Servei de noms DNS

El protocol TCP/IP identifica cada node d'una xarxa per la seva adreça IP. Per tant, dins d'una mateixa xarxa no podrem tenir mai dues adreces IP idèntiques.

Adreça IP: nombre de 32 bits dividit en 4 blocs de 8, que segueix el format XXX.XXX.XXX.XXX

Degut a que no és massa còmode haver de memoritzar les adreces IP de tots els membres d'una xarxa, es va començar a utilitzar noms enlloc de les adreces IP. Per implementar-ho, es definien uns arxius HOST.TXT, que contenien, per a cada màquina, la seva adreça IP i el nom equivalent. Aquests fitxers s'enviaven per correu electrònic a cada membre de la xarxa per tenir tots la mateixa versió. Hi havia un organisme que es deia SRINIC, que era l'encarregat d'aquesta sincronització.

Al anar creixent les xarxes i al aparèixer servidors amb noms duplicats, aquest mètode es va fer inviable i insostenible. Llavors es va crear el sistema de noms de dominis DNS (Domain Name System).

DNS:

DNS és un conjunt de protocols d'aplicació que s'utilitza tant amb TCP com amb UDP.
DNS és una base de dades distribuïda que conté una estructura de noms jeràrquica que permet identificar qualsevol element de la xarxa.

Es defineix el format FQDN (Full Qualified domain name). Es defineix un sistema de noms jeràrquics format per diverses extensions que van separades per punt. Cada extensió ens permet identificar la posició del node dins de la xarxa. Direm que un nom és FQDN si especifica tot el camí per localitzar aquell servidor.

Els noms estaran formats pels diferents dominis als que pertany la màquina. Els dominis de més pes s'escriuran a la dreta del nom i, quan més cap a l'esquerra, menor serà el seu domini.

A Internet, el primer domini pertany generalment a països o a organitzacions internacionals:

- .int: organitzacions internacionals (OTAN)
- .org: organitzacions no gobernamentals
- .net: altres xarxes que s'han unit a Internet
- .com: comercial
- .edu: educacional
- .mil: militar
- dominis de països: estàndard format per dues lletres (.es,.fr,.uk....)

Els dominis són administrats per un organisme anomenat NIC (Network Information Center). A Espanya ho administra EsNIC., que depèn de RedIRIS (xarxa d'investigació espanyola).

Per tant, DNS serà un mecanisme que, en qualsevol moment, permetrà traduir noms (FQDN) en adreces IP.
Cada màquina haurà de ser la responsable d'administrar un domini. Aquest domini serà la part de l'arbre d'adreces format pel node i per tot el que penja d'ell. Podran penjar d'ell màquines i altres dominis.
Els noms de dominis tenen restriccions, només poden contenir caràcters a..z, A..Z i 0..9. No poden contenir '/' ni '_', ni altres caràcters especials.

Una zona serà un arxiu físic que s'utilitzarà per administrar un conjunt de registres de l'espai de noms de DNS. Generalment, un domini estarà dividit en diverses zones per facilitar la seva gestió i permetre que aquesta es pugui distribuir entre diversos elements.

Un subdomini serà una branca que pengi en l'estructura d'arbre d'un domini.

Servidors de noms:

Els servidors de noms emmagatzemen informació sobre l'espai de noms del domini que representen. Tenen autoritat sobre aquest domini.

Quan es configura un servidor de noms, se li informa dels altres servidor de noms que conformen el mateix domini.

En el primer nivell estan els server roots, que són els que administren el primer domini (.com, .org,...). Aquests coneixen els DNS dels servidors que gestionen el primer nivell. Per tant, una consulta es podria esquematitzar e els següents passos:

- El client sol·licita un nom al seu servidor DNS
- El servidor coneix la IP del servidor root corresponent i li passa la consulta
- Aquest, en funció del segon domini, li passa la consulta a un servidor de primer nivell
- Anirem baixant per l'estructura d'arbre passant pels servidors dels diferents dominis que conformen el nom
- Quan arribem al darrer domini, ja podrem obtenir la adreça IP del que desitjàvem
- Cal mencionar que normalment el nostre servidor DNS guardarà una caché amb les adreces que ja ha resolt arran d'anteriors consultes per estalviar repetir búsquedes

Tipus de servidors:

