

使用歷史與即時市區交通資訊之高效率車輛網路 單點路由

張英超
國立彰化範大學 資工系
副教授
icchang@cc.ncue.edu.tw

王元芬
國立彰化範大學 資工系
碩士生
m96612010@mail.ncue.edu.tw

摘要

在本文中，我們建立一個市區車輛網路 (Vehicular Ad Hoc Network, VANET) 的架構，利用電子地圖 (e-map) 及歷史交通流量統計資訊，配合路況中心所得知施工或交通管制即時資訊，比較車輛平均間距是否小於等於車輛網路無線傳輸範圍來篩選出所有可進行逐次跳躍 (hop-by-hop) 傳遞的路段，建立一個路口圖 (Intersection Graph, IG)。接著算出在此 IG 路口圖上，由來源車輛到目的車輛的最小路口成本路由。進而以路口來源路由 (source routing) 的概念，將封包以貪婪法傳遞到達目的地車輛，降低傳輸延遲時間。封包傳輸過程中，即時的交通狀況可能會造成由 IG 圖計算出來的路由中斷，無法繼續傳輸，導致封包遺失。本論文結合歷史統計與即時兩種交通流量資訊，使其發揮互補的效益，改善傳統車輛網路單點路由封包傳輸成功率與延遲。

關鍵詞：車輛隨意網路、車間傳輸、即時資訊、交通統計資料、單點路由協定。

Abstract

In this paper, we propose a city-based Vehicular Ad Hoc Network (VANET) architecture. Using the electronic MAP (e-map), historical traffic statistics and real-time road construction/control information, we create an intersection graph (IG) that includes all road segments, whose average inter-vehicle distances are not larger than the transmission range of the VANET wireless signal. We will further calculate a least-cost routing path consisting of successive intersections on this IG. Then the source vehicle will leverage the source routing approach to forward unicast packets

intersection by intersection to the destination vehicle with reduced transmission latencies. On the other hand, real-time traffic situations like the car accident in the near region or the instantaneous vehicle distribution may cause gaps in the calculated least-cost routing path and in turn result in packet losses. In this paper, we will use real-time traffic information to complement historical one in our proposed unicast routing protocol to improve packet delivery ratio and delay.

Keywords: VANET, IVC (Inter-Vehicle Communication), real-time information, traffic statistics, unicast routing protocol.

1. 簡介

車輛隨意網路的開端是為了提供智慧型運輸系統 (Intelligent Transportation Systems, ITS) 的車間 (inter-vehicles) 及 RSU 通訊，使用基於公用安全相關的網路應用。但近年來車輛網路的應用迅速向其他非安全性服務發展，而各種車用網路裝備也被預定為不久後的未來車輛的標準配備，車輛網路儼然成為各先進國家的既定發展趨勢。以車輛網路的應用來看，網路單點傳輸是必備的技術。目前多數的有線及無線網路服務，均使用網路單點傳輸，無論是網頁流覽、線上購物，或是多媒體傳輸等。因此，研究網路單點傳輸在車輛網路上的適用方法是極其重要的，車輛隨意網路上單點路由選擇協定的經濟潛力也因此浮現。而車輛網路有許多不同於其他無線網路的特性，例如車輛的移動是在既定的道路上，形成特殊的節點分佈；移動速度快，且有明顯相異的移動方向，造成網路拓撲變化劇烈；另外還可能因不同地理區域造成道路交會、車道 (lane) 數的不同，以及道路上包括紅綠燈在內的交通控制機制，影響網路的傳輸。所以在路由協定的設計上，也必須調整成適合車輛網路的環境。對於車輛網路

而言，路由的關鍵並不在於車輛的絕對密度或是單純的車速，而是能掌握可傳輸路由上的車輛分佈，才可能選出最佳的傳輸路由。

要得知道路上車輛的分佈，需要蒐集與分析各項對車輛隨意網路的路由選擇協定有幫助的資訊。我們將其分為靜態的資料與即時的車輛與交通資訊。靜態資訊包括電子地圖、交通流量歷史統計資料(traffic statistics)及紅綠燈分佈，通常不會有經常性的變動。而靜態資訊通常有以下的優點：

- 是一種整體性的資料(global information)：可以提供較大範圍與長期性的資訊，做為整體傳送資料時的考量，並提供車輛在傳輸中斷時選擇處理方式的依據。
- 更新少：不需隨時更新，網路成本消耗低。
- 可事先處理：包括靜態資訊下載與路由計算。

利用現有的道路及交通流量統計資訊，是目前消耗網路資源最小，而最有效獲得整體車輛網路車間距離資訊的方法。在車輛網路上使用歷史統計資訊最具代表性的即為 VADD[15]以及 DIR[3]。然而，關於車流量的資訊卻有不符合傳輸即時狀況的疑慮。

即時資訊包括即時車流、交通事故、道路工程與交通管制等，最能反映車輛網路的最真實現況。取得方式有車間互相交換(V2V)、中央資料庫、RSU 蒐集與三者交互的做法。在路由選擇上即時資訊的必要性與特色如下：

- 對於瞬間的路況準確性高。
- 變動快。
- 取得成本較高。
- 不同區域的資訊需要的使用者不同。

在車輛網路上大多数的路由協定仍採取全部利用即時資訊，例如 OPERA [5]、GPSR_L[14]、GPCR[11]以及 RBVT[13]等。然而，利用即時資訊的最大挑戰在於收集和維護時的網路耗費(cost)，過大的耗費可能引起無線網路的壅塞甚至癱瘓。另外的問題在於如何保持即時資訊的新鮮度，若是為了維護新鮮度而不斷更新，又可能造成更多的耗費。第三項問題在於取得範圍，取得範圍越大，所需的成本可能倍增，而且更不易確保資訊新鮮；取得的範圍太小，又可能不足以幫助路由，形成其他方面的網路資源消耗。

