TWNIC 93年DNS教育訓練計畫

DNS 技術研討班

TWNIC 技術組編撰

初版:2003.07.05
修訂:2004.04.28

本課程目之目的主要在於讓使用者了解網域名稱及DNS的作用，能夠設定一個基本的DNS服務，主要內容分爲下列五個部分:

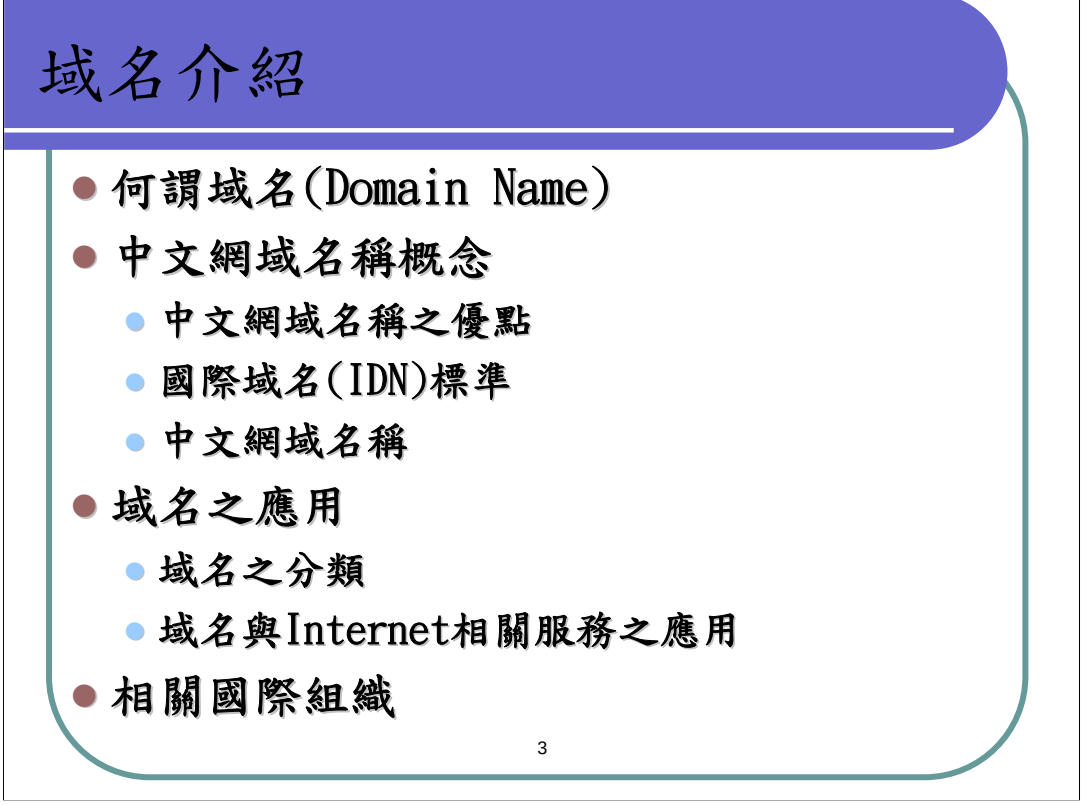
1.域名介紹:網域名稱的概念及其應用方式的介紹

2.DNS基本概念及運作:DNS發展的歷史和整個DNS架構，運作原理以及網域正反解所代表的意義等等介紹

3.域名的申請與DNS指定:以TWNIC所提供的註冊系統爲例來了解Domain Name的申請以及設定方式

4.DNS設定介紹:介紹BIND以及Windows的DNS Server設定， RR的說明，正反解zone的設定

5.DNS各種問題之探討與觀念介紹: 一般常見之設定錯誤及觀念錯誤及其可能造成的影響說明



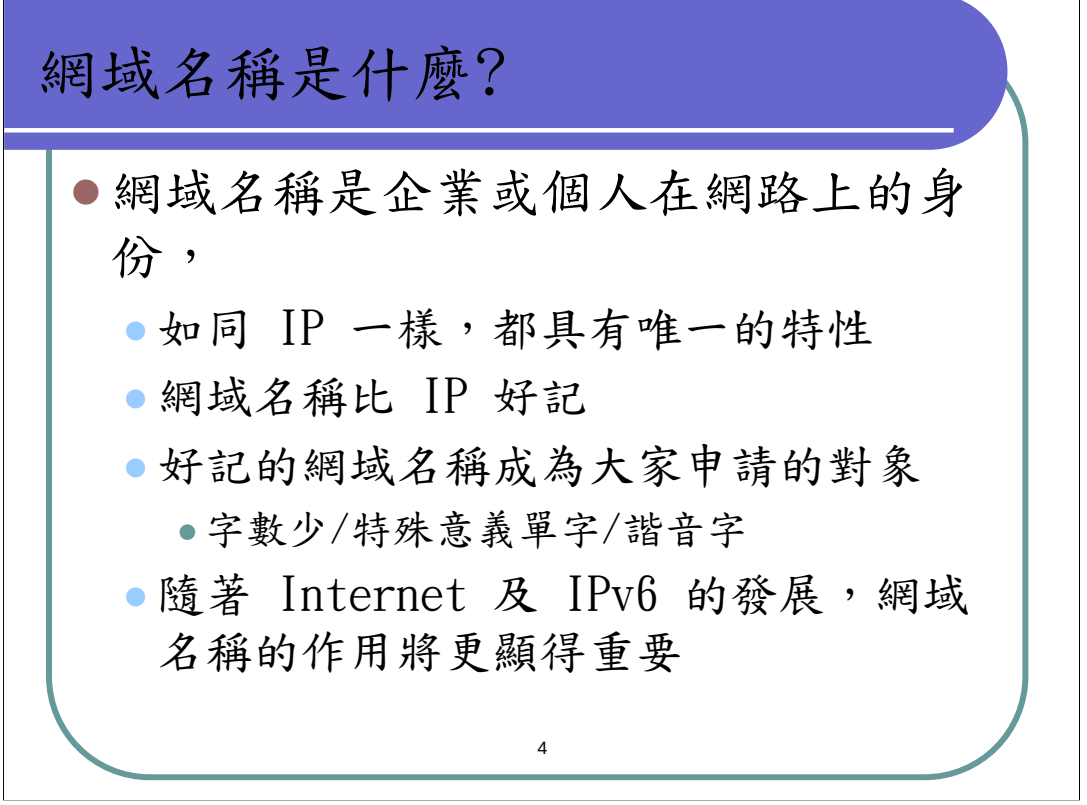
域名介紹主要針對網域名稱的基本概念，作一簡單介紹，主要內容包含下列幾點:

1.Domain Name 與 Domain Name System (DNS)

2.IDN的發展以及相關技術標準及中文網域名稱的發展狀況

3.網域名稱種類的介紹以及網際網路相關服務與網域名稱的相互關係

4.相關的國際網路組織介紹其主要任務及成立原由



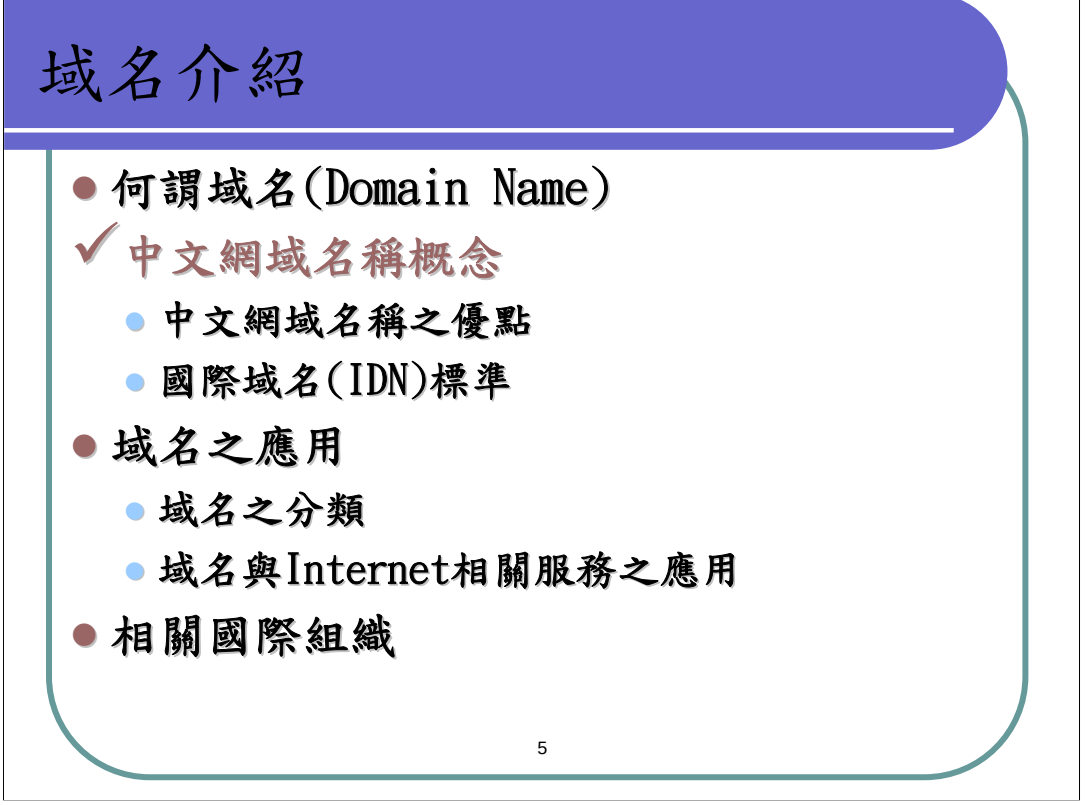
當你連上一個網址如在URL打上：http://www.twnic.net.tw的時候，你已經是使用了DNS的服務了.

但如果您知道這個www.twnic.net.tw的IP地址，直接輸入210.17.9.228也同樣可以到達這個網址.www.twnic.net.tw只是讓人們方便記憶的.一些比較有意義的文字記憶(如：www .twnic.net.tw)，比記憶 IP位址(如：210.17.9.228)，往往容易得多，而且當更換IP時(如更換ISP時)，也不用一一通知所有使用者說我的網頁換IP了，只要在DNS設定改一下，使用者用原來的網址(www .twnic.net.tw)一樣可以連得上.

http://www.ietf.org/rfc/rfc1034.txt (DOMAIN NAMES - CONCEPTS AND FACILITIES)

http://www.ietf.org/rfc/rfc1035.txt (DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION)

在上述 RFC 中定義 domain name 的 label 由 a-z(大小寫)， 0-9 及 “-” 所構成(“-”不得在第一及最後一個字出現)，長度最長爲 63 個 bytes，labels 間以 “.”連結，每個 domain name 最長爲 255 bytes，最多由 127 個 label 組成.



域名介紹主要針對網域名稱的基本概念，作一簡單介紹，主要內容包含下列幾點:

1.Domain Name 與 Domain Name System (DNS)

2.IDN的發展以及相關技術標準及中文網域名稱的發展狀況

3.網域名稱種類的介紹以及網際網路相關服務與網域名稱的相互關係

4.相關的國際網路組織介紹其主要任務及成立原由

中文網域名稱之優點

- 優點
  - 就國人而言中文字較英文字好記

    總統府 => http://www.president.gov.tw/

    中文域名 => http://總統府.tw/
  - 能和企業名稱一致

    統一企業 =>http://www.uni-president.com.tw/

    中文域名 => http://統一企業.tw

以往的網域名稱只能使用RFC1034，1035中規範的英文字元(a-z，0-9 與 -)，現在新的RFC3490-RFC3492 則可以使用國際化域名，國際化域名讓非英語系國家可直接採用其母語作網域名稱，能夠符合區域性的需求，加速Internet的普及

國際域名(IDN)標準(1)

- 經過IETF IDN Working Group耗時3年多的討論，於2003年3月發布通過國際化域名(IDN)技術標準有關之3篇RFCs IDN標準協定IDNA-NAMEPREP-PUNYCODE，這3篇RFC的內容請參考：

1. RFC 3490 IDNA: Internationalizing Domain Names in Applications
2. RFC 3491 Nameprep: A Stringprep Profile for Internationalized Domain Names
3. RFC 3492 Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications

UNICODE 是國際化域名所使用的基本字集，包含世界上所有國家所使用的文字，參考網頁(http://www.unicode.org/)，由於DNS 是以ASCII爲基礎的系統， 而UNICODE 的系統中， 字元已超過現有網域名稱使用的範圍

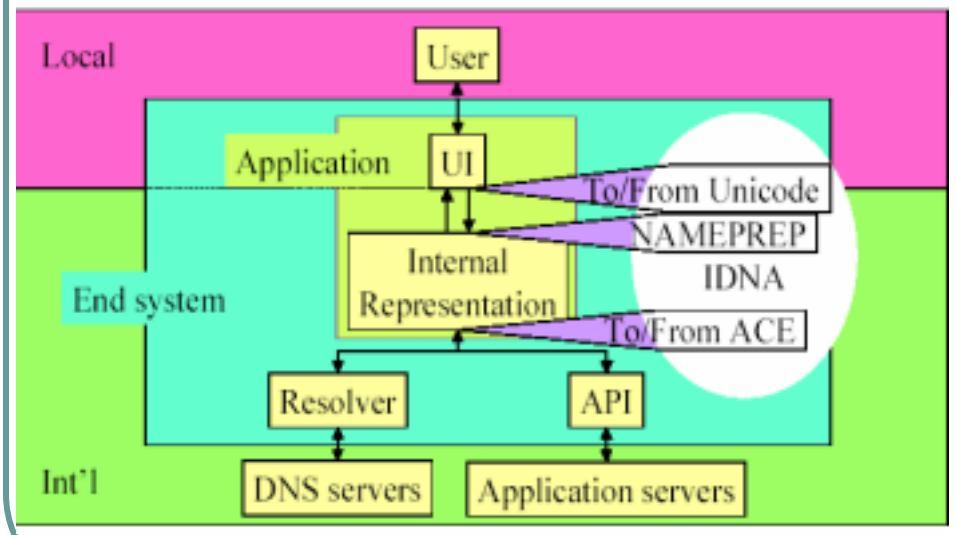
於是IDNA這篇RFC便提出了一個可以使UNICODE可以在DNS系統上實作的機制

而 Nameprep 包含了Mapping，Normalization，Prohibit 三部份 Punycode 則是一個將UNICODE字元轉換成爲ACE (ASCII Code Enable) 的方法

有關IDN詳細資料請參考http://cdns.twnic.net.tw/seminar3/agenda602.htm

IETF先前所提出配合IDN Nameprep 的 Stringprep RFC:RFC3454 Preparation of Internationalized Strings (stringprep)

國際域名(IDN)標準(2)



8

整個IDN架構圖

1. 使用者端送出查詢
2. 系統(目前系統並不直接支援，所以必須外加client 程式)將所輸入的字元轉換成UNICODE字元 ex: 台灣BIG5->UNICODE， 大陸GB->UNICODE
3. UNICODE經過NAMEPREP處理
4. 轉換成punycode後才送往DNS Server或者其他的應用

這樣的好處是應用程式方面不需要大幅度的修改
只須小部份的設定調整即可

至2003年7月止，支援IDN的client:
Mozilla-1.3b (http://www.mozilla.org/)
Opera6 (http://www.opera.com/)

國際域名(IDN)標準(3)

- 中文字應透過某種編碼方式轉換成英文字，並與舊有的 DNS 系統相容
- DNS 設定及伺服器設定應都根據這個編碼定義
- 目前國際上認可的編碼為 AMC-ACE-Z，稱為 puny code (RFC 3492)

EX:

http://xn--fiq43lrrlz83a.tw/ 台網中心.tw
http://xn--fiq64bh55hj6p.tw/ 中華電信.tw
http://xn--nqq28iuws7nz.tw/ 數位聯合.tw

PUNYCODE則是將一個經過NAMEPREP處理過之IDN，從一個8位元的編碼的形式轉換為7位元編碼的形式.目前Internet DNS一向是7位元ASCII編碼的環境，經過這個PUNYCODE轉碼程序，便將IDN從一個8 位元的IDN轉換為與現有DNS環境相容的編碼

國際域名(IDN)標準(4)

- 目前IDN標準所不足的部分
  - 各語系的特定需求
    - 中文缺少繁簡間的關係定義
  - UNICODE部分語系字符集有重疊狀況
    - 中文，日文，韓文漢字部分使用相同字符集
- 解決方案
  - IDN Admin Guideline (draft-jseng-idn-admin-03)
  - CDNC 字表

10

目前IDN的規範中並沒有針對繁簡的問題作處理:

Ex:

話<->话 (繁體與簡體意義完全相同)

UNICODE中字型一樣時，字碼相同

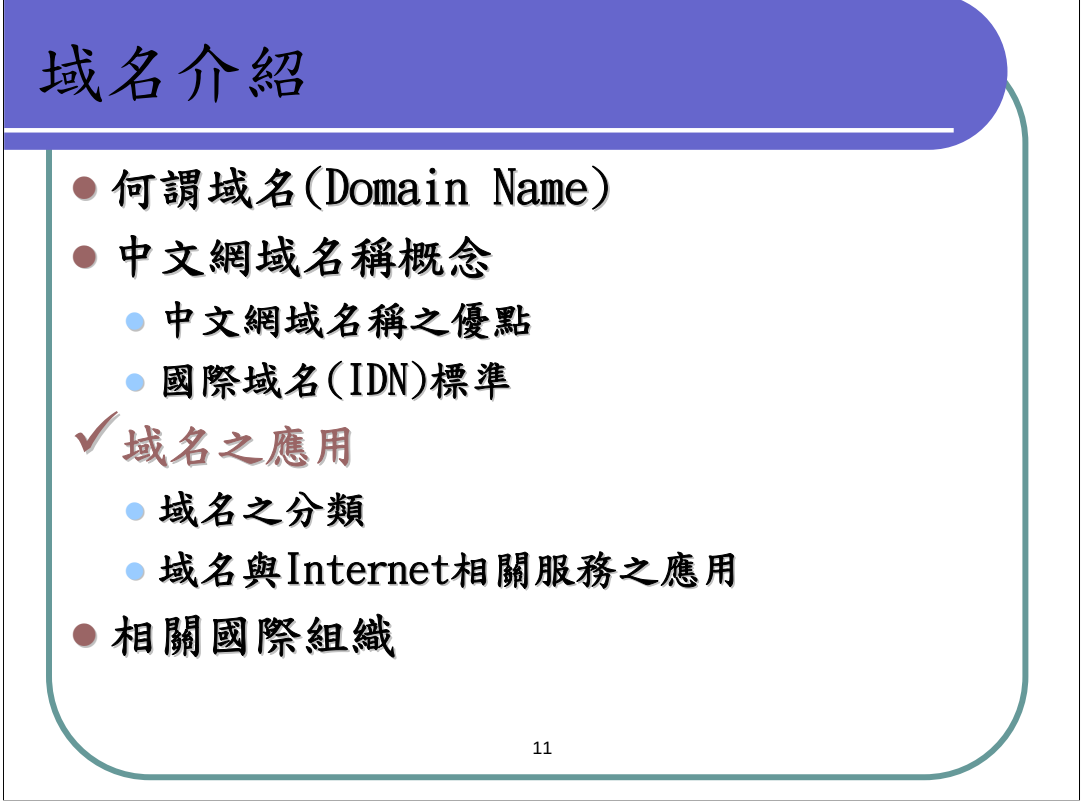
Ex:

台灣使用的BIG5與大陸使用的GB中與日本使用的JIS和韓國使用的KSC中的 “大”這個字，對應到 UNICODE 都是同一個字(字碼是 5927)

IDN Admin Guideline (ftp://ftp.rfc-editor.org/in-notes/internet-drafts/draft-jseng-idn-admin-03.txt)

此篇是由JET (CNNIC，TWNIC，KRNIC，JPNIC等專家組成)提出，主要依據各語系所提出之字表處理每個原型字與對照字和相關字之間的關係

目前 TWNIC 已於五月份將字表向標檢局提出申請



域名介紹主要針對網域名稱的基本概念，作一簡單介紹，主要內容包含下列幾點:

1.Domain Name 與 Domain Name System (DNS)

2.IDN的發展以及相關技術標準及中文網域名稱的發展狀況

3.網域名稱種類的介紹以及網際網路相關服務與網域名稱的相互關係

4.相關的國際網路組織介紹其主要任務及成立原由

域名之分類

- 分類：在區分不同的屬性
  - Top Level Domain（TLD）頂級域名
    - gTLDs:
      - com/net/org/gov/edu/…　共13類
    - ccTLDs:
      - tw/cn/jp/us 共 243 個
  - Second Level Domain（第二層域名）
    - com.tw/org.tw/ 等
- 目前 tw 之第二層域名
  - com.tw/net.tw/org.tw/edu.tw/gov.tw/mil.tw
  - idv.tw/game.tw/club.tw/ebiz.tw

gTLD

COM，NET，ORG，MIL，GOV，EDU，INT，ARPA aero， biz，coop， info， museum， name， pro

ccTLD

TW，CN，JP…(ISO 3166-1)

在ccTLD依各國管理政策有下列三種情況:

單位名稱，如: ibm.uk

屬性名稱，如: com.tw， net.tw， ne.jp

地理名稱，如: tokyo.jp

ICANN在最近新增了七個gTLD:
aero， .biz， .coop， .info， .museum， .name， .pro

域名與Internet相關服務之關係

- 為 Internet 服務最基礎的一環
- 提供機器名稱與 IP 位址雙向對映的機制
  - WWW　www.hinet.net <-> 168.95.1.82
  - MAIL　msa.hinet.net <-> 168.95.4.211
- 網域名稱比 IP 容易記， 且具代表意義
- 使用網域名稱讓系統更具移值性，當 IP 變動，只需更改 DNS 設定即可，程式 網頁等不需更改
- 隨著 IPv6 (16 bytes) 的推展，更需要使用網域名稱

13

使用者經常使用正解來查詢IP

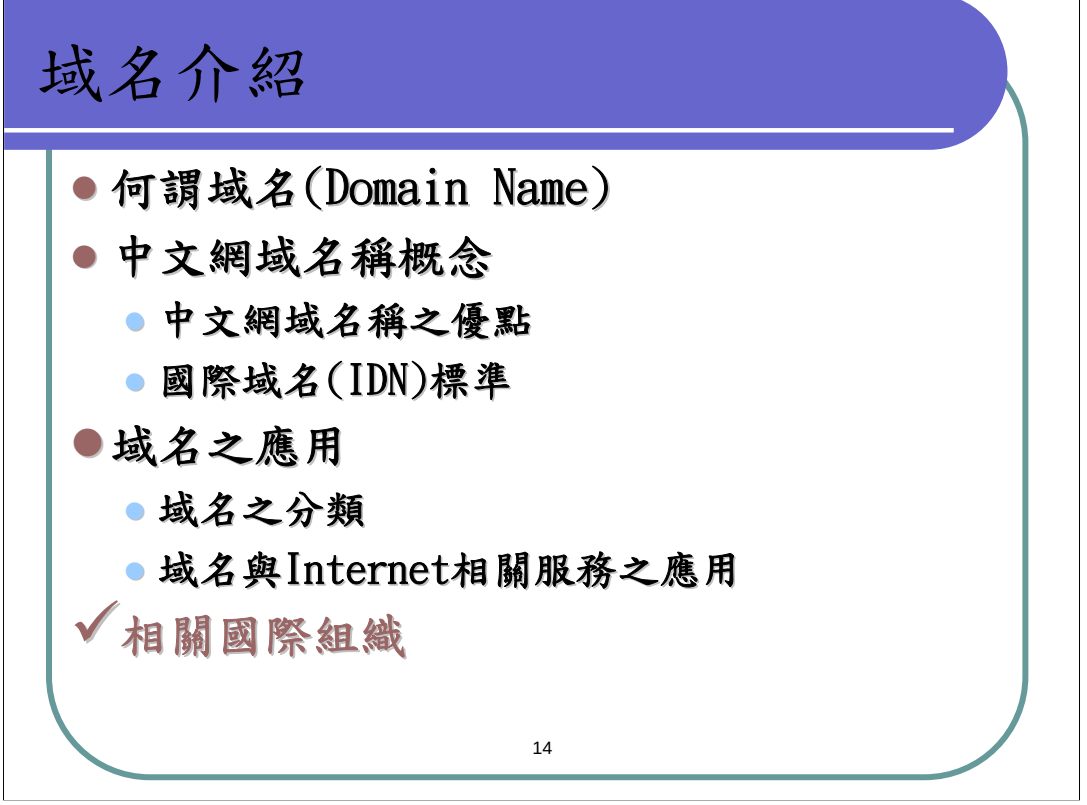
電腦經常使用反解來查主機名稱

電腦會作反解的行爲主要有下列兩個目的：

1.讓使用者或管理者容易了解連線情形，如在畫面或Log中顯示網域名稱會比顯 示IP讓人更容易了解連線對象是那個單位

2.安全考量，如某些server會以連線對象的IP查網域名稱，再查網域名稱對應的IP是否一致

DNS提供主機名稱及IP的精確對應，但DNS並不是一個目錄服務，無法提供檢索功能



域名介紹主要針對網域名稱的基本概念，作一簡單介紹，主要內容包含下列幾點:

1.Domain Name 與 Domain Name System (DNS)

2.IDN的發展以及相關技術標準及中文網域名稱的發展狀況

3.網域名稱種類的介紹以及網際網路相關服務與網域名稱的相互關係

4.相關的國際網路組織介紹其主要任務及成立原由

相關國際組織(1)

- IANA:

  Internet Assigned Numbers Authority 一個由IAB

  所資助的組織，任務是管理與分配 IP 位址，

  ccTLD 的註冊與管理

- ICANN

  The Internet Corporation for Assigned Names and Numbers 負責 Internet IP address 分配及網域名稱架構規範的管理單位

15

IANA: http://www.iana.net

ICANN: http://www.icann.org

ICANN是一個非營利社團法人組織，負責網路位址(IP address)空間的分配，協定參數的配置，網路名稱系統的管理與根伺服器系統的管理.目前由IANA和其他單位與美國政府約定並行管理.

