

# 1 簡介

在歐、美、亞洲已有不少國家佈建了智慧型運輸系統(Intelligent Transport Systems, ITS)，此系統雖然使人們生活水準提升，但現今的智慧型運輸系統還是相當依賴基礎建設的架構，如：路旁的感應裝置、攝影相機…等。智慧型運輸系統的領域很廣，其中車用隨意網路 (Vehicular Ad Hoc Networks, VANET) 近來更受到大家的重視，不少學者紛紛投入此一研究。因為此類系統，除了有路側裝置對車輛(Roadside-to-Vehicle, R2V)的網路架構外，更有著車輛對車輛(Vehicle-to-Vehicle, V2V)的溝通模式，如此減少了基礎建設的使用，進而節省大量的佈建成本。再者，利用此 V2V 的架構，即使在缺少基礎建設的鄉村地區，也能有基本車間網路可供使用。

VANET 研究還有個令人最感興趣的地方是其應用可使行車安全性跟舒服度提升。據統計，在美國每年平均有 250 萬人於交通事故中受傷，而死亡的人數更有 4 萬人之多；而以人口數較少的台灣而言，據 2005 年的統計，因交通意外死亡人數竟也高達 2,894 人[1]。假若我們可以好好利用車間網路來傳送訊息，把安全警告封包運用在車禍碰撞警告、紅燈減速提醒等應用上，將更能保障我們的生命安全。有數種可能的網路架構可以連結車間的通訊：如直接無線車輛對車輛間的隨意網路；或是僅一步跳躍後連上有線的後端網路形式；最後就是混合前二者通訊形式的網路架構。在混合式架構裡，車間通訊可不依靠基礎建設的架構，但卻利用基礎建設架構的優點，讓在某些情況下傳送資訊，進而提升整個網路的效能。

儘管 VANET 的應用可在上述的網路架構上運作，但由於 VANET 的特殊性，使得一些在 MANET 上之著名繞徑方法無法直接應用在 VANET 的環境中。因此，各種針對 VANET 環境之繞徑方法紛紛被提出，其中利用道路車輛密度來做路徑選擇的道路導向繞徑 (road-based routing) 便是主要的方式之一。

以前 MANET 繞徑是把通訊儀器當成節點，兩個儀器若位在彼此的傳輸範圍內，則存在一個無形的邊；而道路導向繞徑是以道路為基本構成的繞徑方法，此繞徑把街道圖轉換成一個道路網路拓撲；其中交叉路口被當成節點，而兩個路口所夾的道路成為邊，如圖 1-1 所示。

一個繞徑協定的好壞，可由下列幾個方向去評斷：封包傳遞率之高低、點對點的延遲

時間長短或繞徑協定的控制負載程度等。在 VANET 中，封包傳遞率若想要提高、點對點的延遲時間要縮短、繞徑協定的負載要小，欲達成上述目標，則車輛密度是個重要關鍵。一般來說，車輛密度越高的道路，能選的封包傳遞路線也較多，繞徑協定就是要選擇此類的路線，是故道路導向繞徑把車輛密度較大的邊，設定為有比較高的權重值，則被選擇成為路徑的機率將會比較大。所以，車輛密度的精準度會影響整個繞徑協定的效能。但傳統研究 VANET 繞徑的方法裡，都會把車輛密度假設為已知，或使用集中式的伺服器，不然就使用歷史的統計資料替代。無論如何，這些方法都不足以代表即時的路況資訊。因此在這篇論文裡，如何得到即時的車輛密度將是我們關注的焦點，所以我們提出了一個精準的即時車輛密度測量協定(Real-time Traffic Estimation Protocol, RTEP)。實驗模擬結果顯示 RTEP 有不錯的準確度。RTEP 可用於 VANET 之各種應用中，並能改善延遲太久或只使用歷史車流資料而造成誤差過大的缺陷。

本論文架構如下：在第二章我們將介紹一些 VANET 背景知識跟相關研究。而 RTEP 方法則將在第三章做一詳細的介紹。第四章將描述 RTEP 準確度的模擬探討。最後此論文的結論則至於第五章。

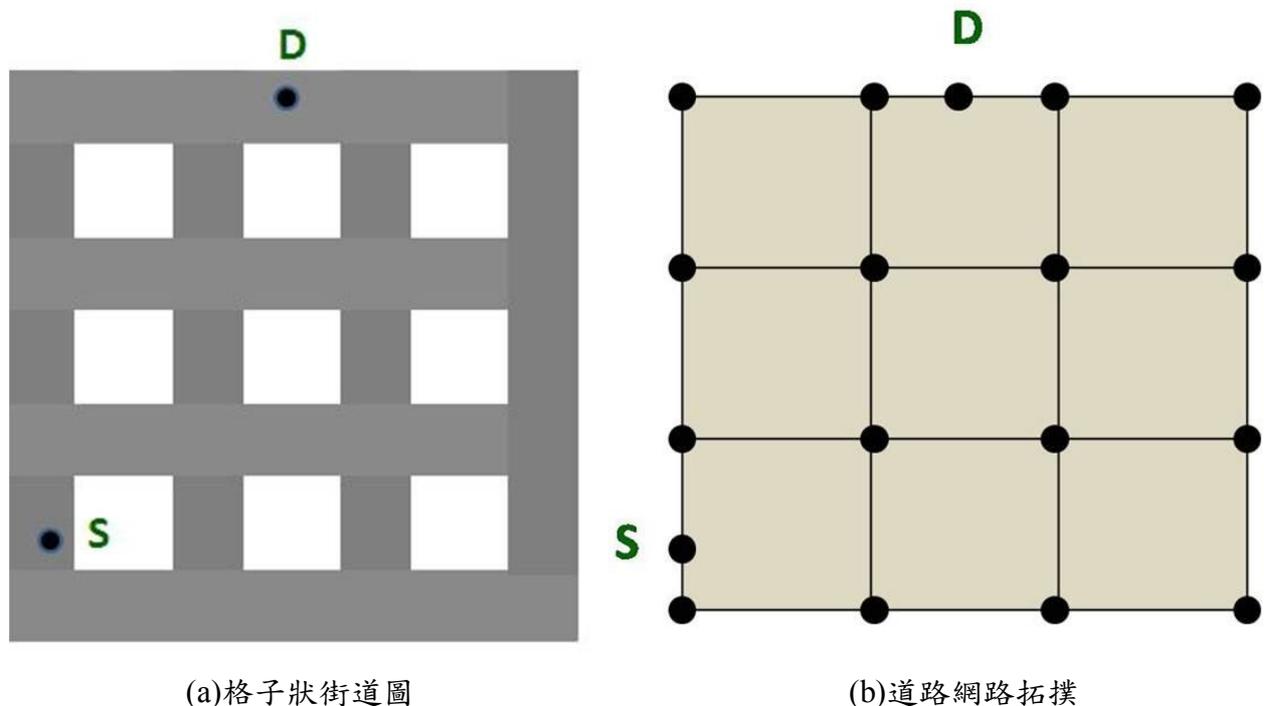


圖 1-1 街道圖轉網路拓撲示意圖

## 2 相關背景與研究

在此一章節裡，我們將介紹 VANET 背景知識跟相關研究。

### 2.1 VANET 簡介

VANET 全名為 Vehicular Ad Hoc Networks，是由車輛(vehicles)組織成隨意網路(ad hoc network)型態的一種網路。參與 VANET 的車輛必須裝備一些儀器，使其有成為網路節點的能力。其中計算跟通訊的裝置是最基本需具備的儀器。計算裝置提供節點計算的能力，使車輛能夠自我判斷封包的行為模式；而通訊裝置則負責傳遞資料與交換資料等用途。為了使 VANET 應用更加廣泛，許多輔助的設備也被安裝在車上。例如：全球定位系統(Global Positioning System, GPS)，可輕鬆達到定位的效果，使車輛能夠得知本身的移動軌跡；若再加上事前安裝好的電子地圖，配合 GPS 所獲得的座標，車輛即可得到導航的輔助；一些感應裝置，(如：測量距離或溫度等裝置)將能更有效應用在行車安全上。