從現存的三類路由協定看來，不論是以位置為基礎的路由協定(Position-Based Routing Protocol)、運用即時路況資訊與位置路口觀念

的路由協定、運用歷史交通統計資訊與位置路口觀念的路由協定等，大多數僅使用統計或即時資訊，使得它們只能適用在部份車輛網路的情況下。本文的目標是整合車輛隨意網路上無線傳輸最重要的數種靜態與即時資訊，一方面利用的電子地圖(e-map)及整體道路歷史交通流量統計資訊，以我們先前研究的路口圖(Intersection Graph, IG)設計[1]中的方法來比較車輛間距是否小於或等於無線訊號傳輸範圍來篩選出可進行逐次跳躍傳遞的路段，建立一個路口圖，進而利用最小成本路由演算法算出封包在此路口圖上傳遞所經過的最小成本路由。另一方面利用路況系統中心收集施工與交通管制訊息，同時運用 RSU 蒐羅附近區域內車輛傳送的問候封包(Hello message)，統計其所攜帶的即時交通流量資訊，提供給車輛有效地修正上述最佳路由的規劃，以適應即時路況，做出正確的路由調整。相較於基於貪婪法(Greedy-based)的即時路由演算法與上述幾種方法，我們的方法更具體整體規劃，兼顧即時的路況。

本篇論文尚有四段。下一段將討論在車輛隨意網路上不同觀念的路由協定的作法。第三段則介紹本文提出的方法，包括如何取得靜態的交通流量統計資訊與即時路況資訊，以及如何根據這兩種路況資訊，設計出高效率的單點路由協定。第四段顯示我們的方法和其他方法在模擬上的比較結果。最後是結論。

2. 相關研究

多年來，在車輛隨意網路上已發展出許多路由方法，這些方法來自不同的觀念與使用的資訊，例如來自傳統隨意網路的方法，由市區環境產生出的路口觀念方法等，而有些使用即時車輛位置或交通統計資料，以下我們將其區分為三種。

單純以位置為基礎的路由協定(Position-Based Routing Protocol)是由無線感測網路的路由選擇方式發展而來的方法，通常會以調整過的貪婪法和分區路由選擇(Face Routing)兩種技術完成。使用這類方法的優點是，所有參與傳輸的節點，都只找出所要轉遞的下一個跳躍(next hop)節點，因此可以避免拓撲改變的影響，然而必須伴隨頻繁的問候訊息的發送與鄰居資訊的更新，增加額外耗費。GPSR_L 是修改自 GPSR[6]，增加問候訊息的欄位，企圖以預測的方式減少訊息的發送，同時保持鄰居資料的新鮮，卻仍會因為貪婪法是

以直線距離判斷是否接近目的地節點，使得路由的中間節點可能陷入區域最大值的情況，而必須進入分區路由選擇模式。此時會造成較大的網路資源消耗，同時可能引起不必要的路由迴路。Motion Vector (MOVE) [7]將車輛的移動性及其相對速度考慮進去，利用三角函數去計算出節點本身之速度向量與目的地的夾角，夾角越小代表這個節點越往目的地靠近。比較節點本身與目的地的夾角及其他鄰居節點與目的地的夾角，封包會傳往夾角較小的鄰居節點，若封包原本就在夾角較小的節點則不將封包傳送出去，選擇儲存在原本的節點與原本的節點一起移動。如此作法便可使封包越來越接近目的地。但這樣的作法在節點移動時可能就必須不斷地計算角度，而且若是來源端與目的端皆移動的情況下可能會無法得知對方的地理位置資訊而無法計算，導致在此種情況失去其準確性。

運用即時路況資訊與位置路口觀念的路由協定一開始的研究並無使用電子地圖，而著重在測得路由轉彎時位於道路的交叉路口的車輛(position-anchor)，例如 GPCR 和 CAR[12]的方法。這些方法都需要額外的計算來確知路口的車輛，同時要在有效期內維護路口得到的路由資訊，將消耗大量的網路資源，這些都是使用即時資訊所需付出的代價。此外 GPCR 必須依靠路口節點去選擇該往哪個路段前進，若是錯過路口節點或路口節點不存在時可能會導致路由有交錯的情形，也就是變成非平面化的地圖，此時可能就會有迴圈產生；接下來的 OPERA 和 CBRP[10]則是認為路口交通資訊對都市格狀車輛網路的重要性，然而 OPERA 必須假設在每一路口都設置固定節點，雖然對車輛資訊的獲得有幫助，卻也可能造成路口附近的傳輸瓶頸而產生新的問題；CBRP 強調路口交通資訊應為即時性的(real-time)，卻沒有提出如何得到這些即時的資訊，而這些資訊是車輛網路上極可能消耗大量資源才能得到的。RBVT 則是利用泛濫法建立與維護來源路由(source routing)的路由，在此路由上使用貪婪法傳輸無線訊息。這種方法雖然可以輕易避免路由迴路的問題，使用泛濫法卻需要付出相當大的成本，而且沒有任何其他資訊(例如交通統計資料)的輔助，仍無法保證即時資訊可維持新鮮度的時間。GeoCross[8]改善 GPCR 的路由方法，在封包的格式中加入額外的欄位，分別存放走過的節點、路段、路口、設定為不可走的路由及來源端送出封包經過的第一個路段等。

GeoCross 利用資料封包中的新增欄位去偵測是否路由產生了迴圈，且移除有交錯的路段去解決迴圈的問題，因此解決了 GPCR 的缺點；GPSRJ+[9]修改了控制訊息加上自己的位置及鄰居所在的路段，使得鄰居接收到控制訊息後可以知道 1-hop 鄰居所在的路段，再經由此資訊去選擇該往哪個路段去傳送封包，可以解決 GPSR 遇到 local maximal 必須切換至 Perimeter mode 導致 hop 數變多的問題；也可解決 GPCR 傳送封包時必須傳到路口再選擇，某些情況下可以直接略過路口的節點，減少 hop 數與延遲時間。但若是遇到訊號範圍包含多個路口時，GPSRJ+的方法就無法有效解決，並且在針對區域最大值問題的解決，也顯得片面而效果有限。

