

## 智慧電網下以 IPv6 實行 SCTP 多重定址機制之研究

呂冠達<sup>a</sup>、吳坤熹<sup>a</sup>

國立暨南國際大學資訊工程研究所<sup>a</sup>

**摘要**— 隨著通訊技術的演進，無線網路成為了目前熱門的通訊技術之一。而智慧電網的建設中，大量仰賴低耗能感測網路來進行溝通，也致使相關的研究蓬勃發展。Stream Control Transmission Protocol (SCTP) 為新一代的傳輸協定，結合 TCP 與 UDP 的優點，再加上其他協定所沒有的機制，使得 SCTP 能滿足高性能傳輸的需求。本研究主要將 SCTP 運用到智慧電網的傳輸網路中，藉此提高網路的穩定度，並降低因網路中斷所造成的損失。<sup>1</sup>

**關鍵詞**：SCTP、智慧電網、無線網路。

### 一、研究背景

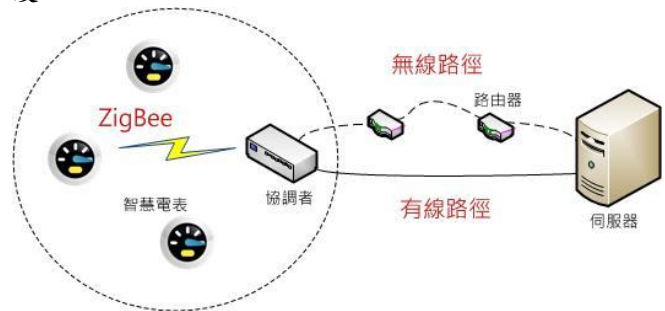
近年來，無線網路已逐漸滲透至人們的日常生活當中。繼 3G、WiMAX、LTE (Long-Term Evolution) 等長距離的無線通訊技術後，短距離無線通訊技術因其應用貼近於日常生活，儼然已成為當前發展的熱門焦點。其中，IEEE 802.15.4 就是針對此一應用所發展出的低功率、低速率和短距離傳輸的無線通訊標準。然而，由於 IEEE 802.15.4 只規定網路架構的底層，即 OSI (Open Systems Interconnection) 網路七層模型中所定義的實體層 (Physical Layer) 與資料鏈結層 (Data-Link Layer)，因此建構在資料鏈結層之上的其他階層可以選擇採用不同的通訊技術，進而產生不同的通訊網路，ZigBee[1] 和 6LoWPAN[2][3] 就是目前 IEEE 802.15.4 上較受歡迎的兩種網路協定。

智慧電網 (Smart Grid) [4][5]，為目前全球熱門的科技議題之一，透過智慧電網的技術可以有效的監控各地用電狀況，並且適度分配電量的傳送比例，達成資源的有效利用，進而減少某些地區用電量不足或是過度浪費的情形。智慧電網是一雙向溝通的輸電網路，即電力公司可以傳送電力與訊息給用戶端，而用戶端則會傳送用電訊息給電力公司，或更進一步選擇將剩餘的電力回賣給電力公司。它包含了一個智慧電表基礎建設 (Advanced Metering Infrastructure; AMI)，用來記錄所有電能的流動。在基礎建設中，假設選擇以 ZigBee 作為其中的傳輸協定，則協調者 (coordinator) 扮演了組織網路的角色，而智慧電表等相關設備會加入到協調者所形成的 ZigBee 網路之下，變成該網路下的其中一個節點；之後協調者則負責收集資料數據，再與後端伺服器作溝通。

然而，在一般的校園環境中，建築物內部多半已建置乙太網路 (Ethernet) 等有線網路基礎建設，假設現

今要在校園內建構智慧電網，當然可以利用既有的乙太網路進行建築物與建築物之間的通訊。除此之外，亦可利用 ZigBee 形成一個智慧電網傳輸網路，並藉由 ZigBee 路由器 (ZigBee Router) 轉送封包，將封包傳送至較遠的節點。如果系統伺服器端是架設在學校的計算機中心，當協調者需要與伺服器端作通訊時，最直接的方式是經由建築物內部原有的有線網路來達到傳輸目的；同時，也可以選擇在協調者與伺服器間，建置若干個路由器，透過無線網路的方式傳輸。如圖一。

