UNIVERSIDAD DEL AZUAY

FACULTAD DE CIENCIAS DE LA ADMINISTRACION

ESCUELA DE INGENIERÍA DE SISTEMAS

Título: "DNS "Sistema De Resolución De Nombres" en Redes IP conceptos y aplicación**"**

Monografía previa la obtención del Título de Ingeniero de Sistemas

Autor: Juan Diego Pesántez Palacios

CUENCA – ECUADOR

2004

Dedicatoria

Este trabajo está dedicado a todos quienes me brindaron su apoyo y cariño, a mis padres y hermanas siendo ellos el pilar fundamental en las etapas de mi vida, los cuales me dieron la inspiración y la ayuda necesaria para culminar con éxito mis estudios.

Juan Diego

Agradecimiento

Agradezco a la Universidad del Azuay por la Formación académica brindada y al Ing. Francisco Salgado por haber colaborado con su tiempo, orientándome y cultivando un buen fundamento teórico como profesional.

Las ideas y opiniones vertidas en la presente monografía son de exclusiva responsabilidad de su autor.

Juan Diego Pesántez Palacios

INTRODUCCIÓN

Un servidor de dominio (DNS) consta de una base de datos de nombres distribuida (catorce servidores raíz en el mundo).

Los nombres de la base de datos DNS establecen una estructura lógica de árbol, conocida como espacio de nombres de dominio. Cada nodo, tiene un nombre que puede contener subdominios. Estos se agrupan en zonas, que permiten la administración distribuida del espacio de nombres.

El nombre identifica la posición del dominio en la jerarquía lógica de DNS respecto de su dominio principal, al separar cada rama del árbol con un punto.

Los DNS (Servidores de Dominio), facilitan a los usuarios el uso del Internet y las redes corporativas, al no tener que memorizar números; siendo ésta la base de las redes IP.

Para el desarrollo de este tema se han utilizado herramientas en Linux y Windows con el fin de interactuar entre los mismos.

CAPITULO I

1. DNS

1.1 Protocolo DNS

El servidor más famoso de nombres es **BIND**, que viene de The Berkeley Internet Name Domain, porque fue ideado por un estudiante de esa universidad, que tantos progresos ha brindado a la informática. Distintas organizaciones se han estado ocupando de este programa hasta su versión 4.9, pero en Mayo de 1997, dio el salto hacia una nueva versión la 8, porque la versión 4 había quedado excesivamente antigua. La versión 4 ya no se desarrolla más, excepto para parches de seguridad. Y es que la seguridad ha sido siempre el punto débil de este protocolo, protagonista de grandes delitos cibernéticos. La excesiva confianza en el momento de su diseño lo han vuelto débil y fácilmente quebradizo con la tecnología actual en la Internet de hoy.

Un gran avance ocurrió en Febrero de 2003, cuando el país de España fue elegido para alojar un servidor raíz. Esto no es algo demasiado común, teniendo en cuenta que en el planeta sólo existen catorce servidores raíz. "El nuevo root server o Servidor Primario de Dominios, que funcionará en Madrid, a cargo de la asociación Espanix, es una de las dos copias de uno de los trece grandes controladores de la Red en todo el mundo. Son los llamados 'root servers', grandes entramados de cables y circuitos que hacen posible navegar por la red. Diez están en Estados Unidos, uno, en Japón, Reino Unido y Suecia."

1.1.1 Transporte TCP / UDP

Trabaja en la capa de aplicación (se asigna a la capa 7 del modelo OSI). Si el segmento a enviar es menor que 512 Bytes se utiliza el protocolo UDP, de lo contrario El uso de TCP como protocolo de transporte se reduce a dos situaciones concretas: por una parte, para transportas las respuestas mayores de 512 octetos de longitud, cosa poco habitual, pero que sucede con determinados dominios. Si tenemos bloqueado el tráfico TCP hacia y desde el puerto 53 veremos que la resolución de nombres mediante DNS funciona casi siempre, pero falla a la hora de consultar información en determinados dominios.

La otra situación en la que se usa protocolo TCP es en las transferencias de zona desde el servidor primario a los servidores secundarios. Si bien el proceso de transferencia lo inician siempre los secundarios hacia el primario por razones de fiabilidad se establece una conexión TCP desde cada secundario al primario para descargarse una copia de las bases de datos de las zonas DNS para las que es autoritativa.

Los resolvers envían consultas UDP primero a los servidores, para obtener un mejor rendimiento, y sólo acuden a TCP si los datos devueltos están truncados.

1.1.2 Cabecera del protocolo DNS

El "resolver" envía la trama al servidor de nombres. Sólo la cabecera y la sección "question" se utilizan para la consulta. Las respuestas o retransmisiones de las consultas usan la misma trama, pero son llenadas por más secciones de la misma (las secciones "answer/authority/additional").

-. ARcount. Un entero sin signo de 16 bits que especifica el número de RRs en la sección "additional records".

-. ANcount. Un entero sin signo de 16 bits que especifica el número de RRs en la sección "answer".

-. NScount. Un entero sin signo de 16 bits que especifica el número de RRs en la sección "authority".

-. QDcount. Un entero sin signo de 16 bits que especifica el número de entradas en la sección "question".

IDENTIFICACION	PARAMETROS
Qdcount	Ancount
Nscount	Arcount
Sección Pregunta	
Sección Respuesta	
Sección Autoritaria	
Sección información Adicional	

-. Op code. Campo de 4-bit que especifica el tipo de consulta:

-. 0 consulta estándar (QUERY).

-. 1 consulta inversa (IQUERY).

-. 2 solicitud del estado del servidor (STATUS).

-. Se reservan los otros valores para su uso en el futuro.

-. RA. Flag de recursividad disponible. Indica si el servidor de nombres soporta resolución recursiva.

-. Rcode. Código de respuesta de 4 bits. Los posibles valores son:

-. 0. Ningún error.

-. 1. Error de formato. El servidor fue incapaz de interpretar el mensaje.

-. 2. Fallo en el servidor. El mensaje no fue procesado debido a un problema con el servidor.

-. 3. Error en nombre. El nombre de dominio de la consulta no existe. Sólo válido si el bit AA está activo en la respuesta.

-. 4. No implementado. El tipo solicitado de consulta no está implementado en el servidor de nombres.

-. 5. Rechazado. El servidor rechaza responder por razones políticas. Los demás valores se reservan para su usuario en el futuro.

-. RD. Flag de recursividad. Este bit se copia en la respuesta e indica al servidor de nombres una resolución recursiva.

-. QR. Flag que indica consulta (0) o respuesta (1).

1.1.3 Historia Dns, Concepto e implementación

El Sistema de nombres de dominio (DNS) es un conjunto de protocolos y servicios sobre una red TCP/IP que permite a los usuarios de la red utilizar nombres amigables jerárquicos cuando busque otros host (es decir, equipos) en lugar de tener que recordar y usar sus direcciones IP. En la actualidad, este sistema cada vez se usa más en Internet y en muchas empresas privadas. Tal es el caso en que alguna vez se ha utilizado un explorador WEB, una aplicación de Telnet, un programa de FTP o cualquier otro programa de TCP/IP parecido de Internet, entonces probablemente haya usado un servidor DNS.

La función más conocida de los protocolos DNS es la asignación de nombres amigables a direcciones IP. Por ejemplo, si la dirección IP del sitio FTP de Tesis.com es **192.168.1.1**, la mayoría de la gente llega a este equipo especificando **ftp.tesis.com** y no la dirección IP, que es menos amigable. Además de ser más fácil de recordar, el nombre es más fiable. La dirección numérica podría cambiar por muchas razones, pero el nombre siempre sirve.

Antes de la implementación de DNS, la utilización de nombres de equipos que fueran amigables se realizaba a través del uso de archivos **HOST**S que contenían una lista de nombres y sus direcciones IP asociadas. Este archivo en Internet, se administraba de forma centralizada y cada ubicación descargaba periódicamente una nueva copia. A medida que fue creciendo el número de equipos en la red de Internet, esta solución se complicó demasiado y surgió la necesidad de ver algo mejor. Esta solución se convirtió en DNS.

Según el Dr. Paul Mockapetris, el diseñador principal de DNS, el objetivo original del diseño de DNS era reemplazar este gran archivo HOSTS, administrado de forma

Singular, por una base de datos distribuida ligera que permitiera la existencia de un espacio jerárquico de nombres, la distribución de la administración, tipos de datos extensibles, tamaño de base de datos virtualmente ilimitado y un rendimiento razonable.

La implementación más conocida del protocolo DNS '**BIND**' se programó en Berkeley, originalmente, para el sistema operativo UNIX BSD 4.3. El nombre 'BIND' significa **Berkeley Internet Name Domain** (Dominio de nombres de Internet de Berkeley).

Ventajas

- Desaparece la carga excesiva en la red y en los hosts: ahora la información esta distribuida por toda la red, al tratarse de una BBDD distribuida.
- No hay Duplicidad de Nombres, el problema se elimina debido a la existencia de dominios controlados por un único administrador. Puede haber nombres iguales pero en dominios diferentes.
- Consistencia de la Información: ahora la información que esta distribuida es actualizada automáticamente sin intervención de ningún administrador.

Nos ayuda a:

- Eliminar el problema de nombres repetidos (a cada organización se le asigna un dominio único
- Elimina el problema de carga y tráfico de red en una sola máquina ya que la información esta distribuida. (Y esta disponible de manera redundante).

1.2 Tipos Servidores

Los servidores DNS almacenan información acerca del espacio de nombres del dominio, y son conocidos como **servidores de nombres**. Los servidores de nombres suelen ser responsables de una o más zonas (entendiendo como zona un archivo físico que almacena registros de la base de datos de una parte del espacio de nombres DNS). El servidor de nombres se dice que tiene autoridad sobre esas zonas. Cuando se configura un servidor de nombres DNS, se indica cuáles son los restantes servidores de nombres DNS que se encuentran en el mismo dominio.

1.2.1 Servidor de nombres principal

Es un servidor de nombres que obtiene los datos de sus zonas de archivos locales. Los cambios en una zona, como la adición de dominios, se realizan en el servidor de nombres principal.

1.2.2 Servidor de nombres secundario

Obtiene los datos de sus zonas de otro servidor de nombres de la Red que tiene autoridad para esa zona (normalmente de un servidor de nombres principal). El proceso de obtención de información de estas zonas (es decir, el archivo de base de datos) por red se conoce como una **transferencia de zona**.

La razón fundamental para la existencia de un servidor de nombres secundario es la de la redundancia. Se necesitan al menos dos servidores de nombres DNS que sirvan cada zona, uno principal y al menos uno secundario, para que en caso de fallo, alguno de ellos responda a las peticiones de nombres.

