

Introducción al DNS

Javier Smaldone (<http://blog.smaldone.com.ar>)
5 de diciembre de 2006

Introducción

Este artículo tiene como objetivo realizar una introducción al funcionamiento del "*Sistema de Nombres de Dominio*" (**DNS**, por "*Domain Name System*").

El **DNS** se utiliza principalmente para la *resolución de nombres*, esto es, decidir qué *dirección IP* pertenece a determinado nombre completo de *host*.

La versión original (online) de este artículo, además de los comentarios de los lectores, puede encontrarse en: <http://blog.smaldone.com.ar/2006/12/05/como-funciona-el-dns/>.

Licencia

Este tutorial se distribuye bajo una Licencia Creative Commons Atribución-No Comercial-Compartir Obras Derivadas Igual 2.5 Argentina.

Usos del DNS

El **DNS** se utiliza para distintos propósitos. Los más comunes son:

- **Resolución de nombres:** Dado el nombre completo de un *host* (por ejemplo *blog.smaldone.com.ar*), obtener su *dirección IP* (en este caso, *208.97.175.41*).
- **Resolución inversa de direcciones:** Es el mecanismo inverso al anterior. Consiste en, dada una *dirección IP*, obtener el nombre asociado a la misma.
- **Resolución de servidores de correo:** Dado un *nombre de dominio* (por ejemplo *gmail.com*) obtener el servidor a través del cual debe realizarse la entrega del correo electrónico (en este caso, *gmail-smtp-in.l.google.com*).

Por tratarse de un sistema muy flexible, es utilizado también para muchas otras funciones, tales como la obtención de claves públicas de cifrado asimétrico y la validación de envío de e-mails (a través de mecanismos como SPF).

Terminología básica

Antes de proseguir, es necesario introducir algunos términos básicos para evitar confusiones y ambigüedades. Otros términos más complejos serán tratados más adelante.

- **Host Name:** El nombre de un *host* es una sola "*palabra*" (formada por letras, números y guiones). Ejemplos de nombres de *host* son "*www*", "*blog*" y "*obelix*".
- **Fully Qualified Host Name (FQHN):** Es el "*nombre completo*" de un *host*. Está formado por el

hostname, seguido de un punto y su correspondiente *nombre de dominio*. Por ejemplo, "*blog.smaldone.com.ar*"

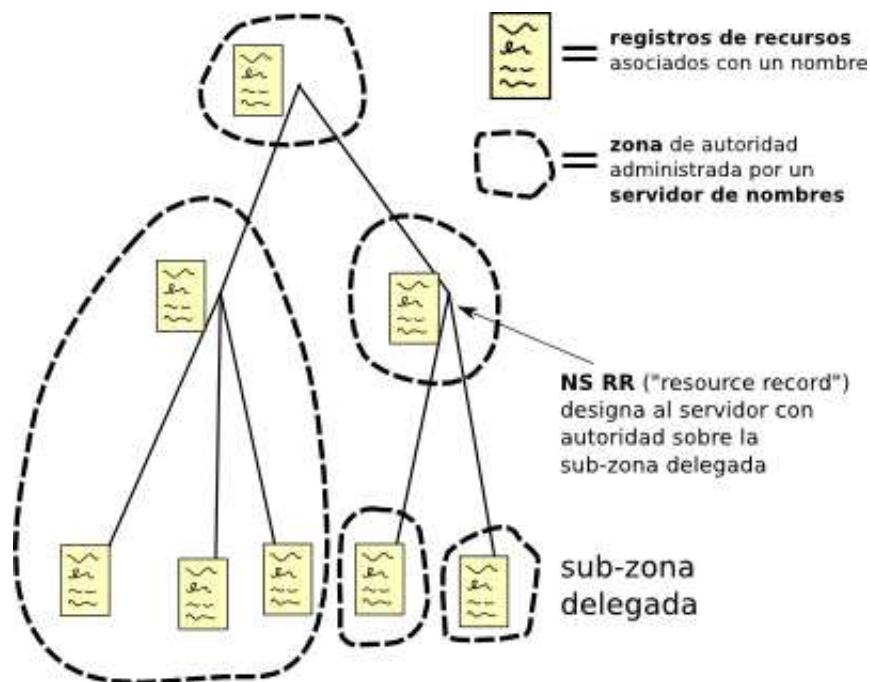
- **Domain Name:** El *nombre de dominio* es una sucesión de nombres concatenados por puntos. Algunos ejemplos son "*smaldone.com.ar*", "*com.ar*" y "*ar*".
- **Top Level Domains (TLD):** Los *dominios de nivel superior* son aquellos que no pertenecen a otro dominio. Ejemplos de este tipo son "*com*", "*org*", "*ar*" y "*es*".

