5.1 網路參數設定使用的指令



任何時刻如果你想要做好你的網路參數設定，就得要瞭解底下這些相關的指令才行！

* ifconfig：查詢、設定網路卡與 IP 網域等相關參數；
* route：查詢、設定路由表 (route table)

5.1.1 手動/自動設定與啟動/關閉 IP 參數： ifconfig, ifup, ifdown



這三個指令的用途都是在啟動網路介面，底下我們就分別來談一談！



ifconfig

ifconfig主要是可以手動的啟動、觀察與修改網路介面的相關參數，包括IP參數以及MTU等等都可以修改，他的語法如下：

|  |
| --- |
| [root@www ~]# ifconfig {interface} {up|down} <== 觀察與啟動介面  [root@www ~]# ifconfig interface {options} <== 設定與修改介面  選項與參數：  interface：網路卡介面代號，包括 eth0, eth1, ppp0 等等  options ：可以接的參數，包括如下：  up, down ：啟動 (up) 或關閉 (down) 該網路介面(不涉及任何參數)  mtu ：可以設定不同的 MTU 數值，例如 mtu 1500 (單位為 byte)  netmask ：就是子遮罩網路；  broadcast：就是廣播位址啊！  # 範例一：觀察所有的網路介面(直接輸入 ifconfig)  [root@www ~]# ifconfig  eth0 Link encap:Ethernet HWaddr 08:00:27:71:85:BD  inet addr:192.168.1.100 Bcast:192.168.1.255 Mask:255.255.255.0  inet6 addr: fe80::a00:27ff:fe71:85bd/64 Scope:Link  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  RX packets:2555 errors:0 dropped:0 overruns:0 frame:0  TX packets:70 errors:0 dropped:0 overruns:0 carrier:0  collisions:0 txqueuelen:1000  RX bytes:239892 (234.2 KiB) TX bytes:11153 (10.8 KiB) |

直接輸入ifconfig會列出目前已經被啟動的卡，不論這個卡是否有給予IP，都會被顯示出來。如果是輸入ifconfig eth0，則僅會秀出這張介面的相關資料，不管該介面是否有啟動。所以如想知道某張網路卡的Hardware Address，直接輸入『ifconfig "網路介面代號"』即可！至於上表出現的各項資料是這樣的(資料排列由上而下、由左而右)：

* eth0：就是網路卡的代號，也有 lo 這個 loopback；
* HWaddr：就是網路卡的硬體位址，俗稱的MAC；
* inet addr：IPv4的IP位址，後續的 Bcast, Mask 分別代表的是 Broadcast 與 netmask 喔！
* inet6 addr：是IPv6的版本的 IP，我們沒有使用，所以略過；
* MTU：就是第二章談到的[MTU](http://linux.vbird.org/linux_server/0110network_basic.php#tcpip_link_mtu)啊！
* RX：表網路由啟動到目前為止的封包接收情況，packets表封包數、errors表封包發生錯誤的數量、dropped 表封包由於有問題而遭丟棄的數量等等；
* TX：與RX相反，為網路由啟動到目前為止的傳送情況；
* collisions：表封包碰撞的情況，如果發生太多次，表網路狀況不太好；
* txqueuelen：代表用來傳輸資料的緩衝區的儲存長度；
* RX bytes, TX bytes：總接收、傳送的位元組總量

大致上可瞭解到網路情況，尤其是RX, TX內的error數量，以及是否發生嚴重的collision情況，都是需要注意的喔！

|  |
| --- |
| # 範例二：暫時修改網路介面，給予 eth0 一個 192.168.100.100/24 的參數  [root@www ~]# ifconfig eth0 192.168.100.100  # 如果不加任何其他參數，則系統會依照該 IP 所在的 class 範圍，自動的計算出  # netmask 以及 network, broadcast 等 IP 參數，若想改其他參數則：  [root@www ~]# ifconfig eth0 192.168.100.100 \  > netmask 255.255.255.128 mtu 8000  # 設定不同參數的網路介面，同時設定 MTU 的數值！  [root@www ~]# ifconfig eth0 mtu 9000  # 僅修改該介面的 MTU 數值，其他的保持不動！  [root@www ~]# ifconfig eth0:0 192.168.50.50  # 介面eth0:0是在該實體網卡上，再模擬一個網路介面，即在一張網路卡上設定多個IP！  [root@www ~]# ifconfig  eth0 Link encap:Ethernet HWaddr 08:00:27:71:85:BD  inet addr:192.168.100.100 Bcast:192.168.100.127 Mask:255.255.255.128  inet6 addr: fe80::a00:27ff:fe71:85bd/64 Scope:Link  UP BROADCAST RUNNING MULTICAST MTU:9000 Metric:1  RX packets:2555 errors:0 dropped:0 overruns:0 frame:0  TX packets:70 errors:0 dropped:0 overruns:0 carrier:0  collisions:0 txqueuelen:1000  RX bytes:239892 (234.2 KiB) TX bytes:11153 (10.8 KiB)  eth0:0 Link encap:Ethernet HWaddr 08:00:27:71:85:BD  inet addr:192.168.50.50 Bcast:192.168.50.255 Mask:255.255.255.0  UP BROADCAST RUNNING MULTICAST MTU:9000 Metric:1  # 仔細看，是否與硬體有關的資訊都相同啊！沒錯！因為是同一張網卡嘛！  # 那如果想要將剛剛建立的那張 eth0:0 關閉就好，不影響原有的 eth0 呢？  [root@www ~]# ifconfig eth0:0 down  # 關掉 eth0:0 這個介面。那如果想用預設值啟動 eth1：『ifconfig eth1 up』即可達成  # 範例三：將手動的處理全部取消，使用原有的設定值重建網路參數：  [root@www ~]# /etc/init.d/network restart  # 剛剛設定的資料全部失效，會以 ifcfg-ethX 的設定為主！ |

ifconfig可暫時手動來設定或修改某個介面卡的相關功能！設定錯誤也不打緊，因為可利用/etc/init.d/network restart重啟整個網路介面，那之前手動的設定資料會全部都失效！要啟動某個網路介面，但又不讓他具有IP參數時，直接給他ifconfig eth0 up即可！這個動作經常在無線網卡當中會進行，因為我們必須要啟動無線網卡讓他去偵測 AP存在與否！

