

行政院國家科學委員會補助專題研究計畫 成果報告
 期中進度報告

開發一個高品質的 IEEE 802.11p 及 IEEE 1609 車輛無線網路模擬系統 (1/3)

計畫類別： 個別型計畫 整合型計畫

計畫編號：97-2221-E-009-065-MY3

執行期間：2008 年 8 月 1 日至 2009 年 7 月 31 日

計畫主持人：王協源

共同主持人：無

計畫參與人員：

成果報告類型(依經費核定清單規定繳交)： 精簡報告 完整報告

本成果報告包括以下應繳交之附件：

- 赴國外出差或研習心得報告一份
- 赴大陸地區出差或研習心得報告一份
- 出席國際學術會議心得報告及發表之論文各一份
- 國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、
列管計畫及下列情形者外，得立即公開查詢

涉及專利或其他智慧財產權， 一年 二年後可公開查詢

執行單位：國立交通大學資訊工程系

中華民國 98 年 5 月 31 日

一、計畫中文摘要及關鍵詞

智慧型運輸系統是一個正在快速興起的產業，很多專家學者都看好它會成為台灣的第三個產值超過兆元的產業。美國及世界其他很多先進國家也都看好這個產業，都正在積極投入人力及資金進行研發。車輛無線網路 (vehicular wireless networks) 是近幾年來剛興起且非常熱門的研究領域，目的就在於利用車輛之間的短距無線通訊網路來達成智慧型運輸系統所要達成的種種目標及建造出各式各樣有用的應用服務。被此新穎車輛無線網路所致能的應用有非常多種。目前專家學者將它們歸類為 (一) 車輛及行人安全應用 (二) 道路及交通資訊應用、及 (三) 一般資訊及娛樂應用。希望能利用車輛無線網路科技來實現這些應用以達成造福人群的目的。

有鑑於這種新網路的重要性，IEEE society 已經成立 802.11p working group 來制訂適合這種網路的通訊協定及系統。802.11p 通訊協定標準是以 802.11a 通訊協定標準為基礎，其上再依據車輛無線網路的種種獨特性質而做修正及增加新功能。除了 802.11p 及 802.11a，IEEE society 已經制定了 IEEE 1609 通訊協定標準的草案。IEEE 1609 包含 1609.1, 1609.2, 1609.3, 1609.4 四個子標準，分別處理車輛無線網路在通訊協定堆內各層的問題，這些層包含實體層、媒體存取控制層、網路層、傳輸層、應用層及網路安全的種種議題。

由於車輛無線網路是一個非常新的網路，目前市場上無任何產品可供實驗進行研究，因此開發一個能模擬此類網路的軟體網路模擬器是非常重要的。這是因為若能有一個高品質的此類網路模擬器，很多為此類網路所設計的新穎通訊協定設計及應用服務就可在此模擬器上進行功能驗證及效能評估。除此之外，由於不用去租借多部車輛及購買多套硬體網路設備、且不用同時召集多人一起進行實驗，模擬研究的成本也比真實道路實驗小很多。另外從安全上來考量，模擬研究也比在路上進行真實實驗來的更加安全。總之，為此類新穎的車輛無線網路開發一個高品質的網路模擬器是一件非常重要且有價值的工作。

在這個所提持續三年的研究計畫中，第一年我們將著重於開發 802.11p、802.11a 及 802.11g 模擬系統。第二年，我們將著重於開發 IEEE 1609.1, 1609.2, 1609.3 及 1609.4 模擬系統，第三年我們將著重於開發道路網路的模擬及符合真實世界車流運動行為的車輛行為模擬。由於這些通訊協定標準都非常複雜完整，預計開發這些模擬系統的工作份量都很大。另外各模擬模組開發完後尚須要經過嚴謹繁複耗時的 validation 及 verification 程序其模擬結果的正確性才能被信賴，因此預計各年度的研究工作份量都會相當吃重。除此之外，由於這些模擬子系統最後需要整合在一起，它們之間的整合、整合測試、及除錯預期也會相當地費時費力。

開發一個高品質車輛無線網路模擬器需要長期的研究投入，由於其對於研究此一新穎的網路是非常重要的且不可或缺，因此本計畫預計用三年的時間來持續開發此一重要研究工具。

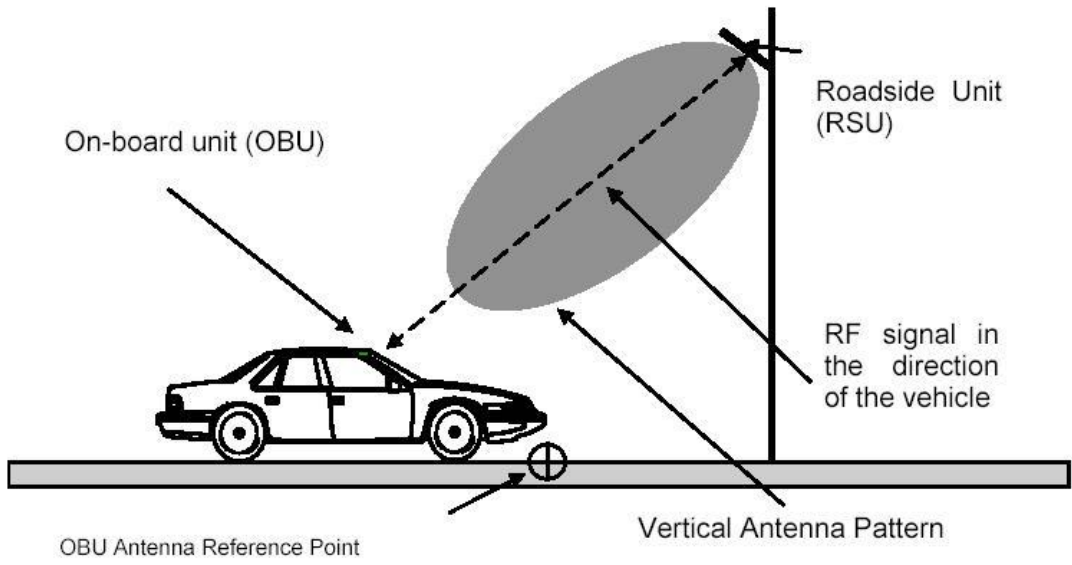
關鍵字：智慧型運輸系統、車輛無線網路、電腦模擬

二、第一年計畫介紹

DSRC (Dedicated Short Range Communication) 為目前車間短距離通訊的一種技術，它是利用裝置於路旁的定點通訊感應設備(Road Side Unit, RSU)，與車上單元(On Board Unit, OBU)(如圖一)互動進行通訊，旨在提高道路安全和運輸效率，並提供駕駛人網路服務。此技術可以提供駕駛人在行進途中，透過車上無線通訊裝備、路測裝置或車與車間的相互傳輸，接收各種即時的資訊服務(路況導航、即時路況資訊、多媒體娛樂等)，以提升整體路網行車效率和交通安全。DSRC 又分為車間通訊(V2V)及車路通訊(V2R)兩種(如圖二)。車間通訊指的是車輛在一定的距離範圍內，透過車上裝置相互傳輸交換訊息。而車路通訊是透過交控中心進行資訊整合，透過路邊裝置將資訊傳送至車上。其基本應用包括：高速高路收費站的自動收費(如圖三)、天氣惡化導致路況變化的消息通報、警示經過道路上的修路訊息及加油站的自動付款系統等等。

目前 DSRC 各國應用發展的情況都不大相同，各國採用的 RF 技術、通訊協定及採用的頻帶都不盡相同。值得注意的是 802.11p 為一以 802.11a 為基礎發展出傳輸於 5.9 GHz 頻段的新無線技術，5.9 GHz 為美國交通運輸的專屬頻道透過 LOS (line of sight) 以及 NLOS (non-line of sight) 的方式提供動態車輛安全資訊的傳遞，並同時提供了寬頻、即時、長距離以及雙向通訊的特性。且 802.11p 是由 802.11a 所修改過來的通訊協定，所以未來 802.11p 也可相容連結於 802.11a 的 hotspot 上網。

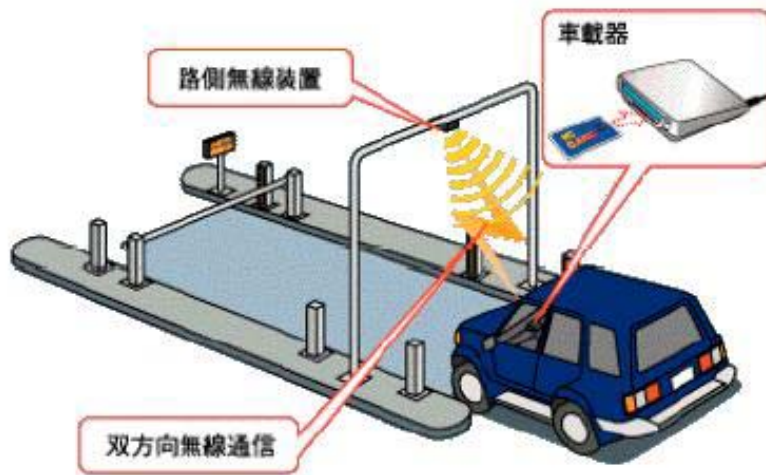
車間通訊網路是一個非常新的網路拓撲架構，且目前市場上並沒有一套完整且便利的網路模擬系統，所以依照標準規格 802.11p 為基礎所開發出來的網路模擬系統將可以用來模擬未來真實世界中車間通訊服務各種服務應用的實際狀況，不但可以大量地節省時間上、人力上的成本，而且可以提供尚未實際在真實世界上提供的車間通訊服務良好的測試平台，以利於觀察到實際應用於真實世界上所可能發生的問題，避免空間、時間和財力上直接的耗費，最重要的，還是可以降低實驗人員在安全上的考量。有了此套模擬器，我們將可以透過幾台具有高速運算能力的電腦，在實驗室內先觀察測試結果，並調整系統參數或修改車間通訊協定來找出合適的解決方案，這對於開發未來實際服務的應用，可說是事半功倍且不可或缺的工具。