Arquitectura del DNS

El sistema DNS funciona principalmente en base al protocolo **UDP**. Los requerimientos se realizan a través del *puerto 53*.

El sistema está estructurado en forma de "*árbol*". Cada *nodo* del árbol está compuesto por un grupo de servidores que se encargan de resolver un conjunto de dominios (*zona de autoridad*). Un servidor puede delegar en otro (u otros) la autoridad sobre alguna de sus sub-zonas (esto es, algún *subdominio* de la zona sobre la que él tiene autoridad). Un subdominio puede verse como una *especialización* de un dominio de nivel anterior. Por ejemplo, "*smaldone.com.ar*" es un subdominio de "*com.ar*", que a su vez lo es del **TLD** "*ar*".

El siguiente diagrama ilustra esto a través de un ejemplo:



Los servidores con autoridad sobre los **TLD** son los llamados "*root servers*" (o "*servidores raíz*") del sistema. Estos son fijos, ya que rara vez cambian, siendo actualmente 13.

Tomemos como ejemplo el dominio "*com.ar*". Este dominio pertenece al **TLD** "*ar*".

Los servidores con autoridad sobre el dominio "*ar*" son:

```
ns-ar.ripe.net
merapi.switch.ch
uucp-gw-1.pa.dec.com
uucp-gw-2.pa.dec.com
ns.uu.net
ns1.retina.ar
athea.ar
ctina.ar
```

En tanto que los servidores con autoridad sobre "*com.ar*" son:

```
merapi.switch.ch
relay1.mecon.gov.ar
ns.uu.net
ns1.retina.ar
athea.ar
ctina.ar
```

Podemos ver que *ns.uu.net*, *ns1.retina.ar*, *athea.ar* y *ctina.ar* tienen autoridad tanto sobre "*com.ar*" como sobre "*ar*".

El proceso de resolución de nombres

Cuando una aplicación (cliente) necesita resolver un **FQHN** envía un requerimiento al *servidor de nombres* configurado en el sistema (normalmente, el provisto por el **ISP**). A partir de entonces se desencadena el proceso de resolución del nombre:

1. El servidor de nombres inicial consulta a uno de los *servidores raíz* (cuya *dirección IP* debe conocer previamente).
2. Este devuelve el nombre del servidor a quien se le ha delegado la sub-zona.
3. El servidor inicial interroga al nuevo servidor.
4. El proceso se repite nuevamente a partir del punto 2 si es que se trata de una sub-zona delegada.
5. Al obtener el nombre del servidor con autoridad sobre la zona en cuestión, el servidor inicial lo interroga.
6. El servidor resuelve el nombre correspondiente, si este existe.
7. El servidor inicial informa al cliente el nombre resuelto.

Ilustremos esto con un ejemplo concreto. Supongamos que el navegador necesita resolver el nombre "*blog.smaldone.com.ar*".

1. El sistema tiene configurado el servidor de nombres *200.49.156.3* (perteneciente al proveedor argentino *Fibertel*). Por lo tanto envía a éste el requerimiento de resolver "*blog.smaldone.com.ar*".
2. El servidor de *200.49.156.3* envía la consulta *root server 198.41.0.4*.

