

# 國立交通大學

## 網路工程研究所

### 碩士論文



連通性觸發之即時交通資訊車際網路路由

Connectivity-driven Real-time Information VANET Routing

研究生：莊敬中

指導教授：陳 健 教授

中華民國 一 百 年 六 月

連通性觸發之即時交通資訊車際網路路由

# Connectivity-driven Real-time Information VANET Routing

研究生：莊敬中

Student : Chin-Chung Chuang

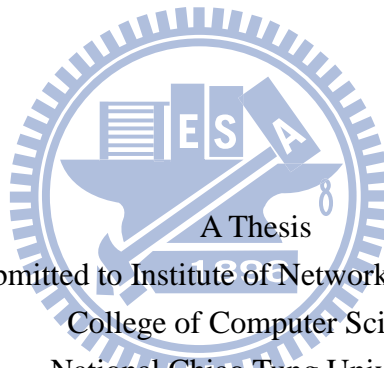
指導教授：陳 健

Advisor : Chien Chen

國立交通大學

網路工程研究所

碩士論文



Submitted to Institute of Network Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2011

Hsinchu, Taiwan, Republic of China

中華民國一〇一年六月

# 連通性觸發之即時交通資訊車際網路路由

研究生：莊敬中

指導教授：陳 健

國立交通大學網路工程研究所

## 中文摘要

車際網路是近年來新興的研究議題。利用車際網路上的車間通訊開發的各種應用，可以幫助人們生活更加便利且安全。這些應用要得以實現，都需要依靠路由機制讓資料可以在兩車輛間順利傳送。在路由機制的設計上，先前已經有很多研究出現，各種選擇最佳路徑時的考慮因素，如道路上的車子數量，道路長度等都被提出。本文提出一個利用即時交通資訊的路由協定，藉由路段上位於路口的車輛發動探測封包，經由路段上的車輛攜帶和傳送到達同一路段上的另一路口，由此測量得到各條路段上最新的封包傳送延遲時間，並且考慮車輛在道路路口轉彎時會減速，車輛數量較多的特性，在不加入額外路邊單元的情況下，設計機制將收集到的延遲時間資訊暫時儲存在路口範圍內，同時嘗試將此資訊擴散到鄰近路口並暫存。當封包經由車輛轉傳到路口，需要對下一個即將轉傳的路段作優先權計算時，便可得到此資訊，在沒有得到即時延遲資訊的路段，因為車輛上有預先紀錄的歷史交通資料(車輛數，道路速限)，車輛可以依據先前研究中的算式估算各路段的傳輸延遲時間，混合有即時延遲時間資訊路段的和經由歷史資料計算出延遲的路段，得到最短延遲時間的路徑。進一步考慮更新即時交通資訊所帶來的大量網路負擔，我們研究各個路段的連通性，先判斷路段是否為接近連通，也就是使用無線傳輸可以快速傳送資料經過此路段，中途不會因傳輸中斷而需要由車輛攜帶，這些路段對傳送延遲時間的影響較大，針對這些路段進行即時資

訊更新的動作，藉以減少網路負擔量同時增進傳輸延遲時間。連通性的考慮方法為利用車輛在行駛的特性，當車輛自由行駛時，道路上車輛的分佈呈現指數分佈。道路上每隔一段長度內有車輛的機率可以用道路上的車密度得到。於是只要在傳輸範圍內存在有一台以上車輛出現的機率極高時，便代表可以往前傳送。連續數個傳輸範圍內都有車輛可以幫忙轉傳，傳輸的總長度大於道路長度時，道路接近於完全連通。但考慮車輛在道路上的分佈並非均勻且實際每次往前轉傳時轉傳距離不等於傳輸範圍，必須針對轉傳時的轉傳距離做計算，最後以連續數個期望轉傳距離內皆有車輛的機率當作連通機率，可求出車輛密度和連通機率的關係。

透過模擬比較和其他路由協定的傳輸延遲時間，傳輸成功率和協定本身對網路造成的負擔。在正常交通情況下結果略好，但當交通情況出現和統計資料有重大差異時，本協定可以增進封包傳輸的延遲時間。

關鍵字: 車際網路、路口基礎路由、即時交通資訊、道路連通性



# Connectivity-driven Real-time Information VANET Routing

**Student: Chin-Chung Chuang**

**Advisor: Dr.Chien Chen**

## **Institute of Network Engineering**

National Chiao Tung University

### **Abstract**

VANET is a popular research topic in recent years. Along with inter-vehicle communication in VANET, a number of applications have been developed to help people's lives. To make these applications come true, we need a routing protocol to deliver packets between vehicles. In this paper, we propose a routing protocol using real-time traffic information. Vehicles on intersections are triggered to send connectivity packets to vehicles on adjacent intersections conditionally. These packets traverse road segments in two ways, forwarding and carrying, and allow us to measure the latest road delivery delay. We consider the feature of vehicles slowing down and gathering on intersections, so the number of cars is more there. We try to design a mechanism to store delay information on intersection area without Road Side Unit (RSU). Meanwhile, we will disseminate it to cars on neighbor intersections and store on them in the same way. When packets are forwarded to vehicles on one intersection, the routing protocol will calculate road priority and assign packets to vehicles on the road with higher priority as possible. In this step, if one vehicle can get real-time delay information, we can use it. Otherwise we use statistical density data to estimate road delay for roads without real-time information as in previous works. It mixes these two kinds of road delay sources to decide the next forwarding road. Furthermore, we take periodical update overhead into account. We discuss road connectivity probability and find out the relation between this probability and the number of cars on the

roads. When there's a higher connectivity probability on a road, it means packets can be forwarded through road segment by wireless transmission without carrying, thus the delay is shorter. During the traversal process of connectivity packets, we judge whether a road segment is connected. If a road is connected, then we update road delay of the road. These roads have significant impact on end-to-end delay, so we focus and do update action on them. Then we can have the benefit of reducing overhead and improving delay at the same time. To research connectivity, we assume the distribution of inter-distance of cars is exponential when drivers go on road freely, referencing traffic research. For one car, if there is another one within its wireless transmission range, it can forward packets to that car and packets can move ahead to destination intersection for the expected forwarding distance. We use average vehicle spacing to find the expected forwarding distance. From the start of one intersection, we calculate the probability that a vehicle exists within the transmission range of one car. When this situation happens many times, the total forwarding distance exceeds the road length. We say the road is connected and use this process to find out the connectivity probability. When we sense that the connectivity probability is high enough, we send the connectivity packets. Through simulations, we find out that our method works and has better performance in some cases. In traffic congestion cases, simulation result shows that our method can improve delay and reduce overhead compared with other routing protocols.

*Index Terms-VANET, Intersection-based routing, Real-time information, Road connectivity*

## 誌謝

本篇論文的完成，我要感謝這些日子以來給予我協助與勉勵的人。首先我要感謝我的指導教授 陳健博士，老師耐心且不厭其煩地指導我，在研究上和做人處事的態度上都處處提點我，即使這段期間遭遇了不少挫折，陳老師對我的指導與教誨使我在遇到困難時都能順利突破，在研究上處處碰壁時指引明路讓我得以順利完成本篇論文，在此表達最誠摯的感謝。同時也感謝我的口試委員，交通大學的陳志成教授、簡榮宏教授以及清華大學的張適宇教授，他們提出了許多寶貴的意見，讓我受益良多。

感謝和我一起努力過的學長張哲維、陳盈羽，由於他們的指點和協助，使我能突破瓶頸，研究也更為完善。另外我也要感謝實驗室的同學，黃鼎峰和孫冠宇，感謝他們給我的鞭策和激勵，讓我得以成長和得到心靈上的力量，也感謝他們陪我度過這段辛苦的研究生活，在我需要協助時總是不吝伸出援手，讓我得以突破重重難關。同時感謝我的學弟們，王柏翔、彭宣翰、鄭元碩、張大均、黃俊憲、蔡世仁等人，感謝他們和我一起在學術上的討論以及為我的研究生活增添歡笑愉快的氣氛，讓我的生活更加多采多姿，在煩躁不順遂時還可以堅持地走下去。

最後，我要感謝家人對我的關懷與支持，他們含辛茹苦的栽培，使我得以無後顧之憂的專心於研究所課業與研究，他們耐心地等待我完成學業並且陪我度過這艱難的考驗，我要向他們致上最高的感謝。

# 目錄

中文摘要.....	i
Abstract .....	iii
誌謝.....	v
目錄.....	vi
圖目錄.....	vii
第一章 簡介.....	1
第二章 相關研究.....	4
第三章 路徑協定.....	10
第一節 連通性偵測 .....	10
第二節 路段延遲更新 .....	16
第三節 混合歷史資料和即時資訊計算路徑 .....	17
第四章 模擬.....	20
第五章 結論.....	29
第六章 參考文獻.....	30





# 圖目錄

圖 一 連通道路範例.....	12
圖 二 期望傳輸距離計算方式.....	12
圖 三 各種連通機率模型結果比較(道路速限 65km/hr).....	14
圖 四 各種連通機率模型結果比較(道路速限 28km/hr).....	15
圖 五 道路連通性偵測.....	16
圖 六 道路區域儲存即時交通資訊機制.....	17
圖 七 混合即時道路延遲及歷史延遲計算最短延遲路徑.....	19
圖 八 不同連通道路比率下之傳輸成功率.....	22
圖 九 不同車輛數量下之傳輸成功率.....	23
圖 十 不同連通道路比率下之傳輸延遲.....	24
圖 十一 不同車輛數量下之協定負擔.....	25
圖 十二 不同連通道路比率下之繞路封包比率.....	26
圖 十三 車輛數量:150 不同連通道路比率下之繞路封包平均傳輸延遲.....	27
圖 十四 車輛數量:200 不同連通道路比率下之繞路封包平均傳輸延遲.....	27
圖 十五 車輛數量:250 不同連通道路比率下之繞路封包平均傳輸延遲.....	27
圖 十六 車輛數量:300 不同連通道路比率下之繞路封包平均傳輸延遲.....	28
圖 十七 車輛數量:350 不同連通道路比率下之繞路封包平均傳輸延遲.....	28



# 第一章 簡介

車際網路是近年來熱門的研究議題，藉由車輛之間的無線傳輸形成網路，資料可以在道路上快速流通，任意車輛之間可以直接傳輸或經由多重跳躍收送封包，只要車輛上有無線傳輸裝置即可。在架構上可以選擇是否要架設其他基礎設施來幫助資料的傳送。透過車輛的移動性可將資料送達其他網路架構無法送達之處，使用車際網路的好處為成本低且傳輸快速。自從相關標準(Dedicated Short-Range Communications[1]、802.11p)被提出之後，各種在車際網路上的應用紛紛被開發出來[2][3]，如加強行車安全性的車輛防撞警告，到實用性的車隊通訊，停車位告知，甚至是娛樂性的即時交通流量資訊、旅遊規劃等。這些應用大大提升了人們的生活品質。

要讓這些應用得以實現，許多研究議題紛紛被提出並討論。基本的網路構成要件，MAC 層的存取機制設計和封包如何在相鄰兩車間傳送的問題已經廣泛的被討論。除了這些要件，車輛可能需要傳送資料到數公里外的車輛或固定點，此時需要實現遠距離兩車之間多重跳躍傳送，必須要有一個可靠且可以符合傳輸延遲時間要求的協定。可以因此達成的應用如車輛行駛在道路上，可以向遠方的商店訂購物品，當車輛到達商店便可拿取所需商品。

在車際網路中的路由協定設計上會遇到一些困難。雖然車際網路是隨意網路的一種，但車際網路有其不同於傳統隨意網路的特性。首先，隨意式網路上的節點移動速度較慢，網路拓撲變化較小，車輛是以高速移動，網路拓撲變化較大，無法利用紀錄網路上所有節點間的位置建立一條長期存在的路徑，兩車輛之間的無線傳輸品質也不穩定，因為節點的位置隨時在變化且容易受到障礙物阻礙。再來因為車輛被限制在道路上，被道路間建築物區塊分隔的車輛無法進行傳輸，只有在特定區域上有車輛出現可以進行無線傳輸，隨意式網路上的節點可以分散在地圖上的任意位置，所以每個節點周圍都可能其他節點可以幫助傳送資料，車際網路如果在車輛密度不夠高的道路上，比較容易出現周圍無節點，傳輸中斷無法再往目的地轉傳的情形。

先前研究的路由協定，一開始提出的方法是直接套用隨意網路上既存的方法，如 AODV[4]，DSR[5]。這些協定是以拓撲為基礎，也就是紀錄路徑上的每一個經過的節點，路徑選擇的依據是最短延遲時間，當有封包要傳送時，依據路徑上紀錄的節點逐次往下一轉傳點傳輸，最後到達目的地。這種方法在隨意網路上有不錯的效能，但在車際網路上因為網路拓撲變化大，路徑上的節點時常且快速移動，傳送過程中紀錄的路徑會時常斷裂，此時需要啟動路徑復原機制重新尋找路徑，產生大量的控制封包負擔，傳送成功率下降，封包傳送延遲時間也會增加。而且需要在封包來源點啟動查詢封包才能在路徑上節點都留下路徑紀錄，封包才能從來源點被送出，發送查詢封包到得到路徑需要時間，造成應用層產生的封包無法即時快速的被送出，需要等待一段時間，增加封包傳輸延遲時間。另一種隨意網路上的協定改以位置為基礎，如 GPSR[6]，這類協定不需在封包從來源點送出前建立路徑，來源點知道目的地的位置和周圍鄰

居節點的位置之後，以貪婪法選擇位置最靠近目的地的鄰居節點送出，鄰居節點收到後也是作相同動作直到送到目的地。如果傳送途中某一節點附近沒有更靠近目的地的鄰居節點可以轉傳，使用右手法則繞路而行。如此在鄰居節點快速變動的情形下也可以找到路徑到達目的地，只要中間轉傳節點都可遇到更靠近目的地的節點，每個封包傳送經過的路徑與節點不必相同，這類協定的缺點是沒有考慮到道路性質，只用位置來選擇下一節點，容易傳送到車輛密度低的道路或和目的地所在道路不相連的道路，增加傳送延遲時間。