如同在第一章已經提及的，VANET 的網路架構有三種可能的選擇，。在此論文裡，V2V 的架構是主要討論範圍，我們所提出來的 RTEP 也是在其上運作。V2V 即所謂車輛跟車輛之間的直接通訊而不透過基礎建設的一種架構；V2V 亦有 IVC(Inter-Vehicle Communication)之稱。以車輛為節點，只要在車輛彼此間的通訊範圍內，皆可以作交換資料，這跟 ad hoc network 十分的類似。簡單的說，車輛在道路上行走，若車輛彼此在通訊範圍內，且有一條可利用的連線，多輛車之間將形成一個網路，並可以多跳躍傳送訊息，這就形成一個 VANET。因此，VANET 其實就是 ad hoc network 的一個特例。在文獻[2]裡提到 ad hoc network 的許多特色，VANET 有著幾點特性不同於 ad hoc network：(1)VANET 移動更加的快，而速度大致是可預測的；(2) VANET 的拓撲是極動態的，由於車輛的高速移動性，車輛間彼此的連結狀況也會變動的很快；(3) VANET 節點的移動是被限制住的，不像 ad hoc network 的節點能有隨意方向的移動，車輛只能限制在道路上行駛，正因為這點，我們可以大致預測出車輛在某段時間後的位置。(4)網路規模會大的多，只要有車輛的存在，網路規模就可以無限的延伸。(5) VANET 之間的連線會常斷離，像是因紅綠燈的阻隔，或者是人為的加減速。(6)車子使用汽油等當作能源，故在 VANET 裡，能量節約將不是主要議題。在文獻[11]裡，作者把 VANET 跟 MANET 作了比較(如表 2-1)，這些差異顯示出我們要在 VANET 散佈資料會變的比 ad hoc network 上更加困難。

表 2-1 MANET 跟 VANET 的差異處

	MANET	VANET
節點數目	通常在 100~1000 之間	沒有明確界限，可能為上百萬輛的車輛
移動區域	通常為幾萬平方公尺之內	沒有明確界線，可以大到是一個國家的範圍
移動性	低度~中度	高度
節點移動軌跡	隨機方向	大多為單一方向
節點的分布	隨機且均勻的	稀疏且分散的

## 2.2 資料散播型態與資料傳遞機制

VANET 上之繞徑協定得依據下列目標所設計：要即時性、延遲要小、要可靠性且封包傳遞率要高，而為了符合不同的車間應用，散播資料的方式亦可分為下列幾個種類：單點傳播(unicast)、多點傳播(multicast)、任一點傳播(anycast)和搜尋傳播(scan)[3]。一般而言，當接收資料者是特定的節點時，就是所謂的單點傳播，資料使用某個傳遞機制(下一段裡會加以敘述)，將其傳送給目的地節點。假若目的地節點是固定不動，這是容易解決的情況，此時資料可就精確的目的地座標傳送；反之，若目的地節點會移動，則可能需要先估算出目的地車輛可能會在的位置，當封包傳送到附近時，再進行當地的散佈。而多點傳播即當一個資料要傳送給多個節點使用時。任一點傳播簡單來說就是傳資料給目的群組中任何一個人即可達成目標。以求救信號而言，只要任何一個警局或警車收到求救訊息就可以了。搜尋傳播是資料將環繞某個範圍，直到找到符合要求的節點，像是搜索鬧道口連上網際網路，就是這一類的代表。

事實上任一種應用服務，都有可能使用多個散佈資料的方法。就像是我們開車時，若需要某個地區的旅店資料，我們就會先採多點傳播來詢問鄰近的節點有無這類的資訊，一但有符合的資訊，其鄰近的節點將會採單點傳播傳送回來給我們。這就是兩種散佈資料的方法組合成的一個 VANET 運用。

資料傳遞的機制是指資料要如何能在網路裡移動的方法，這可說是繞徑協定的核心主軸。若網路是連結的，則資料的傳遞就方便許多，我們可以使用一連串的節點當成是資料

傳送的路徑[4]。然而，在 VANET 裡，高速移動的特性，將使兩點間連線處於極不穩定的情況，因此先找出路徑往往是無法精準利用的。在動態的網路裡，位置導向繞徑是個解決前述問題不錯的選擇。以位置資訊為主的方法[6]，是不需事先決定好一條路徑讓封包走，而是每個節點自己隨時要決定是否接收封包與是否要將封包傳遞下去。但位置導向繞徑還是無法解決網路斷離的情況，一但出現這個問題時，根本沒有臨近的節點可以幫忙傳遞資料。以下是改進網路斷離情況，目前上被提出的方法簡述。其中，廣播協定(broadcast protocol)是一個簡單的改進方法，這可以稍微改善網路密度太小所造成的問題，然而這會造成另一個廣播風暴的問題。流言協定(Gossip protocol)[7]雖然使用廣播，但只允許某部分的中間節點幫忙做再廣播的動作，這減少了網路上封包的負載程度，但是還不足以確保封包能到達目的地(缺乏可靠性)。另外，為了解決網路斷離、節點間路徑不存在的情況，容遲協定(delay tolerant protocol)[8, 9]被一些學者所提出，此類協定在無鄰近的節點或沒合適的節點時，車輛節點自己產生或要幫別人轉送的資料可以先存在自己的緩衝區裡，等到有機會時，有車輛進入傳送範圍內，才將資料複本傳送出去，並留著原始的資料，以備未來再做傳送的準備，此種方法可大大提升遠徑協定的可靠性。



## 2.3 車輛密度的計算

在傳統測量車流量的方法，主要是依靠在路旁或公共設施上架設感測裝置，例如：攝影機跟雷達，在[16, 17]的方法就是利用此種方式。此類方法的缺點是沒有架設感測裝置的路段，就無法得知其道路的車輛密度，若要每條路都裝上感測裝置將會花費很大的佈建跟維護成本，因此使用上受到了極大的限制。[18]提出了一個減少感測裝置的方法，這篇文章利用中點法可以算出理想的感測裝置個數，進而降低一些成本的花費。

以往都是利用集中式的方式去計算並散播車輛密度，除了成本的因素，資訊散播的延遲也是一大問題。車輛密度由路旁裝置偵測出以後，路旁裝置再往後端伺服器傳送資訊，後端伺服器整合計算所有路段的車輛密度，然後再散佈回各路段上，這過程有著數十分鐘的延遲，或多或少精準度會造成一定的誤差，若只是想知道此路段的擁塞程度則尚可使用，但若想用來做 VANET 上的應用，則即時性遠遠不足。因此，利用車輛間的直接通訊來計算車輛密度，漸漸地被許多學者所重視，使用 VANET 的無線網路架構，不只能降低基礎建設的成本其即時性也可獲得大大的改善。

若能跟對向車道的車做資訊交換，即可知道其他路段的擁塞程度[12]，進而事先做行

車路線的規劃。這種方式是交換在某時間範圍內在別條道路上的行駛時間，並跟路段長度與運算出來的值做比較，就可大略得知此路段是否正在塞車。而不選擇跟同向車做資訊交換，主要是因為同向車輛所擁有的資訊相似度較高，因此可降低了整個車間網路的封包負載程度。

在文獻[13, 14, 15]是利用跟周圍的車輛做直接溝通，進而求出路段上的車輛密度的方法。其中，文獻[13]的主要想法是，若在少部分的車輛上裝置可偵測車輛數的儀器，則偵測出的車輛數除以可偵測的範圍，將可獲得當地的車輛密度，進而散播給周圍的車子，當周圍的車子收到此資訊後與自己已有的做一平均，則可得到此路段的車密度。文獻[14]提出了一個交通資訊的系統，主要是在說明如何把資訊散播出去，以其資訊內容為基礎來做散佈的化簡，在距離很近的兩輛車，我們僅以一筆資料來代表此兩輛車，這樣就能在一個廣播週期內散播更多的資訊。文獻[15]是把一條道路切成數個區塊，區塊長度是以無線傳輸範圍來決定，計算此區塊的當地車輛密度則由離區塊中心點最近的車輛負責。透過彼此的溝通，進而求出整條路上的平均車輛密度。

無論如何，上述的方法所獲得的車輛密度準確率都不高；像是用收集當地車輛密度的平均來代表整條路的車輛密度，除非車輛分布很均勻，否則誤差將會很大。而我們所提出的 RTEP 方法將可以避免這類問題。

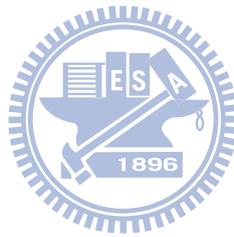


## 2.4 車輛密度與繞徑協定

如前所言，VANET 是 MANET 的一種特殊情況，主要的 VANET 繞徑，也是從 MANET 演變而來。而使用道路的車輛密度來決定封包傳遞路線的方式，現在已逐漸變成 VANET 繞徑協定的主流。以下將對 VANET 繞徑協定的演變作一介紹。