圖一、OBU 和 RSU 通訊



圖二、車間通訊和車路通訊



圖三、DSRC 應用於 ETC 系統上

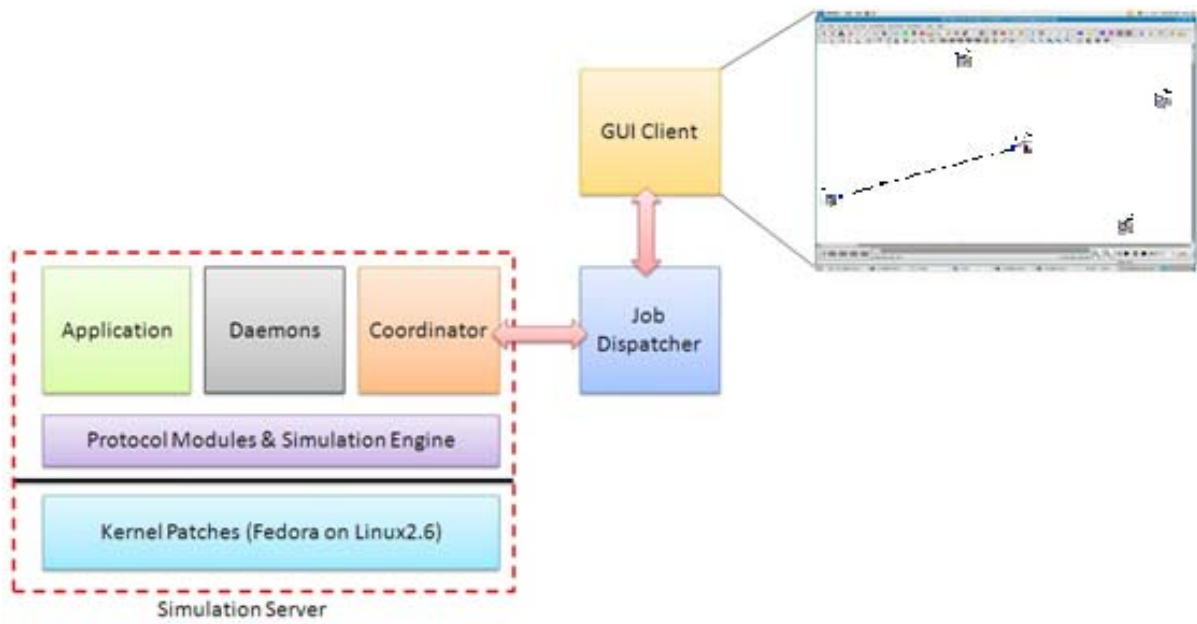
三、模擬平台介紹

NCTUns 是一個功能強大的研究工具，它提供兩個獨特的功能，是一般傳統的模擬器難以實現的。第一個是能夠使用現實世界 Linux 核心的 TCP/UDP/IP 協議進行模擬。這種功能使它能夠產生接近真實的模擬結果。第二個是能運行現實生活中的應用程式。此功能是它能夠利用現實世界的網路流量來進行模擬。因為這兩個獨特的功能，NCTUns 可以產生更準確的模擬結果。

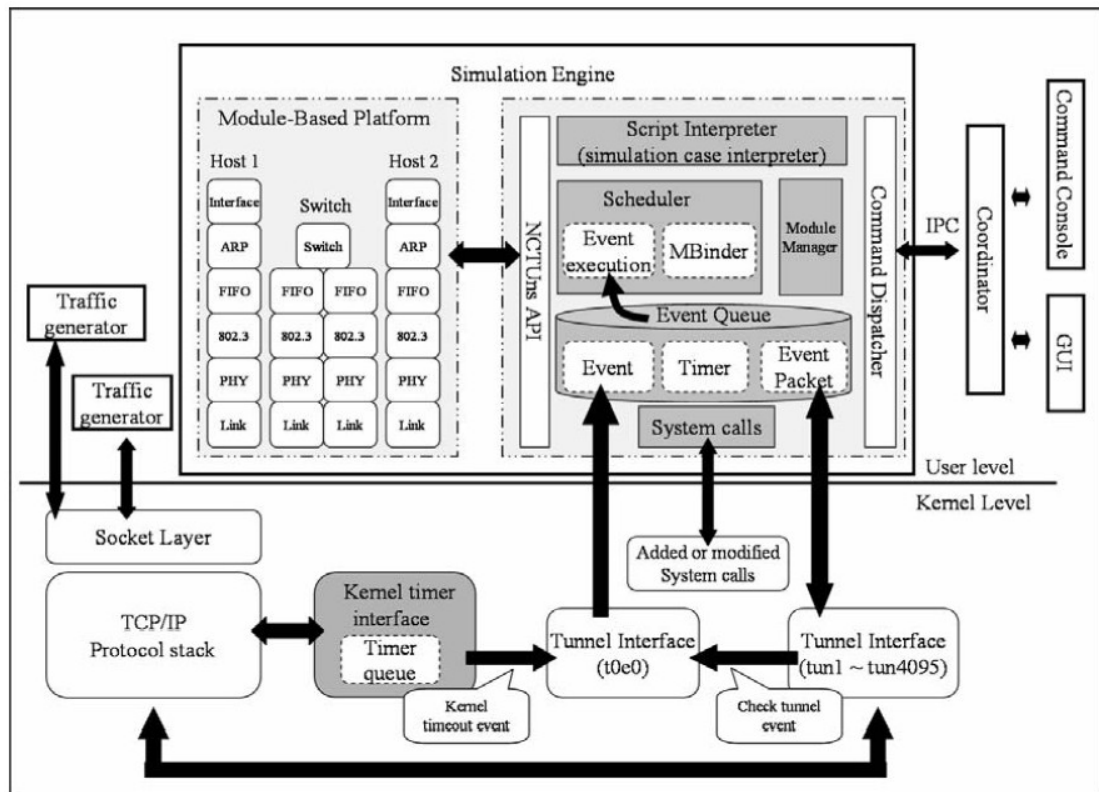
如圖四所示，NCTUns 一共由以上幾個元件所組成，在進行模擬的工作時，會先由 GUI Client 與 Job dispatcher 溝通，在此處的 Job Dispatcher 是一個負責工作排程用的元件，當有 Simulation Server 開始運作時，都會向此元件進行註冊的動作。而 GUI Client 若產生一個模擬或是擬真的工作時，會向 Job Dispatcher 要求可用的 Server，此時若有 Server 尚未運作，則 Job Dispatcher 會將該 Server 的 IP 位置傳給 GUI Client，讓 GUI Client 能夠將工作交給該 Server 上的 Coordinator。當 Server 上的 Coordinator 收到一份工作時，則會開始執行 Simulation Engine。Simulation Engine 為模擬器的核心元件，它根據模擬或擬真的工作描述檔來將整個環境建立起來，包括底層用來模擬的 Protocol Stack、按照使用者設定於哪時應該要讓哪支程式開始運行等，並進行網路傳輸事件的模擬。在模擬的工作結束後，即會將工作回傳給 Coordinator，由 Coordinator 將這些描述結果的檔案回傳給 GUI Client，使得 GUI 能夠將結果以動畫的方式呈現給使用者，也能透過結果的數據來畫出效能的圖形。

模擬的過程中，當某支應用程式將封包自虛擬的網路設備傳送出去時，其將會透過 Kernel 的 Socket Layer，並經過 TCP/IP Protocol Stack 來進行一些在傳送時應該進行的動作，之後會送到一個虛擬的網路介面卡。這個網路介面卡稱為 Tunnel Interface，當初 NCTUns 在開發時，為了能讓封包自 Kernel 導至位於 User Level 的 Simulation Engine，因此進行了一系列的 Kernel Patch 的更動，此更動流程除了增加了 System Call 之外，另外還有 Tunnel Interface 的建立、某些 Kernel Module 的設計等，使得不需要將這些網路處理的流程也一起放到 Simulation Engine 當中造成模擬的失真。而 Tunnel Interface 在收到封包後，Simulation Engine 會將這些封包從 Tunnel Interface 收上來，並以事件的方式進行處理。處理過程中透過圖五 Simulation Engine 左半部的 Module-Based Platform，將封包導到 Engine 內的虛擬模組，並一層一層的往實體層送下去。這些虛擬模組大多為資料連結層的 Protocol 與實體層的機制或 Model，是由 C++ 撰寫而成的，透過撰寫 Class 與 C++ 繼承的機制，開發者可以自由的擴充，因此能夠很容易的在這套平台上開發新的 Protocol。

而封包由 Protocol Module 送出經 link 傳送後，還會再被導到目的地端的 Protocol Module 內，並往上送至 Interface Module。送至該 Module 時，Engine 會將此封包導入對應的 Tunnel Interface 當中，緊接著，透過 Kernel 內對於 TCP/IP Protocol 的處理，使得其可由 Socket Layer 送到對應的目的地端的應用程式中。



圖四、NCTUns 的架構圖



圖五、Protocol Modules & Simulation Engine 內部架構

四、第一年計畫所開發的 IEEE 802.11p 模擬環境介紹

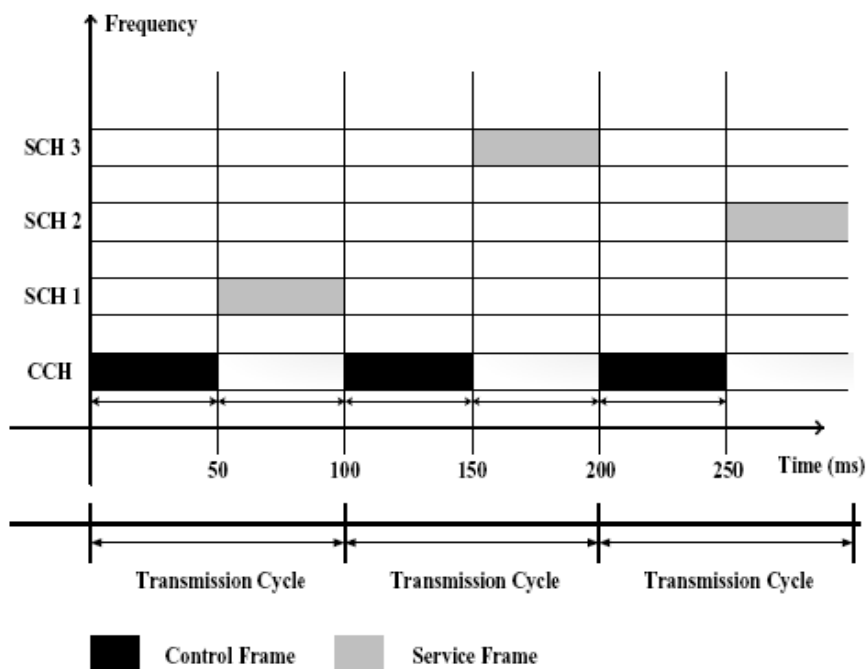
這個章節將簡單地介紹 IEEE 802.11p 網路協議以及我們如何在 NCTUns 上模擬 IEEE 802.11p 網路協議。