當網路設備擁有多個有線或無線的介面時，理論上可以在單一介面故障時切換至另一網路介面，提高通訊系統的可靠度。然而實務上各種應用是否能善用此一優點，則取決於傳輸層 (Transport Layer) 協定的支援程度。



圖一：實驗環境

### 二、提出方法

在傳統的網路架構中，目前普遍使用的傳輸層通訊協定分別為 TCP (Transmission Control Protocol) 與 UDP (User Datagram Protocol) 兩種，TCP 提供可靠的傳輸方式，確保資料可以正確地傳送到目的地；而 UDP 則是一種非可靠性的傳輸方式，並不檢查封包是否正確的傳輸。然而 TCP 在通訊兩端點作傳輸時，一次只能使用一組 IP (Internet Protocol) 位址，存取一種網路介面，如果同時存在多條網路傳輸路徑，將無法被有效的利用。換句話說，當兩端點選擇其中一條路徑傳送時，其餘的傳輸路徑就會被閒置，造成資源浪費。對於本實驗環境來說，則勢必有線路徑與無線路徑只能選擇其中一種存取介面。

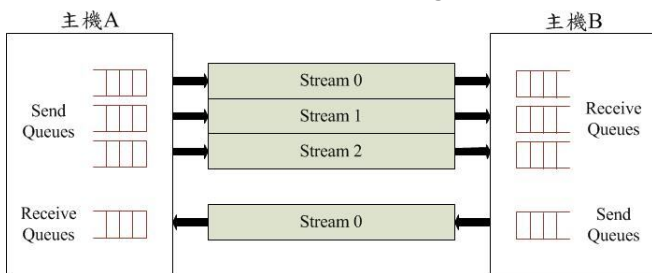
SCTP (Stream Control Transmission Protocol; 串流控制傳輸協定) 是 2000 年由 IETF (Internet Engineering Task Force) 的信號傳輸 (SIGTRAN) 工作小組定義的一種傳輸協定[6]。SCTP 與 TCP 及 UDP 相同，皆是位於 OSI 網路七層模型中傳輸層的通訊協定。由於 TCP 與 UDP 提供了完全互斥的傳輸型態，兩者已經漸漸不符合實際傳輸的需求，而 SCTP 透過結合 TCP

<sup>1</sup> 本研究由國科會贊助，計畫編號 NSC 99-2218-E-029-001、NSC 99-2221-E-260-012。

與 UDP 各自的優點，再加上其他協定所沒有的機制，致使 SCTP 能滿足高性能傳輸的需求，因此可以完全取代舊有 TCP 的傳輸，也能夠代替部分 UDP 的傳輸。SCTP 具有多重定址 (Multi-homing) 的功能，解決了無法同時存取多個傳輸介面的問題。多重定址機制主要提供兩端點之間能夠使用多條路徑連接，當網路斷線或是封包遺失需要重傳時，可以藉由備用路徑來回復連線以及快速重傳遺失的封包，如圖二所示。支援 SCTP 的兩端點可以藉由關聯 (association) 建立多條路徑，圖中主機 A 與主機 B 各有兩個 IP 位址，即表示有兩個傳輸介面。連線雙方會選定一條路徑作為傳輸的主要路徑，當傳輸過程發生封包遺失時，則會選擇其中一條備援路徑重送封包。另外，SCTP 也支援多資料流 (Multi-streaming) 的功能，允許應用程式根據不同的服務，提供多條的資料流 (Stream) 來傳輸封包。根據圖三所示，主機 A 與主機 B 之間建立了多條資料流，主機 A 向主機 B 要求三條資料流，資料流序號分別為 0 至 2，而主機 B 只向主機 A 要求一條，資料流序號則為 0。假設主機 A 傳向主機 B 的資料流 1 發生封包遺失，而資料流 1 等待封包重送的時間內，其餘資料流可以繼續自身的傳輸服務，不需要等待封包的重送。



