

如何為您的網路選擇最佳路由器交換路徑

目錄

[簡介](#)

[處理序交換](#)

[中斷上下文交換](#)

[快速交換](#)

[最佳交換](#)

[Cisco Express Forwarding](#)

[哪種交換路徑最佳？](#)

[相關資訊](#)

簡介

適用於各種Cisco路由器和Cisco IOS®版本的交換路徑過多。哪個最適合您的網路，它們如何工作？本白皮書嘗試解釋以下每一條交換路徑，以便您能夠就適合網路的交換路徑做出最佳決策。

首先，檢查轉發過程本身。透過路由器轉送封包有三個步驟：

1. 判斷封包的目的地是否可達。
2. 確定通向目的地的下一跳，以及可以到達該下一跳的介面。
3. 重新寫入封包上的媒體存取控制(MAC)標頭，使其成功到達下一個躍點。

這些步驟中的每一個對資料包到達其目的地都至關重要。

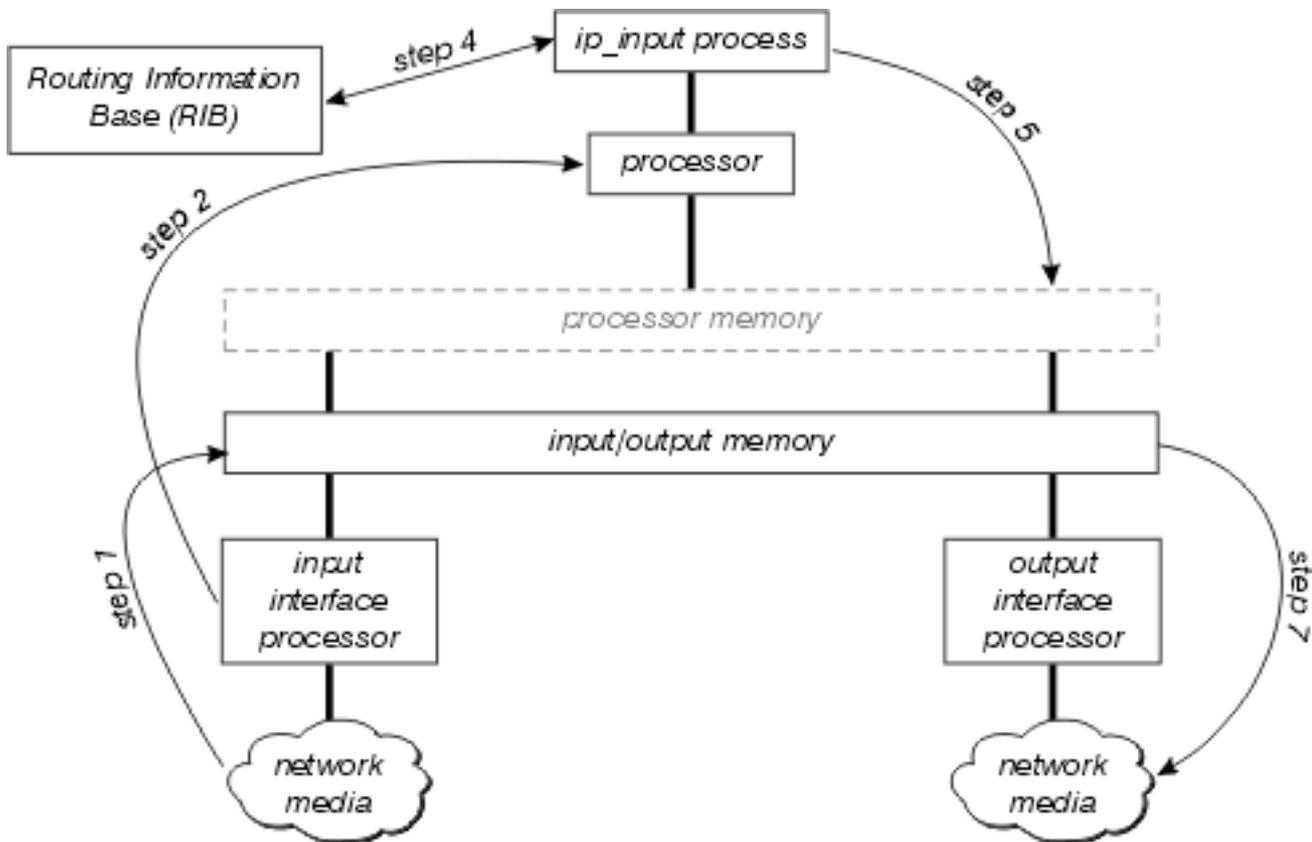
注意：在本文檔中，IP交換路徑用作示例；實際上，此處提供的所有資訊都適用於其他通訊協定的對等交換路徑（如果存在）。

