

利用 VLAN protocol 技術建立 Linux 和
Cisco 設備之間的完善互動

VLAN protocol 與 Cisco 設備的 介接關係與設定

對於一個剛跨進 IT 的領域的初學者 (rookies)，需要學習的領域包括網路、系統和程式設計，而且每一個領域都必須花費相當多的時間去熟悉。其實在 IT 業界中的面試分類，通常也是根據這三類來應徵人員。在業界待了十幾年的筆者，往往發現熟悉網路的工程師不擅長系統和程式設計，瞭解系統的管理員不諳網路與程式設計，而程式設計高手對網路及系統生疏。究其原因，在於一頭栽進某項工作之後，如果想要轉換到另一個截然不同的領域時，幾乎就是重新開始，有多少人能夠承受原本的高薪降為低薪、原本受人推崇的經驗變得一文不值呢？因此，轉換跑道的意願就逐漸降低，甚至打消念頭。

不過，如果在 IT 業界想要往上爬的話，深入了解網

路、系統、程式設計三項領域，是不可或缺的先決條件。有鑒於在台灣對於網路和 Linux 深度結合的文章並不多，因此 Ben 哥以及奇科電腦教育的學員們特別以上課內容為基礎並輔以圖文表現的方式，針對 VLAN 在 Linux 設備上的運用加以解說，務求各位讀者完全了解其中的設定方式，並講解 VLAN 與 Cisco 設備的連結方式與技巧。

本技術文件實驗所需的設備清單如下：

- ◆Linux 機器乙台（具備 100Mb/s 以上的乙太網路卡）
- ◆Cisco 2950 系列第二層交換器
- ◆乙太網路線乙條

何謂 VLAN (虛擬區域網路)

VLAN 就是能夠在乙太網路交換器、第三層交換器 (Layer 3 Switch)、路由器或其他支援 VLAN 的設備中，區隔區域網路的一種設定。以 IEEE 的定義來解釋 VLAN，或許過於生硬且不易了解，以下透過較為生活化的方式進行解說：

讀者們早期做網路連結時，必定都用過集線器 (HUB)。集線器的功能是傳遞或加強電氣訊號，是以 OSI 七層定義中的第一層 (物理層) 為基礎所用的設備，現在已經鮮少出現在市面上，取而代之的是所謂的集線交換器 (SWITCH HUB)，也就是最基本的交換器。為什麼叫 SWITCH HUB 呢？其實這和技術一點關係也沒有，純粹是行銷上的一種名詞定義。那 SWITCH 跟 HUB 又有什麼差異呢？HUB 不用多做解釋，因為剛剛已經做了最簡單的定義，而 SWITCH，可解析 OSI 七層中第二層 (資料鏈結層) 的資料，簡單來說，就是能夠學習和記憶連接機器的 MAC (Media Access Control) 位址，根據 MAC 位址來過濾封包的目的 (Forwarding/Filtering)，進而提升連接機器之間的傳輸效率。

各位讀者如果曾經使用過以上的產品，會發現每一台連接的機器都必須在同一個區域網路 (network) 內，才能進行資料交換，如果想要讓一台 HUB 或是 SWITCH HUB 能夠擁有兩個或兩個以上完全獨立的區域網路，是辦不到的。因此一般的解決方案是，使用兩個或是兩個以上的 HUB 或 SWITCH HUB 來解決。但如果改用支援 VLAN 的交換器，便可以只使用一台設備區隔多個區域網路。

舉例來說，Cisco 的小型交換器——2950、3550、3650、3750 等系列的交換器，每一台至少都提供了 50 個以上的連接埠，而每個連接埠都可以設定成不同的區域網路。在接下來的例子裡就表示有 50 個以上的 VLAN 存在於交換器內，藉由 VLAN 的標準與不同廠牌的交換器，以 trunk 達成資料交換。



▲ Linux 主機 I (個人 PC，內建 100Mb/s 速率以上的乙太網路卡)



▲ Linux 主機 II (機櫃中的 Linux 主機，具備 100Mb/s 速率以上的乙太網路卡)