En el proceso de resolución de nombres, los servidores de nombres almacenan en caché las respuestas obtenidas fuera de su zona para evitar tiempo en la resolución de respuestas a peticiones similares. En este proceso, se realiza la búsqueda a través de la jerarquía de nodos de nombres del DNS hasta encontrar la resolución de la petición. Existe un **tiempo de vida** (TTL Time To Live) que se especifica a través de los datos que se intercambian los servidores de nombres, y que controla el tiempo que se almacenarán estos datos. Evidentemente, a menor tiempo de vida, mayor carga para el servidor de nombres, pero más fiabilidad de los datos del dominio. Para permanecer sincronizado, los servidores de nombres secundarios consultan a los primarios regularmente (típicamente cada tres horas) y reejecutan la transferencia de zona si el primario ha sido actualizado.

1.2.3 Caching (a.k.a. hint)

Son aquellos que no tienen autoridad sobre ningún dominio: se limitan a contactar con otros servidores para resolver las peticiones de los clientes DNS. Estos servidores mantienen una memoria caché con las últimas preguntas contestadas. Cada vez que un cliente DNS le formula una pregunta, primero consulta en su memoria caché. Si encuentra la dirección IP solicitada, se la devuelve al cliente; si no, consulta a otros servidores, apunta la respuesta en su memoria caché y le comunica la respuesta al cliente.

1.2.4 Forwarding (a.k.a. Proxy, cliente, remoto)

El servidor Forwarding (a.k.a. Proxy, cliente, remoto) es uno que transmite simplemente a todas las peticiones otro DNS y deposita los resultados. Sin embargo un forwarding DNS se puede manejar de dos maneras donde está lento o se complica el acceso a una red externa: Al almacenar en el servidor local DNS reduce el acceso externo y acelera respuestas y quita tráfico innecesario. El servidor remoto del DNS proporciona la ayuda recurrente de la pregunta reducción en tráfico a través del acoplamiento, los resultados en una sola pregunta a través de la red.

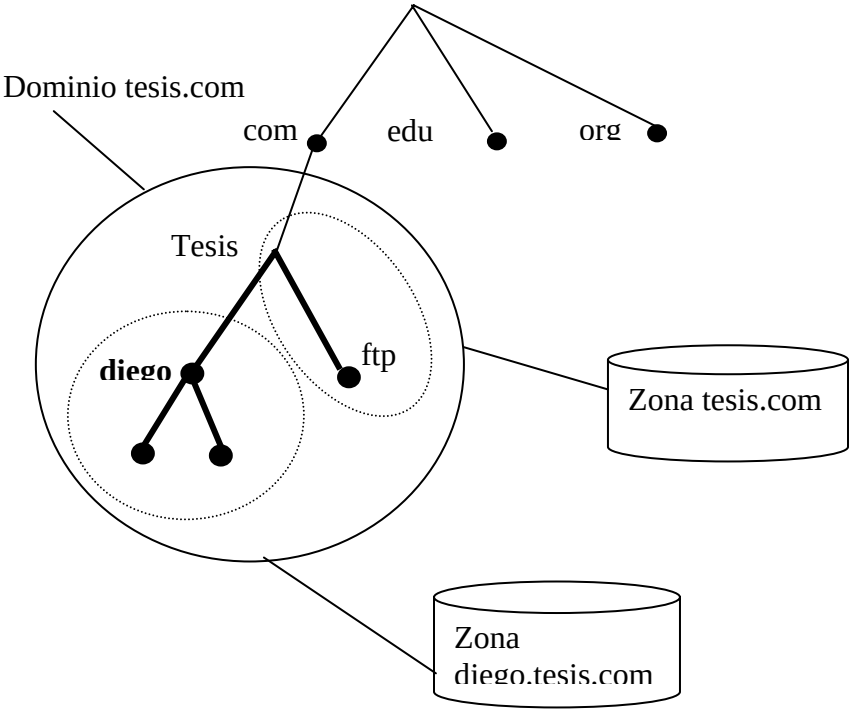
1.3 Dominios

Cada nodo del árbol de una base de datos DNS, junto con todos los nodos por debajo del mismo, se llama **dominio**. Los dominios pueden contener host (**equipos**) y otros dominios (**subdominios**). Por ejemplo, el dominio tesis.com , podría contener a la vez equipos, como diego.tesis.com, y subdominios, como subdom.tesis.com, que a su vez podría contener host, como por ejemplo diego.subdom.tesis.com.

Cada dominio consta de una cadena de menos de 255 caracteres, formada por etiquetas separadas por puntos.

Por norma general, los nombres de host y de dominio tienen restricciones en cuanto a los caracteres que pueden formarlos. Sólo está permitido el uso de los caracteres `a-z', `A-Z', `0-9', y `-'.

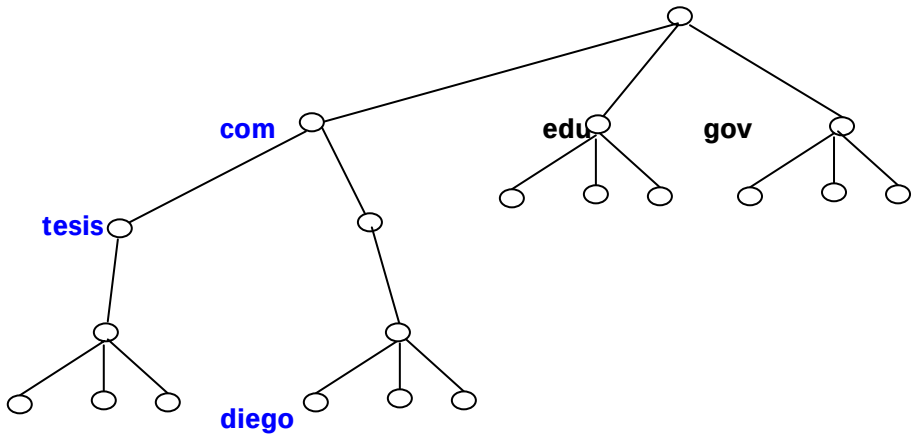
Por lo general existen dominios geográficos y por organización



1.4 Organización y Estructura

Un sistema de nombres de dominio (DNS) consta de una base de datos de nombres distribuida. Los nombres de la base de datos DNS establecen una estructura lógica de árbol, conocida como **espacio de nombres del dominio**. Cada nodo o dominio del espacio de nombres del dominio tiene un nombre y puede contener subdominios. Los dominios y subdominios se agrupan en zonas para permitir la administración distribuida del espacio de nombres.

El nombre del dominio identifica la posición del mismo en la jerarquía lógica del DNS respecto de su dominio principal, al separar cada rama del árbol con un punto. En la siguiente figura se muestran varios dominios superiores, entre los que se encuentra **tesis**, y un host llamado **diego**, dentro del dominio **tesis.com**. Si alguien quisiera contactar con ese host, usarían el nombre completo **diego.tesis.com**.



1.5 Componentes Del Sistema

Para su funcionamiento, el DNS utiliza tres componentes principales:

- **Clientes DNS** (resolvers). Los clientes DNS envían las peticiones de resolución de nombres a un servidor DNS. Las peticiones de nombres son preguntas de la forma: ¿Qué dirección IP le corresponde al nombre nombre.dominio?
- **Servidores DNS** (name servers). Los servidores DNS contestan a las peticiones de los clientes consultando su base de datos. Si no disponen de la dirección solicitada pueden reenviar la petición a otro servidor.
- **Espacio de nombres de dominio** (domain name space). Se trata de una base de datos distribuida entre distintos servidores

1.6 Zona Y archivos de Zona

Una **zona** es alguna parte del espacio de nombres DNS cuyos registros de la base de datos existen y se administran en un archivo determinado de zona. Puede configurarse un único servidor DNS para administrar uno o múltiples archivos de zona. Cada zona está anclada en un determinado nodo del dominio, llamado 'dominio raíz' de la zona. Los archivos de zona no contienen necesariamente todo el árbol (es decir, todos los subdominios) bajo el dominio raíz de la zona.

Es muy importante entender la diferencia entre una zona y un dominio. Una zona es un archivo físico compuesto de registros del recurso que define un grupo de dominios. Un dominio es un nodo del espacio de nombres DNS y todos los subdominios que se encuentran por debajo.

A continuación enumerare algunos de estos registros para ver el formato general de un archivo DNS:

Registro SOA

El primer registro de cualquier archivo de base de datos es el registro SOA. Este registro se forma con una serie de parámetros a tener en cuenta.

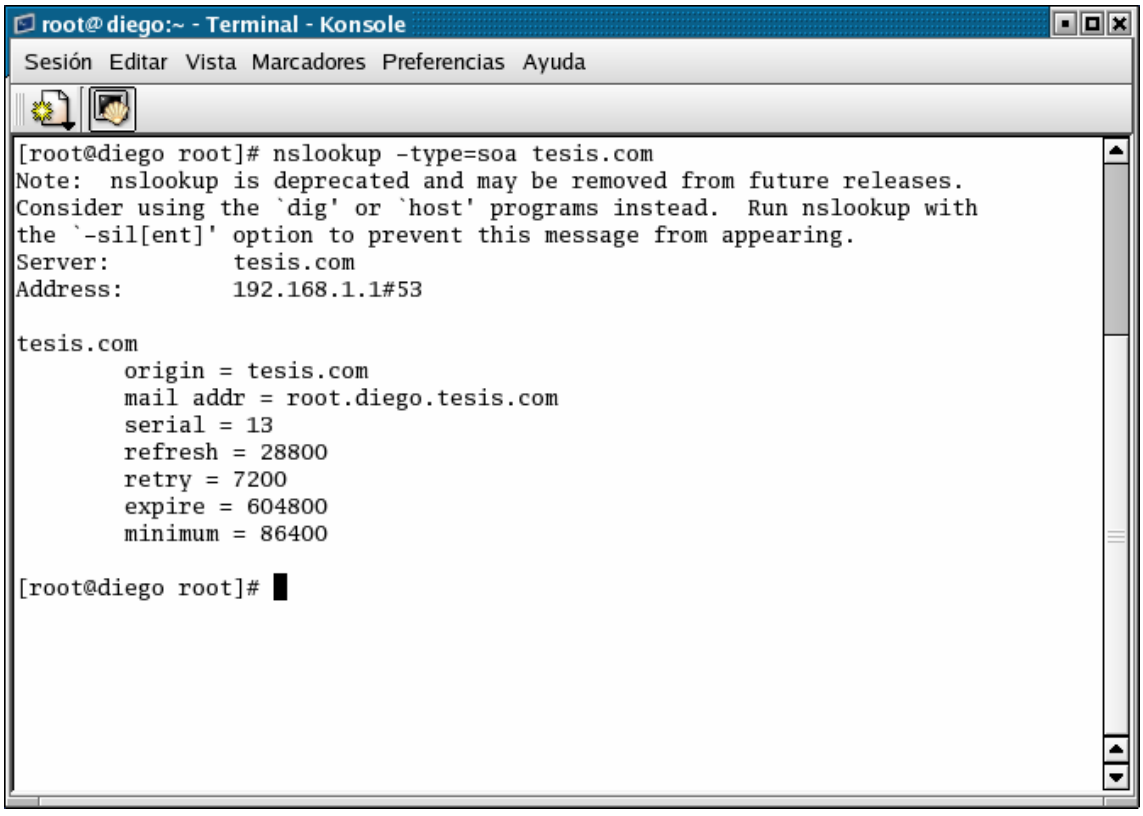
Host origen: El host en el que se mantiene el archivo. **Correo electrónico de contacto**: La dirección de correo electrónico de Internet de la persona responsable del archivo de base de datos de este dominio. No olvide que, en lugar de escribir el símbolo '@' en el nombre de correo electrónico, como se suele hacer, éste se sustituye por un punto cuando se coloca en el archivo de zonas. **Número de serie**: El "número de versión" de este archivo de base de datos. Este número debería aumentar cada vez que cambie el archivo de base de datos. **Tiempo de actualización**: El tiempo (en segundos) que esperará un servidor secundario entre comprobaciones de su servidor maestro para ver si el archivo de base de datos ha cambiado y si hay que pedir una transferencia de zona. **Tiempo de reintento**: El tiempo (en segundos) que esperará un servidor secundario antes de volver a intentar una transferencia de zona que haya fallado. **Tiempo de caducidad**: El tiempo (en segundos) que un servidor secundario seguirá intentando descargar una zona. Cuando haya pasado este tiempo, se rechazará la información antigua de la zona. **Tiempo de vida** (TTL): El tiempo (en segundos) que un servidor DNS tiene permitido acumular en la caché cualquier registro del recurso de este archivo de base de datos.