IEEE 802.11p 網路協議

802.11p又稱Wireless Access in the Vehicular Environment (WAVE) 主要是用在車用電子的無線通訊上。其中定義在實體層(physical layer)的屬性快速變化下以及在非常短的通訊期間內溝通該如何運作。這樣的目的是希望確保在溝通期間非常短和快速變化的環境下，訊息能夠在完整的傳輸在802.11的infrastructure或Ad-hoc網絡下。802.11p提供傳輸IP封包和WSMs(WAVE Short Messages)。

如圖六所示，802.11p將使用的頻譜分為CCH(control channel)和SCHs(service channels)。CCH是讓端點的用來交換控制信息，而SCHs是讓端點交換封包和WSMs。頻寬在時間軸上分成一個一個的 transmission cycle。而每一個 Transmission Cycle 包含一個 control frame 和 service frame。其中在 draft standard 裡建議在一個 control frame 或 service frame 的時間區間為 50 毫秒。

802.11p將網路節點分為兩類，OBU(On-board Unit)和RSU(Road-side Unit)。網路封包的傳輸只能允許在WBSS(Wave-mode Basic Service Set)發生。其中起始WBSS的端點稱做WBSS provider，而加入WBSS的端點稱做WBSS users。為了建立WBSS，WBSS provider必須定期在CCH廣播WSA(Wave-mode Service Announcement)。WSA包括WBSS的ID和WBSS使用的SCH。端點可以在CCH偵測到全部WSA的訊息。在知道想要加入的WBSS的SCH後，端點可以藉由改變SCH到特定的WBSS的SCH來加入此WBSS。



圖六、CCH 和 SCH 的時間區間

我們如何使用 NCTUns 模擬 802.11p

我們新增了兩個新的按鈕表示 802.11p 的 OBU(On-board Unit)和 RSU(Road-side Unit)。如下圖所示，圖的左邊是 OBU，右邊是 RSU。



首先我們先介紹 OBU 的設定視窗，OBU 的設定視窗裡有包括六個子視窗，分別是 1) IEEE 802.11(p) Provider Setting；2) IEEE 802.11(p) User Setting；3) Application；4) Path (for Testing)；5) Down Time and 6) Mobile IP。下面將介紹 IEEE 802.11(p) Provider Setting 和 IEEE 802.11(p) User Setting 兩個子視窗的用法。

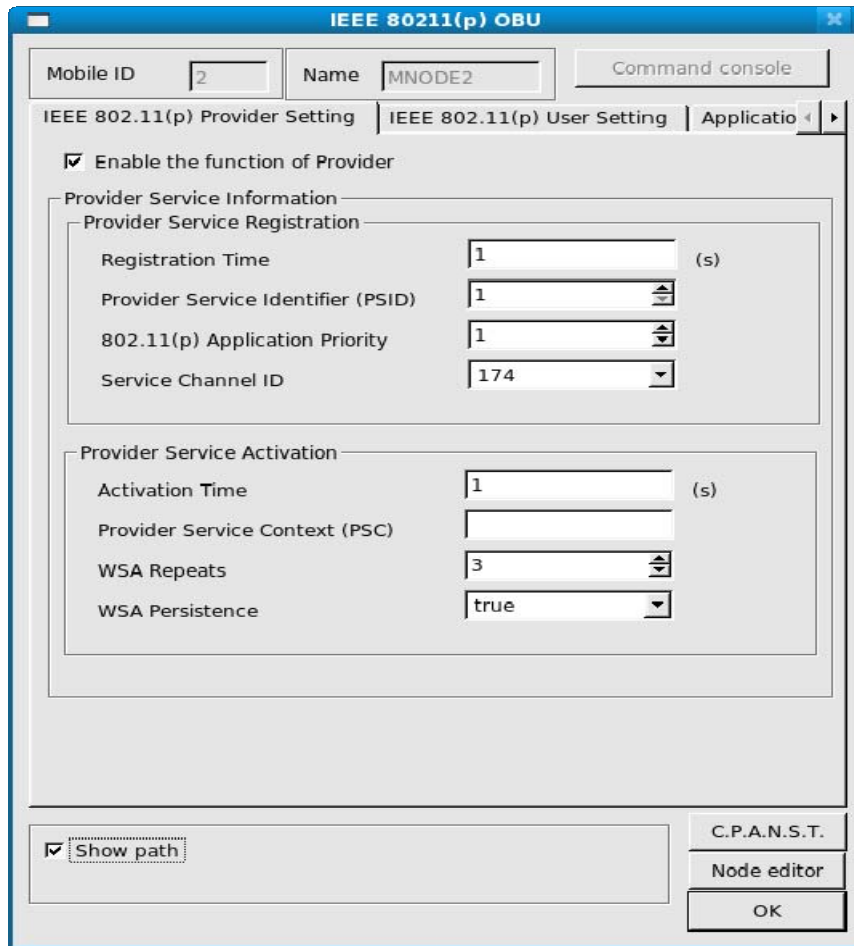
IEEE 802.11(p) Provider Setting

如圖七所示，在 IEEE 802.11(p) Provider Setting 視窗中，使用者可以指定讓 OBU 提供什麼應用程式在勾選 “Enable the function of Provider” 後，使用者可以在 Provider Service Information (PSI) 視窗設定 Provider 的參數。PSI 視窗包含兩個部分，第一個部分是 provider service registration。為了讓 OBU 提供應用程式服務，須先設定下面四個欄位來註冊這個應用程式，分別是 1) the Registration Time；2) the Provider Service Identifier (PSID)；3) the 802.11(p) Application Priority 和 4) the Service Channel ID。第一個欄位指定什麼時候 OBU 註冊應用程式。第二個欄位指定這個應用程式的 ID。第三個欄位指定應用程式的優先等級，數字愈大代表在廣播訊息的時候較高的優先權。最後一個欄位指定應用程式要使用的 Service Channel 的 ID，其中允許使用的 ID 有：174, 175, 176, 180, 181 和 182。PSI 視窗的第二個部分是 provider service activation。在註冊完應用程式後，需設定下面四個欄位來指定什麼時候啟動應用程式，分別是 1) the Activation Time；2) the Provider Service Context (PSC)；3) the WSA repeats 和 4) the WSA persistence。第一個欄位指定什麼時候啟動應用程式，這個欄位的時候一定要比 the Registration Time 的時間大。第二個欄位指定應用程式的名字。第三個欄位指定在一個 control frame 裡 WSA 重複的數目。第四個欄位指定是否會持續在每個 control frame 傳送 WSA。

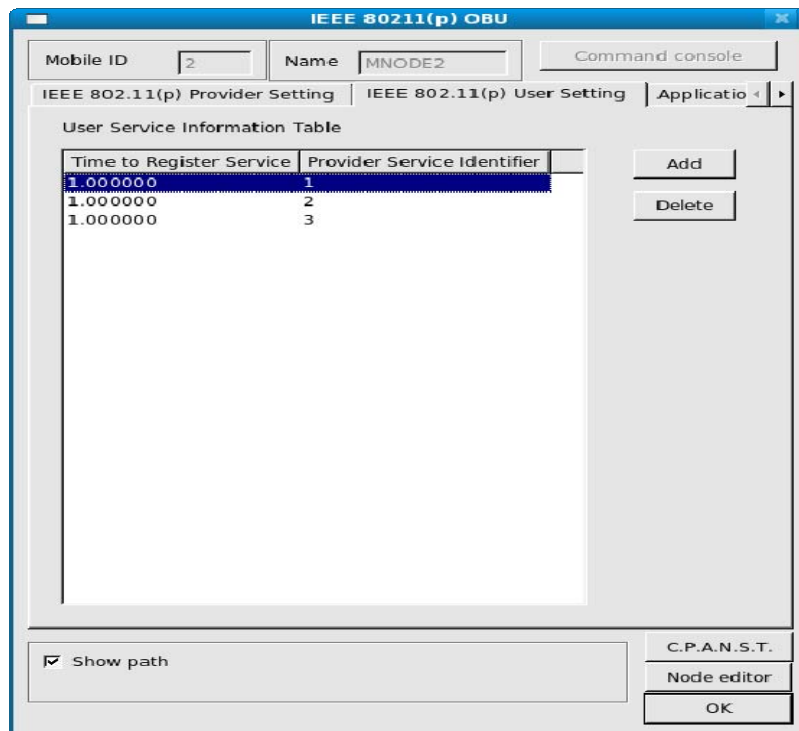
IEEE 802.11(p) User Setting

在 IEEE 802.11(p) User Setting 設定視窗中，使用者可以指定 OBU 在什麼時間開始加入特定 ID(PSID)的 service provider 的 WBSS。如圖八所示，OBU 在模擬時間 1 秒後，如果收到 ID 為 1、2 或 3 的 service provider 發送的 WSAs，則 OBU 會被觸發去加入這些 service provider 的 WBSSs。反之，如果在一秒之前收到這些 service provider 的 WSA 的話，則不會加入這些 service provider 的 WBSSs。

