UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

UPGRADING NAME SERVER CREDIBILITY: UN BREVE ANÁLISIS A LOS DOMINIOS DE LA ZONA .CL

TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIAS, MENCIÓN COMPUTACIÓN

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

NELSON DAVID MARAMBIO QUIROZ

PROFESOR GUÍA:
JAVIER BUSTOS JIMÉNEZ

MIEMBROS DE LA COMISIÓN:
JOSÉ PIQUER GARDNER
TOMÁS BARROS ARANCIBIA
JAVIER CAÑAS ROBLES

SANTIAGO DE CHILE
2023

Resumen

MEJORANDO LA CREDIBILIDAD DE LOS SERVIDORES DE NOMBRE: UN BREVE ANÁLISIS A LOS DOMINIOS DE LA ZONA .CL

El sistema de nombres de dominio permite la traducción de nombres a direcciones IP's. Un componente primordial en él es el *resolver*, el cual se encarga de entregar la información deseada al usuario. Hoy en día, existen dos tipos de *resolvers*: 'child-centric' y 'parent-centric', haciendo que el protocolo de nombres de dominio no sea estándar. Recientemente, Huque et al. propone nuevas recomendaciones para estandarizar este comportamiento. Sin embargo, estas recomendaciones podrían traer consecuencias en el funcionamiento del protocolo, tales como aumento en el tiempo de respuesta, o que un dominio se vuelva inalcanzable. Si un dominio se vuelve inalcanzable, entonces los usuarios no podrán acceder a sus servicios. Si hablamos de servicios fundamentales, tales como salud o educación, lo anterior podría generar un gran impacto en las personas. Es por esto que esta investigación se enfoca en encontrar las consecuencias de las recomendaciones propuestas, y también en encontrar aquellos dominios en Chile que se verán afectados en caso de que estas recomendaciones se implementen. Como resultado de la investigación, se encontró que el tiempo de respuesta de las consultas no se ve afectado de forma significativa, y además se encontró un caso donde un servidor mal configurado puede hacer inalcanzable a una zona. Además, al analizar los dominios en Chile, se encontró que los dominios afectados los cuales quedarían inalcanzables sus zonas son un 0.57 % del total. Finalmente se recomienda arreglar los servidores mal configurados, para así evitar que existan dominios inalcanzables si la nueva propuesta llega a ser implementada.

A todas las personas que me apoyaron.
En especial a mi familia, que estuvo en todo momento.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

Hoy en día el internet es utilizado ampliamente en el mundo. Uno de sus componentes importantes es el sistema de nombres de dominio, el cual fue creado para mejorar la experiencia de los usuarios. Este sistema permite traducir nombres de dominio a direcciones *IP*, y es considerado un servicio crítico [7, 9, 15, 16]. Es por esta razón que cualquier falla en él, podría dejar miles de servicios inalcanzables para sus usuarios [3].

Este sistema, más conocido como DNS [1, 2], está compuesto principalmente de tres actores: el espacio de nombres de dominio, los *name servers*, y los *resolvers*. El espacio de nombres de dominio es una estructura en forma de árbol que mantiene la información asociada con los nombres, como por ejemplo las direcciones *IP*, y los servidores a cargo de secciones más bajas en el árbol. Dentro de este árbol, esta información es almacenada en una estructura conocida como *resource record*, la cual posee varios tipos dependiendo de la información guardada. Por otro lado, los *name servers* mantienen la información de este árbol, y se dicen *autoritativos* cuando poseen la información completa de una parte de este. Finalmente, los *resolvers* son programas que extraen información desde los *name servers* para responder las consultas de los usuarios.

Los *resolvers* hacen todas las consultas que sean necesarias, para así llegar a la información que requiere el usuario. Esto incluye consultas a diferentes *name servers* para encontrar información que se ubica más abajo dentro del árbol de dominio. Este proceso anterior es conocido como delegación, y consiste en la redirección de la consulta a otra zona (o *name server*) más abajo en el árbol.

Para que un resolver sepa que debe realizar la consulta a otra zona, los name server almacenan resource records de tipo NS. Estos no son información propiamente tal de la zona que los almacena, debido a que solo permiten realizar la delegación de una consulta. Por esta razón, estos resource records también se encuentran en la zona que les pertenece. Lo anterior puede apreciarse en la figura 1.1, en donde la zona '.cl' posee resource records pertenecientes a la zona 'uchile.cl', y estos también son encontrados en la zona anteriormente nombrada.

Lo anterior generó un pequeño problema en la implementación de los *resolvers*. En el algoritmo presentado en el *RFC 1034* solo se especifica que la información de delegación debe guardarse en el caché, pero no especifica si se refiere a la información que tiene el padre, o la

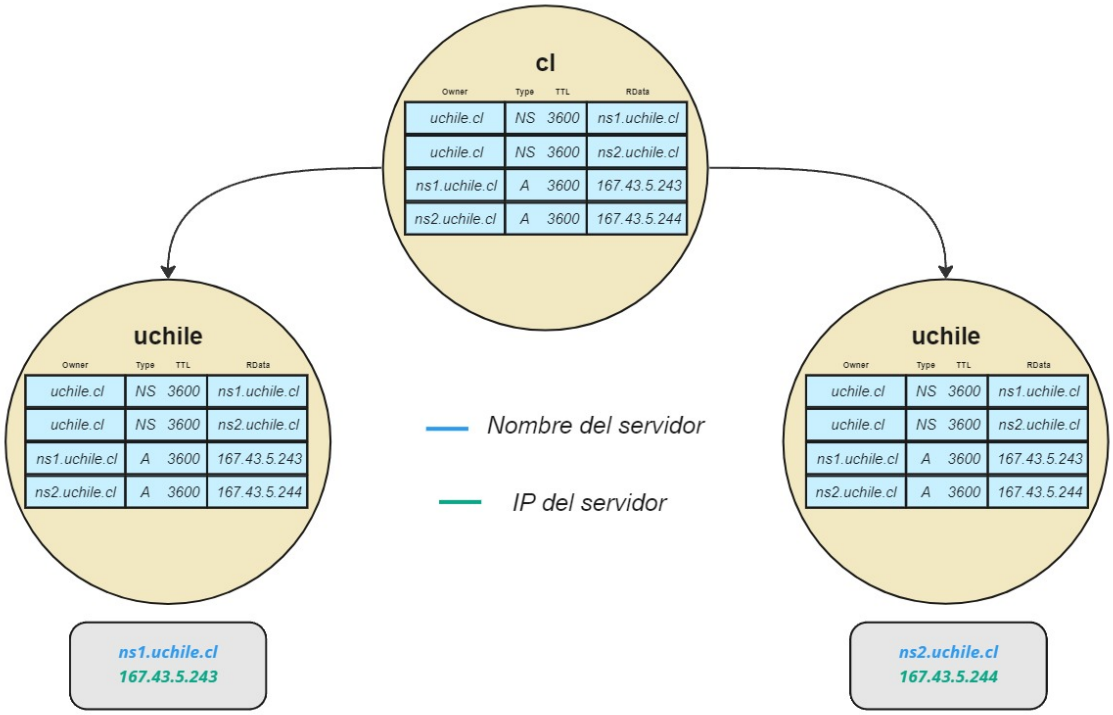

Figura 1.1: Ejemplo de registros NS en una zona padre y su zona hija.

información que posee el hijo [1]. Esto dio lugar a una pequeña inconsistencia en la manera de funcionar en los *resolvers*: algunos guardaban en el caché los registros *NS* del padre, y otros los del hijo. Estos *resolvers* son conocidos como *parent centric* y *child centric* respectivamente [13]. Esta inconsistencia genera que el protocolo de nombres de dominio no sea estándar, haciendo que el comportamiento de los *resolvers* no sea predecible. Por otro lado, los datos de delegación guardados en el caché en un *resolver* parent-centric, pierden credibilidad, debido a que estos datos no son extraídos de la zona *autoritativa* correspondientes a ellos, sino que provienen de la zona padre.

Es por esto que el año 2020 se publicó un *draft* que busca estandarizar el funcionamiento del algoritmo del *resolver* en estos casos, y así hacer que el comportamiento de estos algoritmos sea determinista [10]. Además, busca aumentar la credibilidad de los datos de delegación guardados en el caché. El nuevo algoritmo realiza un cambio en el caso de una delegación. El nuevo *draft* nos recomienda realizar una consulta de validación al momento de delegar a una nueva zona. Esto quiere decir, que de forma paralela el resolver intentará conseguir los *resource records* de tipo NS desde la zona que delegó. Si logra conseguir estos registros, entonces reemplaza los registros provenientes del padre. Por el contrario, si no logra conseguirlos, se queda con la información del padre.

Finalmente, con lo anterior se logra la estandarización del algoritmo del *resolver*, y también, aumentar la credibilidad de los *resource records* de tipo *NS* alojados en el caché, gracias a que estos vendrán de una respuesta autoritativa.

1.1. Problema

El problema que genera el cambio del algoritmo del *resolver* previamente presentado, es que no sabemos qué consecuencias podría producir en el comportamiento del protocolo. Por ejemplo, si aumenta el tiempo de respuesta de las consultas, entonces los usuarios tendrán que esperar más tiempo para acceder a servicios que usen DNS. Por otro lado, si el nuevo algoritmo propuesto presenta algún caso de fallo, entonces varios dominios podrían dejar de funcionar, dejando a varios usuarios sin los servicios que ellos ofrecen.

En este trabajo se busca investigar cuáles serán las consecuencias que traerá el nuevo estándar al protocolo de nombres de dominio.

1.2. Preguntas de Investigación

- El nuevo estándar busca solucionar algunos problemas con respecto al *resolver* DNS, pero ¿Cuál es el *tradeoff* que se genera al solucionar estos problemas?
- ¿Se verán afectados los servidores que actualmente usan el protocolo antiguo? ¿De qué forma?
- ¿Cuántos dominios en Chile se verán afectados? ¿Cuáles y por qué?

1.3. Hipótesis

La nueva propuesta logra estandarizar el algoritmo del *resolver* en caso de delegación, sin producir una falla en más del 10 % de los dominios nacionales.

1.4. Objetivo General

El objetivo general de esta investigación es encontrar las consecuencias que tendrá el nuevo estándar en el funcionamiento operacional del protocolo de nombres de dominio.

1.5. Objetivos específicos

- Analizar y comparar el tiempo de respuesta del nuevo algoritmo con el algoritmo actual.
- Analizar y comprobar el comportamiento del algoritmo en los diferentes casos en que los *resource records* de delegación de un dominio no estén bien configurados.
- Encontrar los diferentes dominios en Chile que se verán afectados con los nuevos cambios.

1.6. Metodología

1.6.1. Desarrollo

Como parte del trabajo ingenieril, será necesario desarrollar un *resolver* con el algoritmo usado actualmente (basado en el RFC 1034) y otro con el algoritmo nuevo propuesto en el *draft*. Lo anterior será utilizado para realizar los diferentes experimentos de la investigación. Cabe destacar, que las herramientas mencionadas serán desarrolladas por el autor, debido a que tomar implementaciones de terceros y modificarlas puede ser un trabajo muy complejo, puesto que habría que entender el código completo de sus implementaciones. Además, estas implementaciones de terceros incluyen más funcionalidades, que en el caso de esta investigación no tienen impacto alguno (solo aumentar la complejidad del entendimiento).

Finalmente, se propone el desarrollo de una interfaz web, para que las diferentes personas encargadas de las zonas puedan revisar si sus dominios se verán afectados o no.

1.6.2. Investigación

Por otro lado, como parte del trabajo de investigación, se analizarán de forma teórica los diferentes casos en donde los *resource records* de delegación no sigan el estándar especificado, o más bien, no estén configurados correctamente. Con lo anterior se busca encontrar aquellas configuraciones que sean potenciales casos a fallar, también encontrar la razón de porqué lo hacen. Para esto será necesario analizar profundamente el comportamiento del resolver en el RFC 1034, 1035, y en el nuevo draft antes mencionado. Además, se buscará investigar si el tiempo de respuesta de las consultas es afectado por el nuevo algoritmo. Finalmente, se buscará encontrar cuales son aquellos dominios en nuestro país que fallarán en caso de que se implemente el nuevo algoritmo.

1.6.3. Experimentación

Finalmente, para completar el trabajo de investigación, se realizarán los diferentes experimentos relacionados al análisis de tiempo de respuesta, análisis de casos mal configurados, y finalmente, el análisis a los dominios pertenecientes a la zona '.cl'. Estos experimentos serán abordados de forma más profunda más adelante en la lectura.

1.7. Resultados Esperados

Se espera encontrar los dominios en Chile que podrían fallar debido al nuevo estándar. Con esto se podrá avisar a los encargados de estos dominios, para que así puedan solucionar el problema antes de que entre en vigencia el nuevo algoritmo del resolver.

Por otra parte, se espera obtener un análisis del tiempo de respuesta de ambos algoritmos, para así saber si el nuevo algoritmo afecta el rendimiento de los resolvers o no.

Finalmente, se contribuirá con una aplicación web que permitirá encontrar si un dominio se verá afectado o no con la problemática investigada.

Capítulo 2

Domain Name System

En este capitulo, se verá de forma detallada el funcionamiento del sistema de nombres de dominio, junto con los nuevos cambios propuestos a investigar, para así poder entender detalladamente el análisis que busca realizar la presente investigación.

