

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 (2/3)

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：2009 年 8 月 1 日至 2010 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 99 年 5 月 31 日

摘要

智慧型運輸系統(ITS)是近年一項備受矚目的重要議題，其藉由應用車輛無線網路，提供即時的資訊而達到增進運輸系統的安全，效率與舒適性。然而舊有的 IEEE 802.11 系列標準已不適用於車輛無線網路此一新興的網路型態，因此 IEEE 訂定了 802.11p/1609 標準，使其能符合智慧型運輸系統的相關應用。由於智慧型運輸系統與車用無線網路在商業發展及學術研究上的巨大潛力，有越來越多團隊投入此一研究領域。但目前市場上並沒有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。

本計畫去年度已完整實作 IEEE 802.11a/p 於 NCTUns 上，今年度則成功實作 IEEE 1609 於 NCTUns 上，並與去年的結果整合為一套完整的車用無線網路模擬系統。

關鍵字： ITS, IEEE 802.11p, IEEE 1609, NCTUns, 模擬

Abstract

Intelligent Transport Systems(ITS) has been a very popular issue in recent years. It improves the safety, efficiency and comfort of transport systems by real-time information obtained from Vehicular Wireless Networks(VWN). And the original IEEE 802.11 standard is not suitable for the novel network type. IEEE society defined IEEE 802.11p and 1609 standard to suit the special needs of VWN. Because of the massive potential of ITS and VWN in commercial development and academic study, they have gained intensive research interest. However, there are no products available on the market for doing ITS experiments. As such, developing a high-quality network simulator for such a new network is extremely important and useful.

We have already completely implemented IEEE 802.11a/p standard on NCTUns last year, and for this year, we successfully implemented IEEE 1609 standard and integrated them to form a complete VWN simulation system.

keywords : ITS, IEEE 802.11p, IEEE 1609, NCTUns, simulation

1. 前言

智慧型運輸系統(Intelligent Transport Systems, ITS)是近年一項備受矚目的重要議題，其藉由應用車輛無線網路，提供即時的資訊而達到增進運輸系統的安全，效率與舒適性。而對於車輛無線網路此一新興的網路類型，舊有的 IEEE 802.11 系列標準並不適用於高速移動物件上的無線通訊，於是 IEEE society 成立了 802.11p working group 制訂 IEEE 802.11p 通訊協定(又稱 WAVE, Wireless Access in the Vehicular Environment)，其由 IEEE 802.11a 擴充延伸而得，以符合 ITS 的各種相關應用。以 802.11p 作為底層的通訊協定，IEEE society 隨後制訂較高階的 1609 標準。它可細分為四個子標準。

1609.1：針對 WAVE 無線網路存取環境中資源管理進行規範，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：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

1609.4：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。

2. 研究目的

車輛無線網路為一新興的網路類型，很多為此類網路設計的通訊協定設計與應用服務不斷發表，但目前在市面上並無任何產品可供實驗進行研究，對於此一領域的開發者可說十分不便，若能提供根據官方規格制定並具有公信力的模擬器，此類網路相關的研究便能在其上進行功能驗證與效能評估，有效提升車輛無線網路領域的研究環境。再者使用模擬進行研究，並不需要去租借多部車輛與購買多套硬體設備，也不需多位實驗人員，成本比起真實道路實驗要小的多；而在安全方面考量，模擬研究也比真實道路實驗來的安全。無論就成本，效率亦或安全性方面考量，開發一可靠的模擬器都是重要且勢在必行的。

在前一年的計畫裡，我們已經成功地將 802.11a/p 實作於 NCTUns 網路模擬器上，這一年的目標就是在 NCTUns 上實作 IEEE 1609 並將 802.11p 與 1609 的四個子標準作結合，形成一套完善的車輛無線網路模擬系統。