5.1.2 路由修改： route



兩部主機之間一定要有路由才能夠互通TCP/IP的協定，否則就無法進行連線！一般來說，只要有網路介面，該介面就會產生一個路由，所以我們安裝的主機有一個eth0的介面，看起來就會是這樣：

|  |
| --- |
| [root@www ~]# route [-nee]  [root@www ~]# route add [-net|-host] [網域或主機] netmask [mask] [gw|dev]  [root@www ~]# route del [-net|-host] [網域或主機] netmask [mask] [gw|dev]  觀察的參數：  -n ：不要使用通訊協定或主機名稱，直接使用 IP 或 port number；  -ee ：使用更詳細的資訊來顯示  增加 (add) 與刪除 (del) 路由的相關參數：  -net ：表示後面接的路由為一個網域；  -host ：表示後面接的為連接到單部主機的路由；  netmask ：與網域有關，可以設定 netmask 決定網域的大小；  gw ：gateway 的簡寫，後續接的是 IP 的數值喔，與 dev 不同；  dev ：如果只是要指定由那一塊網路卡連線出去，則使用這個設定，後面接 eth0 等  # 範例一：單純的觀察路由狀態  [root@www ~]# route -n  Kernel IP routing table  Destination Gateway Genmask Flags Metric Ref Use Iface  192.168.1.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0  169.254.0.0 0.0.0.0 255.255.0.0 U 1002 0 0 eth0  0.0.0.0 192.168.1.254 0.0.0.0 UG 0 0 0 eth0  [root@www ~]# route  Kernel IP routing table  Destination Gateway Genmask Flags Metric Ref Use Iface  192.168.1.0 \* 255.255.255.0 U 0 0 0 eth0  link-local \* 255.255.0.0 U 1002 0 0 eth0  default 192.168.1.254 0.0.0.0 UG 0 0 0 eth0 |

觀察route與route -n的輸出可發現加-n參數的主要是顯示出IP，至於使用route顯示的則是『主機名稱』！在預設的情況下，route會去找出該IP的主機名稱，如果找不到，會顯示的鈍鈍的，所以通常直接使用route -n！

* Destination, Genmask：這就是分別是network與netmask！這兩個就組合成為一個完整的網域！
* Gateway：該網域是通過哪個gateway連接出去的？如果顯示0.0.0.0表示該路由是直接由本機傳送，亦即可以透過區域網路的MAC直接傳訊；如果有顯示IP，表示該路由需要經過路由器(通訊閘)才能夠傳送出去。
* Flags：總共有多個旗標，代表的意義如下：
  + U (route is up)：該路由是啟動的；
  + H (target is a host)：目標是一部主機 (IP) 而非網域；
  + G (use gateway)：需要透過外部的主機 (gateway) 來轉遞封包；
  + R (reinstate route for dynamic routing)：使用動態路由時，恢復路由資訊的旗標；
  + D (dynamically installed by daemon or redirect)：已經由服務或轉 port 功能設定為動態路由
  + M (modified from routing daemon or redirect)：路由已經被修改了；
  + ! (reject route)：這個路由將不會被接受(用來抵擋不安全的網域！)
* Iface：這個路由傳遞封包的介面。

觀察上面的路由排列順序，依序是由小網域(192.168.1.0/24是Class C)，逐漸到大網域(169.254.0.0/16是Class B)，最後是預設路由(0.0.0.0/0.0.0.0)。判斷某個網路封包如何傳送時，會經由這個路由來判斷！舉例來說，上頭有三個路由，若有傳往192.168.1.20的封包，首先會找192.168.1.0/24 這個網域的路由，找到就直接由eth0傳送出去；如果是傳送到Yahoo呢？Yahoo 的主機IP是119.160.246.241，通過判斷1)不是192.168.1.0/24，2)不是169.254.0.0/16結果到達3)0/0 時傳出去，透過eth0將封包傳給192.168.1.254。因此當你重複設定多個同樣的路由時，例如在你的主機上的兩張網路卡設定為相同網域的IP時，會出現如下的情況：

|  |
| --- |
| Kernel IP routing table  Destination Gateway Genmask Flags Metric Ref Use Iface  192.168.1.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0  192.168.1.0 0.0.0.0 255.255.255.0 U 0 0 0 eth1 |

路由是依照順序來排列與傳送的，不論封包是由那個介面(eth0, eth1)所接收，都會由eth0傳送出去，所以一部主機上設定兩個相同網域的IP本身沒有意義！除非是類似虛擬機器 (Xen, VMware等軟體)所架設的多主機時，才有這個必要～

|  |
| --- |
| # 範例二：路由的增加與刪除  [root@www ~]# route del -net 169.254.0.0 netmask 255.255.0.0 dev eth0  # 上面這個動作可以刪除掉 169.254.0.0/16 這個網域！  # 請注意，在刪除時，需要將路由表上面出現的資訊都寫入包括 netmask , dev等等參數  [root@www ~]# route add -net 192.168.100.0 netmask 255.255.255.0 dev eth0  # 透過route add來增加一個路由！請注意，這個路由的設定必須要能夠與你的網路互通。  # 舉例來說，如果我下達底下的指令就會顯示錯誤：  # route add -net 192.168.200.0 netmask 255.255.255.0 gw 192.168.200.254  # 主機內僅192.168.1.11這個IP，所以不能與192.168.200.254這網段直接用MAC互通！  [root@www ~]# route add default gw 192.168.1.250  # 增加預設路由的方法！請注意，只要有一個預設路由就夠了喔！  # 同樣的，那個 192.168.1.250 的IP也需要能與你的 LAN 溝通才行！  # 在這個地方如果你隨便設定後，記得使用底下的指令重新設定你的網路  # /etc/init.d/network restart |

當出現『SIOCADDRT: Network is unreachable』錯誤時，肯定是由於gw後面接的IP無法直接與你的網域溝通(Gateway並不在你的網域內)，所以趕緊檢查一下是否輸入錯誤！

|  |  |
| --- | --- |
| **Tips:** 如果接觸到一新環境內的主機，在不想更動原系統的設定檔情況下，然後預計使用本書的網路環境設定時，手動的處理就變成：『ifconfig eth0 192.168.1.100; route add default gw 192.168.1.254』這樣就搞定了！直接聯網與測試。等到完成測試後，再給她/etc/init.d/network restart恢復原系統的網路即可。 |  |