近來新出現的車際網路路由利用到新出現的技術數位地圖和車輛與道路之間的關係。首先假設車輛都裝載有行車裝置，可以裝載地圖。有了地圖資訊後，在計算最佳路徑時可以考慮道路拓樸、車輛密度、車速等交通資訊，傳送封包時盡量經過車輛密度高、車速快且實際上可以連接到目的地所在道路的那些路段，避免傳送到無法到達目的地的道路。再者因為有地圖，可以將車輛在可能行走的道路抽象化為圖形，將路口轉化成節點，路口之間的路段以有權重的邊代表，這邊的權重可以是傳輸延遲時間、車密度、連通機率，路段長度等。原本路徑是以路徑經過的節點表示，現在可以用圖形表示，以路口和路段組成，不需確切知道路徑上每一節點的身分，只要在封包到達路口時，決定要往那一個路口轉傳，決定之後，依靠該路段上的車輛轉傳，只要有車輛位於路段上且在傳輸範圍內便可幫忙轉傳到下一路口，計算上簡單快速，也不需在節點紀錄路徑。在這種架構下，各個協定在選定下一個路口的優先順序時有些許不同，像 GyTAR[7][8]僅僅是傳送到路口時，貪婪地將和現在路口直接相鄰的道路作優先權計算，並未將現在路口到目的地所在路口的中間的所有路段加以計算。

改良過後的協定如 MDDV[9]、VADD[10]、SADV[11]等，會將封包現在所在路口到目的地所在路口之間的所有道路轉化成圖，根據歷史統計資料的交通資訊(車輛密度、道路速限)計算出每條路段的權重，但並不是直接套用常見最短路徑演算法，因為每個路段不一定都存在有車輛，還要再乘上每個路口可以轉傳到下一鄰近路口的機率，利用方程式求解。這樣可以盡量將封包往車輛數較多的道路傳送，維持最佳傳送延遲時間。這些協定已經考慮到全域最佳解，但是仍有其缺點存在，在 SADV 中提到，像 VADD 協定在估計每個路段的傳輸延遲時間時，是使用統計資料(平均車輛密度)。當車輛數隨著時間變化而不同時，計算出的延遲不能反應出即時的最佳路徑。於是提出了即時交通資訊的概念，就是利用車輛收集各個路段傳輸延遲時間，在資料封包傳送時附加欄位測量路段傳輸時間，然後在計算路徑時加以利用，但其特點為使用到路邊單元，也就是要在路口多架設數個路邊單元，負責測量路段延遲時間，這樣就需要額外成本，而且並不是隨時都有資料封包經過路口，不能隨時都能得到路段延遲資訊。類似概念，在 RBVT[12]中也想要利用交通資訊，但其中提到的交通資訊是指道路是否連通，先去測量哪些路段是連通的，也就是在路段中都持續有車輛可以幫忙轉傳，中途不需由車輛攜帶後等到有車輛在傳輸範圍內才能轉傳，傳輸延遲時間極短，然後在計算路徑時直接將那些不連通的路直接排除，只考慮有連通的路，並未提到會利用時間資訊。這些使用即時資訊的協定，會需要使用額外控制封包去得到即時資訊，對網路造成額外負擔。

於是我們想要改良使用即時交通資訊的路由協定，首先我們考慮車輛在路口時的數量會比在路段上較多的特性，不使用路邊單元，設計機制讓車輛在路口自己形成暫存單位將收集到的即時交通資訊儲存在路口。同時為了解決收集道路資訊時需要傳送大量控制封包造成網路負擔過大，我們研究道路連通性和車輛密度的關係，想要利用這個關係當作門檻，利用道路上的車輛分散式的去偵測道路密度，當車輛密度到達一定值，整條路段呈現極高機率連通時才進行更新動作，只針對那些目前交通情況適合作無線傳輸的路段更新，其他交通狀況變化不大的道路不做更新，藉此減少頻繁更新的負擔。

本篇論文接下來的部份包括，第二章介紹車際網路上和本論文相關的路由協定及研究。第三章提出我們改良過後的路由協定和使用的收集道路資訊機制，道路連通性與車輛數量的關係。第四章透過模擬實驗比較我們的方法和其他先前研究的方法，在各項度量上的結果。第五章是此篇論文的結論及未來展望。



## 第二章 相關研究

現今已經有很多車際網路路由的研究被提出，根據特性不同可以分類為以下二種，以位置資訊為基礎和以拓樸資訊為基礎的路由協定。

以拓樸資訊為基礎的路由，必須在封包送出前先發出路徑要求去知道網路上來源點至目的地之間節點的連結情形，每個節點紀錄到目的地的最佳下一個轉傳點。封包送出後，依據各中繼節點紀錄找下一轉傳點，路徑是由節點組成。簡單來說就是各個節點都需要維持自己可以知道的全域路徑表，紀錄每個目的地對應到的下一轉傳點身分以及經過多少轉傳次數才能到達，這種方法被認為在車際網路上比較不可行，因為車輛會在道路上快速移動，路徑會快速地改變，要在路徑表中維持任兩點的正確連結紀錄是很困難的事。

以位置資訊為基礎的路由，節點需要知道自己的位置、鄰近單跳節點的位置，和目的地的位置，之後再經由不同的最佳化條件(傳輸延遲最短、傳輸成功率最高)。在封包送出後依照位置去選擇下一轉傳點，不必要傳給某一特定節點，只要有位置更接近目的地的點存在即可轉傳。而且因為現今 GPS 裝置的普及性快速增加，使得以位置為基礎的路由更加可行實用。

在以位置為基礎的路由協定中，近來新的研究方向著重於利用更多有關車際網路特性的資訊來幫助選擇路徑，藉以達到傳輸成功率較高和傳輸延遲較短的目的。目前新提出的方法中，有一類是加入數位地圖資訊，包括道路分佈及連接情形，道路速限，路口是否有紅綠燈，各時段的平均車速和車輛數量，利用車際網路上一些可預測的移動性來幫助資料傳送。同時在路徑的建立上，捨棄以往以車輛當作路徑紀錄的基本單位，改以路口為路徑的紀錄單位。就是每條路徑建立完成後，是紀錄中間經過的路口編號，根據此順序從起點到達目的地，在路口依據優先順序找對應道路上的車輛前往下一個路口，中間經過的車輛，是從路口與路口中間道路上的車輛中，根據當時情況選擇數台車輛繞行過這條道路即可。所以路徑不必紀錄中間要經過哪幾台車，符合車際網路車輛快速移動導致車與車之間的連結快速變化經常斷裂的情形。而且車輛本身就被限制在道路上行走，只有在路口才有可能改變行進方向，所以只要規定在路口時候決定下一步要傳送到哪一個路口即可，在道路中間傳輸不會超出道路範圍傳送到其他道路上，整條路徑要紀錄的資訊就只有路口順序，資料量大幅減少。在這類路由方法中，較知名的方法有 VADD，另外還有 GPCR[13]、GyTAR、A-STAR[14]等，在此就 VADD 做簡單介紹。

VADD 是在車際網路中利用多重跳躍的路由協定，它將道路地圖抽象化成圖，其中路口當作頂點，道路當作邊，道路會將路口連接起來。道路成為頂點之間的連結，道路的權重就是預測的傳送延遲。利用已知數位地圖上的統計資料(車輛平均速度、車輛密度)計算封包傳送過每一條路段的延遲時間。在傳送時可以使用的方法有藉由車輛之間的無線傳輸和 carry-and-forward。基本概念是當轉傳方向決定好後，有車輛在當

前車輛的轉傳方向且在傳輸範圍內時便可用無線傳輸，若無車輛可轉傳則當前車輛暫存封包直到遇到其他可以幫助傳送的車輛才轉傳，這樣在車輛密度較低的情況下，封包不會馬上被丟棄而是可以繼續等待更好的車輛幫忙轉傳，增加封包傳送至目的地的機率。每次當有車輛要傳送封包時，起點車輛先利用 Location service 等方法知道目的地車輛位置，然後將起點到目的地圍成一矩形，取出地圖上這個範圍內的道路，計算每條道路預估的延遲時間，根據車輛在各個路口的停留時間和轉彎機率，計算封包在各路口時轉傳到鄰近各路段的機率。對於每一個和起點路口相鄰的路段，計算使用此路段到達目的地路口的預計延遲時間。此延遲時間為封包在此路段所需經過的延遲加上到達鄰近路口後轉傳到各路段的機率乘上用各路段傳送到目的地的延遲。

被圍成矩形內的所有路段的到達目的地預計延遲時間列出來後，可得到方形矩陣，對於矩陣求解得到每一路段到目的地的延遲時間的解，這樣將起點到目的地的最短時間路徑計算出來，決定各相鄰路段的優先順序，每當封包在路口時便往優先權較高的路段傳送。封包傳送途中，會根據現在攜帶封包車輛是在路口範圍或是道路中間分成兩種模式：在路口範圍時就計算各相鄰路段的優先權並依路口優先順序找鄰接道路上是否有車輛可以幫忙傳送，若沒有任何車輛可以轉傳就暫存封包在記憶體。在道路中間時則使用 GPSR 找尋在同一條道路上更接近目的地的車輛轉傳，若沒有車輛可以幫忙轉傳則暫存封包在車輛上。當封包接近到終點一定範圍內，就用 greedy-forwarding 找尋地理位置上更接近目的地的車輛轉傳。這個方法有考慮道路的連通性和車輛只有在路口才會變換方向的特性，節省路徑紀錄的大小。在車輛密度中等以上且車輛數量穩定的情形下有很好的效果，但在車輛密度低且變化大的情形下效果會變差。原因之一是道路上車輛密度低時，在封包到達路口時，優先權較高的道路上都沒有車，導致最後只能利用優先權較低的道路傳送，造成傳送的路徑不是最佳。另外因為估計的道路延遲時間是根據統計的平均道路密度計算出來，但車輛密度隨時間變化，導致估計的延遲時間不準確。因為以上原因，有研究改善了 VADD 的缺失，在此介紹 SADV。

SADV 考慮到(1)當封包傳送到路口時，優先權最高的路段上此時不一定有車輛。想要利用在路口擺設靜止的節點來幫助封包傳送，當封包在路口要轉傳到下一個路口時，如果優先權最高的路段上沒有車輛可以轉傳過去，可以先暫存在靜止節點，等待到有車輛出現時再轉傳出去，這樣可以盡量保持封包在最佳路徑上傳送。(2)之前的方法中提到的延遲時間的得到是利用統計資料計算得到，無法針對車輛流量的變化調整，此方法可以利用靜止節點來做延遲時間的測量，測量的方法是當有封包從一個路口要送到另一路口時，在封包中插入時間欄位。當封包到達另一路口時，將當前時間和封包中的送出時間紀錄相減得到這條路的延遲。然後收到的路口靜止節點再將此延遲時間資訊包裝成延遲時間更新封包，用廣播方式告訴其他節點，其他節點收到後根據加權公式計算後修改自己的延遲時間紀錄。當有封包要送出時，分成兩種模式，道路和路口模式，在道路時，使用 geographic forwarding 傳送到路口的靜止節點，就是選擇地理位置上最接近路口靜止節點的車輛幫忙傳送。當封包到達路口時，利用延遲時間紀錄和 VADD 的公式計算最佳路徑，當最佳路徑上的下一個路段上有車輛可以幫忙傳送就將封包送出，沒有的話靜止節點將此封包暫存，直到最佳路徑上有車輛可以幫

忙傳送、或暫存空間滿了之後因為暫存管理策略從暫存中拿出並根據路口優先權選擇當時次佳路徑上的車輛傳送。這個方法因為有在路口放置節點幫忙，可以在道路上沒有車輛時還可以暫存，增加封包行走最佳路徑的機率，而且有近乎即時的延遲資訊更新，可以根據道路的變化計算出和統計值不同的最佳路徑，得到的路由效果會比之前的 VADD 較佳。但其缺點是必須在路口增設類似 Road Side Unit (RSU) 的裝置，成本增加，另外在延遲資訊的更新方法中，要有封包需要被傳送時才有機會更新節點的延遲資訊，更新的資訊在有資料封包送出前無法得到，傳播延遲時間更新封包是利用廣播方式沒有明確限制傳播範圍，造成一些多餘的負擔。之後的研究 RBVT-P 對於更新封包的更新方式做了不一樣的用法。

RBVT 的基本概念是想利用即時交通資訊製造出以路口為基礎的路徑，在路口中間則是可以自由選擇車輛傳送，不必紀錄路徑中每一台經過的車輛，同時考慮到路口間連通的機率。這邊提出的路由方法有兩種，RBVT-R: reactive routing protocol，需要路徑時才啟動機制，當有封包要從來源點送出時，用 flooding 方式發送路徑要求封包去找到到達目的地的路徑，為了避免造成太大的負擔，重覆起點和重覆序號的要求封包會被丟棄，中間各個點收到後，並不馬上廣播，而是等待一段和前一個節點距離成倒數的時間，如果這段時間內沒有聽到其他節點廣播，自己才廣播，確保每次傳輸都可以傳送到最遠距離的車輛和減少網路上的交通量。來源點送要求封包出去，一開始封包內路徑紀錄為空，中間收到的節點檢查是否到達和前一個節點不同的路段，如果是的話，把新多出的路口加入路徑紀錄中，直到要求封包到達目的地。目的地回傳回覆封包給來源點，紀錄來源點到目的地以路口構成的路徑，來源點收到回覆封包後開始傳送資料。另外針對車輛會移動的問題，有路徑維持的機制，來源點或目的地移動到新的路段時，原本路徑上必須經過的路口便不需要再經過，此時來源點或目的地會發送路徑更新封包到目的地或來源點，將不會經過的路口從路徑紀錄中刪除。在路徑途中遇到沒辦法再往前轉傳的情形時，有錯誤處理機制，來源點等待一段時間，在此時間內嘗試使用這條路徑，直到時間過後還是無法傳送便由來源點重新發起要求機制。這個方法和之前 MANET 上的 AODV 相似，只是將路徑改為以地圖路口為基礎的方式，較不同的是第二種路由方式 RBVT-P。

