網域拓撲探測與延遲測量

論文領域: 網路管理

孫文駿 林盈達

國立交通大學資訊科學系
新竹市大學路 1001 號
TEL：（03）5712121 EXT. 56667
E-MAIL：gis88501@cis.nctu.edu.tw, ydlin@cis.nctu.edu.tw

主要聯絡人：孫文駿，FAX: (03)571-2121 EXT 59263

摘 要

網際網路是一個複雜的迷宮，想要了解一個網域須掌握兩個重要的元素，一是網域的拓撲(topology)，另一個是網域內傳輸的延遲(delay)。傳統探測一個網域的方法，採用"ping"的工具程式逐一 ping 向每一個節點，再依照回傳的結果勾勒出遠端網域的架構，這種方式費時又容易出錯。在此我們利用 SNMP(Simple Network Management Protocol)所提供的 MIB(Management Information Base)資料庫，讀取遠端路由器(router)或主機(host)上的路由表(Routing Table)，來探測遠端網域的拓撲，較為快速且正確。我們整理出一套完整的演算法，利用 SNMP 及 ICMP(Internet Control Message Protocol)來探測一個給定但未知網域的拓撲(topology)並測量網域內部節點間資料傳輸的延遲(delay)。另外，將介紹市面上易於取得的分享軟體(shareware)，來幫助我們完成拓撲探測(topology probing)和延遲測量(delay measurement)的工作。我們並以交大校園網路為例，進行一次完整的拓撲探測和延遲測量。

關鍵字: ICMP、SNMP、MIB、topology、delay、probing、measurement、domain

一、動機

網際網路的世界越來越龐大，全世界使用 Internet 的人口與日俱增，根據 Internet Software Confortium 的統計[1]，到 1999 年 7 月，直接連上 Internet 的主機(host)大約有

56,218,000 台，包括了 240 個國家。較 1999 年 1 月的 43,230,000 台，約增加了 12,988,000 台，成長的速度非常驚人。Internet 上的節點數大量增加，使得 Internet 的拓撲(Topology)更加複雜，網管的困難度也相對提升。對一般使用者而言，Internet 有如迷宮般的難以捉摸。

傳統的方法中，想要探測一個遠端網路的拓撲，就是利用工具程式"ping"。從一個網域的第一個 IP Address 逐一 ping 到最後一個 IP Address，分別記錄封包的路徑和傳輸的時間，再依照這些資料，勾勒遠端網域的架構。這種方式工程耗大，而所得到的資訊又不完整，在描繪網路拓撲的時候很容易出錯。

因此我們希望找出一套新的方法，利用這套方法，可以讓我們正確而快速地了解遠端特定的網域。也就是說這套方法可以完成兩件事，一是探測網域拓撲(topology probing)，一是測量網域傳輸延遲(delay measurement)。

二、拓撲探測和延遲測量的原理

了解一個的網域有兩個重要的元素，一是網域的拓撲(topology)，另一個是網域內傳輸的延遲(delay)。第一步要先找出特定網域中完整的拓撲，這包括網域中主機(host)的數量、機器連接(connect)的情形、IP address 的配置和子網域(subnet)分配。第二步是對網域中每一個 hop 的 delay 進行測量(measurement)接下來我們將分別介紹拓撲探測和延遲測量的其本原理。

2.1 拓撲探測的原理

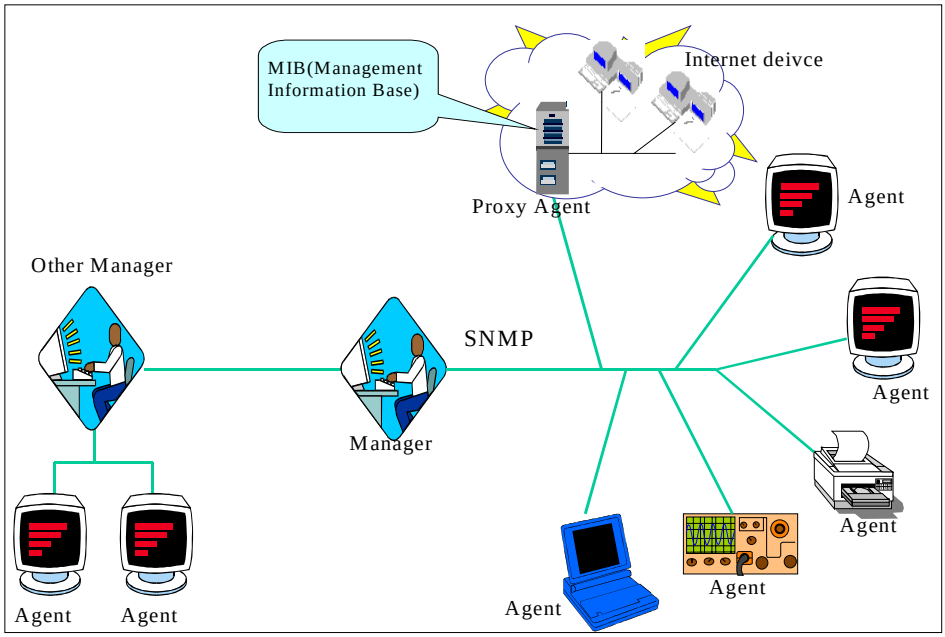
查出網路拓撲最好的方法是取得路由表(Routing Table)。路由表中記錄了 IP Address 的前置地址(prefix)及該往那個方向傳送。支援 SNMP(Simple Network Management Protocol)[2]的設備(device)會把自己的路由表放在 MIB(Management Information Base)[3]資料庫中，因此我們可以透過 SNMP 的訊息取得遠端路由器(Router)的路由表。

2.1.1 SNMP 簡介

Internet 上各種裝置(device)種類繁多，各家商廠的網管系統並不相容。SNMP 是一個開放的網路協定標準，用以協助及整合 Internet 的管理工作。SNMP 第一版在 1990 年五月由 RFC 1157[4]所定義完成。在 RFC 1441[5]到 RFC 1452[6]中，針對第一版的缺

點加強改善，完成了目前通行的 SNMPv2。SNMP 推出後廣受好評，而且很快地爲各家廠商接受，主要的原因在於它”simple”——如同它的名字。

在 SNMP 的網管環境如圖一，它把網路上的設備分爲兩類，一類叫管理者(Manager)，另一類叫受管者(Agent)，受管者分散在網路中，負責監視並記錄網路裝置的運作情形、資料流量、傳送過程等資訊。而管理者視情形向受管者取回各項記錄，經分析判斷後，再向受管者下達指令調整運作的方式，以增進網路的服務能力。



