

網路拓樸探索界限機制⁺

陳彥錚

國立暨南大學資訊管理學系
ycchen@ncnu.edu.tw

黃世育 溫君偉

銘傳大學資訊管理研究所
syhuang@mcu.edu.tw, wenjimmy@ms24.hinet.net

摘要

本研究在於探討網路拓樸探索界限問題，並提出可能之界限規則來設計網路拓樸探索演算法，以進行網路拓樸探索界限機制之研究。近年來，已有一些學者提出有關網路拓樸探索方式，然而我們發現這些現有的網路拓樸探索方式缺乏有效的界限機制來規範正確的探索範圍，導致網管人員所得到的網路拓樸是一個部份殘缺或多餘累贅的情況，也因此增加了網管工作許多不必要的麻煩，因此，本研究依據常見網路架構之特性提出三個有效的界限機制，用以改進目前網路拓樸探索方式，以增進探索程式的執行效率與探索範圍的正確性。

關鍵詞：網路拓樸探索、界限機制、網路管理

⁺本篇論文之研究由國科會計劃補助，計劃編號：NSC 90-2213-E-260-026-。

Abstract

In the recent years, several network topology discovery algorithms have been proposed. However, we found that these network topology discovery algorithms did not consider how to precisely bound the discovery only within the networks interesting for the network administrator. As a consequence, the discovered topology information is usually only parts of networks to be managed. The discovered topology information may contain a lot of redundant and useless networks and nodes beyond the managed network. Considering the common topology features of most existing networks, we will propose three boundary rules as follows: (1). Domain boundary (2). LAN/WAN boundary (3). IP address exclusive boundary. Based upon the above boundary rules, we will propose some more efficient network topology discovery algorithms. Our algorithms have the advantage to precisely bound the discovery area.

Keywords: topology discovery, boundary, network management

1. 前言

由於網路的快速成長，網路的互連日趨複雜，使得網路變的愈來愈龐大，而網管人員有責任去維護網路設備的正常運作，以維持網路的服務品質。但面對如此龐大且複雜的網路環境，要能有效管理網路是一件非常困難的工作，因此如何去得到一個適當的網路拓樸資訊，是一個非常重要的問題。

傳統的網路拓樸探索方法利用 ICMP (Internet Control Message Protocol) [1]通訊協定，對某一網

域的節點利用 Ping 的方式來詢問其是否存在，並不斷對網域的每個節點重複執行 Ping 這個動作，將正常運作的節點記錄下來以描繪出網路的拓樸架構。此種方式雖可把網路探索出來，但所花費的時間成本卻非常多。因此一般拓樸探索程式均會利用 SNMP (Simple Network Management Protocol) [2]到已知的網路設備去讀取與網路拓樸相關的資訊，主要包括 SNMP 網管資訊庫 (Management Information Base, MIB) [3]中的路由表 (Routing Table)，藉由路由表的路由資訊，可以一步一步去了解子網路的存在與其連結情形。

利用 SNMP 進行網路拓樸探索並非創新的觀念，然而現有的網路拓樸探索方式卻沒有深入探討如何提供一個有效的界限機制來規範正確的探索範圍，導致網管人員所得到的網路拓樸常是一個非網管人員所需要的網路拓樸架構。本研究認為目前網路拓樸探索方式必須要有一個適當的界限機制 (Boundary Heuristic)，並透過這些界限機制讓拓樸探索程式能更精確且有效率的找到所需要的網路拓樸架構，我們將以 TCP/IP 網路環境為研究對象，提出一些網路拓樸探索的界限規則，網管人員可利用這些界限規則改進網路拓樸探索的正確性，進而減輕因網路拓樸探索所造成額外的網管工作負擔。