圖二：Multi-homing

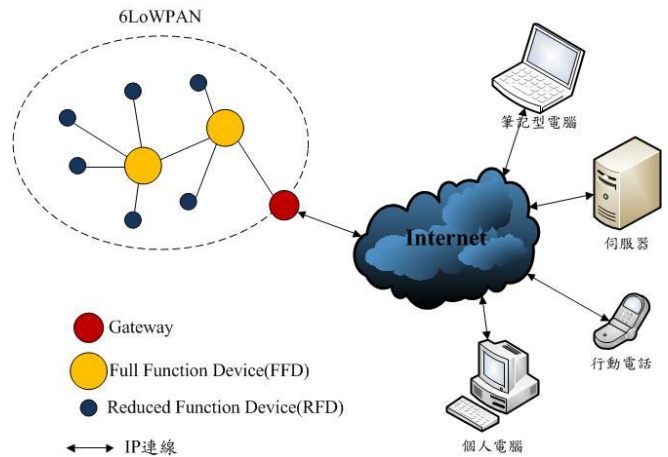


圖三：Multi-streaming

根據上述 SCTP 的特色，運用在本實驗環境中，可以讓有線路徑及無線路徑同時運作，達到多重路徑傳輸及備援的效果。因此本研究主要的方法在於將 SCTP 協定實行在協調者與伺服器之間的傳輸上，藉此增加資料的傳輸效能，並且免去當路徑斷線時，無法及時回傳資料的問題。

另外，本研究選擇以 6LoWPAN (IPv6 over Low Power Wireless Personal Area Networks) 作為無線傳輸方式。6LoWPAN 的發展始於 2004 年 11 月，由 IETF 組成一個工作組，致力訂定基於 IPv6 的低功率無線個人網域傳輸標準，即 IPv6 over IEEE 802.15.4。圖四即為 6LoWPAN 網路基本的架構[7]。相較於其他基於 IEEE 802.15.4 的傳輸標準，6LoWPAN 具有的優勢為 IP 網路技術的廣泛運用和其已發展成熟的特性。而 IPv6 也正在加速其普及的速度，相對於要發展新技術而言，IPv6 更易於被接受，且能直接與網際網路上既有的設備溝通，可充分利用現有的 IP 技術進行發展。由於使用

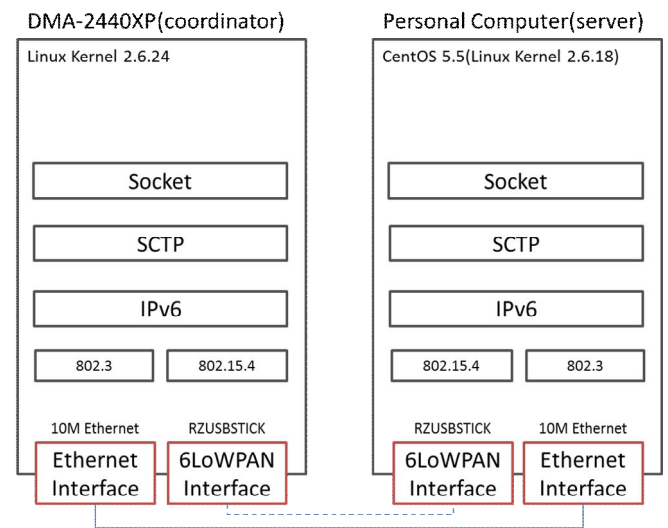
IPv6，6LoWPAN 具有大量 IP 位址，對於佈署大規模及高密度的節點有著莫大的幫助。並且，6LoWPAN 所形成的網路可以透過 Mesh 方式通訊，所以無須接入現行的有線網路當中。假設現今校園網路骨幹路由器發生故障，造成全校性的網路斷線，此時協調者依然可以透過 6LoWPAN 順利的將資料送回給伺服器。同時藉由有線及無線多種路徑進行通訊，以確保資料可順利傳輸。



圖四：6LoWPAN 架構

### 三、實驗結果

本研究的實作平台架構如圖五所示。長高科技的 DMA-2440XP 嵌入式平台，經過系統的移植，將扮演一個協調者的角色。另一台安裝 Linux CentOS 5.5 版作業系統的個人電腦則扮演伺服器的角色。由於 DMA-2440XP 與伺服器最後都會具有兩個以上的傳輸介面，因此，透過 SCTP 通訊協定的移植，進而實現在這兩節點之間的傳輸。



圖五：實作平台架構

#### 3.1 DMA-2440XP 平台移植

DMA-2440XP 平台採用 Samsung 公司 ARM9 系列中的 S3C2440 處理器，時脈穩定運行在 400MHz。DMA-2440XP 主要是針對嵌入式系統愛好者所設計的低成本及高效能之硬體平台。DMA-2440XP 平台擁有豐富的周邊介面和優秀的擴充性，其架構包含二個乙太網路介面，分別是 CS8900 支援 10M 乙太網路和 DM9000 支援