- Servidors mestres: Servidors que són els responsables d'una zona. Totes les peticions referents a aquella zona acabaran passant per ells. Es divideixen en dues categories:

    - Primaris: són els que mantenen la zona. Obténen la informació dels seus arxius locals. Per tant, són els que realment emmagatzemen l'estructura de noms de la seva zona. Qualsevol canvi a la zona s'haurà de realitzar en aquest servidor

    - Secundaris: Obté les dades del servidor primari. Per tant, és una còpia d'ell. Caldrà assegurar-nos que qualsevol canvi al servidor primari es vegi reflexat al secundari. Els servidors secundaris són necessaris per assegurar redundància (per assegurar la tolerància a errors) i per treure feina al servidor primari

- Servidor Caché: No mantenen la informació de cap zona. La seva única finalitat és la de comunicar-se amb servidors mestres per sol·licitar resolucions de noms i, un cop obtingudes, anar-les guardant fent de caché, per tal d'estalviar tràfic de xarxa i augmentar la velocitat de resposta a les consultes

El concepte de primari o secundari és a nivell de zona. Per tant, un servidor pot ser primari per a una zona i secundari per a una altra.

Forwarders: Seran servidors de noms que poden rebre peticions d'altres servidors de noms i que intentaran resoldre aquelles sol·licituds que els que els demanin no hagin pogut resoldre localment. Els forwarders seran els que podran realitzar consultes per internet. Per tant, a la configuració dels servidors de noms clients hi haurà sentències per indicar qui seran els seus forwarders.

Esclaus: són servidors de noms que si no són capaços de resoldre una adreça localment, únicament podran realitzar sol·licituds als forwarders. No poden demanar informació a cap servidor que no sigui un dels que té definits com a forwarders.

Tipus de consultes que un client pot realitzar a un servidor DNS:

1) Recursiva: Consulta típica entre un client i un servidor de noms. El servidor no podrà transferir la consulta a un altre servidor. O la resol ell o retorna un error
2) Interactiva: Consulta en la que un servidor tracta d'obtenir la millor resposta, fent sol·licituds a altres servidors
3) Inversa: Consulta en la que, a partir d'un domini, volem conèixer la IP corresponent

Els servidors fan cache amb tots els dominis que han resolt. Existeix un temps de vida (TTL) de la informació d'aquesta caché. El servidor, al cap del temps especificat eliminarà la informació de la caché.

Configuració del client DNS:

Els clients DNS hauran de tenir un arxiu en el que configuraran els servidor DNS als que accediran. A Linux aquest arxiu es trobarà a /etc/resolv.conf.

En aquest fitxer podrem trobar les següents directives:

- domain: indicarà el domini local. Quan s'intenti resoldre un nom de màquina, se li afegirà aquest domini per construir el FQDN del host sol·licitat
- search: llista de dominis que s'intentarà afegir al nom de la màquina. Per tant, aquesta sentència ens permetrà definir patrons de recerca, ens permetrà posar 'beavis' quan en realitat es muntarà el DNS 'beavis.altredomini.com" si hem definit la sentència search altredomini.com

  Aquestes dues sentències (domain i search) són excloents, o es posa l'una o l'altra
- namesarver: indiquem un servidor del que serem clients. A la configuració Standard només trobarem sentències d'aquest tipus.
  Per tant, si instal·lem un servidor DNS a la nostra màquina, li haurem d'afegir una sentència nameserver 127.0.0.1

BIND:

No tothom necessitarà instal·lar-se un servidor de noms. Per exemple, les màquines domèstiques que només accedeixin a webs no necessitaran instal·lar-se aquesta eina. Normalment, la instal·lació d'un servidor de noms anirà lligada a la instal·lació d'un servidor web. Tota xarxa haurà de tenir almenys un servidor de noms.

Quan instal·lem un servidor web, necessitarem tenir instal·lats dos servidors de noms, un primari i un secundari. La configuració més estesa consisteix en tenir el servidor primari configurat amb un servidor de noms a la màquina i un servidor secundari en un altre lloc al que redirigirem les consultes si falla el primari.

Instal·lació de bind:

A Debian, podem utilitzar apt-get, que ens instal·larà automàticament els paquets necessaris.

A Red Hat, podrem instal·lar el paquet corresponent (bind<versió>).