▲ Cisco 2950 系列第二層交換器



▲ 乙太網路線

認識 802.1Q

IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks 文件編號為 802.1Q-1998，於西元 1998 年 12 月 8 日定稿，明確地定義了「虛擬橋接區域網路」(Virtual Bridged LANs) 的架構以及所提供的服務，同時詳細說明了服務所需要的其他協定 (protocols) 和演算法 (algorithms)。

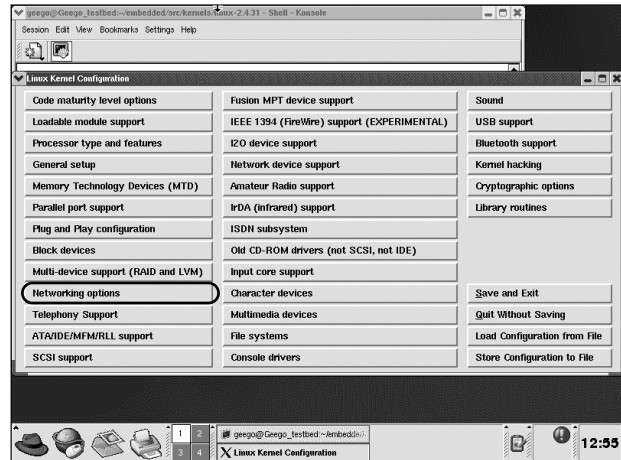
在 IT 業界裡，經常使用專有名詞 802.1Q 和 VLAN 協定，來代表 VLAN 的功能。因此，當有人談到 802.1Q 時指的是 VLAN，提到 VLAN 就代表 802.1Q。基本上，802.1Q 最大的作用是，在乙太網路框架 (Frame) 的表頭上加上一個標記 (Tag)，藉此分辨來自不同 VLAN 的封包。

準備 Linux 的設備

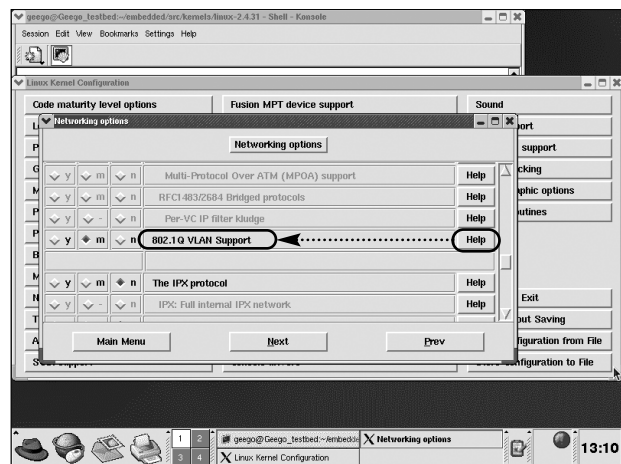
VLAN 在 IT 業界的應用極為廣泛，幾乎所有中高級網路或是電腦設備都提供該項功能。

除了使用硬體 ASIC (Application-Specific Integrated Circuit) 支援 VLAN 功能的設備以外，VLAN 亦可以透過軟體的方式安裝在作業系統內，讓不同的作業系統都能擁有 VLAN 的功能。例如，在微軟視窗系統上，Intel 網路卡的驅動程式就可以支援 VLAN，如果是 Linux 系統，只要使用的 Linux 系統核心支援 VLAN，那麼任何一款網路卡都可以使用。

接下來，讓我們看看 VLAN 的功能位於系統核心選項的何處。以下是在 Linux 核心 2.4.31 版本下執行「make xconfig」命令後的選項畫面。



VLAN 的設定選項位於 [Networking options] 控制項目內，點選進入後，將捲軸往下拉，就可以看到 802.1Q VLAN 選項。請在實驗之前先點選 [Help]，概略了解一下 VLAN 的相關訊息來源，以及設定 VLAN 時所需的命令設定。