傳統的 MANET 繞徑，如 AODV[20]跟 DSR[21]，這種繞徑方式是用一連串的節點當作封包傳遞的路徑，這類的繞徑應用在 VANET 上效果會很差，原因如下：VANET 的網路範圍很大，在路徑尋找上有一定難度，另一原因是：車輛的高移動性將造成車輛間的連線極不穩定，因此就算找到了一條路徑，可能在下一秒就會失效。另外以地理位置為基礎的繞徑協定，將會有著比較穩固的連線，如 GPSR[22]繞徑方法，每個點將會利用目的位置和鄰近點的位置，來決定封包轉送的下一個跳躍點，事先並不會先把傳遞路線規劃好。但此種繞徑的缺點是容易因障礙物的阻隔，而找不到下一個跳躍點，就像是在城市裡沒有出口的巷子，或是被房子等建築物所阻隔。

以道路為基礎的繞徑方法[23, 24]改善了前述 GPSR 的問題，有了道路的電子地圖，可以事先規劃封包要沿著哪條路走，進而避免走到死胡同。這類繞徑是用最短路徑的演算法來找尋路徑，但若希望減少封包到達目的地所花的時間，在因沒考慮車輛密度的情況下，很容易選到路上車輛極少且很難找到下一個跳躍點的路徑。在文獻[25]中，作者試著用歷史(像是用月平均或季平均)的車輛密度資料來作為繞徑的選擇依據，然而這還是不足以精準地反應出目前的道路車輛密度。我們所提出的 RTEP 也正是為了解決此問題而設計。



### 3 即時車輛密度測量協定(RTEP)

此一章節將介紹我們所提出來的即時車輛密度測量協定(RTEP)。一般來說，一個好的繞徑協定是需要根據網路的特性來設計的，以 VANET 而言，車輛密度即是此類網路在設計繞徑協定時需要考慮的。RTEP 即是利用車輛直接通訊的優點，進而快速且準確的求出車輛密度，當車輛密度越準確，則 VANET 繞徑的效能也將會變的更好。

圖 3-1 是一附有車輛密度的電子地圖例子，我們將以此圖作道路導向繞徑的介紹，圖上的數字為標準化過的車輛密度，其中 S 為來源節點，D 為目的地節點，較粗的線條即是一條由比較高的車輛密度路段所組成的封包傳遞路線。我們的目的就是想提供各路段車輛密度值即時且準確的估計。

RTEP 是由兩種訊息(hello 訊息跟偵測訊息)、鄰近節點表格、偵測演算法和封包傳送機制等四部份所構成。在本論文中我們討論單一路段如何採即時分散式的方式算出路段的車流密度，在 RTEP 中，當車輛新進入一條路段，就會初始偵測演算法，首先利用 hello 訊息判斷要不要進一步傳送偵測訊息，若需要，則偵測訊息往另一端路口傳遞並加總鄰近節點表格裡的紀錄，直到偵測演算法終為止，最後再把車輛估算值放入 hello 訊息再散播回整條路告知此路段的所有車輛。以下我們將針對 RTEP 之方法細節節詳細描述之。

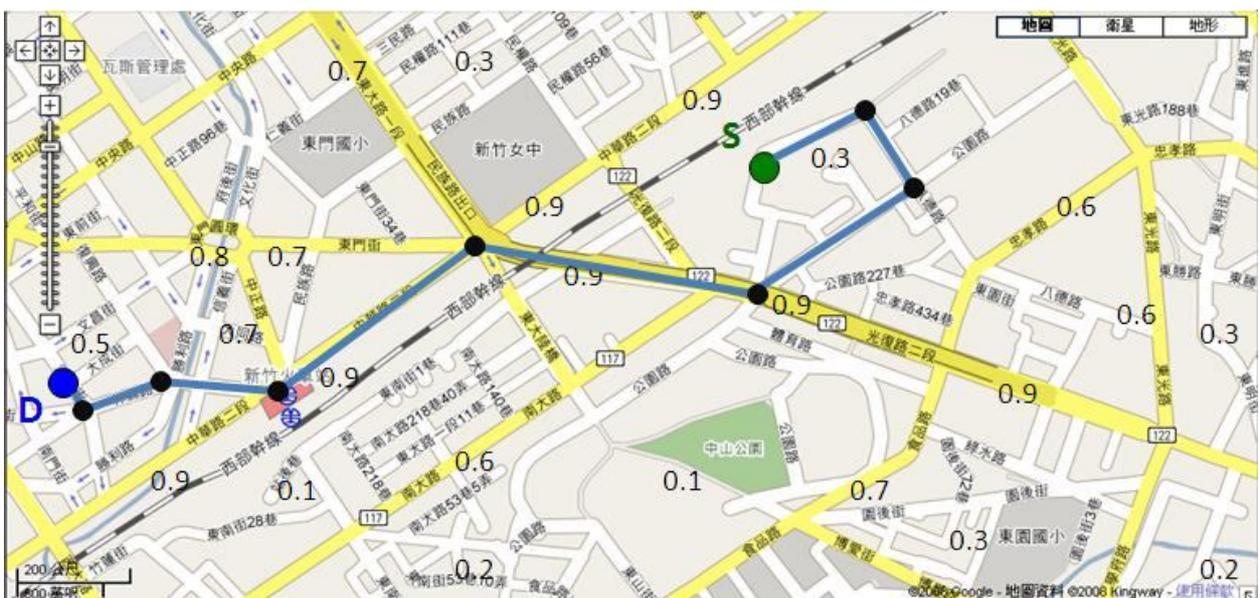


圖 3-1 有車輛密度的電子地圖

### 3.1 前提假設與車流名詞定義

在我們的偵測協定裡，存在著下列幾個基本假設。我們假定每輛車子均裝設有車上機(On-Board Unit, OBU)、道路電子地圖和全球衛星定位系統。車上機提供車輛獨立計算且與其他車輛交換資料的能力，是加入 VANET 的基本裝置。又因在執行 RTEP 時需要知道整個街道的拓撲，如哪條路接著哪條路，路口座標又是多少，所以必須要有道路電子地圖在車上。而全球衛星定位系統則是用來做為車子定位的裝置，有了定位出來的座標，封包才能依據其做傳送方向的選擇。

在後續小節裡，我們會一直使用到車輛估算值，因此我們先把此一名詞與其他常用之名詞，如：車輛密度(traffic density)與車流量(traffic flow)先作一清楚的區分。車輛密度是指每條車道上每公里的車輛數；車流量則是代表每秒通過此路口的車輛數。而我們所定義之車輛估算值則為每公里上的車輛數；亦即車輛估算值 = 車輛密度 \* 車道的數目。

### 3.2 hello 訊息與鄰近節點表格

在 VANET 環境裡，因為車子移動性高，自己的位置和鄰近點的鏈結狀況可能會隨時在變動，所以可使用 hello 訊息來隨時更新鄰居節點的位置。在我們的 RTEP 裡，每輛車要以某長度的時間週期，定期廣播一個訊息，我們稱為 hello 訊息。廣播週期( $T_0$ )越短，則越能準確告知自己目前的位址給鄰近節點，然而廣播次數增多，則會增加網路的負擔，反而會影響整個網路接收資料的效能。

我們的 hello 訊息主要有三個功用，一是要能告訴鄰近節點自己的車輛身分代號跟相同的移動資訊，如此一來，車輛才能依據此資訊來選擇下一個跳躍點的判斷；二是當偵測演算法結束後，利用 hello 訊息可以把最新的車輛估算值往回散播進而更新整條路舊有的車輛估算值；三是週期性的 hello 訊息可以用來幫助判斷需要開始傳送偵測訊息的時機。

hello 訊息的組成如圖 3-2 所示。欄位 Car\_ID 是指發送 hello 訊息那輛車的身分代號，如此收到 hello 訊息的車輛即可用此欄位來判斷是那輛車發出來的 hello 訊息。欄位 Road\_ID 是說明此車在哪條道路上行駛，此欄位也可以防止記錄到別條道路上的車輛。欄位 Position 是指發送 hello 訊息那輛車的位置，位置以經度、緯度及高度所構成，此欄位的用途主要是因為位置導向繞徑需要有位置的資訊才有辦法運作。而欄位 Velocity 則是用來判斷車輛的速率跟方向，資料在傳遞時，此資訊也是考慮下一個跳躍點的因素之一。欄位 Traffic\_Value 就是指最後由 RTEP 所計算出來的車輛估算值。欄位 Timestamp 則是車輛估

Car_ID	Road_ID	Position	Velocity	Traffic_Value	Timestamp
--------	---------	----------	----------	---------------	-----------

