報告題名：以 Linux 建構無線網路之家庭影音

作者：呂思霈
系級：通訊系四年級
學號：D9157856

開課老師：趙啓時 博士

課程名稱：專題研究

開課系所：通訊工程學系

開課學年：九十三 學年度 第 一 學期

逢 甲 大 學 通 訊 工 程 學 系

專題報告：以 Linux 建構無線網路之家庭影音

指導教授：趙啟時 教授 專題生：呂思霈

摘要

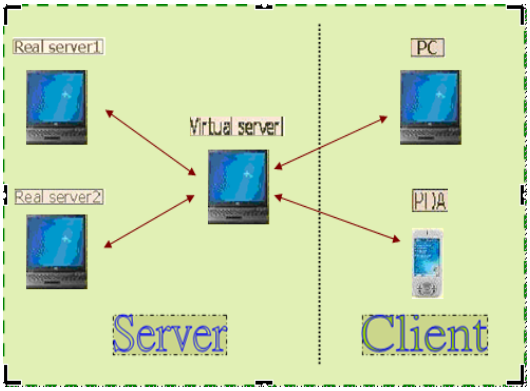
隨著科技的進步，家電資訊化便是一種趨勢，尤其是把媒體影像功能整合在一起，或許會是不錯的主意。例如把 MP3 Player、DVD/VCD、TV、FM 等，整合在一台多媒體中央伺服器上，再以無線網路的架構，傳送至家中每個角落的接收端，這樣不僅省去同一設備卻要購買多台的浪費，也可以減少家中電線亂竄，破壞美感。

本研究著重於利用開放原始碼(open source)，結合 Linux 伺服器、Ad-Hoc 無線網路、負載平衡、零配置等技術，並以家庭環境為背景，來實現此一操作平台。此平台提供消費者省去手動配置設定檔的步驟，並且透過 Ad-Hoc 無線網路的傳輸，使消費者可以輕易地使用其他裝置連上網路，讓在外的使用者，能夠在無線區域內接收數位內容。本研究的目的，在於整合各種數位內容，提供使用者簡易的操作，未來更可以擴大為廣域網路的服務，讓消費者不管身處何地，即能觀看影音內容，又能透過網路電話來交談或是遙控家電，如此，才能稱得上--無線數位內容服務平台。

目錄

第一章 架構

整個環境架構可以分為 server 端和 client 端，家中的電腦



（筆記型或是桌上型）或是 PDA，經由虛擬伺服器，導向真實伺服器，此時，真實伺服器可以架設多台，透過負載平衡技術，將客戶端的需求依據真實伺服器狀況，由虛擬伺服器分配到較適合的真實伺服器上，藉此達到分流的效果，也可以讓伺服器有最大的效能。

第二章 測試環境

因為是以無線網路當作傳輸的介面，因此，無線網路的環境就是必備的，接下來就是系統的核心部分，採用 Linux 做為作業系統，確保伺服器的穩定、效能，以及眾多的免費軟體，來替伺服器升級跟增加新功能，再來就是此專題所需的硬體設備，使用三台筆記型電腦當作伺服器，其中一台為虛擬伺服器，另兩台則為真實伺服器；一台電腦，和一台 PDA 當作客戶端，來做簡易的測試。

無線網卡採用的是 Lucent Technologies WavelLAN/IEEE Adapter:PC24E-H-FC(PCMCIA)，採用的晶片為 Orinoco，另外經測試可用的無線 USB 網卡有：

1. D-link DWL-120 USB，採用的晶片為 Atmel
2. Corega WLUSB-II stick-11 v2，採用的晶片為 Prism2/2.5/3

在挑選無線網路卡的時候，要特別注意所使用的 Linux 版本是否援，尤其 Linux 並不像微軟 Windows 對硬體支援性高，尤其對新硬體、新產品，往往 Linux 還來不及更新驅動程式，導致無法使用的情況，就拿無線網路來說，在經過六款無線網卡的測試，也只有三張網卡能正常使用，即使不同網卡使用相同晶片也可能出現不支援的情況，因此，在挑選的時候盡量找尋支援性較高的晶片，如：Prism2/2.5/3、Orinoco，**參考文獻 6** 就有提供各品牌的無線網卡所對應的晶片。或是選擇有標明支援 Linux 的無線網卡，都是不錯的選擇；但是，就算如此，還是有可能出現因為 Linux 的版本不同而產生有的能支援有的不能支援的情況，就像 D-link DWL-120 USB 和 Corega WLUSB-II stick-11 v2，在 Mandrake10 能夠直接使用，但在 Fedroa3 裡可能就要另外安裝驅動程式了，對 Linux 剛接觸的人是很棘手的問題，這涉及到核心的編譯，rpm、tarball 指令的運用，因此這算是有相當門檻的起步。

在作業系統方面則採用了 Kernel 為 2.6.10-1.770_FC3 的 Fedora Core3 版本。使用此版本的理由很簡單，就是它能夠直接支援 Lucent 的 PC24E-H-FC(PCMCIA)網卡，這樣可以省去不少麻煩。而不使用微軟的 Windows 是因為他有令人詬病的安全性問題，加上如果要使用伺服器版本

的話，那又是另外一筆花費，雖然幾乎所有的硬體它都能支援，但以 Linux 崛起的速度，相信不久的未來，各硬體廠商也會陸續加入 Linux 的驅動程式。

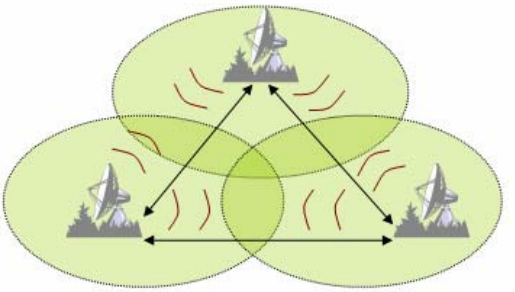
要察看自己 Linux 的核心很簡單，如下圖，只要在指令列上打上 #uname –r 就會顯現出來了。

再來是準備要來當伺服器的設備，只要是普通的筆記型電腦並且使用有 PCMCIA 擴充插槽即可，好來使用 PCMCIA 的無線網卡；準備三台筆記型電腦，三台皆用來當作伺服器端，另外再準備一台 PC 和 PDA，用來當作客戶端來測試，而 PC 和 PDA 只需有無線上網的功能，因為客戶端只是用來作連線的測試，並沒有設定的動作。

第三章 無線網路架設

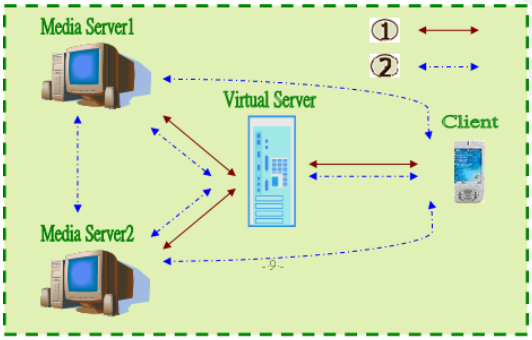
第一節 綱要

無線網路可以分為三種模式，基地台連接模式(Infrastructure or Station)、點對點連接模式(Ad-Hoc)、基地台模式（HostAP），有鑑於架設 HostAP 模式無線網路的困難度，根據 IBM 和 HP 的官方技術文件，並不是什麼樣的無線網路卡都可以支援 HostAP 架設，需要使用 Prism2/2.5/3 晶片的無線網卡，並且安裝驅動程式 Host AP driver（**參考資源 10**），經過一連串的設定，伺服器才具備 AP 模式；而其原理是，因為現在市面上出售的 802.11x 產品，不管是網路卡還是 AP，晶片廠商大都會使用相同的核心無線晶片組，只不過 HostAP 在基礎架構上還增加了一些嵌入式控制系統，控制晶片組的輸入和輸出，讓整個晶片組可以工作在普通的 Station 模式或 HostAP 模式，通過在相同的晶片組上提供不同的韌體，來實現不同的功能。但是，並不一定要使用 HostAP 才能達到傳輸的目的，我們知道 HostAP 是透過整合控制連線來達成，而 Ad-Hoc 模式則是透過互相連線來工作，那麼只要讓 Ad-Hoc 能夠分配連線不就可以達到該功能嘛！



如上圖：點對點連接(Peer-to-Peer：P2P)只要每個半徑範圍內的訊號有相互的覆蓋，即可互相通訊。而所謂的點對點模式（Ad-Hoc），顧名思義就是單一設備來進行直接點對點的連接，每個設備只要裝有 802.11x 的無線網卡，在有效的半徑範圍內，使用共通的網路設定，就能夠互相的直接溝通，通常這個有效半徑範圍在 50～00 公尺內，但是跟環境因素的影響有著相當大的關係。如：牆壁的阻隔、天候的好壞、是不是有別的使用相同通道的無線網路存在…等等，都是影響無線網路品質的重要因素。

在決定好要使用連接的模式之後，接下來採用 DHCP 技術來分配 IP，並且利用 DHCP 和 Ad-Hoc 的結合，可以模擬出 HostAP 的效果，有如採用 Ad-Hoc 傳輸模式的 HostAP，上圖為整個架構模式，每個客戶端都直接與主機連接，取得 IP 後便可以互相連接，這也是 DHCP 的最大好處，簡化客戶端的設定過程，甚至主機只要接入網路，每個客戶端都可以透過主機使用網路資源。由於主機只提供請求和配給 IP 的處理，真正的連線則是透過每個客戶端直接的連接，因此會比真正 HostAP 模式還來得有效率，下圖為實作伺服器端接入網路的工作過程：



步驟①：Media Servers 和 Client 分別向 Vitual Server 索取 IP。

步驟②：當 IP 分配好之後，這四台設備即可互相連接溝通。

完成以上的步驟，那麼無線網路的架構大致上已經規劃好了，接下來要看看如何實作。

第二節 實作前準備

實作的部分要用到的指令有：ifconfig、iwconfig、ifdown、ifup，要更動到的檔案有：/etc/sysconfig/network-scripts/ifcfg-X

ifconfig、iwconfig、ifdown、ifup 皆為網路指令，ifconfig 為設定網路參數使用的指令，ifup 和 ifdown 為啟動與關閉某個網路介面卡；而 iwconfig、iwlist、Iwpriv、Ifrename 為無線網路指令，iwconfig 為設定無線網路參數使用的指令；iwlist 為初始化掃描頻率、列表頻率、比特率和密鑰。Iwspy 為獲得每個節點的連接品質。Iwpriv 為允許針對特定於 Wi-Fi 驅動程式的無線擴展進行操作。Ifrename 為允許使用基於固定標準的名稱介面。假如沒有這些指令，是無法架設無線網路的，得先安裝 wireless tool 的套件，找尋套件請使用 **參考資源 2,3** 來找尋安裝。而 /etc/sysconfig/network-scripts/ifcfg-X 則為網路啟動的設定檔，X 代表著所使用的介面卡代號。這個設定檔有可能因為版本不同而有所不同的路徑。

要會更改設定檔必須先瞭解一些參數，底下為一些常更改的參數：

1. DEVICE=eth0 顧名思義，為網路卡的代號，也是上面所說的 X。
2. BOOTPROTO=static static 是採用固定 IP 的意思，假使要使用 DHCP 模式來取得 IP，則把 static 改成 dhcp 即可。
3. IPADDR=XXX.XXX.XXX.XXX IP 位置
4. NETMASK=XXX.XXX.XXX.XXX 子網路遮罩
5. NETWORK=XXX.XXX.XXX.XX

X　網段

6. BROADCAST=XXX.XXX.XXX.XXX　廣播位址
7. GATEWAY=XXX.XXX.XXX.XXX　預設通訊閘

有了這些參數之後，就可以開始來修改設定檔了。

第三節 實作

首先，要來編輯 Main Server（Virtual Server）主機上的設定檔 /etc/sysconfig/network-scripts/ifcfg-eth1。

```
[root@media ~]# cd /etc/sysconfig/network-scripts/
[root@media network-scripts]# vi ifcfg-eth1
```

```
IPV6INIT=no
ONBOOT=no
USERCTL=no
PEERDNS=yes
GATEWAY=10.10.1.1
TYPE=Wireless
DEVICE=eth1
HWADDR=00:02:2d:1b:5e:81
BOOTPROTO=static
NETMASK=255.255.255.0
DHCP_HOSTNAME=
IPADDR=10.10.1.1
NETWORK=10.10.1.0
BROADCAST=10.10.1.255
DOMAIN=
ESSID=base
CHANNEL=6
MODE=Ad-Hoc
RATE=Auto
```

因為網路規劃中是要這台 Main Server 有提供 DHCP 的功能，因此 IPADDR 設定為 10.10.1.1 為 Main Server 的 IP，也代表著 DHCP Server 的 IP，NETWORK 設定為 Main Server 的網段，也是 DHCP 可以分配的網段，另外 ESSID 為此無線網路的群組名稱，設定為 base，CHANNEL 為無線網路要使用的通道，設定為 6，傳輸模式 MODE 設定為 Ad-Hoc 模式，傳輸速率 RATE 設定為 AUTO，可以自動偵測所使用無線網卡的最高速率為多少，如此一來，簡易的網路功能就已經設定好了。另外，在安全的需求下，如果需要有 WEP Key 加密的功能，需要額外

建立一個 KEY 檔名稱為 key-ethX，X 為所使用的無線網卡代號，皆下來在設定檔中新增 KEY＝xxxxxxxxxx 十位數，如此一來，當別人要連上 base 這個群組的網路時，就需要輸入密碼了，這樣子也就不用怕頻寬被別人盜用了。

再來是要啟動這塊網路卡，以及檢查網路功能是否正常。在設定完網路卡之後，別忘了要先重新啟動，這樣才能設定檔的變更才能生效。

```
[root@media network-scripts]# ifdown eth1;ifup eth1
```

在重新啟動後，還要檢查網路卡是否正常啟動，有時候會因為設定錯誤，但系統也沒出現錯誤訊息，因而無法使用。

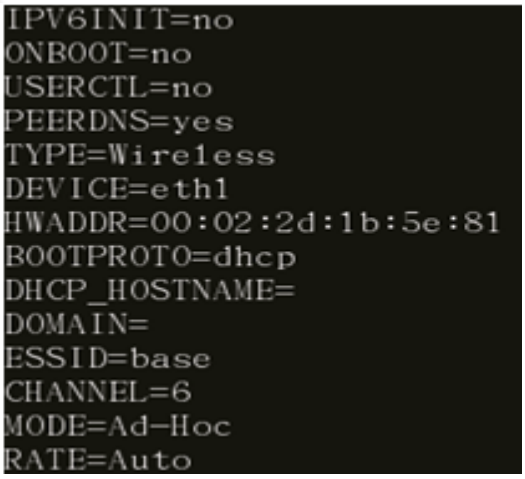
```
[root@media network-scripts]# ifconfig
eth1      Link encap:Ethernet  HWaddr 00:02:2D:1B:5E:81
          inet addr:10.10.1.1  Bcast:10.10.1.255  Mask:255.255.255.0
          inet6 addr: fe80::202:2dff:fe1b:5e81/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1415 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2282 errors:11 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:406437 (396.9 KiB)  TX bytes:1598845 (1.5 MiB)
          Interrupt:10 Base address:0x100
```