5.1.5 手動使用 DHCP 自動取得 IP 參數： dhclient



如果你使用DHCP協定在區域網路內取得IP，有個更快速的作法，就是利用dhclient這個指令～因為這個指令才是真正發送 dhcp要求工作的程式！如果不考慮其他的參數，使用底下的方法即可：

|  |
| --- |
| [root@www ~]# dhclient eth0 |

這樣就可以立刻叫我們的網路卡以dhcp協定去嘗試取得IP喔！

5.2 網路偵錯與觀察指令



網路偵錯流程還有些重要的偵測指令也得要來瞭解一下才好！

5.2.1 兩部主機兩點溝通： ping



ping主要透過[ICMP封包](http://linux.vbird.org/linux_server/0110network_basic.php#tcpip_network_icmp)來進行整個網路的狀況報告，最重要的是ICMP type 0, 8這兩個類型，分別是要求回報與主動回報網路狀態是否存在的特性。要特別注意的是，ping還是需要透過[IP封包](http://linux.vbird.org/linux_server/0110network_basic.php#tcpip_network_head)來傳送ICMP封包的，而IP封包裡面有個相當重要的TTL屬性，這是很重要的一個路由特性。

|  |
| --- |
| [root@www ~]# ping [選項與參數] IP  選項與參數：  -c 數值：後面接的是執行 ping 的次數，例如 -c 5 ；  -n ：在輸出資料時不進行 IP 與主機名稱的反查，直接使用 IP 輸出(速度較快)；  -s 數值：發送出去的 ICMP 封包大小，預設為 56bytes，不過你可以放大此一數值；  -t 數值：TTL 的數值，預設是 255，每經過一個節點就會少一；  -W 數值：等待回應對方主機的秒數。  -M [do|dont] ：主要在偵測網路的 MTU 數值大小，兩個常見的項目是：  do ：代表傳送一個 DF (Don't Fragment) 旗標，讓封包不能重新拆包與打包；  dont：代表不要傳送 DF 旗標，表示封包可以在其他主機上拆包與打包  # 範例一：偵測一下 168.95.1.1 這部 DNS 主機是否存在？  [root@www ~]# ping -c 3 168.95.1.1  PING 168.95.1.1 (168.95.1.1) 56(84) bytes of data.  64 bytes from 168.95.1.1: icmp\_seq=1 ttl=245 time=15.4 ms  64 bytes from 168.95.1.1: icmp\_seq=2 ttl=245 time=10.0 ms  64 bytes from 168.95.1.1: icmp\_seq=3 ttl=245 time=10.2 ms  --- 168.95.1.1 ping statistics ---  3 packets transmitted, 3 received, 0% packet loss, time 2047ms  rtt min/avg/max/mdev = 10.056/11.910/15.453/2.506 ms |

ping最簡單的功能就是傳送ICMP封包去要求對方主機回應是否存在於網路環境中，上面的回應訊息中，幾個重要的項目是：

* 64 bytes：表示這次傳送的 ICMP 封包大小為 64 bytes 這麼大，這是預設值，在某些特殊場合中，例如要搜索整個網路內最大的 MTU 時，可以使用 -s 2000 之類的數值來取代；
* icmp\_seq=1：ICMP 所偵測進行的次數，第一次編號為 1 ；
* ttl=243：TTL與IP封包內的TTL是相同的，每經過一個帶有 MAC 的節點(node)時，例如 router, bridge 時， TTL 就會減少一，預設的TTL為255，透過-t 150之類的方法來重新設定預設TTL數值；
* time=15.4 ms：回應時間，單位有ms(0.001秒)及us(0.000001秒)，一般來說，越小的回應時間，表示兩部主機之間的網路連線越良好！

如果你忘記加上 -c 3 這樣的規定偵測次數，那就得要使用 [ctrl]-c 將他結束掉了！

|  |
| --- |
| 例題：寫一支腳本程式ping.sh，透過這支腳本程式，你可以用ping偵測整個網域的主機是否有回應。此外，每部主機的偵測僅等待一秒鐘，也僅偵測一次。  答：由於僅偵測一次且等待一秒，因此ping的選項為：-W1 -c1，而位於本機所在的區網為192.168.1.0/24，所以可以這樣寫(vim /root/bin/ping.sh)：  #!/bin/bash  for siteip in $(seq 1 254)  do  site="192.168.1.${siteip}"  ping -c1 -W1 ${site} &> /dev/null  if [ "$?" == "0" ]; then  echo "$site is UP"  else  echo "$site is DOWN"  fi  done |

如果你的主機與待偵測主機並不在同一個網域內，那麼TTL預設使用255，如果是同一個網域內，TTL預設則使用64喔！

用ping追蹤路徑中的最大MTU數值: 加大訊框(frame)對於網路效能是有幫助的，因為封包打包的次數減少，加上整個傳輸的媒體都能夠接受這個frame 而不需要重新進行封包的拆解與重組，那效能當然會更好，修改frame大小的參數就是[MTU](http://linux.vbird.org/linux_server/0110network_basic.php#tcpip_link_mtu)！MTU修改可以透過[ifconfig](http://linux.vbird.org/linux_server/0140networkcommand.php/#ifconfig)或者是[ip](http://linux.vbird.org/linux_server/0140networkcommand.php/#ip_cmd)等指令來達成。  
由於本地端的網路卡MTU也會影響到偵測，所以如果想要偵測整個傳輸媒體的MTU數值，那每個可以調整的主機就得要先使用ifcofig或ip先將MTU調大，然後再去進行偵測，否則就會出現像上面提供的案例一樣，可能會出現錯誤訊息的！不過要連上Internet的主機，不要隨便調整MTU，因為無法知道Internet上面每部機器能夠支援的MTU到多大！另外，其實每種連線方式都有不同的 MTU 值，常見的各種介面的 MTU 值分別為︰

|  |  |
| --- | --- |
| 網路介面 | MTU |
| Ethernet | 1500 |
| PPPoE | 1492 |
| Dial-up(Modem) | 576 |

