ;
231             p->check[x] = 1 ;
232             p->count = 1 ;
233             p->cell = x ;
234         }
235         int sum = 0 ;
236         for(int i=0;i<patternNum;i++)
237             sum+=p->check[i] ;
238         if(sum>=3)
239             p->flag = true ;
240             break ;
241         }
242     }
243 }
244 }
245 }
246 }
247 while(in2 >> IMSI >> Time)
248 {
249     string type = "" ;
250     string subType = "" ;
251     int cellId ;
252
253     in2 >> tmp ;
254     type += tmp ;
255     while(in2 >> tmp)
256     {
257         if(tmp[0]>=65 && tmp[0]<=90)
258             break ;
259         type += " " ;
260         type += tmp ;

```

```

261     }
262     subType += tmp ;
263     while(in2 >> tmp )
264     {
265         if(tmp[0]>=48 && tmp[0]<=57)
266             break ;
267         subType += " " ;
268         subType += tmp ;
269     }
270     in2 >> cellId >> tmp >> tmp >> tmp >> tmp >> tmp ;
271     hour = atoi(Time.substr(0,2).c_str()) ;
272     minute = atoi(Time.substr(3,2).c_str()) ;
273     second = atoi(Time.substr(6,2).c_str()) ;
274     int nowTime = hour*3600+minute*60+second ;
275     if(subType == "Normal loc updating")
276     {
277         if(cellId%10000 == 8440 || cellId%10000 == 9357 || cellId
           %10000 == 8422 || cellId%10000 ==7175 || cellId%10000
           ==3236)
278         {
279             treenode *p = Nhead ;
280             while(p != NULL)
281             {
282                 if(p->IMSI > IMSI)
283                 {
284                     p = p->left ;
285                 }
286                 else if(p->IMSI < IMSI)
287                 {
288                     p = p->right ;
289                 }
290                 else
291                 {
292                     if(p->time < nowTime && nowTime-p->time<=3600)

```

```

293         {
294             Timeslot tmp ;
295             tmp.begin = p->time ;
296             tmp.end = nowTime ;
297             p->T.push_back(tmp) ;
298             p->flag = true ;
299         }
300         break ;
301     }
302 }
303 }
304 }
305 }
306 while(in3 >> IMSI >> Time)
307 {
308     string type = "" ;
309     string subType = "" ;
310     int cellId ;
311
312     in3 >> tmp ;
313     type += tmp ;
314     while(in3 >> tmp)
315     {
316         if(tmp[0]>=65 && tmp[0]<=90)
317             break ;
318         type += " " ;
319         type += tmp ;
320     }
321     subType += tmp ;
322     while(in3 >> tmp )
323     {
324         if(tmp[0]>=48 && tmp[0]<=57)
325             break ;
326         subType += " " ;

```



```

327     subType += tmp ;
328 }
329 in3 >> cellId >> tmp >> tmp >> tmp >> tmp >> tmp;
330 hour = atoi(Time.substr(0,2).c_str()) ;
331 minute = atoi(Time.substr(3,2).c_str()) ;
332 second = atoi(Time.substr(6,2).c_str()) ;
333 int nowTime = hour*3600+minute*60+second ;
334 if(subType == "Normal loc updating")
335 {
336     if(cellId%10000 == 3578 || cellId%10000 == 3642)
337     {
338         treenode *p = Shead ;
339         while(p != NULL)
340         {
341             if(p->IMSI > IMSI)
342             {
343                 p = p->left ;
344             }
345             else if(p->IMSI < IMSI)
346             {
347                 p = p->right ;
348             }
349             else
350             {
351                 if(p->time < nowTime && nowTime-p->time<=3600)
352                 {
353                     Timeslot tmp ;
354                     tmp.begin = p->time ;
355                     tmp.end = nowTime ;
356                     p->T.push_back(tmp) ;
357                     p->flag = true ;
358                 }
359                 break ;
360             }

```

```

361         }
362     }
363 }
364 }
365 in1.close() ;
366 in2.close() ;
367 in3.close() ;

```

### 統計交遞次數與通話時間的程式碼