使用 ifconfig 指令後可以觀察 IP、BROADCAST、NETMASK 的數值是否正常，如果正常，通常網路就已經可以使用了；由於是使用無線網路，因此檢查步驟還要多一個，那就是使用 iwconfig 指令來

觀察。

這裡要檢查 Mode、Bit rate、ESSID 是否正確，這些數值都要跟剛剛設定檔案裡的數值是一樣的，否則一定是設定錯誤，而無線網路不能正常運作，通常也都是設定檔出了錯誤，這時候就要檢查設定檔是否有遺漏了什麼，或是哪裡的數值設定錯誤，如果一切正常，那麼就可以進行客戶端的設定了。而當要連上有加密的無線網路時，記得也要設定密碼，否則訊息都正確了，卻上不了網，更誤以為是設定錯誤，而不知如何是好；假使還是出現錯誤，請按照順序檢查，首先先檢查網路卡的設定檔，再來是檢查 network 的設定檔，如果在錯誤訊息裡看到網路卡搜尋的網域不是自己設定的網域，那麼就先檢查 network 這個設定檔了，network 的絕對位址為 /etc/sysconfig/network，使用編輯模式查看裡頭的 GATEWAY="" 和 GATEWAYDEV=""，是否為空白，假使這個參數選項有設定參數的話，那麼在啟動網路時，會優先並且只尋找這個 gateway，也就會出現啟動網路卡時，結果卻尋找不是此網卡設定的 gateway，而造成啟動失敗；在啟動失敗後，最重要的是先改掉 network 的設定，但別急著重新啟動，因為剛剛啟動失敗的程序沒有被正常的關閉，因此，無法再啟動網路卡，這時要先用#ps –aux 指令查看程序列表，找到剛剛所開啟的啟動網路卡指令，在使用＃kill -9 PID 指令（PID 指的是這個程序的號碼），把這個程序給關閉，最後再一次的重新啟動，這時網路就會正常被啟動。

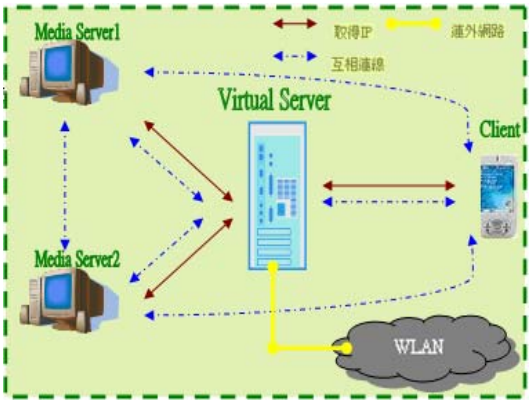
要進行客戶端的設定，一樣是要編輯客戶端的 ifcfg-X(X 為客戶端要啟動網路卡的代號)，跟伺服器端的設定比起來要進行客戶端的編輯，就沒有較繁複的更動了。



從上圖中可以看到，跟伺服器端的設定檔比起來簡潔許多，基本上只要把 BOOTPROTO＝static 的 static 改成 dhcp，把 ONBOOT＝no 的 no 改成 yes 即可，這代表著以後只要一開機，就會自動搜尋現有的無線網路設備，取得 IP 位址，當然如果所要連接的無線網路有加密的話，也要在同目錄下（/etc/sysconfig/network-scripts/），新增 key-ethx，這樣才能正常的連上網路。

第四章 伺服器架設

第一節 綱要



架設伺服器前需要經過詳細的規劃，真正有用的服務才進行架設的動作，否則開啟了一些用不到的服務，反而開啟無謂的連接埠，造成安全上的危害；從架構圖上可以看到，需要有一台 SERVER 能夠分配 IP，也需要有負載平衡的功能，目的是用來分擔頻寬的瓶頸，達到更高效能的服務；由於連外網路的需求，也需要支援 NAT 的封包偽裝功能，並且還要有提供客戶端多媒體應用的介面，那就需要網頁伺服器的架設。因此依照架構所需要的功能，我們大致上可分為四種伺服器的架設，第一種 DHCP Server、第二種 HTTP Server、第三種 NAT Server、第四種 Cluster Server。架設 DHCP Server 主要功能在於，能夠分配 IP 給接入網路的客戶端或是伺服器端，使得客戶端或是伺服器端能夠簡易的接入網路，提供服務或是使用資源；HTTP Server 功能在於，能夠提供給客戶端一個資源網站入口，讓使用者能夠清楚知道有哪些資源可以運用。NAT Server 功能在於，能夠讓客戶端或是伺服器端連上外部網路，獲取外部網路的資源，並且可以跟 Cluster Server 搭配使用來建構 VS-NAT 模式。

第二節 DHCP Server 原理

DHCP（Dynamic Host Configuration Protocol）主機最主要的工作，就是自動將網路參數配置正確的分配給網域中的各個電腦，讓客戶端的電腦可以在開機的時候就能立即自動的取得網路參數值，這些數值包括 IP、NETMASK、NETWORK、GATEWAY、DNS 等等。而 DHCP 的運作方式可以分為四個步驟：

第一，當客戶端開機或是重新啟動網路卡時，就會發送訊息到網路上，並且採用廣播方式，當有提供 DHCP 服務的主機接收到了，就會回應，否則一般電腦會將該封包丟棄。

第二，DHCP Server 收到要求的封包時，會依照自己設定檔裡的定義，在設定好的 IP 段裡，選取網域裡沒有被佔用的 IP 來分配給客戶端使用，並且會夾帶一個租約期限，來限制每個 IP 被使用的期限。

第三，客戶端接受到來自 DHCP Server 網路參數的回應，並且更改自己的網路環境，當客戶端接受回應的訊息之後，會先發出 ARP 封包，確定 DHCP Server 分配的 IP 沒有被佔用；如果該 IP 被佔用了，那麼客戶端將不會接受 DHCP 的回應，而且再一次的發出 DHCP 的請求封包，直到客戶端接受為止；客戶端會把接受的網路參數，替換自己的網路設定檔當中，同時也會對 DHCP Server 發出確認封包，告訴 Server 該 IP 已經使用，而

Server 也會把該資訊記錄下來。

第四，當租約到期或是客戶端離線時，該 IP 則會被 Server 收回；因此，當租約到期後，客戶端又會再次的向 DHCPServer 發出請求，取得 IP。

IP 的取得方式有兩種：

- 靜態方式（static）：當某些主機有特殊需求，如提供網內的一些服務時，就需要一個固定的 IP 來做為其他客戶端指向的目標，假使服務主機還使用動態 IP，那麼客戶端豈不是要常常重新設定主機位址；因此，DHCP Server 提供一種依據 MAC 位址，來分配某一個固定的 IP。
- 動態方式（dynamic）：客戶端每次連上 DHCP 所取得的 IP 都不會是固定的，而是由 DHCP Server 來分配。

所謂的租約，就是每個 IP 所能使用的時間，這是用在動態 IP 上面的；有個這個租約的動作，主要是可以在每過一段時間就重新讓客戶端取得 IP，這樣的好處是，一些停滯沒有在使用的 IP 可以被釋放出來。

第三節 DHCP Server 實作

在建構 DHCP Server 之前，先檢查 DHCP 套件是否有安裝，可以使用指令＃rpm –qa | grep dhcp 來達成。

```
[root@media RPMS]= rpm -qa | grep dhcp
dhcp-3.0.1-11.i386.rpm
dhcp-devel-3.0.1-11.i386.rpm
```

確定安裝了之後，就可以開始來實作了。

首先先設定 dhcpd.conf，dhcpd.conf 位在 etc 資料夾底下，使用 vi 來編輯。

```
ddns-update-style interim;
ignore client-updates;
subnet 10.10.1.0 netmask 255.255.255.0 {
        option routers 10.10.1.1;
        option subnet-mask 255.255.255.0;
        option domain-name-servers 10.10.1.1;
        option broadcast-address 10.10.1.255;
        range 10.10.1.30 10.10.1.40;
        default-lease-time 216000;
        max-lease-time 416000;
}
```

這邊可以看到 subnet 設定為 10.10.1.0 跟網路卡的設定是一樣的，而 netmask、broadcast…等等也都是一樣的，這邊要注意的是 range 參數，10.10.1.30 10.10.1.40 是 DHCP 分配的 IP 範圍，兩個端點間是鍵入空格而不是逗號，default-lease-time 為此 IP 的存活時間，時間到了就會依據 max-lease-time，這是 IP 最長存活時間，只要時間超過最大存活時間，則此 IP 會被收回重新分配一個 IP 給該用戶，接下來設定啟動 dhcp 服務檔 ，如圖：

```
# Command line options here
DHCPDARGS="eth1"
```

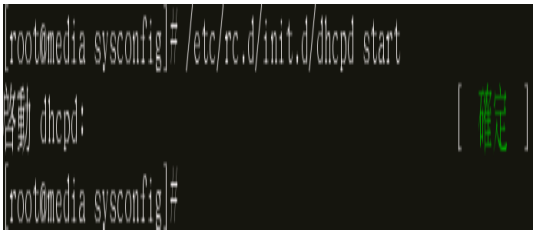
DHCPDARGS 為要啟動 DHCP 服務的網路卡代號，這裡設定成 eth1，也是啟動網路的 eth1。

再來設定網路，檔案為 /etc/sysconfig/network。

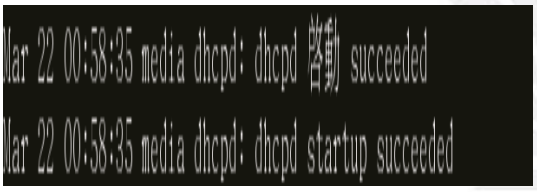
```
NETWORKING=yes
HOSTNAME=media.base.org
GATEWAY=
GATEWATDEV=
```

這裡的 GATEWAY 和 GATEWATDEV 設為無，主要是避免網路在啟動之初，去尋找這個 GATEWAY 位置。

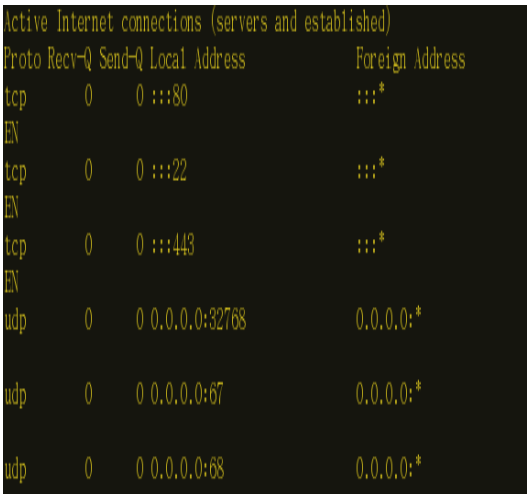
在一切設定就緒後，接下來就是要啟動 DHCP 的服務，#/etc/rc.d/init.d/dhcpd start 直接使用這個指令即可，或是#service dhcpd start 也可。



看到啟動 dhcpd:後面那個綠色的確定，代表剛剛的設定無誤，SERVER 服務有正常的啟動，如要更加仔細檢查是否啟動，或是出現錯誤，這時就要查看 log 檔案了，log 檔的存放位址為/var/log/messages，執行#vi /var/log/messages，並且搜尋 dhcpd 就可以找到訊息，如圖：



DHCP 正確被啟動了，或是查看 port 是否被開啟也可以當作檢查的依據，執行#netstat –an | less 如圖：



知道 dhcp port 為 67 port，而上圖顯現出 67 port，代表著 67 port 被開啟了，因此可以很確切的知道 dhcp 服務被啟動了，這樣一來，主機即擁有 DHCP 服務功能了。

第四節 HTTP Server 原理

HTTP（HyperText Transfer Protocol）是目前網路的資料傳遞協定。當客戶端在網址列輸入網址後，會先經過 DNS 解析得到網路主機的 IP，然後會發出一個資料封包，以 http 協定連到網路主機，並且使用 http 來取得資料，同時也會使用 TCP 協定；而當網路主機收到資料封包後，會根據客戶端的請求，提供相關的資訊來回應，而大部分的時候，皆會使用 http 協定傳送具有 HTML 語法的網頁到客戶端的瀏覽器；最後客戶端的瀏覽器接收到資料後，便會把 HTML 語法透過轉換後，在瀏覽器上呈現出來。

第五節 HTTP Server 實作

一樣的先查看套件是否安裝。透過指令#rpm –qa | grep http 來行使。

確定安裝後，就開始編輯設定檔了。

要編輯的檔案是在/etc/httpd/conf/底下的 httpd.conf，因此使用指令 #vi /etc/httpd/conf/httpd.conf 來編輯。

```
ServerRoot "/etc/httpd"
PidFile run/httpd.pid
Timeout 120
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15
<IfModule prefork.c>
StartServers       8
MinSpareServers    5
MaxSpareServers   20
ServerLimit      256
MaxClients       256
MaxRequestsPerChild  4000
</IfModule>
```

ServerRoot“/etc/httpd” 為最上層的目錄，也就是放置一些設定檔的目錄；MaxKeepAliveRequests 100 為在連續連線中所能允許的最大連線數，這可以視架設網站的規模來決定數值的大小；KeepAliveTimeout 15 為同一個連線的客戶端在 15 秒內沒有送出請求，那麼該連線就會被視同斷線；MinSpareServers 與 MaxSpareServers 是開啟 httpd 服務數目的地方，這個的意思為，當 MaxSpareServers 設定為 5 的時候，如果有超過 5 個人同時上站，那麼超過的人，就無法順利的連上網站，那就得重新再傳送一次需求，因此這也需要考量所架設的網站規模來決定大小，並且也得依據伺服器硬體的效能來考量。

```
<IfModule worker.c>
StartServers         2
MaxClients         150
MinSpareThreads     25
MaxSpareThreads     75
ThreadsPerChild     25
MaxRequestsPerChild  0
</IfModule>
Listen 80
User apache
Group apache
ServerAdmin root@localhost
```

MaxClients 150 為伺服器最大所能容納 150 個客戶端數，數值也是由網站規模和硬體能來作考量；Listen 80 為伺服器所使用的 port；因為安全關係，避免把 User 和 Group 設定成 root，主要是因為，在設定 SpareServers，會開啟一些程序，這些程序的擁有者和擁有群組，就是這裡設定的 User 和 Group，假使 HTTP Server 被入侵了，也不至於被取得 root 的權限。

```
DocumentRoot "/var/www/html"
<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>
<Directory "/var/www/html">
    Options FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
```

DocumentRoot "/var/www/html"，這是放置主網頁的地方，以及主網頁目錄的基本設定，如沒有特殊須求，使用預設值即可，如以下參數為可更改：

1. ExecCGI ：使該目錄具有使用 CGI 的能力！
2. FollowSymLinks ：讓在別的目錄 link 到目錄或檔案，也可以連接出去！
3. Includes ：在 Server 端的工

作可進行！

4. Indexes ：如果在該目錄底下找不到預設的網頁名稱時，就顯示整個目錄下的檔案！

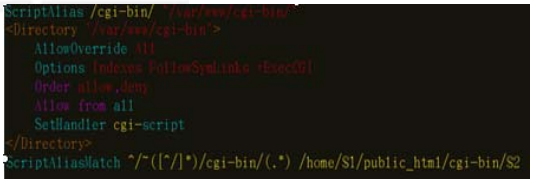
5. MultiViews ：可以編寫多個不同語言的檔案在同一個目錄下或是同一個檔案，有點類似多國語系的支援！

