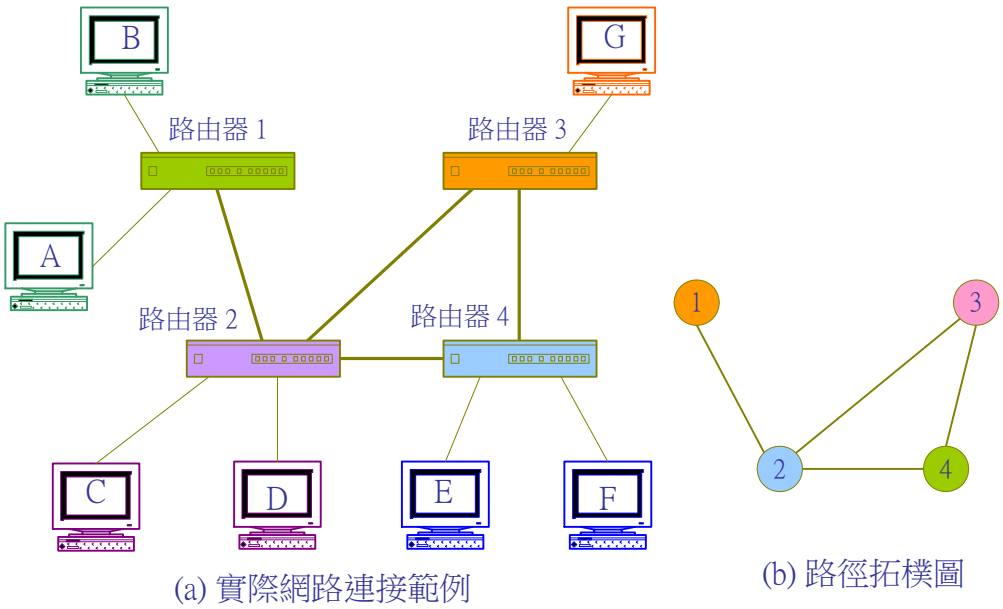
# 第六章 IP Routing 協定

## 6-1 IP Routing 簡介

IP 協定最主要的功能就是如何在網路叢林中找到一條路徑可以到達目的地，這就是『**路徑選擇**』( **Routing** ) 技術。路徑選擇功能主要是由網路上所有『**路由器**』( **Router** ) 共同來達成 ( 有些主機也具有此功能 )，因此，路徑選擇是一種分散式處理方式。它的工作原理是當封包由路由器的某一個埠進來時，路由器依照本身的『**路由表**』( **Routing Table** ) 查出應該往哪一個埠送出，而轉送到其它路由器，再由下一個路由器決定路徑傳送。至於下一個路徑是否可到達目的，就不是本身可以管轄的範圍，而是由下一個路徑的路由器負責，因此又稱為『**下一跳躍路徑選擇**』( **Next-hop Routing** )。

圖 6-1 (a) 為一般網路架構圖，我們將其轉換為路徑拓樸圖，如圖 6-1 (b) 所示。在每一個路由器上都有建立路由表 ( 如 表 6-1 )，對於每一個目的地都有標明下一個路由器位置 ( 下一站 )。如工作站 A 欲傳送封包給工作站 F，當它的封包進入路由器 1，路由器 1 由它的路由表中查詢到應往下一個路由器 2 傳送。當這個封包進入路由器 2 後，由路由器 2 轉送到路由器 4。再由路由器 4 將封包傳送給工作站 F。




(a) 實際網路連接範例　　(b) 路徑拓樸圖

**圖 6-1 路徑選擇範例**

**表 6-1 路由表範例 ( 如圖 6-1 )**

| 路由器 1 | | 路由器 2 | | 路由器 3 | | 路由器 4 | |
|---|---|---|---|---|---|---|---|
| 目的地 | 下一站 | 目的地 | 下一站 | 目的地 | 下一站 | 目的地 | 下一站 |
| 1 | | 1 | 1 | 1 | 2 | 1 | 2 |
| 2 | 2 | 2 | | 2 | 2 | 2 | 2 |
| 3 | 2 | 3 | 3 | 3 | | 3 | 3 |
| 4 | 2 | 4 | 4 | 4 | 4 | 4 | |

對於路由表的建立主要有兩大類：

**(1) 靜態路徑選擇（Static Routing）**技術：表示路由表是靜態的，不會隨時改變，一般有下列兩種方法：

- 固定路徑選擇法（Fixed Routing）
- 熱馬鈴薯法（Hot-potato）

**(2) 動態路徑選擇（Dynamic Routing）技術**：路由表可能經由路由器之間隨時交換訊息，計算出新的路由表，目前較常用的路徑選擇技術有下列兩種：

- 鏈路狀態路徑選擇（Link-State Routing, LS Routing）技術
- 距離向量路徑選擇（Distance Vector Routing, DV routing）技術

# 6-2 Static Routing 技術

**『靜態路徑選擇』(Static Routing)**表示路由表的內容是靜態的，它不會隨網路狀態隨時變更，也就是說，路由器之間並不交換訊息來探討網路狀態而隨時變更路由表。一般靜態路徑選擇使用於較小區域網路範圍內，而系統管理者對網路狀況較容易掌控時所採用。但話說回來，絕大部份的網路工程師所面臨的問題都是靜態路徑選擇，除非是大都會、或是區域網路環境較大的網路工程師，或許較有機會面臨動態路徑選擇的問題。以下介紹兩種靜態路徑選擇技術：固定路徑選擇法和熱馬鈴薯法。

## 6-2-1 固定路徑選擇法

**『固定路徑選擇』(Fixed Routing)**是利用人工建立之路由表，建立後除非再用人工修改，否則將永遠不會變更。系統管理者利用固定路由表規劃網路架構，而網路中主要的路徑分配是利用固

定路由表來完成。如果想要更改網路型態，除了變更實體連線外，主要還必須設定固定路由表來決定網路上實際的分配。如果僅更改網路實體架構，而沒有重新設定固定路由表的話，網路將會發生嚴重的錯誤，也可能會因此而癱瘓。( 建立路由表方法請參考第九章介紹 )

## 6-2-2 熱馬鈴薯法

**『熱馬鈴薯法』( Hot-potato )** 又稱為洪氾法 ( Flooding )，也是一種靜態演算法。其功能是：當封包由路由器的某一個埠口進入後，該路由器便將它複製成多份，往其它埠口發送，即不管封包的目的位址，收到後就往外丟 ( 好像手拿到熱馬鈴薯，燙到手馬上往外丟 )。在理想狀態之下，至少會有一個封包到達目的地。為了避免封包在網路上永無止境的傳遞，在封包內裝設一個跳躍計數器。封包每經過一個路由器，就將計數器的值減一，如果路由器發現某一封包的記數器的值為 0，便將該封包拋棄而不再發送。一般我們都會預估網路最大的範圍 ( 路由器的數量 )，而取一半路徑的數量作為計數器的基準值。如圖 6-2 所示，封包由路由器 A 進入欲傳送到路由器 C。首先該封包被路由器 A 複製兩份，分別發送到路由器 B 和 E，再由它們繼續往前發送，最後至少會有一個複製封包到達路由器 C。

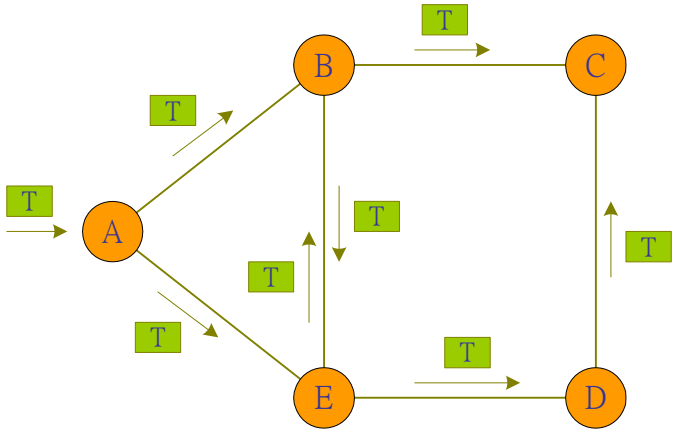



圖 6-2 熱馬鈴薯法

雖然我們用跳躍記數器來限制封包的壽命，但只要發出一個封包便會在網路上產生無數的封包，封包數還會隨著路由器的數量而增加，這種現象稱之為**『封包風暴』( Packet Storm )**。一種修正方法是：每一封包上編有特殊序號，路由器紀錄所經過的封包序號，如果某一封包已被複製轉送過，當它又從另一埠口進入時，便將其拋棄而不再轉送。這樣的話，就可以減少許多重複轉送的機會。但要維護紀錄序列表也是件頭痛的問題，因為當每一封包進入時，都必須搜尋或登錄紀錄表，而且也很難預估同一封包下一次何時會再重複進入。因此，必須再加入登錄時間，登錄時間經過某段時間後便將該紀錄刪除。

另一種修正版本可能較適合，稱之為『選擇性洪氾法』( Selective Flooding )，其功能是：封包進入後，除了搜尋紀錄表外，並非往所有路徑複製轉送，而只發送到比較有可能到達目標位址的路徑上。每一個路由器的埠口可能前往的目標位址，可由人工事先輸入 ( 固定路由 )。除非有特殊情況，否則封包風暴問題已大大改善。熱馬鈴薯方法是使用在廣播訊息較多的網路上，例如，分散式資料庫系統必須隨時廣播更新資料庫訊息。因此，一般使用於較特殊網路，或是小型區域網路上。

## 6-3 Dynamic Routing 簡介

所謂『動態路徑選擇』(Dynamic Routing)？即是『網路上各個路由器隨時互相傳遞網路最新狀態，每個路由器收到相鄰之間路由器的訊息，再依照這些資料建立路由表。』由此可見，動態路由表會隨時依照網路狀態變化中。

在網路大環境隨時變動情況下，使用靜態路由選擇非常不方便，必須針對網路情況，隨時修改路由表。動態路徑選擇會依照網路情況隨時修正路由表，當網路上有任何更動，動態路由選擇器會隨時更新資料，而計算出下一個路徑應該是哪一個路由器的效率最高。雖然動態路由選擇能隨時提供最佳路徑，但為了維持這個功能，必須隨時在網路上收集最新資訊，也是會浪費不少頻寬。所以一般我們在私有網路上儘量使用靜態路由選擇，以加快網路效率，而且系統管理者也非常容易掌握自己網路狀況。對於連結到網路外的公眾網路儘量使用動態路由選擇，網路中隨時改變的確很難掌控。目前所有路由器都具有靜態和動態路由選擇功能，一般系統管理者也可以在主要幹線建立靜態路由選擇，其他就利用動態路徑選擇隨時依照網路情況建立路由表，這樣網路變異性最高，也是最常用的方法。

大型網路中連接許多異質性網路 ( Heterogeneous Network )，動態路由選擇必須由各個路由器之間互相交換訊息，各家廠商所生產的路由器也不盡相同，因此在路由器之中必須有共通的『路徑協定』( Routing Protocol ) 來完成。各種路徑協定都有自己的路徑選擇方法，以下我們介紹兩種目前網路上較常用的動態路徑選擇技術。

## 6-4 Link-Static Routing

『鏈路狀態路徑選擇』( Link-State Routing，LS Routing ) 是屬於動態演算法。路由器必須隨時依照最新消息計算出路由路徑，也隨時更新路由表。它是一種『半集中式』( Quasi-centralized ) 的路徑選擇演譯法( Routing algorithm )。首先每一個路由器必須定期測量它和鄰近路由器之間的費用，這費用可能和佇列延遲、頻寬等因素有關，不同通訊協定都有各自的定義。這費用又稱為『鏈路費

用』( Link Cost )。當每一路由器測出相鄰之間的費用後，定期廣播給其他『所有』路由器，該廣播的訊息又稱為『鏈路狀態』( Link State ) 訊息。同樣的，任何一部路由器也會接收到其他路由器廣播的鏈路狀態，再依照這些訊息計算出到達其他路由器的最短路徑，並建構路由表。路由表上註明欲往哪一個路由器的下一個路由器位置( 如圖 6-1 所示 )。因此在 LS Routing 演譯法下每一個路由器建立路由表的步驟如下：

**(1)** 利用 Hello 封包查詢相鄰路由器

**(2)** 計算鏈路費用

**(3)** 建立鏈路狀態封包並廣播給所有路由器

**(4)** 計算出最短路徑及更新路由表

以下分別說明各步驟的處理程序：

## 6-4-1 利用 Hello 封包查詢相鄰路由器

當路由器啟動後，立即發送 Hello 封包 ( Hello Request ) 給所有相鄰路由器 ( 或定期發送 )。當其它路由器收到 Hello 封包後，也必須立即回覆 Hello 封包 ( Hello Response )，並告知路由器名稱。如圖 6-3 所示，路由器 A 發送 Hello 查詢封包 ( H_Re ) 給相鄰路由器 ( B、C、D )，相鄰路由器也回應 Hello 封包 ( H_Rp ) 給它。路由器也因這樣而知道有哪些路由器和它相鄰。

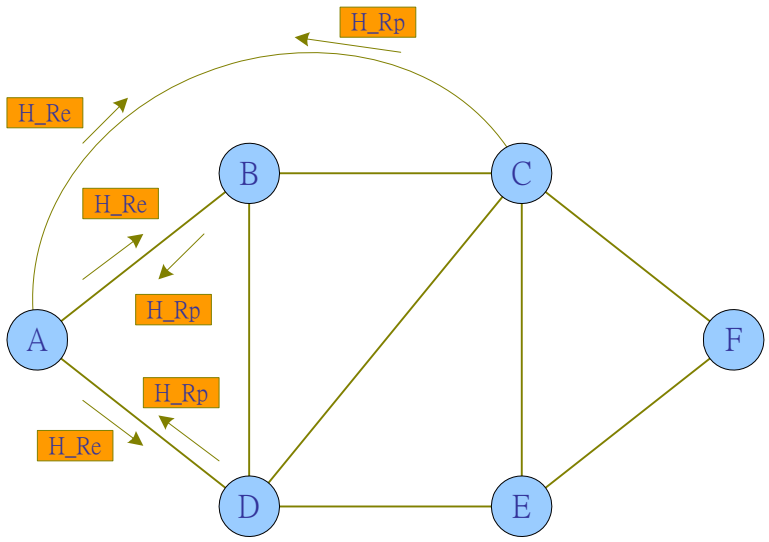



圖 6-3 發送 Hello 封包查詢及回應

## 6-4-2 計算鏈路費用

發送 Hello 之路由器，由相鄰路由器的回應 Hello 封包，而得知它們之間的路徑費用。再由這些訊息計算出到達相鄰路由器之間的鏈路費用。依照各種通訊協定的需求，針對路徑費用的定義也不盡相同，可能採用傳輸延遲、佇列延遲、或頻寬容量等等因素，但最容易取得的是傳輸延遲。當路由器發送查詢 Hello 封包（H_Re）後，到它收到某一路徑回應（H_Rp）的時間，計算之間差異，再取一半的值（除以 2），這就是該路徑的傳輸延遲時間，一般都以毫秒（ms）為單位。在圖 6-4 之中，我們假設所有路由器都經過廣播 Hello 封包後，計算出它們之間的路徑費用（也是鏈路狀態），並假設鏈路兩端所計算費用都相同（如，A → B 和 B → A）。

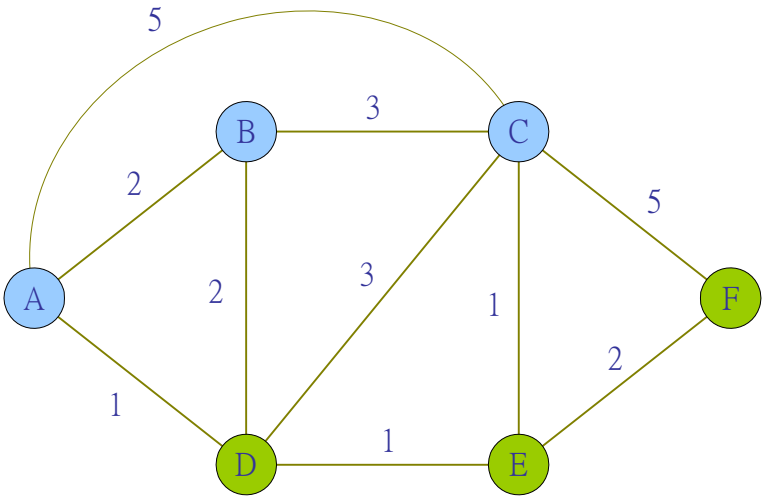



圖 6-4 鏈路狀態範例

## 6-4-3 建立鏈路狀態並廣播給所有路由器

由上一步驟，各路由器已得知相鄰路由器之間的狀態值（如圖 6-4），並各自建立鏈路狀態封包，如圖 6-5 所示。

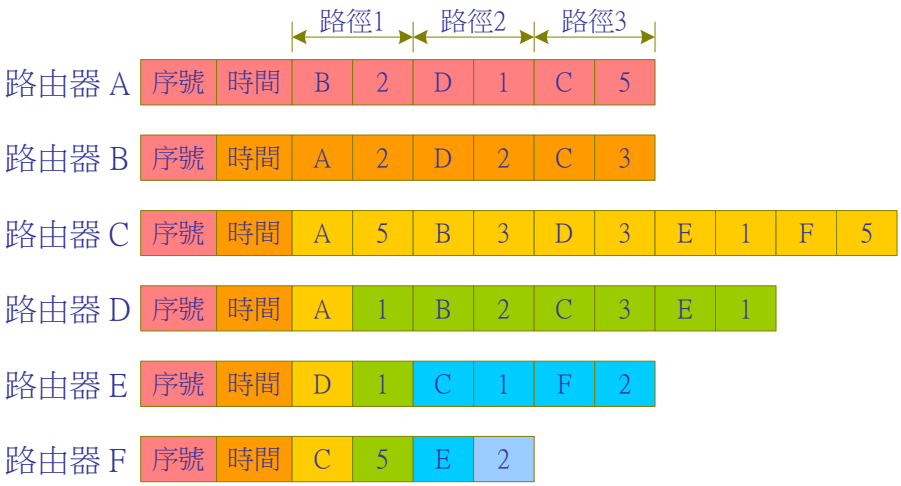


圖 6-5 鏈路狀態封包範例

各路由器再將它的鏈路狀態封包廣播給網路上『所有』路由器。同樣的，每一部路由器也收到所有其他路由器的鏈路狀態封包，再由這些訊息計算出欲往某一路由器的最佳路徑，也建立了路由表。既然任何一部路由器都可由網路收到所有鏈路狀態，並算出自己的路由表，因此我們稱之為『半集中式』的路徑選擇法。

針對廣播鏈路狀態封包可能造成廣播風暴（Broadcast Storm）的問題，其類似熱馬鈴薯方法發送。每一個路由器收到封包後，便往其他路徑複製轉送，才可以將封包廣播到所有路由器上。因此，我們必須在封包上編有封包序號和時間戳記（Time-stamp）。當封包進入路由器後檢查該封包是否來過，如果來過便將其拋棄不再轉送。還有時間戳記是用來更新封包序號紀錄，決定是否可以刪除。

## 6-4-4 計算出最短路徑及更新路由表

當每一路由器收到其他所有路由器的鏈路狀態封包，必須計算出它到達任何一部路由器的最佳路徑。在圖形理論中，有許多尋找最短路徑的演算法，其中較被常用的是 Dijkstra′ s shortest path algorithm。其演譯法如下

## Dijkstra's shortest path algorithm

Define:

N: set of all nodes such that shortest path from source to these nodes is known. N initially is empty.

