

路由器工作原理

張政賢、陳恆毅、林盈達

國立交通大學資訊科學系

摘要

路由器在互為連接的網路系統中扮演著一個舉足輕重的角色，而現今路由器扮演的角色愈來愈多元化而其所具備的功能也愈來愈多，可是路由器的基本工作原理及精神仍是不變的。

在本文中，我們先介紹在一個路由器上存在的協定及各個協定不同之處，再來我們將一個路由器的項目區分成標準化和非標準化來討論，進而以圖表來解說一個路由器的一般軟、硬體的架構，最後我們更以圖表示範解說路由器之間如何交換資訊及如何轉送網路封包。

1. 路由器協定

路由器主要的工作在於傳輸網路上的封包(packet)，對於網路傳輸協定必須要能辨別處理，尤其我們一般說路由器指的是 IP(Internet Protocol)路由器當然對於 IP 協定必須能夠支援，才能識別封包的標頭的資訊，必需得知封包的來源與目的地址等等資訊，以便完成封包的傳輸工作。

同時在網路上不只有像 IP 一樣單純傳遞封包的協定，為了 IP 能夠正常無誤的運作，對於網路上的一些控制協定也必須能夠支援，如 ICMP, ARP 和 RARP 等協定，對於這些協定將在下面簡介。

IP 協定的用途只局限在於資料的傳輸，欠缺錯誤回報與修正的模組，而這部份就由 ICMP(Internet Control Message Protocol)來填補。當封包的傳遞發生錯誤時，路由器會發出 ICMP 的錯誤訊息，傳回給送出封包的機器，以便修正錯誤。另外也提供了一

些查看遠端機器的命令，以便掌握網路的狀況。

ICMP 主要的工作列表如表一：

Echo Request & Reply	要求遠端主機回應的要求訊息及回應訊息。
Destination unreachable	封包無法到的達目的地的錯誤訊息
Time exceeded	經過太多節點的錯誤訊息
Parameter problem	封包標頭錯誤的錯誤訊息
Redirect	將封包的傳遞路徑改向

表一：ICMP 主要的工作列表

IP 協定是 Network Layer 的協定，在這個 Layer 之上 IP address 才有意義，之下的 Layer 都是憑 MAC address 來辨認，ARP(Address Resolution Protocol) 就提供了將 IP address 轉成 MAC address 的協定，使得下層的 Layer 知道將資料往哪送。而 RARP(Reverse Address Resolution Protocol) 的動作便是將 ARP 反過來，以 MAC address 得知 IP address。

有時在兩台路由器之間會存在著數條路徑，要如何決定選擇哪一條路徑是不容易的事，於是這些事就交給路由器去決定在路由器之間也會互相交換資訊，以求得最佳路徑的解，在 Internet 裡有兩種協定被廣為使用而成標準的，分別為 RIP (Routing Information Protocol)與 OSPF(Open Shortest Path First)。

RIP 的原理是將 RIP 的要求封包對自己的鄰近的路由器發送出去，接受到的路由器若採用 RIP 的協定，便將 Routing Table 傳回來，發出要求的路由器統計抵達每個目的位置所要經過的節點數，將最短的寫在自己的 Routing Table 上。

OSPF 跟 RIP 最大的不同處在於說它不是兩個鄰近路由器之間彼此交換，而是對整體網路廣播，而路由器收集這些訊息建構 Routing Table。另一個不同處在於路由器以距離參數(Distance metric)取代單純的經過節點數，以連結狀況更新距離參數，然後用動態規劃(Dynamic Programming)的演算法算出最短路徑，將下一個節點與計算出的總合的距離參數填入 Routing Table。

2. 路由器的工作項目

路由器的工作可以分成兩個部份，一是被訂立出來各家廠商都須遵守的標準協定，一則是說由各家廠商自行發揮的非標準的部份，在這部份各廠商由不同的方法去實現標準的協定。

對於 IP 協定所規定的 Subnet 與 Netmask 等規則是廠商所要遵守的第一個協定，不遵守的話就無法進行 Routing，若各家有各家的定址方式，各個廠商的路由器之間就無法交換封包。而各式封包的 Forwarding 也遵循著這協定。

