

教育部八十八年度通訊科技教育改進計畫

教材名稱：通訊協定實驗教材

編號：18

執行單位：國立中正大學電機系

計畫主持人：陳景章

職稱：教授

教材編撰人：侯廷昭

職稱：副教授

教材網址：<http://hccc.ee.ccu.edu.tw/tch/pub/>

連絡電話：05-2720411-6349 e-mail：tch@ee.ccu.edu.tw

一、目標

網際網路的盛行引起了研究人員再度探討 TCP/IP 這個通訊協定的興趣。為了要讓網際網路具有更好的服務品質，TCP/IP 這個通訊協定需要予以加強。通訊協定這門課程的設計理念是要讓學生不只是了解 TCP/IP 協定的工作原理，而且要能夠具有設計和修改協定的能力，如此學生才有辦法將新的 idea 放入協定程式中實驗。

二、課程規劃重點

本課程著重於探討網際網路 TCP/IP 協定組中的核心協定，如 IP, TCP, ICMP, ARP, UDP 等的工作原理及其撰寫技巧。實驗環境則以無線區域網路為主要平台（ATM 區域網路為輔助平台，了解 TCP/IP 協定在無線環境的行為及效能。

三、教材適用對象

本教材適用對象具有基本電腦網路知識且對開發通訊協定程式有興趣的研究生。這門課的先修課程是電腦網路，學生有了基本的 OSI layered model 觀念及 TCP/IP 的基本了解之後，才比較能了解程式的意義。

四、教材內容設計

為了要達成澈底了解協定程式且能加以修改的

目標，我們需要一本介紹協定程式碼的教材及一個完全公開的作業系統平台。在眾多有關 TCP/IP 協定的教科書中，唯有 D.E. Comer 的 Internetworking with TCP/IP, Vol. II, 詳細的介紹如何撰寫 ARP, IP, ICMP, TCP, UDP, SNMP, RIP 等協定。因此這本書被我們選作課堂講授的主要教材。這本書內的程式是以 XINU 這個作業系統為平台，雖然 XINU 是可公開取得，但考慮其架設困難度及使用並不普及的情況下，我們決定選用 FreeBSD 為我們的實驗平台。因此整個實驗的設計是環繞在 FreeBSD 這個作業系統環境。如此的作法，學生同時學到二種不同的 Implementation，可以互相比較。

課堂講授的 Topic 共有下列十五單元：

- (1a) The Structure of TCP/IP in an Operating System,
- (1b) Network Interface Layer
- (2) Address Discovery and Binding (ARP)
- (3) IP : Global Software Organization
- (4) IP : Routing Table and Routing Algorithm
- (5) IP : Fragmentation and Reassembly
- (6) IP : Error Processing (ICMP)
- (7) IP : Multicast Processing (IGMP)
- (8) UDP : User Datagrams
- (9) TCP : Data Structure and Input processing
- (10) TCP : Finite State Machine Implementation
- (11) TCP : Output Processing
- (12) TCP : Timer Management

- (13) TCP : Flow Control and Adaptive Retransmission
- (14) TCP : Urgent Data Processing and the Push Function
- (15) Socket-Level Interface

五、實驗內容設計

本課程設計了七個實驗。實驗一是 FreeBSD 的安裝，可讓學生熟悉系統的環境及相關設定。實驗二是無線區域網路卡的安裝，建立無線區域網路的環境。實驗三是 ATM 網路卡的安裝，建立 IP over ATM 的實驗環境。實驗四是 ARP 和 ICMP 的量測，讓學生觀察並分析 ARP 和 ICMP 的行為。實驗五是 FTP 的量測，讓學生觀察 client 和 server 之間建立 TCP 連線的 Three-way Handshaking 的程序及 FTP 的步驟。實驗六是要學生修改 TCP congestion window 的程式，並觀察其效應。實驗七是 Socket Programming 的訓練，讓學生撰寫簡單的 mail 或 talk 的網路程式。以下針對這七個實驗做更進一步的說明。