Fent rpm –q bind podrem comprovar si ja el tenim instal·lat o no.

A Fedora Core 7 també podrem fer: yum install bind

Instal·larem la versió 9.4 de bind, que serà la que portarà Red Hat.

Un cop instal·lat, el podrem arrencar i aturar fent:

```
/etc/rc.d/init.d/named start
/etc/rc.d/init.d/named stop
/etc/rc.d/init.d/named restart
```

Això no funciona si no tenim generat el fitxer de configuració (/etc/named.conf).

Això despertarà el dimoni named. Aquest dimoni, el primer que farà serà llegir el fitxer de configuracions per esbrinar quin ha de ser el seu funcionament.

per comprovar que s'ha connectat, podrem realitzar un ping a una IP coneguda
També podríem executar l'eina nslookup. Aquesta eina ens permet generar accessos als servidors definits a resolv.conf. Li indicarem un nom de servidor o una IP i obtindrem una resposta.

Si realitzem dues vegades seguides la mateixa crida a nslookup, podrem observar que a la segona ens indicarà 'Non-authoritative answer:...'. Això indica que la resposta s'ha obtingut de la caché del servidor DNS.

Configuració de bind:

Les versions anteriors a la 8 tenien les configuracions a l'arxiu /etc/named.boot. En canvi, des de la versió 9, l'arxiu de configuracions es diu /etc/named.conf.

Per poder arrencar i/o aturar el servei caldrà tenir generat el fitxer /etc/named.conf.

Existeix un script (named.bootconf.pl) que converteix automàticament les configuracions del format antic al nou. En el nou format hi ha noves directives que permetran suportar actualitzacions dinàmiques i millores de rendiment.

Per fer que una modificació a l'arxiu de configuracions tingui efecte, caldrà reinicialitzar bind fent renamed.

La versió que ens arriba de named.conf originalment és la següent:

```
// generated by named-bootconf.pl

options {
      directory "/var/named";
      /*
       * If there is a firewall between you and nameservers you want
       * to talk to, you might need to uncomment the query-source
       * directive below.  Previous versions of BIND always asked
       * questions using port 53, but BIND 8.1 uses an unprivileged
       * port by default.
       */
      // query-source address * port 53;
};

//
// a caching only nameserver config
//
controls {
      inet 127.0.0.1 allow { localhost; } keys { rndckey; };
};
zone "." IN {
      type hint;
      file "named.ca";
};

zone "localhost" IN {
      type master;
      file "localhost.zone";
      allow-update { none; };
};

// zona inversa
zone "0.0.127.in-addr.arpa" IN {
      type master;
      file "named.local";
      allow-update { none; };
};

include "/etc/rndc.key";
```

La primera sentència (options) permet definir opcions. En aquest cas, la única opció que tenim definida és la que ens indica a on es trobaran els arxius de zona (directory).

Posteriorment, tenim un bloc (zone) per a cada zona en el que indiquem el nom de la zona, el seu tipus, el fitxer a on es troben les dades i, finalment, si permet modificacions o no.

Destacar que la zona inversa necessita que es posin les IPs al revés.

Veiem detalladament el format de l'arxiu de configuracions named.conf:

El format general és:

```
sentència {
        opcions de la sentència;
        };
```

Els comentaris es poden marcar amb //,/* o amb #

Sentència acl:

Permet definir adreces per ser utilitzades posteriorment en altres sentències. Si li posem '!' al davant d'un nom o una IP, estarem indicant que aquest no s'ha d'incloure.
Ja hi ha diverses acl predefinides:

- any: permís a tots els hosts
- none: no es dona permís a cap host
- localhost: permís només a adreces locals
- localnets: permís per a tots els hosts de les xarxes locals

Exemple:

```
acl miacl {
        !192.168.20.20;192.168.30.30;
};
```

Sentència logging:

Permet configurar els logs que generarà el servidor DNS.
Si no s'especifica aquesta sentència, s'utilitzarà el log per defecte, que contindrà missatges importants (errors,...)

```
logging{
        category default {null};
};
```

provocaria que tots els missatges sense categoria (default) es despreciéssin

Utilitzant channel podrem definir accions més concretes a realitzar davant de certs events:

```
channel default_syslog{
        syslog daemon;   // identifica la facility daemon (podria ser una altra)
        severity info;
};
```

channel per defecte que envia al sistema syslog els missatges generats pel DNS de nivell info o superior

```
channel warning{
        file "/var/log/avisos";
        severity warning;
};
```