處理序交換

過程交換是交換路徑中最小公分母；它適用於每個IOS版本、每個平台以及交換的每種流量型別。過程交換由兩個基本概念定義：

- 用於重寫資料包上MAC報頭的轉發決策和資訊取自路由表（來自路由資訊庫，或RIB）和地址解析協定(ARP)快取，或取自包含對映到直接連線到路由器的每台主機的IP地址的MAC報頭資訊的其他表。
- 封包是由IOS中執行的正常程式交換的。換句話說，轉發決策是由通過IOS排程程式排程的進程做出的，該進程作為路由器上其他進程（如路由協定）的對等體運行。路由器上正常運行的進程不會中斷，以處理交換資料包。

下圖說明了進程交換路徑。



詳細檢查此圖表：

1. 介面處理器首先檢測到網路介質上有資料包，並將此資料包傳輸到路由器的輸入/輸出儲存器。
2. 介面處理器產生接收中斷。在此中斷期間，中央處理器確定此資料包的型別（假定它是IP資料包），並在必要時將其複製到處理器記憶體中（此決定取決於平台）。最後，處理器將資料包放在相應進程的輸入隊列上，並釋放中斷。
3. 下次運行排程程式時，它會記錄ip_input輸入隊列中的資料包，並排程此進程運行。
4. ip_input運行時，它會諮詢RIB以確定下一跳和輸出介面，然後諮詢ARP快取以確定此下一跳的正確物理層地址。
5. 然後ip_input會重寫封包的MAC標頭，並將封包放在正確傳出介面的輸出佇列中。
6. 將封包從傳出介面的輸出佇列複製到傳出介面的傳輸佇列；任何出站服務品質都發生在這兩個佇列之間。
7. 輸出介面處理器檢測其傳輸隊列上的資料包，並將資料包傳輸到網路介質上。

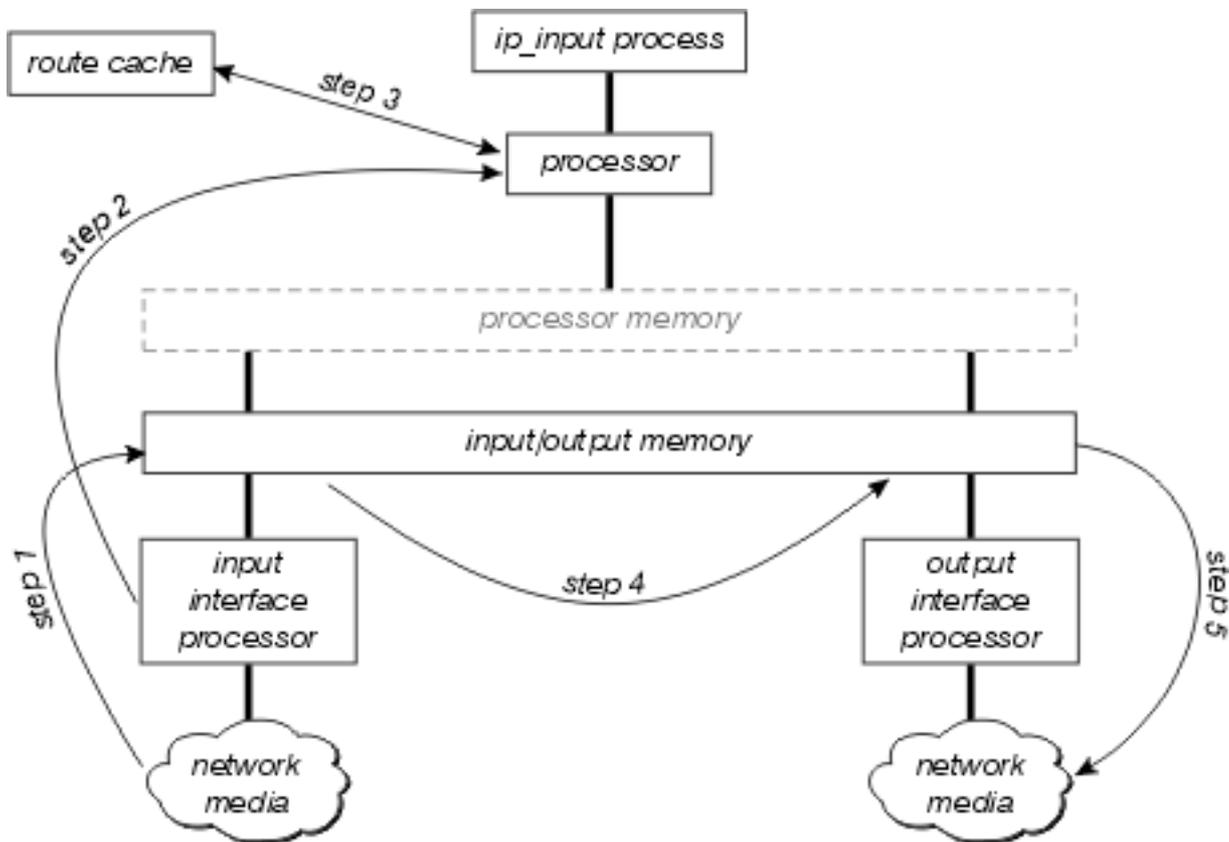
影響封包交換的幾乎所有功能(例如網路位址轉譯(NAT)和原則路由)都會在流程交換路徑中首次出現。這些功能經過驗證和最佳化後，可能會或可能不會出現在中斷情景切換中。