6. All ：全部的屬性都啟動，但是不包含 MultiViews ！

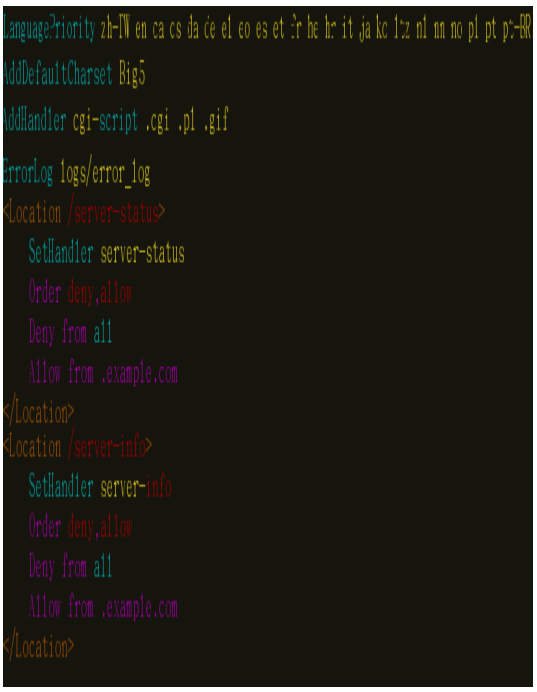
瞭解這些參數後，就能依據自己的需求，來設定自己的目錄屬性了，但這裡乃採用預設值即可。

UserDir public_html 這是在設定個人家目錄下的首頁其所在的目錄，讓主機上的使用者都可以建立屬於自己個人網頁；預設是 public_html，如/home/base/public_html，這就是使用者的首頁目錄，而緊接的是 public_html 的屬性設定。

DirectoryIndex 是預設的首頁檔案名稱，可以設定多個，有優先順序排列，越前面的越先被顯示出來，在這之下的是專門用來放置圖片的目錄，而且這裡還有個小技巧，那就是 Alias /images/ /home/images/ public_html‘，這代表著把/images/指向 /home/images/public_html 目錄，簡便的地方就是可以省略個人用戶的網址有個”～”符號。如：http://domainname/~images/ > http://domainname/images/，如此一來，網址不就變的更簡易了。

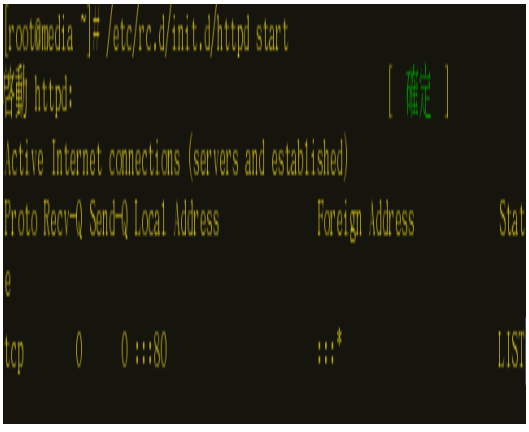


如果要在主機上使用 CGI 程式的話，那麼這個 CGI 目錄的設定就要加入在設定檔裡。



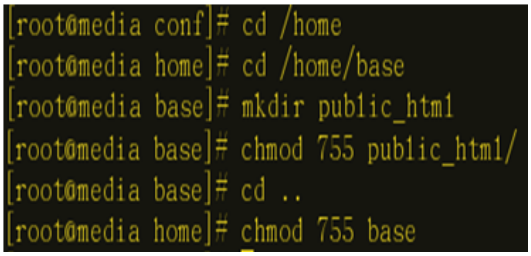
一開始為語言優先順序的設定，這裡設定為中文繁體，並且把預設的編碼設定為 BIG5；如果要讓主機也能夠支援 Perl 的程式，那就要在 AddHandler cgi-script .cgi 後面再加上.pl；再來就是伺服器的狀態以及說明的功能開啟。

這樣大致上網頁伺服器就大致設定完成了，然後要啟動伺服器，使用指令#/etc/rc.d/init.d/httpd start 或 server httpd start。



這裡可以看到 httpd 被成功的啟動了，接著檢查 port 來看看是否被開啟，80port 確實被開啟了，那麼 HTTP Server 就架設成功。

既然 httpd 已經啟動了，再來就是建立用戶的個人網頁；首先，先在個人目錄裡建立 public_html 目錄。



#chmod 指令為更改權限，755 代表著擁有著讀、寫、執行的權力，而同群組和其他人，只擁有讀和執行的權限，如此一來別人就能瀏覽此目錄裡的網頁，如果沒有這一個步驟，那麼來瀏覽的人就會看到網頁呈現錯誤。

個人目錄設定好了之後，接下來就是要建立一個網頁來測試看看了，使用指令#vi index.htm

網頁建立好了，一樣要先更改網頁的權限，改成 755 這樣別人才能夠瀏覽。

測試網頁是否能連上 http://localhost/~base/index.htm

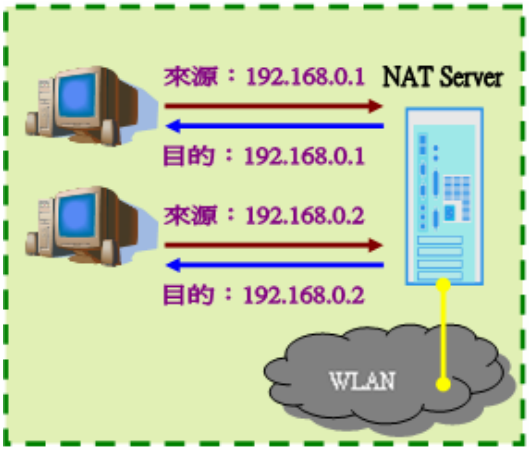
測試成功。

在設定檔中有把主機狀態開啟，因此只要在網址列鍵入：http://localhost/server-status 即

可觀看主機狀態。

第六節 NAT Server 原理

要能夠連上 Internet 必須擁有真實 IP（Public IP），但是在目前 IPv4 協定下，IP 數已經明顯不足，因此才會發展出使用私有 IP（private IP），再透過 NAT 的偽裝技術連上 Internet；如圖，其運作原理可以分為 5 個動作：

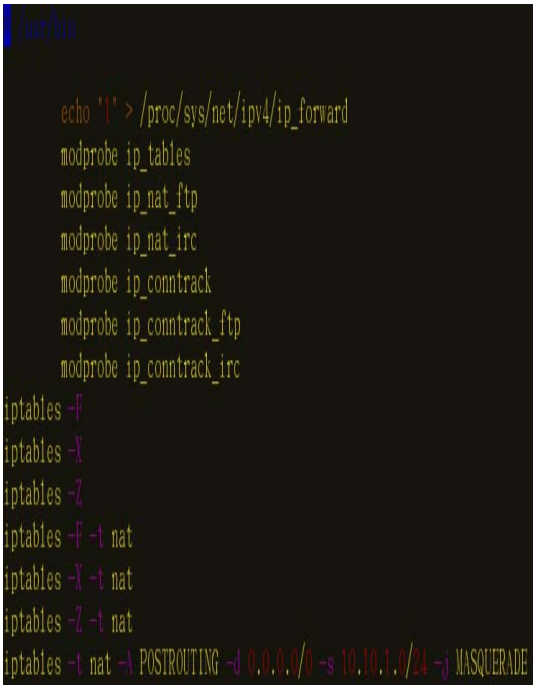


- 首先，客戶端的 gateway 要設定成 NAT 的位址，確保客戶端啟動網路時是連向 NAT。
- 當客戶端有要連到外部網路時，此時該封包就會被送到 NAT 上，而這時封包標頭的來源 IP 為 192.168.0.1 或是 192.168.0.2。
- 而這時候，NAT 會將封包標頭偽裝成自己 Public IP，並且記錄此連線封包是屬於哪一個客戶端的。
- 當外部網路的主機回應時，NAT 主機就會去對照路由的記錄，查到接收到的封包是哪一個客戶端的。
- NAT 主機把封包轉送給所對應的客戶端。

第七節 NAT Server 實作

要在 Linux 建構簡易 NAT Server 是非常容易的辦到的，NAT 架設只需幾行指令就可以了，另外可以把指令寫成 script 來完成自動化。基本上只要透過 iptables 來偽裝內部 IP 成外部 IP 連外的 ip 即可達到 nat 的作用。

直接製作 script，執行指令#vi nat.sh



一開始，先把 ip_forward 功能開啟，再把一些需要的模組加入到系統中，最後再用 iptables 來實現偽裝 IP 的效果，藉此達到 NAT 功能。iptables 有幾個重要的參數需要瞭解：

1. –F 清掉所有訂定的規則
2. –X 清掉所有使用者訂定的 tables
3. –Z 將所有的 tables 的計數和流量統計都歸零
4. –t nat NAT tables
5. –A 新增一條規則
6. –j 動作，MASQUERADE 則為偽裝的動作

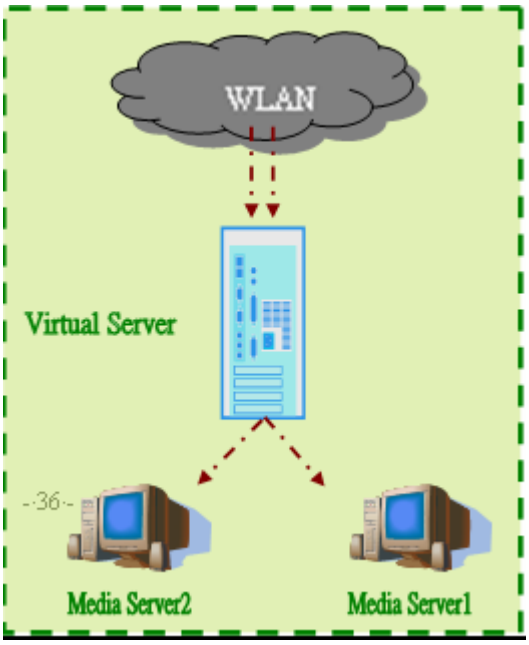
有了這些參數，就可以組合成一個具有 NAT 功能的 script，完成 script 的編寫之後，還是要記得更改檔案的權限，執行指令#chmod +x nat.sh ，＋x 這樣做的目的是讓檔案有可執行的能力，有別於先前說說的數字如 755，其實都是一樣的意思。

最後再來測試 script ，執行指令#./nat.sh，並且執行指令#iptables –t nat –L 來觀看結果。

確認偽裝的指令有被加入到 nat table 裡面即可，這樣這台主機就具有 NAT 功能了。

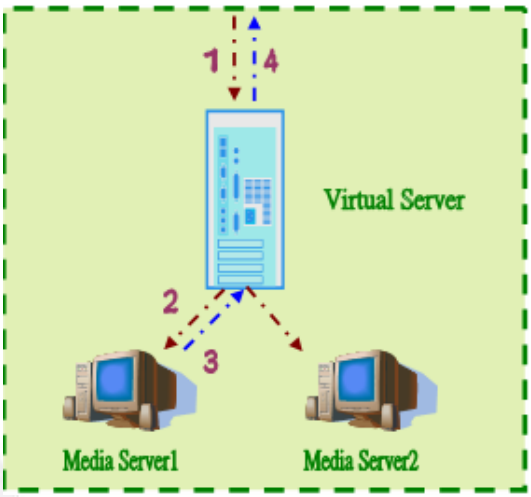
第八節 Cluster Server 原理

當今網路的迅速成長，隨著 Internet 服務的日新月異，網路流量也呈爆速的成長，因此，各家公司的伺服器要如何來應付這樣的流量？而叢集伺服器（Cluster Server）就是為了因應這樣的狀況所產生的，藉由分流的技術，把流量分別導引到不同的伺服器上，以降低伺服器的負載情形。Cluster Server 使用 LVS（Linux Virtual Server ） 來實施負載平



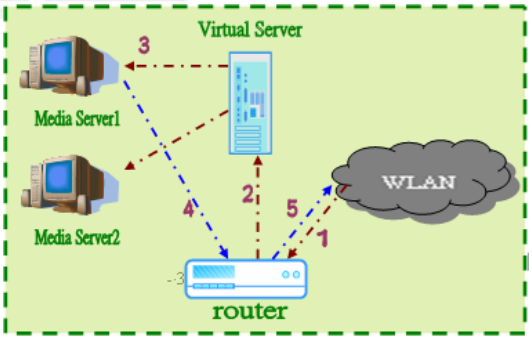
衡叢集，LVS 基本架構是採用一台導引器稱為虛擬伺服器，來處理要求的連線，並且依據設定好的調度演算法，判斷後端適合分配的真實伺服器，分配要求的連線到該台真實伺服器，所謂的調度演算法基本上可以分為四種，第一種輪序（Robin Robin）簡稱為 rr，將連線按照順序輪流的分配給真實伺服器，第二種加權輪序（Weighted Round Robin）簡稱 wrr，將每個真實伺服器定義權重，權重越重的伺服器所分配到的連線就越多，第三種最少連線數（Least-Connection）簡稱 lc，當虛擬伺服器分配給一個真實伺服器時，會將該真實伺服器的連線記錄上加一，等到真實伺服器處理完該連線後，即將記錄減一，當有新連線時，虛擬伺服器就會比較各真實伺服器的連線數，而將連線分配給連線數最少的真實伺服器，第四種加權最少連線數（Weighted Least-Connection）簡稱 wlc，就是加權輪序和最少連線數演算法相結合。接下來 LVS 也可以分為 VS-NAT、VS-DR、VS-TUN 三種模式：

- **VS-NAT**：使用 NAT 技術，在虛擬伺服器上設定 NAT，讓所有的流量都經過虛擬伺服器。其過程如圖：



外界先有請求封包傳到虛擬伺服器，再透過虛擬伺服器分配到 media Server1，media Server1 再將所要求的資訊透過虛擬伺服器傳送給客戶端。

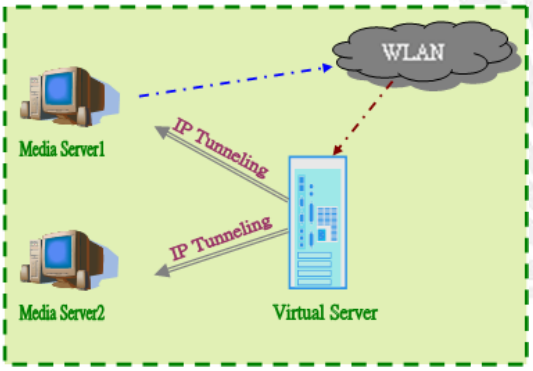
- **VS-DR**：跟 VS-NAT 類似，VS/DR 通過改寫請求封包的 MAC 位址，將請求傳送給真實伺服器，而真實伺服器將回應直接返回給客戶端。這種方法沒有支援 IP 隧道協議的要求，但是要



求虛擬伺服器和真實伺服器都有一塊網卡連在同一網段上。而其過程如圖：第一步驟，客戶端發出請求封包後，封包經由 Internet 傳送到終端網路設備（router）；第二步驟，router 再把封包傳送到虛擬伺服器；第三步驟，

虛擬伺服器依據分配原則，把請求封包改送給真實伺服器；第四步驟，真實伺服器把所需要的資訊傳送給 router 而不是虛擬伺服器；第五步驟，router 把接收到的封包傳回客戶端。

