國立暨南國際大學資訊工程學系

碩士論文

智慧電網下 6LoWPAN 協定傳輸及 SCTP 多重連接機制之研究

The Study of 6LoWPAN with SCTP Multi-homing in Smart Grid

指導教授：吳坤熹博士

研究生：呂冠達

中華民國一〇〇年七月二十九日

國立暨南國際大學資訊工程學系

碩士論文

智慧電網下 6LoWPAN 協定傳輸及 SCTP 多重連接機制之研究

The Study of 6LoWPAN with SCTP Multi-homing in Smart Grid

指導教授：吳坤熹博士

研究生：呂冠達

中華民國一〇〇年七月二十九日

國立暨南國際大學碩士論文考試審定書

資訊工程學系 學系（研究所）

研究生 呂冠達 所提之論文

智慧電網下 6LoWPAN 協定傳輸及 SCTP 多重連接機制之研究

The Study of 6LoWPAN with SCTP Multi-homing in Smart Grid

經本委員會審查，符合碩士學位論文標準。

學位考試委員會

溫志宏 委員兼召集人

[illegible] 委員

吳坤熹 委員

中華民國 一00 年 07 月 29 日

致謝

在研讀碩士班的兩年光景中，首先要誠摯地感謝指導教授吳坤熹老師的悉心指導與啟發。於專業領域上，教導我養成具有獨立思考解決問題的能力；而當研究中遇到無法解決之問題時，也能適時提供我思考方向和寶貴地意見，協助我突破困境；於待人處世上，老師不吝分享自身豐富的人生經驗與生活哲學，使我學習到在面對任何情況下應具備有的態度。老師的諄諄教誨讓我受益良多，在此獻上最深的謝意與感激。除此之外，感謝口試委員溫志宏教授和王忍成教授在口試期間給予的指正及建議，使得本論文可以順利地完成，內容能更加完備。

回想第一次隻身來到偌大地暨南校園那一刻的徬徨和無助，很慶幸自己能進入新世代網路電話實驗室（LAB409），進而在這個大家庭中學習與成長。兩年的研究生活中，感謝已經畢業的嘉裕學長、文仁學長、筱婷學姊、韋立學長、韋勳學長、韋霖學長、瑋勵學長、霓雅學姊和信富學長，在我初進實驗室懵懵懂懂的時候，給予我很多課業及生活上的鼓勵，讓我能很快地融入實驗室生活。另外感謝博士班惟綸學長與實驗室同窗同學豈嘉、俊克、創宏和麗雯，大家一起打拼，往共同目標邁進的情景，都令我難以忘懷。還有感謝碩一的揮雄、佳紋、書丞、翰銓、鈺萍及大學部的伯岡和雅玲等學弟妹，有你們幫忙處理實驗室中的大小雜事，使我能夠專心致力於研究。因為有大家的陪伴，我的碩士生涯才會如此多彩多姿，在歡笑和汗水夾雜的過程中，圓滿且順利的度過，這都將成為我永遠珍貴的回憶。

最後感謝我的家人在這段非常時期給予我莫大的支持與鼓勵，讓我毫無後顧之憂，能全力完成我的研究。最最後感謝在這碩士生涯過程中陪伴及鼓勵我的所有親朋好友們。在此致上我最真誠的感謝，願將碩士班畢業的喜悅分享給大家。

論文名稱：智慧電網下 6LoWPAN 協定傳輸及 SCTP 多重連接機制之研究

校院系：國立暨南國際大學科技學院資訊工程學系　　頁數：62

畢業時間：中華民國一〇〇年七月　　學位別：碩士

研究生：呂冠達　　指導教授：吳坤熹博士

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。

關鍵詞：SCTP；智慧電網；無線網路

Title of Thesis: The Study of 6LoWPAN with SCTP Multi-homing in Smart Grid

Name of Institute: Department of Computer Science and Information Engineering, College of Science and Technology, National Chi Nan University Pages: 62

Graduation Time: July 2011 Degree Conferred: Master

Student Name: Kuan-Ta Lu Advisor Name: Dr. Quincy Wu

Abstract

With the evolution of communication technology, wireless networks have become widely adopted in many modern applications. The communication in a Smart Grid system relies on a large number of low-power sensors, which stimulates vigorous development of related researches. Stream Control Transmission Protocol (SCTP) is a new transport-layer protocol. It combines the advantages of both TCP and UDP, and provides new features such as multi-homing and multi-streaming. SCTP can provide high-performance transmission in both wired and wireless networks. This paper applied SCTP in a wireless sensor network, and studied how it can improve network reliability.

Key words: SCTP; Smart Grid; wireless sensor network

目錄

學位考試委員會審定書 I
致謝 II
摘要 III
Abstract IV
目錄 V
圖目錄 VII
表目錄 IX
第一章 緒論 1
1.1 研究背景 1
1.2 研究動機 4
1.3 論文架構 5
第二章 背景知識及文獻探討 6
2.1 網際網路 6
2.2 IPv6 10
2.3 SCTP 13
2.3.1 區塊綁定（Chunk Bundling） 14
2.3.2 四方交握（4-way Handshake） 16
2.3.3 多重定址（Multi-homing） 18
2.3.4 多資料流（Multi-streaming） 19
2.3.5 路徑監測（Heartbeat） 20
2.3.6 選擇性回應（SACK） 21
2.4 IEEE 802.15.4 21
2.4.1 實體層 23

2.4.2 媒體存取控制層 23
2.5 ZigBee 26
2.5.1 網路層 28
2.5.2 應用層 30
2.6 6LoWPAN 30
2.6.1 適應層 33
2.6.2 路由協定 36
第三章 系統實作 38
3.1 DMA-2440XP 平台移植 38
3.2 Atmel RZRAVEN 開發板移植 43
第四章 效能分析 46
4.1 實驗環境與方法 46
4.2 Socket 程式撰寫 50
4.3 數據量測與分析 51
第五章 結論及未來展望 53
參考文獻 55

圖目錄

圖 1-1、OSI 模型 1
圖 1-2、智慧電網架構圖 2
圖 1-3、實驗假設環境 4
圖 2-1、網路連線示意圖 6
圖 2-2、TCP/IP 模型 7
圖 2-3、IPv4 標頭格式 8
圖 2-4、TCP 標頭格式 9
圖 2-5、TCP 三方交握 10
圖 2-6、UDP 標頭格式 10
圖 2-7、IPv6 標頭格式 11
圖 2-8、IPv6 延伸標頭 12
圖 2-9、EUI-64 位址轉換 12
圖 2-10、SCTP 關聯傳輸 14
圖 2-11、SCTP 標頭格式 15
圖 2-12、資料區塊格式 16
圖 2-13、SCTP 四方交握 17
圖 2-14、Multi-homing 18
圖 2-15、Multi-streaming 19
圖 2-16、ZigBee/IEEE 802.15.4 協定堆疊架構 22
圖 2-17、超級訊框結構 24
圖 2-18、設備傳送資料給協調者 25
圖 2-19、協調者傳送資料給設備 26
圖 2-20、ZigBee 應用領域 27

圖 2-21、ZigBee 通訊協定堆疊 28
圖 2-22、ZigBee 支援網路拓樸 29
圖 2-23、6LoWPAN 結構 31
圖 2-24、6LoWPAN 協定架構 32
圖 2-25、第一個切割標頭 34
圖 2-26、網狀標頭 35
圖 2-27、發送標頭 35
圖 2-28、6LoWPAN 標頭次序 36
圖 2-29、6LoWPAN 路由決策階層 36
圖 2-30、LOAD 路由協定 37
圖 2-31、HiLow 路由架構 37
圖 3-1、實作平台架構 38
圖 3-2、DMA-2440XP 平台外觀 39
圖 3-3、核心配置畫面 41
圖 3-4、BusyBox 配置畫面 43
圖 3-5、嵌入式 Linux 系統啟動畫面 43
圖 3-6、RZUSBSTICK 模組 44
圖 3-7、Contiki 網路架構 46
圖 4-1、實驗環境 47
圖 4-2、傳輸時間軸 52
圖 4-3、多重路徑傳輸點陣圖 53

表目錄

表 2-1、SCTP 和 TCP 及 UDP 比較表 13

表 2-2、區塊的形態 15

表 2-3、IPv6 標頭壓縮比較表 33

表 2-4、發送標頭位元序列表 35

第一章 緒論

1.1 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。繼先前 3G、WiMAX（Worldwide Interoperability for Microwave Access）、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 標準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，也就是 OSI（Open Systems Interconnection）網路七層模型[1]中所講述的實體層與資料鏈結層（參見圖 1-1），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協定，進而產生不同的通訊網路。其中，ZigBee[2]和 6LoWPAN[3][4]就是目前 IEEE 802.15.4 上較受歡迎的兩種網路協定。

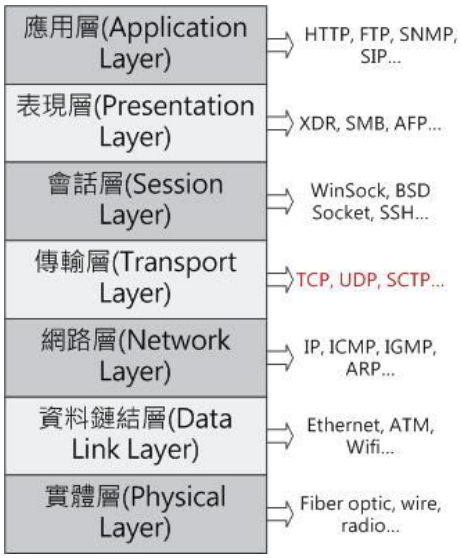

圖 1-1、OSI 模型

智慧電網（Smart Grid）[5][6]，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輸比例，達成資源的有效利用，藉此減少某些地區用電量不足或是電量過度浪費的情形。智慧電網是一雙向溝通的輸電網路，可以動態調度電力。智慧電表基礎建設（Advanced Metering Infrastructure；AMI）是智慧電網中一個關鍵的部份，用來記錄所有電能的流動。系統中通訊傳輸的需求，根據傳輸環境的差異會有不同考量，目前常見的選擇有 Wi-Fi、GPRS（General Packet Radio Service）、ZigBee 和 WiMAX 等無線網路通訊技術。對於短距離傳輸來說，雖然 Wi-Fi 或 GPRS 傳輸的技術較為成熟，而且價格也相對便宜，不過以省電的立場來看，由於 ZigBee 具有低速率、低耗電、低成本和支援大量網路節點的特性，適合用在低耗能環境的建置，因此 ZigBee 常被選擇運用於智慧電網的傳輸協定。在基礎建設中，協調者（Coordinator）負責組織網路，而智慧電表等相關設備會加入到協調者所形成的網路之下，成為該網路下的一個節點；協調者則負責收集資料數據，再與後端伺服器作溝通，如圖 1-2。

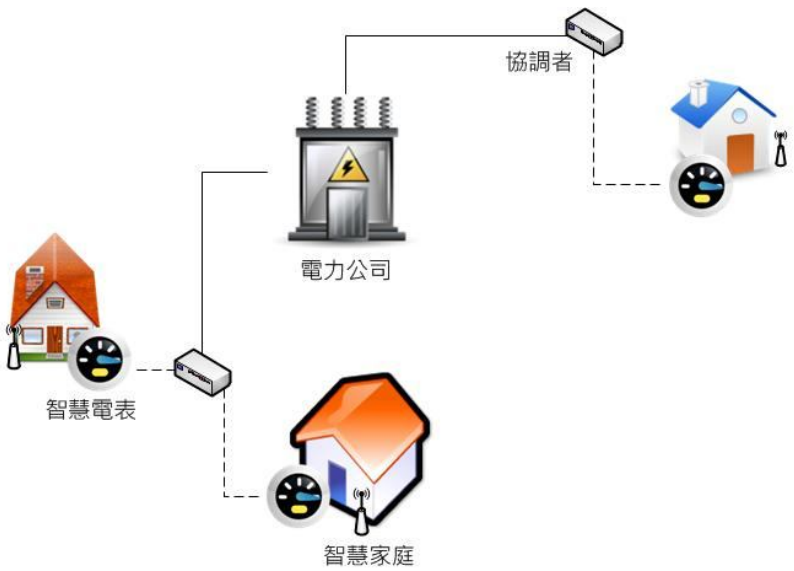

圖 1-2、智慧電網架構圖

與智慧電表相比，由於傳統電表是採用機械方式監控用電量，難免會造成監控上

誤差；隨著使用時間越久而造成設備的老化，誤差就會越來越大。再加上傳統電表需要仰賴人工方式挨家挨戶的抄寫各個電表用電量，更增加了許多不可預期的因素影響電量準確性。而智慧電表是透過數位化的方式，能即時的觀察用電情況，使得電量監控可以更精確；透過此一即時監控措施，可以了解各種單一家電用品的使用情況，並進一步根據智慧電表監控結果，手動或自動控制家電用品的使用，因此也增加時間電價[7]實行的可行性。為了鼓勵用戶改在離峰時間使用電器，許多電力公司會採用時間電價的收費模式，即在尖峰時間所用的電將會比離峰時間用的電來得昂貴。許多用戶就會為了降低電費的支出，當離峰時間電費便宜時，可以選擇使用耗電量較高的電器設備，例如洗衣機、空調等。而在尖峰時間電費高漲時，則關閉一些非必要的電器產品來降低電力需求。但傳統的機械式電表無法支援時間電價的功能，必須仰賴智慧電表才可能達成此功能。除此之外，智慧電網已經將電流轉變成資訊流的形式，因此所有用戶的用電情況，都可以透過智慧電網回傳給電力公司，而不再需要人工使用紙筆抄寫。對於電力公司而言，因為清楚地掌握用戶的電力需求，就能更有效的配置電力傳輸系統，以達到節約能源的目的。知名雜誌數位時代曾經報導，美國杜克能源（Duke Energy）預估，在美國推行的智慧電網計畫中，預計在三年內可以減少三百五十至一千八百七十億仟瓦的用電量，相當於四百萬至兩千萬輛行駛中的汽車所排放的碳排放量[8]。不只如此，智慧電網還具有電力整合利用的備援概念，與各種形式的發電方式如風力、火力和太陽能等，構成一個能源安全體。舉例來說，當用戶家中建置有太陽能發電，並且同時擁有蓄電設備。那麼平時太陽能發電量可以儲備在蓄電設備裡，當作備援的電力。假設太陽能發電量大於用戶所使用的量時，多出來的電能，則可以選擇透過智慧電網回售給電力公司[9]。藉著智慧電網，電力公司與用戶之間會有更多的互動，而不再只是單純電力供給而已。而要能即時地在智慧電網中調配電力，智慧電表基礎建設中所須具備一套可靠的通訊系統，是確保智慧電網順利運作的關鍵。

1.2 研究動機

在一般的校園或住家環境中，建築物內部泰半已建置乙太網路（Ethernet）等有線網路基礎建設，使建築物與建築物之間可以互相通訊。因此，假設現今要在校園內建構智慧電網，如果系統的伺服器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，也可以在協調者與伺服器間，建置若干個無線路由器，透過無線網路的方式傳輸，如圖 1-3 所示。在有線路徑部分，我們選擇一般的乙太網路做傳輸，而無線路徑則以 6LoWPAN（IPv6 over Low Power Wireless Personal Area Networks）的傳輸方式來完成。6LoWPAN 所形成的網路可以透過網狀拓樸（Mesh Topology）方式通訊，所以無須接入現行的有線網路當中。使用多路徑的優點是，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。

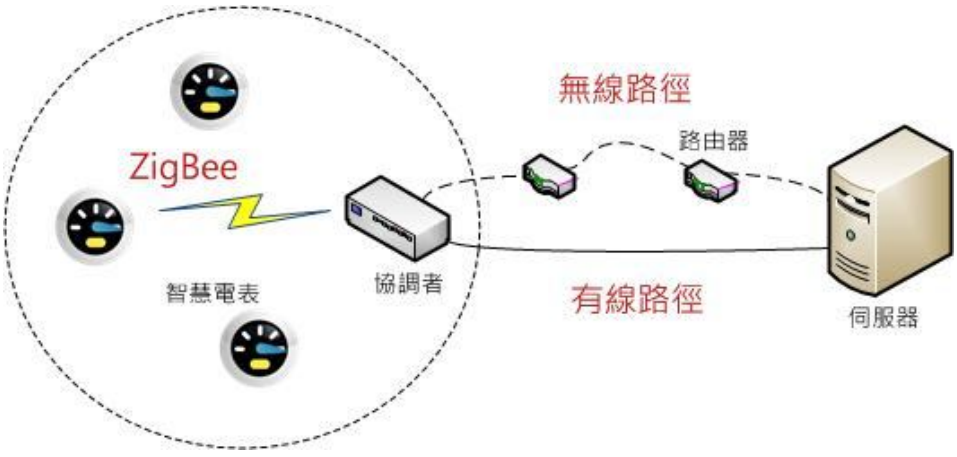

圖 1-3、實驗假設環境

根據圖 1-3 而言，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。隨著網際網路的蓬勃發展，網路傳輸技術也愈趨多元，因此各種網路通訊裝置上漸漸的也都配置多樣的網路存取介面，例如 Wi-Fi、3G、乙太網路和藍牙（Bluetooth）等，以供各種不同網路環境使用。由於傳統 TCP（Transmission Control Protocol）連線時在通訊兩端點一次只能使用一組 IP（Internet Protocol）位址，存取單一網路介面。如果同時存在多個網路傳輸路徑，將無法被有效的利用。意即當兩端點選擇其中一條路徑傳送時，其餘傳輸路徑就會被閒置，造成資源浪費。因此，多重路徑傳輸成為網路上一個很重要的研究議題。