中斷上下文交換

中斷上下文交換是Cisco路由器所使用的第二種主要交換方法。中斷上下文交換和進程交換之間的主要區別如下：

- 處理器上當前正在運行的進程被中斷，以交換資料包。封包會按需交換，而不是僅在可安排ip_input程式時才交換。
- 處理器使用某種形式的路由快取來查詢交換資料包所需的所有資訊。

下圖說明了中斷上下文交換：



詳細檢查此圖表：

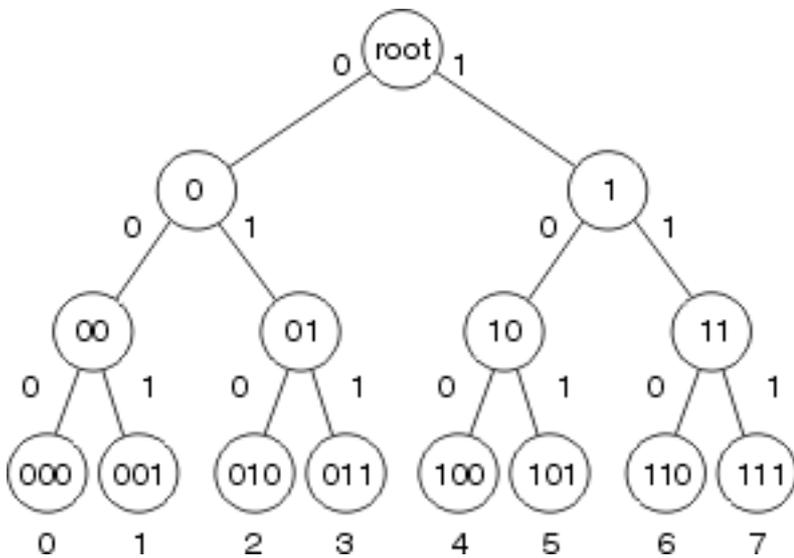
1. 介面處理器首先檢測到網路介質上有資料包，並將此資料包傳輸到路由器的輸入/輸出儲存器。
2. 介面處理器產生接收中斷。在此中斷期間，中央處理器確定此資料包的型別（假定它是IP資料包），然後開始交換資料包。
3. 處理器會搜尋路由快取，以確定資料包的目標是否可訪問、輸出介面應是什麼、通往該目標的下一跳是什麼，最後確定資料包成功到達下一跳時應具有哪個MAC報頭。處理器使用此資訊重寫資料包的MAC報頭。
4. 現在，資料包將被複製到出站介面的傳輸或輸出隊列（取決於各種因素）。現在接收中斷返回，並且發生中斷之前在處理器上運行的進程繼續運行。
5. 輸出介面處理器檢測其傳輸隊列上的資料包，並將資料包傳輸到網路介質上。

