

經由實際操作深入體驗 Cisco 路由的強大效能

實作 Cisco 路由器 與 Linux 的動態路由

文◎奇科電腦教育資深網路技術顧問 Ben 哥和 CCIE 講師群共同編撰

筆者教授 Linux 相關課程已有六年以上的經驗，看過不少沒基礎或程度不錯的學員。以網路工程師的角度來看，系統工程師的網路知識是相當的薄弱。因此筆者在這幾期中，希望以 Linux 結合 Cisco 路由器（Router）的方式，來加強系統工程師對網路路由（Routing）的觀念。

實驗所需的設備如下：

1. Linux 機器兩台
2. Cisco 3500 系列第二層交換機和 2600 系列路由器各乙台
3. 乙太網路線數條

實驗所需的 Linux 軟體清單：

1. Zebra (<http://www.zebra.org/>) 路由通訊免費軟體

讀者必須具備的基本知識：

1. 簡單的 OSI 七層架構觀念。
2. 對 IP 網路、廣播位址，子網路遮罩的了解及計算。

何謂路由 (Routing)

路由就是由甲電腦傳輸資料到乙電腦，以 OSI 第三層——網路層 Network Layer（例如以 IP 為基準）來轉送封包 (Packet) 的一種機制。思科系統股份有限公司 (Cisco Systems, Inc.) 對路由的定義記載「http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/routing.htm#xtocid2」網頁之中。

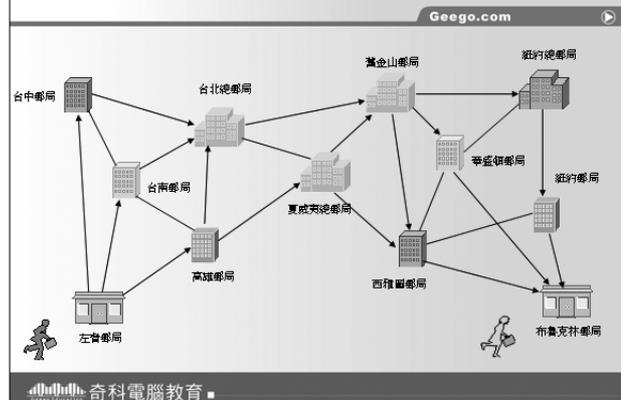
電腦與電腦之間的資料傳輸必須經由網路，而網路可以透過直接連接兩台電腦的方式，或是以一個或一個以上的節點來構成。OSI 的第三層（網路層）的定址就是網路工程師所謂的邏輯定址 (Logical Address)，而資料鏈結層 (Data Link Layer) 中的實體定址 (Physical Address)，在區域網路 (LAN) 中叫做 MAC 位址。以目前的網路來說，在網路層的邏輯定址中主要的通訊協定有 IP、IPX、AppleTalk 等，其中又以 IP 通訊協定為目前最大的主流，無庸置疑地，Cisco Systems, Inc. 是全球公認普及 IP 網路的最大推手。

IP 在網路世界中所扮演的角色，就好比日常生活中每戶家庭的「住址」一樣，IP 就是每台電腦的住址。有了住址，發件者送出去的資料，才能根據地址送達收件者，收件者也只能根據資料上內存的資料來源位址回傳資料給發件者。

因此，經由路由 (Routing) 的機制，可以達成電腦之間資料互相交換的目的。

為什麼需要路由

哪一條是最佳路徑?



▲路由機制的運作

如果沒有路由的話，資料將無法有效率地被轉送。

我們可以從生活中的寄信流程，大致了解為什麼需要路由機制。試想，在台灣的我們想要寄一封明信片給遠在紐約布魯克林的朋友，必須在明信片上填寫「對方的地址」和「自己的地址」。投入郵筒之後，郵局會根據明信片上「收件者地址」經由不同的郵局轉送到紐約的布魯克林郵局，最後送達朋友的家裡。

那麼問題就來了，郵件到達郵局後，該郵局可以轉送的其他郵局相當的多，例如，我人在高雄左營，郵件到達左營郵局後，內部的作業必須決定下一個郵局該是高雄郵局、台南郵局或是屏東郵局呢？郵務系統本身就是一個大型的網路系統，每間郵局身負轉送郵件的使命，只要郵件一到某個郵局，其目標就是迅速準確地把郵件送到下一個郵局，也就是郵件透過郵局的「路由」機制順利有效地到達目的地。

網路的架構是由無數的特殊電腦相連而成的，這些特殊的電腦有一個固定的名稱叫「路由器」。其實路由器就像是郵局，以目的地位址 (Destination IP) 為根據來轉送封包給下一個路由器，而下一個路由器也根據目的地位址，來轉送封包到下一個路由器，反覆以上的步驟，封包就可以順利地到達目的地的位址。