Hay que tener en cuenta que cualquier nombre de dominio del archivo de base de datos que no termine con un punto, tendrá el dominio raíz anexado al final. Para que un registro de recurso abarque una línea en un archivo de base de datos, los saltos de línea deben incluirse entre paréntesis. En un archivo de zona, el símbolo '@' representa el dominio raíz de la zona.

Ejemplo:

1;	número de serie
10800;	actualizar [3 horas]
3602;	reintentar [1 hora]
604800;	caducar [7 días]
86400;	tiempo de vida [1 día]

Ejemplo:



Registro NS

El Registro NS (**Name Server** o Servidor de Nombres) enumera los servidores de nombres de este dominio, permitiendo que otros servidores de nombres miren los nombres de su dominio.

Registro MX

El Registro de Intercambio de Correo (**Mail exchange**) en mi caso (sendmail bajo Linux Red hat 9.0) indica qué host procesa el correo de este dominio. Si existen múltiples registros de intercambio de correo, el resolver de nombres intentará ponerse en contacto con los servidores de correo en orden de preferencia, empezando por los valores inferiores (mayor prioridad) hasta el valor superior (menor prioridad).

Registro A

Un Registro de dirección A (**Address**) sirve para asociar nombres de host a direcciones IP dentro de una zona. Éstos son los registros que componen la mayor parte del archivo de base de datos. Su formato es el siguiente

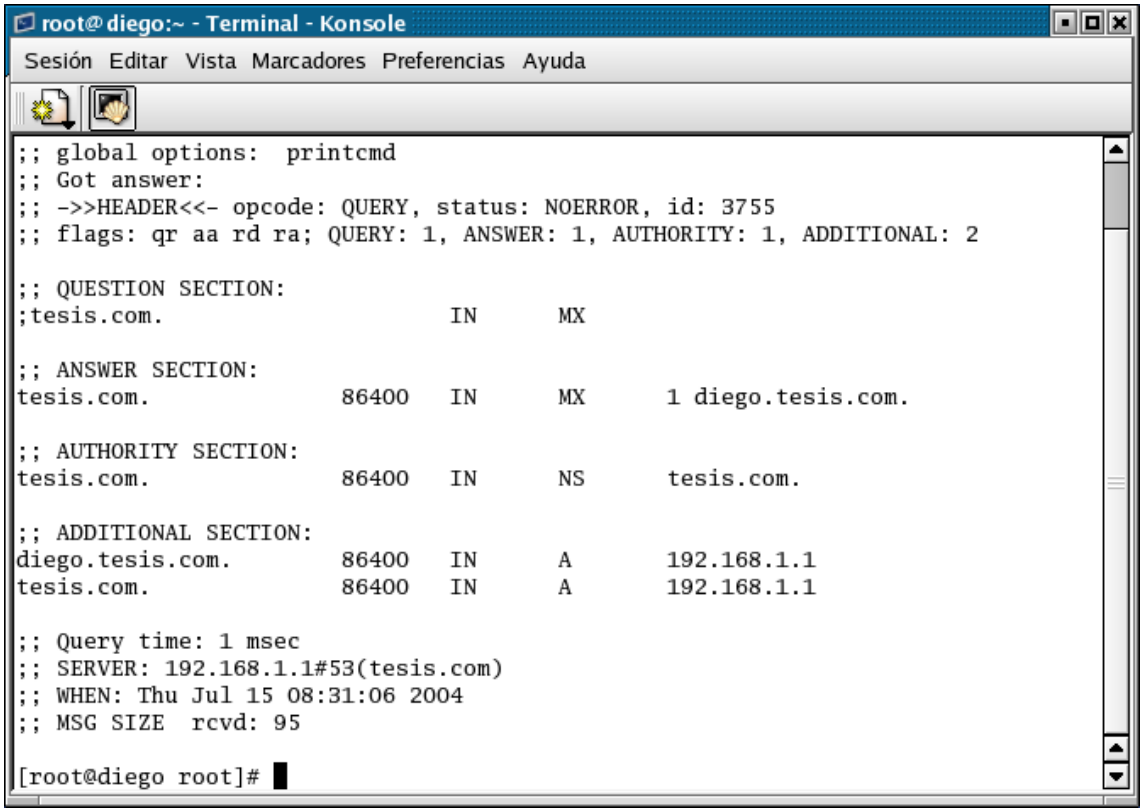
nombrehost IN A direcciónIPdehost

Ejemplos:

diego IN A 192.168.1.1

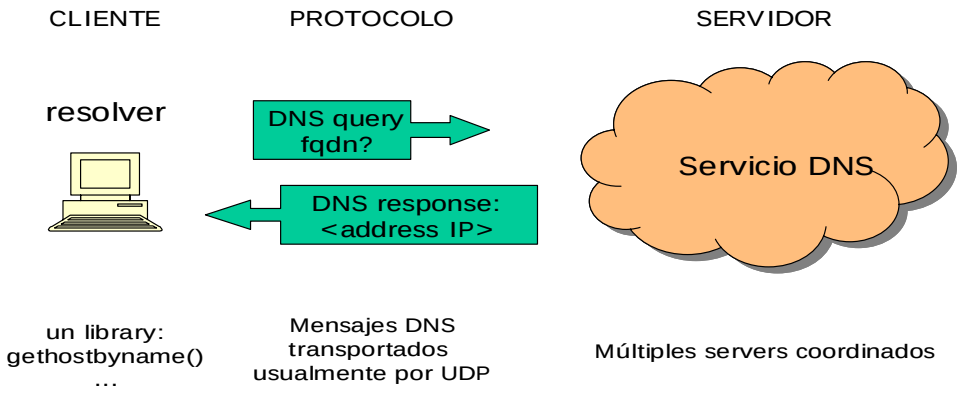
El Registro CNAME

Estos registros también reciben el nombre de **alias**, aunque son conocidos como entradas de "nombre canónico" (CNAME o **Canonical Name**). La utilidad principal de los mismos es la de usar más de un nombre para apuntar a un único host. Esto puede simplificar operaciones como albergar a la vez un servidor FTP y un servidor web en el mismo equipo.



1.7 Consultas

1.7.1 Arquitectura



Frame 1: query a un DNS server:

La pregunta es: “indíqueme la dirección IP correspondiente al host que se llama daniela.tesis.com”, y está dirigida al server DNS cuyo address de IP es 192.168.1.2, conocido previamente

Frame 2: respuesta del DNS server

El DNS server 192.168.1.1 nos contesta que el nombre “daniela.tesis.com” es en realidad.

1.7.2 Consultas recursivas

En una consulta recursiva, se pide al servidor de nombres que responda con los datos pedidos, o con un error que indique que el nombre del dominio especificado o los datos del tipo pedido no existen. El servidor de nombres no puede simplemente mandar al solicitante a un servidor de nombres distinto.

1.7.2.1 Esquema de resolución

1. Nuestro ordenador (cliente DNS) formula una **pregunta recursiva** a nuestro servidor DNS local (por lo general all proveedor de Internet).
2. El servidor local es el responsable de resolver la pregunta, aunque para ello tenga que reenviar la pregunta a otros servidores. Suponemos que no conoce la dirección IP asociada a www.tesis.com; entonces formulará una **pregunta iterativa** al servidor del dominio raíz.
3. El servidor del dominio raíz no conoce la dirección IP solicitada, pero devuelve la dirección del servidor del dominio com.
4. El servidor local reenvía la pregunta iterativa al servidor del dominio com.
5. El servidor del dominio com tampoco conoce la dirección IP preguntada, aunque sí conoce la dirección del servidor del dominio tesis.com, por lo que devuelve esta dirección.
6. El servidor local vuelve a reenviar la pregunta iterativa al servidor del dominio tesis.com.
7. El servidor del dominio tesis.com conoce la dirección IP de www.tesis.com y devuelve esta dirección al servidor local.
8. El servidor local por fin ha encontrado la respuesta y se la reenvía a nuestro ordenador "**192.168.1.1"**.

1.7.3 Consultas iterativas

En una consulta iterativa, el servidor de nombres consultado devuelve al solicitante la mejor respuesta que tiene disponible. Este tipo de consulta la suele realizar un servidor DNS a otros servidores DNS después de haber recibido una consulta recursiva de un resolver.

En una pregunta iterativa, el servidor de nombres envía la mejor respuesta que puede o que conoce en ese momento al peticionario. Esta respuesta puede ser la resolución del nombre, o puede referirse a otro servidor de nombres que sea capaz de responder al cliente original de la petición.

Primero consulta sus datos locales, si no está allí busca entonces en su caché y si aún no encuentra nada entonces devuelve la respuesta (servidor) más cercano al dominio buscado. Si el servidor falla, no lo vuelve a reintentar. Las bibliotecas del resolver hacen búsquedas recursivas e iterativas, mientras que entre servidores de nombres solo se hacen búsquedas iterativas.

La siguiente ilustración muestra un ejemplo de ambas preguntas: recursiva e iterativa. En este ejemplo, un cliente en una corporación, está preguntando al servidor de DNS por la dirección IP de http://www.tesis.com/.

1) El resolver envía una pregunta DNS recursiva al servidor local DNS preguntando por la dirección IP de http://www.tesis.com/. El servidor local de nombres, es responsable de resolver el nombre y no puede referirse a otro servidor de nombres para resolverlo.

2) El servidor de nombres local chequea sus zonas, y no encuentra zonas correspondientes a la petición de dicho dominio. Este entonces, envía una pregunta iterativa para http://www.tesis.com/ al servidor principal de nombres (root name server).

3) El servidor principal de nombres, tiene autoridad para el dominio principal (oot domain) y va a responder con la dirección IP del servidor de nombres para el dominio de más alto nivel .com.