閱讀此說明後，第一個問題會浮現在腦海中：「快取中有什麼？」有三種可能的答案，具體取決於中斷上下文切換的型別：

- [快速交換](#)
- [最佳交換](#)
- [Cisco Express Forwarding](#)

快速交換

快速交換使用二叉樹儲存轉發資訊和MAC報頭重寫字串，以便快速查詢和參考。下圖說明了一個二叉樹：



在快速交換中，可達性資訊通過資料包目的地的二叉樹上是否存在節點來表示。每個目標的MAC報頭和出站介面都作為節點資訊的一部分儲存在樹中。二叉樹實際上可以有32個級別（為了便於說明，上面的樹是極其縮寫的）。

要搜尋二叉樹，您只需在要查詢的（二進位制）數字中從左開始（使用最有效的數字），然後根據該數字在樹中向右或向左分支。例如，如果您正在查詢與樹中的數字4相關的資訊，您將從右分支開始，因為第一個二進位制數字是1。您可以順著樹向下移動，比較（二進位制）數字中的下一個數字，直到到達結尾。

快速交換的特性

快速交換具有若干特徵，這些特徵是二叉樹結構和作為樹節點一部分儲存MAC報頭重寫資訊的結果。

- 由於路由表和快速快取內容（例如MAC報頭重寫）之間沒有關聯，因此構建快取條目涉及必須在進程交換路徑中完成的所有處理。因此，快速快取條目在資料包進行進程交換時建立。
- 由於ARP快取中的MAC標頭（用於重寫）與快速快取的結構之間沒有關聯，因此，當ARP表改變時，快速快取的部分必須失效（並通過封包的處理交換重新建立）。
- 快速快取只能為路由表中的任何特定目標構建一個深度（一個字首長度）的條目。
- 在快速快取中無法從一個條目指向另一個條目（MAC報頭和出站介面資訊應位於節點中），因此必須在構建快速快取條目時解析所有路由遞迴。換句話說，遞迴路由無法在快速快取中解析。

老化快速交換條目

為了防止快速交換條目丟失與路由表和ARP快取的同步，以及防止快速快取中未使用的條目不適當地消耗路由器上的記憶體，每分鐘會隨機使快速快取的1/20失效。如果路由器記憶體下降到非常低的水印以下，則每分鐘會有1/5的快速快取條目失效。

快速交換首碼長度

如果快速交換生成條目只能為每個目標生成一個字首長度，則其字首長度是多少？在快速交換的術語中，目的地是路由表中的單一可達目的地或主要網路。決定構建給定快取條目的字首長度的規則如下：