運用歷史交通統計資訊與位置路口觀念的路由協定中，VADD 的作者將適用於市區的傳輸過程分為三個模式，相互交替直到封包傳送至目的地。在路口模式(Intersection Mode)時，利用交通統計資料設計出機率延遲時間模型，來判斷接下來該將封包轉遞給那一個方向道路的車輛，若不在路口模式又遇到無法向前轉遞時則以車速攜帶封包在車輛本身的記憶體中(carry-and-forward)。VADD 雖然完全不需付出泛濫法的成本，也不需下載、更新資料，卻可能產生訊息傳輸途中迴路(loop)或繞遠路的情況，反而增加傳輸延遲時間。DIR 修改 CAR 的路由錨點成為對角線錨點路由，並以加上紅綠燈變換時間修改 VADD 利用交通統計資料的延遲模型(delay model)，計算每段路延遲成本的期望值。同時將其原本是單點(Unicast)的傳播方式擴展為支援多路由的路由(Multipath)，以對角線錨點分離路由的方法，使資料在每一段的傳輸過程中，隨時都可以動態的調整傳輸路由，避免傳輸延遲過高或是路由中斷。然而方法中卻未考慮來源車和目的地車移動對所選擇的路由可能造成的影響。這類方法一旦即時路況和統計資料不同步，沒有任何應變措施，統計資料無法更新，也缺乏考慮其他對路由具影響的即時路況。

縱觀以上的路由選擇方法，都缺乏整體運用對路由有益的資訊來選擇真正穩定的傳輸路由，造成封包傳輸的遺失與成本的增加。因此，本計畫將討論如何結合歷史交通與即時路況兩種資訊的優點，使其發揮互補的效益，提升單點路由協定的效能。

3. 整合歷史交通統計與即時資訊的單點路口圖路由架構

本文的目標在提出一個能整合可利用在車輛網路路由的靜態歷史交通資訊與即時路況資訊的架構(如圖 1)，利用其設計出適合應用在車輛網路的路由協定。以下內容包括此架構的環境及元件，在此架構下各元件如何取得與處理交通統計資料和即時路況訊息，接著以這些資訊建立高效率的路口圖路由，以及因應資料封包傳輸時即時路線中斷的情形。

3.1 整合靜態與即時資訊的架構

我們所提出的架構包含三個主要元件，路況系統中心、RSU 以及車輛網路上的有效(active)車輛。以下分別敘述各元件的環境假設與在架構中的功能。

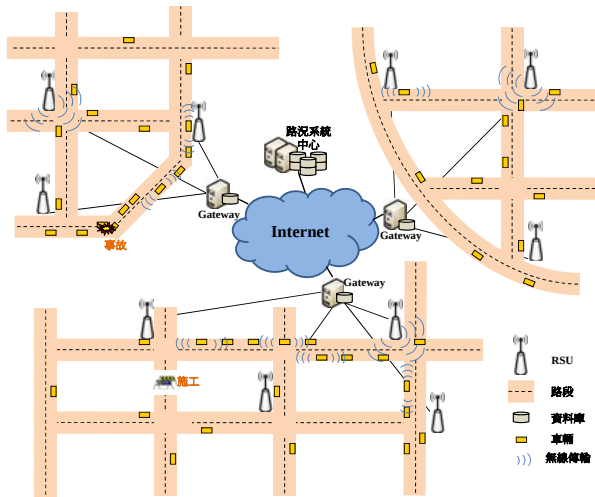


圖 1 以路口圖為基礎的單點路由架構

- 車輛：假設道路上所有車輛都備有電子地圖(e-map)，裝配有 OBU (On Board Unit) 可供車輛作車間 (Inter-Vehicle Communication, IVC) 與對 RSU (Vehicle-Roadside Communication, VRC) 執行無線通訊、並有 GPS (Global Positioning System) 可得知車輛本身位置。因此車輛可事先備有或即時透過 RSU 下載與歷史交通資料，也可收到 RSU 廣播或向 RSU 請求區域即時資訊，亦可送、收、轉遞(forward)其他車輛的訊息。在架構中的車輛將在問候訊息夾帶本身的即時資訊，例如速度與方向(假設與車輛的方向燈連動)；在遇交通事故時發出或轉遞交通安全訊息。
- RSU：假設其分佈在有歷史交通統計資料的路口，且每一 RSU 都能以有線的方式

透過閘道器(gateway)連結至網際網路。在架構中的功能為儲存(區域性)路口本身及所連接路段已做前置處理的交通歷史統計資訊與即時資訊；接收並廣播相關路段的交通事故訊息，將交通事故訊息送往路況系統中心；蒐集並儲存車輛問候訊息(hello message)中的車流資訊，週期性地送至路況系統中心，並用路況系統中心送回的累計資料更新 RSU 本身的統計資料。提供車輛下載各項交通訊息，若 RSU 本身無法提供，將向路況系統中心請求。

- 路況系統中心：目前提及類似路況系統中心功能文獻，例如 VADD 中提到使用 Yahoo Maps，是以提供電子地圖和歷史交通統計資訊為主。本架構的路況系統中心透過有線網路連結至網際網路。在靜態資料方面，負責將交通歷史統計資料做前置處理，並儲存其結果。在即時資訊方面，蒐集其他相關網站的路況資訊，包括施工、集會遊行或慶典活動等造成的交通管制訊息；接收由 RSU 送的交通安全訊息(safety message)，同時主動送至其他相關的數個 RSU；並把 RSU 週期性送回的即時車流資料和已儲存的車流資料做累計，成為新的統計資料，再送回該 RSU。另外提供 RSU 對靜態與即時資訊的請求，並控制其本身與所有 RSU 資料的同步化。