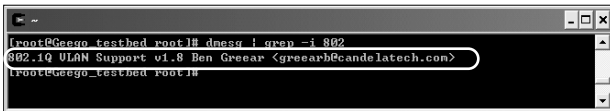
接著，將 VLAN 功能設定成為模組。有一點必須注意的是，如果 VLAN 模組沒有載入核心，就無法由「dmesg」指令的輸出訊息，或是從「/var/log/messages」檔案的內容得知該核心是否能夠支援 VLAN 功能。當核心模組編譯完成之後，802.1Q 模組的檔案名稱是「8021q.o」。

說到這裡，各位已經了解使用 VLAN 之前，首先必須確定目前的系統核心是否支援 VLAN 功能，這可以經由兩種方式查證：一是使用「dmesg」命令，二是查看「/proc/net/vlan」目錄是否存在。

dmesg 命令搭配 grep 所得的輸出訊息：

在核心不提供 VLAN 功能的情況下，執行「`dmesg | grep -i 802`」命令是不會出現任何與 802.1Q 相關的輸出。

而當核心支援 VLAN 功能時，則會出現類似以下畫面的輸出訊息。



```
[root@Geego_testbed root]# dmesg | grep -i 802
802.1Q VLAN Support v1.8 Ben Greear <greearb@candelatech.com>
[root@Geego_testbed root]#
```

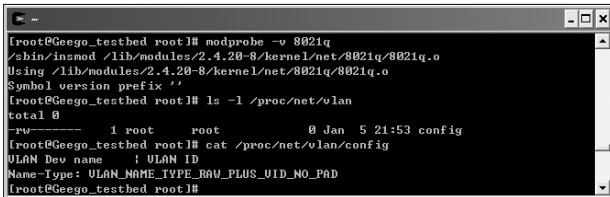
檢查「`/proc/net/vlan`」檔案是否存在：

核心如果沒有提供 VLAN 功能，「`/proc/net/vlan`」目錄是不存在的，如圖所示。



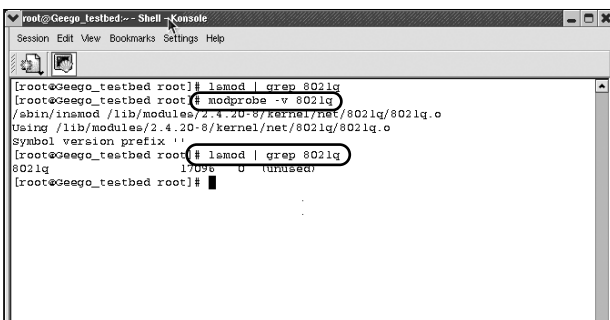
```
[root@Geego_testbed root]# ls -l /proc/net/vlan
ls: /proc/net/vlan: No such file or directory
[root@Geego_testbed root]#
```

若核心支援 VLAN 功能，將會出現下列的輸出訊息。



```
[root@Geego_testbed root]# modprobe -v 8021q
/sbin/insmod /lib/modules/2.4.20-8/kernel/net/8021q/8021q.o
Using /lib/modules/2.4.20-8/kernel/net/8021q/8021q.o
Symbol version prefix ''
[root@Geego_testbed root]# ls -l /proc/net/vlan
total 0
-rw-r----- 1 root root 0 Jan 5 21:53 config
[root@Geego_testbed root]# cat /proc/net/vlan/config
VLAN Dev name 1 VLAN ID
Name-Type: VLAN_NAME_TYPE_RAW_PLUS_VLAN_NO_PAD
[root@Geego_testbed root]#
```