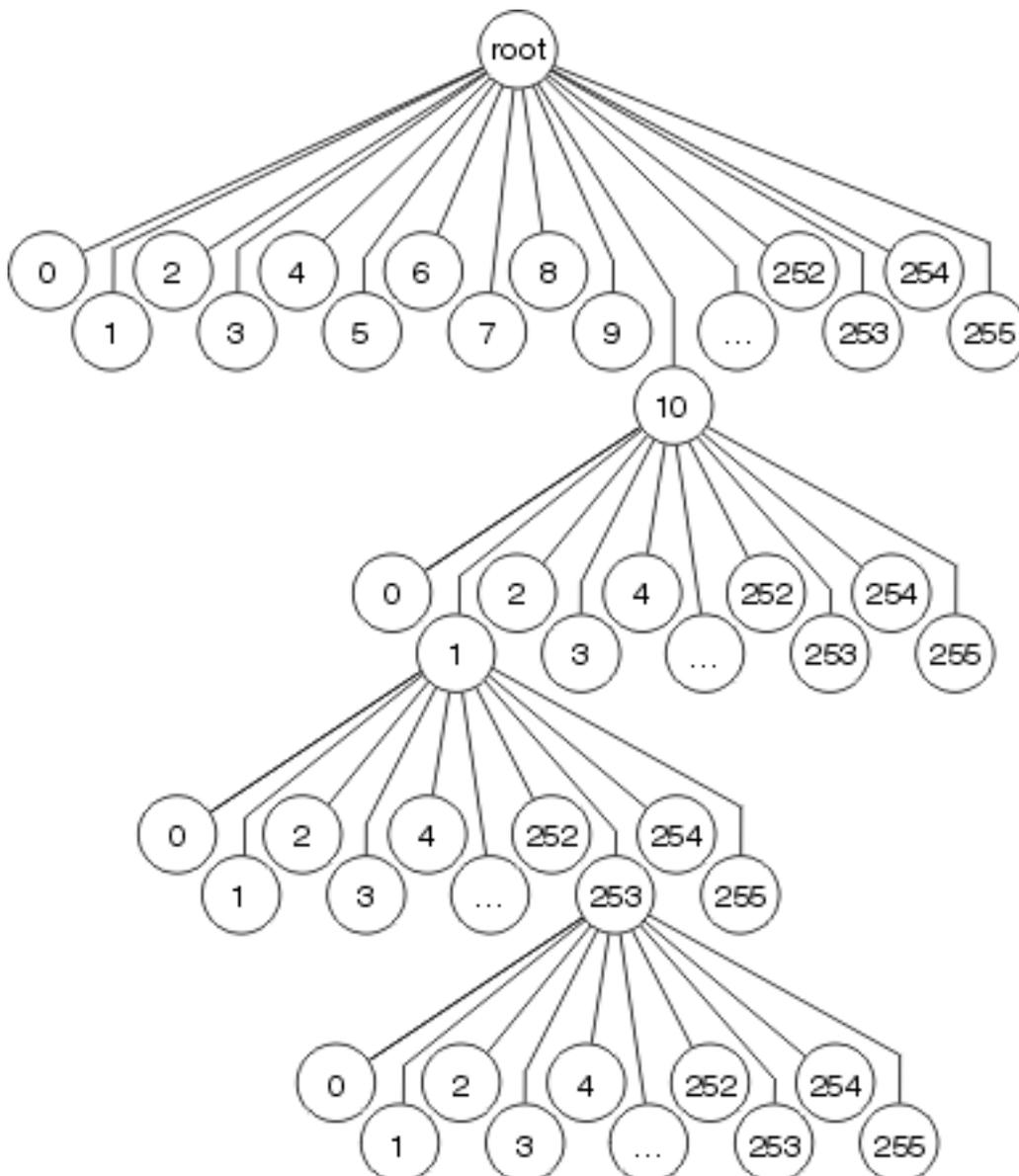
- 如果構建快速策略條目，則始終快取到/32。
- 如果根據ATM虛擬電路上的多重通訊協定(MPOA VC)建立專案，請一律快取到/32。
- 如果網路未劃分子網（它是主要網路條目）：如果是直接連線，請使用/32;否則，請使用主網路掩碼。
- 如果是超網，請使用超網的掩碼。
- 如果網路已劃分子網：如果是直接連線，請使用/32;如果有多條通往此子網的路徑，請使用/32;在所有其他情況下，請在此主要網路中使用最長的首碼長度。

負載共用

快速交換完全基於目的地；負載共用基於每個目標。如果特定目的地網路具有多條等價路徑，則 fast cache 對於該網路中可到達的每台主機都有一個專案，但所有目的地為特定主機的流量都跟隨著一條連結。