淺談 Linux 的路由

Linux 的路由表可以經由「`route -n`」命令輸出：

```
root@linux_MBT_zebra root# route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.6.0 0.0.0.0 255.255.255.0 U 0 0 0 eth1
10.0.0.0 0.0.0.0 255.255.255.0 U 0 0 0 eth1
211.72.111.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
192.168.8.0 0.0.0.0 255.255.255.0 U 0 0 0 eth2
169.254.0.0 0.0.0.0 255.255.0.0 U 0 0 0 eth2
127.0.0.0 0.0.0.0 255.0.0.0 U 0 0 0 lo
0.0.0.0 211.72.111.254 0.0.0.0 UG 0 0 0 eth0
```

透過以下圖表就可以瞭解圖片中路由表每一項領域的涵義：

各領域名稱 (field name)	目的 (Objective)
Kernel IP routing table	核心內的 IP 路由表
Destination	目的地的 IP 網路位址 (Network Address)
Gateway	閘道器的 IP 位址
Genmask	Destination 的子網路遮罩
Flag	旗標。用來代表此路由目前的狀態
Metric	需要經過幾個網路節點 (hops) 才能到達路由的目標網路位址。通常被路由 daemon 所使用，好比 RIP 等
Ref	參考到此路由規則的數目。通常被路由 daemon 所使用，像是 RIP 等
Use	有幾個轉送封包參考到此筆路由規則。只有在執行「 <code>route -C</code> 」或「 <code>route -F</code> 」命令時才會顯示
Iface	Interface (介面)

每一列代表一筆路由規則，綜合每筆規則即成一個路由表。對於系統工程師來說，大部分只會參考 Destination、Gateway、Iface，頂多再加上 Genmask 一項。每當封包到達 Linux 主機之後，會根據最長符合原理 (Longest Prefix Match)，對每一條路由規則進行比對 (Matching) 的動作，而每一條路由規則都包含了轉送介面 (Iface)，用來決定封包將從哪一個介面轉送出去。

路由的種類

一般來說，Linux 上的路由都屬於靜態路由 (Static route)。靜態路由的產生，是系統管理員使用「`route`」命令所加入的靜態路由規則，也就是藉手動輸入的方式來加入路由規則。

相對於靜態路由，另一種當然稱之為動態路由 (Dynamic Route)。動態路由就是無須手動輸入路由規則，其路由規則是本機與不同機器彼此經由路由程式 (Routing daemon) 相互交換路由規則而來。一般在網路界常聽到的 RIPv1、RIPv2、IGRP、EIGRP、OSPF、ISIS、BGP 等，都是路由的通訊協定。

可想而知，當網路上有數十或數百台以上的路由器時，如果我們使用靜態路由的話，就必須在每一台路由器上輸入數百條的路由規則，這不僅曠日費時，而且往後要更動 IP 位址時，更可能牽一髮而動全身，因此唯有靠動態路由的自動路由規則交換機制，才能有效率地交換數百條以上的路由規則。

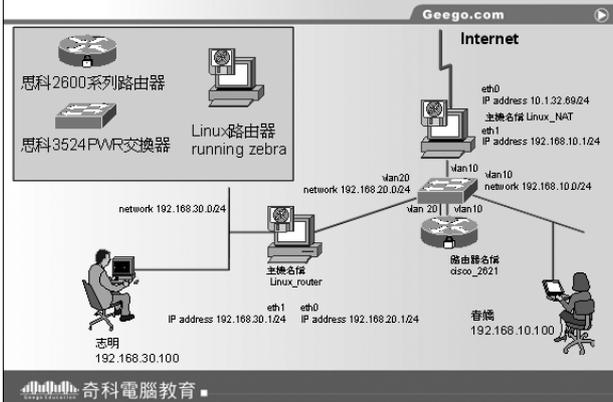
動態路由通訊協定分為「Distance-Vector」和「Link State」兩大類。對這兩類路由通訊協定的差別有興趣的讀者，建議不妨進修 Cisco 的 CCNA 課程來實際了解網路路由的原理。

RIP (Routing Information Protocol) 路由通訊協定簡介

RIP 是一個相當早期且屬於 Distance-Vector 類的路由通訊協定，以 UDP 的資料格式透過連接埠 520 來交換路由資料。RIP 於西元 1982 年出現在美國加州柏克萊大學所發展的作業系統 BSD (Berkeley System Distribution) 4.2 版中，繼而在網路和系統界逐漸流行。

實驗環境介紹

網路拓撲 (Network Topology)



▲網路拓撲圖

建議讀者先研究一下這張網路拓撲圖，如果十分清楚要將哪些路由規則加在哪些設備上，讓春嬌跟志明的電腦可以交換資料，對瞭解網路的運行原理將有實質上的幫助。同時，對接下來的實驗也會比較容易上手，並真正瞭解其中的奧妙。

這裡先給各位一個提示，所有的設備在沒有額外設定之前，春嬌是不能夠 ping 志明的電腦。

NOTE 在實驗開始之前，請先把 Linux 機器上的 iptables 防火牆規則移除，執行「iptables -F && iptables -X」命令即可。

以下是各台實驗設備的 IP 位址設定和路由表內容：

```

~Cisco_2621_奇科電腦教育
interface FastEthernet0/0
ip address 192.168.10.2 255.255.255.0
duplex auto
speed auto
!
interface FastEthernet0/1
ip address 192.168.20.2 255.255.255.0
duplex auto
speed auto
!