```

1 while(in >> IMSI >> Time)
2 {
3     string type = "" ;
4     string subType = "" ;
5     int RNC ;
6     int cellId ;
7     int sessionId ;
8     int finalId ;
9
10    in >> tmp ;
11    type += tmp ;
12    while(in >> tmp)
13    {
14        if(tmp[0]>=65 && tmp[0]<=90)
15            break ;
16        type += " " ;
17        type += tmp ;
18    }
19    subType += tmp ;
20    while(in >> tmp )
21    {
22        if(tmp[0]>=48 && tmp[0]<=57)
23            break ;
24        subType += " " ;
25        subType += tmp ;

```

```

26     }
27     RNC = atoi(tmp.c_str()) ;
28     in >> cellId >> sessionId >> finalId >> tmp >> tmp >> tmp;
29     hour = atoi(Time.substr(0,2).c_str()) ;
30     minute = atoi(Time.substr(3,2).c_str()) ;
31     second = atoi(Time.substr(6,2).c_str()) ;
32     int nowTime = hour*3600+minute*60+second ;
33
34     Phone *p;
35
36     if(type == "Voice call")
37     {
38         if(head == NULL)
39         {
40             if(finalId == 0)
41             {
42                 head = new Phone(nowTime) ;
43                 head->IMSI = IMSI ;
44                 head->cellId = cellId ;
45                 head->sessionId = sessionId ;
46                 head->call = true ;
47                 head->next = NULL ;
48                 dir=search(IMSI,nowTime) ;
49                 if((cellId == targetCell1 || cellId == targetCell2))
50                 {
51                     head->time = nowTime ;
52                     head->deltaT = nowTime/deltaT ;
53                     head->check = 0 ;
54                     N[nowTime/deltaT]++ ;
55                     head->through = true ;
56                     head->filter = false ;
57                 }
58             }
59         }

```

```

60     else
61     {
62         bool f = false ;//flag of IMSI exists
63         p = head ;
64         while(1==1) // check IMSI exists or not
65         {
66             if(p->IMSI == IMSI)
67             {
68                 f = true ;
69                 if(finalId == 1)
70                 {
71                     if(p->call == true)
72                     {
73                         p->call = false ;
74                         dir=search(IMSI,nowTime) ;
75                         if((cellId == targetCell1 || cellId ==
76                             targetCell2) && p->cellId == cellId )
77                         {
78                             p->through = true ;
79                             p->filter = false ;
80                             if(p->check == 1)//handover in
81                             {
82                                 if(nowTime-p->time <=
83                                     limiteErlang && nowTime-p->
84                                     time >= 10)
85                                 {
86                                     if(dir == 1)
87                                     {
88                                         Nrecord[p->time/deltaT].
89                                             handoverIn++ ;
90                                         AddErlang(p,nowTime,1) ;
91                                     }
92                                     else if(dir == 2)
93                                     {

```

```

90         Srecord[p->time/deltaT].
           handoverIn++ ;
91         AddErlang(p,nowTime,2) ;
92     }
93     N[p->time/deltaT]++ ;
94 }
95 }
96 else if(p->check == 0)
97 {
98     if(nowTime-p->time <=
           limiteErlang && nowTime-p->
99         time >= 10)
           {
100         if(dir == 1)
101         {
102             AddErlang(p,nowTime,1) ;
103         }
104         else if(dir == 2)
105         {
106             AddErlang(p,nowTime,2) ;
107         }
108     }
109 }
110 }
111 else if((cellId != targetCell1 && cellId
           != targetCell2) && (p->check == 2 || p
           ->check == 3))//session end outside
112 {
113     p->through = true ;
114     p->filter = false ;
115     if(p->check == 2 || ( p->check == 3
           && p->outtime-p->time >=10))
116     {
117         if(dir == 1)

```

```

118         {
119             Nrecord[p->outtime/deltaT].
120                 handoverOut++ ;
121             AddErlang(p,p->outtime,1) ;
122         }
123     else if(dir == 2)
124     {
125         Srecord[p->outtime/deltaT].
126             handoverOut++ ;
127         AddErlang(p,p->outtime,2) ;
128     }
129 }
130 }
131 p->sessionId = 0 ;
132 p->check = -1 ;
133 p->cellId = cellId ;
134 p->filter = false ;
135 p->through = false ;
136 p->starttime = 0 ;
137 }
138 else
139 {
140     if(p->call == false) //new call
141     {
142         p->call = true ;
143         p->sessionId = sessionId ;
144         p->starttime = nowTime ;
145         dir=search(IMSI,nowTime) ;
146         if((cellId == targetCell1 || cellId ==
147             targetCell2))
148         {
149             p->time = nowTime ;
150             p->deltaT = nowTime/deltaT ;

```