為了因應日趨複雜的網路環境，IETF（Internet Engineering Task Force）在 2000 年十月制定了一種新興的傳輸層協定，稱為 SCTP（Stream Control Transmission Protocol）[10]。SCTP 具有多重定址（Multi-homing）及多資料流（Multi-streaming）等特性，提供了同時在多個存取介面傳輸的機制。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。將此機制運用到智慧電網環境中，則可以讓有線路徑與無線路徑同時運作，達到多重路徑傳輸及備援的效果。本研究主要的目的在於將 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。

1.3 論文架構

全文共分為五個章節。第一章是緒論，主要描述本論文的研究背景及動機；第二章將介紹網路傳輸協定的架構與相關知識，另外包含低速率無線個人區域網路協定之探討；第三章的系統實作主要分為兩個部分作說明，分別是平台與傳輸器兩大部分；第四章會針對最後實作出來的環境，作效能的量測與解析；第五章為本論文之總結與

未來展望。

第二章 背景知識及文獻探討

2.1 網際網路

網際網路（Internet）即是透過網路線或者是無線網路技術，將數台電腦主機、網路印表機和 NAS（Network-attached storage）等相關周邊設備以網際網路協定（Internet Protocol）連接起來，使得資料可以藉由這些連線達到互相傳輸的能力。計算機網路組成的元件，如圖 2-1，可以分為節點（Node），舉凡一般具有 IP 位址的設備，例如個人電腦、網路印表機、檔案伺服器和數據機等，都被歸類為網路節點之一。在這些設備當中，都會內建或外插一到數張的網路卡（Network Interface Card；NIC）。在個人電腦上，目前最常見的就是乙太網路卡。交換器和集線器主要的功能是匯集區域網路（Local Area Network；LAN）中的網路設備，使各分支的設備可以連接到網路主幹上；透過路由器，可以把數個區域網路連接形成一個較大的廣域網路（Wide Area Network；WAN）。而路由器上皆會維護一個路由表（Routing table）以供找尋路徑時查詢。根據上述的基本網路元件，用不同的連接方式將會形成不一樣的網路拓樸。

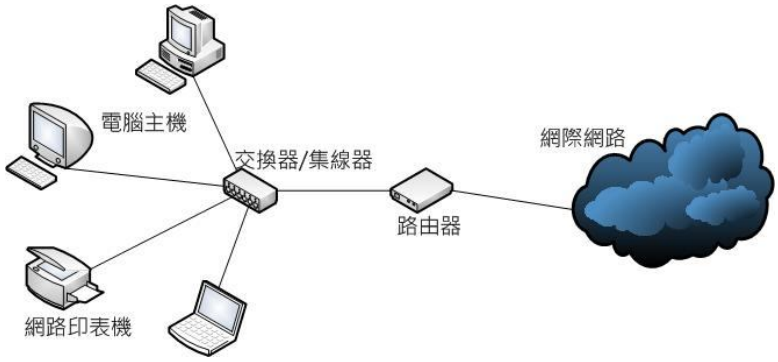

圖 2-1、網路連線示意圖

開放式通訊系統互連參考模型（Open System Interconnection Reference Model）簡稱 OSI 模型，是國際標準化組織（International Standard Organization；ISO）為了解決不同網路之間因為不相容造成彼此無法互相溝通的問題，提出一個試圖讓不同網路能互連的標準框架。OSI 模型主要分為七個階層，如圖 2-2(a)所示，每一個階層各自有提供不同的功能。

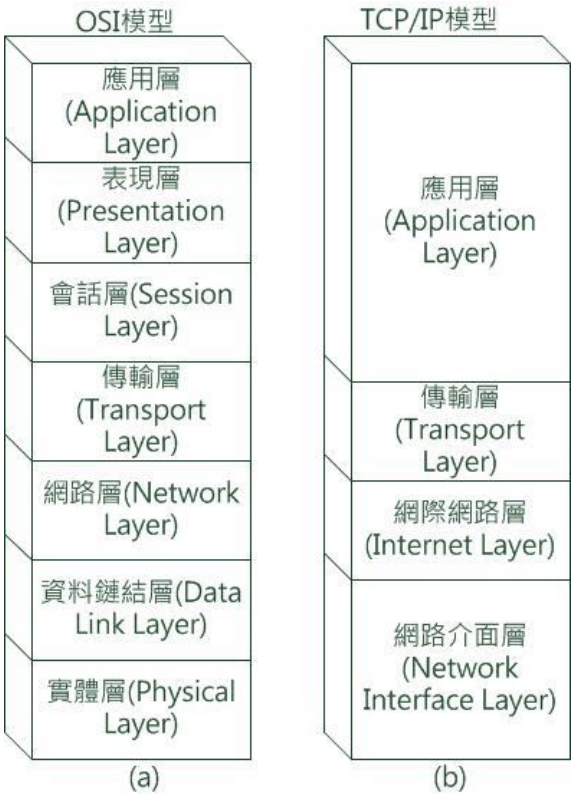

圖 2-2、TCP/IP 模型

雖然 OSI 模型的定義非常嚴謹，但也因為太過於嚴謹，因此程式的撰寫變得相對困難，造成發展上許多的阻礙。於是，由美國國防部尖端研究企劃署（Defense Advanced Research Project Agency；DARPA）所提出的 ARPANET 發展而來的 TCP/IP 模型徹底解決了這個問題。TCP/IP 模型同樣具有階層的概念，只是將其簡化成四個階層的架構，如圖 2-2(b)。底層的網路介面層主要是與硬體有關，最常被使用的就是乙太網路，是由 IEEE 802.3 工作群組制定乙太網路相關的標準規格。乙太網路標準的網路拓樸為匯流

排型拓樸（Bus Topology），傳輸主要是由乙太網路卡對乙太網路卡之間的資料封包傳遞。每張乙太網路卡在出廠時皆會賦予一組獨一無二的網路卡卡號，即 MAC（Media Access Control）位址。

TCP/IP 就字面上來看，代表著 TCP[11]與 IP[12]兩種通訊協定。IP 是網際網路層的通訊協定，目前使用最為廣泛的是網際網路協定第四版（Internet Protocol version 4；IPv4）。IPv4 封包的標頭如圖 2-3 所示，每個封包有著不同的標頭內容，其中 TTL（Time to Live）欄位表示封包的存活時間，範圍為 0-255，每當通過一個路由器時，TTL 值就會減 1，直到 TTL 值等於 0 時就會丟棄該封包；Protocol 欄位值則是來自傳輸層的協定代碼，如 TCP 代碼為 6、UDP（User Datagram Protocol）代碼為 17 或者是 ICMP（Internet Control Message Protocol）[13]代碼為 1；Source Address 與 Destination Address 欄位分別代表來源端和目的端 IP 位址。IPv4 位址是由 32 位元所組成，位址就像是現實生活中每戶人家的門牌號碼一樣；只要寫對位址，郵差就可以幫你把信送到目的地。IP 位址的取得方式可以是直接手動設定或由動態主機設定協定（Dynamic Host Configuration Protocol；DHCP）[14]伺服器發送等方式。

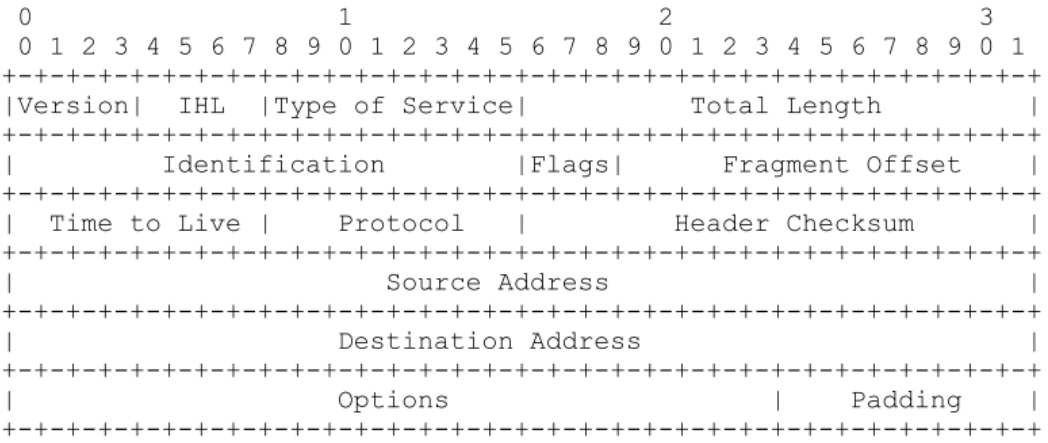

圖 2-3、IPv4 標頭格式[12]

TCP 是傳輸層的通訊協定，為連線導向的協定；至於同階層的 UDP[15]，則為非

連線導向協定。TCP 封包的標頭如圖 2-4 所示，Source Port 與 Destination Port 欄位表示來源端和目的端的埠。雖然封包的傳送是藉由 IP 位址來連接兩端，但這樣是不夠的，還需要兩端點啟動相對應的埠，才能順利建立通道。埠最大可以達到 65535 號，其中有一些埠是規定好的特權埠（Privileged Ports）。舉例來說，21 埠代表 FTP（File Transfer Protocol）連線、23 埠為 Telnet 連線和 80 埠則表示 HTTP（Hypertext Transfer Protocol）連線等。TCP 一般被稱作可靠的連線機制，其中最重要的特色就是三方交握（3-way Handshake）（參考圖 2-5）[16]。當用戶端想要與伺服器連線時，會先發送一個 SYN（Synchronize）封包，伺服器在接收到這個封包之後，則會回覆 SYN/ACK（Synchronize/Acknowledge）封包給用戶端，確認要與用戶端建立連線。接下來用戶端再回覆一個 ACK 封包給伺服器，就完成了連線的建立。UDP 與 TCP 不一樣，在傳送的過程中，伺服器不會在收到每個封包時，回覆任何回應封包，因此它不可靠。UDP 封包的標頭如圖 2-6 表示，由於少了嚴密的檢查與確認機制，所以封包的標頭資料相對較少，封包因此可以填入更多的資料。

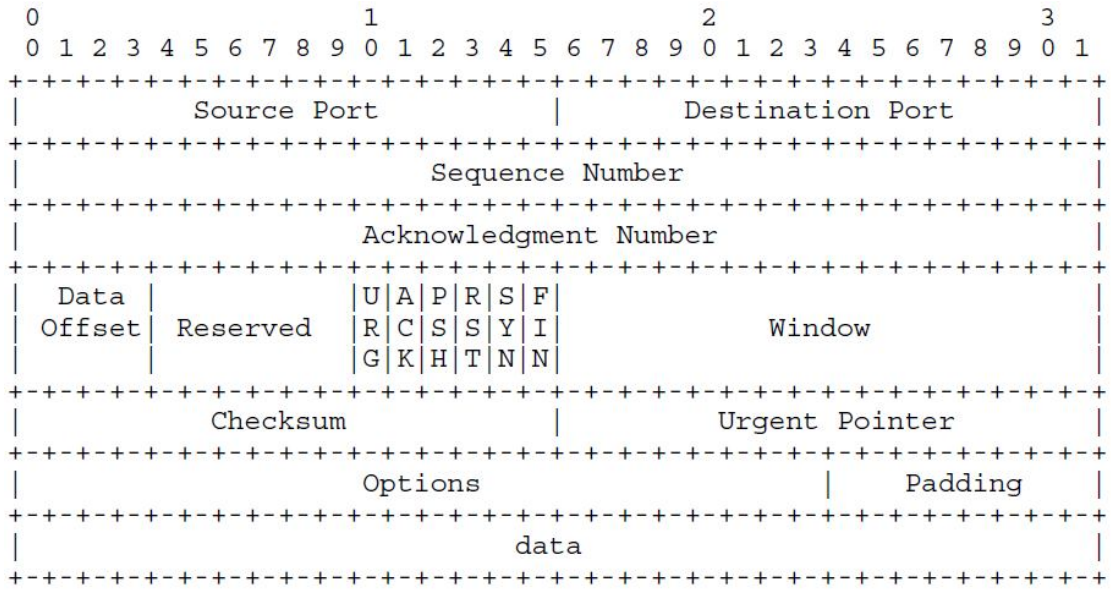

圖 2-4、TCP 標頭格式[11]

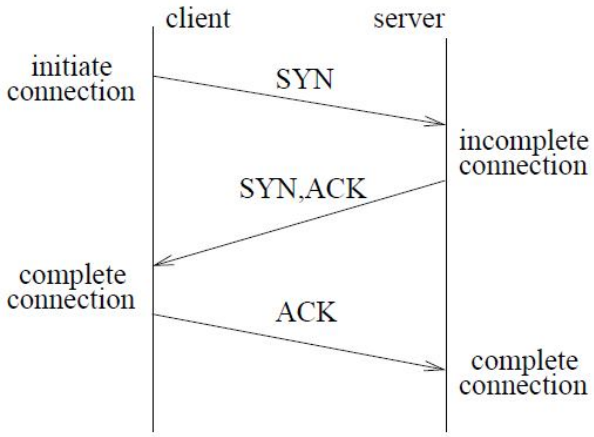

圖 2-5、TCP 三方交握

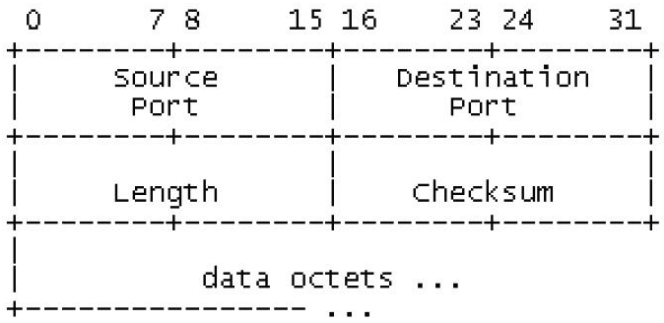

圖 2-6、UDP 標頭格式[15]

2.2 IPv6

網際網路協定第六版（Internet Protocol version 6；IPv6）[17]是 IPv4 的下一代標準。由於 IPv4 位址在今年 2 月已經消耗殆盡[18]，因此驅使了 IPv6 技術的發展。IPv6 具有比 IPv4 更多的位址數量，原因在於，IPv6 位址的長度為 128 位元，擁有 2^{128} 個位址，相較於 IPv4 位址的 32 位元，能提供充裕的 IP 位址[19]。在現行的網際網路架構下，都是朝 All-IP[20]核心網路的方向發展，舉凡各類電腦設備、通訊裝置甚至是智慧型家電產品等，都會分配一個 IP 位址，更加深了 IPv6 發展的重要性。

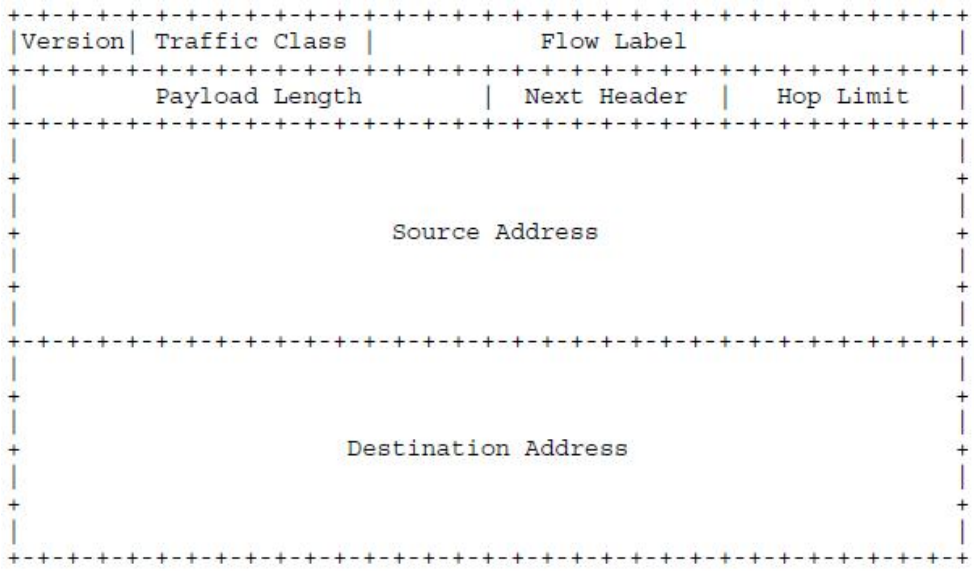

圖 2-7、IPv6 標頭格式[17]

根據上圖 2-7 的 IPv6 封包基本標頭格式，基本標頭固定為 40 位元組，相較於 IPv4 標頭，它的欄位更加的簡化。雖然 IPv6 位址多 IPv4 位址四倍，但是封包標頭僅增加為兩倍而已，設計上極有效率。此外，IPv6 把一些非必要的欄位移到延伸標頭（Extension Headers），提供較為彈性的選擇，藉此節省標頭的空間且提高處理效率。延伸標頭是由 Next Header 欄位所指定，目前定義的標頭有 Hop-by-Hop Options、Routing、Fragment 和 Destination Options 等。每一種標頭都含有 Next Headers 欄位，當有需要時才加入該延伸標頭，標頭之間會根據 Next Headers 所指的位址，找到下一個標頭，如圖 2-8 所示。Source Address 與 Destination Address 欄位需分別填入來源及目的端 IPv6 位址，每一個位址有 128 位元，採用十六進位的表示方式。而 IPv6 位址是由兩個邏輯部分組合而成，前半段為 64 位元的網路前置碼（Prefix），後半段則是 64 位元的主機位址。主機位址一般都根據 MAC 位址自行產生，此方式稱作 EUI-64（參考圖 2-9）[21]。因此，IPv6 支援自動組態（Auto-configuration）的定址方式，自動組態可以分為具狀態（Stateful）和無狀態（Stateless）兩種，具狀態自動組態是由動態主機設定協定第六版（Dynamic Host Configuration Protocol version 6；DHCPv6）[22]伺服器分配 IPv6 位址、參數及相關的網路訊息；而無狀態自動組態則會根據所在網