3. 198.41.0.4 le informa que el servidor con autoridad sobre "ar" es *athea.ar*, cuya **dirección IP** es 200.16.98.2. (En realidad, informa la lista de todos los servidores con tal autoridad, pero para simplificar el ejemplo tomaremos solamente uno.)
4. 200.49.156.3 envía nuevamente el requerimiento a *athea.ar* (el cual, recordemos, también tiene autoridad sobre "com.ar").
5. *athea.ar* responde que la autoridad sobre *smaldone.com.ar* la tiene *ns1.mydomain.com* cuya **dirección IP** es 64.94.117.213.
6. 200.49.156.3 envía ahora la consulta a *ns1.mydomain.com*.
7. *ns1.mydomain.com* informa que la **dirección IP** de "blog.smaldone.com.ar" es 208.97.175.41.
8. Finalmente, 200.49.156.3 devuelve este resultado a la aplicación que originó la consulta.

Mecanismos de caché

Cada vez que un servidor de nombres envía una respuesta, lo hace adjuntando el tiempo de validez de la misma (**TTL** o "*tiempo de vida*"). Esto posibilita que el receptor, antes la necesidad de volver a resolver la misma consulta, pueda utilizar la información previamente obtenida en vez de realizar un nuevo requerimiento.

Esta es la razón por la cual los cambios realizados en el **DNS** no se propagan instantáneamente a través del sistema. Dependiendo de la naturaleza de los mismos (y de la configuración de cada servidor), la propagación puede tardar desde algunos minutos hasta varios días.

Correo electrónico y resolución de nombres

Normalmente los usuarios de correo electrónico redactan su mensajes usando un cliente de correo y enviándolo a través de un servidor **SMTP** provisto por su **ISP** o a través de un sistema de *correo vía web* (*webmail*). En cualquier caso, una vez que el mensaje es recibido por el servidor, debe ser entregado al destinatario. Aquí interviene el sistema **DNS**:

1. El servidor del emisor solicita al **DNS** (de acuerdo al mecanismo analizado anteriormente), la entrada **MX** del dominio del receptor del mensaje. **MX** significa "*mail exchanger*", esto es, el nombre del servidor (o los servidores) encargado de recibir los mensajes destinados a determinado dominio.
2. El **DNS** devuelve el **FQHN** y la **dirección IP** del *mail exchanger*.
3. El servidor del emisor se conecta al puerto 25, mediante **TCP**, del servidor del destinatario y entrega el mensaje según el protocolo **SMTP**.
4. El proceso podrá continuar si el servidor receptor del mensaje no es el último de la cadena. Existen servidores que actúan como "*puertas de enlace*" o "*gateways*" de correo electrónico, y que se encargan de recibir los mensajes de determinados dominios para luego enviarlos a otros servidores.

Tipos de registro en un servidor de nombres

Un servidor de nombres puede almacenar distinta información. Para ello, en cada zona de autoridad dispondrá de entradas de distinto tipo. Entre los más importantes se encuentran:

- **A (Address)**: Este registro se utiliza para traducir nombres de *hosts* del dominio en cuestión a direcciones IP.
- **CNAME (Canonical Name)**: El *nombre canónico* es un alias para un *host* determinado. (No define una *dirección IP*, sino un nuevo nombre.)
- **NS (Name Server)**: Especifica el servidor (o servidores) de nombres para un dominio.
- **MX (Mail Exchange)**: Define el servidor encargado de recibir el correo electrónico para el dominio.
- **PTR (Pointer)**: Especifica un "*registro inverso*", a la inversa del registro **A**, permitiendo la traducción de *direcciones IP* a nombres.
- **TXT (Text)**: Permite asociar información adicional a un dominio. Esto se utiliza para otros fines, como el almacenamiento de claves de cifrado, "*DomainKeys*" o "*Sender Policy Framework*".