ICANN董事會由十九位董事所組成，九位 At-Large 董事，九位 ICANN 三個支援組織所提名的董事，及一位總裁

中文: 網際網路名稱與號碼指配機構

1998年11月在美國商業部的主導下成立.

總部設於美國加州 Marina del Rey

相關國際組織(2)

- IETF

  Internet Engineering Task Force 由與網際網路（Internet）相關的產業及學術機構人員所組成的組織，為網際網路最主要的標準制定組織，Internet Society下的一個委員會.

- APTLD

  Asia Pacific Top-level Domain Forum
  討論亞太區Domain Name， DNS， NIC相關議題.會員包括亞太區主要地區及國家: 台灣、中國、日本、南韓、紐西蘭、澳洲、馬來西亞、泰國、新加坡、香港、越南等.

16

IETF: http://www.ietf.org

IETF創於1986年，是Internet技術標準的組織

IETF體系結構分爲三類，一個是網際網路架構委員會（IAB），第二個是網際網路工程指導委員會（IESG），第三個是在八個領域裏面的工作組（Working Group）.標準制定工作由工作組承擔.IAB成員由IETF與會人員選出，主要是監管各個工作組的工作狀況，它必須非常認真的考慮Internet是什麼，它正在發生什麼變化以及我們需要它做些什麼等問題.

APTLD即將於馬來西亞立案註冊.秘書處設於台灣， 由本中心擔任.

APTLD的成立宗旨，主要是爲了要滿足以下三個需求：

1、建立一個公開討論的場域，讓有關亞太區域網域名稱議題的資訊和專門知識得以進行交換和分類；

2、培育及促進亞太地區網域名稱相關組織積極參與地方性或國際性的公開討論；

3、提昇網域名稱系統的穩定性與持續性.

相關國際組織(3)

- APNIC

  Asia-Pacific Network Information Center， 亞太網路資訊中心

  - 主要掌管亞太地區的新IP address 申請， 及反解網域註冊. 詳情， 請參考 http://www.apnic.net .
  - 其他地區， 例如歐洲由 RIPE 代管.美洲由 ARIN 代管

APNIC: http://www.apnic.net

亞太網路資訊中心(APNIC)爲亞太地區的區域級網際網路註冊組織，負責亞太地區網際網路位址等相關資源的分配

APNIC是一個以會員爲基礎的非營利組織，同時會員也可以透過由下往上的決策管道，公開且民主地決定APNIC的各種政策與運作方向.身爲亞太地區最高之網際網路登記註冊組織，APNIC在其組織章程中載明其運作的目的在於：

1、提供亞太地區網際網路IP位址資源之註冊與分配服務，使全球網際網路得於暢通；

2、協助亞太地區網路團體發展有效分配網際網路資源的程序、機制與標準；

3、提供會員有關網際網路技術與政策的教育機會；

4、建立有益會員的公開政策與角色，尋求最符合會員的合理規範與制度.

相關國際組織(4)

- 各國NIC

  各個國家地區往往都有類似的單位， 如中國大陸(CNNIC)，日本(JPNIC)， 韓國(KRNIC)， 香港(HKNIC)，新加坡(SGNIC)，台灣(TWNIC)等.

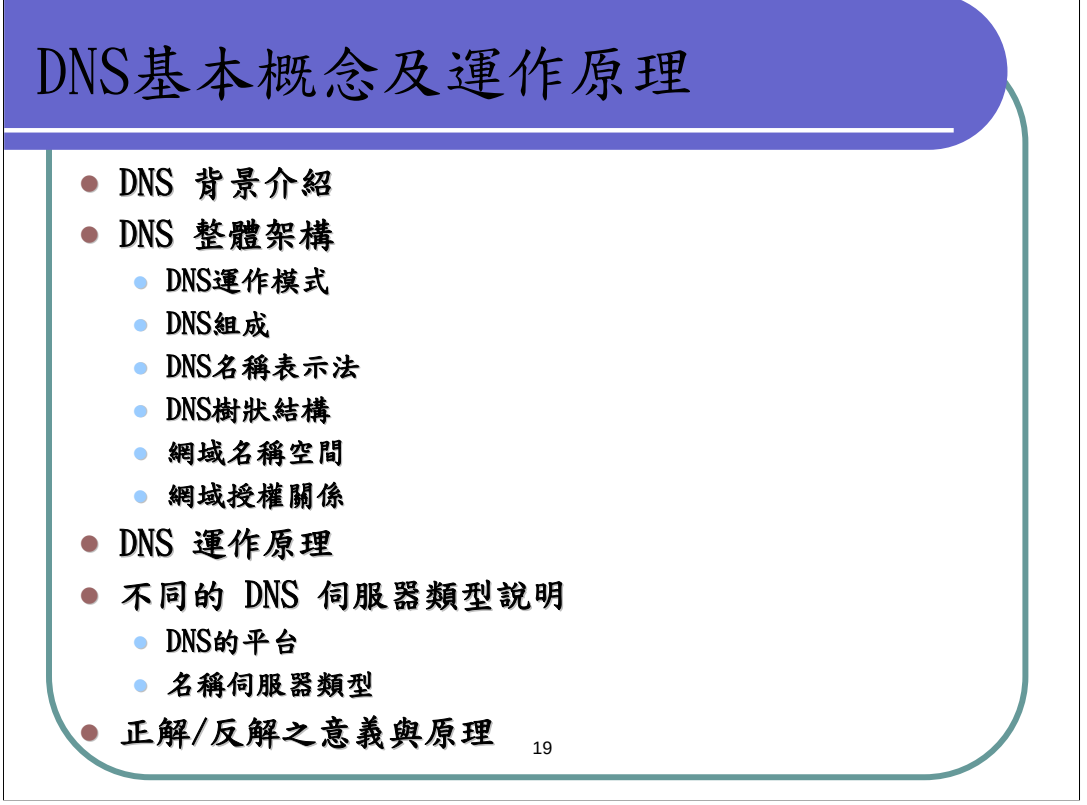
- CDNC
  - 兩岸四家網路資訊中心（即中國的CNNIC、台灣的TWNIC、香港的HKNIC、澳門的MONIC）發起組建的中文域名協調聯合會，為一個獨立、非營利的組織，該機構將在國際上擔負起中文域名的協調和規範工作

各國NIC主要由 ccTLD 組成

ccTLD(country code TLD)如“.tw”(台灣)，“.jp”(日本)，“.uk”(英國)…(ISO-3166所定義的2個byte國碼)

CDNC: http://www.cdnc.org

2000年5月19日，中文域名協調聯合會（CDNC）由兩岸四地網路資訊中心（CNNIC、TWNIC、HKNIC、MONIC）在北京正式發起成立.作爲一個爲獨立、非盈利的組織，該機構將在國際上擔負起中文域名的協調和規範工作



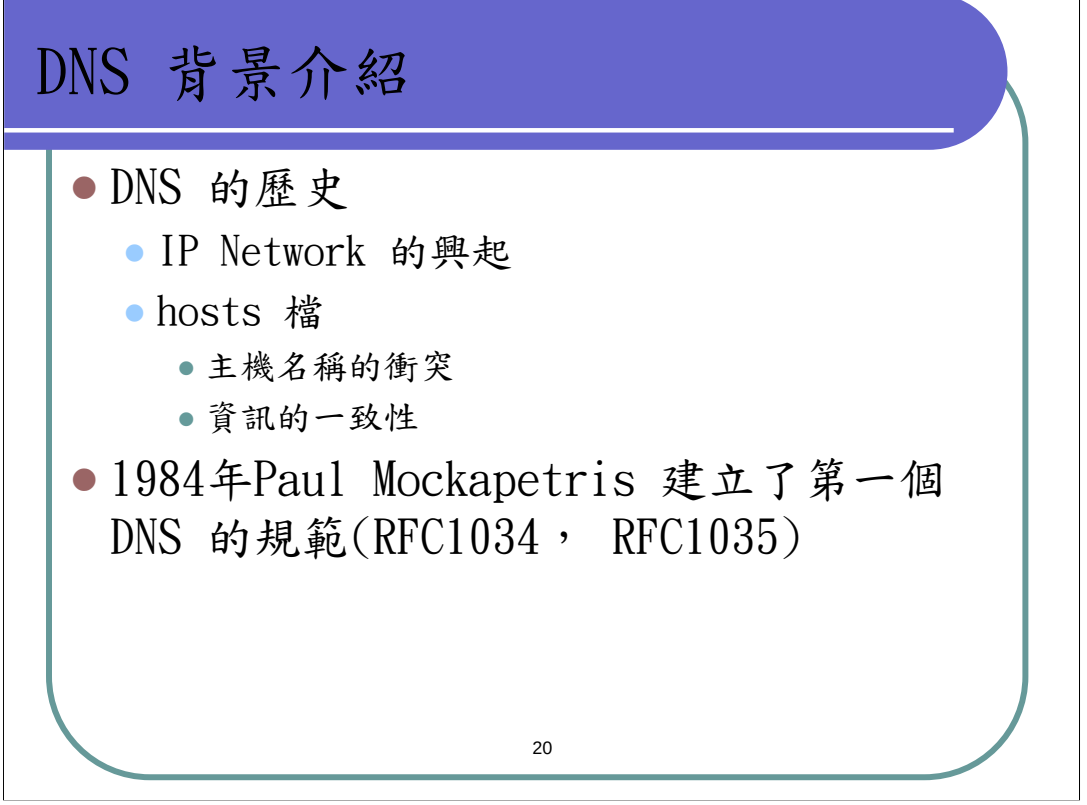
這個章節包含

1.DNS發展歷史

2.DNS的運作概念，整體架構以及必須了解的DNS基礎知識

3.DNS查詢的流程

4.網域名稱正解/反解所各自代表的意義以及影響



在 Unix-Base 的環境，hosts 位於 /etc 下，而 Window 的環境下則置於 /Win/system32/drivers/etc下，其格式爲:

;IP Hostname Alias

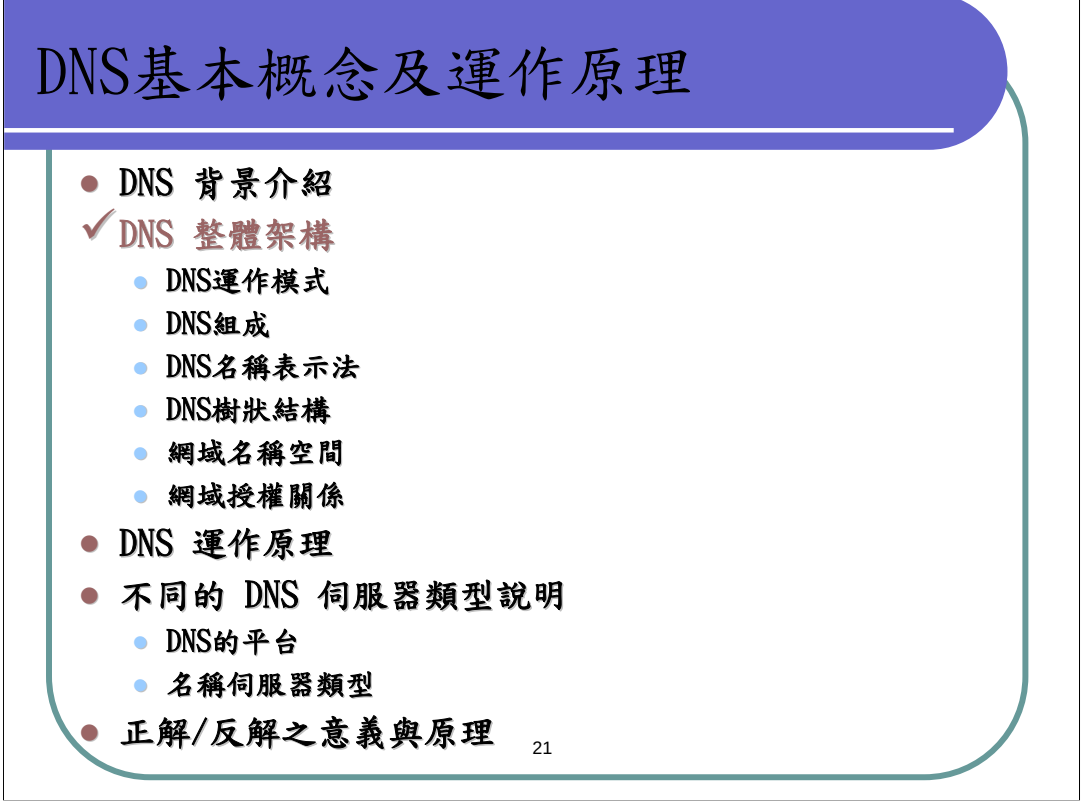
211.72.211.71 pc071.twnic.net.tw pc071

而一般作業系統設定之解析(Resolver) 流程順序都會設成先讀這個檔，若查得到則使用該 Hostname 之IP，若查不到則詢問 DNS，故 Hosts 檔目前仍經常使用，尤其在其Alias 及功能，可以簡化不少鍵盤輸入的長度，如 telnet pc071，即會連線到 211.72.211.71，而可不必輸入過長之 pc071.twnic.net.tw

Hosts 之另一個較少人知功能在於在機器的反解，因爲有許多服務會檢查反解是否設立，而設於 DNS 對於一般人而言較有困難，偷懶的方法即在於 hosts 中指定，但同樣的，僅限於本機有效.

使用 Hosts 雖有不少好處，但有時難免會用 DNS 的資料會有不同的結果，維護 hosts 檔資料的正確性有時還是有必要的，這是不少人常忽略的地方.

RFC882/1034/1035 DOMAIN NAMES - CONCEPTS and FACILITIES 等多篇 RFC，主要在介紹網域名稱的概念及其階層屬性之概念，並訂定出許多 DNS 不同的記錄特性



這個章節包含

1.DNS發展歷史

2.DNS的運作概念，整體架構以及必須了解的DNS基礎知識

3.DNS查詢的流程(DNS 觀念圖文解說)

4.網域名稱正解/反解所各自代表的意義以及影響

DNS 運作模式

- 名稱查詢之服務
- 分散式
  - 自己的資料由自己維護，而其他人的資料則分散在全球
  - 沒有一台電腦會有全部的DNS資料
  - 全球最大的分散式資料庫系統
  - 以樹狀結構的方式找到目的位址(每個結點需將授權)
- 穩定
  - 負載平衡:可由 Master 主機自由的複製到 Slave 主機
  - 備援:一個網域名稱可有多台主機共同服務(輪流查詢)
- 樹狀結構
  - 經由全球唯一的 Root Server 達到正確搜尋的目的
  - Root Server 共十三部，每一部可能都有許多 Mirror (如 f.root-servers.net 有二三十部)
- 效率
  - 使用 UDP 封包
  - 查詢速度基本上都在 100 msec 內
  - 經由 Cache 來加快 DNS 的查詢

22

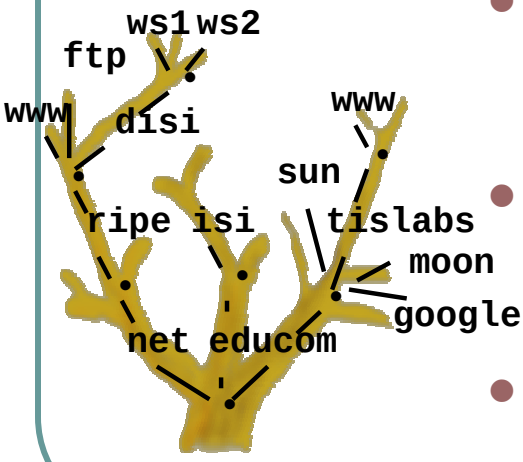
目前全球有超過一億部的 DNS 系統，以上述的特性運作，可正確且快速的解析到網域名稱與IP的對應，這些對應都由 Root (“.”) 開始，故其地位相當重要

若一個網域名稱有三部 DNS 主機，其間的資料同部以 Zone Transfer 達成，即 DNS 間即可進行同步做業，而不需其他額外成本(Master vs Slave)，而三部 DNS 主機再接受查詢時，是以隨機方式進行，即是每次查的主機皆不同，進而達到負載平衡之效果.

此外， DNS 查詢使用 UDP 封包，其效率與速度皆極快，並能降低 DNS 的負載，根據我們的測試，在 LAN 內 (100 BaseT)，使用一般 PC 即可達到每秒近 6000 次的查詢，使用好一點的機器則可以有更高的效率.

不過，UDP 封包有 512 bytes 的限制，故 DNS 查詢所帶的資料不可超過 512 bytes， 所以一個 UDP 封包 大抵上只能容納 13 部 Root Server，若含有 IPv6 的主機的話，基本上這個數字會再降低(IPv6 的 Address 有 16 個 bytes)

DNS 組成



- DNS 系統如同一樹狀結構
- 每一個分支以 “.” 分隔
- 其限制
  - 最多 127 層
  - 每個分支最長63 字元

domain name 的 label 由 a-z(大小寫)， 0-9 及 “-” 所構成(“-”不得在第一及最後一個字出現),長度最長爲 63 個 bytes, labels 間以“.”連結，每個 domain name 最長爲 255 bytes，最多由 127 個 label 組成

Top Level Domain (TLD，頂級域名)

gTLD (Global TLD)

COM，NET，ORG，MIL，GOV，EDU，INT，ARPA aero， biz，coop， info， museum， name， pro

ccTLD(Country Code TLD)

TW，CN，JP…(ISO 3166-1)

在ccTLD依各國管理政策有下列三種情況:

單位名稱，如: ibm.uk

屬性名稱，如: com.tw， net.tw， ne.jp

地理名稱，如: tokyo.jp

DNS 名稱表示法

Fully Qualified Domain Name (FQDN)

WWW.EP.NET.

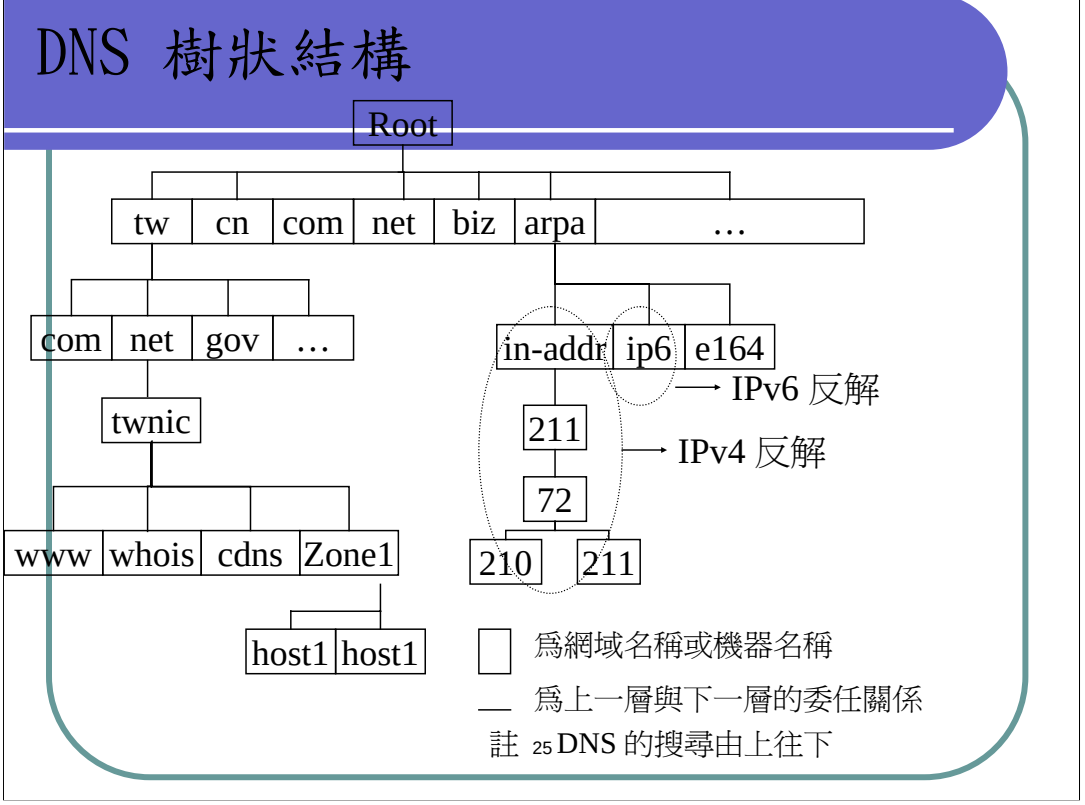
注意結尾的點

- 每一個名稱間以 . 隔開
- 一個 FQDN 可以對應到不同的位置或服務
  - 一個名稱對應到多個 IP 稱為 Round Robin
  - 一個名稱對應到不同的服務如 MX
- 每個 FQDN 如同 IP 一般皆具有唯一性

Ex:

www.ipv6.org.tw. 對應到 210.17.9.228 (IPv4,A)

或 2001:c50:ffff:1:2e0:18ff:fe95:b229 (IPv6,AAAA)



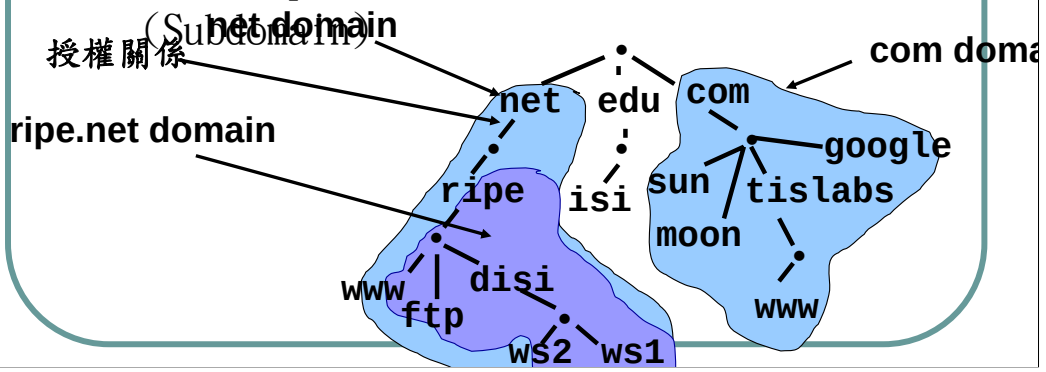
DNS 的階層式架構並不僅此於其隸屬關係，亦是一個查詢時的流程，當我們要查詢 www.twnic.net.tw 時，當您的 Client DNS 收到這個查詢請求時，並不知道要去何處詢問，所以會從根 (Root .)開始詢問，一路的找下來，並找到其結果爲 210.17.9.228，所以，這個根就很重要了，根沒有了，全世界整個 DNS就會出現很大的問題.目前根伺服器在全世界共有13部，它們扮演著極其重要的角色.依此推論，如果 tw 這個位置的 DNS 出了問題，代表著所有台灣網域名稱的服務也會跟著出問題，故所處位置愈上層，其重要性也愈高，其穩定度也就更受關切.故，行政院資安會報將 TWNIC 列爲二十個重要的系統，並屬於 A+ 級，要求二十個重要單位要在今年內取行資安認證，而 TWNIC 於今年三月份亦已通過 BS7799 之認證，故在一般的公司， DNS 的安全亦不可不慎.