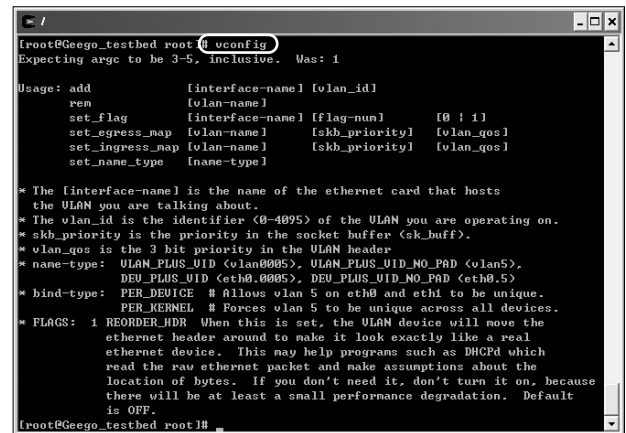
截至目前為止，我們已經學會了如何判斷核心是否支援 VLAN 的功能。接下來，看看該如何將 802.1Q 模組載入核心：使用「`modprobe`」模組命令載入「`802.1q.o`」模組，並且利用「`lsmod`」命令確認模組是否已經載入到核心內，如圖所示。



```
[root@Geego_testbed root]# modprobe 8021q
[root@Geego_testbed root]# modprobe -v 8021q
/sbin/insmod /lib/modules/2.4.20-8/kernel/net/8021q/8021q.o
Using /lib/modules/2.4.20-8/kernel/net/8021q/8021q.o
Symbol version prefix ''
[root@Geego_testbed root]# lsmod | grep 8021q
8021q 17096 0 (unused)
[root@Geego_testbed root]#
```

確定目前使用的 Linux 核心已經支援 VLAN 功能之後，接下來開始設定。一般安裝 Linux 的過程中，都會安裝 VLAN 相關的命令「`vconfig`」。如果沒有的話，可以安裝相關的 rpm 套件，套件名稱通常以「`vconfig`」開頭，例如「`vconfig-1.6-2.i386.rpm`」。

此外，也可以從「<http://www.candelatech.com/~greear/vlan.html>」或「<http://rpm.pbone.net/>」下載原始碼安裝，目前最新的版本是 1.9 版。緊接著，介紹 `vconfig` 的設定選項和使用方法，只須執行「`vconfig`」命令即可顯示相關的解說畫面。



```
[root@Geego_testbed root]# vconfig
Expecting arg to be 3-5, inclusive. Was: 1

Usage: add [interface-name] [vlan-id]
rem [interface-name] [vlan-id]
set_flag [interface-name] [flag-num] [0 | 1]
set_egress_map [vlan-name] [skb_priority] [vlan_qos]
set_ingress_map [vlan-name] [skb_priority] [vlan_qos]
set_name_type [name-type]

* The [interface-name] is the name of the ethernet card that hosts
the VLAN you are talking about.
* The vlan_id is the identifier (0-4095) of the VLAN you are operating on.
* skb_priority is the priority in the socket buffer (sk_buff).
* vlan_qos is the 3 bit priority in the VLAN header
* name-type: VLAN_PLUS_VLAN (vlan0005), VLAN_PLUS_VLAN_NO_PAD (vlan5),
DEU_PLUS_VLAN (eth0.0005), DEU_PLUS_VLAN_NO_PAD (eth0.5)
* bind-type: PER_DEVICE # Allows vlan 5 on eth0 and eth1 to be unique.
PER_KERNEL # Forces vlan 5 to be unique across all devices.
* FLAGS: 1 REORDER_HDR When this is set, the VLAN device will move the
ethernet header around to make it look exactly like a real
ethernet device. This may help programs such as DHCPd which
read the raw ethernet packet and make assumptions about the
location of bytes. If you don't need it, don't turn it on, because
there will be at least a small performance degradation. Default
is OFF.
[root@Geego_testbed root]#
```

VLAN 以數字來做編號，也就是所謂的 VLAN ID，使用的方法非常的簡單，只需一行命令就可以建立一個 VLAN。

VLAN 介面的形式為「<網路卡介面> . <VLAN ID>」，請特別注意，網路卡介面與 VLAN ID 之間有一個句點，有別於 IP Alias 的冒號（:）。如果想在 eth0 介面卡上設立 VLAN 介面的話，以下兩點須特別注意：

1. eth0 介面卡必須是啟用（up）的狀態，也就是說當我們執行「`ifconfig`」命令時可以看到 eth0。如果沒有顯示，可以使用「`/sbin/ifconfig eth0 up`」命令加以啟動。
2. eth0 介面卡不能設定 IP 位址，只有 VLAN 介面卡才可以設定。稍後，將會詳細介紹如何在 VLAN 介面上設定 IP 位址。