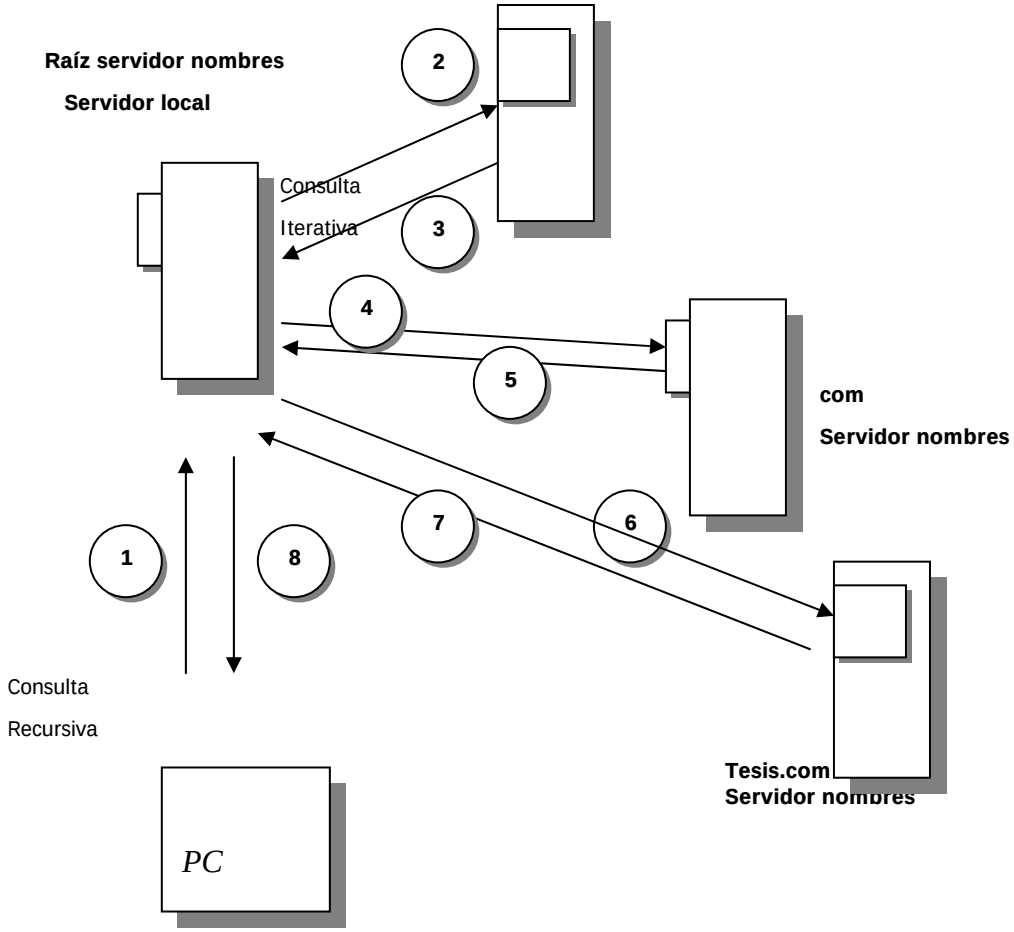
4) El servidor de nombres local envía una pregunta iterativa para http://www.tesis.com/ al servidor de nombres .com.

5) El servidor de nombres .com devuelve la dirección IP del servidor de nombres que está dando servicio al dominio tesis.com

6) El servidor de nombres local, envía una pregunta iterativa para http://www.tesis.com / al servidor de nombres tesis.com

7) El servidor de nombres tesis.com devuelve la dirección IP correspondiente a http://www.tesis.com.

8) El servidor de nombres local envía la dirección IP de http://www.tesis.com/ al cliente (resolver) original. (**“192.168.1.1”)**



1.7.4 Consultas Inversas

Los clientes DNS también pueden formular **preguntas inversas**, esto es, conocer el nombre de dominio dada una dirección IP. Para evitar una búsqueda exhaustiva por todo el espacio de nombres de dominio, se ha creado un dominio especial llamado in-addr.arpa. Cuando un cliente DNS desea conocer el nombre de dominio asociado a la dirección IP 192.168.1.1, formula una pregunta inversa a **1.1.168.192.in-addr.arpa**. La inversión de los bytes es necesaria debido a que los nombres de dominio son más genéricos por la derecha, al contrario que ocurre con las direcciones.

La funcionalidad de consultas inversas de DNS es importante, pues algunas aplicaciones ofrecen la posibilidad de implementar la seguridad basada en los nombres de los hosts de conexión. Por ejemplo, si un cliente intenta vincularse a un volumen NFS (Sistema de archivos de red o Network File System) con este arreglo de seguridad, el servidor NFS entraría en contacto con el servidor DNS y realizaría una consulta inversa del nombre en la dirección IP de los clientes. Si el nombre del host devuelto por el servidor DNS no se encuentra en la lista de acceso del volumen NFS, o si no se ha encontrado por DNS, entonces se denegaría la petición de montar NFS. La consulta inversa se utiliza a menudo para solucionar problemas.

CAPITULO II

2. SEGURIDAD

2.1 Amenazas a la seguridad

Para poder determinar las amenazas potenciales y las contramedidas posibles es primeramente necesario entender los flujos de datos normales en un sistema del DNS.

Como es el caso del spoofing en Dns. Sucede cuando un tercero intercepta y cambia o toma otra dirección con fines no conocidos, como fuese el caso en el que se toma la dirección de Hotmail.com, y todos aquellos que intenten ingresar en Hotmail.com se vean direccionados a mipagina.com y uno se ve sorprendido y piensa que escribió mal o que sucedió, por aquello que ahora en la mayoría de los casos se tiene actualización dinámica de direcciones IP.

Otra amenaza latente es tener la dirección fija del servidor de mails tal es el caso que al tener fija un intruso puede conectarse vía Telnet a Tesis.com obtener la dirección IP del registro MX **"MAIL"** y empezar a enviar mails con usuarios inexistentes o utilizando el nombre de uno de los usuarios emulando como si fuese el usuario el que estaría enviando el mensaje

2.2 Tipos de seguridad

La siguiente clasificación nos permite simplemente que seleccionemos soluciones y las estrategias apropiados para evitar o asegurar nuestro sistema

1.- La fuente primaria de los datos de la zona es normalmente los archivos de la zona y no olvidarse del archivo de named.conf que contiene las porciones de datos interesantes. Estos datos deben ser protegidos y sostenidos con seguridad. Esta amenaza se clasifica como local y es manejada típicamente por la buena administración del sistema.

2.- Si se hace funcionar los servidores auxiliares se hará transferencias de la zona. Si no se hace funcionar los servidores auxiliares, se podrá hacer funcionar con múltiples masters y eliminar la amenaza de la transferencia íntegramente. Esto se clasifica como amenaza del Servidor-Servidor (transacción).

3.- El BIND por defecto es deny, actualizaciones dinámicas de la zona. Si se permite este servicio pues se plantea una amenaza seria a la integridad de sus archivos de la zona y se protege. Esto se clasifica como amenaza del Servidor-Servidor (transacción).

4.- La posibilidad del peligro cache remoto debido al spoofing del IP, a la interceptación de los datos y a otros cortes es una llamada del juicio si se está utilizando un sitio web simple. Si el sitio es de alto perfil, Esto se clasifica como amenaza del Servidor-Cliente.

5.- Se entiende que ciertos grupos están mirando ya las implicaciones para los discernidores de imágenes seguros pero en fecha principios de 2004 esto no ha sido estandardizada. Esto se clasifica como amenaza del Servidor-Cliente

6.- Amenaza física, colocar el servidor en un lugar seguro, al cual solo tenga acceso el administrador.

2.3 Seguridad local

La administración normal del sistema se asegura de que los archivos de la configuración y de la zona sean sostenidos con seguridad, apropiada y los permisos de escritura aplicados y el control de acceso físico sensible a los servidores puede ser suficiente

CAPITULO III

3. MAPEO INVERSO

3.1 Revisión del Mapeo Inverso

Una pregunta normal del DNS estaría de la siguiente forma ' cuál es el IP del host=diego en domain=tesis.com '. Hay veces sin embargo cuando deseamos poder descubrir el nombre del anfitrión que IP address = 192.168.1.1 esto se requiere a veces para los propósitos de diagnóstico actualmente que se utiliza más con frecuencia para los propósitos de la seguridad de remontar a un hacker o el spammer, muchos sistemas modernos utiliza de hecho la traza o mapeo inverso para proporcionar la autentificación simple usando el look-up dual, para nombrar al IP.

Para realizar el mapeo inverso se diseño o implemento: IN-ADDR.ARPA llamado **Domain Name** (reservado) especial. Este dominio permite todas las direcciones apoyadas del Internet IPv4 (y ahora IPv6).

3.2 Archivos IN-ADDR.ARPA

Se define la estructura normal del Domain Name como árbol que empezaba con la raíz. Escribimos un Domain Name normal A LA IZQUIERDA a la DERECHA pero la estructura jerárquica CORRECTA a la IZQUIERDA.

```
Domain Name = diego.tesis.com
el nodo más alto en árbol es = com
después (más bajo) = tesis
después (más bajo) = diego
```

Se escribe una dirección IPv4 como:

```
192.168.1.1
```

Esta dirección IPv4 define un anfitrión = 1 en una gama de dirección de la clase C (192.168.1.x). En este caso la parte más importante (el nodo más alto) a la izquierda (192) no el CORRECTO. Esto es incorrecto y haría imposible construir una estructura arborescente sensible que se podría buscar en un solo curso de la vida.

La solución es invertir el orden de la dirección y poner el resultado bajo dominio especial IN-ADDR.ARPA (donde se verá esto también escrita como in-addr.arpa que sea ACEPTABLE puesto que los dominios son caso insensibles pero el caso debe ser preservado así que utilizaremos IN-ADDR.ARPA).

Finalmente la parte pasada IPv4 de la dirección (1) es el host address y los anfitriones, de nuestra lectura anterior, se definen típicamente dentro de un archivo de la zona así que no haremos caso de él y utilizaremos solamente la base de la dirección de la clase C. El resultado de nuestras manipulaciones es:

```
IP address = 192.168.1.1
Base de la clase C = 192.168.1; omite el host address = 1
Base invertida de la clase C = 1.168.192
Agregado al dominio de IN-ADDR.ARPA = a 1.168.192.IN-ADDR.ARPA
```

CAPITULO IV

4. ADMINISTRACION DE UN ARBOL DNS

4.1 Autoridad

Para su administración, el árbol (único) del DNS se divide en zonas.

La entidad administrativa que tiene autoridad sobre una porción (zona) del espacio de nombres, puede agregar, eliminar o cambiar labels dentro de esa zona. Por ejemplo, la entidad administrativa que tiene dominio sobre la zona que comienza en .com.ec es el NIC de Ecuador.

Una zona comienza en un cierto nodo, e incluye todos los nodos descendientes de éste, excepto los nodos pertenecientes a sub-zonas cuya autoridad haya sido previamente delegada a otras entidades.