最佳交換

最佳交換在256路多路樹（256路樹）中儲存轉發資訊和MAC報頭重寫資訊。使用mtree可減少查詢字首時必須執行的步驟數，如下圖所示。

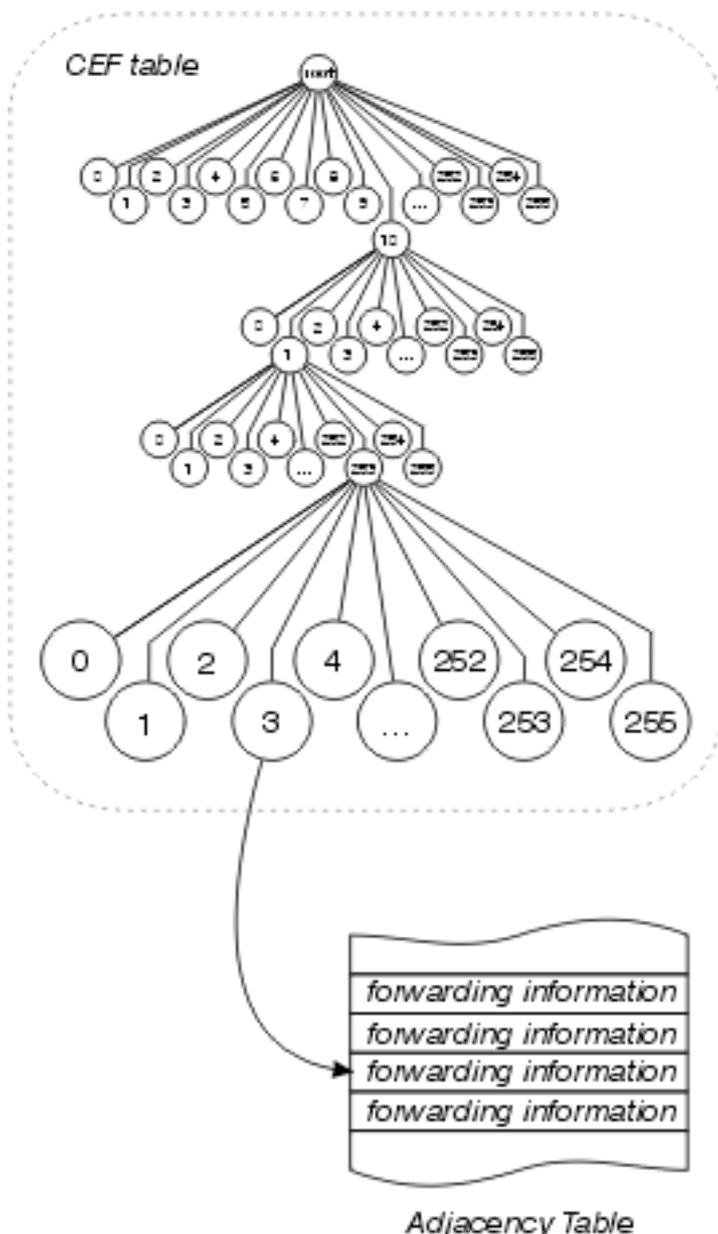


每個八位組用於確定在樹的每個級別要採用的256個分支中的哪一個，這意味著在查詢任何目標時最多涉及4個查詢。對於更短的字首長度，僅需要一到三個查詢。MAC報頭重寫和輸出介面資訊作為樹節點的一部分儲存，因此快取失效和老化仍會發生，就像快速交換一樣。

最佳交換還與快速交換一樣確定每個快取條目的字首長度。

Cisco Express Forwarding

Cisco Express Forwarding還使用256路資料結構來儲存轉發和MAC報頭重寫資訊，但它不使用樹。Cisco Express Forwarding使用trie，這意味著要搜尋的實際資訊不在資料結構中；相反，資料儲存在單獨的資料結構中，trie只是指向它。換句話說，思科快速轉發不是將出站介面和MAC報頭重寫儲存在樹本身，而是將此資訊儲存在稱為鄰接表的單獨資料結構中。



可達性資訊 (在Cisco Express Forwarding表中) 和轉發資訊 (在鄰接表中) 的這種分離提供了許多優點：

- 鄰接表可以與Cisco Express Forwarding表分開構建，這樣兩者都可以在不交換任何資料包的情況下構建。
- 用於轉發資料包的MAC報頭重寫不儲存在快取條目中，因此MAC報頭重寫字串中的更改不需要