前段所述爲一網域名稱求 IP 的流程 (正解)，這張表上同時亦列出 IP 的階層關係，因爲域名之特性是有後往前推(後序)，故 IP 在網路上定成名稱時，其結構即爲 x.210.72.211.in-addr.arpa.，查詢此一名稱即代表要查其域名(反解).其中 arpa. 是用於數字上，in-addr 表示 internet address (一般人往往不知爲何要設 in-addr.arpa.)，ip6 用於 IPv6 位址.

上圖 tw cn 該排，通常我們會稱爲 TLD (Top Level Domain) ， 其中國碼的部份爲 ccTLD，非國碼的部份爲 gTLD，而其下的則稱爲 SLD (Second Level Domain，ex: com.tw)，目前除了13部的 Root 伺服器外，共有 243 個 ccTLD 及 13 個 gTLD， 而台灣目前的 SLD 計有 Com.tw/net.tw/org.tw/edu.tw/ mil.tw/gov.tw/game.tw/ ebiz.tw/club.tw/idv.tw (稱爲屬性型，因其帶有特性)等，及 中文.tw (稱爲泛用型) 等不同之 SLD.

網域名稱空間

- 網域即是一個名稱空間（namespaces）
- 在 .com 之下的稱為 com 網域
- 在 ripe.net 之下的稱為 ripe.net 網域，同時也是 .net 網域
  - 此時 ripe.net. 可說是 net 的Zone (Subdomain)



DNS是一個分層級的分散式名稱對應系統，在最頂端的是一個“root”，接下來是TLD(Top Level Domain)，TLD又分爲gTLD(generic TLD)如“.com”，”.org”，”.net”，”.edu”，”.gov”，”.mil”，”int”，”.arpa”及ccTLD(country code TLD)如“.tw”(台灣)，“.jp”(日本)，“.uk”(英國)…(ISO-3166所定義的2個byte國碼)

目前“.com”，”.net”由networksolutions(已被verisign買下)公司所經營，”.edu”，”.gov”，”.mil”分別爲美國的教育單位、政府單位、軍事單位，”.int”爲一些國際間的需求(如internet fax)使用，“.arpa”原本爲arpanet(internet的前身)單位所使用，現爲DNS反解等使用.

ICANN在最近新增了七個gTLD:
aero， .biz， .coop， .info， .museum， .name， .pro

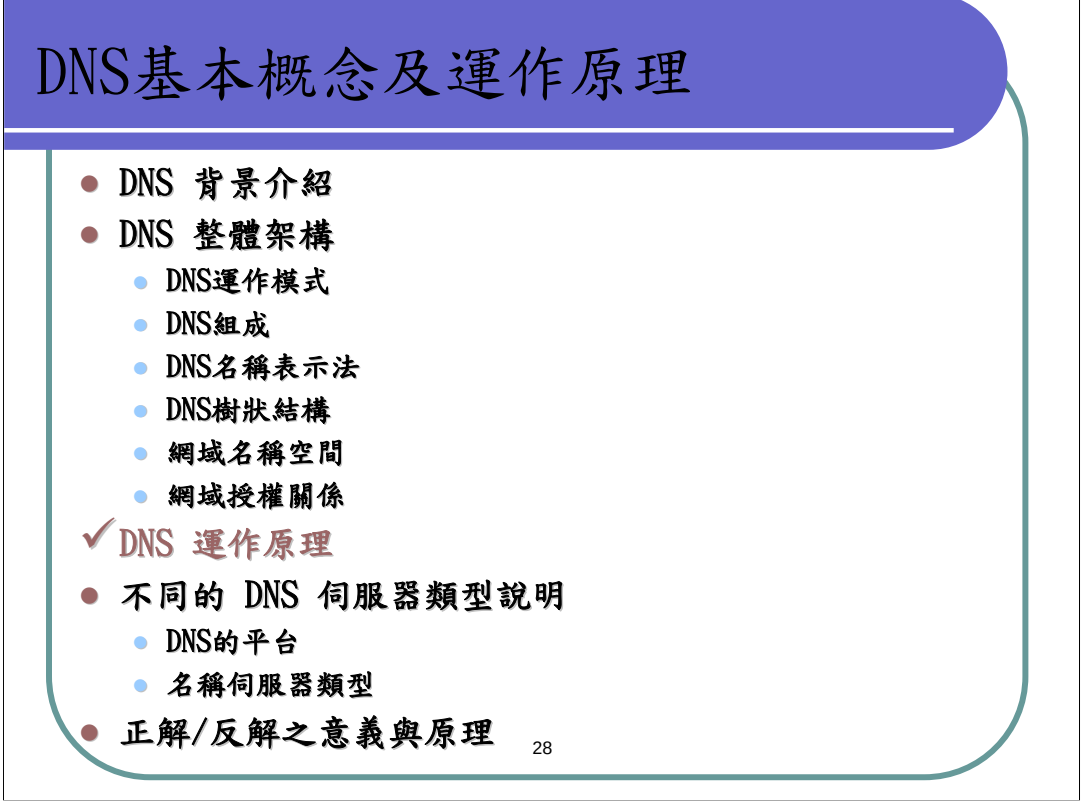
網域授權關係

- 網域名稱的管理者可以建立不同的子網域給不同的部門或單位使用
  - 如學校系所，或較大之公司
- 其亦可將此子網域授權他部門自行管理
- 在上一層的網域需指出這種授權的關係
  - 以 NS 記錄達到授權關係
- 整個 DNS 樹狀結構即是依此完成
  - 如 ntcu.edu.tw 下系所 cise.nctu.edu.tw 等

DNS 中的 NS 紀錄就用來指定下一層 sub-domain name server所在的位置,Ex:

```
$TTL 86400
twnic.net.tw        IN SOA  dns.twnic.net.tw. snw.twnic.net.tw. (
                    19991268   ; serial
                    7200       ; refresh (2 hours)
                    1800       ; retry (30 minutes)
                    3600000    ; expire (5 weeks 6 days 16 hours)
                    172800     ; minimum (2 days)
                    )
                NS     dns.nic.net.tw.
                NS     dns.twnic.net.tw.
                NS     twnic.net.tw.
                A      211.72.210.250
agent           NS     ns1
                NS     ns2
```

agent.twnic.net.tw 就是一個子網域，name server爲 ns1.twnic.net.tw 及 ns2.twnic.net.tw，在這兩部 DNS 主機上再建立 agent.twnic.net.tw 的網域名稱，外界即可查詢到這個子網域， com.tw 亦是依此例之方式達到 xxx.com.tw 之效果



這個章節包含

1.DNS發展歷史

2.DNS的運作概念，整體架構以及必須了解的DNS基礎知識

3.DNS查詢的流程

4.網域名稱正解/反解所各自代表的意義以及影響

運作原理(1)

- 當被詢問到有關本域名之內的主機名稱的時候，DNS伺服器會直接做出回答(此一答案稱為權威回答(Authoritative Answer)，此一主機稱為權威主機)
- 如果所查詢的主機名稱屬於其它域名的話，會檢查快取(Cache)，看看有沒有相關資料
- 如果沒有發現，則會轉向root伺服器查詢，然後root伺服器會將該域名之授權(authoritative)伺服器(可能會超過一台)的地址告知

即是若您的 DNS 管理這個網域名稱時，有人向你查詢這個名稱的資料，可以直接做出回答，因您的 DNS Server 即是負責此一網域名稱之主機，所以稱爲權威主機，而回應的結果稱爲權威回答

在每一個名稱伺服器中都有一個快取暫存區(Cache)，這個快取暫存區的主要目的是將該名稱伺服器所查詢出來的名稱及相對的IP位址記錄在快取暫存區中，這樣當下一次還有另外一個用戶端到次伺服器上去查詢相同的名稱時，伺服器就不用在到別台主機上去尋找，而直接可以從暫存區中找到該筆名稱記錄資料，傳回給用戶端，加速用戶端對名稱查詢的速度，而您的 FQDN 資料要被人快取多久，則是由您的記錄上的 TTL 欄位決定(後面章節將介紹到 TTL)

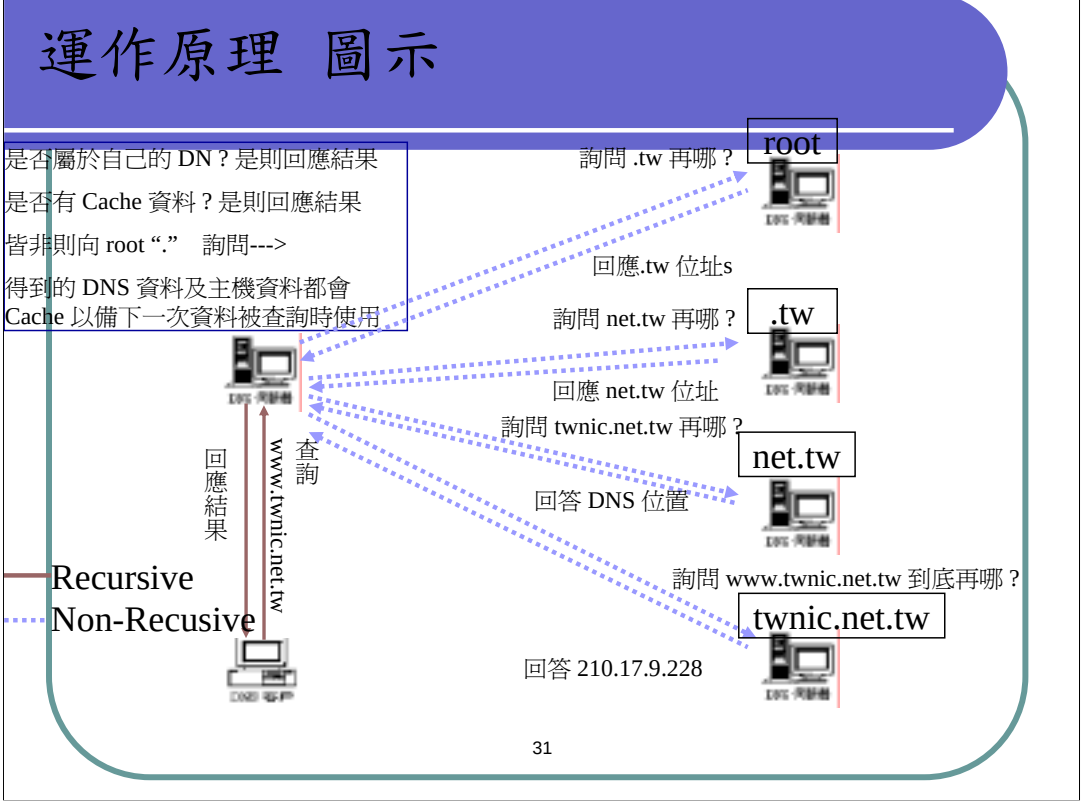
如果名稱伺服器在資料記錄查不到且快取暫存區中也沒有時，伺服器首先會才會向 Root Server 查詢，所以 Root Server 在 Internet 上扮演極爲重要的角色，目前 Root Server 由 ICANN 管理.

運作原理(2)

- 本地伺服器然後會向其中的一台伺服器查詢，並將這些伺服器名單存到記憶體中，以備將來之需(省卻再向root查詢的步驟)
- 遠方伺服器回應查詢
- 將查詢結果回應給客戶，並同時將結果儲存一個備份在自己的快取記憶裡面
- 如果Cache資料的時間尚未過期之前再接到相同的查詢，則以存放於快取記憶裡面的資料來做回應

30

1. DNS用戶端向指定的DNS伺服器查詢網際網路上某台主機名稱
2. 當DNS伺服器在該資料記錄找不到用戶所指定的名稱時，會轉向該伺服器的快取暫存區找尋是否有該資料
3. 當快取暫存區也找不到時，會向最接近的名稱伺服器去要求幫忙找尋該名稱的IP位址
4. 在另一台伺服器上也有相同的動作的查詢，當查詢到後會回覆原本要求查詢的伺服器
5. 該DNS伺服器在接收到另一台DNS伺服器查詢的結果後，先將所查詢到的主機名稱及對應IP位址記錄到快取暫存區中
6. 最後在將所查詢到的結果回覆給用戶端



Recursive 遞迴查詢:

使用者只送出一個查詢， 由DNS Server 完成其他所需的查詢後回應

Non-Recusive 反覆查詢(非遞迴查詢)

每一個查詢直接給予指示性的回應， 由使用者端再進行後續的查詢

通常Client到Server之間是 Recursive 查詢

Server到Server之間是Interactive 查詢

DNS解析流程(1)

- 讓我們一步一步來看DNS解析的步驟:

Pc001.abc.com.tw

```
ping www.twnic.net.tw.
```

32

1.當被詢問到有關本域名之內的主機名稱的時候，DNS伺服器會直接做出回答;

2.如果所查詢的主機名稱屬於其它域名的話，會檢查記憶體，看看有沒有相關資料;

3.如果沒有發現，則會轉向root伺服器查詢;

4.然後root伺服器會將該域名之授權(authoritative)伺服器(可能會超過一台)的地址告知;

5.本地伺服器然後會向其中的一台伺服器查詢，並將這些伺服器名單存到記憶體中，以備將來之需(省卻再向root查詢的步驟);

6.遠方伺服器回應查詢;

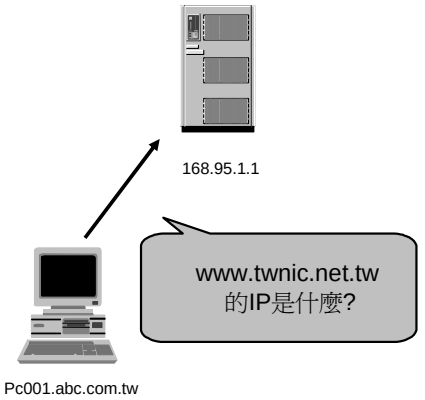
7.將查詢結果回應給客戶，並同時將結果儲存一個備份在自己的快取記憶裡面;

8.如果在存放時間尚未過時之前再接到相同的查詢，則以存放於快取記憶裡面的資料來做回應.

以上八點建議您能充份了解，對於在實做 DNS Server 時，可容易掌握問題所在

DNS解析流程(2)

- 個人電腦向他設定的DNS 168.95.1.1查詢www.twnic.net.tw的IP



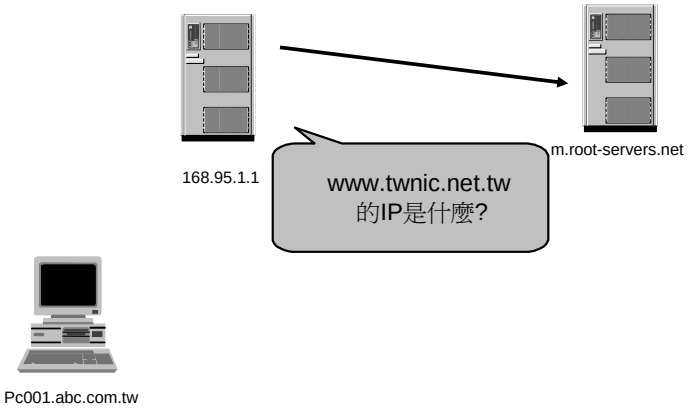

```
ping www.twnic.net.tw.
```

33

使用者電腦中所定義的name server爲一recursive DNS server(即會反復向其他DNS問出結果再將答案給使用者)， 而且這部name server在電腦中一定都是設定成IP， 因爲在向這部name server查詢前都還不會有網域名稱與IP的對應

DNS解析流程(3)

- 168.95.1.1會向root server M查詢www.twnic.net.tw的IP address

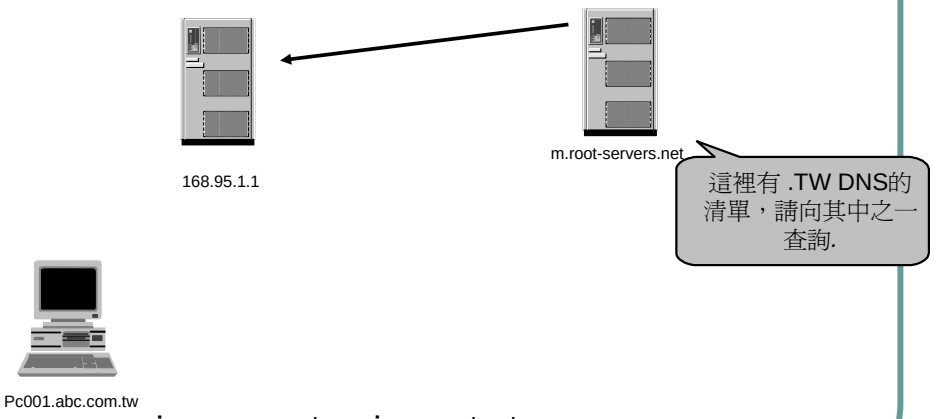



```
ping www.twnic.net.tw.
```


當一部name server剛啓動時，cache中是空的，除了named.cache檔案中所定義的十三部root server外沒有其他的資料，所以一開始一定要向這十三部root server之一發出查詢請求(隨機選擇)

DNS解析流程(4)

- M root server會回應 .TW 的dns在那裡

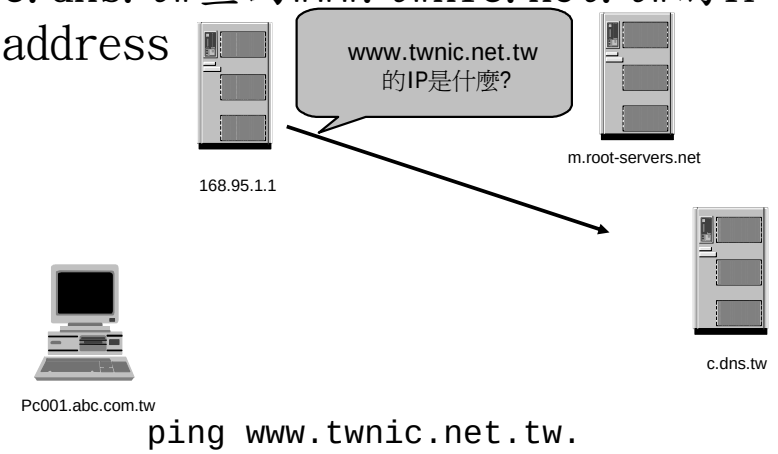


這種回應的型式我們稱爲 “referral” (即回應 NS 記錄)

一般而言對一個authority的name server都會config成這種型式,只會告訴查詢者下一站到那查詢，而不會主動到外面將結果查回來給查詢者,其原因在於遞迴主機會 Cache 別人的資料， 這可能會造成 DNS 欺騙的問題，如果你的DNS 主機被欺騙了，對您可能沒有什麼關係，但對上層 (root ， .tw or com.tw ...)等，因爲大家都會用到，所以會設成非遞迴，因爲非遞迴不會 Cache (不會代查別人的就不會 cache)， 會回應自己的 domain 及 root server list，故能有較高的安全性. 另外，非遞迴因爲只有收到，回應自己的資料，所以可以承受較高的查詢壓力(沒有幫別人代查的負擔)

DNS解析流程(5)

- 168.95.1.1會向.TW name server: c.dns.tw查詢www.twnic.net.tw的IP address

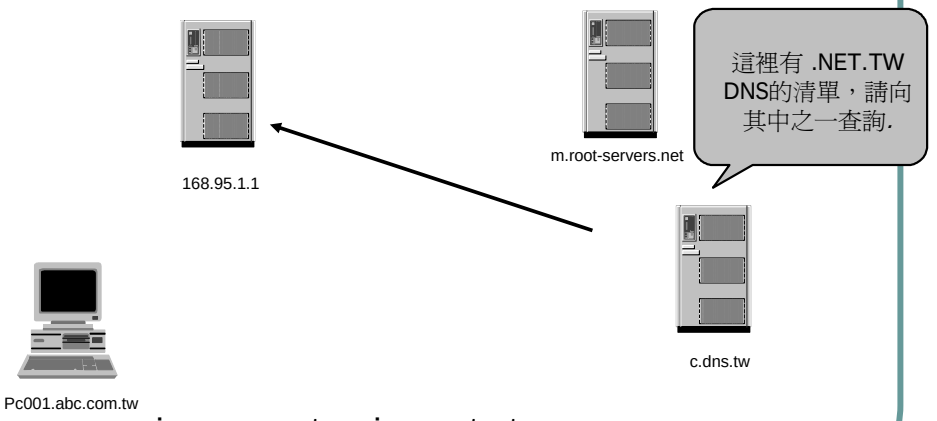



```
ping www.twnic.net.tw.
```


相同的 .TW DNS 也只是將 net.tw 的DNS在那告訴查詢者

DNS解析流程(6)

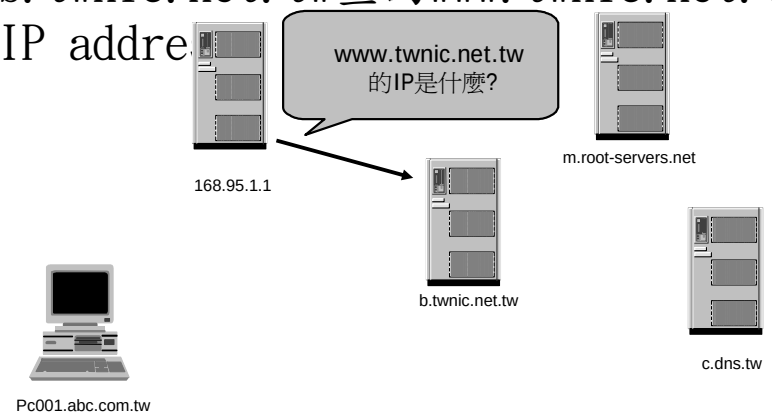
- c.dns.tw回應net.tw的DNS在那裡




```
ping www.twnic.net.tw.
```

DNS解析流程(7)

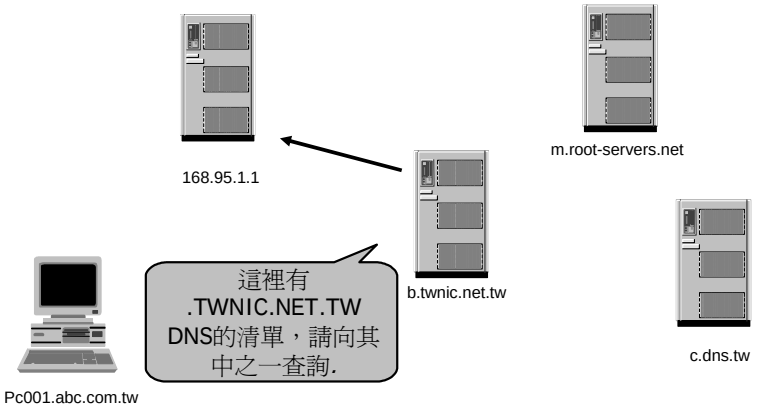
- 168.95.1.1會向.TW name server: b.twnic.net.tw查詢www.twnic.net.tw的IP addre




```
ping www.twnic.net.tw.
```

DNS解析流程(8)

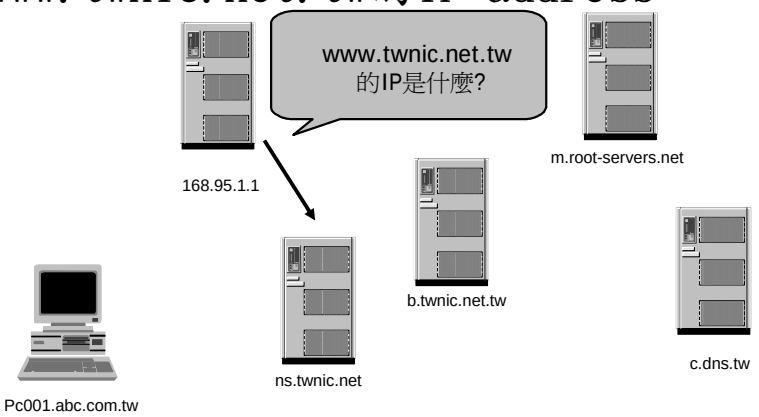
- b.twnic.net.tw回應twnic.net.tw的DNS在那裡




```
ping www.twnic.net.tw.
```