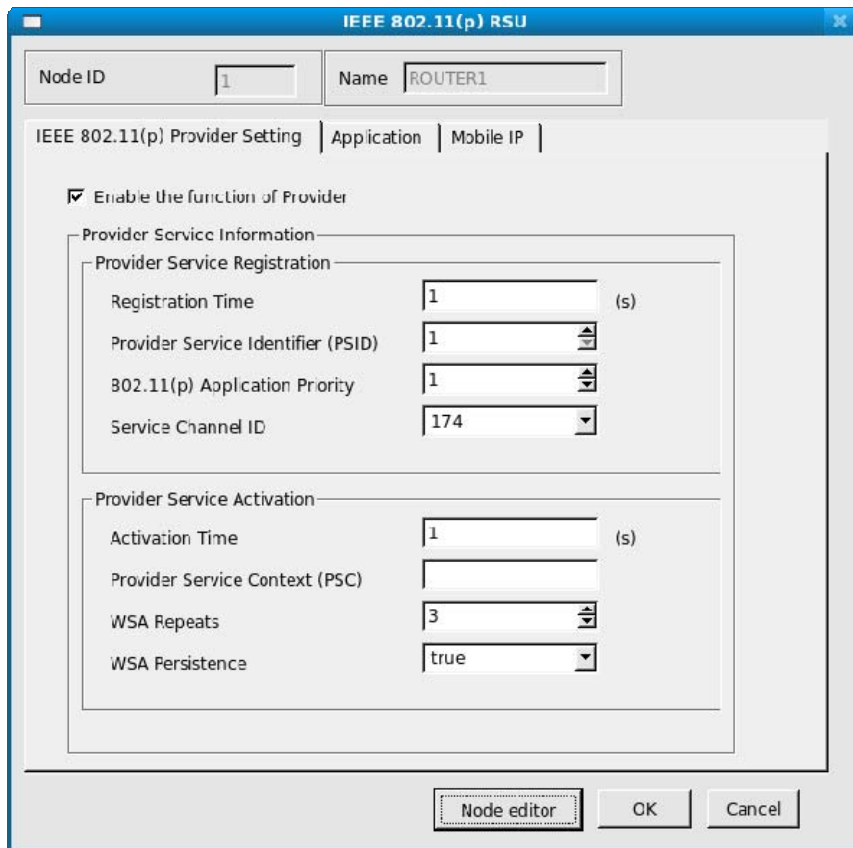
接下來我們先介紹 RSU 的設定視窗，RSU 的設定視窗裡有包括三個子視窗，分別是 1) IEEE 802.11(p) Providing Setting；2) Application and 3) Mobile IP。如圖九所示，IEEE 802.11(p) Providing Setting 和 OBU 的使用方法一樣，我們就不再說明。



圖七、OBU的IEEE 802.11(p) Provider Setting設定視窗



圖八、OBU的IEEE 802.11(p) User Setting



圖九、RSU的IEEE 802.11(p) Provider Setting設定視窗

五、模擬結果驗證

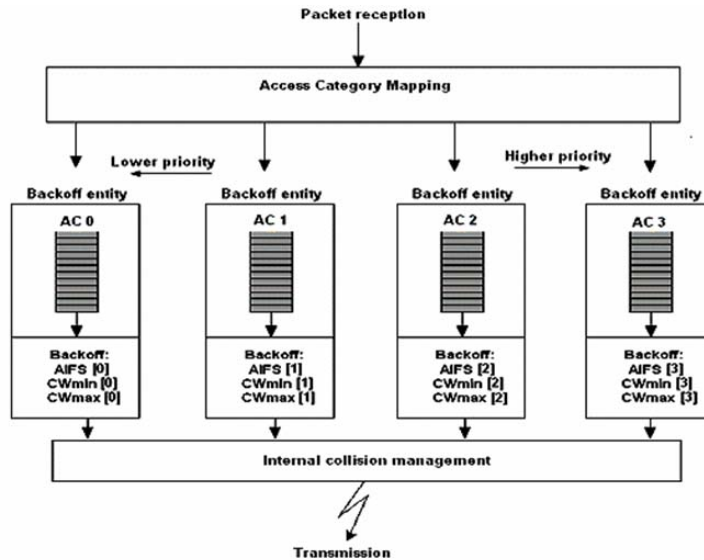
在 802.11p 網路中有兩個重要的特性：

CCH/SCH

在 MAC 層，WAVE mode 把頻寬分為一個 control channel(CCH)和許多的 service channels (SCH)。CCH 可以用來傳輸 WAVE short message (WSM) 以及告知有哪些 WAVE services。SCH 則是用來傳輸應用程式的資料封包。根據 draft standard，CCH 和 SCH 的 frame duration 都是 50 milliseconds。

QoS

WAVE 的 MAC 層有 QoS 的機制，messages 分成四成不同的 AC (Access Category) priority，分別對應到四個 queue，其中 AC0 優先權最低，AC3 最高。如下圖十一所示。

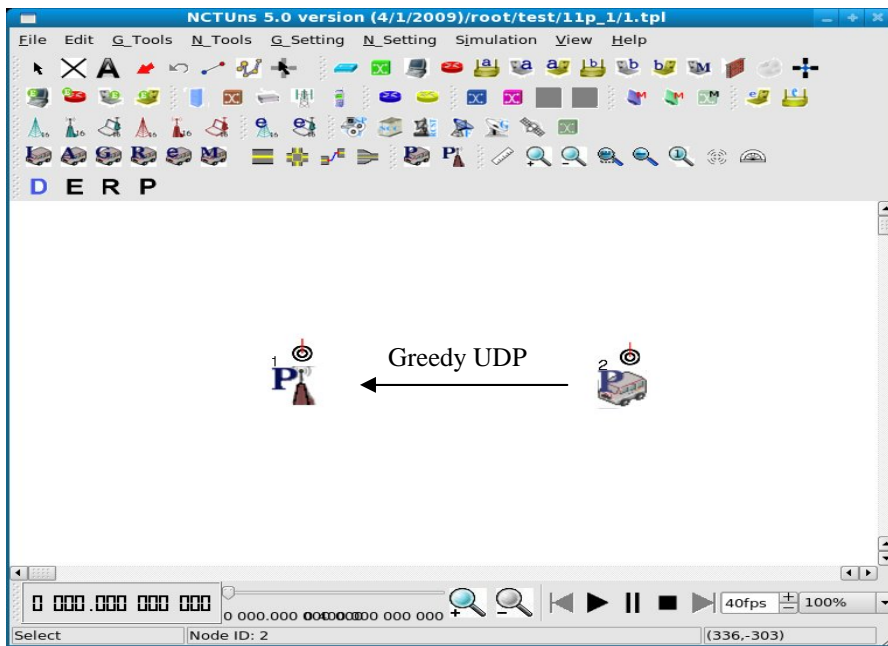


圖十、priority queue

底下我們將說明四個 case。case1 和 case2 是模擬 CCH/SCH 的切換對 throughput 和 latency 的影響。case3 和 case4 是模擬 priority queue(QoS)對 throughput 和 latency 的影響。藉由這四個 case 的模擬驗證在 NCTUns 實做 802.11p 是否符合標準。

Case 1

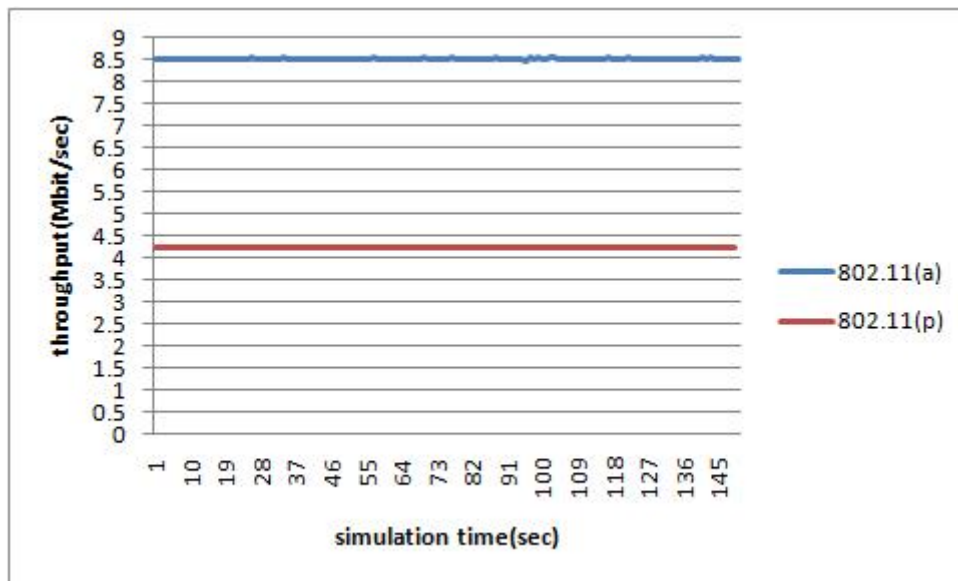
如圖十一所示，模擬的網路拓撲裡有一個 802.11(p) OBU(IP:1.0.1.1)和一個 802.11(p) RSU(IP:1.0.1.2)，其中 OBU 是 802.11(p) provider，RSU 為 802.11(p) user。在這個模擬 case 裡，為了比較 CCH/SCH 對頻寬的影響，在實體層(physical layer)把 phy data rate 都設為 12 Mbps 的環境下，我們使用 greedy(在 NCTUns 上我們可以使用 stg 以及 rtg 這兩支程式) UDP 從 RSU 傳輸封包到 OBU，因此 RSU 的設定為“stg -u 1000 100 1.0.2.1 -p 2007”，OBU 設定為“rtg -u -p 2007”。(-u 代表 greedy udp, 1000 是 packet size, 100 是傳輸時間。)



圖十一、case1的模擬拓撲

我們另外測試一種沒有CCH/SCH的模擬當作對照，因此我們選擇和802.11(p) 實體層類似的802.11(a)來比較，也是用兩個ad hoc mobile node，頻寬都設為 12 Mbps，並使用greedy UDP 來傳輸封包。圖十二為模擬的throughput比較結果。

由模擬結果(圖十一)可知，在頻寬為 12Mbps 的網路下，802.11(a) greedy UDP 的 throughput 大約為 8.5Mbit/sec，而 802.11(p)網路由於有 CCH/SCH 的切換，data 傳輸最多只能用到一半的頻寬，因此可以看到整體的 throughput 只有 4.2 Mbit/sec，這個結果也驗證了 CCH/SCH 的切換對 throughput 的影響。