```

149         p->check = 0 ;
150         N[p->time/deltaT]++ ;
151         p->through = true ;
152         p->filter = false ;
153     }
154 }
155 else //handover
156 {
157     dir=search(IMSI,nowTime) ;
158     if((cellId == targetCell1 || cellId ==
159         targetCell2) && (p->cellId!=
160         targetCell1 && p->cellId !=
161         targetCell2))//handoverIn
162     {
163         if(p->check == 2 && nowTime-p->
164             outtime<HItime)
165         {
166             p->check = 1 ;
167         }
168         else if(p->check == 3 && nowTime-p->
169             outtime<HItime)
170         {
171             p->check = 0 ;
172         }
173         else if((p->check == 2 || p->check ==
174             3) && nowTime-p->outtime>=HItime)
175         {
176             if(p->check == 2 || ( p->check ==
177                 3 && p->outtime-p->time >=10)
178                 )
179             {
180                 if(dir == 1)
181                 {

```

```

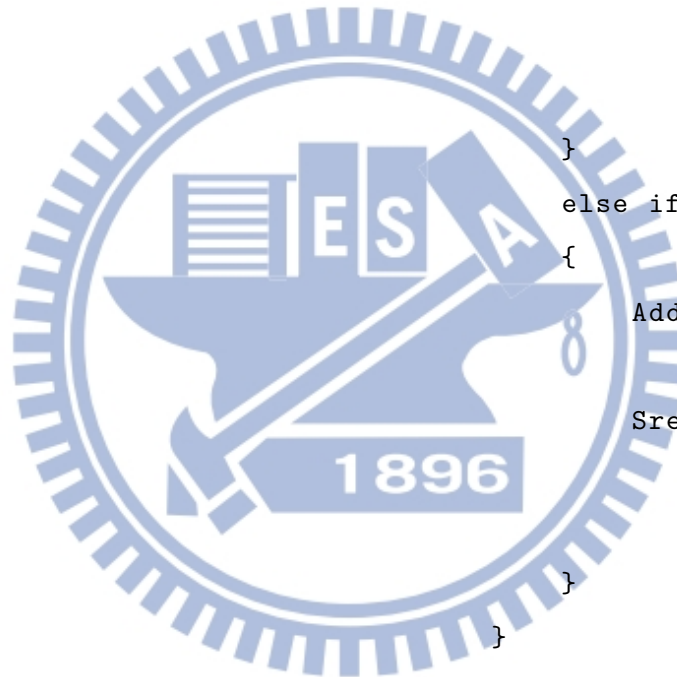
174         AddErlang(p,p->outtime,1)
           ;
175         Nrecord[p->outtime/deltaT
           ].handoverOut++ ;
176     }
177     else if(dir == 2)
178     {
179         AddErlang(p,p->outtime,2)
           ;
180         Srecord[p->outtime/deltaT
           ].handoverOut++ ;
181     }
182 }
183 p->time = nowTime ;
184 p->deltaT = nowTime/deltaT ;
185 p->outtime = 0 ;
186 p->check = 1 ;
187 p->sessionId = sessionId ;
188 }
189 else
190 {
191     p->time = nowTime ;
192     p->deltaT = nowTime/deltaT ;
193     p->check = 1 ;
194 }
195 }
196 else if((p->cellId == targetCell1 || p->
cellId == targetCell2) && (cellId !=
targetCell1 && cellId != targetCell2))
//handoverOut
197 {
198     if(cellId%10000 != p->cellId%10000)//
leave this BS set
199     {

```

```

200         if(p->check == 0)
201         {
202             if(nowTime-p->time <=
                limiteErlang && nowTime-p
                ->time >= 10)
203             {
204                 if(dir == 1)
205                 {
206                     AddErlang(p,nowTime
                        ,1) ;
207                     Nrecord[nowTime/
                        deltaT].
                        handoverOut++ ;
208                 }
209                 else if(dir == 2)
210                 {
211                     AddErlang(p,nowTime
                        ,2) ;
212                     Srecord[nowTime/
                        deltaT].
                        handoverOut++ ;
213                 }
214             }
215             p->check = -1 ;
216             p->time = 0 ;
217             p->deltaT = 0 ;
218             p->sessionId = sessionId ;
219         }
220         else if(p->check == 1 && nowTime-
                p->time >= Hitime)
221         {
222             if(nowTime-p->time <=
                limiteErlang)
223             {

