Network Working Group P. Mockapetris
Request for Comments: 1034 ISI
RFCs obsoletos: 882, 883, 973 Noviembre 1987

Traducido al castellano por Ramiro Encinas en Enero 2007

NOMBRES DE DOMINIO - CONCEPTOS E INSTALACIÓN

1. ESTADO DE ESTE MEMORÁNDUM

Este RFC es una introducción al Sistema de Nombres de Dominio (DNS), y no tiene en cuenta muchos detalles que pueden encontrarse en el RFC-1035 "Nombres de Dominio - Implementación y Especificaciones". El RFC-1035 asume que el lector está familiarizado con los conceptos descritos en este memorándum (RFC-1034).

El protocolo oficial está compuesto por un subconjunto de funciones DNS y tipos de datos DNS. También incluye consultas estándar, sus respuestas y la mayoría de los formatos de datos de las clases de Internet, (direcciones de host).

De todas formas, la intención del sistema de dominios es escalable. Los investigadores contínuamente proponen, implementan y experimentan con nuevos tipos de datos, tipos de consultas, clases, funciones, etc. Normalmente los componentes del protocolo oficial no suelen cambiar y trabajan en servicio de producción. Los componentes experimentales son extensiones más allá del protocolo oficial. Las características experimentales u obsoletas están claramente indicadas en estos RFC's, y tal información ha de ser utilizada con precaución.

El lector debe tener especial cuidado de no depender de los valores que aparecen en los ejemplos, debido a que su propósito es fundamentalmente pedagógico. La distribución de este memorándum es ilimitada.

2. INTRODUCCIÓN

Este RFC es una introducción de los estilos de los nombres de dominio, su uso para correo de Internet, soporte de direcciones de host, y los protocolos y servidores utilizados para implementar instalaciones de nombres de dominio.

2.1 La historia de los nombres de dominio

El ímpetu por el desarrollo del sistema de dominios ha crecido en Internet:

- El mapeo de las direcciones a nombres de host están gestionados por el Network Information Center (NIC) en un sólo fichero (HOSTS.TXT), el cual fue distribuido a todos los hosts mediante FTP. [RFC-952, RFC-953].

  El ancho de banda de red total consumido en la distribución de este escenario es proporcional al cuadrado del número de hosts en la red. Incluso cuando se utilizan múltiples niveles de FTP, la carga de salida FTP en los NIC de los hosts es considerable. No se tuvo una buena previsión del futuro y explosivo crecimiento en el número de hosts.

- De la misma manera ocurrió con las redes. Los hosts de la ARPANET original fueron sustituidos por redes locales de estaciones de trabajo. Las organizaciones administraron localmente sus propios nombres y direcciones, pero tuvieron que esperar bastante al NIC para que HOSTS.TXT estuviera actualizado con esos cambios en Internet. Las organizaciones también querían alguna estructura local en el espacio de nombres.

- Al volverse más sofisticadas las aplicaciones en Internet, se creó la necesidad de tener un servicio de nombres de propósito general.

El resultado fueron varias ideas acerca del espacio de nombres y su gestión [IEN-116, RFC-799, RFC-819, RFC-830]. Los propósitos cambiaron, pero la idea común era la de un espacio de nombres jerárquico que se correspondiera con la estructura de la organización y del uso del "." como caracter de unión entre los niveles de la jerarquía. En los [RFC-882, RFC-883] se describió un diseño utilizando una base de datos distribuida y recursos generalizados. En base a la experiencia de varias implementaciones, el sistema evolucionó en el escenario que describe este memorándum.

Los términos "dominio" o "nombre de dominio" son utilizados en muchos contextos más allá de lo que DNS describe aquí. El término nombre de dominio se utiliza muy a menudo para referirse a un nombre con una estructura indicada por puntos, sin relación alguna con DNS. Esto se cumple en el direccionamiento de correo electrónico [Quarterman 86].

2.2 Objetivos del diseño DNS

Los objetivos del diseño DNS es su estructura. Son:

- El objetivo principal es un espacio de nombres consistente que sea utilizado para referirse a los recursos. Para evitar problemas causados por extensiones especiales, los nombres no necesitarán contener identificadores de red, direcciones, rutas o información similar como parte del nombre.

- La capacidad total de la base de datos y la frecuencia de las actualizaciones debe ser mantenida de una forma distribuida, con cachés locales para mejorar el rendimiento.

  Los intentos de tener una copia consistente de la base de datos completa serán cada vez más costosos y difíciles, por tanto es mejor olvidarse de ello. El mismo principio es aplicado a la estructura del espacio de nombres, y a los mecanismos particulares para crear y eliminar nombres, que también han de ser distribuidos.

- Si comparamos el coste de la adquisición de datos, la velocidad de las actualizaciones y la precisión de las cachés, la prioridad es la adquisición de datos.

- La instalación debe ser generalmente sencilla y no restringirla sólo a una aplicación. Tenemos que utilizar nombres para recuperar direcciones de host, datos de buzones de correo, y otros datos aún indeterminados. Todos los datos asociados con un nombre se etiquetan con un tipo, y las solicitudes pueden limitarse a un sólo tipo.

- Debido a que queremos que el espacio de nombres sea manejable para que distinga entre redes y aplicaciones, puede utilizarse el mismo espacio de nombres con diferentes familias de protocolos o

gestores. Por ejemplo, aunque todos los protocolos incluyen dirección o direcciones, los formatos de direcciones de host son distintos dependiendo de los protocolos. El DNS etiqueta todos los datos con una clase, como el tipo, de forma que podemos utilizar de forma paralela diferentes formatos para datos de tipos de direcciones.

- Queremos que las transacciones entre los servidores de nombres sean independientes del sistema de comunicaciones que las hace posible. Algunos sistemas pueden utilizar datagramas para consultas y respuestas, y sólo establecer circuitos virtuales para las transacciones de confianza (actualizaciones de bases de datos, transacciones grandes), mientras que otros sistemas pueden utilizar sólo circuitos virtuales.

- El sistema ha de ser manejable independientemente de las capacidades del host, de forma que pueda utilizarse tanto en ordenadores personales como servidores, aunque quizá se utilice de formas distintas.

2.3. Suposiciones sobre el uso

La organización del sistema de dominios deriva de algunas suposiciones sobre las necesidades y patrones de uso de su comunidad de usuarios y está diseñado para descartar muchos de los problemas complicados encontrados en los sistemas de bases de datos de propósito general.

Las suposiciones son:

- El tamaño total de la base de datos será inicialmente proporcional al número de hosts que utilicen el sistema, y podrá crecer de forma eventual proporcionalmente con el número de usuarios de los hosts, y de igual forma con las direcciones de correo y con cualquier otra información que se sume al sistema de dominios.

- La mayoría de los datos del sistema cambiarán muy lentamente, (por ej. direcciones de correos, direcciones de hosts). El sistema debe adaptarse a los cambios rápidos de datos que así lo requieran (segundos o minutos).

- Los límites administrativos utilizados para distribuir las responsabilidades de la base de datos corresponderán normalmente a organizaciones con uno o más hosts. Cada organización que tenga responsabilidad para un conjunto de dominios en particular deberá tener servidores de nombres redundantes, ya sean hosts propios o hosts que utilice fuera de la organización.

- Los clientes del sistema de dominio deberán identificar a los servidores de nombres de confianza preferentemente antes de aceptar referencias de servidores de nombres que no son de confianza.

- El acceso a la información es más crítico que las actualizaciones instantáneas o las garantías de consistencia. Esto es, el proceso de actualización permite actualizaciones que se difunden fuera del sistema de dominio de los usuarios en vez de que garantice que todas las copias se actualizan a la vez. Cuando las actualizaciones no están disponibles por fallo de red o de host, lo habitual es seguir con la información antigua hasta que se actualice. El modelo general dice que las copias se distribuyan con tiempos de refresco. El responsable de la distribución establece el valor de tiempo de refresco y el receptor de la

distribución es responsable de realizar el refresco. En situaciones especiales se pueden establecer intervalos muy cortos de refresco o el propietario puede prohibir las copias.

- En cualquier sistema que tenga una base de datos distribuida, puede existir un servidor de nombres en particular realizando consultas que sólo puedan ser respondidas por algún otro servidor. Las dos formas generales de ver esto son la "recursividad", en la cual el primer servidor prosigue con la consulta del cliente en otro servidor, y la "iteratividad", la cual el servidor refiere al cliente a otro servidor y deja que el cliente prosiga con la consulta. Ambos procesos tienen ventajas y desventajas, pero la "iteratividad" es la preferida para los estilos de datagramas de acceso. Los sistemas de dominios requieren una implementación de la iteratividad, y permiten la recursividad opcionalmente.

El sistema de dominios asume que todos los datos originados en los ficheros maestros se distribuyen a los hosts del sistema de dominios. Estos ficheros maestros son actualizados por administradores de sistema locales. Los ficheros maestros son ficheros de texto legibles por un servidor de nombres local, y de esta manera se hace disponible desde los servidores de nombres a los usuarios del sistema de dominio. Los programas de usuario acceden a los servidores de nombres a través de programas estándar llamados resolutores.

El formato estándar de los ficheros maestros permite que pueda ser intercambiado entre hosts (vía FTP, mail, u otro mecanismo); esta ventaja es útil cuando una organización quiere un dominio, pero no quiere un servidor de nombres. La organización pueden mantener los ficheros maestros de forma local utilizando un editor de texto, enviarlos a un host remoto fuera de la organización que ejecuta un servidor de nombres, y por tanto coordinar con el administrador de sistemas del servidor de nombres para cargar los ficheros.

Los servidores de nombres de cada host y los resolutores son configurados por un administrador local de sistemas [RFC-1033]. En cada servidor de nombres, estos datos de configuración incluyen la identidad de los ficheros maestros locales e instrucciones en cada fichero maestro no local para cargarse en servidores fuera de la organización. El servidor de nombres utiliza los ficheros maestros o copias para cargar sus zonas. En el caso de los resolutores, los datos de configuración identifican a los servidores de nombres que deben ser primarios.

El sistema de dominio define los procedimientos para acceder a los datos y para referirse a otros servidores de nombres. El sistema de dominio también define los procedimientos para cachear datos y para refrescos periódicos de los datos definidos por el administrador de sistemas.

El administrador de sistemas ofrece:

- La definición de los límites de zona.
- Los ficheros maestros de datos.
- Actualizaciones de ficheros maestros.
- Establecimiento de las políticas deseadas de refresco.

El sistema de dominios ofrece:

- Formatos estándar para datos de recursos.
- Métodos estándar para consultar la base de datos.
- Métodos estándar para refrescar los datos locales de los servidores de nombres desde servidores de nombres fuera de la organización.

2.4. Elementos de un DNS

Un DNS tiene tres componentes principales:

- El ESPACIO DE NOMBRES DE DOMINIO y REGISTROS DE RECURSOS, que son especificaciones para un árbol estructurado de espacio de nombres y datos asociados con los nombres. Conceptualmente, cada nodo e hijo del árbol del espacio de nombres del dominio nombra a un conjunto de información, y solicita operaciones para extraer tipos de información específicos de un conjunto en particular. Una consulta nombra el nombre de dominio de interés y describe el tipo de recurso de información deseado. Por ejemplo, Internet utiliza alguno de sus nombres de dominio para identificar hosts; las consultas para direcciones de recursos devuelven la dirección del host de Internet.

- SERVIDORES DE NOMBRES, que son programas de servidor donde se aloja la información de la estructura de un árbol de dominio y la establece. Un servidor de nombres puede cachear la estructura o establecer información en cualquier parte del árbol de dominio, pero en general, un servidor de nombres en particular tiene toda la información de los subconjuntos del espacio de dominios, y punteros a otros servidores de nombres que pueden utilizarse para indicar información de cualquier parte del árbol de dominio. Los servidores de nombres conocen las partes del árbol de dominio para las cuales tienen información completa; un servidor de nombres es la AUTORIDAD para esas partes del espacio de nombres. La información Autoritativa se organiza dentro de unidades llamadas ZONAS, y esas zonas pueden distribuirse automáticamente en los servidores de nombres para ofrecer un servicio redundante para los datos de la zona.

- RESOLUTORES, que son programas que extraen información de servidores de nombres en respuesta a consultas de los clientes. Los resolutores deben tener acceso al menos a un servidor de nombres que pueda responder directamente la consulta, o proseguir con la consulta utilizando referencias a otros servidores de nombres. Un resolutor normalmente es una rutina de sistema directamente accesible por programas de usuario; de ahí que no sea necesario un protocolo entre el resolutor y el programa de usuario.

Estos tres componentes se corresponden básicamente a los tres niveles o vistas de un sistema de dominio:

- Desde el punto de vista del usuario, se accede al sistema de dominio con un procedimiento simple o una llamada del SO a un resolutor local. El espacio de nombres consiste en un sólo árbol y el usuario puede consultar información de cualquier sección del árbol.

- Desde el punto de vista del resolutor, el sistema de dominio se compone de un número desconocido de servidores de nombres.

  Cada servidor de nombres tiene una o más piezas del total de los datos del árbol, pero el resolutor ve cada una de esas bases de datos, esencialmente, de una forma estática.

- Desde el punto de vista del servidor de nombres, el sistema de dominio consiste en conjuntos separados de información local llamadas zonas. El servidor de nombres tiene copias locales de algunas de las zonas. El servidor de nombres debe refrescar sus

zonas de forma periódica desde las copias maestras de los ficheros locales o de servidores de nombres fuera de la organización. El servidor de nombres debe procesar concurrentemente las consultas que lleguen desde los resolutores.

Por el interés del rendimiento, las implementaciones pueden cumplir estas funciones. Por ejemplo, un resolutor ubicado en la misma máquina que el servidor de nombres puede compartir una base de datos que incluya las zonas gestionadas por el servidor de nombres y la caché gestionada por el resolutor.