圖十二、802.11(a)和 802.11(p)的 throughput 比較

Case2

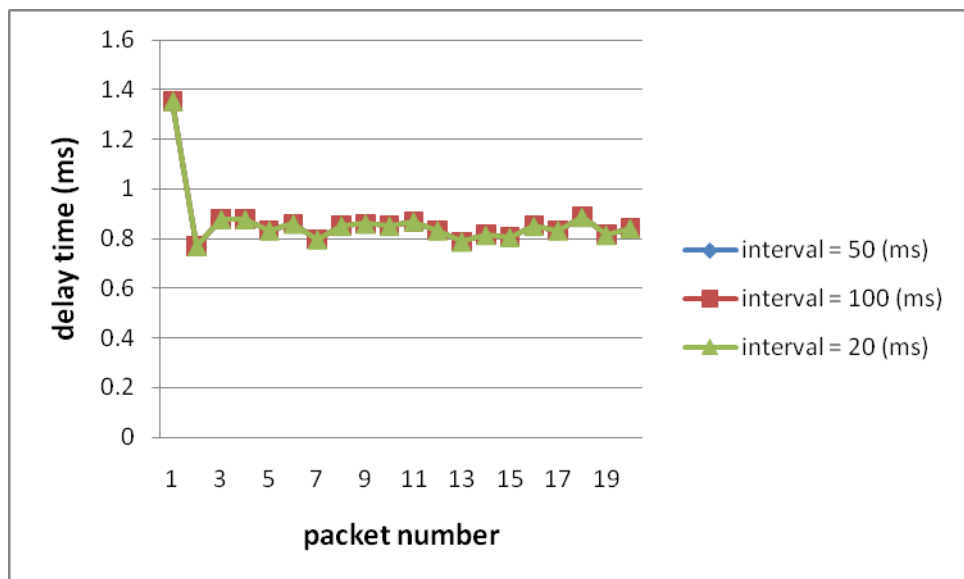
我們使用和 case1 一樣的網路拓撲而且也是使用 greedy UDP 傳輸並記錄 packet delay time，由於 greedy 會不斷的猛送封包，可能造成 packet queue 的長度一直增加使得封包的 delay time 受到 queue delay 的影響，為了避免這種情形，我們在使用 stg 程式時需額外吃設定檔來控制每個封包的傳輸時間，因此 RSU 的設定為 “stg -i udp.cfg 1.0.2.1 -p 2007”，OBU 設定為 “rtg -u -p 2007”。(-u 代表 greedy udp)。其中 udp.cfg 設定如下：

```
type: udp
  2 start_time: 0
  3 on-off: 1
  4
  5 on:   time: 60 const 0.05 length: const 1000
  6
  7 end
```

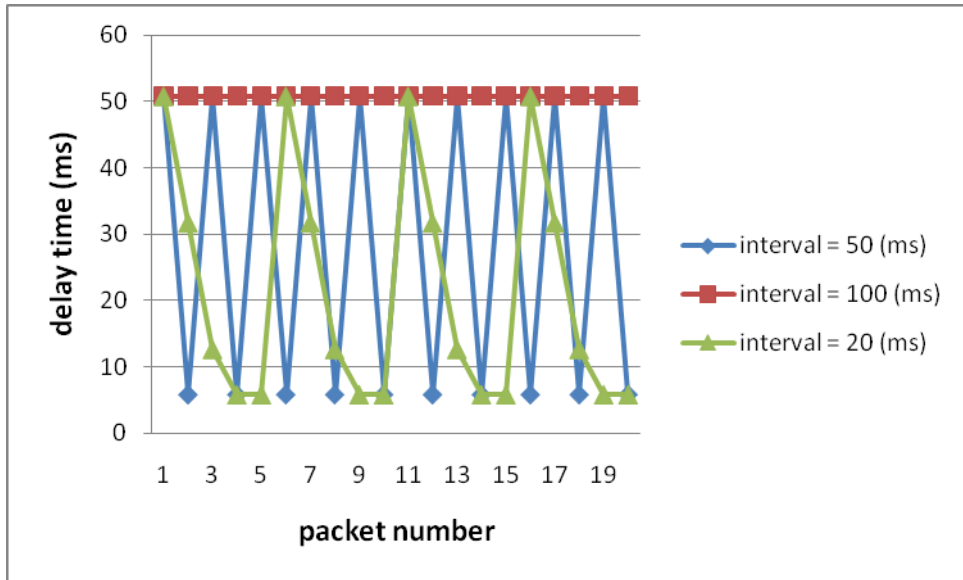
udp.cfg

Time 60 const 0.05 是指總共執行 60 秒，const 是 CBR(const bt rate)，後面接的 0.05 代表每 0.05 送一次。

我們先跑 802.11(a) 並分別調整 packet 傳送的時間間隔為 50ms，100ms，20ms，結果如圖十三所示。packet delay time 平均大概在 0.8 毫秒左右，而第一個 packet 的 delay time 會比較高是因為 802.11(a) 的 IP 和 MAC address 的對應是利用 ARP protocol，一開始 sender 並不知道 receiver 的 MAC address，因此要建立 ARP table 需要花比較長的時間。

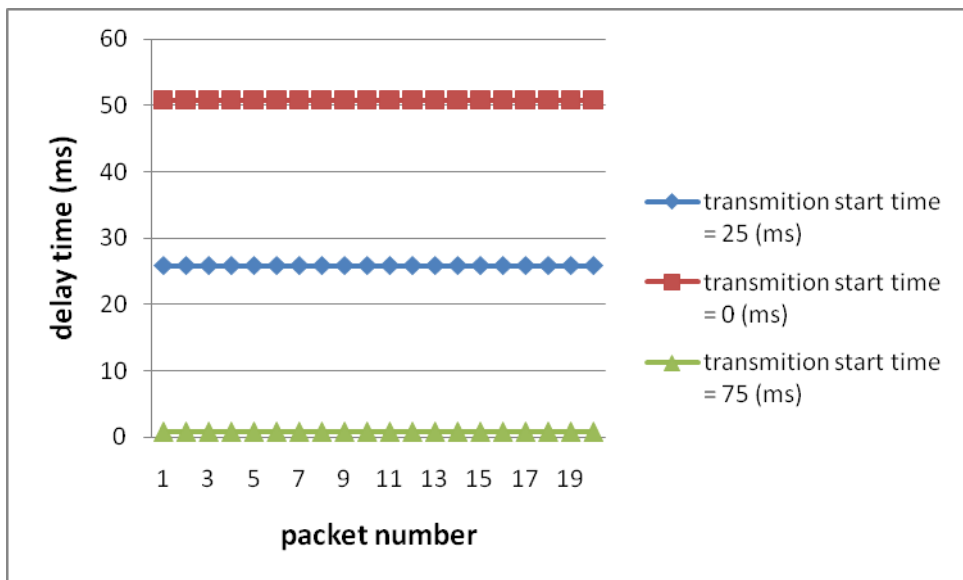


圖十三、802.11(a)的 delay time

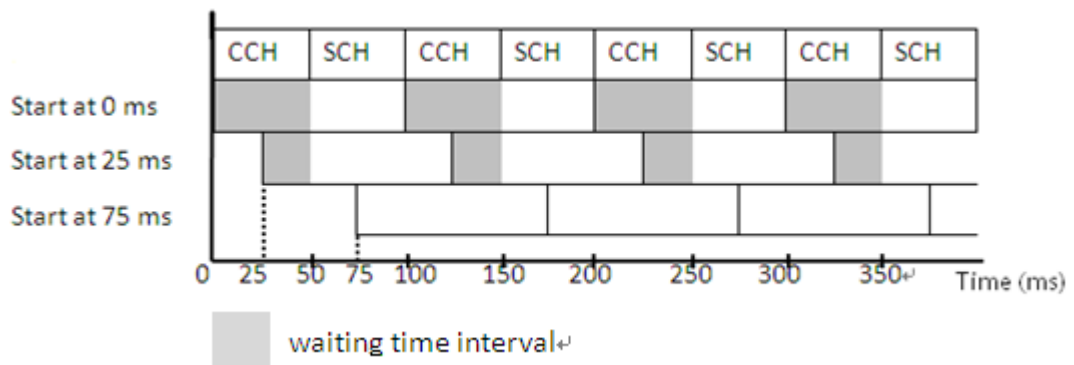


圖十四、802.11(p)的 delay time

圖十四顯示了在 802.11(p)的網路下，透過不同的 packet 傳時間間隔所造成的 packet delay time。由於 CCH/SCH 的切換時間為 50ms，因此當我們將傳送時間間隔調成 50ms 時剛好有一半的機率會碰到 Control Channel 因此 delay time 會來到 45ms，若剛好在 Service Channel，則幾乎不會有 delay time。當我們將傳送時間間隔調成 20ms 時大概有 20%的機率會碰到 Service Channel，因此如綠色線條所示。而當我們將傳送時間間隔調成 100ms 時，也就是剛好等於一個 CH+SH，此時 packet 不是都在 CH 就是都在 SH，圖十五及圖十六即針對傳送時間間隔為 100ms 而起始傳送時間的分析。



圖十五、802.11(p)傳送時間間隔為 100ms, 不同傳送起始時間的 delay time



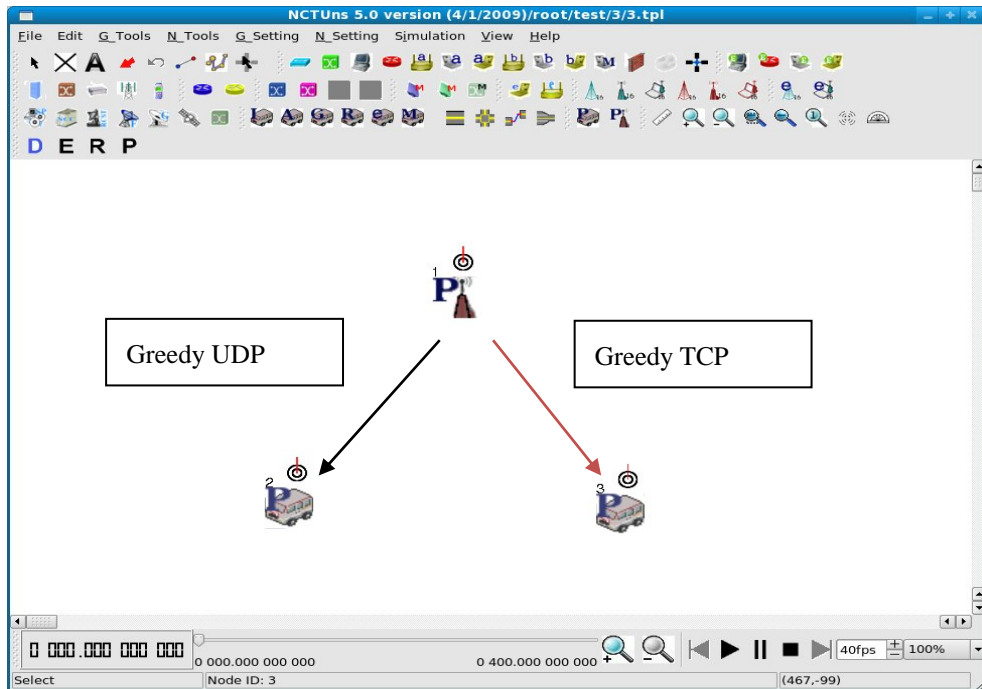
圖十六 不同傳送起始時間，相同傳送間隔之delay time 分析圖

如圖十六所示，當傳送起使時間為 0 時，恰好在 CCH 的開端，所以要等待一個 CCH 的時間，即 50 ms，剛好對照到圖十五的紅線。而當傳送起使時間為 25 ms 時，恰好在 CCH 的中間，所以只要再等 25ms 就可以到 SCH，對照到如十五的藍線。而當傳送起使時間為 75 ms 時，恰好在 SCH，data 可以不用等待就送出去，因此 delay time = 0，對照到圖十五的綠線。