10/100M 乙太網路，另外還有二個串列埠、一個 USB Host、一個 USB Device、SD 卡介面、GPRS 模組、CAMERA、RS458 及 7 吋 LCD 面板等等。圖六 是 DMA-2440XP 平台外觀。外部記憶體方面，平台上包含 64MB 的 SDRAM、2MB 的 NOR FLASH 與 64MB 的 NAND FLASH，透過 TOP\_J3 跳線的配置可以選擇從何種 FLASH 方式啟動。當 TOP\_J3 跳線不接時，選擇從 NOR FLASH 啟動；相反地，當 TOP\_J3 跳線接上時，則從 NAND FLASH 啟動。除此之外，DMA-2440XP 不僅提供完整的底層驅動，並提供了 WinCE 5.0/6.0 和 Linux 2.6.14/2.6.24 下各個介面的驅動程式，還提供圖形介面控制這兩個系統下各驅動的範例程式，其中 Linux 系統的圖形介面為 MiniGUI。



圖六：DMA-2440XP 平台外觀

由於晶片、網路和感測器等技術不斷地發展，嵌入式系統便成為後電腦時代中發展的重要一環，廣為應用在科學研究、工業工程、軍事技術以及商業等方面。除此之外，開放原始碼的提倡發展，更加深了嵌入式 Linux 作業系統在嵌入式領域占有的地位。它不僅繼承了 Linux 核心穩定性強、軟體豐富等特性，而且也支援目前市面上大多數的主流處理器和硬體平台。因此，嵌入式 Linux 技術有著明確的發展潛力與市場需求。當然，嵌入式 Linux 作業系統本身能力會有一定的局限，主要的原因是，相對於一般個人電腦，嵌入式平台上所具備的快閃儲存記憶體量相對較小，並且處理器效能也沒有個人電腦來得強大，所以開發難度較高，系統功能較特定。但是正因為如此，大大提升了嵌入式系統在運行上的穩定度。嵌入式系統的軟體發展方法稱為交叉平台開發 (cross-platform development)，對於系統和應用軟體都是透過此方式來發展。嵌入式系統的軟體會在宿主機器 (host) 上作開發，然後在目標機器 (target) 上執行——本研究中即在 DMA-2440XP 平台上執行。進行交叉平台開發的主要工具為交叉編譯器 (cross compiler)，透過交叉編譯器在宿主機器上編譯生成可以在目標機器上執行的程式碼，而後可經由串列埠或網

路等方式將程式碼傳輸並裝載到目標機器上執行。本研究主要用到的編譯器是 arm-gcc，它是 gcc 的 arm 版本，目前 Linux 作業系統主要是以 GCC (GNU Compiler Collection) [8]編譯器進行移植的。

本研究所使用的 SCTP protocol stack 為 LKSCTP，即 Linux Kernel Stream Control Transmission Protocol，是於 2001 年期間所提出的計劃，主要目標為在 Linux 核心上實現 SCTP 通訊協定，它是一個 GNU GPL 開放源碼軟體。此計劃在 2001 年一月釋出了第一版 SCTP 開發套件，而目前 LKSCTP 已經內建於 Linux 2.6 核心系列裡面。根據上一段文章描述，因為快閃儲存記憶體量很小的關係，導致系統功能較特定且簡易，所以目前常見嵌入式 Linux 系統並不包含本研究需要的 IPv6 與 SCTP 協定功能。因此本研究必須重新編譯 Linux 核心，以加入所需的功能。

Linux 核心移植一般包括核心配置、核心編譯及核心載入三大部分。首先需要下載欲修改的 Linux 2.6.24 核心原始碼，在終端機視窗鍵入 make menuconfig 命令，接著會出現核心配置的介面，此時有很多配置的選項，根據本研究的需求，最後調配出適合本研究的嵌入式 Linux 核心。然後在終端機視窗鍵入 make zImage，透過 arm-linux-gcc-4.0.3 交叉編譯，執行完成之後，會產生 image 和 zImage 兩個核心映像檔，其中 zImage 為壓縮後的映像檔。由於本研究選擇使用 NAND FLASH 作為儲存記憶體，因此將 zImage 以 TFTP 方式載入到 NAND FLASH 對應的 kernel 分區中。核心移植至此算是完成了。