圖一：SNMP 網管環境

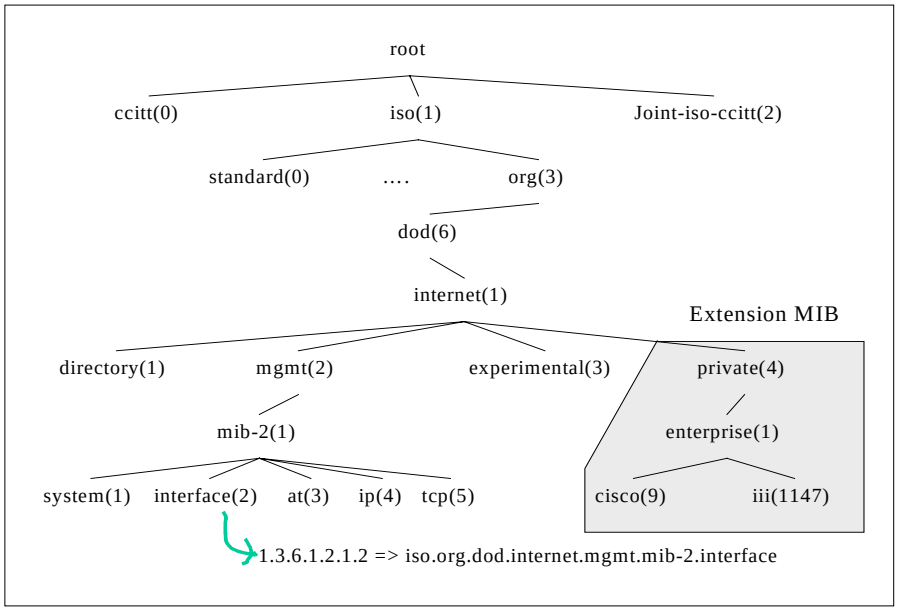
但是許多較早時期的設備並不支援 SNMP。針對這個問題，提出了中介受管者(Proxy Agent)的角色。中介受管者負責和不支援 SNMP 的設備溝通，而後提供資訊給管理者並接受管理者的命令調整網路設備的運作。

2.1.2 MIB 簡介

管理者(Manager)要如何取得每一個受管者(Agent)紀錄的網管資訊呢？不同的網路設備各有不同的特性，管理者須要一個共通而且明確的方式取得不同網路設備的網管資訊。爲了建立一個共同的資料存取介面，定義了 MIB(Management Information Base)。

MIB 是一種資料庫，受管者把所有網管相關的訊息放入資料庫中以供存取。資料庫需要有索引才能正確的找到資料，MIB 所使用的索引稱 OID(Object IDentifier)。OID 將 MIB 架構成一個樹狀的資料庫[7]，如圖二所示，每一項資料就是一片樹葉(leaf)，相關的資料形成一棵子樹(subtree)。由上而下，依照相關的組織、特性，而分佈成整棵樹

(tree)。樹上的的每一個節點(node)都有一個正整數的號碼，這個號碼在節點的兄弟(brother)之間是唯一的。透過這個號碼就組成了索引資料庫的 OID。當我們要拜訪 MIB 中某一個節點時，由樹根開始，逐一向下指定每一層子樹的 OID 號碼，我們便可找到要拜訪的樹葉。依照 OID 的規定，每一層的數字間以句點(dot)區隔。



圖二：MIB 的資料結構

以圖二爲例，如果我們要尋找「interface」這個節點，我們由樹根開始，經由 iso(1)到 org(3)到 dod(6)到 internet(1)到 mgmt(2)到 mib-2(1)到 interface(2)，因此 OID 爲 1.3.6.1.2.1.2。管理者只要將 1.3.6.1.2.1.2 的訊息送給受管者，就可以取得「interface」的資料。

2.1.3 利用 MIB 資訊找出網路架構

MIB II[8]屬於標準的 MIB 資料庫，如圖二在整個 MIB 中被定義在 1.3.6.1.2.1 的位置，主要是爲了路由器的控制管理所設計的。MIB II 共分爲 system、interface、at、ip、icmp、tcp、udp、egp、transmission、snmp 共 10 個類別(Group)，記錄各種控制、傳輸、路由等資訊，這裡我們所關心的是它的第二個類別 interface 和第四個類別 ip。Interface 類別下有一個 Interface Table，Interface Table 可以告訴我們這台路由器每一個通訊埠(port)現在的使用狀況。在 ip 類別下，我們可以取得 Routing Table 及 Net to Media Table。Routing Table 紀錄了不同的 ip address 應該送往那一個節點轉送，而 Net to Media Table 告訴我們 Router 的每一個埠(port)和哪些機器直接相連。Routing Table 及 Net to Media Table 詳細的欄位分別列於表一及表二。

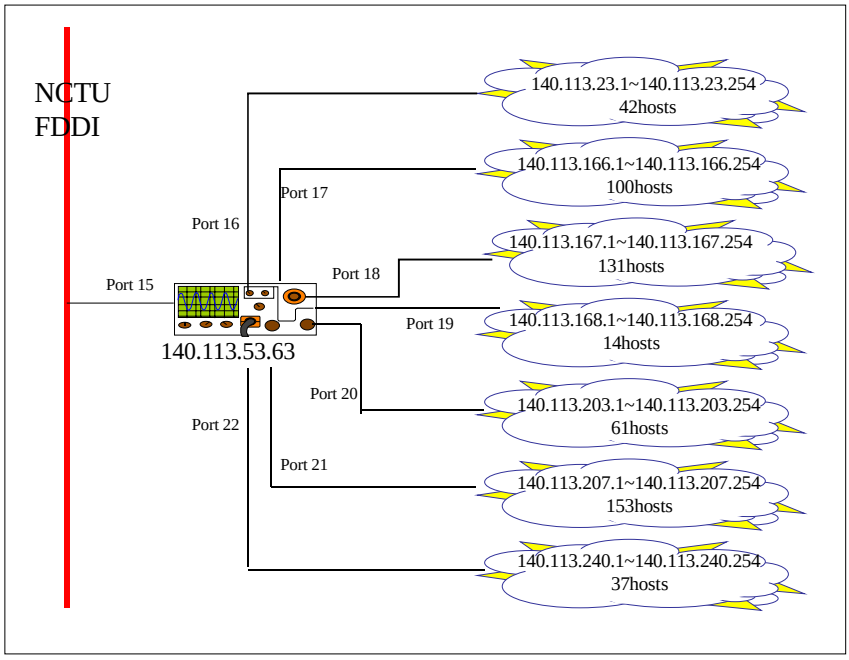
欄位名稱	資料型態	說明
ipRouteDest	IPAddress	destination IP address
ipRouteIfIndex	INTEGER	經由那一個 port 路由
ipRouteMetric1	INTEGER	primary routing metric(和 iprouteproto 有點關)
ipRouteMetric2	INTEGER	routing metric 替代方案
ipRouteMetric3	INTEGER	routing metric 第三個方案
ipRouteMetric4	INTEGER	routing metric 第四個方案
ipRouteNextHop	IPAddress	下一個 hop 的 IP 位址
ipRouteType	INTEGER	route 方式(直接給誰/還要轉接)
ipRouteProto	INTEGER	路由使用的通訊協定
ipRouteAge	INTEGER	這一個路由記錄上一次修改現在經過幾秒
ipRouteMask	IPAddress	路由位址的 Net Mask

表一：Routing Table 欄位說明

欄位名稱	資料型態	說明
ipNetToMediaIfIndex	INTEGER	interface 的索引值
ipNetToMediaPhysAddress	OCTET STRING	和 interface 直接相連機器的 MAC 位址
ipNetToMediaNetAddress	IpAddress	和 interface 直接相連機器的 IP 位址
ipNetToMediaType	INTEGER	ipNetToMedia 記錄的型態

表二：Net To Media Table 欄位說明

藉由以上兩張表，我們就可以找出一個路由器和那些其他的路由器相連接，而路由器本身直接管理那些子網域(subnet)，當然我們也可以用同樣的方式找出每一台主機的串連狀況。在此，我們比較關心的是路由器間的串連和子網域的分配。圖三，我們以交通大學資訊科學系的一顆路由器爲例，說明透過 MIB 的查詢，我們可以取得什麼樣的資訊。