3. ESPACIO DE NOMBRES DE DOMINIO y REGISTROS DE RECURSOS

3.1. Especificaciones y terminología del espacio de nombres

El espacio de nombres de dominio es una estructura de árbol. Cada nodo y hoja en el árbol corresponde a un conjunto de recursos (que debe estar vacío). El sistema de dominio no distingue entre el uso de los nodos interiores y hojas, y este memorándum utiliza el término "nodo" para referirse a ambos.

Cada nodo tiene una etiqueta, con un tamaño que varía entre 0 y 63 octetos. Los nodos hermanos no tienen porqué tener la misma etiqueta, mientras que la misma etiqueta puede ser utilizada por nodos que no son hermanos. Una etiqueta está reservada, que es null (tamaño cero) y es utilizada por el root.

El nombre de dominio de un nodo es la lista de etiquetas en la ruta del nodo desde la raíz del árbol. Por convención, las etiquetas que componen un nombre de dominio se leen de izquierda a derecha, desde la más específica (la más baja, lejos del raíz) a la menos específica (la más alta, cerca de la raíz).

Internamente, los programas que manipulan nombres de dominio deben representarlos como secuencias de etiquetas, donde cada etiqueta tiene un octeto de tamaño seguido de un octeto string. Debido a que todos los nombres de dominio terminan en el raíz, que tiene el string null de una etiqueta, estas representaciones internas pueden utilizar un tamaño de byte cero para terminar un nombre de dominio.

Por convención, los nombres de dominio pueden guardarse en mayúsculas y/o minúsculas, pero las comparaciones para todas las funciones presentes de un dominio se realizan sin tener en cuenta las mayúsculas/ minúsculas, asumiendo el conjunto de caracteres ASCII, y con orden ascendiente desde el bit 0. Esto quiere decir que puedes crear un nodo con la etiqueta "A" o un nodo con la etiqueta "a", pero no ambas como hermanas; puedes referirte a ellas como "a" o "A". Cuando recibes un nombre de dominio o etiqueta, puedes utilizar este caso. Lo racional de esta opción es que algún día puede que necesites añadir nombres de dominio completamente en binario para nuevos servicios; respetando los servicios existentes.

Cuando un usuario necesita introducir un nombre de dominio, se omite la longitud de cada etiqueta y las etiquetas se separan con puntos ("."). Como un nombre de dominio completo termina con la etiqueta raíz, éste debería terminar con un punto. Esta propiedad sirve para distinguir entre:

- un string de caracteres que representan un nombre de dominio completo (a menudo llamado "absoluto"). Por ejemplo, "poneria.ISI.EDU."
- un string de caracteres que representan el comienzo de etiquetas de un nombre de dominio incompleto, y debería ser completado por algún software local con información del dominio local (a menudo

llamado "relativo"). Por ejemplo, "poneria" dentro del dominio ISI.EDU.

Los nombres relativos pertenecen a un origen bien conocido o a una búsqueda en una lista de dominios. Los nombres relativos aparecen con mayor frecuencia en la interfaz de usuario, donde su representación varía entre implementaciones. También aparecen en ficheros maestros donde son relativos a un sólo nombre de dominio origen. La interpretación más común utiliza el raíz "." como el origen, o uno de los miembros de la búsqueda en la lista, por eso un nombre relativo con varias etiquetas suele omitir el punto "." de origen para ahorrar un caracter.

Para simplificar las implementaciones, el número total de octetos que representan un nombre de dominio (la suma de todos los octetos de todas las etiquetas y la longitud de las etiquetas) está limitado a 255.

A un dominio se le identifica por un nombre de dominio, y consiste en la parte del espacio de nombres de dominios que pertenece al nombre de dominio que especifica el dominio. Un dominio es un subdominio de otro dominio si está contenido en un dominio. Esta relación se puede comprobar si el nombre del subdominio termina con el nombre de dominio donde está contenido. Por ejemplo, A.B.C.D es un subdominio de B.C.D, C.D, D, y " ".

3.2. Pautas administrativas en uso

Como parte de la política, las especificaciones técnicas DNS no siguen una estructura de árbol en particular o reglas para seleccionar etiquetas; su meta es ser lo más general posible, y puede utilizarse para crear aplicaciones de forma arbitraria. El sistema fue diseñado para que el espacio de nombres no tenga porqué ser organizado dentro de los límites de red, servidores de nombres, etc. El razonamiento de esto no es que el espacio de nombres tenga una semántica implícita, sino que la elección de la semántica debe ser abierta para el problema tratado, y que las diferentes partes del árbol pueden tener semánticas implícitas distintas. Por ejemplo, el dominio IN-ADDR.ARPA está organizado y distribuido por redes y direcciones de host porque su rol es traducir desde números de red o de host a nombres; los dominios NetBIOS [RFC-1001], [RFC-1002] son planos porque así lo pide su aplicación.

De todas formas, hay algunas pautas que aplicar a las partes "normales" del espacio de nombres utilizado para hosts, direcciones de correo, etc., que hacen que el espacio de nombres sea más uniforme, extensible y que minimiza problemas de conversión de tablas de host antiguas. Las decisiones políticas de los niveles altos de el árbol tienen origen en el RFC-920. La política actual para los niveles altos se expone en el [RFC-1032]. La conversión MILNET se trata en el [RFC-1031].

Los dominios inferiores que, de forma eventual se dividen en múltiples zonas, deben proporcionar ramas en lo alto del dominio para que esa división eventual pueda realizarse sin cambiar nombres. Las etiquetas de nodos que utilizan caracteres especiales, como dígitos de control, etc., pueden no funcionar en software antiguo que dependa de opciones más restrictivas.

3.3. Pautas técnicas en uso

Antes de que DNS se utilice para manejar información de nombres para alguna clase de objeto, necesitamos dos cosas:

- Una convención para mapear entre nombres de objetos y nombres de dominio. Esto describe el acceso a la información de un objeto.

- Tipos de RR y formatos de datos para describir el objeto.

Estas reglas pueden ser muy simples o muy complejas. Muy a menudo, el diseñador debe tomar los formatos de cuenta existentes y planificarlos para permitir la compatibilidad futura. Pueden requerirse múltiples mapeos o niveles de mapeos.

Para los hosts, el mapeo depende de la sintaxis existente de los nombres de hosts que son un subconjunto de la representación normal en texto de los nombres de dominio, que, junto con los formatos RR describen direcciones de hosts, etc. Como necesitamos un mapeo inverso fiable de direcciones a nombres de hosts, dentro del dominio IN-ADDR.ARPA también se define un mapeo de direcciones especial.

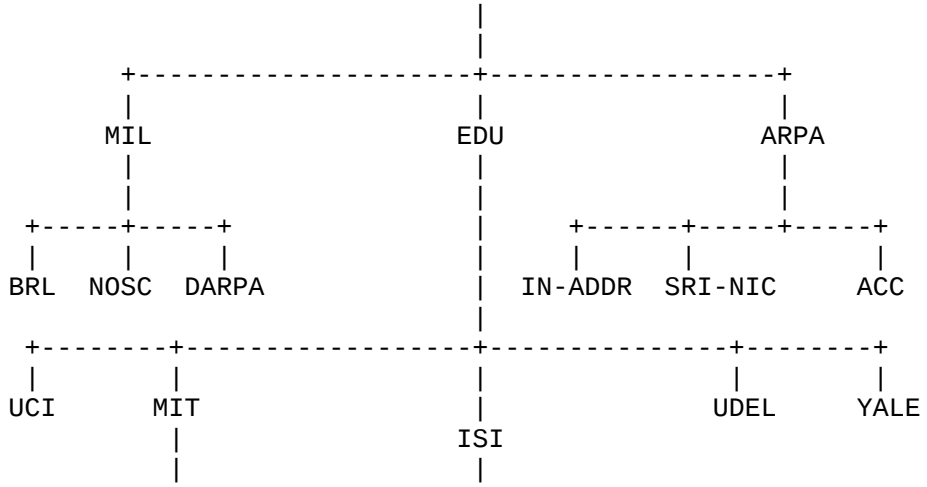
Para direcciones de correo, el mapeo es un poco más complejo. La dirección típica de mail <parte-local>@<dominio-mail> se mapea en nombre de dominio mediante la conversión de <parte-local> en una sola etiqueta (sin los puntos que pueda contener), convirtiendo <dominio-mail> en un nombre de dominio utilizando el formato de texto habitual para nombres de dominio (los puntos indican la separación de etiquetas), y se concatenan las dos partes para formar un sólo nombre de dominio.

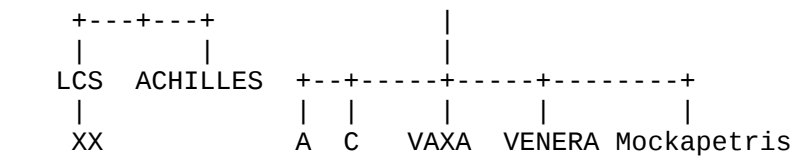
De esta manera, la dirección de correo HOSTMASTER@SRI-NIC.ARPA es representado en nombre de dominio como HOSTMASTER.SRI-NIC.ARPA. Es conveniente saber que las razones detrás de este diseño también deben tomarse del contexto de cuentas para los servidores de correo [RFC-974].

La definición de estas reglas no conciernen al usuario típico, pero debe entender que normalmente estas reglas son el resultado de numerosos compromisos entre los deseos de tener una alta compatibilidad con el pasado, interacciones entre diferentes definiciones de objetos, y la inevitable urgencia de añadir nuevas características cuando se definen las reglas. La forma en que el DNS soporta algunos objetos es a menudo más crucial que las restricciones inherentes de tal.

3.4. Ejemplo de espacio de nombres

La siguiente figura muestra una parte del espacio de nombres de dominio actual, y es utilizado en varios ejemplos de este RFC. Nótese que el árbol es un subconjunto más pequeño del actual espacio de nombres.





En este ejemplo, el dominio raíz tiene tres subdominios inmediatos: MIL, EDU, y ARPA. El dominio LCS.MIT.EDU tiene un subdominio inmediato llamado XX.LCS.MIT.EDU. Todas las hojas también son dominios.

3.5. Sintaxis de nombres preferida

Las especificaciones DNS intentan ser lo más generales posibles en cuanto a las reglas para construir nombres de dominio.

La idea es que pueda expresarse cualquier objeto existente como un nombre de dominio con los mínimos cambios. Sin embargo, cuando asignamos un nombre de dominio a un objeto, el usuario prudente seleccionará un nombre que cumpla las reglas del sistema de dominios y las del objeto, ya estén publicadas o implícitas por los programas existentes.

Por ejemplo, cuando nombramos a un dominio mail, el usuario puede cumplir las reglas de este memorándum y las del RFC-822. Cuando creamos un nuevo nombre de host, se seguirán las antiguas reglas para HOSTS.TXT. Esto evita problemas con el software antiguo a la hora de utilizar nombres de dominio.

La siguiente sintaxis da pocos problemas con muchas aplicaciones que utilizan nombres de dominio (mail, TELNET).

<dominio> ::= <subdominio> | " "

<subdominio> ::= <etiqueta> | <subdominio> "." <etiqueta>

<etiqueta> ::= <letra> [[<ldh-str>] <let-dig>]

<ldh-str> ::= <let-dig-hyp> | <let-dig-hyp> <ldh-str>

<let-dig-hyp> ::= <let-dig> | "-"

<let-dig> ::= <letter> | <digit>

<letra> ::= cualquiera de los 52 caracteres alfabéticos desde la A a la Z, y de la a a la z

<digito> ::= cualquiera de los 10 dígitos entre 0 y 9

Nótese que mientras se tiene en cuenta el uso de mayúsculas y minúsculas en los nombres de dominio, no es significativo. Esto es, dos nombres con las mismas letras pero unas con mayúsculas y otras minúsculas se tratan de igual forma.

Las etiquetas deben seguir las reglas de los nombres de hosts de ARPANET. Deben comenzar con una letra, terminar con una letra o dígito, y sólo pueden tener contenido de letras, dígitos y guiones. Las etiquetas tienen un límite de 63 caracteres.

Por ejemplo, los siguientes strings identifican a los hosts en Internet:

A.ISI.EDU XX.LCS.MIT.EDU SRI-NIC.ARPA

3.6. Registros de recursos

Un nombre de dominio identifica a un nodo. Cada nodo tiene un conjunto

de información del recurso, que debe estar vacío. El conjunto de información del recurso asociado con un nombre particular está compuesto por distintos registros de recursos (RRs). El orden de los RRs en un conjunto no es significativo y no es necesario en los servidores de nombres, resolutores, u otras partes del DNS.

Cuando hablamos de un RR específico, asumimos que tiene lo siguiente:

propietario es el nombre de dominio donde se encuentra.

tipo es un valor codificado de 16 bit que especifica el tipo de recurso en este registro de recursos. Los tipos se refieren a recursos abstractos.

Este memorándum utiliza los siguientes tipos:

A	una dirección de host
CNAME	identifica a un nombre canónico de un alias
HINFO	identifica la CPU y el OS del host
MX	identifica a un servidor de correo para el dominio [RFC-974]
NS	el servidor de nombres autoritativo para el dominio
PTR	puntero a otra parte del espacio de nombres de dominio
SOA	identifica el comienzo de la zona de autoridad

clase valor codificado de 16 bits que identifica a una familia de protocolos o una instancia de un protocolo.

Este memorándum utiliza las siguientes clases:

IN	el sistema de Internet
CH	el sistema Caos

TTL tiempo de vida del RR. Este campo es un entero de 32 bits en unidades de segundo, y fundamentalmente lo utilizan los resolutores cuando cachean RRs. El TTL describe el tiempo de duración que puede estar cacheado un RR antes de que pueda ser eliminado.