Bind, "el" servidor de nombres

Prácticamente el único software utilizado en los servidores de nombres de *Internet* es **bind** ("*Berkeley Internet Name Domain*"), creado originalmente en la *Universidad de California*, y actualmente propiedad del *Internet Systems Consortium*.

Este programa, distribuido bajo una licencia libre, es utilizado en prácticamente todos los sistemas *Unix* del mundo. Esto ha sido considerado un problema de seguridad, al punto que se ha propuesto la migración de algunos *root servers* a otro sistema, ya que la aparición de algún problema de seguridad en **bind** podría implicar la caída de todo el **DNS** de *Internet*.

Uso del DNS en una red local

Ya en redes de tamaño medio (quizás más de 5 equipos) es conveniente la utilización de **DNS**. Esto nada tiene que ver con el **DNS** de *Internet* (aunque el servidor local puede estar vinculado a este sistema).

Básicamente, es conveniente montar un servidor local de **DNS** por los siguientes motivos:

- **Agilizar el acceso a Internet**: Al tener un *servidor de nombres* en nuestra propia red local (que acceda al **DNS** de nuestro proveedor o directamente a los *root servers*) se agiliza el mecanismo de resolución de nombres, manteniendo en *caché* los nombres recientemente usados en la red y disminuyendo el tráfico hacia/desde *Internet*.
- **Simplificar la administración de la red local**: Al contar con un **DNS** propio (ya sea uno o varios servidores de nombres) es posible definir *zonas locales* (no válidas ni accesibles desde *Internet*) para asignar nombres a cada uno de los *hosts* de la **LAN**. De esta forma es posible, por ejemplo, referirnos a la impresora de red como "*hplaser.mired.local*" en vez de "*192.168.0.2*" y a nuestro servidor de correo interno como "*smtp.mired.local*" en vez de "*192.168.0.3*". (Pensemos, por ejemplo, que ocurriría con las configuraciones de las aplicaciones si un día decidimos cambiar el esquema de *direcciones IP* de nuestra red.)

Problemas del DNS

El principal problema que presenta el **DNS** es que, al estar basado en **UDP** (protocolo de transporte que no garantiza la recepción de la información enviada), tanto las consultas como las respuestas pueden "*perderse*" (por ejemplo, a causa de congestionamiento en algún enlace de la red). Es común apreciar cómo, en el caso de servidores y redes no muy bien configuradas, la resolución de nombres se resiente sensiblemente ante cualquier anomalía (saturación de tráfico o del servidor de nombres local).

Otro inconveniente, que ya hemos hecho notar, es la lentitud de la propagación de las modificaciones en el sistema, producto de la propia arquitectura del mismo.

Pero quizás el mayor problema no sea inherente al sistema mismo, sino a la pésima configuración de los servidores de muchos **ISP**. *Fibertel*, el proveedor que utilizo, es un notable ejemplo de esta falencia. Una buena solución a esta situación es ejecutar un *servidor de nombres* en alguna PC de la red local, de forma tal que se comunique directamente con los **root servers** (evitando de esta forma pasar a través de los servidores de nombres de nuestro proveedor).

Herramientas para aprender más

En sistemas **Unix** el comando **dig** (ver "**man dig**") permite realizar requerimientos "*a mano*" para poder investigar un poco más sobre el funcionamiento del **DNS** y, cómo no, también para detectar y solucionar problemas en la red.

Los usuarios de sistemas **Windows** disponen del comando **nslookup** (aunque no tan potente como **dig**), para el mismo propósito.

Lectura adicional

- La página de Wikipedia sobre **DNS** contiene bastante información y buenos enlaces sobre este tema.
- El "*DNS Cómo*" explica la configuración de **bind** en **GNU/Linux**.
- El **RFC 1591** explica detalladamente la estructura del **DNS**.
- Los **RFC 1034** y **1035** (ambos en inglés), describen completamente el **DNS**.