DNS解析流程(9)

- 168.95.1.1會向ns.twnic.net查詢www.twnic.net.tw的IP address

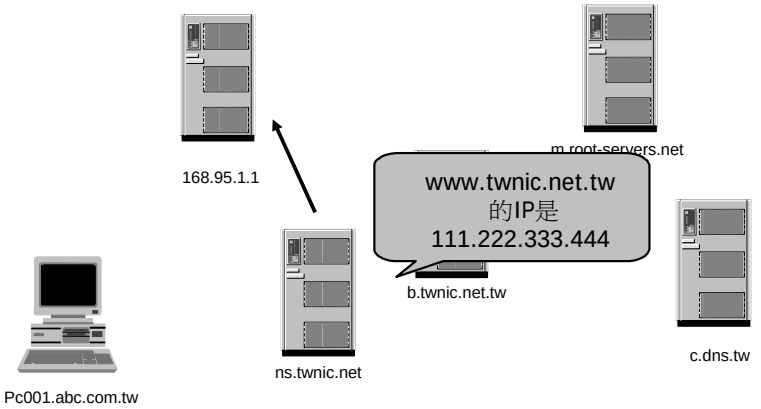



```
ping www.twnic.net.tw.
```


以上 DNS 主機選擇皆是隨機的， 上層 Server 會告訴查詢者其所有相關的 Name Server (NS) 列表，由 168.95.1.1 自己選擇向這 N 部那一部查詢下一站

DNS解析流程(10)

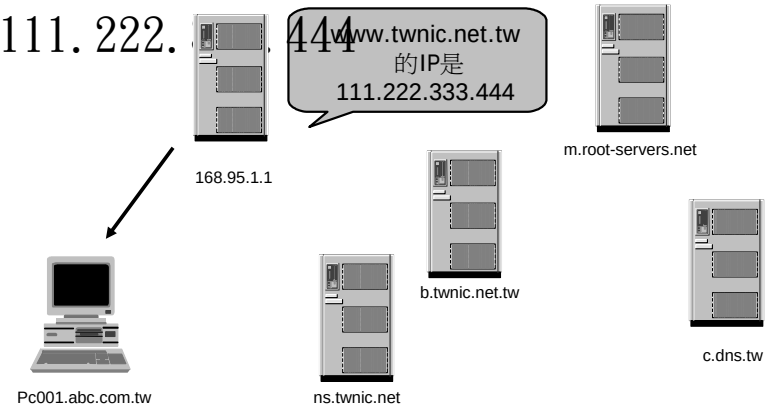
- ns.twnic.net回應www.twnic.net.tw的IP是什麼




```
ping www.twnic.net.tw.
```

DNS解析流程(11)

- 168.95.1.1回應pc001.abc.com.tw www.twnic.net.tw的IP是 111.222.333.444




```
ping www.twnic.net.tw.
```

DNS解析流程Caching(1)

- 在前次查詢後168.95.1.1知道了下列紀錄:
  - TW的dns及其IP
  - NET.TW的dns及其IP
  - TWNIC.NET.TW的dns及其IP
  - WWW.TWNIC.NET.TW的IP
- 讓我們看下一次的解析流程

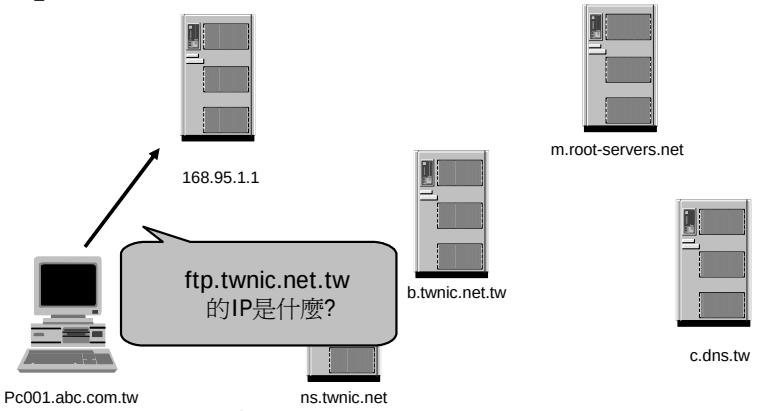
Pc001.abc.com.tw

```
ping ftp.twnic.net.tw.
```


所有DNS查詢的結果都會放在cache中以加速查詢,所以當下一次查詢相同網域名稱的資料時就不會從 root server往下問.

DNS解析流程Caching(2)

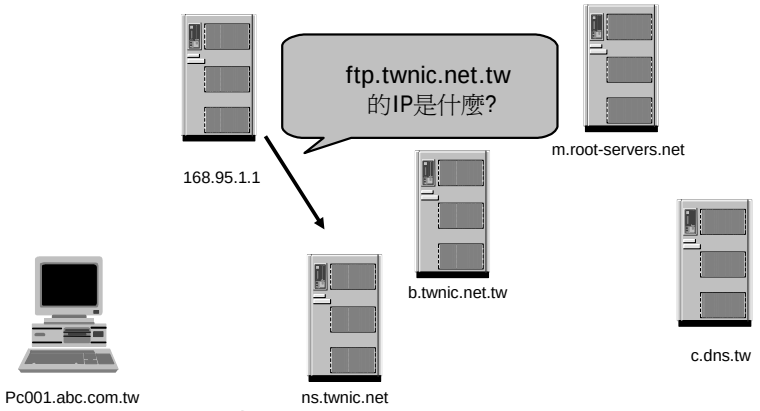
- 個人電腦向他設定的DNS 168.95.1.1查詢ftp.twnic.net.tw的IP




```
ping ftp.twnic.net.tw.
```

DNS解析流程Caching(3)

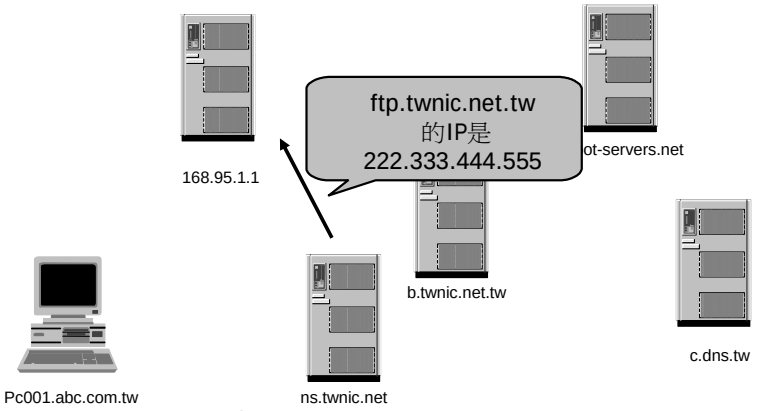
- 168.95.1.1已經有twnic.net.tw的NS紀錄， 所以直接過去詢問ftp.twnic.net.tw的IP




```
ping ftp.twnic.net.tw.
```

DNS解析流程Caching(4)

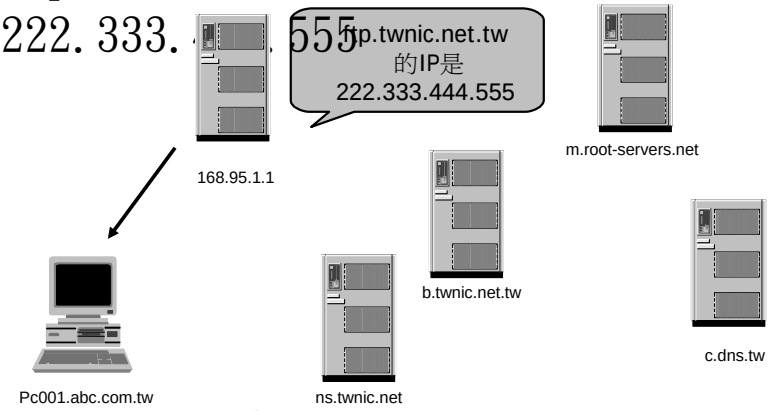
- ns.twnic.net回應ftp.twnic.net.tw的IP是什麼




```
ping ftp.twnic.net.tw.
```

DNS解析流程Caching(5)

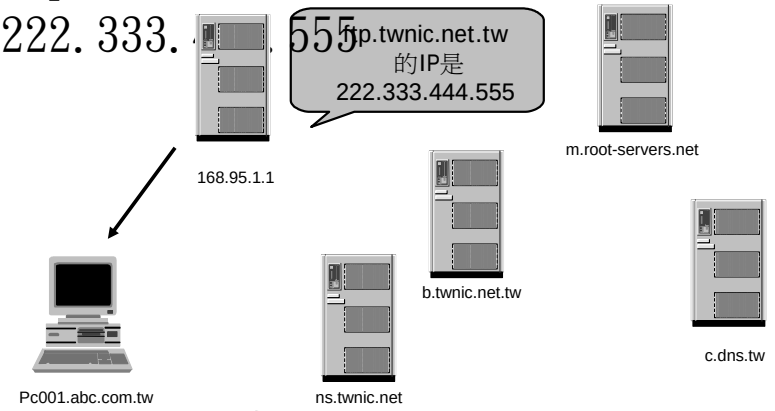
- 168.95.1.1回應pc001.abc.com.tw ftp.twnic.net.tw的IP是222.333.444.555




```
ping ftp.twnic.net.tw.
```

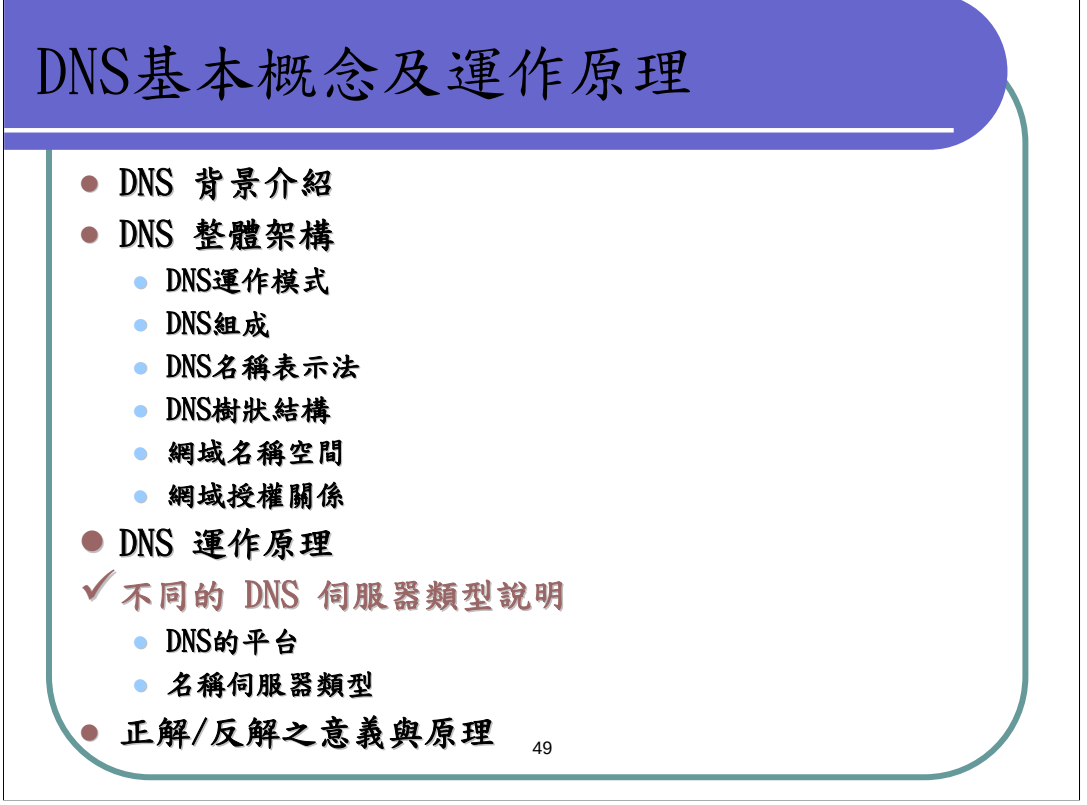
DNS解析流程Caching(5)

- 168.95.1.1回應pc001.abc.com.tw ftp.twnic.net.tw的IP是 222.333.444.555



ping **ftp**.twnic.net.tw.

以上範例之 IP 僅爲 sample ， 實際上僅以您自己自己之狀況調整，若您有 sniffer (封包監測)之類軟體，當您重開 DNS 後(避免 Cache 因素)，以上列測試時，在在抓取封包，即可明白其查詢流程.



這個章節包含

1.DNS發展歷史

2.DNS的運作概念，整體架構以及必須了解的DNS基礎知識

3.DNS查詢的流程

4.網域名稱正解/反解所各自代表的意義以及影響

DNS 的平台

- UNIX
  - 常見為ISC BIND
  - 共約發行三十幾個版本 (4.X~9.X)
  - 最新版本 4.9.9(不再維護)， 8.4.1，9.2.2
  - 建議使用 9.2.2 版本
  - 穩定，可靠，最多人使用
- Windows
  - 可見於Windows Server 級的版本
  - 簡單設定是其優點
  - GUI 設定
  - 根據 BIND 4.x 修改而來

V8.x及V9.x皆整套程式重新改寫

/etc/named.boot (v4) -> /etc/named.conf (v8)

zone transfer (named vs named-xfer)

no more fork for each AFXR jobs

(for v 8.1.X and later)

BIND **4.x/8.x** 版本的重大差異

新的功能

negative caching

bind_notify， IPv6 support， RFC 1535 compliant， …

BIND **8.x/9.x** 版本的重大差異

新的功能

DNSsec

IPv6 support(A6)

Domain alias(DNAME)

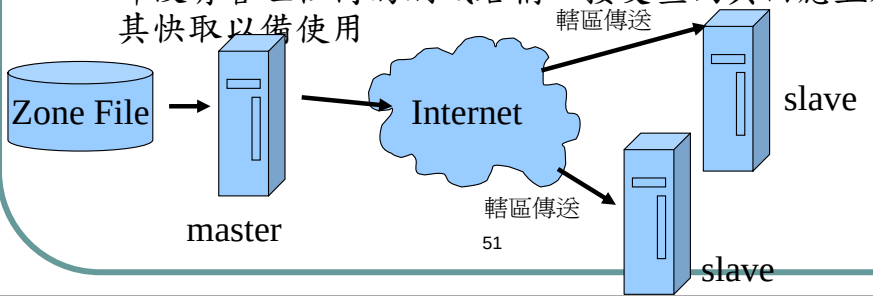
多國語文(8 bits)

目前使用最普遍的爲 9.x (主要因爲 OS distribution 所附的版本現已多 9.X)

但 ISP (HINET，SEEDNET) 仍有使用 8.x 之狀況

名稱伺服器類型

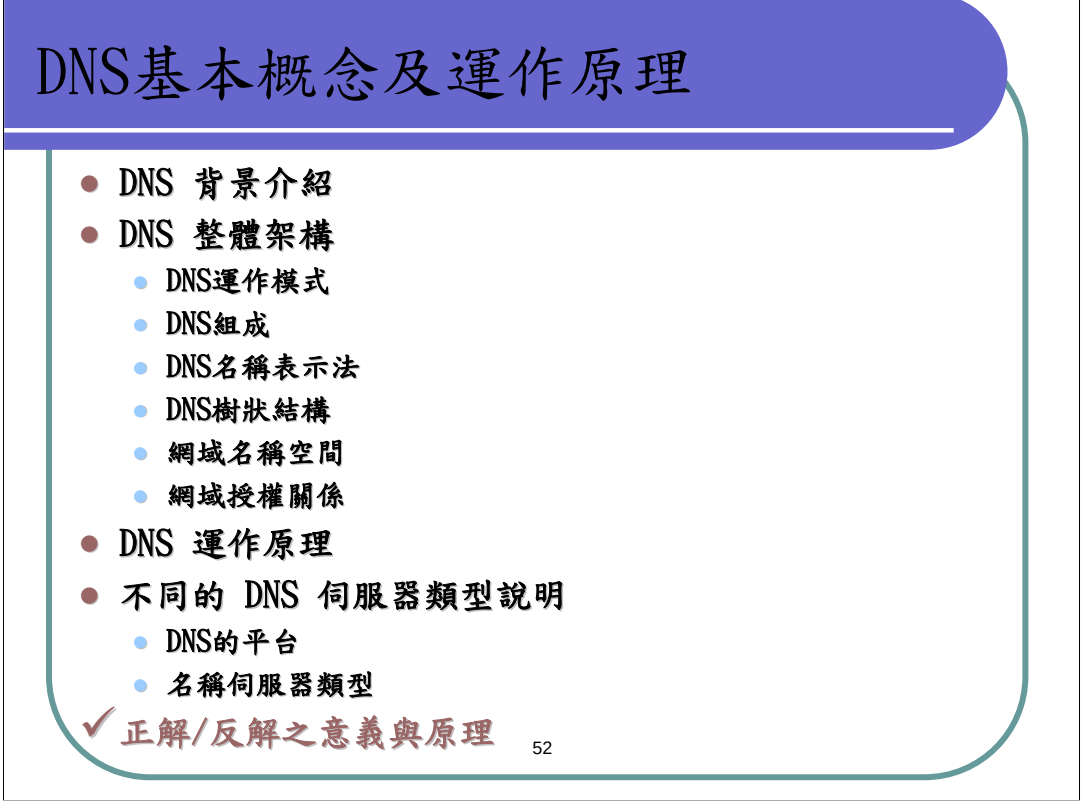
- 權威主機(Authoritative)
  - 可管理或回答其網域名稱之答案
  - Master 主機指 DNS 所管轄的資料是從檔案（ Zone File）中而來
  - Slave 主機指 DNS 所管轄的資料是以轄區傳送（Zone Transfer，簡稱 AXFR）從 Master 而來
- Cache-Only 主機
  - 即沒有管理任何的網域名稱，接受查詢與回應並將其快取以備使用



在舊版的Bind(4.x)稱爲primary與secondary，在新版的Bind(8.x以後)稱爲master與slave

master DNS伺服器是架設在某一個網域下被主要授權並控制所有名稱記錄的主控制伺服器，管轄著所有該網域的記錄資料，這些記錄資料只有master可以修改.

但如果在一個比較大型的網路中，DNS伺服器就會變得很繁忙了，所以您可以設定多個DNS來分擔master的工作，但您或許不願意到每一個DNS伺服器去更新資料吧? 而且就算您願意這樣做，也容易出現錯誤或資料不同步的情形.這樣您可以設定其它的伺服器爲slave DNS來複製master上面的記錄資料，這樣，其它的電腦可以被指定到不同的DNS做查詢，既可以分擔master的工作，而且資料也可以自動進行同步工作.您可以設定DNS資料同步的時間間隔，在dns檔案中的Refresh設定就是了.同時您還會看到Serial，當slave的上面的serial數字少於它，資料就會被複製，否則會被忽略.所以更新的 Zone File 的資料時，一定要記得更新 serial 數字 +1.



這個章節包含

1.DNS發展歷史

2.DNS的運作概念，整體架構以及必須了解的DNS基礎知識

3.DNS查詢的流程

4.網域名稱正解/反解所各自代表的意義以及影響

正解/反解之意義與原理

- 正解 (forward domain): 由機器名稱對應至 IP
- 反解 (reverse domain): 由 IP 對應至網域名稱
  - 反解的 DNS Query 遠比正解高出許多，這是一般人常忽略之處
- 正反解一致有其必要
  - 國內的系統較不嚴謹，比較不會檢查正反解的一致性，但國外有許多比例都會進行這個部分的確認
  - 由來源 IP 查反解名稱，依結果再查正解，並檢驗其結果
  - 有部分的Mail Server也會使用正反解確認的機制來減少SPAM的問題
- 網路上的 DNS 查詢何種較多 ？
  - 反解多，正比反約 2:3 ，原因是多數的服務皆會進行IP 來源的反查所致(Ex: WWW，MAIL，Firewall ….)

正解

com.tw，org.tw，net.tw，idv.tw ， game.tw ， club.tw ， ebiz.tw ， tw --- twnic及registrars

http://rs.twnic.net.tw

edu.tw---教育部

gov.tw---研考會

http://rs.gsn.gov.tw

com，org，net---verisign的registrars

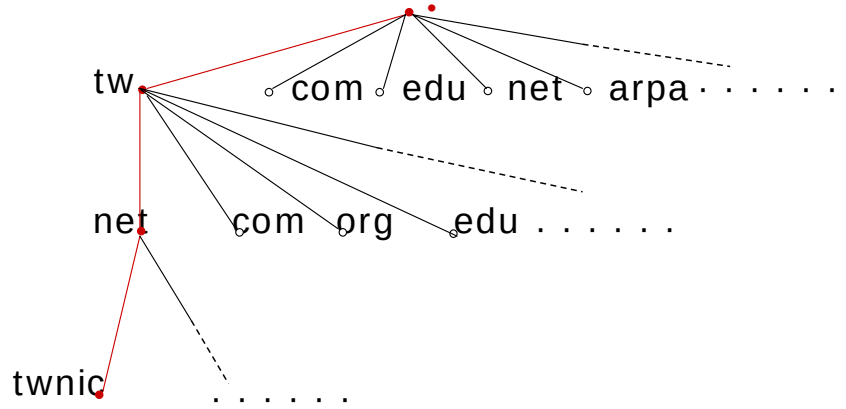
http://www.internic.net/alpha.html

反解

核發IP的ISP

正解之原理

- 正解 twnic.net.tw.



DNS是一個分層級的分散式名稱對應系統，在最頂端的是一個“root”，接下來是TLD(Top Level Domain)，TLD又分爲gTLD(generic TLD)如“.com”，”.org”，”.net”，”.edu”，”.gov”，”.mil”，”int”，”.arpa”及ccTLD(country code TLD)如“.tw”(台灣)，“.jp”(日本)，“.uk”(英國)…(ISO-3166所定義的2個byte國碼)

目前“.com”，”.net”由 verisign 所經營，”.edu”，”.gov”，”.mil”分別爲美國的教育單位、政府單位、軍事單位，”.int”爲一些國際間的需求(如internet fax)使用，“.arpa”原本爲arpanet(internet的前身)單位所使用，現爲DNS反解等使用.

ICANN在新增了七個 gTLD:

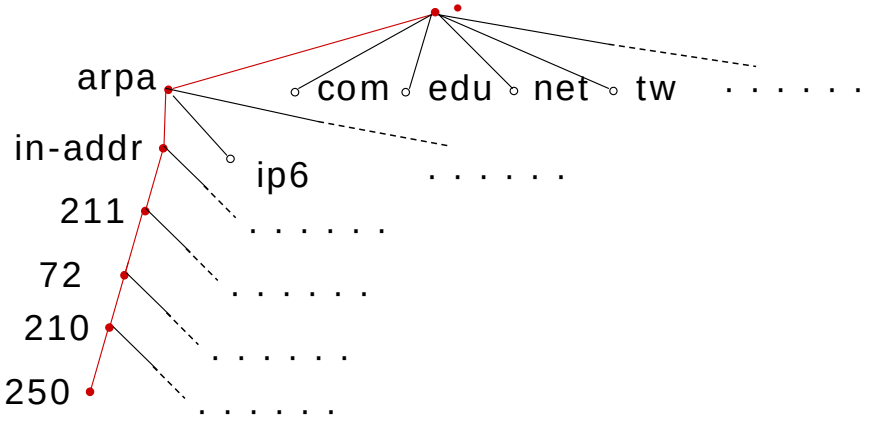
aero， .biz， .coop， .info， .museum， .name， .pro

第二層網域名稱SLD(Second Level Domain)在gTLD爲單位名稱，如ibm(ibm.com)，fitpi(fitpi.com)等，在ccTLD依各國管理政策有下列三種情況:

1. 單位名稱，如: ibm.uk
2. 屬性名稱，如: com.tw， net.tw， ne.jp
3. 地理名稱，如: tokyo.jp

反解之原理

- 反解 250.210.72.211.in-addr.arpa.



55

由網域名稱查IP的行爲稱爲正解，由IP查網域名稱的行爲稱爲反解，電腦會作反解的行爲主要有下列兩個目的：

1. 讓使用者或管理者容易了解連線情形，如在畫面或Log中顯示網域名稱會比顯示IP讓人更容易了解連線對象是那個單位

2. 安全考量，如某些server會以連線對象的IP查網域名稱，再查網域名稱對應的IP是否一致

根據我們的調查，國內 IP 的反解比率約四成，而中國大陸不到 5%，造就國內高反解比率(與國際相較)主要因爲 TANET 早期的推動及近來 ISP 的處理方式，而國際上 Internet 有所謂的主機數量調查，台灣排名約在十二名左右(不約 hinet.net 僅以 .tw 結算).