D(v): cost of known least cost path from source to node v.

c(i, j): cost of link i to j, c(i, j)=∞ if i, j not directly connected.

p(v): previous node (neighbor of v) along shortest path from source to v.

Algorithm:

source node: A

iterative: after k steps know shortest path to nearest k neighbors

**(1)** Initialization:

N = {A}

For all nodes v

If v adjacent to A

Then D(v) = c(A, v)

Else D(v) = ∞

**(2)** Loop

find w not in N such that D(w) is smallest

add w into N

update D(v) for all not in N:

D(v) = min(D(v), D(w)+c(w, v))

Until all nodes in N

## end of algorithm

我們用圖 6-4 為範例，每一路由器所收到的鏈路狀態封包如圖 6-5 所示。又以路由器 A 為例，尋找最短路徑的步驟如圖 6-6 所示。所有路由器演譯法演算後所建立的路由表如表 6-2 所示。

| Step | N | D(B), p(B) | D(C), p(C) | D(D), p(D) | D(E), p(E) | D(F), p(F) |
|---|---|---|---|---|---|---|
| 0 | A | 2, A | 5, A | 1, A | ∞, -- | ∞, -- |
| 1 | AD | 2, A | 4, D | | 2, D | ∞, -- |
| 2 | ADE | 2, A | 3, E | | | 4, E |
| 3 | ADEB | | 3, E | | | 4, E |
| 4 | ADEBC | | | | | 4, E |
| 5 | ADEBCF | | | | | 4, E |

Ex. in Step1 D(C) = D(D) + c(D, C) = 1+ 3 = 4

**圖 6-6 以路由器 A 為範例之演算過程**

**表 6-2 所有路由器之路由表（以圖 6-4 為例）**

| 路由器 A | | 路由器 B | | 路由器 C | | 路由器 D | | 路由器 E | | 路由器 F | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 目的地 | 下一站 | 目的地 | 下一站 | 目的地 | 下一站 | 目的地 | 下一站 | 目的地 | 下一站 | 目的地 | 下一站 |
| A | | A | A | A | D | A | A | A | D | A | E |
| B | B | B | | B | B | B | B | B | D | B | E |
| C | D | C | C | C | | C | E | C | C | C | E |
| D | D | D | D | D | E | D | | D | D | D | E |
| E | D | E | D | E | E | E | E | E | | E | E |
| F | D | F | D | F | E | F | E | F | F | F | |

## 6-4-5 鏈路狀態的異常狀態

鏈路狀態之路徑選擇方法雖然簡單，但會有下列異常狀態：

- **廣播風暴（Broadcast Storm）：**網路上每一個路由器必須隨時計算相鄰之間的鏈路費用，廣播給網路上所有的路由器。如果每次廣播的時間間隔太長，則網路上任何變更將無法即時反映給所有路由器，會造成連結上困難。但廣播間隔時間太短容易造成廣播風暴。而且為了廣播路徑狀態給所有路由器也會佔用不少頻寬。

- **報喜不報憂：**對每個路由器都是主動廣播它所計算的鏈路狀態。任何路由器啟動時都會按時發送鏈路狀態。但當中某一路由器發生故障，它便失去廣播狀態之能力。其它路由器要能偵測出這一條鏈路是故障的，這可能要經過一段不短的時間，在這期間內也容易造成連結上的錯誤。

當然，上列的異常狀況有許多方法來克服，還不至於太困難。但一般來說，任何一個路由器要將它的鏈路狀況廣播給網路上所有路由器，在較大的網路上使用也太不夠經濟，網路上路由器愈多佔用的頻寬也就愈大。

## 6-5 Distance Vector Routing

**『距離向量路徑選擇法』( Distance Vector Routing，DV Routing )** 是一種動態的分散式路徑選擇演算法。由此演算法所實現的通訊協定之下的路由器，它們的路由表是由鄰近相鄰路由器之中共同建立而成，因此稱之為**『分散式』**。網路上每一個路由器都必須維護一個二維的**『向量表』( Vector Table )**，向量表格內紀錄本身路由器到每一個路由器的已知的最佳距離。路由器定期和鄰近路由器交換向量表 ( 並不廣播給所有路由器 ) 來建立路由表。當路由器接收到鄰近傳來的向量表時再修正本身的向量表，向量表的內容就一直修正再傳送，整個網路狀況也就能漸漸地傳遞到每一個路由器上。隨著時間路由器上之路由表的資料漸漸完備，也漸漸能找出最佳路徑。向量表只傳送給相鄰的路由器，對網路的頻寬也佔用較少，也不會造成廣播風暴。

對於向量表中各個欄位，如果使用路由器之間距離的數值，稱之為**『距離向量』( Distance Vector )**。表示兩個路由器之間相距幾個路由器，向量表又稱為**『距離表』( Distance Table )**。但目前使用之 DV routing 並不只估計相鄰之間的距離，我們會將各個路由器之間的佇列延遲、頻寬和網路壅塞情況來計算成為相鄰之間的**『距離費用』( Distance Cost )**。假設路由器已知它相鄰路由器之『距離』，如果距離費用是封包跳躍次數，則其值剛好為 1；如果距離費用為佇列長度，則路由

器僅需計算每個佇列；如果距離費用是延遲時間，則路由器可利用一個特殊的 Echo 封包要求鄰近路由器回應，再計算要求和回應之間時間的差距，再取它的一半值就是延遲時間。

## 6-5-1 距離向量演算法的推演

我們以圖 6-7 為範例來說明路由器之間距離向量的傳遞，和路由表建立的過程。在圖中兩個端點（如，A、B）之間的標示值（如，6），為該兩端點之間的距離費用，其值的來由也許經過：跳躍次數、延遲時間等等因素所計算出來的。並假設距離向量傳遞路徑為：A → B → C → D → E。當距離向量進入某一路由器後，便和本身的路由表搜尋出所有可能到達的路徑，再由新建立的路由表之中尋找出最短路徑。緊接著，又將該最短路徑的路由表傳遞給下一個路由器。一直到最後，觀察路由器 E 的路由表的變化結果，便可瞭解距離向量演算法的運作情形。

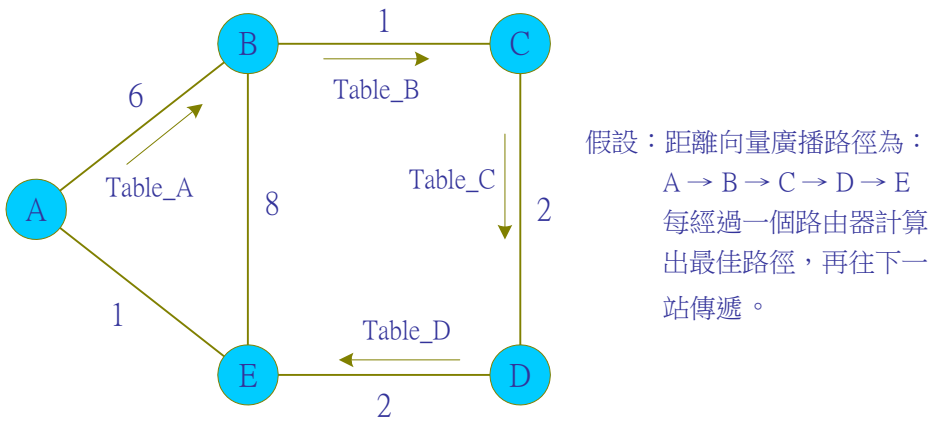



圖 6-7 距離向量演算法範例

圖 6-8 (a) ~ (e) 為各個路由器的起始距離向量表，並假設所有路由器都未和其他路由器交換訊息。針對每一個向量表，我們記錄了：從該路由器到其他路由器所經由不同相鄰路由器的費用。例如 DE(A, B)是從 E 到 A 經由 B 的費用。其他空白部份表示沒有訊息，也可說是路徑費用無限大（∞）。

(a) 路由器 A

| 目的地 \ 經由 | B | E |
|---|---|---|
| B | 6 | |
| C | | |
| D | | |
| E | | 1 |

$D^A(B, B) = 6$
$D^A(E, E) = 1$

(b) 路由器 B

| 目的地 \ 經由 | A | C | E |
|---|---|---|---|
| A | 6 | | |
| C | | 1 | |
| D | | | |
| E | | | 8 |

$D^B(A, A) = 6$
$D^B(C, C) = 1$
$D^B(E, E) = 8$

(c) 路由器 C

| 目的地 \ 經由 | B | D |
|---|---|---|
| A | | |
| B | 1 | |
| D | | 2 |
| E | | |

$D^C(B, B) = 1$
$D^C(D, D) = 2$

圖 6-8 (a) ~ (c) 路由器的起始路由表

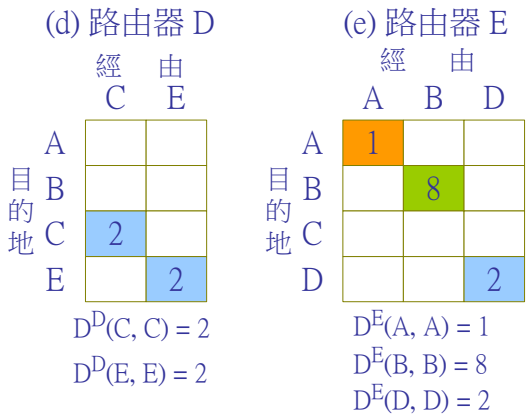



圖 6-8 (d)、(e) 路由器的起始路由表

向量表傳遞後，路由器依照進入的向量表和本身路由表，建構新的路由表，其步驟如下：

**(1)** 路由器( 如 B )首先登錄自己和進入向量表的路由器( 如 A )之間的費用( DB(A, A) = 6 )。

**(2)** 再搜尋向量表 ( 如圖 6-8 (a) ) 中可能到達的端點 ( 其值不是 ∞ )，兩個路徑費用的合如小於表中內所紀錄的值，便取代它。

**(3)** 搜尋索有路徑後，建立新的路由表，再由新的路由向量表找出最短路徑，再將其傳遞給下一個路由器。

我們以圖 6-7 為例子，來推演距離向量演算法建構新路由表的過程，而距離傳遞方向假設只有單一路徑，其方向為： A → B → C → D → E。我們以下列步驟來觀察推演的過程：

## (A) 路由器 A 建構起始路由表

路由器 A 沒有收到其他路由器的向量表，依照自己向鄰近路由器查詢之距離費用( 如 6-8 (a) ) 建構出路由表。如圖 6-9 所示，其中最短路徑是由距離向量表中，查出每列的向量值最低者，並將其建立成路由表。如圖中，它可經由 B 到達 B 的費用是 6( DA(B, B) )為該列最小值；又 DA(E, E) = 1 也是該列最小值。路由器 A 建構路由表後，再將該路由表傳遞給 B ( A → B )，緊接著下一步驟。

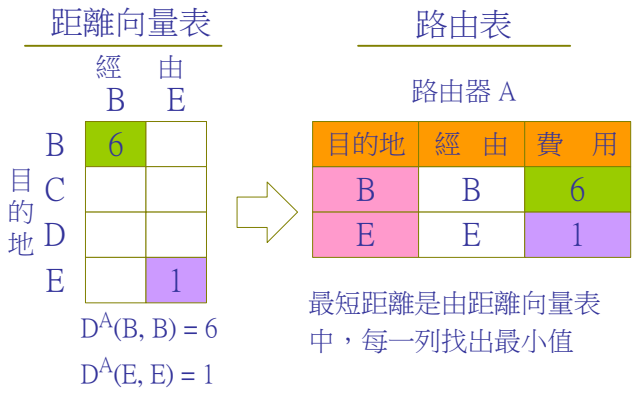



圖 6-9 路由器 A 自行建構路由表

## (B) 路由器 A 傳送給路由器 B ( A → B )

路由器 A 將路由表 ( 圖 6-9 ) 傳遞給路由器 B，B 起始距離向量表如圖 6-8 (b)，再利用這兩個向量表建立出新的距離向量表。首先，B 由路由表中查出它自己經由 A 到 A 的費用 ( DB(A, A) = 6 )。再由 A 的向量表中查詢可能到達的端點 ( B、E )，得知可經由 A 到達 E ( DA(E, E) = 1 )。則將兩個端點路徑費用的和 ( DB(A, A) + DA(E, E) = 7 ) 填入經由 A 到 E 的欄位內。再由表中的每一列的最小值，找到最短路徑 ( 如第四列 )，建構出路由表如圖 6-10 所示。並將路由表傳遞給 C ( B → C )，接下一步驟。

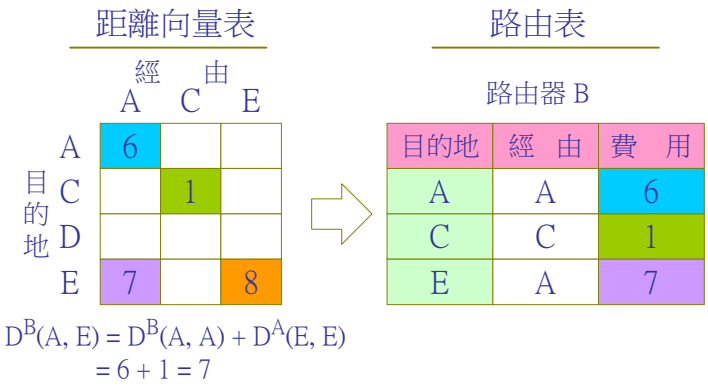



圖 6-10 路由器 B 經過 ( A → B ) 後建立新的路由表

## (C) 路由器 B 傳送給路由器 C ( B → C )

路由器 B 將路由表 ( 圖 6-10 ) 傳遞給路由器 C，C 起始距離向量表如圖 6-8 (c) ，再利用這兩個向量表建立出新的距離向量表如圖 6-11 所示。首先 C 由本身路由表中查詢出到達路由器 B 之間的費用 ( DC(B, B) = 1 )，在搜尋 B 的向量表可能到達的端點 ( A、E )。

**(1)** 到達 A( 經由 B 到達 A ):DC(B, B) + DB(A, A) = 1 +6 = 7，原向量表中的值為無限大( ∞ )，因而取代為 7。

**(2)** 到達 E ( 經由 B 到達 E ) : DC(B, B) + DB(A, E) = 1 +7 = 8，填入表格。

**(3)** 搜尋完後所建立之向量表，再找出最短路徑，建立出新的路由表，如圖 6-11 所示。並將路由表傳遞給 D ( C → D )，接下一步驟。

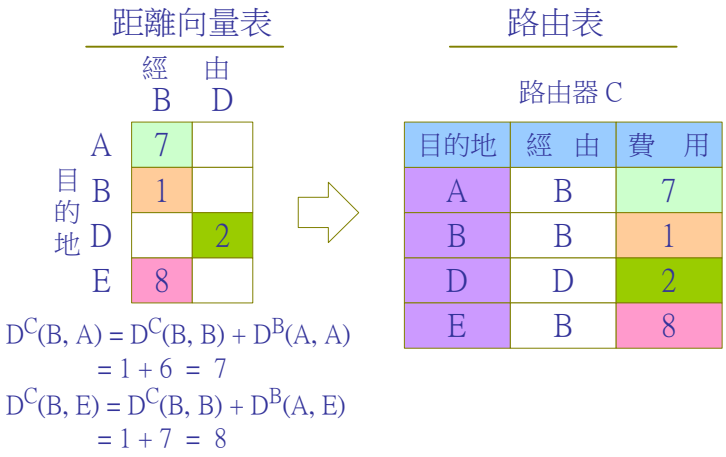



**圖 6-11 路由器 C 經過 ( B → C ) 後建立新的路由表**

## (D) 路由器 C 傳送給路由器 D ( C → D )

路由器 C 將路由表 ( 圖 6-11 ) 傳遞給路由器 D。又 D 的起始距離向量表如圖 6-8 (d) ，再利用這兩個向量表建立出新的距離向量表。而 D 到 C 的費用是 2 ( DD(C, C) )，另由向量表 ( 圖 6-11 ) 中查詢出可經由 C 到達之端點為 A、B、E。搜尋步驟如下：