- **VS-TUN**：虛擬伺服器把請求封包通過 IP 隧道轉發至真實伺服器，而真實伺服器將所需要的資訊直接回應給客戶，所以虛擬伺服器只處理請求封包。IP 隧道（IP tunneling）是將一個 IP 封包封裝在另一個 IP 封包的技術，這可以使得目標為一個 IP 位址的資料封包能被封裝和轉送到另一個 IP 位址。IP 隧道技術亦稱為 IP 封裝技術(IP encapsulation)。IP 隧道主要用於移動主機和虛擬私有網路（Virtual Private Network），在其中隧道都是靜態建立的，隧道一端有一個 IP 位址，另一端也有唯一的 IP 地址。而其過程如圖：



利用 IP 隧道技術，虛擬伺服器將請求封包封裝轉送給真實伺服器，而所請求的資訊封包直接從真實伺服器直接回應給客戶。而這裡是用動態地選擇一台伺服器，將請求封包封裝和轉送給選出的伺服器。這樣，就可以利用 IP 隧道的原理將一組伺服器上的網路服務組成在一個 IP 位址上的虛擬網路服務。如圖為三種模式的特性：

	VS-NAT	VS-DS	VS-TUN
外送連線處理方式	以虛擬伺服器爲對外、內連線的通訊閘	真實伺服器使用原有通訊閘直接回應	真實伺服器使用原有通訊閘直接回應使用者
節點間連線	內部網路	單一網段的區域網路	任何網段均可
真實伺服器理論上最大數量	10~20	>100	>100
叢集效率(MAX)	Low	Best	Good
對真實伺服器的需求	不拘	需支援封包過濾或忽略ARP	需支援IP Tunneling技術

依據官方三種模式的性能報告，如果是使用 100M 的網路卡，採用 VS/TUN 或 VS/DR 調度技術，群組系統的吞吐量可高達 1Gbits/s；如使用 1000M 的網路卡，則群組系統的最大吞吐量可接近 10Gbits/s。

第九節 Cluster Server 實作

在架設之前先確認一下，IPVS 模組、ipvsadm 工具是否被支援與安裝。首先，檢查 ipvs 套件，使用指令 #rpm –qa | grep ipvsadm。

```
[root@media RPMS]# rpm -qa | grep ipvsadm
ipvsadm-1.24-5
[root@media RPMS]#
```

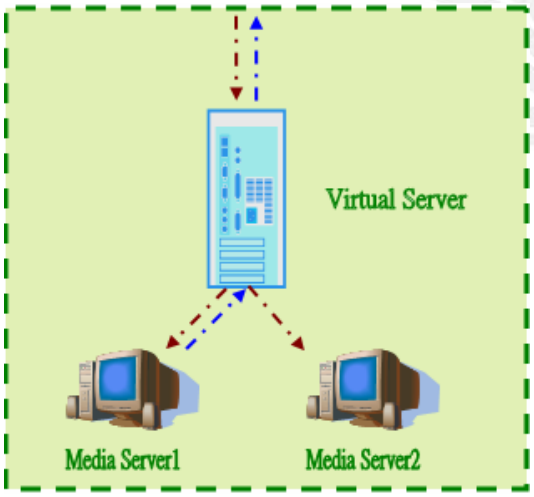
檢查 IPVS 模組是否被加入系統，使用指令#grep “CONFIG_IP_VS” /boot/config-`uname -r`。

```
[root@media ~]# grep 'CONFIG_IP_VS' /boot/config-`uname -r`
CONFIG_IP_VS=m
# CONFIG_IP_VS_DEBUG is not set
CONFIG_IP_VS_TAB_BITS=12
CONFIG_IP_VS_PROTO_TCP=y
CONFIG_IP_VS_PROTO_UDP=y
CONFIG_IP_VS_PROTO_ESP=y
CONFIG_IP_VS_PROTO_AH=y
CONFIG_IP_VS_RR=m
CONFIG_IP_VS_WRR=m
CONFIG_IP_VS_LC=m
CONFIG_IP_VS_WLC=m
CONFIG_IP_VS_LBLC=m
CONFIG_IP_VS_LBLCR=m
CONFIG_IP_VS_DH=m
CONFIG_IP_VS_SH=m
CONFIG_IP_VS_SED=m
CONFIG_IP_VS_NQ=m
CONFIG_IP_VS_FTP=m
```

確認所有模組都被支援了。

■ **VS-NAT 架構：**

真實伺服器透過虛擬伺服器跟客戶端溝通，由於此方式只要將真實伺服器的通訊閘，設定成虛擬伺服器的位址即可，就不用再做其他的設定。



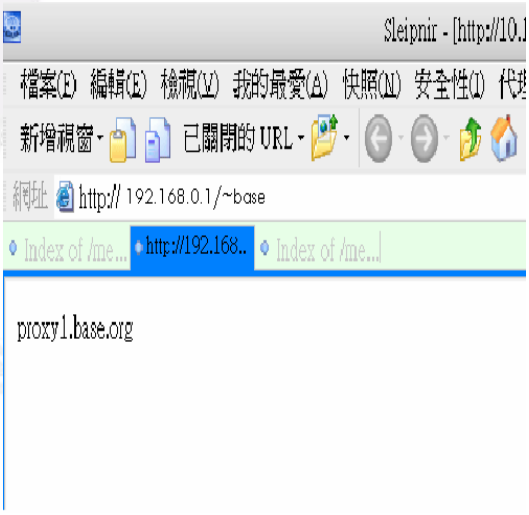
VS-NAT 設定：

先在虛擬伺服器上設定 NAT，再寫 script 來完成叢集的設定。使用指令 #vi vs-nat.sh

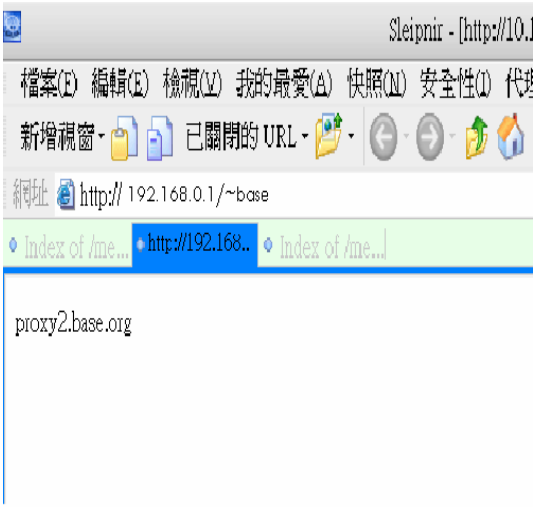
```
#/usr/bin
        ipvsadm -A -t 192.168.0.1:80 -s rr
        ipvsadm -a -t 192.168.0.1:80 -r 10.10.1.38 -m
        ipvsadm -a -t 192.168.0.1:80 -r 10.10.1.37 -m
~
```

一開始是 ipvsadm 的編寫，-A 為增加服務的意思、-t 為使用 TCP 通訊協定、-s 為使用哪種調度演算法、-a 為增加一個真實伺服器服務、-r 設定真實伺服器的 IP、-m 設定 VS-NAT 模式，由上設定可以知道，設定 192.168.0.1：80 為虛擬伺服器提供服務，並且使用輪序調度演算法來分配連線，10.10.1.37、10.10.1.38 為真實伺服器，採用 VS-NAT 模式。因為 VS-NAT 的特性，所以在真實伺服器上，不需再做特別的設定。

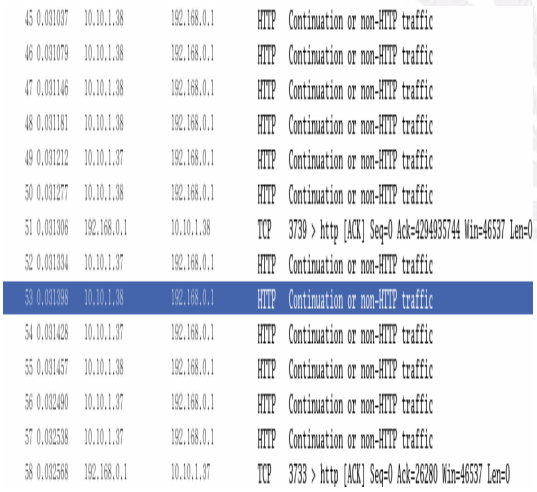
VS-NAT 實測：



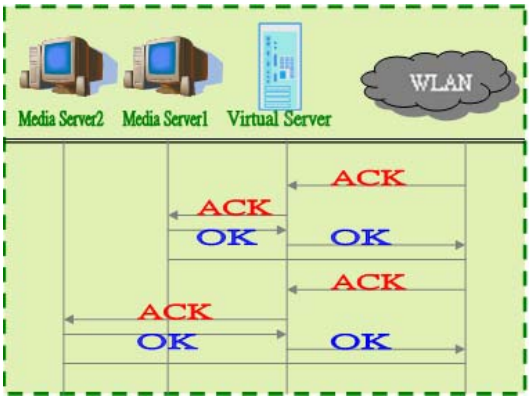
如圖可以看到，一開始連上主機所呈現出來的網頁，為第一台服務主機，這是因為 VS-NAT 虛擬伺服器採用的調度演算法為輪序，因此在沒有先前的連線紀錄時，會先把連線分配給第一台服務主機，然後，再經過重新整理後，會變成下圖。

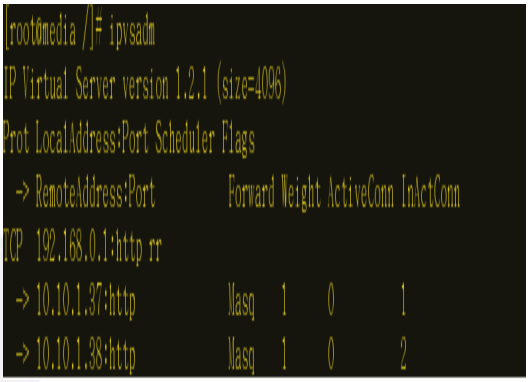


如圖可以看到，客戶端的連線是導向第二台服務主機，因為第一台服務主機已經被分配連線一次了，而經過輪序的分配原則，第二次分配連線就要導向第二台服務主機；這邊也可以看到，雖然被導向第二台服務主機，但是網址卻沒有改變，由此，可以看出負載平衡對客戶端來說，有如隱形般沒有感覺。



由封包內容可以很清楚的看到輪序的過程，如以下圖示，兩個客戶端發出請求後 192.168.0.1 分別把請求傳送給 10.10.1.37 和 10.10.1.38；10.10.1.37 和 1010..38 再透過虛擬伺服器跟客戶端連線，



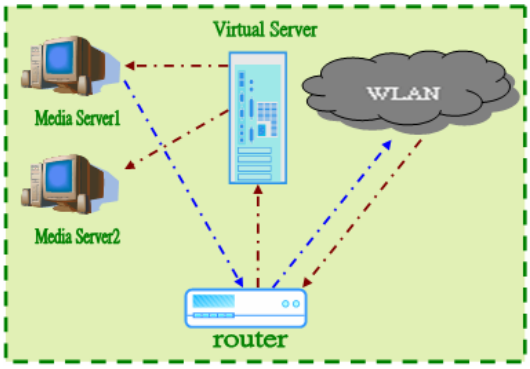


被虛擬伺服器轉送的連線數會顯示到 InActConn 這個項目，也可以看到封包轉送模式是使用 Masq(偽裝)，代表著使用 VS-NAT 模式。

由以上的說明及設定可以知道，VS-NAT 的導向完全都是由虛擬伺服器來控制，這樣的好處就是設定簡易，不需在各個真實伺服器上作設定，並且只要有支援 TCP/IP 的作業系統上都可以運作，而壞處就是流量都集中在虛擬伺服器上，容易造成瓶頸，因此這種方法只適合用在小流量的服務。

■ **VS-DR 架構：**

如圖，客戶端發出請求後，透過虛擬伺服器分送給真實伺服器，真實伺服器再透過原有的網路設備來跟客戶端溝通；因此，把真實伺服器的預設通訊閘設定成 router 的 IP 即可。



VS-DR 設定：

一樣寫一個 script 來完成真實伺服器的所有動作。使用指令 #vi vs-drreal.sh。

```
# /usr/bin
    echo "1" > /proc/sys/net/ipv4/ip_forward
    iptables -t nat -A PREROUTING -p tcp -d 10.10.1.1 --dport 80 -j REDIRECT
--to-port 80
```

啟動 ip_forward，再用 iptables 把接收到的封包重導向到 80port，這樣做，是因為真實伺服器接到虛擬伺服器轉送過來的封包不是自己的，因此透過這樣的重導向方式，使真實伺服器能夠接收封包。

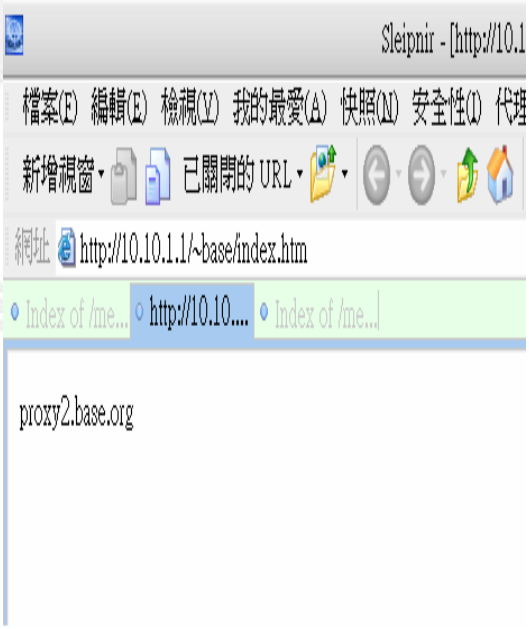
接下來是虛擬伺服器的設定，使用指令#vi vs-drvip.sh。

```
# /usr/bin
    echo "1" > /proc/sys/net/ipv4/ip_forward
    ipvsadm -A -t 10.10.1.1:80 -s rr
    ipvsadm -a -t 10.10.1.1:80 -r 10.10.1.37:80 -g
    ipvsadm -a -t 10.10.1.1:80 -r 10.10.1.38:80 -g
```

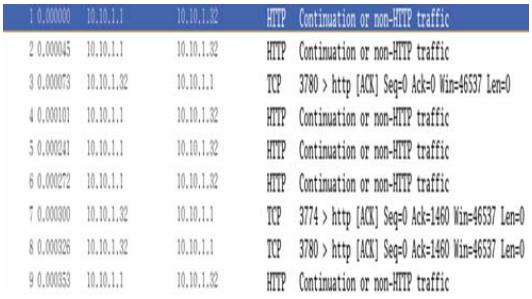
首先先開啟 ip_forward，讓這台虛擬伺服器能夠支援封包轉送的功能；新增 10.10.1.1：80，10.10.1.1 為虛擬伺服器的 IP 位址，這裡使用 rr 為調度演算法，下面兩項則為新增兩個真實伺服器 10.10.1.37 和 10.10.1.38，然後使用-g 參數，設定分流模式為 VS-DR。