RDATA es el tipo y a veces los datos que dependen de la clase que describe el recurso:

A	Para la clase IN, dirección IP de 32 bit Para la clase CH, un nombre de dominio seguido de una dirección Caos octal de 16 bit.
CNAME	un nombre de dominio.
MX	un valor preferentemente de 16 bit (si tiene menos mejor) seguido de un nombre de host que actúe como servidor de

	correo para el dominio propietario.
NS	un nombre de host.
PTR	un nombre de dominio.
SOA	varios campos.

El nombre propietario es implícito muy a menudo en vez de formar parte integral del RR. Por ejemplo, muchos servidores de nombres internamente forman una estructura de árbol o hash para el espacio de nombres, y encadenan RRs fuera de los nodos. El resto de las partes del RR son la cabecera indexada (tipo, clase, TTL) que permanece para todos los RRs, y la parte variable (RDATA) que mantiene las necesidades del recurso descrito.

El TTL es el límite de tiempo de duración de un RR en caché. Este límite no se aplica a los datos autoritativos en las zonas, aunque estos también tienen un tiempo de vida, pero controlado por las políticas de refresco de la zona. El TTL es asignado por el administrador de la zona donde se origina el dato. Mientras que los TTLs cortos se utilizan para minimizar la caché, y cero para prohibir el cacheo, la realidad en el rendimiento de Internet sugiere que esos tiempos sean de días para el típico host. Si se tiene en cuenta algún cambio, el TTL puede reducirse antes del cambio para minimizar inconsistencias durante el cambio, y después del cambio se incrementa el TTL para dejarlo como estaba.

Los datos en la sección RDATA de los RRs son una combinación de strings binarios y nombres de dominio. Los nombres de dominio, con frecuencia se utilizan como "punteros" a otros datos en el DNS.

3.6.1. Expresión textual de los RRs

Los RRs se representan de forma binaria en los paquetes del protocolo DNS, y normalmente se representan en alto nivel cuando se guardan en un servidor de nombres o resolutor. En este memorándum, adoptaremos un estilo similar al utilizado en los ficheros maestros para mostrar el contenido de los RRs. En este formato, la mayoría de los RRs se muestran en una línea, aunque es posible que las líneas que tenga a continuación utilicen referentes.

El comienzo de la línea es el propietario del RR. Si la línea comienza con un blanco, se asume que el propietario es el mismo que el del RR anterior. Las líneas en blanco a menudo se incluyen para una mejor visualización.

Después del propietario, va el TTL, el tipo, y la clase del RR. La clase y el tipo utilizan la mnemotécnia de antes, y el TTL es un entero antes del campo tipo. Para evitar ambigüedades en la sintaxis, la mnemotécnia del tipo y la clase son distintas, los TTLs son enteros, y la mnemotécnia del tipo es siempre la última. Para más claridad, a menudo se omiten los valores de la clase IN y el TTL en los ejemplos.

Los datos de recursos de la sección RDATA de un RR dado utiliza la representación típica de esos datos.

Por ejemplo, la representación de los RRs de un mensaje es:

```
    ISI.EDU.        MX      10 VENERA.ISI.EDU.
                    MX      10 VAXA.ISI.EDU.
    VENERA.ISI.EDU. A       128.9.0.32
                    A       10.1.0.52
    VAXA.ISI.EDU.   A       10.2.0.27
                    A       128.9.0.33
```

El RRs del MX tiene una sección RDATA que consiste en un número de 16 bit seguido de un nombre de dominio. La dirección de los RRs es una IP estándar de Internet de 32 bit.

El ejemplo anterior muestra seis RRs, con dos RRs para cada uno de los tres nombres de dominio.

De la misma forma veremos:

```
XX.LCS.MIT.EDU. IN      A       10.0.0.44
                CH      A       MIT.EDU. 2420
```

Este ejemplo muestra dos direcciones para XX.LCS.MIT.EDU, cada una con dos clases distintas.

3.6.2. Alias y nombres canónicos.

En los sistemas existentes, los hosts y otros recursos tienen a menudo varios nombres para identificar el mismo recurso. Por ejemplo, los nombres C.ISI.EDU y USC-ISIC.ARPA identifican al mismo host. De forma similar, en el caso de direcciones de correo, muchas organizaciones proporcionan varios nombres que apuntan a la misma dirección de correo; por ejemplo Mockapetris@C.ISI.EDU, Mockapetris@B.ISI.EDU, y PVM@ISI.EDU apuntan a la misma dirección de correo (aunque el mecanismo que hay detrás de esto es un tanto complicado).

La mayoría de estos sistemas saben que uno de los nombres es el canónico o principal y los otros son alias.

El sistema de dominios proporciona una característica del uso del RR del nombre canónico (CNAME). Un RR CNAME identifica a su nombre propietario como un alias y especifica el correspondiente nombre canónico en la sección RDATA del RR. Si en un nodo hay un RR CNAME, no debería haber más datos presentes; así nos aseguramos de que los datos para el nombre canónico y sus alias no sean distintos. Esta regla también asegura que un CNAME cacheado puede ser utilizado sin que sea comprobado por un servidor autoritativo para otros tipos de RR.

Los RRs CNAME tienen una acción especial en el software DNS. Cuando un servidor de nombres no encuentra el RR deseado en el conjunto de recursos asociados con el nombre de dominio, comprueba si el conjunto de recursos es un registro CNAME con una clase que coincida. Si es así, el servidor de nombres incluye el registro CNAME en la respuesta y reinicia la consulta en el nombre de dominio especificado en el campo de datos del registro CNAME. La única excepción a esta regla es que las consultas que coincidan con el tipo de CNAME no se reiniciarán.

Por ejemplo, supongamos que un servidor de nombres procesó una consulta para USC-ISIC.ARPA, preguntando por la información del tipo A, y tiene los siguientes registros de recursos:

```
USC-ISIC.ARPA   IN      CNAME   C.ISI.EDU

C.ISI.EDU       IN      A       10.0.0.52
```

Ambos RRs pueden devolverse en la respuesta, mientras que una consulta del tipo CNAME o una consulta * puede devolver el CNAME.

Los nombres de domino en los RRs que apunten a otro nombre siempre deberán apuntar al nombre primario y no a un alias. Esto evita accesos indirectos a la información. Por ejemplo, la dirección para nombrar el RR del ejemplo anterior debería ser:

```
52.0.0.10.IN-ADDR.ARPA  IN      PTR     C.ISI.EDU
```

en vez de que apunte a USC-ISIC.ARPA. Bien, por el principio de fiabilidad, el software de dominio no debería fallar cuando se presenten cadenas o bucles CNAME; las cadenas CNAME han de ser seguidas y los bucles CNAME serían errores.

3.7. Consultas

Las consultas son mensajes que pueden ser enviados a un servidor de nombres para obtener una respuesta.

En Internet, las consultas se transportan en datagramas UDP o sobre conexiones TCP. La respuesta del servidor de nombres puede ser la respuesta en sí a la consulta, referencias a otro conjunto de servidores de nombres, o algún error.

Generalmente, el usuario no realiza las consultas directamente, las envía al resolutor y éste envía una o más consultas a los servidores de nombres. Después, el resolutor reparte los errores y las referencias resultantes. Las posibles consultas que pueden realizarse dependen de los tipos de servicios que soporte el resolutor.

Las consultas y respuestas tienen un formato de mensaje estándar. El formato del mensaje tiene una cabecera que contiene un número de campos ordenados que siempre están presentes, y contienen cuatro secciones donde van los parámetros de la consulta y los RRs.

El campo más importante de la cabecera es un campo de cuatro bit llamado opcode que separa las distintas consultas. De los 16 valores posibles, uno (consulta estándar) forma parte del protocolo oficial, dos (consulta inversa y estado de la consulta) son opcionales, uno (terminación) no se utiliza, y el resto no está asignado.

Las cuatro secciones son:

Pregunta	Lleva el nombre de la consulta y otros parámetros de tal.
Respuesta	Lleva los RRs que responden directamente a la consulta.
Autoridad	Lleva los RRs que describen a otros servidores autoritativos. Pueden llevar el RR SOA para los datos autoritativos en la sección de respuesta.
Adicional	Lleva los RRs que pueden ser útiles para otros RRs de otras secciones.

Nótese que el contenido, no el formato, de estas secciones varía con el opcode de la cabecera.

3.7.1. Consultas estándar

Una consulta estándar especifica un nombre de dominio destino (QNAME), un tipo de consulta (QTYPE), y una clase de consulta (QCLASS) y pregunta por los RRs que coincidan. Este de tipo de consulta engloba a la mayoría de consultas DNS que utilizan el término "query" y es el estándar a menos que se indique otra cosa. Los campos QTYPE y QCLASS son de 16 bit, y son superconjuntos de las clases y tipos definidos.

El campo QTYPE debe contener:

<cualquier tipo>	coincide con el tipo (A, PTR).
AXFR	QTYPE de transferencia especial de zona.

MAILB coincide con todas las direcciones de correo de los RRs (MB y MG)

* coincide con todos los tipos del RR.

El campo QCLASS debe contener:

<cualquier clase> coincide con la clase (IN, CH).

* coincide con todas las clases del RR.

Mediante el nombre de dominio de la consulta, QTYPE, y QCLASS, el servidor de nombres busca a los RRs que coincidan. Adicionalmente, para los registros relevantes, el servidor de nombres puede devolver RRs que apunten hacia el servidor de nombres que tiene la información deseada o RRs que se esperan que sean útiles para la interpretación de los RRs relevantes. Por ejemplo, un servidor de nombres que no tiene la información requerida puede conocer a un servidor de nombres que sí la tiene; un servidor de nombres que devuelve un nombre de dominio en un RR relevante puede también devolver el RR que une el nombre de dominio con una dirección.

Por ejemplo, un email que se envía a Mockapetris@ISI.EDU puede preguntar al resolutor por la información del servidor de correo de ISI.EDU en la forma QNAME=ISI.EDU, QTYPE=MX, QCLASS=IN. La respuesta de la sección de la respuesta en sí puede ser:

```
ISI.EDU.        MX      10 VENERA.ISI.EDU.
                MX      10 VAXA.ISI.EDU.
```

mientras que la sección adicional puede ser:

```
VAXA.ISI.EDU.   A       10.2.0.27

                A       128.9.0.33
VENERA.ISI.EDU. A       10.1.0.52
                A       128.9.0.32
```

Debido a que el servidor asume que si el origen quiere información del servidor de correo, es probable que quiera las direcciones de los otros servidores de correo después.

Nótese que QCLASS=* requiere una interpretación especial respecto a la autoridad. Debido a que un servidor de nombres en particular puede no conocer todas las clases disponibles del sistema de dominio, puede que nunca sepa si es autoritativo para todas las clases. De ahí que las respuestas a QCLASS=* nunca son autoritativas.

3.7.2. Consultas inversas (opcionales)

Los servidores de nombres también pueden realizar consultas inversas que mapean un recurso en particular a un nombre de dominio o nombres de dominio que tengan ese recurso. Por ejemplo, mientrs que una consulta estándar puede mapear un nombre de dominio a un RR SOA, la consulta inversa correspondiente puede mapear el RR SOA a un nombre de dominio.

La implementación de este servicio en el servidor de nombres es opcional, pero todos los servidores de nombres deben ser capaces al menos de entender una consulta inversa y devolver una respuesta de error en caso de que no soporte ese tipo de consultas.

El sistema de dominio no puede garantizar las consultas inversas porque el sistema de dominio está organizado por nombre de dominio en vez de

direcciones de host o cualquier otro tipo de recurso. Las consultas inversas son fundamentalmente útiles para depuración y tareas de mantenimiento de la base de datos.

Las consultas inversas no pueden devolver el TTL apropiado, y no pueden saber si el RR en cuestión pertenece a un conjunto o no (por ejemplo, una dirección de un host que tenga múltiples direcciones). Por tanto, los RRs devueltos en las consultas inversas nunca deben cachearse.

Las consultas inversas NO son un método aceptable para mapear direcciones de host a nombres de host; utiliza mejor el dominio IN-ADDR.ARPA.

En el [RFC-1035] hay una exposición detallada de las consultas inversas.

3.8. Consultas de estado (Experimental)

Por definir.

3.9. Consultas de terminación (Obsoleto)

Los servicios opcionales de terminación descritos en los RFCs 882 y 883 han sido eliminados. Los servicios rediseñados pueden estar disponibles en un futuro, o los opcodes pueden ser utilizados para otro uso.

4. SERVIDORES DE NOMBRES

4.1. Introducción

Los servidores de nombres son los repositorios de información que forman la base de datos del dominio. La base de datos se divide en secciones llamadas zonas, que están distribuidas entre los servidores de nombres. Mientras que los servidores de nombres pueden tener varias funciones opcionales y recursos de datos, la tarea esencial de un servidor de nombres es responder consultas utilizando los datos de sus zonas. Por diseño, los servidores de nombres pueden responder consultas de una manera simple; la respuesta siempre puede ser generada utilizando datos locales o pueden dar una referencia a otros servidores de nombres "cercanos" a la información deseada.

Una zona dada puede estar disponible desde varios servidores de nombres para asegurar su disponibilidad en caso de un fallo de comunicaciones o de host. Por decreto administrativo, requerimos que cada zona esté disponible al menos en dos servidores, y muchas zonas tienen mucha más redundacia.

Un servidor de nombres dado soportará normalmente una o más zonas, pero esto sólo le dará información autoritativa de una pequeña sección del árbol de dominio. También debe haber datos no autoritativos cacheados en otras partes del árbol. El servidor de nombres marca sus respuestas a las preguntas para que el origen pueda saber si las respuestas vienen de datos autoritativos o no.