3.2 歷史交通統計資料的使用

以下分為三個步驟說明：

步驟 1. 有 RSU 路口的前置處理

假設所得到的交通歷史統計資料如臺北市交通管制工程處提供的交通流量資料[2]。詳細前置處理與接下來路口圖建立的部分內容參照我們先前研究的靜態路口圖設計。

步驟 2. 路口圖(Intersection Graph)的建立

將每一個路段都如 IG 方法進行檢驗，以(1)處理歷史交通統計資訊，計算出各個路段的平均車間距。

$$D_{veh} = \frac{T_{total}}{N_{veh}} \times V \quad (1)$$

當每路口的資料依上述方法處理完後，我們設 IEEE 802.11p 的傳輸半徑 R 為最大車間距門檻值，如方程式(2)。

$$R \geq D_{veh} \quad (2)$$



圖 2 以路口圖為基礎的單點路由架構

由(1)與(2)兩個方程式計算篩選的結果，即可在電子地圖上進行 IG 的建立。定 R 為車間距門檻值作為篩選可靠傳輸路段，是因為當 $R \geq D_{veh}$ 時，即表示這段道路的平均車間距，能以車間多個跳躍的無線傳輸來傳送封包，因此這個路段的兩個路口都成為 IG 中的節點。把所有車間距小於或等於 R 者，資料路口標於電子地圖上，以兩相鄰路口之間的路段視為 IG 的邊(edge, E)，邊的端點路口當作 IG 的節點(node, vertex, V)。測量相鄰兩 IG 節點正中央位置之間的距離當作是邊的長度 L ，定義方程式(3)計算權重(weight)為 W ，即傳輸的跳躍數(hop count)，以此做出一個具有權重的 IG 圖。因此執行完此三階段，將得到如圖 2. 的 IG 路口圖： $IG = (V, E)$ ，做為路由演算法計算時所依據的網路拓樸。

$$W = \begin{cases} \lfloor L/R \rfloor, & \text{if } L/R \in \mathbb{Z} \\ \lfloor L/R \rfloor + 1, & \text{if } L/R \notin \mathbb{Z} \end{cases} \quad (3)$$

步驟 3. 無 RSU 路口的前置處理

由於取得的歷史交通統計資料並非所有路口都有統計資料，若前置處理後沒有完整資料的路段都不能加入路口圖中，將會有很多事實上平均車間距小於等於無線傳輸半徑的路段，尤其是幹道，卻被排除在 IG 之外，產生 IG 上可逐次跳躍傳輸成功的路段較少，可能造成傳輸路徑壅塞，封包被丟棄(drop)的可能升高，以及端至端延遲 (End-to-End Delay) 增加。因

此，在上述處理完後，我們把剩餘的資料，並將已知資訊利用內差法推測沒有完整資料而未加入路口圖的路段車流。

例如圖 2. 中，南京東路口和復興北路交叉路口(I17)沒有統計資訊，因而與其連接的四個路段沒有完整資料，但是四個路段的另一端路口都有統計資料，所以我們用這些路口的統計資料推算出這四路段的車流。計算的方法我們以南京東路和敦化北路交叉路口(站牌號碼 SI120)之間的車流量為例，如圖 3. 所示。SI120 路口往西的車流統計資料是存在且確知資料(1700 輛車)，卻沒有南京東路和復興北路交叉路口(I17)往東的統計資料。而 SI120 路口的資料有由西面進入路口的車流統計資料(1567 輛車)，並且 SI120 路口正好是南京東路和復興北路交叉路口(I17)往東的相鄰路口，我們以此資料代表 I17 南京東路和復興北路交叉路口往東的車流。其他三個路段的求法以此類推。以上述方式仍無法推算出車流的路段，則以等比的線性內差法 (Interpolation method) 來推測。

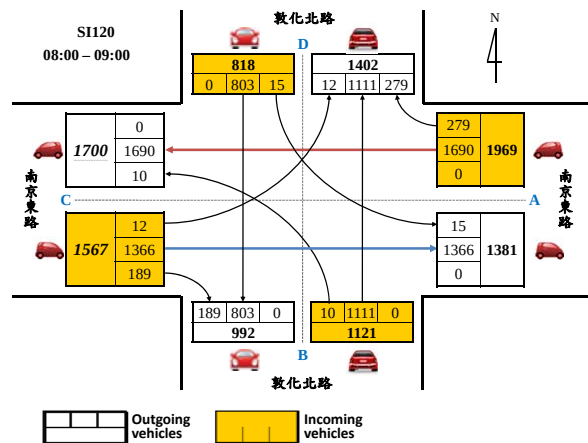


圖 3 道路及交通流量統計資料：和平東路與敦化北路交叉路口(SI120)

3.3 即時路況資訊的蒐集與運作

在本文中的即時資訊包含三類，前二類可造成車輛無法通過該路段(gap)，使得車輛網路長時間或暫時性的間斷。

◆ 第一類的即時資訊為施工或集會遊行、慶典活動造成的交通管制：

通常事先可得知開始與結束時間，且為時較長。蒐集的方法可由相關網站、調頻電台廣播(警廣)、RDS-TMC 的廣播頻道等得知。在本架構中蒐集的元件為路況系統中心。而由於各種資訊的形式與格式不盡相同，因此目前暫假設由人工蒐集輸入路況系統中心的資料庫

中，提供由 RSU 發出請求來查詢。

◆ 第二類為交通事故資訊：

假設發生事故的車輛會發出交通安全訊息，直接或經過一到數個跳躍 (hop) 的轉送到達最近的 RSU。收到交通安全訊息的 RSU 會將此訊息透過閘道器送至路況系統中心的資料庫，路況系統中心則會主動送至其他相關影響範圍內的數個 RSU。

◆ 第三類為車流資訊：

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Vehicle_ID															Direction															Speed									
Position																																							
Last RSU_ID															Timestamp																								

圖 4 由 GPSR 修改後的問候訊息格式

著重於使用 RSU 蒐集經過路口車輛的問候訊息。如此可避免大規模的泛濫法取得車輛即時分佈的耗費，同時累計的結果具長時間的可靠度。問候訊息的格式內容修改了原始 GPSR 的車輛 ID (Vehicle_ID)、車輛位置(Position)與時間戳(Timestamp)的欄位之外，另加上車輛的方向(Direction)、速度(Speed)與所經過的上一個 RSU (Last RSU_ID)三個欄位。我們假設 OBU 和車輛的方向燈作連動，而所有駕駛都遵守轉彎前打方向燈的規則，將可以得到方向資訊；或者由 GPS 座標計算或車用陀螺儀等方式得到方向資訊。因此我們將可得到與交通歷史統計資料相同形式的資料。然後 RSU 將蒐集到的車流資訊週期性地經由閘道器送往路況中心。路況系統中心經過累加與處理後，回傳該 RSU 執行更新，並提供更新後的統計資訊給發出請求的 RSU 查詢之用。