5.2.2 兩主機間各節點分析： traceroute



ping是兩部主機之間的回聲與否判斷，那追蹤兩部主機之間通過的各個節點(node)通訊狀況的好壞得使用traceroute指令！

|  |
| --- |
| [root@www ~]# traceroute [選項與參數] IP  選項與參數：  -n ：可以不必進行主機的名稱解析，單純用 IP ，速度較快！  -U ：使用 UDP 的 port 33434 來進行偵測，這是預設的偵測協定；  -I ：使用 ICMP 的方式來進行偵測；  -T ：使用 TCP 來進行偵測，一般使用 port 80 測試  -w ：若對方主機在幾秒鐘內沒有回聲就宣告不治...預設是 5 秒  -p 埠號：若不想使用 UDP 與 TCP 的預設埠號來偵測，可在此改變埠號。  -i 裝置：用在比較複雜的環境，如果你的網路介面很多很複雜時，才會用到這個參數；  舉例來說，你有兩條 ADSL 可以連接到外部，那你的主機會有兩個 ppp，  你可以使用 -i 來選擇是 ppp0 還是 ppp1 啦！  -g 路由：與 -i 的參數相仿，只是 -g 後面接的是 gateway 的 IP 就是了。  # 範例一：偵測本機到 yahoo 去的各節點連線狀態  [root@www ~]# traceroute -n tw.yahoo.com  traceroute to tw.yahoo.com (119.160.246.241), 30 hops max, 40 byte packets  1 192.168.1.254 0.279 ms 0.156 ms 0.169 ms  2 172.20.168.254 0.430 ms 0.513 ms 0.409 ms  3 10.40.1.1 0.996 ms 0.890 ms 1.042 ms  4 203.72.191.85 0.942 ms 0.969 ms 0.951 ms  5 211.20.206.58 1.360 ms 1.379 ms 1.355 ms  6 203.75.72.90 1.123 ms 0.988 ms 1.086 ms  7 220.128.24.22 11.238 ms 11.179 ms 11.128 ms  8 220.128.1.82 12.456 ms 12.327 ms 12.221 ms  9 220.128.3.149 8.062 ms 8.058 ms 7.990 ms  10 \* \* \*  11 119.160.240.1 10.688 ms 10.590 ms 119.160.240.3 10.047 ms  12 \* \* \* <==可能有防火牆裝置等情況發生所致 |

traceroute會針對欲連接的目的地之所有node進行UDP的逾時等待，上例中由主機連接到Yahoo時，經過12個節點以上，traceroute會主動的對這12個節點做UDP的回聲等待，並偵測回覆時間，每節點偵測三次，最終回傳像上頭顯示的結果。你可以發現每個節點其實回覆的時間大約在50 ms以內，算是還可以的Internet環境了。

比較特殊的算是第10/12個，回傳星號，代表該node可能設有某些防護措施，讓我們發送的封包資訊被丟棄所致。因為我們是直接透過路由器轉遞封包，並沒有進入路由器去取得路由器的使用資源，所以某些路由器僅支援封包轉遞，並不會接受來自用戶端的各項偵測。因為traceroute預設使用UDP封包，如果你想嘗試使用其他封包，那麼-I或-T可以試看看囉！

由於目前 UDP/ICMP的攻擊層出不窮，因此很多路由器可能就此取消這兩個封包的回應功能。所以我們可以使用TCP來偵測呦！ 例如使用同樣的方法，透過等待時間 1 秒，以及 TCP 80 埠口的情況下，可以這樣做：

|  |
| --- |
| [root@www ~]# traceroute -w 1 -n -T tw.yahoo.com |

5.2.3 察看本機的網路連線與後門： netstat



如果覺得某個網路服務明明啟動了，但無法造成連線，首先應要查詢一下自己的網路介面所監聽的埠口(port)是否真的有啟動！

|  |
| --- |
| [root@www ~]# netstat -[rn] <==與路由有關的參數  [root@www ~]# netstat -[antulpc] <==與網路介面有關的參數  選項與參數：  與路由 (route) 有關的參數說明：  -r ：列出路由表(route table)，功能如同 route 這個指令；  -n ：不使用主機名稱與服務名稱，使用IP與port number，如同 route -n  與網路介面有關的參數：  -a ：列出所有的連線狀態，包括 tcp/udp/unix socket 等；  -t ：僅列出 TCP 封包的連線；  -u ：僅列出 UDP 封包的連線；  -l ：僅列出有在 Listen (監聽) 的服務之網路狀態；  -p ：列出 PID 與 Program 的檔名；  -c ：可以設定幾秒鐘後自動更新一次，例如 -c 5 每五秒更新一次網路狀態的顯示；  # 範例一：列出目前的路由表狀態，且以 IP 及 port number 顯示：  [root@www ~]# netstat -rn  Kernel IP routing table  Destination Gateway Genmask Flags MSS Window irtt Iface  192.168.1.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0  169.254.0.0 0.0.0.0 255.255.0.0 U 0 0 0 eth0  0.0.0.0 192.168.1.254 0.0.0.0 UG 0 0 0 eth0  # 其實這個參數就跟 route -n 一模一樣，對吧！這不是 netstat 的主要功能啦！  # 範例二：列出目前的所有網路連線狀態，使用 IP 與 port number  [root@www ~]# netstat -an  Active Internet connections (servers and established)  Proto Recv-Q Send-Q Local Address Foreign Address State  ....(中間省略)....  tcp 0 0 127.0.0.1:25 0.0.0.0:\* LISTEN  tcp 0 52 192.168.1.100:22 192.168.1.101:1937 ESTABLISHED  tcp 0 0 :::22 :::\* LISTEN  ....(中間省略)....  Active UNIX domain sockets (servers and established)  Proto RefCnt Flags Type State I-Node Path  unix 2 [ ACC ] STREAM LISTENING 11075 @/var/run/hald/dbus-uukdg1qMPh  unix 2 [ ACC ] STREAM LISTENING 10952 /var/run/dbus/system\_bus\_socket  unix 2 [ ACC ] STREAM LISTENING 11032 /var/run/acpid.socket  ....(底下省略).... |

netstat的輸出主要分為兩大部分，分別是TCP/IP的網路介面部分，及傳統Unix socket部分。socket與FIFO檔案就是在Unix介面用來做為程式資料交流的介面了，也就是上頭表格內看到的Active Unix domain sockets的內容～