RBVT-P: 周期地去探尋道路上的路徑，然後使用看到的近乎即時的道路地圖計算起終點之間的最短距離路徑。以道路為基礎的拓撲探索是利用每隔一段周期隨機選出一些節點發送 connectivity packet(CP)。每一個節點獨立地決定是否要產生 CP，決定的依據是看現在網路上車輛數量，歷史的交通資訊，和上一次收到 CP 的時間。當車輛決定要發送 CP 時，會定義要包含的範圍，也就是要探索多大的區域然後送出，中間利用 Depth First Search(DFS)繞行整個範圍，收集道路上是否為連通的資訊，最後返回發起 CP 的路段，在路段上第一個收到返回 CP 的車輛再將 CP 的內容散發更新封包。散發的範圍是之前 CP 規定的探詢範圍，散發的內容是道路的拓撲，和道路是否為連通，收到更新封包的車輛會更新自己的路由表，並依照此路由表來計算最短距離路徑。計算時只使用那些被標記為連通的路段，計算好的結果放在封包中，封包依照此路徑選擇路徑中道路上的車輛來幫忙傳送，但是當封包經過的轉傳節點有更新的路由資

訊，就會使用更新的資訊得到的路徑來傳送。在車輛和車輛之間的轉傳，比較不同的是採用 receiver-based forwarding，就是不指定下一個轉傳點，每一次傳輸都是廣播，收到的一群車輛中，每台車輛會使用條件函式算出各自的等待時間，這邊考慮的條件有可以前進到目的地的距離，無線傳輸的最佳距離，和接收到的功率，這三項的加權，擁有最短等待時間的車輛廣播將封包往前傳，其他等待時間較長的車輛聽到廣播後便將封包丟棄表示已經有人幫忙往前傳送。RBVT-P 的好處是可以隨著交通的變化，即時收集道路資訊來計算最短路徑，但是在車輛分布較低時，CP 封包的傳送變得較困難，因為要傳送繞經整個區域，需要整個區域內有足夠車輛讓封包可以繞完並回到發起的路段，整個資訊才能收集完全並散發，而且繞行的時間可能過久讓更新的資訊的新鮮度下降。在發送 CP 的參數上，要多頻繁發送且發送範圍要多廣也無明確定義，這些參數會影響網路效能。頻繁的發送 CP 封包可能會導致網路上的負擔過重，反而導致真正的資料無法送達，而且在交通狀況沒有多大變異性的情形，如果用已知統計值就可以得知道路上的情況而且道路情況也和事實近似的話，其實不用頻繁的送封包去探詢，真正有需要時再探詢即可。

基於以上論文中所產生的問題，我們尋找以下幾篇論文做參考，期望可以解決其中幾個重要問題，並發展出較符合實際情形的效能提高的路由協定。在[15]這篇文章中探討車際網路上儲存局部資訊之能力並分析不同交通狀況下的結果。想法是在彼此在互相傳輸範圍內的車輛形成一個小型的 Mesh network，就是 VMesh，其中每一台車輛都至少和 mesh 當中的任一車輛可以透過單跳或多重轉傳互相通訊，在這個 VMesh 當中的每一台車輛都擁有相同的資訊，只要有一台得到特定資訊，便可以用廣播或多次轉傳送給在同一 mesh 的其他車輛。實際運作流程如下，道路上隨時隨地可能有特定情形發生如車禍，當事件發生時，有一定的機率剛好有一個在 mesh 內的車輛經過並收到事件發生的訊息，事件發生的訊息就可以被保留在事件發生的附近範圍內，藉由 mesh 之間的互相傳送，就是當任一 mesh 的任一車輛收到事件發生的資訊後，它會傳送給同一 mesh 中的車輛並暫存此訊息，如果有新成員加入此 mesh，便分享給它。當遇到其他 mesh 時，將資訊送給其他 VMesh。每個在 mesh 中的成員都進行此分享的動作。在快要離開這個範圍時，車輛會尋找有無即將進入範圍內的車輛並將資訊傳送出去，在離開後將資訊刪除。只要一直有車輛進入事件發生的周遭範圍內，資訊便可以被保存在這個範圍，類似暫時的儲存裝置，讓進入這個範圍內的車輛都可以得到這個資訊，可以用在安全或其他用途。給定車輛速度、交通密度、和儲存在事件發生周圍多大的範圍，推算出訊息可以儲存住的時間，和事件發生後會被任一 mesh 收到的機率( $P_c$ )。經過模擬後， $P_c$  和推導出的公式大致符合，可儲存時間依據參數不同從十秒到數十秒。歸納出影響儲存時間的參數有無線傳輸範圍和保存範圍的大小。這篇文章提出利用車際網路將某些資訊保留在特定區域的方法，並分析實際效能，這個特點可以拿來幫助車際網路上的路由方法。

文章[16]中，Link-state 是介於 proactive routing(隨時任兩個節點互傳資訊更新路徑表)和 reactive routing(需要傳送資料時才發送路徑要求封包)的方法，就是當節點發現自己和周圍節點的連結發生變化，連結斷掉或新產生連結時，傳送更新訊息給周圍幾



個跳躍內的節點讓節點可以知道最新拓樸，進而選擇路徑。本篇文章分析了在無線隨意網路上使用 link-update 的路由方法，藉由限制更新的範圍和頻率減少造成網路的負擔，使其可以適用於範圍較大的網路，並詳細地定義路由方法造成的各種負擔，在最佳化負擔的情形下找出適當的更新跳躍數和頻率，並非傳送給每個節點，讓所有點都更新成最新資訊，而是讓整個網路上的各個節點在所知拓樸不同的情形下選擇路徑。首先每個點每隔一段區間才送 link-update 到周圍一定跳躍次數內的節點，累積一段時間內的所有連結變化，減少了更新時所需的負擔，但還是有些許控制封包的產生。而且因為並不是所有點都在知道最新拓樸的情形下選擇路徑，會導致選擇的路徑不是最佳，造成資料封包在網路上必需經過較多次傳送才能到達目的地，於是就多產生了一些負擔。定義傳送更新訊息的封包和選擇到次佳路徑造成的負擔，兩者相加為總負擔，對於此函數微分，最後得到最佳更新跳躍數和頻率。根據此頻率，每隔一段時間，如果有連結情形改變，就發送更新訊息到最佳更新跳躍數內的節點。當有封包要傳送時，起點檢查路由表中是否有到目的地的路徑，下一個轉傳點是誰，中繼的轉傳點收到封包時根據自己知道的路由表選擇下一點，依此到達目的地。所以一開始離目的地較遠時所知的路由資訊比較不準確，越靠近目的地，越可以得到較精確的路由資訊，選擇較佳路徑。利用 link-update 特性，在車際網路上，或許可以採用這種方式，在造成網路負擔不大的情形下，利用更新方式得到較好的路徑。

在探討車際網路中車輛間因為持續快速的移動導致車輛間距離過大，無線傳輸中斷的問題中，[17]利用觀察實際交通流量畫出車輛間距的機率分佈，再使用統計學上的 K-S test 說明道路上車輛間距近似為指數分佈。並指出車輛間距超過傳輸範圍的情形在非尖峰時段發生機率極高。接下來有大量研究探討車際網路的連通性問題，重要的議題為探討網路上任兩點之間存在有完全連通的路徑和就一個路段而言，在兩路口的間車輛數夠多可以讓在路口的任兩輛車用無線傳輸即可互相收送資料，中途每一節點都可找到下一轉傳點，不需要在任一轉傳點暫存。

就路段連通性而言，CAR[18]想要利用先計算出每條道路的連通機率當作最短路徑演算法中的權重來找起點至目的地的最佳路徑。它利用統計資料得到的車輛數配合提出的連通率模型得到連通機率。使用的模型利用排列組合將所有可能不連通的情形得到並找出不連通機率，連通機率等於一減掉不連通機率。計算不連通機率的方法如下：將道路長以車長  $d$  為單位切成格子狀共  $m$  個格子，傳輸範圍  $R$ 。當出現  $n_0=R/d$  個連續空格都沒有車子存在，代表這段道路有不連通的部分。這樣的方式雖然將所有情形都考慮到，但計算複雜度極高，當路長越長時，使用到的階層數就越大。而且並未考慮車輛在道路上其實是有機率分佈的，此法的計算是假設車輛隨意的散佈在道路上，如果考慮車輛在道路上的分佈，可以用較簡單的方法分析連通率。

綜合以上研究的問題和解法，我們想要利用即時的交通資訊，讓封包在傳送時可以依照當時道路上交通狀況來計算出最佳路徑，在不多加入 RSU，只利用道路上的車輛來節省成本。規定更新的方法讓更新的資訊可以盡快地收集交通資訊並回傳，同時儲存在路口區域讓經過路口的封包都可以因為知道最新的資訊而選擇較佳路徑。另外

針對更新的頻率做改良，並非每隔一段時間就更新交通資訊，而是在車輛密度有重大改變的情形，可能影響傳送延遲時，利用車輛數和連通性的關係找出連通機率極高適合無線傳輸的路並進行更新，進而減少負擔又能增進路由效能。



## 第三章 路徑協定

問題描述：我們針對的問題是在車際網路上，移動中的車輛要持續傳送封包給位於道路上的靜止點，要想辦法使得封包傳輸的延遲時間變短，封包更快到達目的地。

本協定主要是想利用道路上的車輛去作道路即時交通資訊(路段傳輸延遲時間)的更新並且在路口製造暫時儲存機制讓收集到的道路資訊可以增加其被利用到的機會。我們認為道路上的交通情況是瞬間變化極大的，如果封包在傳送途中有某些路段的車輛數增加或減少，和統計得到的結果相差到一定量時，此時在路徑的選擇上應該不同。也就是說，在封包傳送途中如果可以一直收到來自各路段的即時延遲，可能可以得到延遲更短的路徑。決定是否發動更新機制的門檻值由道路連通性決定，我們認為當道路已經被判定為連通，代表此時往此路段傳送可以快速傳送經過整個路段，這時交通情況一定是和統計資料相差極大。當分散式的機制偵測道路密度到達一定值，連通機率極高時，才進行更新動作。總結本協定大致分為以下部份：

1. 如何由道路車輛密度得知連通機率和分散式偵測機制。
2. 透過路口車輛發動更新機制並將收集到的資訊儲存在路口區域。
3. 將即時延遲時間配合統計資料計算的延遲時間計算道路優先權。

### 第一節 連通性偵測

以無線傳輸的假設，只要兩輛車在彼此的傳輸範圍內，傳送資料時便可以順利傳送給對方。於是我們說從路口一端開始，每隔傳輸範圍內有一台車以上，便可向前轉傳一段距離，我們稱之為傳送距離。將道路用傳輸範圍內切成若干等份，每隔傳輸範圍內都有至少一台車輛以上，需要轉傳(路長/傳送距離)次數恰可直接傳送經過整段路而不會在傳送途中某一段沒有車輛在欲轉傳的方向而傳輸中斷，封包需要由車輛攜帶才能傳送過整個路段。對任一台車而言，我們先得到其前方傳輸範圍內有一台車以上的機率，然後在此情況下，它可以往前傳送經過傳送距離，當以傳送距離轉傳了(路長/傳送距離)次時，表示已經傳輸經過完整條道路，圖一為道路連通的例子。

已知車輛在道路上自由行駛的情形，車輛在路上的分佈為指數分佈[10]。也就是說，給定一段長度，會出現一台車輛的機率為指數，參數為道路車輛密度。如果要知道對於一段道路長為  $L$  的路段，傳輸範圍  $R$ ，要達到連通機率為  $P_{con}$  所需的車輛數，必須計算達到此連通機率所需道路車輛密度，再將此密度乘上道路長度就可得到所需車輛數。對於任一台車輛， $\rho_i$  為道路車輛密度，它前方傳輸範圍  $R$  內有一台車以上出現的機率為將機率密度函數從零至傳輸範圍作積分，結果如式子(1)。

$$\int_0^R \rho_i e^{-\rho_i s} ds = 1 - e^{-R\rho_i} \quad (1)$$

如果要成功由一端路口到達另一路口，至少需要轉傳  $L/R$  次跳躍。但這是假設每

次轉傳都可以成功往前轉傳到前方  $R$  公尺處。如果在轉傳過程中每次都可以以傳輸範圍前進的話，那只要對於每隔  $R$  公尺的那些車輛而言，其前方  $R$  公尺內都有至少有一台車輛存在即可，於是就是連乘式子(1) $L/R$  次。得到的  $P_{con}$  為式子(2)

$$P_{con} = \left[1 - e^{-R\rho_i}\right]^{\frac{L}{R}} \quad (2)$$

在這邊我們要求的是給定一個  $P_{con}$ ，想要知道  $\rho_i$ 。因此可以透過以下推導

$$1 - e^{-R\rho_i} = \sqrt[\frac{L}{R}]{P_{con}}$$

$$1 - \sqrt[\frac{L}{R}]{P_{con}} = e^{-R\rho_i}$$

$$\ln\left(1 - \sqrt[\frac{L}{R}]{P_{con}}\right) = -R\rho_i$$

得到最後的  $\rho_i$

$$\rho_i = \frac{\ln\left(1 - \sqrt[\frac{L}{R}]{P_{con}}\right)}{R} \quad (3)$$

最後得到的車輛數便是式子(4)的結果

$$\rho_i \times L \quad (4)$$

但是由於車輛在道路上的位置不會是均勻的，前方傳輸範圍內有至少一台車，不代表每次往前傳輸都可以往前進  $R$  公尺。於是我們想要知道每次轉傳時平均往前的傳輸距離是多少。首先根據[9]車輛之間的距離為指數分佈。機率密度函式為式子(5)

$$f_s(s) = \lambda_s e^{-\lambda_s s} \quad (5)$$

$$\lambda_s = \lambda_t / \bar{V}$$

$\lambda_s$ :平均每公尺的車輛數

$\lambda_t$ :道路上平均每秒觀察到的車子數量

要知道在傳輸範圍  $R$  內最遠的車輛距離為何?依照指數分佈，平均車間距為

$$\bar{s} = 1 / \lambda_s$$

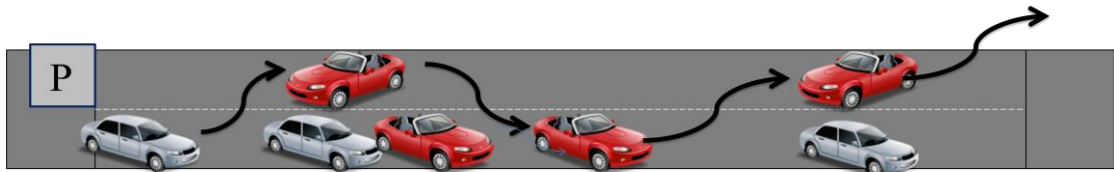
期望傳輸距離便是傳輸範圍內可以轉傳到的最遠車輛的距離，每次轉傳時可以轉傳這段距離。計算上看傳輸範圍內，車輛以平均車間距相隔，累計這些車輛間距的總和，在不超過傳輸範圍的情況下，可以存在幾台車。以傳輸範圍減去車輛間距的總和便是傳輸範圍內最遠的車輛離當前車輛的距離，也就是在找下一轉傳點時，可以轉傳前進的傳輸距離，計算方式如圖 二，結果為式子(6)。

$$E[\text{Forward\_distance}] = R - (R \bmod \bar{s}) \quad (6)$$

將此當作每次傳送可以往前前進的距離，取代式子(3)中的傳輸範圍  $R$ ，原本求連通率的式子會變成式子(7)