路下的網路前置碼和自動產生的 64 位元主機位址，結合成一個 IPv6 位址，之後需執行重複位址檢測（Duplicate Address Detection；DAD）[23]，以確保該位址是否唯一。除此之外，IPv6 支援服務質量（Quality of Service；QoS）機制[24]，透過 Traffic Class 與 Flow Label 欄位，可以對特定的資料流加上標記或是提高優先權來完成特殊的服務，對於即時性的多媒體傳輸會很有助益。

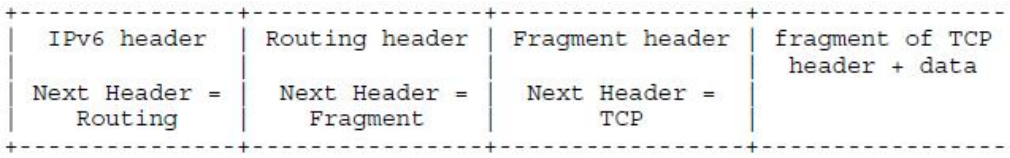

圖 2-8、IPv6 延伸標頭

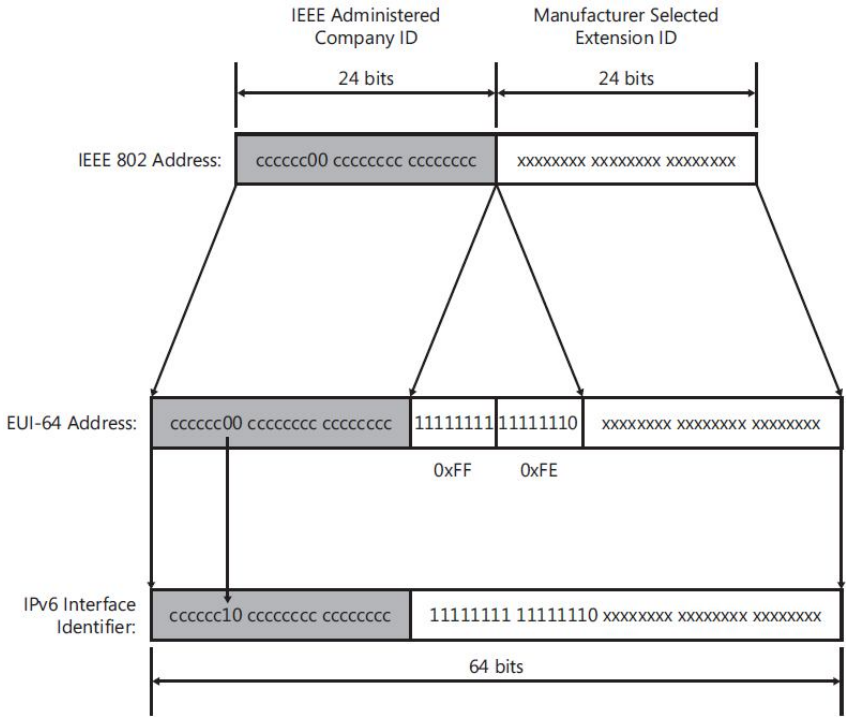

圖 2-9、EUI-64 位址轉換

2.3 SCTP

串流控制傳輸協定（Stream Control Transmission Protocol；SCTP）於 2000 年十月由 IETF 的信號傳輸(SIGTRAN)工作組定義的一種傳輸協定,並且被規範在 RFC4960[10]標準中。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（參考圖 1-1)。由於 TCP 可靠但額外負擔（overhead）大，而 UDP 負擔小卻不可靠，因此新提出的 SCTP 透過結合 TCP 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，幾乎可以完全代替舊有 TCP 的傳輸，也能夠部分代替 UDP 的傳輸。根據表 2-1[25]，可以明確比較出三種協定之間的差異。有別以往 TCP 在傳輸實際資料前需要在兩端點間建立一條連線（Connection），SCTP 改用更廣義的關聯（Association）概念。即兩端點間連線不再只是單一通道，中間傳輸可以有多組 IP 位址或是多條的資料流，透過關聯讓兩端點能夠相互的傳輸，藉此達到所謂的多重定址(Multi-homing)或是多資料流(Multi-streaming)傳輸特性,如圖 2-10。

表 2-1、SCTP 和 TCP 及 UDP 比較表

Services/Features	SCTP	TCP	UDP
Full-duplex data transmission	yes	yes	yes
Connection-oriented	yes	yes	no
Reliable data transfer	yes	yes	no
Partially reliable data transfer	optional	no	no
Ordered data delivery	yes	yes	no
Unordered data delivery	yes	no	yes
Flow and congestion control	yes	yes	no
Explicit congestion notification support	yes	yes	no
Selective acks	yes	optional	no
Preservation of message boundaries	yes	no	yes
Path maximum transmission unit discovery	yes	yes	no
Application data fragmentation/bundling	yes	yes	no
Multistreaming	yes	no	no
Multihoming	yes	no	no
Protection against SYN flooding attack	yes	no	n/a
Half-closed connections	no	yes	n/a

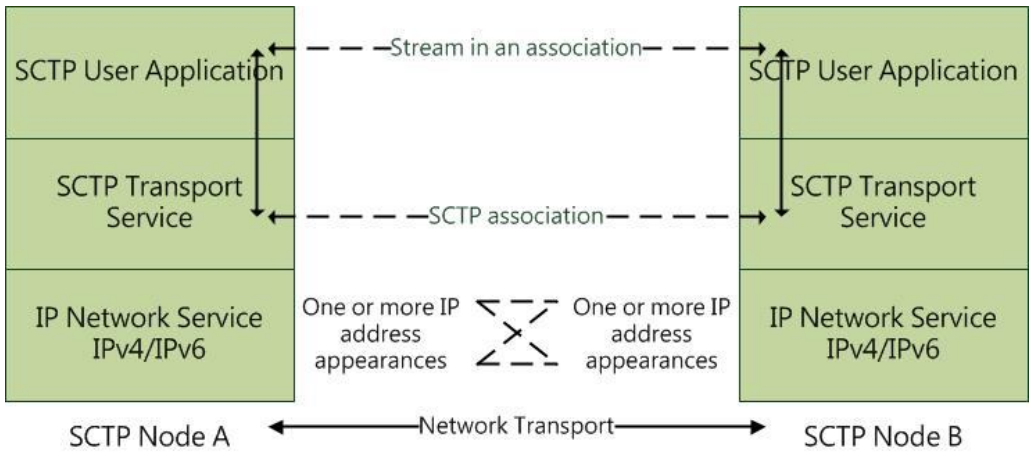

圖 2-10、SCTP 關聯傳輸

2.3.1 區塊綁定（Chunk Bundling）

TCP 是一種位元組導向（byte-oriented）的通訊協定，它會將來自上層的訊息封裝後，以位元組資料流的形式傳輸，因此所有訊息的邊界就會變得模糊不清。相對地，SCTP 提供的是訊息導向（message-oriented）通訊協定，以訊息作為傳輸單位。但是，假設來自上層的訊息大小超過路徑最大傳輸單元（Maximum Transmission Unit；MTU），則仍會被分割成數段，在目的端才會重組成一個訊息。圖 2-11 為 SCTP 標頭格式，可分為一般標頭（Common Header）及區塊（Chunk）兩部分，而區塊又分成資料區塊（Data chunk）或控制區塊（Control chunk）。一般標頭主要包含有來源端與目的端埠欄位，Verification Tag 欄位是一個 32 位元的亂數號碼，用來對應到一個關聯，在連線期間作為辨識之用，最後則是 32 位元的檢查碼。

在 TCP 中，它的封包標頭與封包格式都是固定，毫無彈性可言，然而，對 SCTP 來說，除了基本標頭欄位以外，所有的控制資訊及資料都被定義成一個個的區塊，每一個區塊大小是以 32 位元組為單位，控制區塊必需安排在資料區塊之前。另外，為了減輕 SCTP 封包標頭傳輸造成額外的負擔，SCTP 可以讓多個資料區塊或是不同類型

的控制區塊放在同一個 SCTP 封包裡，甚至是重送的資料區塊和準備傳送的資料區塊也可以綁在一起傳送。而封包內含區塊的數量並未限制，僅需遵守 MTU 大小的規範[26]。SCTP 支援多達 255 種區塊，目前僅定義其中的 14 種（參考表 2-2），因此具有相當大的擴充性與使用彈性，我們可以從 Type 欄位指定該區塊的形態。

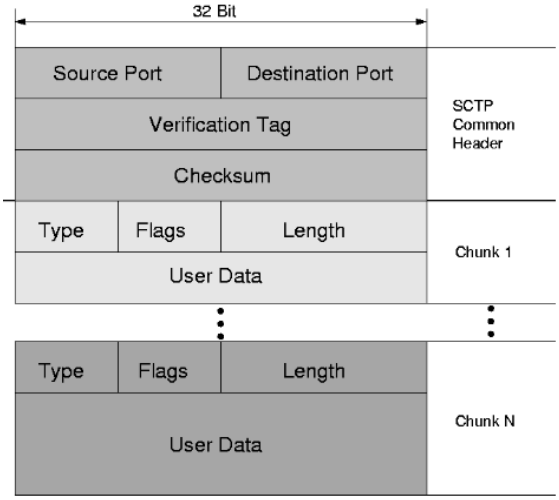

圖 2-11、SCTP 標頭格式

表 2-2、區塊的形態

ID Value	Chunk Type
0	Payload Data (DATA)
1	Initiation (INIT)
2	Initiation Acknowledgement (INIT ACK)
3	Selective Acknowledgement (SACK)
4	Heartbeat Request (HEARTBEAT)
5	Heartbeat Acknowledgement (HEARTBEAT ACK)
6	Abort (ABORT)
7	Shutdown (SHUTDOWN)
8	Shutdown Acknowledgement (SHUTDOWN ACK)
9	Operation Error (ERROR)
10	State Cookie (COOKIE ECHO)
11	Cookie Acknowledgement (COOKIE ACK)
12	Reserved for Explicit Congestion Notification Echo (ECNE)
13	Reserved for Congestion Windows Reduced (CWR)
14	Shutdown Complete (SHUTDOWN COMPLETE)
15 ~ 62	Reserved by IETF
63	IETF - defined Chunk Extensions
64 ~ 126	Reserved by IETF
127	IETF - defined Chunk Extensions
128 ~ 190	Reserved by IETF
191	IETF - defined Chunk Extensions
192 ~ 254	Reserved by IETF
255	IETF - defined Chunk Extensions

另外，SCTP 除了承接像 TCP 一般的有序傳輸以外，還提供亂序傳輸的服務。如圖 2-12 所示，資料區塊包含一個 U 欄位，U 指的是 Unordered bit。假設此位元被設為 1，表示這是一個亂序傳輸的資料區塊，因此不會分配資料流內的序號（Stream Sequence Number）給資料區塊，接收端則會忽略該欄位。傳輸序號（Transmission Sequence Number；TSN）表示每一個資料區塊的序號，而資料流編號（Stream Identifier）則定義下一個使用者資料屬於哪一個資料流。透過這些欄位，SCTP 可以達到兩個層次的亂序傳輸概念，分別為資料流的亂序，即資料流之間彼此獨立且沒有相依性；而資料流內部訊息可以選擇依序傳輸，抑或是亂序傳輸的方式。

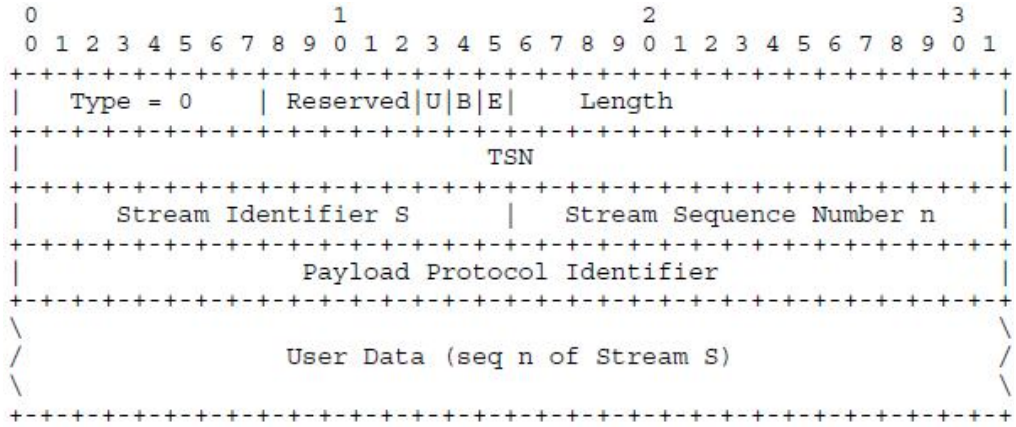

圖 2-12、資料區塊格式

2.3.2 四方交握（4-way Handshake）

在 TCP 協定中，採用三方交握（參考圖 2-5）的方式來建立連線，但是其中卻存在有安全性的漏洞，導致 TCP/IP 網路很容易受到阻斷服務（Denial-of-Serves； DoS）攻擊[27]。原因在於，三方交握機制本身是半開放（half-open）模式，當惡意的攻擊者有意攻擊伺服器時，會先發出大量的 SYN 封包，向伺服器請求連線。然而，當伺服器回覆 SYN/ACK 封包後，卻遲遲等不到用戶端回應 ACK 封包，即第三次握手無法完

成，造成伺服器需消耗大量的 CPU 時間和記憶體來維護 TCB（Transmission Control Block）列表。而伺服器還會持續重送 SYN/ACK 封包，等待攻擊者的回應。對一般的用戶端來說，伺服器因忙著服務攻擊者偽造的請求，所以伺服器已經處於失去回應的狀態。

SCTP 則是採用四方交握（4-way Handshake）機制，如圖 2-13[28]，提供 cookie 驗證的機制，且伺服器端不用建立任何 TCB 列表來儲存連線資訊及分配任何資源，藉此預防大量 SYN 封包的攻擊。其連線建立的過程為，首先，用戶端發送 INIT（Initial）封包請求連線，伺服器則回應 INIT-ACK 封包，其中夾帶 cookie 指出要建立關聯所需要的全部狀態資訊。用戶端在接收到 INIT-ACK 封包後以 COOKIE-ECHO 訊息的方式發送回伺服器，然後，伺服器會根據接收到的 cookie 來建立關聯，並回送 COOKIE-ACK 確認關聯已建立。根據上述的方式，是將狀態資訊儲存在用戶端，而非伺服器本身，在收到 COOKIE-ECHO 後，才會分配所需要的資源。因此，即使接收到再多的 INIT 封包，接收端也不會有過多的負擔，如此就不會像 TCP 容易遭受 DoS 攻擊。

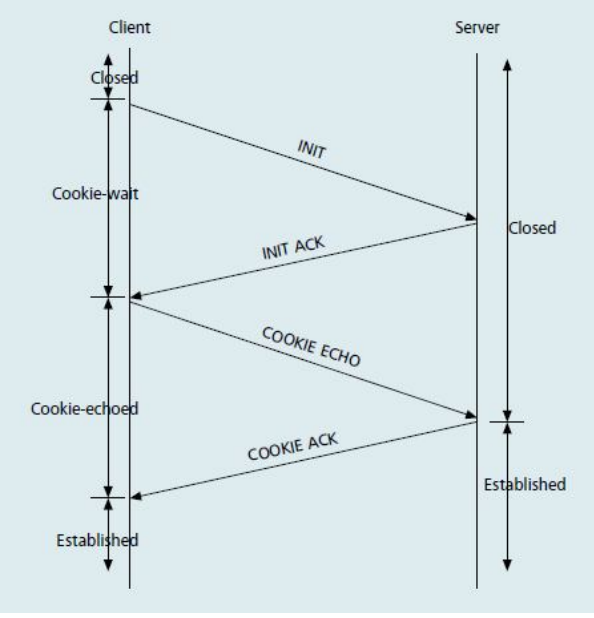

圖 2-13、SCTP 四方交握

2.3.3 多重定址（Multi-homing）

SCTP 提供多重定址的功能，當主要路徑（Primary Path）失效時，可以使用備援路徑（Redundancy Path）傳送資料。如圖 2-14 所示，支援 SCTP 的兩端點可以藉由關聯建立多條路徑。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，而傳送端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；如果主要路徑發生斷線或是故障的情況，傳送端會從備援路徑中選擇一條作為新的主要路徑。透過多重定址的機制，可以達到錯誤時快速復原的功能，增強網路服務的穩健性，也就是經由路徑之間的切換，進而提高網路的容錯能力。

圖 2-14、Multi-homing

Junichi Funasaka 等學者[29]針對 Shigeru Kashihara 等學者[30]所提出的演算法作修改。由於標準的 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，但是對於即時資料傳輸來說，假設主要路徑的封包遺失率或延遲時間品質下降，則應該立即作路徑的切換，而不是等到路徑失效後才切換。所以標準策略儼然不適合即時資料的傳輸，因此作者提出一個新的路徑交換演算法用來降低傳輸延遲且增加資料傳輸到達率。利用計算傳輸路徑的瓶頸頻寬與傳輸延遲時間來作判斷，然後根據相異的四個不同案例討論。即(1)只有主要路徑滿足需求頻寬、(2)只有備用路徑滿足需求頻寬、