使快取條目失效。

- 您可以直接指向轉發資訊，而不是遞迴的下一跳，以便解析遞迴路由。

基本上，所有快取老化都已消除，並且快取是根據路由表和ARP快取中包含的資訊預構建的。無需處理交換機任何資料包來構建快取條目。

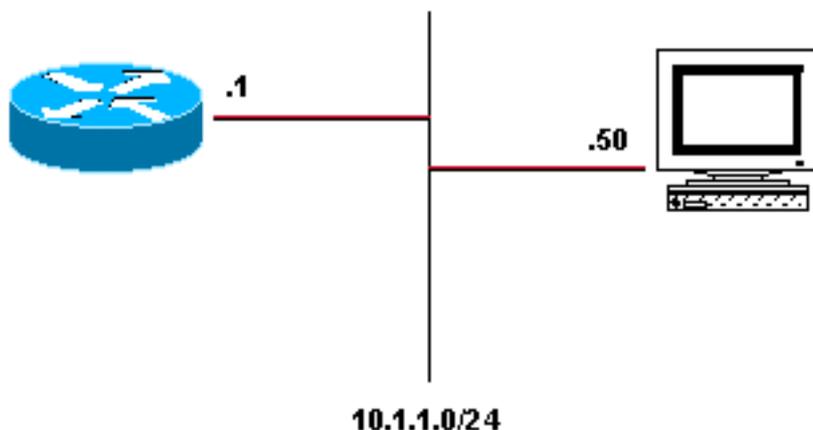
鄰接表中的其他條目

鄰接表可以包含MAC報頭重寫字串以外的條目和出站介面資訊。可以放入鄰接表中的各種條目包括：

- **cache** — 用於到達特定相鄰主機或路由器的MAC報頭重寫字串和出站介面。
- **receive** — 路由器應接收發往此IP地址的資料包。其中包括路由器本身上配置的廣播地址和地址。
- **drop** — 應該丟棄目的地為此IP地址的資料包。這可用於訪問清單拒絕或路由到NULL介面的流量。
- **punt** - Cisco Express Forwarding無法交換此封包；將其傳遞給下一個最佳交換方法（通常是快速交換）進行處理。
- **glean** — 直接連線下一跳，但當前沒有MAC報頭重寫字串。

收集鄰接關係

彙總鄰接條目表示應該直接連線特定下一跳，但是沒有MAC報頭重寫資訊。這些是如何構建和使用的？如下圖所示，運行Cisco Express Forwarding並連線到廣播網路的路由器預設會構建許多鄰接表條目。



預設情況下構建的四個鄰接表條目是：

```
10.1.1.0/24, version 17, attached, connected
0 packets, 0 bytes
  via Ethernet2/0, 0 dependencies
  valid glean adjacency
10.1.1.0/32, version 4, receive
10.1.1.1/32, version 3, receive
10.1.1.255/32, version 5, receive
```

請注意，共有四個條目：三人獲勝，一人獲利。每個接收條目代表一個廣播地址或在路由器上配置的地址，而收集條目代表所連線網路上的剩餘地址空間。如果收到主機10.1.1.50的封包，路由器會嘗試交換該封包，但發現該封包已解析為此緊密鄰接。然後，Cisco快速轉發發出訊號，表明

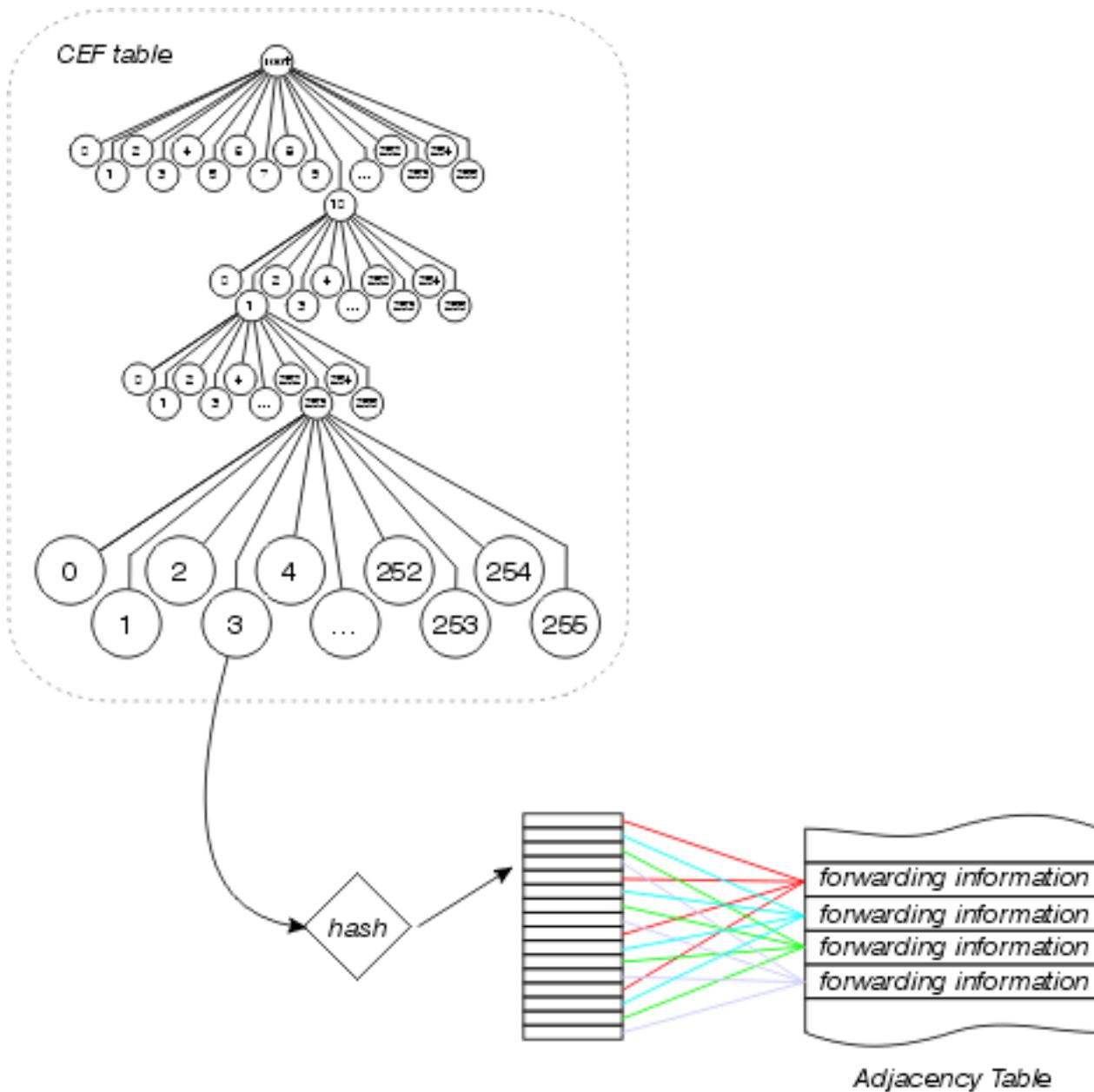
10.1.1.50需要ARP快取條目，ARP進程將傳送ARP資料包，並根據新的ARP快取資訊構建相應的鄰接表條目。完成此步驟後，鄰接表包含一個10.1.1.50條目。

```
10.1.1.0/24, version 17, attached, connected
0 packets, 0 bytes
  via Ethernet2/0, 0 dependencies
    valid glean adjacency
10.1.1.0/32, version 4, receive
10.1.1.1/32, version 3, receive
10.1.1.50/32, version 12, cached adjacency 208.0.3.2
0 packets, 0 bytes
  via 208.0.3.2, Ethernet2/0, 1 dependency
    next hop 208.0.3.2, Ethernet2/0
    valid cached adjacency
10.1.1.255/32, version 5, receive
```