```

▲ Cisco_2621 的介面 IP 設定

```

~Cisco_2621_奇科電腦教育
Cisco_2621#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route

Gateway of last resort is 192.168.10.1 to network 0.0.0.0

C    192.168.10.0/24 is directly connected, FastEthernet0/0
C    192.168.20.0/24 is directly connected, FastEthernet0/1
S*   0.0.0.0/0 [1/0] via 192.168.10.1
Cisco_2621#

```

▲ Cisco_2621 的路由表

```

~Linux_NAT_zebra_奇科電腦教育
[root@localhost root]# ifconfig | grep -01 eth
eth0      Link encap:Ethernet HWaddr 00:0C:76:41:CA:D8
          inet addr:10.1.32.69 Bcast:10.1.32.255 Mask:255.255.255.0
--
eth1      Link encap:Ethernet HWaddr 00:08:01:7B:74:6B
          inet addr:192.168.10.1 Bcast:192.168.10.255 Mask:255.255.255.0
[root@localhost root]# netstat -nr
Kernel IP routing table
Destination Gateway      Genmask         Flags MSS Window irtt Iface
192.168.10.0 0.0.0.0      255.255.255.0   U     0 0 0 eth0
10.1.32.0    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
127.0.0.0   0.0.0.0      255.0.0.0       U     0 0 0 lo
0.0.0.0     10.1.32.1    0.0.0.0         UG    0 0 0 eth0
[root@localhost root]#

```

▲ Linux_NAT 機器的設定

```

~Linux_NAT_zebra_奇科電腦教育
[root@Linux_NAT_zebra root]# iptables -L -n -t nat
Chain PREROUTING (policy ACCEPT)
target prot opt source destination

Chain POSTROUTING (policy ACCEPT)
target prot opt source destination

Chain OUTPUT (policy ACCEPT)
target prot opt source destination
[root@Linux_NAT_zebra root]# iptables -t nat -A POSTROUTING -o eth0 -j SNAT --t
o 10.1.32.69
[root@Linux_NAT_zebra root]# iptables -L -n -t nat
Chain PREROUTING (policy ACCEPT)
target prot opt source destination

Chain POSTROUTING (policy ACCEPT)
target prot opt source destination
SNAT all -- 0.0.0.0/0 0.0.0.0/0 to:10.1.32.69

Chain OUTPUT (policy ACCEPT)
target prot opt source destination
[root@Linux_NAT_zebra root]#

```

▲簡單的 NAT 設定

```

~Linux_router_zebra_奇科電腦教育
[root@Linux_router root]# ifconfig | grep -01 eth && netstat -nr
eth0      Link encap:Ethernet HWaddr 00:0C:76:41:CA:FA
          inet addr:192.168.20.1 Bcast:192.168.20.255 Mask:255.255.255.0
--
eth1      Link encap:Ethernet HWaddr 00:50:22:E9:CD:E7
          inet addr:192.168.30.1 Bcast:192.168.30.255 Mask:255.255.255.0
Kernel IP routing table
Destination Gateway      Genmask         Flags MSS Window irtt Iface
192.168.20.0 0.0.0.0      255.255.255.0   U     0 0 0 eth0
192.168.30.0 0.0.0.0      255.255.255.0   U     0 0 0 eth1
169.254.0.0 0.0.0.0      255.255.0.0     U     0 0 0 eth1
127.0.0.0   0.0.0.0      255.0.0.0       U     0 0 0 lo
0.0.0.0     192.168.20.2 0.0.0.0         UG    0 0 0 eth0
[root@Linux_router root]#

```

▲ Linux_router 機器的設定

至於交換器的設定，筆者在上一期（11期）已介紹過了，這裡就不再贅述。

接下來，請讀者們使用「ping」這個指令來檢視各個設備之間的連線是否正常。如果連線一切正常的話，再試著直接從 Linux_NAT 機器上執行「ping」指令來檢查與 Linux_router 之間的連線狀況。

從圖片中的顯示結果發現，根本 ping 不到，這種結果在原先的預期之中。

```

~Linux_NAT_zebra_奇科電腦教育
[root@localhost root]# ping 192.168.20.1
PING 192.168.20.1 <192.168.20.1> 56(84) bytes of data.

-- 192.168.20.1 ping statistics --
17 packets transmitted, 0 received, 100% packet loss, time 16017ms
[root@localhost root]#