La entidad que tiene autoridad sobre una zona se ocupa de mantener los servidores de DNS correspondientes a esa zona (authoritative servers)

Hay 2 tipos de authoritative servers: Primario (sólo uno) y secundarios. Los secundarios actualizan su información automáticamente, consultando al primario.

Del lado cliente, la resolución suele ser mediada por un server especial, llamado caching-only nameserver, en el cual no se configura información sobre ninguna zona. En estos servers se configuran solamente los addresses de IP de los “root nameservers”

4.2 Delegación In-addr.arpa

En el DNS se reserva un TLD (top level domain) para una zona que permite resolver nombres a partir de addres de IP.

Los administradores deciden qué datos se colocan allí. Éstos deben asegurar que la información sea coherente (el sistema no incluye chequeos de consistencia de la información ingresada para las distintas ramas del árbol)

Los root-nameservers son autoritativos para las zonas correspondientes a la zona in-addr.arpa, y delegan a otros nameservers con correspondencia a las clases IP tradicionales (A, B y C). Éstos a su vez pueden sub-delegar siguiendo la serie de números decimales que representan los bytes del address de IP.

4.3 Características de la arquitectura

El mapeo de números de IP a nombres en el DNS es arbitrario cualquier número de IP (independientemente de, por ejemplo, su parte de net-id) puede asociarse con cualquier nombre en el DNS (la asignación de nombres es independiente de la numeración de IP)

La ubicación de un host en el sistema DNS es independiente del ruteo de los datagramas (éste se hace exclusivamente a partir del id de net -o sub/super-net del número de IP)

La autoridad en el dominio in-addr.arpa sí está relacionada con la conexión física (se delega según las componentes del número de IP)

4.4 Limitación de las zonas

Las zonas se delimitan cuando se delega autoridad sobre una porción de la zona a otra entidad administrativa.

La delegación implica imposibilidad de modificar datos de la zona delegada mientras la delegación sea efectiva (se mantiene la posibilidad de revocar la delegación) la necesidad por parte de la entidad a la que se delega de mantener un name server para la zona delegada.

4.5 Delegación Incorrecta

Ocurre cuando un server delega en otra autoridad sobre una zona, pero éste no ha sido configurado como autoritativo para esa zona.

Se genera tráfico innecesario sobre la internet, y demoras debido a las consultas no respondidas hechas a los lame servers.

Se compromete la disponibilidad debido a una falsa percepción de redundancia.

4.6 Mecanismo de delegación

Se utilizan registros especiales (registros NS) en el server de la zona "padre" (parent), que indican el número de IP del nameserver de la zona "hija" (child).

El nameserver hijo también incluye registros NS correspondientes a los nameservers autoritativos para su zona. Estos deben corresponder con los consignados en el padre.

glue A records: se utilizan en el padre para indicar el address de IP de los nameservers autoritativos para la zona hija. Sólo son necesarios si el nombre de éstos está dentro de la zona hija.

CAPITULO V

5. PRÁCTICO CONFIGURACIONES EJEMPLO

5.1 Configuración e instalación de un DNS bajo Linux Red Hat 9.0

Adherir una zona
Archivo de Zona
Actualización dinámica
Forwarders
Adherir un nuevo host
El Nuevo Host

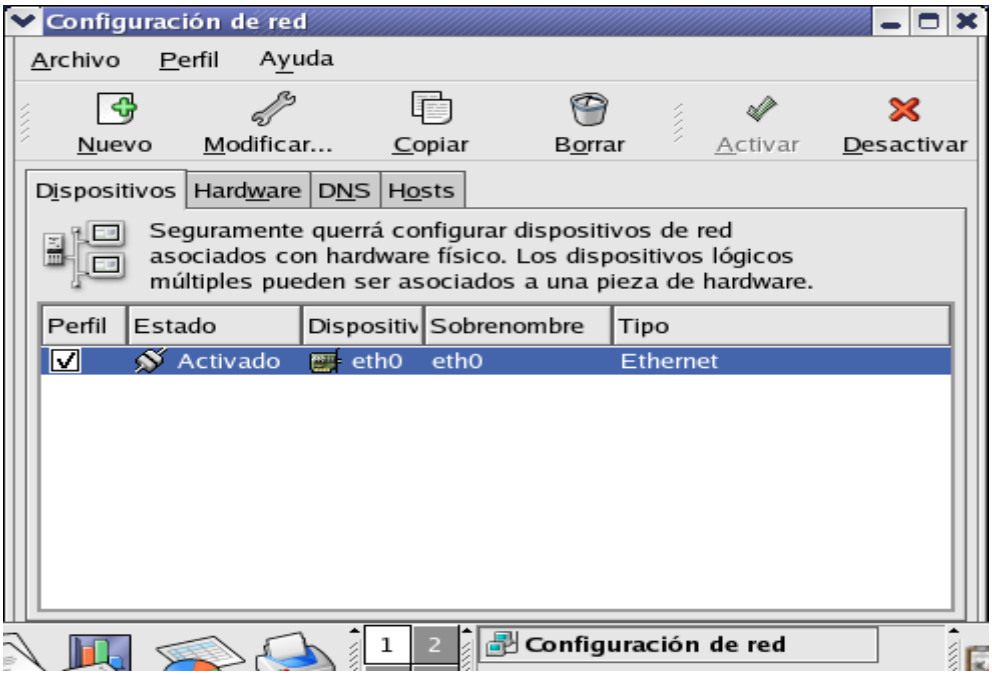
Luego de haber analizado y conocido como se estructura un servidor DNS he implementado un servidor DNS interno bajo Linux Red hat 9.0, el cual se podría usar dentro de una Lan (Local Area Network) de una manera amigable con el fin de que este a su vez nos permita instalar un servidor de mails, ftp, los cuales también se implementarán (**"Service Sendmail",Vsftp**).

Pasos para la Instalación y Configuración

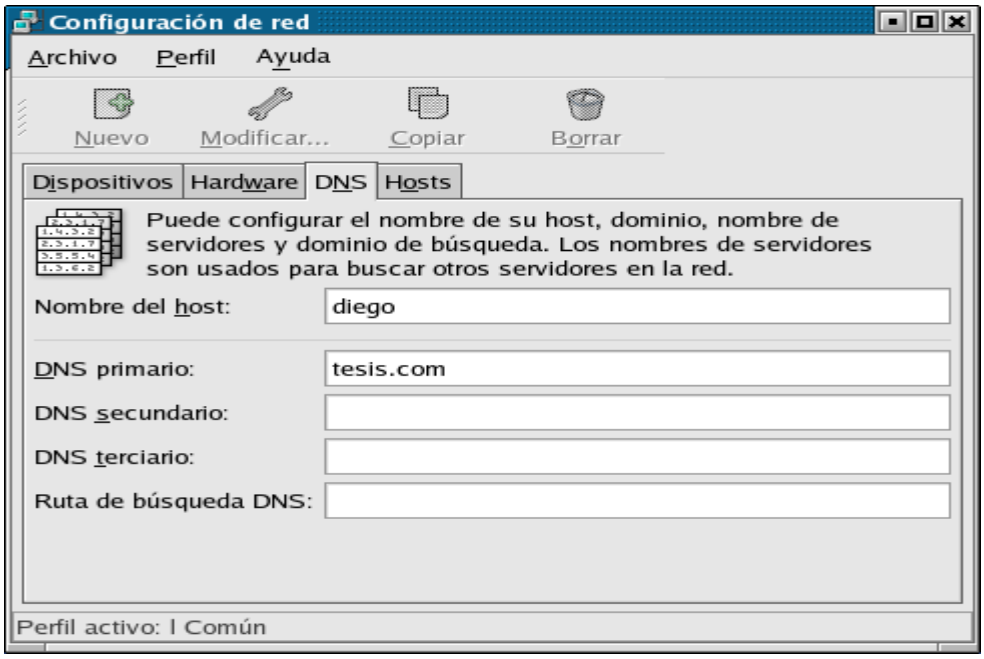
Nota:

Para mi aplicación he definido el dominio **tesis.com**, el cual será el servidor DNS con la dirección **192.168.1.1,** y el host **diego.**

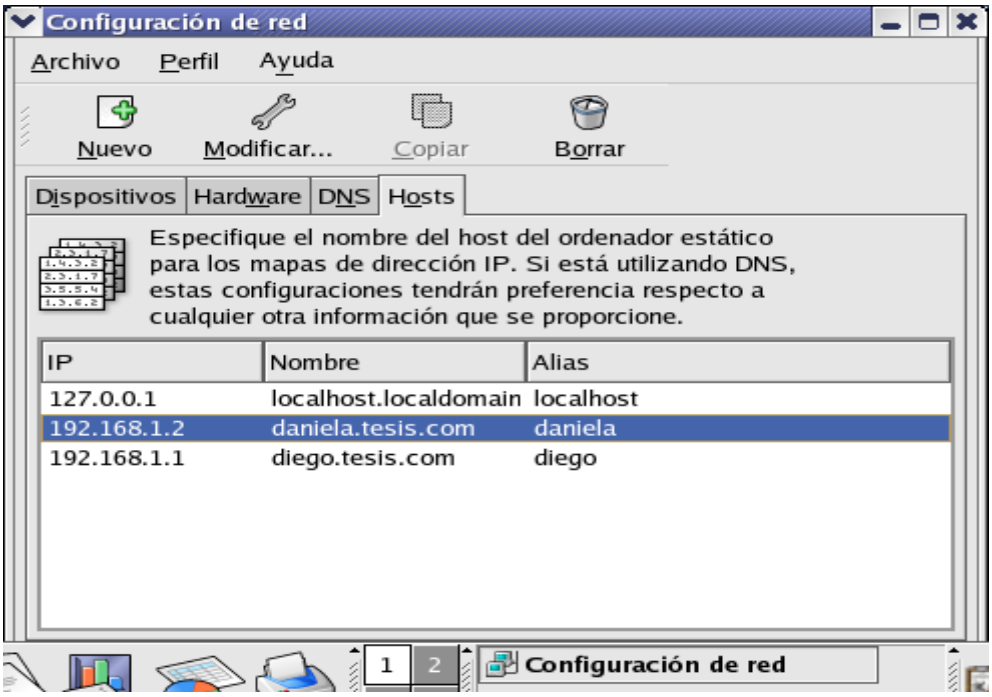
Luego de haber instalado los paquetes de Linux, nos corresponde configurar la placa de red.