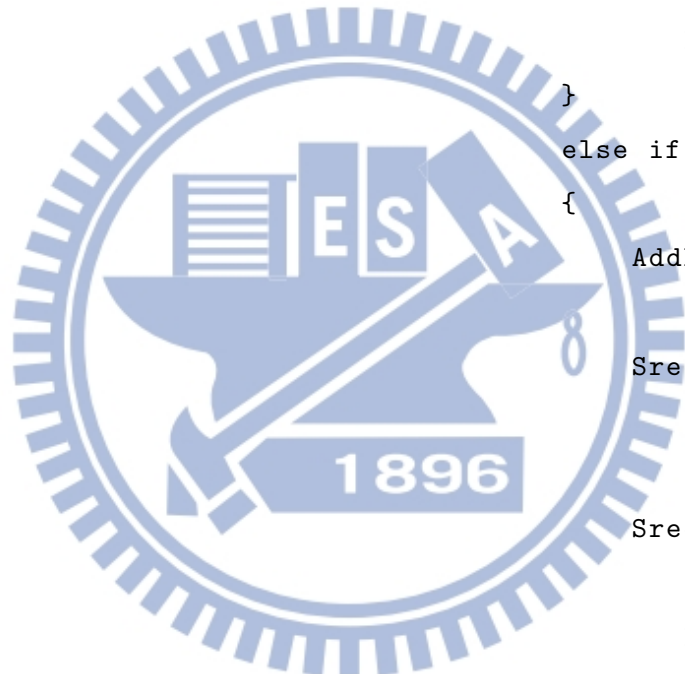
```



```

224         p->through = true ;
225         p->filter = false ;
226         if(dir == 1)
227         {
228             AddErlang(p,nowTime
229                 ,1) ;
230             Nrecord[p->time/
231                 deltaT].handoverIn
232                 ++ ;
233             Nrecord[nowTime/
234                 deltaT].
235                 handoverOut++ ;
236         }
237         else if(dir == 2)
238         {
239             AddErlang(p,nowTime
240                 ,2) ;
241             Srecord[p->time/
242                 deltaT].handoverIn
243                 ++ ;
244             Srecord[nowTime/
245                 deltaT].
246                 handoverOut++ ;
247         }
248         N[p->time/deltaT]++ ;
249     }
250     p->check = -1 ;
251     p->time = 0 ;
252     p->deltaT = 0 ;
253     p->sessionId = sessionId ;
254 }
255 }
256 else if(nowTime-p->time >= H0time)
257 {

```



```

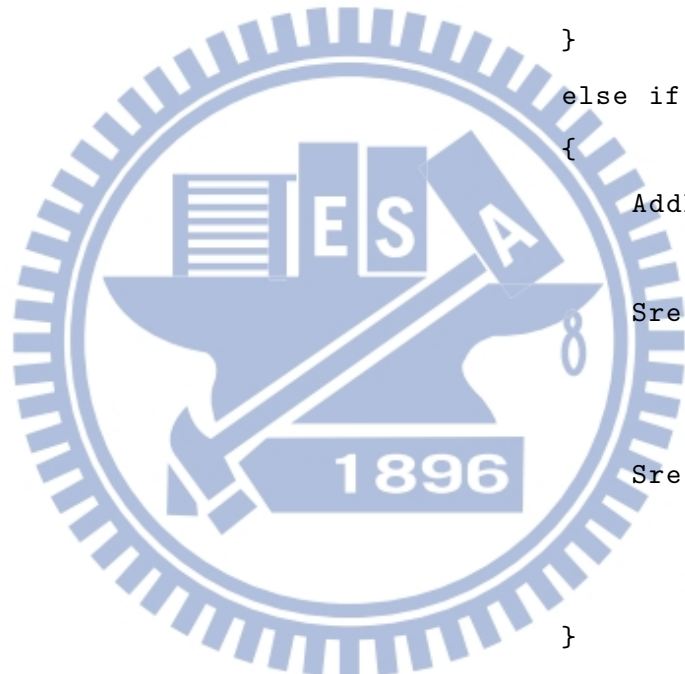
248         if(p->check == 0)
249             {
250                 if(nowTime-p->time <=
251                     limiteErlang)
252                     {
253                         if(dir == 1)
254                             {
255                                 AddErlang(p,nowTime
256                                     ,1) ;
257                                 Nrecord[nowTime/
258                                     deltaT].
259                                 handoverOut++ ;
260                             }
261                         else if(dir == 2)
262                             {
263                                 AddErlang(p,nowTime
264                                     ,2) ;
265                                 Srecord[nowTime/
266                                     deltaT].
267                                 handoverOut++ ;
268                             }
269                         }
270                     p->check = -1 ;
271                     p->time = 0 ;
272                     p->deltaT = 0 ;
273                     p->sessionId = sessionId ;
274                 }
275             else if(p->check == 1 && nowTime-
276                 p->time >= HItime)
277                 {
278                     if(nowTime-p->time <=
279                         limiteErlang)
280                         {
281                             p->through = true ;

```