```

Linux 設備的設定

在 Linux 上有幾個相當有名的路由程式，包括 routed、zebra、gated 等，在這一次的文章中，筆者選用 zebra 為實驗中的路由程式，原因無他，只因業界在 Linux 為基礎的網路設備上，大都使用 zebra。套一句筆者常跟學生說的話，zebra 是經過業界淬鍊過的程式，絕對可以信賴。更重要的是，zebra 的介面與 Cisco 的 IOS 相似度極高，讓熟悉 Cisco 設備的工程師可以省下不少重新摸索的時間。另外，zebra 在 Fedora Core 4 及 CentOS 4 上已經是標準套件。

在此以 Red Hat 9 架構的 Linux 實驗平台，並使用檔名為 zebra-0.93b-1.i386.rpm 的 zebra rpm 套件，來介紹如何安裝 zebra。請先將 zebra 安裝好，然後繼續下面的操作說明。

從 zebra-0.93b-1.i386.rpm 內包裝的檔案清單中可以看出 zebra 有那些組態檔和執行檔。

```
~Linux_NAT_zebra 奇科電腦教育
[geego@Linux_NAT_zebra tmp]$ rpm -ql zebra-0.93b-1.i386.rpm | head -20
warning: zebra-0.93b-1.i386.rpm: U3 DSA signature: NOKEY, key ID db42a60e
/etc/logrotate.d/zebra
/etc/pass.d/zebra
/etc/rc.d/init.d/bgpd
/etc/rc.d/init.d/ospf6d
/etc/rc.d/init.d/ospf6d
/etc/rc.d/init.d/ripd
/etc/rc.d/init.d/ripngd
/etc/rc.d/init.d/zebra
/etc/zebra
/usr/bin/utysb
/usr/bin/bgpd
/usr/bin/ospf6d
/usr/bin/ospf6d
/usr/bin/ripd
/usr/bin/ripngd
/usr/bin/zebra
/usr/share/doc/zebra-0.93b
/usr/share/doc/zebra-0.93b/AUTHORS
/usr/share/doc/zebra-0.93b/COPYING
/usr/share/doc/zebra-0.93b/ChangeLog
[geego@Linux_NAT_zebra tmp]$
```

Zebra 的組態檔是「/etc/zebra/zebra.conf」，但是剛剛安裝的「zebra.conf」並沒有什麼內容，因此必須將「/usr/share/doc/zebra-0.93b/zebra.conf.sample」範例檔複製到「/etc/zebra/」目錄內並更名為「zebra.conf」，就建立好「/etc/zebra/zebra.conf」組態檔。另外，在「/usr/share/doc/zebra-0.93b/」目錄下還有一些 sample 檔，請複製一份到「/etc/zebra/」目錄下，並且將副檔名「.sample」刪除。

```
~Linux_NAT_zebra 奇科電腦教育
[geego@Linux_NAT_zebra zebra-0.93b]$ pwd ; ls
/usr/share/doc/zebra-0.93b
AUTHORS          INSTALL          REPORTING-BUGS  tools
bgpd.conf.sample NEWS             ripd.conf.sample  utysb.conf.sample
bgpd.conf.sample2 ospf6d.conf.sample  ripngd.conf.sample  zebra.conf.sample
ChangeLog        ospf6d.conf.sample SERVICES         zebra.html
COPYING          README          TODO
```

接下來，使用 root 的身分執行 zebra 路由程式，然後檢查 zebra 是否在執行程序表中。

```
~Linux_NAT_zebra 奇科電腦教育
[root@Linux_NAT_zebra ~]# service zebra start
啟動 zebra:
[root@Linux_NAT_zebra ~]# ps -ef | grep zebra
root      4698      1  0 14:38 ?        00:00:00 /usr/sbin/zebra -d
root      4703    4612  0 14:38 pts/0    00:00:00 grep zebra
[root@Linux_NAT_zebra ~]#
```

根據以上的指令輸出結果，確認 zebra 已經開始執行了。使用 zebra 的好處就是它那神似於 Cisco IOS 的控制介面。zebra 的組態檔「/etc/zebra/zebra.conf」，不像平常設定其他應用程式一樣非得需要使用編輯器（如 vi）來手動編輯不可，直接透過 zebra 的控制介面就可以修改。Zebra 一旦啟動之後就會開始聽取連接埠 2601 的要求，因此只要藉由「telnet」指令連線到本機（localhost）的 2601 連接埠，即可進入 zebra 的控制介面。登入時所需的密碼是「zebra」。

```
~Linux_NAT_zebra 奇科電腦教育
[geego@Linux_NAT_zebra geego]$ telnet localhost 2601
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^'.

Hello, this is zebra (version 0.93b).
Copyright 1996-2002 Kunihiro Ishiguro.

User Access Verification
Password:
Router> sh ver
Zebra 0.93b (i386-redhat-linux).
Copyright 1996-2002, Kunihiro Ishiguro.
Router>
```