實驗一 安裝 FreeBSD 作業系統

本課程是要探討 TCP/IP 這個通訊協定的核心，目前大家最常用的作業系統應該都是 Windows 或 MacOS，為何我們不採用這兩種大家常用且較簡單的作業系統，反而採用 FreeBSD 這種使用者較少，且操作較困難的作業系統，這是因為 Windows 及 MacOS 並不提供程式碼給使用者，所以當我們要去瞭解系統內 TCP/IP 如何運作時，就有困難存在，但 FreeBSD 有提供完整的程式碼供使用者研究，再加上 TCP/IP 在 UNIX 系統上已經使用很久了，也相當成熟了，而其他的作業系都是這幾年才將 TCP/IP 納入其系統，穩定度也較差，所以我們才會選擇採用 FreeBSD 這套作業系

統。

安裝作業系統基本上是參照安裝手冊的說明進行。首先製作 FreeBSD 的開機片，開機，作硬體設定，最後安裝軟體。

實驗設備：PC 一部，網路卡或 FreeBSD 光碟。

實驗二 安裝 IEEE 802.11 wireless LAN card

為了不受有線環境的牽制，無線傳輸的網路環境越來越受到重視。但不同的作業系統對於無線網路的支援程度不一。本實驗目的把符合 IEEE802.11 的無線區域網路安裝在 FreeBSD UNIX 環境下，作為以後進一步實驗 TCP/IP 網路的基本平台。FreeBSD 為 PC 上多種 UNIX-like 作業系統之一。UNIX system 透過 kernel 來與 hardware 溝通，而 device driver 則是 kernel modules 用以控制周邊設備的運作，所以上層的應用程式無須做任何更動而可透過 UNIX system 操作新加入的各項周邊設備。故在 FreeBSD 中安裝新的無線網路卡後，將其 device driver 設定到編譯 kernel Makefile 中，並重新編 kernel。就可驅動此新加入的無線網路卡。

實驗設備：LANEscape WL2420/ISA 無線網路卡，Firmware 1.2 以上，FreeBSD 2.2.8，CD-ROM，PC 一部。

實驗三 安裝 ForeRunner ATM Adapter card

ATM (非同步傳輸模式；Asynchronous Transfer Mode)可處理多種不同型態的流量，從對傳輸時間敏感、不敏感以及資料正確性要求高、要求低不同的需求，都可在同一個 ATM 介面及網路上傳輸。本實驗的目的是將 ATM 網路卡架設在 FreeBSD 的作業系統下，以作為進一步 TCP/IP 網路實驗課程的基本實驗平台。

ATM 是以細胞 (cell) 為主的交換及多工

技術，主要是做為一個各式服務的多功能、預接式的傳輸模式。ATM 也可做為像 ATM Forum 中所指定的區域網路及專屬網路技術。ATM 屬於非同步方式，因為它不像同步傳輸模式 (STM; Synchronous Transfer Mode) 傳送每個數據所使用的時槽週期是相同的。

ATM 提供一個在網路架構上定義多工和交換法則標準的可能，以及 SONET/STM 提供非常高速的實際傳輸標準。ATM 也提供了多種服務品質(QoS) 的等級來符合不同應用系統的延遲需求及丟棄的效能。因此，ATM 的目標是將整個網路可經由 ATM 及 ATM 適應層的交換及多工方法來支援全部的服務，例如：語音、封包數據 (SMDS、IP、訊框傳送)、視訊、影像、電路模擬等。ATM 可經由交換式虛擬電路，提供變動頻寬(bandwidth-on-demand) 以及支援接近區域網路存取的頻寬。

ATM 的定義是一個介面及協定，作為在一個共同的傳輸媒體上，交換變動位元速率及固定位元速率的流量。整個 B-ISDN 的協定架構通常就是指 ATM。ATM 常被視為一項技術，由符合 ATM 協定標準組成的硬體及軟體所構成，提供一個具多工、交互連接及交換功能的網路。ATM 技術範圍從網路介面卡、多工器、相互連接，甚至是一個完整的交換器。當 ATM 技術具有成為網路基礎架構核心的優點時，ATM 硬體及相關的軟體可一同提供主幹技術作為高等通訊網路。事實上，有許多的專家，視 ATM 為主的架構將成為未來提供數據及語音的平台。ATM 也提供一個具擴充能力的基礎架構，從校園環境到中央處理中心。擴充能力可依介面速度，交換器容量、網路大小及定址方式而有所不同。ATM 的主要處理單位是細胞 (Cell)。ATM 標準之中定義長度為 53 個位元組的固定大小細胞，其中有 5 個位元組是頭欄部分，48 個位元組是負載。