4.2. División de la base de datos en zonas

La base de datos se particiona de dos formas: por clases, y por "cortes" entre nodos del espacio de nombres.

La partición por clases es simple. La base de datos para cualquier clase se organiza, se delega y se mantiene de forma separada del resto de otras clases. Por convención, los espacios de nombres son los mismos para todas las clases. Las clases separadas puede tomarse como una tabla de árboles de espacios de nombres paralelos. Nótese que los datos de los nodos serán diferentes para diferentes clases paralelas. Las razones más

comunes para crear una nueva clase es la necesidad de un nuevo formato de datos para tipos existentes o el deseo de tener un manejo separado de otra versión del espacio de nombres existente.

En una clase, se puede realizar un "corte" en el mismo espacio de nombres entre dos nodos adyacentes. Después del corte, cada grupo de espacios de nombres conectados son una zona separada. Se dice que la zona es autoritativa para todos los nombres en la región de conexión. Nótese que el "corte" en el espacio de nombres puede estar en diferentes sitios para diferentes clases, los servidores de nombres pueden ser distintos, etc.

Estas reglas dicen que cada zona tiene al menos un nodo, y por tanto un nombre de dominio, para el cual es autoritativo, y todos los nodos de una zona en particular están conectados. Dada la estructura de árbol, cada zona tiene un nodo superior cercano al raíz más elevado que cualquier otro nodo en la zona. El nombre de este nodo se utiliza a menudo para identificar la zona.

Es posible, aunque no muy útil, particionar el espacio de nombres para que cada nombre de dominio esté en una zona separada o para que todos los nodos estén en sólo una zona. En vez de esto, la base de datos se particiona en puntos donde una organización en particular quiere tener control sobre un subárbol.

Una vez que una organización controla su propia zona, puede, de forma unilateral cambiar los datos de la zona, crear nuevas secciones en el árbol conectadas a la zona, borrar nodos, o delegar nuevas subzonas bajo su zona.

Si la organización tiene una subestructrua, puede querer realizar delegaciones anidadas para el control del espacio de nombres. En algunos casos, esas divisiones son realizadas para un buen mantenimiento de la base de datos.

4.2.1. Consideraciones técnicas

Los datos que describen una zona tienen cuatro partes principales:

- Datos autoritativos para todos los nodos dentro de la zona.
- Datos que definen el nodo superior de la zona (pensando en que forman parte de los datos autoritativos).
- Datos que describen subzonas delegadas (cortes sobre el fin de la zona.
- Datos que permiten el acceso a subzonas de servidores de nombres (a veces llamados datos "glue" (pegamento)).

Todos estos datos se expresan como RRs, por eso una zona puede ser totalmente descrita en el término de un conjunto de RRs. Zonas enteras pueden transferirse entre servidores de nombres transfiriendo los RRs, o transportados en series de mensajes, o subiendo el fichero maestro mediante FTP, el cual es una representación textual.

Los datos autoritativos para una zona simplemente son todos los RRs vinculados a todos los nodos desde el superior de la zona hasta los nodos hojas o nodos del punto final del límite de la zona.

Los RRs que describen el nodo más superior de la zona son especialmente importantes para la gestión de la zona (lógicamente están pensados como parte de los datos autoritativos). Estos RRs son de dos tipos: RRs del servidor de nombre que listan, uno por RR, a todos los servidores para

la zona, y un sólo RR SOA que describe los parámetros de gestión de la zona.

Los RRs que describen el límite del final de zona son RRs NS que nombran a los servidores de las subzonas. Debido a que los cortes están entre nodos, estos RRs NO forman parte de los datos autoritativos de la zona, y deberían ser exactamente los mismos que los correspondientes RRs en el nodo superior de la subzona. Debido a que los servidores de nombres siempre están asociados con límites de zona, los RRs NS sólo se encuentran en nodos que están en el nodo superior de alguna zona. En los datos que hacen posible una zona, los RRs NS están en el nodo superior de la zona (y son autoritativos) y realizan el corte al final de la zona (donde no son autoritativos), pero nunca en medio.

Una de las metas de la estructura de la zona es que cualquier zona tiene todos los datos requeridos para establecer comunicaciones con los servidores de nombres para cualquier subzona. Esto es, las zonas padres tienen toda la información necesaria para acceder a los servidores de las zonas hijas. Los RRs NS que nombran a los servidores para las subzonas, a menudo no son suficientes para realizar esta tarea porque nombran a los servidores, pero no dan sus direcciones. En particular, si el nombre del servidor de nombres es él mismo en la subzona, podemos encontrarnos con la situación de que los RRs NS nos digan que para saber la dirección del servidor de nombres, tenemos que contactar con el servidor utilizando la dirección que queremos saber. Para solventar este problema, una zona contiene RRs "glue" (pegamento) que no forman parte de los datos autoritativos, y son RRs de direcciones para servidores. Estos RRs sólo son necesarios si el nombre del servidor de nombres está detrás del límite, y sólo es utilizado como parte de la referencia de la respuesta.

4.2.2. Consideraciones administrativas

Cuando alguna organización quiere el control de su propio dominio, el primer paso es identificar a la zona padre apropiada, y conseguir que los propietarios de la zona padre nos den la delegación del control.

Como no hay ninguna limitación técnica en particular para decir donde puede realizarse esto en el árbol, hay algunos grupos administrativos de difusión en el [RFC-1032] que tratan sobre el nivel superior de la organización, y en las zonas de niveles medios son libres de crear sus propias reglas. Por ejemplo, una universidad puede elegir utilizar una sóla zona, mientras que otros pueden elegir una organización con subzonas dedicadas a departamentos individuales o escuelas. El [RFC-1033] cataloga el software DNS disponible y debate sobre los procedimientos administrativos.

Una vez seleccionado el nombre apropiado de la nueva subzona, los nuevos propietarios deben demostrar la redundancia requerida del servidor de nombres. Nótese que aquí no es necesario que los servidores de una zona residan en un host con un nombre que esté en el dominio. En muchos casos, una zona será mucho más accesible para internet si sus servidores están distribuidos a lo largo de Internet, en contraposición de que los servidores de la zona estén ubicados en la misma dependencia de la organización que maneja la zona. Por ejemplo, en el DNS actual, uno de los servidores de nombres para el Reino Unido, o dominio UK, está ubicado en US. Esto permite a los hosts de US conseguir información de UK sin utilizar el ancho de banda limitado del transatlántico.

Como último paso de la instalación, la delegación de los RRs NS y RRs glue necesaria para realizar la delegación efectiva debería añadirse a la zona padre. Los administradores de ambas zonas pueden asegurarse de que los RRs NS y glue que marcan ambas caras del corte son consistentes en el tiempo.

4.3. Los servidores de nombres por dentro

4.3.1. Consultas y respuestas

La actividad principal de los servidores de nombres es responder consultas estándar. Tanto la consulta como su respuesta tienen un formato de mensaje estándar descrito en el [RFC-1035]. La consulta tiene un QTYPE, QCLASS, y un QNAME, que son los tipos y clases de la información deseada y el nombre en cuestión.

La forma en que el servidor de nombres responde depende de si opera en modo recursivo o no:

- El modo más simple del servidor es el no-recursivo, con esto puede responder consultas utilizando sólo la información local: la respuesta puede contener un error, la respuesta, o una referencia a otro servidor cercano a la respuesta. Todos los servidores de nombres deben implementar consultas no-recursivas.

- El modo más simple para el cliente es el recursivo. En este modo el servidor de nombres tiene el rol de un resolutor y devuelve un error o la respuesta, pero nunca una referencia a otro servidor. Este servicio es opcional en el servidor de nombres, y éste puede restringir su uso a los clientes.

El servicio recursivo es útil en varias situaciones:

- un cliente relativo que sólo pueda soportar una respuesta directa a la consulta.

- un cliente que necesite pasar de un protocolo a otro o que tenga otros límites y pueda enviar consultas a un servidor que puede actuar como intermediario.

- una red donde queramos centralizar la caché en vez de tener una caché separada para cada cliente.

Un servicio no-recursivo es apropiado si el origen es capaz de seguir las referencias y puede guardar información para futuras consultas.

El uso del modo recursivo está limitado a casos donde tanto el cliente como el servidor de nombres están deacuerdo en utilizarlo. Este acuerdo es negociado por el uso de dos bits en los mensajes de consulta y respuesta:

- Recursión disponible o bit RA, se activa o no por un servidor de nombres en todas sus respuestas. El bit está activo si el servidor de nombres está dispuesto a ofrecer un servicio recursivo al cliente, independientemente de si éste quiere un servicio recursivo o no para sus consultas. Esto es, la señal RA indica disponibilidad en vez de uso.

- Consultas que contienen un bit de deseo de recursión o RD. Este bit especifica si el cliente quiere el servicio de recursión para esta consulta. Los clientes pueden requerir el servicio de recursión de cualquier servidor de nombres si éste les ha enviado antes un RA, o si los servidores permiten ofrecer el servicio a través de un acuerdo privado o de alguna otra forma fuera del protocolo DNS.

El modo recursivo se produce cuando una consulta con RD llega al servidor que está dispuesto a ofrecer el servicio recursivo; el cliente puede verificar que el modo recursivo está disponible comprobando si RA y RD figuran en la respuesta. Nótese que el servidor de nombres nunca

podría realizar el servicio recursivo a menos que se le responda con un RD, porque podría dar problemas al servidor de nombres y a sus bases de datos.

Si se solicita el servicio de recursión y está disponible, la respuesta recursiva a una consulta será una de las siguientes:

- La respuesta a la consulta, posible prefacio de uno o más de un RRs CNAME que especifica los alias encontrados para la respuesta.

- Un error de nombre indicando que el nombre no existe. Esto puede incluir los RRs CNAME que indican que el nombre de la consulta original fue un alias para el nombre que no existe.

- Una indicación de error temporal.

Si el servicio de recursión no es solicitado o no está disponible, la respuesta no-recursiva será una de las siguientes:

- Un error autoritativo de nombre indicando que el nombre no existe.

- Una indicación de error temporal.

- Alguna combinación de:

  RRs que responden la consulta, junto con una indicación de si los datos vienen de una zona o están cacheados.

  Una referencia a servidores de nombres que tienen zonas que son cercanas antecesoras al nombre buscado.

- RRs que el servidor de nombres cree que serán útiles al origen.

4.3.2. Algoritmo

El algoritmo actual utilizado por el servidor de nombres dependerá del sistema operativo local y las estructuras de datos utilizadas para almacenar los RRs. El siguiente algoritmo asume que los RRs están organizados en varias estructuras de árbol, una por cada zona, y otra para la caché:

1. Establecer o no el valor de recursión disponible en la respuesta dependiendo de si el servidor de nombres dará el servicio de recursividad. Si el servicio de recursividad está disponible y es solicitado con el bit RD en la consulta, ir al paso 5, sino ir al paso 2.

2. Buscar en las zonas disponibles la zona más cercana al antecesor del QNAME. Si se encuentra dicha zona, ir al paso 3, sino ir al paso 4.

3. Comenzar a comparar, etiqueta por etiqueta de la zona. El proceso de comparación puede terminar de formas distintas:

   a. Si coincide el QNAME completo, hemos encontrado el nodo. Si los datos del nodo son un CNAME, y QTYPE no coincide con CNAME, copiar el RR CNAME dentro de la sección de respuesta, cambiar QNAME al nombre canónico en el RR CNAME, y volver al paso 1.

   En cualquier otro caso, copiar todos los RRs donde coincidan el QTYPE dentro de la sección de respuesta e ir al paso 6.

   b. Si una coincidencia nos lleva fuera de los datos

autoritativos, tendremos una referencia. Esto sucede cuando encontramos un nodo con RRs NS marcando cortes a lo largo de la parte inferior de una zona.

Copiar los RRs NS para la subzona dentro de la sección autoritativa de la respuesta. Poner las direcciones disponibles en la sección adicional, utilizando RRs glue si las direcciones no están disponibles desde los datos autoritativos o desde la caché. Ir al paso 4.

c. Si no se produce la coincidencia de alguna etiqueta (la etiqueta correspondiente no existe), mirar si la etiqueta "*" existe.

Si la etiqueta "*" no existe, comprobar si el nombre que buscamos es el QNAME original de la consulta o un nombre que hemos seguido por un CNAME. Si el nombre es el original, establecemos un error autoritativo de nombre en la respuesta y salimos. En cualquier otro caso salimos.

Si la etiqueta "*" no existe, comparar los RRs de este nodo con el QTYPE. Si hay alguna coincidencia, los copiamos en la sección de respuesta, pero establecemos a QNAME como propietario del RR, y no al nodo con la etiqueta "*". Vamos al paso 6.

4. Comenzar la comparación en la caché. Si encontramos al QNAME en la caché, copiamos todos los RRs adjuntos donde coincida el QTYPE dentro de la sección de respuesta. Si aquí no hubo delegación de datos autoritativos, buscar al mejor de la caché, y ponerlo en la sección autoritativa. Ir al paso 6.

5. Utilizar el resolutor local o una copia de su algoritmo (ver la sección del resolutor de este memorándum) para responder la consulta. Guardar los resultados, incluyendo cualquier CNAME intermedio, en la sección de respuesta de la respuesta.

6. Utilizando sólo datos locales, intentar añadir otros RRs que puedan ser útiles a la sección adicional de la consulta. Salir.

4.3.3. Wildcards

En el algoritmo anterior, los RRs con nombres de propietario que comiencen con la etiqueta "*" tienen un tratamiento especial. Estos RRs se conocen como wildcards. Los RRs wildcard son instrucciones para sintetizar RRs. Cuando se tienen las condiciones apropiadas, el servidor de nombres crea RRs con un nombre de propietario igual al nombre de la consulta y los contenidos los toma de los RRs wildcard.