本研究針對界限機制的觀念提出多個界限規則，並設計一主要界限機制拓樸探索演算法，將所提之界限規則附加於此演算法中，其分別為節點領域名稱比對、區域與廣域網路分界和 IP 位址排除法。節點領域名稱比對是利用 DNS Server 所提供之服務機制去詢問節點其領域名稱，藉由其領域名稱來判斷節點所在網域是否為探索範圍。區域與廣域網路分界則是在進行拓樸探索時，除了可發現有那些子網域存在之外，並可探索其網路的連結方式，藉由此區域與廣域網路判斷來決定探索範圍。IP 位址排除法則是一般網路的連結方式是由多個子網路匯集於一點再與其它類似的網路互連，其網路架構類似一個樹狀的階層式架構，因此我們只要指定一個或少數幾個不屬於探索範圍之 IP 位址作排除之動作，當探索程式發現包含這些 IP 位址之網路即停止探索工作。

2. 網路拓樸探測原理

探測一個網路拓樸最常採用的技術是利用 SNMP (Simple Network Management Protocol) 的方式去取得 MIB 管理資訊庫的資料，因為 MIB 裡存放了各個網路設備本身的路由資訊 (即 Routing Table)，Routing Table 記錄了 IP 位址的目的地網域

(Destination) 及下一個轉送的路由器 IP 位址，藉由這些資訊我們可以概略的描繪出網路的拓模。

2.1 簡單網路管理協定

由於網路上的設備種類繁多，不同廠商的網管系統有不同的協定，而 SNMP 網路管理協定則是一個開放的網路管理協定，可以讓網路設備容易的去交換網路管理資訊。SNMP 是一個遵循網路標準規格的網路管理協定，SNMP 第一版在 1990 年 5 月由 RFC 1157 所定義完成，在 RFC 1441 到 RFC 1452 中，則是提供較 SNMPv1 更多安全性方面的擴充，發展成 SNMPv2 [4]，SNMPv3 也於近年被提出。目前，各網路設備仍以支援 SNMPv1 為主。

SNMP 管理模式定義了兩種管理實體 (Management entities)，一類是網路管理工作站 (Network Management Station, NMS)，另一類是代理工作站 (Agent)，所謂網路管理工作站是一部具有資訊處理能力的電腦，在上面執行網路管理軟體 (Manager)，用以監控所管轄的網路元件 (Network Element)。所謂網路元件是指網路上的硬體設備，例如：路由器、橋接器、伺服器或終端機等設備，並在上面執行中介軟體 (Agent) 記錄網路元件的運作情形，並負責接受網路管理工作站所下達的指令，調整其運作方式。其運作情形如圖 1 所示。

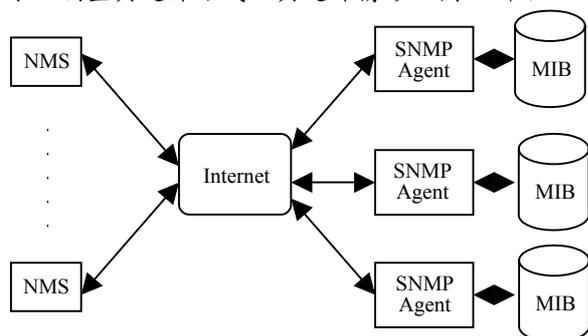


圖 1. SNMP 網路管理模式

2.2 管理資訊庫

由於不同的網路設備因廠商不同有其不同的特性，而網路管理工作站需要一個共通的標準介面來讀取網路設備的網管資訊，而管理資訊庫則是將各個網路設備相關的網管資訊儲存起來，與網際網路協定相關的 MIB 樹主要是定義在 MIB-II (RFC1213 [3])，其資料架構是一個樹狀結構。

管理資訊庫類似一個資料庫，負責儲存所有網管相關的訊息，管理資訊庫之網管資訊均以物件的方式呈現，由於相關的網管物件數量龐大，為能唯一辨別每一網管物件，SNMP 採用與目錄服務相同的物件識別機制，即每個網管物件均定義了一個物件識別碼 (Object Identifier, OID)，一個物件之 OID 記錄了該物件在一個樹狀圖中的位置。如圖 2 所示，每一項資料對應至樹的一片樹葉，相關的資料形成一棵子樹，由上而下，依照相關的組織、特性形成整棵樹。樹上的每一個節點都會被賦予一個特