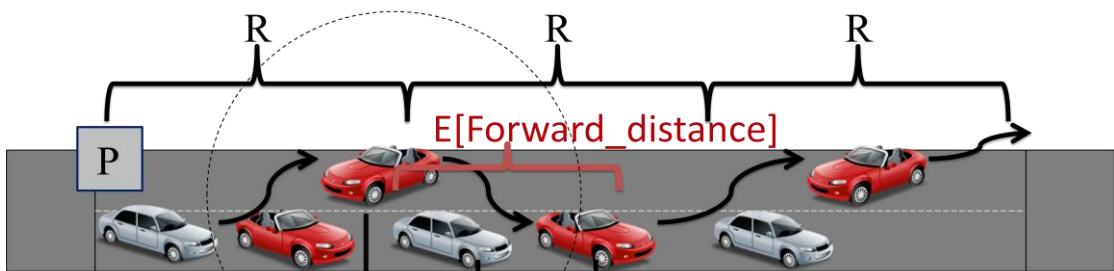
$$\rho_i' = \frac{\ln \left( 1 - \frac{l_i}{E[\text{Forward\_distance}] \sqrt{C_p}} \right)}{R} \quad (7)$$

修改過後真正所需的車輛數為  $\rho_i' \times L$ 。



Connected

圖一 連通道路範例



圖二 期望傳輸距離計算方式

在此我們對使用車輛分布來推導車輛連通機率的結果做模擬實驗，目的是要看預測的結果和實際值，也就是車輛真的在路口傳送封包的結果，以及和其他方法中提到的模型計算出來的結果做比較。越接近真實車輛產生的曲線代表能越準確地知道車輛數大於某一定值，道路上車輛就有很高的機率一定會連通。

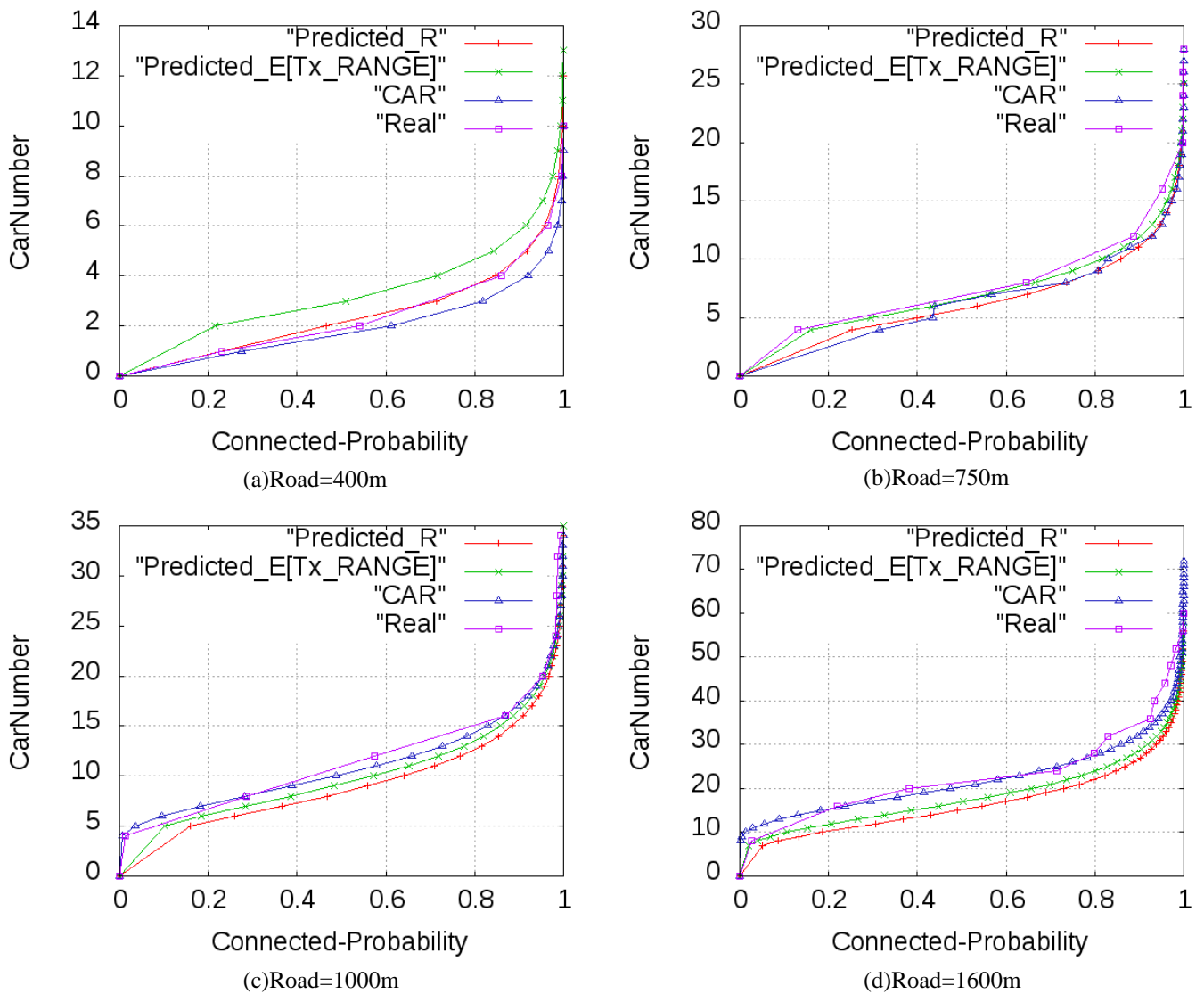
我們使 NS2(ns-2.31)[14]來模擬實際傳輸的結果以驗證理論連通率計算的準確性。比較實際傳輸的連通率，Qing Yang[13]提出的方法 CAR 以及[12]一樣是以指數分佈分析，但是以每次期望傳輸都可以前進傳輸範圍來比較。我們用車輛移動產生器[15]產生車輛的移動，模擬不同的道路密度，模擬時間 500 秒，分別在路段的兩個路口，都各自由路口一端往另一路口傳送封包，看有多少封包不會在傳送過程中遭遇到傳送路徑斷裂現象，然後除以總共送出的封包數當作連通率。針對不同道路長，不同速限，不同車輛密度，每個情境重複十次，詳細參數設定如表格一。

Real 是實驗結果，CAR 是另一個研究連通性的研究，使用組合方法計算出所有連通的情形，Predicted\_R 是假設車輛分佈為指數分佈，但每次往前傳輸都假設可以前進無線傳輸範圍，最後是我們修改後的 Predicted\_E[Tx\_RANGE]。

表格 一 模擬參數設定

Parameter	Value
MAC	802.11
Tx_Range	250m
Packet sending rate	1 packet/second
Road Length	400m,750m,1000m,1600m
Vehicle Speed	0~65km/hr(avg:54km/hr), 0~28km/hr(avg:18km/hr)

可以看到圖 三在路長為 400m 時，用我們機率分佈得到的曲線在同樣連通率下，會預測出需要多一點車輛才會連通，和實際值有所誤差，使用期望傳輸距離在道路長度短時沒有發揮作用，反而會高估所需的車輛數而產生誤差，表示車輛在道路長度不長，車輛分佈密集時，還是可以每次轉傳時都跳躍傳輸範圍。在路長 750m 時，使用期望跳躍距離後，我們可以比原本每次都以傳輸範圍跳躍的結果和 CAR 更接近真實值。在道路長度逐漸變長 1000m、1600m，用機率分佈的結果會比 CAR 差，因為車輛散佈在道路上的位置情形變化較大，分佈情形不均勻，實際上的車輛分佈和期望的指數分佈有所落差，車輛的連通性比較差無法照機率分佈這樣理想。CAR 會將車輛在道路上所有位置的可能性都計算出，但當道路長度越長，所需計算的階層數會隨之增大，計算上較為複雜。但當車輛數變多時，車輛有越高的機率平均散佈在各處，預測的結果就越靠近實際值。總體來說使用期望傳輸距離會比使用傳輸範圍較符合實際值。圖四為車速較慢時的結果。車速較慢時，各種結果和實際值的差距皆變大，因為車速慢，車輛需要較多時間才能均勻散佈，一開始車輛產生的位置可能也會影響連通率，等到車輛行駛一段時間分佈均勻後結果應該與車速快時相同，車速快慢不會影響連通率的高低。



圖三 各種連通機率模型結果比較(道路速限 65km/hr)

一旦知道在某個道路長度下，使用任一連通機率模型計算需要多少車輛才能讓連通機率極高，適合無線傳輸。要做的事便是利用道路上的車輛去偵測目前交通狀況看車輛數是否已到達門檻。使用方法如下：如圖五將道路以道路長度用兩倍無線傳輸範圍切成等分格子，在格子內每輛車子依據自己和鄰居們的位置，判斷自己是否為最靠近格子中心點車輛，最接近的車輛負責偵測此格子內的連通情形。因為位於格子中心的話，可以收到來自格子內所有車輛的位置封包而知道格子內所有車輛的位置。最靠近中心的車輛發動判斷動作，判斷自己所在格子是否可以完全連通。判斷標準為，因為每台車輛會透過位置封包得知鄰居資訊，於是對任一車輛而言，它都可以知道自己前後傳輸範圍內鄰居資訊，也就是說，它可以知道某一個大小為兩倍傳輸範圍內，總共車輛的數目。將這個車輛數對照前面已知的要達到連通機率為一定值時需要多少車輛。但前面得到的是總共道路長要幾台車，這邊感應到的是兩倍傳輸範圍內的車輛數，這邊我們將車輛數依感應距離作正規化。例如某一路段長為 1000m，需要 58 輛

車出現在道路上時，才有 0.9 的連通機率。現在對於每一個格子內最靠近中心點的車輛，當這台車發現自己的鄰居數目超過  $58 \times (500/1000) = 29$  時，認為在自己所在的格子內，車輛數量夠多，可以幫助整條道路形成連通。這時這個格子是接近連通的，對整條路而言還未知。於是要透過交換訊息的方式，來讓各個格子的中心車輛得知其他格子的連通情形。

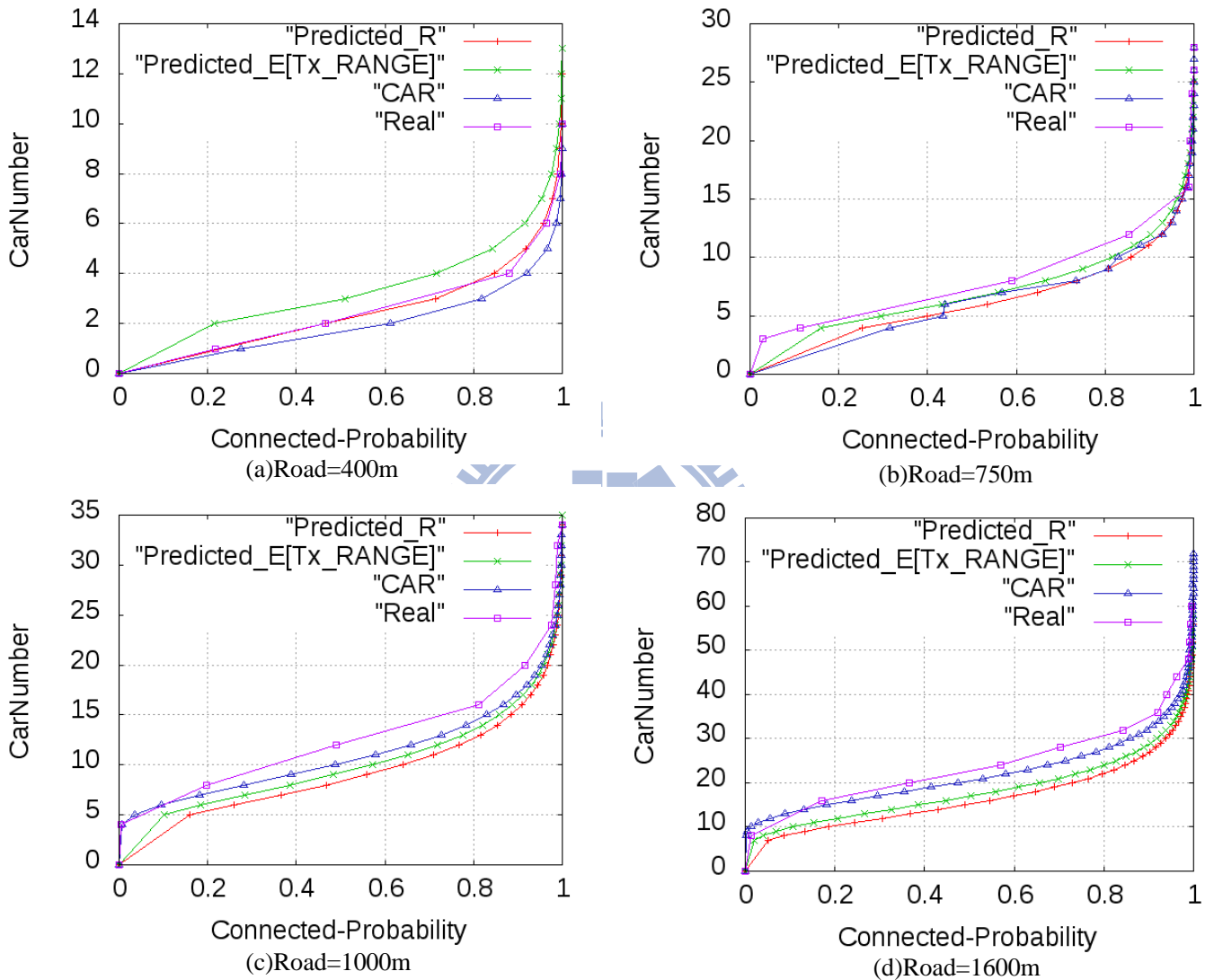


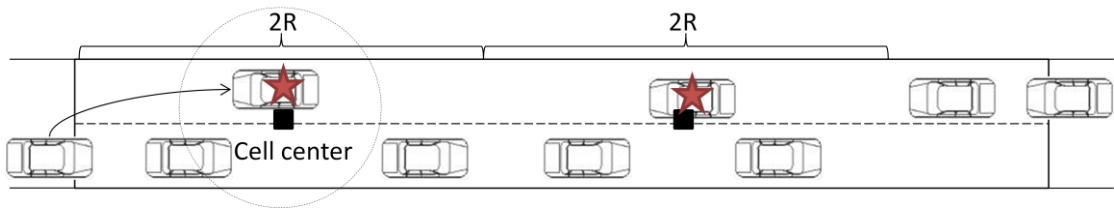
圖 四 各種連通機率模型結果比較(道路速限 28km/hr)