圖 3-2 hello 訊息欄位架構

算值產生的時間。

每輛車都會有鄰近節點表格，其作用是把收到的 hello 訊息裡的內容記錄下來，並儲存到自己的鄰近節點表格裡(如圖 3-3)，在此表格中若欄位 Invalid 為 1，即代表此一記錄所對應到的車輛正處於傳輸範圍之內。由於 VANET 的網路拓撲變化是動態的，因此鄰近節點表格必須做隨時更新的動作，一但發現有某車子已不在通訊範圍內，就必需把此筆資料做移除，則欄位 Invalid 主設定為 0。因此藉由計算鄰近節點表格中不同的資訊數目，即 1 的個數，就可以得知當地的車輛數。當收到一個不同的 hello 訊息，就會新增一筆資料，並把 hello 訊息裡的資訊記錄下來。

鄰近節點表格的組成如圖 3-3 所示。欄位 Invalid 如前所述主要是用來判斷此筆資料是不是可利用，0 代表此資料失效，而 1 則為可利用。欄位 Car\_ID 標明了鄰近的車子的身分代號，在傳遞封包的步驟裡可以指名誰可接收封包或轉傳封包。欄位 Position 是代表鄰近節點的位置，位置以經度、緯度及高度所表示，在判斷誰可收封包或誰可以轉送封包將以這個欄位為依據。欄位 Velocity 是記錄鄰近節點的速度，功能則如同 Position 欄位。欄位 Timestamp 是收到這個資料項目的時間，此欄位之功能是用來更新此筆資料的依據(以秒為單位)，欄位裡的值愈久遠，則此筆記錄的資訊誤差將愈大。因此，我們設定在當某筆記錄大於一個門檻值(T3)時，就將此筆資料予以刪除。此時代表沒收到此輛車的 hello 訊息，將判斷其非有效的臨近節點，可能此節點行駛離開了傳送範圍或車上機損壞故障之類的情況，因此，在這些情況將會把此筆資料從鄰近節點表格裡移除。當然 T3 的設定將影響鄰近點鏈結狀態的準確性。若 T3 的值越小，則更新頻率愈大，鄰近節點表格的可靠度會提升，整體的繞徑效能也可獲得改善。

Invalid	Car_ID	Position	Velocity	Timestamp
---------	--------	----------	----------	-----------

圖 3-3 鄰近節點表格架構

### 3.3 偵測訊息

偵測訊息是影響 RTEP 效能好壞的主要關鍵，此訊息最後必須總和出整個路段的車輛估算值。與其他論文的計算方法不同之處在於，我們不是利用當地車流密度來做平均，而是算出整個路上的車輛數後，再除以路段長。一般採當地車流密度去做平均的作法，在車輛分布不均勻的道路中，所算出的車輛密度準確率將大受影響，然而我們的 RTEP 卻能改善此問題。

圖 3-4 是偵測訊息的組成欄位。欄位 Destination of intersection 指的是偵測訊息要往哪個路口送以座標表示，我們可以利用自己所在的位置、車速與電子地圖作搭配，找出此目的地路口的座標值。欄位 Source 代表一開始發動此一偵測訊息的車輛。欄位 Sum of vehicles 代表截至目前為止，以累積到的車輛總數。當然，若傳到最後的計算終止點，此值將是此路段的路段車輛總數。而 hello 訊息欄位 traffic\_value 擺放的車輛估算值跟最後欄位 Sum of vehicles 裡的路段車輛總數關係如下：

$$\text{車輛估算值} = \text{路段車輛總數} / \text{路段長} \quad (\text{輛/公里})$$

欄位 Timestamp 指的是此路段車輛總數產生的時間。

偵測訊息被發起傳送後，就必須計算當地的車輛數，並累加於欄位 Sum of vehicles 裡，直到傳送到最後一個計算終止的車輛，它最後會得到整段路段上的車輛，再把此值除以路段長並擺放入 hello 訊息的欄位 Traffic\_Value 裡。累加的方法如下式：

$$\begin{aligned} & \text{傳送出去當時的 Sum of vehicles} \\ &= \text{接收到的 Sum of vehicles} + \text{當地車輛數(local vehicles)} + 1(\text{自己}) \end{aligned}$$

為了不讓每個跳躍的當地車輛數有嚴重出現重複計算的情形，我們把當地車輛數定義如下：

$$\text{當地車輛數} = \text{兩個跳躍點之間的車輛數}$$

如圖 3-5，填滿的節點為偵測訊息的傳送路徑，而當地車輛數就是指兩個實心點間的車輛數，例如：B 點的當地車輛數為跟 C 點之間的車輛數，其值等於 2。

Destination of intersection	Source	Sum of vehicles	Timestamp
-----------------------------	--------	-----------------	-----------

圖 3-4 偵測訊息欄位架構

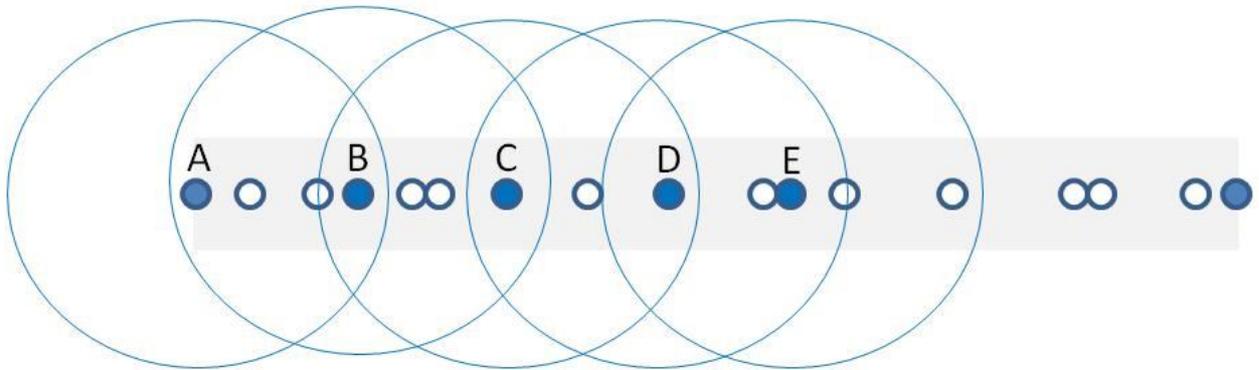


圖 3-5 當地車輛算法示意圖

### 3.4 偵測演算法



當車輛一進入新的路段，我們的偵測演算法即會開始運作。此演算法主要由兩個判斷與四個動作所組成，如圖 3-6 所示。以下將就此些動作即判斷作一細部描述。

**動作一：**當車輛進入新路段後，需要等待一段時間( $T_1$ )來接收 hello 訊息，因為我們是利用 hello 訊息裡的資訊來判斷要不要傳送偵測訊息，因為若一進入新路段就做是否傳送偵測訊息的判斷，此時根本都還來不及接收到四周 hello 訊息，這樣將導致 hello 訊息因碰撞而接收不到。所以我們設定在進入新路段的車輛必須先等待  $T_1$  秒。

**判斷一：**在經過了  $T_1$  秒後，需不需要傳送偵測訊息。則用下列三點依據來作為判斷依據：

- 1) 這條路是否從沒計算過車輛估計值，若都沒算過即開始送偵測訊息。
- 2) 先前的偵測訊息是不是傳送失敗了，亦即離現在的時間若已大於  $T_2$  秒，就開始傳送偵測訊息。