El Sistema de Nombre de Dominio o Domain Name System (DNS) es un sistema que permite la traducción de nombres a direcciones IP [1, 2]. Esto con el fin de facilitar la experiencia de los usuarios al momento de realizar alguna conexión con un servidor. Un ejemplo cotidiano de uso es cuando nos intentamos conectar a una página web, en donde solo ponemos el nombre de dominio de la página.

Este sistema está compuesto principalmente por 3 componentes: el espacio de nombres de dominio, los servidores de nombres, y los resolvers.

2.1. Espacio de nombres de dominio

El espacio de nombres de dominios corresponde a una estructura en forma de árbol en la cuál se guarda la información asociada a los nombres. Cada nodo y hoja de este árbol representa un nombre, en el cual se guarda la información asociada. En la imagen 2.1 se puede apreciar un ejemplo de la estructura del árbol de dominio. Dado que la estructura corresponde a un árbol, para obtener el nombre de dominio del nodo, debemos unir los nombres desde el nodo hasta la raíz. Como ejemplo, en la figura 2.1 el nombre de dominio de la hoja más a la izquierda sería 'uchile.cl.'. Además mencionar que dentro de estos nodos la información se almacena en una estructura conocida como Resource Record. De esta forma, cada nodo podría tener varios Resource Records en su interior, o incluso no poseerlos.

2.1.1. Resource Records

Tal como se mencionó anteriormente, los Resource Records son estructuras de datos que guardan una única información sobre un dominio. Estos no solo guardan un dato, si no que

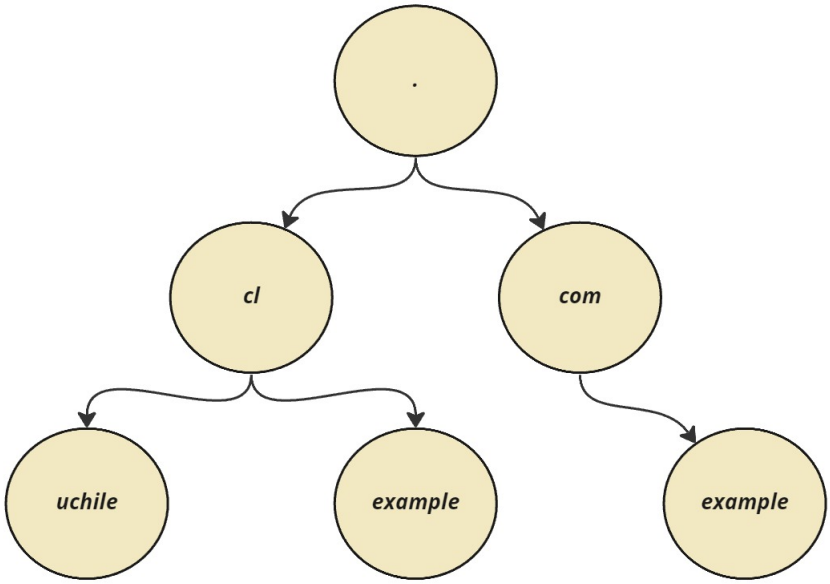

Figura 2.1: Ejemplo simple de un espacio de nombres de dominio

además contienen información adicional que permite facilitar el proceso de extracción de información. Algunos campos de información adicional son los siguientes:

- Owner: Corresponde al nombre de dominio que corresponde la información del Resource Record.
- Type: Corresponde al tipo de información que guarda el Resource Record. Los tipos más conocidos son los siguientes:
  - A: Identifica a una dirección IP para el dominio.
  - NS: Identifica el nombre de un servidor para el dominio.
  - Cname: Identifica un nombre para un alias.
- Ttl: Corresponde a la duración de validez que tendrá la información si es guardada en memoria caché. Esta información viene en segundos, y el caché solo guardará la información hasta que el Ttl expire. Expirando la información, es eliminada del caché y tendrá que ser preguntada nuevamente.
- Rdata: Corresponde a la información que guarda el Resource Record. Esta dependerá del tipo del Resource Record.

Para poder entender de mejor manera los Resource Records, a continuación se mostrará un ejemplo para cada tipo.

En la imagen 2.2, se puede apreciar un ejemplo de Resource Record de tipo A. En este se guarda una dirección IP que le corresponde al dominio 'uchile.cl'. Esta dirección IP se encuentra en el campo Rdata, y corresponde a '205.54.23.123'.

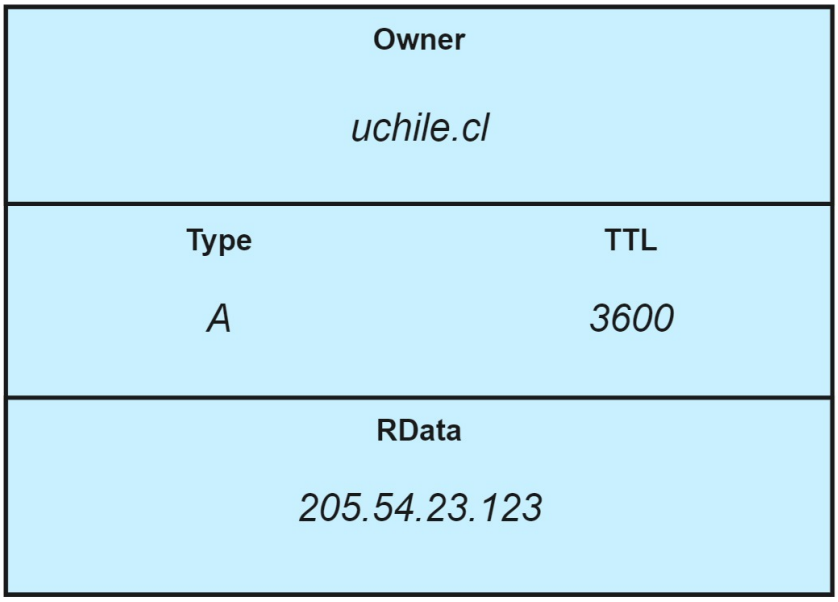

Figura 2.2: Ejemplo de Resource Record de tipo A

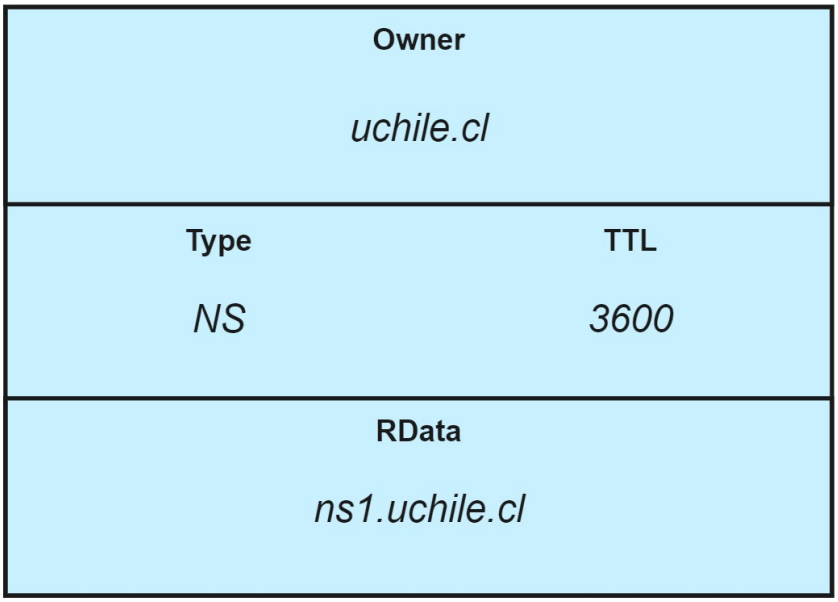

Figura 2.3: Ejemplo de Resource Record de tipo NS

En la imagen 2.3, se puede apreciar un ejemplo de Resource Record de tipo NS. En este se guarda un nombre de dominio que le corresponde a un servidor para 'uchile.cl'. Este nombre se encuentra en el campo Rdata, y corresponde a 'ns1.uchile.cl'.

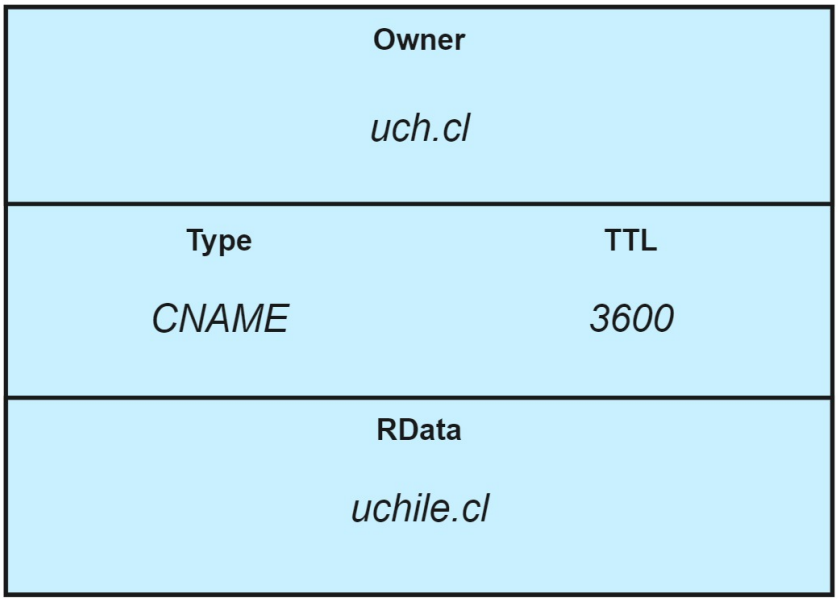

Figura 2.4: Ejemplo de Resource Record de tipo CNAME

En la figura 2.4, se puede apreciar un ejemplo de Resource Record de tipo Cname. En este se guarda un nombre de dominio que le corresponde al dominio 'uch.cl'. Esto significa que 'uch.cl' también puede ser conocido como 'uchile.cl'.

Finalmente en la figura 2.5 se aprecia un ejemplo de como los nodos se ven realmente con la información en su interior.

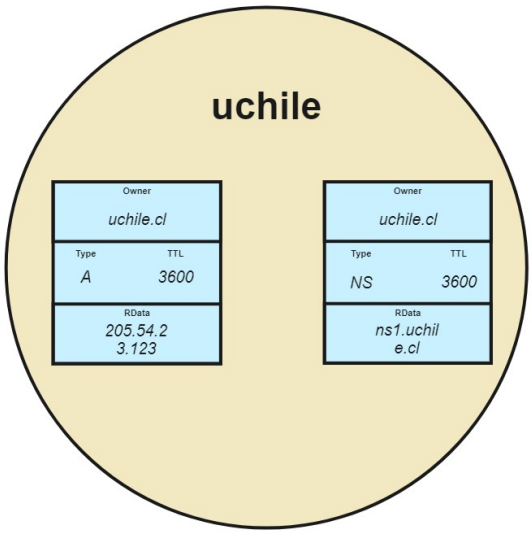

Figura 2.5: Ejemplo de la información del nodo 'uchile'

2.2. Servidores de Nombre

Los servidores de nombres o Name Servers (NS) son servidores que guardan la información del espacio de nombre de dominios. La información que almacenan son conocidas como *'zonas'*, las cuales corresponden a un conjunto de nodos. De esta forma, los servidores de nombre almacenan zonas para que luego puedan ser consultadas. En la figura 2.6 se muestra un ejemplo de cómo el espacio de nombres de dominio es divido en zonas.

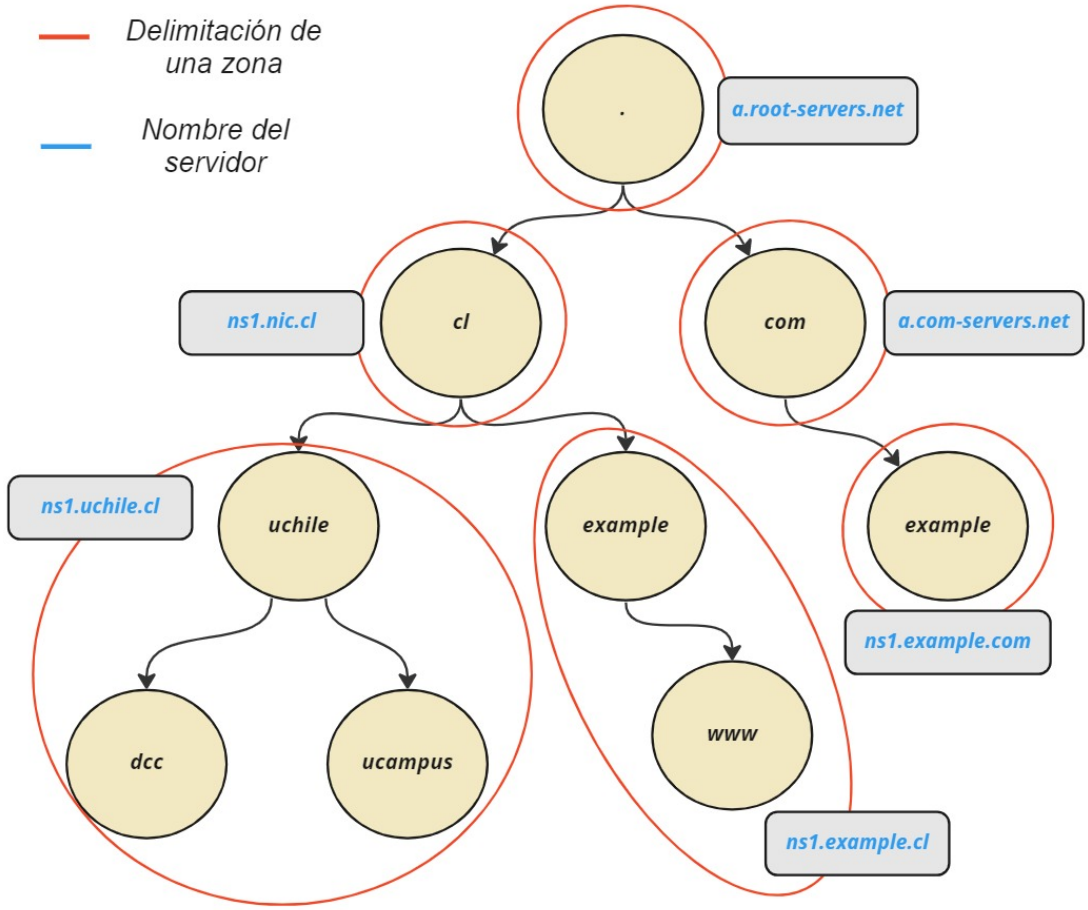

Figura 2.6: Ejemplo del espacio de dominio divido en zonas.