(3)兩種路徑皆滿足需求頻寬及(4)兩種路徑皆不滿足需求頻寬。最後實驗結果證明該演算法對即時傳輸應用是有效的。但是，假設路徑的傳輸頻寬會經常性變化，則無法達到應有的效果。Guanhua Ye 等學者[31]提出一個新的 SCTP 架構，稱作 IPCC-SCTP（Independent per Path Congestion Control for SCTP）。標準 SCTP 的擁塞控制是針對一個關係（per association）作為判斷依據，但是在多重路徑傳輸環境下，每一條路徑狀況皆不一樣，不能用統一的控制觀點來操作。因此作者改用以單一路徑（per path）來作擁塞控制，利用 PSN（Path Sequence Number）的概念可以達到單一路徑擁塞控制。模擬結果指出，IPCC-SCTP 不僅克服多重路徑下擁塞控制的問題，也適合用在多樣的多重路徑機制應用。

2.3.4 多資料流（Multi-streaming）

有別於 TCP 使用單一資料流的傳輸方式，SCTP 可以針對應用層的不同需求，提供多條資料流進行資料傳輸。傳送端與接收端在建立關聯時，會先協商資料流的數量，就如同 TCP 同時在兩端點建立多條連線。對於某些獨立且不具相依性的資料，可以由傳送端區分為多條資料流進行傳輸，而資料流中每一個封包都會分配一組資料流序號。當同一資料流內發生封包遺失或延遲的狀況，則不會影響到其他資料流，因而減少傳統 TCP 常出現的 HoL（Head-of-Line） Blocking 問題[28]。

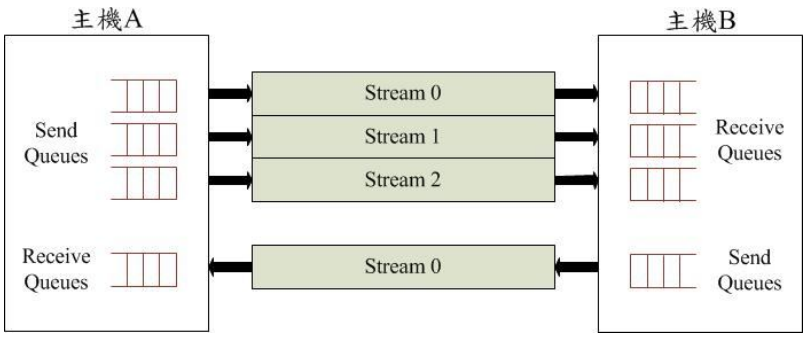

圖 2-15、Multi-streaming

根據圖 2-15 所示[32]，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 間的資料流 1 發生封包遺失，而資料流 1 等待封包重送的期間，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。舉例來說，多媒體的資料可以切割成不同的多媒體檔案，分別指派給不同的資料流傳輸。或者像我們瀏覽網頁時，網頁資料通常包含純文字敘述、圖片或聲音檔等。如果以 TCP 進行傳輸，過程中如有封包遺失，那之後的傳輸就必須等待封包重傳之後，才得以繼續。然而，如果是使用 SCTP，則可以減低等待傳輸的情況發生。

Kim 等學者[33]試著將 IPTV（Internet Protocol Television）建構在 SCTP 上，透過實驗比較 HoL Blocking 對 TCP 與 SCTP 傳輸效能的影響。SCTP 連線中採用三條資料流，結果證明多資料流的 SCTP 能克服 HoL Blocking 問題。當路徑封包遺失率為 10%時，SCTP 比 TCP 多 15%的傳輸率。最後結論為網路傳輸遺失率越大，SCTP 與 TCP 的傳輸率相差越多。而 Atiquzzaman 等學者[34]表示，當大量的手持式設備存在於無線網路環境中，則各個接收端暫存器容量有限；在此情況下，SCTP 比單一資料流的 TCP 具有較佳的傳輸率，進而證明 SCTP 多資料流機制可以降低接收端暫存器的需求，因此對於無線網路傳輸有很大的益處。

2.3.5 路徑監測（Heartbeat）

由於 SCTP 具有路徑備援的概念，藉此可以提高網路容錯的能力。然而，並非所有備援路徑皆會隨時進行著封包的傳送。因此，假設某路徑處於閒置的狀態時，就會有適當的路徑監測訊息產生，經由該路徑傳送到另一端節點。而此節點會立即回覆相對應的確認訊息。Heartbeat 傳輸間隔時間可以依使用者要求而有所調整，一般預設為 30 秒傳送一次；傳送 5 次沒有回應，則判斷該路徑失效。SCTP 可以藉由這種機制

監測路徑的可用情況，並確保路徑處於正常狀態。而且透過 Heartbeat 傳輸，也可以很精確的測量路徑來回的延遲時間（Round Trip Time；RTT），或者是運用在其他效能的量測。與 TCP 相比，雖然 TCP 具有類似 keep-alive 的機制，但這並不是傳輸層認可的機制，需要由特定的應用程式加以控制。

2.3.6選擇性回應（SACK）

當關聯建立完成後，兩端點間會開始傳送資料區塊。不論是有序傳輸或亂序傳輸，接收端的確認封包皆是以 TSN 為單位，並不提供單獨資料流的確認機制。SACK（Selective Acknowledgement）的研究在 TCP 方面已經相當成熟，但是網路上 TCP 傳輸節點並不完全支援這個機制[35]，因此 SCTP 明確的將 SACK 定義成標準機制。在收到資料區塊後，接收端會回應 SACK 區塊，用來確認每一個收到的資料區塊。傳統的 TCP 回應機制只能確認連續收到的位元組資料，而 SACK 則可以確認不連續的資料區塊，SACK 能完整的描述接收端資料區塊的接收狀態給發送端，因此有利於發送端作出傳輸資料的判斷。當封包遺失的情況下，SACK 會回應給發送端遺失且需要重傳的封包 TSN。然而，即使 SCTP 發現 TSN 有缺口（gap）或順序錯誤，仍會發送後面的資料區塊；與此相比，TCP 則是回覆已接收到的封包序號給發送端，而當 TCP 發現封包序號有缺口時，會等到該缺口補上後，才會傳送剩餘的封包。

2.4 IEEE 802.15.4

IEEE 802.15.4[36][37]低速無線個人區域網路（Low-Rate Wireless Personal Area Networks；LR-WPAN）標準，是由電機電子工程師學會（Institute of Electrical and Electronics Engineers；IEEE）在 2003 年所提出的標準，致力於實體層（Physical Layer；

PHY）與媒體存取控制層（Media Access Control Layer；MAC）的規範，如圖 2-16 所示。媒體存取控制層為資料鏈結層的子層，此子層與實體層相連接，負責定義傳輸媒體存取的方式。IEEE 802.15.4 具有低資料傳輸速率、低功率消耗、低成本、低複雜度和短距離傳輸等特性，其應用目標在於家庭網路（Home Networks）、汽車網路（Automotive Networks）、工業網路（Industrial Networks）和遠端量測（Remote Metering）等環境下。而 IEEE 802.15.4 標準中，定義了在一個 IEEE 802.15.4 網路下，會具有兩種不同類型的設備，分別為全功能設備（Full Function Device；FFD）和精簡功能設備（Reduced Function Device；RFD）。經由數個設備所構成的無線個人區域網路（Wireless Personal Area Networks；WPAN），稱作個人操作範圍（Personal Operating Space；POS），其中至少要有一個全功能設備作為協調者的角色，用以組織網路。一個全功能設備可以與其他全功能設備或是精簡功能設備互相溝通，但是精簡功能設備僅能和一個全功能設備溝通。因此，在定義上，精簡功能設備通常會是一個功能較為簡單的應用裝置，例如燈光開關、溫度感測裝置等。由於其裝置不會傳送大量的資料，並且同一時間只與一個全功能設備溝通，所以只需要較小的記憶體容量與資源。

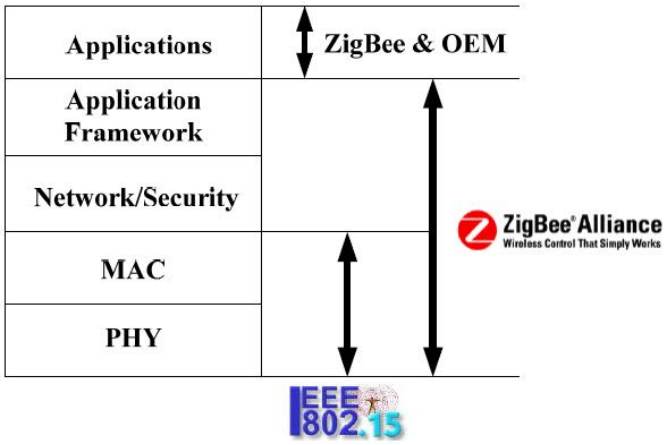

圖 2-16、ZigBee/IEEE 802.15.4 協定堆疊架構

2.4.1 實體層

在 IEEE 802.15.4 的標準中，實體層主要負責開關無線電收發器、選擇通道、偵測電力和傳送與接收封包等功能。除此之外，實體層採用直接序列展頻（Direct Sequence Spread Spectrum；DSSS）技術，可使用三種不同的頻段[38]，藉此滿足全球不同國家對於免使用執照頻道（License-Free Band）的規範。頻道總共有 27 個，在相同頻段範圍內的頻道可以互相轉換，甚至允許動態選擇頻道：

1. 歐洲地區的 868.0~868.6MHz 頻段，具有一個通道，編號為 0，提供 20Kbps 傳輸速率。
2. 美國地區的 902.0~928.0MHz ISM 頻段，具有 10 個通道，編號為 1~10，各個通道提供 40Kbps 傳輸速率。
3. 全球通用的 2.4~2.4835GHz ISM 頻段，具有 16 個通道，編號為 11~26，各個通道提供 250Kbps 傳輸速率。

2.4.2 媒體存取控制層

另外，媒體存取控制層主要定義了兩種傳輸模式，分別為信標網路（Beacon-enabled Network）與無信標網路（Non Beacon-enabled Network）。無信標網路採用基本的 CSMA/CA（Carrier Sense Multiple Access/Collision Avoidance）[39]機制來做競爭，即當某個設備想傳輸資料，它必須等待一個隨機時間，然後偵測頻道是否閒置。如果是閒置狀態則馬上傳送資料，否則再等待一個隨機時間後再嘗試。而在信標網路狀態下，包含有超級訊框（Superframe）的架構，超級訊框結構是由信標（Beacon）內容指定，協調者會定時發送信標，信標與信標之間最多可分為 16 個時槽（Slot），其架構如圖 2-17[36]。所有設備都是以信標作時間同步，並且在這 16 個時槽之中，

選擇一個時槽作為資料的傳遞。

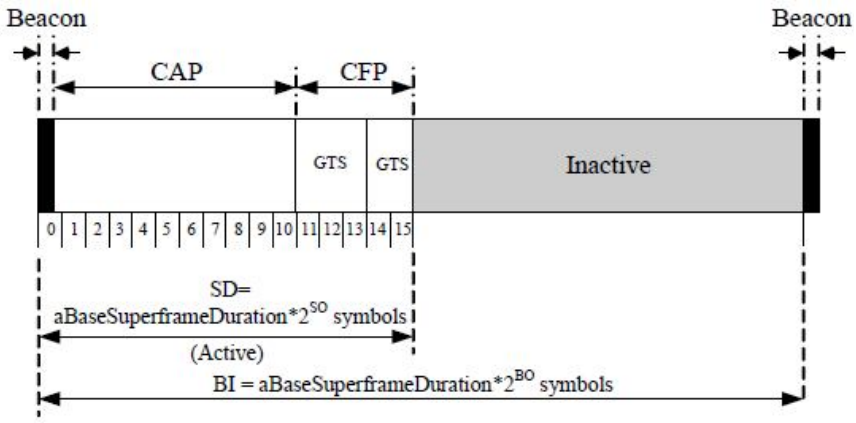

圖 2-17、超級訊框結構[36]

根據圖 2-17，超級訊框包含活動週期（Active Period）跟非活動週期（Inactive Period），設備只能在活動週期進行資料的傳遞，非活動週期則是進入省電模式。然而，超級訊框又以有無使用保證時槽（Guaranteed Time Slots；GTS）來區別。具有保證時槽的超級訊框，在活動週期又可區分為競爭存取週期（Contention Access Period；CAP）與無競爭週期（Contention Free Period；CFP）。無競爭週期是由數個保證時槽所組成的，一次最多可以分配七個保證時槽。在競爭存取週期，任何設備都會使用 CSMA/CA 機制來做競爭。當經過互相競爭之後，搶得時槽者則可以開始傳遞資料；反之，若是設備以預先要求的方式，經由協調者的同意，而在無競爭週期分配到保證時槽，只要輪到該設備被分配的保證時槽時，資料則會直接傳送出去，不需要經過相互的競爭。

其資料傳輸模式定義了以下三種模型[40]：

1. 設備傳送資料給協調者

   在信標網路中，如果設備要傳送資料給協調者，必須先取得信標與協調者做同步，接著使用時槽型（Slotted）CSMA/CA 機制競爭頻道使用權。設備會在

得到使用權後開始傳送資料給協調者，在協調者成功接收資料之後，協調者可以選擇性傳送回覆訊框（Acknowledgment Frame；ACK Frame）給設備，如圖 2-18(a)；而無信標網路中，當有設備要傳送資料給協調者時，該設備會以非時槽型（Unslotted）CSMA/CA 機制競爭頻道使用權。同樣的，設備在得到使用權後即傳送資料給協調者，待協調者成功接收資料，則可以選擇性傳送回覆訊框給設備，如圖 2-18(b)。

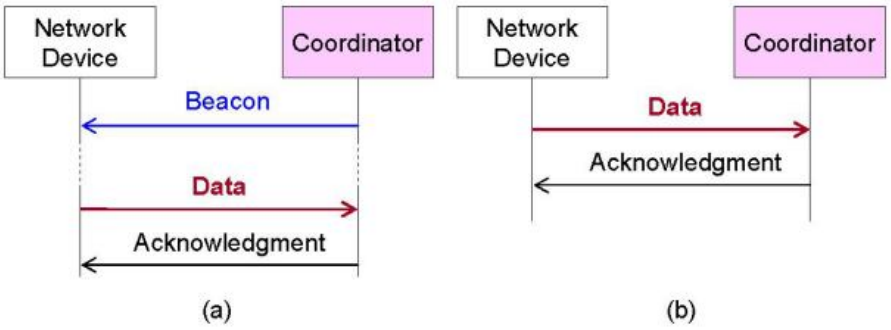

圖 2-18、設備傳送資料給協調者

2. 協調者傳送資料給設備

在信標網路中，協調者會在定期發送的信標中告知設備有資料要傳送。當設備收到信標後會先判斷協調者是否有資料要送給它，如果確認有資料要接收，則設備會使用時槽型 CSMA/CA 機制競爭頻道使用權。設備取得使用權後則傳送資料要求訊框（Data Request Frame）給協調者，待協調者成功收到資料要求訊框，會傳送一個回覆訊框給設備。接著須確認是否有其他資料等待傳送給該設備。若有，則透過時槽型 CSMA/CA 機制競爭頻道使用權；若無，即直接開始傳送資料。最後，設備會傳送回覆訊框給協調者，表示整個傳送動作完成，如圖 2-19(a)；無信標網路中，當協調者有資料要傳送給設備時，會先將資料儲存起來，設備會定期詢問是否有資料要傳輸。其做法為，設備會以非時槽型 CSMA/CA 機制的方式，傳送資料要求訊框給協調者。

在協調者成功接收後，會回傳一個回覆訊框確認是否有資料在等待傳送。若有，則協調者會使用非時槽型 CSMA/CA 機制進行資料的傳送；反之，如果沒有資料在等待，協調者會傳送零負載（payload）的資料訊框（Data Frame）給設備。當設備傳送一個回覆訊框給協調者，至此，整個傳送的動作到這裡完成，如圖 2-19(b)。

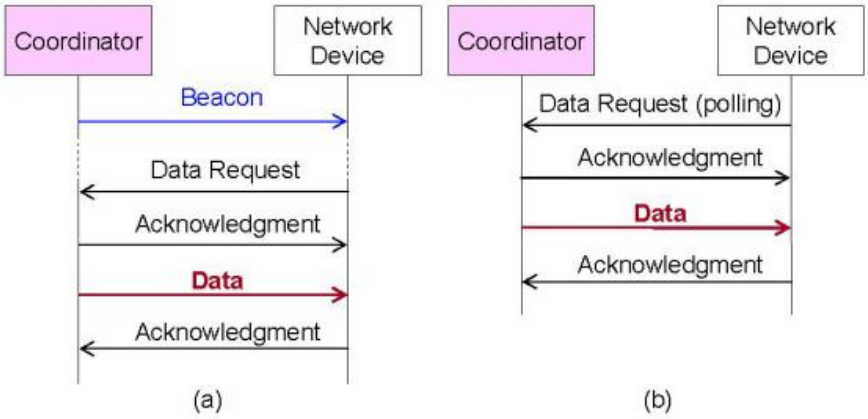

圖 2-19、協調者傳送資料給設備

3. 設備與設備間對等傳送

在點對點拓樸中，任一設備可以與傳輸範圍內的其他設備直接溝通。根據此網路型態，為了讓設備之間能夠正常的傳送資料，設備必須持續的保持在接收狀態，即無法進入睡眠模式，並且需要與其他設備保持同步。如此，則可以使用非時槽型 CSMA/CA 機制進行資料的傳送。

2.5 ZigBee

ZigBee[41][42]的命名，源自於蜜蜂在發現花粉時，展現如同 ZigZag 形狀的舞蹈。然而，看似隨意在跳的花舞，實際上是將有花和蜂蜜的地方，正確地傳達給其他同伴知道。ZigBee 早先亦被稱為 HomeRF Lite、RF EasyLink 或 FireFly 無線通訊技術，目前