3) 現有的車輛估計值是不是已過期了，我們可用現在時間減去欄位 timestamp 裡的時間是否大於  $T_3$  秒，來作為是不是過期的判斷。

若上述判斷為是，則將往下進行動作二，否則將結束偵測演算法。雖然我們目的是要

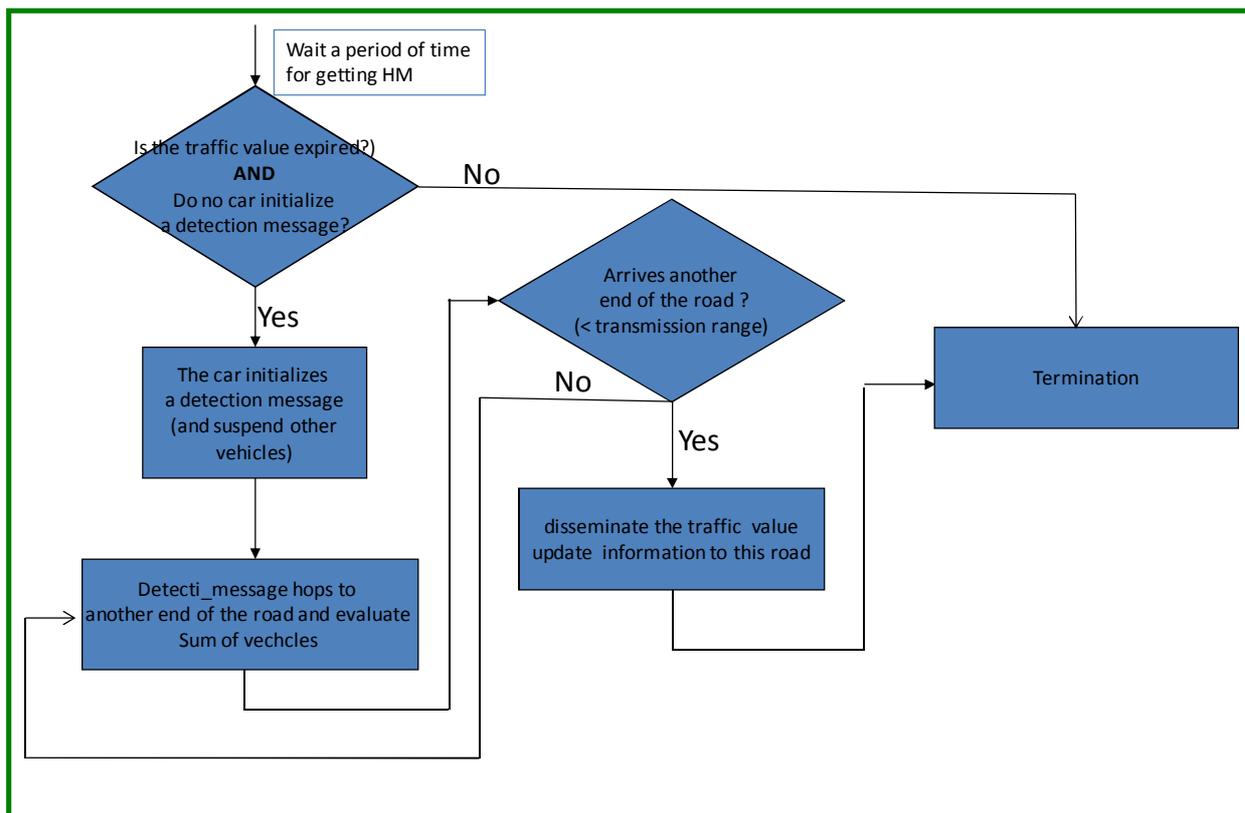


圖 3-6 偵測演算法流程圖

算出路段的車輛密度，但並非每一輛車都需要發送偵測訊息，因為這樣只會增加整個網路不必要的負擔，所以判斷一是十分重要且必需的。

**動作二：**在開始初始偵測訊息時，需要把前述所等待的  $T_1$  秒裡有新進的車輛加入計算(圖 3-7 可看出因等待  $T_1$  秒而新進來了兩輛車)，因此在傳送偵測訊息給下一個跳躍點前，除了原本的定義外，必需把修正車輛也加入欄位 Sum of vehicles 之中，如此即可減少因等待而造成的誤差。

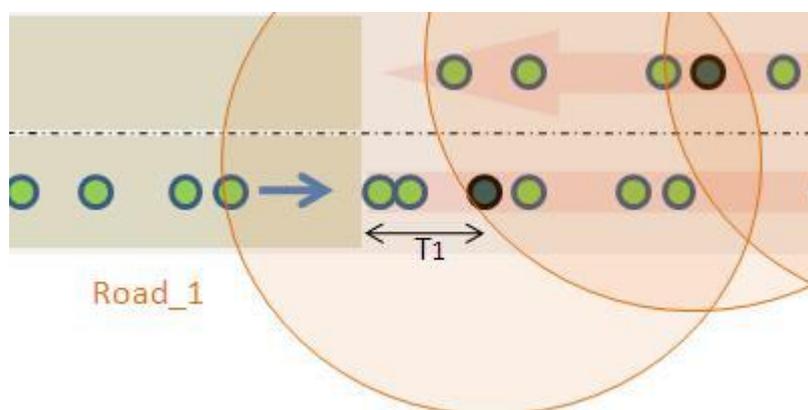


圖 3-7 偵測演算法初始步驟中車輛數修正示意圖

**動作三：**往目的路口傳遞偵測訊息，並在每個中間跳躍點累計當地車輛數至欄位 Sum of vehicles，至於如何選擇下一個跳躍點的方法，將在第 3.5 節作詳細介紹。

**判斷二：**在作傳遞偵測訊息的過程裡，是否車輛本身位置離目的路口已經小於傳輸半徑，若答案為真，則結束繼續傳遞偵測訊息。否則，重複執行動作三。

**動作四：**算出車輛估算值後，利用 hello 訊息來散播新的 traffic\_value 到這條路段上，讓所有車輛更新估算值。

### 3.5 封包傳送機制

此封包傳送機制是用來說明封包如何被傳遞到目的路口，在我們的 RTEP 裡，是以位置導向傳遞作為我們主要的基礎方法，另外再加上安全傳送距離的判斷機制。其中位置導向傳遞需要知道鄰近點的資訊，而因我們有使用 hello 訊息傳播的方式，每輛車均會廣播 hello 訊息給鄰近的車輛讓他們知道傳送車輛的身分代號、位置、速度等資訊，如此一來，每一個傳送車輛即能知道鄰近車輛的位置。

而選擇下一個跳躍點的方式就是選鄰近節點表格裡最接近目的路口的點，所以封包也有可能被傳遞到對向車道去。如圖 3-8 所示，箭頭為封包的傳遞路徑。由於 RTEP 是一種分散式運作的方法，因此，在資料傳至每個當地跳躍點車輛時，才會決定出下一個跳躍點。

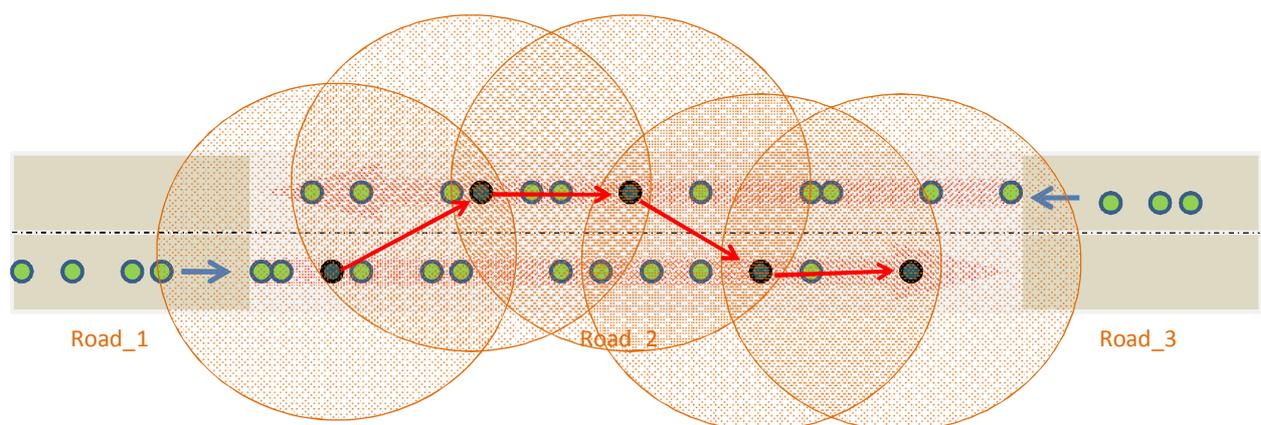


圖 3-8 封包傳遞路徑圖

由於 VANET 的高移動性、hello 訊息週期性以及鄰近節點表格更新的頻率等，都會影響鄰近節點位置資訊的準確率，若選太接近傳輸邊界的車輛作為轉傳點，這雖然可減少跳躍數及時間延遲，但卻多了鏈結不穩定性的反效果，導致在決定下一個跳躍點後，卻很快鏈結斷離而找不到此一節點的情況發生，封包將會被丟棄，偵測訊息的傳遞率更會因此而下降。因此，我們提出了一安全傳送距離的判斷機制來作為尋找跳躍點的依據，以改進前述遺失之發生可能。

在此安全範圍的定義如下：

安全傳輸距離 = 最大傳輸半徑 - (車輛最大速率\*修正的時間)

修正的時間 = 現在的時間 - 鄰近節點表格裡欄位 Timestamp 的時間

因為我們都是用鄰近節點表格來作傳遞機制，但裡面的位置資訊並非就是現在的分布狀況，所以需要作修正。按上列安全傳輸範圍的定義是以最大的傳輸半徑減去一個危險區塊長度，而危險區塊長度的估計則是用車輛最大速度乘以修正的時間來估算。