交換流程由最靠近一條路段的路口的格子開始發動。當格子的中心車輛發現自己所在格子是連通時，它生成一個控制封包，內含自己所在位置，選擇靠近相鄰路口的下一個格子傳送過去。選擇最接近下一個格子的中心點的車輛傳送，位於下一個格子的車輛收到之後，以位置資訊判斷此封包是否來自和自己所處格子不同的其他格子，如果是來自相同格子則往下一格子轉傳。如果是來自不同格子，看自己是否為目前格子的最接近中心點車輛，是的話就判斷連通性，車輛數超過某連通機率所需之車輛數便轉傳到下一格子，車輛數低於連通所之車輛數則丟棄此控制封包。自己不是最靠近



中心點車輛便往目前格子的最靠近中心點的車輛轉傳。

此封包經過多次轉傳後到達路口範圍。在路口範圍內的車輛，要做的事就是負責接收封包，如果可以成功接收得到，那麼我們說，這整條道路有很高的機率趨近於連通，適合作無線傳輸。而且因為車輛必須沿著道路行駛，到達路口前不會突然消失，這個偵測到的車子數量在數秒內總值不會變動太大，此資訊在短期內有一定可信度。我們在每5秒鐘執行一次此動作，以平均車速15~30m/s車輛最大會移動的距離為150m。偵測時收集到的車輛大部份還會停留在同一路段。



圖五 道路連通性偵測

## 第二節 路段延遲更新

收集路段延遲的目的是想要讓道路上的車輛可以盡可能地知道最靠近目前時間點，各路段的傳輸延遲時間，在計算路徑上加以利用。

得到路段延遲時間的來源有二。一是在資料封包中附加資訊，紀錄資料封包在相鄰兩路口經過的時間。另外再加上我們要做的，主動送出控制封包去探詢路段延遲。位於路口區域的車輛發送一個封包讓其通過整條道路去得知目前傳輸所需時間，封包格式如下:起點路口編號，目的地路口編號，送出時間。

對於目前路口，它會對所有相鄰連接到這個路口的道路各自發送探詢封包，封包沿著道路上的車輛貪婪轉傳或攜帶到達目的地路口。位於目的地路口的車輛取出送出時間標記，將收到時間減去送出時間便是實際的傳輸延遲。對於目前所在路口  $I_i$ ，其直接連接的道路個數  $n$ ，各自連接到個路口  $I_1, I_2, \dots, I_n$ 。我們對這  $n$  條道路各發送一個封包。起點  $I_i$ ，目的地  $I_1, I_2, \dots, I_n$ ，裡面存放封包送出時的時間  $\text{Send\_Time}$ 。最後位於路口區域  $I_1, I_2, \dots, I_n$  的車輛收到，收到的時間紀錄在  $\text{CURRENT\_TIME}$  欄位。

$$\text{delay}(I_i \rightarrow I_1) = \text{CURRENT\_TIME} - \text{Send\_Time}$$

如果封包沒有辦法傳送到目的地路口，而且封包的跳躍數大於三且仍然停留在起點路口，我們將其丟棄並可以得知這條路段目前不是完全連通的。

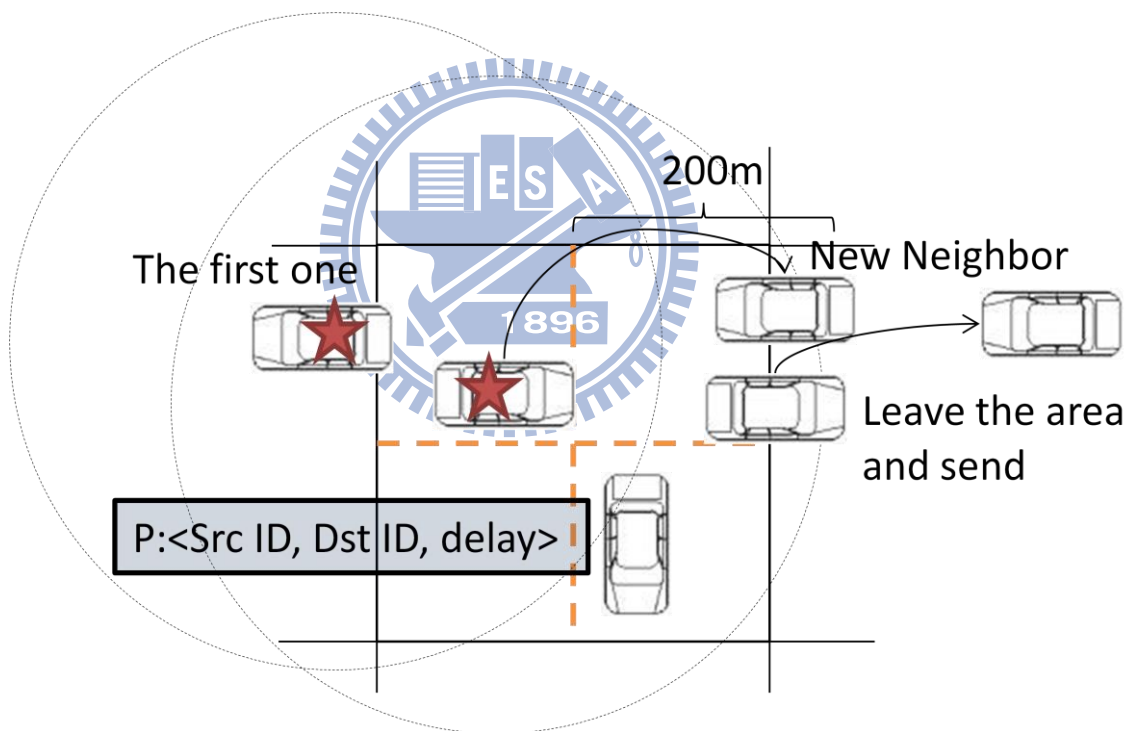
結合上一節提到的連通性偵測機制，我們將偵測連通性所需的封包和這裡提到的路段延遲探測封包合成同一封包。和原本路段延遲封包一定會在各路口發動且一定會持續轉傳到目的地路口不同。我們會發動且會將探測封包持續轉傳的依據是，在每一個格子內都要保證車輛數大於連通機率所需之車輛數，這樣可以讓每一個成功送達目的地路口的探測封包都是在連通機率达到達某一值所收集到的結果。

如果探測封包可以順利到達目的地路口，使用以下機制利用車輛製造儲存效果。如圖六錯誤! 找不到參照來源。首先第一個在目的地路口接收到探尋封包的車輛，廣

播一次收集到的延遲資訊給鄰居車輛。查詢自己的鄰居表格，將這些鄰居另存為這個探尋封包的接收者，也就是會得到這個探尋封包資訊的人。之後除非碰到新的鄰居而且不是曾經收過這個探尋封包的車輛，才會再廣播並更新所紀錄的探尋封包接收者名單。直到離開路口區域時，找尋尚未在路口區域但即將進入路口區域內的車輛，將此延遲資訊傳送給它。

在路口區域收到位於同樣位於目的地路口車輛轉傳的延遲資訊封包的車輛，收到後一樣先廣播一次並建立探尋封包接收者名單，在離開路口區域前如同上段方法找尋未曾接收過此資訊的車輛轉傳此資訊並更新探尋封包接收者名單。每一個即時資訊都有一個過期時間，時間到了就將此資訊刪除，確保可信度。

當目的地路口的車輛收到探尋封包後，它也會反向送一個封包，格式為將探尋封包的目的地路口改成起點路口，起點路口改成目的地路口。加上封包送出時間和探尋封包測量到的路段延遲。和探尋封包不同的是，這個封包內已經含有路段延遲時間，所以在傳送過程中經過的每一台車輛都可以更新這個延遲時間。當這個封包到達目的地路口時，同上我們更新車輛知道的路段延遲並且儲存此資訊在路口範圍內的車輛。



圖六 道路區域儲存即時交通資訊機制

### 第三節 混合歷史資料和即時資訊計算路徑

封包傳送流程：當有封包要從來源點送出時，不必先建立路徑。透過定位服務器找到目的地位置，然後依照來源點現在位置作不同動作。車輛會定期傳送位置封包告訴鄰居車輛自身位置，封包傳送途中每一轉傳點指定下一個轉傳點將封包轉傳給它。

在 VADD 中已經建立了一套計算路徑的機率模型，在此延用其模型針對延遲時間

計算做部份修改。以下介紹封包在路徑協定上經過的流程並加入我們修改的部份。

首先車輛在道路行駛時依據其所在位置被歸類於不同模式。每條道路和相鄰道路相接的部份為路口，如果車輛位於此區域則進入路口模式。如果位於道路上且非路口區域則為道路模式。最後在靠近目的地所在路口一段距離的範圍內為目的地模式。在不同模式下執行的動作不同。

因為是以路口為基礎的協定，車輛在路口時才需要去決定下一個轉傳路口和負責轉傳的車輛。在道路中間時只需朝著特定路口貪婪傳送找尋下一個轉傳車輛即可。

路口模式時我們需要對連接到當前路口的相鄰路口作優先權計算，並決定由要轉傳到哪台車輛。首先要知道各路段期望傳輸延遲時間。對於一路段我們利用以下算式表示其傳送延遲。如式子(8)，如果車輛擁有即時延遲資訊便使用此資訊當路段延遲不必計算。否則沿用先前研究 VADD 中的算式估算路段延遲時間。

$$d_{ij} = \begin{cases} (1-e^{-R \cdot \rho_{ij}}) \cdot \frac{l_{ij} \cdot C}{R} + e^{-R \cdot \rho_{ij}} \cdot \frac{l_{ij}}{v_{ij}} \\ \text{Real-time delay}(I_i \rightarrow I_j) \end{cases} \quad (8)$$

符號表示定義如下：

$r_{ij}$ : 從路口  $I_i$  到  $I_j$  的路段

$l_{ij}$ :  $r_{ij}$  的長度

$\rho_{ij}$ :  $r_{ij}$  上的車輛密度

$v_{ij}$ :  $r_{ij}$  上的平均車速

$d_{ij}$ : 路口  $I_i$  到  $I_j$  的期望傳輸延遲

對於當前路口和目的地路口，將中間所有道路找出做路徑延遲時間的計算，當前路口和目的地路口中間的道路，以路段為邊，路口為頂點的方式抽象化成圖，邊上的權重為延遲時間。

符號表示定義如下：

$D_{mn}$ : 在路口  $I_m$  選擇路口  $I_n$  的上車輛傳送到目的地路口的總延遲時間

$P_{nj}$ : 封包在路口  $I_n$  轉傳到  $I_j$  的機率

$N(j)$ : 在路口  $I_j$  的鄰居路口集合

$$D_{mn} = d_{mn} + \sum_{j \in N(n)} P_{nj} \times D_{nj}$$

在路口  $I_m$  經路口  $I_n$  到目的地路口的延遲時間為先以  $d_{mn}$  時間經過  $r_{mn}$  到路口  $I_n$  後，透過  $I_n$  的鄰居路口到目的地路口。所以將在路口  $I_n$  轉傳到各鄰居路口的機率  $P_{nj}$  求出，分別乘上經由這些鄰居路口傳送到目的地路口的延遲時間，這樣就是總延遲時間。 $P_{nj}$  的求法，由車輛在路口時轉彎到各鄰居路口的機率和遇到車輛在各個鄰居道路上的機率算出。

求  $D_{nj}$  時需要繼續往下將經由  $I_j$  的各個鄰居路口的到目的地路口的延遲時間求出，

但無法直接求得，要以類似遞迴的方式將所有等式列出。直到路口與目的地路口直接相接， $D_{mn}=d_{mn}$  可以直接得到值為止，每個等式以以下型式：

$$\begin{aligned} x_1 &= d_1 + P_{11}x_1 + P_{12}x_2 + \dots + P_{1n}x_n \\ x_2 &= d_2 + P_{21}x_1 + P_{22}x_2 + \dots + P_{2n}x_n \\ &\vdots \\ x_n &= d_n + P_{n1}x_1 + P_{n2}x_2 + \dots + P_{nn}x_n \end{aligned}$$

共  $n$  個等式，其中  $n$  為所有路口個數。 $x_1, x_2, \dots, x_n$  為上面提到的  $D_{mn}$ 。 $P_{11}, P_{12}, \dots, P_{1n}$  為上面的  $P_{ij}$ 。於是我們最後將這個  $n$  個線性方程式利用矩陣高斯消去法求解。就可以知道在當前路口，要選擇哪個鄰近道路上的車輛，往各個鄰近路口傳送過去的延遲時間是多少。剩下的就是去鄰居表格中找對應路口上的車輛轉傳，依優先權先選路徑延遲短的道路上的車輛，直到沒有車輛可以幫忙轉傳才由當前車輛繼續攜帶此封包直到遇到可以轉傳的車輛。

道路模式要做的事便是使用 GPSR，在路口決定好往哪個路口前進後，在該條道路上時，就一直使用貪婪法往該路口前進，每次找更接近該路口的車輛轉傳，沒有更接近路口的車輛時一樣由當前車輛攜帶直到遇到更接近該路口的車輛。