圖三：一台路由器中 MIB 記錄的資訊

我們可以逐一查詢和 140.113.53.63 直接相連的每一顆路由器，然後再查取與之相連的路由器，逐步向外，便可找出整個網域(domain)內的拓撲 topology。在稍後的章節中我們會更仔細說明網域拓撲探測(topology probing)的詳細步驟。

2.2 網域延遲測量的原理

測量遠端某一節點與自己通訊間詳細的延遲情形，要追蹤(trace)封包經過的路徑，而後逐一測量每一段的延遲時間。測量網路上遠端兩個節點(A 點與 B 點)間的延遲，需要送出一個封包通過指定的兩個節點後返回，將封包往返的時間扣除到達 A 點與自 B 點回來時間，方可得到封包通過 A、B 兩點間的延遲時間。

2.2.1 ICMP 簡介

TCP/IP(Transmission Control Protocol/Internet Protocol)是一個組合式的通訊協定，他是藉著多個協定互相分工，彼此支援來完成資料傳輸的工作。其中 IP 協定負責提供了資料包(datagram)的傳送、路徑路由、檢核(checksum)等功能，但是有異常狀況時，接收端(receiver)必須將異常狀態反應給發送端(sender)，ICMP(Internet Control Message Protocol)[9]就是扮演通知各種異常狀況的角色。

ICMP 是在 IP 層(Internet Protocol Layer)上層的通訊協定，ICMP 的資料也就是透過 IP 的封裝(encapsulated)傳送給發送端。ICMP 中定義了九種訊息格式(message type)[10]。分爲兩類，第一類錯誤訊息(error message)有五種訊息格式，第二類資訊訊息(information message)有四種訊息格式。「錯誤訊息」當 IP 傳輸有異常狀況發生時，路由器(Router)或接收端(Receiver)主動向發送端(Sender)送出錯誤訊息，提醒發送端停止或調整資料的傳送。「資訊訊息」是使用者主動提出要求，希望目的機器回應簡單的訊息。五種錯誤訊息格式整理如表三，四種資訊訊息格式整理如表四。

訊息名稱	意 義
Destination Unreachable	Packet Could not be delivered
Time Exceeded	Time To Live field hit 0
Parameter problem	Invalid header field
Redirect	Teach a router about geography
Source Quench	Choke packet

表三：ICMP 中五種錯誤訊息

訊息名稱	意 義
Echo Request	Ask a machine if it is alive
Echo Reply	Yes, I am alive
Timestamp Request	Ask a machine timestamp
Timestamp Reply	Echo timestamp request

表四：ICMP 中四種資訊訊息

2.2.2 Echo Request/Echo Reply 訊息

Echo Request/Echo Reply 是 ICMP 中最爲常用的兩種訊息格式(message type)。Echo Request 就是要求目的機器(destination machine)做一個簡單的回應，以檢查網路是否暢通以及目的機器是不是仍然正常(alive)。Echo Reply 訊息就是專門用以回應 Echo Request，告訴查詢的機器「我很好！」。Echo Request/Echo Reply 屬於資訊訊息(information message)唯有在使用者下達指令時才會發出訊息。

最常用來發送 Echo Request 訊息的公具程式就是"ping"。藉由 ping 的動作，使用者可以馬上得到兩個項資訊，第一項資訊是目的機器(destination machine)是否仍然正常運作，第二項資訊是一個封包來回須要多少時間。Ping 程式只是一個簡單的迴圈架構，送出一個 ICMP 封包後，就一直傾聽(listen)，等待回傳的訊息。如果超過使用者指定時間就告訴使用者「Time out」。

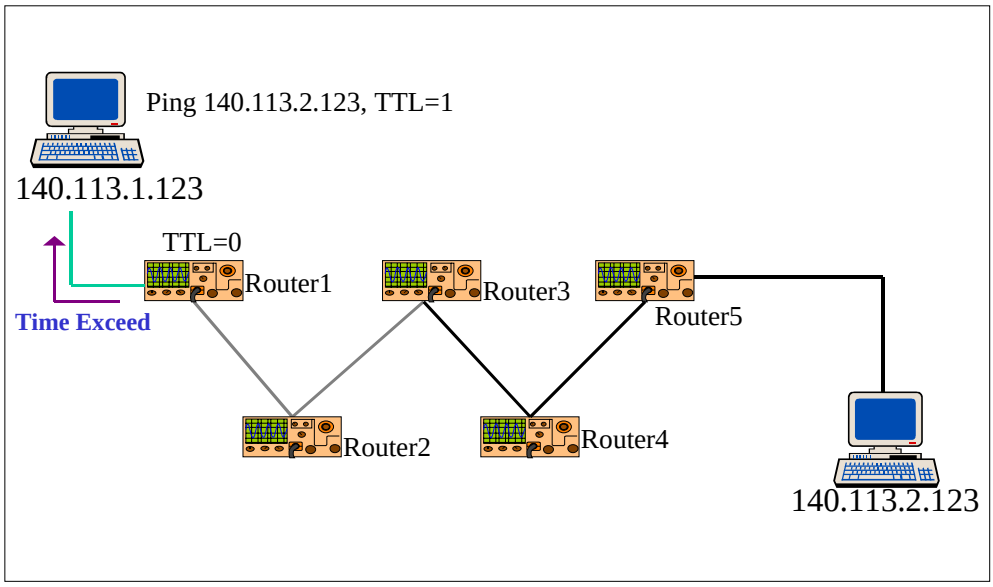
2.2.3 Time Exceeded 訊息

在 IP 標頭(IP Header)內，有一個欄位叫 TTL（Time To Live），長度 8bits，爲了防止封包在網路中漫無止境地傳送而設，與路徑的選擇無關。當封包從發送端(source)發出時， TTL 欄位就被填入一個正整數，此後這個封包每經過一個 hop，TTL 欄位的數字就減 1。當 TTL 遞減爲零，這個封包就會被視爲過時的資訊而被收到它的 hop 直接丟棄。

通常，在封包一開始當被發送端(source)發送出去時，TTL 欄位會填上最大正整數 255。如果一個封包的 TTL 值被遞減到了 0，代表這個封包迷路了，或是遇到了一個無限迴圈。在 IP 協定中，最後一個 hop 會自動放棄這個封包。發生這種情況時，封包遭到丟棄而不能正確傳遞至目的地。但是封包的發送端並不知道，自己被丟失了一個封包，因此丟棄它的 hop 將回傳一個 ICMP 的訊息告訴發送端，這個訊息稱之爲「Time Exceeded」訊息。