Esta característica se utiliza a menudo para crear una zona que servirá para reenviar mail desde Internet a algún otro sistema de correo. La idea general es asumir que existe cualquier nombre presentado al servidor desde una consulta, para la zona correspondiente, cumpliendo ciertas propiedades, a menos que, por el contrario exista una evidencia explícita. Nótese que el uso del término zona utilizado, en vez de dominio, es intencionado; por defecto no se propaga por los límites de la zona, sin embargo una subzona puede parecer un dominio por similitud a las configuraciones por defecto.

Los contenidos de los RRs wildcard siguen las reglas usuales y formatos de los RRs. Los wildcards en la zona tienen un nombre propietario que controla los nombres de la consulta con los que coincida. El nombre propietario de los RRs wildcard tienen la forma "*.<cualquierdominio>, donde <cualquierdominio> es cualquier nombre de dominio.

<cualquierdominio> no debería contener otras etiquetas *, y debería ser autoritativo para la zona. Los wildcards se aplican potencialmente a descendientes de <cualquierdominio>, pero no para <cualquierdomino> en sí mismo.

Otra forma de ver esto es que la etiqueta "*" siempre coincide al menos con una etiqueta completa y a veces con más, pero siempre con etiquetas completas.

Los RRs Wildcard no funcionan:

- Cuando la consulta está en otra zona. Esto es, la delegación cancela el funcionamiento por defecto del wildcard.

- Cuando se sabe que el nombre de la consulta o un nombre entre el dominio del wildcard y el nombre de la consulta existe. Por ejemplo, si un RR wildcard tiene como nombre de propietario "*.X", y la zona también tiene RRs a B.X, los wildcards pueden aplicarse a consultas para el nombre Z.X (sabiendo que ahí no hay información explícita para Z.X), pero no para B.X, A.B.X, o X.

La etiqueta A * no tiene un efecto especial si aparece en un nombre de consulta, pero puede utilizarse para comprobar wildcards en una zona autoritativa; una consulta como esa es la única forma para conseguir una respuesta que contenga RRs con un nombre de propietario con * en él. El resultado de tal consulta no se cacheará.

Nótese que los contenidos de los RRs wildcard no se modifican cuando se utilizan para sintetizar RRs.

Para ilustrar el uso de los RRs wildcard, supongamos una compañía grande con una gran red (no TCP/IP) que quiere crear una puerta de enlace de mail. Si la compañía se llamara X.COM, y tuviera una máquina para la puerta de enlace que soporte TCP/IP llamada A.X.COM, los siguientes RRs deberían estar en la zona COM:

```
X.COM           MX      10      A.X.COM

*.X.COM         MX      10      A.X.COM

A.X.COM         A       1.2.3.4
A.X.COM         MX      10      A.X.COM

*.A.X.COM       MX      10      A.X.COM
```

Esto devolvería un RR MX de A.X.COM a cualquier consulta MX con cualquier nombre de dominio terminado en X.COM. Se requieren dos RRs wildcard porque el efecto del wildcard en *.X.COM es implícito en el subárbol A.X.COM por los datos explícitos de A.X.COM. También se requieren los datos MX explícitos de X.COM y A.X.COM, y uno de esos RRs deberían coincidir con el nombre de la consulta de XX.COM.

4.3.4. Cacheo de respuesta negativa (opcional)

El DNS ofrece un servicio opcional que permite a los servidores de nombres distribuir, y resolver en caché TTLs con resultados negativos.

Por ejemplo, un servidor de nombres puede distribuir un TTL con un error de nombre, y un resolutor al recibir tal información puede asumir que el nombre no existe durante el período del TTL sin consultar datos autoritativos. De forma similar, un resolutor puede hacer una consulta con un QTYPE que coincida con múltiples tipos, y cachear el hecho de que algunos tipos no están presentes.

Esta función puede ser particularmente importante en un sistema que implementa nombres cortos utilizados en listas de búsqueda debido a su uso extendido, requiriendo un sufijo hacia el final de la lista de búsqueda. Cuando se utilice esto, se generan múltiples errores de nombre.

El método radica en que el servidor de nombres puede añadir un RR SOA a la sección adicional de una respuesta cuando es autoritativa. El SOA debe estar en la zona donde estuvo el origen de los datos autoritativos en la sección de respuesta, o si procede, donde estuvo el error de nombre. El MINIMO campo del SOA controla el tiempo de cacheo del resultado negativo.

En algunas circunstancias, la sección de respuesta puede contener varios nombres de propietario. En este caso, el mecanismo SOA sólo puede ser utilizado para los datos que coincidan con QNAME, que es el único dato autoritativo de esta sección.

Los resolutores y servidores no deberían añadir SOAs a una sección adicional de una respuesta no autoritativa, o deducir resultados que no vengan directamente de la respuesta autoritativa. Existen varias razones para esto, incluyendo: la información cacheada normalmente no es suficiente para que coincidan los RRs y sus nombres de zona, los RRs SOA pueden cachearse en las consultas directas de SOA, y los servidores de nombres no son requeridos para sacar los SOAs de la sección autoritativa.

Esta funcionalidad es opcional, aunque se espera una versión refinada que forme parte del protocolo estándar en un futuro. Los servidores de nombres no son requeridos para añadir RRs SOA en todas las respuestas autoritativas, ni los resolutores para cachear resultados negativos. Se recomiendan ambas cosas. Todos los resolutores y servidores de nombres recursivos son requeridos para que ignoren al menos el RR SOA cuando esté presente en una respuesta.

Algunos experimentos han sido propuestos para utilizar esta funcionalidad. La idea es que si sabe que los datos cacheados vienen de una zona en particular, y si se obtiene una copia autoritativa de la zona del SOA, y si el SERIAL de la zona no ha cambiado desde que el dato fue cacheado, entonces el TTL de los datos cacheados pueden reiniciarse al valor MINIMO de la zona si éste es menor. Este procedimiento está indicado para propósitos de planeamiento, y todavía no está recomendado.

4.3.5. Mantenimiento y transferencias de zona

Parte del trabajo del administrador de zona es mantener todas las zonas en todos los servidores de nombres autoritativos de las zonas. Cuando se producen cambios inevitables, éstos deben distribuirse a todos los servidores de nombres. Esta distribución puede realizarse mediante FTP o algún otro procedimiento puntual, pero el método preferido es la transferencia de zona que forma parte del protocolo DNS.

El modelo general de la transferencia automática de zona o refresco se da en los servidores de nombres maestros o primarios para la zona. Los cambios son coordinados en el primario, típicamente editando el fichero maestro de la zona. Después de la edición, el administrador indica al servidor maestro que cargue la nueva zona. Los otros servidores no maestros o secundarios para la zona, de forma periódica, comprueban los cambios (en un intervalo seleccionable) y obtienen las nuevas copias de la zona cuando los cambios se hayan realizado.

Para detectar cambios, los servidores secundarios comprueban el campo SERIAL del SOA de la zona. De forma adicional, el campo SERIAL del SOA de la zona siempre avanza cuando se produce cualquier cambio en la zona.

El avance es un incremento simple, o puede estar basado en la fecha y la hora del fichero maestro, etc. El propósito es hacer posible la determinación de cual de las dos copias de una zona es más reciente comparando sus números de serie. Los números de serie utilizan una secuencia aritmética, por tanto hay un límite teórico en cuanto a la rapidez de actualización de una zona, básicamente las copias antiguas mueren antes de que el número de serie llegue a la mitad de su rango de 32 bits. En la práctica, la operación de comparación es correcta si se produce entre los límites del número más positivo y más negativo de los 32 bits.

La comprobación períodica de los servidores secundarios se controla con los parámetros del RR SOA de la zona, que establece el intervalo mínimo aceptable de comprobación. Los parámetros se llaman REFRESH, RETRY, y EXPIRE. Incluso cuando se carga una nueva zona secundaria, ésta espera los segundos de refresco (REFRESH) antes de comprobar un nuevo número de serie de la zona primaria. Si no se completa esta comprobación, se comprueba de nuevo cada (RETRY) segundos. La comprobación es una consulta simple al RR SOA de la zona primaria. Si el campo del número de serie de la zona secundaria es igual al número de serie devuelto por la zona primaria, quiere decir que no se han realizado cambios, y el intervalo de refresco espera a reiniciarse. Si la zona secundaria ve que es imposible realizar una comprobación del número de serie en el intervalo de expiración (EXPIRE), asumirá que su copia de la zona es obsoleta y la descartará.

Cuando la comprobación dice que la zona ha cambiado, el servidor secundario solicitará una transferencia de zona mediante una solicitud AXFR para la zona. El AXFR puede provocar un error, como un rechazo, pero normalmente es respondido por una secuencia de mensajes de respuesta.

El primer y último mensaje deben contener los datos del nodo más alto autoritativo de la zona. Los mensajes intermedios llevan el resto de los RRs de la zona, incluyendo tanto RRs autoritativos y no autoritativos. El flujo de mensajes permite al servidor secundario reconstruir una copia de la zona. Debido a que la exactitud es esencial, se debe utilizar TCP o algún otro protocolo fiable para las solicitudes AXFR.

Cada servidor secundario puede realizar las siguientes operaciones contra un maestro, pero también puede realizar esas operaciones contra otros servidores secundarios. Esta estrategia puede mejorar el proceso de transferencia cuando el primario no está disponible debido a una caída o problemas de red, o cuando un servidor secundario tiene mejor acceso de red a un secundario "intermedio" que al primario.

5. RESOLUTORES

5.1. Introducción

Los resolutores son programas que hacen de intérprete entre los programas de usuario y los servidores de nombres de dominios. En un caso simple, un resolutor recibe una petición de un programa de usuario (por ej. correo, TELNET, FTP, etc.) mediante una llamada de una subrutina, llamada del sistema, etc., y devuelve la información deseada de forma compatible con los formatos de datos del host local.

El resolutor está ubicado en la misma máquina donde está el programa que realiza las consultas de los servicios del resolutor, pero es necesario consultar a servidores de nombres de otros hosts. El tiempo que necesita un resolutor en cumplir su tarea varía dependiendo de si los datos los tiene en la caché local o los consigue de varios servidores. Este tiempo oscila entre milisegundos y segundos.

Un objetivo importante de un resolutor es eliminar el retraso de red y la carga del servidor de nombres de la mayoría de las peticiones respondiéndolos desde su caché con resultados previos. Esto indica que las cachés compartidas por múltiples procesos, usuarios, máquinas, etc., son más eficientes que las cachés no compartidas.

5.2. Intérprete Cliente-resolutor

5.2.1. Funciones típicas

El intérprete de cliente hacia el resolutor está influenciado por las convenciones del host local, pero el intérprete resolutor-cliente típico tiene tres funciones:

1. Conversión de nombres de host a direcciones de host.
Esta función es, a menudo definida por similitud en base a la función del antiguo HOSTS.TXT. Dada una cadena de caracteres, el llamador quiere una o más direcciones IP de 32 bits.

El DNS lo convierte en una petición de RRs A. Debido a que DNS no mantiene el orden de los RRs, esta función puede optar entre ordenar las direcciones devueltas o seleccionar las "mejores" direcciones si el servicio devuelve sólo una opción para el cliente. Es recomendable que se devuelvan varias direcciones, pero devolver una dirección puede ser la única forma de emular los servicios del antiguo HOSTS.TXT.

2. Conversión de direcciones a nombres de host
Esta función sigue a menudo la forma de funciones antiguas. Dada una dirección IP de 32 bits, el origen quiere una cadena de caracteres. Los octetos de las direcciones IP se dan la vuelta, se utilizan como componentes de nombre, y se les pone el sufijo "IN-ADDR.ARPA". Para esto se utiliza una consulta de tipo PTR que consigue el RR con el nombre primario del host. Por ejemplo, una petición para el nombre de host que corresponde a la dirección IP 1.2.3.4 busca RRs PTR para el nombre de dominio "4.3.2.1.IN-ADDR.ARPA".

3. Función lookup general

Esta función recupera información arbitraria del DNS, y no forma parte de sistemas anteriores. El llamador proporciona un QNAME, QTYPE, y QCLASS, y quiere todos los RRs que coincidan. Esta función utiliza el formato DNS en vez del formato del host local para todos los datos RR, y devuelve todo el contenido RR (como el TTL) en vez del formulario procesado con las convenciones locales citadas.

Cuando el resolutor realiza la función indicada, normalmente tiene uno de los siguientes resultados para devolver al cliente:

- Uno o más RRs dados con los datos solicitados.

En este caso el resolutor devuelve la respuesta en el formato apropiado.

- Un error de nombre (NE).

Esto sucede cuando el nombre referenciado no existe. Por ejemplo, un usuario puede haberse equivocado en el nombre de host.

- Un error de dato no encontrado.

Esto sucede cuando el nombre referenciado existe, pero no existen los datos del tipo apropiado. Por ejemplo, una función de dirección de host aplicada a un nombre de dirección de correo puede devolver este error debido a que el nombre existe, pero no hay RR de direcciones.

Es importante saber que las funciones de conversión entre nombres de host y direcciones pueden combinar las condiciones de error "error de nombre" y "dato no encontrado" dentro de un sólo tipo de retorno de error, pero la función general no. Una razón para esto es que las aplicaciones pueden preguntar primero por un tipo de información sobre un nombre seguido de una segunda pregunta al mismo nombre para otro tipo de información; si se combinan los dos errores, las consultas inservibles pueden ralentizar la aplicación.