然而，作業系統正常運作的前提下，還需要一個檔案系統，在系統啟動時核心需要檔案系統來掛載。簡單地說，檔案系統是用於作業系統上明確的檔案和資料結構，而檔案系統的存在，使得資料可以被有效且透明地讀取或寫入。在嵌入式 Linux 中較常見的檔案系統有 RomFS、EXT2、RAMDISK、Cramfs 等，每一種類型選擇會因為目標用途差異有不同的考量。由於本研究的 DMA-2440XP 平台上具有較大量的 Flash，並且需要較多的記憶體供應用程式使用，所以本研究選擇 Cramfs 作為嵌入式 Linux 平台的檔案系統。

本研究通過 Busybox 工具包建立 Cramfs 檔案系統。Busybox 計劃是在 1996 年發起的，本身為一個很成功的開放源碼軟體計畫，它整合了一百多個常用的 Linux 命令和工具軟體。而 Busybox 支援多樣函式庫的特性，讓使用者可以非常方便地在 Busybox 中制定所需的應用程式，然後透過動態連結 Busybox 二進位檔，便能有效地減小程序的體積，如此使得 Busybox 在嵌入式系統的開發過程具有很大優勢。本研究選用的版本是 busybox-1.11.1，同樣是在終端機視窗鍵入 make menuconfig 進行檔案系統的配置，然後鍵入 make 命令編譯 busybox，此處使用的交叉編譯器則是 arm-linux-gcc-3.4.1，接著於終端機視窗鍵入 make install 命令，如果過程順利，安裝成功後會在預設\_install 目錄下看見基本檔案系統的內容。隨後新建一個系統目錄，稱作 rootfs，將\_install 目錄下的所有檔案複製到當前目錄，並且新增該檔案系統所需的設定檔。除此之外，由於本研究會使用 SCTP

通訊協定，但是基本檔案系統並不包含 SCTP 相關函式庫和軟體工具，因此本研究必須透過交叉編譯方式生成 ARM9 平台可執行的 LKSCTP 工具套件，其中包括 SCTP 程式執行必備的函式庫與 `sctp_test` 等命令工具。最後將所有檔案放到 `rootfs` 目錄底下，執行 `./mkcramfs`，即會產生本研究所需要的 `root_dma.cramfs` 檔案系統，把剛生成的 Cramfs 檔案系統燒寫到 NAND FLASH 中，重新啟動 DMA-2440XP 平台，就可以正常開啟本研究移植的嵌入式 Linux 系統了。

### 3.2 Atmel RZRAVEN 開發板移植

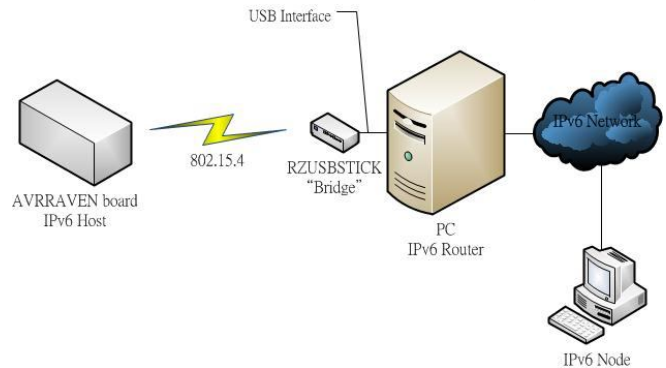
根據本研究假設的實驗環境中，DMA-2440XP 平台除了具備乙太網路介面之外，還需要包含有 IEEE 802.15.4 傳輸介面。目前 DMA-2440XP 平台上，僅有乙太網路介面，並未內建 IEEE 802.15.4 的無線傳輸介面。因此本研究另外採用 Atmel RZRAVEN 2.4GHz 開發板作為 IEEE 802.15.4 傳輸擴充介面。RZRAVEN 開發板套件包含二個 AVRRAVEN 模組與一個 RZUSBSTICK 模組。AVRRAVEN 模組內嵌有 ATmega3290P 微控制器用來驅動模組上的 LCD 面板，以及內嵌 ATmega1284P 微控制器用以驅動 AT86RF230 802.15.4 2.4GHz 無線電收發器；而 RZUSBSTICK 模組即是所謂的 USB dongle，模組上內嵌 AT90USB1287 微控制器，同樣也是用來驅動 AT86RF230 無線電收發器的。