統一稱為 ZigBee。ZigBee 通訊協定主要是建立在 IEEE 802.15.4 標準之上。根據圖 2-16，IEEE 802.15.4 標準規範通訊協定底部兩層，而由 ZigBee 聯盟（Alliance）[43]所定義的標準則是規範其上部各層。ZigBee 是一種具有低傳輸速率、短距離、低成本和低耗能等特性的技術，主要應用在監測與控制網路方面。ZigBee 支援主從式（Master-Slave）或對等式（Peer-to-Peer）運作，具有高擴充性。一個 ZigBee 設備可以對應 255 個設備連結，而單一 ZigBee 網路內最多可同時擁有 65535 個設備連結，並且採用 128 位元高階加密標準（Advanced Encryption Standard；AES）[44]的加密技術和循環冗餘校驗（Cycle Redundancy Check；CRC）之錯誤檢查碼機制，因此也具有高安全性及高可靠度。如圖 2-20 所示，目前常見的應用是以家庭自動化為設計目標，另外也包含智慧型大樓、工業與環境控制及個人醫療照護等應用。

圖 2-20、ZigBee 應用領域

ZigBee 聯盟是一個由許多廠商所組成的產業聯盟，致力於制定開放性全球通用的標準，並且協助業界推廣可靠度高、低成本、低耗能和無線網路連結的監測與控制等功能之產品。而 ZigBee 聯盟是成長快速的非營利性產業聯盟，其成員包括全球各大

半導體廠商、技術供應商和原始設備製造商（Original Equipment Manufacturer；OEM）等，任何單位皆可以申請加入。此外，ZigBee 聯盟已經具備產品認證的機制。產品在貼上 ZigBee 認證標誌之前，都必須經過嚴格的品質與運作標準測試。唯有皆以 ZigBee 標準為基礎所製造出的無線網路產品，才能讓跨廠商的產品可以彼此互通。因此，選擇以 ZigBee 標準進行開發的廠商，無需再投入大量成本開發獨自的 ZigBee 通訊協定，可以專心致力於產品的創新，增加各類不同的應用。下圖 2-21 是 ZigBee 聯盟所制定標準的通訊協定堆疊架構圖[45]。

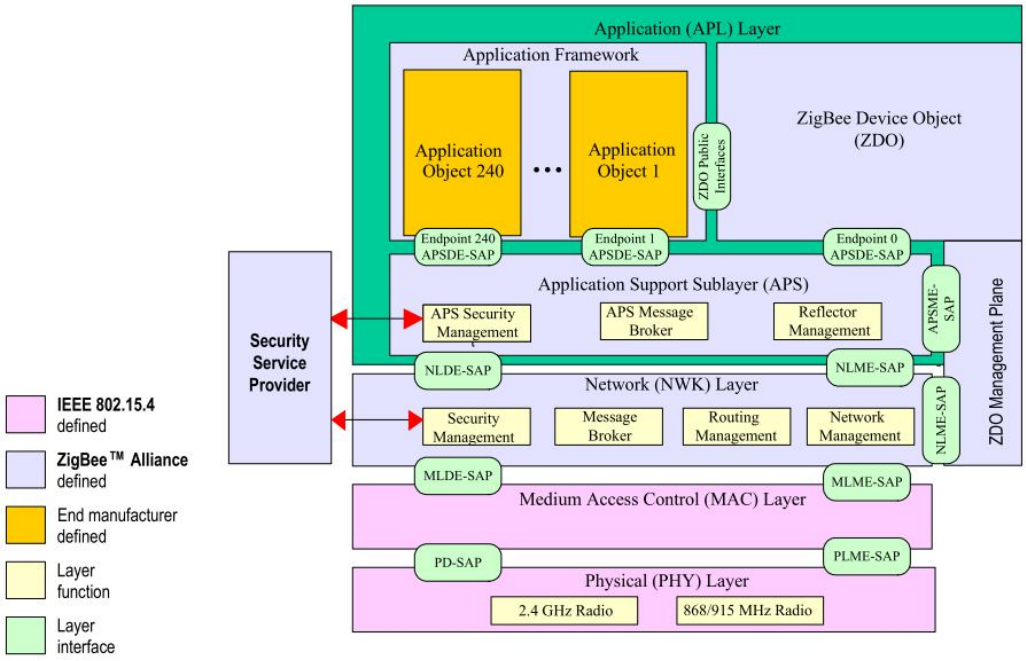

圖 2-21、ZigBee 通訊協定堆疊

2.5.1 網路層

網路層（Network Layer；NWK）主要負責網路的建立與管理，並且具有自我組織和自我修復的功能，所謂自我組織功能即是網路內節點有能力去偵測到其他存在的節點，並且將這些節點加入現有的網路內，而不需要人為的介入；自我修復指的是當網

路發生錯誤時，同樣不需要人為的介入即可自我復原。延續 IEEE 802.15.4 中對設備的分類，在 ZigBee 網路層也對設備定義了三種角色，分別是：ZigBee 協調者（ZigBee Coordinator；ZC），負責網路的建立與位址的分配，必須透過全功能設備才能完整地實作出來；一個 ZigBee 網路通常僅具有一個 ZigBee 協調者。ZigBee 路由器（ZigBee Router；ZR），主要負責尋找、建立、以及修復資料封包路由路徑（Routing Path），然後負責轉送資料封包，本身也是全功能設備，可以接受其它設備加入和分配網路資源管理能力。ZigBee 終端設備（ZigBee End Device；ZED），只能選擇加入已經形成的 ZigBee 網路，可以傳送與接收資料封包，但是並不具有轉送封包的能力，視需要可選擇使用全功能設備或精簡功能設備來實作。另外，ZigBee 網路層提供了可靠安全的傳輸，其支援的網路拓樸又分為星狀拓樸（Star）、樹狀拓樸（Tree）及網狀拓樸（Mesh）等三種典型網路拓樸，如圖 2-22 所示。

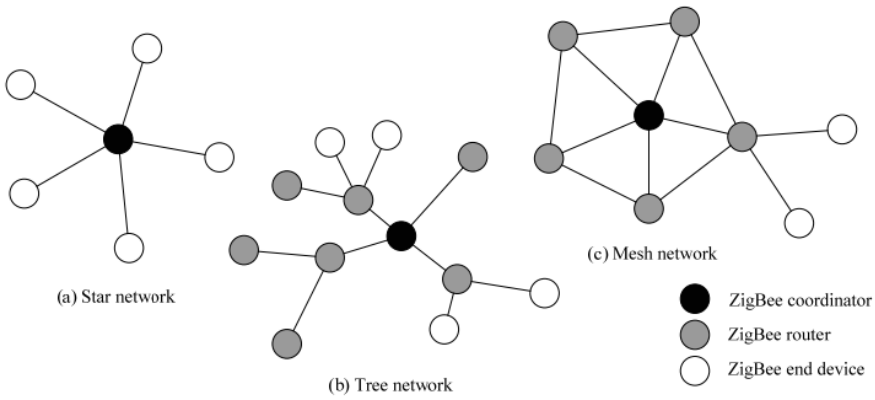

圖 2-22、ZigBee 支援網路拓樸

ZigBee 設備可以支援兩種定址方式，一個是 64 位元延伸位址，即 IEEE 位址；另一個是 16 位元短位址。基本上，在 ZigBee 網路中，每一個設備都包含有自身獨一無二的 64 位元延伸位址，設備可以利用這個延伸位址互相溝通。而當設備加入任一 ZigBee 網路時，則會從其父設備上獲得 16 位元短位址，這是此 ZigBee 網路內的唯一位址。短位址經常用以取代延伸位址做通訊，藉此以減低記憶體空間和頻寬的浪費。

每個節點的路由表上存放著目的端設備與下一個將到達設備的位址，因此 ZigBee 網路中各個設備都必須具備明確且唯一的位址，以保證封包能正確到達目的端。

2.5.2 應用層

應用層（Application Layer；APL）主要分成三個部分，首先是與網路層連接的應用程式支援子層（Application Support Sub-Layer；APS），然後是 ZigBee 設備物件（ZigBee Device Object；ZDO）以及應用程式框架（Application Framework；AF）。ZigBee 應用層架構涵蓋了服務的觀念，而服務就是所謂的功能，如 ZigBee 風扇有可以調節葉片轉速的功能，即表示此 ZigBee 設備會提供這樣的服務。在 ZigBee 設備物件中的端點（Endpoint），就代表了這個設備的服務。將來透過這個端點，設備可以提供服務給他人使用或是使用他人的服務，以前例分別代表風扇本身與遙控器之間的通訊，所以端點很類似 Socket 的角色。當設備加入一個 ZigBee 網路時，ZigBee 設備物件會開始做初始化的動作，然後透過應用程式支援子層提供設備搜尋（Device Discovery）與服務搜尋（Service Discovery）機制，並且搭配事先定義的設備描述（Device Description）規範，將與自己相關的設備和服務，紀錄在應用程式支援子層的綁定表（Binding Table）內。在此以後，所有服務的使用都需要透過綁定表來查詢設備資訊。而應用程式框架下的應用程式配置文件（Application Profile）會根據不同的設備功能有不同的設備描述規範。

2.6 6LoWPAN

6LoWPAN（IPv6 over Low Power Wireless Personal Area Networks）[46][47]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。相較於其他基於 IEEE 802.15.4 的傳輸標準，如上述 ZigBee，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，相對於發展新的一個網路層協定，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。由於是使用 IPv6 技術，因此賦予 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。下圖 2-23 表示 6LoWPAN 的傳輸結構，當 6LoWPAN 網路內的精簡功能設備欲傳送訊息給網路外之 IP 設備時，首先必須將封包傳送給上層全功能設備，然後全功能設備會透過路由機制一層一層的將封包送給 6LoWPAN 閘道器（Gateway）。6LoWPAN 閘道器與 IPv6 網域相連接，經過封包底層的轉換，則可以使用 IP 位址將封包正確傳送到達目的地。

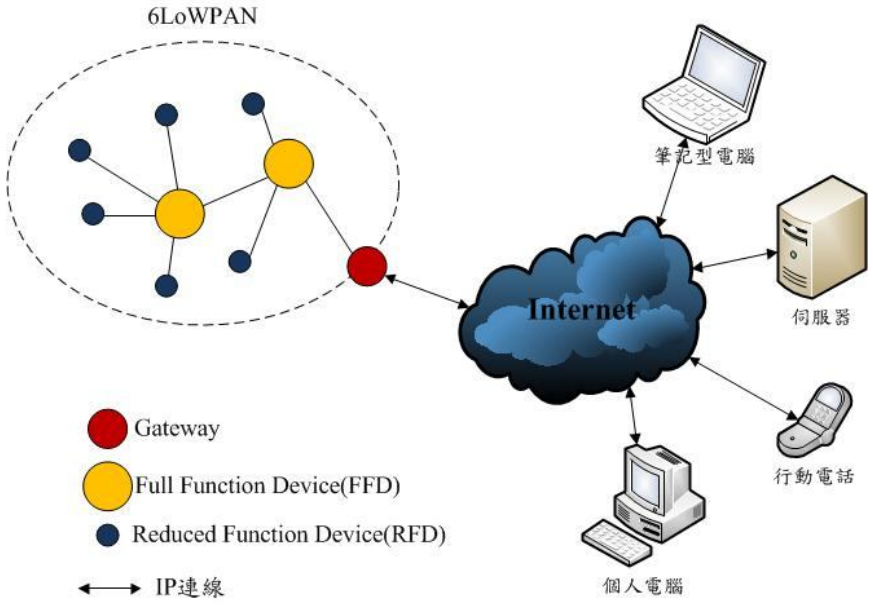

圖 2-23、6LoWPAN 結構

根據圖 2-24 所示，描述了 6LoWPAN 協定架構的參考模型。相較於 ZigBee 網路模型，ZigBee 和 6LoWPAN 都是建構在 IEEE 802.15.4 標準之上，而在上層部分，6LoWPAN 網路層則改採用 IPv6 協定。由於 IEEE 802.15.4 標準在媒體存取控制層定義的最大傳

送訊框大小為 127 位元組，而媒體存取控制層本身的標頭佔 25 位元組，所以負載大小剩餘 102 位元組。然而，假設選擇使用安全傳送方式，根據不同的演算法會再消耗不同大小的位元組，如 AES-CCM-128 需要 21 位元組，將導致留給負載最少只有 81 位元組。因此，一個完整的 IPv6 封包並不符合一個 IEEE 802.15.4 訊框，原因在於，IPv6 基本標頭為 40 位元組，扣除這 40 位元組，會只剩下 41 位元組供上層使用，然後再保留傳輸層 UDP 協定的 8 位元組標頭或是 TCP 協定的 20 位元組標頭，最後能供應用層傳送資料的位元組就變得所剩無幾。除此之外，在 IPv6 網路中，媒體存取控制層最小的最大傳輸單元需為 1280 位元組，但是 IEEE 802.15.4 最多只提供 102 位元組訊框，所以並不能一次完整的封裝 IPv6 封包。為了解決上述問題，讓媒體存取控制層與網路層能實現無縫連接，6LoWPAN 主要在這兩層之間加入一適應層（Adaptation Layer），用來完成封包切割與重組、標頭壓縮和封包繞路等工作[48]。

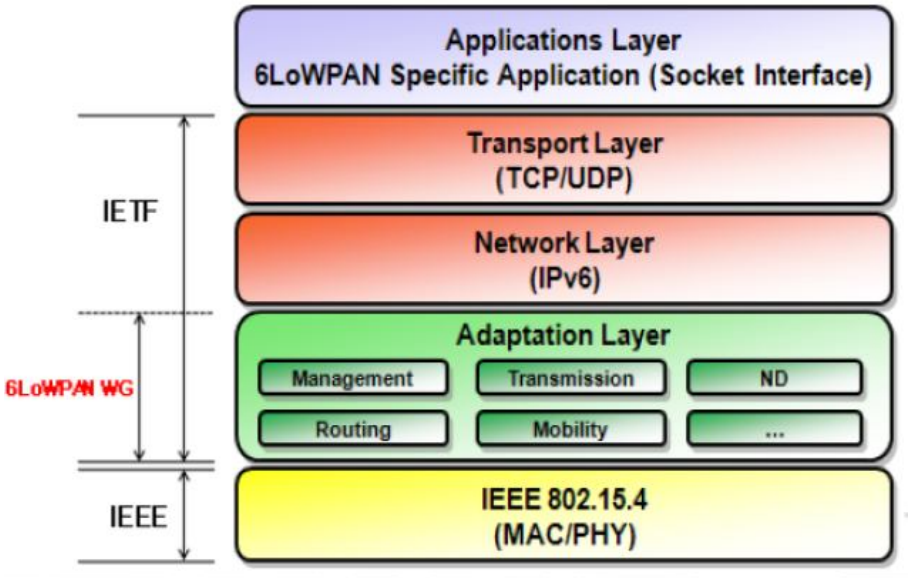

圖 2-24、6LoWPAN 協定架構

2.6.1 適應層

6LoWPAN 在標頭壓縮部分，定義了 HC1 編碼壓縮機制。對於 IPv6 標頭來說，其壓縮策略為：描述長度和位址的欄位，可以經由資料鏈結層標頭來推導或是透過一些簡單的計算公式取得實際的欄位值，所以可以被壓縮；另外，不管在適應層、網路層或傳輸層，皆存在一些相同的欄位值，這些欄位也都是被壓縮的目標。因此，為了降低封包傳送過程中，標頭所佔的負載量，透過壓縮上述的欄位，藉此可以增加資料傳輸量。下表 2-3 顯示在壓縮前後，標頭欄位的變化。

表 2-3、IPv6 標頭壓縮比較表

Header Field	IPv6 header length	6LoWPAN HC1 length	Explanation
Version	4 bits	-----	Assuming communicating with IPv6.
Traffic class	8 bits	1 bit	0 = Not compressed. The field is in full size. 1 = Compressed. The traffic class and flow label are both zero.
Flow label	20 bits		
Payload length	16 bits	-----	Can be derived from MAC frame length or adaptation layer datagram size (6LoWPAN fragmentation header).
Next header	8 bits	2 bits	Compressed whenever the packet uses UDP, TCP or Internet Control Message Protocol version 6 (ICMPv6).
Hop limit	8 bits	8 bits	The only field always not compressed.
Source address	128 bits	2 bits	If Both source and destination IPv6 addresses are in link local, their 64-bit network prefix are compressed into a single bit each with a value of one. Another single bit is set to one to indicate that 64-bit interface identifier are elided if the destination can derive them from the corresponding link-layer address in the link-layer frame or mesh addressing header when routing in a mesh.
Destination address	128 bits	2 bits	
HC2 encoding	------	1 bit	Another compression scheme follows a HC1 header.
Total	40 bytes	2 bytes	Fully compressed, the HC1 encoding reduces the IPv6 header to two bytes.