Envia els missatges d'avís o de nivell superior al fitxer especificat

Sentència options:

Permet configurar opcions globals del servidor:

- directory <path>: indica el camí a aplicar a tots els arxius mencionats a l'arxiu de configuracions
- notify yes/no: fer que el servidor primari enviï o no missatges NOTIFY als secundaris que es produeixi algun canvi
- recursion yes/no: indiquem si el servidor farà cerques per tot l'arbre DNS per obtenir la traducció o aquesta feina l'haurà de fer el client
- forwarders {llista de IPs}: llista de servidors als que reenviarem les sol·licituds que el servidor no sigui capaç de resoldre
- forward first/only: indica si es consultarà primer als servidors reenviadors abans de mirar la informació local o s'anirà directament a consultar als reenviadors sense consultar la informació local
- allow-query {llista de IPs}: permet limitar les màquines a les que se'ls permetrà que ens facin consultes. Per defecte, no hi ha limitació

- allow-transfer{llista de IPs}: permet limitar les màquines que podran realitzar transferències de zona. Aquestes seran els servidors secundaris. Els autoritzem per actualitzar les dades des del servidor primari (que som nosaltres)

```
options {
        directory "/var/named";
        allow-transfer "192.168.100.1";
};
```

permetem que transfereixi els arxius de base de dades únicament a la IP 192.168.100.1. Aquesta sentència estarà relacionada amb els servidor secundaris, com ja veurem més endavant.

- allow-update{llista de IPs}: màquines de les que s'acceptaran actualitzacions dinàmiques. Per defecte no se'n accepten. També està relacionat amb els servidors secundaris.

Sentència zone:

Permet definir una zona. La sintaxi és:

zone <nom zona>{
type <tipus>;
file <path fitxer>;
};

path_fitxer (combinat amb directory) indica el fitxer que conté els registres de la zona
tipus: "master" indica que el servidor és primari (conté els arxius de zona). "slave" indica que el servidor és secundari, per tant, els arxius de zona que conté són còpia dels d'algun servidor primari.

Dins d'una definició de zona també podem posar directives com les de option que provocaran un comportament diferent per a aquella zona (allow-query, allow-transfer, notify, allow-update,...)

A cada sentència zone definim un arxiu a on s'especificaran les característiques d'aquell domini. Anem a veure l'estructura d'aquests fitxers:

Arxius de registres de zona:

Totes les línies que comencin amb ';' seran línies de comentari.
Els registres de l'arxiu de zona s'anomenen registres de recursos. El seu format és el següent:

[domini] [ttl] [classe] tipus dades

domini: indica el domini al que ens referim, relatiu a la zona. Amb @ marcarem l'origen o arrel de la zona
ttl: temps pel que és vàlid un registre dins de la caché d'un servidor que es descarregui la informació
classe: Amb TCP/IP sempre posarem 'IN', que és el valor corresponent a una adreça IP
tipus: aquest és el camp més important i el desglossarem a continuació
dades: dependrà del tipus de registre

Si a domini, ttl o classe no indiquem cap valor, s'assumirà el mateix valor que el del darrer registre.

Tipus:

SOA: start of authority. Descriu les característiques i configuracions dels registres que conformaran la zona. Cada fitxer de base de dades de zona haurà de contenir una sentència SOA. Les dades que es poden especificar són:
- origen: nom FQDN del servidor de noms responsable de la zona. És molt important que acabi en '.' per indicar que és un nom absolut i no relatiu
- contacte: adreça de correu de contacte de l'administrador del servidor DNS. Caldrà substituir la @ per '.'
- Serial: número de versió de l'arxiu. Aquest número s'haurà d'anar modificant a cada modificació que es realitzi a l'arxiu. Es recomana format AAAAMMDDnn. Aquest número l'utilitzaran els servidor secundaris per esbrinar si cal que es tornin a copiar la informació del servidor primari o no. No oblidar d'incrementar aquest nombre a cada modificació que es realitzi a l'arxiu de zona !!!
- Refresh: indica cada quant temps han de comprovar els secundaris si hi ha hagut algun canvi en el primari