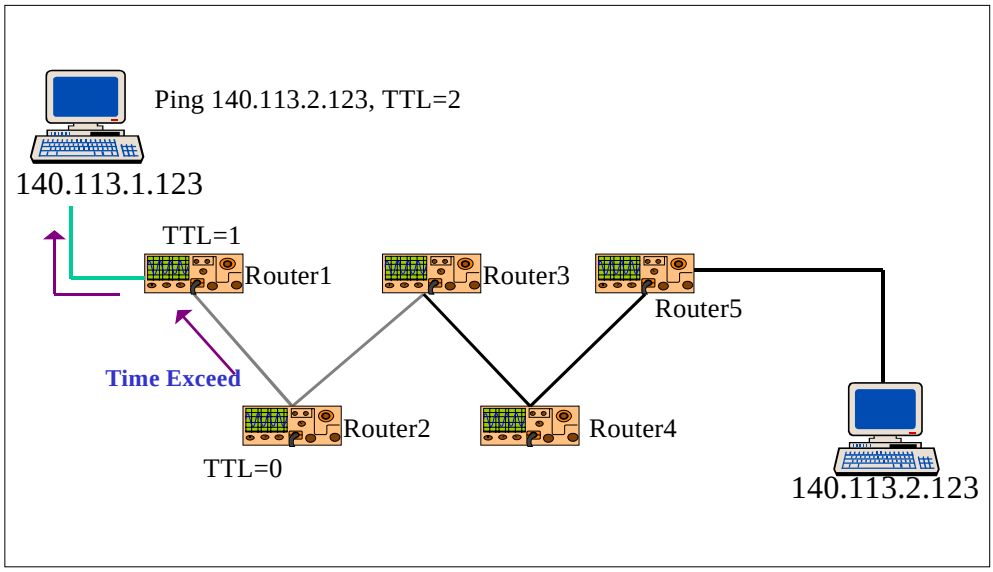
2.2.4 利用 ICMP 測量網路延遲的時間

藉由上面所介紹的兩種 ICMP 功能，我們就可以對 Internet 上任意節點(node)間延遲的時間進行測量。我們送出特定 TTL 值的 Echo Request 訊息，讓中繼的 Router 送回 Time Exceeded 訊息，並紀錄封包往返的時間，算出每一個段網路間封包延遲的狀況。



圖四：ICMP 路徑追縱與延遲測量(1)

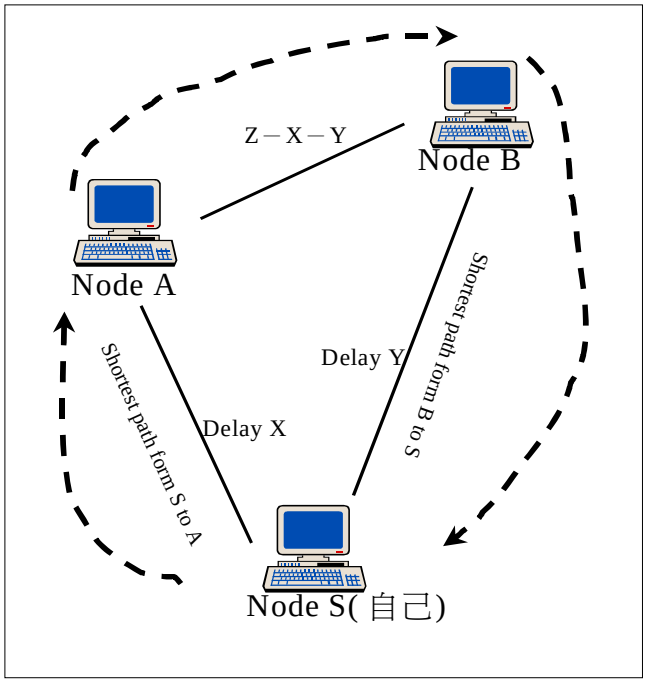
如圖四所示，Router1 至 Router5 是兩點間的正常路徑。我們向目標機器送 ping 封包，設其 TTL=1，Router1 將 TTL 減 1 得到 0，就會送回「Time Exceeded Message」，訊息中包括了 Router1 的 IP Address，於是我們知道封包的第一站就是 Router1。記錄從送出 ping 封包至收到「Time Exceeded Message」的時間，就是往返 Router1 所須要的時間。



圖五：ICMP 路徑追縱與延遲測量(2)

如圖五，我們再送出 ping 封包，但是設 TTL=2，封包到 Router2 時就會被遞減爲 0，我們就會收到 Router2 的「Time Exceeded Message」。將原點與 Router2 間的延遲時間，減去原點與 Router1 間延遲時間，便是 Router1 與 Router2 之間的延遲時間。以此類推，便可以完整地找出原點和目地間的路徑和每一段的延遲時間。

我們利用 LSRR(Loose Source Route)的功能幫助我們指定封包通過的路徑，就可以測量出 Internet 上任意兩點間的延遲時間。LSRR 允許我們指定封包必須先通過那些節點，才到達指定的目的地(destination)，指定通過節點的 IP address 會被依序放在 IP 標頭(IP Header)中的選擇項(option)欄位。傳送封包的路由器會讀取選擇項(option)欄位並依序傳送封包。



圖六：利用 LSRR 測量任意兩點間的延遲時間

如圖六，我們要測量節點 A 至節點 B 的延遲時間，我們先分別測量出節點 S(自己)至節點 A 的延遲時間 X 和節點 S 至節點 B 的延遲時間 Y。再送出一個 ICMP 封包，指定路徑通過 A 再到達 B，則他會通過圖六箭號指示的路線回到節點 S，設延遲時間爲 Z。則 Z－Y－X 即是節點 A 至節點 B 的延遲時間。

三、網域探測與延遲測量工具介紹

上一章我們說明了網域探測與延遲測量基本原理，這樣的程序是一個規律而且重覆多次的步驟，須要透過程式幫我們執行。SNMP 和 ICMP 都是公開的協定，可以藉由 Socket programming 程式開發來完成。這樣的程式開發已經有人完成了，而且分享

在網路上，每一位使用者都可以下載安裝。接下來我們將介紹兩套十分方便的分享軟體(shareware)。

3.1 網域探測工具----Visual MIBrowser

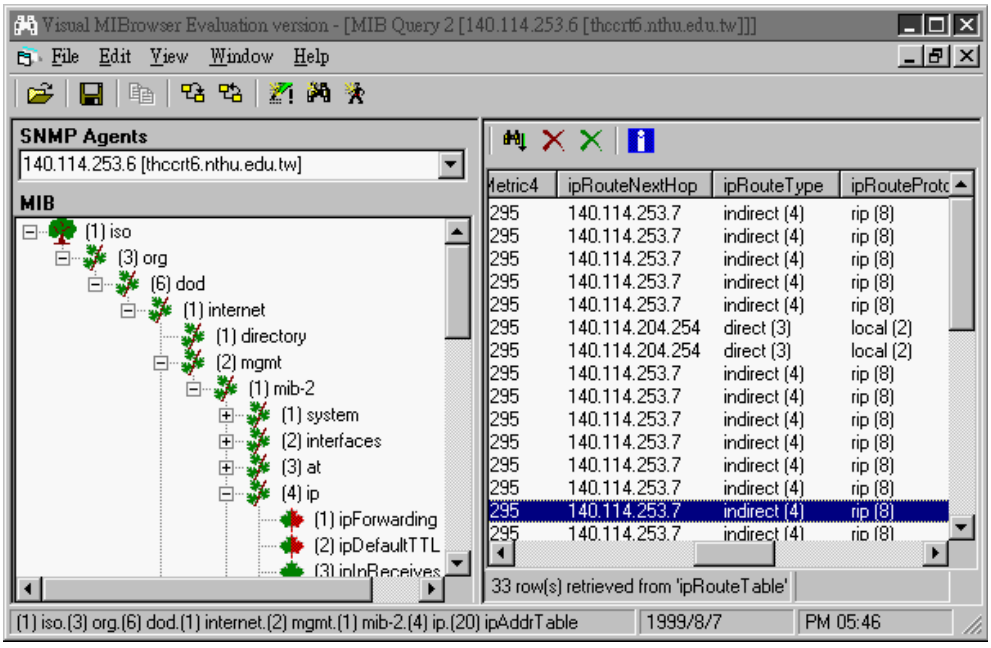
SNMP 是目前網路環境中相當常用的一項網管協定，所以大部份的網路設備都會支援 SNMP。市面上可以瀏覽 MIB 資料庫的軟體很多，常見的有 Visual MIBrowser[11]、OpenView[12]、NetView[13]等工具。部份功能比較完整的軟體須要相當的費用才能取得合法的使用權，如 OpenView、NetView。仍有一些分享軟體(Shareware)歡迎使用者自行下載安裝使用。接下來我們將介紹一套介面方便的分享軟體——Visual MIBrowser。