如果看到以上畫面，就表示已成功進入 zebra 的命令介面。在命令提示字元下輸入問號「?」或命令「list」，就會列出所有可用的命令清單。如果想要儲存設定值，只須在 enable 模式下鍵入命令「write memory」即可。

```
~Linux_NAT_zebra 奇科電腦教育
Router> list
enable
exit
help
list
quit
show debugging zebra
show history
show interface [IFNAME]
show ip forwarding
show ip route
show ip route <bgp|connected|kernel|ospf|rip|static>
show ip route A.B.C.D
show ip route A.B.C.D/M
show ip route A.B.C.D/M longer-prefixes
show ip route supernets-only
show ipv6 forwarding
show ipv6 route
show ipv6 route <bgp|connected|kernel|ospf6|ripng|static>
show ipv6 route X::X::X
show ipv6 route X::X::X/M
show ipv6 route X::X::X/M longer-prefixes
show memory
show memory all
show memory bgp
show memory lib
show memory ospf
show memory ospf6
show memory rip
show table
show version
terminal length <0-512>
terminal no length
who
Router>
```

接著，開始啟動動態路由。Zebra 支援的路由通訊協定有：RIPv1、RIPv2、RIPng、OSPF、OSPF6、BGP4+ 和 BGP4-，在此選定 RIPv2 當作實驗的路由通訊協定。在 zebra 的套件中有一個名為「ripd」的 daemon，其組態檔為「ripd.conf」，在稍早之前我們已經將「ripd.conf.sample」複製到「/etc/zebra」目錄下，並更名為「ripd.conf」。「ripd」這支 daemon 同時支援 RIPv1 和 RIPv2，因此，我們只需啟動 ripd 即可以開始實驗。

請注意，在啟動 ripd 之前必須先確認 zebra 的 daemon 已經事先啟動了，之後才可以啟動 ripd。ripd 本身也是一個 daemon，其控制介面接收的連接埠為「2602」，登入密碼是「zebra」。enable 模式無須密碼輸入即可登入。

```

~Linux_NAT_zebra_奇科電腦教育
[root@Linux_NAT_zebra root]# service ripd start
啟動 ripd:
[root@Linux_NAT_zebra root]# telnet localhost 2602
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is zebra (version 0.93b).
Copyright 1996-2002 Munehiro Ishiguro.

User Access Verification

Password:
ripd> on
ripd# sh ver
Zebra 0.93b (i386-redhat-linux).
Copyright 1996-2002, Munehiro Ishiguro.
ripd#

```

然後，使用「sh run」命令（原意 show running）來查看設定值為何。

```

~Linux_NAT_zebra_奇科電腦教育
ripd(conf)# end
ripd# sh run
Current configuration:
!
hostname ripd
password zebra
log stdout
!
interface lo
!
interface eth0
!
interface eth1
!
line vty
!
end
ripd#

```

然後，依照以下的步驟啟動 RIPv2：

第 1 步 ■ 啟動 RIP version 2

輸入「conf t」（原意 configure terminal）命令進入設定模式，然後鍵入「router rip」命令即可啟動 RIP。雖然在 zebra 中 RIP 預設的版本就是第二版，但為了保險起見，還是建議輸入「version 2」命令來指定執行 RIPv2 通訊協定。

```

~Linux_NAT_zebra_奇科電腦教育
ripd# conf t
ripd(conf)# router rip
ripd(conf-router)# version 2
ripd(conf-router)# end
ripd# sh run

Current configuration:
!
hostname ripd
password zebra
log stdout
!
interface lo
!
interface eth0
!
interface eth1
!
router rip
!
line vty
!
end
ripd#

```

第 2 步 ■ 指定 RIP 所需交換的路由網路

同上一步驟，先進入「router rip」的命令模式，然後執行「network」命令來指定 RIP 所需交換的網路位址和介面。

```

~Linux_NAT_zebra_奇科電腦教育
ripd# conf t
ripd(conf)# router rip
ripd(conf-router)#
default-information Control distribution of default route
default-metric Set a metric of redistribute routes
distance Administrative distance
distribute-list Filter networks in routing updates
end End current mode and change to enable mode.
exit Exit current mode and down to previous mode
help Description of the interactive help system
list Print command list
neighbor Specify a neighbor router
network Enable routing on an IP network
no Negate a command or set its defaults
offset-list Modify RIP metric
passive-interface Suppress routing updates on an interface
quit Exit current mode and down to previous mode
redistribute Redistribute information from another routing protocol
route RIP static route configuration
timers Adjust routing timers
version Set routing protocol version
write Write running configuration to memory, network, or terminal
ripd(conf-router)#