WME 透過 primitive 與 higher layer 以及 MAC layer 溝通，但是我們實作的 WME 並沒有完整的支援 WME 與 higher layer 間的溝通。目前只能支援 application 單向的透過 primitive 來告知 WME WAVE service 的訊息 (例如 provider service 的新增)，其原因要從 WAVE standard 的發展歷史來說明。一開始 IEEE 1609.1 定義了 application 以及 WME 的中介層，稱為 resource manager，application 必須透過 resource manager 來存取 WAVE device 所提供的資源，而在 IEEE 1609.3 所說的 higher layer 正是 resource management。但是，IEEE 1609.0 在之後卻被部分的人認為制定失敗。雖然後來又制定了 IEEE 1609.5 的 communications manager 來取代 IEEE 1609.0 中 resource management，但是在實作 WAVE system 時 IEEE 1609.5 還未發表，因此我們捨棄了在 WAVE device 上 application 透過 higher layer 與 WME 間動態使用 primitive 溝通的功能。最後，我們選擇了折衷的辦法，那就是，一開始就必須設定好 higher layer 何時去使用 WME 提供的 primitives。實際的情形為使用者必須一開始就在指定的檔案(primitive file) 中說明 WME 要在何時執行何種 primitives。在我們的實作中只提供 WME-ProviderService.request，WME-UserService.request，WME-CchService.request 以及 WME-WSMService.request 四種較重要的 service request primitive。檔案的格式如圖 3.2 所示，SIB_Begin 以下 SIB_End 以上為設定檔的內容，所有 node 的 primitives 都設定在此檔案內。NID X 以下 NID Y 以上為 NID X 的 primitives 設定。CDB 與 CDE 中間所夾的內容就是一筆完整的 primitive 設定，其中時間 (Time) 的單位為 $10^{(-7)}$ second，表示 primitive 啟動的時間。

```

SIB_Begin
NID 1
CDB
    Time 20000000
    Primitive provider_service_req
    Action add
    PSID 1
    PSC ""
    AppPriority 1
    Channel 174
    Persistence 1
    Repeats 0
    IPService 1
    IPAddr 1.0.1.1
    ServicePort 1
    MacAddr 0:1:0:0:0:2
    RecipientMacAddr 0:1:0:0:0:3
CDE
CDB
    Time 10000000
    Primitive user_service_req
    Action add
    UserReqType auto_access_on_service_match
    PSID 1
    PSC "test"
    Channel 174
    SourceMac 0:1:0:0:0:2
    AID ""
    ImmediateAccess 1
    IndefiniteAccess 1
    Notify 1
NID 2
CDB
    Time 10000000
    Primitive wsm_service_req
    Action add
    PSID 1
CDE
CDB
    Time 10000000
    Primitive cch_service_req
    Action add
    AppPriority 1
CDE
SIB_End

```

圖 3.2 WME Primitive Setting File

一開始每個 node 的 WME module 會去讀取 primitive file，並且把屬於自己的 primitives 設定按照時間順序存在 queue 中，直到 primitive 的啟動時間到了，WME 就依照 primitive 的 type 去執行不同的 function。

所有 service 的新增，移除或是修改都是藉由 primitives file 所指定的 primitives 來運作。以下將詳細介紹這些 service request 的運作流程：

- Provider Service：

若 provider service request 的 Action 為 add，其代表的意思為 application 想提供 service，請 WME 幫助它完成。收到此 primitive 之後，WME 首先會將 WME-ProviderService.request 的內容記錄在 MIB 中並且開始準備發送 advertisement。如果 WAVE device 之前就是 provider 那麼 WME 就要修改之前的 WSA，並且在原本的 WSA 之後附加此 provider 的資訊，否則就要自己建立新的 WSA。WSA 建立完成之後就要根據 primitive 中所帶的 persistence 以及 repeats 來發送 WSA。最後便是告知 MAC 到達 SCH interval 時要跳到哪個 SCH。