通常建議加上『-n』這個參數，可避過主機名稱與服務名稱的反查，直接以IP及埠口號碼(port number)來顯示，顯示速度會快很多！至於在輸出的訊息當中，關於網路連線狀態的輸出部分主要是分為底下幾個大項：

* Proto：該連線的封包協定，主要為TCP/UDP等封包；
* Recv-Q：非由使用者程式連接所複製而來的總 bytes 數；
* Send-Q：由遠端主機傳送而來，但不具有ACK標誌的總bytes數，指主動連線SYN或其他標誌封包所佔的bytes 數；
* Local Address：本地端的位址，可以是IP(-n參數存在時)，也可是完整的主機名稱。格式為『IP:port』只是IP的格式有IPv4及IPv6的差異。如上在port22的介面中，使用的:::22是針對IPv6的顯示，事實上就相同於0.0.0.0:22 的意思。至於port 25僅針對lo介面開放，意指Internet基本上是無法連接到我本機的25埠口！
* Foreign Address：遠端的主機IP與port number。
* stat：狀態列，主要的狀態含有：
  + ESTABLISED：已建立連線的狀態；
  + SYN\_SENT：發出主動連線 (SYN 標誌) 的連線封包；
  + SYN\_RECV：接收到一個要求連線的主動連線封包；
  + FIN\_WAIT1：該插槽服務(socket)已中斷，該連線正在斷線當中；
  + FIN\_WAIT2：該連線已掛斷，但正在等待對方主機回應斷線確認的封包；
  + TIME\_WAIT：該連線已掛斷，但 socket 還在網路上等待結束；
  + LISTEN：通常用在服務的監聽 port ！可使用『 -l 』參數查閱。

netstat的功能就是觀察網路的連線狀態，而網路連線狀態中，又以觀察『我目前開了多少的port在等待用戶端的連線』以及『目前網路連線狀態中，有多少連線已建立或產生問題』最常見。那你]如何瞭解與觀察呢：

|  |
| --- |
| # 範例三：秀出目前已經啟動的網路服務  [root@www ~]# netstat -tulnp  Active Internet connections (only servers)  Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name  tcp 0 0 0.0.0.0:34796 0.0.0.0:\* LISTEN 987/rpc.statd  tcp 0 0 0.0.0.0:111 0.0.0.0:\* LISTEN 969/rpcbind  tcp 0 0 127.0.0.1:25 0.0.0.0:\* LISTEN 1231/master  tcp 0 0 :::22 :::\* LISTEN 1155/sshd  udp 0 0 0.0.0.0:111 0.0.0.0:\* 969/rpcbind  ....(底下省略)....  # 上面最重要的是-l的參數，因為可以僅列出有在Listen的port |

很多的網路服務其實僅針對本機的lo開放而已，網際網路是連接不到該埠口與服務的。而由上述的資也可看到，啟動port 111的，其實就是rpcbind那隻程式，如想要關閉這個埠口，可使用kill刪除 PID 969，也可使用killall刪除rpcbind這個程序即可。如此一來， 很輕鬆的你就能知道哪個程式啟動了哪些埠口囉！

|  |
| --- |
| # 範例四：觀察本機上頭所有的網路連線狀態  [root@www ~]# netstat -atunp  Active Internet connections (servers and established)  Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program  tcp 0 0 0.0.0.0:111 0.0.0.0:\* LISTEN 969/rpcbind  tcp 0 0 0.0.0.0:22 0.0.0.0:\* LISTEN 1155/sshd  tcp 0 0 127.0.0.1:25 0.0.0.0:\* LISTEN 1231/master  tcp 0 52 192.168.1.100:22 192.168.1.101:1937 ESTABLISHED 4716/0  ....(底下省略).... |

上頭的特殊字體表目前已經建立的一條網路連線，由遠端主機192.168.1.101啟動一個大於1024的埠口向本地端主機192.168.1.100的port 22的連線：『Client端是隨機取一個大於1024的port進行連線』，此外『只有root可以啟動小於1024以下的 port 』！如果想要砍掉這條連線，用kill看到到PID為4716！Unix socket通常是用在一些僅在本機上運作的程式所開啟的插槽介面檔，如X Window都是在本機上運作，那何必啟動網路的port呢？當然可以使用Unix socket！

|  |
| --- |
| 例題：請說明服務名稱與 port number 的對應在 Linux 當中，是用那個檔案來設定對應的？  答：/etc/services |

5.3 遠端連線指令與即時通訊軟體



遠端連線其實是在不同的電腦之間進行登入！我們可以透過 telnet, ssh 或者是 ftp 等協定來進行遠端主機的登入。

5.3.1 終端機與 BBS 連線： telnet



telnet的資料在傳送時是使用明碼(原始的資料，沒有加密)，所以資料在Internet上跑時，會比較危險一點(怕被別人監聽)。

|  |
| --- |
| [root@www ~]# telnet [host|IP [port]]  # 範例一：連結到台灣相當熱門的 PTT BBS 站 ptt.cc  [root@www ~]# yum install telnet <==預設沒有安裝這軟體  [root@www ~]# telnet ptt.cc  歡迎來到 批踢踢實業坊 目前有【100118】名使用者與您一同對抗炎炎夏日。  請輸入代號，或以 guest 參觀，或以 new 註冊:  [高手召集令] 台灣駭客年會 暑假與你駭翻南港 http://reg.hitcon.org/hit2011  要學電腦，首選台灣大學資訊訓練班! http://tinyurl.com/3z42apw |