5.2.2 Alias

Cuando se intenta resolver una consulta en particular, el resolutor puede descubrir que el nombre en cuestión es un alias. Esto se produce cuando encuentra un RR CNAME, y siempre que sea posible, el resolutor debe comunicar esto al cliente.

En la mayoría de los casos un resolutor simplemente reinicia la consulta al nuevo nombre cuando encuentra un CNAME. Sin embargo, cuando se realiza la función general, el resolutor puede que no persiga a los alias cuando el RR CNAME coincida con el tipo de consulta. Esto permite consultas dependiendo de si tiene o no un alias presente en su pregunta. Por ejemplo, si el tipo de consulta es CNAME, el usuario está interesado en en RR CNAME en sí, y no en los RRs de los nombres a los que apunta.

Pueden ocurrir varias condiciones especiales con los alias. No es un error, pero se deberían descartar niveles múltiples de alias debido a su ineficiencia. Los bucles de alias que apuntan a nombres que no existen deberían ser capturados y emitir un error al cliente.

5.2.3. Fallos temporales

Menos en un mundo perfecto, todos los resolutores puede que no resuelvan una consulta en particular en alguna ocasión. Esta condición puede estar causada por un resolutor que se caiga de la red por un fallo de conexión o un problema con la puerta de enlace, o menos a menudo por un fallo o indisponibilidad de todos los servidores de un dominio en particular.

Es esencial que este tipo de condiciones no se presenten como un error de nombre o de dato no presente en las aplicaciones. Esta clase de problemas molestan a los humanos, y pueden sembrar confusión cuando los sistemas de correo utilizan DNS.

En algunos casos se puede producir un problema temporal bloqueando las consultas indefinidamente, pero normalmente no es una buena elección, particularmente cuando el cliente es un proceso servidor que puede moverse entre otras tareas. La solución recomendada es tener siempre el fallo temporal como uno de los posibles resultados de la función del resolutor, incluso cuando se piense que puede ser producido por la emulación de un HOSTS.TXT existente.

5.3. El resolutor por dentro

Cada implementación de resolutor utiliza algoritmos muy parecidos, y normalmente consume mucho más en indicar errores lógicos de varios tipos que en el funcionamiento normal. Esta sección resume una estrategia básica recomendada para la operación del resolutor, dejando más detalles en el [RFC-1035].

5.3.1. Resolutores incompletos

Una opción para implementar un resolutor es mover la función de resolución de una máquina local a un servidor de nombres que soporte consultas recursivas. Esto es un método fácil para dar un servicio de dominio en un PC que libera de sus funciones la de resolver, o centralizar la caché para toda la red u organización.

Todo lo que se necesita es una lista de direcciones de servidores de nombres que realizarán las consultas recursivas. Este tipo de resolutor necesita presumíblemente la información en un fichero de configuración, y debido a esto, probablemente carece de sofisticación para localizarlo en la base de datos del dominio. El usuario también necesita comprobar que los servidores listados realizarán el servicio recursivo; un servidor de nombres es libre para rechazar los servicios recursivos para cualquiera o todos los clientes. El usuario podría consultar al administrador de sistema local para encontrar los servidores de nombres que puedan realizar el servicio.

Este tipo de servicio tiene algunos inconvenientes. Debido a que la consulta recursiva puede llevarse a cabo en un tiempo arbitrario, la comprobación de cambios en la zona puede tener dificultades para optimizar los intervalos de retransmisión en cuanto al trato con los paquetes UDP perdidos y servidores muertos; el servidor de nombres fácilmente puede sobrecargarse por comprobaciones de cambio de zona si se interpreta a las retransmisiones como consultas nuevas. Una respuesta puede ser el uso del TCP, pero TCP puede disminuir la capacidad del host de forma similar a un resolutor real.

5.3.2. Recursos

Adicionalmente a sus propios recursos, el resolutor también puede tener acceso compartido a zonas mantenidas por el servidor de nombres local. Esto da al resolutor la ventaja de tener un acceso más rápido, pero el resolutor debe tener cuidado para que la información en caché nunca anule los datos de la zona. En este punto, el término "información local" se refiere al significado de la unión de la caché y las zonas compartidas, entendiendo que los datos autoritativos siempre tienen preferencia a los datos en caché cuando ambos están presentes.

El siguiente algoritmo del resolutor asume que todas las funciones han sido convertidas a una función general lookup, y utiliza las siguientes estructuras de datos para representar el estado en progreso de una consulta en el resolutor:

SNAME — el nombre de dominio que buscamos.

STYPE — el QTYPE de la consulta.

SCLASS — el QCLASS de la consulta.

SLIST — una estructura que describe a los servidores de nombres y la zona donde el resolutor envía la consulta. Esta estructura guarda un registro de las mejores búsquedas y los servidores de nombres implicados; se actualiza cuando llegan nuevos cambios. Esta estructura incluye el equivalente a un nombre de zona, los servidores de nombres de la zona, las direcciones de los servidores de nombres, y un histórico con información acerca de cual sería el mejor servidor de nombres para la siguiente búsqueda. El nombre de zona equivalente es el que coincida con las etiquetas desde el raíz cuyo SNAME es el mismo que el de la zona consultada; esto se utiliza como medida para "conectar" el SNAME con el resolutor.

SBELT una estructura "cinturón de seguridad" con la misma forma que SLIST, que se inicializa desde un fichero de configuración, y especifica servidores que tendrían que ser utilizados cuando el resolutor no tiene ninguna información local para seleccionar un servidor de nombres. -1 indica que no hay ninguna etiqueta que coincida.

CACHE una estructura que guarda los resultados de las consultas realizadas. Debido a que los resolutores son responsables de eliminar viejos RRs cuyo TTL ha expirado, la mayoría de las implementaciones convierten el intervalo especificado en los RRs que llegan a algún tipo de tiempo absoluto cuando el RR se guarda en caché. En vez de contar los TTLs de forma individual, el resolutor ignora o descarta viejos RRs cuando se ejecutan en el transcurso de una búsqueda, o los descarta durante el barrido periódico de limpieza de memoria consumida por RRs viejos.

5.3.3. Algoritmo

El nivel superior del algoritmo tiene cuatro pasos:

1. Ver si la respuesta viene de la información local, y si se ha devuelto al cliente.

2. Encontrar los mejores servidores para preguntar.

3. Enviarles las consultas hasta que uno responda.

4. Analizar la respuesta, entre:

   a. si la respuesta responde a la pregunta o contiene un error de nombre, cachear los datos y devolverlos al cliente.

   b. si la respuesta contiene una mejor delegación a otros servidores, cachear la información de delegación, y volver al paso 2.

   c. si la respuesta tiene un CNAME y no es la respuesta en sí, cachear el CNAME, cambiar el SNAME por el nombre canónico del RR CNAME e ir al paso 1.

   d. si la respuesta contiene un fallo de servidor u otro contenido raro, borrar el servidor de SLIST y volver al paso 3.

El paso 1 busca en la caché por los datos deseados. Si los datos están en la caché, se asume que será suficiente en condiciones normales. Algunos resolutores dan la opción al cliente de ignorar la caché y consultar con un servidor autoritativo. Por defecto, esto no está recomendado. Si el resolutor tiene acceso directo a zonas de servidores de nombres, debería comprobar si los datos deseados están presentes de forma autoritativa, y utilizarlos de forma preferente a los datos en caché.

El paso 2 busca un servidor de nombres para preguntar por los datos requeridos. La estrategia general es buscar los RRs de servidores de nombres que estén localmente disponibles, comenzando con SNAME, después con el nombre de dominio padre de SNAME, el abuelo, y así hasta llegar al raíz. De este modo, si SNAME es Mockapetris.ISI.EDU, este paso buscará por los RRs NS de Mockapetris.ISI.EDU, después ISI.EDU, luego

EDU, y después . (la raíz). Estos RRs NS listan los nombres de los hosts para una zona en o superior de SNAME, copia los nombres en SLIST y guarda sus direcciones como datos locales. Si el resolutor comprueba que las direcciones no están disponibles, entonces tiene varias opciones; la mejor es comenzar con procesos paralelos de resolución para las direcciones mientras se continúa con las direcciones disponibles. Obviamente, el diseño de las opciones son complicadas y supone carga del host local. Las prioridades recomendadas para el diseño del resolutor son:

1. Limitar el total de trabajo (paquetes enviados, procesos paralelos iniciados) para que el origen no pueda entrar en un bucle infinito o evitar una reacción en cadena de peticiones con otras implementaciones INCLUSO SI ALGUIEN TIENE ALGÚN DATO MAL CONFIGURADO.

2. Responder si es posible.

3. Evitar transmisiones innecesarias.

4. Conseguir la respuesta lo antes posible.

Si falla la búsqueda de los RRs NS, el resolutor inicializará SLIST desde el cinturón de seguridad SBELT. La idea básica es que cuando el resolutor no tiene ni idea de a cual servidor preguntar, debería utilizar la información de un fichero de configuración que contenga varios servidores útiles. Sin embargo, hay situaciones especiales, y lo normal es incluir en el fichero de configuración dos servidores raíz y dos servidores del dominio del host. La razón de que sean dos es por redundancia. Los servidores raíz darán acceso eventual a todo el espacio de dominio. Los dos servidores locales permitirán al resolutor continuar con sus resoluciones de nombres locales si la red local se aisla de internet debido a un fallo de puerta de enlace o conectividad.

De forma adicional a los nombres y direcciones de los servidores, la estructura de datos SLIST puede ordenarse para utilizar los mejores servidores primero, y asegurarse de que todas las direcciones de todos los servidores funcionan con round-robin. La ordenación puede ser una simple función de direcciones preferidas de la red local sobre otras, o puede involucrar estadísticas de eventos pasados, como tiempos de respuestas recibidos y cálculos de medias estadísticas.

El paso 3 envía consultas hasta que recibe una respuesta. La estrategia está en realizarlas de forma cíclica a todas las direcciones de todos los servidores con un tiempo determinado entre cada transmisión. En la práctica es importante utilizar todas las direcciones de un host con más de una tarjeta de red, y una política más agresiva de retransmisión provocaría la ralentización de las respuestas cuando se utilicen varios resolutores para el mismo servidor de nombres e incluso, de forma ocasional, para uno solo. El SLIST normalmente contiene los valores de los datos que controlan los tiempos de expiración y guarda un registro de las transmisiones realizadas.

El paso 4 implica el análisis de las respuestas. El resolutor debería ser bastante paranoico en dicho análisis. Se debería comprobar también que la respuesta coincide con la consulta enviada utilizando el campo ID de la respuesta.

La respuesta ideal es la que viene de un servidor autoritativo que da el dato requerido o un error de nombre. El dato se pasa al usuario y se queda en la caché para usos posteriores si su TTL es mayor a 0.

Si la respuesta muestra una delegación, el resolutor debería comprobar si la delegación está "cercana" a la respuesta teniendo en cuenta los

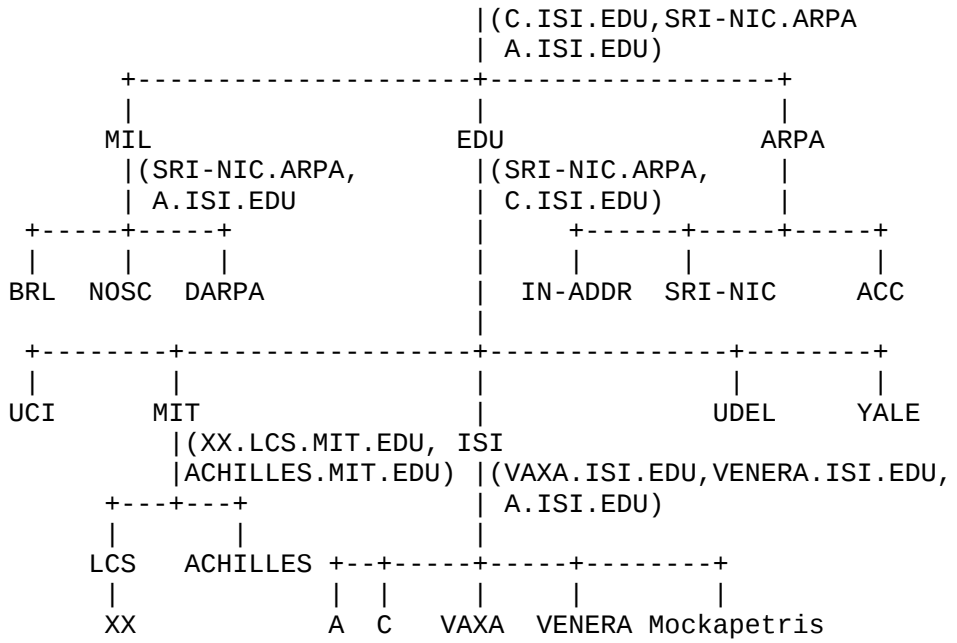
servidores que están en SLIST. Esto se puede llevar a cabo comparando la coincidencia en SLIST con la encontrada en SNAME y los RRs NS en la delegación. Sino, la respuesta sería falsa y debería ser ignorada. Si la delegación es válida los RRs NS de la delegación y cualquier RRs con direcciones de los servidores deberían ser cacheados. Los servidores de nombres se introducen en SLIST, y se reinicia la búsqueda.

Si la respuesta contiene un CNAME, la búsqueda se reinicia en el CNAME a menos que la respuesta tenga los datos del nombre canónico o si el CNAME es la respuesta en sí.

Los detalles y trucos de la implementación están en el [RFC-1035].

6. UN ESCENARIO

En nuestro espacio de dominio de ejemplo, suponemos que queremos separar el control administrativo para las zonas raíz, MIL, EDU, MIT.EDU y ISI.EDU. Tenemos que localizar los servidores de nombres así:



En este ejemplo, el servidor de nombres autoritativo se muestra entre paréntesis en el punto del árbol del dominio donde asume el control.