```

273         p->filter = false ;
274         if(dir == 1)
275         {
276             AddErlang(p,nowTime
277                 ,1) ;
278             Nrecord[p->time/
279                 deltaT].handoverIn
280                 ++ ;
281             Nrecord[nowTime/
282                 deltaT].
283                 handoverOut++ ;
284         }
285         else if(dir == 2)
286         {
287             AddErlang(p,nowTime
288                 ,2) ;
289             Srecord[p->time/
290                 deltaT].handoverIn
291                 ++ ;
292             Srecord[nowTime/
293                 deltaT].
294                 handoverOut++ ;
295         }
296         N[p->time/deltaT]++ ;
297     }
298     p->check = -1 ;
299     p->time = 0 ;
300     p->deltaT = 0 ;
301     p->sessionId = sessionId ;
302 }
303 }
304 else if(p->check == 1 && nowTime-p->
305     time < HItime)
306 {

```





```

296         p->check = -1 ;
297         p->time = 0 ;
298         p->outtime = 0 ;
299         p->deltaT = 0 ;
300     }
301     else if(p->check == 1 && nowTime-p->
302           time >= HItime)
303     {
304         if(dir ==1)
305         {
306             Nrecord[p->time/deltaT].
307                 handoverIn++ ;
308         }
309         else if(dir == 2)
310         {
311             Srecord[p->time/deltaT].
312                 handoverIn++ ;
313             N[p->time/deltaT]++ ;
314             p->through = true ;
315             p->filter = false ;
316             p->check = 2 ;
317             p->outtime = nowTime ;
318         }
319     }
320     else if(p->check == 0)
321     {
322         p->check = 3 ;
323         p->outtime = nowTime ;
324     }
325 }
else if((p->check == 2 || p->check == 3))
{
    if(p->check == 2 || ( p->check == 3
        && p->outtime-p->time >=10))

```

```

326         {
327             if(dir == 1)
328             {
329                 AddErlang(p,p->outtime,1) ;
330                 Nrecord[p->outtime/deltaT].
331                 handoverOut++ ;
332             }
333             else if(dir == 2)
334             {
335                 AddErlang(p,p->outtime,2) ;
336                 Srecord[p->outtime/deltaT].
337                 handoverOut++ ;
338             }
339             p->check = -1 ;
340             p->time = 0 ;
341             p->deltaT = 0 ;
342             p->sessionId = sessionId ;
343         }
344         p->cellId = cellId ;
345         p->sessionId = sessionId ;
346     }
347     break ;
348 }
349 if(p->next == NULL)
350     break ;
351 p = p->next ;
352 }
353 if(f == false)
354 {
355     if(finalId == 0)
356     {
357         p->next = new Phone(nowTime) ;

```

```
358     p = p->next ;
359     p->IMSI = IMSI ;
360     p->cellId = cellId ;
361     p->sessionId = sessionId ;
362     p->call = true ;
363     p->next = NULL ;
364     dir=search(IMSI,nowTime) ;
365     if((cellId == targetCell1 || cellId ==
366         targetCell2))
367     {
368         p->time = nowTime ;
369         p->deltaT = nowTime/deltaT ;
370         p->check = 0 ;
371         N[p->time/deltaT]++ ;
372         p->through = true ;
373         p->filter = false ;
374     }
375 }
376 }
377 }
378 }
```