如上所示，我們可以透過telnet輕易的連結到BBS上面，如果你的主機有開啟telnet伺服器服務的話，同樣的利用『telnet IP』並且輸入帳號與密碼之後，就能夠登入主機了。除了連結到伺服器以及連結到BBS站之外，telnet還可以用來連結到某個port (服務)上頭吶！舉例來說，我們可以用telnet連接到port 110，看看這個port是否有正確的啟動呢？

|  |
| --- |
| # 範例二：偵測本機端的 110 這個 port 是否正確啟動？  [root@www ~]# telnet localhost 110  Trying 127.0.0.1...  telnet: connect to address 127.0.0.1: Connection refused  # 如果出現這樣的訊息，表這個 port 沒有啟動或這個連線有問題，因為你看到那個refused！  [root@www ~]# telnet localhost 25  Trying ::1...  Connected to localhost.  Escape character is '^]'.  220 www.centos.vbird ESMTP Postfix  ehlo localhost  250-www.centos.vbird  250-PIPELINING  250-SIZE 10240000  ....(中間省略)....  250 DSN  quit  221 2.0.0 Bye  Connection closed by foreign host. |

根據輸出的結果能知道這通訊協定 (port number 提供的通訊協定功能) 是否有成功的啟動！每個port所監聽的服務都有其特殊的指令，如上述的port 25就是在本機介面所提供的電子郵件服務，那個服務所支援的指令就如同上面使用的資料一樣，但其他的port不見得支援這個『ehlo』的命令，因不同的port有不同的程式！支援的命令就不同囉！

5.3.2 FTP 連線軟體： ftp, lftp



如想要一口氣傳送個幾百MB的檔案，恐怕還是得要透過FTP這個通訊協定才行。



ftp

ftp 這個指令很簡單，用在處理FTP伺服器的下載資料。這裡使用崑山科大的 FTP 伺服器為例：

|  |
| --- |
| [root@www ~]# ftp [host|IP] [port]  # 範例一：連線到崑山科大去看看  [root@www ~]# yum install ftp  [root@www ~]# ftp ftp.ksu.edu.tw  Connected to ftp.ksu.edu.tw (120.114.150.21).  220---------- Welcome to Pure-FTPd [privsep] ----------  220-You are user number 1 of 50 allowed.  220-Local time is now 16:25. Server port: 21.  220-Only anonymous FTP is allowed here <==訊息要看啊！這個 FTP 僅支援匿名  220-IPv6 connections are also welcome on this server.  220 You will be disconnected after 5 minutes of inactivity.  Name (ftp.ksu.edu.tw:root): anonymous <==鳥哥這裡用匿名登入！  230 Anonymous user logged in <==嗯！確實是匿名登入了！  Remote system type is UNIX.  Using binary mode to transfer files.  ftp> <==最終登入的結果看起來是這樣！  ftp> help <==提供需要的指令說明，可以常參考！  ftp> dir <==顯示遠端伺服器的目錄內容 (檔名列表)  ftp> cd /pub <==變換目錄到 /pub 當中  ftp> get filename <==下載單一檔案，檔名為 filename  ftp> mget filename\* <==下載多個檔案，可使用萬用字元 \*  ftp> put filename <==上傳 filename 這個檔案到伺服器上  ftp> delete file <==刪除主機上的 file 這個檔案  ftp> mkdir dir <==建立 dir 這個目錄  ftp> lcd /home <==切換『本地端主機』的工作目錄  ftp> passive <==啟動或關閉 passive 模式  ftp> binary <==資料傳輸模式設定為 binary 格式  ftp> bye <==結束 ftp 軟體的使用 |

FTP使用兩個port分別進行命令與資料的交流。首先是登入，上頭的表格中填入帳號與密碼。由於崑山科大僅提供匿名登入，而匿名登入者的帳號是『anonymous』所以直接填寫那帳號即可。如果是私人的FTP，才需要一組完整的帳號與密碼！

登入FTP主機後，就能夠使用ftp軟體的功能進行上傳與下載的動作，幾個常用的ftp內指令如上表，可連到大學的FTP網站後，使用help(或問號 ?)來參考可用的指令，然後嘗試下載以測試使用一下這個指令！這樣以後沒有瀏覽器的時候，也可到ftp 下載了！另外要注意的是，離開ftp軟體時，得要輸入『 bye』喔！不是『exit』！

如果由於某些理由，讓你的FTP主機的port開在非正規的埠口時，那你就可以利用底下的方式來連接到該部主機喔！

|  |
| --- |
| [root@www ~]# ftp hostname 318  # 假設對方主機的 ftp 服務開啟在 318 這個 port 啊！ |

5.4 文字介面網頁瀏覽



文字界面下上網瀏覽的好工具！分別是links及wget，但是，必需要確定你已經安裝了這兩個套件才行。

5.4.2 文字介面下載器： wget



如果links是在進行網頁的『瀏覽』，那 wget就是在進行『網頁資料的取得』。舉例來說，我們的Linux核心是放置在 www.kernel.org 內，同時提供ftp與http來下載。可以使用 lftp 來下載資料，但如果想要用瀏覽器下載就利用wget！

|  |
| --- |
| [root@www ~]# wget [option] [網址]  選項與參數：  若想要連線的網站有提供帳號與密碼的保護時，可以利用這兩個參數來輸入喔！  --http-user=usrname  --http-password=password  --quiet ：不要顯示 wget 在抓取資料時候的顯示訊息  更多的參數請自行參考 man wget 吧！ ^\_^  # 範例一：請下載 2.6.39 版的核心  [root@www ~]# wget \  > http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.39.tar.bz2  --2011-07-18 16:58:26-- http://www.kernel.org/pub/linux/kernel/v2.6/..  Resolving www.kernel.org... 130.239.17.5, 149.20.4.69, 149.20.20.133, ...  Connecting to www.kernel.org|130.239.17.5|:80... connected.  HTTP request sent, awaiting response... 200 OK  Length: 76096559 (73M) [application/x-bzip2]  Saving to: `linux-2.6.39.tar.bz2'  88% [================================> ] 67,520,536 1.85M/s eta 7s |

不必透過瀏覽器，只要知道網址後，立即可以進行檔案的下載。

5.5 封包擷取功能



由於網路連線出問題，使用類似ping的軟體卻無法找出問題，最常見的是因為路由與IP轉遞後所產生的一些困擾，最簡單的方法是『分析封包的流向』！透過分析封包的流向，可以瞭解一條連線應該是如何進行雙向的連線的動作，也就會清楚可能發生的問題所在了！底下我們就來談一談這個 tcpdump 與圖形介面的封包分析軟體吧！