3.4 路由規劃

假設一開始所有車輛都已由所屬的 RSU 下載目前時段經前置處理的交通歷史統計資訊與 IG 圖，並且假設 C_{src} 從距離最近的 RSU 與路況中心可以隨時獲得經由 C_{dst} 回報所經過的 RSU 位置、速度與方向的資訊。本計畫將以「IG 圖」取代過去以「車輛」為節點的網路拓樸來進行路由選擇，做為之後資料傳輸時來源路由的路由。接下來分別說明使用的代號表、 C_{src} 執行路由規劃的步驟及其範例、以及來源路由表頭(source routing header)的格式。以下是使用的代號表：

C_{src} ：來源車輛。

C_{dst} ：目的地車輛。

I_{src} ：來源路口。

I_{dst} ：目的地路口。

I_{vit} ：虛擬路口 (Virtual Intersection)。

I_{intn} ：最小成本路由上的第 n 個中繼路口。

I_{srcN} ： C_{src} 所在的路段上移動將到達的下一路口。

I_{dstN} ： C_{dst} 所在的路段上移動將到達的下一路口。

3.4.1 路由規劃的步驟

路由規劃是由 C_{src} 根據上述資料進行最小成本路由的計算。其步驟如下：

- 由 RSU 獲取路況系統中心的第一、二類即時資訊，檢查是否有原來穩定路段中斷，設定這些中斷路段成為集合 B，將集合 B 從 IG 的路段集合 E 中刪除。(E ← E-B)
- 判斷 C_{src} 是否位於路口，若是，則該路口即為 I_{src} ($I_{src} \in V$)；若不是，則設 C_{src} 當時的座標為一個虛擬路口 (Virtual Intersection, I_{vit})，且 I_{src} 就是這個虛擬路口 I_{vit} ($I_{src} \leftarrow I_{vit}$)。
- 判斷 C_{dst} 是否位於路口，若是，則該路口即為 I_{dst} ($I_{dst} \in V$)；若不是，則設 C_{dst} 當時的座標為一個虛擬路口，且 I_{dst} 就是這個虛擬路口 I_{vit} ($I_{dst} \leftarrow I_{vit}$)。
- I_{src} 及 I_{dst} 路口節點設定完成後，基於靜態歷史交通資訊與第一、二類即時路況，以方程式(3)計算出每一路段的成本，包括真實路口之間的正常路段，以及並非位於真實路口的 C_{src} 和 C_{dst} 與真實路口之間的虛擬路段，利用 Dijkstra[4]或類似演算法計算由 I_{src} 到 I_{dst} 的最小成本路由。計算時 C_{src} 和 C_{dst} 目前行進方向將會到達的第一個的路口 I_{srcN} 與 I_{dstN} ，在最小成本路由的選擇上擁有最高優先權。
- 將計算完成後最小成本路由上的所有路口依順序記錄在集合 A 中。
(A ← < I_{src} , I_{int1} , I_{int2} , ..., I_{intn} , I_{dst} >)
- 去掉因 C_{src} 和 C_{dst} 的行進方向影響路由上的真實路口
 - 若 C_{src} 位於路口，且 C_{src} 行進的方向朝向第一個中繼路口 ($I_{srcN} = I_{int1}$)，則集合 A 去掉 I_{src} 。(A ← A - < I_{src} >)
 - 若 C_{dst} 位於路口，且 C_{dst} 行進的方向

朝向最後一個中繼路口 ($I_{dstN} = I_{intn}$)，
則集合 A 去掉 I_{dst} 。 ($A \leftarrow A - \langle I_{dst} \rangle$)

- 去掉集合 A 中的虛擬路口
 - 若 I_{src} 為虛擬路口，則集合 A 去掉 I_{src} 。 ($A \leftarrow A - \langle I_{src} \rangle$)
 - 若 I_{dst} 為虛擬路口，則集合 A 去掉 I_{dst} 。 ($A \leftarrow A - \langle I_{dst} \rangle$)
- 在集合 A 的最前面加上 C_{src} ，最後面加上 C_{dst} 。