Visual MIBrowser 由 NuDesign Team 開發完成。是一套在 Windows 環境下使用的工具程式，它提供了方便的介面，使用者無須花費大量的心力在 OID 的索引，只要透過圖形介面的點選，就可以得到想要的 MIB 資料值。在網址 NuDesign Team 的網頁上就可以免費下載安裝。Visual MIBrowser 的特性整理在表五。

Visual MIBrowser	
Develop by	NuDesign Team Inc.
Platform	Windows95/98 WindowsNT4
Download	http://www.nudesignteam.com/
Interface	GUI
Special feature	*Support MIB-II *SNMP Agent Discovery *Store result to text file

表五：Visual MIBrowser 功能簡介

圖七是 Visual MIBrowser 操作的主畫面。畫面可分爲三個部份，左上角的 SNMP Agent 列示框(list box)，左方是 MIB 的樹狀展開圖，右半邊則是結果顯示視窗。我們先在左上角的 SNMP Agent 列示框(list box)中選擇你所要查詢的 SNMP Agent 的 IP 位址，然後展開 MIB 樹狀圖，點選要查詢的查料項，下達「query」指令，就可以在右半邊看到查詢的結果。



圖七：Visual MIBrowser 操作介面

3.2 延遲測量工具介紹----Visual Route

在第 2.2.4 節中我們說明了測量網路延遲的方式，看起來步驟十分複雜，幸而已經有許多工具軟體幫我們處理了這個複雜的程序。網路測量工具眾多，漸漸趨向整合性和圖形化，而且功能常常彼此重疊，我們在此只選擇一項介紹。

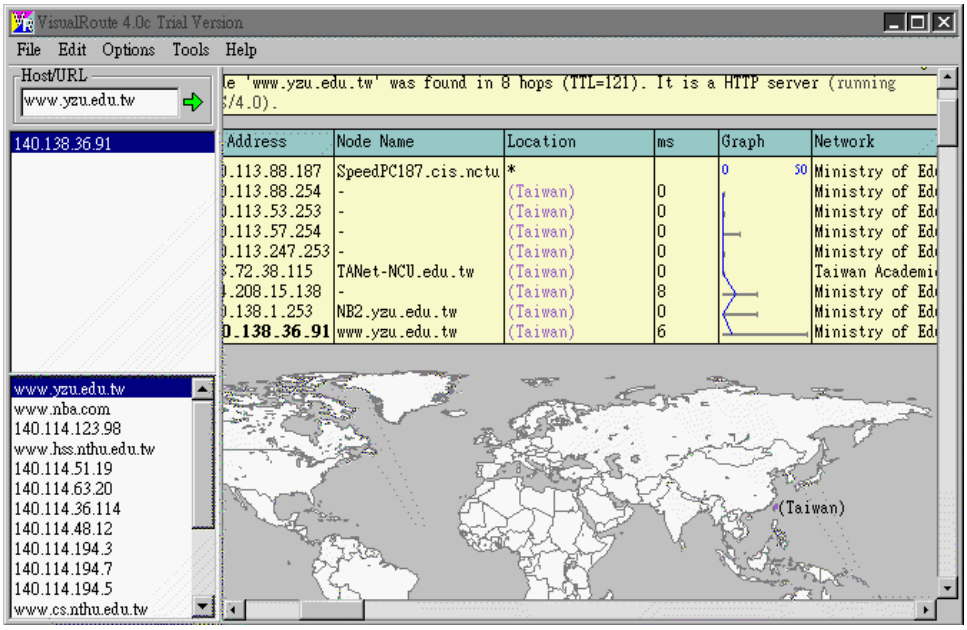
Visual Route 由 Datametrics System Corporation[14]開發， Visual Route 的主要功能是探測路徑的工具。只需指定目的地的位址，Visual Route 就會將封包所經過的每一個路由器表列出來，路徑中的每一個 hop 的延遲情形也會回報在表列之中，還可自動分析網路問題癥結，另外也有增強的圖形功能，所以利用這種測量工具來得知網路塞車狀況是十分方便的。我們將 Visual Route 的特性整理在表六。

Visual Route	
Develop by	Datametrics System Corporation
Platform	Windows95/98 WindowsNT4
Download	http://www.visualroute.com/
Interface	GUI
Special feature	*Map function. *Store trace result to text file. *Loose Source Route

表六：Visual Route 功能簡介

圖八是 Visual Route 的主畫面，大致上可以分爲左右兩個部分。左邊最上方的文字框(text box)是用來輸入欲觀察的目的地，它可以是 URL、host name 或是 IP

address。它的下方有一個 Loose Source Route 欄位，它是用來指定路徑所必須經過的 hosts，雖然它只有一個欄位，但是可以填入多個位址，位址間用空白隔開。若不想使用 LSRR 功能，只要將此欄位保持空白即可。再下來是 Visual Route 將 host name 轉成 IP address 之後的列表，由於可能有數個 IP address 共用一個 host name，所以 Visual Route 會優先觀察排在首位的站址，若要切換到其他的 IP 站址(IP address)，只要用滑鼠在 IP 站址上面 double click 即可。最下面是最近觀察過的站址列表，當然也可以從此處直接選擇站址來觀察路徑。