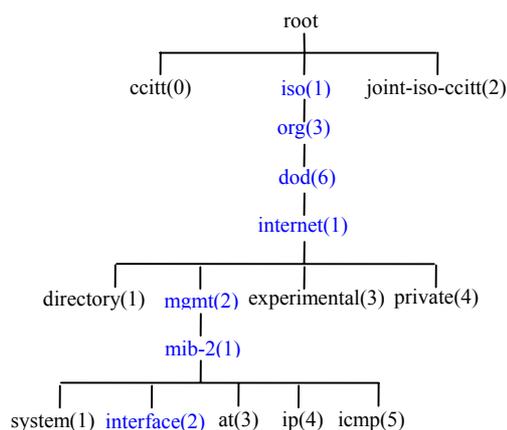


圖 2. MIB 的樹狀結構

定的整數號碼，當我們找尋某一個節點時，則由樹根開始尋找，例如我們要尋找 interface 這個節點，則需經由 iso(1)到 org(3)到 dod(6)到 internet(1)到 mgmt(2)到 mib-2(1)到 interface(2)，因此 interface 物件的 OID 便為 1.3.6.1.2.1.2。

2.3 MIB-II 的相關拓模應用

MIB-II[3]是一個標準的 MIB 資料庫，因為一般來說只要是支援 SNMP 的網路設備大都有支援 MIB-II，而 MIB-II 主要是定義在 1.3.6.1.2.1，共包括十個群組 (Group)，其資料如表 1 所示：

表 1. MIB-II 十個群組

群組	物件 ID	說明
System	MIB-2 1	系統描述
Interface	MIB-2 2	介面卡資訊
At	MIB-2 3	網路位址的轉換表
Ip	MIB-2 4	IP 路由的資料
Icmp	MIB-2 5	ICMP I/O 統計
Tcp	MIB-2 6	TCP 連結參數與狀態
Udp	MIB-2 7	UDP 傳輸統計
Egp	MIB-2 8	EGP 傳輸狀態
Transmission	MIB-2 10	傳輸媒體資料
Snmp	MIB-2 11	SNMP 訊息統計

對於拓模探測而言，我們最主要使用到 MIB-II 裡所包含的 interface 群組和 ip 群組。interface Group 包含 interface Table，interface Table 可以告訴我們這台 Router 每一個 Port 的使用情形。在 ip Group，我們可以取得 Routing Table、ip Address Table 和 ipNetToMedia Table。Routing Table 記錄了不同的 IP 位址要從哪個節點轉送出去，ip Address Table 則記錄了節點介面的 IP 位址資訊，而 ipNetToMedia Table 則告訴我們 Router 目前鄰近節點的 IP 位址與實體位址資訊。底下則將相關拓模探測所使用到之 Table 欄位資訊作一說明，如表 2、表 3、表 4 所示。

傳統探測一個網域最主要的方法是去取得 Routing Table，因為 Routing Table 記錄了不同 IP 位

址的目的地網域及其傳送的方向。因此，當我們得

表 2. Interface Table 欄位說明

欄位名稱	資料型態	說明
ifIndex	INTEGER	介面卡編號
ifDescr	DisplayString	網路介面的描述
ifType	INTEGER	網路介面型態
ifPhysAddress	PhysAddress	介面卡的實體位址
ifOperStatus	INTEGER	網路介面運作狀態

表 3. Routing Table 欄位說明

欄位名稱	資料型態	說明
ipRouteDest	ipAddress	路由位址的目的地
ipRouteIfIndex	INTEGER	介面卡編號
ipRouteNextHop	ipAddress	Next Hop 的 IP 位址
ipRouteType	INTEGER	路由的型態
IpRouteProto	INTEGER	使用的通訊協定
IpRouteMask	ipAddress	路由位址的 Mask