若 provider service request 的 Action 為 change，其代表的意思為 application 想改變 service 的參數，請 WME 幫助它完成。此時 WME 會根據 primitive 的內容來修改 MIB 中相對應的欄位，並且修改已存在的 WSA 成為 primitive 所要的樣子。如果 SCH 有改變的話則要重新通知 MAC，另外 WSA 的發送頻率有改變的話也要做處理。

若 provider service request 的 Action 為 del，其代表的意思為 application 想停止 service 的提供，請 WME 幫助它完成。此時 WME 會根據 primitive 的內容來刪除 MIB 中相對應的欄位，並且將 WSA 中相應的 provider 訊息刪除。如果 WSA 內沒有其他的 provider 的訊息則停止發送 WSA，並且告知 MAC SCH interval 請留在 CCH。

- User Service：

WME-UserService.request 有新增與刪除兩種 action。新增表示 application 對於 primitive 所指定的 service (以 PSID 來區分) 有興趣，刪除則相反。收到 user service request 時 WME 一開始只會去修改 MIB 的欄位，之後如果 WME 收到 WSA 中包含 application 所感興趣的 service，WME 會自動加入此 service 直到此 service 不再提供，或是 application 對這類型的 service 不再感興趣為止。如果 application 對很多 service 都感到興趣的話，WME 會選擇最高 priority 的 service 加入。

- CCH Service：

WME-CchService.request 有新增與刪除兩種 action，用來新增或是刪除 CCH 的 priority。由於 CCH 只有一個，因此 WME 只會使用到最大的 priority。WME 收到此 primitive 時只會去修改 MIB 的欄位，直到 SCH interval 轉換到 CCH interval 那一瞬間，如果 WAVE device 想要繼續留在 SCH 它必須去比較目前所使用的 provider service 的 priority 是否比 CCH

priority 大，如果是則會在繼續留在 SCH 否則就會切換回 CCH。

- WSM service :

WME-WSMService.request 有新增與刪除兩種 action，新增或是刪除 PSID。WME 可以用 PSID 來判別 WSM 由哪一種 application 送出，application 可以藉此決定只收哪種 PSID 的 WSM。WME 收到此 primitive 時只會去修改 MIB 的欄位，直到 WAVE device 收到 WSM 時，WME 會根據 application 之前所註冊的 WSM 的 PSID，來決定要不要收此 WSM。

目前 WME 只能服務一個有能力收發 WSM 的 application，因此只要是 WSM service 有註冊過的 PSID 都會送到同一個 application，WME 並沒有 route WSM 到正確 application 的能力。使用這種作法的原因是因為在實做 WME 當時考慮到 WAVE 標準對於 application 的定義並不明確。WME 必須要維護 available service。在此使用的方法為收集其他 WAVE device 所發出的 advertisement，如果在此次 CCH interval 有收到關於某 service 的 advertisement 表示此 service 可以使用，WME 會在 CCH interval 結束時搜尋 MIB 中記錄的 available service，如果 available service 中有對應到 application 使用 user service request primitive 註冊的 service 則 WME 會自動加入此 service，詳細的運作情況如圖 3.3 所示。