實驗設備：個人電腦 PC 兩台，ATM 網路卡 (ForeRunner PCA-200E)兩片，集線器 (Hub) 一台，雙絞線兩條，fiber 一對。

實驗四 ARP 和 ICMP 的量測與分析

本實驗藉由實際操作 NetXRay 或 Surveyor 等網路監控軟體，觀察主機以 telnet 連線 (1) 本地電腦 (local) (2)遠端電腦 (remote) 以及(3) 不存在電腦 (unreachable) 時，在網路流傳之 ARP 和 ICMP 的封包出現情形，並加以分析。

ARP (Address Resolution Protocol)：在區域網路中，當一個乙太網路訊框 (frame) 由一部機器傳送到另一部機器時，是由一組 48 個位元的乙太網路位址 (MAC Address) 來決定此訊框需送達那個介面，而非由軟體驅動程式檢查資料報裡的目的地 IP 位址。ARP 協定負責依據目的地的 IP 位址找出其 MAC 位址。此行為能自動執行，並與應用程式或系統管理員扯不上關係。

ICMP(Internet Control Message Protocol): 通常被視為 IP 層的一部份。它傳遞錯誤的訊息和其他需要注意的情況。ICMP 的訊息常是對 IP 層或更高層的協定才有效 (TCP 或 UDP)。一些造成錯誤的 ICMP 的訊息會傳回給使用者處理程序來處理。

實驗設備：個人電腦 PC 兩台，其作業系統為 FreeBSD，網路卡使用無線網路卡，測試端亦使用相同設備，測試軟體為 NetXRay 3.0.3。另每次實驗前需執行 `arp -a -d` 以清除 ARP cache。

實驗五 FTP 連線觀測與分析

本實驗藉由實際操作 NetXRay 或 Surveyor 等網路監控軟體，觀察 FTP 連線時，

在網路 TCP 層的封包出現情形，並加以分析各封包之出現原因。

FTP 的連線可以分成兩大類，其一為 Control Connection，其二為 Data Connection。Control Connection 用來傳輸控制資訊之用。例如建立連線、釋放連線、或其他控制命令及其回覆，都是經由 Control Connection。Data Connection 才是真的用來傳輸資料的部分。

建立連線:

1. 一開始,Client 端會向 Server 發出要求建立連線的 request。
2. FTP 的內定目的 port 為 21。故 Client 端自行取一個 port (如 1031)，再跟 Server 端的 port 21 要求建立連線。(剛開始之 Sequence Number 亦自取, Acknowledge Number 則設為 0)。
3. 要求建立連線之封包，內容的 SYN 旗標被設為 TRUE。表示這是要求建立連線之封包。
4. 當 Server 端收到建立連線之要求，會回覆一個 SYN+ACK 的封包，表示同意建立連線。
5. 當 Client 端收到 Server 的回覆之後，也再度回覆一個 ACK 封包，以確認連線。至此完成 Three way hand-shaking 的連線程序。

建立連線所使用的 Channel 是 Control Channel，值得一提的是，Control Channel 乃是使用跟 telnet protocol 相同的機制。換句話說，我們也可以利用 telnet 的 client 程式，經由一定的程序，進行 ftp 的連結。

命令溝通:

正如前面所述，FTP 的 Control Connection 所使用的協定跟 telnet 相同的。所有的命令也是透過相同的機制來傳送。一般而言，當 server 接收到一個 client 端所發出的命令，在處理完畢之後，會再送出一個對應的回覆，通知 client

端。

FTP 所定義的命令如下:

1. Access Control Commands

User Name (User)、Password (Pass)、Account (Acct)、Change Working Directory (Cwd)、Change To Parent Directory (Cdup)、Structure Mount (Smnt)、Reinitialize (Rein)、Logout (Quit)