- Retry: indica el temps que ha d'esperar un secundari per tornar a intentar copiar dades del primari quan la còpia anterior li hagi fallat
- Minimum: temps que serà vàlida una còpia que els secundaris hagin pogut realitzar del primari quan aquests no aconsegueixin tornar a connectar amb el primari

Exemple:

```
@                   1D IN SOA   @ root (
                                42          ; serial (d. adams)
                                3H          ; refresh
                                15M         ; retry
                                1W          ; expiry
                                1D )        ; minimum
```

A: associa adreces IP amb noms de màquina. Al camp de dades hi posem la adreça IP. El nom de la màquina es sobreentén que és el nom oficial

Exemple:

```
pcl     IN          A       192.168.1.1
```

CNAME: Associa un alias amb un nom canònic(un registre de tipus A)

Exemple:

```
Mail    IN          CNAME       pc1
```

PTR: associa una adreça IP d'una màquina amb el seu nom per a realitzar recerques inverses

Exemple:

```
                    IN          PTR         pc1
```

```
;
        IN      NS      dns.sudominio.com.

5               IN              PTR             www.sudominio.com.
10              IN              PTR             dns.sudominio.com.
20              IN              PTR             mail.sudominio.com.
```

Els nombres corresponen al final de la IP. És molt important posar el domini **sencer** i **acabat n '.'**, ja que així s'entén que són noms de domini complerts, ja que en cas contrari es completarien

NS: indica els servidors primaris i secundaris que mantenen els registres de zona. Cal definir un registre NS pel propi nom local:

```
                    IN          NS          ns.ElMeuDomini.local
```

MX: indica el servidor de correu de la zona. Si hi ha més d'un servidor de correu, haurem d'indicar la prioritat de cadascun d'ells. Caldrà definir una sentència de tipus A pels noms indicats aqui

Exemple:

```
                    IN          MX          10      mail1
                    IN          MX          20      mail2
```

Tindrem sentències A definides per mail1 i mail2. La prioritat 10 farà que primer es tingui en compte mail1 i, si no funciona, agafarà mail2

Tot arxiu de registres de base de dades hauran de complir:

- És obligatori que l'arxiu comenci amb una sentència SOA i tingui almenys una sentència NS. La resta és opcional
- L'ordre serà: SOA, NS, MX, A, CNAME

Servidor de noms primari:

Anem a veure els passos a donar si volem configurar una el nostre servidor DNS per fer que reconegui una zona que anomenarem thico.org:

Volem que el nostre servidor ens permeti resoldre els noms:

www.thico.org	192.168.8.8
dns.thico.org	192.168.9.9

A l'arxiu de configuració (/etc/named.conf), caldrà que hi afegim les següents zones:

```
zone "thico.org" IN {
      type master;
      file "thico.db";
      allow-update { none; };
};

zone "162.198.in-addr.arpa" IN {
      type master;
      file "thico.inv";
      allow-update { none; };
};
```

És molt important que el mateix nom que donem a la zona (tant en la normal com en la inversa, quadri amb el que després indiquem a l'arxiu de definició de la zona!!

A més a més, caldrà que definim els arxius corresponents a aquestes dues zones. Per tant, crearem:

1) L'arxiu de base de dades corresponent a la zona a la que donem servei:

/var/named/chroot/var/named/thico.db:

```
$TTL  86400
@           IN SOA      thico.org.        root (
                              427          ; serial (d. adams)
                              3H           ; refresh
                              15M          ; retry
                              1W           ; expiry
                              1D )         ; minimum

              IN NS           @

www         IN A            198.162.8.8
dns         IN A            198.162.9.9
```

2) L'arxiu de base de dades correspondent a la zona inversa, per permetre que a partir de la IP puguin obtenir el nom:

/var/named/chroot/var/named/thico.inv:

```
$TTL  86400
@       IN      SOA     thico.org. root  (
                                    1997022701 ; Serial
                                    28800      ; Refresh
                                    14400      ; Retry
                                    3600000    ; Expire
                                    86400 )    ; Minimum
              IN      NS      @

8.8     IN PTR          www.thico.org.
9.9     IN PTR          dns.thico.org.
```

Les configuracions de la zona inversa han d'anar a l'arxiu corresponent (en aquest cas thico.inv), ja que si les intentem ficar al mateix arxiu que la definició de zona (thico.db) no funcionarà.