還有 RIP 與 OSPF 這兩個 Routing Protocols 已經是網路世界中不可或缺的重要協定，在多數網路上的骨幹都是使用這兩種協定來交換資訊，不支援的話對排除錯誤與平衡負載將有很大的限制，甚至連傳送封包的工作都無法作。

有關於非標準的廠商實作部份包括程式碼與資料結構方面，標準只訂立溝通所必須的項目與介面，至於內部的程式架構均交給廠商發展。尤其對於 Routing Table 的建立，在標準部份的協定中只定義一些必須的欄位，對於如何實作並不加以限制。事實上這部份影響效率極大，好的實作方法將可帶來大筆的利潤。

對於封包的處理，如何以辨識最少的位元數而能辨識出此封包的處理方法或傳遞位置是很重要的課題，一般的做法是將整個封包的標頭先抓下來，根據標頭內的各項資訊分辨處理方式。不過對高速的路由器來說，這種方式的延遲太重了，所以改進為封包在傳遞進路由器之中時就已經開始分析，加快處理的腳步，尤其以傳送封包的加速最明顯，在目的的位置傳入路由器後，封包本體就不經處理直接經過路由器送出去了。

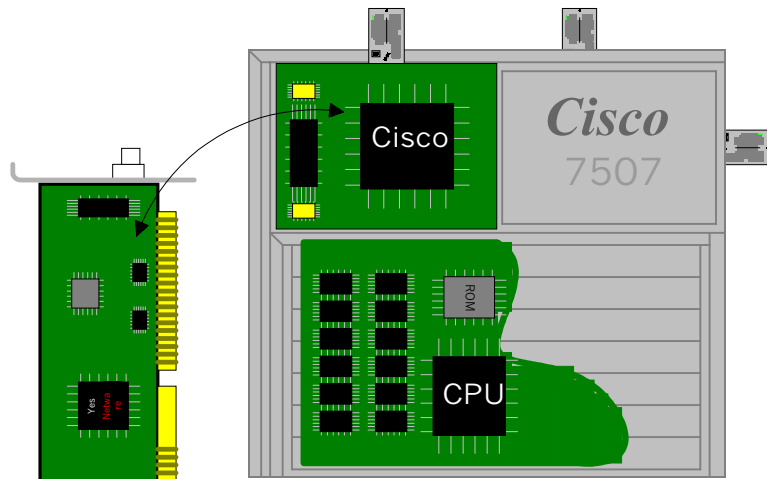
對於 Routing Table 的儲存格式，好的資料格式與搜尋演算法能使能快速搜尋到所需要的欄位，且並不需要很長的時間去建立與修正。在這方面各家廠商也都有自己的方式，如使用緊湊(hash)函數，平均樹(AVL tree)的使用，等等利於搜尋的資料結構都有被使用，也有的廠商是綜合使用，搭配種種的演算法，產生最快速的組合。

另外，除 RIP 與 OSPF 外，尚存在一些單一廠商使用的 Protocol，以 Cisco 來說，它使用了自行開發的 IGRP 等等的特殊協定，自己公司的產品都支援這些協定，在一個全是 Cisco 路由器的網路中使用這個協定將可有較佳的效率，但混雜在其中的他廠路由器將處處受限而不能有較好的效率。

3. 硬體架構

其實一台路由器的硬體架構和一台電腦的其本組成是相差不遠的，我們可以把一台路由器看成是一台電腦再加上網路設備而成。在一台路由器中，除了網路的傳送部份外，尚需要一個可以執行 routing algorithm 的運算部份、儲存 routing information(如 routing table)的記憶單元，當然囉，一台路由器運作時所執行的程式也需要儲存在路由器內，有些路由器是使用磁碟機，有些是使用 ROM 或是 Flash ROM 來存放執行欲使用的程式。我們從圖一中可以很清楚地看到，我們可以把一台路由器的網路傳送部份對映到一般的網路卡，而在圖一的下半部即可以比擬成一台針對 routing 作