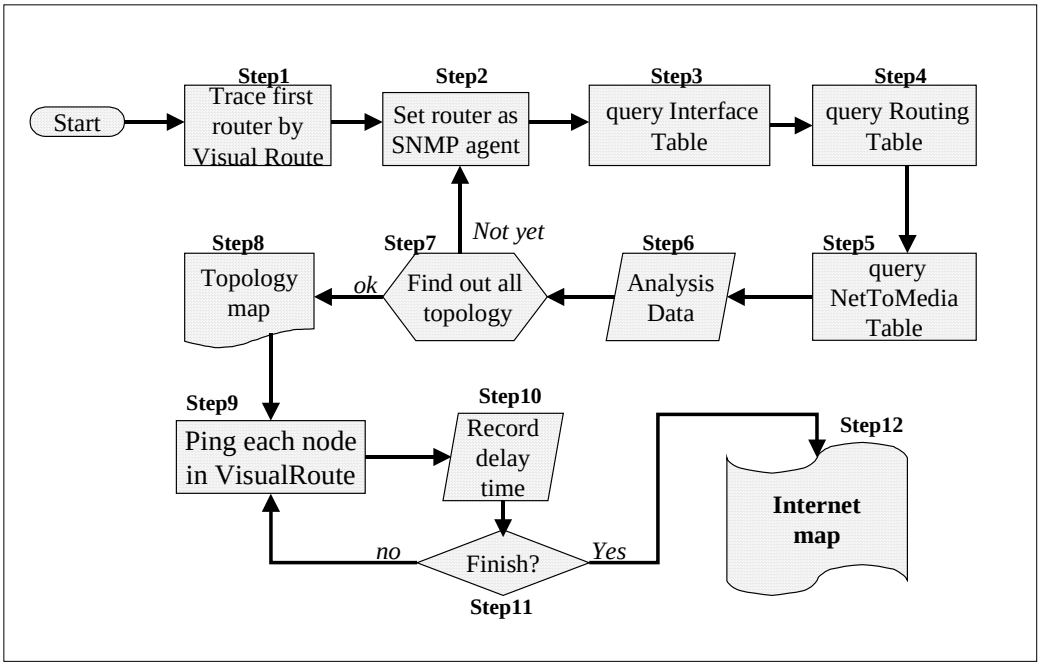
圖八：Visual Route 操作介面

右邊的畫面是路徑分析後的結果，由上而下可分爲三個部份，分別是 Visual Route Analysis、Trace Route Table 以及 Trace Route Map。Visual Route Analysis 是分析封包不能抵達目的地的原因。Trace Route Table 是最主要的資訊來源，它會將路徑上所有的 host 的位址、最大延遲及平均延遲等資訊列出。Trace Route Map 是將路徑用視覺化的方式在世界地圖中展現出來，這樣可更明顯看出實際的連接情形。

Visual Route 還有一些功能在這裡尚未提及，請自行至 Datametrics System Corporation 網站瀏覽查詢，將可得到更詳實的資料。其實除了 Visual Route 以外，其他尚有許多工具軟體具有相同的功能，例如：PingPlus[15]、Trace route[16]、PingSim[17]等，如果有興趣，可以自行在網路上尋找。

四、演算法

我們在此整理了一套完整的演算法，透過這套程序，我們可以探測 Internet 上任何一個我們想了解的網域(domain)。我們可以快速地掌握遠方一個子網域中網路配置的拓撲和網域中每一段鏈結(link)的延遲情形。對一位 Internet 的使用者而言，網路不再是幽不可見的迷宮，對網管人員而言，可以更簡單迅速地找出網路塞車的地方和塞車的原因。



圖九：流程圖

這一套演算法的流程整理如圖九。首先我們要能夠找到一顆屬於該網域內的路由器。從這顆路由器開始，找出和所有它相串連的路由器，再逐一拜訪每一顆路由器，找到更下層的路由器，重覆這個步驟，直到我們得到了整個網域的拓撲(topology)。再來我們要測量網域內每一個節點間的延遲時間。我們利用 Visual Route 分別對網路末端節點發送封包，並記錄封包傳輸的延遲時間。重覆此一動作，直到我們完全測量出整個網域延遲的情形。下面我們仔細條列整個演算法。

Step1: Trace first router by Visual Route 首先，我們須要設定我們要探測的網域。然後 ping 該網域下的任何一台機器，例如該網域內的 WEB 站台。我們可利用先前介紹的工具 VisualRoute 來進行，這樣我們很快就可以在指定的網域下找到一台存活(alive)的路由器。這台路由器就是我們的第一個入口。

Step2: Set router as SNMP agent 我們在 Visual MIBrowser 環境下，把 SNMP Agent 指向我們的第一台路由器。Visual MIBrowser 提供了一個 SNMP Agent Discovery 的功能，會自動偵測出遠端的 SNMP Agent。

Step3: Query Interface Table 展開 MIB tree，查詢指定路由器的 Interface Table，在標準 MIB II 的資料結構中，Interface Table 的 OID 號碼是 1.3.6.1.2.1.2.2 (iso.org.dod.internet.mgmt.mib-2.interface.ifTable)。Interface Table 存放路由器傳輸埠(port)相關的資訊。

Step4: Query Routing Table 展開 MIB tree，查詢指定路由器的 Routing Table，在標準 MIB II 的資料結構中，Routing Table 的 OID 號碼是 1.3.6.1.2.1.4.21 (iso.org.dod.internet.mgmt.mib-2.ip.ipRouteTable)。Routing Table 存放這台路由器在進行 IP 封包轉送時使用的資訊。

Step5: Query NetToMedia Table 展開 MIB tree，查詢這台路由器的 Net To Media Table，在標準 MIB II 的資料結構中，Net To Media Table 的 OID 號碼是 1.3.6.1.2.1.4.22 (iso.org.dod.internet.mgmt.mib-2.ip.ipNetToMediaTable)。Net To Media Table 可以告訴我們和路由器直接串連機器的 IP address 及 MAC address。

Step6: Analysis Data 分析 Step3 至 Step5 所取得的 MIB 資料值，整理這些資料。Visual MIBrowser 可以讓使用者把查詢的結果儲至文字檔(text file)中。這些資料提供了五項資訊。(1)該路由器目前有多少埠對外串接，(2)路由器直接管理那些子網域，(3)每一個子網域下串連多少台機器，(4)有那些其他的路由器和自己串接，(5)每一個 IP 應送往那一個路由器。

Step7: Find out all topology? 是不是想探測網域下的每一個路由器都已經查詢過了？如果還沒跳往 Step2。查詢完成，繼續 Step8。

Step8: Topology map 將探測到的資訊加以整理，就得到網域的拓撲。

Step9: Ping each node by Visual Router 接下來，我們利用 Visual Route 對網域下的每一個路由器進行 ping 的動作。

Step10: Record delay time 封包所經過的路徑及每一段的延遲時間 Visual Route 會回應在螢幕的右側。另外我們亦可利用 LSRR 欄位指定封包經過的路徑，加強測量的便利 (使用 LSRR 的技巧已在前面章節說明)。