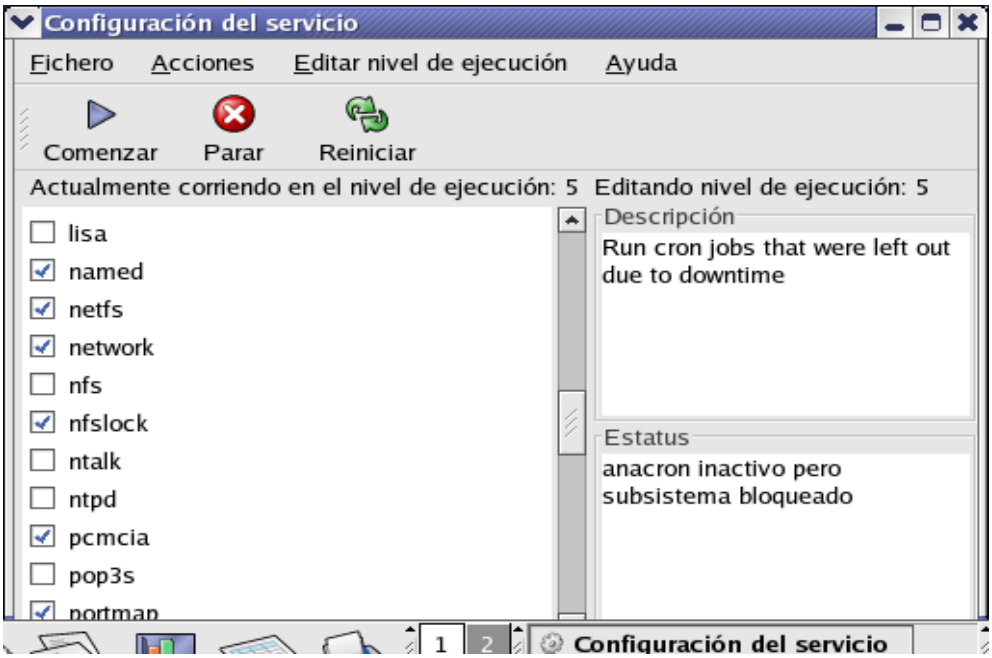
Seguido de esto procedemos a dar nombre a nuestra máquina, y asignar el servidor DNS primario



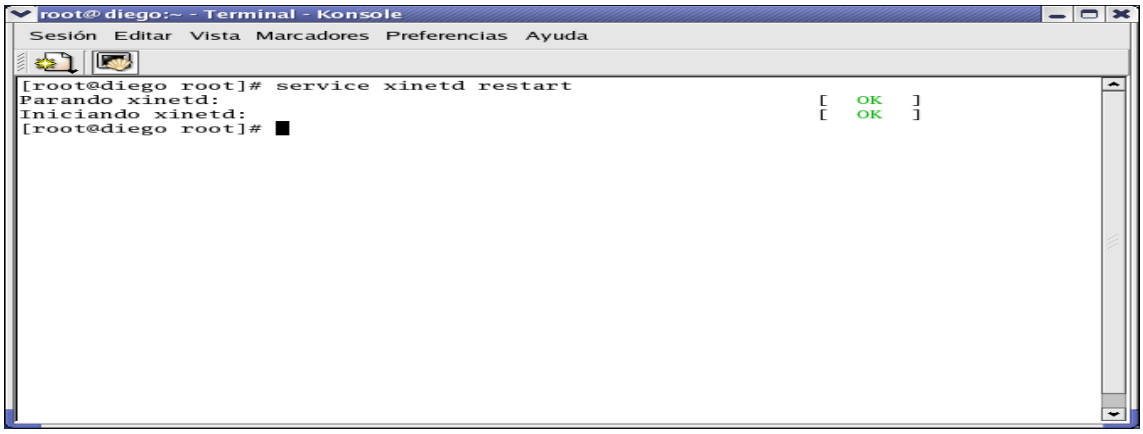
Una vez que damos un nombre a nuestro host, procedemos a agregar en la tabla de ruteo aquellas máquinas que tendrán acceso al servidor



Una vez que hemos configurado la red procedemos a activarle y levantar los servicios.



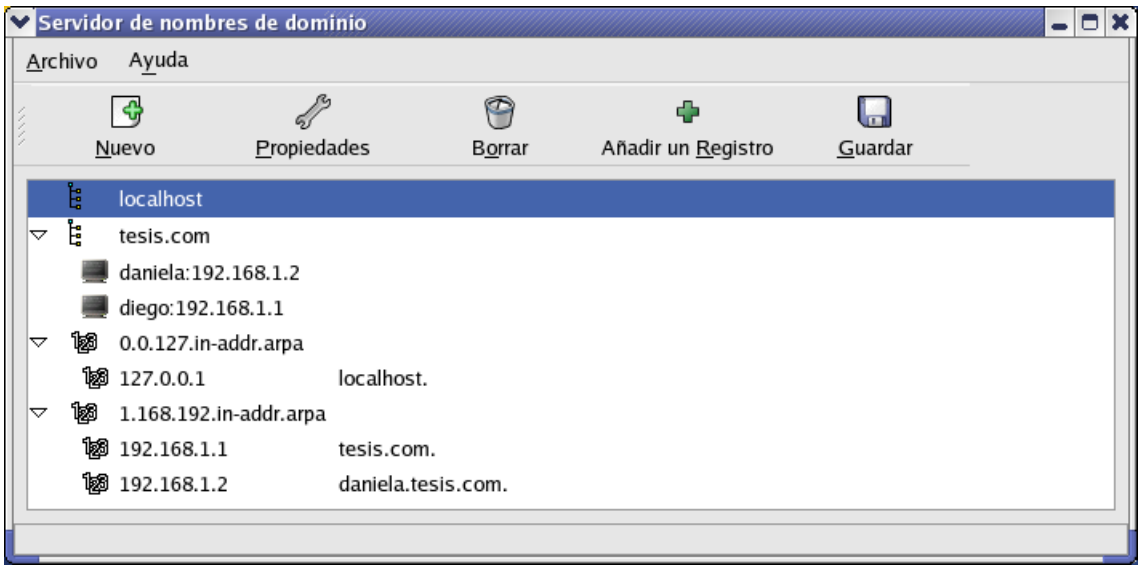
Esta es otra forma de levantar el servicio de red: **service xinetd restart**



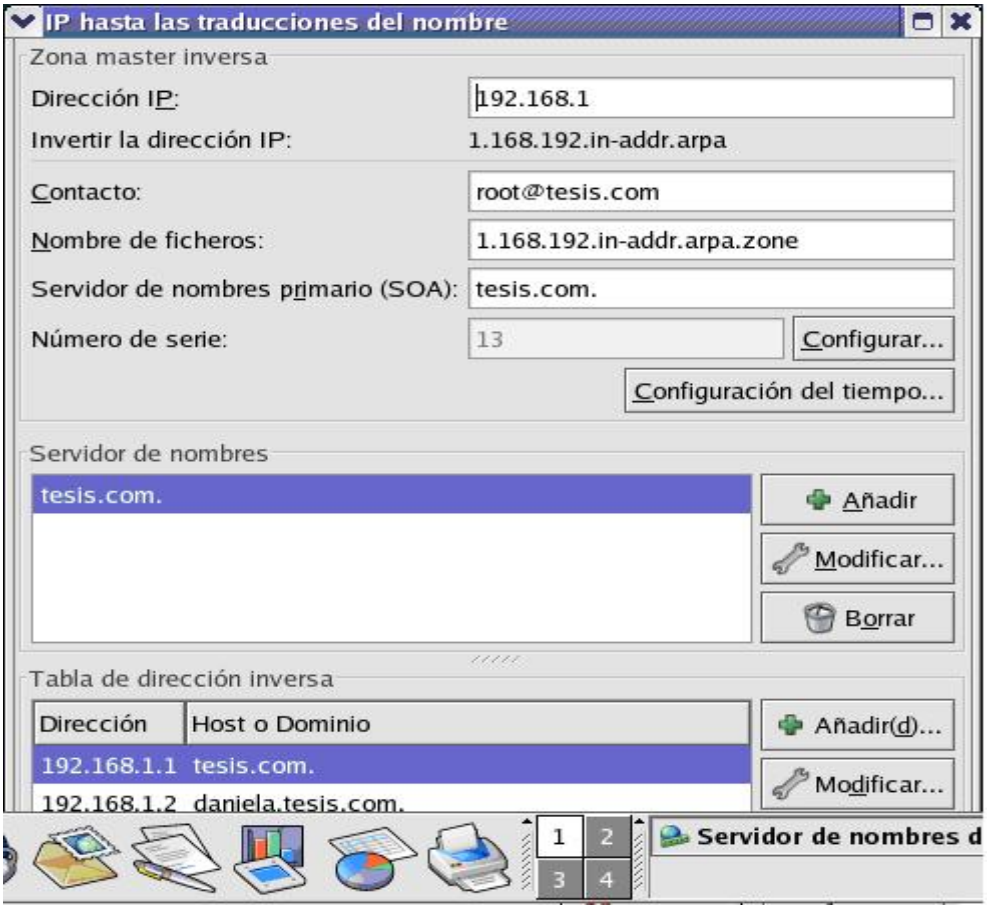
5.1.1 Configurar DNS.

El primer paso es definir una zona de reenvío, y otra inversa con el fin de que el servidor pueda realizar las conversiones adecuadas. En este caso tesis.com es el reenvío master y 1.1.168.192.in-addr.arpa será la que realice la inversa.

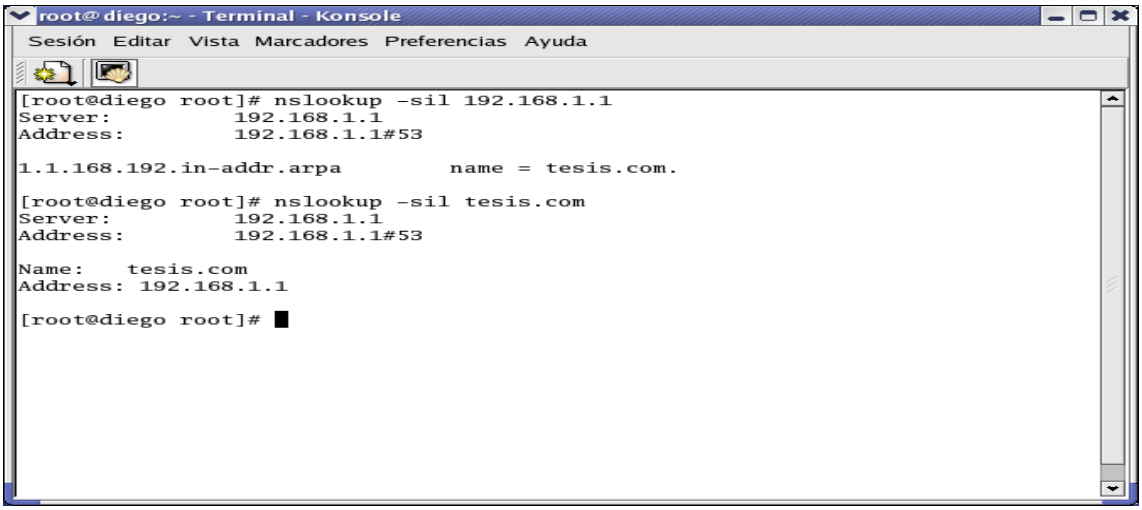
Para implementar la zona master indicamos el dominio en mi caso es **tesis.com** se indica donde va a guardarse es decir el archivo de zona y agregamos los registros que están en mi red.