設計的電腦。我們之所以要把路由器如此地比喻，最主要的是想告訴大家，您目前正在使用的電腦也可以搖身一變而成為一台路由器。



圖一 路由器硬體架構簡圖

由於路由器的處理資料的方式是一個個封包個別的處理的，也就是說，如果您的路由器所需要處理的資料不大，如一條 T1 線路的流量 (1.544Mbps)，還可以使用軟體路由器來連結您的網路系統；如果您的網路流量較大，最好還是使用硬體路由器來連結您的網路，免得因路由器的處理速度太慢而成為網路系統中的瓶頸。目前軟體路由路有兩種發展的方向，一是朝向建全路由器功能而努力，另一是整合其他針對封包做處理的功能，像是擋火牆(Firewall)、或是達成虛擬私人網路(VPN – Virtual Private Network)的網路位址軌換 NAT (Network Address Translation) 等等。

現在一般使用的軟體路由器，大多是使用 GateD 這一套軟體，這是一套歷史十分優久且發展得十分成熟的軟體路由器，在教育和研究的使用用途上是免費的而且有原始碼可以供參考，也

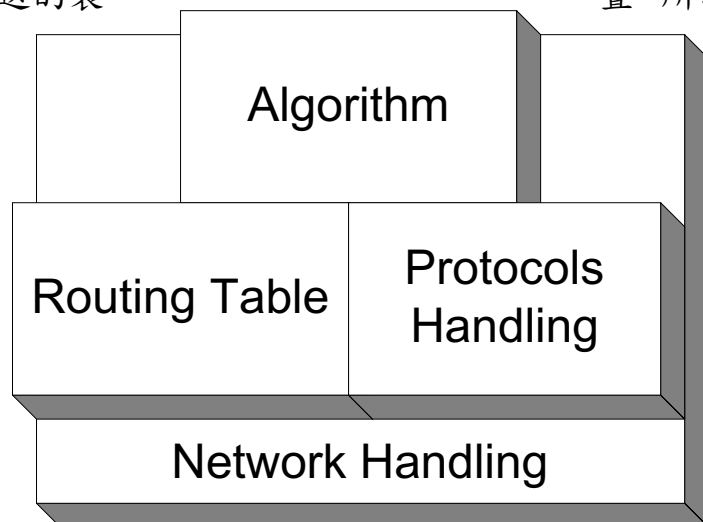
有不少的商業的路由器是使用了 GateD 這一套軟體來做研發的。尤其在 Linux 興起後，建設一台 Linux 工作站並使用 GateD 來做路由器，真是一個價格功能比很高的一種解決方式。如果您的伺服器是使用微軟的 NT 來做為作業系統的，也沒有關係，在 NT Server 或是 Workstation 版本中都內建了支援 RIP 和靜態路由的路由器功能，您只需要在網路設定內將 IP 轉送功能打開，並設定路由即可。

由於目前公司林立，Intranet 方興未艾，所以有很多使用路由器的時機只是將一個 Intranet 的網路連接上租借的網路骨幹而已。這個時候實在是不需太多路由器的功能，其實也只需要靜態路由路徑即可。而這一種環境中又可能會用到虛擬私人網路或是擋火牆，所以第二種方式的軟體路由器就因應而生了。這一種的軟體在 Unix 或是 Wintel 平台都有，而以 Wintel 平台的為最多且設定、使用較為簡便。不過筆者認為這類的軟體的設計立足點不再是從路由器開始，而是如何解決 Intranet 網路連結著眼。以 WinRoute 這一套軟體為例，它可以使用 PPP 的 Modem 或是 ISDN 的撥接，也可以使用擁有單一 IP 位址的 Internet 或是 Cable Modem 的連線來連結整個 Intranet 到 Internet 上。