```

```

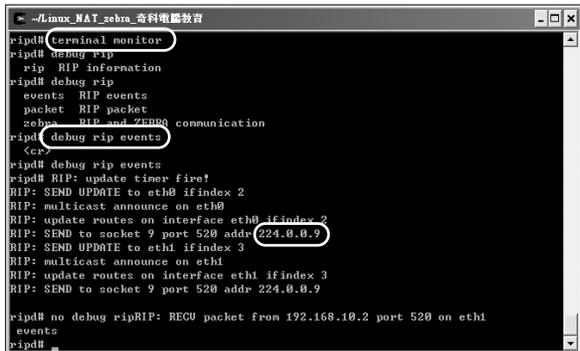
~Linux_NAT_zebra_奇科電腦教育
ripd(conf-router)# network 10.1.32.0/24
ripd(conf-router)# network eth0
ripd(conf-router)# network 192.168.10.0/24
ripd(conf-router)# network eth1
ripd(conf-router)# exit
ripd(conf)# router zebra
ripd(conf-router)# redistribute rip
ripd(conf-router)# end
ripd# sh run

Current configuration:
!
hostname ripd
password zebra
log stdout
!
interface lo
!
interface eth0
!
interface eth1
!
router rip
network 10.1.32.0/24
network 192.168.10.0/24
network eth0
network eth1
!
line vty
!
end
ripd#

```

Special Report 特別企劃

第 3 步 ■ 設定成功之後，就可以使用 ripd 中的「debug」模式來觀察 RIP 的各項行為了。首先使用「terminal monitor」命令啟動，debug 訊息便會顯示在讀者目前使用的終端機上，接著輸入「debug rip events」命令就可以看到 RIP 事件的運作狀況。



```
~Linux_NAT_zebra 奇科電腦教育
ripd# terminal monitor
ripd# debug rip
ripd# debug rip events
ripd# debug rip
events RIP events
packet RIP packet
zebra RIP and ZEBRA communication
ripd# debug rip events
ripd# debug rip events
ripd# debug rip events
RIP: update timer fire!
RIP: SEND UPDATE to eth0 ifindex 2
RIP: multicast announce on eth0
RIP: update routes on interface eth0 ifindex 2
RIP: SEND to socket 9 port 520 addr 224.0.0.9
RIP: SEND UPDATE to eth1 ifindex 3
RIP: multicast announce on eth1
RIP: update routes on interface eth1 ifindex 3
RIP: SEND to socket 9 port 520 addr 224.0.0.9
ripd# no debug ripRIP: REC packet from 192.168.10.2 port 520 on eth1
events
ripd#
```

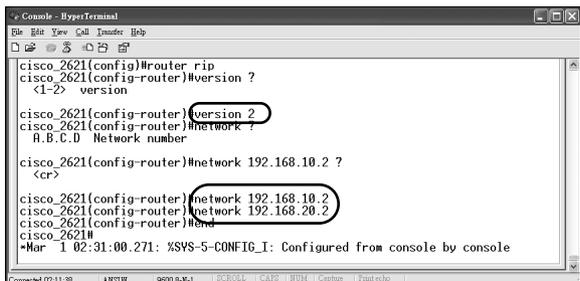
在 debug 訊息中，應該會看到一個 multicast 位址 224.0.0.9，很多路由通訊協定都有規定 multicast 位址來傳送路由訊息。筆者在此特別提醒諸位，網路的運用不僅僅只是設 IP 位址和路由而已，還有許多有趣的知識隱含其中，期待大家去挖掘學習。

剛剛已經完成 Linux_NAT 機器的設定，因此請讀者自行設定好 Linux_router 這台機器，感受一下獨立操作時可能會遇到的問題和挫折。

Cisco 2621 路由器的設定

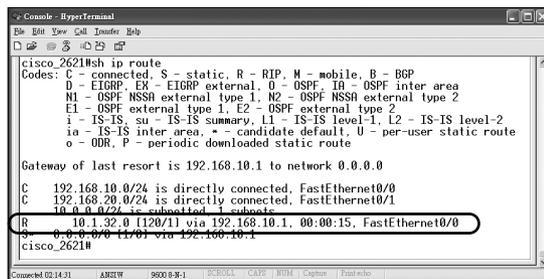
相較於設定 Linux 上的 RIP 路由通訊協定，Cisco 路由器上的設定相對簡單了許多。再者，經過剛剛在 zebra 上練習過一連串の設定命令之後，再設定 Cisco 路由器就不會陌生了，其中的操作過程簡單介紹如下：

第 1 步 ■ 首先，啟動 RIP 路由通訊協定。



```
~Linux_NAT_zebra 奇科電腦教育
cisco_2621(config)#router rip
cisco_2621(config-router)#version ?
<1-2> version
cisco_2621(config-router)#version 2
cisco_2621(config-router)#network ?
A.B.C.D Network number
cisco_2621(config-router)#network 192.168.10.2 ?
<cr>
cisco_2621(config-router)#network 192.168.10.2
cisco_2621(config-router)#network 192.168.20.2
cisco_2621(config-router)#end
cisco_2621#
-Mar 1 02:31:00.271: %SYS-5-CONF1G_I: Configured from console by console
```