5.5.1 文字介面封包擷取器：tcpdump



tcpdump不但可分析封包的流向，連封包的內容也可以進行『監聽』，如果使用的傳輸資料是明碼，在router或hub上就可能被人家監聽！(註：tcpdump必須使用root的身份執行)

|  |
| --- |
| [root@www ~]# tcpdump [-AennqX] [-i 介面] [-w 儲存檔名] [-c 次數] \  [-r 檔案] [所欲擷取的封包資料格式]  選項與參數：  -A ：封包的內容以 ASCII 顯示，通常用來捉取 WWW 的網頁封包資料。  -e ：使用資料連接層 (OSI 第二層) 的 MAC 封包資料來顯示；  -nn：直接以IP及port number顯示，而非主機名與服務名稱  -q ：僅列出較為簡短的封包資訊，每一行的內容比較精簡  -X ：可以列出十六進位 (hex) 以及 ASCII 的封包內容，對於監聽封包內容很有用  -i ：後面接要『監聽』的網路介面，例如 eth0, lo, ppp0 等等的介面；  -w ：將監聽所得的封包資料儲存下來，用這個參數就對了！後面接檔名  -r ：從後面接的檔案將封包資料讀出來。『檔案』是已經存在，並由 -w 所製作出來的。  -c ：監聽的封包數，如沒這個參數， tcpdump會不斷的監聽，直到輸入[ctrl]-c 為止。  所欲擷取的封包資料格式：我們可以專門針對某些通訊協定或者是IP來源進行封包擷取，  那就可以簡化輸出的結果，並取得最有用的資訊。常見的表示方法有：  'host foo', 'host 127.0.0.1' ：針對單部主機來進行封包擷取  'net 192.168' ：針對某個網域來進行封包的擷取；  'src host 127.0.0.1' 'dst net 192.168'：同時加上來源(src)或目標(dst)限制  'tcp port 21'：還可以針對通訊協定偵測，如 tcp, udp, arp, ether 等  還可以利用 and 與 or 來進行封包資料的整合顯示呢！  # 範例一：以 IP 與 port number 捉下 eth0 這個網路卡上的封包，持續 3 秒  [root@www ~]# tcpdump -i eth0 -nn  tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes  17:01:47.360523 IP 192.168.1.101.1937 > 192.168.1.100.22: Flags [.], ack 196, win 65219,  17:01:47.362139 IP 192.168.1.100.22 > 192.168.1.101.1937: Flags [P.], seq 196:472, ack 1,  17:01:47.363201 IP 192.168.1.100.22 > 192.168.1.101.1937: Flags [P.], seq 472:636, ack 1,  17:01:47.363328 IP 192.168.1.101.1937 > 192.168.1.100.22: Flags [.], ack 636, win 64779,  <==按下 [ctrl]-c 之後結束  6680 packets captured <==捉下來的封包數量  14250 packets received by filter <==由過濾所得的總封包數量  7512 packets dropped by kernel <==被核心所丟棄的封包 |

我們可以約略區分為數個欄位，我們以範例一當中那個特殊字體行來說明一下：

* 17:01:47.362139：這個是此封包被擷取的時間，『時:分:秒』的單位；
* IP：透過的通訊協定是 IP ；
* 192.168.1.100.22 >：傳送端是192.168.1.100這個 IP，而傳送的port number為 22，大於 (>) 的符號指的是封包的傳輸方向！
* 192.168.1.101.1937：接收端的IP是192.168.1.101，且該主機開啟port 1937來接收；
* [P.], seq 196:472：這個封包帶有PUSH的資料傳輸標誌，且傳輸的資料為整體資料的196~472 byte；
* ack 1：ACK 的相關資料。

該封包是由192.168.1.100傳到192.168.1.101，透過的port是由22到1937，使用的是PUSH的旗標，而不是SYN之類的主動連線標誌。務必到[TCP 表頭資料](http://linux.vbird.org/linux_server/0110network_basic.php#tcpip_transfer_tcp)的部分去瞧一瞧！

一個網路很忙的主機上，想取得某主機對連線的封包資料時，使用tcpdump配合管線命令與正規表示法也可，不過畢竟不好捉取！透過tcpdump的表示法功能，能輕易的將所需要的資料獨立的取出來。簡化如只取port 21的連線封包，可以這樣做：

|  |
| --- |
| [root@www ~]# tcpdump -i eth0 -nn port 21  tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes  01:54:37.96 IP 192.168.1.101.1240 > 192.168.1.100.21: . ack 1 win 65535  01:54:37.96 IP 192.168.1.100.21 > 192.168.1.101.1240: P 1:21(20) ack 1 win 5840  01:54:38.12 IP 192.168.1.101.1240 > 192.168.1.100.21: . ack 21 win 65515  01:54:42.79 IP 192.168.1.101.1240 > 192.168.1.100.21: P 1:17(16) ack 21 win 65515  01:54:42.79 IP 192.168.1.100.21 > 192.168.1.101.1240: . ack 17 win 5840  01:54:42.79 IP 192.168.1.100.21 > 192.168.1.101.1240: P 21:55(34) ack 17 win 5840 |

這樣就僅提出port 21的資訊而已，封包的傳遞都是雙向的，client端發出『要求』而server端則予以『回應』，所以是有去有回！可經過這個封包的流向來瞭解到封包運作的過程。

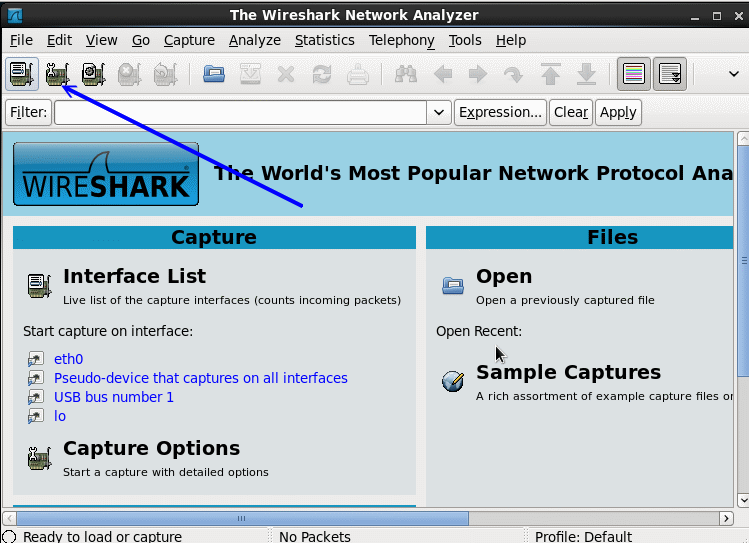
|  |
| --- |
| 例題：如何用tcpdump監聽來自eth0介面卡，通訊為port 22，封包來源為192.168.1.101的資料？  答：tcpdump -i eth0 -nn 'port 22 and src host 192.168.1.101' |