Además, los servidores de nombres raíz están en C.ISI.EDU, SRI-NIC.ARPA, y A.ISI.EDU. El dominio MIL es hijo de SRI-NIC.ARPA y A.ISI.EDU. El dominio EDU es hijo de SRI-NIC.ARPA. y C.ISI.EDU. Nótese que los servidores pueden tener zonas que sean contiguas o no. En este escenario, C.ISI.EDU tiene zonas contiguas en los dominios raíz y EDU. A.ISI.EDU tiene zonas contiguas en los dominios raíz y MIL, pero también tiene una zona no contigua en ISI.EDU.

6.1. El servidor de nombres C.ISI.EDU

C.ISI.EDU es un servidor de nombres para los dominios raíz, MIL, y EDU de la clase IN, y puede tener zonas para esos dominios. Los datos de zona para el dominio raíz pueden ser:

```
.       IN      SOA     SRI-NIC.ARPA. HOSTMASTER.SRI-NIC.ARPA. (
                        870611          ;serial
                        1800            ;refresh every 30 min
```

```
                                300             ;retry every 5 min
                                604800          ;expire after a week
                                86400)          ;minimum of a day
                        NS      A.ISI.EDU.
                        NS      C.ISI.EDU.
                        NS      SRI-NIC.ARPA.

    MIL.        86400   NS      SRI-NIC.ARPA.
                86400   NS      A.ISI.EDU.

    EDU.        86400   NS      SRI-NIC.ARPA.
                86400   NS      C.ISI.EDU.

    SRI-NIC.ARPA.       A       26.0.0.73
                        A       10.0.0.51
                        MX      0 SRI-NIC.ARPA.
                        HINFO   DEC-2060 TOPS20

    ACC.ARPA.           A       26.6.0.65
                        HINFO   PDP-11/70 UNIX
                        MX      10 ACC.ARPA.

    USC-ISIC.ARPA.      CNAME   C.ISI.EDU.

    73.0.0.26.IN-ADDR.ARPA.  PTR    SRI-NIC.ARPA.
    65.0.6.26.IN-ADDR.ARPA.  PTR    ACC.ARPA.
    51.0.0.10.IN-ADDR.ARPA.  PTR    SRI-NIC.ARPA.
    52.0.0.10.IN-ADDR.ARPA.  PTR    C.ISI.EDU.
    103.0.3.26.IN-ADDR.ARPA. PTR    A.ISI.EDU.

    A.ISI.EDU. 86400 A      26.3.0.103
    C.ISI.EDU. 86400 A      10.0.0.52
```

Estos datos están representados como si estuvieran en un fichero maestro. La mayoría de los RRs son entradas de una línea; la única excepción aquí es el RR SOA, que utiliza "(" para comenzar un RR multilínea y ")" para indicar su fin. Debido a que la clase de todos los RRs de una zona deben ser la misma, con que lo especifique el primer RR es suficiente. Cuando un servidor de nombres carga una zona, obliga a los RRs autoritativos a tener al menos el campo TTL del SOA, aquí tenemos 86400 segundos, o un día. Los RRs NS marcan la delegación de los dominios MIL y EDU, junto con los RRs glue de las direcciones de host de los servidores, y no forman parte de los datos autoritativos en la zona y por tanto tienen TTLs explícitos.

Cuatro RRs van adjuntos al nodo raíz: el SOA que describe la zona raíz y los tres RRs NS que listan los servidores de nombres para la zona raíz. Los datos del RR SOA describe el mantenimiento de la zona. Los datos de la zona están mantenidos en el host SRI-NIC.ARPA, y el responsable adjunto de la zona es HOSTMASTER@SRI-NIC.ARPA. Una dato clave del SOA es el TTL mínimo de 86400 segundos, que quiere decir que todos los datos autoritativos de la zona tienen al menos ese TTL, a no ser que se especifique explícitamente un valor más alto.

Los RRs NS de los dominios MIL y EDU marcan los límites entre la zona raíz y las zonas MIL y EDU. Nótese que esto es un ejemplo, las zonas bajas tienen que estar soportadas por servidores de nombres que también están soportados por la zona raíz.

El fichero maestro para la zona EDU debe ser relativo al origen EDU. Los datos de zona del dominio EDU deberían ser:

```
    EDU.  IN SOA SRI-NIC.ARPA. HOSTMASTER.SRI-NIC.ARPA. (
                            870729 ;serial
```

```
                              1800 ;refresh every 30 minutes
                              300 ;retry every 5 minutes
                              604800 ;expire after a week
                              86400 ;minimum of a day
                              )
                        NS SRI-NIC.ARPA.
                        NS C.ISI.EDU.

      UCI 172800 NS ICS.UCI
                        172800 NS ROME.UCI
      ICS.UCI 172800 A 192.5.19.1
      ROME.UCI 172800 A 192.5.19.31
      ISI 172800 NS VAXA.ISI
                        172800 NS A.ISI
                        172800 NS VENERA.ISI.EDU.
      VAXA.ISI 172800 A 10.2.0.27
                        172800 A 128.9.0.33
      VENERA.ISI.EDU. 172800 A 10.1.0.52
                        172800 A 128.9.0.32
      A.ISI 172800 A 26.3.0.103

      UDEL.EDU.  172800 NS LOUIE.UDEL.EDU.
                        172800 NS UMN-REI-UC.ARPA.
      LOUIE.UDEL.EDU. 172800 A 10.0.0.96
                        172800 A 192.5.39.3

      YALE.EDU.  172800 NS YALE.ARPA.
      YALE.EDU.  172800 NS YALE-BULLDOG.ARPA.

      MIT.EDU.  43200 NS XX.LCS.MIT.EDU.
                          43200 NS ACHILLES.MIT.EDU.
      XX.LCS.MIT.EDU.  43200 A 10.0.0.44
      ACHILLES.MIT.EDU. 43200 A 18.72.0.8
```

Nótese aquí el uso de nombres relativos. El nombre propietario para ISI.EDU. es relativo, como los dos contenidos del RR del servidor de nombres. Los nombres de dominio relativos y absolutos pueden ser intermezclados en un maestro.

6.2. Ejemplo de consultas estándar

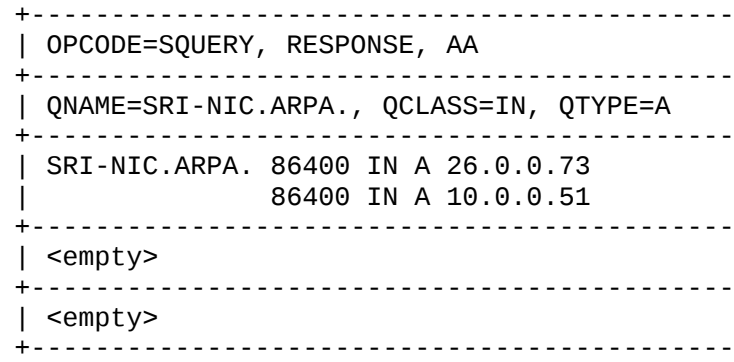
Las siguientes consultas y respuestas muestran el funcionamiento del servidor de nombres. A menos que se especifique lo contrario, las consultas no tienen el requerimiento de recursión (RD) en la cabecera. Nótese que las respuestas a consultas no recursivas se realizan dependiendo del servidor preguntado, y no dependen del origen de la consulta.

6.2.1. QNAME=SRI-NIC.ARPA, QTYPE=A

La consulta puede ser así:

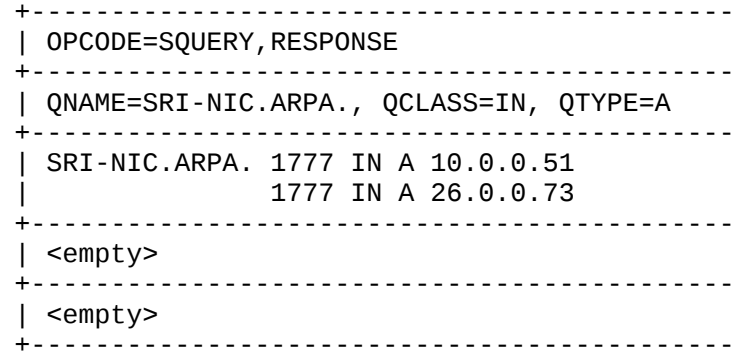
Header	OPCODE=SQUERY
Question	QNAME=SRI-NIC.ARPA., QCLASS=IN, QTYPE=A
Answer	<empty>
Authority	<empty>
Additional	<empty>

La respuesta desde C.ISI.EDU puede ser:



La cabecera de la respuesta es parecida a la cabecera de la consulta, excepto el RESPONSE que es un bit activado que indica que este mensaje es una respuesta y no una consulta. La respuesta también tiene el bit de respuesta autoritativa (AA) que indica que los RRs de la direcciones en la sección de respuesta (Answer) vienen de datos autoritativos. La sección de consulta (Question) es igual a la misma sección de la consulta.

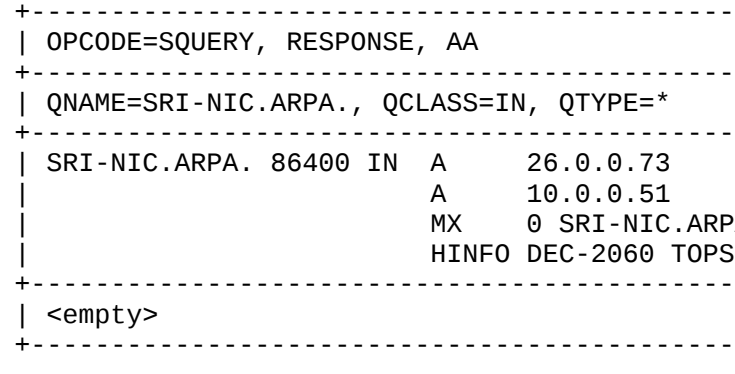
Si la misma consulta se enviara a algún otro servidor que no es autoritativo para SRI-NIC.ARPA, la respuesta puede ser:



Esta respuesta es distinta de la anterior en dos formas: la cabecera no tiene AA, y los TTLs son diferentes. La diferencia es que los datos no vienen de una zona, sino de una caché. La diferencia entre el TTL autoritativo y el TTL que tenemos aquí es producida por la caducidad de los datos en la caché. La diferencia del orden de los RRs no es significativa.

6.2.2. QNAME=SRI-NIC.ARPA, QTYPE=*

Una consulta similar a la anterior, pero utilizando un QTYPE de *, puede recibir la siguiente respuesta desde C.ISI.EDU:



```
Additional | <empty>                                              |
           +-----------------------------------------------------+
```

Si una consulta similar fuera dirigida a dos servidores de nombres que no son autoritativos para SRI-NIC.ARPA, las respuestas pueden ser:

```
           +-----------------------------------------------------+
Header     | OPCODE=SQUERY, RESPONSE                             |
           +-----------------------------------------------------+
Question   | QNAME=SRI-NIC.ARPA., QCLASS=IN, QTYPE=*             |
           +-----------------------------------------------------+
Answer     | SRI-NIC.ARPA. 12345 IN     A       26.0.0.73        |
           |                            A       10.0.0.51        |
           +-----------------------------------------------------+
Authority  | <empty>                                             |
           +-----------------------------------------------------+
Additional | <empty>                                             |
           +-----------------------------------------------------+
```

```
           +-----------------------------------------------------+
Header     | OPCODE=SQUERY, RESPONSE                             |
           +-----------------------------------------------------+
Question   | QNAME=SRI-NIC.ARPA., QCLASS=IN, QTYPE=*             |
           +-----------------------------------------------------+
Answer     | SRI-NIC.ARPA. 1290 IN HINFO  DEC-2060 TOPS20        |
           +-----------------------------------------------------+
Authority  | <empty>                                             |
           +-----------------------------------------------------+
Additional | <empty>                                             |
           +-----------------------------------------------------+
```

Ninguna de las dos respuestas tienen AA, por eso ninguna respuesta viene de datos autoritativos. Los contenidos y TTLs son diferentes porque vienen de dos cachés correspondientes a dos servidores en tiempos distintos, y por eso el primer servidor cachea la respuesta a una consulta QTYPE=A y el segundo la cachea a una consulta HINFO.

6.2.3. QNAME=SRI-NIC.ARPA, QTYPE=MX

Este tipo de consulta podría ser de un programa de correo intentando averiguar información acerca del destino HOSTMASTER@SRI-NIC.ARPA. La respuesta desde C.ISI.EDU sería:

```
           +-----------------------------------------------------+
Header     | OPCODE=SQUERY, RESPONSE, AA                         |
           +-----------------------------------------------------+
Question   | QNAME=SRI-NIC.ARPA., QCLASS=IN, QTYPE=MX            |
           +-----------------------------------------------------+
Answer     | SRI-NIC.ARPA. 86400 IN     MX      0 SRI-NIC.ARPA.|
           +-----------------------------------------------------+
Authority  | <empty>                                             |
           +-----------------------------------------------------+
Additional | SRI-NIC.ARPA. 86400 IN     A       26.0.0.73        |
           |                            A       10.0.0.51        |
           +-----------------------------------------------------+
```

Esta respuesta tiene el RR MX en la sección de respuesta. La sección adicional tiene los RRs de las direcciones debido a que el servidor de nombres C.ISI.EDU supone que el solicitante necesitará las direcciones para para ubicar la información de MX.