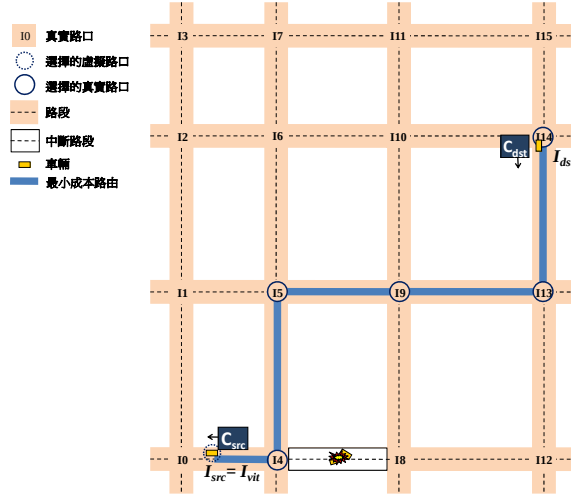


圖 5 IG 路由規劃

圖 5 是 C_{src} 將傳送資料封包給 C_{dst} 所做路由規劃的例子。 C_{src} 得知經前置處理的歷史交通統計資訊與 IG 圖如圖 5 的 16 個路口與其間的路段。向 RSU 取得的第二類即時資訊中表示，路口 I4 到路口 I8 之間發生交通事故，造成車流中斷， C_{src} 因此將此路段從路段集合 E 中刪除。由於此時 C_{src} 位於路段上而非路口，所以設 C_{src} 當時的座標為一個虛擬路口，而這個虛擬路口即為 I_{src} ；而 C_{dst} 正好位於路口 I14，故路口 I14 即為 I_{dst} 。 C_{src} 以上述設定，利用最小成本路由演算法（假設成本定義為路由長短/傳輸跳躍次數）計算出的最小成本路由為 $I_{src} \rightarrow I4 \rightarrow I5 \rightarrow I9 \rightarrow I13 \rightarrow I14$ (I_{dst}) 放入集合 A ($A \leftarrow \langle I_{src}, I4, I5, I9, I13, I14 \rangle$)。由於 I_{src} 是一個虛擬路口，因此集合 A 要去掉 I_{src} ，成為 $\langle I4, I5, I9, I13, I14 \rangle$ 。而因為 C_{dst} 位於路口且行進朝向最後的中繼路口 I13，所以集合 A 要去掉 I_{dst} ，也就是路口 I14，使得集合 A 成為 $\langle I4, I5, I9, I13 \rangle$ 。之後在集合的最前加上 C_{src} ，最後加上 C_{dst} ，得到最終的集合為 A: $\langle C_{src}, I4, I5, I9, I13, C_{dst} \rangle$ 。

3.4.2 來源路由表頭(Source Routing Header)的格式

C_{src} 在傳送資料封包時，將集合 A 中的路口放入來源路由表頭中，依此路由順序送至 C_{dst} 。來源路由表頭的格式如圖 6：

0	1	2	3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1			
Next Header	Type	Dir	Num. of Target
Target_ID [0]			
...			
Target_ID [n]			
$C_{dst_Direction}$		C_{dst_Speed}	Timestamp
$C_{dst_Position}$			

圖 6 IG 來源路由表頭的格式

- **Next Header**：8 位元。表示跟隨在 IG 來源路由表頭之後的協定的型態，例如傳輸層的表頭。
- **Type**：4 位元。表示此封包的型態，如果是使用 IG 路由的資料封包，Type=1。
- **Direction**：4 位元。表示此資料封包傳送的方向，如朝向 C_{dst} 傳送，則 Direction=1。如遇到目前路段中斷的情況，必須反向送回前一個已經經過的中繼路口，則 Direction=0。
- **Number of Target (N)**：最小成本路由集合 A 的元素個數-1，也就是除掉 C_{src} 之後所需經過的中繼路口的數目（包含 C_{dst} ）。
- **Segments Left**：路由剩餘的路口數目，也就是到達 C_{dst} 之前所需經過的中繼路口數目。最高為 N-1，當此欄位為 0 時，表示經過最後的中繼路口，將要到達 C_{dst} 的路段。
- **Target_ID[0..N]**：來源路由的中繼路口順序，由集合 A 取得每個元素。Target_ID[0]= C_{src} ；Target_ID[n]= I_{intn} ，這裡 $1 \leq n < N$ ；Target_ID[N]= C_{dst} 。這裡用路口的 ID 表示，假設使用 IG 路由的系統，都有將路口 ID 轉成路口中心座標的功能。
- **$C_{dst_Direction}$** ：在 C_{src} 送出封包時 C_{dst} 的方向。
- **C_{dst_Speed}** ：在 C_{src} 送出封包時 C_{dst} 的速度。
- **Timestamp**： C_{src} 送出封包的時間戳記。
- **$C_{dst_Position}$** ：在 C_{src} 送出封包時 C_{dst} 的位置。使用 IG 路由的車輛，都能參考封

包表頭相關欄位($C_{dst_Direction}$,
 C_{dst_Speed} , $C_{dst_Position}$, Timestamp)
 預測 C_{dst} 目前位置的功能。

3.5 資料傳送

當 C_{src} 算出最小成本路由之後，隨即以路由貪婪的方式，將資料封包傳送至 C_{dst} 。也就是說，封包傳送的下一轉遞車輛的選擇，是以最小成本路由上趨近「目標路口 (Target Intersection)」的貪婪法進行。以圖 12 為例，一開始 C_{src} 算出最小成本路由 $A = A : \langle C_{src}, I3, I4, I7, I10, C_{dst} \rangle$ ，所以送出的封包內 IG 來源路由表頭中的 Type 欄位值為 1 (Type=1)，表示是使用 IG 路由的資料封包。Direction 欄位值為 1 (Direction=1)，表示封包送往 C_{dst} 。Number of Target 欄位值為 5 (Number of Target=5)，表示路由過程中包含 C_{dst} 總共需趨近 5 個目標路口。Segments Left 欄位值為 4 (Segments Left = 4)，表示傳送路由上有 4 個中繼路口 ($n=4$)，且尚有 4 個中繼路口未經過。Target_ID [0..5] = { C_{src} , I3, I4, I7, I10, C_{dst} }。而此時的目標路口為 I3 (Target_ID [1])。從 C_{src} 到路口 I3 的路段上，收到封包的車輛會選擇最接近路口 I3 中心點座標的鄰居車輛為下一轉遞車輛，直到收到封包的車輛中，存在和路口 I3 座標範圍相同的鄰居車輛或路口 RSU 時，將封包隨機轉遞給這些在路口的鄰居的其中一個。在路口的車輛或 RSU 收到封包後，會把 Segments Left 欄位值減去 1 (Segments Left = 3)，此時的目標路口則變為路口 I4 (Target_ID [2])。從路口 I3 到達路口 I4 之間的路段上的運作和前一路段相似，只是選擇的鄰居車輛要最接近的座標為路口 I4 的中心點，而不再是路口 I3 的中心點。同樣地，在收到封包的車輛中，發現有和路口 I4 座標範圍相同的鄰居時，將封包隨機轉遞給這些在路口鄰居的其中一個。在路口收到封包者，會再將 Segments Left 欄位值減去 1 (Segments Left = 2)，目標路口也再改為下一個中繼路口 I7 (Target_ID [3])。這些循環地運作將一直重覆，直到 Segments Left 欄位值成為 0 時，也就是封包的傳送進入了最後路段。此時利用 IG 來源路由表頭中的 $C_{dst_Direction}$ 、 C_{dst_Speed} 、Timestamp 以及 $C_{dst_Position}$ 預測出 C_{dst} (Target_ID [5]) 當時的座標，此時轉遞車輛的選擇才會真正使用 C_{dst} 的座標，然後持