VS-DR 實測：

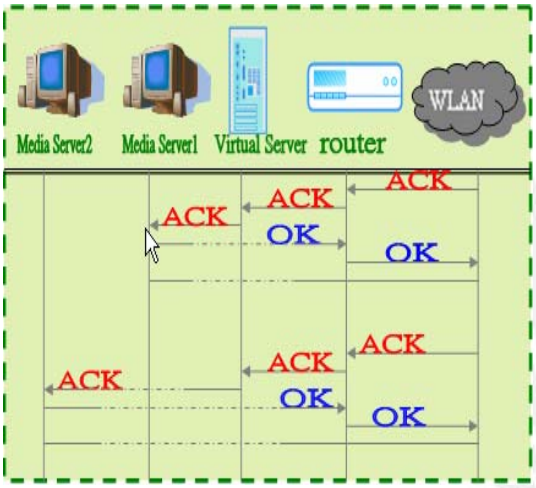
當我們一開始連線時，網頁所呈現的是第一台服務主機的內容，在重新整理過後，變成下圖。



網頁呈現第二台服務主機的內容，這代表連線被正確的導向第二台服務主機，而且網址也沒有變動，這樣客戶端就會完全感受不出來後端伺服器的存在。



於 VS-DR 是透過原有的網路設備連線，因此可以看到只有一台網路設備在與虛擬伺服器溝通，不會顯示出後台真實伺服器的連線狀態。而客戶端連向虛擬伺服器時，其過程跟 VS-NAT 有著些許的不同，



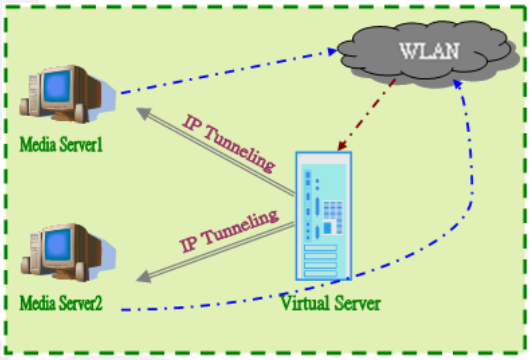
如上圖：客戶端傳送請求封包給路設備(router)，router 再把請求封包傳送到虛擬伺服器，虛擬伺服器藉由分配原則，把封包傳送給真實伺服器，而真實伺服器卻把回應的封包直接傳送給 router，跳過虛擬伺服器，正是因為這樣，大大減少虛擬伺服器的工作量，也減少了網路流量，VS-DR 的效能才會比 VS-NAT 要好；由於虛擬伺服器只需要處理請求的封包就可以了，因此就可以擴大群組，根據官方的數據，VS-DR 可以擴充真實伺服器的數目到 100 台之多。

```
[root@media /]# ipvsadm
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  10.10.1.1:http rr
  -> 10.10.1.38:http              Route   1      1          0
  -> 10.10.1.37:http              Route   1      1          0
```

由上圖可以看到，封包轉送模式是透過 Router，正在連線數顯示在 ActionConn 項目下。

■ VS-TUN 架構

在使用 VS-NAT 時，連線的請求和回應，都需要透過虛擬伺服器來做處理，因只要真實伺服器數量一多，那麼虛擬伺服器將會是很大的瓶頸所在，而使用 VS-TUN 最大的優點就是



他可以跨網域，可以把請求和回應分開來，利用 IP Tunnel 技術，將請求封裝轉向給真實伺服器，真實伺服器再直接回應給客戶端 。

VS-TUN 設定：

先在虛擬伺服器上設定另一個 IP，來作為 Tunnel。使用指令 #vi ifcfg-eth1:0。

```
IPV6INIT=no
ONBOOT=no
USERCTL=no
PEERDNS=yes
#GATEWAY=10.10.1.1
TYPE=Wireless
DEVICE=eth1:0
#HWADDR=00:02:2d:1b:5e:81
BOOTPROTO=static
#BOOTPROTO=dhcp
NETMASK=255.255.255.0
DHCP_HOSTNAME=
IPADDR=10.10.1.2
NETWORK=10.10.1.0
BROADCAST=10.10.1.255
DOMAIN=
#ESSID=default
ESSID=base
CHANNEL=6
#MODE=Managed
MODE=Ad-Hoc
#RATE=11Mb/s
```

這裡的設定跟網路卡設定是一樣的，IP 設定為 10.10.1.2，DEVICE 設定為 eth1:0，而 eth1:0 的意思為，在網路卡 eth1 上，再建立一個網路服務。

啟動 eth1:0 ，使用指令#ifup eth1:0。

```
eth1:0    Link encap:Ethernet  HWaddr 00:02:2D:1B:5E:81
          inet addr:10.10.1.2  Bcast:10.10.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:10 Base address:0x100
```

確認 eth1:0 被啟動。

接下來在真實伺服器上建立 Tunnel。使用指令#vi ifcfg-tunl0。

```
# Lucent Technologies WaveLAN/IEEE Adapter
DEVICE=tunl0
ONBOOT=yes
BOOTPROTO=static
NETMASK=255.255.255.255
DHCP_HOSTNAME=
IPADDR=10.10.1.2
DOMAIN=
#HWADDR=00:02:2d:1b:5e:73
USERCTL=no
PEERDNS=yes
#GATEWAY=
BROADCAST=10.10.1.255
TYPE=Wireless
IPV6INIT=no
ESSID=base
CHANNEL=6
MODE=Ad-Hoc
RATE=11Mb/s
```

DEVICE 設定為 tunl0，IP 設定為跟虛擬伺服器 eth1:0 一樣的 IP：10.10.1.2，NETMASK 設定為 255.255.255.255，其他設定則相差不多。

再來就是啟動 tunl0。使用指令#ifup tunl0。

```
tunl0     Link encap:IPIP Tunnel  HWaddr
          inet addr:10.10.1.2  Mask:255.255.255.255
          UP RUNNING NOARP  MTU:1480  Metric:1
          RX packets:62375 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2561388 (2.4 MiB)  TX bytes:60 (60.0 b)
```

可看到 Link encap 的地方為 IP Tunnel，代表 IP Tunnel 被啟動。

在虛擬伺服器上的設定，透過編寫 script 來達成。使用指令#vi vs-tunl.sh。

```
/# /usr/bin
touch /va/lock/subsys/local
echo 0 > /proc/sys/net/ipv4/ip_forword
route add -host 10.10.1.2 dev tunl0
echo 1 > /proc/sys/net/ipv4/conf/all/arp_ignore
echo 1 > /proc/sys/net/ipv4/conf/lo/arp_ignore
ipvsadm -A -t 10.10.1.1:80 -s rr
ipvsadm -a -t 10.10.1.1:80 -r 10.10.1.37 -i
ipvsadm -a -t 10.10.1.1:80 -r 10.10.1.38 -i
```

關閉 ip_forward，開啟 arp_ignore，增加 route 到 tunl0，ipvsadm 參數-i 為使用 VS-TUN 模式。因為 VS-DR 和 VS-TUN 有 arp 回應的問題，關閉 arp 回應就可以解決此問題。

編寫 VS-DR script。使用指令#vi vs-dr-nonarp.sh。

```
# /usr/bin
ip rule add prio 99 from 10.10.1/24 table
99
ip route add table 99 blackhole 10.10.1.1

ip rule add prio 100 table 100
ip route add table 100 local 10.10.1.1
dev lo
```

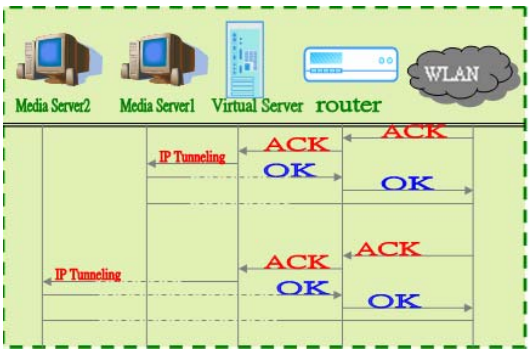
編寫 VS-TUN script。使用指令#vi tun-nonarp.sh。

```
# /usr/bin
ip rule add prio 99 from 10.10.1/24 table
99

ip route add table 99 blackhole 10.10.1.2

ip rule add prio 100 table 100
ip route add table 100 local 10.10.1.2
dev lo
```

VS-DR 和 VS-TUN 只差在 IP 設定的不同，VS-DR 是作在 10.10.1.1，而 VS-TUN 則是作在 10.10.1.2。大致上 Cluster Server 三種模式都架設成功了。

VS-TUN 實測：

VS-TUN 和 VS-DR 的實測結果是差不多的，差別只是 VS-TUN 的虛擬伺服器傳輸到真實伺服器的方式是採用



IP Tunneling。

連線到網頁瀏覽，正確無誤的導向了第一台服務主機，而在重新整理過後，就會變成下圖。

順利的變成第二台服務器了，跟 VS-DR 很相似，這是因為 VS-DR 和 VS-TUN 連結虛擬伺服器，和真實伺服器連結客戶端的模式是一樣的，僅僅只差在虛擬伺服器分配連線的方法不一樣，而且這兩種模式的效能非常相近，幾乎都可以把群族擴充到 100 台伺服器之多，但由於此種方法需要作業系統支援 IP Tunneling 或者 IP

Encapsulation 的協議，才可運作，因此目前大部分的測試或是實作運用，都是實施在 Linux 的作業系統上。

第五章 UPNP 支援

第一節 綱要

UPnP是針對智慧家電、無線設備以及各種外觀尺寸的個人電腦的普遍對等（peer-to-peer）網路連接而設計的一種架構。它旨在為家庭、小型企業、公共場所中或連接到Internet的ad-hoc網路或未管理網路提供易於使用、靈活且基於標準的連接。UPnP是一個充分利用TCP/IP和Web技術的分散式開放型網路體系結構，除了能夠在家中、辦公室和公共場所聯網設備之間的完整控制和資料傳輸之外，還可建立無縫緊密的連接網路。

UPnP不僅僅只是即插即用模式的簡單擴展。它設計用於支援零配置、“不可見”聯網，以及對眾多廠商的廣泛設備類型的自動發現。這就代表著，一台設備能夠動態加入一個網路，獲取一個IP位址，通報其功能，以及瞭解其他設備的存在和功能。DHCP和DNS伺服器為可選伺服器，僅當它們在網路上存在時可以使用。最後，設備能夠順利地自動離線，而不會造成任何的影響。

UPnP充分利用了包括IP、TCP、UDP、HTTP和XML在內的網路組件。而其網路溝通是以XML來表達，並通過HTTP進行傳輸。

何為 UPnP 的“通用性”？不使用設備驅動程式；取而代之的是通用協定。UPnP 網路不依賴於任意媒體。UPnP 設備可以在任何作業系統上採用任何編譯程式語言來實現。UPnP 通過使用瀏覽器和傳統應用程式控制來使廠商能夠控制設備的用戶介面（UI）並實現交互。

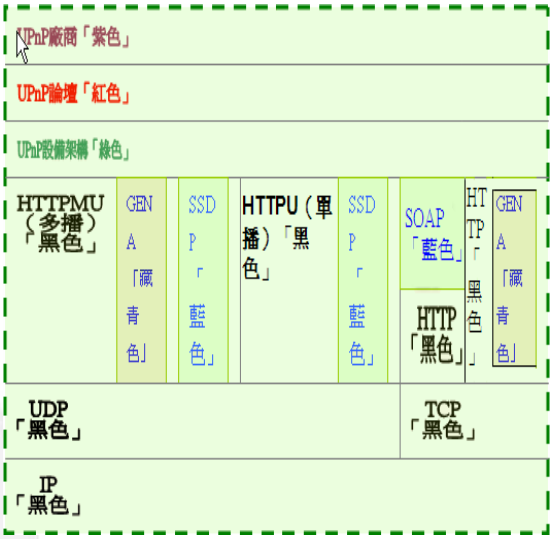
微軟提出UPNP，全名為Universal Plug and Play，是DHCP的延伸，UPnP最大的好處就在於，任何設備(比如說:電視、冰箱、PDA、音響.……etc.)只要加入網路，便能馬上使用；而所有網路上的設備，即會被通知有新設備的加入，進而每個設備間可以互相溝通，更可以控制它，使用者不需做任何的設定，因此Auto即是UPnP的代名詞。看在未來家電採用UPNP技術的關係，因此加入UPNP的支援，好讓未來有支援UPNP的硬體，能夠簡易的就接入原有的網路；微軟提出了一組有趣的API：NAT Traversal，搭配UPnP閘道器解決了目前NAT大部分的問題，由於IP位址的不足，NAT(Network Address Translation)被廣泛的使用，然而NAT遭遇許多問題；當在NAT下提供網際網路服務時，最大的問題就是NAT伺服器需設定連接埠對應(port-mapping)，否則網際網路上的電腦無法與提供服務的主機連線，而設定連接埠對應需要手動設定，這點造成非常大的困擾，如果NAT下有幾百台電腦，那設定、修改連接埠對應可不是件容易的事。許多應用程式會假設客戶端是使用真實IP,在傳送資料到伺服器時,會將IP嵌入到應用程式封包表頭，當伺服器收到後直接以表頭內的IP位址回傳資訊,如果客戶端在NAT下，由於私有IP(Private IP)無法透過網際網路傳送，因此會造成伺服器端無法回傳資料到客戶端的情形。應

用程式使用不同連接埠傳送，如有些網路應用程式使用連接埠x傳送資料，但是預期用連接埠y接收資料，NAT伺服器看到從x連接埠流出封包，因此替x作port-mapping，但卻不知道是要用連接埠y接受封包，因此會把流向連接埠y的封包丟棄。由於以上問題，在有了UPnP後，如果使用的閘道器有支援UPnP(微軟對支援UPnP 的閘道器統稱IGD：Internet Gateway Device)，那麼既然UPnP 可以知道裝置資訊，設定狀態變數，那不就能利用這些資訊來知道目前的狀態是不是在NAT下，進而自己設定連接埠對應這類動作來解決一些NAT的問題，因此，微軟提出了NAT Traversal 這套API，NAT Traversal ，而其可以提供以下功能：

- 確認NAT 是否存在
- 從他處（例如閘道器）取得IP 位址
- 取得靜態連接埠的對應資訊
- 加入靜態連接埠的資訊，除非之前已經設定了外部連接埠
- 在不刪除連接埠的情形下，啟用或停用該連接埠
- 為靜態連接埠設定一個名稱
- 刪除靜態連接埠的對應資料
- 從區域網路上取得靜態連接埠的清單

有了這套API 後，可以知道，設定連接埠對應這件事完全可以由應用程式自動化完成了，應用程式可以自己指定它所需的連接埠對應資訊。因此，許多NAT衍生的問題都能透過NAT Traversal解決了。然而目前還未有支援的硬體供測試，因此只能使用現有的軟體來做測試。目前測試目標為讓main server提供UPNP，再使用具有UPNP功能的軟體來測試。

第二節 運作原理



在最高一層，消息在邏輯上僅僅包含關於廠商設備的 UPnP 廠商特定資訊。移至下一層協定棧後，廠商內容由 UPnP 論壇工作委員會定義之資訊提供。來自以上各層的消息儲存在 UPnP 特定協定中。之後，以上消息通過採用簡單服務發現協定（SSDP）、通用事件通知架構（GENA）和簡單物件訪問協定（SOAP）來進行格式化。然後消息通過運行於 UDP 上的多播或單播類 HTTP，或是運行於 TCP 上的標準 HTTP 進行傳輸。最終，以上所有消息均通過 IP 進行傳輸。