路由器收到的發往10.1.1.50的下一個資料包將通過此新鄰接關係進行交換。

[負載共用](#)

Cisco Express Forwarding還利用Cisco Express Forwarding表和鄰接表之間的分離功能，提供比任何其他中斷上下文交換模式更好的負載共用形式。在Cisco Express Forwarding表和鄰接表之間插入一個負載共用表，如圖所示：



思科快速轉發表指向此負載共用表，該表包含指向可用並行路徑的各種鄰接表條目的指標。源地址和目的地址通過雜湊演算法來確定每個資料包使用哪個負載共用表條目。可以配置每個資料包負載共用，在這種情況下，每個資料包使用不同的負載共用表條目。

每個負載共用表有16個條目，其中根據路由表中的流量共用計數器劃分可用路徑。如果路由表中的流量共用計數器全為1（與多個等價路徑的情況一樣），則每個可能的下一跳都會從負載共用表接收相等數量的指標。如果可用路徑的數量不能均勻地劃分為16個（因為有16個負載共用表條目），則某些路徑將具有比其它路徑更多的條目。

從Cisco IOS軟體版本12.0開始，負載共用表中的條目數會減少，以確保每個路徑都有相應數量的負載共用表條目。例如，如果路由表中有三個等價路徑，則僅使用15個負載共用表條目。

哪種交換路徑最佳？

只要可能，您就會希望路由器在中斷上下文中交換，因為它至少比進程級交換快一個數量級。Cisco Express Forwarding交換肯定比任何其他交換模式更快、更好。如果正在運行的協定和IOS支援，我們建議您使用Cisco快速轉發。如果存在多個並行鏈路，流量應該通過它們進行負載共用，則情況尤其如此。存取[Cisco Feature Navigator](#)(僅供註冊客戶使用)頁面以確定您需要哪些IOS以提供

CEF支援。

相關資訊

- [如何驗證Cisco Express Forwarding Switching](#)
- [使用Cisco Express Forwarding排除並行鏈路上的負載均衡故障](#)
- [使用CEF進行負載平衡](#)
- [思科快速轉送支援頁面](#)
- [IP 路由支援頁面](#)
- [Cisco IOS交換服務組態設定指南12.1版](#)
- [技術支援 - Cisco Systems](#)