**(1)** 到達 A：DD(C, A) = DD(C, C) + DC(B, A) = 2 + 7 = 9。

**(2)** 到達 B：DD(C, B) = DD(C, C) + DC(B, B) = 2 + 1 = 3。

**(3)** 到達 E：DD(C, E) = DD(C, C) + DC(B, E) = 2 + 8 = 10。

**(4)** 搜尋完後所建立之向量表及所建構出新的路由表如圖 6-12 所示。再將它的路由表傳遞給 E，緊接下一步驟 ( D → E )。

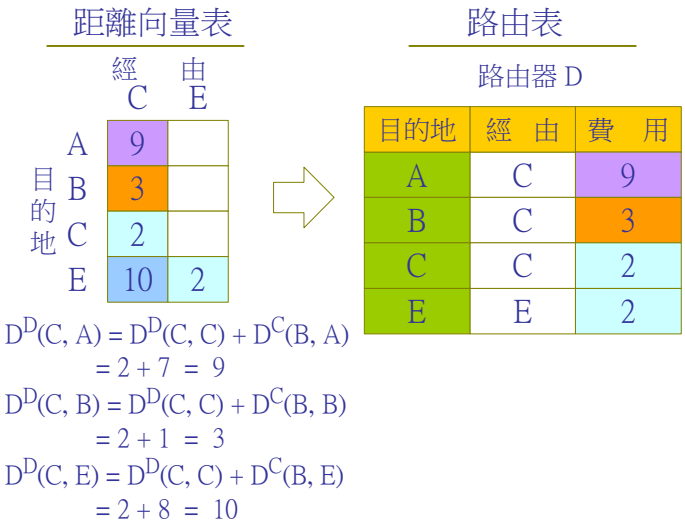



圖 6-12 路由器 D 經過（C → D）後建立新的路由表

## (E) 路由器 D 傳送給路由器 E（D → E）

路由器 D 將路由表（圖 6-12）傳遞給路由器 E。又 E 的起始距離向量表如圖 6-8 (e) ，再利用這兩個向量表建立出新的距離向量表。搜尋步驟如下：

**(1)** 到達 A：DE(D, A) = DE(D, D) + DD(C, A) = 2 + 9 = 11。

**(2)** 到達 B：DE(D, B) = DE(D, D) + DD(C, C) = 2 + 3 = 5。

**(3)** 到達 C：DE(D, C) = DE(D, D) + DD(C, C) = 2 + 2 = 4。

**(4)** 搜尋完後所建立之向量表及所建構出新的路由表如圖 6-13 所示。

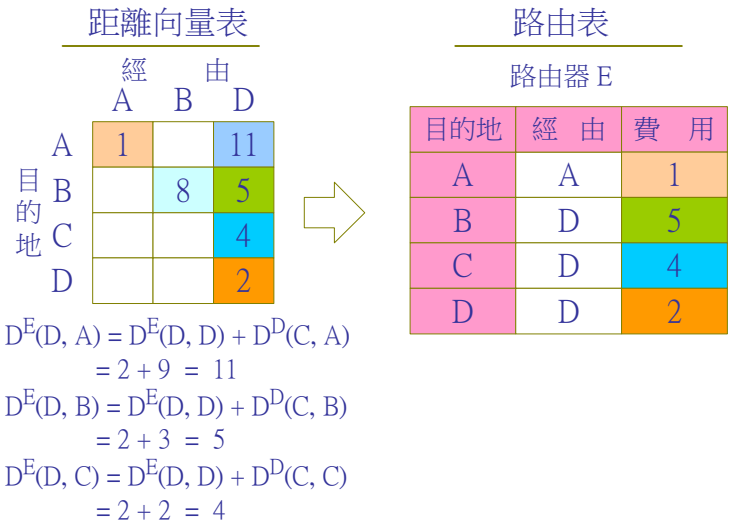
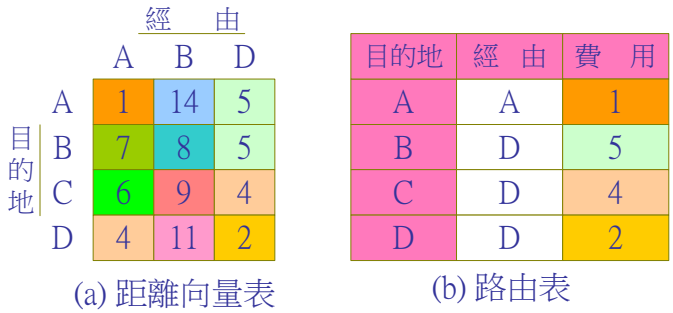



圖 6-13 路由器 E 經過（D → E）後建立新的路由表

以上所推演的過程之中，只有單一路徑 ( A → B → C → D → E )。當然，在正常情況下，路由器會隨時交換它們之間的距離向量表。假設，它們之間的路徑費用沒有改變，路由器 E 的向量表和路由表將如圖 6-14 (a)、(b) 所示。



經由

| 目的地 | A | B | D |
|---|---|---|---|
| A | 1 | 14 | 5 |
| B | 7 | 8 | 5 |
| C | 6 | 9 | 4 |
| D | 4 | 11 | 2 |

(a) 距離向量表

| 目的地 | 經由 | 費用 |
|---|---|---|
| A | A | 1 |
| B | D | 5 |
| C | D | 4 |
| D | D | 2 |

(b) 路由表

**圖 6-14 路由器 E 的距離向量表和路由表**

但是如何來建立和更新路由表？一般我們會使用圖形理論中的分散式-非同步的最短路徑演算法。較常用的是 Bellman-Ford Algorithm，演譯法如下：

## Bellman-Ford algorithm (at node X)

1. Initialization:

   for all adjacent nodes (column) v

   D(*, v) = ∞

   D(v, v) = c(X, v)

2. Loop

   Execute distributed topology update procedure

Forever

## end of Bellman-Ford algorithm

## Topology Update Algorithm

At node X:

1. wait (until I see a link cost change to neighbor Y, or until I receive control message from neighbor W)
2. if (c(X, Y changes by δ)

   /* change in cost to my neighbor, Y */

   change all column-Y entries in distance table by δ

if this changes cost of least cost path to some node Z, send control message to neighbors with my new minimum cost to Z.

3. if (control message received from my neighbor W)

/* shortest path via W to some node Z has changed */

DX(Z, W) = c(X, W) + new distance from W to Z

If cost of my least cost path to Z has changed send control message to neighbors with my new minimum cost to Z.

** end of Topology Update Algorithm

## 6-5-2 迴路問題

距離向量之路徑選擇法雖然較適合於大網路上使用，但也可能造成兩種異常狀態：**『迴路問題』（Looping）**和**『震盪情形』（Oscillations）**。我們先來討論迴路問題，其最主要的現象是：好消息傳的快，壞消息傳的慢。當路由器啟動時會即時計算本身的向量表，並向鄰近路由器通告，使整個網路迅速建立起來。網路使用一段時間後，各個路由器都已尋找出最佳路徑到其他路由器。但當中有任何路徑發生故障，必須再靠路由器之間的傳遞來發現路徑不通，重新計算最短路徑可能必須要一段傳遞時間。如圖 6-15 為好消息傳得快的範例，假設起始狀態之前端點 A 和 B 之間斷線，因此，各端點到 A 之間的跳躍距離為無限大（∞）。當第一次傳遞時，B 到 A 之間距離向量就被設定為 1。經過四次傳遞後，便可以將 A 路徑恢復的消息傳遞給所有端點。

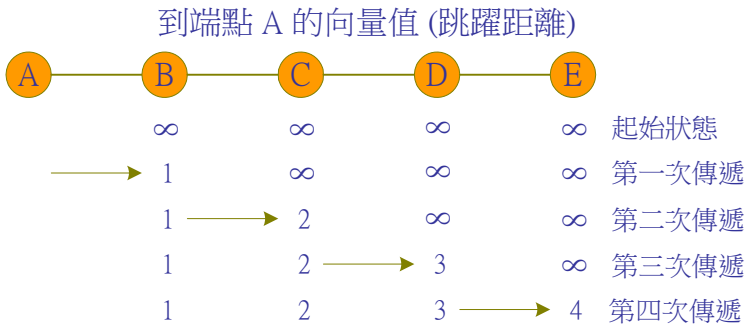



**圖 6-15 迴路問題範例（好消息傳得快）**

圖 6-16 為壞消息傳得慢的範例，起始狀態是所有路徑都正常，各端點上也都有紀錄前往端點 A 的距離向量（跳躍距離）。當 A 到 B 之間的路徑已斷線第一次傳遞時， B 無法傳送到 A，而將距離設為無限大，但它將這個訊息告訴 C，但 C 的紀錄裡有一路徑可到達 A 其向量值為 2，並將該值傳給 B，B 因而設定其前往 A 的向量值為 3。第二次傳遞，B 告訴 C 經由它那裡到達 A 的路徑為 3，因此，C 又將其到達 A 的向量值改為 4。第三次，C 將它到達 A 的向量值傳遞

給 B 和 D，它們又將前往 A 的向量值改為 5 和 5。依此類推，如果要將 A 斷線的消息傳遞給所有端點，也許需要無止境的迴路。

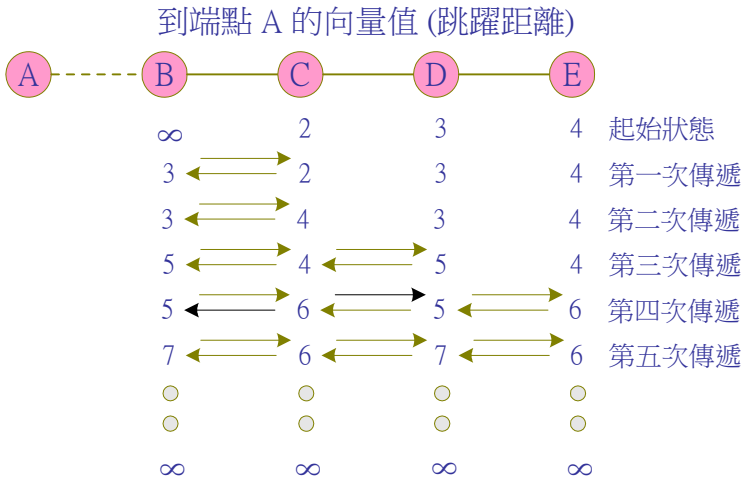



圖 6-16 迴路問題範例（壞消息傳得慢）

解決迴路問題在許多文獻中提出各種解決方法，但以下列兩種方法較為常用：

## (A) 水平分離切口（Split Horizon Hack）

路由器在發送距離向量表時，都先假設某一邊路徑已有斷線的可能，而去試探它。做法如下：如果路由器往某一路徑是最佳路徑，下一次發送向量表時，發送給該端點之路徑費用設定為無限大，而另一方向以正常向量值發送。宛如，水平切開兩邊缺口。如果對方路徑還是正常，下次發送正常向量值；如果已斷線就給無限大的向量值。在圖 6-17 之中，端點 C 送前往 A 端點的向量值無限大給 B，而以正常的向量值 2 給另一邊的 D。如果當時 A 已斷線，B 找不到最佳路徑，就將該向量值設定為無限大，並以下次傳送給 C。第二次切離時，D 將前往 A 的向量值設定為無限大並傳給 C，另一方面，它以正常向量值 3 傳給 E。端點 C 收到該向量值後，也找不到另一路徑可到達 A，便將該向量值設定為無限大。依此類推，每分離切割一次，就將 A 斷路的訊息往前發佈一個端點。也可以避免產生向量值傳遞迴路的問題。

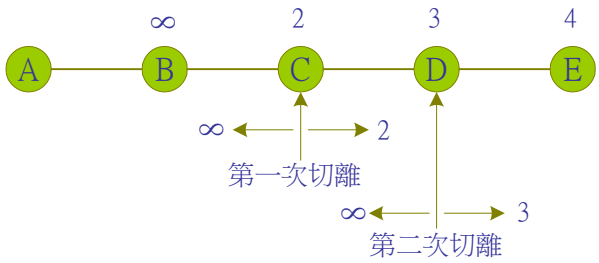



圖 6-17 水平分離切口法範例

### (B) 暫停（Hold Down）

當某一路由器發現路徑故障，首先將故障路徑的費用設定為無限大，通知鄰近路由器。等待一段時間後，收到其他的向量表，再依照當時情況決定最短路徑。

## 6-5-3 向量表震盪情形

當路徑費用是以資料流量大小來評估的情況下，路由器發現某一路徑傳輸流量較低，也將該路徑的向量值設定為最低，並發送給相鄰路由器。因此，所有的傳輸訊息便往該路徑發送，這個路徑費用又變得最高。路由器又將這個訊息告訴相鄰路由器，它們又停止往這個路徑傳送，這條路徑費用又變成最低。依此類推，這條路徑費用就一直震盪（Oscillations）不停。解決方法是：(1) 不要定期的交換路徑資訊。交換訊息的時間加長，以減少震盪的現象。(2) 減低資料流量對路徑費用的比率。以減少資料流量瞬間改變對整個路由表的影響。

以上所介紹的路徑選擇技術，是目前 Internet 網路上的路徑協定普遍所使用的，尤其在動態路徑協定方面，幾乎都是使用 LS Routing 和 DV Routing 兩種技術，我們瞭解這兩種技術原理之後，以下針對 Internet 網路上，各種路徑協定來加以介紹，但不再重複介紹它的運作原理，請讀者自行參考。

# 6-5 Internet 路徑選擇

**『網際網路』(Internet)** 是目前全世界使用最廣泛的網路系統，連結上億台電腦，並分布於全世界任何一個角落，本節就以 Internet 網路的基本架構來探討其連結技術。如果 Internet 網路像圖 5-13( IP 路徑選擇範例 )一般的擴充，隨著網路愈來愈大時，要由網路中每一個路由器的紀錄，來計算通往所有路由器的最短路徑已漸不可能。而且網路愈大時，每個路由器內的路由表，也相對變得非常的長，每一個封包進入時，勢必浪費不少時間在於路由表上搜尋最短路徑，且路由表的維護也非常繁重。因此，Internet 網路上採用階層式路徑選擇的方法，類似目前電話系統之搜尋號碼位置的方式，以減低每個路由器上，路由表的建立及搜尋時間，以提高路徑選擇的效率。

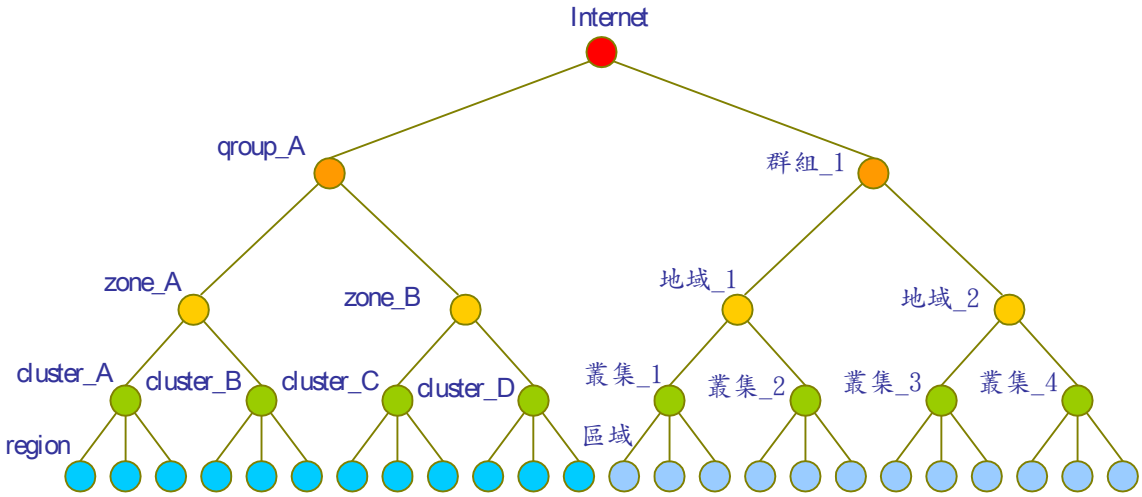



圖 6-18 Internet 網路架構圖

圖 6-18 表示 Internet 網路架構圖，使用階層式路徑選擇法，路由器則被劃分於所謂的區域（region）之內。每一個路由器非常清楚自己位於哪一個區域之內，對自己區域內之其它路由器的路徑選擇也非常清楚，但對本區域以外的路徑選擇便全然不知。至於和其它區域（region）之間的路徑選擇，就必須靠上層的路由器轉送，而無法直接繞徑（routing）。我們將若干個區域（region）集合成叢集（cluster），或又將若干個叢集組合成地域（zone），再將數個地域集合成群組（group），一直下去直到能完全分辨為止。其實，一般我們大型網路就是以階層式串接各地的區域網路，使用階層式路徑選擇法是最恰當不過的。

我們先以兩階層式路徑選擇法來介紹，如圖 6-19 所示。我們將一個組織單位（也許是一個公司、學校、教育部、國防部）的網路系統稱之為**『自治系統』（Autonomous System，AS）**，它是由若干個稱之為**『網域』（Domain）**的網路所構成，這些網域是分佈在組織單位中的任何角落，也許是單位內的小部門（如，分公司、系、所、學校）的網路系統。每個網域上至少有一個**『內部閘門』（Interior Gateway）**和其它網域的內部閘門串接。一個自治系統就由這些內部閘門所串接而成，但至少有一個**『外部閘門』（Exterior Gateway）**連接至外部網路。內部閘門所串接的網路，一般就稱為骨幹網路（Backbone）。其實圖 6-19 也類似第三章中圖 3-19 和 3-20 區域網路的傳輸骨幹。（注：所謂外部/內部閘門，其實也是一個路由器，而只是扮演的角色不同而已。）

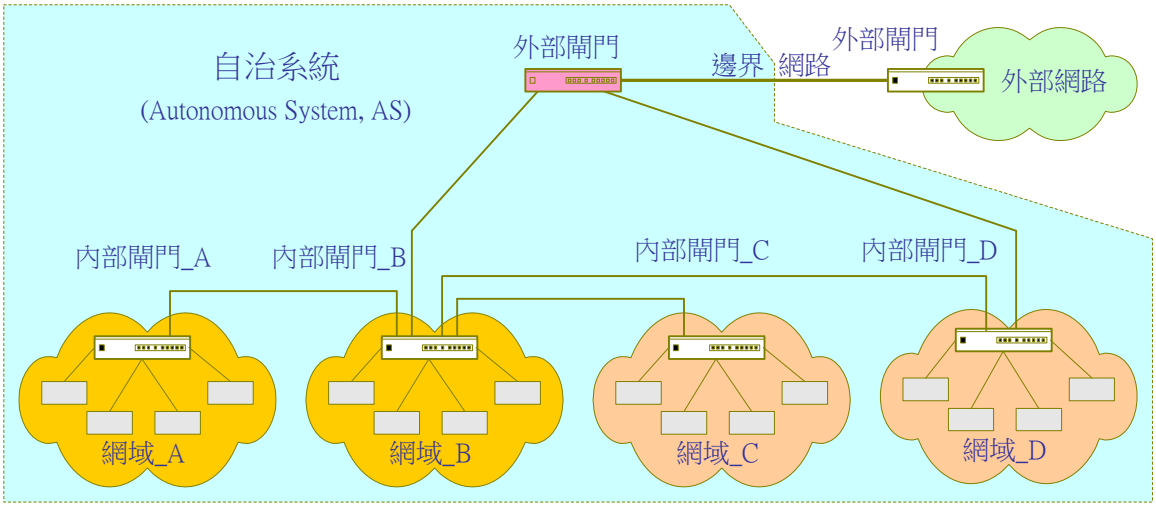



圖 6-19 兩階層式網路架構

依照圖 6-19 的網路架構，有三種不同層次的路徑選擇方式，又每一層次的路徑選擇都有不同的協定標準如下：

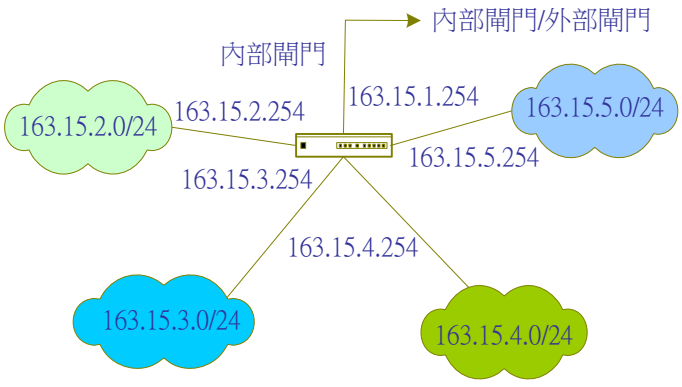
- **網域內路徑選擇**
  1. **靜態路徑選擇**
- **自治系統內路徑選擇**
  1. **RIP 路徑協定**
  2. **IGRP 路徑協定**
  3. **EIGRP 路徑協定**
  4. **OSPF 路徑協定**
- **自治系統之間路徑選擇**
  1. **CIDR 路徑協定**
  2. **BGP 路徑協定**