表 4. ipNetToMedia Table 欄位說明

欄位名稱	資料型態	說明
ipNetToMediaIfIndex	INTEGER	介面卡編號
ipNetToMediaPhysAddress	PhysAddress	直接相連機器的 MAC 位址
ipNetToMediaNetAddress	IpAddress	直接相連機器的 IP 位址
ipNetToMediaType	INTEGER	ipNetToMedia 的型態

知一個路由器的 IP 位址時，我們可以透過路由的型態 (ipRouteType) 來得知與此路由器相連的子網域 (subnet)，當 ipRouteType 值為 direct，表示目的地網路 (ipRouteDest) 與此路由器相連，另外當 ipRouteType 的物件值 indirect 時，而此時 ipRouteNextHop 所記錄的節點必是與此路由器相連的其它路由器 IP 位址，因此我們可以藉此來得知路由器彼此相連的情形，並可查詢此路由器的 ipNetToMedia Table，因為它記錄了與此路由器直接相連主機的 IP 位址，透過此種方式重複對路由器去讀取 Routing Table 跟 ipNetToMedia Table 的資訊，即可得到整個網路的拓樸架構。

2.4 相關網路拓樸探索技術

我們首先針對目前所使用之拓樸探索技術做一分析與整理[6-12]，了解這些拓樸探索方式之原理及其優缺點。目前探索網路拓樸的方式最主要使用到四種技術：

1. SNMP 網管通訊協定：一般網路設備如路由器、橋接器或交換器均支援 SNMP 網管通訊協定，這些設備會將網路管理資訊以符合 SNMP 通訊標準的方式儲存於 MIB 資訊庫中，讓網路管理工作站透過 SNMP 的方式至遠端存取 MIB 資訊。此種方式是探測一個網路拓樸最常使用之技術，但 SNMP 具簡單的身份驗證機制，於管理站執行之探索程式必須知道網路設備之社群名稱

(Community Name) 才能順利讀取 MIB 資訊。

2. ICMP 通訊協定：探測某一節點是否存在我們均會使用 Ping 的程式來判斷，此種利用 ICMP 通訊協定所設計之 Ping 程式可以用來協助探索某一網路有哪些節點存在。此種方式適用於缺乏 SNMP MIB 資訊之網路環境，因一般探索程式會至 MIB 讀取 ipNetToMedia Table 來判斷目前所存在之節點，但此資訊也僅止於最近有通訊行為之節點，為了能正確找出網路拓樸，仍會搭配 SNMP 與 ICMP 通訊協定來尋找網路。雖然利用 Ping 的方式可正確找出網路節點，但 Ping 程式對一個不存在的節點其必須等待 Time Out 的時間卻非常多，因此對一個只有少數節點存在的網路進行 Ping 的方式會影響執行效率。

3. Traceroute 技術：Traceroute[5]利用 IP 通訊協定為避免無窮迴圈所定義的 TTL (Time To Live) 欄位所設計，主要是探索發送端到目的端的連結路徑，以藉此了解網路拓樸資訊。當已知所有 IP 位址之範圍，便可從網管系統所在節點對其它節點進行 Traceroute 探測，最後將所得結果進行整理便可得到整個網路拓樸資訊。但利用此方式對作兩點進行一次 Traceroute 之代價很高，執行效率是一大問題。

4. DNS 技術：DNS Server 包含了其網域上所有的名稱和相對應的 IP 位址，除此之外，我們也可向 DNS 詢問某一領域的所有主機名稱，藉此服務可得知所探索的領域有哪節點有註冊領域名稱及其 IP 位址，來概略得知網路探索範圍。但此方式對未向 DNS Server 註冊之節點可能無法得知其網路的存在，且許多 DNS Server 均會因為安全的考量關閉詢問所有領域名稱之功能，因此，目前較少利用此方式去詢找網路拓樸。