No ens oblidarem de fer un restart de l'eina Bind per assegurar que les configuracions tenen efecte.

Comprovacions sobre la configuració:

Per comprovar que el servidor de noms funciona, haurem d'habilitar un client:

Posarem la IP del servidor a la clàusula **"nameserver"** de l'arxiu **/etc/resolv.conf** del client

Comprovarem les configuracions realitzades utilitzant l'eina nslookup.

N'hi haurà prou amb fer "nslookup **www.thico.org**". És a dir, passar-li el nom que volem resoldre. Aquesta eina ens tornarà la IP que ha resolt i també ens indicarà la IP del servidor que l'ha resolt.

També podríem entrar dins de nslookup. Entrem al seu prompt '>' i, a partir d'aquí, podrem anar-li fent consultes:

```
>set q=ns              // ens mostrarà els ns (servidors de noms)
> set q=mx             // ens mostrarà els mx (servidors de correu)
> set q=any            // ens mostrarà tota la informació de dominis
>elmeudomini.com      // ens mostrarà la IP que correspon al DNS
>127.0.0.1            // ens mostrarà el DNS corresponent
```

Servidor de noms secundari:

El primer que caldrà fer és declarar el servidor secundari dins els arxius del primari:

A l'arxiu de configuració del servidor primari afegirem:

```
$TTL  86400
@       IN      SOA     thico.org. root  (
                                  1997022701 ; Serial
                                  28800      ; Refresh
                                  14400      ; Retry
                                  3600000    ; Expire
                                  86400 )    ; Minimum
              IN      NS      @
            IN      NS      dns2.thico.org.

8.8     IN PTR          www.thico.org.
9.9     IN PTR          dns.thico.org.
```

Cal tenir en compte que si hem configurat el servidor primari per permetre transferències de zona des de certs servidors secundaris, caldrà afegir-hi el nou (opció allows-transfer de la sentència options) a l'arxiu /etc/named.conf del servidor primari:

```
options {
      directory "/var/named";
      dump-file "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
      allow-transfer
            {192.168.1.2;};
        notify yes;
      /*
       * If there is a firewall between you and nameservers you want
       * to talk to, you might need to uncomment the query-source
       * directive below.  Previous versions of BIND always asked
       * questions using port 53, but BIND 8.1 uses an unprivileged
       * port by default.
       */
       // query-source address * port 53;
};
```

Amb aquestes modificacions ja tenim preparat el servidor primari per a tenir en compte al nou secundari que estem definint.

Anem a veure l'arxiu de configuració (/etc/named.conf) del que farà de servidor secundari, al que afegirem:

```
zone "thico.org" IN {
      type slave;
      masters{
            192.168.1.1;  IP del servidor primari
            };
};

;  zona inverse:

zone "162.198.in-addr.arpa" IN {
      type slave;
      masters{
            192.168.1.1;   IP del servidor primari

            };
};
```

Un cop fet això, reiniciarem els dos (primari i secundari) i, a partir de llavors, el secundari guardarà una còpia dels arxius de base de dades del primari. Aquesta còpia s'actualitzarà cada vegada que el primari modifiqui la seva base de dades.

Per comprovar aquest funcionament no haurem de buscar la còpia d'aquests arxius, ja que el secundari no en fa una còpia al seu sistema de fitxers amb el mateix nom, sinó que els guarda de forma que no ens indica ni el format ni el nom dels fitxers en que els guarda.

Per tant, per comprovar que el servidor secundari funciona, la millor opció serà modificar l'arxiu (/etc/resolv.conf) del secundari per eliminar la IP del primari o fer un stop d'aquest primari i veure que en fer nslookup, el nom se'ns està resolent i, a més, l'està resolent el secundari. Per tant, aquest té una còpia correcta dels arxius del primari i ha aconseguit resoldre el nom sense el primari.

Accés des de windows:

Des d'una màquina windows, podem configurar una connexió (conexiones de red) posant-li com a servidor DNS una màquina Linux que tinguem fent de servidor DNS. Un cop fet això, a la màquina windows, executarem des de l'entorn MS-DOS l'eina **nslookup www.thico.org** tal qual ho faríem a la prova anterior des de linux.