2. Transfer Parameter Commands

Data Port (Port)、Passive (Pasv)、Representation Type (Type)、File Structure (Stru)、Transfer Mode (Mode)

3. Ftp Service Commands

Retrieve (Retr)、Store (Stor)、Store Unique (Stou)、Append (With Create) (Appe)、Allocate (Allo)、Restart (Rest)、Rename From (Rnfr)、Rename To (Rnto)、Abort (Abor)、Delete (Dele)、Remove Directory (Rmd)、Make Directory (Mkd)、Print Working Directory (Pwd)、List (List)、Name List (Nlst)、Site Parameters (Site)、System (Syst)、Status (Stat)、Help (Help)、Noop (Noop)。

資料傳輸:

當 ftp 過程中，有大筆資料需要傳輸的話（例如傳送目錄資訊，傳送大量訊息，或傳送資料本身），都需要重新建立連線。此處所謂之建立連線，乃是建立 data connection。data connection 建立之後，即利用 sequence number，與對應之 acknowledge number 做查核封包，以確認接收方完整接收所需之資料，並將下一筆未傳輸完成的資料繼續傳送。在我們於 ftp client 端，做 get 或 put 的指令，或是利用 Cute-FTP 做拖曳檔案的動作，都會先建立 data connection。而一般 data connection 的 port 都會設定在 port 20。

結束連線:

當我們資料傳輸完成，要結束 FTP 連線的時候，client 端會送出 FIN 之封包，要求中

斷連結。當 server 端收到這個要求的時候，會回覆一個 ack 及另外一個 FIN 封包。當 client 端收到此兩個封包之後，再度回覆一個 ACK 封包。連線即宣告結束。

實驗設備：PC 兩部 Surveyor 或 NetXRay 軟體，網路卡。

實驗六 TCP congestion control 的觀察與分析

藉由修改 FreeBSD 之網路核心程式，加入記錄封包 sequence number 及 Sending Window、Congestion Window、Round Trip Time 的資訊，並加以觀察及分析。

由於 TCP 層的網路傳輸控制位於 FreeBSD 的核心 (kernel) 之中，我們若欲記錄該傳輸控制，便必須修改 FreeBSD 的原始程式碼，加入可茲記錄的 code。當任何網路所提供的負載超過所能處理的範圍時，會產生壅塞的狀況。而雖然網路層也處理壅塞，但大部分還是由 TCP 處理，這是因為解決壅塞的方法，實際上是放慢資料的傳輸率。當連線建立，他們會選擇一固定長度的 Window。接收端可以依據自己緩衝區的大小來決定其視窗大小。如果傳送端同意此大小，則因接收端緩衝區溢滿的問題將不再發生，但網路上的壅塞仍有可能。這兩個問題可以區別為網路容量及接收端容量分別處理。首先，傳送端需有兩個視窗，一個是接收端所同意的視窗，一個則是壅塞視窗 (Congestion window)。每個視窗皆對傳送端可能的位元組反應。可傳送的位元數，為兩個視窗中的最小者。當連結建立，傳送端會將壅塞視窗長度啟使化，然後送出最大的區段 (MSS)，若能及時獲得回復 (ack)，則可再將壅塞視窗變為兩倍。如此一來，壅塞視窗將成等比級數成長，直到發生逾時或是達到接收端視窗的長度。這種演算法可稱為慢速啟動 (slow start)。

再來是門檻 (threshold) 的概念。啟使為 64K，當逾時發生，門檻設定為目前壅塞視窗的一半，而壅塞視窗則重設為一個最大區段 (MSS)。之後再度使用慢速啟動，當慢速啟動再次遇到門檻時，指數成長必須停止，每次成功的傳輸會讓壅塞視窗以線性方式成長 (每次傳輸成功增加一個 MSS)。

實驗設備：PC 一部，無線網路卡，Access Port，Excel。

實驗七 Socket programming (a mail program or a talk program)

本實驗以實際的應用程式寫作，訓練使用 network socket system call，了解並且實作 mail program 或 talk program。