4. 軟體架構

在上一節中，我們說明了一台路由器的硬體架構，也提示了：一般的電腦也可以成為舉足輕重的路由器。由於一般的電腦也可以接上網路傳送的裝置，所以

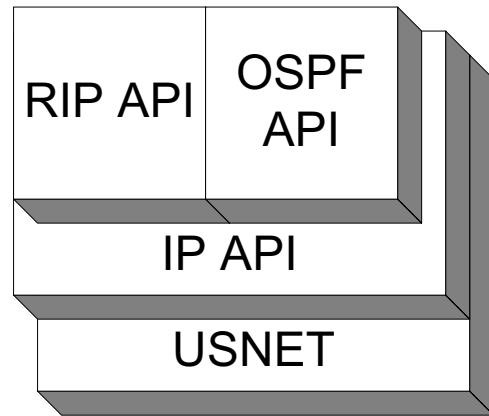
我們要將一台電腦當成一台路由器來轉送網路上的封包最重要的就是一軟體，也就是說我們必需把之前所提過一台路由器的項目全部實作在一般電腦的軟體執行平台上。我們可以從圖表二可以得知一個軟體



圖二 路由器所需的軟體元件

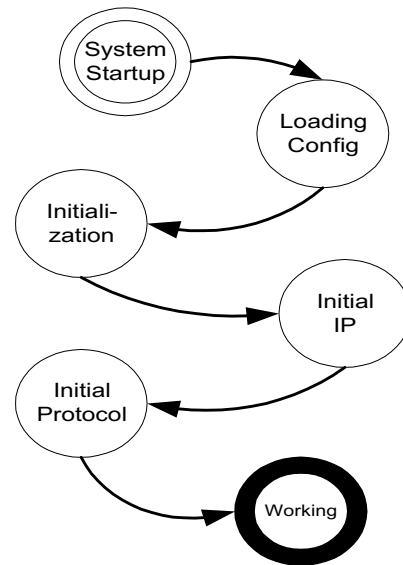
架構的路由器需要那些部份的軟體元件。在圖二中我們可以看到在最底層的軟體路由器架構是關於網路傳輸的處理函式，來處理每個封包的傳送與接收工作，再來是存放如 routing table 的資料結構和路由器協定的固定處理流程，當然囉，在這兩個部份都需要一些演算法來處理。

我們這一次也解析查看了 Phase 2 這一家專門在寫路由器程式碼的廠商的產品，我們可以發現他們的程式碼有如圖三的階層狀態，雖然他們的程式碼是用 C 寫成，可是卻可以發現它有濃厚的 OOP 的概念，他們先寫作可以處理 IP routing 的程式框架，再在其上套上支援不同路由器協定的元件即可，如此的架構對於目前眾多路由器協定的支援可謂是十分聰明的一種方式。在圖三，我們可以看出來整個 Phase 2 的程式碼是架構在 USNET 之上，也就是 Phase 2 的程式碼不從事和硬體交涉的封包讀取和接送，所以它就不必煩惱在不同路由器之間的硬體差異。



圖三 Phase 2 的 API 階層示意圖

如圖四，當使用 Phase 2 的路由器開機時，它會先去載入一些先前的網路設定，像是 IP 位址，靜態路徑及一些有關路由器效能的設定值等等。再來開始啟始化 IP 網路及一些路由器協定，最後進入了開始工作的狀態。



圖四 Phase 2 的程式流程示意圖

5. Control Packet 執行流程

當路由器收到一個封包時，將會對此封包的標頭加以解析，會先分析它所屬於的協定，當此協定屬於一般的 IP 協定時，則會將此封包轉送出去，不然會送到對應的協定模組去處理。在路由器上大多有 ICMP、RIP、OSPF 等等協定的模組，專職處理對

應的協定，以回應訊息或更新本身資訊。如圖五所示。

Phase 2 的做法是在路由器初始化時將各模組給註冊起來，當收到一個封包，取出標頭分析其的協定種類後，就檢查註冊資訊，查出相對應的協定處理模組，在模組內再將封包做進一步的解析，取出該協定的標頭，執行此協定命令，送出回應訊息或更新系統內的資訊。如 RIP 的 request 跟 respond 分別會送出 Routing Table 或更新 Routing Table Entry。