2.5 傳統探索網路拓樸之界限機制

了解目前所使用之拓樸探索技術及其基本原理，可將重點針對本研究所提之界限機制上，雖然一般拓樸探索方式並沒有具體指出其所採用之界限機制，但我們從一些學者之演算法分析可整理成以下幾種目前所採用之界限方法：

1. IP 位址列舉法：即列出所有待探索網路之 IP 位址，此種方式為最精確的方法，但當所欲探索之 IP 位址不可得或不足時，採用此種方式所找出之網路拓樸可能並不完整。但此方法可用來衡量其它界限機制之精確性，越接近此方法所探索出來之網路拓樸其精確性越高。

2. Hop 數限制法：此方法預先設定所探索之 Hop 數，並由網管系統所在網路中心位置向外進行拓樸探索，當探索範圍所經過之 Hop 數超過預先設定值便停止探索工作。但此方式只有在網管系統位在網路正中心時才可恰巧找到我們所要的網路。

3. 節點數限制法：預先設定所要探索之節點數目，並由網管系統所在網路為中心向外進行拓樸探索，當所找到的節點數目超過系統設定值便停止進行探索工作。但此方法並不考慮網路探索範圍之正確性。

3. 網路拓樸探索演算法設計

以下我們將針對所提之界限規則分別設計拓樸探索演算法，首先我們先提出一個與界限規則獨立的具界限機制拓樸探索演算法，然後再針對我們所提之界限規則，分別敘述其演算法，這些界限規則包括節點領域名稱比對、區域與廣域網路分界和 IP 位址排除法等機制。

3.1 具界限機制拓樸探索演算法

具界限機制拓樸探索演算法主要有一界限規則 (Boundary Heuristic) 來規範所探索的網路。具界限機制拓樸探索演算法顯示於表 5。此演算法針對每一新發現的子網域進行界限判斷，如果所探索之子網域符合網路探索範圍，探索程式則會將此子網域納入 SubnetList (此列表會紀錄符合條件之子網域)，並取得此網域內之節點，若所探索之路由器為 SubnetList 網域內之節點 IP 位址，則會將此路由器加入 RouterList 作為下一個探索之 Router。

表 5 之界限機制拓樸探索演算法使用九個函式，其功能說明如下：

1. default_Router(): 讀取本系統之 default Router。
2. add_element(): 將 Router 存放至 RouterList。
3. get_element(): 取出 Router 並移除此 Router 於 RouterList。
4. get_direct_connected_subnet(): 將直接相連之子網域存放至 candidateSubnetList。
5. isInScope(): 判斷子網域是否為探索範圍。
6. addSubnet(): 將符合之子網域存放至 SubnetList。
7. find_all_devices(): 尋找子網域目前所存在之節點。
8. find_next_hop_router(): 將下一個轉送節點之 Router 存放至 RouterList。
9. isInSubnet(): 將符合探索範圍之 Router 存放至 RouterList。

表 5 所示之界限機制拓樸探索演算法是由執行拓樸探索所在節點之預設路由器 (default Router) 逐步向外擴張尋找網路拓樸，它會先取得此 Router 之相連子網域，並將其存放至 candidateSubnetList (此列表記錄所有未判斷之子網域)，再藉由本演算法所設計之界限規則對所有子網域作一判斷，若所得之子網域屬於探索範圍則會存放至 SubnetList，並對此一子網域去取得其內部節點，接著將相符合於網域內之路由器存放至 RouterList (此列表記錄尚未探索之路由器)，最後將 RouterList 所存放之路由器逐一重複上述步驟去取得其相關拓樸資訊，即可

產生所要探索之網路拓樸架構。

表 5. 具界限機制拓樸探索演算法

```

BEGIN
dr = default_Router ();
add_element ( dr, RouterList );
While ( ! empty ( RouterList ) ) do {
  r = get_element (RouterList);
  candidateSubnetList =
    get_direct_connected_subnet ( r );
  foreach s in candidateSubnetList {
    if ( isInScope(s, BoundaryHeuristic));
    addSubnet(s, SubnetList);
    find_all_devices ( s );
  }
  R = find_next_hop_router ( r );
  foreach nhr in R do {
    if (isInSubnet(nhr,SubnetList));
    add_element ( nhr, RouterList );
  }
}
END