以下就分別來介紹以上這種路徑選擇方式。

# 6-6 網域內路徑選擇

網域內路徑選擇是電腦和內部閘門之間的路由方式，一般內部閘門都是使用多埠路由器（Multi-port Router）（或由多個路由器所構成），每一個埠口設定一個網路位置，埠口所連接的電腦，都屬於該網路的一份子。系統管理者也對整個網路內的成員都非常清楚，因此內部閘門主要以『靜態路徑選擇』（Static routing）為主。又內部閘門至少有一個連接埠口，連結到其它網域的內部閘門或外部閘門，如果內部閘門有不認識的網路位址（表示不在本網域內），便直接由該埠口發送出去，至於是否可以到達目的位址，這就不在管轄之內。

我們用圖 6-20 來說明網域內路徑選擇方式。圖中有一個多埠口路由器作為內部閘門，連接了四個網路，並有一個連接埠口接到外部網路，而以靜態路徑選擇方式，路由表（固定路由表）如圖中所示。當封包由任何一個埠口進入時，內部閘門依照它的網路位址，轉送到網路所屬的埠口上。例如，一個目的位址為 163.15.2.4 的封包進入，內部閘門就依照網路位址（163.15.2.0/24），由路由表上查詢出，應該轉送到 163.15.2.254 的埠口上。假如另有一個目的位址為 138.14.3.2 的封包進入，內部閘門在路由表上無法找出相對應的路徑，便將該封包轉送到 163.15.1.254 埠口（Otherwise）上，該埠口連結至外部網路，至於該封包如何尋找出下一個路徑，就由外部之其它路由器負責了。




內部閘門路由表

| 目的網路位址 | 轉送位址 |
|---|---|
| 163.15.2.0/24 | 163.15.2.254 |
| 163.15.3.0/24 | 163.15.3.254 |
| 163.15.4.0/24 | 163.15.4.254 |
| 163.15.5.0/24 | 163.15.5.254 |
| Otherwise | 163.15.1.254 |

圖 6-20 網域內之路徑選擇範例

## 6-7 自治系統內路徑選擇

如圖 6-19 所示，一個自治系統可由若干個網域所構成，每一個內部閘門管理一個網域。如果網域內所傳送的封包目的位址，是屬於本網域所管轄的網路內，封包將被隔離於內部傳送，但如果封包的目的位址超過網域所管轄範圍，該封包將會被送出該網域的內部閘門，且於若干個內部閘門（其它網域）之間，尋找出可到達目的地的最佳路徑。

通常企業內之網路範圍較小，且多半屬於同一權責單位所管，所以內部閘門之間的路徑選擇演譯法，通常皆使用**『鏈路狀態路徑選擇法』( Link-State Routing, LS Routing )**( 請參考 6-3 節說明 ) 或**『距離向量路徑選擇法』( Distance–Vector Routing, DV Routing )**( 請參考 6-4 節說明 )。兩者間的不同點在於內部閘門之間互相傳遞的訊息為何，LS Routing 是傳遞本身和相鄰內部閘門之間的鏈路狀況；而 DV Routing 是計算所有可能到達目的地，所經過的**「跳躍次數」( hop count )**傳遞給其他內部閘門。每一個內部閘門接收到這些訊息後，再計算出最佳路徑填入路由表，至於進入內部閘門的封包就依照路由表上，查出最佳路徑，並發送到下一個內部閘門，再由下一個內部閘門決定往哪一個路徑傳送。因此，所有內部閘門之間必須存在一個共通的通訊協定，以便傳遞網路之間的訊息，依此建立動態路由表，目前網際網路上較常用的**『路徑協定』( Routing Protocol )**有：

- Routing Information Protocol (RIP)
- Interior Gateway Routing Protocol ( IGRP )
- Enhanced Interior Gateway Routing Protocol ( EIGRP )
- Open Shortest Path First ( OSPF )

# 6-8 RIP 路徑協定

**『路徑訊息協定』( Routing Information Protocol, RIP )**( RFC 1058 ) 是由 Xerox 公司的 Palo Alto Research Center( PARC )所發展出來，目前是 Unix 電腦上的共通路徑選擇協定，可執行 routed 或 gated 命令來啟動 RIP 程式。 RIP 採用**『距離向量路徑選擇法』( Distance –Vector Routing )**，首先路由器( 內部閘門 )紀錄每個進入封包的來源位址和計算其所經過路徑的數目( hop )( 可由 IP 封包之 TTL 欄位數值計算出 )，在每一段時間 ( 一般設定 30 秒 ) 內廣播給相鄰的路由器。每一個路由器從自己所計算的訊息和其他路由器所傳遞過來的訊息之中計算出最佳路徑 ( 請參閱 6-4 節之演算法 )，再更新路由表。目前 RIP 協定在 Internet 網路上應用有兩種版本規格：RIP 和 RIP 2，以下分別介紹其訊息格式及運作方式。

## 6-8-1 RIP 與 RIP 2 訊息格式

RIP 訊息是以 UDP 協定 ( 埠口 520 ) 包裝，也是透過 IP 協定傳送，圖 6-21 為 RIP、UDP 和 IP 封裝格式。

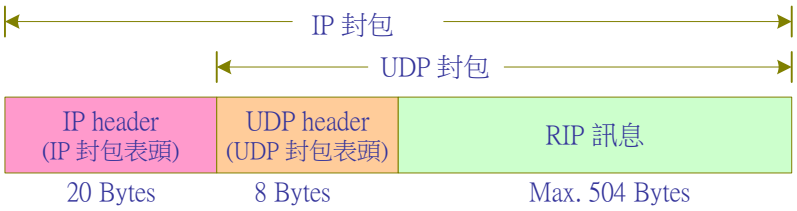



圖 6-21 RIP 訊息封裝

## (A) RIP 訊息格式

圖 6-22 為 RIP 封包格式，各欄位功能如下：

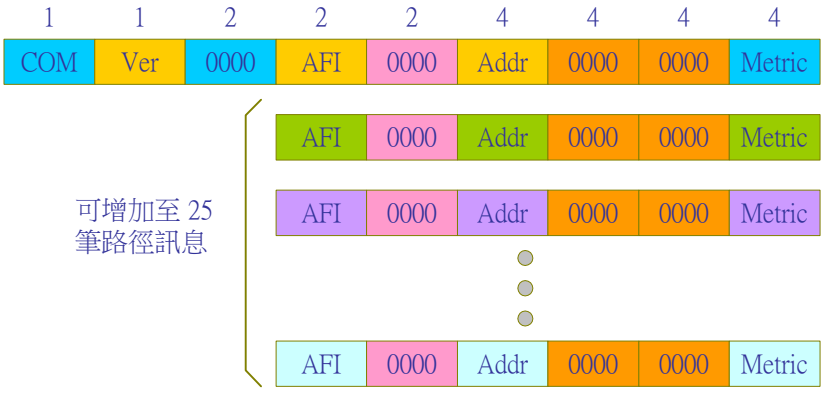



圖 6-22 RIP 封包格式

- **命令（Command, COM）：**表示此封包的命令是要求訊息（Request）（COM = 1）或回應訊息（Reply）（COM = 2），另外兩個沒有正式說明文件的命令：poll（COM = 5）和 poll-reply（COM = 6），作為要求或回應全部或部份路由表使用。其它命令（COM = 3 ~ 4）都未使用。

- **版本（Version, Ver）：**表示此封包的版本（Ver = 1）。

- **位址家族標示（Address family Identifier, AFI）：**在 RIP 協定中允許不同網路之間的訊息傳遞，AFI 表示所傳遞訊息之網路型態或位址格式。如果 AFI = 2 表示 IP 位址格式。

- **位址（Address, Addr）：**訊息之位址，IP 位址表示之。

- **路由值（Metric）：**到達位址欄位內之 IP 位址所必須經過的跳躍次數（hop count）。最高值為 15，如果超過 15（16）表示不可到達。Matric 的計算方式如圖 6-23，每經過一個路由器就增加一，譬如路由器 1 和 3 之間的 Metric 為 2。

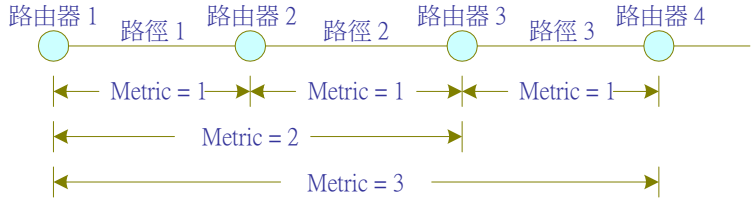



圖 6-23 路徑訊息之 Metric 計算方式

在一個 RIP 封包內最多可增列 25 筆路徑訊息，每一筆訊息的長度為 20 Bytes，因此 RIP 訊息最長為 504（20 × 25 + 4） Bytes，還不超過 UDP 封包最長 512 Bytes 限制，在一般環境下都可順利傳輸（MTU 限制）。其實 RIP 封包內設計每筆訊息可存放 14 位元組長的位址，但使用 IP 位址格式只用到 4 個位元組，其餘都設為 0。

## (B) RIP 2 訊息格式

圖 6-24 為 RIP 2 封包格式，由圖中我們可發現 RIP 2 充分利用 RIP 的位址的空白欄位，填入更多的訊息。以下介紹所增加的 3 個訊息欄位：

- **路由網域（Routing Domain）：**提供路由程式的辨識記號，也就是執行該程式的程序識別碼（Process ID）。
- **路由標籤（Route Tag, RT）：**提供一個方法來區分內部閘門和外部閘門之間的路徑訊息。
- **次網路遮罩（Subnet Mask, SM）：**提供該筆訊息的次網路遮罩，如都為 0 表示沒有提供 SM 資料。
- **下一路徑（Next Hop, NH）：**到達該筆訊息之位址的下一路徑。

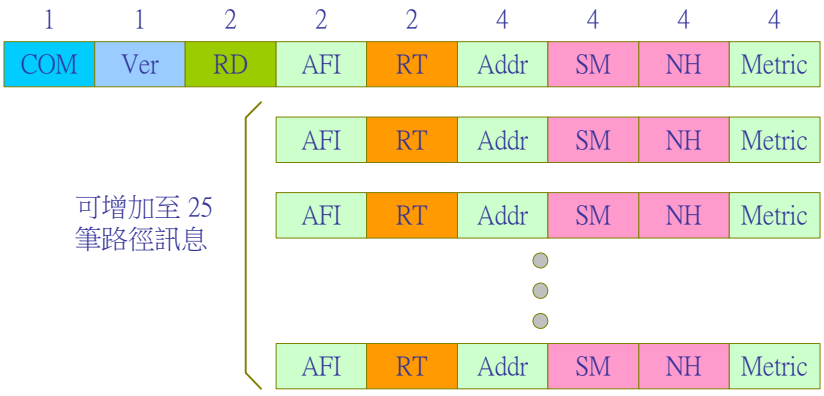



圖 6-24 RIP 2 封包格式

# 6-8-2 RIP 運作程序

RIP 協定是使用路由器上眾所皆知（Well-Know）的 UDP 埠口 520，它運作程序如下：

1. **初始化：**當 RIP 啟動時，便偵測所有運作的介面，並針對每一介面送出請求的封包（RIP Request）（或廣播方式），詢問其它路由器的路由表。對方路由器的埠口也是 UDP 520，而該 RIP Request 的 AFI = 0、Addr = 自己的 IP 位址、Metric = 16。

2. **收到請求：**路由器收到 RIP Request，即將本身的路由表以 RIP Response 回應給詢問者。另一種情況，如果 RIP Request 內所紀錄的路徑訊息，本身路由表內無資料，表示可能無法到達，便將該路徑的 Metric 設定為 16（表示無窮大的值），也一起回應給詢問端。

3. **收到回應：**詢問端收到 RIP Response，就利用 RIP Response 上所登錄的路徑訊息，來更改本身路由表。新的紀錄可以被加入，已存在的紀錄可以被更改或刪除。

4. **定期更新路由表：**一般系統都設定每 30 秒，路由器會將本身路由表得全部或部份，廣播給相鄰的路由器，以更新路由資訊。

5. **被動更新：**當路由器發現本身和相鄰之間的跳躍數有變更時，隨時發送更新資訊給其它相鄰之路由器，但只發送變更部份。

如果路由器發現它相鄰的路由器已超過三分鐘沒有發送訊息，則將前往該路由器的跳躍值更改為 16，以隔離往該路由器之路徑。又針對 RIP 有一些缺點，RIP 2 將其改進如下：

- 為了減少廣播 RIP 封包的數量，儘可能減少自治系統（AS）之間的 RIP 訊息流通，因此在 RIP 2 的路徑訊息內有一個 RT（Route Tag）以區分不同自治系統，如果路由器收到的路徑訊息是不屬於本身自治系統，便可將其拋棄。

- 一般路徑資訊上的 IP 位址都以網路位址來表示，但也有可能以路由器本身的 IP 位址，如此便很難區分網路位址範圍，尤其是在有子網路分割的環境裡，因此 RIP 2 增加『次網路遮罩』（Subnet Mask, SM），以提供網路號碼的識別。

- RIP 雖然提供路由器之間交換路由表，但沒有提供最佳路徑選擇，RIP 2 的每一筆路徑訊息裡，也提供下一路徑（Next Hop, NH）訊息，以提供最佳路徑給其它路由器參考。

## 6-8-3 RIP 運作範例

我們用圖 6-25 來說明 RIP 協定的運作程序，也讓我們對 DV Routing 演譯法有更進一步的認識。假設圖中各路由器的路由表皆為起始狀態（未和其它路由器交換任何訊息），其中 Dest. 欄位表示目的網路（Destination）、Next H 欄位為下一路徑（Next-Hop）的路由器、Metric 為向量值（跳躍數量）、D( Direct )表示直接到達網路。假設訊息傳遞方式為 R1 到 R2、再到 R3、再到 R4，依此方向來看各路由表變化情形（當然，實際在運作的網路會雙方向廣播）。

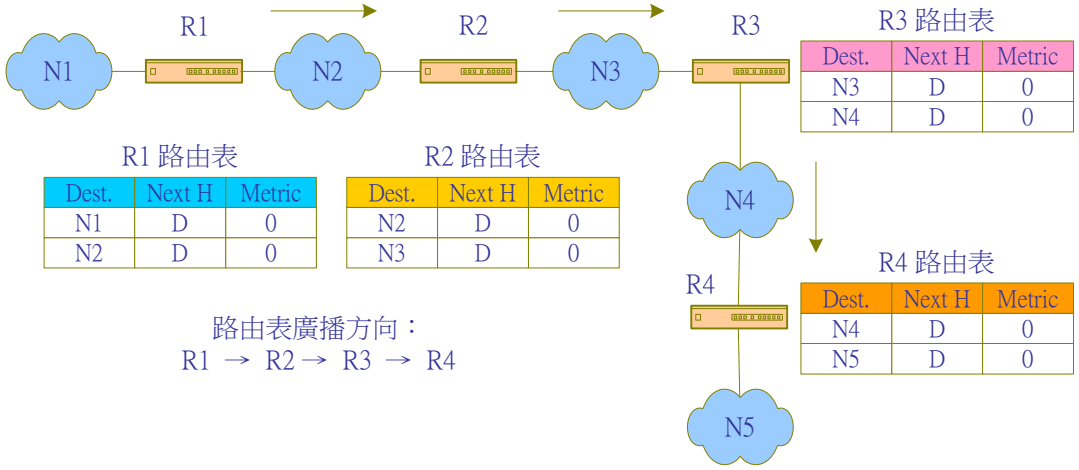



**圖 6-25 RIP 運作範例**

○ **R1 廣播路由表給 R2**：R1 廣播路由表給 R2 後，R2 建立新的路由表，如圖 6-26 所示。

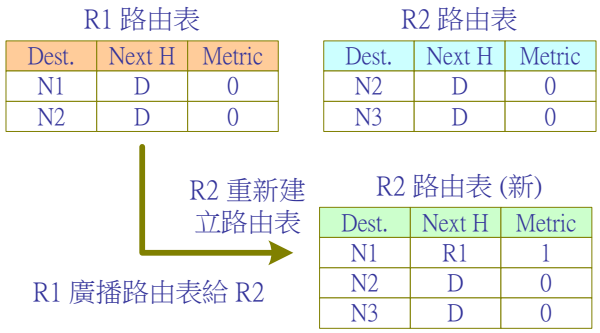



**圖 6-26 R1 廣播路由表給 R2**

○ **R2 廣播路由表給 R3**：R2 廣播新的路由表給 R3 後，R3 建立新的路由表，如圖 6-27 所示。

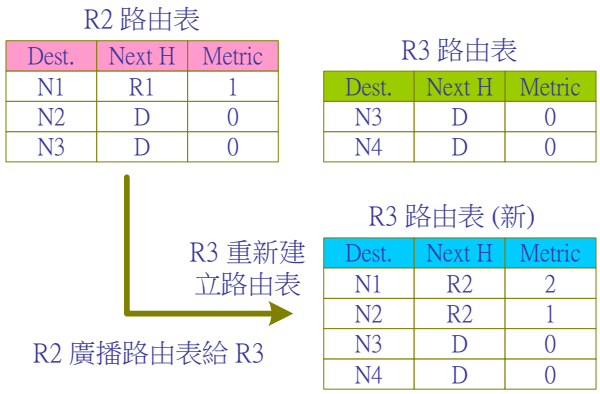


圖 6-27 R2 廣播路由表給 R3

○ **R3 廣播路由表給 R4**：R3 廣播新的路由表給 R4 後，R4 建立新的路由表，如圖 6-28 所示。

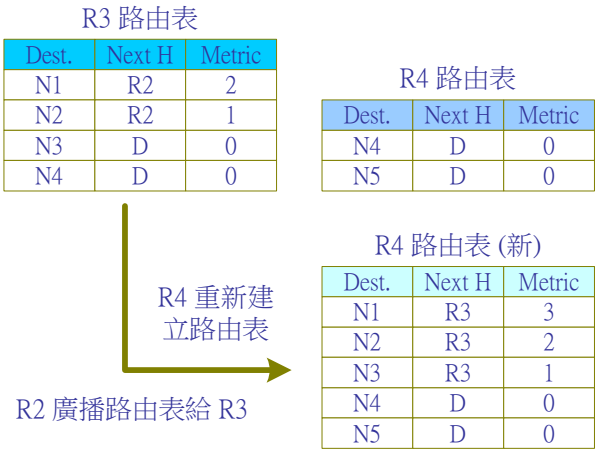



圖 6-28 R3 廣播路由表給 R4

RIP 協定的最大限制就是跳躍距離最大 15 個區段，目前網路中自治系統環境也愈來愈大，在一個自治系統之路由器也許會超過這個數目。又因採用距離向量法可能會發生訊息過慢收斂問題，也就是說當網路變更或故障時，無法在快速的時間內傳遞及更新所有路由器上的路由表，造成封包回繞或到達不了目的地。解決方法有水平分割法、以毒攻毒法等等（請參閱 6-4-2 節）。