續到鄰居車輛中存在 C_{dst} 時，將封包轉遞給 C_{dst} ，完成一次資料封包的傳送。

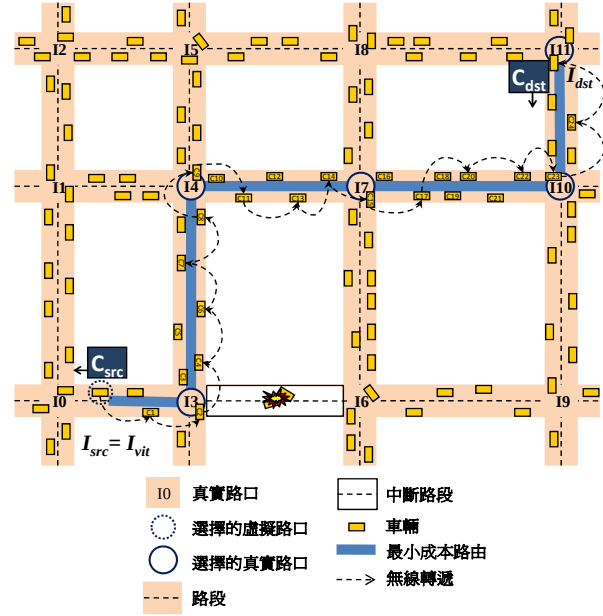


圖 7 IG 路由資料封包傳送的過程

收到資料封包的車輛或 RSU 將進行下列演算法：

$N = \text{Number of Target};$

判斷 Segments Left 是否為 0

是，則判斷自己是否為 C_{dst} (Target_ID [N])

是，則封包到達目的地

否，則判斷鄰居車輛中是否有 C_{dst}

是，轉遞封包給 C_{dst}

否，則以 $C_{dst_Direction}$ 、 C_{dst_Speed} 、Timestamp 與 $C_{dst_Position}$ 預測出 C_{dst} 當時的座標，目標座標設為 C_{dst} 的座標，將封包轉遞給最接近目標座標的鄰居

否，則判斷自己是否位於目標路口的座標範圍內，目標路口的 ID 為 Target_ID [m]，這裡 $m \leftarrow N - \text{Segments Left}$

是，則 Segments Left $\leftarrow \text{Segments Left} - 1$ ；

更新目標路口為 Target_ID [m]，這裡 $m \leftarrow N - \text{Segments Left}$

判斷是否有位於目標路口的鄰居

是，則判斷位於路口的鄰居數目是否大於 1

是，則隨機選擇其中一個鄰居，轉遞封包給這個鄰居

否，則轉遞封包給唯一在目標路口的鄰居

否，則目標座標設為目標路口的中心點座標，將封包轉遞給最接近目標座標的鄰居

4. 模擬與分析

表 1 模擬參數表

項目	設定值
模擬範圍	4200 公尺 × 4200 公尺
車輛數目	965 輛
CBR 送、收車輛對數	10 - 50 輛
訊號範圍	250 公尺
模擬時間	150 秒
車速	3 - 10 公尺/秒
CBR 傳送率	10 封包/秒
MAC 協定	IEEE 802.11 DCF RTS/CTS
實體傳播	TwoRayGround
資料封包大小	512 位元組

在網路的效能評估部份，我們使用的網路模擬工具為 NS-2.34，模擬參數如表 1。模擬環境是如本文第三節所述，是根據台北市道路電子地圖截取其中棋盤狀的地理區域(如圖 2)，共有 25 個路口，40 個路段，路段上的車輛分佈以交通流量統計資訊設定平均的車間距，無統計資訊的路段則使用內插法設定平均車間距，形成車輛網路的初始拓樸。模擬中初始車速為 6 公尺/秒，所有車輛每 5 秒隨機變動 1-2 公尺/秒，最高及最低車速不超出 3-10 公尺/秒。由於篇幅有限的關係，在此只顯示出發生第一類及時訊息的道路障礙造成道路無法逐次跳躍傳輸的平均封包成功送達率與端到端延遲時間。

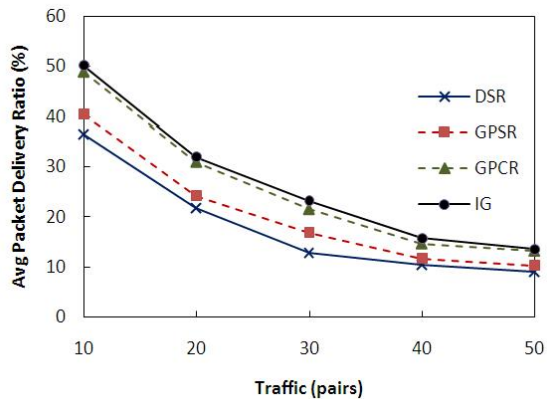


圖 8 所有路段上有兩段道路障礙造成道路無法逐次跳躍傳輸的平均封包成功送達率

圖 8 的橫軸為 CBR 送、收車輛對數，可以觀察到 DSR 及 GPSR 無論送、收對數的多少，封包成功送達率都低於 GPCR 與我們所提出的 IG。但道路障礙造成道路無法逐次跳躍傳輸的道路數量對 DSR 封包成功送達率的影響也不大，因為 DSR 是不以距離來決定轉送的節點，不會產生判斷錯誤的情形。而 GPSR、GPCR 與 IG 這類基於位置的路由協定，在無道路障礙的情況下，表現的差別不大，僅有 GPSR 略為遜色，GPCR 的表現幾乎和 IG 一樣。不過當路段中有道路障礙無法逐次跳躍的情況發生時，GPSR 和 GPCR 平均封包成功送達率即會依次明顯遞減，而 IG 則只會因可選擇傳送的路段減少所造成較高的網路流量負擔而使平均封包成功送達率稍微下降。可見得我們所提出的方法優於其他路由協定。