待安全範圍被估算出來後，我們 RTEP 則會將超過安全距離的車輛排除使之不會成為下一個跳躍點，如圖 3-9 所示，原本 A 車的下一個跳躍點是應 C 車，但由於 C 車在安全傳輸範圍以外，故改選 B 車。

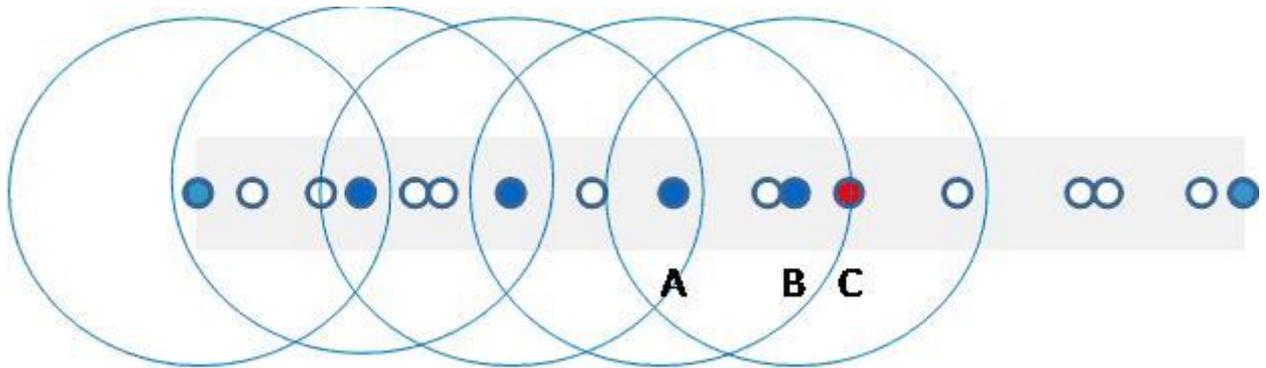


圖 3-9 安全傳輸範圍示意圖

# 4 模擬結果探討

在此我們將展示 RTEP 的效能，藉由車輛密度、車輛速度跟資料流大小的調整來作為模擬探討。主要本章架構如下：在 4.1 節我們將介紹實驗所使用的模擬平台及所撰寫的程式檔描述；4.2 節將描述模擬參數的相關設定；模擬結果將於 4.3 節中呈現。

## 4.1 模擬環境

本篇論文的模擬平台是用 NS2 網路模擬器[26]，使用版本為 2.31。在媒介存取控制層(Media Access Control Layer, MAC Layer)，是使用預設值 802.11 的分散式協調功能(Distributed Coordination Function, DCF)。

RTEP 是一個在網路層上操作的協定，我們採用 C++語言來進行撰寫，並安裝在 NS2 上。主要由五個檔案所構成，作用分別介紹於下：

### 1) RTEP.cc

實作所有的計時功能、RTEP 的偵測功能，還有對情境敘述檔案的溝通都寫在這一部分。情境敘述的程式碼，是使用工具命令語言(Tool Command Language;TCL)來撰寫。

### 2) RTEP.h

定義在 RTEP.cc 裡會用到的變數及功能。

### 3) Neighbor\_Table.cc

實作整個鄰近節點表格，如:資料新增及更新資料等動作。

### 4) Neighbor\_Table.h

定義在 Neighbor\_Table.cc 裡會用到的變數及功能。

### 5) RTEP\_pkt.h

定義我們的 RTEP 會用到的封包格式，即 hello 訊息與偵測訊息。

## 4.2 模擬參數

我們是模擬一條雙向道路來估算實質的效果，如圖 4-1 所示，模擬的車輛會從此路的兩側路口駛入，並往另一端路口駛去，車輛不會停下或倒退行駛。車輛進入路段的間隔採指數分配(即所謂 exponential interarrival time)。車輛的速度以正常範圍 8 公尺/秒到 20 公尺/秒之間，換算成時速，大致為每小時 30~70 公里。路口座標指的是路段兩端的正中間，車輛會往此座標作封包傳遞的方向。另外緊鄰著路口中心還會擺置兩台固定的無線儀器(如圖 4-1 的 A 和 B 兩節點)來負責產生跟接收資料流，在傳輸層我們採用 UDP 來傳送大小為 1000 位元組的封包，並使用固定位元速率(Constant Bit Rate，簡稱為 CBR)來傳遞。詳細參數列於表 4-1 中。



圖 4-1 模擬道路圖

表 4-1 環境參數表

實驗環境	媒介存取控制層	802.11 DCF
	傳輸半徑	250 公尺
	路段長	1 公里
	道路型態	兩線道的雙向道路
RTEP 參數設定	hello 訊息週期( $T_0$ )	0.5 秒
	偵測演算法的等待時間( $T_1$ )	2 秒
	判斷偵測訊息是否成功的等待時間( $T_2$ )	3 秒
	車輛估算值的有效時間( $T_3$ )	5 秒
模擬參數設定	車輛進入間隔	exponential distribution $\lambda=1, 5, 10, 15, 20$ 秒
	車輛速度	8~20 公尺/秒
	資料流(CBR)	0, 1, 2, 3, 4, 5 千位元/秒

## 4.3 模擬結果

在我們的模擬數據，記錄了每個偵測訊息的相關結果，如表 4-2 所示。主要想計算每筆數據之平均準確率(%)。當然準確率愈高代表我們 RTEP 的效能越好，而且估算出的車輛數離實際值差距很小。準確率的定義如下：

$$\text{準確率(\%)} = 100 (\%) - \text{誤差率(\%)}$$

$$\text{誤差率(\%)} = (|\text{實際車輛} - \text{偵測車輛}| / \text{實際車輛}) * 100$$

而準確率要每次發送偵測訊息時，就計算一次；假如偵測訊息沒有估算成功，將以過去中最近算出的車輛估計值作代替。

表 4-2 模擬數據範例

時間(秒)	發送點	演算法的終點	偵測車輛(輛)	實際車輛(輛)	DM的平均延遲時間(秒)	平均準確率(%)	HM的平均延遲時間(秒)
151.70	648	171	144	146	0.2440 秒	98.77	1.853 秒
157.11	656	572	152	150			
162.69	666	570	156	155			
.	.	.	.	.			
.	.	.	.	.			
.	.	.	.	.			

表 4-2 中各欄位代表之意義說明如下：

時間(秒)：偵測訊息(DM)所產生的時間

發送點：產生 DM 的節點

演算法的終點：算出車輛估算值的節點

偵測車輛：估算出的路段車輛數

實際車輛：真實的路段車輛數

DM 的平均延遲時間：DM 從產生到達終點所花的時間

平均準確率(%)：把每筆 DM 的準確率加總並除以數據個數

HM 的平均延遲時間：hello 訊息(HM)散播車輛估算值至整條路段所花的時間

在第一個模擬實驗中，我們希望求得車輛密度(或簡稱車密度)的大小對於準確率的影響程度。我們用 5 個  $\lambda$  值，各為 1、5、10、15、20，來表示不同的車密度。其換算關係如下：

$$\begin{aligned} \text{平均車密度} &= \frac{\text{車輛數}}{\text{路段長}} \\ &= \frac{2 \times \text{路段長}}{\lambda \times \text{平均車速}} \\ &= \frac{2}{\lambda \times \text{平均車速}} \end{aligned}$$

我們從圖 4-2 可以看出就算車密度改變，但在足夠車輛的情況下，準確率還是能維持在 90% 以上；即使車輛變稀疏，像是每一公里上平均有 7.15 輛車，準確率還是能有 75% 左右。這結果說明了，當 DM 傳遞率下降時(圖 4-3 所示)，車輛估算值將無法定時更新，只能用歷史估算出的值作代替，所以準確率會下降。DM 的傳遞率，亦稱到達率，定意為偵測訊息沒有遺失的機率。