# 6-9 IGRP 路徑協定

**『內部閘門路徑協定』( Interior Gateway Routing Protocol, IGRP )**是由 Cisco 公司於 1980 年中期發展出來，提供比較完整的自治系統（Autonomous System, AS）內之路徑選擇，也是針對 RIP 協定的功能增強。RIP 提供使用較小自治系統內，而且是在同等級( Homogeneous )網路之間使用，也限制 16 個跳躍距離。IGRP 提供較大型且複雜的自治系統內的路徑選擇協定。IGRP 和 RIP 的不同點如下：

- IGRP 可以服務較大的自治系統，跳躍距離不受限於 15。
- IGRP 可以提供多條路徑選擇，RIP 只提供單一最佳路徑。
- IGRP 可以重新配置於 RIP、OSPF、EIGRP 之協定內，也就是說可以共同使用及轉換。
- IGRP 提供快速更新資料計時器（Flush timer），如有資料變動，將更新之資料於迅速告知相鄰路由器（一般設定 10 秒）。

- IGRP 廣播訊息週期是每 90 秒一次。

基本上，IGRP 也是採用『**距離向量演算法**』來計算最佳路徑，但它的向量值（metric）不只使用跳躍距離。IGRP 的向量值可由下列參數的組合：網路間延遲時間（internetwork delay）、頻寬（bandwidth）、可靠度（reliability）與負載（load）。網路間延遲時間可由進入封包內所紀錄的發送時間和實際接收到時間的差異計算出來。頻寬可以將傳輸速率分為不同等級 1 到 255 之間來計算，例如將 1200 bps 到 10Mbps 的傳輸速率以 1 到 24 的級數之間來分別。至於向量值（metric）對於這些參數的權重比率值必須由系統管理員來設定，一般內定值（default）只會採用 delay 和 bandwidth 兩個參數，並以最佳權重比率計算。

路由器間利用相互之間訊息傳遞來建立路由表，其中最大的困擾就是收斂問題。網路上任何區段發生故障，或網路架構變更之訊息，無法立即傳遞給有關的路由器，造成網路之傳遞訊息暫時性的不正確，必須經過一段時間的訊息更新後，才能到達穩定狀況，這段時間稱之為『**收斂時間**』。IGRP 為提高路由選擇效率，採取多種方法來縮短收斂時間，以及預防網路不穩定，方法如下列說明：(如圖 6-29 所示)

- Flash Update：使用 Flash Update 訊息，以便快速通知相鄰路由器網路有變更，使加快收斂時間。
- Hold-Down Timer：使用 Hold-Down Timer 計時器，以預防路徑回繞。
- Split Horizon：用來防止傳回不正確訊息。
- Poison Reverse：用來移除不正確路徑。

在圖 6-29 中，假設網路 C 發生故障，Router_4 發現通往網路 C 的路徑已不通，立即廣播 Flash Update 訊息給相鄰之路由器。Router_3 接收到 Flash Update 訊息，知道經由 Router_4 到達網路 C 路徑已不通，立即啟動 Hold-Down Timer 並將往網路 C 之路徑刪除。並且啟動 Split Horizon ，將欲往網路 C 的路徑隔離，以防止任何封包欲經由 Router_3 傳送到網路 C。也就是說，要到網路 C 的封包不可再經由 Router_3 送往 Router_4，必須經由其他路徑。如果 Router_1 還未更新路由表，發送 Update Router 訊息給 Router_3，並告知經由 Router_3 可到達網路 C。則 Router_3 回應 Poison Reverse 給 Router_1 經由 Router_3 到達網路 C 的路徑為無限大。因此，Router_1 便知道必須移除該路徑。

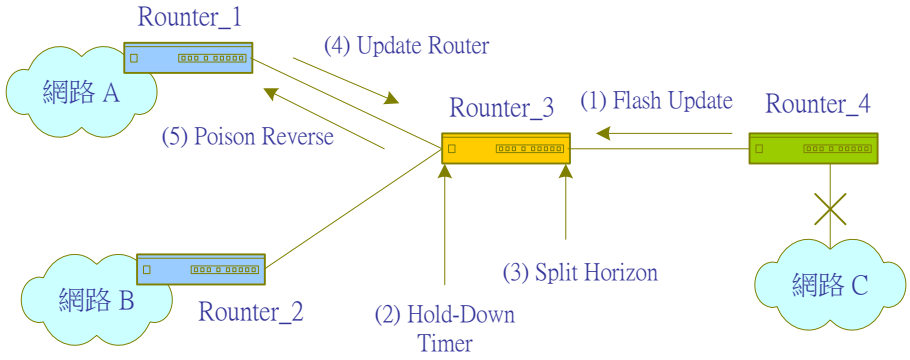



圖 6-29 IGRP 預防網路震盪範例

# 6-10 EIGRP 路徑協定

**『加強型內部閘門路徑協定』( Enhanced Interior Gateway Routing Protocol, EIGRP )** 是由 Cisco 公司針對 IGRP 功能的加強，使其更適合較大型的網路間路徑選擇協定。EIGRP 的路徑選擇演算法是整合**『鏈路狀態法』( LS Routing )** 和**『距離向量法』( DV Routing )**，成為一個稱之為**『擴張型更新演算法』( Diffusing-Update Algorithm, DUAL )**。另外，EIGRP 和其他路徑選擇協定有下列四個主要不同點：

**(1) 提供重新配置 ( Redistribution ) 功能**以整合不同網路的路徑選擇協定，如 Apple-Talk、IP 和 Novell Netware 之間。在 Apple-Talk 網路之下，重新配置是由 RTMP ( Routing Table Maintenance Protocol ) 所建立的路由表；在 IP 網路下，重新配置是由 RIP 、OSPF ( Open Shortest Path First )、EGP ( Exterior Gateway Protocol )、或 BGP ( Border Gateway Protocol ) 等協定所建立的路由表；Novell 網路下，重新配置是由 Novell RIP 等協定所建立的路由表，使這些異質網路 ( Heterogeneous Network ) 之間可經由 EIGRP 作最佳路徑選擇。

**(2) 快速收斂**。在 EIGRP 之下的所有路由器皆有儲存其相鄰路由器之路由表，因此它可以快速更新替代路徑，如果沒有適當路徑，路由器會發送查詢訊息給相鄰的路由器，這查詢訊息會一直被傳遞到找出適當路徑為止。

**(3) 提供可變長度的網路遮罩**。路由器會自動收集網路號碼的範圍，更進一步，EIGRP 可以被規劃為總結 ( summarize ) 任意位元長度的遮罩。

**(4) EIGRP 並非週期性的廣播訊息**，而是當本身路由表有所變更時，才將更新部份廣播給其他路由器，因此 EIGRP 使用頻寬比 IGRP 用的少。

為增強 EIGRP 的功能，它使用了四個主要技術：

(1) **鄰居發現與復原（Neighbor discovery/recovery）**。路由器必須隨時注意相連接網路之間是否有發生不可到達或停止工作的情況，當它發現某一路徑的負載特別低，便週期性發送 Hello 封包詢問對方，如一直沒有收到回應，表示該網路已不正常工作，則必須更新路由表並通知其他相鄰路由器。任何路由器接收到 Hello 封包必須即時回應。

(2) **可靠的傳輸協定（Reliable Transport Protocol）**。為了保證訊息封包都能按順序及安全到達相鄰路由器，EIGRP 提供多點廣播（Multicast）和單一廣播（Unicast）兩種封包。對於多重存取（Multiaccess）網路，則使用多點廣播封包；如是單一存取網路（如 Ethernet），則使用單一廣播封包。當廣播封包是 Hello 時不用回應確認訊息；但廣播更新（Update）封包時，接收者必須回應確認訊息。

(3) **DUAL 狀態轉換（DUAL Finite-State Machine）**被崁入計算和搜尋最佳路徑演算法內。DUAL 整合距離向量演算法和鏈路狀態演算法，能隨時找出最佳路徑更新路由表。

(4) **協定相依模組（Protocol-Dependent Module）**。特定網路層路徑選擇協定之間的連結可採用不同模組，這對網路的擴充性較高。

# 6-11 OSPF 路徑協定

**『開放式最短路徑優先』（Open Shortest Path First, OSPF）**是在 1980 年中期由 IETF（Internet Engineering Task Force）發展出來，主要應用於 IP 網路中內部閘門之間的路徑選擇協定。和 RIP 相比較，OSPF 能適用於較大網路或異質網路上。OSFP 有兩個重要特性：(1) 是開放性架構（Open），它的規格是公開性的（RFC 1247），任何廠商可任意安裝在自家電腦上，並修改或增加其功能。(2) 它是最短路徑演算法（SPF），在所有路徑之中最短路徑，一般都參考採用 Dijkstra Algorithm（請參考 6-3-4 節）。

## 6-11-1 拓樸圖資料庫

和其他路徑選擇協定的另一不同點是，OSPF 採用**『鏈路狀態路徑選擇法』（Link-State routing）**。在 OSPF 下的路由器定時傳送**『鏈路狀態宣傳』（Link-State Advertisement, LSA）**訊息給同等階級地區的其他路由器，LSA 訊息包含有連接介面、路由值（Metric）、以及其他相關變數值。OSPF 路由器計算這些參數後，並以最短路徑演算法找出，針對網路上（自治系統內）所有路由器中的最

短路徑。另外和使用距離向量法的 RIP 或 IGRP 有很大的不同，它們皆傳送某部份( 或更新部份 ) 的路由表給其他路由器；而 OSPF 是傳遞路由器所管轄內之『**路由拓樸圖**』給相鄰之路由器。

OSPF 能將自己管轄的自治系統 ( Autonomous System, AS ) 以階層式分割為若干個小區域 ( 如圖 6-30 所示 )。基本上，OSPF 是做自治系統內 ( intra-AS ) 路徑選擇的功能，但它也有能力處理接收和傳送路徑於自治系統之間 ( inter-AS )。被分割的小區域一般都稱為網域 ( Domain )，一個網域內也許連接數個路由器和若干個主機。網域之間連接的路由器稱之為邊界路由器( Border Router ) 或稱骨幹路由器( Backbone Router )，如圖中的 Router_4、Router_5、Router_12、Router_11、Router_10、以及 Router_6，它們之間連線稱為骨幹( Backbone )。骨幹路由器和一般網域內路由器( 如 Router_3 等 ) 處理不同的拓樸圖資料庫 ( Topological Database )。

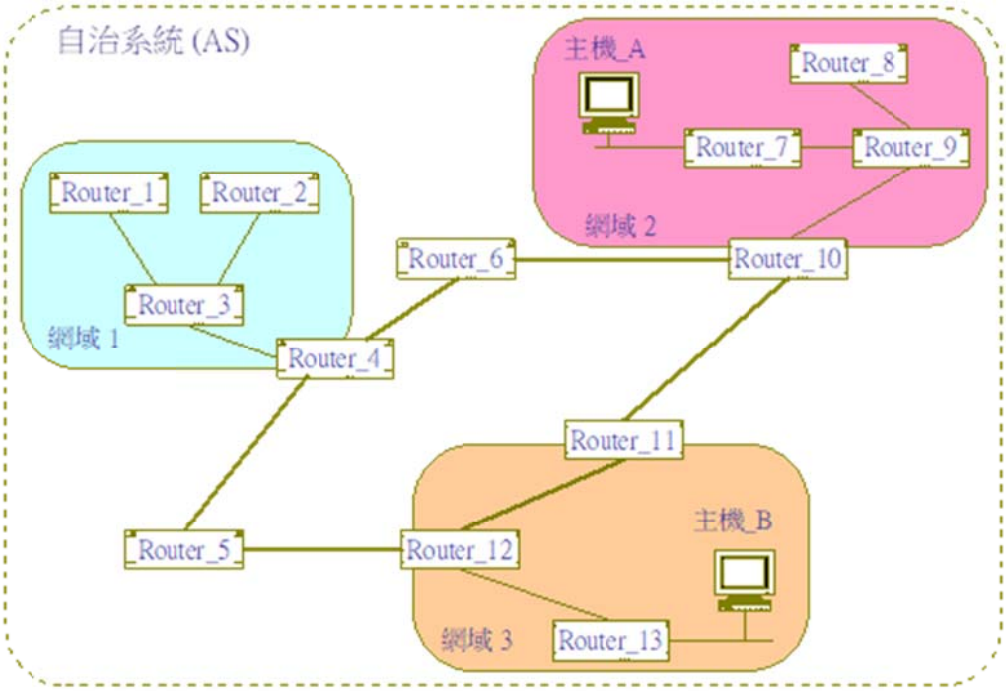



**圖 6-30 自治系統內 OSPF 的拓樸圖**

網域內所有路由器接收網域內其他路由器所傳送的 LSA 建立網路架構圖，並將其建構於拓樸圖資料庫內。骨幹路由器不僅必須建構網域內的拓樸圖，還必須建立骨幹的拓樸圖資料庫，因此在任一部骨幹路由器上可觀察到所有網域和骨幹網路的拓樸圖。在 OSPF 中有兩種路徑選擇功能：**(1) 網域內 ( intra-domain )**，處理封包的目的和來源位址皆屬於本網域之路徑選擇；**(2) 網域間 ( inter-domain )**，跨越不同網域必須透過骨幹路由器轉送。如圖 6-30 網域 3 的主機_B 欲傳送封包到網域 2 的主機_A，該封包被傳送到 Router_13、再往前送到 Router_11、再送到 Router_10 ( inter-domain )；之後再經由 Router_9 轉送到 Router_7 到達主機_A ( intra-domain )。

因此，骨幹路由器在跨越不同網域之間的路徑選擇，必須搜尋較複雜的拓樸圖資料庫，尤其在做連續封包傳送時，每個封包都必須搜尋資料庫。為了節省搜尋時間及次數，骨幹路由器可以建立虛擬鏈路（Virtual Link），來連結經常使用的路徑。但虛擬路徑在網域內路由器之間是共享而非專屬。骨幹路由器也可以學習經由外部閘門所傳過的路徑訊息，作自治系統之間的路徑選擇功能。

## 6-12-2 訊息格式

OSPF 訊息的傳輸也不同於 RIP 方式，RIP 是利用 UDP 封包封裝，再以廣播方式傳送給相鄰路由器。而 OSPF 有自己的協定號碼（Protocol=89），可以直接包裝在 IP 封包內（如圖 5-10），並用多點傳輸（Class D 位址），因此可以減少網路負荷。圖 6-31 為 OSPF 封包格式，其中各欄位功能如下：

- **版本（Version, Ver）**：表示該封包之 OSPF 的版本。
- **型態（Type）**：表示該封包的工作型態，有下列四種型態：
  - **Hello**：建立和管理相鄰路由器關係。
  - **Database Description**：描述拓樸圖資料庫的內容。這些訊息將因調整資料庫而被改變。
  - **Link-state Request**：向相鄰路由器要求傳遞某些片段的拓樸圖資料庫。這些訊息被傳送是因為某些路由器發現資料庫的內容已經失去時效性，要求重新更新。
  - **Link-state Update**：回應 Link-state Request 要求。傳送中的訊息可能經由 LSA 訊息修正過。
  - **Link-state Acknowledge**：確認接收到回應訊息。
- **封包長度（Length, Len）**：整個封包的長度，以位元組為單位。
- **Router ID**：發送封包的來源路由器之識別碼。
- **Area ID**：來源封包之區域（或網域）的識別碼。
- **Checksum（CS）**：檢查集之檢查碼。

- **認證型態（Authentication Type, AT）**：內容為認證型態。所有 OSPF 的交換訊息都必須經過認證，任何區域可自行規劃認證型態。
- **認證（Authentication, Auth）**：內容為認證訊息。
- **資料（Data）**：傳送給上層通訊協定之包裝資料。


**圖 6-31 OSPF 封包格式**

未來 OSPF 將被加入等效價格（equal-cost）與多重路徑選擇（multipath routing）的功能，使路徑選擇能依照上層服務型態（Type-of-Service, ToS）的要求來配置路徑。ToS 路徑選擇是由上層通訊協定依照服務的特殊需求而制定，因此可達到服務品質（Quality of Service, QoS）的需求。在應用上，譬如某個封包特別緊急，如果 OSPF 有不同優先等級鏈路，就可讓它優先通過。OSPF 提供一個或多個路由值（metric）計算路徑效率，如果只採用一個路由值，將沒有所謂 ToS 功能，我們可按照 ToS 的需求採用多種有關的路由值。例如在 IP ToS 之下，我們可採用延遲、傳輸量、可靠度等等參數，再由 OSPF 計算出等效價格（equal-cost）的路徑。

# 6-12 自治系統之間路徑選擇

傳送封包的目標網路位址不在自治系統（Autonomous System, AS）內，則必須被傳送到自治系統外部，由**『外部閘門』（Exterior Gateway）**之間尋找適合的路徑傳送，如圖 6-32 所示。外部閘門之間必須按照某個共通的協定來互相傳送路徑訊息及尋找路徑，此協定稱之為**『外部閘門協定』（Exterior Gateway Protocol, EGP）**。外部閘門和自治系統之間宛如國家邊界，因此又稱為**『邊界閘門』（Border Gateway）**。邊界閘門所扮演的角色非常的重要，它就像國家的邊界之出入境管理局一樣，負責審核各種封包是否可以進出自治系統。一般自治系統的封包進出都有依其政策型態（Policy dependent）規定，好比如研究機構、或國防單位的管理就更加嚴謹，一般環境也必須預防破壞份子的入侵。一般外部閘門之間的都採用**『距離向量路徑選擇法』（Distance Vector Routing）**，目前在 Internet 網路上較常用的是**『邊界閘門協定』（Border Gateway Protocol, BGP）**，我們就以 BGP 協定協定介紹外部閘們之間路徑選擇之功能。（注意 BGP 是 EGP 協定中的一個，兩者名稱不要混擾）