藉由以下圖片說明如何在 Linux 的介面卡上設定 VLAN。首先執行「ifconfig eth0 up」命令啟動 eth0 介面卡，然後使用「ifconfig eth0」命令觀察 eth0 是否已經設定 IP 位址。圖中的 eth0 介面卡資訊內沒有出現 IP 位址，接下來使用「vconfig add eth0 6」命令建立一個 VLAN，VLAN ID 設為 6。

然後，利用 ls 指令查看「/proc/net/vlan」目錄下有何變化，發現多了一個名為「eth0.6」的檔案，並且可使用「ifconfig eth0.6」命令來確定 VLAN 介面是否存在。

```
[root@Geego_testbed root]# for ip in 6 10 20 30 40 50 60 70 80 90
> do
> ifconfig eth0.$ip 192.168.$ip.1 up
> ifconfig eth0.$ip ! head -2
> echo "-"
> done
eth0.6 Link encap:Ethernet HWaddr 00:0C:29:24:9E:80
inet addr:192.168.6.1 Bcast:192.168.6.255 Mask:255.255.255.0
-
eth0.10 Link encap:Ethernet HWaddr 00:0C:29:24:9E:80
inet addr:192.168.10.1 Bcast:192.168.10.255 Mask:255.255.255.0
-
eth0.20 Link encap:Ethernet HWaddr 00:0C:29:24:9E:80
inet addr:192.168.20.1 Bcast:192.168.20.255 Mask:255.255.255.0
-
eth0.30 Link encap:Ethernet HWaddr 00:0C:29:24:9E:80
inet addr:192.168.30.1 Bcast:192.168.30.255 Mask:255.255.255.0
-
eth0.40 Link encap:Ethernet HWaddr 00:0C:29:24:9E:80
inet addr:192.168.40.1 Bcast:192.168.40.255 Mask:255.255.255.0
-
eth0.50 Link encap:Ethernet HWaddr 00:0C:29:24:9E:80
inet addr:192.168.50.1 Bcast:192.168.50.255 Mask:255.255.255.0
-
eth0.60 Link encap:Ethernet HWaddr 00:0C:29:24:9E:80
inet addr:192.168.60.1 Bcast:192.168.60.255 Mask:255.255.255.0
```

增加 VLAN 介面之後，接著介紹如何移除 VLAN 介面。以「vconfig」命令搭配「rem」選項，再以 VLAN 介面為其引數即可移除，完整命令為「vconfig rem eth0.6」。

```
[root@Geego_testbed root]# ifconfig eth0 up
[root@Geego_testbed root]# ifconfig eth0
eth0 Link encap:Ethernet HWaddr 00:0C:29:24:9E:80
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:57874 errors:0 dropped:0 overruns:0 frame:0
TX packets:366 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueue:len:100
RX bytes:5415958 (4.8 Mb) TX bytes:24576 (24.0 Kb)
Interrupt:10 Base address:0x10a4

[root@Geego_testbed root]# vconfig add eth0 6
Added VLAN with VID == 6 to IF ==eth0:-
[root@Geego_testbed root]# ls /proc/net/vlan
config eth0.6
[root@Geego_testbed root]# ifconfig eth0.6
eth0.6 Link encap:Ethernet HWaddr 00:0C:29:24:9E:80
BROADCAST MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueue:len:0
RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)
```

```
[root@Geego_testbed root]# vconfig rem eth0.6
Removed VLAN -eth0.6:-
[root@Geego_testbed root]# ls /proc/net/vlan
config eth0.20 eth0.40 eth0.60 eth0.80
eth0.10 eth0.30 eth0.50 eth0.70 eth0.90
[root@Geego_testbed root]#
```