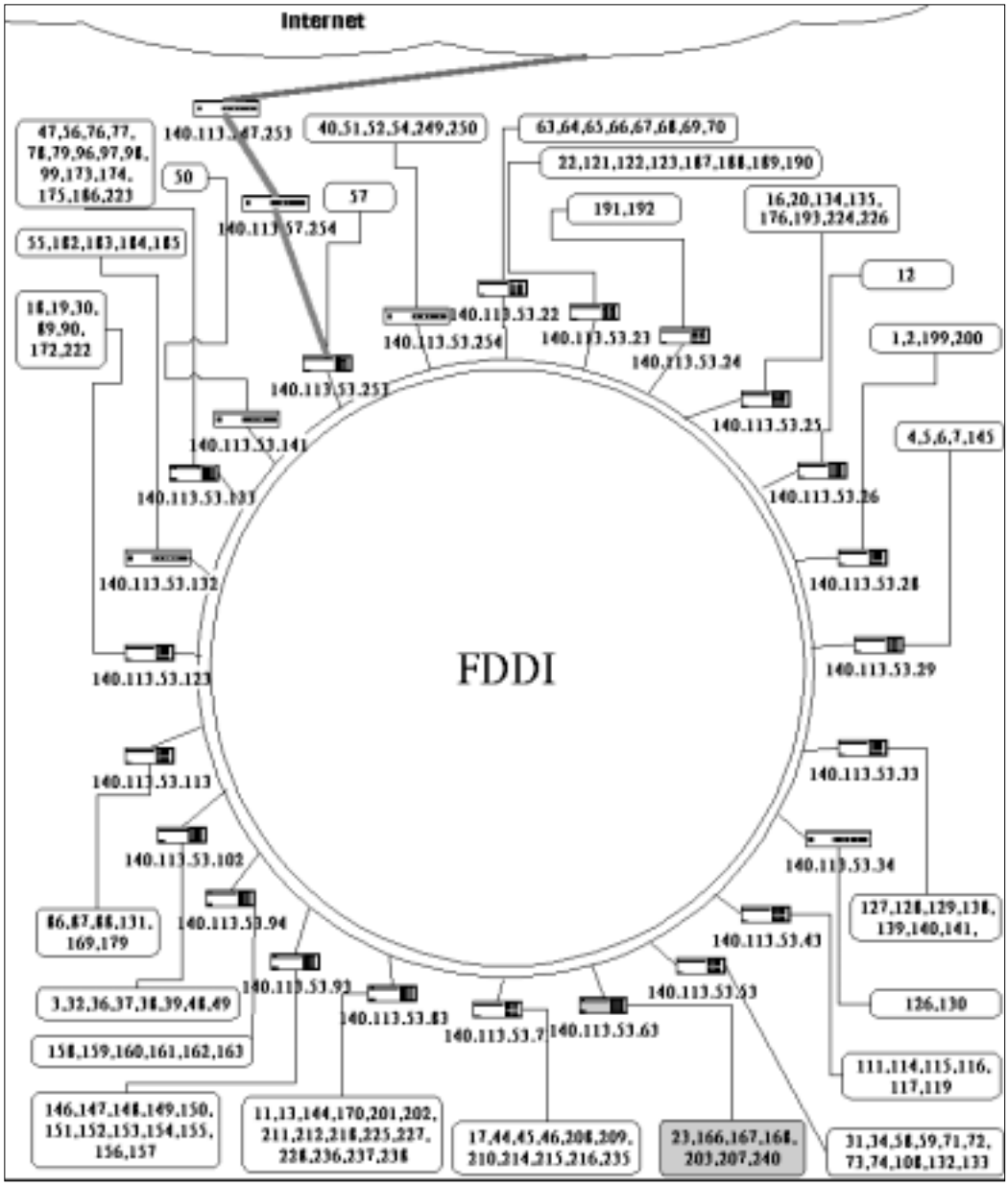
Step11: Finish? 重覆 Step9 和 Step10，直對對每一段網路延遲時間的測量都完成。

Step12: Internet map 一張漂亮的網路配置圖就在你眼前。

五、範例——交大校園網路探測與測量

現在我們以交通大學的校園網路爲例子，依照第四章介紹的演算法，使用第三章介紹的工具 Visual MIBrowser 和 Visual Route，對交通大學的校園網路進行拓撲探測(topology probing)和延遲測量(delay measurement)。

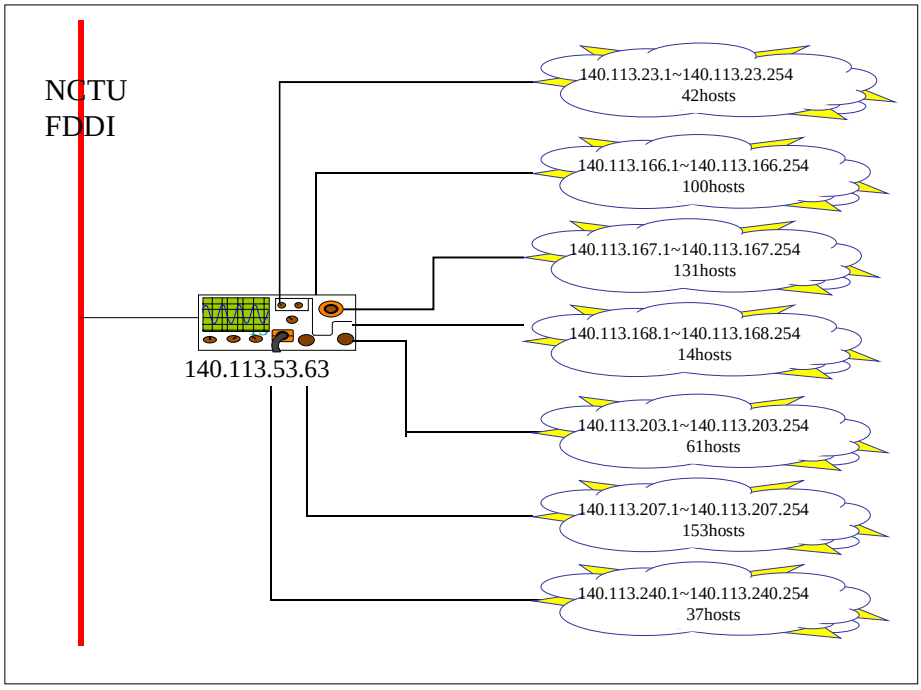
第一步，我們在 Visual Route 中 ping www.nctu.edu.tw，利用 Visual Route 的 trace 功能，找出與 www.nctu.edu.tw 最接近的路由器 IP 位址 140.113.53.254。140.113.53.254 就是我們的第一個入口。我們將從它開始找出整個交大校園網路的拓撲。



圖十：交通大學校園網路架構

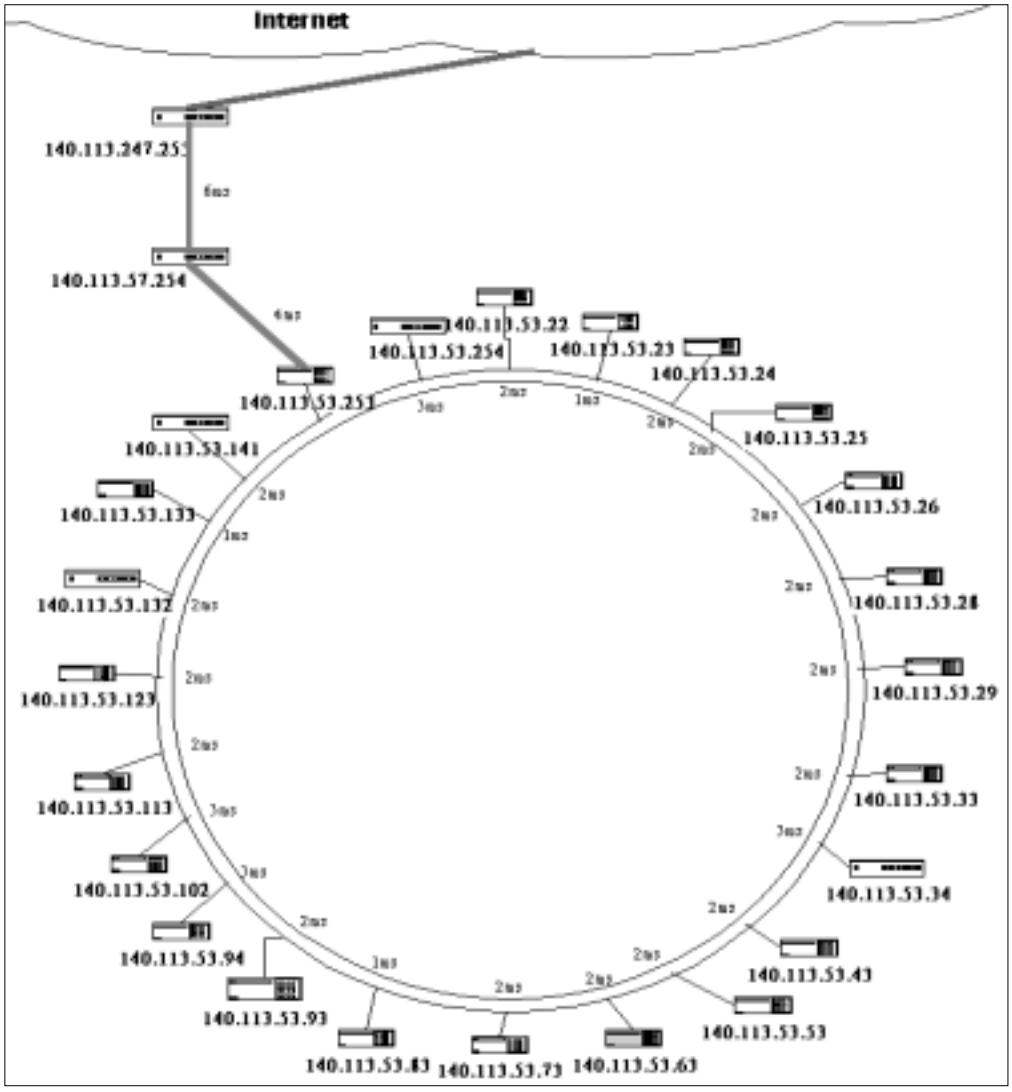
我們開啓 Visual MIBrowser，並在 SNMP Agent 視窗中加入 140.113.53.254。然後在 Visual MIBrowser 主畫面的 SNMP Agent 列示窗中選取 140.113.53.254。並查詢 (query) 140.113.53.254 的 Interface Table、Routing Table 及 Net To Media Table。然後再逐一查詢和 140.113.53.254 連接的每一顆 router，如同第四章演算法的 Step2 至 Step7，我們便可得到交大校園網路完整的拓撲。