UPnP網路互連的基礎是IP定址。每台設備均必須配有動態主機配置協定（DHCP）用戶端，並在設備首次與網路連接時搜索DHCP伺服器。如果DHCP伺服器可以使用，即網路處於管理狀態，則設備必須採用分配給它的IP位址。如果沒有DHCP伺服器可用，即網路處於未管理狀態，則設備必須利

用Auto IP來獲取一個位址。簡言之，Auto IP說明了一台設備如何從一組保留位址中智慧地選出一個IP位址，以及如何能夠在處於管理和未管理狀態的網路間輕鬆切換。如果設備在DHCP取得過程中獲得了一個功能變數名稱（例如通過一台DNS伺服器或通過DNS轉發），則設備應當在後來的網路操作中採用該名稱；否則即應採用其IP位址。

如果取得了一個IP位址，則UPnP網路的第1步是發現。在一個設備添加到網路上之後，UPnP發現協定允許該設備向網路中的控制點宣告其服務。同樣，當一個控制點被添加到網路後，UPnP發現協定允許該控制點在網上搜索定義中的設備。兩種情況下的根本資訊交換均為一個發現消息，包含有關該設備或其服務之一的一些基礎資訊（例如其類型、識別字和指向更詳細資訊的一個指標）。UPnP發現協定基於簡單服務發現協定（SSDP）。以下發現部分說明了設備如何進行宣告，控制點如何搜索，以及有關發現消息格式的詳細資訊。

UPnP網路中的第2步是描述。控制點在發現一個設備之後仍然對其瞭解的資訊太少。為了使控制點瞭解到更多關於設備及其能力的資訊或與設備進行交互，則控制點必須取得來自該設備在發現消息中所提供之URL的設備描述。設備可能包含其他邏輯設備，以及功能單元或服務。對於設備的UPnP描述通過XML來表達，並包括諸如模型名稱和號碼、序列號、製造商名稱和廠商專門網站URL等專門針對廠商的製造商資訊。該描述還包括一列任意的嵌入式設備或服務，以及用於控制、事件觸發和展示的URL。對於每項服務，此描述均包括一列命令或動作，而服務（參數或變數）對於每個動作做出回應；針對服務的描述還包括一列變數；這些變數模型化服務在運行時的狀態，並通過資料類型、範圍和事件特徵進行描述。以下關於描述的部分說明了設備如何被描述，以及這些描述如何被控制點取得。

UPnP網路中的第3步是控制。當一個控制點取得設備描述後，該控制點可將動作發至一個設備的服務。為此，控制點將一條適當的控制消息發至服務的控制URL（在設備描述中提供）。控制消息同樣利用簡單物件訪問協定（SOAP）通過XML來表達。以下關於控制的部分說明了有關動作、狀態變數以及控制消息格式的描述。

UPnP網路的第4步是事件觸發。針對服務的UPnP描述包括一個服務回應的動作列表，以及一個對伺服器運行時狀態進行展示的變數列表。在這些變數變更時服務會發佈更新，一個控制點可以預訂接收此資訊。服務通過發送事件消息來發佈更新。事件消息包含一個或多個狀態變數名和這些變數的當前值。這些消息同樣通過XML來表達，並採用通用事件通知架構（GENA）格式。當控制點首次預定時，會發送一個特殊的初始事件消息；此事件消息包含所有事件變數的名稱和值，並允許訂閱者對服務狀態模式進行初始化。為了支持擁有多個控制點的環境，事件觸發設計用於將任何動作的效果通知所有控制點。因此，所有訂閱者均會收到全部的事件消息。訂閱者收到關於所有已變更事件變數的事件消息，此事件消息無論

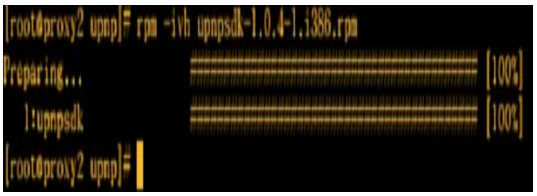
狀態變數為何，改變都會被發送（由於回應一個要求動作，或由於服務建立狀態的變更）。以下關於事件觸發的部分說明了事件消息的預訂和格式。

UPnP網路中的第5步是展示。如果設備有用於展示的URL，那麼控制點就可以通過此URL取得一個頁面，在瀏覽器中載入該頁面，並且根據頁面的功能，支援用戶控制設備和/或瀏覽設備狀態。每一項完成的程度取決於展示頁面和設備的具體功能。

第三節 實作

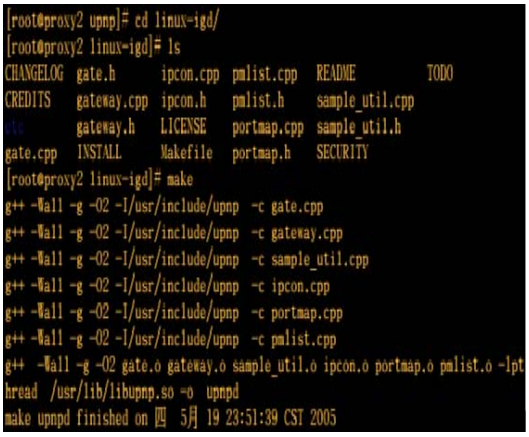
安裝所需要的軟體：NAT Server、linuxigd-0.92、upnpsdk-1.0.4.i386，測試軟體: RaidenFTPD V2.4 build 2166 、FlashFXP v3.0。linuxigd 為 Internet Gateway Device，可用來提供 Gateway 功能，而 upnpsdk 是提供 UPNP 功能，兩個組合起來就形成一台支援 UPNP 的 GATEWAY；RaidenFTPD 是支援 UPNP 協定的 FTP Server，FlashFXP 則是 FTP Client 端。

upnpsdk 安裝，從 http://sourceforge.net/projects/upnp/下載 UPnP SDK。使用指令＃rpm –ivh upnpsdk-1.0.4-1.i386.rpm。



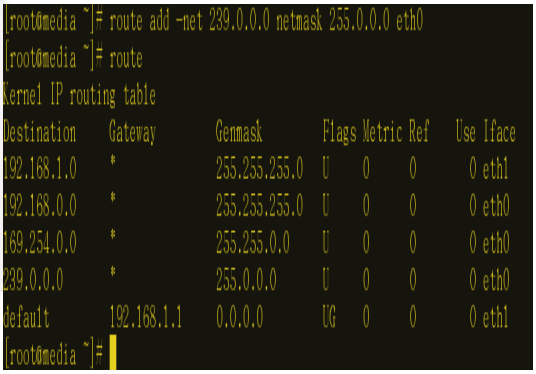
看到 100%即代表安裝成功。

linuxigd 安裝，從 http://sourceforge.net/projects/linux-igd/ 下載 Linux GID 。



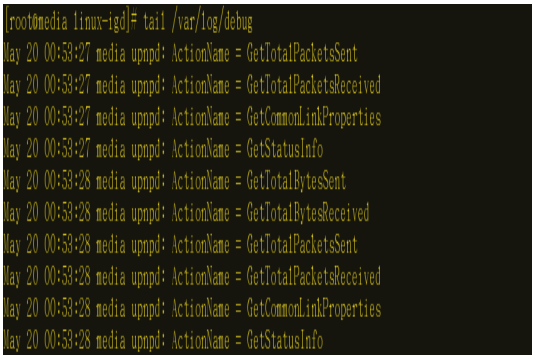
進入 Linux GID 資料夾下，使用 make 指令即可完成。接下來要增加廣播路由。

使用指令# route add -net 239.0.0.0 netmask 255.0.0.0 eth0 。



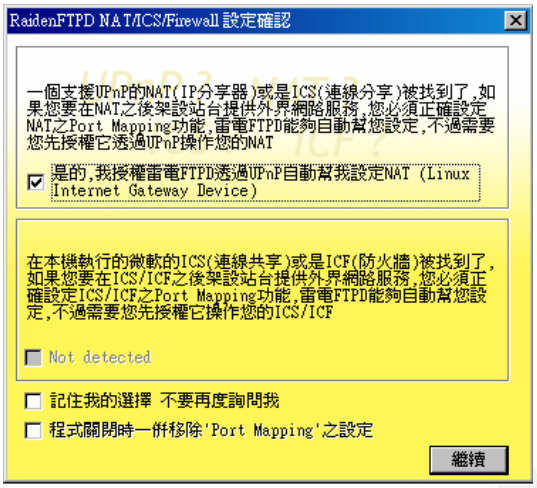
使用 route 指令即可查看剛剛是否有加入新路由。

在設定完畢之後，再來就是啟動 UPNP。使用指令#upnpd eth1 eth0 ，eth1 為 Output interface、eth0 為 Input interface，最後還要檢查 log 檔是否 UPNP 被正確的啟動。#tail /var/log/debug

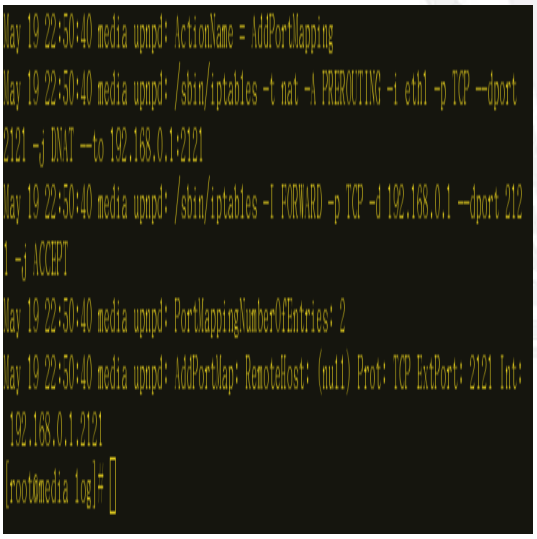


看到 upnp 已經接收到訊息了，那就代表 upnp 正確啟動。然後，用軟體來實測一下，UPNP 的運作狀況。

首先先安裝測試軟體，使用 RaidenFTPD 架設 FTP Server。由圖可看出，此軟體已經偵測到 Linux Internet Gateway Device。

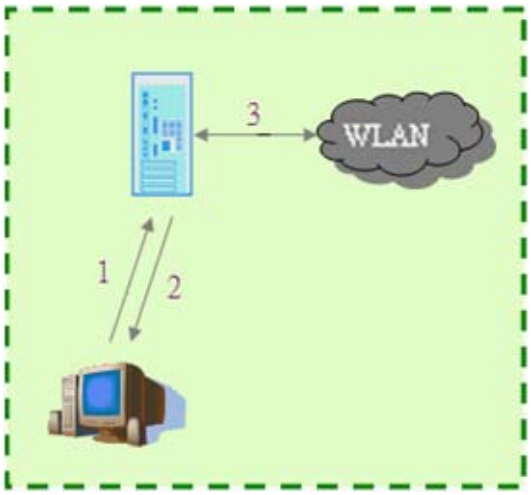


這時候可以檢查一下 IGD 的 log 檔，



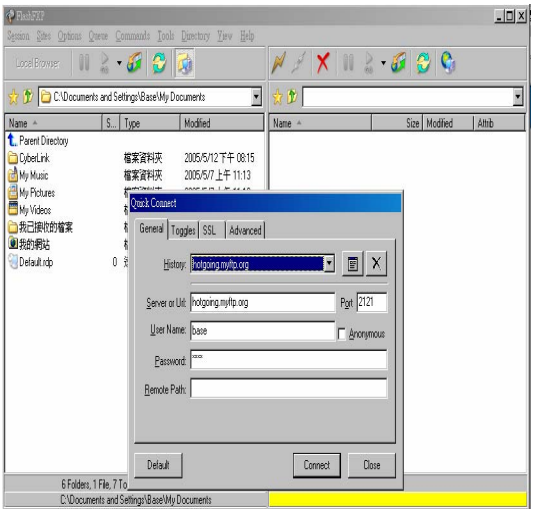
由內容可以發現，它是透過 port mapping 的方式來達成目的，自動把 port：2121 對應給 192.168.0.1。

而其中的構想，大概就如下圖：



第一步，服務伺服器向 IGD 主機發佈消息，訊息內容為包含開啟埠的資訊，當 IGD 主機接收到訊息封包後，使用 iptables 指令，把服務主機的 port 對應到 IGD 主機上；第二步，主機把開啟埠成功的訊息回應給服務主機；第三步，外部網路可以經由 IGD 主機跟服務主機連線。而整個流程就是，當 IGD 主機接收到服務主機發送的封包時，就會透過 iptables 的指令，自動做 port-mapping 的動作；從中發現，以往要手動設定的 port-mapping 就可以交給電腦來處理了，也就完成自動化（Auto）的概念。

再來是使用 FTP Client 來做連線測試，在沒有設定 NAT 對應的 port 的情況下，一般來說這樣是無法連線



的，因為外部網路並不知道內部網路 2121 是要給誰的；但是由於 UPNP 的技術，可以克服這樣缺點，因為有了自動的 port-mapping，2121 port 就會自動的被對應到服務主機上，外部網路也可以順利連上服務主機，不會再出現錯誤訊息，按下連線就如圖所示：

成功的連線，已經進入 FTP 裡的資料夾了。

服務主機上也出現了連線訊息，這樣也代表所架設的 UPNP 系統成功的被支援了。

第六章 監控

我們採用 MRTG 這個主要套件來監控流量，首先，檢查套件是否安裝，我們所需要用到的套件有：perl、gd、libpng、net-snmp、zlib。

#rpm -qa perl；
#rpm -qa gd ;
#rpm -qa libpng ;
#rpm -qa | grep snmp ;
#rpm -qa zlib ;
#rpm -qa mrtg ;

```
[root@media etc]# rpm -qa perl
perl-5.8.5-9
[root@media etc]# rpm -qa gd
gd-2.0.28-1.30.1
[root@media etc]# rpm -qa libpng
libpng-1.2.8-1.fc3
[root@media etc]# rpm -qa snmp
[root@media etc]# rpm -qa | grep snmp
net-snmp-libs-5.1.2-11
net-snmp-5.1.2-11
net-snmp-utils-5.1.2-11
[root@media etc]# rpm -qa zlib
zlib-1.2.1.2-1
[root@media etc]# rpm -qa mrtg
mrtg-2.10.15-1
```

檢查無誤後，開始來做設定動作吧！有支援 snmp 的硬體，就可以很輕易的偵測到該硬體的狀態了，所以一開始先來編輯 snmp。

vi /etc/snmp/snmp.conf

```
access  notConfigGroup ""      any       noauth    exact  mib2 none none

view mib2   included  .iso.org.dod.internet.mgmt.mib-2 fc
```

加入 mib2，MIB(Management Information Base)稱為資訊管理庫，是 SNMP 用來收集各種不同類型網路設備所提供的資訊的資料庫。完成編輯後，啟動 SNMP，# service snmpd start

SNMP 啟動後，接下來是建立 mrtg.cfg。

＃cfgmaker pulic@10.10.1.1 public@10.10.1.37 pulic@10.10.1.38 > /var/www/html/mrtg.cfg