Cabe destacar que un name server es llamado 'Autoritativo' si posee toda la información de la sección del árbol de dominio que almacena. Esto permite saber si la información que entrega el servidor es confiable o no.

Por otro lado, la función principal de los name servers es responder consultas usando los datos almacenados en él. Para lograrlo, utilizan el algoritmo presentado en la figura 2.7.

En el paso 1, el name server busca si tiene disponible alguna de las zonas del dominio preguntado o de sus ancestros. Si esta zona existe, continua en el paso 2. En otro caso, el algoritmo termina.

Con respecto al paso 2, compara el nombre de la consulta con los nodos de la zona,

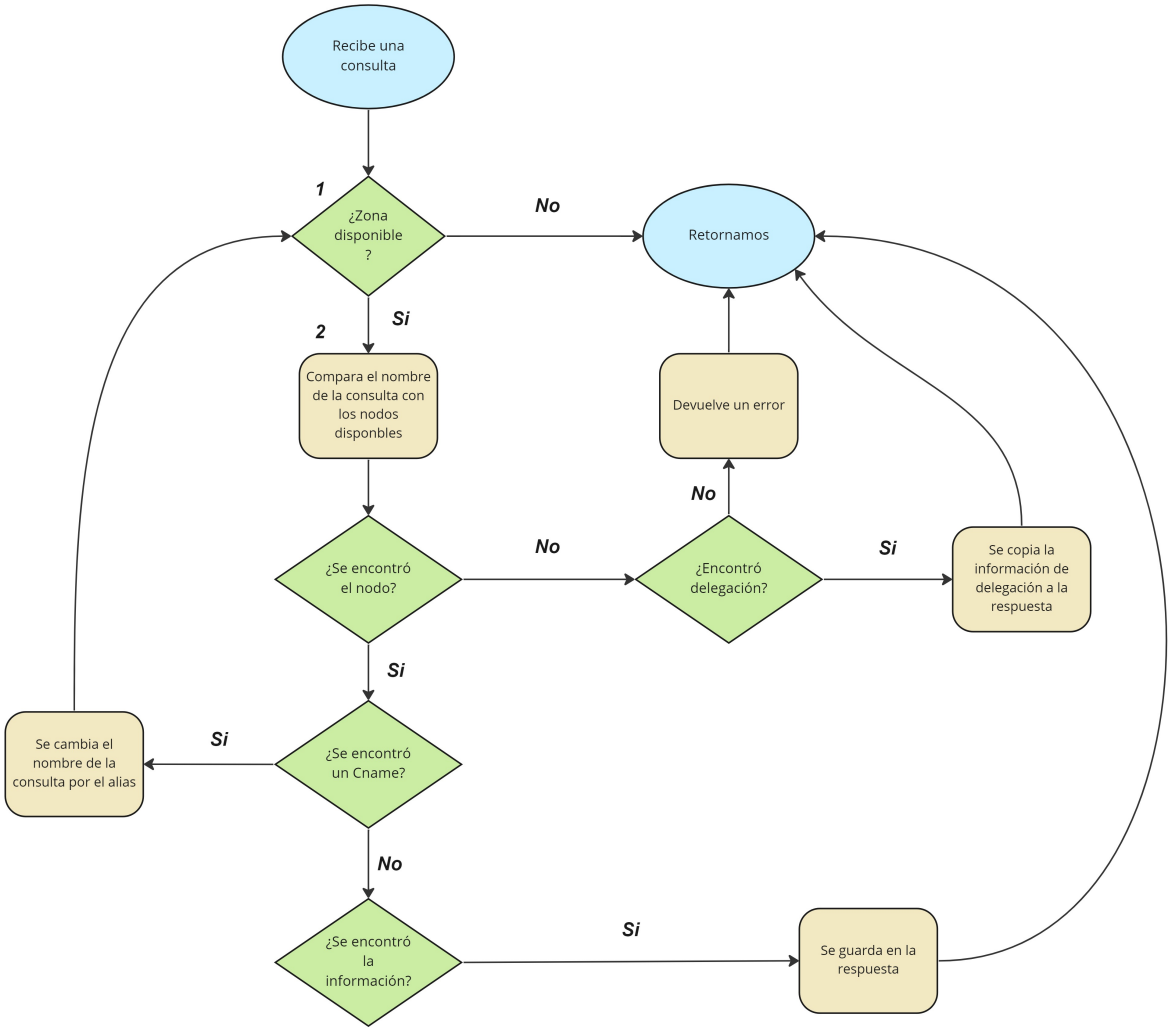

Figura 2.7: Diagrama del funcionamiento de los name servers.

partiendo en la raíz y bajando hacia las hojas, lo cual puede terminar de varias maneras:

1. Si encuentra todo el nombre de la consulta, entonces se encuentra el nodo buscado.
   Si los datos en el nodo corresponden a CNAME, y el tipo buscado no es CNAME, se cambia el nombre de buscado en la consulta por el nombre canónico del CNAME, y se vuelve al paso 1. En otro caso, se copian todos los Resource Records que coincidan con el tipo consultado a la respuesta.
2. Si buscando el nodo, se llega a una hoja con solo Resource Records de tipo NS, entonces significa que otro servidor tiene la información. Para ello se copian los Resource Records de tipo NS en la respuesta.
3. Si no se puede encontrar el nodo, entonces se devuelve un error autoritativo de nombre o NXDOMAIN.

Cabe destacar que el algoritmo anterior es una versión simplificada, pero contiene todo lo esencial para que el name server funcione correctamente. Es importante mencionar que al proceso del paso 2.2 se le conoce como *delegación*, y corresponde cuando un name server no posee la información buscada, pero conoce a otro servidor que podría tenerla.

Por otra parte, existe un caso particular en donde Resource Records de tipo A se encuentran junto a los de tipo NS en la delegación. A estos se les conoce como glue records, y es obligación que se encuentren cuando el nombre del Name Server a preguntar es subdominio de la zona actual. En este caso estos glue records se adjuntan en la respuesta, junto a los Resource Records de tipo NS en el paso 2.2.

Para entender de mejor manera el funcionamiento, a continuación se mostrarán algunos ejemplos del algoritmo. Para esto, se utilizará el ejemplo de zona mostrado en la figura 2.8.

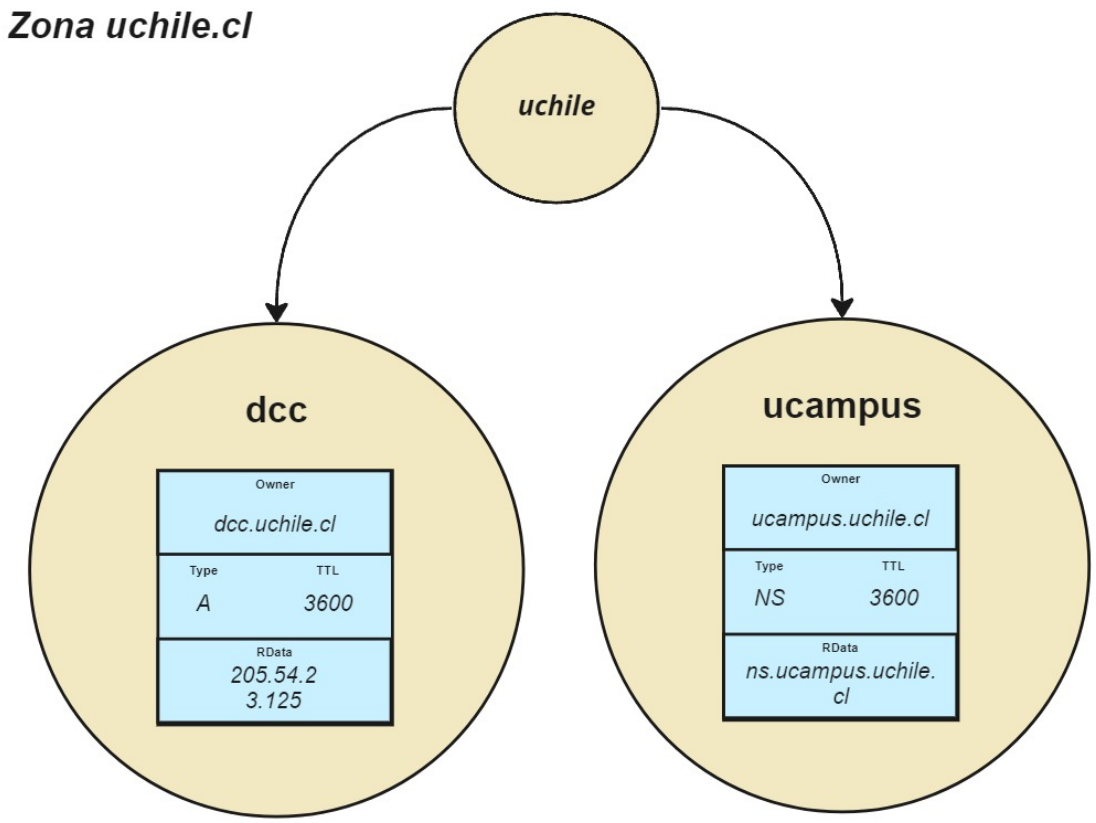

Figura 2.8: Ejemplo de la zona 'uchile.cl'.

Para el primer ejemplo, consultaremos la información de tipo A para el dominio 'dcc.uchile.cl'. Cuando el name server recibe esta consulta, lo primero que hará será revisar sus zonas para ver si alguna de ellas coincide con el nombre de dominio o sus ancestros. En este caso como la zona corresponde a 'uchile.cl' (que es ancestro de 'dcc.uchile.cl'), entonces encontrará dicha zona. Para continuar se irán comparando los nombres de los nodos con el nombre de la consulta. Como la zona corresponde a 'uchile.cl', entonces al comparar la raíz con el nombre de

la consulta, se encuentra que el primer nodo calza con el nombre, por lo tanto el Name Server procederá a buscar la parte restante. Luego, como el existe el nodo 'dcc', entonces hemos llegado al nodo esperado. Finalmente se buscarán aquellos Resource Records que coincidan con el tipo A, y se devolverán como respuesta.

Para el segundo ejemplo, consultaremos la información de tipo A para el dominio 'ucampus.uchile.cl'. Como ya sabemos, el servidor buscará si alguna zona coincide con el nombre a consultar. Al igual que el caso anterior, encontrará dicha zona. De forma similar, se irán comparando los nombres de los nodos con el nombre de la consulta. Como la zona corresponde a 'uchile.cl', entonces al comparar la raíz con el nombre a consultar, se encuentra que el primer nodo calza con el nombre.Luego, el name server buscará si existe un nodo con el nombre 'ucampus'. Como el nodo existe, hemos llegado al nodo esperado. Como el nodo es una hoja y posee Resources Records de tipo NS, entonces nos damos cuenta que corresponde a una delegación. Finalmente el servidor copia los Resource Records de tipo NS, y junto a ello los glue records que se encuentran en el nodo.

Para el último ejemplo, consultaremos la información de tipo A para el dominio 'ejemplo.uchile.cl'. Como ya sabemos, el servidor buscará si alguna zona coincide con el nombre a consultar. En este caso, encontrará dicha zona. Para continuar, se irán comparando los nombres de los nodos con el nombre de la consulta. Como la zona corresponde a 'uchile.cl', entonces al comparar la raíz con el nombre a consultar, se encuentra que el primer nodo calza con el nombre. Continuando, el Name Server buscará si existe un nodo hijo con el nombre 'ejemplo'. Luego, como el nodo no existe, entonces el servidor responderá con un error.

Complementando los ejemplos anteriores, en DNS existen dos tipos de respuestas: las respuestas positivas y las respuestas negativas. Las respuestas positivas corresponden cuando la respuesta entregada posee la información deseada por el cliente. En cambio las respuestas negativas corresponden a errores, o cuando la información deseada no existe. De las respuestas negativas, las más relevantes son 'NODATA' y 'NXDOMAIN'. 'NODATA' corresponde a cuando un servidor autoritativo no encuentra la información deseada. Por otro lado, 'NXDOMAIN' corresponde cuando el nodo de un dominio no es encontrado en la zona. Un ejemplo de 'NODATA' sería si en la zona de ejemplo de la figura 2.8 se consultará por la información de tipo CNAME para el dominio 'dcc.uchile.cl'. Por otra parte, un ejemplo para 'NXDOMAIN' es el ejemplo número 3 de los casos anteriores.