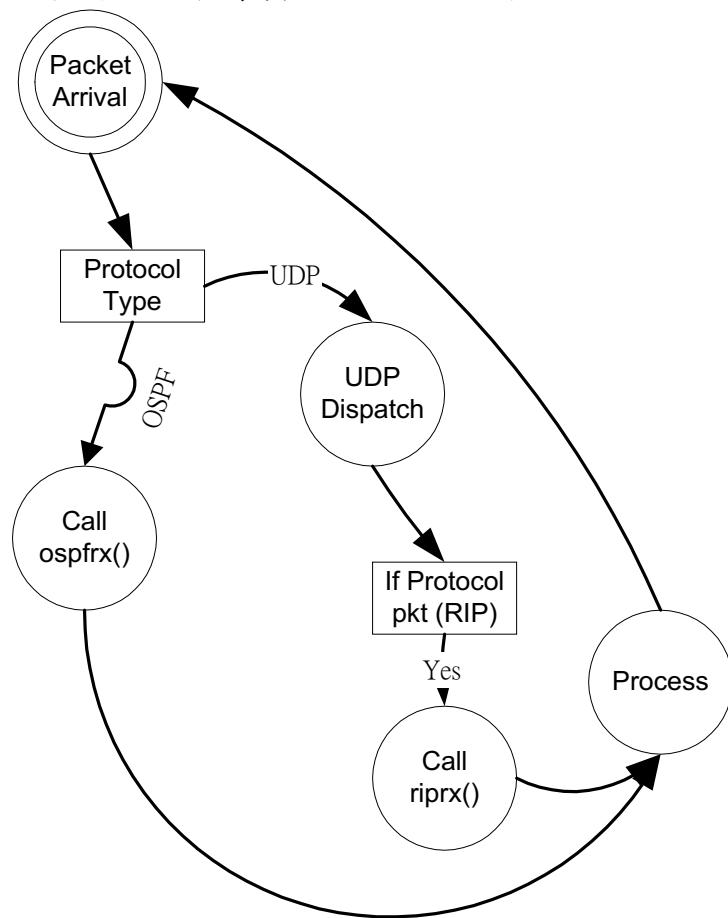
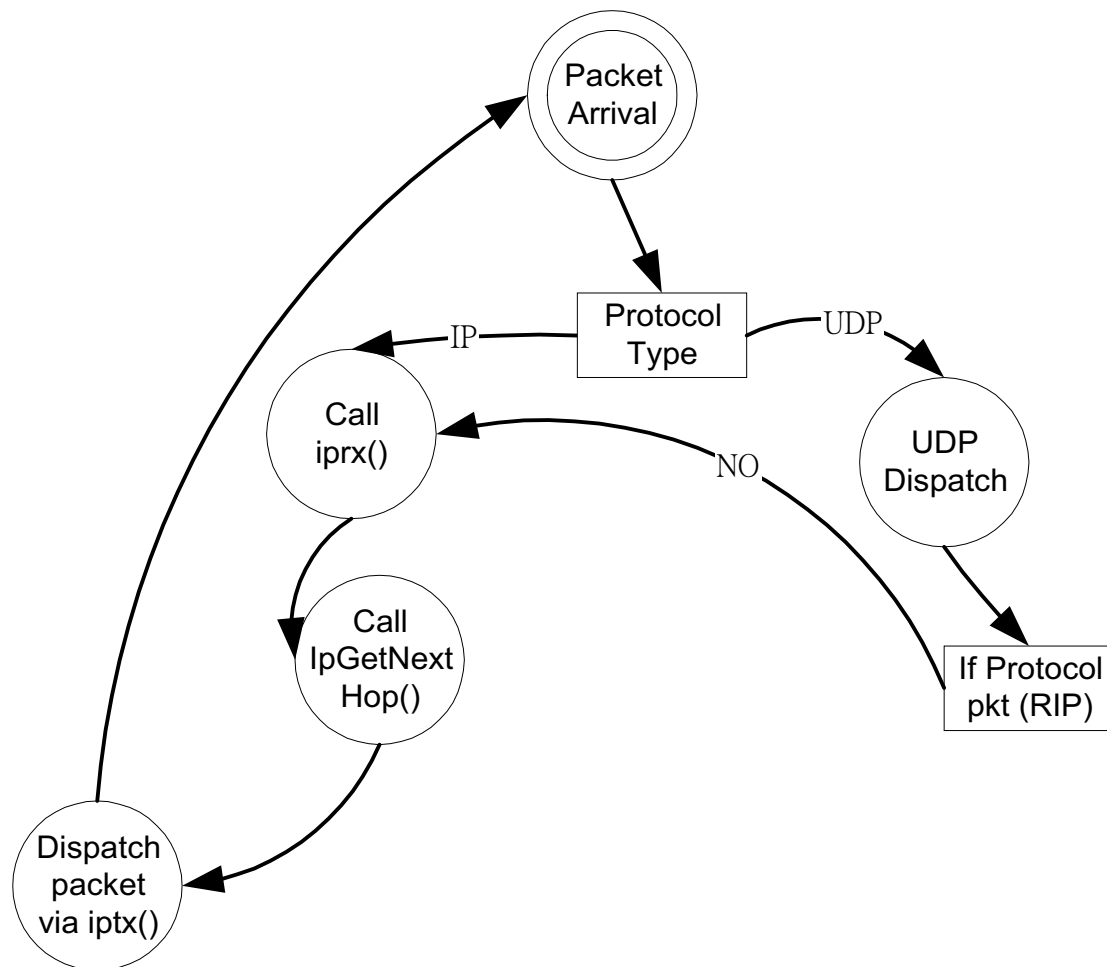