適應層另外支援封包切割與重組機制，當 IPv6 封包無法整個填入媒體存取控制層的訊框時，會將封包切割成數個部分，以便透過多個訊框把封包傳送給接收端，由接收端負責重組。根據圖 2-25，顯示在經過封包切割後，第一個部分切割標頭（Fragmentation Header）的結構，大小為 4 位元組。而接續在第一個部分之後的封包切割標頭，則多增加額外的 Offset 欄位；此欄位長度為 1 位元組，所以該標頭會佔

5 位元組。Datagram size 欄位用來指出原始 IP 封包在切割前完整的大小；從同一個封包切割而來的部分，該欄位會具有相同的值。而 Datagram tag 欄位則是用來辨識從哪一個原始封包所切割而來的；同樣的，由同一個封包切割取得的，此欄位值都會是一樣。

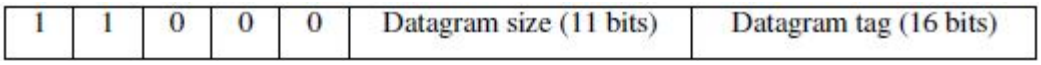

圖 2-25、第一個切割標頭

除此之外，為了支援第 2 層繞路與傳送 IPv6 封包，6LoWPAN 在適應層定義了網狀繞路標頭（Mesh Header），如圖 2-26 所示。因為 IEEE 802.15.4 標準包含 16 位元和 64 位元位址兩種，所以透過 Originator(O)與 Final destination(F)欄位值，其值為 1 表示使用 16 位元位址，或值為 0 則表示使用 64 位元位址，藉此來判斷標頭後方位址欄位。Hop left 欄位是用來限制傳送端與目的端之間最大的節點跳躍數，雖然 4 位元的 Hop left 只能提供跳躍 15 個節點，但是在一般網路環境下已然足夠；而 0xF 欄位值是保留用來指定額外的跳躍數，最多可以增加到 255 個節點。當執行繞路協定時，傳送端會在 Originator address 欄位放入自己的位址，且在 Final address 欄位填上最後目的端位址；同時針對媒體存取控制層訊框標頭，其 Source address 欄位同樣放入自己的位址，而 Destination address 欄位則是填下一個要傳送節點位址，如此可以將訊框傳送出去。假設是目的端節點收到該封包，目的端即處理掉封包；否則，該節點會將 Hop left 欄位值減 1，然後更改訊框標頭的 Source address 和 Destination address 欄位值，變成自己的位址與下一個要傳送節點位址。最後當 Hop left 欄位值等於 0 時，則丟棄該封包。

1	0	O	F	Hops left (4 bits)	Originator address (16-64 bits)	Final address (16-64 bits)

1	0	O	F	0xF	Hops left (8 bits)	Originator address (16-64 bits)	Final address (16-64 bits)

圖 2-26、網狀標頭

由上述標頭可以發現，不同的機制會有不同的標頭，而這些標頭可以透過前 2 位元來分辨。位元序列 11 表示是封包切割標頭，位元序列 10 則表示是網狀繞路標頭。為了支援與非 6LoWPAN 網路共存的環境，因此位元序列 00 選擇用來辨識是否為非 6LoWPAN 網路訊框。而位元序列 01 為發送標頭（Dispatch Header）使用，如圖 2-27 所示。根據表 2-4，它定義多個型態的發送標頭，後段 6 位元序列可以用來指出接下來標頭的型態。目前發送標頭只定義 5 種型態。

0	1	Dispatch (6 bits)

0	1	0x3F	Dispatch (8 bits)

圖 2-27、發送標頭

表 2-4、發送標頭位元序列表

01	000001	The following bits are IPv6 uncompressed header
01	000010	The following bits are IPv6 HC1 compressed encoding
01	010000	The following bits are broadcast header
01	111111	The following 8 bits are an additional field for dispatch value.

6LoWPAN 在這些標頭使用上是採取各自獨立的概念，即表示設備在傳送封包時只選擇加入需要的標頭。舉例來說，當設備想要傳送一個簡單的訊息封包，而且距離目的端只有一個跳躍數，在此情況下，切割標頭與網狀標頭就可以捨棄而不使用。然而，這些標頭的先後次序是有被定義的，次序分別為網狀標頭、廣播標頭、切割標頭及最後的負載，如圖 2-28 所示。

IEEE 802.15.4 frame	Mesh addressing	Broadcast header	Fragmentation header	Dispatch header	Compressed IP header	Payload

圖 2-28、6LoWPAN 標頭次序

2.6.2路由協定

根據路由決策所在的階層，6LoWPAN 中的路由機制可以分為兩類：Mesh-under 與 Route-over[49]，如圖 2-29 所示。Mesh-under 是在適應層進行路由決策；由於使用 IEEE 802.15.4 媒體存取控制層位址作為路由位址，所以為非 IP 路由協定。相反的，Route-over 則是在網路層執行路由決策，並且透過數個節點充當路由器，進而完成 IP 網路路由。

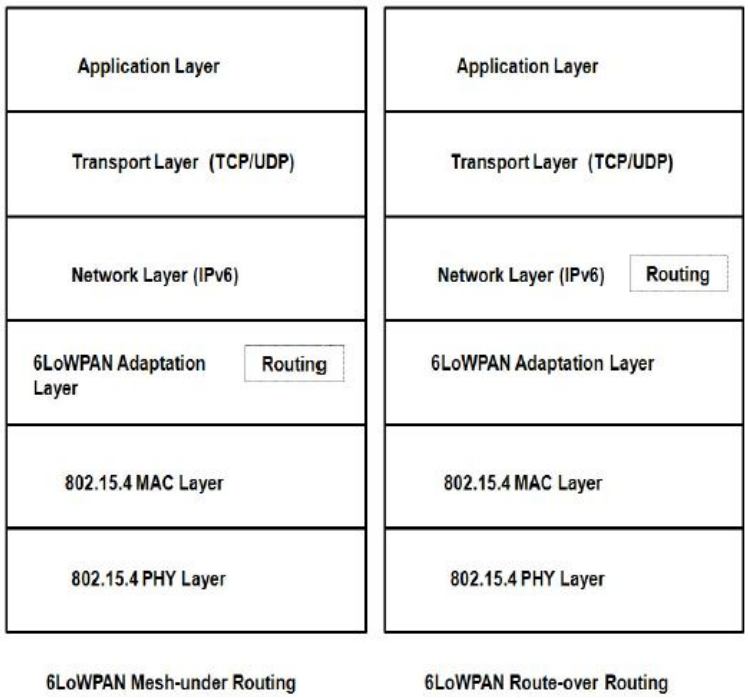

圖 2-29、6LoWPAN 路由決策階層

由於 6LoWPAN 設備的資源有限，所以在現有的路由機制中可選擇適用於

6LoWPAN 環境的機制相對較少[50]。AODV（Ad-Hoc On-demand Distance Vector）[51]路由協定具有簡易找尋路徑的特色，是少數適用於 6LoWPAN 環境路由。然而，為了更符合 6LoWPAN 環境路由，則必須針對 AODV 路由協定作些許的修改，LOAD（6LoWPAN Ad-Hoc On-demand Distance Vector Routing）[52]和 DYMO-low（Dynamic MANET On-demand for 6LoWPAN Routing）[53]就是以 AODV 為基礎發展的路由協定，如圖 2-30 所示。除此之外，還有階層式的 HiLow（Hierarchical Routing over 6LoWPAN）[54][55]路由協定（參考圖 2-31），也是針對 6LoWPAN 環境所發展而來的。

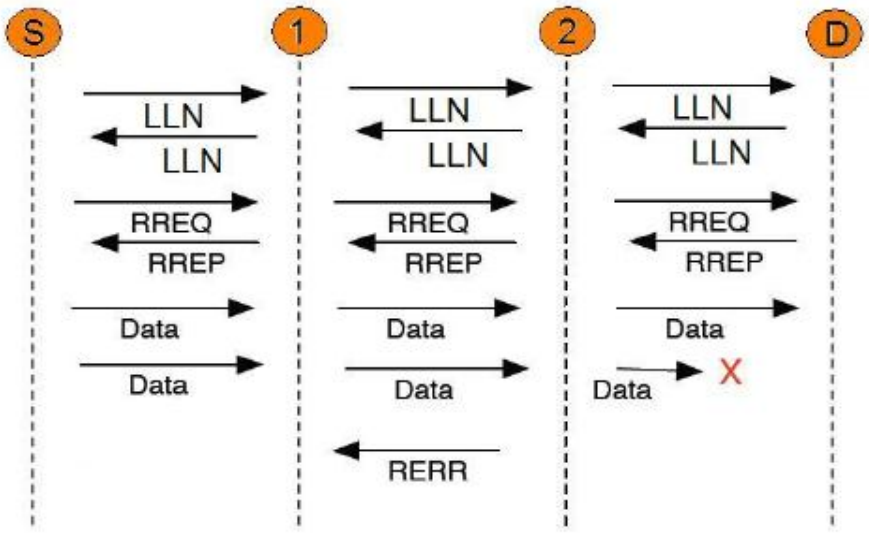

圖 2-30、LOAD 路由協定

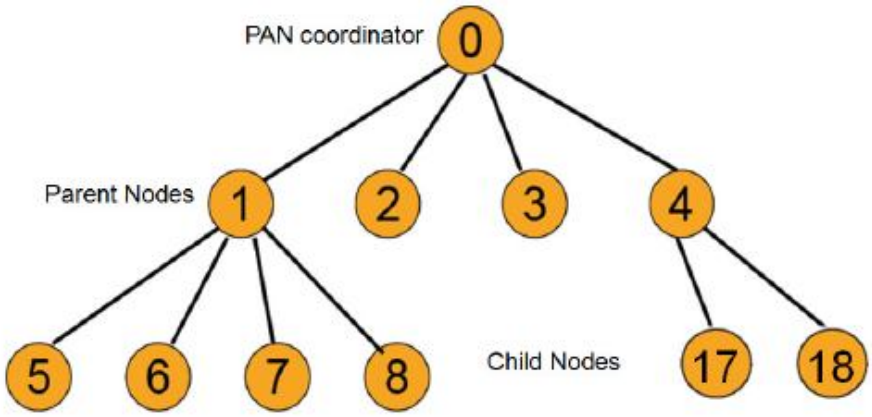

圖 2-31、HiLow 路由架構

第三章 系統實作

本論文的系統實作平台架構如圖 3-1 所示。選擇長高科技[56]的 DMA-2440XP 平台作為嵌入式系統移植平台，將扮演一個協調者的角色；而另外選擇一台安裝 Linux 作業系統的個人電腦來扮演伺服器的角色。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多重路徑傳輸。

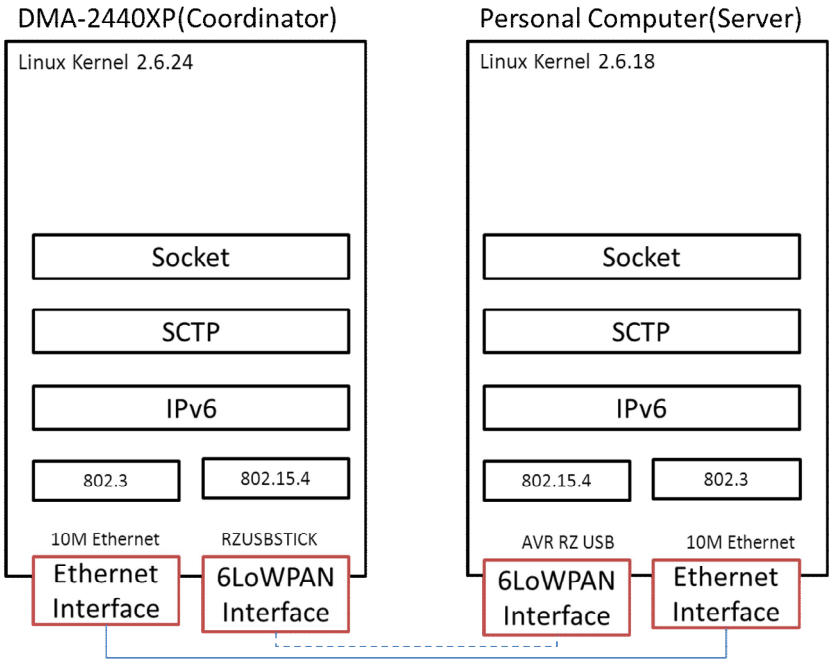

圖 3-1、實作平台架構

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 及七吋 LCD 面板等。圖 3-2 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，更另外提供圖形介面控制這兩個系統下各介面的範例程式，其中 Linux 系統的圖形介面採用 MiniGUI[57]。

圖 3-2、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[58]。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記憶體（FLASH）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 —— 對應於本研究，即是在 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。本研究使用的編譯器是 arm-gcc，它是 gcc 的 arm 版本，目前 Linux 作業系統主要是以 GCC（GNU Compiler Collection）[59]編譯器進行移植的。

本研究所使用的 SCTP 協定堆疊為 LKSCTP（Linux Kernel Stream Control Transmission Protocol）[60]，是於 2001 年期間所提出的計劃，它是一個 GNU General Public License（GNU GPL）授權開放原始碼軟體[61] ，主要目標為在 Linux 核心上實現 SCTP 通訊協定。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。由於平台上快閃記憶體量很小的關係，導致系統功能較特定且簡易，所以目前常見的嵌入式 Linux 系統並不包含本研究需要的 IPv6 與 SCTP 協定功能。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。首先本研究選擇 Linux 2.6.24 核心原始碼作為修改核心基礎，在終端機視窗鍵入 make menuconfig 命令，接著會出現核心配置的介面（參考圖 3-3），此時有很多配置的選項。根據本

研究的需求，我們勾選了 IPv6 與 SCTP 相關選項，依照平台快閃記憶體配置大小，最後調配出合適的嵌入式 Linux 核心。然後在終端機視窗鍵入 make zImage，透過 arm-linux-gcc-4.0.3 版編譯器作交叉編譯。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 作為快閃儲存記憶體，因此將 zImage 透過 Trivial File Transfer Protocol（TFTP）[62]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。

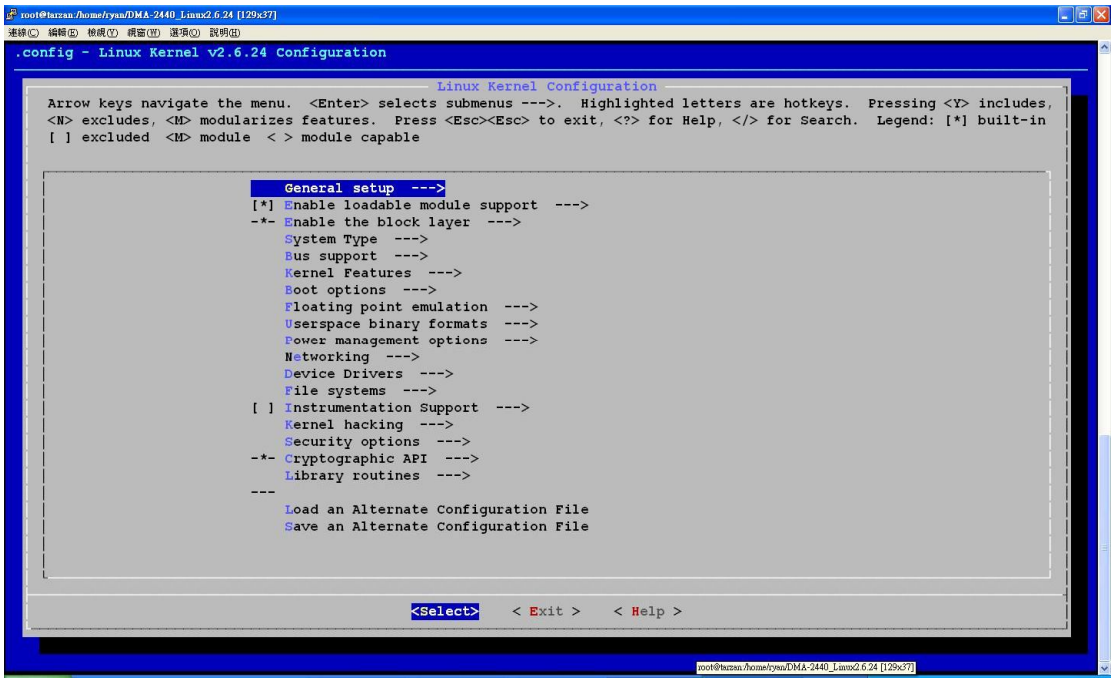

圖 3-3、核心配置畫面

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，檔案系統是用於定義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（The Second Extended Filesystem）、RAMDISK、Cramfs（The Compressed ROM Filesystem）[63]等，每一種類型選擇會因為目標用途差異有不同的考量。Cramfs 是一個壓縮的檔案系統，因此在系統啟動時，不需要一次性地將整個檔案系統中所有內容都解壓縮到記憶體（RAM）裡。當系統需要存取某些資

料的時候，會立即計算出該資料在 Cramfs 映像檔中的位置，再將該資料解壓縮到記憶體之中。由於本研究的 DMA-2440XP 平台上具有較大量的快閃記憶體，並且需要較多的記憶體供應用程式使用，所以最後選擇 Cramfs 作為嵌入式 Linux 平台的檔案系統。

本研究利用 BusyBox[64]工具包建立 Cramfs 檔案系統。BusyBox 計劃是在 1996 年由 Bruce Perens 發起的，本身為一個很成功的開放原始碼軟體計畫，其目的在為 Linux 作業系統建立可以開機且具有救援工具的單一磁片系統。直到之後由 Dave Cinege 接管計畫，Dave Cinege 對 BusyBox 做了許多修改及增加特性，並轉移成以發展針對嵌入式系統為目標。BusyBox 整合了一百多個常用的 Linux 命令和工具軟體，甚至還包含 HTTP 伺服器和 Telnet 伺服器。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。透過動態連結 BusyBox 二進位檔，便能有效地減小程式的體積，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 版，同樣是在終端機視窗鍵入 make menuconfig 進行檔案系統的配置（參考圖 3-4）；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 版；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（參考圖 3-5）。