Cfgmaker 是用來建立 mrtg.cfg，而 pulic@10.10.1.1 public@10.10.1.37

pulic@10.10.1.38 是我們所要監控的 IP 流量，並且把這些 log 都產生到 mrtg.cfg。

```
[root@media ~]# cfgmaker public@10.10.1.1 public@10.10.1.37 public@10.10.1.38 >
/var/www/html/mrtg.cfg
--base: Get Device Info on public@10.10.1.1:
--base: Vendor Id:
--base: Populating confcache
--snpo: confcache public@10.10.1.1: Descr lo --> 1
--snpo: confcache public@10.10.1.1: Descr eth0 --> 2
--snpo: confcache public@10.10.1.1: Descr tun10 --> 3
--snpo: confcache public@10.10.1.1: Descr eth1 --> 4
--snpo: confcache public@10.10.1.1: Descr sit0 --> 5
--snpo: confcache public@10.10.1.1: Type 24 --> 1
--snpo: confcache public@10.10.1.1: Type 6 --> 2
--snpo: confcache public@10.10.1.1: Type 131 --> 3
--snpo: confcache public@10.10.1.1: Type 6 --> 4 (duplicate)
--snpo: confcache public@10.10.1.1: Type 131 --> 5 (duplicate)
--snpo: confcache public@10.10.1.1: Ip 10.10.1.1 --> 4
--snpo: confcache public@10.10.1.1: Ip 127.0.0.1 --> 1
--snpo: confcache public@10.10.1.1: Ip 192.168.0.2 --> 2
--snpo: confcache public@10.10.1.1: Eth  --> 1
--snpo: confcache public@10.10.1.1: Eth 00-03-9d-73-33-d8 --> 2
--snpo: confcache public@10.10.1.1: Eth 00-00-00-00-33-d8 --> 3
--snpo: confcache public@10.10.1.1: Eth 00-02-2d-1b-5e-81 --> 4
--snpo: confcache public@10.10.1.1: Eth 00-00-00-00-5e-81 --> 5
--base: Get Interface Info
```

如圖可以看到，系統正在擷取所要監控主機的 info，這麼一來就可以清楚的知道各主機的流量了。然後，要來修改 mrtg.cfg 的一些內容，把語言改成中文繁

```
Options[]: growright, bits
HtmlDir: /var/www/mrtg
ImageDir: /var/www/mrtg
LogDir: /var/lib/mrtg
ThreshDir: /var/lib/mrtg
Language:big5
```

體，也把 Options 的部分改成”growright，bits”，主要目的在於讓顯示流量方向為由右往左，並且以 bits 來計算流量。

再來是建立 mrtg 主網頁。

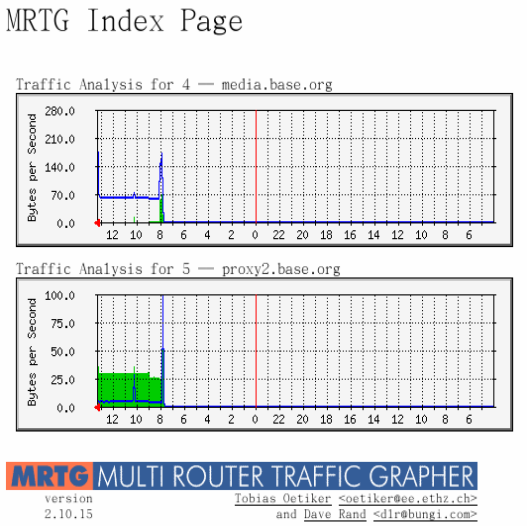
＃indexmaker /var/www/html/mrtg.cfg > /var/www/html/mrtg/index.html

加入系統的排程，讓系統每隔一段時間就可以自動產生 mrtg 主網頁。

＃crontab –u root –e

```
*/5 * * * * env LANG=C /usr/bin/mrtg /var/www/html/mrtg.cfg
```

此寫法是讓系統每隔五分鐘，就會去執行＃/usrbin/mrtg /var/www/html/mrtg.cfg 這項指令，這也代表主機每五分鐘就會去偵測流量一次。



完成設定之後，來測試主機是否能偵測到流量。如我們所願，運作正常。

另外，MRTG 也可以讓我們偵測 CPU 負載、偵測線上人數。

- **偵測 CPU 負載：**

  第一步先建立 mrtg.cpu。

＃vi mrtg.cpu

第二步就是要建立 mrtg.cfg.cpu。

＃vi mrtg.cfg.cpu

第三步就是執行 mrtg.cfg.cpu。

#/usr/bin/mrtg /var/www/html/mrtg/ mrtg.cfg.cpu

第四步就是加入系統排程。

＃crontab –u root –e

如此就完成 CPU 偵測的設定了。

- **偵測線上人數：**

第一步先建立 mrtg.man。#vi mrtg.man

第二步修改 mrtg.cfg. man。#vi mrtg.cfg.man

第三步執行 mrtg.cfg.man 。

#/usr/bin/mrtg /var/www/html/mrtg/mrtg.cfg.man 第四步加入系統排成程。

＃crontab –u root –e

如此一來監控也完成了，可以在主機上去監控其他的主機設備了，也可以即時知道哪台主機有異常，能夠在短時間內做出反應。

第七章 展示網頁

第一節 綱要

接下來規劃的是網頁，目前網頁所要呈現出來的就是隨選視訊，因此，除了簡易的登入頁面，還要有資源展示頁面和多媒體播放頁面；有了以上的目標，所要設計的頁面總共有四個，分別為 index.php（主網頁）、getpawd.php（登入）、main.php（資源展示頁）、mediaplayer.php（多媒體播放頁）；而此網站登入的流程為，登入主網頁，輸入密碼並且登入到資源展示頁面，選擇所要播放的多媒體，之

後在多媒體播放頁面播放，這樣就完成展示網頁的簡易設計了。

第二節 網頁編寫

```
#index.php

<body>

<img src="1.png" width="827" height="652">
<h5 align="center">
  <form name="form1" method="post" action="getpawd.php">
<font color="#6F83AA"><span class="style2" style="font-size:14pt;">管理者密碼</span>
</font>:
  <input name="pawd" type="password" size="30" maxlength="12">
  <input type="submit" name="Submit" value="送出">
  </form>
</h5>
</body>
```

#index.php 主要是登入第一個會看到的頁面，並且也是輸入登入密碼的所在；這裡使用 form 表單，並使用 post 模式來傳送密碼，把輸入的 key word 定義為密碼(password)模式，命名為 pawd，以便在密碼處理的頁面能夠使用這些傳過來的資訊。上面所說的 POST 就是外部傳遞變數；一般 Server 端要取得 Client 端所輸入的資料都是透過 HTML From 表單來傳送，而傳送的方式（method）有 Get、Post 二種。在使用上，只需要將 HTML Form 表單中的 method 屬性等於 get 或是 post。利用 Get 方式傳送資料時，是將資料直接加在 URL 後；而 Post 方式傳送則是先將資料轉成標準輸入，然後再傳送。這樣就可以知道為什麼這裡的表單要使用 POST，就是避免密碼讓別人一看就清楚了然。而這邊也有一個小細節要注意，就是一開始安裝的 PHP，裡頭有一個設定檔 php.ini，這裡面有個參數 register_globals = Off 要改成 On，否則變數是無法在網頁間傳遞的。

```
#getpawd.php

<body>
<?

$newpw = 12345;

if ( $newpw == $pawd){

echo "<script>window.location ='main.php';</script>";

}
else {

echo "you type wrong";

echo "<META HTTP-EQUIV=\"Refresh\" CONTENT=\"3; URL=index.php\">";
}

?>
</body>
```

首先，從主頁面傳過來的資訊，透過變數，導引到密碼處理頁面；而上面的程式碼，一開始先設定密碼為 12345（這是簡易的作法，如果要多人登入系統，就還要配合資料庫並且重新設計頁面），等待從主頁面傳送過來的資料，然後再進行比對，假使輸入的密碼跟所設定的密碼相符合的話，就會開啟資源展示的頁面；假使是輸入錯誤的密碼，那麼頁面上就會顯示出”you type wrong”的訊息，並且在經過 3 秒後重新整理頁面，把頁面轉向主頁面。

#main.php

```
<body>
<form name="form1" method="post" action="mediaplayer.php">
  <p><label>  <img src="tile_blue_1.gif" width="88" height="69" border="0"></p>
  <p><input type="radio" name="mediafile" value="final">  Final</label>
  <br>  <label>  <input type="radio" name="mediafile" value="final2">  Final2</label>
  <br><label>  <input type="radio" name="mediafile" value="final3">  Final3</label>
  <br><label>  <input type="radio" name="mediafile" value="final4">  Final4</label>
  <br> </p>
 <input type="submit" name="Submit" value="確定">
</form>


</body>
```

接下來要設計的是資源展示網頁。資源展示網頁，主要就是網站擁有者想要呈現什麼樣的資源，都可以放在這個頁面，而且只要透過簡單的排版和 HTML 語言的編寫，就可以動態的讓播放多媒體的頁面載入，而不需要繁複的輸入多媒體資源的來源位址；由 HTML 語言中可以看到，這是用很簡單的 form 來完成的，只要替每個檔案定義名稱，加入 form 的選項中，就可以讓使用者選取並且播放。

#mediaplayer.php

```
<body background="back3_2_2_1.gif">
<p align="center" class="style1">
<b><font face="王漢宗中隸書繁" color="#CC0000"><span class="style1">現在播放的是
</span></font></b> : <? echo "$mediafile" ;?>
<p align="center">

<!-- <embed src=\"$media\" width="400" height="320" autostart="true" >-->
<EMBED  width=360 height=270 type=video/x-msvideo loop="ture" autostart="true"
src= "<?echo "$mediafile.mpg"; ?>">
</embed>

</body>
```

先來看看第三行，<?echo $mediafile";?>這行是用來顯示目前播放的檔案名稱，使用方法是透過變數名稱，也就是之前在資源展示網頁所點選的檔案名稱，這樣就可以很清楚的告訴使用者，目前所播放的檔案是屬於哪一個；再來是第七行，這是定義播放程式的語法，使用的是微軟的 mediaplayer 插件，可以自行定義一些參數，以下則是一些參數的介紹。

embed 是個很簡便的語法，可以用來播放影音檔，通常它是用來播放 Windows Media Player 支援的格式，但也可用來播放一些其他格式，只是要注意更多細節，本文最後再略微提一下這些其他格式，主要還是針對播放 Media Player 支援的格式來討論。（如 WMA、WMV、ASF、MPG、AVI）

<embed src=檔案位址>上面這便是 embed 最簡單、最簡潔的寫法，只要這樣寫便可播放影音檔了，以下再開始討論各注意事項及設定參數。

關於結束標籤：

通常語法標籤都是成對的，有開始就有結束，有〈b〉就有〈/b〉，有〈font〉就有〈/font〉。可是 embed 語法可以不用寫〈/embed〉，完全不會有任何影響，又省事。

關於尺寸：

如上的最簡潔寫法，尺寸可以完全不設定，這是最理想的。播放音樂檔時，會自動呈現一條完整的播放 Bar，如圖所示（Windows Media Player 版本若不同播放 Bar 外觀也可能不

同）。播放影片時（WMV、ASF、MPG、AVI），除了播放 Bar，還會以影片的原始尺寸播放出畫面，畫面大小會自動調整。

如果因各種因素而想設定尺寸，例如想刻意拉大縮小影片的畫面，或是想改變播放 Bar 的大小，只要加進尺寸參數 width=寬度 height=高度 即可，範例如下：

<embed src=檔案位址 width=寬度 height=高度>

無論設定任何尺寸，都是包含播放 Bar 也算在內的，試一下便可瞭解。

關於自動播放：

如上的最簡潔寫法，其預設就是會自動播放。如果不想自動播放，加入參數 autostart=false 即可，範例如下：

<embed src=檔案位址 autostart=false>

關於循環播放：

如上的最簡潔寫法，其預設就是不會循環播放的，就只播放一次。如果要循環播放，加入參數 loop=true 即可，範例如下：

<embed src=檔案位址 loop=true>

關於 Tracker：

如果加進這句參數 ShowTracker=false，播放 Bar 就會變成像下圖那樣，應該有看出少了什麼東西吧，就是會少了播放的進度 Bar。語法範例如下：

<embed src=檔案位址 ShowTracker=false>

關於 Position Controls：

只要加入這句參數 ShowPositionControls=false，播放 Bar 就會變成如下圖那樣，看出少了什麼東西了嗎？就是一些控制鈕都被隱藏起來了語法範例如下：

<embed src=檔案位址 ShowPositionControls=false>

關於 Audio Controls：

就是有關於音量控制的參數，只要加入這句參數

ShowAudioControls=false，控制音量的 tracker 跟靜音鈕都會消失。語法範例如下：

<embed src=檔案位址 ShowAudioControls=false>

關於預設的音量大小：

在語法裡面如果沒有寫任何關於音量大小的設定，播放時預設的音量大小如下上圖，大約是 50%的音量。如果希望一開始播放就以 100%的最大音量播放，可以加入這句參數 Volume=0，請注意，是等於「零」，不是英文字母的大寫 O。播放時的情況如下下圖，語法範例如下：

<embed src=檔案位址 Volume=0>

關於資訊視窗（part 1）：

只要加進這句參數 ShowStatusBar=true，播放 Bar 下方便會多出一行資訊視窗，如下圖。語法範例如下：

<embed src=檔案位址 ShowStatusBar=true>

這個參數是非常實用的，因為這行資訊視窗會秀出很多有用資訊如下載進度、播放進度、曲名、藝人名稱.....等等。秀出下載進度、播放進度尤其體貼，可以讓瀏覽者大概掌握到底要等多久才會開始播放。建議用 embed 語法播放影音檔時最好都加上這句參數。

關於資訊視窗（part 2）：

還有另外一種秀出資訊視窗的參數 ShowDisplay=true，會一口氣秀出四行資訊，每行各秀出一種資訊，如上圖。語法範例如下：

<embed src=檔案位址 ShowDisplay=true>

這個參數就比較無所謂了！因為這些

資訊，只要用上一個參數 ShowStatusBar=true 便可全部呈現出來，而且是單行資訊，但是這裡用交替出現的方式呈現所有資訊。有點顯得這個 ShowDisplay=true 參數占空間，一行顯示一種資訊，而最有用的下載進度、播放進度等卻不會顯示，跟 ShowStatusBar=true 參數比起來較不當！

關於防右鍵：

可以試試在播放 Bar 上按右鍵看內容，便可看到檔案的真實位址！或是在播放影片時也可以在畫面上按右鍵看內容，也可看到檔案真實位址。加上這句參數 EnableContextMenu=false 便可防止在播放 Bar 或影片畫面上按右鍵。語法範例如下：

<embed src=檔案位址 EnableContextMenu=false>

可是這語法有一點沒有什麼用，因為光是在播放面版上防堵意義不大，還要搭配整個頁面的防堵才能收到效果。

關於隱藏面版：

想要隱藏面版，只要加上這句參數 hidden=true，整個播放面版就會不見，什麼都看不到，但還是會播放歌曲（只要沒有取消自動播放）。語法範例如下：

<embed src=檔案位址 hidden=true>

什麼東西都沒有，卻又會播放歌曲，就跟最傳統的背景音樂語法〈bgsound src=檔案位址〉很相似！embed 加上 hidden=true 後，看不到播放面版，也不會影響版面（完全不佔位置），表面上的確和 bgsound 語法一模一樣，兩者可以互相取代。但在功能上，兩者還是有差別的！