6.2.4. QNAME=SRI-NIC.ARPA, QTYPE=NS

C.ISI.EDU puede responder a esta consulta con:

```
              +-------------------------------------------------+
    Header    | OPCODE=SQUERY, RESPONSE, AA                     |
              +-------------------------------------------------+
    Question  | QNAME=SRI-NIC.ARPA., QCLASS=IN, QTYPE=NS        |
              +-------------------------------------------------+
    Answer    | <empty>                                         |
              +-------------------------------------------------+
    Authority | <empty>                                         |
              +-------------------------------------------------+
    Additional| <empty>                                         |
              +-------------------------------------------------+
```

La única diferencia entre la respuesta y la pregunta es AA y el bit RESPONSE de la cabecera. La interpretación de esta respuesta es que el servidor es autoritativo para el nombre, y el nombre existe, pero no hay RRs del tipo NS.

6.2.5. QNAME=SIR-NIC.ARPA, QTYPE=A

Podemos ver este tipo de consulta si un usuario se equivoca con un nombre de host.

C.ISI.EDU puede responder con:

```
              +-------------------------------------------------+
    Header    | OPCODE=SQUERY, RESPONSE, AA, RCODE=NE           |
              +-------------------------------------------------+
    Question  | QNAME=SIR-NIC.ARPA., QCLASS=IN, QTYPE=A         |
              +-------------------------------------------------+
    Answer    | <empty>                                         |
              +-------------------------------------------------+
    Authority | . SOA SRI-NIC.ARPA. HOSTMASTER.SRI-NIC.ARPA.    |
              |       870611 1800 300 604800 86400              |
              +-------------------------------------------------+
    Additional| <empty>                                         |
              +-------------------------------------------------+
```

Esta respuesta dice que el nombre no existe. Esta condición está indicada en la cabecera, en concreto en la sección de código de respuesta (RCODE).

El RR SOA de la sección autoritativa es la información opcional negativa de caché que dice al resolutor que asuma que el nombre no existe para el SOA durante 86400 segundos como mínimo.

6.2.6. QNAME=BRL.MIL, QTYPE=A

Si esta consulta se envia a C.ISI.EDU, la respuesta puede ser:

```
              +-------------------------------------------------+
    Header    | OPCODE=SQUERY, RESPONSE                         |
              +-------------------------------------------------+
    Question  | QNAME=BRL.MIL, QCLASS=IN, QTYPE=A               |
              +-------------------------------------------------+
    Answer    | <empty>                                         |
              +-------------------------------------------------+
    Authority | MIL.            86400 IN NS     SRI-NIC.ARPA.   |
              |                 86400    NS     A.ISI.EDU.      |
              +-------------------------------------------------+
    Additional| A.ISI.EDU.              A       26.3.0.103      |
              | SRI-NIC.ARPA.           A       26.0.0.73       |
```

```
                |                           A       10.0.0.51      |
                +--------------------------------------------------+
```

Esta respuesta tiene la sección de respuesta vacía, pero no es autoritativa, es una referencia. El servidor de nombres de C.ISI.EDU dice que no es autoritativo para el dominio MIL, y refiere al origen a los servidores de A.ISI.EDU y SRI-NIC.ARPA, puesto que sabe que éstos son autoritativos para el dominio MIL.

6.2.7. QNAME=USC-ISIC.ARPA, QTYPE=A

La respuesta a esta consulta desde A.ISI.EDU puede ser:

```
                +--------------------------------------------------+
    Header      | OPCODE=SQUERY, RESPONSE, AA                      |
                +--------------------------------------------------+
    Question    | QNAME=USC-ISIC.ARPA., QCLASS=IN, QTYPE=A         |
                +--------------------------------------------------+
    Answer      | USC-ISIC.ARPA. 86400 IN CNAME     C.ISI.EDU.     |
                | C.ISI.EDU.     86400 IN A         10.0.0.52      |
                +--------------------------------------------------+
    Authority   | <empty>                                          |
                +--------------------------------------------------+
    Additional  | <empty>                                          |
                +--------------------------------------------------+
```

Nótese que el bit AA de la cabecera garantiza que el dato que coincide con QNAME es autoritativo, pero no dice nada sobre si el dato para C.ISI.EDU es autoritativo. Esta respuesta puede ser completa porque A.ISI.EDU parece ser autoritativo para ambos dominios ARPA donde se encuentra USC-ISIC.ARPA y en el dominio ISI.EDU se encuentran los datos de C.ISI.EDU.

Si la misma consulta fuera enviada a C.ISI.EDU, la respuesta puede ser la misma que la anterior en el caso de que su propia dirección estuviera cacheada en su caché, pero también puede ser:

```
                +--------------------------------------------------+
    Header      | OPCODE=SQUERY, RESPONSE, AA                      |
                +--------------------------------------------------+
    Question    | QNAME=USC-ISIC.ARPA., QCLASS=IN, QTYPE=A         |
                +--------------------------------------------------+
    Answer      | USC-ISIC.ARPA.   86400 IN CNAME   C.ISI.EDU.     |
                +--------------------------------------------------+
    Authority   | ISI.EDU.        172800 IN NS      VAXA.ISI.EDU.  |
                |                        NS      A.ISI.EDU.       |
                |                        NS      VENERA.ISI.EDU.  |
                +--------------------------------------------------+
    Additional  | VAXA.ISI.EDU.   172800    A       10.2.0.27       |
                |                 172800    A       128.9.0.33      |
                | VENERA.ISI.EDU. 172800    A       10.1.0.52       |
                |                 172800    A       128.9.0.32      |
                | A.ISI.EDU.      172800    A       26.3.0.103      |
                +--------------------------------------------------+
```

Esta respuesta es autoritativa para el alias USC-ISIC.ARPA, y además hace referencia a los servidores de nombres de ISI.EDU. Este tipo de respuesta es muy distinto dado que la consulta es para el nombre de host del servidor de nombres preguntado, pero puede ser común para otros alias.

6.2.8. QNAME=USC-ISIC.ARPA, QTYPE=CNAME

Si esta consulta es enviada tanto a A.ISI.EDU o C.ISI.EDU, la respuesta puede ser:

```
              +---------------------------------------------------+
    Header    | OPCODE=SQUERY, RESPONSE, AA                       |
              +---------------------------------------------------+
    Question  | QNAME=USC-ISIC.ARPA., QCLASS=IN, QTYPE=A          |
              +---------------------------------------------------+
    Answer    | USC-ISIC.ARPA. 86400 IN CNAME     C.ISI.EDU.      |
              +---------------------------------------------------+
    Authority | <empty>                                           |
              +---------------------------------------------------+
    Additional| <empty>                                           |
              +---------------------------------------------------+
```

Puesto que QTYPE=CNAME, el RR CNAME en sí responde la consulta, y el servidor de nombres no intenta la búsqueda para C.ISI.EDU. (Puede tener como excepción alguna posibilidad en la sección adicional).

6.3. Resolución de ejemplo

Los siguientes ejemplos muestran las operaciones que un resolutor debe realizar para su cliente. Asumimos que el resolutor comienza sin caché, como sería el caso después de un inicio del sistema.

También asumimos que el sistema no es un host que tenga datos de zona y que el host está localizado en algún lugar de la red 26, y que la estructura de su cinturón de seguridad (SBELT) tiene la siguiente información:

```
    Match count = -1
    SRI-NIC.ARPA.   26.0.0.73       10.0.0.51
    A.ISI.EDU.      26.3.0.103
```

Esta información especifica a los servidores para resolver nombres, sus direcciones, y un contador de coincidencia a -1, que quiere decir que los servidores no están muy cerca del destino. Nótese que -1 no supone una medida con mucha precisión, sino un valor que utilizará el algoritmo en fases posteriores.

Los siguientes ejemplos muestran el funcionamiento de la caché, por ello cada ejemplo asume que la consulta anterior se completó.

6.3.1. Resolver MX de ISI.EDU.

Supongamos que la primera consulta del resolutor viene de un programa de correo electrónico que tiene un email para PVM@ISI.EDU. El programa de correo electrónico pregunta por los RRs del tipo MX del nombre de dominio ISI.EDU.

El resolutor puede buscar los RRs MX de ISI.EDU en su caché, pero una caché vacía no es útil. El resolutor puede reconocer que necesita consultar a servidores externos e intentar determinar los mejores servidores para la consulta. Este rastreo puede buscar los RRs NS de los dominios ISI.EDU, EDU, y el raíz. Estos rastreos también pueden fallar si vienen de alguna caché. Como último recurso, el resolutor puede utilizar la información desde SBELT, copiándolo en su estructura SLIST.

En este punto, el resolutor puede que necesite elegir una de las tres direcciones para probar. Dado que el resolutor está en la red 26, debería elegir primero entre 26.0.0.73 o 26.3.0.103. Entonces puede enviar una consulta con el siguiente formulario:

```
                +---------------------------------------------------+
```

```
Header     | OPCODE=SQUERY                                        |
           +------------------------------------------------------+
Question   | QNAME=ISI.EDU., QCLASS=IN, QTYPE=MX                  |
           +------------------------------------------------------+
Answer     | <empty>                                              |
           +------------------------------------------------------+
Authority  | <empty>                                              |
           +------------------------------------------------------+
Additional | <empty>                                              |
           +------------------------------------------------------+
```

El resolutor puede entonces esperar una respuesta o un tiempo agotado. Si se agota el tiempo, puede intentar de nuevo con servidores distintos, esto es, diferentes direcciones para los mismos servidores, y por último puede utilizar direcciones ya utilizadas.
Puede recibir una respuesta desde SRI-NIC.ARPA:

```
           +------------------------------------------------------+
Header     | OPCODE=SQUERY, RESPONSE                              |
           +------------------------------------------------------+
Question   | QNAME=ISI.EDU., QCLASS=IN, QTYPE=MX                  |
           +------------------------------------------------------+
Answer     | <empty>                                              |
           +------------------------------------------------------+
Authority  | ISI.EDU.        172800 IN NS       VAXA.ISI.EDU.     |
           |                           NS       A.ISI.EDU.        |
           |                           NS       VENERA.ISI.EDU.   |
           +------------------------------------------------------+
Additional | VAXA.ISI.EDU.   172800    A        10.2.0.27         |
           |                 172800    A        128.9.0.33        |
           | VENERA.ISI.EDU. 172800    A        10.1.0.52         |
           |                 172800    A        128.9.0.32        |
           | A.ISI.EDU.      172800    A        26.3.0.103        |
           +------------------------------------------------------+
```

El resolutor recibe una respuesta con una delegación cercana a ISI.EDU donde coinciden las tres etiquetas con su SLIST. El resolutor puede entonces cachear la información de la respuesta y utilizarla para crear un nuevo SLIST:

```
Match count = 3
A.ISI.EDU.      26.3.0.103
VAXA.ISI.EDU.   10.2.0.27       128.9.0.33
VENERA.ISI.EDU. 10.1.0.52       128.9.0.32
```

A.ISI.EDU también aparece en esta lista como antes, pero es pura coincidencia. El resolutor puede comenzar la transmisión de nuevo y esperar respuestas. De forma eventual puede recibir una respuesta:

```
           +------------------------------------------------------+
Header     | OPCODE=SQUERY, RESPONSE, AA                          |
           +------------------------------------------------------+
Question   | QNAME=ISI.EDU., QCLASS=IN, QTYPE=MX                  |
           +------------------------------------------------------+
Answer     | ISI.EDU.                MX 10 VENERA.ISI.EDU.        |
           |                         MX 20 VAXA.ISI.EDU.          |
           +------------------------------------------------------+
Authority  | <empty>                                              |
           +------------------------------------------------------+
Additional | VAXA.ISI.EDU.   172800  A  10.2.0.27                 |
           |                 172800  A  128.9.0.33                |
           | VENERA.ISI.EDU. 172800  A  10.1.0.52                 |
           |                 172800  A  128.9.0.32                |
```

```
                +-------------------------------------------------+
```

El resolutor puede agregar esta información a su caché, y devolver los RRs MX a su cliente.

6.3.2. Conseguir el nombre de host de la dirección 26.6.0.65

El resolutor puede convertir esto en una consulta de RRs PTR de 65.0.6.26.IN-ADDR.ARPA. Esta información no está en caché, por tanto el resolutor puede buscar servidores para preguntar. Puede que no coincida ningún servidor, por tanto puede utilizar de nuevo el SBELT. (Nótese que los servidores del dominio ISI.EDU están en la caché, pero ISI.EDU no es un antecesor de 65.0.6.26.IN-ADDR.ARPA, y por eso se utiliza el SBELT).

Debido a que esta consulta está en los datos autoritativos de ambos servidores en SBELT, de forma eventual uno de ellos puede devolver:

```
                +-------------------------------------------------+
    Header      | OPCODE=SQUERY, RESPONSE, AA                     |
                +-------------------------------------------------+
    Question    | QNAME=65.0.6.26.IN-ADDR.ARPA.,QCLASS=IN,QTYPE=PTR |
                +-------------------------------------------------+
    Answer      | 65.0.6.26.IN-ADDR.ARPA.    PTR     ACC.ARPA.    |
                +-------------------------------------------------+
    Authority   | <empty>                                         |
                +-------------------------------------------------+
    Additional  | <empty>                                         |
                +-------------------------------------------------+
```

6.3.3. Conseguir la dirección de host de poneria.ISI.EDU

Esta consulta puede ser un tipo A para poneria.ISI.EDU. El resolutor puede que no encuentre ningún dato cacheado para este nombre, pero puede encontrar RRs NS en su caché de ISI.EDU cuando busque en servidores externos. Utilizando estos datos, podemos construir un SLIST así:

```
    Match count = 3

    A.ISI.EDU.      26.3.0.103
    VAXA.ISI.EDU.   10.2.0.27       128.9.0.33
    VENERA.ISI.EDU. 10.1.0.52
```

A.ISI.EDU es el primero asumiendo que el resolutor ordena la lista de forma preferente, y A.ISI.EDU está en su misma red.

Uno de esos servidores puede responder a la pregunta.