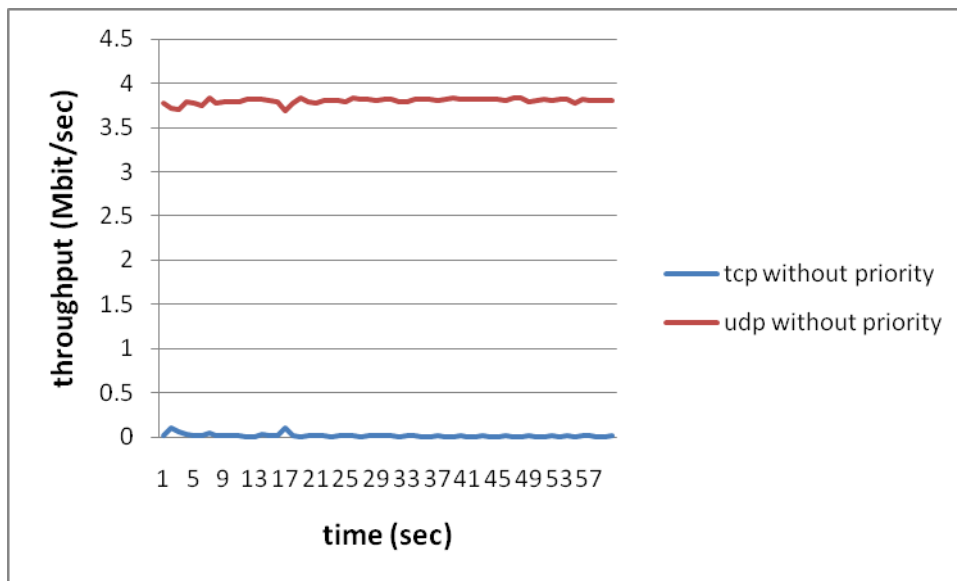
Case 3.

模擬的網路拓撲裡有一個 802.11(p) RSU(IP:1.0.1.1)和兩個 802.11(p) RSU(IP:1.0.1.2 和 1.0.1.3)，其中兩個 OBU 是 802.11(p) user，RSU 為 802.11(p) provider。這個 case 要展示的是實做 priority queue 對 throughput 的影響。左邊的 traffic flow 是用 greedy UDP，右邊的 traffic flow 是用 greedy TCP。我們有提供 8 種 priority(0~7)分別對應到 port 2000~2007，2007 的 priority 最高。若沒有指定 port 就代表沒有 priority 的區別。圖十六和圖十七是這兩個模擬的 throughput 比較結果。我們將從兩種角度來分析 priority queue 對 throughput 的影響。

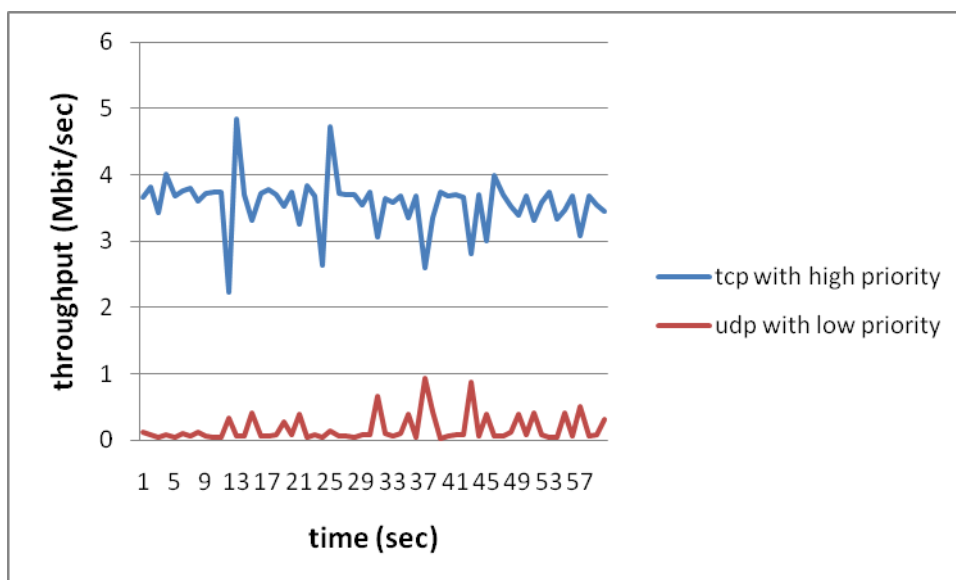
第一種情形如圖十七所示，RSU 對兩個 OBU 傳送封包，所以 RSU 須執行兩支 `stg(stg -u 1000 150 1.0.1.2 for 左邊的 OBU , stg -t 150 1.0.1.3 for 右邊的 OBU)`



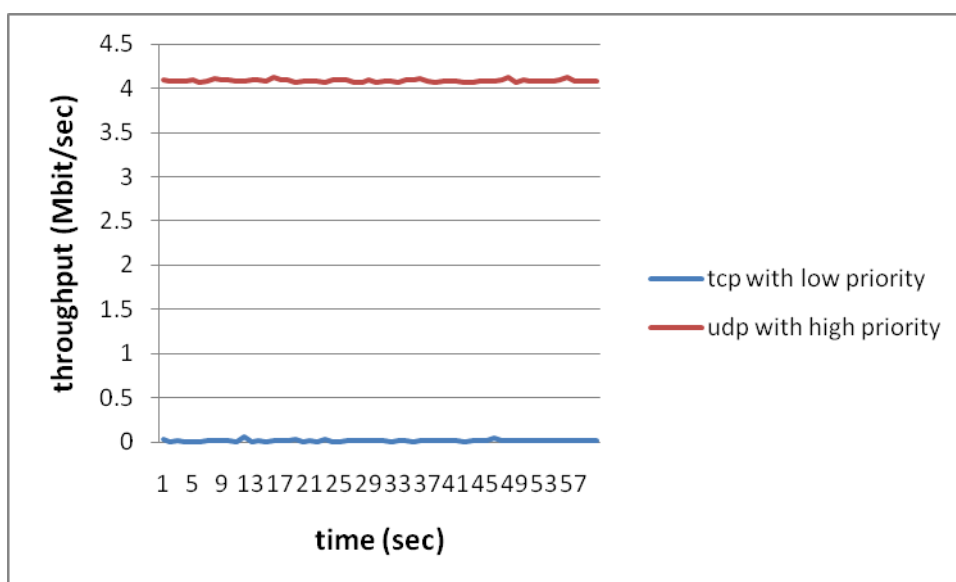
圖十七、priority 影響 throughput 的模擬拓撲



圖十七(a)、沒有 priority queue 的兩條網路流量 throughput 的比較



圖十七(b)、有 priority queue 的兩條網路流量 throughput 的比較



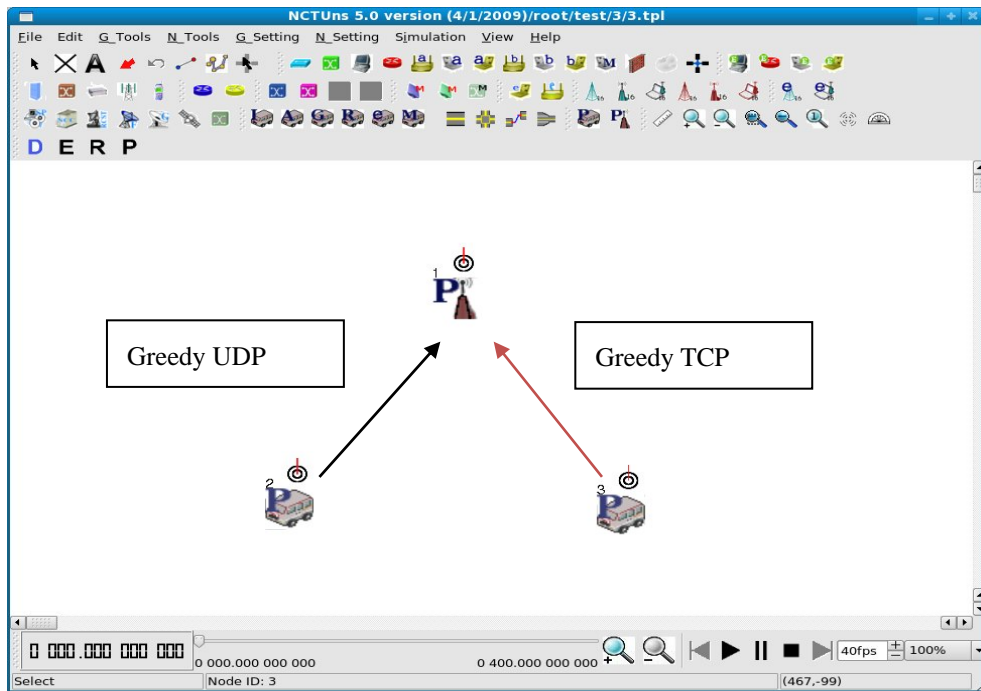
圖十七(c)、有 priority queue 的兩條網路流量 throughput 的比較