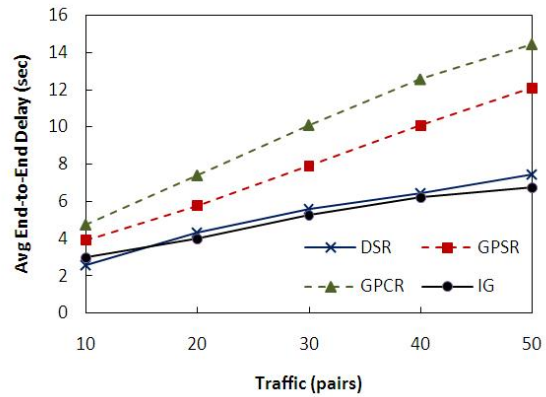


圖 9 所有路段上有兩段道路障礙造成道路無法逐次跳躍傳輸的平均端到端延遲時間

圖 9 是端到端延遲時間的情況。其中 IG 的延遲僅在封包送、收對數為 10 的時候，稍微高於 DSR，剩下的其他時候都完全不高過於其他所，可說明其成功降低了原本路口觀念路由協定所增加的延遲時間。原因是 IG 以路口的方法所降低的跳躍數超過 GPSR 的 20% 以上。這使得轉送的次數減少，路口車輛的流量負擔也同時下降。這種成效在道路障礙的路段增加時更為明顯。圖 9 也說明了當發生道路障礙無法逐次跳躍的路段時，GPSR 就有可能進入邊緣模式(perimeter mode)，造成跳躍次數大幅上揚且延遲時間上升，並且依無法逐次跳躍路段數量的增加而遞增。由此可證明利用歷史交通流量統計資料加上更新即時路況對路由協定的效能有明顯的助益。

5. 結論

本文中，我們以「路口」代替位置高度變動的「車輛」，用「計算」而非泛濫法傳送大量無線控制訊息的方式得到最小成本路由，減少網路資源消耗。而封包傳送以趨近「目標路口」而非「目的地車輛」的貪婪法，避免以位置為基礎的路由經常出現的區域最大值的問題。最小成本路由的建立有交通歷史統計資訊的支持，因此是「長期穩定的」。同時使用多種即時資訊，讓基於交通歷史統計資訊建立的路由能根據即時交通狀況進行調整。

參考文獻

- [1] 張英超、李原嘉、王元芬，“應用於高效率車輛單點路由協定的交通統計路口圖設計”，**NCS 2009 全國計算機會議論文集**，pp. 271-279, November 2009。
- [2] 臺北市交通管制工程處全球資訊網：交通流量調查資料，
<http://www.bote.taipei.gov.tw/ct.asp?xItem=660485&CtNode=20205&mp=117031>, December 2010.
- [3] Y.-S. Chen, Y.-W. Lin, and C.-Y. Pan, “DIR: Diagonal-Intersection-Based Routing Protocol for Vehicular Ad Hoc Networks,” **Telecommunication System**, Vol. 46, Issue 4, May 2011.
- [4] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, **Introduction to Algorithms**, New York: McGraw-Hill, 1990, ch. 24.
- [5] M. Ghaffari and F. Ashtiani, “A New Routing Algorithm for Sparse Vehicular Ad-Hoc Networks with Moving Destinations,” **in proc. of IEEE Wireless Communications and Networking Conference**, pp. 1-6, April 2009.
- [6] B. Karp and H. Kung, “GPSR: Greedy Perimeter Stateless Routing for Wireless Networks,” **in proc. of 6th ACM Mobile computing and networking**, pp. 243-254, 6-11 August 2000.
- [7] J. LeBrun, C.-N. Chuah, D. Ghosal, and M. Zhang, “Knowledge-Based Opportunistic Forwarding in Vehicular Wireless Ad Hoc Networks,” **in proc. of 61st. IEEE Vehicular Technology Conference**, Vol. 4, pp. 2289-2293, May, 2005.
- [8] K.-C. Lee, P.-C. Cheng, M. Gerla, “GeoCross: A Geographic Routing Protocol in the Presence of Loops in Urban Scenarios,” **Ad Hoc Networks**, Vol 8, No. 5, pp. 474-488, July, 2010.
- [9] K.-C. Lee, J. Haerri, U. Lee, and M. Gerla, “Enhanced Perimeter Routing for Geographic Forwarding Protocols in Urban Vehicular Scenarios,” **in proc. of IEEE Globecom conference**, pp. 1-10, 2007.
- [10] T. Li, Y. Li, and J. Liao, “A Contention-Based Routing Protocol for Vehicular Ad Hoc Networks in City Environments,” **in proc. of 29th IEEE Distributed Computing Systems Workshops**, pp. 482-487, June 2009.
- [11] C. Lochert, M. Mauve, H. Fussler, and H. Hartenstein, “Geographic Routing in City Scenarios,” **ACM Mobile Computing and Communications Review**, Vol. 9, No. 1, pp. 69-72, January 2005.
- [12] V. Naumov and T. Gross, “Connectivity-Aware Routing (CAR) in Vehicular Ad Hoc Network,” **in proc. of 26th IEEE International Conference on Computer Communications**, pp. 1919-1927, May 2007.
- [13] J. Nzouonta, N. Rajgure, G. Wang, and C. Borcea, “VANET Routing on City Roads using Real-Time Vehicular Traffic Information,” **IEEE Transactions on Vehicular Technology**, Vol. 58, No. 7, pp. 3609- 3626, September 2009.
- [14] S. Rao, M. Pai, M. Boussedjra, and J. Mouzna, “GPSR-L: Greedy Perimeter Stateless Routing with Lifetime for VANETS,” **in proc. of 8th IEEE ITS Telecommunications**, pp. 299-304, October 2008.
- [15] J. Zhao and G. Cao, “VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks,” **in proc. of 25th IEEE International Conference on Computer Communications**, pp. 1-12, April 2006.