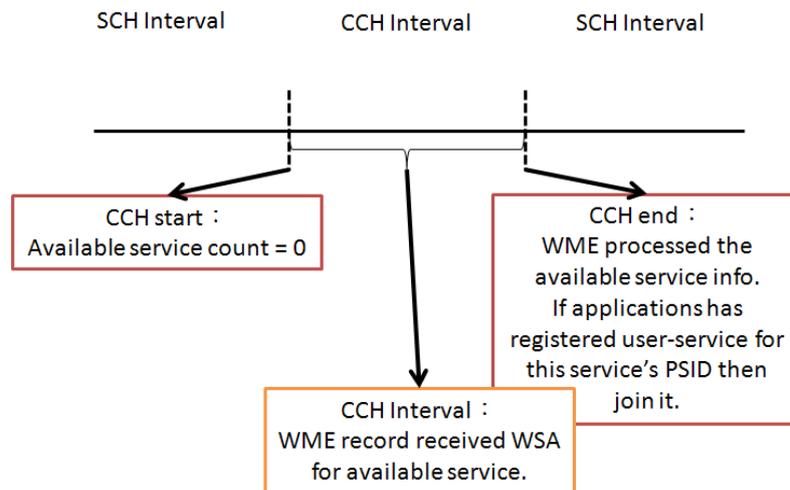


圖 3.3 Recording Available Services in CCH intervals

IEEE 1609.3 定義了 WAVE system 發送 IP 封包的條件限制，這個機制目前實作在 WME 中。其實際運作情況如圖 3.4 所示，如果 IP 封包是由 higher layer 送往 WME，則 WME 會檢查目前 WAVE device 是否有加入或是提供 service (user or provider)。若 WAVE device 目前為 user 則只能將封包送往提供自己 service 的 provider，若 WAVE device 為 provider 則直接將封包送出去，發生其他的情況則直接將封包丟棄。如果 IP 封包是由 lower layer 送往 WME，

WME 同樣要檢查目前 WAVE device 是否有加入或是提供 service (user or provider)。若 WAVE device 目前為 user 則只能收取自己 provider 所發出的封包，若 WAVE device 為 provider 則直接將封包向上送，其他情況則直接將封包丟棄。

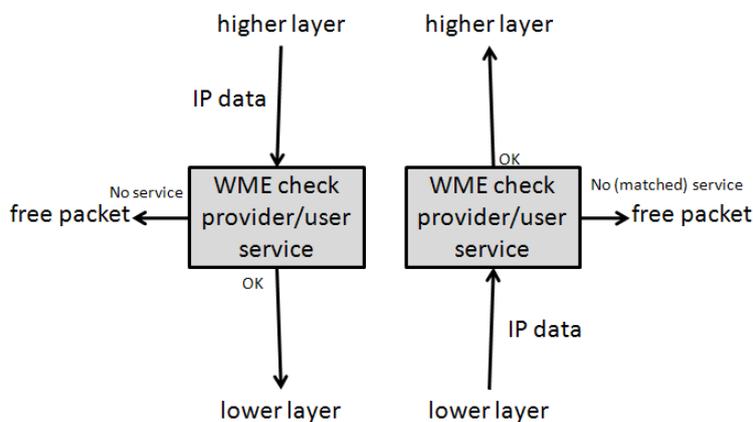


圖 3.4 The Processing of IP Packets in WME

WSMP Implementation :

WSMP 是專門為了 WAVE system 所設計的 protocol，它與 IP protocol 的地位相同。在 NCTUns 所使用的 IP protocol 是修改 Linux kernel 內建的 IP protocol 而來，但是 Linux kernel 並沒有內建 WSMP，因此必須自行實作。為了讓 WAVE applications 可以透過 function call 的方式來發送 WSM，以下有兩種方法來實作 WSMP。第一種方法比較直覺，將 WSMP 實作在 Linux kernel 中，但是缺點是實作的難度較高以及每次 Linux 改版都要重新維護。第二種方法是將 WSMP 實作在 NCTUns 中，application 要發送 WSM 時利用 IPC 將 WSM 送至 NCTUns 內。

在此使用上述的第二種實作方式。WSM 收送的情形如圖 3.5 所示，當 application 要發送 WSM 時，其會透過 IPC 將 WSM 送至相對應的 WME，之後再由 WME 將 WSM 送出。當另一方的 WME 收到 WSM 時，它會透過 signal 來通知 application，最後 application 透過 IPC 讀取 NCTUns 內的 WSM。