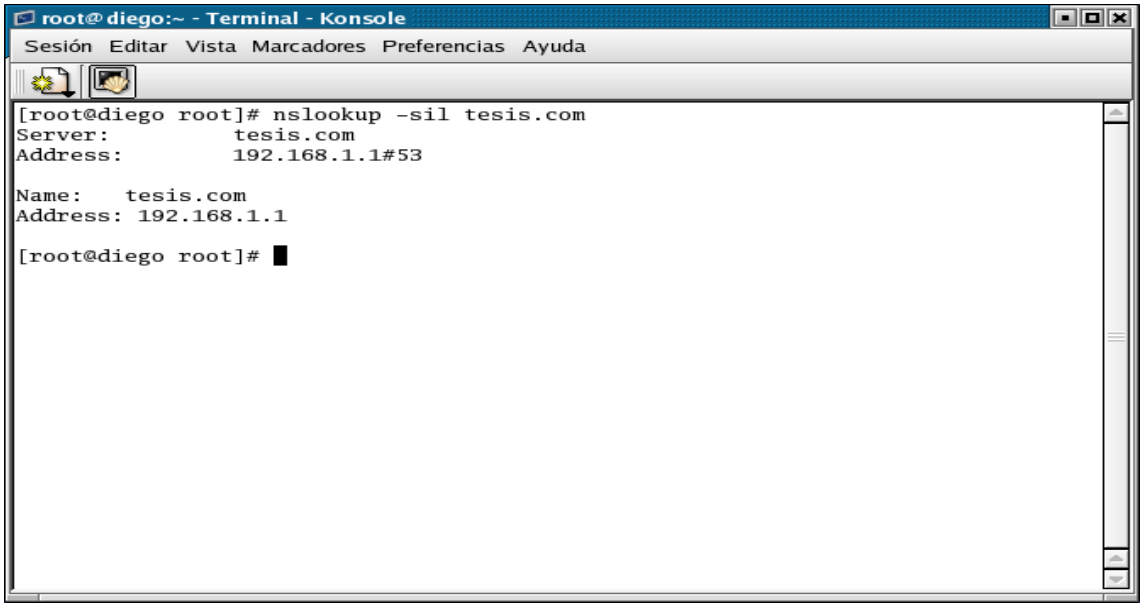


Para implementar la zona inversa, ingresamos los primeros tres octetos de una dirección IP en forma ordenada de izquierda a derecha y luego el servidor de nombres , seguido de esto las direcciones de cuales queremos que resuelva la inversa.



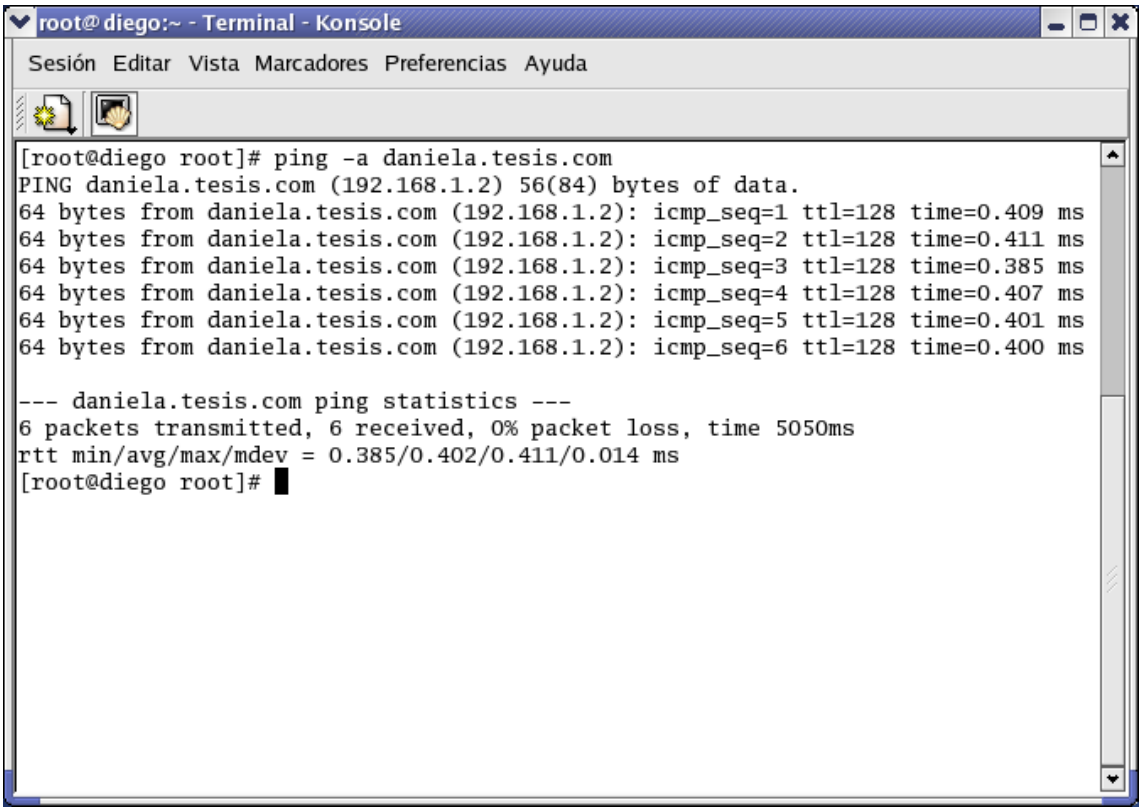
Una vez que se ha realizado lo anterior se procede a comprobar y verificar el funcionamiento del DNS mediante el comando **NSLOOKUP.** Primero con la dirección ip y nos da como respuesta **tesis.com,** lo cual nos indica que esta la configuración correcta, ahora vamos a verificar la resolución de nombres "INVERSA" damos tesis.com y nos da como resultado **192.168.1.1 con lo cual esta bien configurado.**

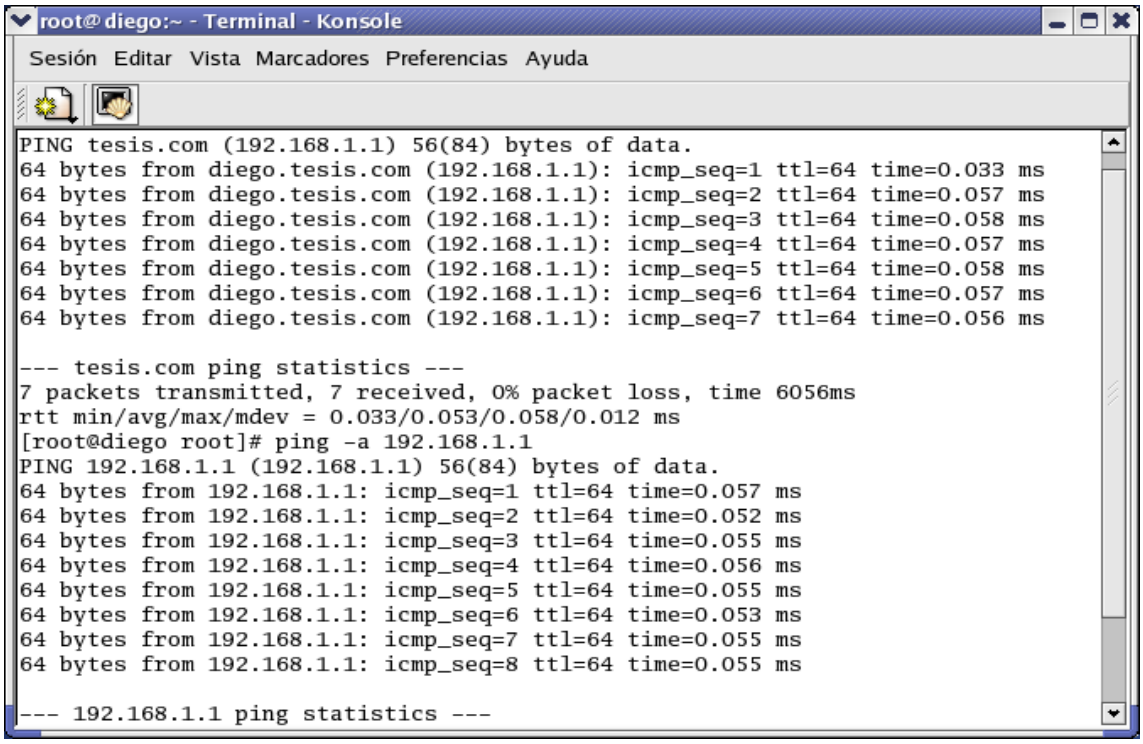




Otra forma de comprobar es realizar pings a dichas máquinas.

Ping al cliente daniela.tesis.com y al servidor





5.2 Configuración Cliente

Ahora procederemos a configurar el cliente en Windows xp en el cual se agregara la dirección del DNS a mas de tener su dirección Ip en este caso 192.168.1.2, nombre de la máquina será **daniela.**

5.3 Servidor de Correo Sendmail

Una vez configurado correctamente el servidor de dominios **"DNS"**, será posible enviar y recibir correo electrónico.

Lo primero será establecer que es lo que tenemos en la red local y que es lo que haremos con esto. Determinar que máquinas hay en mi red local, específicamente las direcciones IP, de las que necesitan poder enviar y recibir correo electrónico y cuales NO deben hacerlo.

Determinar como desea recuperar los mensajes de correo electrónico que arriben al servidor. POP3 o IMAP.

POP3 vs IMAP

La diferencia principal es que IMAP almacena los mails en el servidor y solo se muestran las cabeceras de los mismos en los clientes, al contrario en POP3 los mails se recuperan desde el cliente pero no quedan almacenados en el servidor.

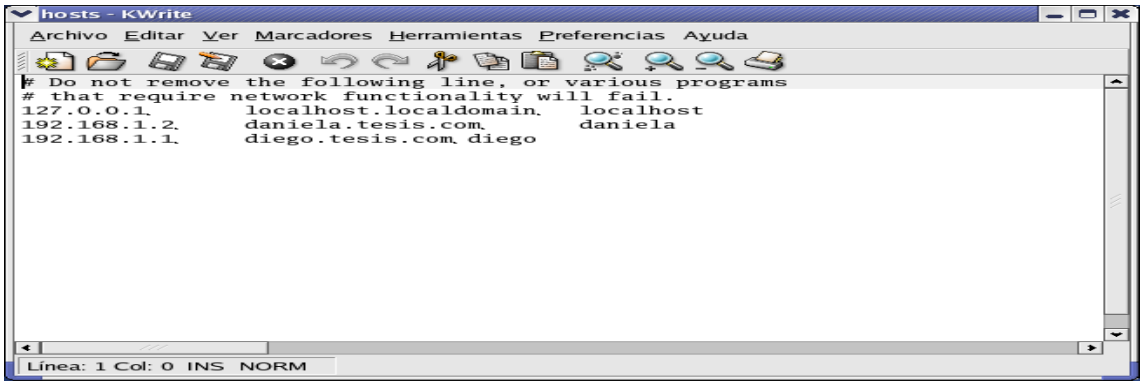
Debemos definir el nombre de la máquina que funcionará como servidor de correo. Y para ello utilizaremos el diego.tesis.com y se puede comprobar en /etc/sysconfig/network y /etc/hosts:

Para /etc/sysconfig/network, es decir, el nombre que asignamos a la máquina, correspondería lo siguiente:

NETWORKING=YES
HOSTNAME=diego.tesis.com

Para */etc/hosts*, es decir, la información de los hosts y las direcciones IP, correspondería lo siguiente en la cual deberemos agregar.