2.3. Resolvers

En la sección anterior, se habló de como los Name Servers responden las consultas. Es por esto que a continuación hablaremos de los Resolvers, que corresponden a un programa que realiza consultas a los Name Servers para así extraer la información que ellos poseen. En la gran mayoría de casos, los resolvers reciben consultas de clientes, como por ejemplo nuestro navegador, y devuelven la información deseada.

Un objetivo muy importante de los resolvers es disminuir el delay de la red, y la sobrecarga de los Name Servers, y es por esta razón que los resolvers poseen una memoria caché. Además, estos necesitan dos estructuras más para su funcionamiento: la 'Sbelt' y la 'Slist'. Dado que

el resolver necesita estas 3 estructuras antes mencionadas, comenzaremos hablando primero de ellas.

2.3.1. Caché

En el caso de DNS, el caché se encarga de guardar la información proveniente de consultas hechas a los Name Servers. Como las respuestas vienen en un formato de Resource Record, entonces el caché almacena en su interior Resource Records de distintos tipos para diferentes nombres de dominio.

Es importante mencionar que los datos recibidos y guardados en el caché, poseen un índice que permite saber su prioridad. Este es conocido como 'data ranking' y está estipulado en el RFC 2181 [8]. En este ranking entre más pequeño sea el índice, mayor será la prioridad de la información. Un pequeño resumen de las prioridades es el siguiente:

1. Datos provenientes de un archivo de zona primaria, que no sean 'glue data'.
2. Datos de una transferencia de zona, que no sean 'glue data'.
3. Datos incluidos en la respuesta de una consulta a un servidor autoritativo.
4. Datos de delegación de una respuesta autoritativa.
5. 'Glue records' de una zona primaria.
6. Datos provenientes de una respuesta no autoritativa.
7. Información adicional de una respuesta autoritativa, datos de delegación de una respuesta no autoritativa e información adicional de una respuesta no autoritativa.

Ahora que ya se sabe que es el 'data ranking', podemos explicar como funciona el caché. Cuando llega una respuesta, el caché realiza lo siguiente para realizar el guardado de datos:

- Si no existe en el caché el tipo de Resource Record para el dominio que recibe, entonces lo guarda inmediatamente con su respectivo 'data ranking'.
- Si el tipo de Resource Record existe para el dominio en el caché, entonces revisa el 'data ranking'. Si el 'data ranking' de la información guardada es mayor al de la información recibida, entonces borra del caché la información, y guarda la recibida. En caso contrario, simplemente no guarda la información recibida y se queda con la que ya poseía. En caso de que la información recibida sea diferente a la almacenada, y además posea el mismo data ranking, queda a manos del resolver saber que datos guardar [8].

Cabe destacar que no solo respuestas positivas son guardadas en el caché, sino también las respuestas negativas. En el RFC 2308 [4] y 8020 [5] se habla sobre este tema, pero ninguno menciona el 'data ranking' de las respuestas negativas. Es por esta razón, que en la presente

investigación se utilizaron los mismos 'rankings' dispuestos anteriormente para el caso de respuestas negativas.

Por último, mencionar que los datos guardados en el caché no tienen una duración de validez ilimitada, sino que está definida por el campo TTL que se encuentra en los Resource Records. Luego pasado aquel tiempo, la información queda inutilizable por el resolver. Para los casos de respuestas negativas, el TTL utilizado es aquel *default* que tiene la zona desde donde proviene la respuesta.

2.3.2. Sbelt y Slist

El 'Sbelt' corresponde a una estructura que guarda en su interior direcciones IP's de Name Servers correspondientes a la raíz del espacio de nombres de dominio. Esta estructura es necesaria, debido a que el resolver necesitará al menos un servidor para comenzar a realizar sus consultas.

Por otro lado, la 'Slist' es similar al 'Sbelt', en su interior guarda direcciones IP's de Name Servers para realizar consultas. La gran diferencia es que la 'Slist' se va construyendo durante el proceso de búsqueda de la respuesta, y se construye a partir de los datos almacenados en el caché. Cabe destacar que al explicar el proceso de funcionamiento del resolver, se verá en más detalle el funcionamiento de la 'Slist'.

2.3.3. Funcionamiento del Resolver

Luego de hablar sobre las componentes necesarias para el resolver, podemos proceder a conocer como estos funcionan. De forma general, el resolver comenzará preguntando desde la raíz del árbol de dominio, e irá bajando en el árbol hasta llegar al servidor que tiene la información que busca. Para ello, los resolvers siguen el flujo mostrado en la figura 2.9. Este flujo es indicado en el RFC 1034 [1].

En el paso 1, el resolver intentará buscar la respuesta a la consulta en la memoria caché. Si la logra encontrar, entonces simplemente la devuelve al cliente.

En el paso 2, se buscará obtener el mejor servidor para realizar la consulta. Para ello, buscará en el caché aquellos Resource Records de tipo NS que coincidan con el nombre de dominio de la consulta. Si no son encontrados, entonces seguirá con el padre del dominio, y así sucesivamente. Al encontrar Resource Records de tipo NS, son copiados a la Slist, y se buscan en el caché las direcciones IP's disponibles. Si no encuentra alguna IP, entonces el resolver comienza a realizar consultas para obtenerlas. Existe el caso donde la Slist no consigue encontrar ningún Resource Record de tipo NS. En este caso la Slist es inicializada con las direcciones IP's de la Sbelt.

Para el paso 3, el resolver comenzará a enviar la consulta a los diferentes servidores guardados en la Slist, para ello correrá un timeout entre cada consulta. Si el timeout se cumple, entonces consultará al siguiente Name Server, siguiendo la lista de forma circular. Cabe destacar que existen casos donde ninguno de los servidores responde y el resolver podría

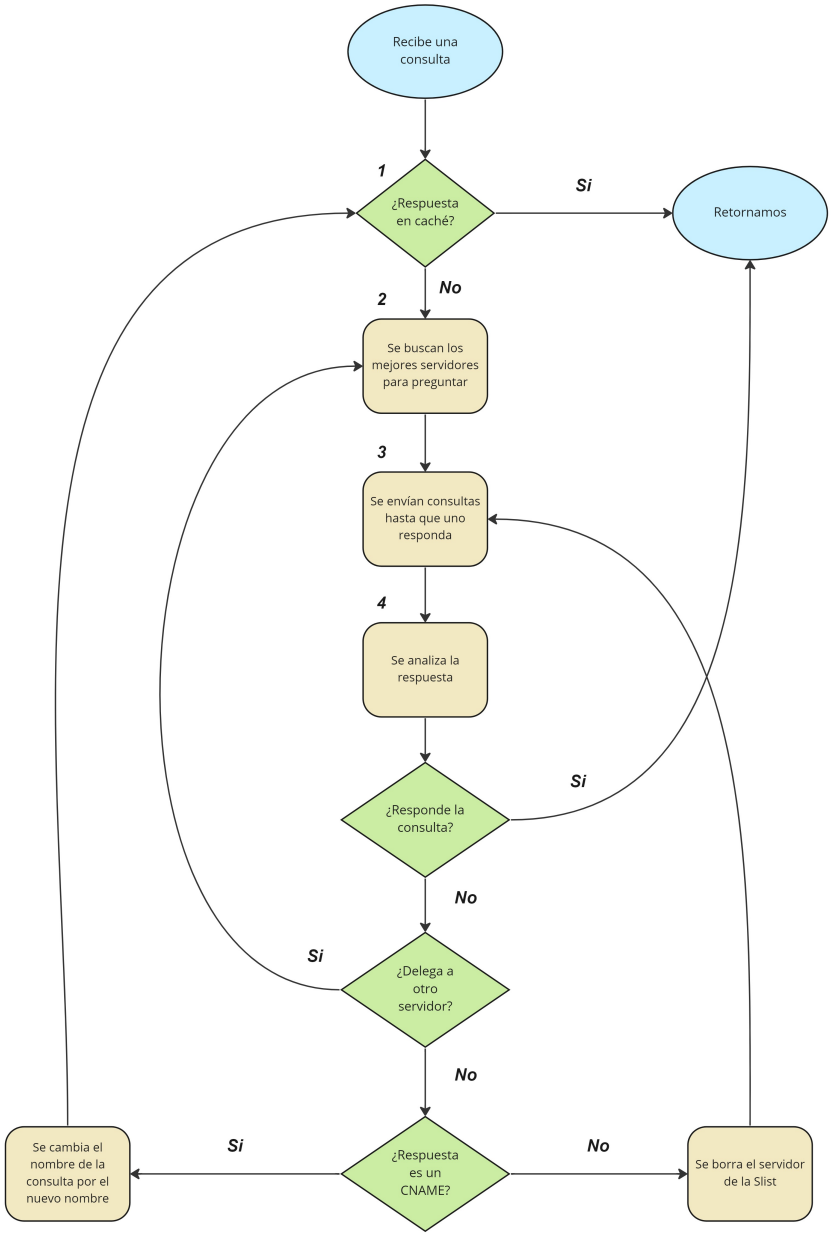

Figura 2.9: Flujo del funcionamiento del resolver.

quedar infinitamente preguntando. Es por esto, que existe un número de consultas limitado que puede realizar, por lo que en cada consulta, este número va disminuyendo. Finalmente si llega a 0, entonces el resolver arrojará un error conocido como 'Temporary Error'.

Finalmente en el paso 4 se analiza la respuesta. Si la respuesta es la deseada, o entrega un error, entonces se devuelve al cliente. Si muestra alguna delegación, se guarda la información de delegación en el caché, se vuelve a construir la Slist y se vuelve a consultar. Por último, si la respuesta contiene un CNAME, y este tipo no era el consultado, entonces se cambia el dominio de la consulta por el nombre encontrado en el Resource Record, y se reinicia la consulta.

A continuación se dejará un ejemplo explicado en detalle del funcionamiento del resolver. Para ello supondremos que existe el árbol de dominio mostrado en la figura 2.10. Además, destacar que cada zona del árbol esta demarcada en rojo, con el nombre del servidor a su lado.

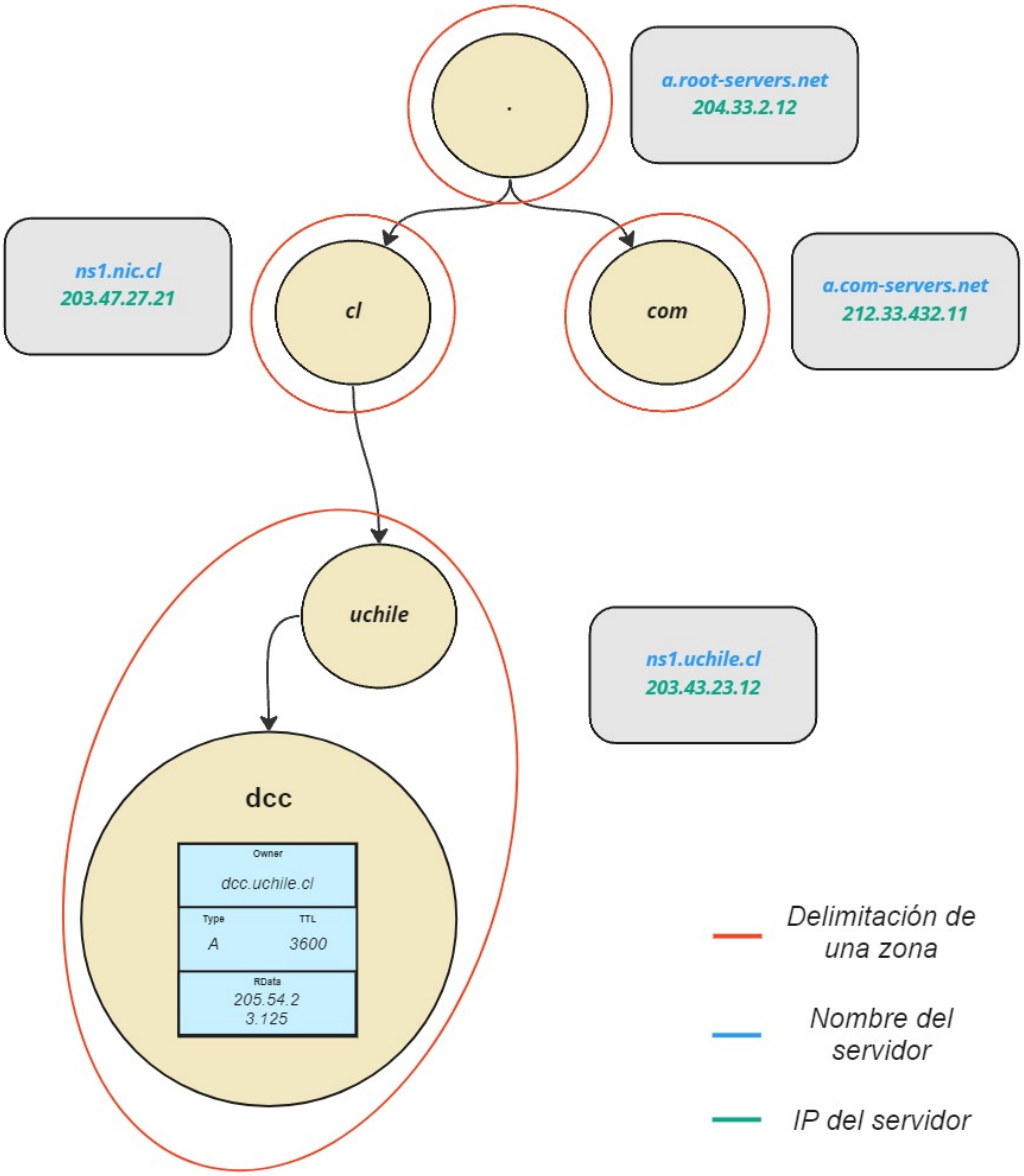

Figura 2.10: Árbol de dominio utilizado para el ejemplo del funcionamiento del resolver.