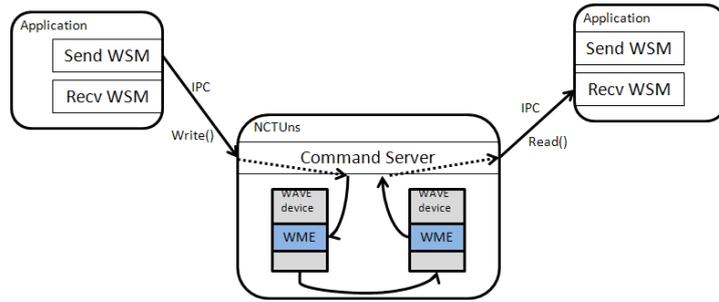


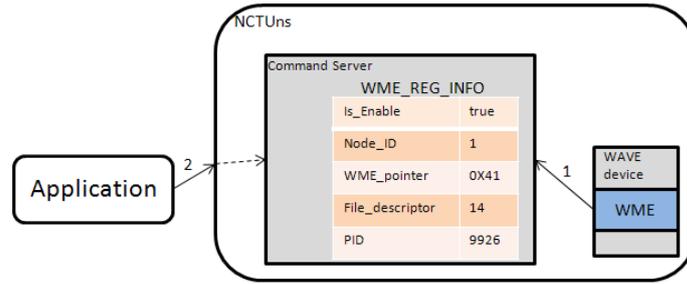
圖 3.5 Send/Receive a WSM

NCTUns 透過其內的 command server 來收送 WSM，之後再由 command server 來將 WSM 送至相對應的 WME module 或是 application，也就是由 command server 來負責做 IPC。

NCTUns 在開始模擬前會先將其所有將要運行的 module 初始化，接下來才會開始 fork 出即將要在 NCTUns 上模擬執行的 process。為了要讓 application 將 WSM 送到正確的 WME，以及 WME 將 WSM 送到正確的 application，因此我們利用 NCTUns 會先初始化完所有的 module 後才會 fork process 的特性。

其細節如圖 3.6 所示，在 WME 作初始化時先在 command server 內建立一個資料結構 WME_REG_INFO，記錄 WME 所在的 node id 以及 WME module 所在的位置。當 application 被 fork 出來時 application 會透過 IPC 向 command server 告知此 application 是由哪個 node 所執行，application 和 command 間溝通所利用的 file descriptor 以及 application 的 process id (PID)。Command server 會根據 application 所給的訊息 (node id) 去找相對應的 WME_REG_INFO，找到之後便將 application 的 PID 紀錄在內。如此，WME_REG_INFO 內便有 WME 以及 application 的共同資訊，以及個別資訊 (共同資訊為 node id，個別資訊為 WME module 位置以及 application 之 PID)。Application 想發送 WSM 時先透過 IPC 傳送給 command server，由 commander 去尋找相對應的 WME_REG_INFO，利用其內的資訊將 WSM 送至正確的 WME module。

WME module 想把 WSM 傳給 application 時，WME 直接將 WSM 交給 command server 處理，讓 command server 去尋找相對應的 WME_REG_INFO，利用其內的資訊將 WSM 透過 IPC 送至正確的 application process，並且發送一個 signal (kill(PID, SIGANL_TYPE)) 通知此 application 來收取 WSM。



1. WME constructed WME_REG_INFO. (IS_Enable = false, Node_ID = 1, WME_pointer = 0X41)
2. Application enabled WME_REG_INFO. (IS_Enable = true, File_descriptor = 14, PID = 9926)