圖 五 控制封包的執行流程

6. Data Packet 執行流程

一個路由器最主要的工作當然就是把每一個經過的封包快速且正確地轉送到特定的網路上，所以我們現在來看看一個路由器將資料封包從接收到轉送出去的流程，首先，讓我們先來看看圖六。

一開始，接收一個傳送給路由器的封包，然後暫存在路由器的網路部份，再來路由器開始處理這一個封包的標頭，以別辨這一個封包的網路協定類別，由於如果一個封包是資料封包，那可能是 IP 封包或是 UDP 封包，所以路由器還需要得知一個 UDP 封包是否是一個 RIP 的封包；如果這一個封包是資料封包，那這



時會呼叫 iprx()來處理這一個封包，查看它的目標位址後再呼叫 IpGetNextHop()來取得要轉送路徑，並使用 iptx()來分送這一個資料封包。

7. 總結

在看似無遠弗界、天涯若比鄰的網際網路中就是靠著不少像是路由器這一類的連接器將各個分散在世界各地的網路系統連接起來。然而在現今的網路使用中，光是網網相連亦不敷我們使用了，我們現在追求的卻是更好的傳輸品質與效率，也由此，愈來愈多的 Routing 協定也因應而生，對於支援服務品質(QoS)的 Routing 協定正在被學界跟業界研究中，在不久的將來應可以享受穩定頻寬的網路，而不再因為網路擁塞而產生時間的浪費；路由器的設計也愈來愈新穎，ATM 網路的交換器(Switch)技術的刺激，誕生了 Level 2 的交換集線器(Switching Hub)及具有封包交換傳送功能的 Level 3 的交換器，使用硬體的交換取代軟體的路由，轉送封包的速度也提升了；服務也愈來愈周到，人們的生活

也因此而不斷的改進。

8. 參考資料

- TCP/IP and Related Protocols, Uyles Black, McGraw-Hill, 1998
- Routing in the internet, Christian Huitema, prentice Hall, 1995
- Phase 2 路由器 Source Code
- RFC 791 Internet Protocol. J. Postel. September 1981.
- RFC 792 Internet Control Message Protocol. J. Postel. September 1981.
- RFC 1058 Routing Information Protocol. C.L. Hedrick. June 1988.
- RFC 1723 RIP Version 2 - Carrying Additional Information. G. Malkin. November 1994.
- RFC 1247 OSPF Version 2. J. Moy. July 1991.
- RFC 2328 OSPF Version 2. J. Moy. April 1998.
- RFC 2386 A Framework for QoS-based Routing in the Internet. E. Crawley, R. Nair, B. Rajagopalan, H. Sandick. August 1998
- GateD, routing application, reference URL <http://www.gated.org>
- WinRoute, routing application, reference URL <http://www.winroute.com>