En la figura 2.10, podemos ver 4 zonas en el árbol de dominio. La zona de la raíz, la zona de '.cl', la zona de '.com', y la zona de 'uchile.cl'. Cabe destacar que entre cada nodo enlazado de diferentes zonas, existe una delegación junto con su información, que será omitida para simplificar el diagrama. Además asumiremos que las direcciones IP's de los servidores son aquellas que se encuentran en la imagen.

El resolver siempre comenzará con su Sbelt inicializada, en donde tiene guardada la IP del servidor raíz. Por otro lado, la Slist siempre comenzará vacía, pero no así el caché,

que dependerá si el resolver ha realizado otras consultas anteriormente o no. En este caso asumiremos que el resolver no ha hecho consultas, por lo que comenzará vacío. El estado del resolver en este momento es el mostrado en la figura 2.11.

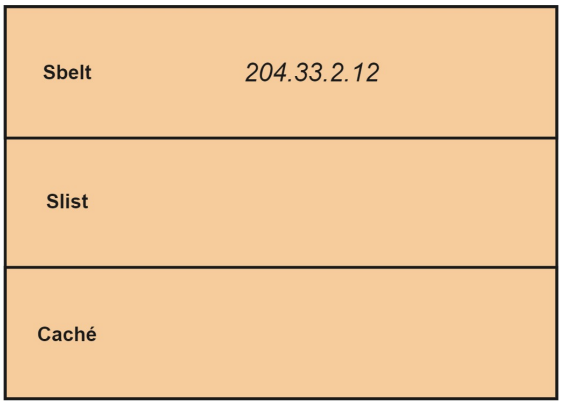

Figura 2.11: Estado del resolver al comenzar.

A continuación el resolver recibe una consulta por información de tipo A para el dominio 'dcc.uchile.cl'. Lo primero que hará será revisar en su caché si existe la información buscada, pero como este está vació, entonces no logrará encontrarla. Luego, el resolver intentará crear la Slist para la consulta, para ello, primero revisará en el caché si posee algún Resource Record de tipo NS para 'dcc.uchile.cl', pero como el caché sigue vació, entonces no encontrará. Al no encontrar los Resource Records, entonces intentará buscarlos para el padre de 'dcc.uchile.cl', es decir, para 'uchile.cl'. Al igual que el caso anterior, no podrá encontrarlos, por lo que buscará para 'cl'. Finalmente al no encontrar Resource Records de tipo NS en el caché que le sirvieran para crear la Slist, entonces la Slist se inicializa desde el Sbelt. El estado del resolver en este momento se ve como en la figura 2.12.

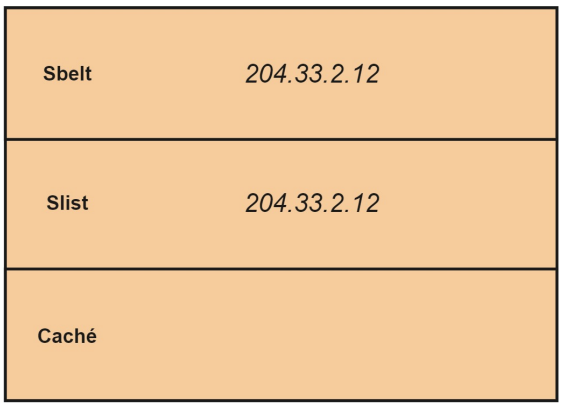

Figura 2.12: Resolver al tomar la Slist del Sbelt.

Dado que el resolver ya creó una lista con las mejores opciones de servidores a preguntar, entonces lo que hará será enviarle a estos servidores la consulta. Recordar que la consulta corresponde a la misma que fue recibida: información del tipo A para 'dcc.uchile.cl'.

Como el servidor raíz solo tiene información acerca de 'cl' y no de 'dcc.uchile.cl', entonces él responderá con la información de delegación, es decir, con los Resource Records de tipo NS y la IP asociada a él. Es por esto que el resolver cuando reciba la respuesta, entrará al paso 4.b, guardando la información recibida en el caché, y luego volviendo al paso 2 del algoritmo. En ese momento, el estado del resolver quedaría como se indica en la figura 2.13.

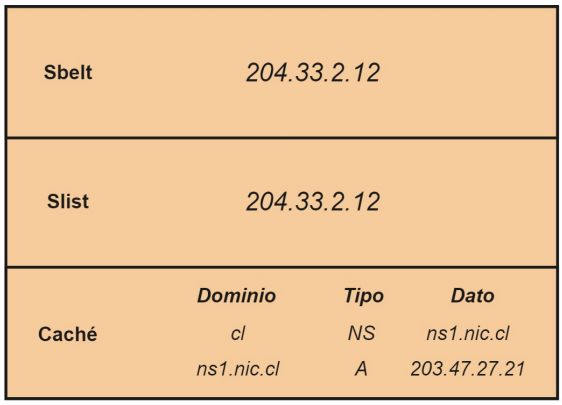

Figura 2.13: Resolver con información de delegación en el caché.

Puesto que el resolver volvió al paso 2, entonces tendrá que volver a construir una nueva Slist. Esto lo hará de forma similar a la vez anterior, pero con la diferencia que ahora cuando busque los Resource Records de tipo NS para 'cl', si encontrará. Al encontrarlos, el resolver los agrega a la Slist, y buscará a su vez en el caché la información correspondiente a la IP. En este caso si se posee la IP en el caché, por lo que la agregará a la Slist. El estado del resolver se ve como en la figura 2.14.

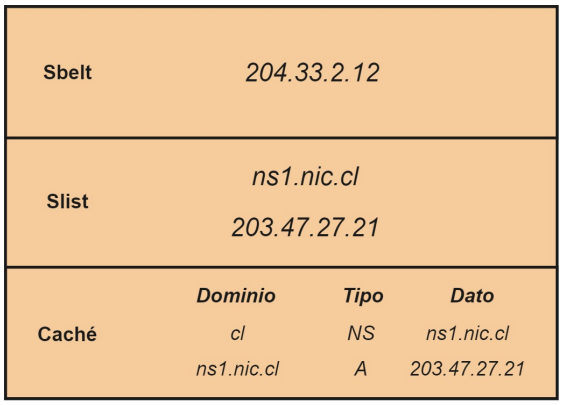

Figura 2.14: Resolver con Slist creada con la información de delegación.

Siguiendo el algoritmo, el resolver le preguntará al servidor de 'cl', por información de tipo A para 'dcc.uchile.cl'. De forma similar al servidor raíz, el servidor responderá con la información de delegación de 'uchile.cl'. El resolver recibirá la información, y la guardará en el caché. El estado del resolver cambiará, y será el mostrado en la figura 2.15.

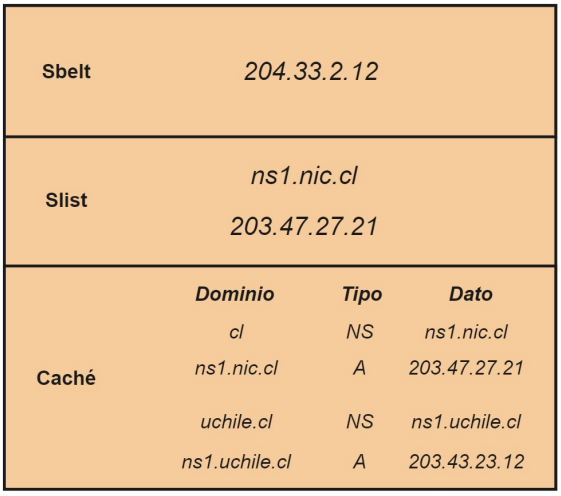

Figura 2.15: Resolver con información de delegación de uchile guardada.

Dado que el resolver debe continuar nuevamente con el paso 2, entonces procederá a crear una nueva lista de servidores para preguntar. Al igual que los casos anteriores, preguntará por los distintos dominios que se preguntaron al comienzo, pero esta vez se detendrá en 'uchile.cl'. Añadirá el Resource Record de tipo NS y su respectiva IP a la nueva Slist. El resolver en este preciso momento se encuentra como se muestra en la figura 2.16.

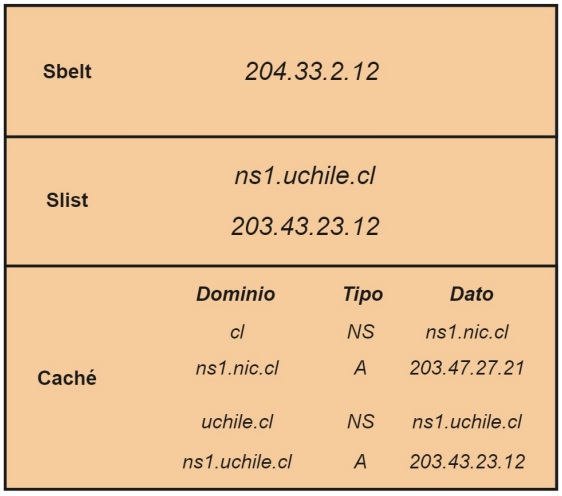

Figura 2.16: Resolver con información de delegación de uchile en Slist.

Para continuar, el resolver preguntará al servidor de 'uchile.cl' por información de tipo A de 'dcc.uchile.cl'. En este caso como el servidor de 'uchile.cl' posee la información que el resolver esta consultando, entonces la devuelve como respuesta al resolver. El resolver al recibir esta respuesta, guarda su información en el caché para futuras consultas, y devuelve la información al cliente. Finalmente, luego de terminar la consulta, el resolver queda como se indica en la figura 2.17.

Sbelt	*204.33.2.12*		
Slist	*ns1.uchile.cl* *203.43.23.12*		
Caché	***Dominio***	***Tipo***	***Dato***
	cl	*NS*	*ns1.nic.cl*
	ns1.nic.cl	*A*	*203.47.27.21*
	uchile.cl	*NS*	*ns1.uchile.cl*
	ns1.uchile.cl	*A*	*203.43.23.12*
	dcc.uchile.cl	*A*	*205.54.23.125*

Figura 2.17: Estado del resolver al finalizar la consulta.

2.4. Resolvers Parent-centric y Child-centric

En el ejemplo anterior existe un pequeño detalle omitido. Esto corresponde a que la información de delegación (correspondiente a los Resource Records de tipo NS) se encuentran no solamente en la zona padre de la zona a delegar, sino que también se encuentran en la propia zona delegada. Esto sucede debido a que la información de delegación es necesaria en la zona padre para así poder alcanzar otros servidores, pero a su vez esta corresponde a información propiamente tal de la zona. En la figura 2.18 se muestra un ejemplo, en donde la zona 'cl' posee la información de delegación de 'uchile.cl', pero a su vez 'uchile.cl' posee esta información dentro de su zona.

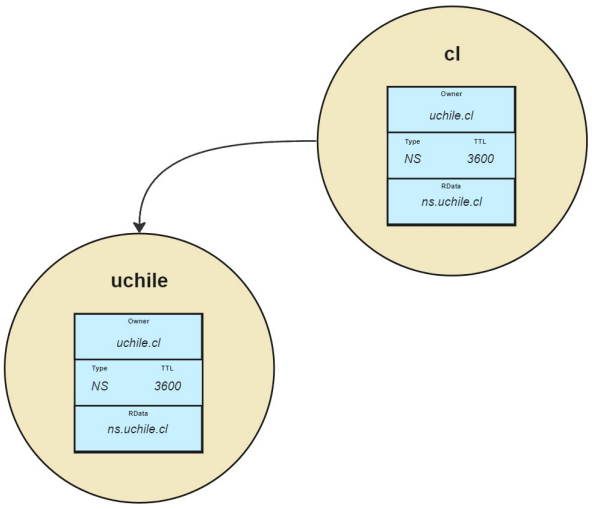

Figura 2.18: Ambas zonas poseen la información de delegación.

Si recordamos el algoritmo del resolver, al analizar una respuesta se debe guardar la información de delegación en el caché. Debido a que la información de delegación se encuentra en dos diferentes zonas, surgieron dos implementaciones de resolvers: los parent-centric y los child-centric [13, 15]. Los resolvers parent-centric son aquellos que confían en la información de delegación proveniente de la zona padre, y que guardan aquella información en el caché. Por otro lado, los resolvers child-centric confían en la información de delegación que se encuentra en la propia zona delegada, por lo que consultan la información a la zona, y es aquella la que guardan en el caché. Esto sucede debido a que la información proveniente de la propia zona delegada debería ser más confiable, puesto que esa información pertenece a la zona propiamente tal.

Lo anterior logra generar una pequeña inconsistencia en el algoritmo del resolver, haciendo que este no sea predecible, y a su vez, provocando que el protocolo de nombres de dominio no sea estándar.

2.5. Estandarización del algoritmo del Resolver

Gracias a la inconsistencia generada en el algoritmo del resolver, en el año 2020 fue publicado un nuevo 'draft' [10] para el protocolo de nombres de dominio. El objetivo principal de este 'draft', es estandarizar el algoritmo del resolver, y a su vez, aumentar la credibilidad de la información de delegación guardada en el caché del resolver. Para esto el autor propone principalmente dos cambios, estos son conocidos como Upgrading NS credibilty y Delegation Revalidation.