```

函式 isInScope() 中之 BoundaryHeuristics 為界限規則，藉由表 5 所示之演算法可了解一個具有界限機制的網路拓樸探索是如何執行，以下為方便闡述我們所提之界限規則，我們將只描述與各界限規則相關的演算法步驟，即從尚未執行界限判斷的子網路串列中 (candidateSubnetList)，找出符合界限規則之子網路並將之放至另一子網路串列 (SubnetList)，以下針對每一界限機制演算法做一介紹與說明。

表 6. 節點領域名稱比對之演算法

```

foreach s in candidateSubnetList {
  D=find_all_devices ( s );
  foreach node in D {
    hostname = getDomainName ( node )
    if (hasNoDomainName(hostname)) next;
    if ( isInDomain(hostname,DomainName)) {
      add_subnet ( s, SubnetList )
    }
    else {
      add_subnet ( s, ExcludeSubnetList )
    }
  }
}
SubnetList=SubnetList ∪
(candidateSubnetList-ExcludeSubnetList);

```

3.2 界限機制 1 - 節點領域名稱比對

節點領域名稱比對主要是由給定之領域名稱 (DomainName) 來規範所探索的網路。此界限機制可讓網管人員只針對欲納管的網域進行拓樸探索。表 6 之演算法預設條件是先給定一個所要搜尋的領域名稱，並將路由器所探索之相連子網域及網域內節點加以分析，針對每個網域下之節點至 DNS Server 詢問其領域名稱，若所得之領域名稱與所要搜尋的領域名稱相符，則會將此節點所在之子網域納入 SubnetList (此列表會記錄符合條件之子網域)；如果所得之領域名稱不屬於所要探索的網

路，即為其它之領域名稱，則將此節點所在之子網域納入 ExcludeSubnetList（此列表會記錄不符合條件之子網域），重複上述步驟之節點領域判斷，最後再將 candidateSubnetList 與 ExcludeSubnetList 所記錄之子網域作比對，並將相符合之子網域移除於 candidateSubnetList，最後再與 SubnetList 聯集即為所要探索之子網域。

3.3 界限機制 2 - 區域與廣域網路分界

一般而言，網管的工作範圍常在區域網路的環境，例如校園網路或企業內網路，因此，網路拓樸探索範圍也多會侷限在區域網路內，當我們在進行網路拓樸探索時，除了可發現哪些子網路外，並可探索這些網路的連結方式，用以決定網路拓樸之界限。針對區域與廣域網路分界界限機制，我們提出兩種探索方式，說明如下。

3.3.1 介面型態辨識

MIB 資料庫包含了 Interface Table，而 Interface Table 記錄了被管理物件之介面特性，而本演算法則利用 Interface Table 之介面型態（Interface Type）來判斷網路連結型態為區域網路或廣域網路。此界限規則在於探索以區域網路方式建立之企業內部網路為主，因此演算法會先給定一已知區域網路型態列表（LanTypeList），它記錄所有要探索之區域網路型態，當所探索到之子網域其對應之介面型態屬於區域網路時，則會將此子網域納入 SubnetList。

表 7 演算法包括函式 getRouterInterfaceIndex、getInterfaceType 和 isLan，getRouterInterfaceIndex 是取得 Routing Table 中某一目的地子網路其所對應之網路卡介面編號，此一介面編號即為介面表（interface Table）中介面卡編號（ifIndex）、getInterfaceType 則是取得子網域其介面編號之對應型態（ifType）、isLan 則是將介面型態與 LanTypeList 作比對，以判斷其型態是否為區域網路。

表 7. 介面型態辨識之演算法

```
foreach s in candidateSubnetList {
  rIfIndex=getRouterInterfaceIndex(s);
  rIfType=getInterfaceType(rIfIndex);
  if (isLan(rIfType, LanTypeList)) {
    add_Subnet(s, SubnetList);
  }
}
```