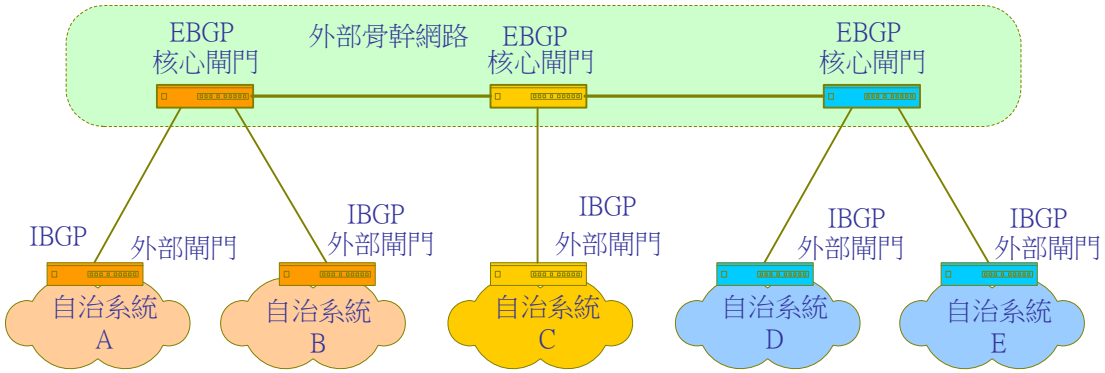



圖 6-32 邊界網路架構範例

在 Internet 網路上的外部閘門的路由器又被分為『核心閘門』( Core Gateway ) 和『非核心閘門』( Non-core Gateway )( 即是一般外部閘門 ) 兩種。非核心閘門必須紀錄自己管轄內，自治系統的網路拓樸之路徑選擇資料，以及核心閘門之間的路徑選擇資料，核心閘門只負責核心閘門之間的路由資料。一般核心閘門是由『網際網路操作中心』( Internet Network Operation Center, INOC ) 所管理，各自治系統之外部閘門( 非核心閘門 )只要連接到 INOC，由 INOC 管理整個核心骨幹。在自治系統之間的路徑協定有下列兩個主要標準：

- Classless Inter-Domain Routing ( CIDR )
- Border Gateway Protocol ( BGP )

以下分別介紹這兩個路徑協定：

# 6-13 CIDR 路徑協定

『無層級網域間路徑選擇』( Classless Inter-Domain Routing, CIDR ) 是專門用來解決目前 IP 位址不符所需，和減低路由表負荷的問題。最近幾年來 Internet 網路快速成長，很明顯的面臨幾個嚴重的問題，我們將其歸類以下三個問題：

(1) B 級 ( Class B ) 的網路位址漸被耗盡。在一個中型的機構裡 ( 或 ISP )，等級 B 的網路位址空間已漸不符所需，而不得不採用等級 C ( Class C ) 的位址空間。但是等級 C 的每一網路位址只有 254 個主機位址這又嫌太少；然而一個等級 B 的網路位址可容納 65534 個主機位址又太多，一般一個網路位址所管轄的主機並沒有哪麼多。

(2) 路由表過於龐大。網路快速成長造成每一邊界路由器上的路由表過於龐大，當某一封包進入時，必須在這路由表上搜尋最佳路徑太過於費時，又過於龐大的路由表維護也不容易。

**(3)** 甚至 32 位元的 IP 位址已快被耗盡。

其中第三個問題，32 位元的 IP 位址（IPv4）已漸不符所需，早已提出 IPv6（5-6 節介紹）來解決此問題，但是 IPv6 的實施費用過於龐大，網路上大部份的路由器都必須做適當的修正才可達到，也因此稱之為**『未來 IP』( Future IP )**。我們希望在最小的變更範圍內可以克服目前所面臨的第一和第二個問題，因此提出網域間路徑選擇技術。

雖然加入 C 級網路位址可以解決等級 B 的網路位址不足的問題，但卻延伸了另一個問題，每一個 C 級網路需要一個路由表，又造成路由表成長過大。譬如一個 ISP 使用到位址區塊 195.10.×.× 等級 C 的位址空間，這個區塊包含著 256 個網路位址，由 195.10.0.× 到 195.10.255.×，並且該 ISP 已將這些網路位址分配所屬的次網路。當執行路徑訊息傳遞時，該 ISP 的路由器必須將這 256 個網路區塊的路由表傳送給它的前端路由器，如此在各路由器之間路由表的傳遞就過於龐大。

## 6-14-1 CIDR 運作原理

CIDR 便是用來解決 Internet 網路路由表過大的方法，由 RFC 1518 與 RFC 1519 所描述規範，此方法又稱為**『超級網域化』( Supernetting )** 技術。CIDR 的觀念是利用一種方法將若干個 IP 網路位址，組合成一個網域空間，並摘要成一個較小的數字寫入路由表。例如，某 ISP 有 8 個次網路，每一次網路被分配 16 個 C 級的網路位址，而這 16 個網路位址都被分配使用，以致該次網路可將這 16 位址總結( Summarization )成一個次網域，而以一個較小數字表示，並寫入 ISP 的路由表。8 個次網域就構成一個**『超級網』( Supernet )**，也是由一個數字來代表路由，而此 ISP 是由同一點進入 Internet ，則這 8 個次網域只需要一筆路由表紀錄就可以，因此可以大大減低路由表的紀錄空間。

但必須有下列三個條件，才能允許總結發生：

1. 若干個 IP 位址被總結在一起，構成一筆路由路徑，它們的位址必需擁有相同的高位元，也就是說，必需相同的網路位址。

2. 路由表和路徑選擇演算法必需被延伸，才能以 32 位元的 IP 位址和 32 位元的網路遮罩來決定路徑選擇的依據。

3. 正在使用中的路徑選擇協定必需被延伸，以載送 32 位元 IP 以外的 32 位元的網路遮罩。但目前 OSPF 和 RIP 2 都具有攜帶 32 位元之網路遮罩的能力。

由上述三個條件所構成的超級網域化如圖 6-33 所示，假設某一 ISP（或自治系統） 使用到等級 C 的網路位址區塊是 195.10.×.×，而將該位址區塊分配給 8 個次網路使用，每一次網路享有 32 個等級 C 的網路位址，也就是由 8 個次網域構成一個超級網，而各網域之間的路徑選擇就稱為『網域間路徑選擇』（Inter-Domain Routing）。

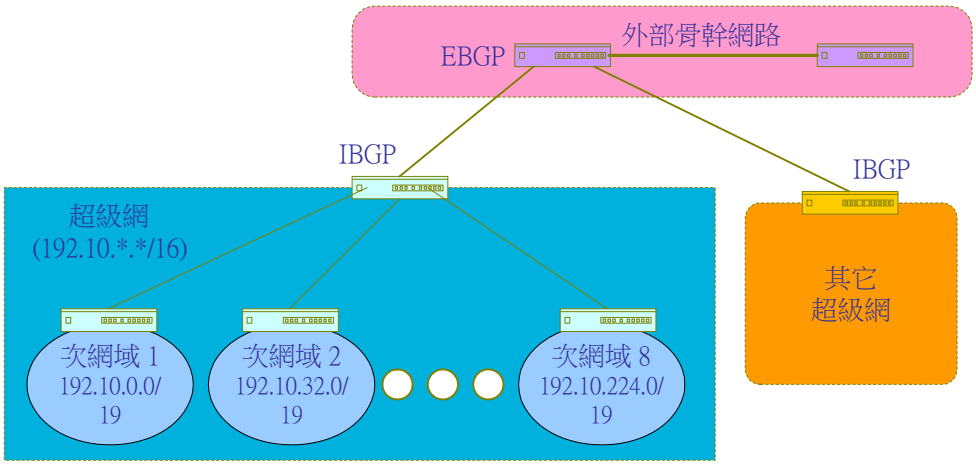



圖 6-33 超級網架構範例一

每一次網域的 IP 位址都有相同的高位元位址（較高位元之 16 位元），如果以網路遮罩表示為 255.255.224.0，因此，同一網域內（或自治系統之內）之 IP 位址和網路遮罩相配，必定會得到一個和其它網域的數值不同，而將此數值填入路由表，有相同數值的路由值就會被選擇繞送到該網域位址。對於 ISP（或自治系統之外）以外的地區也是一樣，由 192.10.0.0/255.255.0.0 也會得到一個獨立的網路數值，以作為路徑選擇的依據。超級網連結到外部網路是透過『內部邊界閘門』（Interior Border Gateway Protocol, IBGP），因此該 IBGP 就必須具有 CIDR 之功能，同樣的，在外部骨幹網路上的『外部邊界閘門』（External Border Gateway Protocol, EBGP）也必須具備有 CIDR 的功能。

## 6-14-2 CIDR 路徑選擇

但並非所有等級 C 的網路位址都像圖 6-33 般的簡單分配，如依照等級 C 的位址分配，它的網路位址是前面 24 位元。如圖 6-34 中，有可能將 195.10.×.× 中的某些網路位址分配給其它的 ISP 或地區使用，譬如，195.10.0.× 到 195.10.31.× 分配給另一個 ISP 使用，造成另一個超級網。因此路由值得設定就不能只觀察網路位址的最高位元相同的設定，而且必須增加利用另一個技

巧，藉由它的最佳符合（Best Match）的永遠是最長符合（Longest Match）網路遮罩者優先選擇路徑。譬如，EBGP 收到一個目的位址為 195.10.3.34 /255.255.224.0 的封包，它的網路遮罩長度為 19 位元（較高位元為 1 的數目），則優先繞徑到 IBGP-A，如另有一封包的目的位址為 192.10.34.56 /255.255.0.0，則會被轉送到 IBGP 路由器上。

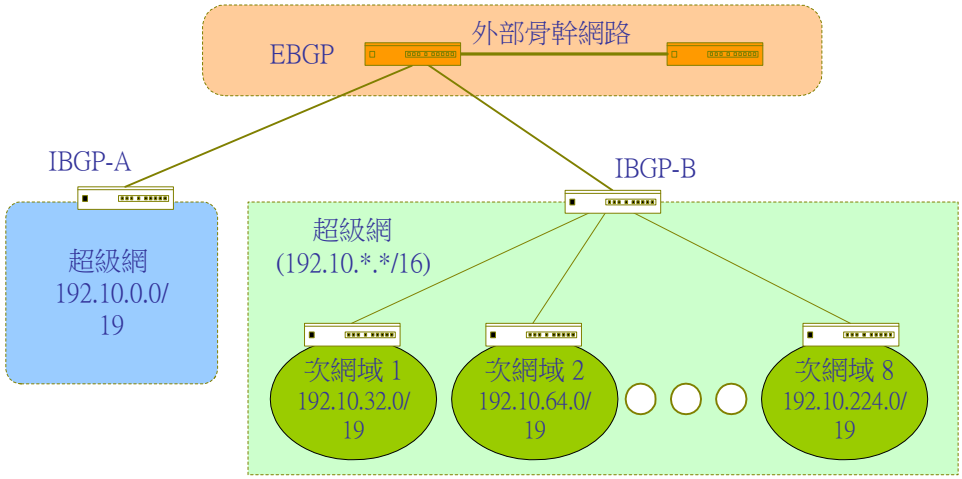



**圖 6-34 超級網架構範例二**

由以上的介紹，我們大略可將 CIDR 的路徑選擇技術歸略以下重點：

- 路由表是依照 IP 位址與網路遮罩所建構而成，針對每一路徑是由 IP 位址 + 網路遮罩相配，而以相同的最高位元的網路位址（由網路遮罩所指定），總結成一路徑選擇。（路由表的建構請參考 6-14 節）

- 路徑選擇是以最長的網路遮罩（較高位元為 1）為優先，雖然有相同的網路位址，但必須搜尋較長網路遮罩為優先路徑選擇。

由上述兩點的觀察，我們已將網路區塊依照**『IP 位址 + 網路遮罩』**，劃分為多個超級網（Supernet），也是根據整個 32 位元的 IP 位址遮罩運作來決定路徑選擇，不管該 IP 位址是 A 級、B 級或 C 級，都不會有任何差異，因此稱之為**『無層級』（Classless）**，而它的只要運作是網域間（或超級網之間）的路徑選擇，因此稱之為**『無層級網域間路徑選擇』（CIDR）**。

# 6-14 BGP 路徑協定

**『邊界閘門協定』（Border Gateway Protocol, BGP）**（RFC 1267）是一種自治系統之間的路徑選擇協定，目前已修正到第四版本（BGP4），詳細規格由 RFC 1771 上規範。一個自治系統大多是由一個或多個網路所構成，並在一個共通的管理環境及路由條件之下，一般都是由**『網際網路服**

務提供者』( Internet Service Provider, ISP ) 的網路範圍。在一個 ISP 之內的路徑選擇也大多透過『內部閘門協定』( Interior Gateway Protocol, IGP ) 來達成，譬如：RIP-2、OSPF 等等。BGP 主要是被使用於 ISP 之間的路徑選擇。

## 6-15-1 BGP 路徑選擇

基本上，BGP 路由器公佈和交換網路上可到達路徑的訊息給其它 BGP 路由器，該路徑訊息包含本身自治系統內和可到達其它自治系統的路徑訊息。BGP 協定也是屬於『距離向量路徑協定』，也相同的建構本身路由表，再傳送給相鄰的 BGP 路由器以更新路由表，如此週期性的更新路由表。但 BGP 和 RIP 有 BGP 與 RIP 很大的不同點在於，RIP 只宣告可到達路徑的跳躍數目，而 BGP 必需列舉到每一目標的路由。BGP 所言的目標也許是一個自治系統或是一個子網路系統，一個自治系統或許會包含許多網路號碼 ( 或 IP 網路 )。因此 BGP 用一個 16 位元的數字來表示一個自治系統，每一自治系統在包含一系列的網路號碼，這些都按照大小順序排列。

BGP 路由器之間交換路徑訊息有兩種情況：起始信息交換和後來訊息更新。當路由器連結上網路時，BGP 路由器將互相交換路由表，而當路由表有變更時，只交換變更部份並不全部傳送。基本上，BGP 路由器之間並不週期性交換訊息，而是路由表變更或發現更佳路徑時才會傳送。雖然 BGP 也是採用單一路由值 ( Metric ) 來表示一個路徑的費用，以作為最佳路徑選擇的基礎，路由值的評估可能包含：跳躍數、傳輸速率、延遲時間等等，但最主要的還是政策的考量。

BGP 訊息的傳輸相異於 RIP 和 OSPF，RIP 是包裝於 UDP 封包傳輸，OSPF 是直接利用 IP 封包作多點傳輸，而 BGP 是利用 TCP 協定傳輸。首先 BGP 路由器之間必須建立 TCP 連線，且交換整個路由表，從此以後，新增加或變更內容將被視為路由表的變更而傳送出去。

為了提供路徑選擇的效率，BGP-4 採用『無層級網域間路徑選擇』( Classless Inter-Domain Routing, CIDR ) 之技術。BGP-4 路由器之間使用 IP 得前置 ( Prifix ) 位元數 ( IP Netmask 前面連續幾個 1 )來簡化網路的『等級』( Class )，並將自治系統的路徑設定成若干個超級網( Suppernet ) 以簡化路徑訊息。但採用 CIDR 時，IGP 具有傳遞網路遮罩之功能，還好目前使用的 RIP-2 和 OSPF 都具有此功能。但在一自治系統 ( 或超級網 ) 之內也許會有其它次網域，如圖 6-29 所示，因此，在一自治系統內如使用 CIDR 來作路徑選擇，也必須利用 BGP-4 路由器，此路由器稱之為『內部 BGP』( Interior BGP, IBGP )，如果針對外部網路骨幹則稱之為『外部 BGP』( External

**BGP, EBGP )**。基本上，IBGP 負責自治系統和外部骨幹的橋樑，如訊息路徑並非本自治系統之內，便利用 IBGP 轉送給 EBGP，並由 EBGP 轉送到其它自治系統上，其架構如圖 6-35 所示。

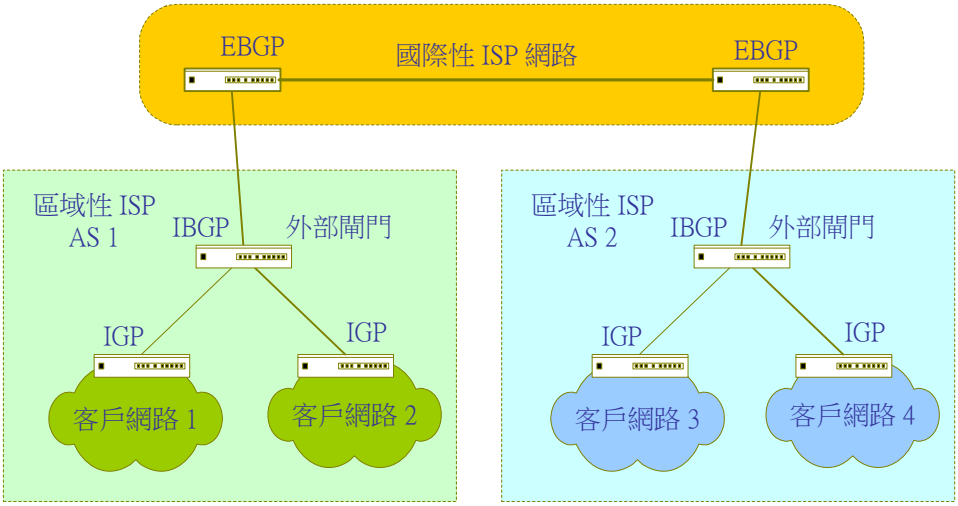



**圖 6-35 EBGP 與 IBGP 路徑選擇**

## 6-15-2 BGP-4 運作方式

BGP-4 是利用 TCP 協定來互相交換訊息，所使用用的著名埠口( Well-Known )為 TCP 179。在 BGP-4 路由器之間又可區分為：前端路由器 ( Front Router ) 和同儕路由器 ( Peer Router ) 兩種運作方式，前端路由器的運作就像圖 6-32 中 IBGP 和 EBGP 之間的運作，基本上 IBGP 處理內部網路的路徑選擇 ( 如 RIP-2 或 OSPF )，並將本身內部網路的路徑圖經 CDIR 協定 ( IP 位址 + 前置位元數 ) 處理後傳送給 EBGP 路由器 ( 以 BGP-4 協定 )，此內部路徑圖又稱為**『AS 圖』**。同儕路徑選擇就如圖 6-32 中，國際性 ISP 網路上的路由器之間的路徑選擇，也是我們主要探討的 BGP-4 的運作方式。又針對一部 BGP-4 路由器所管轄的範圍也許是由多個網路所構成，並且公告路徑訊息不一定要由 BGP 路由器負責，如果針對較複雜的網路環境，甚至可以利用一部主機電腦來處理 BPG 路徑訊息，並負責傳遞給同儕路由器，因此，在 BGP-4 環境下，一個網路路由節點都稱之為**『BGP-4 系統』( BGP-4 System )**。