2.5.1. Upgrading NS credibility

Upgrading NS credibility hace referencia a mejorar la credibilidad de los Resource Records de tipo NS en las delegaciones. Para esto, propone realizar una consulta preguntando por los Resource Records de tipo NS en caso de delegación, parecido a como lo hacen los resolvers child-centric. Lo anterior permite mejorar la credibilidad de estos registros, dado que provienen de la zona propiamente tal. Cabe destacar que esta consulta se haría en paralelo con la consulta actual que el resolver busca responder.

Si se recibe una respuesta para la consulta anterior, entonces esta nueva información de delegación es guardada en el caché, reemplazando a la información proveniente desde la zona padre que anteriormente había sido guardada. Por otro lado si la consulta no recibe una respuesta, entonces quedará en el caché la información proveniente de la zona del padre. Cabe destacar que este cambio logra estandarizar el algoritmo del resolver.

2.5.2. Delegation Revalidation

Cuando la información obtenida del proceso anterior caduca en el caché, ahí es cuando Delegation Revalidation hace su aparición. Como dice su nombre, Delegation Revalidation

busca revalidar la información de delegación que se encuentra en el caché y saber si la delegación desde el padre fue cambiada. Para ello, el resolver preguntará a la zona padre por la información de delegación, y la comparará con la información que tenía guardada. Si alguno de los Resource Records guardados coincide con los del padre, entonces se considera que la información que estaba guardada era valida. En caso contrario, la información que había en el caché es eliminada.

Capítulo 3

Análisis del nuevo algoritmo

En el capitulo anterior se vio en más profundidad el funcionamiento del sistema de nombres de dominio, junto con los cambios que se buscan realizar al resolver. Es por esto que como parte del trabajo de investigación, analizaremos sus principales consecuencias. Si recordamos de la sección anterior, el principal cambio al algoritmo es realizar una consulta para obtener la información de delegación desde la zona en sí donde pertenecen esos registros.

Cabe destacar que en la sección anterior se mencionó que en Delegation Revalidation se busca comparar los registros obtenidos desde la zona padre con los del caché, pero el protocolo nos menciona que ambos registros deben ser iguales y coincidir [1], entonces ¿Por qué compararlos?. La respuesta a esta pregunta se basa en el trabajo de Sommese et al. [15], donde nos muestran que existen servidores mal configurados en los cuales estos registros difieren, y comprenden cerca del 8 %.

Lo anterior podría provocar que en estos casos mal configurados, las zonas de los dominios sean inalcanzables por un periodo de tiempo. Es por estas razones, que se investigará si el tiempo de respuesta de las consultas es afectado con el nuevo algoritmo, y también investigar aquellos casos que podrían llegar a provocar algún fallo para las consultas a la zona.

3.1. Comparación del Tiempo de Respuesta

3.1.1. Análisis teórico

Con respecto al tiempo de respuesta de las consultas, el draft nos recomienda lo siguiente: 'Cuando una respuesta de delegación es recibida durante iteración, una consulta de validación debería ser enviada en paralelo con la resolución de la consulta que lo provoca' [10]. Dado que la consulta de validación es enviada en paralelo, se espera que el tiempo de respuesta de las consultas no cambie de una manera significativa en comparación al algoritmo usado actualmente.

3.1.2. Experimentos

Para validar lo anterior, se utilizaron dos *resolvers*: uno con el algoritmo actual, y otro con el nuevo. Estos resolvers fueron implementados en el lenguaje de programación Rust, y se basan en el RFC 1034 y 1035 [1, 2], los cuales son los principales para el funcionamiento correcto del resolver. Además se le añadió el soporte de 'data ranking', y caché de respuestas negativas para las repuestas de NXDOMAIN, y NODATA.

Validación del resolver desarrollado

Para validar el funcionamiento del resolver desarrollado, se compararon los siguientes aspectos con un resolver existente: respuestas, name servers consultados y tiempo de respuesta. El resolver a ser utilizado corresponde al resolver del sistema operativo *Ubuntu*.

Con ambos resolvers, se procedió a consultar las direcciones IP de 50 dominios de los 100 más visitados en el mundo según 'semrush' [14]. Cada uno de los sitios web fue consultado 384 veces, para así aumentar la credibilidad de los datos obtenidos. Cabe destacar que con las 384 consultas, se logra un intervalo de confianza del 95 %, y un margen de error del 5 %. Este valor fue calculado utilizando la fórmula de *Cochran*.

Con respecto los name servers consultados, el resolver desarrollado mostró seguir las consultas correctas, es decir, todos los name servers consultados correspondían a name servers pertenecientes a los dominios de la cadena de consultas. De manera similar, el resolver desarrollado obtuvo en todas las consultas las respuestas correctas de los dominios.

Finalmente, como se puede apreciar en la figura 3.1, el tiempo de respuesta de ambos resolvers es bastante similar. En la mayoría de los casos el tiempo de respuesta del resolver desarrollado es mayor, pero es un resultado esperable, debido a que los resolvers ya existentes poseen un mayor grado de optimización.

Con lo anterior, podemos validar el comportamiento del resolver desarrollado, y así ser usado en los experimentos de la presente investigación.

Ejecución de los experimentos

Con ambos resolvers, se procedió a consultar las direcciones IP de los sitios web utilizados para la validación del resolver. Además, se agregaron 10 nuevos dominios pertenecientes a *.cl*, para así ver el comportamiento en dominios más cercanos. Cabe destacar que dado que se busca comparar los tiempos de respuestas de ambos algoritmos, los dominios a consultar pueden ser de cualquier parte del mundo. Al igual que en la validación del comportamiento del resolver, se realizaron 384 consultas a cada dominio. Así lograr que la media real del tiempo de respuesta se encuentre dentro de un rango de valores con un 95 % de certeza.

Del experimento se obtuvieron los tiempos de respuesta de ambos *resolvers*, para que así pudieran ser comparados y saber si existe alguna diferencia en el tiempo de respuesta al usar el nuevo algoritmo. Cabe destacar que las consultas se hicieron sin el uso del caché, debido

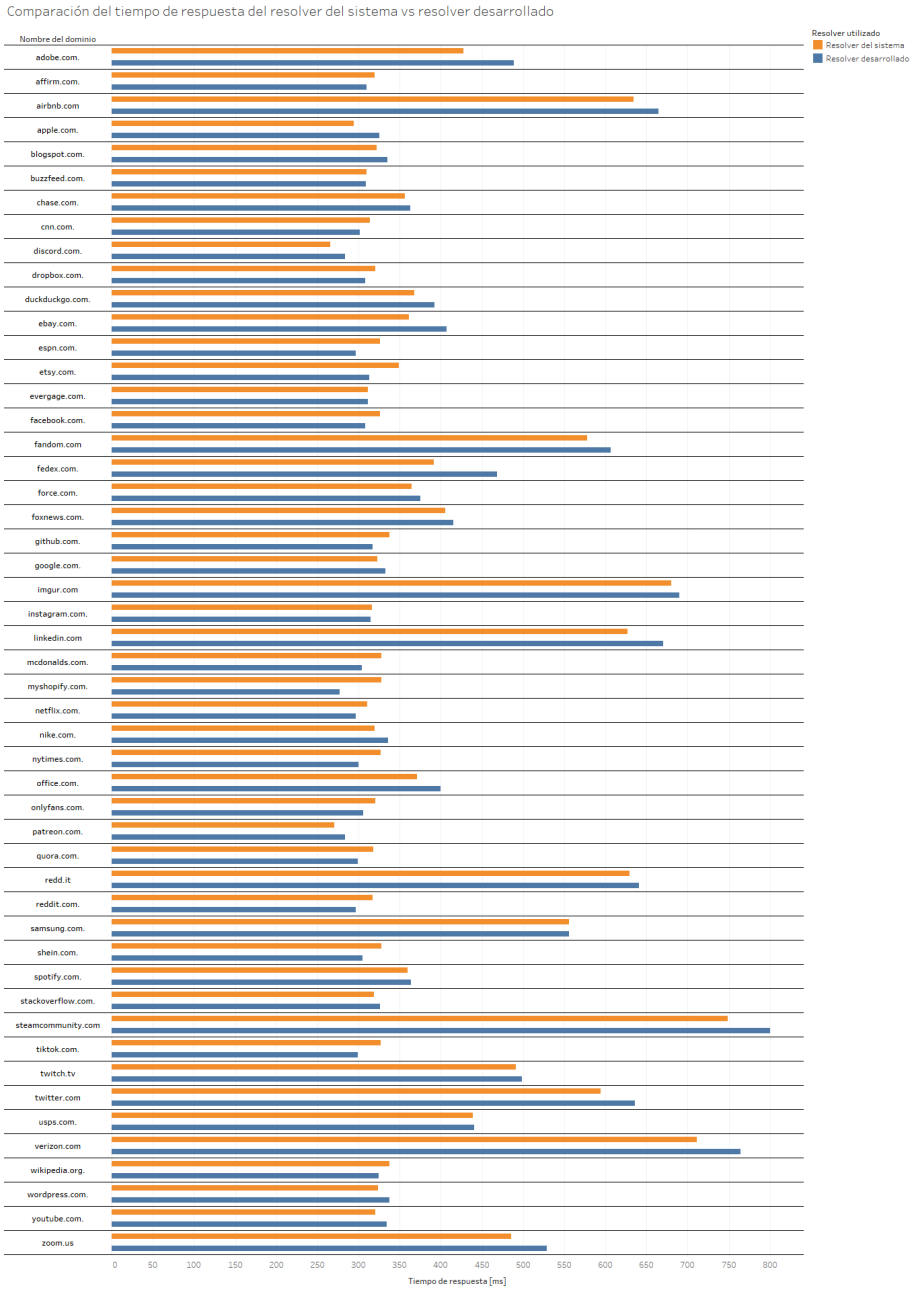

Figura 3.1: Diferencia del tiempo de respuesta entre el resolver desarrollado y el resolver del sistema operativo.

a que el cambio del algoritmo ocurre en caso de una delegación, por lo que la existencia del cache provocaría que el resolver no realice las delegaciones correspondientes.

Debido a que el tiempo de respuesta es afectado por la ubicación de los distintos name servers [12], al realizar las consultas los name servers fueron siempre los mismos para un dominio, para así poder comparar de forma correcta los tiempos obtenidos.

De forma similar, los *autonomous system* también pueden afectar el tiempo de respuesta, debido a sus políticas de ruteo [6]. Es por esto que el experimento fue ejecutado en tres diferentes proveedores de internet. Con lo anterior se logró evitar sesgos dentro de los resultados

de los experimentos, y dar una mayor credibilidad a los resultados.

3.1.3. Resultados

Con los tiempos de respuestas obtenidos, se procedió a calcular el promedio de ellos por sitio web. Estos resultados pueden observarse en el Apéndice A, en donde se muestran los tiempos para los diferentes resolvers y proveedores de internet. Luego se calculó el promedio entre los resultados de los tres proveedores de internet, y se procedió a calcular la diferencia entre el tiempo del algoritmo nuevo y el algoritmo actual. Los resultados pueden observarse en la figura 3.2 y en su porcentaje en la figura 3.3.

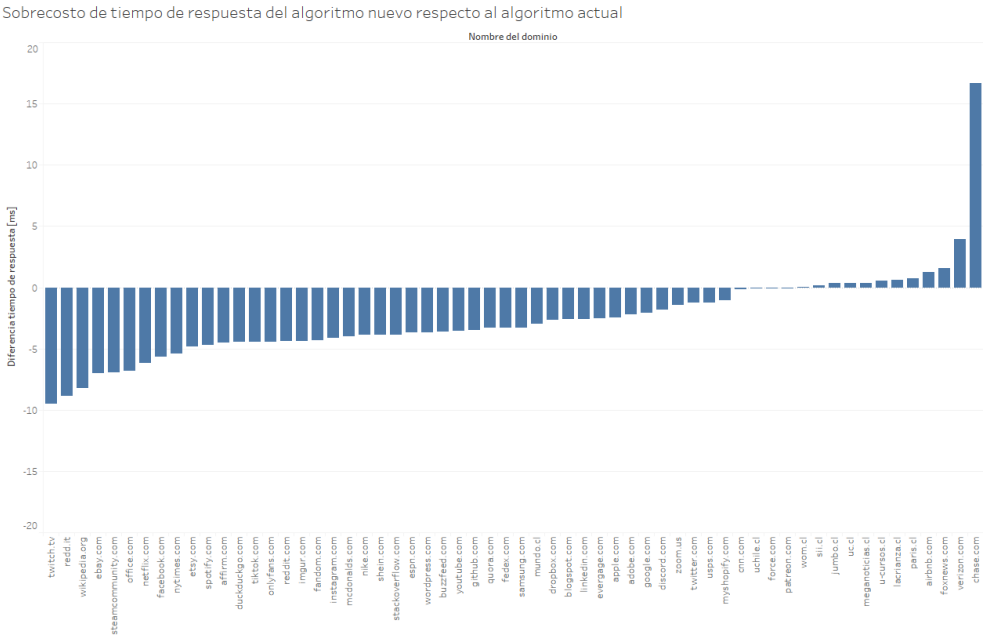

Figura 3.2: Sobrecosto de tiempo de respuesta entre el algoritmo actual y el algoritmo nuevo.

Tal como se puede apreciar en la figura 3.2, las diferencias de tiempos de respuestas entre ambos algoritmos son pequeñas. Además, el porcentaje de estas diferencias es menor al 4 % en el 100 % de los dominios. Esto se puede observar en la figura 3.3.

Luego de apreciar que las diferencias de tiempo de respuesta son tan pequeñas, y de confirmar que las consultas realizadas por los resolvers no tuvieran error alguno, se atribuyen estas pequeñas diferencias de tiempo a externalidades de la red.