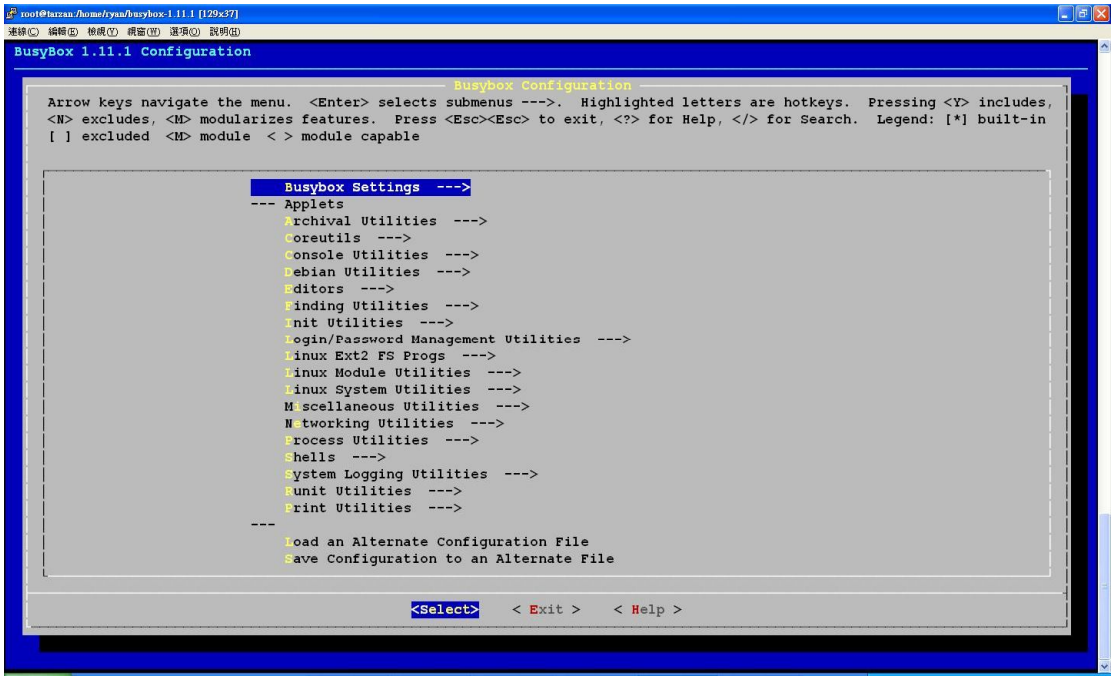

圖 3-4、BusyBox 配置畫面

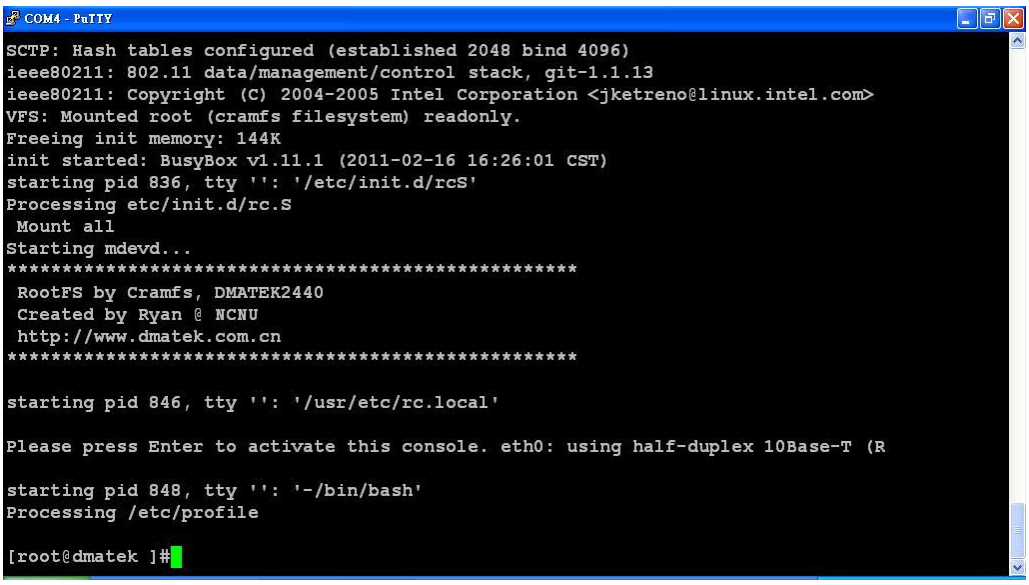

圖 3-5、嵌入式 Linux 系統啟動畫面

3.2 Atmel RZRAVEN 開發板移植

根據本研究假設的實驗環境中，設備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 平台及個人電腦上，僅有乙太網路介面，並未內建支援 IEEE 802.15.4 的無線傳輸介面。因此本研究另外採用 Atmel RZRAVEN 2.4GHz 開發板套件作[65]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 模組是一個無線 USB 傳輸器（USB dongle），如圖 3-6 所示。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。

圖 3-6、RZUSBSTICK 模組

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 為一個開放原始碼作業系統[66]，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 資源的配置甚至只需要 2KB 記憶體和 40KB 唯讀記憶體（Read-Only Memory；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 開發版上實現 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。由於本研究選擇 RZUSBSTICK 模組當作擴充介面，因此針對 AT90USB1287 微控制器的軟體進行編譯。在編譯完成之後，會產生 ravenusbstick.elf 檔案，接著透過 JTAGICE mkII 模擬器將二進位檔燒寫到 AT90USB1287 微控制器上，至此把 RZUSBSTICK 與個人電腦連接，則個人電腦即可成為 6LoWPAN 傳輸網路中的一個節點。如圖 3-7 所示，個人電腦將 RZUSBSTICK 視為一個橋接介面（Bridge），並且使用個人電腦上的原始乙太網路介面作為該介面預設路由器。在 Contiki 計畫當中，RZUSBSTICK 與 Contiki 的結合，表示硬體與韌體的結合，通稱為“Jackdaw”。

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 11 月，Jackdaw 已經可以選擇另外一種 USB CDC-ECM（USB Communications Device Class-Ethernet Control Model）驅動程式來模擬網路介面。可惜的是，該驅動程式只適用於 Linux 作業系統上，對 Windows 作業系統並不支援。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。

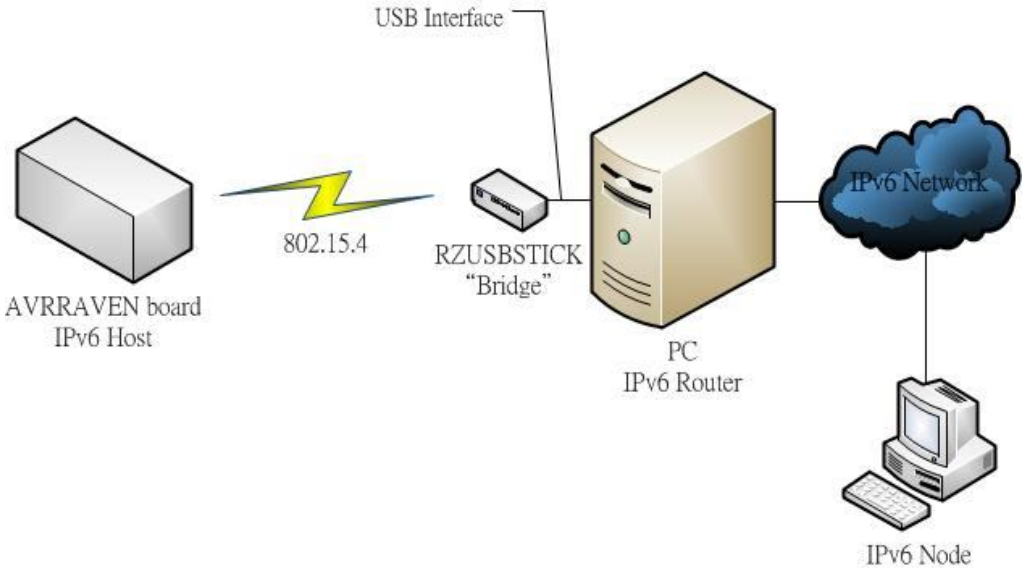

圖 3-7、Contiki 網路架構

第四章 效能分析

4.1 實驗環境與方法

Md. Nurul Islam 等學者[67]提出兩種實驗腳本來測量封包吞吐量（Throughput），分別為傳送不同大小的資料區塊在路徑處於穩定狀態與含有封包遺失狀態下。不同大小的資料區塊會影響傳輸封包負載量，進而影響 SCTP 關聯的吞吐量，最後證明在大資料區塊的情況下，不管是穩定狀態或含有封包遺失狀態，都會有較佳的吞吐量。Thomas Ravier 等學者[68]透過點陣圖（Plots）與統計的方式來研究多重路徑的效能，透過系統實作，可以顯示 SCTP 傳輸效能及行為，如路徑傳輸具有緩慢啟動（Slow-start）的機制等。另外，也可以觀察到路徑轉換機制和在封包遺失的狀況下處理封包之能力。Jinyang Shi 等學者[69]提出在無線網路環境下，使用 SCTP 的多重路徑機制，以 GPRS 當作備用路徑重傳封包，並且運用在電子商務上。透過設備的實作，可以發現 SCTP 機制可以提供較佳的吞吐量，且增加無線網路的強健性。

本實驗採用 NetEM[70]軟體模擬網路環境。NetEM 提供傳送封包遺失、延遲或重複封包等狀態設置，其設置方式是使用 tc 流量控制工具來完成。透過 NetEM 可以針對不同的傳輸路徑給予不同的傳輸狀態。因此，在本實驗環境下，我們選擇 Ethernet 路徑進行環境配置，可以對傳送出去的流量做控制。6LoWPAN 路徑則維持原始狀態。研究假設實驗環境有兩種（參考圖 4-1）：

- ✧ SCTP 單一乙太網路路徑傳送
- ✧ SCTP 多重路徑傳送（乙太網路與 6LoWPAN）

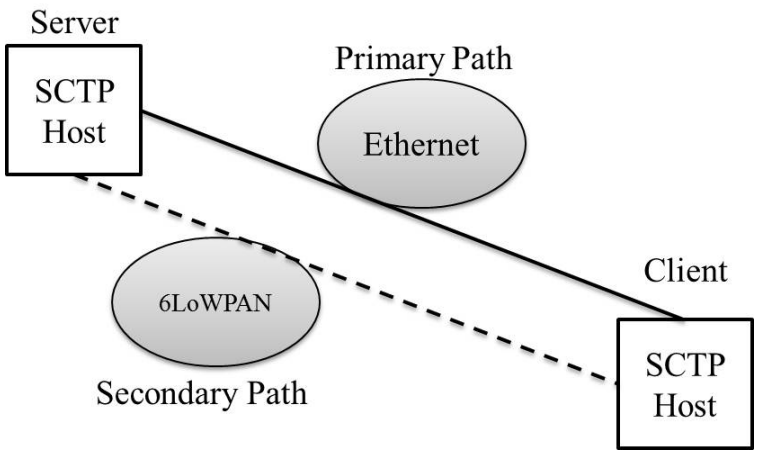

圖 4-1、實驗環境

原始 SCTP 傳輸協定具有的一大特色即是多重路徑下錯誤偵測與回復，錯誤偵測可以分為無法到達遠端位址和無法到達遠端節點之兩種狀況[71]。每一個資料區塊或 Heartbeat 區塊等需要等待回覆的區塊傳輸都可以用來評估遠端位址是否可以到達，當傳送端送出區塊後，在預期的時間內收到回覆即表示遠端位址可以到達；相反的，如未在預期的時間內收到回覆，則表示遠端位址可能暫時無法到達，假設連續偵測到遠端位址無法到達，便會正式判斷該位址失效。而在先前章節已經有提及 Heartbeat 是用來偵測處於閒置狀態的路徑，其閒置狀態判斷依據為在 Heartbeat 區塊傳輸間隔時間內，無其他可以更新 RTT 參數的區塊傳輸，包含資料區塊、INIT 區塊或

COOKIE-ECHO 區塊等，則判斷遠端位址處於閒置狀態。另外，任一遠端位址皆會賦予活躍（Active）或失效（Inactive）兩種狀態設置。以下將簡述偵測無法到達遠端位址的演算法。傳送端針對每個遠端節點都記錄了狀態（State）和錯誤計數器（Error counter）等參數，初始建立連線關係時，位址狀態皆設定為活躍，錯誤計數器設定為 0。當沒有收到資料區塊或 Heartbeat 區塊回覆的狀況發生時，錯誤計數器會開始累加，連續發生則持續累加，待超過 SCTP 協定參數 Path.Max.Retrans 所設定的門檻值（預設為 5）時，遠端位址即會被視為無法到達，狀態改標示為失效；相對於如果有收到區塊回覆，遠端位址將會被視為可以到達，狀態則標示為活躍，錯誤計數器也會被歸零。然而，此演算法有可能會造成誤判，舉例來說，當路徑發生嚴重的碰撞時，會丟棄大量的封包，進而讓傳送端誤認為無法到達該遠端位址。為了降低誤判的發生機率，則可以選擇提高 Path.Max.Retrans 的門檻值，但缺點是延長偵測確認位址失效的時間。不論是資料區塊或 Heartbeat 區塊等，都是根據重傳時間（Retransmission timeout；RTO）參數值來作預期收到回覆的時間，此數值對 SCTP 傳輸效率和穩定性的影響相當大。SCTP 協定參數中關於 RTO 的有 Rto.Initial=3000(ms)、Rto.Max=60000(ms)和 Rto.Min=1000(ms)等，其預設值都是有經過考量設定的，而 RTO 參數值則是以上述提及的資料區塊或 Heartbeat 區塊傳輸取得的 RTT 資訊來計算。對於重傳方面來說，RTO 參數值會變成上一次 RTO 參數值的兩倍，因此當 Path.Max.Retrans 的門檻值提高，所需要花費偵測無法到達位址的時間就會相對加長。除此之外，此演算法還具有一個不明確的狀況，就是當一個區塊送出去之後，並未在 RTO 規定時間內收到回覆，因此傳送端會選擇另一個位址再重送此一區塊，接下來在收到回覆後，傳送端將無法確保是哪一次傳送的區塊所送的回覆，原因在於接收端回覆也可能也經過重送機制的轉換。該狀況會攸關於各位址錯誤計數器是否要累加或是歸零，影響甚大。為了避免不明確狀況發生，演算法另外規定當經過重傳過後的區塊回覆，並不作位址門檻值歸零的動作，只在需要時作門檻值累加。

另外，Heartbeat 區塊傳輸間隔時間也與 RTO 參數值有相關，在 SCTP 協定中含有 Hb.Interval 參數值，用來指定傳輸間隔時間，預設值為 30 秒。而實際上 Heartbeat 區塊傳輸間隔時間為 Hb.Interval 乘上(1+δ)倍，δ是介於-0.5~+0.5 的隨機變數，然後再加上 RTO 參數值，如此才是真正 Heartbeat 區塊傳輸間隔時間。以上所述是偵測無法到達遠端位址的條件；當所有遠端位址都無法到達時，則判定無法到達遠端節點。傳送端針對每一個遠端關係連線都會維護一個全錯誤計數器（Total error counter），它可以用來記錄關係上所有錯誤計數，包含資料區塊及 Heartbeat 區塊等未收到回覆的狀況，當全錯誤計數器的數值超過 SCTP 協定參數 Association.Max.Retrans 所設定的參數值時，則會判斷為無法到達該遠端節點，最後關閉連線關係。Association.Max.Retrans 預設參數值為 10。全錯誤計數器的累加是根據任一遠端位址錯誤計數器加 1，即跟著加 1，然後如有收到任一遠端位址的回覆，則該位址錯誤計數器歸零，全錯誤計數器也會跟著歸零。然而，關於偵測無法到達遠端節點的演算法會遇到 Dormant State 問題，此問題為當所有遠端位址都被判斷成無法到達時，還未達到 Association.Max.Retrans 所設定的門檻值，因此造成關聯連線無法關閉，也無法繼續傳送資料的情形。為了避免這樣的狀況，所以 SCTP 要求 Association.Max.Retrans 參數值不能大於所有遠端位址 Path.Max.Retrans 參數值的總和，這樣可以儘量避免該狀況發生，但是仍無法全部避免。

根據上述演算法，在不同的環境下，參數設置必須在網路信賴度與遠端位址偵測速度之間作權衡，假設我們希望提升偵測速度，則勢必要調降 Hb.Interval、Path.Max.Retrans、Association.Max.Retrans 和 Rto.Max 等參數值，如此即可以達到實驗環境的要求。

4.2 Socket 程式撰寫

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）[72][73]。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一個 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。`sctp_connect()` 語法如下：

```
int sctp_connectx(int sd, struct sockaddr * addrs, int addrcnt)
```

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。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。

4.3 數據量測與分析

為了測量多重路徑下路徑轉換效果，本實驗選擇在預期的時間裡讓網路斷線。為達到這個效果，透過在用戶端與伺服器端執行 NetEM 網路環境模擬軟體。由於 NetEM 只針對發送出去的封包作控制，因此在設定上兩端點都必須同一時間將各自傳送出去的封包遺失率定為 100%，藉此模擬斷線的現象。其設置及移除指令分別如下：

- $ tc qdisc add dev eth0 root netem loss 100%
- $ tc qdisc del dev eth0 root

接著將指令寫入一個 Shell Script 檔案裡，透過此方式執行可以減少手動控制時間的誤差。根據圖 4-2 所示，在關聯開始建立 20 秒之後，以 NetEM 模擬兩端點 100% 遺失封包，製造斷線的現象；然後經過 130 秒再讓連線回復正常，則可以觀察路徑轉換情形。每一次實驗皆會傳送 200 個資料區塊，區塊大小為 998 位元組，然後區塊與區塊間隔傳送時間為 1 秒。

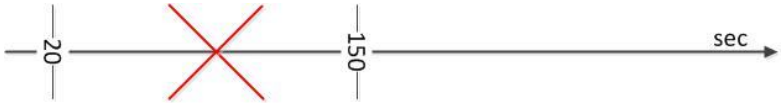

圖 4-2、傳輸時間軸

另外，為了提升遠端位址偵測的速度，本實驗將 SCTP 協定參數作以下調整：

- $ sysctl –w net.sctp.hb_interval=5000(ms)
- $ sysctl –w net.sctp.rto_max=30000(ms)
- $ sysctl –w net.sctp. path_max_retrans=4
- $ sysctl –w net.sctp. association_max_retrans=8