圖十七(a)是沒有用 priority queue 的結果，也就是說 greedy TCP 和 greedy UDP 這兩條 traffic 會同時塞在同一個 output queue，可以看到 TCP 幾乎搶不到頻寬，這是因為這兩條 traffic 開始時間一樣，一旦 UDP 開始猛灌封包就會把 queue 塞滿，如此一來 TCP 封包便會狂掉而開始降速，這樣惡性循環 TCP 更不可能搶到頻寬。若一開始 TCP 有搶到一點頻寬，輪到 UDP 開始灌的時候還是會發生上述的情形，因此 TCP throughput 會很低。

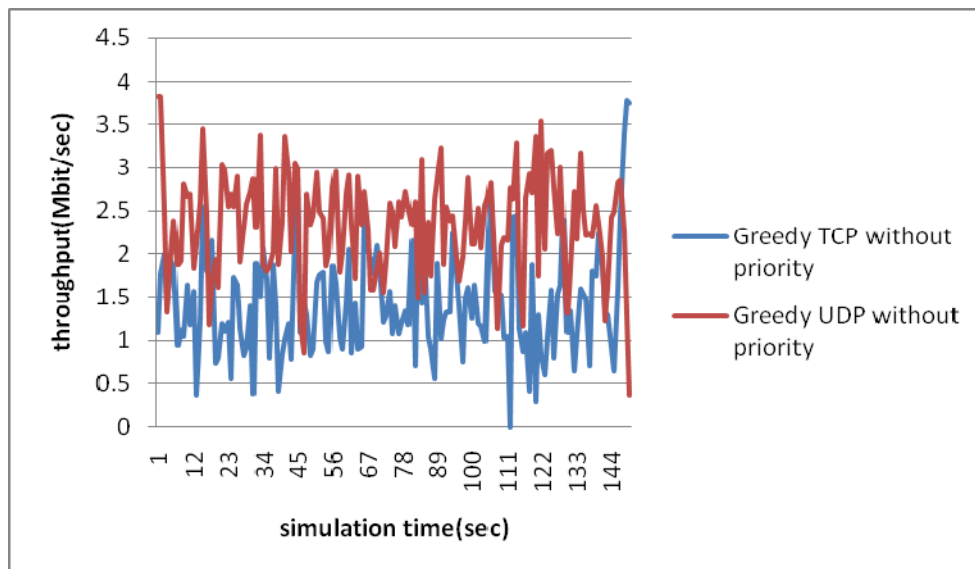
圖十七(b)是將 greedy TCP 給予最高的 priority (stg -u 1000 150 1.0.1.1 -p 2007)，而 UDP 則給予最低的 priority (stg -u 1000 150 1.0.1.1 -p 2000)，可以看到 TCP 的 throughput 可以維持大概在 3.8Mbit/sec，中間有一些地方 throughput 會狂掉是因為我們使用 greedy TCP 猛灌可能會造成 TCP 封包被 drop 掉而使 TCP 降速。

圖十七(c)是給 UDP 較高 priority，因此 UDP 可以很穩定的搶到接近滿的頻寬。

第二種情形如圖十八所示，左邊的 traffic flow 是用 greedy UDP，由 OBU 送到 RSU，右邊的 traffic flow 是用 greedy TCP，由 OBU 到 RSU。

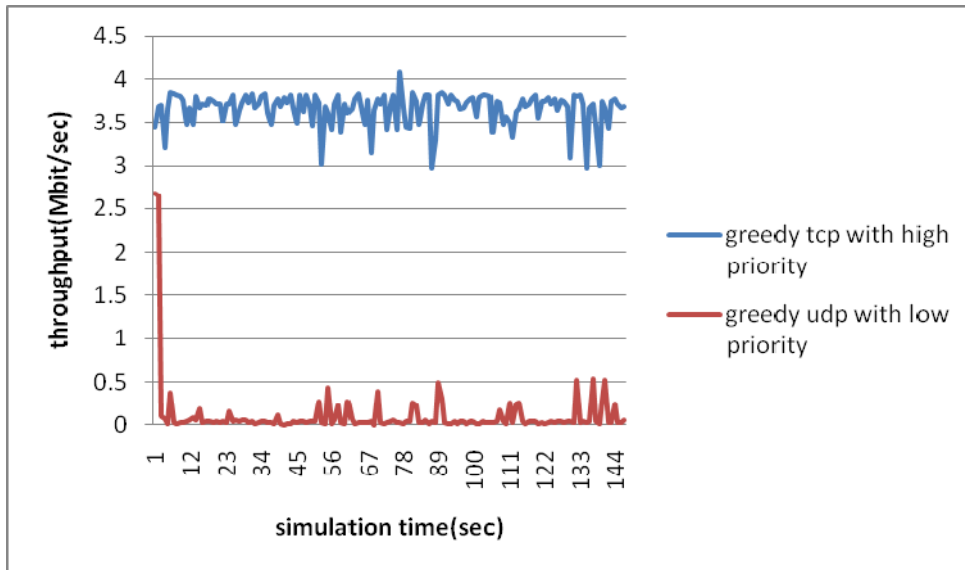


圖十八、priority 影響 throughput 的模擬拓撲



圖十八(a)、沒有 priority queue 的兩條網路流量 throughput 的比較

圖十八(a)是沒有用 priority queue 的結果，但是這兩條 traffic 的發送端不同，所以是在不同的 queue，因此圖中的折線是代表這兩個 queue 中的封包再傳送前要先去聽 MAC 層有無人在送，有的話要 random back off 一段時間。

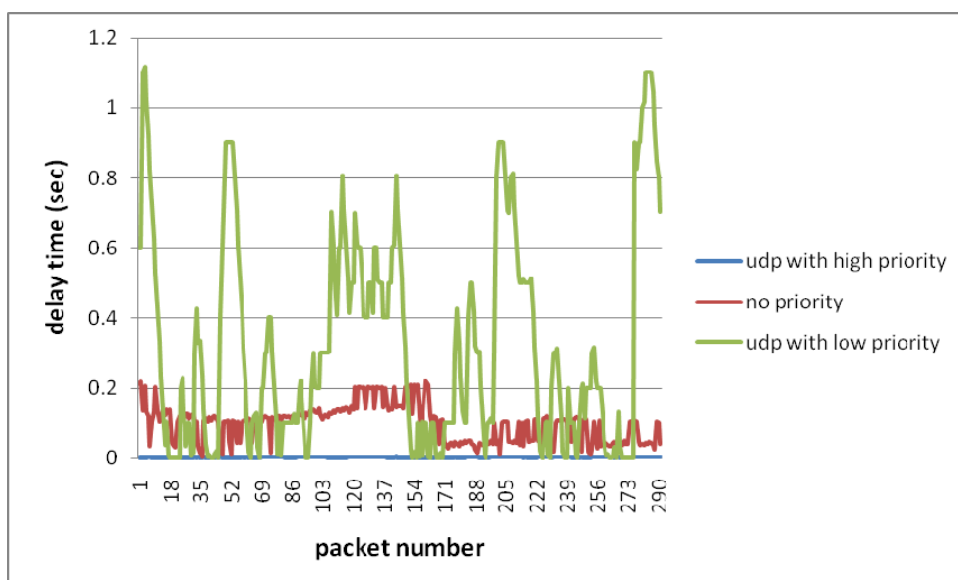


圖十八(b)、有 priority queue 的兩條網路流量 throughput 的比較

圖十八(b)是將 greedy TCP 給予最高的 priority (stg -u 1000 150 1.0.1.1 -p 2007)，而 UDP 則給予最低的 priority (stg -u 1000 150 1.0.1.1 -p 2000)，如此一來右邊 traffic 的 queue 的 priority 比較高，因此等待的 DIFS(Distributed InterFrame Space or DCF InterFrame Space)時間比較短，所以幾乎都是 TCP 搶到頻寬。

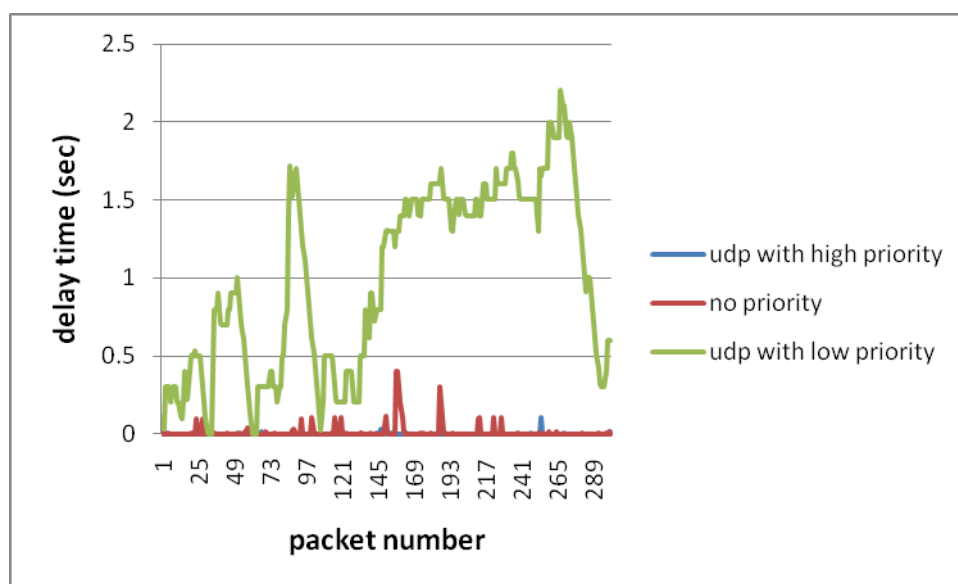
Case 4.