圖 3.6 Construction of the WME_REG_INFO Structure

為了配合 WME 的設計因此所有想發送 WSM 的 application 都必須依照圖 3.7 的格式撰寫。首先 application 要先利用 IPC 開通和 NCTUns 間的通道，接下來要傳送註冊訊息給 NCTUns，並且要對 NCTUns 所發出的 signal 做處理 (signal handler)。Application 收到 NCTUns 所發出的 signal 表示 NCTUns 已經將 WSM 送到之前所建立的 IPC 通道，此時 application 要中斷目前程式的執行，並起去執行 signal handler 以收取 WSM。另外 application 只要把想送的 WSM 包裝成正確的格式，再透過 IPC 送給 NCTUns 便可完成 WSM 的發送。

```

void sig_usr1(int signo)           //signal handler function
{
    sigprocmask(SIG_BLOCK, &usr1_mask, NULL);
    //防止收取封包時遭到 signal 的中斷

    Recv_WSM();

    sigprocmask(SIG_UNBLOCK, &usr1_mask, NULL);
}

int Recv_WSM()
{
    read(file_descriptor, buffer, buffer_size);
    //透過 IPC 收取 WSM
}

int Send_WSM(data, ...)
{
    sigprocmask(SIG_BLOCK, &usr1_mask, NULL);
    //防止發送封包時遭到 signal 的中斷

    WSM_data = packing_WSM(data, ...);
    //將發送的 data 包裝成 WSM 格式

    write(file_descriptor, WSM_data, size);
    //透過 IPC 發送 WSM 給 NCTUns

    sigprocmask(SIG_UNBLOCK, &usr1_mask, NULL);
}

int main()
{
    createTCPSocketForCommunicationWithSimulationEngine(...)
    //開通和 NCTUns 間的連線

    Enable_WME_REG_INFO(...);
    //向 NCTUns 註冊 process 的資訊

    signal(SIGUSR1, sig_usr1);
    //註冊 signal 的 handler function

    ...
    Send_WSM(data, ...);
    ...
}

```

圖 3.7 WAVE Application for Sending or Receiving WSMs

4. 計畫成果自評

計畫第二年預期完成的工作項目條列如下：

- (1) 1609.1 (2) 1609.2 (3) 1609.3 (4) 1609.4

經過審慎的研讀與評估後，我們實作了 IEEE 1609 的四個子標準中的 1609.3 以及 1609.4 這兩個子標準，因為 1609.1 的 resource manager 與 1609.2 的 security 這兩個子標準並不會影響到模擬網路的效能評估，因此 1609 標準在 NCTUns 的實作上，已經開發完畢，此外我們還完成了 1609 與 802.11p 的結合，並且經過了正確性的驗證。有了這一套模擬系統，之後就可以利用 802.11p 與 1609 來模擬網路封包的傳遞、頻道的變換等，並測量其效能，藉由這些模擬，可以讓更多人做相關的研究。

5. 參考文獻

- [1] “Draft P802.11p/D3.0,” the IEEE 802.11 Working Group of the IEEE 802 Committee, July 2007.
- [2] “IEEE 1609.1 Trial-Use Standard for Wireless Accesses in Vehicular Environments(WAVE) – Resource Manager,” IEEE Vehicular Technology Society, October 2007.
- [3] “IEEE 1609.2 Trial-Use Standard for Wireless Accesses in Vehicular Environments(WAVE) – Security Services for Applications and Management Messages,” IEEE Vehicular Technology Society, October 2006.
- [4] “IEEE 1609.3 Trial-Use Standard for Wireless Accesses in Vehicular Environments (WAVE) – Networking Services,” IEEE Vehicular Technology Society, October 2006.
- [5] “IEEE 1609.4 Trial-Use Standard for Wireless Accesses in Vehicular Environments (WAVE) – Multi-channel Operation,” IEEE Vehicular Technology Society, October 2006.
- [6] “IEEE P1609.3TM/D1.0 Draft Standard for Wireless Access in Vehicular Environments (WAVE) – Networking Service,” IEEE WAVE Working Group, December 2008.
- [7] “IEEE P1609.4TM/D1.0 Draft Standard for Wireless Access in Vehicular Environments (WAVE) – Multi-channel Operation,” IEEE Dedicated Short Range Working Group, December 2008.
- [8] Wei-Jyun Hong, “Performance comparison between IEEE 802.11a-based and IEEE 802.11p-based wireless vehicular networks,” M.S. thesis, National Chaio Tong University, June 2009.