以上就是在 Linux 設備中設定 VLAN 時所需的操作步驟。

一個實體介面卡上可以建立相當多的 VLAN 介面。接著，配合 command script 建立其他的 VLAN。

```
[root@Geego_testbed root]# for vlan in 10 20 30 40 50 60 70 80 90
> do
> vconfig add eth0 $vlan
> done
Added VLAN with VID == 10 to IF ==eth0:-
Added VLAN with VID == 20 to IF ==eth0:-
Added VLAN with VID == 30 to IF ==eth0:-
Added VLAN with VID == 40 to IF ==eth0:-
Added VLAN with VID == 50 to IF ==eth0:-
Added VLAN with VID == 60 to IF ==eth0:-
Added VLAN with VID == 70 to IF ==eth0:-
Added VLAN with VID == 80 to IF ==eth0:-
Added VLAN with VID == 90 to IF ==eth0:-
[root@Geego_testbed root]# ls /proc/net/vlan
config eth0.20 eth0.40 eth0.6 eth0.70 eth0.90
eth0.10 eth0.30 eth0.50 eth0.60 eth0.80
[root@Geego_testbed root]#
```



緊接著，設定每一個 VLAN 介面的 IP 位址。同樣地，使用 command script 來加快設定過程。

Cisco Catalyst 2950/3550 /3650/3750 交換器的設定

剛剛在 Linux 設備上建立了 9 個 VLAN，其 ID 分別是 10、20、30、40、50、60、70、80、90，所以在網路設備上，也必須建立相同的 VLAN ID 才可以和 Linux 的設備作介接，此次實驗採用的設備是 Cisco 2950T-24 的第二層交換器，下頁圖中顯示的是其使用的版本。

```

Cisco Internetwork Operating System Software
IOS (tm) C2950 Software (C2950-160412-M), Version 12.1(13)E1, RELEASE SOFTWARE
(fc1)
Copyright (c) 1986-2003 by Cisco Systems, Inc.
Compiled Tue 04-Mar-03 02:14 by yanah
Image text-base: 0x80010000, data-base: 0x805A8000

ROM: Bootstrap program is CALHOUN boot loader

geego_2950 uptime is 17 weeks, 5 days, 20 hours, 38 minutes
System returned to ROM by power-on
System image file is "flash:/c2950-16q412-mz.121-13.E01.bin"

cisco WS-C2950T-24 (RC32300) processor (revision L0) with 20839K bytes of memory
.
Processor board ID FOC074420LP
Last reset from system-reset
Running Enhanced Image
24 FastEthernet/IEEE 802.3 interface(s)
2 Gigabit Ethernet/IEEE 802.3 interface(s)

32K bytes of Flash-simulated non-volatile configuration memory.
Base ethernet MAC Address: 00:0E:38:80:24:C0
Motherboard assembly number: 73-6114-09
--More--
    
```

在這裡順道說明一下，Cisco 絕大部分的交換器和路由器所使用的作業系統稱為 IOS (Internetwork Operating System)，此為 Cisco 設備的核心價值所在，Cisco 生產的設備之所以能在市場擁有高達 85% 以上的佔有率，IOS 功不可沒，這也清楚說明了程式設計和系統調校的確非常重要。

接下來示範如何在 IOS 中建立 VLAN。首先必須進入「enable」模式，在「enable」模式下才能取得設定的基本權限。先查看目前這個交換器內有哪些既有的 VLAN，然後進入 vlan database 建立想要的 VLAN。

```

geego_2950#enable
Password:
geego_2950#show vlan
    
```

VLAN Name	Status	Ports
1 default	active	Fa0/21, Gi0/1, Gi0/2
6 VLAN0006	active	Fa0/2
31 VLAN0031	active	
32 VLAN0032	active	
33 VLAN0033	active	Fa0/4, Fa0/5, Fa0/6, Fa0/7, Fa0/8, Fa0/9, Fa0/10, Fa0/11, Fa0/12, Fa0/13, Fa0/14, Fa0/15, Fa0/16, Fa0/17, Fa0/18, Fa0/19, Fa0/20, Fa0/22, Fa0/24
66 VLAN0066	active	
1002 fddi-default	active	
1003 token-ring-default	active	
1004 fddinet-default	active	
1005 trnet-default	active	