我們使用和 case3 一樣的網路拓撲，也是分兩種情形來分析，但這個 case 主要是測 priority queue 對 latency 的影響。這邊我們討論的是 UDP 的 packet delay time。也就是說我們將比較 greedy TCP 和 greedy UDP 都沒有 priority 的 UDP packet delay time 和 greedy TCP 使用 low priority(port=2000) greedy UDP 使用 high priority (port=2007)以及 greedy TCP 使用 high priority (port=2007) greedy UDP 使用 low priority(port=2000)的情形下 UDP 的 delay time。此外為了避免 greedy 猛灌所形成的 queue delay，這邊也是使用 config 檔來控制封包傳輸時間，我們將起始傳送時間設為 0.05 秒，封包傳輸間隔為 0.1 秒。結果如圖十九(a)及圖十九(b)所示。



圖十九(a)、不同 priority 的兩條網路流量 delay time 的比較

圖十九(a)是針對圖十七所做的 delay time 分析，我們可以看到，當沒有分 priority 時，packet delay time 大約為 0.1 秒左右，當 UDP 有較高 priority 的時候，在 output queue 競爭時就會搶贏 TCP 封包，因此 delay time 接近 0，而當 UDP priority 較低時，就會搶輸 TCP，因此平均 delay time 大概在 0.5 秒左右。



圖十九(b)、不同 priority 的兩條網路流量 delay time 的比較

圖十九(b)是針對圖十八所做的 delay time 分析，在沒有分 priority 的情形下，雖然這兩條 traffic 是不同 queue，但要送往同一個目標，因此會在 MAC 競爭，一但 UDP 搶到頻寬，TCP 就不能送，造成 TCP 降速，所以 UDP delay time 可以接近 0，當 UDP priority 比較高的時候就更容易搶贏，delay time 更低，而當 UDP priority 比較低時就要等 TCP 傳完才能傳因此 delay time 會比較高。

六、結論

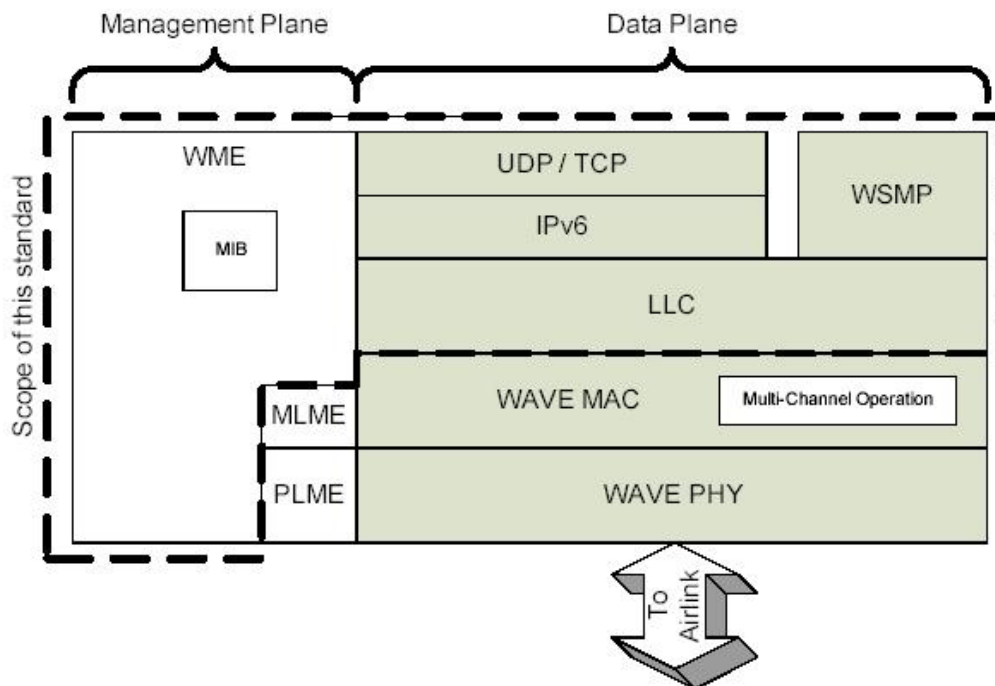
IEEE society 已經成立 802.11p working group 來制訂適合車輛無線網路 (vehicular wireless networks) 的通訊協定及系統。由於車輛無線網路是近幾年來剛興起的熱門研究領域，目前市場上無任何產品可供實驗進行研究，因此開發一個能模擬此類網路的軟體網路模擬器是非常重要的。我們在 NCTUns 上開發 IEEE 802.11p 網路協議，提供使用者有一個容易上手且高品質網路模擬器來進行功能驗證及效能評估。如此不僅能減少實體成本花費還能符合安全上的考量。

經過實驗的驗證，我們可以看出利用 NCTUns 模擬 IEEE 802.11p 的結果符合兩個重要特性：第一個是 CCH(control channel)/SCH(service channel)，根據 draft standard，CCH 和 SCH 的 frame duration 都是 50 milliseconds；第二個是 QoS，MAC 層分成四個 queue 分別對應到不同 priority，其中 AC0 優先權最低，AC3 最高。所以使用者可以放心的利用 NCTUns 來進行 IEEE 802.11p 的模擬研究。

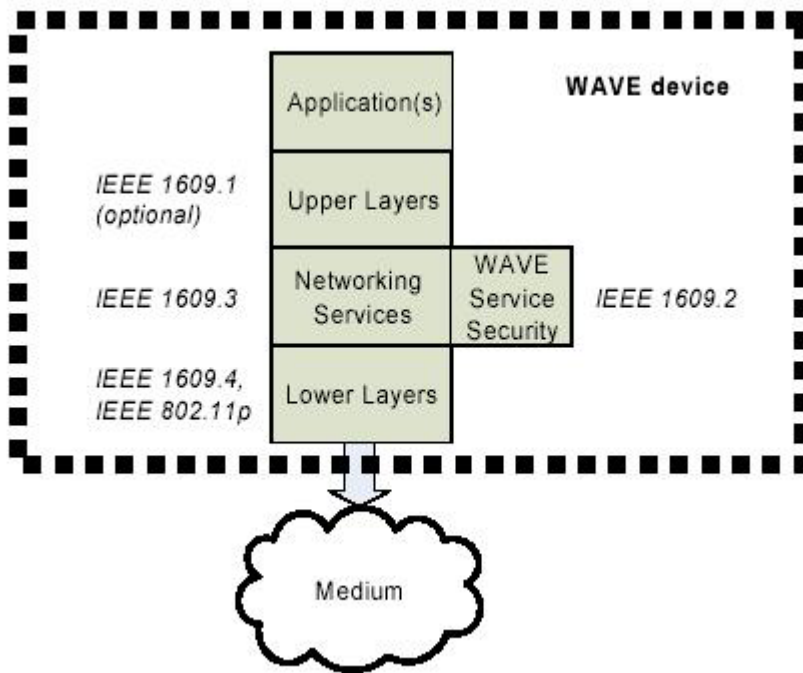
七、明年（第二年）工作計畫

在計畫第二年期，我們會承續第一年在模擬器平台上開發的 802.11p 車間網路通訊規範，將 IEEE 針對無線網路車間通訊所訂定的 1609 四個子標準和 802.11p 結合並在既有的 NCTUns 模擬器平台上開發這些通訊協定規範。802.11p 為智慧型基礎建設 (Intelligent Infrastructure, IF) 的後線系統與智慧型車輛 (Intelligent Vehicles, IV) 的接口，IF 的訊息透過 WAVE 與 IV 做連結，兩者之前相互交換訊息，IV 與 IV 之間也可以相互溝通。WAVE 既然是 IF 與 IV 的接口，所以 IF 與 IV 間和 IV 與 IV 間三者的溝通方式與內容，那麼就必須訂定一套完整的標準，這套標準就是 IEEE 1609。

其中 1609.1 部分針對 WAVE 無線網路存取環境中資源管理 (Resource Management) 進行規範，規定處理許多同時存在的資料流，記憶體管理與其他資源管理間的控制互換流程。1609.2 則是定義了應用和管理資訊的安全服務，規範在 WAVE 無線網路存取環境中通訊協定的安全性，確保通訊不被旁聽，干擾與其他的攻擊行為。1609.3 詳細地說明 WAVE 的網路服務與通訊協定，定義了支援該無線連接的網路和傳輸層中的服務(如圖十七)。而 1609.4 則制定了多頻道的管理規範，說明在 WAVE 系統架構下管理控制頻道 (Control Channel, CCH) 和應用服務頻道 (Service Channel, SCH) 間的通訊規範。在第二年前半期中，我們預計詳細研讀 1609 四個主要規格且分析評估其中適合開發於模擬測試平台上的部分。由於 1609 標準是由 IEEE 為了達成在高速行進間無線網路通訊的標準所制定出來的一套新的規範，其所涵蓋的範圍上從應用層，下至實體層(如圖十八)為一架構相當完整的規範。



圖十七、WAVE Protocol Stack



圖十八、IEEE 1609 WAVE Standard

在第二年前半年中，我們將會把 1609 的四套子規範於模擬器平台整合開發實作出來，因為 IEEE 1609 整體架構非常複雜且完整，所以我們預計將整個規範依照其四套子規範分成四個部分，各別分配一個人去熟讀其內容，然後於每週召開固定的會議，評估和討論各子規範中適合開發於模擬器系統平台上的規範，勾勒出結合 802.11p 和 IEEE 1609 的基本車間網路通訊架構，再將此架構開發於本實驗室的 NCTUns 模擬器系統上，並由開發人員進行初期的除錯及驗證。

在後半年中，我們將會將前半年所開發出來的四套子規範整合鏈結成一套完整的系統，然後再經各開發人員相互合作進行整套規範正確性的除錯和驗證，並對於此套系統在模擬器平台的效能作評估，以求達到利用最小的資源達到最大的模擬效率。並結合模擬器系統上的 GUI 圖形介面，然後在模擬平台上模擬各種組合的測試，再交叉比對所有的模擬數據各種結果，以數學方法去分析所統計出來的模擬數據，並驗證其正確性是否符合規範上所訂定的結果。

開發此套模擬系統最大的難度在於如何正確地整合 IEEE 1609 的四項子規範，並且在整合後進行除錯和驗證。我們預計會每週定期召開會議，由各子規範開發人員對各自負責的部分進行討論和講演，讓大家能夠在最有限的時間內清楚了解各層規範的相關性，並對於規範中有關聯的部分進行評估，避免在整合後期發生必須變更整個大架構的情形，以求達成事半功倍的效能。且目前 IEEE 1609 規範尚在審核階段，未來還是有可能會有相關的新規範會被通過，如第五項標準 IEEE P1609.0 作為一項架構文件正在籌備中，其中將對 WAVE 系統和其組件及運行進行整體的介紹。所以，如何將模擬模組設計的更有彈性以利於日後系統變更和新增規範也是設計及實作整套模擬系統必須詳加考慮的要件。所以，我們必須將各個開發模組明確的訂定區分出來，以確保其關聯性和獨立性。