Socket system call 大量簡化了撰寫網路程式的複雜度，它使我們能像讀寫檔案般地，讀寫網路另一端的資料；其餘的資料處理，則與一般應用程式無異。

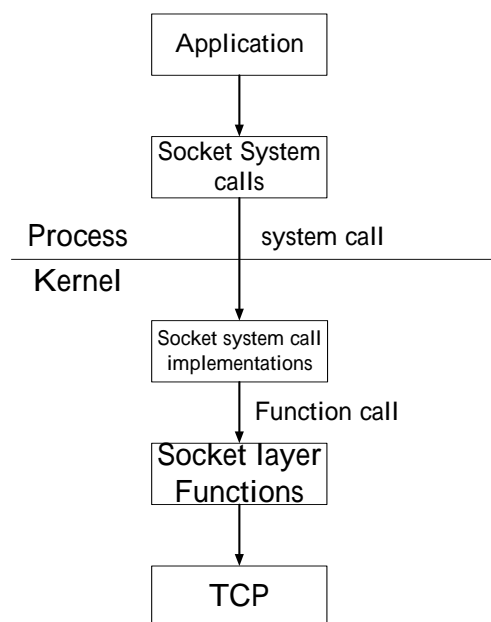


Figure 1. The socket layer converts generic requests to TCP protocol operation.

Socket 概念：當我們需要透過網路讀寫另一端的資料時，應用程式請求作業系統建立一個 socket，透過它，我們就如同使用一般的檔案或裝置一樣，以傳統的讀寫操作來控制它。

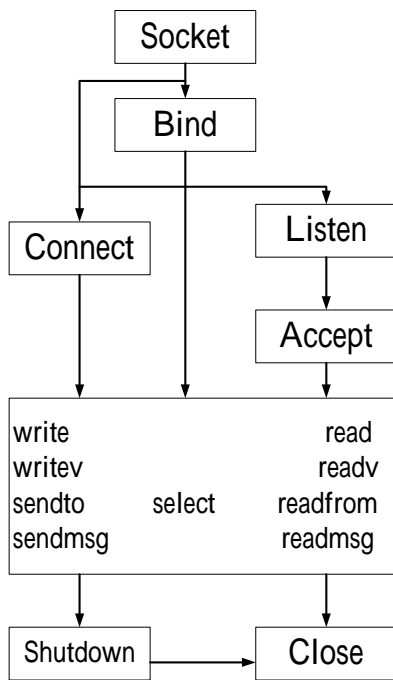


Figure 2. Network system call flowchart

1. 李建達, “FreeBSD 入門與應用,” 博碩文化, 1999。
2. D. E. Comer, “Internetworking with TCP/IP, Vol. I: Principles, Protocols, and Architecture,” 3rd Ed., Prentice-Hall, 1995.
3. D. E. Comer and D. L. Stevens, “Internetworking with TCP/IP, Vol. II: Design, Implementation, and Internals,” 3rd Ed., Prentice Hall 1999.
4. D. E. and D. L. Stevens, “Internetworking with TCP/IP, Vol. III: Client-Server Programming and Applications,” 2nd Ed., Prentice Hall, 1996.

實驗設備： PC 兩部以上，網路卡各一。

六、執行成效

經過實驗一至三的訓練，學生對於灌 OS 或網路卡驅動程式多能駕輕就熟。實驗四、五的量測及數據分析使得學生對 ARP, ICMP, TCP, FTP 等協定的運作有更深刻的體會。實驗六修改 TCP Flow Control Algorithm，訓練學生第一手系統程式修改的能力，不只更透澈了解網路程式的運作，且對 OS Kernel 的架構有更進一步的體會。最後實驗七 Socket Program 的撰寫，讓學生感受一個網路應用程式會運作的喜悅。

部份學生在修畢這門課後，有機會參予 QoS Router 的計畫，實際著手建置 RSVP, ALTQ, diffserv, routing, signaling 等功能於一個 Embedded System with real-time operating system。這門課的訓練讓他們能很快的掌握撰寫網路協定程式的技巧。

七、參考文獻