反解之原理(以IPv6為例)

- IPv6
  - 共 16 bytes，以十六進制表示，每兩個 bytes 為一組，每組間以 ‘:’ 相隔

2001:288:1:1002:2e0:18ff:fe77:f174

舊用法:

4.7.1.f.7.7.e.f.f.f.8.1.0.e.2.0.2.
0.0.1.1.0.0.0.8.8.2.0.1.0.0.2.ip6.int.

新用法:

4.7.1.f.7.7.e.f.f.f.8.1.0.e.2.0.2.
0.0.1.1.0.0.0.8.8.2.0.1.0.0.2.ip6.arpa.

每一欄位開頭的 0 可省略

Ex: 2001:238:882:0:248:54ff:fe53:d3ee

:: 可代表連續多個 0 的欄位(僅可使用一次)

2001:238:F882:0:0:0:0:1 = 2001:238:F882::1

0:0:0:0:0:0:0:1 = ::1

0:0:0:0:0:0:0:0 = ::

2001:0:0:345:0:0:0:1 = 2001:0:0:345::1

由於舊設法與新設法不同，因此目前一般兩種都會設，主要因爲過渡時期，

目前 IPv6 在國內尚有等發展，但或許某一天則爲勢在必行之狀況. IPv6 的反解設定在此處來看是非常長的，故 BIND 在 9.X 版本中則另發展了 Binary Format 的設定方法(見於第二天課程)，有效免除輸入錯誤的困擾

TWNIC 域名的申請與DNS指定

- 各類網域名稱申請資格
- 如何指定 DNS 伺服器
- DNS 指定上常見的錯誤

目前.tw的網域名稱註冊由TWNIC受理註冊業務，TWNIC授權多家代理註冊機構受理註冊事宜(http://rs.twnic.net.tw)

各類網域名稱申請資格(1)

- .com.tw
  依公司法登記之公司或依商業登記法登記之商號；外國公司依其本國法設立登記者，亦同.
- .net.tw
  具第一類電信事業特許執照或網路建(架)設許可證或第二類電信事業許可執照者.
- .org.tw
  依法登記之財團法人或社團法人； 外國非營利組織依其本國法設立登記者，亦同.
- .tw
  在中華民國立案之公司行號、組織機構或具中華民國國籍之國民均得申請.

58

TWNIC的registrars:

一、協志聯合科技股份有限公司(http://reg.tisnet.net.tw)
二、亞太線上服務股份有限公司(http://rs.apol.com.tw/)
三、中華電信數據通信分公司(http://nweb.hinet.net)
四、網路中文資訊股份有限公司(http://www.net-chinese.com.tw)
五、網路家庭資訊服務股份有限公司(http://myname.pchome.com.tw)
六、數位聯合電信股份有限公司(http://rs.seed.net.tw)
七、台灣固網股份有限公司(http://www.anet.net.tw)
八、台灣網路資訊中心(http://rs.twnic.net.tw)
九、Yahoo! 奇摩(http://tw.domain.yahoo.com/)
十、台灣電訊 (http://www.ttn.net.tw)

目前此類網域名稱皆可免文件審查，但細節您可參照 http://www.myhome.net.tw/2004_03/twnic_news/main1.htm 處的說明

各類網域名稱申請資格(2)

- .game.tw

  不限制申請人資格，註冊人可自行依其需求擇定屬性，惟需利用電子郵件方式確認身分.

- .ebiz.tw

  不限制申請人資格，註冊人可自行依其需求擇定屬性，惟需利用電子郵件方式確認身分.

- .club.tw

  不限制申請人資格，註冊人可自行依其需求擇定屬性，惟需利用電子郵件方式確認身分.

- .idv.tw

  凡自然人均可申請，惟需利用電子郵件方式確認身分.

game.tw 由數位聯合電信股份有限公司(http://rs.seed.net.tw) 獨家受理註冊

ebiz.tw及club.tw由台灣固網股份有限公司(http://www.anet.net.tw) 獨家受理註冊

TWNIC 域名的申請與DNS指定

- 各類網域名稱申請資格
- ✓如何指定 DNS 伺服器
- DNS 指定上常見的錯誤

目前.tw的網域名稱註冊由TWNIC受理註冊業務，TWNIC授權多家代理註冊機構受理註冊事宜(http://rs.twnic.net.tw)

TWNIC DNS設定模式介紹

- DNS 設定代管
  - DNS 模式
    - 即為一般的 DNS 指定，使用者需在其指定之 IP 上架設 DNS 伺服器，提供名稱解析
  - 代管模式(主機模式)
    - 使用者不需自設 DNS 伺服器，所指定之主機資料將由代管主機回應結果

61

設定爲DNS模式，使用者必須要自行架設DNS Server或者另外找DNS代管的服務才能正常運作

爲服務許多小型單位以及許多不了解DNS的使用者，TWNIC也提供了主機模式(即DNS代管)，惟僅限於原在 TWNIC 申請註冊之用戶免費服務.

TWNIC DNS相關服務介紹

- 動態 DNS
  - 提供下載 Client
  - 固定時間與伺服器間更新 IP 資訊
  - 可能受限於 FireWall 及 NAT 環境
  - 僅提供 www.DN 及 DN 之 A（Address）記錄
  - 適合沒有固定IP的主機使用
- 網頁轉址
  - 以 frameset 的方式將您的網頁轉至網頁空間
  - 如 http://www.xxx.com.tw ，其實看到的是 pchome 的免費網頁空間

動態DNS: http://rs.twnic.net.tw/dyndn_intro.html

不固定IP的主機(如使用 PPPOE的ADSL，DHCP的Cable，撥接用戶)，欲架設Web、Mail或者FTP 等Server，或者使用者需要網路身份(網域名稱)者

網頁轉址

將您的 www.DN 轉向一般網頁目錄(如 pchome， yahoo 等)，讓人感覺起來是一個獨主的 WWW Server

TWNIC 域名的申請與DNS指定

- 各類網域名稱申請資格
- 如何指定 DNS 伺服器
- ✓DNS 指定上常見的錯誤

目前.tw的網域名稱註冊由TWNIC受理註冊業務，TWNIC授權多家代理註冊機構受理註冊事宜(http://rs.twnic.net.tw)

常見之錯誤(1)

- DNS 指定 : DNS 模式

  指定了 mail.xxx.com.tw 及 www.xxx.com.tw 為 NameServer (NS)，但實際上並沒有架設 DNS Server

  - 問題

    這樣的設定是不正確的，上層的DNS已經將網域委任出去，但是下層卻沒有可以回應的機器，會造成整個網域找不到的現象

  - 正確做法

    需自設 DNS 伺服器或改用代管模式

64

若指定了name server但實際name server 該主機又沒有該域名之 DNS 在運行，會造成一些問題，在新版的DNS server(如BIND 9)有可能造成無法解析的結果

有部分的狀況會可以查得到，例如Hinet的DNS Server (dns.hinet.net)

原因是該DNS Server 的版本較舊會直接對A 紀錄作回應 所以會查得到但在其他較新的DNS Server上 則會出現查不到的現象.因爲國內有很多人將 DNS 指向 hinet， 故問題可能較不明顯，以 TWNIC 常接到的問題都是國內信件會通，但國外有些地方不通，有些原因即是如此所造成的，因爲 DNS 授權出去而沒有對應所照成所致.

(BIND 8 與 9 運作最不同點在於 BIND 9 會使用該 zone 所提供之 NS 記錄，而 BIND8 在這種狀況下會以 .com.tw. 中所提供的 A 記錄，因爲如此，所以當對方使用 BIND 9 的版本時，即會有此類之問題)

實例請參考 http://phorum.study-area.org/viewtopic.php?t=9718

常見的錯誤(2)

- DNS 指定 : DNS 模式

  將 DNS 指定到別人的 Server，如 HINET/SEEDNET

  - 問題

    會發生 Lame Server 的狀況，該 ISP 並不管理您的域名資料，這是一種不良的委任關係，發生原因多是因為使用者需指定兩部以上之主機，而其第二部不知要指向何處而為之，造成DNS的解析發生錯誤

  - 解決方法

    確認該 ISP 同意代管您的域名或為您的 Slave 主機，或是提供兩部以上主機之要求

DNS Server的指定， 一般會建議兩部以上的原因主要是系統備援的概念若是只設一部DNS Server，當這部機器出問題時，或者網路有問題時，將會影響到整個網域都看不到的狀況，但若因此指向不管您的 Domain 的主機，徒增不必要的 traffic， 及查詢時間，舉例來說，若設兩部 DNS 主機，而第二部指向不管您的 Domain 的 HINET， 則當有人寄信給您時，他將有 50% 的機會第一次寄不到而存在佇列(queue) 裏，一般的 Mail Server 多預設佇列資料 4 小時重送，此時仍是隨機二選一情況，故若造成 Lame Server (即這種狀況)， 反而造成您的損失(如訂單晚到…)，兩部的 DNS 要求對您只要好處，雖然增加些許成本，但更能穩定您的 DNS 服務，若此兩部位於不同的線路或 Subnet 上，則更能避免服務的失效.

DNS設定介紹

- 用 BIND 還是用 Windows DNS ?
- BIND 版本差異
- BIND 工具程式
- 設定檔 named.conf
- 根伺服器介紹與設定
- zone file基本設定
- 資源紀錄（SOA，NS，A，CNAME，MX，PTR…）
- 正解/反解設定
- 主要（master）/次要（slave）伺服器的關係容錯及負載平衡功能（Round Robin）
- named之參數說明及啟動與停止
- WINDOWS DNS 設定介紹
- 使用 nslookup/dig 自我檢測

本章節由淺入深教您如何將您的DNS設定完成，當您都了解了DNS的原理後，不論是設定UNIX BIND或Windows DNS，都會是相當容易的一件事從安裝、設定以及資源紀錄的解釋到Zone File帶您一步一步的了解DNS的設定流程，最後教您如何驗證DNS的正確性

用 BIND 還是用 Windows DNS ?

	Windows	BIND
操作	●GUI的設定方式入門容易 ●可用 Windows 其他服務結合 (WINS/AD)	●設定以文字編輯進行 ●較易出錯 ●Unix 環境爲一般人所不熟悉
效率	●查詢數字無統計資料	●每秒可處理上萬次查詢 ●Multi-thread/SSL
穩定性	●視 OS 表現 ●基本上可符合一般企業需求	●穩定性佳 ●版本更新速度較快
安全性	●隨系統版本更新而更新版本	●可從設定面加強安全性 ●較能預防DNS Spoofing
佔有率	●在台灣兩者相當	●在全世界佔大宗
其他		●Root Server 皆以 BIND 爲主

Windows之Microsoft DNS Server由舊版之BIND4改版而成，因此有些較新之功能如NAPTR等皆無法支援

BIND 版本差異

版本	8.X	9.X
編譯	傳統的 Makefile	較人性的 autoconf
工具程式	查詢，診斷，動態更新，轄區傳送	DNSSEC，語法檢查，診斷，動態更新，(預計捨棄 nslookup 而用 dig)
執行模式	單一執行程式	多緒(mutli-thread) 多 CPU 的支援
除錯	僅能從記錄檔分析	提供工具做語法檢查
IPv6	簡單的支援 (AAAA)	多了 A6 offset 的功能
反解	只有一種方法	共三種方法且簡單，對 IPv6 的特性更容易處理
資料庫	無	可支援多種不同的資料庫
資料格式	僅 7bit 資料	可使用 8bit 資料
資源紀錄		新增 NAPTR, A6, DNAME, TKEY

BIND 8.x/9.x 版本的重大差異

新的功能

DNSSEC

IPv6 support

Domain alias(DNAME)

多國語文(8 bits)

建議使用 BIND 9.x 的理由

- 新架設DNS Server
- 視需求而定
  - 可使用自動變數簡化管理
  - BIND8已經是一個維護的版本不會再有新的功能加入
  - 是否使用 IPv6
  - 是否使用新的資源紀錄(NAPTR，A6，DNAME)
  - 對安全的要求等級(DNSSEC，TSIG)

以安全性及擴充性之考量應昇級至BIND9， BIND8已經是一個維護的版本不會再有新的功能加入,一些新的作業系統其內建的DNS軟體也都是BIND9,而BIND 9 的資料結構不再是以往的hash， 而是全新改寫的資料庫架構(Binary Tree)

DNS設定介紹

- 用 BIND 還是用 Windows DNS ?
- BIND 版本差異
- ✓BIND 工具程式
- 設定檔 named.conf
- 根伺服器介紹與設定
- zone file基本設定
- 資源紀錄（SOA，NS，A，CNAME，MX，PTR…）
- 正解/反解設定
- 主要（master）/次要（slave）伺服器的關係容錯及負載平衡功能（Round Robin）
- named之參數說明及啟動與停止
- WINDOWS DNS 設定介紹
- 使用 nslookup/dig 自我檢測

本章節由淺入深教您如何將您的DNS設定完成，當您都了解了DNS的原理後，不論是設定UNIX BIND或Windows DNS，都會是相當容易的一件事

從安裝、設定以及資源紀錄的解釋到Zone File帶您一步一步的了解DNS的設定流程，最後教您如何驗證DNS的正確性

BIND 的工具程式

- named　DNS 伺服器的服務程式，查詢服務
- h2n　將 /etc/hosts 轉成 bind 的 正/反 解檔
- named-xfer　轄區傳送的工具程式
- ndc　named 的　啟動/停止　程式
- dig　DNS 查詢的工具程式
- nsupdate　動態更新程式(請參考 Oreilly DNS 一書)
- nslookup　DNS 查詢程式
- dns*　另有許多 dns 開頭之工具，主要用於產生用於認證之 KEY，如 TSIG/DNSSEC 等
- named-checkconf　檢查 named.conf 語法的工具
- named-checkzone　檢查 zone file 語法的工具
- named-bootconf　將 4.X 的 named.boot 轉成 8.X named.conf 檔

Bind 相關的重要執行檔 (課程因素僅列表了解)

/usr/sbin/named	DNS 伺服器的服務程式，查詢服務
/usr/sbin/dnskeygen	產生 TSIG key ，供 named 做授權認證
/usr/sbin/h2n	將 /etc/hosts 轉成 bind 的 正/反 解檔
/usr/sbin/named-bootconf	將 4.X 的 named.boot 轉成 8.X named.conf 檔
/usr/sbin/named-xfer	轄區傳送的工具程式
/usr/sbin/ndc	named 的　啓動/停止　程式
/usr/bin/dig	DNS 查詢的工具程式
/usr/bin/nsupdate	動態更新程式
/usr/bin/nslookup	DNS 查詢程式

Bind 相關的設定檔

/etc/named.boot	4.X 版的設定檔
/etc/named.conf	8.X 版的設定檔
/var/named/named.ca	Root Server 的所在
/var/named/*	正反解檔等，一般常放的位置

DNS設定介紹

- 用 BIND 還是用 Windows DNS ?
- BIND 版本差異
- BIND 工具程式
- ✓ 設定檔 named.conf
- 根伺服器介紹與設定
- zone file基本設定
- 資源紀錄（SOA，NS，A，CNAME，MX，PTR…）
- 正解/反解設定
- 主要（master）/次要（slave）伺服器的關係容錯及負載平衡功能（Round Robin）
- named之參數說明及啟動與停止
- WINDOWS DNS 設定介紹
- 使用 nslookup/dig 自我檢測

本章節由淺入深教您如何將您的DNS設定完成，當您都了解了DNS的原理後，不論是設定UNIX BIND或Windows DNS，都會是相當容易的一件事從安裝、設定以及資源紀錄的解釋到Zone File帶您一步一步的了解DNS的設定流程，最後教您如何驗證DNS的正確性

設定檔：named.conf

- 路徑
  - /etc/named.conf 或 /usr/local/etc/named.conf
- BIND（named）環境之主要設定檔
- 作用
  - 定義 named 的功能項目 （ options ）
  - 定義 root server 位置 （ zone ）
  - 定義所管轄之網域名稱 （ zone ）
  - 定義反解 （ zone ）
  - 其他，如系統記錄/存取控制列表等…

舊版(BIND4)與新版(BIND8/9)的檔案名稱與格式不同，但BIND8有提供轉換的程式將BIND4的named.boot轉換成named.conf

named.conf: options

```
#/etc/named.conf
options {
directory "/var/named";
pid-file “/var/named/named.pid”;
allow-transfer     { 211.72.211.71/32;211.72.210/24;};
};
```

directory　　zone file 檔案存放位置（預設為 /etc）

pid-file　　DNS 啟動時記錄行程代號(PID)之檔案

allow-transfer　　轄區傳送之設定，定義那些 IP 可與此部 DNS 做 AXFR（未定義則全開，形同 DNS 資料外流）

- 常犯錯誤
  - options 忘了加 “s”，前後以 {} 括住
  - 有關檔案或路徑名稱皆要加 “” 號
  - 每一行的結尾需有 “;” 號
  - 有關 IP 等設定項目亦需加 ; 號
  - pid-file 所指路徑的權限問題要注意

Options 請注意有加 s

Named 相關檔案置於 /var/named 目錄下

允許來自 211.72.211.71 及 211.72.210.0~255的 AXFR 要求

注意: 每個描述要加 ; 號區隔

上例之 /32 或 /24 爲 CIDR 表示法， /32 代表 netmask 有 32 bit 爲 1，即 255.255.255.255，固代表爲一 IP ， /24 爲 255.255.255.0 ， 表示爲一個 network， 該 IP (/32) 或該 Network(/24) 皆可以向本機要求 Zone Transfer.

DNS設定介紹

- 用 BIND 還是用 Windows DNS ?
- BIND 版本差異
- BIND 工具程式
- 設定檔 named.conf
- ✓ 根伺服器介紹與設定
- zone file基本設定
- 資源紀錄（SOA，NS，A，CNAME，MX，PTR…）
- 正解/反解設定
- 主要（master）/次要（slave）伺服器的關係容錯及負載平衡功能（Round Robin）
- named之參數說明及啟動與停止
- WINDOWS DNS 設定介紹
- 使用 nslookup/dig 自我檢測

75

本章節由淺入深教您如何將您的DNS設定完成，當您都了解了DNS的原理後，不論是設定UNIX BIND或Windows DNS，都會是相當容易的一件事從安裝、設定以及資源紀錄的解釋到Zone File帶您一步一步的了解DNS的設定流程，最後教您如何驗證DNS的正確性

在named.conf設定根伺服器

```
zone “.” {
     type hint;
     file “named.cache”;
};
```

- 所有的 DNS 伺服器皆需要知道根伺服器位置
- 根伺服器為所有 DNS 查詢之起源
- 根伺服器列表可由 ftp://ftp.internic.net/domain/named.cache 取得
- hint 字面為暗示之意，即向 DNS 表示如果你沒有資料，可以到根伺服器詢問
- 一部 DNS 僅能有一 hint type

定義 root dns Server 的檔案， 可由 ftp://ftp.internic.net 取得或執行 dig @a.root-servers.net . ns > /var/named/named.cache

在named.conf設定正解網域

```
zone "xxx.com.tw" {
    type master;
    file "xxx.com.tw.hosts";
    allow-transfer { 168.95.1.1;168.95.192.1;};
    };
```

此一域名需上層授權(com.tw) 後別人方可查得到
若您有多個網域名稱即添加類似設定即可
此例為 master 主機設定，slave 主機設定於後述
file 未定路徑即表參照 directory 參數
注意 ; 號問題
此處的 allow-transfer 僅對此網域有效，而 options 中的 allow-transfer 則對此部 DNS 有效

Zone 項下可用的選項功能有:

allow-notify 表示當此 **Zone** 被更新時要通知何部 **DNS** 主機

allow-query 允許那個 **IP** 來查，預設 **any**

allow-transfer 預設 **any**

allow-update 預設 **none**

check-names 語法較負雜，本處不介紹，參考第二天課程

max-transfer-time-in 要求 **Zone Transfer** 時，需在多少時間內完成

max-transfer-idle-in **Zone Transfer Idle** 的時間(秒數)

max-transfer-time-out 同上述，秒數值

max-transfer-idle-out

Notify **Zone File** 更新時要不要主動通知

zone-statistics 這個 **Zone** 的統計要不要做 **(default no)**

statistics-file 統計檔名爲何

其他尚有十幾個參數，尤於甚少用到，故本處不再描述

在named.conf設定反解網域

```
zone  "0.0.127.in-addr.apra"  {
    type master;
    file  "named.local";
};
zone  "210.72.211.in-addr.arpa"  {
    type master;
    file  "211.72.210.rev";
};
```

DNS 反解搜尋由後往前(後序)，故原 127.0.0.x 及 211.72.210.x 之 IP 要反寫

in-addr 為 Internet Address 之意，用於 IPv4 之反解，IPv6 則使用 ip6

apra 為反解之起源

78

127.0.0.1 乃是系統用的 localhost

所以DNS Server 中必須要加入這個反解

IPv6之反解舊的RFC使用ip6.int，新的RFC已經改爲ip6.arpa作爲反解的起源，建議使用者設定時兩種都設，因爲有些舊版的DNS仍然會到ip6.int去尋找

反解名稱爲 IP 反對來寫，您可根據前述之 DNS Tree 結構來看查詢，故要反過來寫，查詢才找得到

named.conf 完整內容範例

```
#/etc/named.conf
options {
    directory "/var/named";
    pid-file "/var/named/named.pid";
    allow-transfer    { 211.72.211.71/32;211.72.210/24;};
    };
zone "." {
    type hint;
    file "named.cache";
};
zone "xxx.com.tw" {
    type master;
    file "xxx.com.tw.hosts";
    allow-transfer { 168.95.1.1;168.95.192.1;};
    };
zone "0.0.127.in-addr，apra" {
    type master;
    file "named.local";
};
zone "210.72.211.in-addr.arpa" {
    type master;
    file "211.72.210.rev";
};
```


大致分爲

Options

Root zone

Localhost zone

正解 zone

反解 zone

named.conf 回顧

- options/root server /正解/反解 為基本設定
- 注意 Zone Transfer 問題
- 若欲為 Cache-Only 主機則可拿掉 正解/反解設定即可（即保留 options/root/localhost）
- 語法及 ; {} “” 等問題需注意（初學常犯）
- 可使用工具程式 named-checkconf，named-checkzone 幫您做語法檢查

用法: named-checkconf [-v] [-t directory] filename

named-checkzone [-d] [-j] [-q] [-v] [-c class] zonename filename

DNS設定介紹

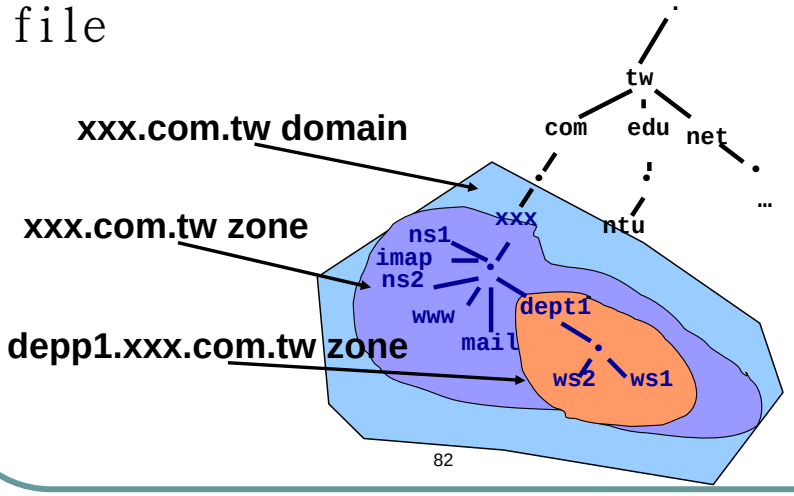
- 用 BIND 還是用 Windows DNS ?
- BIND 版本差異
- BIND 工具程式
- 設定檔 named.conf
- 根伺服器介紹與設定
- ✓ zone file基本設定
- 資源紀錄（SOA，NS，A，CNAME，MX，PTR…）
- 正解/反解設定
- 主要（master）/次要（slave）伺服器的關係容錯及負載平衡功能（Round Robin）
- named之參數說明及啟動與停止
- WINDOWS DNS 設定介紹
- 使用 nslookup/dig 自我檢測