3.3.2 子網路遮罩辨識

由於 interface Table 中的 ifType 所記錄的種類繁多，因此在判斷其型態為區域或廣域網路並不容易，且有時會發生此一型態並不存在，這些原因都會造成拓樸程式執行上的錯誤；為了進一步改進此一原因，我們發現除了藉由 interface Table 之分析外，也可利用 Routing Table 其子網路的遮罩(Subnet

Mask)來推測區域或廣域網路，一般而言，用來連結區域網路與廣域網路的介接存取網路較小，所以其子網路遮罩較大，藉由其切割子網域的大小來得知其網域之連結方式。

表 8 之演算法主要是由 Subnet Mask 來判斷，當一個 C 級網路其切割之網域超過六個子網路時，其子網路的遮罩必須設為 255.255.255.240，即最後一個 byte 大於 224，則此介面卡之連接型態必為廣域網路；反之如果所切割之網域小於六個子網路時，其子網路遮罩必定設為 255.255.255.224 或其它，亦即最後一個 byte 小於或等於 224 時，而此型態的介面多為區域網路。

表 8 之演算法包括函式 getRouterInterfaceIP 和 getMask，getRouterInterfaceIP 會取得其相連之子網域、getMask 則是取得其子網路之遮罩。

表 8. 子網路遮罩辨識之演算法

```
foreach s in candidateSubnetList {
  rIP=getRouterInterfaceIP(s);
  ipRouteMask=getMask(rIP);
  if (ipRouteMask <= '255.255.255.224')
    add_Subnet(s, SubnetList);
}
```

3.4 界限機制 3 - IP 位址排除法

一般網路的連結方式是由多個小網路匯集於一點再與其它類似的網路連結，其網路架構近似樹狀的階層式架構。當所探索的網路近似樹狀架構時，可使用 IP 排除法排除不屬於探索範圍之 IP 位址，且這些 IP 位址所屬的網路必須直接與所欲探索的網路相連，即為此樹狀網路之父節點網路，當拓樸探索程式發現包含這些 IP 位址之子網域時，便停止對此網域進行探索。

表 9. IP 位址排除法之演算法

```
foreach s in candidateSubnetList {
  nId=getNetId(s);
  nMask=getNetMask(s);
  isObv=false;
  foreach obvIP in ObviateList {
    if (nId == (obvIP && nMask))
      isObv=true;
  }
  if (!IsObv) {
    add_Subnet(s, SubnetList);
    find_all_devies (s);
  }
}
```

表 9 之演算法會存在一資料列表（ObviateList）儲存一個或少數幾個不屬於探索範圍的 IP 位址。並假設 candidateSubnetList 所記錄之相連子網域為所要探索之網路，每取得一個所要排除之節點即會判斷其所在網域為何，當此網域與所要搜尋之子網域

相符合時，即排除此節點所在網域，重複針對排除之節點所在網域做判斷，只要探索之網域不存在所要排除之子網域，即可將此子網域納入 SubnetList，並尋找此網域之節點資訊。

此演算法會使用到 getNetId 和 getNetMask 兩個函式，getNetId 是取得 Routing Table 所記錄之相連子網域、getNetMask 取得其相連子網域之網路遮罩，將排除之節點與其網路遮罩做"AND"運算，判斷排除之子網域是否與相連之網域相同，如果相符即移除此網域於探索範圍。

3.4 界限機制之比較分析

上述所提之界限規則主要在於提供一個更有效率的探索方式去尋找網路拓樸，其分別為節點領域名稱比對、區域與廣域網路分界和 IP 位址排除法，且各界限規則因網路環境的不同均有其適用範圍，我們針對演算法之描述將各界限規則整理成如表 10 所示。