目的地模式便是當攜帶封包的車輛接近目的地路口一個固定距離時，便可以直接用距離找最近的鄰居轉傳，可以不用計算和考慮道路地圖，只要以地理位置選則中間節點即可，因為已經夠靠近目的地，不用再做其他選擇。

結合即時資訊和歷史的延遲資訊，利用以上方法，就可以讓封包在從來源點送出後，在封包傳送途中一邊經過轉傳車輛，一邊利用即時資訊而得到更接近最佳的延遲最短路徑。整個過程如圖 七，如果在傳送途中收到即時延遲資訊的話，我們可以在該路口繞路，讓封包繞到當下更快到達目的地的路段。

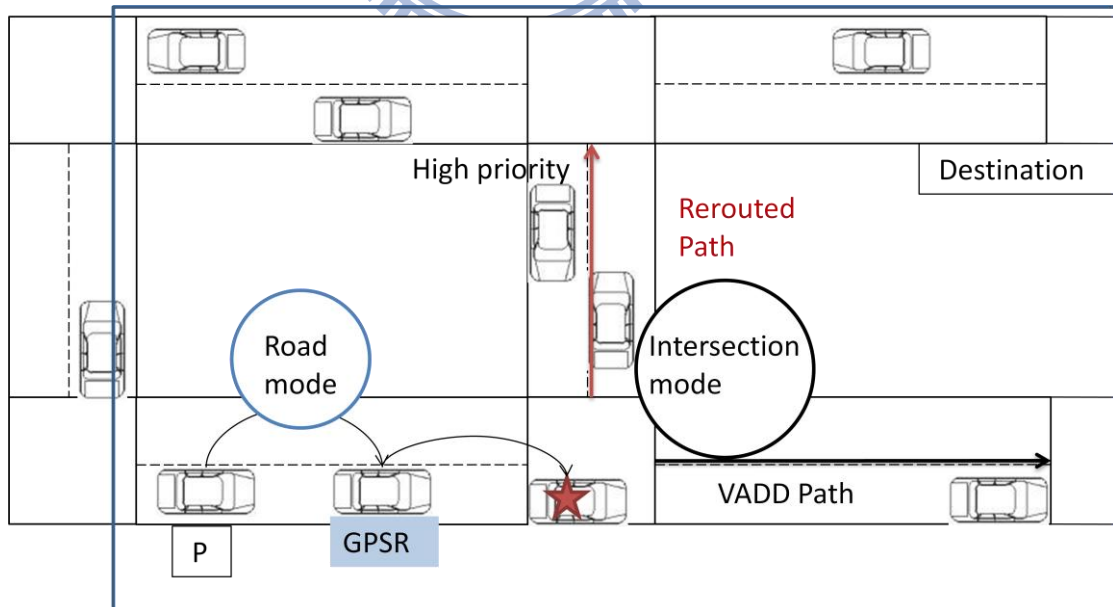


圖 七 混合即時道路延遲及歷史延遲計算最短延遲路徑

## 第四章 模擬

在本節，我們比較使用即時資訊更新的協定及只使用歷史統計資料估算路段延遲時間的 VADD，比較各種路由協定上的度量包括傳送延遲時間、傳輸成功率，以及協定本身造成的負擔。

實驗的地圖大小為 4000×3200m 的矩形格子狀街道地圖，MAC 層的協定是 802.11，有 DCF 功能。車輛移動的軌跡是使用 [20] 產生。

表格 二 模擬參數設定

Parameter	Value
Simulation Area	4000×3200m
# of intersections	24
Intersection area radius	200m
# of vehicles	150,200,250,300,350
# of senders	15
Vehicle velocity	0~100 km/hr
CBR rate	1 packet/second
Beacon interval	0.5 sec

模擬初始時車輛隨機散布在道路上，然後車輛隨機選擇任一路口當作行駛的目的地，依照最短路徑開到該路口，最短路徑演算法中的權重為道路長度除以道路速限。車輛抵達目的地之後若模擬時間尚未結束，再選下一個路口當目的地開過去。

封包傳送的目的地為設在路口的靜止點，於是我們在所有路口中隨機選擇十五個路口節點當作封包傳送目的地，在道路上所有車輛中選十五個車輛當作封包來源點傳送 constant bit rate(CBR) 資料。在模擬時間五十秒後持續傳送資料，傳送速率為每秒一個封包，詳細的實驗參數如表格 二。我們要測量比較的是傳輸成功率，傳輸延遲，和協定本身造成網路的負擔量。

在此我們比較三種方法，第一種是 VADD，原始的使用歷史收集的統計資料來計算路段延遲。第二跟第三種都是有使用即時交通資訊，但其中我們的 Conn-CP 是會先偵測道路連通性，確定連通機率高於一定值才會讓探測封包可以傳送下去並散發到周圍車輛，這邊判斷是連通機率高於 0.8 才算有足夠高的機率連通，和傳統的 Always-CP 是不論何時都要送探測封包相比，我們可以減少造成網路的負擔。

這邊較不同的是，當我們模擬時，我們發現使用即時資訊後封包傳送延遲沒有大幅下降。我們認為要在交通狀況有重大改變時，傳輸延遲才会有比較明顯的改善，如果交通狀況和歷史資料相差不多時，傳輸延遲也會有改善，只是改善的幅度不明顯。觀察其中原因，我們發現在正常交通狀況下，使用到即時資訊進而在封包傳送過程中經計算後選擇和 VADD 不同路徑的封包量其實很少，不到總收到封包數的 1%，而且

這些封包轉換路徑後延遲並不會下降，導致整體平均之後效果不明顯，當我們額外增加有劇烈變化的路段數，我們選擇總道路個數的部分百分比(3%、10%、20%)，也就是選定路段讓該路段車輛數增加由原本不連通的車輛數量增加到可以形成連通的車輛數量，這些道路為確定連通的道路，製造類似塞車或車禍之類的道路狀況異常，讓被即時資訊更新且繞路的封包數增多，再來看改善過後的傳輸延遲，延遲時間確實有縮短。下面各圖的圖例：Conn-CP 為先偵測連通率再決定是否傳送 CP，Always-CP 為隨時隨地都會傳送探測封包去得到最新交通資訊。

## 一、傳輸成功率

就傳輸成功率來看，使用 CP 之後，雖然造成網路負擔增加，傳輸成功率並不會比原本的 VADD 下降太多，約略相同，顯示傳送 CP 封包的負擔不會造成資料封包碰撞情形大幅增加。分析造成傳輸成功率下降的兩個主要原因：一是封包在傳送過程中因為和其他資料封包或者控制封包在干擾範圍內同一時間傳送導致產生碰撞，連續碰撞多次直到超過重傳次數後 MAC 層就會將封包丟棄。二是因為過期，在限定時間內，封包無法送達到目的地，就會被當成過期資料而丟棄。分析車輛數量相同時，增加確定連通的道路比例，各種方法的傳輸成功率都會稍微下降。隨著確定連通的道路比例增多，傳輸成功率會下降越多。對 VADD 而言，雖然它不知道即時資訊，也就是不知道哪些道路現在成為連通的，但在使用歷史資料計算出的優先權高的道路上無車輛時或是優先權高的道路的即為確定連通的道路時，封包都會傳送到確定連通的道路，導致該道路上資料封包數量增加，碰撞機會增加。同樣的現象也出現在使用即時交通資訊的協定，因為它們會知道即時資訊，會將封包都導向連通的道路於是碰撞機率增加。另外我們可以看到隨著車輛數量增加，增加確定連通的道路比例，此時傳輸成功率下降的幅度會變小。如圖 八，在車輛數量為 150 時，確定連通的道路比例為 20% 的傳輸成功率會比不增加連通道路的傳輸成功率下降 8%，這下降幅度隨著總車輛數量增加而變少。在車輛數量為 350 時，確定連通的道路比例為 20% 的傳輸成功率會比不增加連通道路的傳輸成功率只會下降 2%，顯然當總車輛數量增多時，封包因為某些道路無車輛而繞到連通道路的比例會變少，導致碰撞機率下降。對於使用 CP 的方法：Always-CP 和 Conn-CP 發生碰撞的次數會比 VADD 多。因為增加的控制封包導致碰撞次數增加或者是資料封包傳送到連通道路的比例增加導致在確定連通的道路上資料封包互相碰撞的情形增加。但其可以找到更多在傳送期限內可以送達目的地的路徑，於是過期封包發生的次數較 VADD 少。兩者相比碰撞的封包數量增加較多，所以傳輸成功率會比 VADD 略低。Conn-CP 減少了控制封包的傳送量，讓碰撞情形稍少，所以傳輸成功率可以比 Always-CP 略高。將各種協定在不增加連通道路個數的情況相比，如圖 九隨著車輛數量增加，傳輸成功率會逐漸上升，但是當車輛數量增加到 300 和 350 時，傳輸成功率不會再持續上升，而是略微下降到 80%，這可能是因為車輛數量增多時，碰撞情形增加的結果。

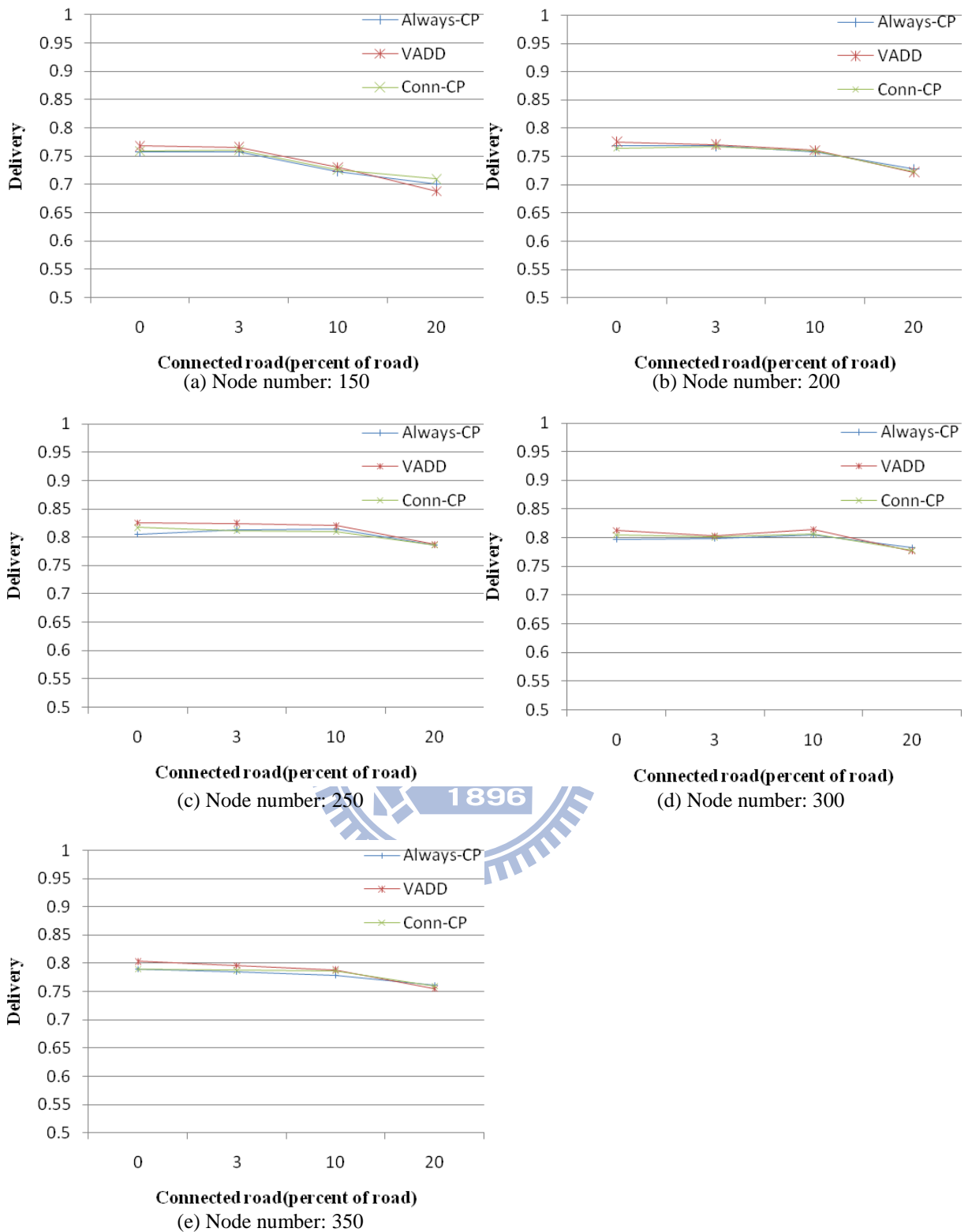
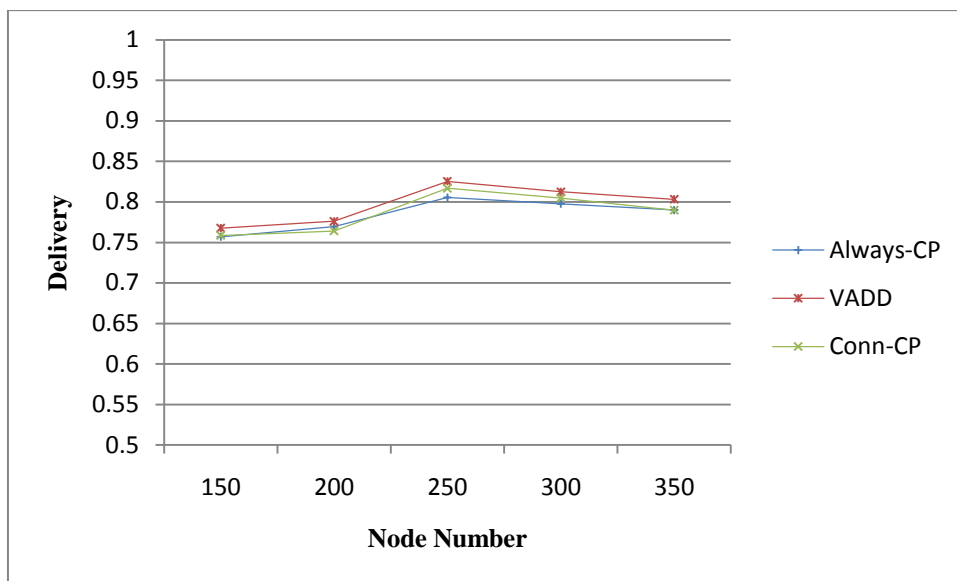