第 2 步 ■ 檢視路由表是否有以「R」為開頭的路由規則。如果有的話，表示 Cisco 的 2621 埠已經成功地經由 RIP 路由通訊協定與 Linux_NAT 機器交換路由資訊了。



```
~Linux_NAT_zebra 奇科電腦教育
cisco_2621#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
I - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route

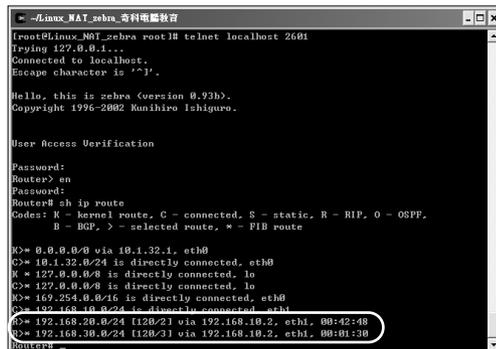
Gateway of last resort is 192.168.10.1 to network 0.0.0.0

C 192.168.10.0/24 is directly connected, FastEthernet0/0
C 192.168.20.0/24 is directly connected, FastEthernet0/1
R 10.0.0.0/24 is subnetted, 1 subnets
R 10.1.32.0 [20/1] via 192.168.10.1, 00:00:15, FastEthernet0/0
R 0.0.0.0/0 [1/0] via 192.168.10.1
```

從圖片中可以看出 Cisco_2621 已經透過 RIP 獲得 Linux_NAT 機器上「10.1.32.0」這筆路由規則，並且將它加入到 Cisco 2621 自己的路由表中。

實驗應有結果

現在開始查看在所有機器上的路由表，與「實驗環境介紹」一節中所列出的各機器的路由表有何差異。首先，查看 Linux_NAT 機器。



```
~Linux_NAT_zebra 奇科電腦教育
[root@Linux_NAT_zebra root]# telnet localhost 2601
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

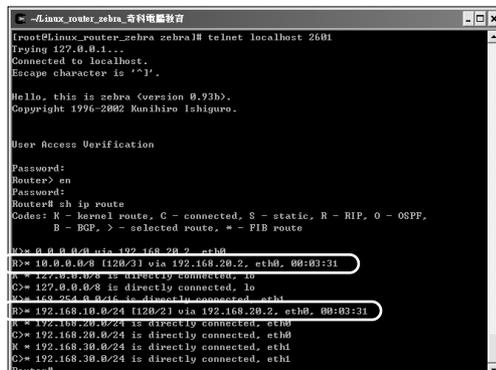
Hello, this is zebra (version 0.93b).
Copyright 1996-2002 Kunihiro Ishiguro.

User Access Verification

Password:
Router> en
Password:
Router# sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
B - BGP, > - selected route, * - FIB route

R> 0.0.0.0 via 10.1.32.1, eth0
C> 192.168.10.0/24 is directly connected, eth0
N * 127.0.0.0 is directly connected, lo
C> 127.0.0.0/8 is directly connected, lo
C> 169.254.0.0/16 is directly connected, eth0
C> 192.168.10.0/24 is directly connected, eth0
R> 192.168.20.0/24 [120/2] via 192.168.10.2, eth1, 00:42:48
R> 192.168.30.0/24 [120/3] via 192.168.10.2, eth1, 00:01:30
Router#
```

接著，檢查 Linux_router 機器上的結果。



```
~Linux_router_zebra 奇科電腦教育
[root@Linux_router_zebra zebra]# telnet localhost 2601
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is zebra (version 0.93b).
Copyright 1996-2002 Kunihiro Ishiguro.

User Access Verification

Password:
Router> en
Password:
Router# sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
B - BGP, > - selected route, * - FIB route

R> 0.0.0.0 via 192.168.20.2, eth0
R> 10.0.0.0 [120/3] via 192.168.20.2, eth0, 00:03:31
C> 127.0.0.0 is directly connected, lo
C> 127.0.0.0/8 is directly connected, lo
C> 169.254.0.0/16 is directly connected, eth0
C> 192.168.10.0/24 is directly connected, eth0
R> 192.168.10.0/24 [120/2] via 192.168.20.2, eth0, 00:03:31
R> 192.168.20.0/24 is directly connected, eth0
C> 192.168.20.0/24 is directly connected, eth0
N * 192.168.30.0/24 is directly connected, eth1
C> 192.168.30.0/24 is directly connected, eth1
Router#
```