192.168.1.1 diego.tesis.com
192.168.1.2 daniela.tesis.com



Es importante tener instalados los paquetes sendmail y sendmail-cf, ya que al utilizar el servidor de correo Sendmail para el envío de nuestros mensajes, mismo que nos permitirá utilizar el servicio de IMAP y POP3. Para asegurarse de esto, se puede utilizar la siguiente línea de comando:

rpm -q sendmail sendmail-cf imap

Se Deben definir los dominios para los cuales se estará permitiendo enviar correo electrónico. Esto se hace generando el fichero /etc/mail/relay-domains:

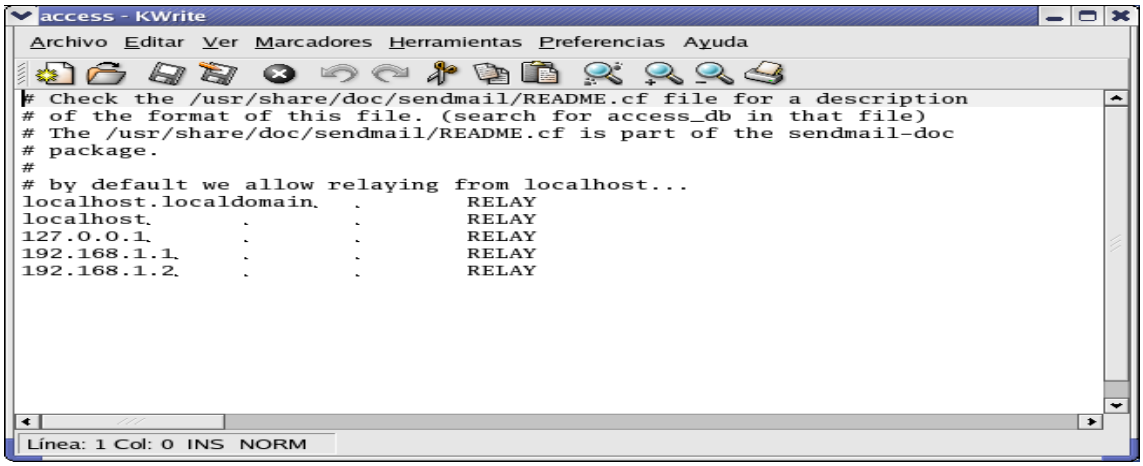
Tesis.com

Abrimos ahora el archivo */etc/mail/access* y agregamos algunas líneas para definir quienes podrán hacer uso de nuestro servidor de correo para poder enviar mensajes:

En mi caso las 2 máquinas:

192.168.1.1 RELAY

192.168.1.2 RELAY



Si colocamos REJECT estamos indicando que esta dirección estará cancelada para envio y recepción de mails.

makemap hash /etc/mail/access.db < /etc/mail/access

Terminados los detalles de la configuración, reiniciamos sendmail del siguiente modo y tendremos listo un servidor de correo que podrá utilizar para enviar mensajes para toda su red local utilizando el servidor SMTP de su proveedor de nuestros servicios tesis.com:

/sbin/service sendmail restart

Por ultimo se debe habilitar los servicios de manera automática e inmediata ejecutando los siguientes comandos.

/sbin/chkconfig ipop3 on
/sbin/chkconfig pop3s on
/sbin/chkconfig imap on
/sbin/chkconfig imaps on

5.4 Comandos de prueba Servidor DNS

Para poder verificar y comprobar el funcionamiento del servidor DNS configurado en plataforma Linux Red Hat 9.0 utilice los siguientes comandos:

El comando dig (*Domain informatiol Groper*) constituye una herramienta para realizar consultas de diverso tipo a un servidor de DNS. Este muestra las respuestas recibidas de acuerdo a su solicitud. Es muy útil para detectar problemas en la configuración de los servidores de DNS debido a su flexibilidad, facilidad de uso y claridad en su salida.

Aunque normalmente las consultas que permite dig se definen en la línea de comando Cuando no se añade ninguna opción o argumento en la línea de comando se consultan los servidores de nombres del dominio raíz (*NS query*).

La forma básica de invocar a dig es:

dig <nombre> [tipo]

donde:

- @servidor - es el nombre o la dirección IP del servidor a consultar.
- nombre - es el nombre de dominio del *record* por el cual se quiere preguntar.
- tipo - es el tipo del *record* por el que se consulta (*ANY*, *NS*, *SOA*, *MX*, etc.). De no indicarse se asumirá *A*.

El comando host es un utilitario que permite hacer búsquedas en el DNS. Se utiliza básicamente para convertir nombres en direcciones IP y viceversa.

Sintaxis: host [opciones] <dominio>

Algunas opciones:

- -t <tipo> : indica el tipo de record a devolver. Puede ser A, ANY, PTR, NS, etc.
- -R <n> : permite modificar el número de intentos que se hacen para obtener la respuesta ya que por defecto es uno.
- -l : lista toda la información del dominio

El comando Nslookup este programa posee dos modos en su comportamiento:

- Interactivo: permite realizar un número ilimitado de consultas diversas acerca de distintos hosts y dominios utilizando a varios servidores de DNS. Provee un prompt en el cual se podrán ejecutar distintos comandos en correspondencia con las acciones a realizar. Para terminarlo se podrá presionar Ctrl-D o utilizar el comando exit.
- No interactivo: se utiliza para realizar una única consulta o sea, para devolver sólo la información exacta de un host o un dominio a partir de un servidor.

El primer modo se obtiene cuando se invoca nslookup sin argumentos o cuando el primer argumento es ``-'' y el segundo, un nombre de dominio o una dirección IP de un servidor de DNS. En cambio el modo no interactivo se alcanza dado que se indica como primer argumento el nombre o dirección IP del host buscado y como segundo, opcionalmente, el nombre o la dirección del servidor a consultar. Además se pueden indicar opciones para expresar que tipo de información se buscará y como se hará. Las

opciones toman la forma <opción>=<valor> y en el modo no interactivo se pueden tomar del fichero oculto .nslookup presente en el directorio base del usuario actual, o de la propia línea de comando precedidas del signo ``-''. En el modo interactivo se expresan a través del comando set

set <opción>[=<valor>] - permite fijar el valor de las opciones que modifican el comportamiento de todas las consultas subsiguientes. Las opciones se describen a través de una palabra, algunas poseen un valor asociado. Entre las opciones válidas se encuentran:

ls [opción]

- all imprime los valores actuales de las opciones activadas.
- -d lista todos los records presentes en el dominio. Es equivalente a ``-t ANY''.
- -h lista la información del procesador y del sistema operativo correspondiente a los hosts del dominio. Es equivalente a ``-t HINFO''.
- -s lista los servicios conocidos que ofrecen los hosts del dominio. Es equivalente a ``-t WKS''.

CAPITULO VI

6. CONCLUSIONES Y RECOMENDACIONES

Como conclusión podemos decir, que el servicio DNS es de suma importancia en el mundo de la informática actual, más aún con el uso y el avance de Internet y de las redes locales dentro de las empresas, ya que este servicio, facilita la localización de los nodos sin tener que conocer las direcciones IP, permite un control local sobre los segmentos de la base de datos en general, logrando que cada segmento esté disponible a lo largo de toda la red Internet.

El sistema de nombres de dominios utiliza un esquema cliente servidor y a su vez resuelve la implementación de un servidor de mails, pues este requiere de un DNS, caso contrario nos veríamos muy afectados en memorizar las direcciones IP de cada uno de los correos electrónicos, es decir un servidor DNS juega un papel importante dentro de una estructura de redes IP.

La arquitectura de un servidor DNS implementado en una plataforma Linux nos brinda una mayor seguridad y una interacción con el administrador a un más alto nivel.

Glosario

Término	Definición
Domain Name	Nombre de dominio DNS primario que se utiliza como raíz para una zona o un registro
fully qualified domain name (FQDN)	Es el hombre completo o dirección el cual va desde la raíz
host name	Es el host o máquina dentro de un dominio ejemplo. diego.tesis.
qualified domain name	Define una parte de un dominio hacia la raíz.
sub-domain name	Es un nombre arbitrario que pertenece a un dominio y va antes del dominio.
Top Level Domain (TLD)	Es el nivel mas alto de un FQDN
Dominios	La estructura básica de un dominio DNS y sus tipos.

Delegación de Dominios	Cómo descentralizar DNS, cómo crear subdominios
Dominio host	El Dominio en el que se ha configurado un espacio de nombres
Registros de Recursos	RR, o dónde guarda la información DNS
Servidores de Nombres	Tipos de servidores de nombres
Resolver	Las bibliotecas que buscan la información por el lado del cliente
Zona	Hay 2 tipos la directa y la inversa
Dominio in-addr.arpa	Dominio DNS especial de nivel superior reservado para invertir la asignación de direcciones IP a nombres de host DNS
Dominio in-addr.arpa	Dominio DNS especial de nivel superior reservado para invertir la asignación de direcciones IP a nombres de host DNS

Cname	Registro que reciben el nombre de **alias**, aunque son conocidos como entradas de "nombre canónico"

Bibliografía

- TCP/IP Ilustrated Volumen I, **Stevens W Richard**
- TCP/IP Tutorial and Technical Overview ,**Adolfo Rodríguez,John Gatrell John Karas**
- DNS and BIND. **Paul Albitz.**
- Tcp/IP Network Administración. **Craig Hunt.**
- Managing IP Networks with Cisco Routers. **Scott M. Ballew.**
- RFC 920: Requerimientos
- RFC 1101: DNS Enconding of Network Names and Other Types
- RFC 1033 : Guía de Administración
- RFC 1034: Conceptos
- RFC 1035: Implementación
- RFC 1591: Estructura y delegación
- RFC 2535: Seguridad DNS
- Apuntes dns (curso de graduación)
- Enciclopedia Encarta 2004

Bibliografía Internet

Autor	Nombre Sitio	Fecha ingreso	URL
András Salamon	DNS	08-Jul-04	www.dns.net/dnsrd/
	SSTUFF	09-Jul-04	www.dnsstuff.com
	SERVERS	11-Jul-04	www.root-servers.org/
	EUD	11-Jul-04	www.eud.com
Alina Castellanos Leyva	ABDN	12-Jul-04	www.erg.abdn.ac.uk/users/ gorry/course/inet-pages/dns.html
Leopoldo de la Fuente Silva	CRAM	11-Jul-04	www.cramsession.com/articles/files/defining-dns.asp
	ZYTRAX	14-Jul-04	www.zytrax.con/books/dns/
	MICROSOFT	10-Jul-04	www.microsoft.com/dns/
	DNS2GO	07-Jul-04	www.dns2go.com
	MICROSOFT	07-Jul-04	www.microsoft.com/technet/Batch2003-11/d5dns_153643/

Índice General