如圖十所示，交通大學採用雙迴路 FDDI 架構做爲整個校園網路的主幹。一共有 24 台 router 與之相連，每一台 router 各自管理數個子網域。其中，140.113.53.253 連向交通大學計算機中心，透過計算機中心的 140.113.57.254 和 140.113.247.253 對外通訊。圖十每一台 router 指向的文字框，表示該 router 直接管理的 IP Subnet。例如：140.113.53.63 就直接管理 140.113.23.0、140.113.166.0、140.113.167.0、140.113.168.0、140.113.203.0、140.113.207.0、140.113.240.0 七個子網域。



圖十一：交通大學資訊科學系網路架構

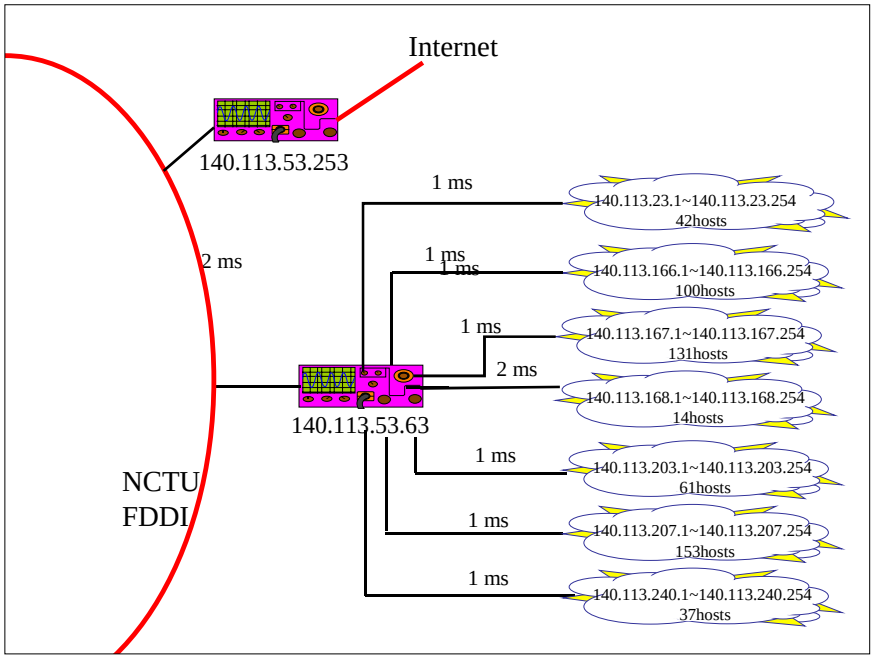
由於交通大學整個校園網路約有一萬多個節點，無法在此鉅細彌遺的表列，特別選擇資訊科學系的子網域爲例子，說明每個子網域下網路的架構。資訊科學系的子網域在路由器 140.113.53.63 之下，共分 7 個子網域，其串接情形及每個子網域上主機 (host)的數量見圖十一。140.113.53.63 共有 8 個通訊埠(port)在運作，其中 15 號通訊埠 (port 15)連向 FDDI 與交通大學其他路由器串接，另外 7 個通訊埠分別管一個子網域。



圖十二：交大校園網路延遲情形

完成拓撲探測，下一步就是進行延遲測量。我們利用 Visual Route 的操作介面可以簡單快速地測量每一個節點與節點間的延遲時間。圖十二是交通大學校園網路 FDDI 主幹上每一台路由器對於計算機中心路由器(140.113.52.253)的封包延遲時間，以及於計算機中心對外的封包延遲時間。由此圖就可以看出每一個子網域對外通訊是否順暢和資料塞車的地方。

圖十三是交通大學資訊科學系每一個子網域間的延遲時間，以及交通大學資訊科學系對外通訊的延遲時間。本章所列示的資訊及數據皆爲於 1999 年 8 月 25 日進行測量之結果。



圖十三：交大資訊科學系網路的延遲時間

六、參考資料

[1] http://www.isc.org/dsview.cgi?domainsurvey/index.html

[2] J.D. Case, M. Fedor, M.L. Schoffstall, J. Davin, ”Simple Network Management Protocol”, RFC 1067, Aug-01-1988.

[3] K. McCloghrie, M.T. Rose, “Management Information Base for network management of TCP/IP-based internets”, RFC1066, Aug-01-1988

[4] J.D. Case, M. Fedor, M.L. Schoffstall, C. Davin, “A Simple Network Management Protocol (SNMP)”, RFC 1157, May-01-1990

[5] J. Case, K. McCloghrie, M. Rose, S. Waldbusser, “Introduction to version 2 of the Internet-standard Network Management Framework”, RFC 1141, April 1993

[6] J. Case, K. McCloghrie, M. Rose, S. Waldbusser, “Coexistence between version 1 and version 2 of the Internet-standard Network Management Framework”, RFC 1452, April 1993

[7] Andrews S. Tanenbaum , “Computer Networks” , 3rd edition, Prentice Hall, 1996

[8] M.T. Rose, “Management Information Base for network management of TCP/IP-based internets: MIB-II”, RFC 1158, May-01-1990

[9] J. Postel, “Internet Control Message Protocol”, RFC 777, Apr-01-1981

[10] Stevens, W. Richard, “TCP/IP Illustrated: the protocol”, Addison-Wesley Publishing Company, 1994.

[11] http://www.nudesign.com/

[12] http://www.openview.hp.com/

[13] http://www.tivoli.com/products/index/netview_distmgr/

[14] http://www.visualroute.com/

[15] http://www.barefootinc.com/main.htm

[16] http://www.traceroute.org/

[17] http://www.xs4all.nl/~houtriet/