本章節由淺入深教您如何將您的DNS設定完成，當您都了解了DNS的原理後，不論是設定UNIX BIND或Windows DNS，都會是相當容易的一件事

從安裝、設定以及資源紀錄的解釋到Zone File帶您一步一步的了解DNS的設定流程，最後教您如何驗證DNS的正確性

Zone file: 正解檔

- 正解檔是 DNS 中最重要的檔案
- 如下面的例，我們將其轉換成 DNS Zone file

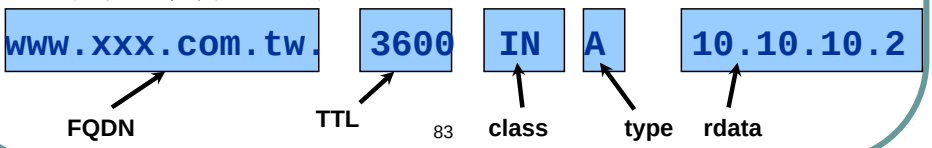


注意網域(domain)及轄區(zone)的不同

一個domain即代表DNS樹狀結構下某個節點下面的全部節點,而轄區只代表該名稱伺服器所管理的區域，通常是該網域的一部份而已. 想像一下，若root server管理整個root網域，而不是root轄區，就可以了解網域與轄區的不同

正解：什麼是資源記錄

- 資源記錄(RR， Resource Record)
  - 名稱(FQDN)
  - 快取時間 (TTL，Time to Live)
  - 網路類別(class)，
  - 資料類型(type)
  - 答案(rdata)
- TTL 是此一筆資料被別的 DNS Cache 的時間值
- IN 即是 Internet
- 資料類型分許多種
- 下列為一筆資源紀錄的內容



TTL與Class是可省略的

FQDN或是RDATA的主機名稱若最後加上‘.’表示爲完整的FQDN， 無需再加上該zone的domain name

正解:資源記錄範例

```
xxx.com.tw.  38400 IN SOA ns1.xxx.com.tw.
  abelyang.twnic.net.tw. (
    2001061501          ; Serial
    43200           ; Refresh 12 hours
    14400           ; Retry 4 hours
    345600          ; Expire 4 days
    7200          ; Negative cache 2 hours
    )
xxx.com.tw.          86400    IN    NS    ns1.xxx.com.tw.
xxx.com.tw.          86400    IN    NS    ns2.xxx.com.tw.
ns1.xxxcom.tw.      86400    IN    A     211.72.211.1
ns2.xxx.com.tw.      86400    IN    A     211.72.211.2
；以下略
```

- 資源記錄的 TYPE 有許多不同類型

常見的TYPE有:

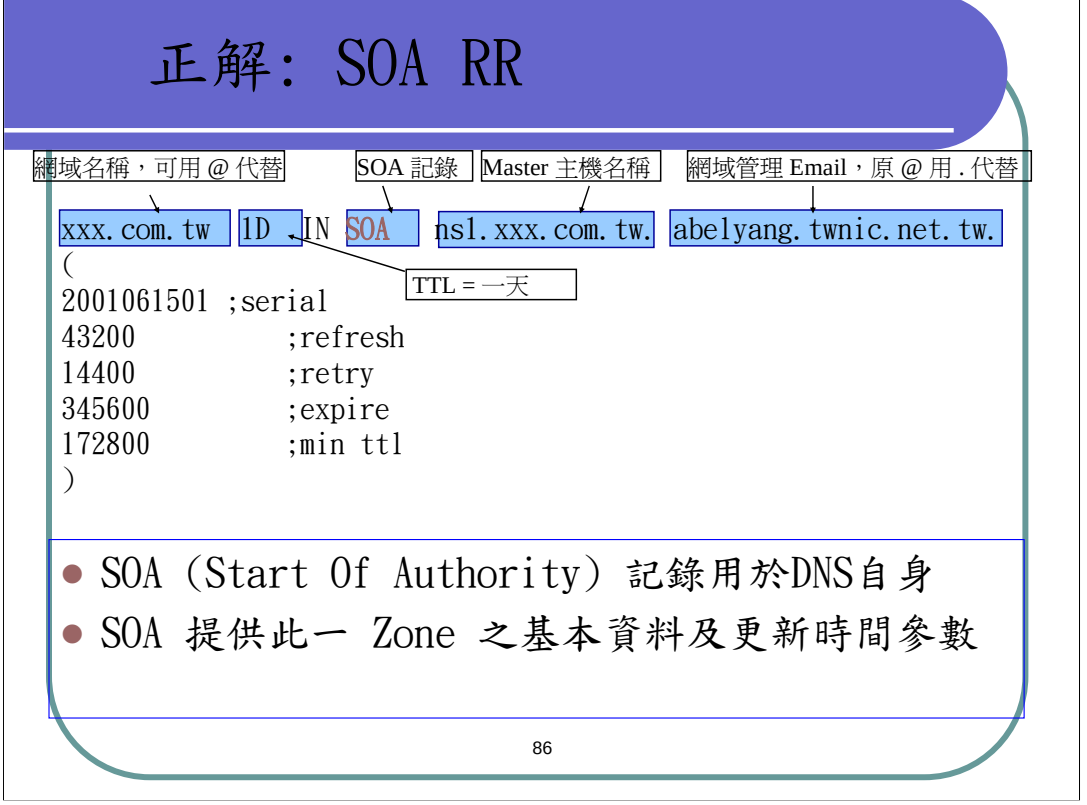
A	IPv4 Address
AAAA，A6	IPv6 Address
CNAME	別名
NS	Name Server
MX	mail exchanger
PTR	Pointer (反解使用)
SOA	Start of Authority

DNS設定介紹

- 用 BIND 還是用 Windows DNS ?
- BIND 版本差異
- BIND 工具程式
- 設定檔 named.conf
- 根伺服器介紹與設定
- zone file基本設定
- ✓ 資源紀錄（SOA，NS，A，CNAME，MX，PTR…）
- 正解/反解設定
- 主要（master）/次要（slave）伺服器的關係容錯及負載平衡功能（Round Robin）
- named之參數說明及啟動與停止
- WINDOWS DNS 設定介紹
- 使用 nslookup/dig 自我檢測

本章節由淺入深教您如何將您的DNS設定完成，當您都了解了DNS的原理後，不論是設定UNIX BIND或Windows DNS，都會是相當容易的一件事

從安裝、設定以及資源紀錄的解釋到Zone File帶您一步一步的了解DNS的設定流程，最後教您如何驗證DNS的正確性



每一個zone都必需要有 SOA 這個RR

;爲註解

2001061501 ;serial序號，用於 AXFR 時，Master 資料是否變動判斷，修改時至少要將序號加一，讓slave server會至master作同步的動作

43200 ;refresh，請 Slave 主機每隔此一時間向 Master AXFR

14400 ;retry，更新失敗，則每隔此一時間再做 AXFR

345600 ;expire，Slave 主機的資料超過此值則放棄此 Zone

172800 ;min ttl，在此 refresh + N x retry 小於此一時間內此會

;做 AXFR，以確保資料的正確性與同步，超過此時間，Slave

;即不會再和 Master 做 AXFR 請求，需重啓 DNS

在 Zone File 中 @ 符號表示現在這個 Zone 的名稱，故原來 Email 的 @ 要以 . 代替,若原來 Email 中有 . ，則需以 \. 表示.

TTL 一般建議值多爲 半天(38400)或一天(86400)，如果您的 DNS 資料很穩定則可設得更高些

SOA 資料中的時間參數在於同步 Master/ Slave 間的 Zone file 資料一致,每次更改了 Zone File 的任何內容，請務必加大序號值，讓 AXFR 能順利進行,相關的時間建議值可參考 RFC 1912 說明

正解: NS/A RR

```
xxx.com.tw.    IN   NS   ns1.xxx.com.tw.
xxx.com.tw.    IN   NS   ns2.xxx.com.tw.
ns1.xxx.com.tw.   IN   A   211.72.211.1
ns2.xxx.com.tw.   IN   A   211.72.211.2
```

- NS（Name Server）用於 DNS 的搜尋
- 每個 Zone File 如何 SOA 一般，皆要有 NS RR，且接於 SOA 之後NS 記錄之 RDATA 若屬同一個zone以內者需接一 A（Address）RR，以標明其 IP Address
- NS 記錄說明了那些主機管理此一網域名稱（權威主機），需與上層（如 TWNIC）的指定一致
- NS 記錄之 RDATA 需接一 FQDN 記錄，不可用 IP ，也不可接到一 CNAME 記錄（ RFC 規範）
- NS 記錄的取用順序是隨機決定的，而非取用第一筆
- A 記錄為指出某一 FQDN 其 IP 為何

Ex:

xxx.com.tw. IN NS ns1.yyy.com.tw.

xxx.com.tw. IN NS ns2.yyy.com.tw.

則不須加 A 紀錄，因爲 yyy.com.tw 這個網域根本不屬這個轄區所管， 加了也沒用

FQDN1 IN NS FQDN2

FQDN2 不得直接寫爲 IP

FQDN2 IN A xxx.xxx.xxx.xxx

FQDN2 不得指定爲 CNAME (如 FQDN2 IN CNAME FQDN3)，在新版的 BIND會造成Error

上述之 “與 TWNIC 指定一致” 說明需熟記，因爲不一致會造成許多問題 (不致於失敗，但可能浪 Traffic 或時間)

正解: CNAME RR

www.xxx.com.tw. 3600 IN A 211.72.211.80
ftp.xxx.com.tw. 3600 IN CNAME www.xxx.com.tw.

- CNAME 用於機器別名，如查詢 FTP ，則會查到 WWW 位址
- 建議使用 A 記錄來替 CNAME，以避免 NS/MX 等出現問題
- CNAME Chain 問題，雖沒有禁止使用，但會導致效率變差甚至錯誤

NS/MX 避免使用 CNAME 紀錄 (CNAME and OTHERS Error)

Ex:

xxx.com.tw. IN NS ns.xxx.com.tw.

ns.xxx.com.tw. IN CNAME www.xxx.com.tw.

www.xxx.com.tw. IN A 1.2.3.4

這種東西就應該避免，會造成錯誤

這種東西就叫CNAME Chain， 應避免

aaa.xxx.com.tw. IN CNAME bbb.xxx.com.tw.

bbb.xxx.com.tw. IN CNAME ccc.xxx.com.tw.

ccc.xxx.com.tw. IN A 1.1.1.1

正解: MX RR

```
@       IN   MX   10   mail
@       IN   MX   20   imap
mail    IN   A        211.72.211.25
imap    IN   A        211.72.211.143
; 此時不可用如 mail IN CNAME www 語法
```

- MX (Mail eXchange) 記錄為 SMTP 服務所使用，其中的 10，20 表示郵件交換時的優先順序(數字小者優先)
- mail 或 imap 亦需接一 A 記錄，而不可使用 CNAME，這是FAQ最常見的問題
- 亦可使用 A 記錄來代 MX使用 (即 DN=FQDN)，但如此僅能使用一部機器當 Mail Server

```
@    IN    A    211.72.211.25
```


MX紀錄指到一個CNAME紀錄，這是Sendmail使用者最常出現的問題，須特別注意, 不同的 MX 記錄郵件如何同步視 Mail Server 性質而定， Mail Server 必須要作一些調整才能夠配合 MX. 不同的 MX 意義在優先權最高(數字最小)之主機失效時，次之 MX 主機能暫時將 Mail 收下 (是收在 queue 下，而不是收在信箱)，並根據 Mail Server 所定義之重試(Retry) 時間，試著將 Mail 轉至 MX 最小之主機，此爲 MX 運作之主要精神所在， MX 做法如下:

$ORIGIN xxx.com.tw.

@ IN MX 10 mail.xxx.com.tw.

@ IN MX 20 redundant.xxx.com.tw.

a IN A 211.72.211.4

b IN A 211.72.211.5

接下來在 mail.xxx.com.tw 機器上面做如下動作:

echo “xxx.com.tw” >> /etc/mail/local-host-names

echo "mail.xxx.com.tw" >> /etc/mail/local-host-names ;service sendmail restart

再於 redundant.xxx.com.tw 機器上面做如下動作:

echo "redundant.xxx.com.tw" >> /etc/mail/local-host-names

echo "xxx.com.tw RELAY" >> /etc/mail/access

echo "mail.xxx.com.tw RELAY" >> /etc/mail/access

makemap hash /etc/mail/access.db < /etc/mail/access ;service sendmail

DNS設定介紹

- 用 BIND 還是用 Windows DNS ?
- BIND 版本差異
- BIND 工具程式
- 設定檔 named.conf
- 根伺服器介紹與設定
- zone file基本設定
- 資源紀錄（SOA，NS，A，CNAME，MX，PTR…）
- ✓ 正解/反解設定
- 主要（master）/次要（slave）伺服器的關係容錯及負載平衡功能（Round Robin）
- named之參數說明及啟動與停止
- WINDOWS DNS 設定介紹
- 使用 nslookup/dig 自我檢測

本章節由淺入深教您如何將您的DNS設定完成，當您都了解了DNS的原理後，不論是設定UNIX BIND或Windows DNS，都會是相當容易的一件事

從安裝、設定以及資源紀錄的解釋到Zone File帶您一步一步的了解DNS的設定流程，最後教您如何驗證DNS的正確性

正解檔完整內容範例

```
xxx.com.tw.      86400 IN SOA    ns1.xxx.com.tw.
root.xxx.com.tw. (
    2002021301    ; serial
    1D        ; refresh
    1H        ; retry
    1W        ; expiry
    2D         ; min ttl
)
xxx.com.tw.         86400    IN    NS    ns1.xxx.com.tw.
xxx.com.tw.         86400    IN    NS    ns2.xxx.com.tw.
Ns1.xxx.com.tw.     86400    IN    A    211.72.211.1
Ns2.xxx.com.tw.     86400    IN    A    211.72.211.2
www.xxx.com.tw.     86400    IN    A    211.72.211.80
ftp.xxx.com.tw.     86400    IN    CNAME    www.xxx.com.tw.
xxx.com.tw.     86400    IN    MX    10    mail.xxx.com.tw.
xxx.com.tw.     86400    IN    MX    20    imap.xxx.com.tw.
mail.xxx.com.tw.     86400    IN    A    211.72.211.25
imap.xxx.com.tw.     86400    IN    A    211.72.211.143
wk1.dept1.xxx.com.tw.     86400    IN    A    211.72.211.101
wk2.dept1.xxx.com.tw.     86400    IN    A    211.72.211.102
```

- 從上述來看是不是又臭又長呢？

記得每個 FQDN 都要有 “.” 結尾,若沒有加的話,有時查詢時出現 xxx.com.tw.com.tw 這樣的東西,即可能是漏了 “.”

正解：以 $TTL/$ORIGIN 來簡化設定

```
$TTL    86400          ; 預設 TTL 值，原來每筆 RR 之 TTL 值可以此值代替
$ORIGIN    xxx.com.tw.   ; 預設附加字尾

@          IN     SOA    ns1      root (
    2002021301     ; serial
    1D         ; refresh
    1H         ; retry
    1W         ; expiry
    2D          ; min ttl
)
              IN    NS     ns1
              IN    NS     ns2
              IN    NS     dns.hinet.net.
              IN    MX     10     mail
              IN    MX     20     imap
nsl              IN    A     211.72.211.1
ns2           IN    A     211.72.211.2
www              IN    A     211.72.211.80
ftp              IN    CNAME     www
mail          IN    A     211.72.211.25
imap          IN    A     211.72.211.143

$ORIGIN dept1.xxx.com.tw.
ws1           38400     IN    A     211.72.211.111
ws2           38400     IN    A     211.72.211.112
```


$TTL 要放在最上面，若 RR 沒有設 TTL 值，代表使用 $TTL 的值

$ORIGIN 設定之後就可以使用 @ 符號來代表該網域管理上也較爲清楚，容易換名字

正解：子網域的分割

```
;在 xxx.com.tw. 的 Zone File 內
$ORIGIN xxx.com.tw.
             IN    NS    ns1
            IN    NS    ns2

ns1            IN    A    211.72.211.1
ns2          IN    A    211.72.211.2
$ORIGIN    dept1.xxx.com.tw.
             IN    NS    ns1
             IN    NS    ns2
ns1            IN    A    211.72.211.101
ns2            IN    A    211.72.211.102
```

- 上述例子 dept1 要自建一 Sub-domain，以管理自己部門之 DNS 資料，需以 NS 記錄再授權出去
- 原在上頁 Zone File 中的 ws1/ws2 等資料應從 xxx.com.tw. 中移除，避免 DNS 混淆
- 於 ns1.dept1.xxx.com.tw 及 ns2 上再建立 dept1.xxx.com.tw. 的 Zone File，並做好 Master/Slave 之區分
- 可參考實例

  http://phorum.study-area.org/viewtopic.php?t=17969&highlight=glue

相當於

```
;
$ORIGIN xxx.com.tw.
                IN   NS   ns1
                IN   NS   ns2
Ns1             IN   A    211.72.211.1
Ns2             IN   A    211.72.211.2
Dept1           IN   NS   ns1.dept1
                IN   NS   ns2.dept1
ns1.depe1       IN   A    211.72.211.101
ns2.depe1       IN   A    211.72.211.102
```

正解: Master/Slave 如何實現

ns1.xxx.com.tw

```
#/etc/named.conf
#其他略
zone  “xxx.com.tw”  {
      type master;
      file  “xxx.com.tw.hosts”;
      allow-transfer {
          211.72.211.2/32;
      };
};
```

ns2.xxx.com.tw

```
#/etc/named.conf
#其他略
zone “xxx.com.tw” {
      type slave;
      masters { 211.72.211.1;};
      file “xxx.com.tw.hosts”;
      allow-transfer {none;};
};
```

- 一般而言 Slave 主機應不允許 AXFR
- Slave 主機啟動後即會和 Master 同步資料，而後資料的同步即是參考 SOA 資訊
- Master 主機更改了資料請務必記得序號需加大，否則即使時間到了 Slave 不會同步資料

在新版的BIND中可設定 notify 的功能讓 master 在有資料異動時主動通知 slave 來更新資料，其通知對象主要根據該 Zone 內之 NS 主機，因爲不是該 Zone 之 NS 主機並沒有意義，有時我們爲了建立備援的DNS 主機，則會使用 also-notify IP 來特別做不宣告的 NS.

Master 應只允許 slave來作zone transfer

容易出錯的部分:Slave 主機設定和 Master 設定稍有不同注意 masters 中的 s

正解: Master/Slave 的同步問題

- AXFR 只有一個方向，由 Slave 向 Master 發出要求
- 如果 Master 更改了資料，以上述 Zone File 的範例而言，最長要一天後 Slave 才會更新，顯然不夠即時
- 解決方法
  - 降低 Refresh 之值：或可改善但仍會有時間差
  - 使用 Bind 8.x/9.x 之 notify 功能
    - 當 Master 重新啟動時，即會送出 notify 訊息至所有 Zone File 中的 NS RR，告知這些機器進行 AXFR
    - 這個功能在 Bind 9.x 中預設即已啟動，若欲關閉

```
options { ….
    notify    no;
};
```

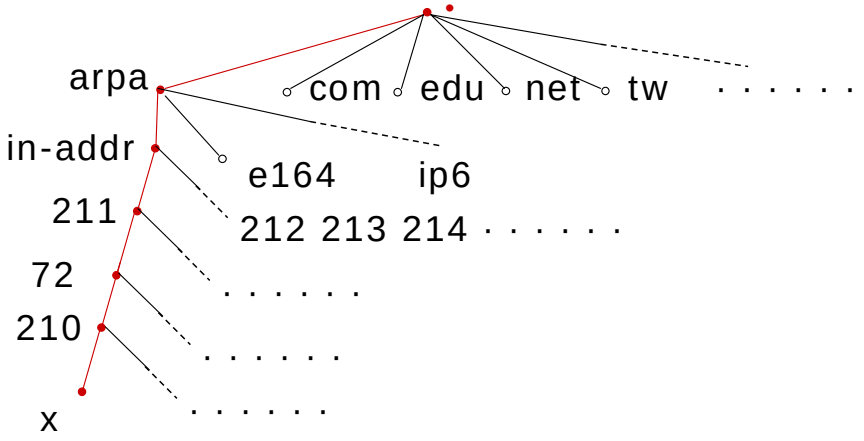

TTL 對 DNS Server的穩定之間有很大的關聯

AXFR 即 Zone Transfer

TTL 值越大表示越不會有變動，若有需要更動 DNS 資料時，建議可以先將 TTL 值改小，以減少修改時資料不同步的時間差，穩定之後再將 TTL 值改回來

Zone File: 反解

- 反解 Subnet 211.72.210.0/24 之反解樹
- Zone 名稱表示為 210.72.211.in-addr.apra



DNS反解

從IP解析出網域名稱

使用PTR紀錄

反解DNS的設定要由核發IP的ISP負責delegation

反解：域名設定

```
Zone  "210.72.211.in-addr.arpa"  {
      type master;
      file  "210.72.211.rev";
};
```

反解依然有 master/slave 主機之分，與正解同理
上例為一 Class C 反解，若為 Class B 則可寫成 72.211.in-addr.arpa.
若不滿一個 Class C，在 DNS 來說則狀況較特殊，於後面補述

反解的zone仍必需有SOA及NS紀錄
DNS也是一樣有分master和slave

反解: Zone File 內容範例

```
$TTL 86400
$ORIGIN     211.72.211.in-addr.apra.
     IN     SOA     ns1.xxx.com.tw.     Root.xxx.com.tw. (
     20030421;
     86400;
     3600;
     864000;
     2D;
)
     IN     NS     ns1.xxx.com.tw.
     IN     NS     ns2.xxx.com.tw.
1     IN     PTR     ns1.xxx.com.tw.
2     IN     PTR     ns2.xxx.com.tw.
3     IN     PTR     pc3.xxx.com.tw.
; 以下類推…..
```

反解之 Zone File 內容與正解類似
反解之 TYPE 為 PTR (Pointer)，指出這個 IP 對應什麼名稱
建議正反解最好一致

對於一些應用程式(如某些mail server)會檢查正反解是否一致， 若不一致可能會被拒絕連線

反解: 利用 $GENERATE 變數簡化

```
;前 SOA 及 NS RR 略
1       IN          PTR     pc1.xxx.com.tw.
2       IN          PTR     pc2.xxx.com.tw.
3       IN          PTR     pc3.xxx.com.tw.
…
31      IN          PTR     pc31.xxx.com.tw.
```

- 上述例子可以 BIND 9.X 之 $GENERATE 功能表示為:

```
$GENERATE 1-31     $     PTR     pc$.xxx.com.tw.
; $ 即表 1-31，將會自動展開成 31 行的 PTR
; BIND 9.X 可省略 IN (Class) 不寫
```

- $GENERATE 亦可用於正解部分，語法相同

$GENERATE 就像是一個迴圈變數，但無法使用巢狀迴圈，通常用於反解或有規則的主機名稱，以簡化設定

反解：不滿一個 Class C 怎麼辦（1）

- 一般而言 ISP 應提供設定功能
  - 如 HINET/SeedNet（僅舉例）
    http://hidomain.hinet.net/rever.html （網頁上直接填寫）
    http://eservice.seed.net.tw/ip_check.htm（以1個 IP 授權）
- 反解授權還是可以小於 Class C
- 如果將 211.72.211/24 切割為 16 個 subnet（即 /28)或更小，分配給 16 家公司，DNS 反解應如何設呢？

100

對不滿一個Class C是使用CNAME與PTR的方式，在後面有詳細介紹

HINET 的做法簡單而容易懂，可直接於其網頁的表格中直接填，而 SEEDNET 則是您填寫後，如果您的 SEEDNET 給您 16 個 IP (210.65.1.0/28)， 則在其反解授權處是一個 IP 一個 NS 的指向，即形成了:

```
$ORGIN 0.65.210.in-addr.arpa.
1   IN NS your_NS
2   IN NS your_NS
……
15   IN NS your_NS
```

或

```
$ORGIN 0.65.210.in-addr.arpa.
$GENERATE 0-15       $    NS    your_NS
```

SEEDNET 做法 USER 若不能充份理解即會造成設定上的問題，可參考 http://phorum.study-area.org/viewtopic.php?t=17795 http://phorum.study-area.org/viewtopic.php?t=12095說明

反解: 不滿一個 Class C 怎麼辦 (2)

```
$ORIGIN 211.72.211.in-addr.arpa.
; 將 16 個子網授權出去:  第一家公司
company1-16     IN      NS      company1_ns1.xxx.com.tw.
company1-16     IN     NS     company1_ns2.xxx.com.tw.
;第二家公司
company2-16     IN      NS      company2_ns1.yyy.com.tw.
company2-16     IN     NS     company2_ns2.yyy.com.tw.
;依此類推 16 家公司,
;以 $GENERATE 的方式,建立 CNAME ,將查詢轉往子網:
$GENERATE 0-15     $     CNAME          $.company1-16
$GENERATE 16-31     $     CNAME          $.company2-16
;依此類推 16 家公司
```

有人查詢 211.72.211.1 之反解時,會查到其 CNAME 至
1.company1-16.211.72.211.in-addr.arpa.
此時 company1 公司的對等反解域名應定義為
zone "company1-16.211.72.211.in-addr.arpa"

詳細資訊請參考: RFC 2317 (Classless IN-ADDR.ARPA delegation)
http://www.ietf.org/rfc/rfc2317

本頁範例爲擁有該CLASS C之業者須設立,主要在將每一個IP 的反解設爲CNAME的形式,根據核發的IP Range 將那一段IP 的反解 CNAME 到一個特定的網域名稱

$GENERATE 0-15 $ CNAME $.company1-16

上述之例子爲16個 IP (211.72.211.0~211.72.211.15) CNAME 到 0.company1-16.211.72.211.in-addr.arpa. ~ 15.company1-16.211.72.211.in-addr.arpa.