由於 RZUSBSTICK 可以直接賦予個人電腦備有 IEEE 802.15.4 介面，因此本研究選擇此模組作為平台介面擴充使用。Contiki 為一個開放源碼作業系統[9]，它具有高度地移植能力、支援多重任務等特性，適合用在著重於高效能記憶體網路的網路嵌入式系統和無線感測網路。而 Contiki 主要就是針對微控制器設計的，使得系統所占記憶體的量相對較小，典型 Contiki 資源的配置只需要 2KB RAM 和 40KB ROM。另外，Contiki 不僅提供 IPv4 與 IPv6 兩種完整的 IP 網路協定，並且也提供了低耗能無線電通訊之架構。目前最新釋出的版本為 Contiki 2.5-rc1。透過 Contiki 作業系統的移植，本研究可以在 RZRAVEN 開發板上實現 6LoWPAN 傳輸網路。

移植的首要任務必須先取得 Contiki 2.5-rc1 的原始碼，然後根據原始碼中的範例程式，將其交叉編譯生成二進位檔，此處的交叉編譯器為 `avr-gcc`。由於本研究是選擇 RZUSBSTICK 模組當作擴充介面，因此只針對該模組進行編譯，編譯完成之後，會產生 `ravenusbstick.elf` 檔案，接著透過 JTAGICE mkII 將二進位檔燒寫到 AT90USB1287 微控制器上，至此把 RZUSBSTICK 插入個人電腦，則個人電腦即可成為 6LoWPAN 傳輸網路中的一個節點。如圖七所示，個人電腦將 RZUSBSTICK 視為一個橋接介面 (bridge)，然後會使用個人電腦上的乙太網路介面作為該介面預設路由。在 Contiki 計畫當中，RZUSBSTICK 與 Contiki 的結合，也就是硬體與韌體的結合，皆通稱為“Jackdaw”，以下敘述也跟著如此稱呼。

Jackdaw 主要運行方式是透過 RNDIS (Remote Network Driver Interface Specification) 模擬成一個網路

介面。RNDIS 是一種常見將 USB 設備模擬作網路介面的驅動程式，目前大多應用在 Windows 作業系統上，但其實也支援部分 Linux 作業系統。不過，在去年十一月，Jackdaw 已經可以選擇另外一種 CDC-ECM (Ethernet Control Model) 驅動程式來模擬網路介面，可是該驅動程式只適用於 Linux 作業系統上，Windows 作業系統則沒有支援。因為 Jackdaw 的緣故，之前編譯的嵌入式 Linux 核心並不支援 RNDIS 相關的驅動程式，為了將介面擴充至 DMA-2440XP 平台，所以在此本研究必須重新編譯一次系統核心，把 RNDIS 相關的驅動程式加入核心配置，待交叉編譯完成，即可重新燒寫 DMA-2440XP 平台上的嵌入式 Linux 核心。至此，Jackdaw 也可以順利地結合到 DMA-2440XP 平台了。



圖七：Contiki 網路架構

### 3.3 Socket 程式撰寫

以上敘述皆是硬體平台移植的部分，當所有步驟都順利操作之後，本研究需要具備的硬體環境也就完成了。接下來，則開始撰寫基於 SCTP 傳輸協定的 socket 程式。相對於傳統 TCP 或 UDP 傳輸協定，SCTP 提供更多樣的專屬 socket API (Application Programming Interface)，由於本研究的實驗環境是假設在多重路徑選擇下，因此 socket 程式目前只針對多重路徑部分撰寫，並未考慮多資料流的情況。

SCTP 允許在建立連線時，同時選擇多個網路介面。不管是伺服器端或用戶端，根據選擇的介面會有不同排列組合。然而，一般 socket 的 `bind()` 只能存取單個 `sockaddr` 結構，也就是一次只能綁定單個 IP 位址，所以 SCTP 採用新的 `sctp_bindx()` 函式，它可以同時綁定以陣列方式存放的多個 `sockaddr`，每一個 `sockaddr` 結構都代表一個 IP 位址，而每一個 IP 位址將具有相同的埠。`sctp_bindx()` 語法如下：

```
int sctp_bindx(int sd, struct sockaddr *addrs, int count, int flags)
```

`flags` 參數值可以填入 `SCTP_BINDX_ADD_ADDR` 表示要增加綁定的 IP 位址，或是選擇填入 `SCTP_BINDX_DEL_ADDR`，則表示要將該 IP 位址從綁定列表中移除。第二個參數則是存有 IPv4 或 IPv6 位址的結構。