BGP-4 系統起始建立路由表後分別傳送給相鄰的同儕系統 ( 或稱同儕路由器 )，而後隨著交換訊息來增加、修正或刪除某些路由表參數。BGP-4 並不需要週期性的廣播路由表給同儕系統，而是當路由表有所變更時，再利用**『Update 訊息』**通知相鄰之 BGP-4 系統有哪些路徑訊息變更，一般時候同儕系統之間週期性以較短的**『KeepAlive 訊息』**告知對方自己還是存在著。當網路有特

殊狀況發生或有異常障礙時，BGP-4 系統會以『**Notification 訊息**』告知相鄰之同儕系統，譬如，TCP 連線中斷。

如果有一自治系統是透過多個 IBGP 路由器( 或稱為 BGP-4 發言者 )連結到外部骨幹網路，如圖 6-36 所示。每一個 BGP-4 發言者是利用內部閘門協定 ( 如 RIP-2 ) 交換訊息所得，因此，在每一個 BGP-4 發言者所建構的 AS-圖和其它發言者不一定相同，當這些訊息都前送到 EBGP 路由器時，可能會造成路徑選擇之間的困擾。在這種情況之下，為了達到 BGP-4 發言者之間訊息資料的一致性問題，並不希望任一發言者可以隨意傳送訊息給前端 BGP，而是在所有發言者之間選一個當作所有訊息的出入( Exit/Entry )端點，並由此出入端點集中管理整個自治系統的 AS-圖，其它發言者由此端點的路由器索取最新路徑訊息，再前送給 EBGP 路由器，並且當其它發言者收到變更網路訊息 ( 內部網路或外部網路 ) 時，必需即時利用內部閘門協定傳送給出入端點路由器。當然也希望內部閘門協定的傳送更新訊息能在 IBGP 發言者傳送更新訊息之前完成。

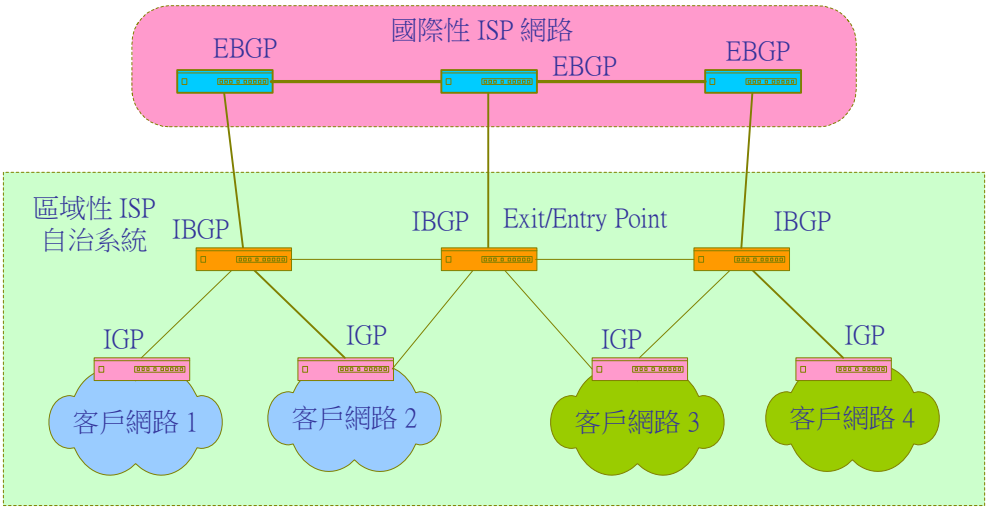



**圖 6-36 多重 BGP 發言者**

## 6-15-3 路徑訊息資料庫

每一個 BGP 發言者維護一只『**路徑訊息資料庫**』( **Routing Information Base, RIB** )，其包含了三個主要部分：

**(1) Adj-RIBs-In**：Adj-RIBs-In 儲存經過 BGP 發言者之間的『**Update 訊息**』學習得來的訊息，這些訊息再經過『**判斷處理**』( **Decision Process** )後所得的路徑訊息，再填入 Adj-RIBs-In 欄位內。

**(2) Loc-RIB**：Loc-RIB 儲存本地路徑選擇訊息，這些訊息也許是由 Adj-RIBs-In 欄位的資料，再經過本地政策所選擇的路徑訊息，以作為本地路徑選擇的主要依據。

**(3) Adj-RIBs-Out**：儲存預備公佈給其它同儕系統的路徑訊。，當本地 BGP 發言者欲公告路徑訊息給其它同濟系統，便將 Adj-RIBs-Out 包裝在『**Update 訊息**』內，傳送出去。

基本上，Adj-RIBs-In 儲存未經處理的路徑訊息，它是由其它同儕系統傳送而來的。Loc-RIB 是依照本地路由策略再加上 Adj-RIBs-In 訊息處理所得的路徑訊息。Adj-RIBs-Out 是經過處理後，發現有變更訊息而必須傳送給其它同儕系統的路徑訊息。

## 6-15-4 路徑訊息宣傳與儲存

為了達成 BGP-4 協定的運作，一個路徑被定義成一個單元的訊息，該訊息是由一對的目的位址所形成的途徑（Path），每一路徑的儲存與宣傳如下：

- 路徑被一對 BGP 發言者以『**Update 訊息**』宣傳，其方式如下：該系統可以到達的 IP 位址儲存『**Update 訊息**』的『**網路層可到達訊息**』(**Network Layer Reachability Information, NLRI**）欄位中，並且該路徑的屬性（Attribute）也儲存於屬性欄位上。
- 路徑訊息被儲存於 RIB 資料庫中，並區分為：Adj-RIBs-In、Loc-RIB 與 Adj-RIBs-Out 三個不同欄位。

BGP-4 提供三種方法來讓 BGP 發言者通知同儕系統有哪些先前所宣傳的路徑已經不再有效，也就是讓 BGP 發言者取消（Withdrawn）該服務路徑：

1. 在先前宣傳之路徑的 IP 前置位址加入『**Update 訊息**』的『**Withdrawn Routes**』欄位上，傳送給相鄰之同儕系統，表示該路徑已不再有效使用。
2. 將另一路徑更新已不再使用路徑的『**網路可到達訊息**』(**NLRI**)，並宣傳出去。
3. 關閉 BGP 發言者之間的連線，表示先前所宣傳的路徑訊息被移除掉。

## 6-15-5 BGP-4 訊息格式

BGP-4 訊息是利用 TCP 連線來互相宣傳，每一封包最大的容量為 4096 個位元組（Bytes），BGP 發言者之間傳遞有：Open Message、Update Message、Notification Message 與 Keep-alive

Meaasge 等四種訊息。這四種訊息都使用相同的封包標頭，標頭長度為 19 Bytes，如圖 6-37 所示，標頭欄位之功能如下：

- **Marker**：內容為一個認證訊息，讓訊息接收者可以預定該值。如果為 Open 訊息但沒有認證功能時，該欄位設定為全部 1；如果有加入認證訊息，接收端可利用該訊息來確定資料的正確性。
- **Length**：表示整個封包的長度，該數值一定在 19 和 4096 之間，BGP 訊息並沒有填補（Padding）資料。
- **Type**：表示該訊息的型態：
  1. 為『開啟訊息』（Open Message）
  2. 為『更新訊息』（Update Message）
  3. 為『通知訊息』（Notification Message）
  4. 為『保持存活訊息』（Keep-alive Message）。
- **Data**：內容為各訊息型態的資料（Open、Update、Notification、或 Keep-alive 訊息），其長度依照各訊息型態而不同。

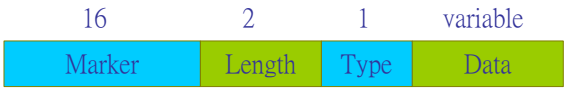



圖 6-37 BGP-4 之封包標頭

以下針對欄位 Type 所區分的四種訊息加以介紹，各種訊息是附加在封包標頭的後面，也就是如圖 6-37 上的 Data 欄位（依照各種訊息而有不同的長度）。

## (A) 開啟訊息（Open Message）

Open Message 是建立兩個閘門之間的交談連線，它是連線後第一個訊息，如欲傳送其他訊息之前，必須使用 Open Message 建立雙方對談連線。圖 6-38 為 Open Message 的資料內容，各欄位功能如下：

- **Version（Ver）**：表示該封包的 BGP 版本。

- **Autonomous System（AS）：**表示該發送封包者所在的自治系統編號。
- **Hold-Time（HT）：**表示保持時間，在這時間內沒有回應的路由器，都被假設已失去功能。
- **BGP Identifier（BGP）：**傳送該封包的外部閘門號碼（IP 位址）。
- **Optional Parameter Length（O-Len）：**表示緊接在後的 Optional 欄位的長度。
- **Optional Parameter：**任意參數。目前僅使用於認證（Authentication）訊息，有兩個部份：Authentication code 和 Authentication data。

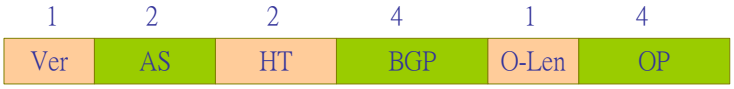



圖 6-38 Open Message 的資料內容

## (B) 更新訊息（Update Message）

Update Message 是被用來更新同儕系統之間的路徑訊息，使各個路由器都能建立一個可觀察整個網路的拓樸圖。Update Message 是由 TCP 連線完成已確定訊息的可靠度。當網路上有任何路徑被抽離，該相連之 BGP 發言者便利用 Update Message 告知相鄰之閘門。圖 6-39 為 Update Message 的資料內容，各欄位功能如下：

- **Unfeasible Router Length（URL）：**表示緊接著後面 Withdrawn Router 欄位的長度。
- **Withdrawn Router（WR）：**表示有那些已被抽離的路由器（IP 位址表示），可變長度表示之。
- **Total Path Attribute Length（TPAL）：**表示後面緊接著兩個有關屬性欄位的長度。
- **Path Attribute（PA）：**路徑屬性。描述該路徑之特性有能是下列屬性：
  - **Origin：**指派屬性（Mandatory attribute）。為原系統指定之路徑。
  - **AS Path：**經系統指定之經由多個自治系統片段所構成的路徑。
  - **Next Hop：**指派屬性。指定經由邊界網路的下一路徑可到達目的位址。
  - **Mult Exit Disc：**選擇屬性（Option attribute）。在多點路徑之中辨別可到達鄰近自治系統之路徑。

- **Local Pref**：任意屬性（Discretionary attribute）。描述任意路由的級數。
- **Atomic Aggregate**：任意屬性。被使用在表現有關路徑選擇的訊息。
- **Aggregator**：選擇屬性。包含有關路徑聚集的訊息。

- **Network Layer Reachability Information（NLRI）**：『網路層可到達訊息』是作為 BGP 發言者宣傳可到達的路徑，它包含一串列的 IP 位址的網路位址（前置位元），每一 IP 前置位元（IP Prefix）表示可到的路徑區段。


圖 6-39 Update Message 的資料內容

## (C) 通知訊息（Notification Message）

當外部閘門發現任何異常狀態，便使用該訊息告知相鄰閘門，或被使用於中斷連線。圖 6-40 為 Notification Message 的資料內容，各欄位功能如下：

- **Error Code（EC）**：該封包表示錯誤的種類，如下列：
  - **Message Header Error**：所傳送的封包標頭發生錯誤。
  - **Open Message Error**：所傳送的 Open Message 錯誤，如版本、自治系統或 IP 號碼、或認證錯誤。
- **Update Message Error**：所傳送的 Update Message 錯誤，如屬性不合等。
- **Hold Time Expired**：表示 Hold Time 溢時，將該區段之 BGP 被視為沒有功能。
- **Finite State Machine Error**：協定流程錯誤。
- **Cease**：結束 BGP 連線，
  - **Error Subcode**：錯誤型態的附加描述碼。
  - **Error Data**：內容為有關錯誤型態的資料。

圖 6-40 Notification Message 的資料內容

### (D) 存活訊息（Keep-alive Message）

用來測試連線中斷或 TCP 連線另一端的 BGP 路由器是否故障了，傳送訊息的建議是每 30 秒一次。該訊息並沒攜帶任何資料，因此沒有 Data 欄位（如圖 6-34），只在 Type 欄位上標示為 Keep-alive Message（Type = 4），整個封包長度為 19 Bytes。BGP 發言者就是利用此短的訊息，週期性的通知同濟系統自己還是存在著。

## 6-15-6 BGP-4 路徑屬性

BGP-4 雖然也是採用『**距離向量演算法**』，但它和 RIP-2 之間有很大的不同點，RIP-2 只利用『**跳躍距離**』(**Hop Count**)來評估路徑費用。而 BGP-4 利用許多『路徑屬性』(Path Attributes)來評估每一條路徑的費用，這些路徑屬性將被包裝在 Update 訊息內（如圖 6-39），以讓 BGP 發言者之間來互相傳遞。BGP-4 路徑屬性可區分為以下四大類：

1. **著名指定性（Well-known Mandatory）**
2. **著名隨意性（Well-known Discretionary）**
3. **選項過渡性（Optional Transitive）**
4. **選項非過渡性（Optional Non-transitive）**

『**著名屬性**』(**Well-known Attribute**)必須經過所有 BGP-4 的製造者共同確認，又著名指定的屬性必需被包含每一 Update 訊息內；而著名隨意的屬性可視環境需要來決定是否要加入到 Update 訊息內。除了著名屬性外，每一路徑也許包含若干個選項性（Optional）屬性，但這些屬性並不需要 BGP-4 製造商共同確認，而是各個廠商依照環境需求而增加，某些選項屬性不被其它廠商採用，在評估路徑費用時可以不用理會。過渡性（Transitive）屬性是屬於較區域性或特殊自治系統所制定的屬性；非過渡性是針對某些特殊路徑所制定，也會制定相對應的路徑規則。在 RFC-1771 中規範有許多路徑屬性，但並非所有屬性都會被一般製造商採用，我們以 Cisco 公司所實現的路徑屬性來介紹，Cisco 採用：Weight、Local Preference、Multi-exit Discriminator、Origin、AS_path、Next hop 與 Community 屬性，分別介紹如下：

## (A) 衡權屬性（Weight Attribute）

衡權屬性是 Cisco 所定義的本地路徑屬性，該屬性並不宣傳給其它同儕系統（或相鄰路由器）。如果路由器學習到同一目的位址有一個以上的路徑可以到達，便將衡權量較高的路徑填入路由表，並宣傳給其它相鄰路由器。如圖 6-41 中，路由器 A （AS 100）學習到（或收到宣傳訊息）兩條路徑可以到達 127.16.1.0/24 網路（AS 200），一條是經由路由器 C，衡權屬性設定為 100；另一條是經由路由器 B，衡權屬性設定為 50，因此，路由器 A 選擇路由器 C 路徑並填入路由表。

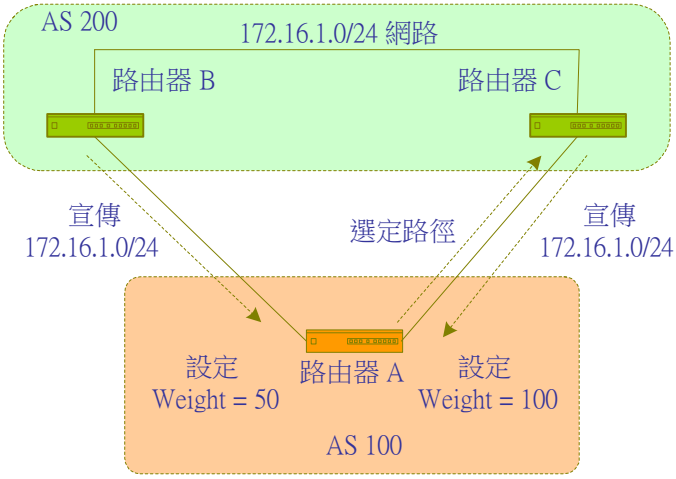



圖 6-41 衡權屬性

## (B) 本地優先屬性（Local Preference Attribute, Local_Pref）

Local_Pref 是屬於『著名隨意性』的屬性，被使用於表示選定本地自治系統的出口。它不同於衡權屬性，本地 BGP 發言者會將該訊息宣傳給相鄰的路由器，尤其在有多點出口的環境裡，Local_Pref 告知相鄰路由器哪一個才是本地自治系統的優先出口位址。如圖 6-42，AS 100 接收到兩個由 AS 200（172.16.1.0/24 網路）所宣傳的訊息，當路由器 A 收到由路由器 C 的宣傳訊息，則依照網路狀態設定為 Local_Pref＝50；路由器 B 收到由路由器 D 的宣傳，則設定 Local_Pref＝100，路由器 A 和 B 互相傳遞訊息後，判斷經由路由器 D 到達 172.16.1.0/24 的 Local_Pref 較高，因此，AS 100 前往 AS 200 網路路徑便選擇經由路由器 D。

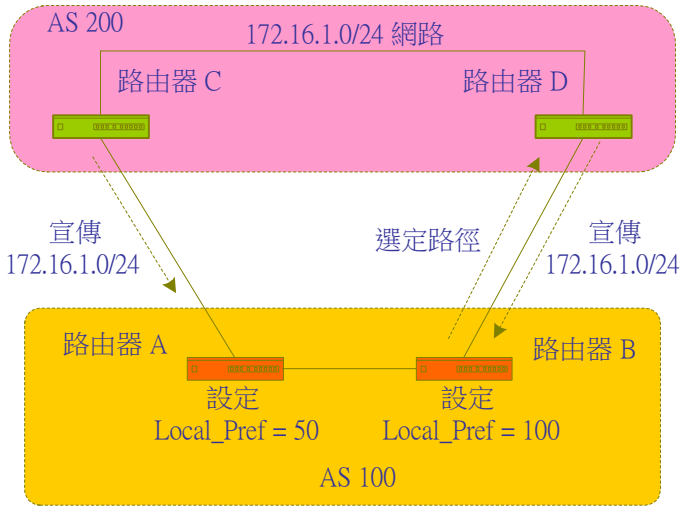



圖 6-42 本地優先屬性

## (C) 多重出口鑑別屬性（Muti-Exit Discriminator Attribute, MED）

或稱為『向量值』( Metric ) 屬性。MED 是用來建議外部自治系統進入本自治系統的向量值。如圖 6-43 所示，AS 200 的路由器 C 向 AS 100 系統的路由器 A 宣傳由本路徑進入本系統的 MED = 10；另一方面，路由器 D 宣傳進入本自治系統的 MED = 5，在 AS 100 內經過訊息交換後，判斷經由路由器 D 到 AS 200 系統費用較低，便選用該路徑到達 AS 200 系統。