查詢範例:

[root@twnic etc]# dig @dns.hinet.net ptr 225.9.17.210.in-addr.arpa.
225.9.17.210.in-addr.arpa. IN PTR
;; ANSWER SECTION:
225.9.17.210.in-addr.arpa. 3600 IN CNAME 225.sub224-255.9.17.210.in-addr arpa.
225.sub224-255.9.17.210.in-addr.arpa. 33294 IN PTR webredir.twnic.net.tw.
….以下略…….

反解: 不滿一個 Class C 怎麼辦 (3)

```
#/etc/named.conf
zone  "company1-16.211.72.211.in-addr.arpa"   {
      type master;
      file  "company1-16.211.72.211.rev";
};
```

```
;file company1-16.211.72.211.rev
;SOA/NS 訊息略
$ORIGIN company1-16.211.72.211.in-addr.arpa.
1     IN     PTR     ns1.xxx.com.tw.
2     IN     PTR     ns2.xxx.com.tw.
$GENERATE 3-15     $     PTR          pc$.xxx.com.tw.
```


承上

company1-16.211.72.211.in-addr.arpa. (擁有 211.72.211.0~15之單位) 的 Name Server 上面須設立該 Zone, Zone File 之內容再指出其反解(PTR) 的結果爲何

可以看得出，小於 /24 之反解設定較爲複雜,反解設定由您取得 IP 的 ISP 負責,ISP 應提供一般反解的網頁設定.而多數用戶認爲反解不重要,不滿 C 不設亦不向ISP 設定或不知道該如何設等等.

不滿一個 Class C 的設定方式，在做 DNS 查詢時，系統會產生兩行 log :

Sep 24 10:40:11 pc071 syslog: gethostby*.getanswer: asked for "37.103.74.204.in-addr.arpa IN PTR" ， got type "CNAME"
Sep 24 10:40:11 pc071 syslog: gethostby*.getanswer: asked for "37.103.74.204.in-addr.arpa"， got "37.32/27.103.74.204.in-addr.arpa

DNS設定介紹

- 用 BIND 還是用 Windows DNS ?
- BIND 版本差異
- BIND 工具程式
- 設定檔 named.conf
- 根伺服器介紹與設定
- zone file基本設定
- 資源紀錄（SOA，NS，A，CNAME，MX，PTR…）
- 正解/反解設定
- ✓ 主要（master）/次要（slave）伺服器的關係容錯及負載平衡功能（Round Robin）
- named之參數說明及啟動與停止
- WINDOWS DNS 設定介紹
- 使用 nslookup/dig 自我檢測

本章節由淺入深教您如何將您的DNS設定完成，當您都了解了DNS的原理後，不論是設定UNIX BIND或Windows DNS，都會是相當容易的一件事

從安裝、設定以及資源紀錄的解釋到Zone File帶您一步一步的了解DNS的設定流程，最後教您如何驗證DNS的正確性

負載平衡功能（Round Robin）

```
www     IN   A    211.72.211.80
www     IN   A    211.72.211.81
pc1     IN   A    211.72.211.80
pc2     IN   A    211.72.211.81
```

- 如上資料，一個 FQDN 有兩個以上之 IP 位址是允許的
- 回答的答案基本上是亂數決定的，不過在 BIND 9.X 中是以有些回答的依據設定 (rrset)
- DNS 僅做名稱之負載平衡，如 www/mail 或他類型的服務之負載平衡要取決其他技術(ex:Level 4 switch)

對於相同名稱而有多筆資料時BIND預設是採用round robin的方式提供答案，因爲client端通常都會採用第一筆資料先作連線因此可達到負載平衡的目的

RRset Ordering(Resource Record Set Ordering)

Fixed:　固定順序

Random:　亂數

Cyclic:　輪詢(round-robin)

DNS設定介紹

- 用 BIND 還是用 Windows DNS ?
- BIND 版本差異
- BIND 工具程式
- 設定檔 named.conf
- 根伺服器介紹與設定
- zone file基本設定
- 資源紀錄（SOA，NS，A，CNAME，MX，PTR…）
- 正解/反解設定
- 主要（master）/次要（slave）伺服器的關係容錯及負載平衡功能（Round Robin）
- ✓ named之參數說明及啟動與停止
- WINDOWS DNS 設定介紹
- 使用 nslookup/dig 自我檢測

105

本章節由淺入深教您如何將您的DNS設定完成，當您都了解了DNS的原理後，不論是設定UNIX BIND或Windows DNS，都會是相當容易的一件事

從安裝、設定以及資源紀錄的解釋到Zone File帶您一步一步的了解DNS的設定流程，最後教您如何驗證DNS的正確性

named 之及啟動與停止

- 基本上只要執行 named 即可啟動 DNS

```
$>named      或
$>/etc/rc.d/init.d/named start
```

- named 啟動狀況會寫到 /var/log/messages 中，只要查看這個檔案即可知道有無錯誤
- 為了方便了解其執行狀況，可以讓 named 於前景執行
- 要停止 named ，可直接 kill 掉其行程即可

```
$>killall -9 named     或
$>kill -9 pid-file     或
$>/etc/rc.d/init.d/named stop
```


在BIND8和BIND9分別有NDC及RNDC這兩個程式以控制BIND的執行及其他動作，可試在以 rndc 指令了解其參數意義，在設定 rndc 時，要先在 named.conf 中設定 key， key 的產生與設定如下:

```
[root@pc071 etc]# rndc-confgen
# Start of rndc.conf 設定一 rndc.conf ，建議與 named.conf 同一目錄
key "rndc-key" {
        algorithm hmac-md5;
        secret "lnWw5u3Jlr/8jaTpjfWF6w==";
};
options {
        default-key "rndc-key";
        default-server 127.0.0.1;
        default-port 953;
};
#在 named.conf 中加入如下設定
 controls {
      inet 127.0.0.1 port 953    allow { 127.0.0.1; } keys { "rndc-key"; };
```

在啓動 DNS 後，即可使用 rndc 來檢測系統(輸入 rndc 會顯示參數)

一般來說， named 在啓動時是以背景方式進行，所有啓動時所發生的正確或錯誤訊息會寫入 syslog 內 (一般即指 /var/log/messages)，此時則需查看

DNS設定介紹

- 用 BIND 還是用 Windows DNS ?
- BIND 版本差異
- BIND 工具程式
- 設定檔 named.conf
- 根伺服器介紹與設定
- zone file基本設定
- 資源紀錄（SOA，NS，A，CNAME，MX，PTR…）
- 正解/反解設定
- 主要（master）/次要（slave）伺服器的關係容錯及負載平衡功能（Round Robin）
- named之參數說明及啟動與停止
- ✓WINDOWS DNS 設定介紹
- 使用 nslookup/dig 自我檢測

本章節由淺入深教您如何將您的DNS設定完成，當您都了解了DNS的原理後，不論是設定UNIX BIND或Windows DNS，都會是相當容易的一件事

從安裝、設定以及資源紀錄的解釋到Zone File帶您一步一步的了解DNS的設定流程，最後教您如何驗證DNS的正確性

Windows DNS：設定

- 以下範例僅以 Windows 2000 Advance Server 為例
- Windows DNS 位於 控制台->服務->系統管理工具->DNS

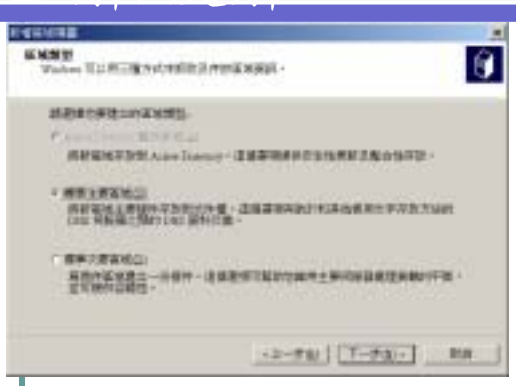
此二項目作用相同，在於設定 DN 之 Master/Slave 及正反解檔
即 SOA 中的 Expire 時間
即以手動方式清除過期(TTL 時間已到)之 RR
所有的 Slave 設定做 AXFR 動作
清除所有快取資料而不論其 TTL是否已到

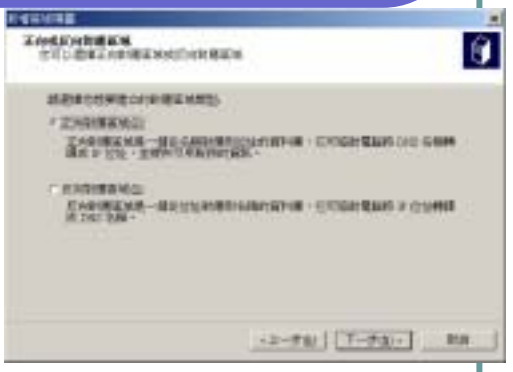
Windows DNS設定與 BIND 類似

但相對簡單的多

若有設定BIND經驗的使用者必能駕經就熟

Windows DNS： Master/Slave 及正解/反解





- 標準主要區域： Master
- 標準次要區域： Slave
- 正向對應：正解
- 反向對應：反解

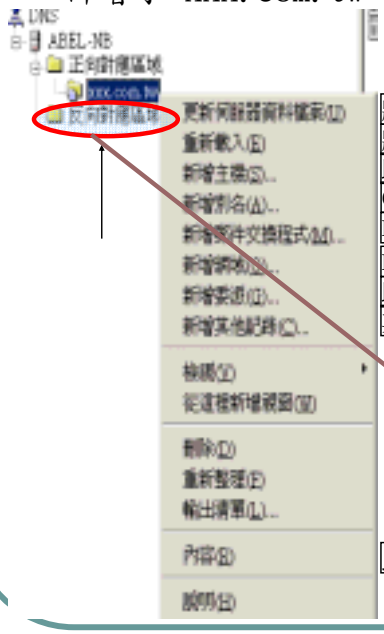
先選擇 DNS Server 型態

標準主要區域(Master)或標準次要區域(Slave)

再選擇正向對應(正解)或反向對應(反解)

Windows DNS: 新增一個標準主要區域-正解

- 新增了 xxx.com.tw



將設定寫回檔案

將 Zone File 重新載入到視窗

即 A RR，可一併建立 PTR 資料

CNAME

MX

DN 下再加一層，如 dept1 (同一 Zone)

即 NS，建立 Subdomain 再授權出去

其他不常用

即反解，如您已了解 Windows DNS 的正解的設功能，這個項目應可迎刃而解

此處非常重要而常爲人所忘.....見下頁

若要新增各種資源紀錄，請於該網域上按右鍵進行新增

Windows DNS: 網域的 內容項目

這一個畫面是一般的 Windows DNS 管理者常常忽略之處

一般：項目中有一 “是否允許動態更新”，主要爲 Windows 的特定需求而設

SOA: 這個畫面之值多要改，依據 RFC 1912 多不有不符合之處

名稱伺服器：需在此加上自身的名稱及 IP 及 Slave 主機資料

區域轉送：即 Zone Transfer，預設爲全開，建議選擇第二項，僅對 NS AXFR ，並勾選通知功能

動態更新主要在配合DHCP的環境下
Windows 能直接將電腦名稱更新至 Windows DNS Server
但Server端與Client 端都需作些許的設定

由於預設的轄區傳送是全部開啓的，許多使用者並沒有注意此一部分

DNS設定介紹

- 用 BIND 還是用 Windows DNS ?
- BIND 版本差異
- BIND 工具程式
- 設定檔 named.conf
- 根伺服器介紹與設定
- zone file基本設定
- 資源紀錄（SOA，NS，A，CNAME，MX，PTR…）
- 正解/反解設定
- 主要（master）/次要（slave）伺服器的關係容錯及負載平衡功能（Round Robin）
- named之參數說明及啟動與停止
- WINDOWS DNS 設定介紹
- ✓使用 nslookup/dig 自我檢測

本章節由淺入深教您如何將您的DNS設定完成，當您都了解了DNS的原理後，不論是設定UNIX BIND或Windows DNS，都會是相當容易的一件事

從安裝、設定以及資源紀錄的解釋到Zone File帶您一步一步的了解DNS的設定流程，最後教您如何驗證DNS的正確性

檢測工具: nslookup/dig

- nslookup
  - UNIX 及 Windows 平台皆有
  - 可使用命令模式與交談模式
  - 較好用但資訊較簡略
- dig
  - 僅存在 Unix-Base 環境，為 BIND 所附
  - 僅有命令模式
  - 較不好用但資訊完整

nslookup [-option ...] [host-to-find | -[server]]

```
# dig -h
Usage:  dig [@global-server] [domain] [q-type] [q-class] {q-opt}
        {global-d-opt} host [@local-server] {local-d-opt}
        [ host [@local-server] {local-d-opt} [...]]
Where:  domain    are in the Domain Name System
        q-class  is one of (in，hs，ch，...) [default: in]
        q-type   is one of (a，any，mx，ns，soa，hinfo，axfr，txt，...) [default:a]
                (Use ixfr=version for type ixfr)
```

檢測及追蹤方式介紹

```
$TTL    86400
@       IN      SOA     ns1     root (
        2002021301      ; serial
        1D              ; refresh
        1H              ; retry
        1W              ; expiry
        2D              ; min ttl
)
$ORIGIN xxx.com.tw.
        IN      NS      ns1
        IN      NS      ns2
        IN      MX      10      mail
        IN      MX      20      imap
ns1             IN      A       211.72.211.1
ns2             IN      A       211.72.211.2
www         IN      A       211.72.211.80
www         IN      A       211.72.211.81
ftp             IN      CNAME   www
mail            IN      A       211.72.211.25
imap            IN      A       211.72.211.143
```


此處將測試資料全貌貼上，以利後續查詢步驟解釋

以nslookup追蹤(1)

```
[root@pc071 named]# nslookup
Default Server:  ns1.xxx.com.tw
Address:  211.72.211.1

> set q=soa
> xxx.com.tw.
Server:  pc071.twnic.net.tw
Address:  211.72.211.71

xxx.com.tw
        origin = ns1.xxx.com.tw
        mail addr =
   root.xxx.com.tw
        serial = 2002021301
        refresh = 86400 (1D)
        retry   = 3600 (1H)
        expire  = 604800 (1W)
        minimum ttl = 172800 (2D)
```

啟動 nslookup 交談模式
連線至 nameserver

設定查詢類別為 SOA 資訊
查 xxx.com.tw.

如 Zone 所列之內容(SOA 訊息)

nslookup之操作說明如下:

set 設定某些值:

debug N:	以 debug level N 來查詢， N 值欲大訊息會愈多. 此時會看到許多查詢時所產生的訊息， 如查 www.kimo.com.tw ， 會列出其查詢的過程， 有關其所產生的訊息分成五大部份， HEADER， QUESTIONS ， ANSWER， AUTHORITY RECORDS ， ADDITIONAL RECORDS ， 這五大部份也就是一般 DNS 封包的格式， HEADER 段內說明了下面每一段有幾節...欲知詳情， 請參考 RFC1034，1035
nodebug:	關閉 debug N 的設定.
querytype=N:	查詢的類別， N 預設爲 A 記錄， 您也可以使用較簡單的表示法， 如 q=ns 或 q=ptr 等描述
class=IN	目前唯一可以使用預設值
timeout=N	每次查詢的 timeout 時間設爲 N 秒， 第一次失敗會將上一次的值 X 2，如 N=5 ，逾時時間則分爲 5，10，20，40
retry=N	逾時後重查的嘗試設爲 N 次， 若 N=4 即如上結果
server NS	切換使用的DNS 伺服器爲 NS， 在切換過去時會以現在的 NS 來查詢對方的 IP
ls –a DN [>FILE]	列出所有的 A 和 CNAME 記錄[到 什麼檔案]

ls 即 AXFR(轄區傳送) ， 若您沒有設定 allow-transfer 則每個人都可以操作 ls 指令

以nslookup追蹤(2)

```
;續前頁
xxx.com.tw        nameserver = ns1.xxx.com.tw
xxx.com.tw        nameserver = ns2.xxx.com.tw
ns1.xxx.com.tw    internet address =
   211.72.211.1
ns2.xxx.com.tw    internet address =
   211.72.211.2
```

當您查詢 SOA 訊息時，一併會列出其 NS 資訊，及 NS的 A 記錄，若您見到的不是這樣的訊息與對應關係，代表您的 DNS 設定有問題

set q=soa 之功能，除 SOA 外，您尚可設定其他 TYPE（如 NS，A，MX，CNAME，PTR ...等不同記錄），以查到您想要的資訊

命令模式（等同與上例）

nslookup -q=soa xxx.com.tw.

nslookup [-option ...] [host-to-find | -[server]]

以nslookup追蹤(3)

```
[root@pc071 named]# nslookup
Default Server:  ns1.xxx.com.tw
Address:  211.72.211.1

> server dns.hinet.net
Default Server:  dns.hinet.net
Address:  168.95.1.1
> set q=ns
> hinet.net
Server:  dns.hinet.net
Address:  168.95.1.1
;以下結果略

>ls -d hinet.net
[dns.hinet.net]
*** Can't list domain hinet.net.: Unspecified error

>server ns1.xxx.com.tw.
>ls -d xxx.com.tw.
;以下會列出 xxx.com.tw. 的 Zone File 內容
```

以 dns.hinet.net 做為 DNS Server

設定查詢 NS 記錄

對 dns.hinet.net 要求 ls (list) -d(domain) 資料 (即 AXFR)，結果當然被拒ls -d 之指令不適用於BIND 9環境

(省略部份訊息)，切回 ns1 並做 AXFR，若允許 Client IP 做 AXFR 則會列出 Zone File 內容

假如您使用“除錯模式”的話，看到的資料還將更多!

以nslookup追蹤(4)

```
>set q=a
>www.xxx.com.tw.
Server:  ns1.xxx.com.tw
Address:  211.72.211.1

Name:    www.xxx.com.tw
Addresses:  211.72.211.80，
   211.72.211.81

>www.msn.com.
Server:  ns1.xxx.com.tw
Address:  211.72.211.1

Non-authoritative answer:
Name:    www.msn.com
Addresses:  207.68.171.245，
   207.68.171.247， 207.68.172.234，
   207.68.173.244， 207.68.173.254，
   207.68.171.244
```

查詢 www.xxx.com.tw.資訊，列出兩個 IP，即是此記錄做 Round Robin，再查一次可能是相同順序，亦可能 81 會排到前面，如果您在看此網站，有時可能會連到 80，但有時又會連到 81，即可達到負載平衡之效果

查詢外面的網域名稱可以查的到，代表root server的設定正常

Server 會一次回應所查詢網域全部的 IP，Client 會依序使用，所以可以達到負載平衡之目的，但在 Windows 2000 以上的版本查詢，因爲 Window 2000 會做 Cache (非 DNS 的 Cache，而是 2000 自己 Cache . 要避免這種狀況，可將服務中的 [DNS Client] 這一項關閉即可，實際案例可參考 http://phorum.study-area.org/viewtopic.php?t=20718&highlight=DNS+Client

以dig追蹤(1)

```
[root@pc071 named]# dig @211.72.211.1 xxx.com.tw ns

; <<>> DiG 8.3 <<>> @211.72.211.1 xxx.com.tw ns
; (1 server found)
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY， status: NOERROR， id: 6
;; flags: qr aa rd ra; QUERY: 1， ANSWER: 2， AUTHORITY: 0， ADDITIONAL:
   2
;; QUERY SECTION:
;;      xxx.com.tw， type = NS， class = IN

;; ANSWER SECTION:
xxx.com.tw.              1D IN NS        ns2.xxx.com.tw.
xxx.com.tw.              1D IN NS        ns1.xxx.com.tw.

;; ADDITIONAL SECTION:
ns2.xxx.com.tw.          1D IN A         211.72.211.2
ns1.xxx.com.tw.          1D IN A         211.72.211.1
```


dig [@server] [-b address] [-c class] [-f file-name] [-k filename] [-p port#] [-t type] [-x addr] [-y name:key] [name] [type] [class] [queryopt...]

此處可見 dig 顯示的資訊多了許多，一般我們常用

$>dig FQDN_or_Domain TYPE

來查詢 DNS 的資料， 這時候的 dig 所查的主機是使用 reslover 中所定義的 DNS 伺服器

若我們要臨時要自訂一個 DNS Server 則可使用

$>dig @DNS_SERVER FQDN_or_Domain TYPE

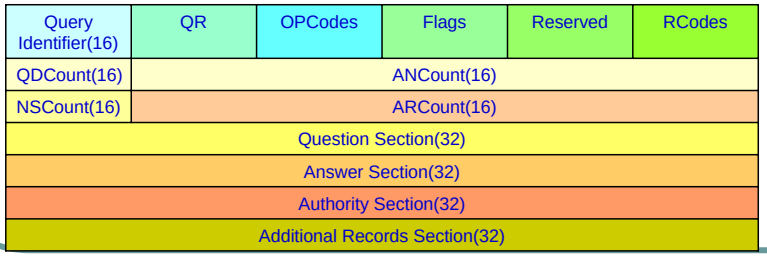
來查詢，其作用就如在 nslookup 中指定 server 一般

以dig追蹤(2)

- 命令格式為

  dig @dns_server domain type

- 由上頁可看出 dig 較 nslookup 複雜許多
- 其列出許多 DNS 封包的欄位資訊，可參考 RFC 1034/1035 或 Oreilly “DNS & BIND” 一書之介紹
- DNS封包格式



DNS 的封包格式

QID

DNS 查詢封包編號，作爲確認依據.

QR

查詢封包爲 0； 回應爲 1 .長度爲 1 bit .

OPCodes

封包類別(QUERY， IQUERY， STATUS， Reserved).長度爲 4 bits.

Flags

共 4 bits ，各表示：AA(Authoritative Answer)、TC(Truncation)、RD(Recursion Desired)、RA(Recursion Avalable).

Reserved

保留未用.

RCodes

回應訊息，長 4 bits ，除 0 及 6-15 保留未用外，1-5 分別爲：Format Error、Server Failure、Name Error、Not Implemented、Refused.

Question Section、Answer Section、Authority Section、Additional Records Section

每一 Section 分爲 NAME、TYPE、CLASS 三個子欄位，分別作爲查詢、應答、授權、額外記錄等封包之資訊，及各自長度.

DNS Running ?