VLAN Type	SAID	MTU	Parent	RingNo	BridgeNo	Stp	BrdgMode	Trans1	Trans2
1	enet 100001	1500	-	-	-	-	-	0	0
6	enet 100006	1500	-	-	-	-	-	0	0
31	enet 100031	1500	-	-	-	-	-	0	0
32	enet 100032	1500	-	-	-	-	-	0	0
33	enet 100033	1500	-	-	-	-	-	0	0
66	enet 100066	1500	-	-	-	-	-	0	0
1002	fddi 101002	1500	-	-	-	-	-	0	0
1003	tr 101003	1500	-	-	-	-	srb	0	0
1004	fddinet 101004	1500	-	-	-	-	ieee	0	0
1005	trnet 101005	1500	-	-	-	-	ibm	0	0

```

Remote SPAN VLANs:

Primary Secondary Type           Ports
    
```

以上顯示的 IOS 命令輸出訊息中，並沒有發現在 Linux 設備上所建立的 VLAN ID，所以必須進入 IOS 的 vlan database 建立這些 VLAN，然後執行「exit」命令離開 vlan database，如下圖所示。

```

geego_2950#vlan database
geego_2950(vlan)#?
VLAN database editing buffer manipulation commands:
abort  Exit mode without applying the changes
apply  Apply current changes and bump revision number
exit   Apply changes, bump revision number, and exit mode
no     Negate a command or set its defaults
reset  Abandon current changes and reload current database
show   Show database information
vlan  Add, delete, or modify values associated with a single VLAN
vtp    Perform VTP administrative functions.

geego_2950(vlan)#vlan 10
VLAN 10 added:
  Name: VLAN0010
geego_2950(vlan)#vlan 20
VLAN 20 added:
  Name: VLAN0020
geego_2950(vlan)#vlan 30
VLAN 30 added:
  Name: VLAN0030
geego_2950(vlan)#vlan 40
VLAN 40 added:
  Name: VLAN0040
geego_2950(vlan)#vlan 50
VLAN 50 added:
  Name: VLAN0050
geego_2950(vlan)#vlan 60
VLAN 60 added:
  Name: VLAN0060
geego_2950(vlan)#vlan 70
VLAN 70 added:
  Name: VLAN0070
geego_2950(vlan)#vlan 80
VLAN 80 added:
  Name: VLAN0080
geego_2950(vlan)#vlan 90
VLAN 90 added:
  Name: VLAN0090
geego_2950(vlan)#exit
APPLY completed.
Exiting...
geego_2950#
    
```

想要讓 Cisco 交換器與 Linux 設備作介接，必須先選定一個連接埠。在目前這個交換器上，FastEthernet 0/22 並未被使用，所以就利用這個連接埠與 Linux 設備上的介面卡作介接。先看看 FastEthernet 0/22 的狀態如何。各位應該沒想到一個簡單的連接埠竟然有如此多的屬性和資訊可以參考，這也正反映了普通和高規格的網路設備之間，為何有驚人價差的原因了。

```

geego_2950#show interfaces fastEthernet 0/22
FastEthernet0/22 is down, line protocol is down (notconnect)
Hardware is Fast Ethernet, address is 000e.388a.24d6 (bia 000e.388a.24d6)
MTU 1500 bytes, BW 100000 Kbit, DLY 1000 usec,
  reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
Auto-duplex, Auto-speed
input flow-control is off, output flow-control is off
ARP type: ARPA, ARP Timeout 04:00:00
Last input 3w0d, output 3w0d, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
  5 minute input rate: 0 bits/sec, 0 packets/sec
  5 minute output rate: 0 bits/sec, 0 packets/sec
19859 packets input, 1986525 bytes, 0 no buffer
Received 2849 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
0 watchdog, 2737 multicast, 0 pause input
0 input packets with dribble condition detected
113884 packets output, 9155131 bytes, 0 underruns
0 output errors, 1 collisions, 2 interface resets
0 babbles, 0 late collision, 4 deferred
0 lost carrier, 0 no carrier, 0 PAUSE output
0 output buffer failures, 0 output buffers swapped out
geego_2950#
    
```