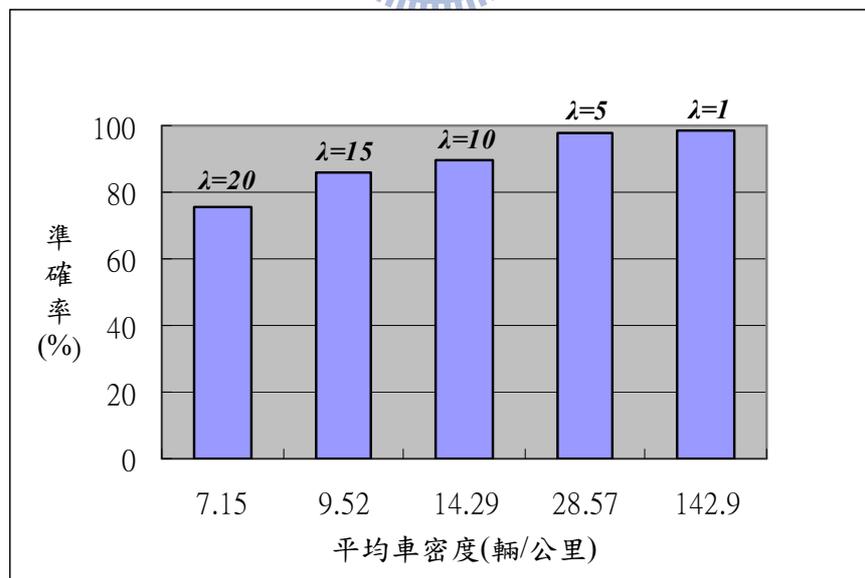


圖 4-2 平均車密度與準確率關係圖

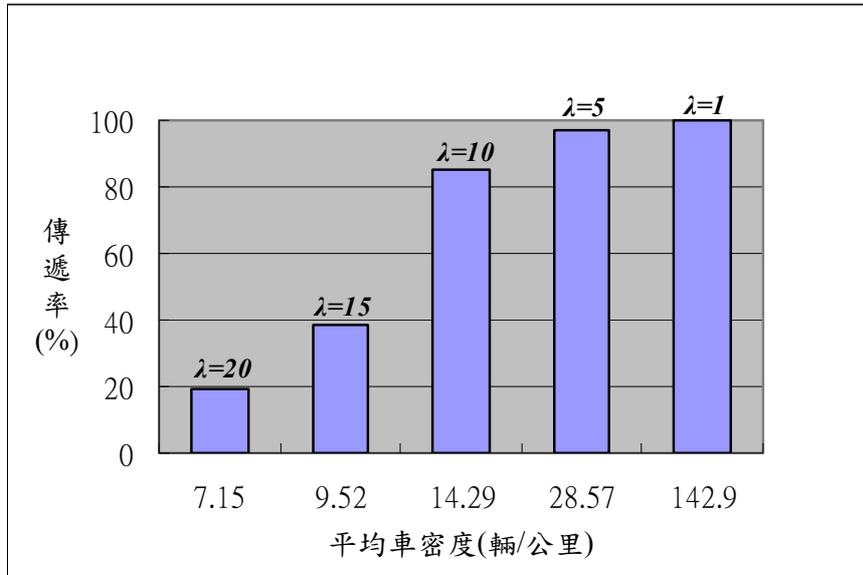


圖 4-3 平均車密度與傳遞率關係圖

在第二個模擬實驗，我們將探討不同平均車速對於準確率的影響。以 5 種平均車速作模擬，從圖 4-4 可以看出，當平均車速變快時，準確率會略為下降，但在平均時速為 60 ~70 公里時，準確率還是能有 97.5%以上。RTEP 是用鄰近節點表格來判斷周圍的節點，所以當平均車速上升，鄰近節點表格所記錄的節點誤差也會變大，導致準確率下滑。

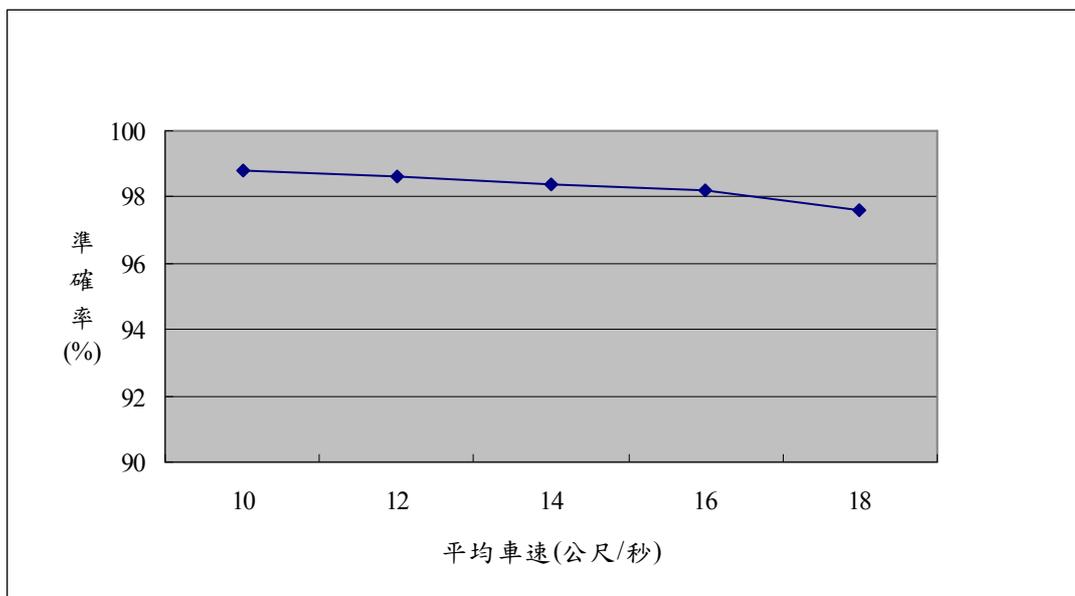


圖 4-4 平均車速與準確率關係圖

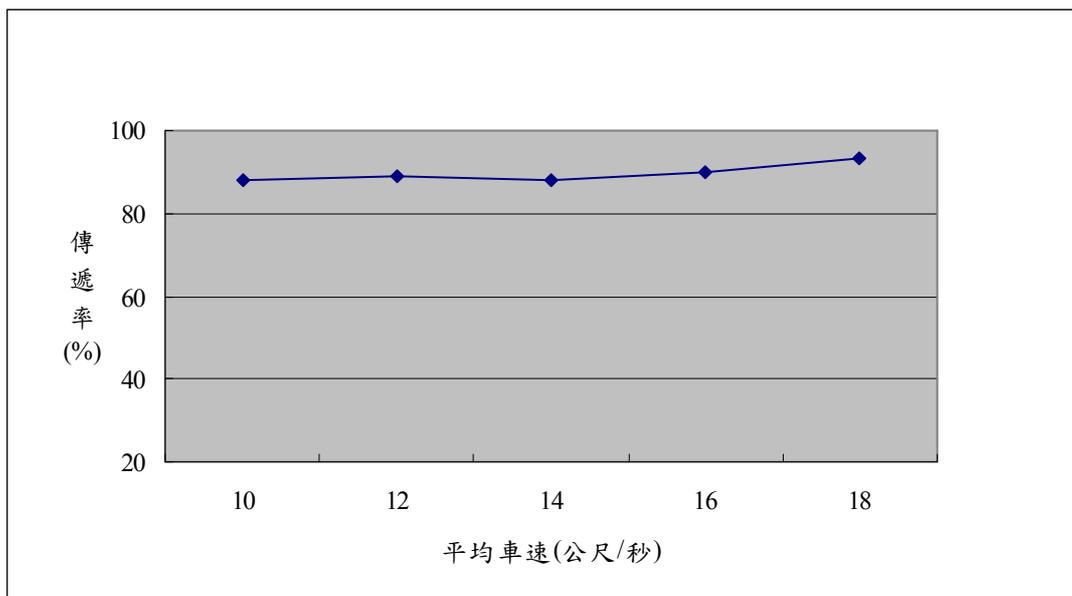


圖 4-5 平均車速與傳遞率關係圖

從圖 4-5 可以發現平均車速不是影響 DM 傳遞率的主要因素。

表 4-3 是車輛相對速度對準確率的影響結果，我們可看出準確率都幾乎在 97% 左右沒有太大變化。相對速度越大代表車間鏈結愈容易斷掉，我們的 RTEP 因有安全範圍的判斷，所以從表 4-3 的 DM 到達率來看，RTEP 是可行的，並不會有常常算不出車輛密度的情況發生，幾乎每 100 個偵測訊息，都能成功傳送 95 個以上。

表 4-3 相對車速改變的一些數據結果

車速 (公尺/秒)	同方向 相對速度範圍 (公尺/秒)	相對速度範圍 (公尺/秒)	DM 到達率%	DM的 平均延遲時間(秒)	平均 準確率 (%)	HM的 平均延遲時間(秒)
13~15	0~2	0~30	97.70	0.2807 秒	97.34	1.996 秒
10~18	0~8	0~36	95.62	0.2330 秒	97.09	1.990 秒
8~20	0~12	0~40	97.06	0.2846 秒	97.82	2.125 秒