- 如何確定 DNS 是否運行呢 ?
  - Port Scan 目標 53/UDP（敏感動作）
  - telnet 53 port
  - nslookup -q=ns . Dns_server（查詢其 Root 記錄）
  - dig @dns_server . Ns
- DNS 不正常原因:
  - 語法錯誤造成 DNS 未啟動
  - 觀念錯誤造成運作不正常
  - 網路是否正常（流量，斷線…）
  - 是否被 Router/Firewall 等檔掉了 53 port
  - 被入侵或欺騙
  - 判別密碼被猜出而改了指向

上述爲DNS Trouble Shooting 該注意的部分，若發生問題可以檢視是否發生上述狀況

DNS 各種問題之探討與觀念介紹

- 設定上常犯的錯誤
- SOA 中的數字意義
- MX 與 CNAME/NS 設定對 Mail Server 的影響
- TTL 設定對網路的影響
- 轄區傳送
- Lame Server
- Bad Referral
- DNS 使用協定之觀念
- 兩部以上 DNS 之作用
- DNS query 的種種行為探討
- 負面答案（negative answer）對系統之影響
- 反解
- DNS 不安全之後果

此部份主要探討一些重點觀念的釐清，及一些常碰到的問題

設定上常犯之錯誤及觀念澄清

- Zone File (ex: xxx.com.tw)

```
$TTL 38400
xxx.com.tw.    IN  SOA  ns1.xxx.com.tw.  abel\.yang.twnic.net.tw.  (
    2003041801    ; serial
    38400         ; refresh
    3600          ; retry
    864000         ; expire
    2D         ; mininum ttl
)
              IN    NS     ns1.xxx.com.tw.
              IN    NS     ns2.xxx.com.tw.
              IN    MX     10     mail
ns1           IN    A      211.72.211.1
ns2           IN    A      211.72.211.2
www           IN    A      211.72.211.3
mail          IN    A      211.72.211.4
dept1     IN    NS     ns1.dept1
              IN    A      211.72.211.101
```


“.” root 別忘了

Email帳號部分若有“.”，要用“\.”表示

SOA 用 .代替 @

TAB鍵或空白分開都是合法的

SOA email的正確性， 有許多人會亂填或者使用root@localhost 但並沒有人去收root的信件

這樣當您的網域有問題別人想聯絡您的時候會發生找不到人的狀況

SOA 中的數字意義

- Serial (yyyymmddvv)
  - Slave 主機向 Master 主機Zone Transfer 要求時，會先判斷之值，若 Slave 主機上之 Serial 小於 Master 之數值時，才會做 AXFR，所以，當我們改變了 Zone File 的內容時，並有Master/Slave 之關係時，應同時增加此值，以確保 Slave 主機亦會同步更新(data synchronization 用)
- Refresh (1D)
  - 此數值於每次AXFR後即開始計時，每隔此一時間，Slave即會向Master要求更新資料，此一動作之第一步即是檢查Serial
- Retry (1H)
  - 當Refresh之動作因不明原因失敗時，為確保更新動作能加快而設定此值
- Expire (7D+)
  - 當此一時間到達時，Slave之Zone File 即會放棄而不使用，也就是Slave主機就不在負責此一網域名稱之查詢處理
- minimun TTL (1H)
  - 當 Refresh + N x Retry > TTL 時，AXFR 的請求即會停止，除非重新啟動 DNS

在 RFC 1982 中定義：serial 欄位爲一 32 位元的無正負別整數， 因此 serial 的範圍應該是 [0..4294967295],一般的數字都是以秒爲單位，但您也可以用 H(小)、D(天)、W(星期)來做單位，如: 3H 和 259200 是一樣的. 設定任何時間都可以但記得要符合以下規則:

expire >= refresh + retry
expire >= 10 * retry

MX/CNAME/NS 合用之影響（1）

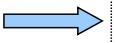
- 根據 RFC 1034， 1035， DNS 設定檔中， NS RR 這一種記錄項目， 只能夠指到一個 正式的 canonical name.（ 也就是有定義成 A RR 記錄項目者 ）
- 如果將 NS 指到一個 alias name（ CNAME RR 的 LHS 左邊 ）， 這樣一個 NS 記錄 項目是無效的 (invalid)， 其它網路上的 DNS 系統， 將不會認可這一個 NS RR.
- 同上理，MX記錄亦應對應到一A RR ，而非 CNAME RR，否則會造成 Mail 退信的狀況
- 常見狀況是因為公司 IP 不足，很可能是一部主機負起 DNS/MAIL/WEB…等相關服務而造成此一問題一再出現

請參考 http://www.sendmail.org/faq/section4.html#4.5

MX/CNAME/NS 合用之影響（2）

- 下列設定將造成該網域名稱的收信狀況不正常(退信訊息:Local configuration error)

```
        IN    NS        dns1
        IN    NS        dns2
        IN    MX   10     mail
dns1    IN    CNAME       www
dns2    IN    CNAME       www
mail    IN    CNAME       www
www         IN    A         211.72.211.1
```



將 CNAME 皆改成 A 記錄即可

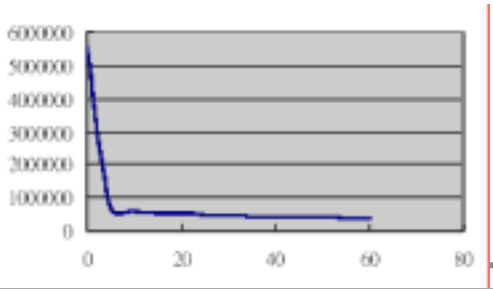
- 直接於 TWNIC DNS 指定上，以 DNS 模式設定兩部以上主機(如下例)，而不設 DNS Server（亦很常見），並於 211.72.211.1 上架設 Mail Server，造成收信不正常（這樣的設定是錯誤的）

```
    mail.xxx.com.tw     211.72.211.1
    www.xxx.com.tw      211.72.211.2
    ……
```


如果擔心會出問題，建議全部使用 A 紀錄代替 CNAME 紀錄,而若 Zone File 較大時更建議可依 IP 來排序,以利判別關係(而名稱來排序較不能判斷出中一個 IP 有幾個名稱)

TTL 設定對網路的影響

- TTL Time To Live， 他人DNS 快取了你的域名的某筆資料時，將在他的機器上暫存多久讓他人於查詢同一筆資料時可快速回應(這個值是由自己指定)
- 當 TTL 為 0 時，表示 RR 不為他部 DNS 主機所快取，不過就實際層面而言很少人會如此設定
- 當要更改 DNS 內容或公司網路變動時，適當的縮小 TTL 值是必要的，可以縮短整體網路的更新時間
- DNS 預設每小時檢查一次所有快取資料是否過期
- 下圖為 www.twnic.net.tw 一個月之查詢量在不同 TTL 值上的變化，我們可以看出，只要設定了 TTL 值，即會減輕 DNS 查詢對網路的負載



圖爲 www.twnic.net.tw 的查詢量分析，當 TTL 定爲 0時 每天有將近六百萬次的查詢，TTL改爲6時卻降到約八十萬次，隨著TTL值的降低 查詢次數亦更著降低.故如動態 DNS 之設定 TTL 不爲 0，設在 20 秒內則可以取得一個較佳的平衡.

轄區傳送（1）

- Zone Transfer（AXFR）在於同步 Master/Slave zone file 一致
- SOA 訊息中的序號是決定是否同步的關鍵
- 同步之動作是由 Slave 向 Master 發出，發出時機視 Refresh 或 Retry 時間而定
- .tw 網域約 25% 可進行 AXFR
- 不論 BIND/Window DNS，AXFR 預設為全開，也就是任何人皆可向您的 DNS 要求 AXFR，故需做一些存取控制，以避免網路資料外流

BIND : allow-transfer { IP1;IP2;};

Windows : 按右鍵選內容，再找“區間轉送項目

使用 nslookup 作zone transfer

nslookup 進入交談模式

server server.host.name.

ls transfer.domain.name.

使用dig作zone transfer

dig @server.host.name. axfr transfer.domain.name.

轄區傳送 (2)

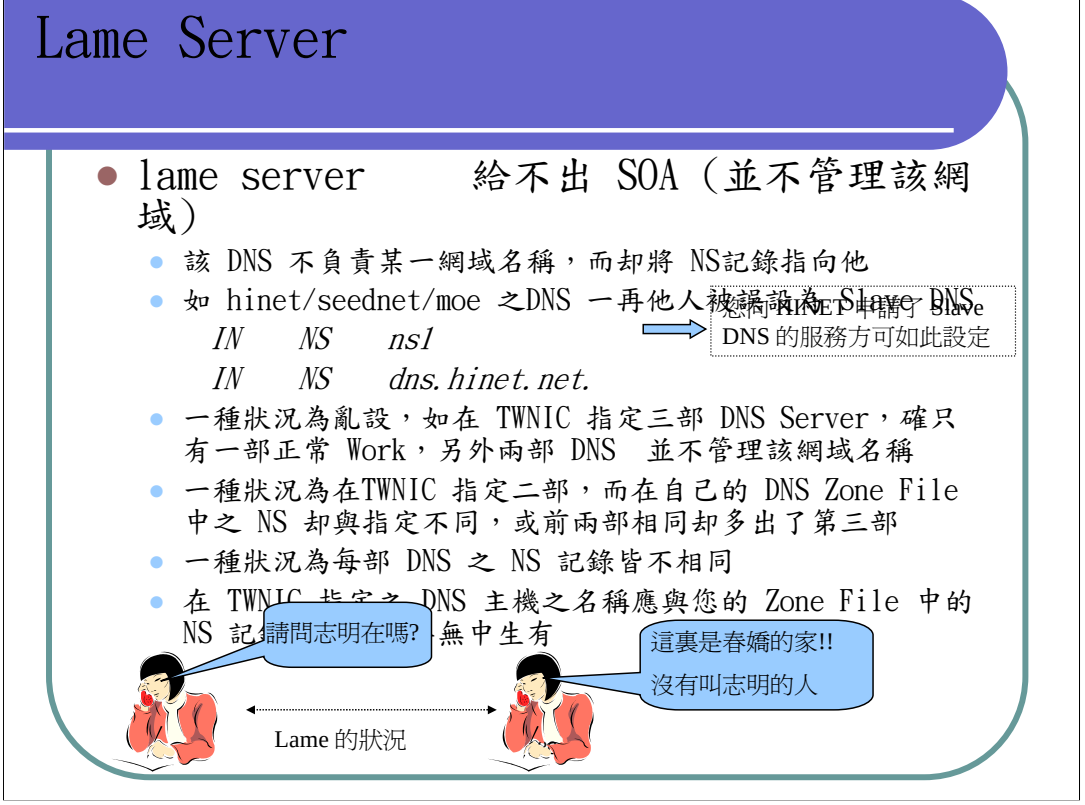
- Master 可利用設定檔中的 notify 或 also-notify 功能，向 Slave 發出通知，請 Slave 再次來確定資料是否同步(notify 主機為 Zone File 中的)

```
notify yes;     // notify 對項為 Zone File 中的 NS 主機
also-notify {IP1;IP2};     //特別對某部 DNS 送出 notify
```

- AXFR 使用 53/TCP port， 且為一筆一筆方式傳送，故較大的 zone file 傳送時會較費時
- AXFR 僅是同步動作，若有其他替代方案，亦可達到相同功能

一般master/slave只需使用DNS Server的zone transfer 機制就能夠更新

但在像 root 這種超大的zone時， 可能就必須使用ftp的方式來傳送， 再重新啓動



簡單地說， 會形成 Lame server 的條件是，

a) NS delegation (被授權)

b) not authoritative. (沒有該 domain zone 全部的資料)
* 也就是，
a) 該 host 沒有跑 named 程式 (or equivalent DNS server)
b) 該 host 有跑 DNS server 程式， 但是起始設定檔 /etc/named.boot (or equivalent) 沒有對應的 primary or secondary 設定行.

若您在 TWNIC 指定了二部 DNS 主機，但實際上僅有一部運作，此時即是所謂 Lame Server 狀況，他人在查詢您的 FQDN 時，有二分之一的機會查不到，若別人寄信給您亦有可能50% 第一時間寄不到.

若您在 TWNIC 指定 ns1 與 ns2 ， 但在您的 Zone file 中卻寫 ns3 時，這種狀況完全不一致時，在 BIND 9.X 時，根據 Mail Server 的不同可能有不同的影響 (Ex: Exchange 可以 Work， 但 Sendmail 可能不正常)，而在 BIND 8.X 就沒有什麼影響

最重要的概念即是您在 TWNIC 的指定要完全與您 Zone File 中的 NS 記錄一致，不論 IP 或 Hostname 皆要相同，並且儘量避免使用 CNAME，如此 DNS 的運作方能正常.

您可使用 http://dns-security.twnic.net.tw/dns_report.php 來檢測自己 DNS 設定上的錯誤，例如 Lame Server，CNAME， TTL …等檢查，您也可使用 http://www.dnsreport.com 亦是相同功能.

Lame Server 案例:

Bad Referral

- Bad Referral 給錯 SOA（可能是該域名的下一層）
  - 上一層的 primary NS server， 已將某一 sub-domain zone， 授權出去
  - 下一層的單位， 被授權的 DNS server， 將 SOA RR 的 domain zone 設錯了
  - 較常發生在反解
  - 如負責 10 個 Class C 的網路 211.72.1-10.x，為偷懶而設成 72.211.in-addr.arpa. 的 Zone
  - 或負責 a.com.tw， b.com.tw， c.com.tw 而設了 com.tw. 的 Zone

131

例如說某公司申請了固定的ADSL，從ISP那裡拿到了8個IP，而ISP方使用一對一的方式設定了這8個IP的反解，但該公司卻直接設定了一整個 Class C 的反解時

就是Bad Referral常出現的狀況.

正解也有可能會出現這樣的問題，但是出現的狀況較少，因爲正解的思考比較直覺，較不易搞混.而 bad Referral 通常不會影響 DNS 的運作，只是在系統記錄上留下許多訊息而以.

DNS 使用協定之觀念

- 基本上Query皆使用 53/UDP 協定
- AXFR 使用 53/TCP 協定
- 平均每個 DNS query 封包約 80-100 個 bytes
- 一個DNS封包最大為512bytes（UDP）
- AXFR 的資料傳輸預設為一筆一筆傳
- 不建議更改 DNS Listen 的 Port，因為 Client 只認 53 port

正常狀況， 99% 以上的 DNS 查詢， 使用 UDP 封包.如果 DNS UDP 低於 90%， 則表示 TCP 量偏高. 則可能顯示:

網路條件可能非常差， 導致許多 DNS UDP 失敗比率太高， 轉而使用 DNS TCP 查詢.

網路使用異常， 諸如 zone transfer 等主要 DNS TCP 應用太過頻繁.

兩部以上 DNS 之作用

- 根據 InterNIC 的規定，NIC 之 DNS 指定應兩部以上
- 兩部 NS 之意義在於
  - 容錯

    只有一部 DNS 失效時，眾多網路服務也跟著失效，且正確的兩部以上主機更應處在不同的網段以降低風險
  - 系統安全

    二部以上的 DNS 主機能互相支援，亦爭取了足夠的處理時間，也有效的降低網路安全的風險
  - 負載平衡

    若您設定了兩部以上的 DNS 主機，當有人連接您的網站前，其查詢乃是兩台主機輪流運作(輪詢，Round-Robin). 在這樣的運作機制下讓您的系統可以更穩定

In a Men & Mice's February 2001 survey of 6000 randomly-selected .com domains， 38% of tested zones had all their DNS servers in one subnet， and only 9.9% maintained a single functioning DNS server. (http://www.menandmice.com/infobase/mennmys/vefsidur.nsf/index/2.1)

TWNIC 於2002年的調查中:所有.tw的網域，只有一部DNS Server佔了15.34%， 所有DNS Server 在同一個網段的佔了13.11%

DNS query 的種種行為探討

- 基本上 DNS query timeout 的時間為 1 秒
- 但 timeout 後，可能啟動 retry 機制，最多 retry 為 4 次
- 每次的 retry，timeout 時間即會加倍，即成 1，2，4，8
- 如果查不到，可能會啟動 default domain/search 之功能
- Default domain/search 亦會有 retry/timeout 等機制
- 如果又 Lame Server ….

因此查詢可能會拖延到 1+2+4+8=15秒 才會出現找不到. 而一般我們常看到的一個問題, telnet 要 15 秒才會看到提示符號 (login:), 即是這個原因, 當我們登入時,此時系統會留下記錄,系統看到來源 IP 會去查其反解, 故若沒有反解情況下, DNS 可能會一直試著去找, 直到時間用完爲止.

Resolver 有 default Domain/Search 功能時,可能在找不到時,以 Default domain 去試,例如,查詢 abcd.com.tw 查不到時,而您的設定了 default domain (unix/linux 在 /etc/resolv.conf, Windows 在 TCP/IP 選項中的進階項目) 爲 com.tw 時,則這個找不到的查詢會變成 abcd.com.tw.com.tw. 這種情形.

負面答案對系統之影響

- Negative Answer 不對的事 DNS 會記著，避免重覆發生
- 預設時間為 1小時
- 不對的是如 Lame Server/Bad Referral/找不到主機/找不到答案…
- 有效的減少失敗查詢對網路流量的影響

只要DNS伺服器有回應一個正確的IP位址並傳送到用戶端，而用戶端將查詢到的結果快取下來，我們就稱爲此種快取記錄爲正確回應的快取(Positive Cache)記錄，另一種快取記錄稱爲不正確回應的快取(Negative Cache)記錄或稱爲負向快取記錄，此種記錄所代表的意思是DNS伺服器沒回應一個正確IP位址的查詢記錄

在 windows 2000的系統中

可以用 ipconfig/displaydns 的指令來查看DNS 快取的狀態

可以用 ipconfig/flushdns 的指令來清掉所有 DNS快取

反解

- 反解　有必要嗎?
- DNS Query 正解約佔 40%，反解佔 60%
- 許多的服務會檢查反解
  - Mail Server　　可定義只接受有反解的主機才能收發信
  - Syslog　　每一個 connect 的動作都會查 IP 反解
  - FTP　　可定義需反解主機
  - 認證服務　　Firewall/Proxy/Router 等亦與反解相關
- 滿一個 Class C 以上之用戶皆應設定反解
- 不滿一個 Class C 需設反解嗎 ?
  - ISP 有義務提供客戶反解服務
  - 客戶有權要求
- 只要有 IP 即應設定反解，以供反查服務，目前台灣的反查比例約 1/3

設定反解的理由:

a) 管理上的理由 (administration)
一個 IP addr. 有註冊， 表示有人使用這個節點. 在 local 看來， 可避免 IP addr. 相衝的問題， 在外人看來， 這一個單位， 是比較 well-organized.

b) security/access control control.

c) performance & load balancing support.
目前， 幾乎所有的網路連線程式， 都有反解查詢的程式碼， 寫在裏面. 也就是， 不管使用 telnet， ftp， www， news， ... 等只要你一連上 remote site， 對方的系統， 都會作反向查詢， 以取得地要的 host/ip 資訊. 就像超級 FAQ， telnet 連上一台機器，過了近 15 秒才會出現提示符號，即是沒有反解記錄的關係.

相關資訊請參考:http://freebsd.ntu.edu.tw/named/DNS-tech/ip-dns-whois.html

DNS 不安全之後果（1）

- 為何會不安全
  - 被入侵: 可能從別的服務程式或直接從 DNS 入侵
  - 被欺騙: 被造假的 DNS 訊息欺騙
  - 若使用 BIND 建議您昇級至 9.2.3 最新版
- 可能的後果
  - DNS 失常

    這是最常見的情況，使用者會感覺到 DNS 失去作用.此時除了重新啟動，還需去了解為什麼 DNS 會失效.
  - 假造網頁

    原來的 www 位於 211.72.211.80，但被改到 1.2.3.4，而其又 mirror 您的網頁，此時很容易套出您使用者的身份與密碼，而又不易被察覺(因為使用者輸入的是 www.xxx.com.tw)(即一般俗稱 man in middle 手法)

137

根據目前已知的DNS版本漏洞資訊，可以歸納出幾種攻擊的模式：

1. Buffer overflow：即緩衝區溢位的攻擊，藉由傳送特定的shell codes，可能會讓攻擊者取得named執行時的權限或是root權限，執行惡意的指令.甚至在被入侵的主機開啓一個remote login shell，即所謂的後門，以方便下次的入侵.例如在2000年4月間流傳的lion worm即是.

2. Crash server：藉由傳送不正常的訊息，導致named處理時產生錯誤crash掉.

3. Denial of Service：藉由傳送不正常的封包或是因爲系統管理者錯誤的設定，導致伺服器無法提供正常的服務，即阻斷服務攻擊(DoS attack)的發生.和crash server不同的是，這種攻擊模式並未造成伺服器當機，只是忙於處理『不正常的』requests.

4. Information leak：即洩漏伺服器資訊的攻擊方式.有心人士可以得知系統相關資訊，並擬定下一波的攻擊行動.

DNS 不安全之後果（2）

- 可能的後果
  - **複製郵件**

    所有的信件到達你的服務器之前可以被拷貝，修改或者刪除.入侵者只要了解郵件伺服器與 DNS 的運作原理輕易即可達成此一目的，而其也可以偽造成您的信件寄出，這些都是可以透過 DNS 完成，而您不會感覺到很明顯的異常.其手法同上一段所述
  - **授權問題**

    某些與信任有關服務（如 mail，firewall，proxy等等）若涉及DNS域名信任時將會無效.如您的防火牆信任 any.com.tw 網域可自由通過，在 DNS 被入侵後防火牆將完全失效.因為入侵者可在您的 DNS 中漆加他機器為 any.com.tw 網域機器的資訊
  - **系統權限**

    當駭客從 DNS 入侵後(指遠端溢位攻擊，remote buffer overflow ），通常亦直接取得 DNS 權限

建議:

以最小的權限執行 DNS Server

如果可能的狀況DNS Server上不要提供其他服務

隱藏版本資訊並注意 Access Control

如果是 BIND DNS 可以使用 chroot 的方式來增加安全性，chroot 做法可參考http://phorum.study-area.org/viewtopic.php?t=18121&highlight=bind+chroot

升級至最新版本的DNS Server

安全性相關問題請參考:

http://www.cert.org.tw/document/docfile/DNS%20Security.pdf

http://www.cert.org.tw/document/docfile/DNS%20survey.pdf

名詞解釋

AXFR　同 zone transfer

CIDR　Classless Inter-Domain Routing， RFC1519

DN　網域名稱(Domain Name)

NS　名稱伺服器(Name Server)

RR　資源記錄 (Resource Record)

TTL　time to live ， 存活時間

delegation　委任/授權

negative answers負面答案，查不到的狀況

Lame Server　不良的委任記錄

zone　轄區

zone transfer　轄區傳送

DNS 相關資源

- RFC 1034: domain names - concepts and facilities
- RFC 1035:domain names - implementation and specification
- RFC 1886:DNS Extensions to support IP version 6
- RFC 1912 Common DNS Operational and Configuration Errors
- RFC 2065:Domain Name System Security Extensions
- RFC 2136:Dynamic Updates in the Domain Name System (DNS UPDATE)
- RFC 3490 IDNA: Internationalizing Domain Names in Applications
- RFC 3491 Nameprep: A Stringprep Profile for Internationalized Domain Names
- RFC 3492 Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications
- BIND 官方網站 http://www.isc.org
- http://www.dns.net/dnsrd/
- http://www.menandmice.com/
- http://dns-learning.twnic.net.tw/
- http://ns.nctu.edu.tw/ http://dnsrd.nctu.edu.tw/
- DNS 錯誤訊息列表 http://www.menandmice.com/docs/named_messages.htm

課後討論

- 有一單位（xxx.com.tw.)遭受來自網路上的 DNS Query 的 DDOS 攻，造成其 T1 的頻寬完全塞滿，每秒查詢萬次以上，無數的主機皆向其詢問， xxx.com.tw. AAAA 的內容，但該網域名稱並無 AAAA 之 Record， 請問如何解決 ？
  （ TTL 及 Cache 因素）

- 某人在 TWNIC 指定其 xxx.com.tw. 之 DNS Server 如下面內容:
  mail.xxx.com.tw. 1.2.3.4
  xxx.com.tw. 1.2.3.4
  而其本身並沒有在這兩個 IP 上架設 DNS Server（有 MailServer)， 請問別人以 user@xxx.com.tw 寄信給他時， 收得到嗎 ？ 為什麼 ？
  （版本差異）

- VeriSign 在 .com .net 的 gTLD 上加了
  $ORIGIN com.
  ** IN A 64.94.110.11*
  請問，對 Internet 各種服務會有什麼影響 ？
  (綜合觀念，telnet /smtp/www ¡4¡)

課後討論

- 上層 DNS 中 .tw 及 .com.tw 的 DNS 查詢量那一個較多？為什麼？
  （TTL 及架構因素）
- 依據 ISC 及 CERT/CC 組織的說明，建議使用 BIND 運行 DNS 的人，為了安全的理由，最好都換到 BIND 9.X 版，為什麼像 HINET/SEEDNET 這些 ISP 都不願意換呢？
  (版本差異)
- 有一個網域名稱叫 tw.com.tw ， 為什麼他的查詢是高達每小時數萬次？管理人員覺得很困擾，用 FireWall 將 DNS query 給檔下來又會發生什麼事？
  (default domain/search 及 loop 問題)