Con esto se puede concluir que el tiempo de respuesta no se ve afectado significativamente en el nuevo algoritmo en comparación al algoritmo actual, gracias a que la consulta de validación es realizada en paralelo.

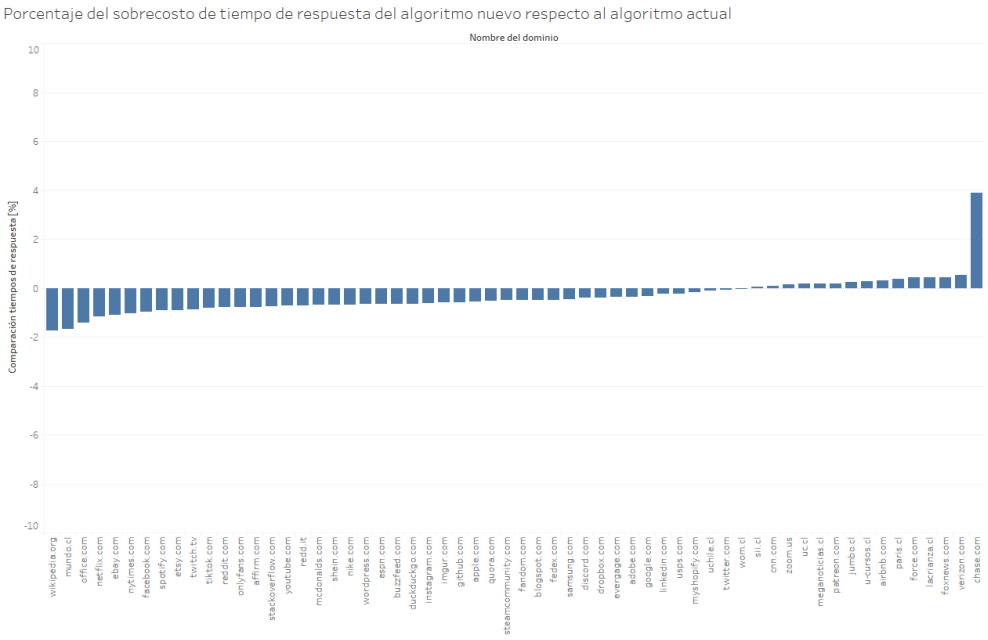

Figura 3.3: Porcentaje del sobrecosto de tiempo de respuesta en base al tiempo del algoritmo actual.

3.2. Análisis de Casos

3.2.1. Análisis teórico

Para continuar con el trabajo de investigación, se analizarán los diferentes posibles casos de mala configuración de los registros NS de los servidores y si tienen o no alguna consecuencia con los nuevos cambios propuestos al algoritmo del resolver. Cuando hablamos de casos mal configurados, nos referimos a aquellas zonas del espacio de nombres de dominio, en donde los registros de delegación difieren entre la zona padre y la zona delegada.

Por el lado de los casos bien configurados, estos no presentarán problemas con el nuevo algoritmo, ya que poseen los mismos registros NS en la zona padre como en la zona hija. Lo anterior hace referencia a que si los NS records de delegación del padre funcionan para el algoritmo actual, entonces al usar los de la zona hija en el nuevo algoritmo propuesto, estos también lo harán.

Para comenzar a hablar sobre los posibles casos problemáticos, primero debemos recordar el comportamiento del algoritmo del resolver al procesar una delegación. Cuando recibe un mensaje de delegación, este guarda los NS records junto a sus IP's en el cache. Para continuar, busca los mejores servidores para preguntar por la consulta, pero como recientemente había guardado la delegación en el caché, entonces el resolver encontrará estos Name Servers como mejor opción a preguntar. Cabe destacar que la cantidad de NS records que el resolver recibe es variada, en algunos casos podría ser solo 1, como en otros podrían ser 2 o más. Luego de encontrar los mejores servidores a preguntar, el resolver envía consultas hasta que alguno de los servidores escogidos responda. En caso de que algún servidor no responda, se espera un

timeout, y se envía la consulta a otro servidor. Este proceso continua enviando consultas hasta que alguno responda, o se acabe la cantidad de intentos que tiene configurado el resolver. Cabe destacar que la selección de los servidores a preguntar, es de manera circular, es decir que después de intentar todos los posibles, vuelve a preguntar al primero. Es por esto que si ninguno de los servidores responde, entonces el resolver no podrá responder la consulta, y arrojará un 'Temporary Error'.

Como en el draft [10] recomienda usar los NS del padre la primera vez que se consulta una zona (para así realizar una consulta paralela para obtener los NS records del hijo), entonces los problemas sucederán en consultas posteriores, y no en la primera consulta.

Conociendo como funciona el algoritmo en caso de delegación y el data ranking, ahora hablaremos de los casos mal configurados, en donde los registros NS de la zona hija no son exactamente los mismos que la zona padre, como nos indica el RFC 1034 [1]. Luego de analizar las diferentes situaciones en donde la lista del padre es diferente a la lista del hijo, se obtuvieron los siguientes casos. Cabe destacar que solo serán analizados aquellos casos donde la configuración permite el funcionamiento del dominio con el algoritmo actual.

1. En la zona hija existe al menos un registro NS que se encuentra en el padre, y aquel servidor responde por la zona.
2. En la zona hija existe al menos un registro NS que redirige a un servidor que responde por la zona en sí. Esto se refiere a que el servidor administra la zona, y por lo tanto puede responder por consultas de ella.
3. En la zona hija, ninguno de los servidores en los registros NS responde. Este caso se refiere a que ninguno de los servidores responde consultas DNS, por lo tanto al preguntarles, no se recibe respuesta alguna.
4. En la zona hija, no existen Resource Records de tipo NS.
5. El servidor responde correctamente a consultas de tipo A, pero responde NXDOMAIN para consultas de tipo NS.

Cabe destacar que debido a que el resolver actual guarda y utiliza la información de delegación del padre [1], es que estos casos funcionan con el algoritmo actual. A continuación se verán los diferentes casos de forma más detallada, y con sus respectivos ejemplos.

Caso 1

Para el primer caso, si existe al menos un registro NS en la zona hija que se encuentre en la zona padre, entonces este caso podrá funcionar correctamente, independientemente de que la IP para dicho registro se encuentre mal en la zona hija. Esto sucede gracias al 'data ranking', debido a que las IP's provenientes de glue data, tienen un mayor ranking que las IP's que llegarían en la consulta que realiza el nuevo resolver. Por lo tanto, seguiría en el caché la IP correcta asociada al registro NS, y el resolver podría consultar sin problemas a la zona.

Supongamos que tenemos las zonas mostradas en la figura 3.4. Para este ejemplo, consideraremos que las IP's *167.43.5.301* y *167.43.5.302* no redirigen a ningún servidor, en cambio, la IP *167.43.5.243* si lo hace. Cuando el resolver tenga que realizar una delegación, tendrá que guardar en el caché el registro NS y la IP que se encuentran en el padre, con un 'data ranking' de 4 y 5 respectivamente. El caché quedaría como se observa en la figura 3.5.

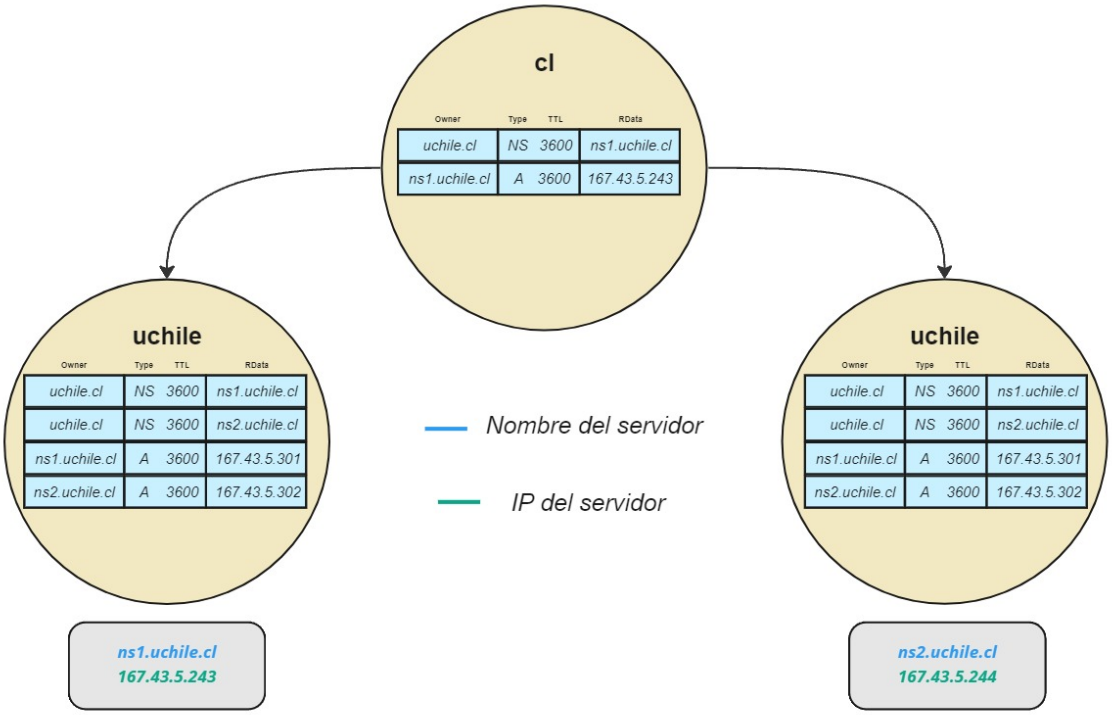

Figura 3.4: Ejemplo de zonas para el primer caso.

	Dominio	Tipo	Dato	Data Ranking
Caché	uchile.cl	NS	ns1.uchile.cl	4
	ns1.uchile.cl	A	167.43.5.243	5

Figura 3.5: Estado del caché al guardar la información de delegación del padre.

Para continuar, el resolver realizará una consulta a la zona hija para obtener los records de tipo NS y las IP's de delegación que se encuentran en la zona. En este caso obtendrá los registros *ns*1.*uchile.cl* y *ns*2.*uchile.cl* junto a sus IP's *167.43.5.301* y *167.43.5.302* (que están mal configuradas en la zona hija). Al guardar estos records en el caché, el resolver se encontrará de que ya existen registros para aquel Name Server, por lo que tendrá que revisar el 'data ranking' antes de continuar. Como los registros NS provenientes de la zona hija vienen de una respuesta autoritativa, entonces estos tienen un mayor ranking que los

que venían desde el padre, quedando estos guardados en el caché. Por otra parte, las IP's provenientes de la zona hija, vendrán en la sección Adicional de una respuesta, por lo que su 'data ranking' es menor a los que ya estaban guardados anteriormente. Es por esto que finalmente queda guardada la IP proveniente desde el padre (configurada como glue record), quedando el caché como se ve en la figura 3.6.

	Dominio	Tipo	Dato	Data Ranking
Caché	uchile.cl	NS	ns1.uchile.cl	3
	uchile.cl	NS	ns2.uchile.cl	3
	ns1.uchile.cl	A	167.43.5.243	5
	ns2.uchile.cl	A	167.43.5.302	7

Figura 3.6: Estado del caché al guardar la información de delegación del hijo.

Finalmente como queda guardada la IP funcional proveniente desde el padre, entonces el resolver no tiene problemas para continuar con sus futuras consultas.

Caso 2

En el segundo caso, si existe al menos un registro NS que dirige a un servidor que responde correctamente a la zona, entonces no existirán problemas con la delegación. Esto sucede debido a que el resolver irá consultando a todos los servidores guardados, hasta que alguno de ellos responda.

Por ejemplo, supongamos que tenemos la zona *uchile.cl*, la cual posee dos *name servers*: *ns1.uchile.cl* con IP *167.43.5.243*, y *ns2.uchile.cl* con IP *167.43.5.244*. En la figura 3.7 podemos ver los *resource records* de la zona, junto con los de su zona padre para delegación.

En la figura, podemos ver que las *IP's* de los *name servers* están mal configuradas. Una de ellas esta configurada correctamente, debido a que dirige al primer *name server*, en cambio la segunda contiene una *IP* que no corresponde a ninguno de los *name servers*. En este caso supondremos que esta *IP* no responde a consultas DNS. Primero el resolver al realizar la delegación, guardará la información proveniente desde el padre con un data ranking de 4 y 5 al igual que el ejemplo anterior, quedando el caché como se muestra en la figura 3.8.

Cuando el *resolver* vaya a consultar la información de delegación a la zona, este obtendrá y guardará ambos registros NS junto a sus *IP's*, gracias al 'data ranking'. Cabe destacar que el Name Server *ns*1.*uchile.cl* no quedará guardado en el caché, debido a que será sobreescrito por los nuevos registros provenientes desde el hijo. Además las nuevas IP's quedarán guardadas en el caché, gracias a que los NS records del padre y del hijo difieren, por lo que no existe ninguna IP asociada a los registros NS de la zona hija en el caché. El caché queda como de observa en la figura 3.9.