5.5.2 圖形介面封包擷取器： wireshark



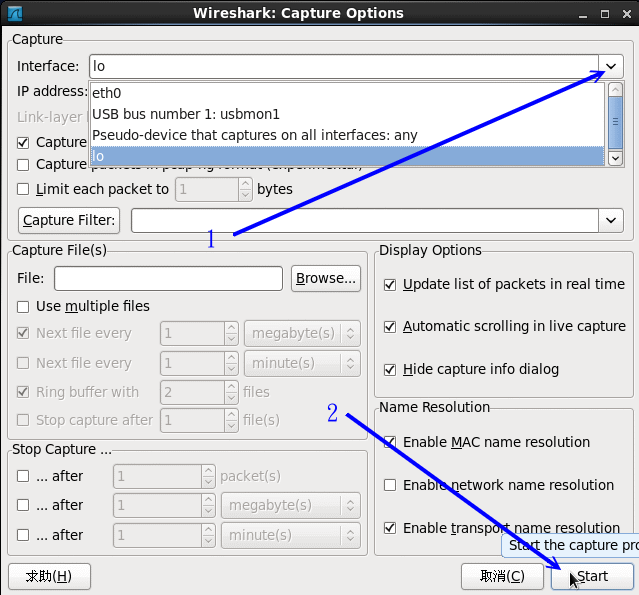
wireshark ([註1](http://linux.vbird.org/linux_server/0140networkcommand.php/#ps1))啟動的畫面如下所示：

圖 5.5-1、wireshark 的使用示意圖



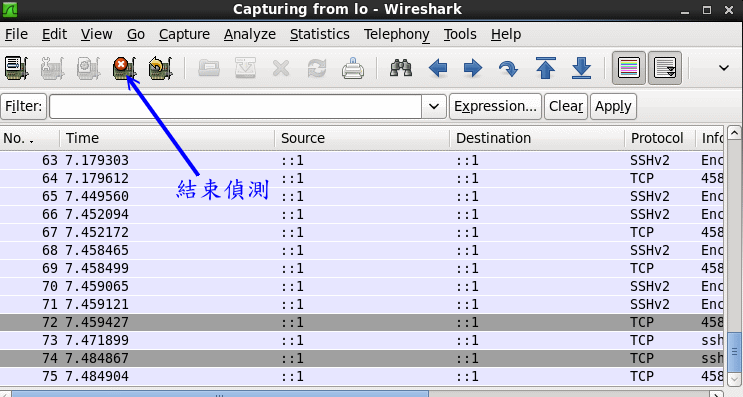
想要開始擷取封包前，要設定一下監聽的介面之類的，點選圖5.5-1 畫面中的網路卡小圖示！出現如下的畫面給你選擇了。

圖 5.5-2、wireshark 的使用示意圖



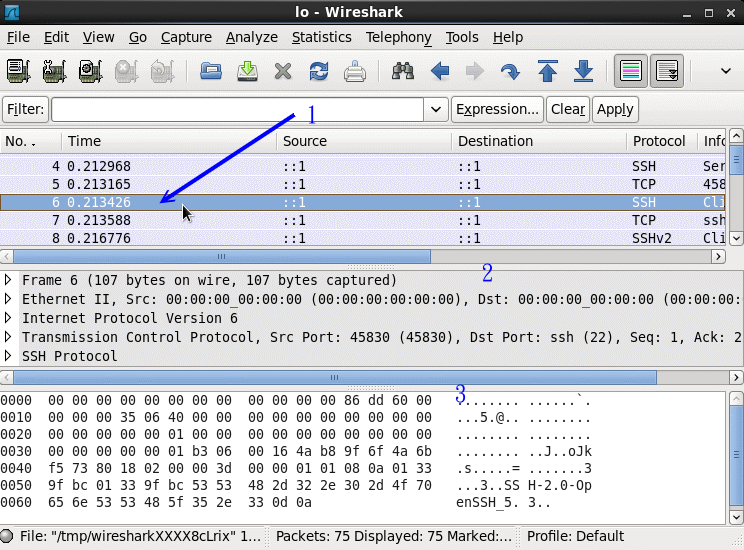
在上圖中，先選擇想要監聽的介面，因擔心外部的封包太多導致畫面很亂，因此這裡使用內部的lo介面作為範例。lo平時是很安靜的！所以在點選了『start』並打開終端機後，使用『ssh localhost』嘗試登入自己，才能夠獲得封包喔！如下圖所示：

圖 5.5-3、wireshark 的使用示意圖



若沒有問題，等到你擷取了足夠的封包想要進行分析之後，按下圖 5.5-3 畫面中的停止小圖示，那麼封包擷取的動作就會終止， 接下來，就讓我們來開始分析一下封包吧！

圖 5.5-4、wireshark 的使用示意圖



整個分析的畫面如上所示，畫面總共分為三大區塊，你可以將滑鼠游標移動到每個區塊中間的移動棒， 就可以調整每個區塊的範圍大小了。第一區塊主要顯示的是封包的標頭資料，內容就有點類似tcpdump的顯示結果，第二區塊則是詳細的表頭資料，包括訊框的內容、通訊協定的內容以及socket pair等等資訊。第三區塊則是16進位與ASCII碼的顯示結果(詳細的封包內容)。

如果你覺得某個封包有問題，在畫面1的地方點選該封包 (圖例中是第6個封包)，那麼畫面2與3就會跟著變動！由於鳥哥測試的封包是加密資料的封包，因此畫面2顯示出封包表頭，但畫面3的封包內容就是亂碼啦！透過這個wireshark 你就可以一口氣得到所需要的所有封包內容啦！而且還是圖形介面的，很方便吧！