用 bgsound 語法播放歌曲沒有串流功能，即使播 WMA 也沒有，一定要檔案完全下載完畢才會開始播放。embed 加 hidden=true 就有串流功

能。有串流就比較好嗎？不見得！是會比較快開始播放歌曲，但如果網路繁忙或連線速度慢，播放就會斷斷續續的。bgsound 語法要整首下載完才播放，等比較久才聽的到，但開始播後就保證不會斷斷續續的。所以，現在頁面要播背景音樂有兩種選擇了，就看要不要串流功能。

播放清單檔：

embed 並非只能直接播放影音檔，它也可以播放清單檔如 M3U、ASX、WPL 等，這樣就可以用一句 embed 語法播放多首歌曲。語法範例如下：

<embed src=檔名.m3u>

當然，想這樣運用的話，就得多費心思製作清單檔，影音檔跟清單檔上傳時也要注意其相對位置都要擺正確。並非所有電腦環境都支援播放各式清單檔，如果 Windows Media Player 版本不夠新，可能就無法播放一些清單檔。還有如果電腦預設使用其他軟體來播放清單檔（非 Media Player)，也可能無法支援用 embed 播放清單檔，得將檔案格式連結回 Media Player 才行。播放清單檔功能如果搭配上述的隱藏面版功能，就什麼都沒看到，卻會自動播放多首背景音樂。

播放 MP3：

用 embed 可以播放 MP3，但有一點要注意，通常大家電腦裡不見得是預設用 Windows Media Player 來播放 MP3，很多人是預設 Foobar、WinAmp...等等，總之只要不是預設用 Media Player，embed 在播放 MP3 時，便會呼叫別的軟體來播放，如 QuickTime 或 Real Player，而這樣播放面版都會很小，變成一定要設尺寸才可以。只有電腦裡預設用 Media Player 播放 MP3，embed 才會呼叫 Media Player 來播放 MP3，也才會是那個熟悉的 Media Player style 的播放面版，各種設定也較為方便。語法範例如下：

<embed src=檔名.mp3>

播放 Flash 檔案 SWF：

embed 可以播放 SWF，事實上 SWF 好像一定要用 embed 才能播放出來。要注意播放時一定要設定尺寸，否則畫面會變成 200*200。語法範例如下：

<embed src=檔名.swf width=寬度 height=高度>

播放 SWF 不會出現任何播放 BAR（除非 SWF 檔案本身有做一個），因為播放 Flash 並非呼叫 Media Player 來播放，而是直接用 IE 播放。

播放 Quicktime 影片檔案 MOV 及 QT：

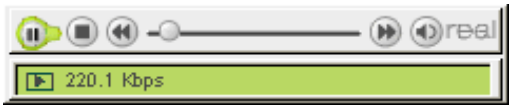
embed 可以播放 MOV、QT 等影片檔，但是電腦裡必須有 QuickTime Player，而且版本要夠新，最好是 6.3 以上，這樣才能看到用 embed 播的 MOV 或 QT 檔。另外，播放時一定要設定尺寸，否則畫面會變很小。語法範例如下：

<embed src=檔名.mov width=寬度 height=高度>

播放 MOV 或 QT 會有播放 Bar，是 QuickTime style 的播放 Bar，尺寸設定亦是將播放 Bar 算在內的，多嘗試幾次便能掌握合適的尺寸。

播放 Real 檔案：

先說明播放 Real 音樂檔。第一，用 embed 播放 Real 音樂檔一定要寫尺寸大小，否則播放面版會很小，甚至看不到。第二，關於「自動播放」若什麼都不寫不去設定，預設是「不會自動播放」，若要自動播放就要加入這句參數 autostart=true。圖示如下，整



個綜合起來語法範例如下：

<embed src=檔名.ram autostart=true width=寬度 height=高度>

還有一個秀出播放面版的語法 controls=controlpanel，若加這句進去，面版會變成像下圖這樣。這個參數有點多餘，多加一句話，秀出的東西卻比上面那個圖示還少，如果就是想讓面版簡潔些，就可用這句語法，高度可以設小一點啦，不需像上圖這麼高。語法範例如下：

<embed src=檔名.ram autostart=true width=寬度 height=高度 controls=controlpanel>

再來說明播放 Real 影片。若用上

述的語法播 Real 影片，只能聽到聲音，看不到畫面，要加上這句參數 controls=ImageWindow 才能看到畫面。當然，尺寸還是要寫，關於自動播放的設定也同上述。語法範例如下：

＜embed src=檔名.ram autostart=true width=寬度 height=高度 controls=ImageWindow＞

用這語法播放影片，畫面先是 Real Player 的 Mark，下方會顯示檔案下載

進度，開始播放後就只有畫面，沒有任何控制面版，整個情況如上圖所示。所以，用這語法播放 Real 影片一定要設為自動播放，否則沒有播放面版根本無法控制。

想要播放出畫面又要有控制面版，有個特殊寫法可以辦的到。原理簡單來說是將整組 embed 語法寫兩次，兩組語法用〈br〉隔開。上面那組語法加入 controls=ImageWindow 參數播出畫面，沒有控制面版。下面那組語法不要多加參數（或是加入 controls=controlpanel），就只有控制面版，看不到畫面。然後兩組語法都要加入一個重要參數 console=_master，兩組語法的播放才會同步並互相關連，操作下面那組語法的面版亦可控制上面那組語法的播放。語法範例如下：

＜embed src=檔名.ram autostart=true width=影片畫面寬度 height=影片畫面高度 controls=ImageWindow console=_master＞

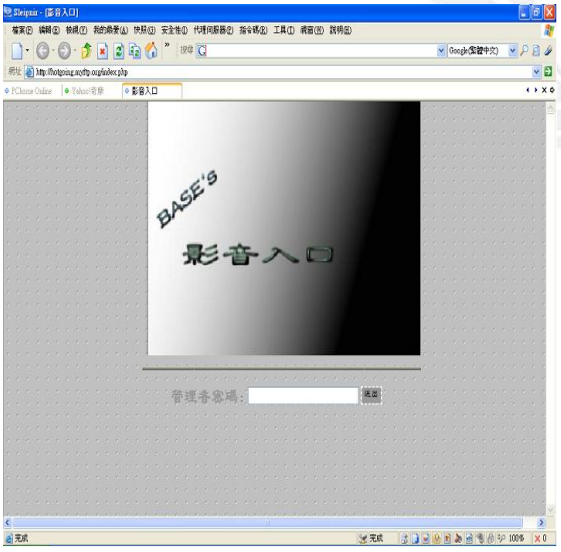
〈br〉

＜embed src=檔名.ram autostart=true width=播放面版寬度 height=播放面版高度 controls=controlpanel console=_master＞

上面這範例，兩組 embed 語法以及〈br〉之間都有換行讓它呈現的更清楚，自行編寫的時候請把它們通通連起來成為一行很長的語法，這樣呈現出來的才會如下圖，畫面下面緊接著是控制面版。如果像上面範例這樣有換行，畫面跟控制面版之間會有空行間隔出現。

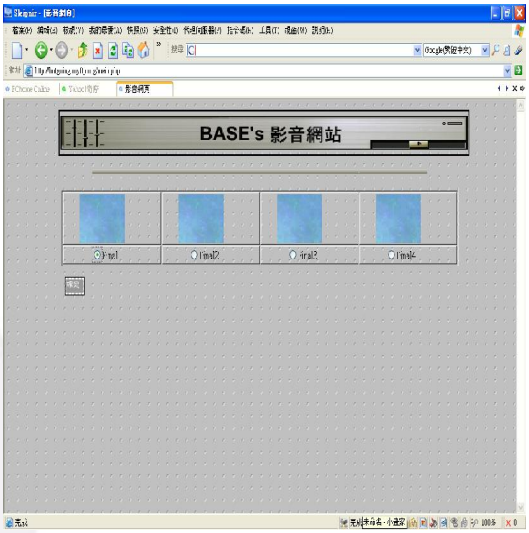
第三節 測試

在一切準備就緒下，現在要來看看最後的展示網頁，進入剛剛設計的登入頁面。

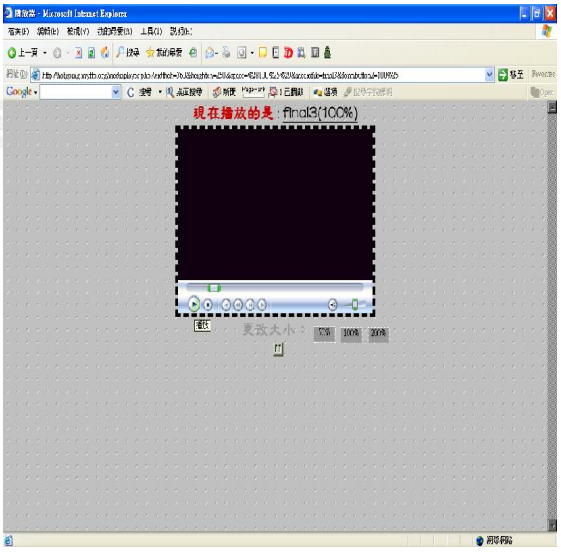


然後把管理者密碼輸入後，就可以進入正式的頁面。此時，輸入的密碼會對應成 form 裡的密碼變數，並且會透過 post 的模式，把變數傳送出去，交由密碼處裡的頁面來作處理。

當密碼處理頁面接收到登入頁面傳來的變數後，會開始比對輸入的密碼和預設的密碼，是否有相吻合，假使有才會讓使用者通行，進入下一個頁面；假使沒有，就會重新回到登入的頁面。



進入資源展示頁面後，可以依自己喜好，選擇想要觀看的影片，在檔名旁邊的選擇鈕點選後，按下確定，就會自動進入播放頁面。



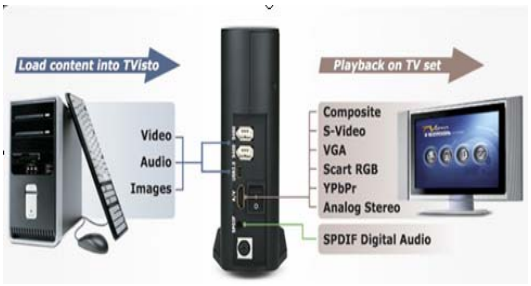
進入播放頁面後，可以看到，在播放器上面有播放的檔名，這個檔名的顯示模式是自動的，是依據所選的檔案名稱，透過變數傳遞，把檔明顯現出

來，再來就是播放的 mediaplayer，在播放器下方，有可以調整視訊框大小的按鈕，可以依據網路速度來選擇播放視訊的大小。因此，由上可以知道，目前播放頁面主要規劃的功能有，顯示播放檔案名稱、調整播放內容大小而已，當然還可以依照需求，加入許多的功能。

第八章 未來願景

在幾乎每個家庭都有電視、音響、影音播放機……等等的情況下，假使有那麼一台能夠控制並且管理這些資源的產品，那麼我想，家庭的娛樂功能勢必會再提升；想想看，未來硬體設備的價格持續下降，在每個來源上加裝一台機器當作 SERVER，不再是大成本的花費時；透過這一台主機來控制這些資源，並且在無線網路技術的提升下，速度及穿透率的增加，使主機涵蓋範圍達到整個家庭，那麼，家裡的每個人，都可以在各個角落，接收到完整且穩定的資源，甚至還可以透過 PDA 或是手機，直接連上無線網路，不管走道哪都可以享受娛樂。然而，把整個架構放大的話，並且運用到更上層的網路供應商的話，也就可以變成目前熱門的 MOD。

下面這裡介紹的幾張圖，是最近在市面上的一個產品，它的特點跟這次所做的專題有些類似，除了在網路部分不支援和架構上有些微差距外，其構想卻蠻相近的。從下面的架構圖可以知道，它的來源主要是經由電腦傳輸到主機的硬碟，然後再跟電視連接；電視可以直接取用主機內預設的選單服務，透過選單的設定，並且選取所要的資源，即可及時播放多媒體。



此產品的名稱為「多媒體家庭劇院」，主要的功能就是提供家庭影音服務，假使在這產品上加上無線網路的部分，不就可以分享給家庭裡的每個角落。

- 主目錄

- 影片瀏覽模式

- 圖片瀏覽模式

- 影像來源格式設定

- 音訊檔播放模式

由這些圖可以看出來，這產品支援了影像、MP3、圖片、電視訊號，如此的多功能，是不是對家庭的娛樂有著很大的提升呢！

參考文獻

1. Linux 系統管理實物-自動化、備援、安全、叢集，施威銘研究室著
2. 鳥哥的 Linux 私房菜-基礎學習篇（增訂版），鳥哥著
3. 鳥哥的 Linux 私房菜-伺服器架設篇，鳥哥著
4. Linux 網路實作經典，高健智、賴阿福著
5. http://linux.tnc.edu.tw/techdoc/perl_intro/book1.html，The linux-wlan[tm] Company
6. http://www.linux-wlan.org/docs/wlan_adapters.html.gz，WLAN Adapter Chipset Directory
7. UPnP:自動化網路設定，蔡孟甫、曹世強、林盈達
8. http://vc3.ath.cx/~harvey/view.php/page/doc，Harvey's WebCabinet
9. http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/，Wireless LAN resources for Linux
10. http://hostap.epitest.fi/，Host AP driver
11. http://www.gentoo.org/doc/zh_tw/handbook/handbook-x86.xml?part=4&chap=4，wireless-tools 設定
12. http://www-128.ibm.com/developerworks/tw/library/wi-run.html，IBM(多種幫助開始使用無線 Linux 的工具和專案)
13. http://www.gentoo.org/doc/zh_tw/handbook/handbook-x86.xml?part=4&chap=4，無線網路
14. http://www.jollen.org/linux/video_streaming/streaming-1.html，Jollen

網路學院，Linux 的應用－ Video Streaming 探討

15. http://www.net-snmp.org/，Net-SNMP
16. http://www.newport-networks.com/whitepapers/nat-traversal.html，NAT traversal
17. http://www.linuxvirtualserver.org/，LVS （Linux Virtual Server）
18. http://www.upnp.org/，UPNP
19. http://linux.tnc.edu.tw/techdoc/perl_intro/book1.html，網路管理語言 Perl 入門與實作
20. PHP&MySQL 完全架站攻略，Luke Welling, Laura Thomson 著
21. http://www.newport-networks.com/whitepapers/nat-traversal.html，NAT Traversal for Multimedia over IP Services - White Paper
22. http://linux-igd.sourceforge.net/，LINUX UPNP INTERNET GATEWAY DEVICE

參考資源

1. http://fr2.rpmfind.net/linux/rpm2html/search.php?query=kernel-ntfs，RPM resource kernel-ntfs
2. http://rpm.pbone.net/，RPM Search
3. http://fr2.rpmfind.net/ , RPM Find
4. http://www.appservnetwork.com/?modules=&applang=tw，AppServ （Windows Server）
5. http://etds.ncl.edu.tw/theabs/index.jsp，全國博碩士論文資訊網
6. http://www.oldlinux.org/cgi-bin/LB5000XP/leoboard.cgi，OldLinux
7. http://twpug.net，台灣 PHP 聯盟
8. http://www.zyxel.dk/Produkter.32+B6JnR4X1p5WEVMcHJvZHVjdHN__fcGkxW3Nob3dVaWRdPTgyJmNIYXNoPWE3NmUyZGVjMzc_.0.html，Linux ZyDAS USB Driver
9. http://www.anywlan.com/，Anywhere WLAN