圖 八 不同連通道路比率下之傳輸成功率



圖九 不同車輛數量下之傳輸成功率

## 二、傳輸延遲

在傳輸延遲上可以看到隨著連通道路個數的比例變多，各協定的延遲都會下降，因為在計算出的路徑上的路段沒有車輛時，各協定都會經由這些連通的道路比較快到達目的地。使用 CP 的方法因為知道即時資訊，可以知道哪些道路傳輸延遲較短而改道。VADD 並非是因為知道即時資訊而改道，可能是因為原本計算出優先權高的道路上無車輛而轉向連通的道路。使用 CP 的方法可以將較多的封包導向延遲變短的路徑上，所以傳輸延遲可以比 VADD 更低，隨著連通道路比例的增加，延遲也下降越多，在車輛數量為 350 時，最多可以比 VADD 下降 20% 的傳輸延遲時間，就不同車輛數量平均而言，可以比 VADD 下降 10%~20%。在道路情形沒有改變的情形下，延遲下降的情形不明顯，這是因為本來 VADD 封包就可以行經車輛密度較高的路段，而我們只是能找到那些車輛密度變化大的道路，如果車輛分佈情形和 VADD 在計算時使用的歷史資料符合。沒有出現道路車輛數變化劇烈的情況時，比較不出差異。因為即使有即時資訊的協定，在計算道路優先權時都可能計算出和 VADD 同樣的優先權，封包會被轉傳到同樣的路徑上，如此一來延遲和 VADD 相比便不會有所差別。另外，Conn-CP 的延遲時間和 Always-CP 相近，可知在發現道路連通時才做更新的動作也可以讓傳輸延遲降低，在道路連通時，封包可以快速通過路段，這時才更新，可以讓經過更新資訊計算過後而變換路徑的封包，比較可能出現變換過後延遲會降低的情形。原本的 Always-CP，可能會使封包變換路徑後，轉向延遲較長的路徑。因為即使道路不連通封包也轉向，但不連通時表示這些道路有可能在傳送途中還是會中斷，在道路上封包需要被攜帶在車輛上，或者經過數次轉傳後還是停留在路口區域內，延遲反而會增加。另外，隨著車輛數量增加，各種協定因為每個節點周圍有較多節點可以幫忙傳送，傳送延遲都會隨之下降。



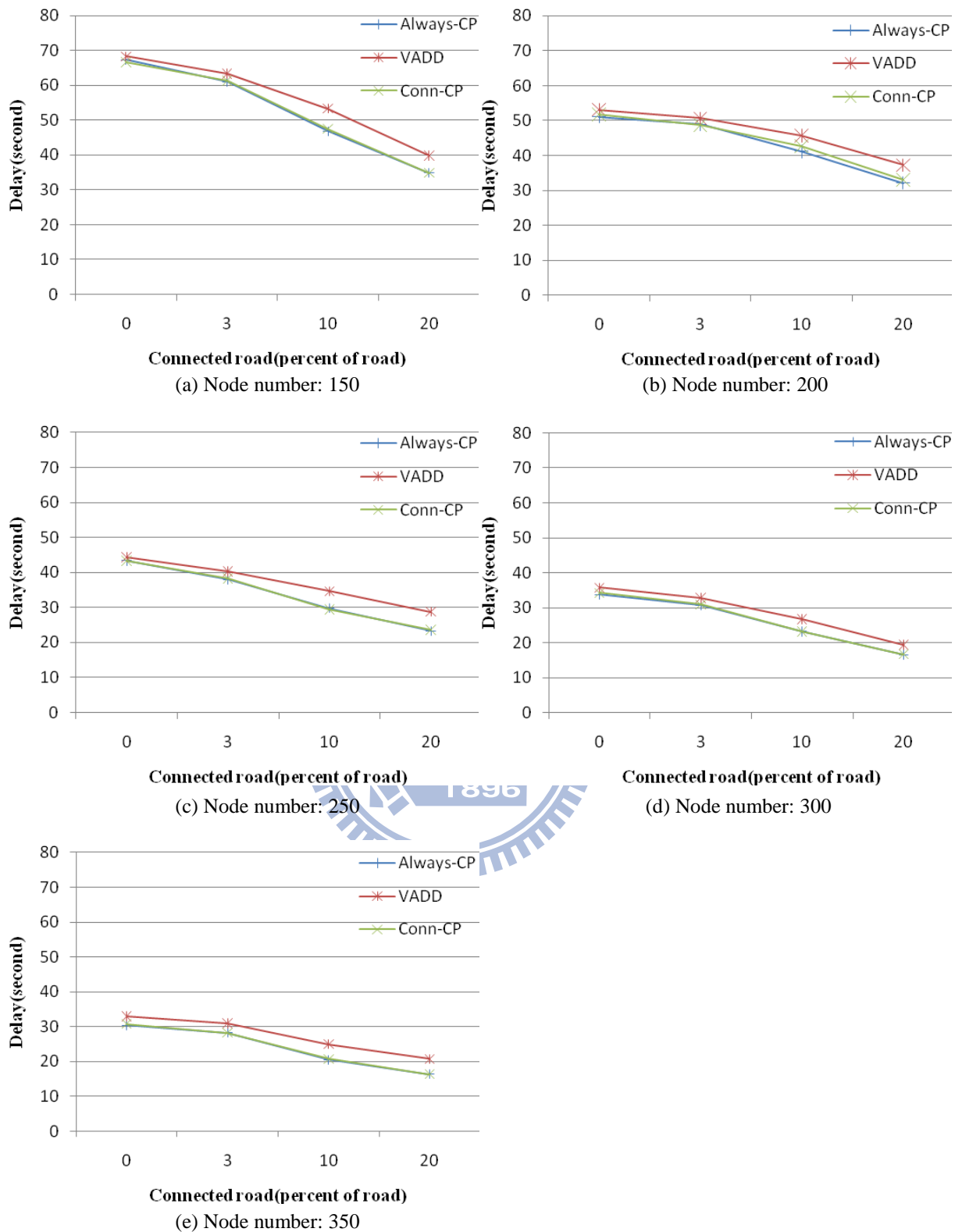


圖 十 不同連通道路比率下之傳輸延遲

### 三、協定負擔

我們分析在模擬中傳送的控制封包總量，以傳送出的控制封包大小總位元數表示，比較各協定對於網路造成的負擔。原本的 VADD 在行進途中只有定期發送的位置封包，這是為了知道鄰居位置所必須的負擔。隨著車輛個數變多，會有些微增加。使用 CP 的

方法需要多傳送封包去探詢道路延遲和儲存動作，需要多一些負擔量。在 Always-CP 下，車輛會定期地去執行探詢動作，所以會有較多的額外負擔，在這邊看到增加的負擔量並沒有非常大，最大增加的負擔約略是原本 VADD 的 20%，平均而言增加的負擔量約為 VADD 的 16%。而使用 Conn-CP，是有條件式的發送探詢封包，可以降低控制封包總量，會比 Always-CP 降低 3~4%，也就是比 VADD 多出 10~16% 的控制封包。使用 Conn-CP，在協定造成的負擔上確實有所減少

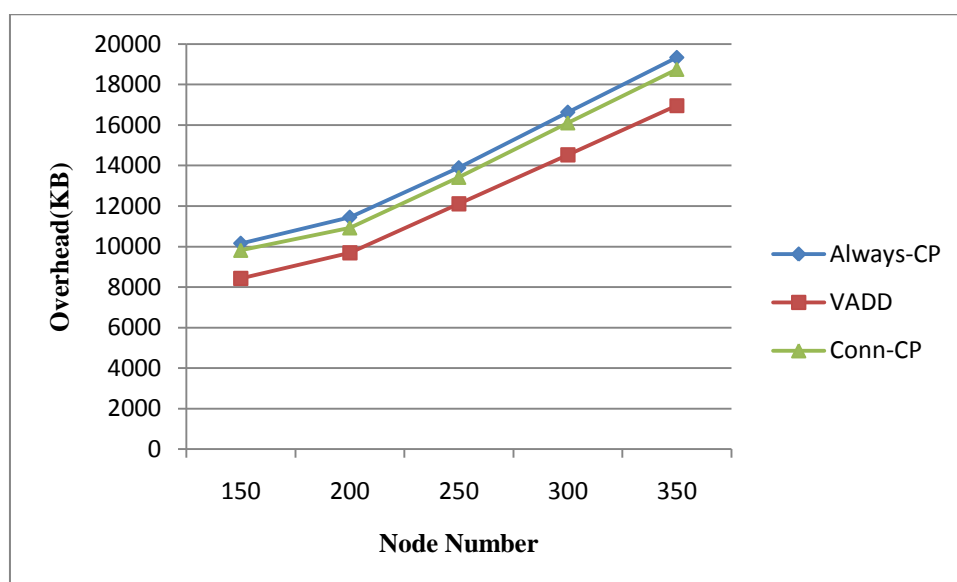


圖 十一 不同車輛數量下之協定負擔

#### 四、繞路封包佔總收到封包的百分比

這邊看的是和 VADD 相比，有多少比例的封包在傳送途中，因為得到即時資訊而經過計算後，在決定道路優先權時，選擇傳往和 VADD 不同的路段，最後成功到達目的地。在此列出不同車輛數量時的結果，可以看到在道路情形沒改變的正常情形下，大概是 10% 的封包會變換路徑，但此時這些封包造成的傳輸延遲有的比走 VADD 原路徑好，有的比走原路徑差，結果並不能造成傳輸延遲下降。隨著已知連通的道路比例增加，繞路封包的比例也會上升，也就是越多的封包因為即時資訊而繞路。但這邊看來繞路封包的比例不會上升劇烈，最高比例就落在 13%~14% 左右，顯示繞路的情形可能還是有限。Conn-CP 的繞路封包比例稍低，因為其只有在連通情形才會發送探詢封包，所以影響的封包個數會較少。

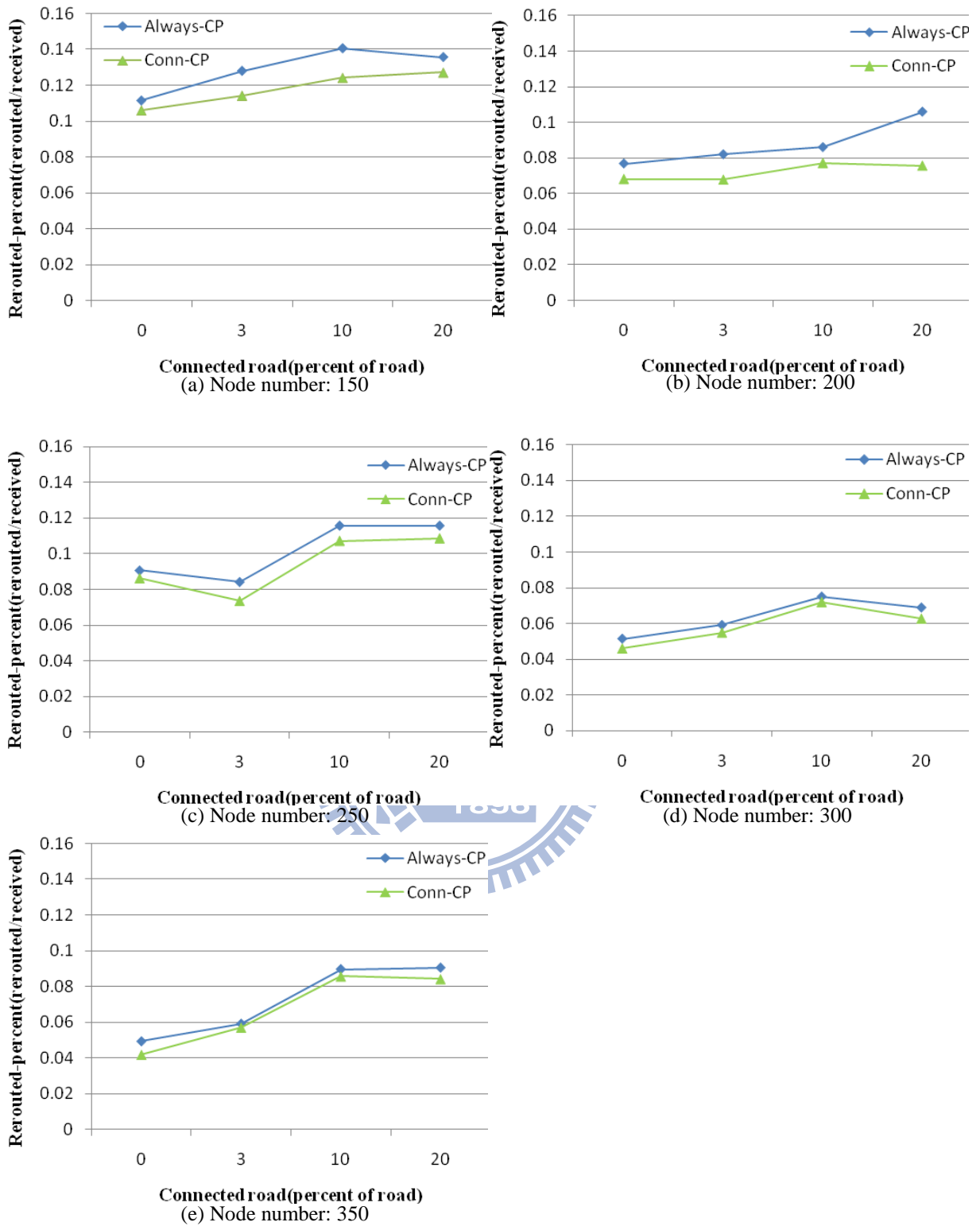


圖 十二 不同連通道路比率下之繞路封包比率

### 五、繞路封包在延遲上的增進

我們取出那些在路徑選擇時，會行走和 VADD 協定不同路徑的封包，去比較這些封包可以比走 VADD 延遲減少的秒數。在正常交通情形下，CP 方法並不能在傳送延

遲上增進，而且有可能因為控制封包變多，讓資料封包傳送途中可能碰撞次數變多或者鄰居資訊不準確而導致延遲反而變差。在交通狀況有改變下，也就是增加確定連通的道路比例，可以看到使用 CP 方法的延遲，可以下降 10~20 秒，和所有封包的平均延遲時間減少的量相比，有產生繞路現象的這些封包，延遲減少的量比較明顯，顯然 CP 有發生效用。Always-CP 下降的秒數會隨著連通道路個數的增加而增加，可見連通道路越多，封包可以走到連續數個連通的道路而讓延遲下降越多。而 Conn-CP 保守地只選擇那些連通的道路去做更新，延遲時間減少的量會比 Always-CP 少。顯然就繞路封包的延遲時間增進而言，在所有道路都做更新得到的好處比只更新連通路段的延遲資訊來得多。