現在，進入設定模式。可以在 enable 模式下執行 IOS 的「configure terminal」命令來進入設定模式。然後使用「interface FastEthernet 0/22」命令進入 FastEthernet 0/22 的介面設定模式，這裡又可以見到單一連接埠可以設定的眾多命令，一整頁都顯示不完。

```

geego_2950(Config)#configure terminal
Enter configuration commands one per line. End with CNTL/Z.
geego_2950(Config)#interface FastEthernet 0/22
geego_2950(Config-if)#
Interface configuration commands:
  arp          Set arp type (arpa, probe, snap) or timeout
  auto         Configure Automation
  bandwidth    Set bandwidth informational parameter
  carrier-delay Specify delay for interface transitions
  cdp          CDP interface subcommands
  channel-group Etherchannel/port bundling configuration
  channel-protocol Select the channel protocol (LACP, PAgP)
  default      Set a command to its defaults
  delay        Specify interface throughput delay
  description  Interface specific description
  dot1x        IEEE 802.1X subsystem
  duplex       Configure duplex operation.
  exit         Exit from interface configuration mode
  help        Description of the interactive help system
  hold-queue   Set hold queue depth
  ip           Interface Internet Protocol config commands
  keepalive    Enable keepalive
  lacp         LACP interface subcommands
  load-interval Specify interval for load calculation for an interface
  logging      Configure logging for interface
  mac          MAC interface commands
  mac-address  Manually set interface MAC address
  mls          mls interface commands
  mvr         MVR per port configuration
  no          Negate a command or set its defaults
  ntp         Configure NTP
  pagp        PAgP interface subcommands
  random-detect Enable Weighted Random Early Detection (WRED) on an
  rmon        Interface
  service-policy Configure QoS Service Policy
  shutdown    Shutdown the selected interface
  snmp        Modify SNMP interface parameters
  spanning-tree Spanning Tree Subsystem
  speed        Configure speed operation.
  storm-control storm configuration
  switchport  Set switching mode characteristics
  timeout     Define timeout values for this interface

```

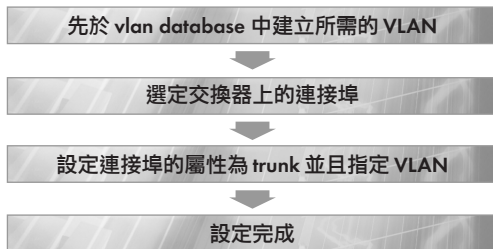
最後，將此連接埠設定為 802.1Q 的模式，如下圖所示。

```

geego_2950(Config-if)#switchport mode trunk
geego_2950(Config-if)#switchport trunk allowed vlan 10,20,30,40,50,60,70,80,90
geego_2950(Config-if)#end
geego_2950#

```

以上就是在 Cisco 2950 交換器中所做的全部設定。如果將剛剛的設定以流程圖表示，內容如下：



經過以上分別針對 Linux 與 Cisco 設備的設定示範，我們讓 Cisco 2950 交換器的 FastEthernet 0/22 連接埠和 Linux 的介面卡 (eth0) 連接，並且以 802.1Q 的通訊協定進行資料傳輸，我們不難看出 Cisco 有 UNIX 的影子存在，筆者接觸過的知名設備不下百種，所有高階設備的作業系統皆以 UNIX 架構為基礎，為的就是穩定和效能。UNIX 好比是科學界中的物理一樣，為一切科學的發展基礎。網路設備的衍生，就如同電機學科是從物理的理論基礎出發，再按其應用方向（電路和機械）深度發展成為一門單獨的學科一樣。再者，分析目前社會發展的趨勢方向，可以想見未來幾年必定迫切需要具備跨領域多項思維能力的人才來處理更深層且更多元化的問題。因此，強烈建議已經熟悉系統的人，雖然你們已經比大多數的人成功了一半，但絕不要因此而停滯不前，趕緊配合原有的優勢拓展個人未來的發展空間，向網路邁進吧！

筆記欄