Wireshark[74]是一種網路封包分析軟體，其功能可以在兩端點擷取網路封包，並且盡可能顯示出最詳細的網路封包資訊。本實驗透過 Wireshark 執行在伺服器端，即可取得乙太網路及 6LoWPAN 網路的接收封包 TSN、SSN 和接收時間等資訊，以供數據分析使用。實驗一開始會在用戶端及伺服器端執行 Socket 程式，各自綁定乙太網路和 6LoWPAN 網路的 IPv6 位址介面，然後選定乙太網路作為主要路徑傳輸。乙太網路路徑以上述 NetEM 控制環境，6LoWPAN 網路則維持原始環境。最後實驗結果可以畫出如圖 4-3 表示的點陣圖。藍色點表示經由乙太網路介面傳送之封包；而紅色點則表示經由 6LoWPAN 網路傳輸之封包。一般來說，在單一路徑傳輸環境下，當路徑斷線時，會造成這一段時間內無法順利傳送封包，嚴重時甚至會造成整個連線關係中斷。然而，若啟動 SCTP 的備用路徑機制，由下圖可以觀察到在主要乙太網路路徑斷線之後，封包可以藉由備用 6LoWPAN 網路路徑，以無線網路方式來傳送封包；當主要路徑回復正常後，再切換回主要路徑傳送。透過備用路徑機制，不僅可以重送遺失封包，也可以作為備援使用，等於加大整體傳送頻寬，而且兩條路徑之間並不互相影響，更增加了傳送的可靠性。

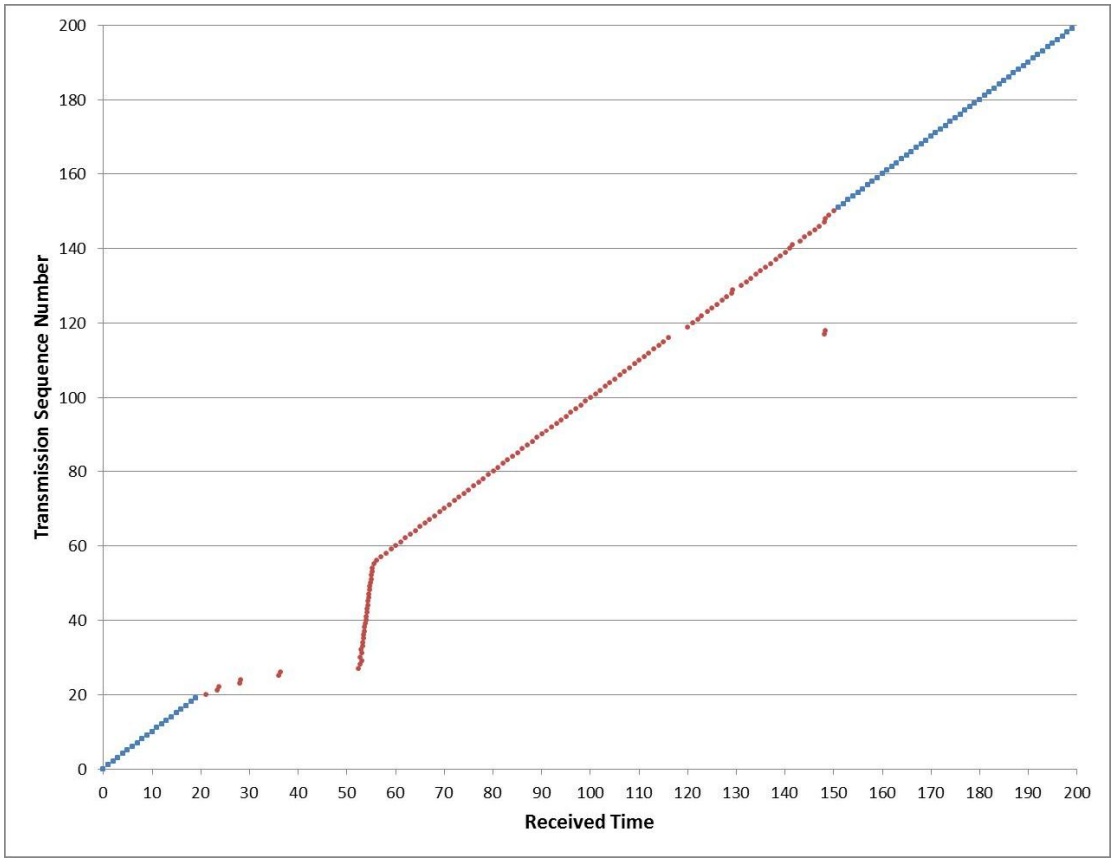

圖 4-3、多重路徑傳輸點陣圖

第五章 結論及未來展望

本研究已經可以把實驗環境完整的實作出來。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，也應該考慮立即作路徑的切換；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] Hubert Zimmermann, “OSI Reference Model-The IS0 Model of Architecture for Open Systems Interconnection,” IEEE Transactions on Communications, vol. COM-28, no. 4, April 1980.

[2] 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, vol. 30, no. 7, pp. 1655-1695, May 26, 2007.

[3] N. Kushalnagar, G. Montenegro, C. Schumacher, “IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals,” RFC4919, August 2007.

[4] G. Montenegro, N. Kushalnagar, J. Hui, D. Culler, “Transmission of IPv6 Packets over IEEE 802.15.4 Networks,” RFC4944, September 2007.

[5] E. Santacana, G. Rackliffe, Le Tang, Xiaoming Feng, “Getting Smart,” IEEE Power and Energy Magazine, vol. 8, no. 2, pp. 41-48, March/April 2010.

[6] A. Vojdani, “Smart Integration,” IEEE Power and Energy Magazine, vol. 6, no. 6, pp. 71-79, November/December 2008.

[7] Francisco J. Nogales, Javier Contreras, Antonio J. Conejo, Rosario Espínola, “Forecasting next-day electricity prices by time series models,” IEEE Transactions on Power Systems, vol. 17, no. 2, May 2002.

[8] 數位時代，“智慧電網 用電快、省、準”，187 期，December 2009.

[9] Isabel Praça, Carlos Ramos, Zita Vale, Manuel Cordeiro, “Mascem: a multiagent system that simulates competitive electricity markets,” IEEE Intelligent Systems, vol.

18, no. 6, pp. 54-60, November/December 2003.

[10] R. Stewart, Ed., “Stream Control Transmission Protocol,” RFC4960, September 2007.

[11] J. Postel, “Transmission Control Protocol,” RFC793, September 1981.

[12] J. Postel, “Internet Protocol,” RFC791, September 1981.

[13] J. Postel, “Internet Control Message Protocol,” RFC792, September 1981.

[14] R. Droms, “Dynamic Host Configuration Protocol,” RFC2131, March 1997.

[15] J. Postel, “User Datagram Protocol,” RFC768, August 28, 1980.

[16] Jonathan Lemon, “Resisting SYN flood DoS attacks with a SYN cache,” Proceedings of the BSDCon '02 Conference on File and Storage Technologies, February 11-14, 2002.

[17] S. Deering, R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification,” RFC2460, December 1998.

[18] Number Resource Organization (NRO) - Free Pool of IPv4 Address Space Depleted. [http://www.nro.net/news/ipv4-free-pool-depleted]

[19] R. Hinden, S. Deering, “Internet Protocol Version 6 (IPv6) Addressing Architecture,” RFC3513, April 2003.

[20] Ian F. Akyildiz, Jiang Xie, Shantidev Mohanty, “A survey of mobility management in next-generation all-IP-based wireless systems,” IEEE Wireless Communications, vol. 11, no. 4, pp. 16-28, August 2004.

[21] Joseph Davies, “Understanding IPv6, Second Edition,” Microsoft Press (ISBN 978-0735624467), January 19, 2008.

[22] R. Droms, Ed. J. Bound, B. Volz, T. Lemon, C. Perkins, M. Carney, “Dynamic Host Configuration Protocol for IPv6 (DHCPv6) ,” RFC3315, July 2003.

[23] N. Moore, “Optimistic Duplicate Address Detection (DAD) for IPv6,” RFC4429, April 2006.

[24] Girish Chiruvolu, Anshul Agrawal, Marc Vandenhoute, “Mobility and QoS support for

IPv6-based real-time wireless Internet traffic," ICC '99 - 1999 IEEE International Conference on Communications, vol. 1, pp. 334-338, June 6-10, 1999.

[25] Armando L. Caro Jr., Janardhan R. Iyengar, Paul D. Amer, Sourabh Ladha, Gerard J. Heinz II, Keyur C. Shah, "SCTP: A Proposed Standard for Robust Internet Data Transport," IEEE Computer, vol. 36, no. 11, pp. 56-63, November 10, 2003.

[26] R. Stewart, C. Metz, "SCTP: New Transport Protocol for TCP/IP," IEEE Internet Computing, vol. 5, no. 6, pp. 64-69, November/December 2001.

[27] Z.W. Park, J.H. Lee, M.K. Kim, "Design of an extended TCP for preventing DOS attacks," Proceedings KORUS 2003. 7th Korea-Russia International Symposium on Science and Technology, vol. 2, pp. 385-389, July 6, 2003.

[28] S. Fu, M. Atiquzzaman, "SCTP: State of the Art in Research, Products, and Technical Challenges," IEEE Communications Magazine, vol. 42, no. 4, pp. 64-76, April 2004.

[29] J. Funasaka, K. Ishida, H. Obata, Y. Jutori, "A study on primary path switching strategy of SCTP," Proceedings of Autonomous Decentralized Systems, ISADS 2005, pp. 536-541, April 4-8, 2005.

[30] S. Kashihara, T. Nishiyama, K. Iida, H. Koga, Y. Kadobayashi, S. Yamaguchi, "Path selection using active measurement in multi-homed wireless networks," Proceedings of the 2004 International Symposium on Applications and the Internet, pp.273-276, 2004.

[31] Y. Fuanhua, T. Saadawi, L. Myung, "IPCC-SCTP: An Enhancement to the Standard SCTP to Support Multi-homing Efficiently," Proceedings of the 2004 IEEE International Performance, Computing, and Communications Conference, pp. 523-530, 2004.

[32] 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.

[33] Sang Tae Kim, Seok Joo Koh, Yong Jin Kim, “Performance of SCTP for IPTV Application,” The 9th International Conference on Advanced Communication Technology, vol. 3, pp. 2176-2180, February 12-14, 2007.

[34] M. Atiquzzaman, W. Ivancic, “Evaluation of SCTP multistreaming over wireless/satellite links,” Proceedings 12th International Conference on Computer Communications and Networks, pp. 591-594, October 20-22, 2003.

[35] A. Medina, M. Allman, S. Floyd, “Measuring the evolution of transport protocols in the Internet,” ACM SIGCOMM Computer Communication Review, vol. 35, no. 2, April 2005.

[36] IEEE 802 Working Group, “Standard for Part 15.4:Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs),” ANSI/IEEE 802.15.4, October 2003.

[37] E. Callaway, P. Gorday, L. Hester, J.A. Gutierrez, M. Naeve, B. Heile, V. Bahl, “Home networking with IEEE 802.15.4: a developing standard for low-rate wireless personal area networks,” IEEE Communications Magazine, vol. 40, no. 8, pp. 70-77, August 2002.

[38] Jianliang Zheng, Myung J. Lee, “Will IEEE 802.15.4 make ubiquitous networking a reality?: a discussion on a potential low power, low bit rate standard,” IEEE Communications Magazine, vol. 42, no. 6, pp.140-146, June 2004.

[39] Eugene Shih, Seong-Hwan Cho, Nathan Ickes, Rex Min, Amit Sinha, Alice Wang, Anantha Chandrakasan, “Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks,” Proceedings of the 7th annual international conference on Mobile computing and networking, pp. 272-286, 2001.

[40] Jin-Shyan Lee, “An experiment on performance study of IEEE 802.15.4 wireless networks,” The 10th IEEE International Conference on Emerging Technologies and

Factory Automation (ETFA'2005), vol. 2, pp. 415-458, September 19-22, 2005.

[41] C. Evans-Pughe, “Bzzzz zzz [ZigBee wireless standard],” IEE Review, vol. 49, no. 3, pp. 28-31, March 2003.

[42] Jin-Shyan Lee, Yu-Wei Su, Chung-Chou Shen, “A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi,” The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON 2007), pp. 46-51, November 5-8, 2007.

[43] ZigBee Alliance. [http://www.zigbee.org/]

[44] “Announcing the Advanced Encryption Standard (AES),” Federal Information Processing Standards Publication 197, November 26, 2001.

[45] ZigBee Alliance, “ZigBee Specification,” ZigBee Document 053474r17, January 17, 2008.

[46] Gee Keng Ee, Chee Kyun Ng, Nor Kamariah Noordin, Borhanuddin Mohd. Ali, “A Review of 6LoWPAN Routing Protocols,” Proceedings of the Asia-Pacific Advanced Network 30th Meeting, August 9-13, 2010.

[47] Zach Shelby, Carsten Bormann, “6LoWPAN: The Wireless Embedded Internet,” Wiley (ISBN 978-0470747995), January 2010.

[48] Xin Ma, Wei Luo, “The Analysis of 6LowPAN Technology,” IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application (PACIIA 2008), vol. 1, pp. 963-966, December 19-20, 2008.

[49] Aminul Haque Chowdhury, Muhammad Ikram, Hyon-Soo Cha, Hassen Redwan, S.M. Saif Shams, Ki-Hyung Kim, Seung-Wha Yoo, “Route-over vs Mesh-under Routing in 6LoWPAN,” Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly, pp. 1208-1212, June 21-24, 2009.

[50] J.W. Hui, D.E. Culler, “Extending IP to Low-Power, Wireless Personal Area Networks,”

IEEE Internet Computing, vol. 12, no. 4, pp. 37-45, July/August 2008.

[51] C. Perkins, E. Belding-Royer, S. Das, “Ad hoc On-Demand Distance Vector (AODV) Routing,” RFC3561, July 2003.

[52] K. Kim, S. Daniel Park, G. Montenegro, S. Yoo, N. Kushalnagar, Ed., “6LoWPAN Ad Hoc On-Demand Distance Vector Routing (LOAD),” draft-daniel-6lowpan-load-adhoc-routing-03, June 19, 2007.

[53] K. Kim, G. Montenegro, S. Park, I. Chakeres, C. Perkins, Ed., “Dynamic MANET On-demand for 6LoWPAN (DYMO-low) Routing,” draft-montenegro-6lowpan-dymo-low-routing-03, June 19, 2007.

[54] K. Kim, S. Yoo, S. Daniel Park, J. Lee, Ed., “Hierarchical Routing over 6LoWPAN (HiLow),” draft-daniel-6lowpan-hilow-hierarchical-routing-01, June 17, 2007.

[55] Choon-Sung Nam, Hee-Jin Jeong, Dong-Ryeol Shin, “Extended Hierarchical Routing over 6LoWPAN,” The 4th International Conference on Networked Computing and Advanced Information Management (NCM 2008), vol. 1, pp. 403-405, September 2-4, 2008.

[56] 長高科技股份有限公司 (DMATEK). [http://www.dmatek.com.tw]

[57] MiniGUI, Beijing FMSoft Technologies Co., Ltd. [http://www.minigui.org/]

[58] Hallinan, Christopher, “Embedded Linux Primer: A Practical, Real-World Approach,” Prentice Hall (ISBN 0131679848), September 5, 2006.

[59] GCC, The GNU Compiler Collection. [http://gcc.gnu.org/]

[60] LKSCTP, The Linux Kernel Stream Control Transmission Protocol. [http://lksctp.sourceforge.net/]

[61] Chris DiBona, Sam Ockman, “Open Sources: Voices from the Open Source Revolution,” O'Reilly Media (ISBN 978-1565925823), January 1999.

[62] K. Sollins, “The TFTP Protocol (Revision 2),” RFC1350, July 1992.

[63] Cramfs, The Compressed ROM Filesystem. [http://sourceforge.net/projects/cramfs/]

[64] BusyBox: The Swiss Army Knife of Embedded Linux. [http://www.busybox.net/]

[65] Atmel RZRAVEN kit. [http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4291]

[66] Contiki: The Operating System for Connecting the Next Billion Devices - the Internet of Things. [http://www.sics.se/contiki/]

[67] Md. Nurul Islam, A. Kara, "Throughput Analysis of SCTP over a Multi-homed Association," Proceedings of the Sixth IEEE International Conference on Computer and Information Technology, pp. 110, September 2006.

[68] Thomas Ravier, Rob Brennan, Thomas Curran, "Experimental studies of SCTP multi-homing," First Joint IEI/IEE Symposium on Telecommunications Systems Research, Dublin, Ireland, November 27, 2001.

[69] Jinyang Shi, Yuehui Jin, Hui Huang, Dajiang Zhang, "Experimental Performance Studies of SCTP in Wireless Access Networks," International Conference on Communication Technology, vol. 1, pp. 392-395, April 9-11, 2003.

[70] Stephen Hemminger, "Network Emulation with NetEm," Open Source Development Lab, April 2005.

[71] Randall R. Stewart, Qiaobing Xie, "Stream Control Transmission Protocol (SCTP): A Reference Guide," Addison-Wesley Professional (ISBN 978-0201721867), November 2, 2001.

[72] A. Jungmaier, "SCTP for Beginners," Computer Networking Technology Group, 2001. [http://tdrwww.exp-math.uni-essen.de/inhalt/forschung/sctp_fb/SCTP4Beginners.html]

[73] Jan Newmarch, "Stream Control Transmission Protocol (SCTP) Associations," Linux Journal - HOWTOs, October 1, 2007. [http://www.linuxjournal.com/article/9749]

[74] Wireshark - the world's foremost network protocol analyzer. [http://www.wireshark.org/]