表 10. 界限機制之比較分析

	節點領域名稱比對	區域與廣域網路分界		IP 位址排除法
		辨識	子網路遮罩辨識	
條件	設定一領域名稱	存在一區域網路型態列表	無	排除節點位址
機制	領域名稱	介面型態	子網路遮罩	節點位址
適用範圍	針對某一或多個網域名稱做拓樸探索	1.單一 LAN 所組成的 Intranet 網路 2.校園網路		階層式網路

4. 結論

一般進行網路拓樸探索時，最快的方法即是至 MIB 管理資訊庫中讀取網路拓樸相關資訊，藉由這些資訊的分析來產生網路拓樸架構，但一般拓樸探索程式卻很難進一步去正確規範所探索的範圍，因此所得到的網路拓樸有可能並非是網管人員所需之網路。因此本研究有鑑於此，針對目前網路的相關特性提出一些有效的界限規則，藉由這些界限規則來改進目前網路的拓樸探索方式，以進一步正確找出所要探索的範圍。

本研究提之拓樸探索演算法均是以集中式的方式向外擴展去尋找網路拓樸，但當所探索的網路範圍很大時就會有一些問題產生，例如搜尋時間成本的增加或大量訊息的傳遞造成網路的負載等，因此，就有學者[13]提出分散式的網路拓樸架構，利用分散式的網路技術去搜尋網路拓樸，其方

法是在網路上配置多個網路管理伺服器在不同的網路去執行拓樸探索工作，最後再將各管理伺服器所探索之資訊做一結合形成網路拓樸架構，但如何讓網路管理伺服器能各職所司，適當的找到其管轄範疇的網路即停止探索工作以避免管轄重疊，這有賴運用界限機制來達成，因此，本研究所提之界限規則可結合於分散式網路管理技術，讓網路管理伺服器能採用各界限機制之特性在不同的網域內進行探索，以明確的規範所探索的範圍。

5. 參考文獻

- [1] J. Postel, "Internet Control Message Protocol", RFC777, 1981.
- [2] J.D. Case, M. Fedor, M.L. Schoffstall, C. Davin, "A Simple Network Management Protocol", RFC1157, 1990.
- [3] K. McCloghrie, T. Rose, "Management Information Base for Network Management of TCP/IP-based internets", RFC1213, March 1991.
- [4] 孫文駿、林盈達,「網域拓樸探測與延遲測量」, 國立中山大學, TANET'99.
- [5] V. Jacobson, et. Al. Traceroute, 1988.
- [6] 李成泰, 儲存為基的 TCP/IP 區域網路拓樸探索與管理系統之建立, 私立輔仁大學資訊管理研究所碩士論文, 2000。
- [7] Hwa-Chun Lin, Shou-Chuan Lai, Ping-Wen Chen, Hsin-Liang Lai, "Automatic Topology Discovery of IP Networks", IEICE Transactions on Information and Systems, April 1998.
- [8] Ramesh Govindan, Hongsuda Tangmunarunkit, "Heuristics for Internet Map Discovery", Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 1371-1380, 2000.
- [9] R. Sianwalla, R. Sharma, S. Keshav, "Discovering Internet Topology", IEEE INFOCOM'99.
- [10] R. Govindan, Anoop Reddy, "An Analysis of Internet Inter-Domain Topology and Route Stability," Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution., vol. 2, pp. 850-857, 1997.
- [11] Yuri Breitbart, Minos Garofalakis, Cliff Martin, Rajeev Rastogi, S. Seshadri, Avi Silberschatz, "Topology Discovery in Heterogeneous IP Networks," Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 1, pp. 265-274, 2000.
- [12] Ramesh Govindan, Hongsuda Tangmunarunkit, "Heuristics for Internet Map Discovery," Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 1371-1380, 2000.
- [13] Hwa-Chun Lin, Yi-Fan Wang, Chien-Hsing Wang, Chien-Lin Chen, "Web-based Distributed Topology Discovery of IP Networks," Information Networking, 2001. Proceedings. 15th International Conference on, pp. 857-862, 2001.