最後，查詢 Cisco 2621 路由器上的結果。

```

Cisco 2621#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        I - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route

Gateway of last resort is 192.168.10.1 to network 0.0.0.0

R    192.168.30.0/24 [120/1] via 192.168.20.1, 00:00:22, FastEthernet0/1
C    192.168.30.0/24 is directly connected, FastEthernet0/0
C    192.168.20.0/24 is directly connected, FastEthernet0/1
O    10.0.0.0/24 [0/0] via 192.168.10.1, 00:00:00, FastEthernet0/0
R    10.1.32.0 [120/1] via 192.168.10.1, 00:00:34, FastEthernet0/0
S    0.0.0.0/0 [1/0] via 192.168.10.1
Cisco 2621#

```

從以上幾張圖片的顯示結果發現，所有的機器上都有 RIP 加入的路由規則（也就是以 R 為開頭的路由規則），因此 Linux_NAT 原本 ping 不到的 192.168.20.1 和 192.168.30.1，現在就可以 ping 到了。

```

root@Linux_NAT_zebra root# ping 192.168.20.1 -c 5
PING 192.168.20.1 (192.168.20.1) 56(84) bytes of data:
64 bytes from 192.168.20.1: icmp_seq=1 ttl=63 time=0.525 ms
64 bytes from 192.168.20.1: icmp_seq=2 ttl=63 time=0.511 ms
64 bytes from 192.168.20.1: icmp_seq=3 ttl=63 time=0.537 ms
64 bytes from 192.168.20.1: icmp_seq=4 ttl=63 time=0.489 ms
64 bytes from 192.168.20.1: icmp_seq=5 ttl=63 time=0.488 ms

--- 192.168.20.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4000ms
rtt min/avg/max/mdev = 0.488/0.516/0.537/0.019 ms

root@Linux_NAT_zebra root# ping 192.168.30.1 -c 5
PING 192.168.30.1 (192.168.30.1) 56(84) bytes of data:
64 bytes from 192.168.30.1: icmp_seq=1 ttl=63 time=0.630 ms
64 bytes from 192.168.30.1: icmp_seq=2 ttl=63 time=0.489 ms
64 bytes from 192.168.30.1: icmp_seq=3 ttl=63 time=0.480 ms
64 bytes from 192.168.30.1: icmp_seq=4 ttl=63 time=0.523 ms
64 bytes from 192.168.30.1: icmp_seq=5 ttl=63 time=0.510 ms

--- 192.168.30.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3996ms
rtt min/avg/max/mdev = 0.480/0.526/0.630/0.057 ms
root@Linux_NAT_zebra root#

```

而 Linux_router，現在也能夠 ping 到位於 Internet 上的機器了。

```

root@Linux_router_zebra zebra# ping 168.95.1.1 -c 5
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=248 time=47.0 ms
64 bytes from 168.95.1.1: icmp_seq=2 ttl=248 time=40.9 ms
64 bytes from 168.95.1.1: icmp_seq=3 ttl=248 time=41.7 ms
64 bytes from 168.95.1.1: icmp_seq=4 ttl=248 time=41.9 ms
64 bytes from 168.95.1.1: icmp_seq=5 ttl=248 time=41.3 ms

--- 168.95.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4045ms
rtt min/avg/max/mdev = 40.902/42.598/47.065/2.262 ms
root@Linux_router_zebra zebra#

```

至此，網路拓撲圖中的春嬌與志明終於可以互通訊息了，實驗預期的結果終於圓滿達成。

經過實際設定 Linux 與 Cisco 設備，我們成功地讓 Cisco 2621 與 Linux 的機器作動態路由 RIP 的路由交換了，省卻了手動設定靜態路由的麻煩。從練習中，不難看出 Linux 高度且全面化地支援各種路由通訊協定。礙於篇幅有限，筆者此次還沒有機會帶大家實驗 OSPF、BGP 等進階的路由設定，不過要將這些進階的路由觀念和設定交代清楚很簡單。對一般的 Linux 使用者來說，想在短時間內融會貫通也不是件簡單的事。

Linux 本身就是一個相當完整的網路設備平台，在業界經常可以看到以 Linux 為平台而衍生出來的產品。有些新創立的公司（Startups）想出非常獨特的點子，藉其強大的工程技術背景採用 Linux 平台來設計研發產品，以最快的速度配合行銷上的市場區隔，創造出符合企業和客戶所需要的設備，因此 Linux 平台的使用在 Time to Market 的層面上功不可沒。為數不少的 Startup 公司，藉此成功地進入美國股票市場，短時間立即致富，享受成功的榮耀。

作者簡介

奇科電腦教育資深顧問 Ben 哥，擁有物理學士以及美國 Santa Clara Univ. 電腦工程碩士學位，有豐富的 UNIX 系統管理、程式設計、網路及深厚的教育經驗，目前擔任資深網路安全顧問一職，專門替國內外各大企業、政府提供網路安全諮詢及設計。

如果您有任何技術上的問題，歡迎您隨時 Bne 哥連絡：

E-mail : ask@geego.com.tw