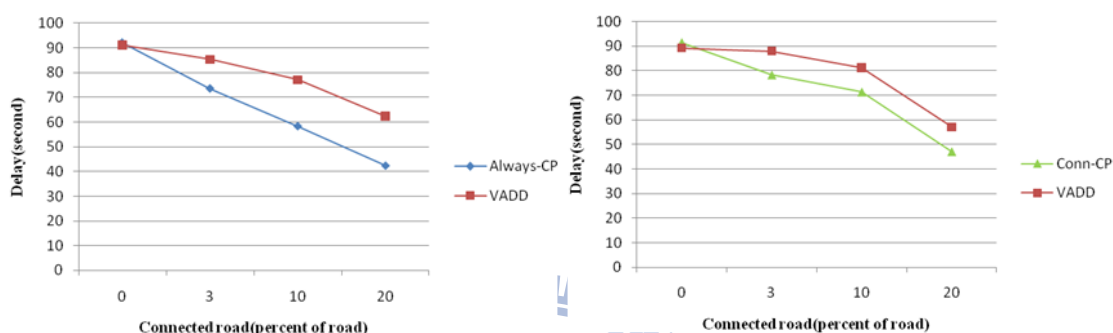


圖 十三 車輛數量:150 不同連通道路比率下之繞路封包平均傳輸延遲

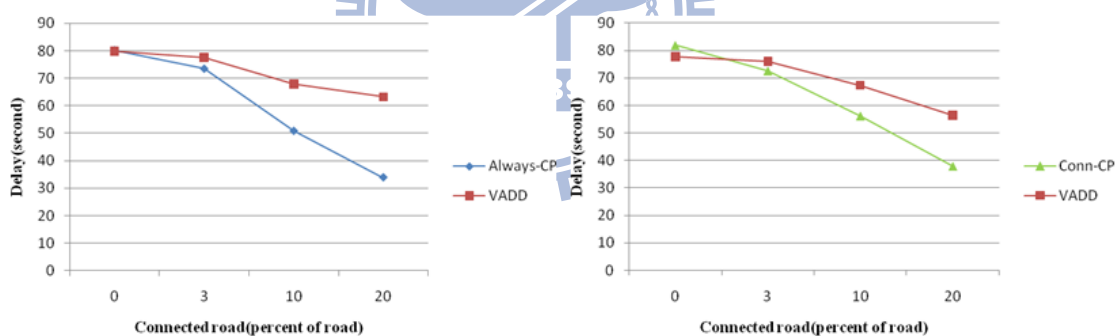


圖 十四 車輛數量:200 不同連通道路比率下之繞路封包平均傳輸延遲

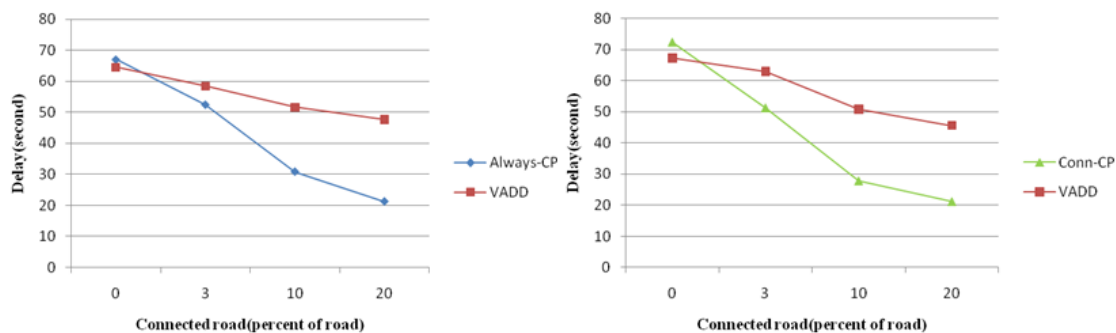


圖 十五 車輛數量:250 不同連通道路比率下之繞路封包平均傳輸延遲

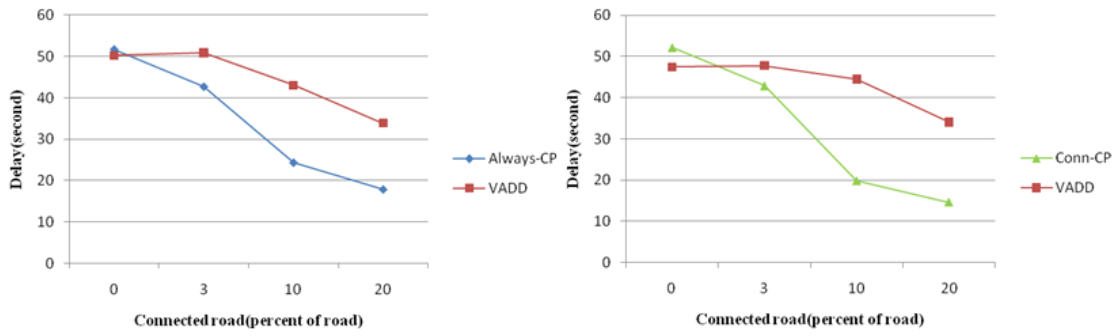


圖 十六 車輛數量:300 不同連通道路比率下之繞路封包平均傳輸延遲

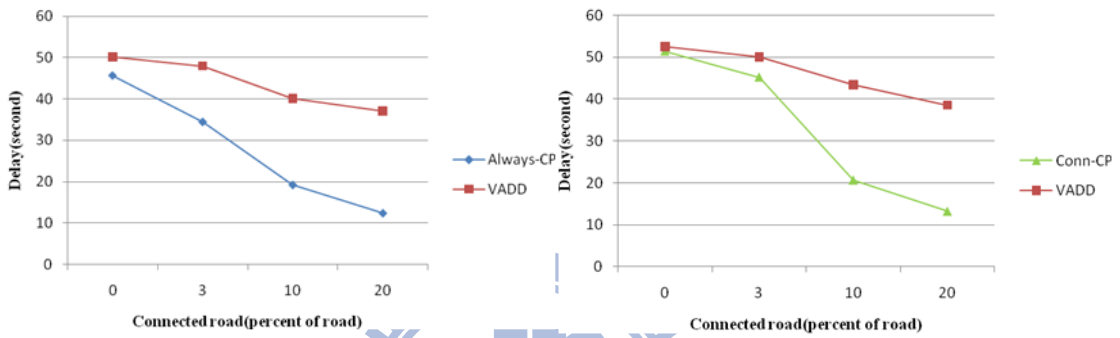


圖 十七 車輛數量:350 不同連通道路比率下之繞路封包平均傳輸延遲



## 第五章 結論

在本文中，我們利用了交通即時資訊達到了隨著道路狀況改變，在封包傳送到路口時，可以即時更改傳送路徑到傳輸延遲更短的路段上，進而使得封包傳輸延遲時間下降的目的。藉由發送額外的控制封包我們可以得到各路段上的最新延遲，然後利用道路上的車輛製造暫存機制儲存此資訊並發送到鄰近路口以增加此資訊的被利用度。為了減少控制封包造成的負擔，我們研究道路連通性與車輛數量的關係，使用連通性偵測找出連通性極高的路段，讓收集交通資訊的動作有條件地在連通機率夠高的道路上執行。在減少協定負擔的情況下確保傳輸延遲極短的路段上的延遲資訊可以被收集且利用。最後在封包傳送時，參考先前研究，在路段上一樣使用地理位置路由傳送，在路口時我們便可以利用收集到的交通資訊和歷史資料計算出的路段延遲計算出較佳的轉傳方向並轉傳讓封包沿著當下最短的路徑抵達目的地。在連通率模型的驗證上，我們透過模擬比較連通率模型和實際值的差異並和其它研究比較，結果顯示我們的模型和其它研究相比可以在計算複雜度較低的情形下，得到更準確或相近的預測結果。在路徑協定的實驗中，我們證實了在傳輸延遲方面，使用即時交通資訊的方法並且有連通性偵測的協定和不使用即時交通資訊的協定相比確實可以減少封包傳輸延遲。隨著交通狀況改變比歷史資料變化越來越大，增進的幅度會越來越多。和無條件傳送控制封包的即時交通資訊協定相比，我們可以減少傳輸負擔，略為提升傳輸成功率，同時所有封包的平均傳輸延遲仍然相近。整體來說本協定可以適用在車際網路上延遲時間限制在十秒至數十秒的環境中，但還是有可以改進的地方。首先針對我們協定本身而言，我們希望可以在不傳送探測封包的情形下，還是可以得到道路上目前最新延遲資訊進而行走最接近道路延遲最短的路徑上，減少傳輸時間。再者針對傳輸延遲時間限制嚴格的應用，因為我們協定極為依靠道路上的車輛傳輸，導致傳輸延遲過長，除非在道路上車輛密度夠高的情形上才能符合延遲時間的要求，需要其他設計來幫助降低傳輸時間，這是可以改進的方向。另外在模擬過程中我們發現了一些現象，就是當道路車輛分布不均勻時，包括我們在內的協定都會將封包頻繁地傳向車輛密度較高的路段，造成那些路段上封包碰撞的機率增加，導致傳輸成功率降低，針對這種情形應該要有負載平衡機制調節封包傳送的行為來減少碰撞機率。經由以上修改後的路徑協定便可以有低延遲及高傳輸成功率的通用性，更適合於車載網路的各種應用。

## 第六章 參考文獻

- [1] “Dedicated Short Range Communications (DSRC) home,” <http://www.leearmstrong.com/dsrc/dsrchomeset.htm>.
- [2] Q. Xu, T. Mark, J. Ko, and R. Sengupta, “Vehicle-to-vehicle Safety Messaging in DSRC,” in *Proceedings of VANET*, Oct. 2004.
- [3] J. Yin, T. ElBatt, G. Yeung, B. Ryu, S. Habermas, H. Krishnan, and T. Talty, “Performance Evaluation of Safety Applications over DSRC Vehicular Ad Hoc Networks,” in *Proceedings of VANET*, Oct. 2004.
- [4] C. PERKINS, Ad hoc on demand distance vector (AODV) routing. Internet-Draft, draft-ietf-manet-aodv-04.txt, Oct. 1999.
- [5] D. B. Johnson, and D. A. Maltz, “Dynamic source routing in ad hoc wireless networks,” in *Mobile Computing*, 1996, on. 5, pp. 153-181.
- [6] B. Karp and H. T. Kung, “GPSR: Greedy Perimeter Stateless Routing for Wireless Sensor Networks,” in *Proceedings of MobiCom*, Boston, MA, Aug. 2000.
- [7] M. Jerbi, R. Meraihi, S.-M. Senouci, and Y. Ghamri-Doudane, “Gytar: Improved greedy traffic aware routing protocol for vehicular ad hoc networks in city environments,” in *Proceedings of 3rd ACM Int. Workshop VANET*, Los Angeles, CA, Sep. 2006, pp. 88–89.
- [8] M. Jerbi, R. Meraihi, S. -M. Senouci, and Y. Ghamri-Doudane, “Towards Efficient Geographic Routing in Urban Vehicular Networks,” *IEEE Transactions on Vehicular Technology*, vol. 58, no. 9, pp. 5048-5059, Nov. 2009.
- [9] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter, “MDDV: A mobility-centric data dissemination algorithm for vehicular networks,” in *Proceedings of 1st ACM Int. Workshop VANET*, Philadelphia, PA, Oct. 2004, pp. 47–56.
- [10] J. Zhao and G. Cao, “VADD: Vehicle-assisted data delivery in Vehicular ad hoc networks,” *IEEE Transactions on Vehicular Technology*, vol.57, no. 3, pp. 1910–1922, May. 2008.
- [11] Y. Ding and L. Xiao, “SADV: Static-Node-Assisted Adaptive Data Dissemination in Vehicular Networks,” *IEEE Transactions on Vehicular Technology*, vol.59, no.5, June 2010.
- [12] J. Nzouonta, N. Rajgure, G. Wang, “VANET Routing on City Roads Using Real-Time Vehicular Traffic Information,” *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, Sep. 2000.
- [13] C. Lochert, M. Mauve, H. Fusler, and H. Hartenstein. “Geographic routing in city scenarios,” *ACM SIGMOBILE Mobile Computing and Communications Review*, pp. 69–72, 2005.

- [14] B.-C. Seet, G. Liu, B.-S. Lee, C.H. Foh, K.J. Wong, K.-K. Lee. “A-STAR: A Mobile Ad Hoc Routing Strategy for Metropolis Vehicular Communications,” *Networking*, pp. 989-999, 2004.
- [15] B. Liu, B. Khorashadi, D. Ghosal, C. Chuah, M. H. Zhang. “Assessing the VANET’s Local Information Storage Capability under Different Traffic Mobility,” *IEEE INFOCOM*, Mar. 2010.
- [16] C. Santivanez, R. Ramanathan, and I. Stavrakakis. “Making link-state Routing scale for ad hoc networks,” in *Proceedings of ACM Mobihoc*, Oct. 2001.
- [17] N. Wisitpongphan, F. Bai, P. Mudalige, and O.K. Tonguz, “On the routing problem in disconnected vehicular ad-hoc networks,” in *26th IEEE International Conference on Computer Communications*, May. 2007, pp. 2291–2295.
- [18] Q. Yang, A. Lim, and P. Agrawal, “Connectivity aware routing in vehicular networks,” in *Wireless Communications and Networking Conference*, 2008, pp. 2218–2223.
- [19] Network Simulator 2 (NS2), <http://www.isi.edu/nsnam/ns/>
- [20] A. K. Saha and D. B. Johnson, “Modeling mobility for vehicular ad hoc networks,” in *Proceedings VANET—Poster*, Philadelphia, PA, Oct. 2004, pp. 91–92.