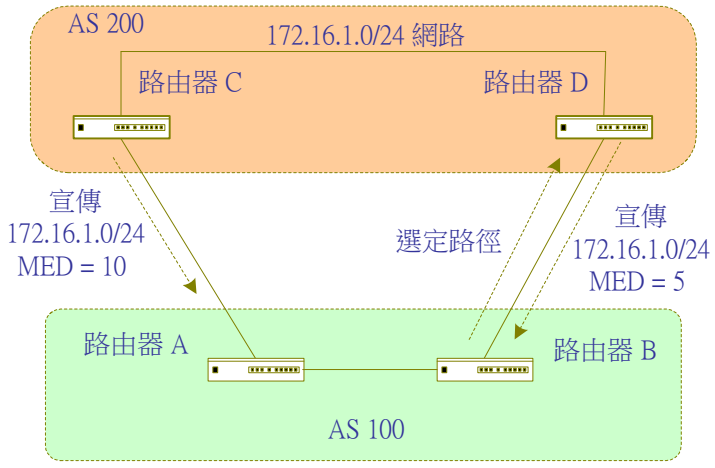



圖 6-43 多重出口鑑別屬性

## (D) 起源屬性（Origin Attribute）

Origin 也是屬於『著名指定屬性』，是用來表示該路由是 BGP 以何種途徑所學習得來。Origin 可能是下列三種數值之一：

- IGP：表示該路由是經由『內部閘門協定』( IGP ) 學習得來的。

- **EGP**：表示該路由是由『**外部邊界閘門協定**』（**Exterior Border Gatway Protocol, EBGP**）學習得來的。

- **Incomplete**：表示該路由起源不明或是經由其它通訊協定學習得來，這種屬性的路由大多是被重新分配到另一個 BGP 上。

## (E) 自治系統路徑屬性（AS_Path Attribute）

AS_Path 也是屬於『**著名指定屬性**』，它是被用來表示某一路由所經過的路徑。當一個路由被宣傳而經過某一路由器時，該路由器便將它的 AS 識別值加入到此路由的次序串列中（AS_Path），再宣傳給其它自治系統，因此，由 AS_Path 屬性就可以觀察到該路由所經過的路徑。由圖 6-44 可以觀察到，AS 1 的起源路由為 172.16.1.0/24，並向 AS 2 與 AS 3 宣傳該路由，宣傳時將 AS_Path 設定為 [1]，當 AS 2 和 AS 3 收到該宣傳，便將自己的 AS 識別碼加到 AS_Path 內後傳遞給下一個自治系統（[3.1] 與[2.1]）。但當 AS 1 收到 AS_Path = [3.1] 或 [2.1]，也就知道該路由是由本地的路由器發出，便拒絕該路由訊息。又譬如 AS 2 收到 AS 3 系統所宣傳的 AS_Path = [3.1]，也判斷自行由 AS_Path = [1] 的路由路徑會較為短捷，因此，它會選用 AS_Path = 1 的路由到達 AS 1。

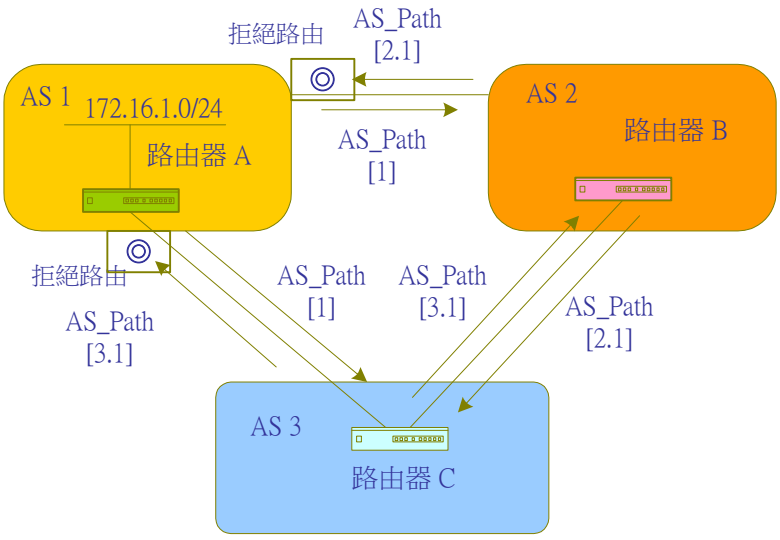



圖 6-44 自治系統路徑屬性

## (F) 下一跳躍屬性（Next-Hop Attribute）

Next-Hop 屬性是針對 EBGP 的下一跳躍的位址規範，也就是邊界路由器的位址。在 RBGP 的同儕路由器之間，Next-Hop 是以一個 IP 位址來表示，也表示由此位址可進入哪一自治系統。如圖 6-45 所示，AS 200 系統宣傳進入 172.16.1.0/24 網路的 Next-Hop = 10.1.1.1，AS 100 的路由

器 A 收到該訊息後，再宣傳給路由器 B，表示欲進入 176.16.1.0/24 網路的下一跳躍位址為 10.1.1.1。

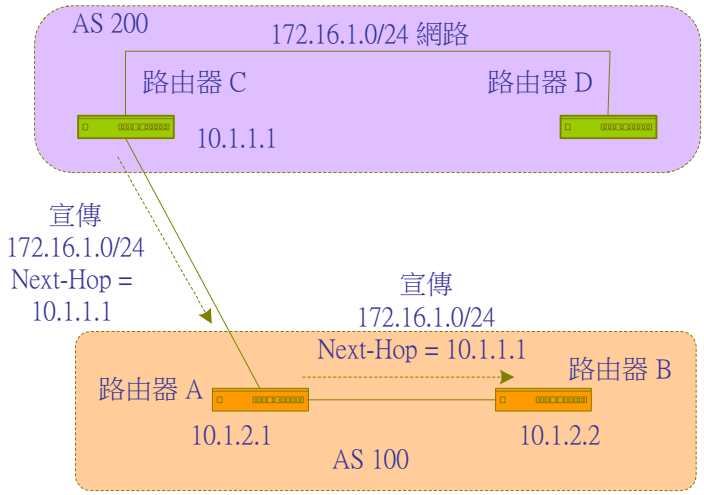



圖 6-45 下一跳躍屬性

## (G) 共同體屬性（Community Attribute）

Community 屬性是提供一種方法來處理群體性之目的位址。可利用 Community 來規劃某一群組目的位址成為一個共同體，以作為決定是否傳送路由訊息給這一群組的成員。一般事先定義有下列三種共同體屬性：

- **No-Export**：不要宣傳此路由給同儕邊界路由器（EBGP）。
- **No-Adverties**：不要宣傳此路由給任何同儕路由器。
- **Internet**：宣傳此路由給 Internet 共同體，所有路由器都屬於此共同體的成員。

圖 6-46～48 為上列三種屬性的宣傳傳遞方式，其中 圖 6-46 表示 No-Export 屬性的宣傳方式；而 圖 6-47 表示 No-Adverties 的宣傳方式；另外 圖 6-48 是 Internet 屬性的方式。

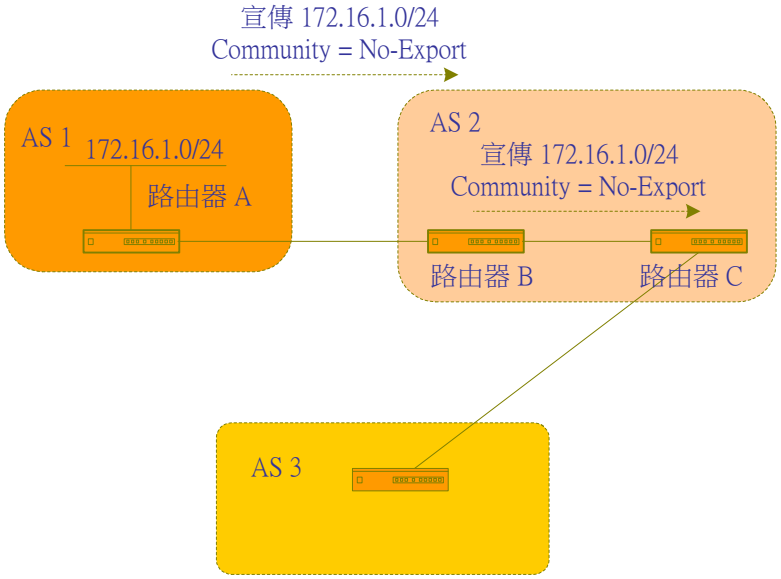



圖 6-46 共同體屬性之 No-Export

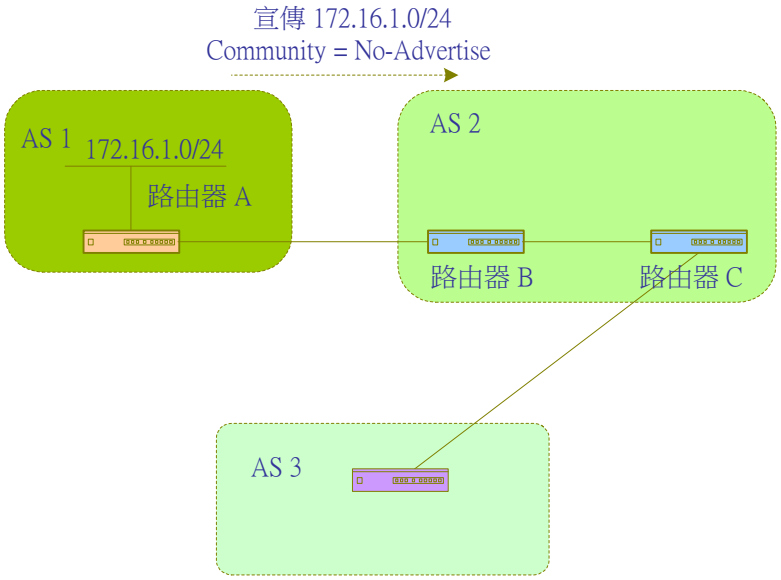



圖 6-47 共同體屬性之 No-Advertise

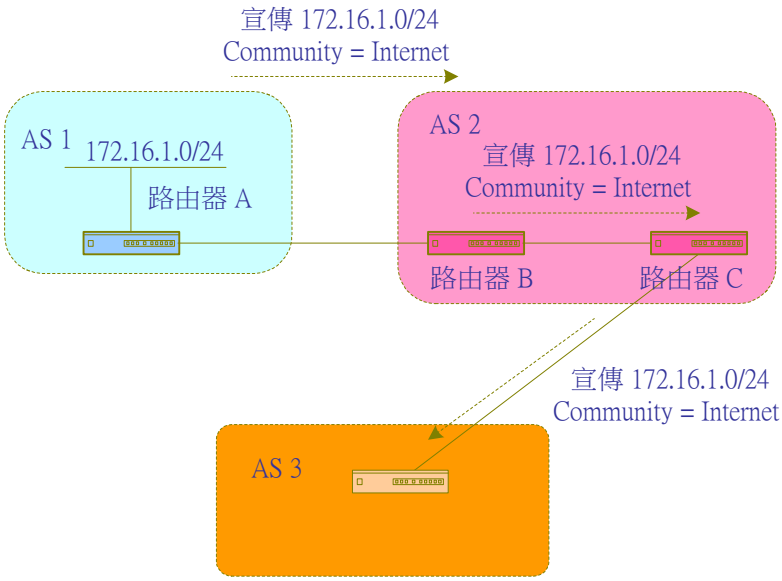


圖 6-48 共同體屬性之 Internet

## 6-15-7 判斷處理

判斷處理（Decision Process）是由週期性的宣傳訊息中選擇適當的路由，當 BGP-4 路由器收到同儕系統所宣傳的訊息將其儲存於 RIB 的 Adj-RIB-in，再經過判斷處理後儲存於 Loc-RIB 資料庫內。由經過處理後的路由及其它訊息儲存於 Adj-RIB-Out，而準備向其他同儕系統宣傳。選擇適當路由是依照每一路由的屬性來作判斷的依據，依照路由的屬性關係來評估每一路由的優先等數（Degree of Preference），以選擇較高的優先等數的路由。在 RFC 1771 中，以三個處理時相（Phase）來測試各種不同的現象：

1. **Phase 1**：負責計算由相鄰自治系統 BGP 發言者所傳送路由的優先等數，並將較高優先等數的路由向本自治系統之 BGP 發言者宣傳，每一較高優先級數路由都是一個獨立路徑。

2. **Phase 2**：Phase 1 完成後，再執行 Phase 2。Phase 2 負責由較高優先級數的路由中選擇較適合的路由，而將其儲存於 Loc-RIB 中，每一路由也是一個獨立路徑。

3. **Phase 3**：當 Loc-RIB 已被更新完成後，再執行 Phase 3。Phase 3 負責散播 Loc-RIB 上的路由給相鄰的同濟自治系統，針對路由的聚集和訊息簡化使達到最佳化的處理，也是在此時相裡完成。

前面我們介紹 Cisco 公司所使用的路由屬性，以下也針對 Cisco 公司的路由選擇規則來介紹。BGP 也許由多個來源的宣傳收到相同的路由，但它只能再選擇其中一個優先級數較高的路由。當某一路由被選擇到時，必須將其存放於 IP 路由表內（或 Loc-RIB），並傳播給相鄰的自治系統，BGP 依照下列準則選擇最佳路由：

- 如果有某一路徑被描述為下一路徑（Next Hop），便往該路徑傳送。但如果該路徑已經到達不了，便將其刪除掉（或不存在下一路徑），再依照下列步驟判斷選擇最佳路由。
- 首先選擇較高衡權（Weigth）屬性的路徑。
- 如果路徑的衡權屬性相同，則選擇最高『本地優先』（Local Preference）屬性的路徑。
- 如果本地優先屬性相同，則選擇本路由器之 BGP 執行中產生的路徑。
- 如果路由都沒有起源屬性，則選擇較短『自治系統路徑』（AS_Path）屬性的路徑。

- 如果都是相同長度的 AS_Path，則依照『起源』( Origin )屬性選擇較低屬性的路徑( IGP 比 EGP 屬性低，EGP 比 Incomplet 屬性低 )。
- 如果起源屬性相同時，則選擇最低『多重出口鑑別』( MED ) 屬性的路徑。
- 如果所有屬性都相同時，則選擇最靠近相鄰的 IGP 路徑。
- 還是無法分出時，則選擇被標示為 BGP 之最小 IP 位址的路徑。

以上是針對網路環境來考量路徑選擇，但 BGP-4 允許以政策為基礎的路徑選擇，政策是由自治系統的系統管理者決定，並由規劃檔中設定成為路由值的一部份。政策的決定並不是通訊協定的一部份，但政策規格允許 BGP 實際應用時有多重選擇時，可作為路由器之間選擇的依據，並控制資料的重新分配，以配合路徑選擇的政策、安全性、商業性的考量。

# 習題

1. 何謂『下一跳躍』( Next-hop ) 路由選擇技術？

2. 何謂『靜態路徑選擇』( Static Routing ) ？其中包含哪兩種技術？

3. 何謂『熱馬鈴薯』( Hot-potato )方法？此方法可能會造成封包風暴，應如何改進來克服它？

4. 何謂『鏈路狀態路徑選擇』( Link-State Routing, LS Routing ) ？並簡述利用此方法建立路由表的步驟。

5. 採用鏈路狀態技術的路徑協定，可能會發生哪些異常狀態？

6. 何謂『距離向量路徑選擇』( Distance Vector Routing, DV Routing ) ？並簡述利用此方法建立路由表的步驟。

7. 採用距離向量技術的路徑協定，可能會發生哪些異常狀態？應如何克服？

8. 採用 DV Routing 技術在何種情況下，會發生路由表震盪現象？應如何克服？

9. 請利用 DV Routing 路徑選擇技術，推演出圖 6-49 網路拓樸圖中路由器 E 的路由表，圖中兩端點之間圍標是該路徑費用，並假設路由表的傳遞方向為 C→A、A→B、B→D、D→E。

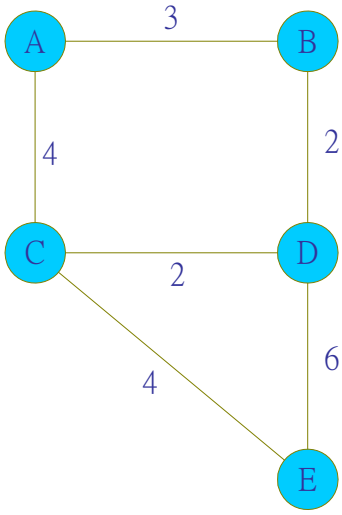



圖 6-49 DV Routing 範例

10. 請設計一個 Internet 網路架構圖，並說明網域內、自治系統內和自治系統之外的範圍。

11. 何謂網域內路徑選擇？一般都採用何種方式？

12. 何謂『路徑協定』(Routing Protocol)？請敘述其功能。

13. 請說明『路徑訊息協定』(Routing Information Protocol, RIP)的運作原理。

14. 請分辨 RIP-1 和 RIP-2 兩者協定，在協定運作方面有何不同？

15. 請說明『內部閘門路徑協定』(Interior Gateway Routing Protocol, IGRP)的運作程序。

16. 請說明 IGRP 協定針對 RIP-2 協動增強了哪些功能？

17. 請說明『加強型內部閘門路徑協定』(Enhanced Interior Gateway Routing Protocol, EIGRP)的運作原理。

18. 請說明 EIGRP 協定比 IGRP 協定增強了哪些功能？

19. 請說明『開放式最短路徑優先』(Open Shortest Path First, OSPF)之路徑協定的運作原理。

20. 何謂『拓樸圖資料庫』(Topological Database)？

21. 請說明採用 OSPF 協定的路由器之間，以何種方式互相交換訊息？

22. 何謂『外部閘門協定』(Exterior Gateway Protocol, EGP)？

23. 何謂『邊界閘門協定』(Border Gateway Protocol, BGP)？

24. 何謂『核心閘門』(Core Gateway)？

25. 何謂『非核心閘門』(Non-core Gateway)？

26. 請說明目前 Internet 網路成長快速，造成 IP 位址分配上有何嚴重問題？

27. 何謂『無層級網域間路徑選擇』(Classless Inter-Domain Routing, CIDR)？

28. 何謂『超級網域化』(Suppernetting)？

29. 請簡述 CIDR 的運作原理？

30. 請簡述 CIDR 的路徑選擇技術？

31. 請簡述 BGP 的路徑選擇技術？

**32.** 在採用 BGP 協定之下，路由器之間以何種方式來互相傳遞訊息？

**33.** 請說明 BGP-4 的運作方式？

**34.** 何謂『**路徑訊息資料庫**』(**Routing Information Base**)？它主要包含哪三大部份？

**35.** 在 BGP-4 協定裡，主要有哪四種訊息？並簡述其功能。

**36.** 請簡述 BGP-4 協定的路徑屬性？它和 RIP-2 有何不同？