除此之外，SCTP 提出新的 `sctp_connectx()` 函式，相對於 `sctp_bindx()`，此函式可以讓用戶端透過多個網路介面作連結，解決了 `connect()` 只能選擇

單個介面作連結的問題。當用戶端使用 `sctp_connectx()` 要求連線時，會由伺服器端選擇，並回應要用哪一個介面來連線。主要是因為伺服器至用戶端通常是一對多的 socket 連線，伺服器端可以接受多個用戶端的連線請求，因此必須經由伺服器端選擇分配介面給各個用戶端使用。一旦連線建立之後，資料即會透過選擇的介面進行傳輸。然而，socket 連線也可以透過 `setsockopt()` 裡的參數值來指定優先選擇介面，`SCTP_PRIMARY_ADDR` 會告知伺服器端在本用戶端中哪一個介面是優先想接收資料的；`SCTP_SET_PEER_PRIMARY_ADDR` 則是告知伺服器端想優先使用伺服器端上的哪一個介面。

當伺服器端和用戶端上的 socket 程式撰寫完成之後，必須要用交叉編譯器來編譯程式，此處的交叉編譯器為 `arm-linux-gcc-3.4.1`。編譯完成將會產生一個執行檔，隨後可以透過 `rz` 與 `sz` 傳輸工具將執行檔上載到嵌入式 Linux 系統內，透過執行 socket 程式，可以讓 DMA-2440XP 平台與伺服器之間，建立一個 SCTP 連線，兩端點都會各自綁定二個介面（乙太網路介面和 6LoWPAN 介面；通常指定乙太網路當作優先傳輸的介面），至此即可相互傳輸資料了。

#### 四、結論

經過上一節描述的步驟，本研究已經可以把實驗環境完全地實作出來。接下來將運用到現實的智慧電網資料傳輸上，透過 SCTP 傳輸協定移植，期待能增加資料傳輸效率，降低封包的遺失，並且可以提升網路的穩健性。

然而，由於現有的 SCTP 多重路徑傳輸策略只有在主要路徑失效後才會進行切換，此策略可能會造成主要路徑長期處於忙碌狀態，但是其餘路徑卻是閒置的，這時即需要一個負載平衡的選擇機制。除此之外，在主要路徑封包遺失率或延遲時間品質下降時，也應該立即作路徑的切換；而不是等到路徑完全失效後才切換。所以明顯地有比現有策略更佳的路徑轉換演算法。本研究希望未來能提出適合智慧電網的路徑選擇機制，藉此更進一步的提高傳輸效能。

#### 參考文獻

- [1] Paolo Baronti, Prashant Pillai, Vince W.C. Chook, Stefano Chessa, Alberto Gotta, Y. Fun Hu, "Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards," *Computer Communications*, Volume 30, Issue 7, 26 May 2007, pp. 1655-1695.
- [2] N. Kushalnagar, G. Montenegro, C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals," RFC4919, August 2007.
- [3] G. Montenegro, N. Kushalnagar, J. Hui, D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," RFC4944, September 2007.
- [4] Santacana, E., Rackliffe, G., Le Tang, Xiaoming Feng, "Getting Smart," *IEEE Power & Energy Magazine*, March/April 2010.

- [5] Ali Vojdani, "Smart Integration," *IEEE Power & Energy Magazine*, November/December 2008.
- [6] P. Natarajan, J. R. Iyengar, P. D. Amer, and R. Stewart, "SCTP: An Innovative Transport Layer Protocol for The Web," *The 15th International Conference on World Wide Web*, Edinburgh, Scotland, May 23-26, 2006.
- [7] Gee Keng Ee, Chee Kyun Ng, Nor Kamariah Noordin and Borhanuddin Mohd. Ali, "A Review of 6LoWPAN Routing Protocols," *Proceedings of Asia-Pacific Advanced Network (APAN) Network Research Workshop (ISBN 978-4-9905448-0-5)*, Hanoi, Vietnam, August 9-13, 2010.
- [8] GCC, The GNU Compiler Collection. [<http://gcc.gnu.org/>]
- [9] Contiki: The Operating System for Connecting the Next Billion Devices - the Internet of Things. [<http://www.sics.se/contiki/>]