最後，在第三個模擬實驗中，我們增加資料流到路段上，並觀察其準確率之變化。在模擬道路圖(圖 4-1)中，我們加入了 CBR 資料流，並從 A 點傳送到 B 點，再探討六種不同的 CBR 對準確率的影響。模擬結果如圖 4-6 所示，我們可以看出改變了資料流後，我們的 RTEP 效能還是不錯的，依然有 95%以上的準確率。但是由於資料流變大後，封包彼此碰撞機率增加，DM 傳遞率會下降(如圖 4-7 所示)，導致準確率也會下降。

由以上三種模擬實驗，我們總結出 DM 傳遞率與平均車速對準確率有比較大的影響。DM 傳遞率下降，估算不完整段道路的車輛，只能用上次估算出的值作代替，所以準確率下降；而車速變快，準確率也將會些微下降；但以平均車密度來說，對其影響是趨近於零的。

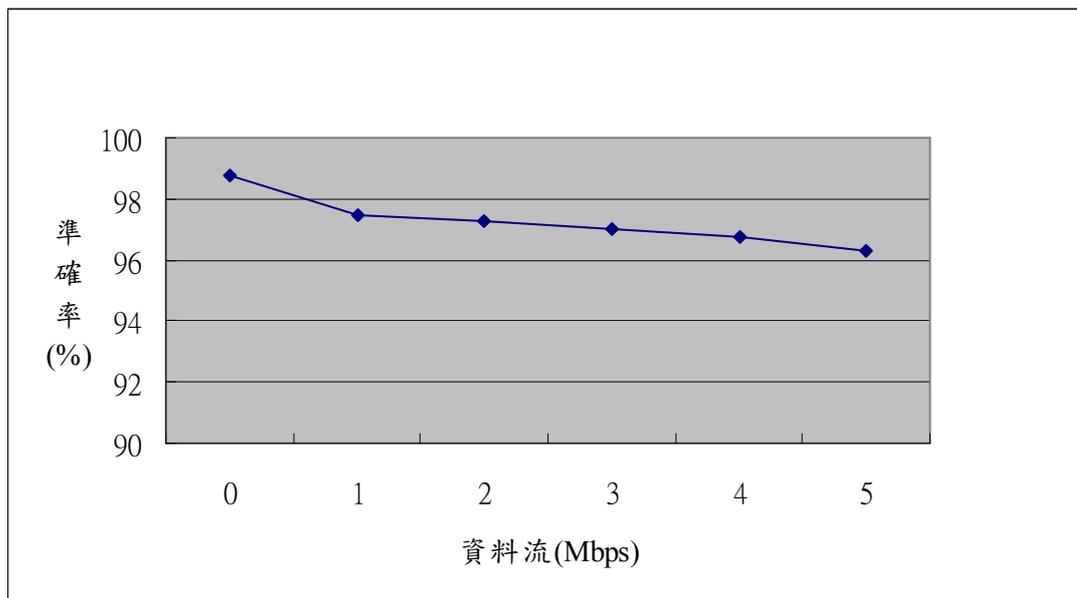


圖 4-6 資料流與準確率關係圖

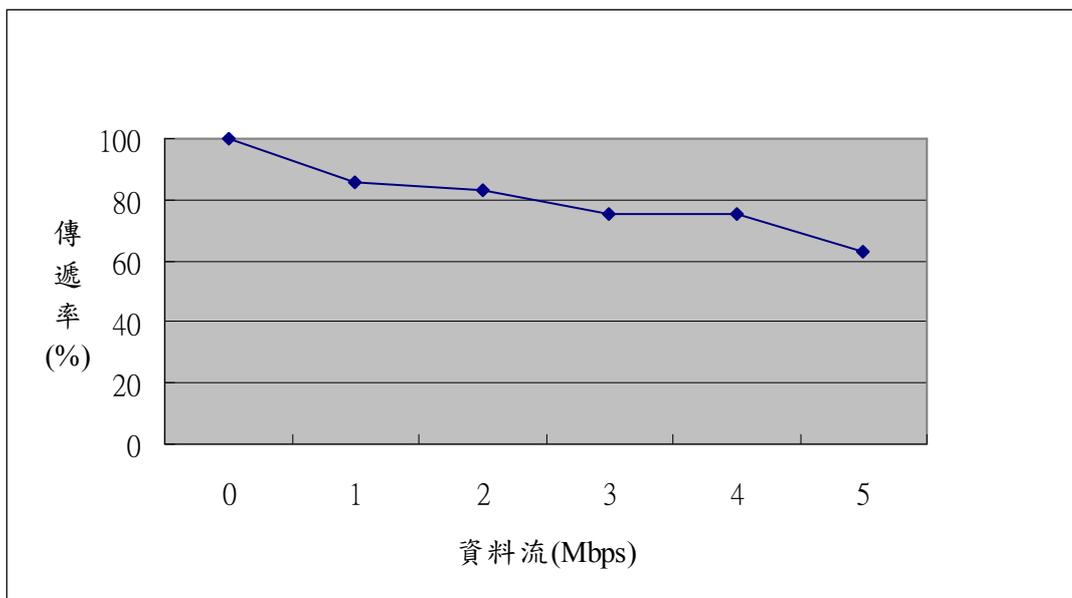


圖 4-7 資料流與傳遞率關係圖

除了模擬一般道路的長度外，另外還模擬了道路長為 5 公里與 10 公里的情形。從圖 4-8 的結果可以看出，雖然道路長度增長，網路斷離的發生率變大，導致 DM 傳遞率下降，但是還不至於造成準確率太大的影響。

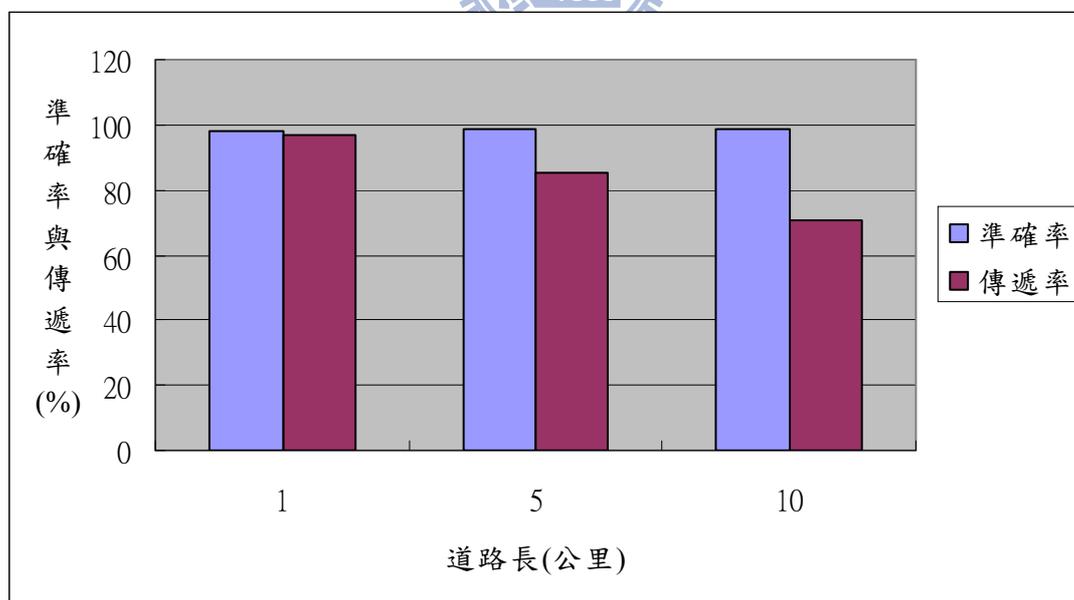


圖 4-8 道路長對於 RTEP 的影響

## 5 結論

RTEP 的準確率跟即時性能帶來不少好處，除了可應用在 VANET 繞徑外，還能用在動態調整 VANET 之頻道間距上，當車輛密度變大時，RTEP 可以適時提供資訊調整傳送資料的頻道時間，減少有資料未傳完的情形發生；RTEP 更可應用在安全輔助上，例如用估算出的車輛密度來決定緊急封包的散播次數或每個接收者重播的等待時間。

從模擬結果可以看出，在不同車輛密度、車輛速度跟資料流的情況下，我們的 RTEP 皆有著不錯的效能，我們所模擬估算出的數值，在非車輛極稀疏的情況下，至少都有著 80~95% 以上的準確率；而使用安全範圍的傳送機制，偵測訊息的傳遞率就算在相對速度變大的路況中，也能維持一定的高傳遞率。