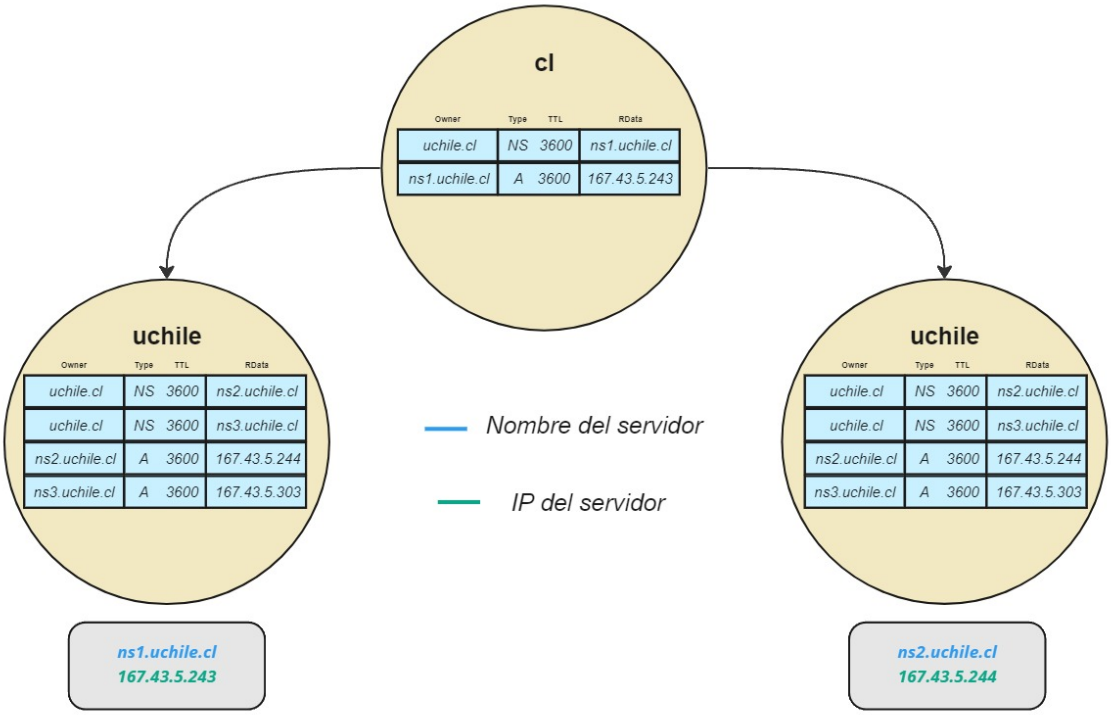

Figura 3.7: Ejemplo de zonas para el segundo caso.

	Dominio	Tipo	Dato	Data Ranking
Caché	uchile.cl	NS	ns1.uchile.cl	4
	ns1.uchile.cl	A	167.43.5.243	5

Figura 3.8: Estado del caché al guardar la información de delegación del padre.

Luego, cuanto el resolver quiera preguntar a esta zona alguna consulta posterior, preguntará a las *IP's* guardadas. Finalmente, como una de las *IP's* corresponde a un name server de la zona, entonces la consulta podrá ser realizada sin ningún problema.

Caso 3

En cuanto al tercer caso, si ninguno de los servidores en los registros NS del hijo responde, entonces el resolver quedará en un loop preguntando. Esto sucederá hasta que en algún momento el resolver arroje un Temporary Error. Continuando con el mismo ejemplo, en la figura 3.10 podemos ver los *resource records* de la zona, junto con los de su zona padre para delegación para este caso.

	Dominio	Tipo	Dato	Data Ranking
Caché	uchile.cl	NS	ns2.uchile.cl	3
	uchile.cl	NS	ns3.uchile.cl	3
	ns1.uchile.cl	A	167.43.5.243	5
	ns2.uchile.cl	A	167.43.5.244	7
	ns3.uchile.cl	A	167.43.5.303	7

Figura 3.9: Estado del caché al guardar la información de delegación del hijo.

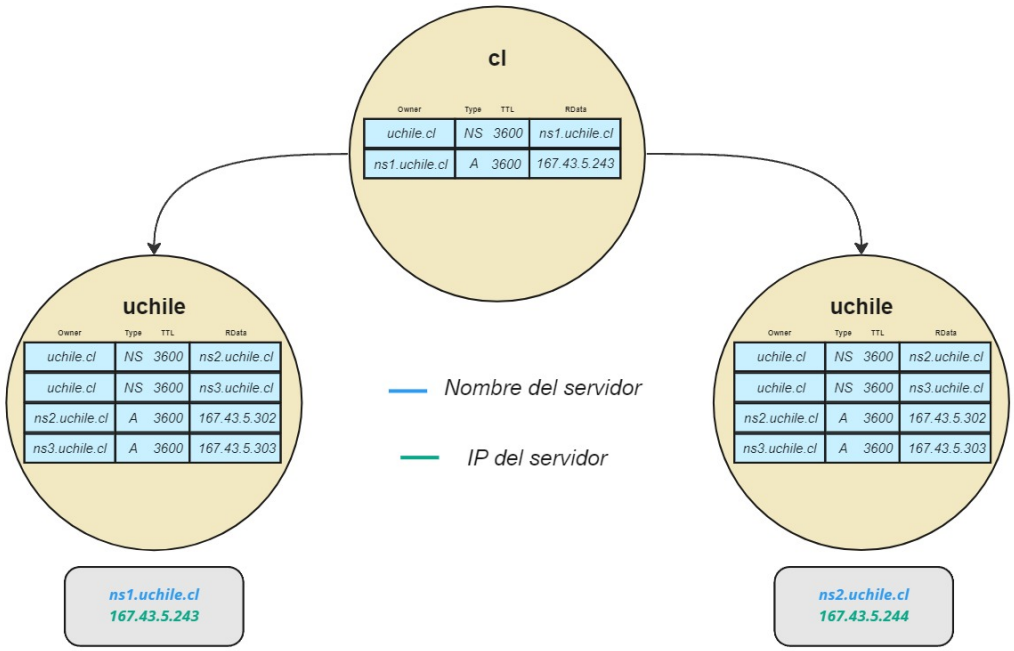

Figura 3.10: Ejemplo de zonas para el cuarto caso.

En la figura, se puede ver que ninguna de las *IP's* corresponde a alguna de los *name server*, y además, supondremos que ninguna responde consultas DNS. De la misma forma que los casos anteriores, se guardará primero la información de delegación del padre en el caché, mostrado en la figura 3.11.

Luego, cuando el resolver guarde la información de delegación obtenida desde la zona hija, esta sobre escribirá los NS ya guardados. Por otro lado, también guardará las nuevas IP's, puesto que no existía información para los dominios de los NS en el caché. Finalmente quedando el caché como en la figura 3.12.

Cuando el resolver quiera realizar consultas por otros dominios en esa zona, usará estos

	Dominio	Tipo	Dato	Data Ranking
Caché	uchile.cl	NS	ns1.uchile.cl	4
	ns1.uchile.cl	A	167.43.5.243	5

Figura 3.11: Estado del caché al guardar la información de delegación del padre.

	Dominio	Tipo	Dato	Data Ranking
Caché	uchile.cl	NS	ns2.uchile.cl	3
	uchile.cl	NS	ns3.uchile.cl	3
	ns1.uchile.cl	A	167.43.5.243	5
	ns2.uchile.cl	A	167.43.5.302	7
	ns3.uchile.cl	A	167.43.5.303	7

Figura 3.12: Estado del caché al guardar la información de delegación del hijo.

registros NS e *IP's* guardadas en el caché. Como ninguna de aquellas responde, entonces el resolver se quedará en un *loop* intentando preguntar. Como ninguna responde, entonces el resolver terminará en un *Temporary Error*, sin poder responder a la consulta.

Caso 4

En cuanto al cuarto caso, si en la zona hija no existen Resource Records de tipo NS, entonces una respuesta de validación vacía será enviada al resolver. Según el RFC 2308 [4], el resolver guardará en el caché esta respuesta como una respuesta negativa. Además nos dice que esta respuesta solo afectará a aquellas consultas que pregunten por el mismo dominio, y mismo tipo de resource record. Por lo tanto, en este caso el resolver no será capaz de seguir respondiendo consultas, gracias a que el 'data ranking' de esta información, no permitirá al resolver guardar la información de delegación para acceder a la zona.

Al igual que los casos anteriores, consideremos la zona de ejemplo situada en la figura 3.13. Primero la información de delegación será guardada en el caché, quedando como se observa en la figura 3.14.

Luego al realizar la consulta por los registros NS a la zona hija, el resolver recibirá una respuesta autoritativa vacía, indicando que no existe la información que se requiere. Es por

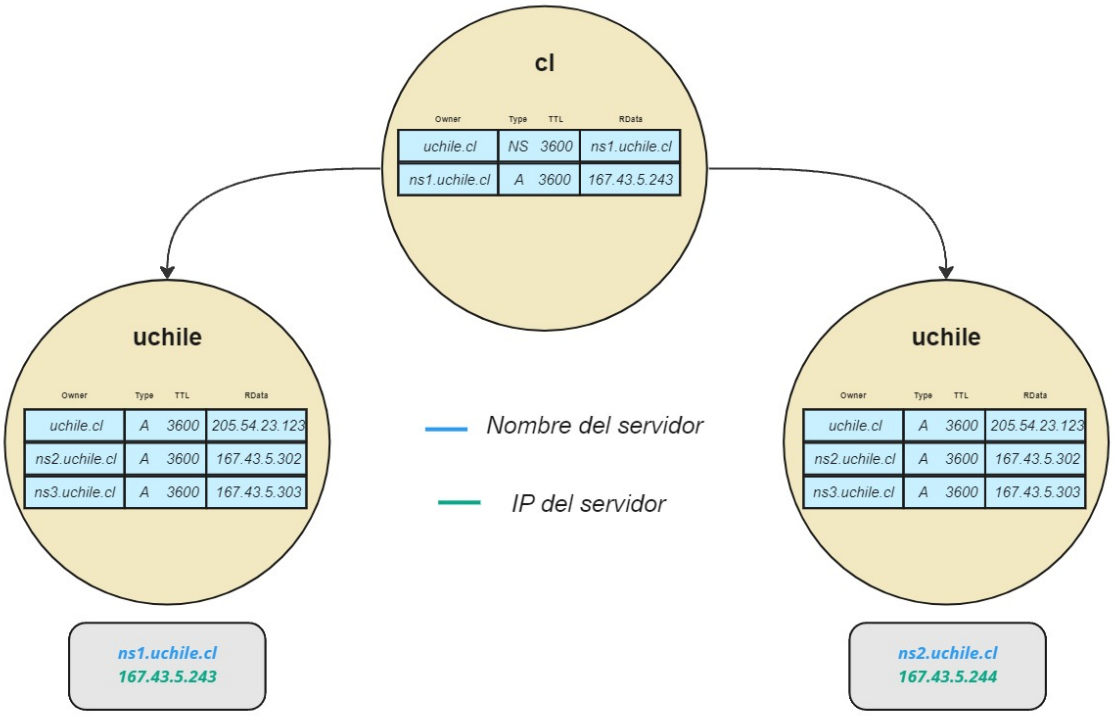

Figura 3.13: Ejemplo de zonas para el quinto caso.

	Dominio	Tipo	Dato	Data Ranking
Caché	uchile.cl	NS	ns1.uchile.cl	4
	ns1.uchile.cl	A	167.43.5.243	5

Figura 3.14: Estado del caché al guardar la información de delegación del padre.

esto que el resolver guardará en el caché esta respuesta negativa, con su respectivo 'data ranking'. Esto se puede apreciar en la figura 3.15.

	Dominio	Tipo	Dato	Data Ranking
Caché	uchile.cl	NS	NO DATA	3
	ns1.uchile.cl	A	167.43.5.243	5

Figura 3.15: Estado del caché al guardar la información de delegación del hijo.

Lo anterior afectará las siguientes consultas al momento de que el resolver quiera construir la lista con los servidores a consultar. Supongamos que se preguntará por algún otro dominio perteneciente a la zona, por ejemplo, 'www.uchile.cl' por información de tipo A. El resolver intentará buscar los NS records de 'uchile.cl', pero se dará cuenta que según el cache esta información no existe, por lo tanto irá a consultar a los dominios padres, es decir, '.cl'. Suponemos que logra encontrar esta información, por lo tanto le enviará la consulta al servidor de '.cl', recibiendo como respuesta la información de delegación de 'uchile.cl'. Al intentar guardar esta información, el resolver no podrá hacerlo, dado a que el data ranking de la información de delegación proveniente de un padre posee menor ranking que la que estaba ya guardada. Finalmente por esta razón, el resolver quedará infinitamente intentando llegar a la zona 'uchile.cl', sin poder lograrlo.

Caso 5

Un último caso problemático (que no tiene que ver con Resource Records mal configurados) tiene que ver con servidores que responden de mala forma a consultas de tipo NS. Estos servidores entregan una respuesta de NXDOMAIN por error, el cual significa que el nombre del dominio no existe. Este tipo de respuestas son errores del servidor, ya que si se pregunta por el mismo dominio pero por otro tipo de Resource Record, entonces responden correctamente. Este caso se estudiará debido a que es conocido [17].

Si la zona hija responde con NXDOMAIN, entonces el resolver guardará en el cache esta respuesta negativa. Al igual que el caso anterior, esta respuesta afectará de la misma forma al resolver. Es decir, no permitirá que este pueda construir la lista de servidores para consultar, logrando que la zona sea inalcanzable.

Cabe destacar que el no poder obtener la información de una zona debido a estas malas configuraciones, solo perdurará hasta que la información del caché caduque, es decir, hasta que se haga la revalidación de la información de delegación (Delegation Revalidation). Por otro lado, se podrá obtener la IP del dominio (gracias al caché), pero no consultar nuevos dominios en aquella zona.

3.2.2. Experimentos

Para comprobar empíricamente si los potenciales casos encontrados fallarán, se creó un Name Server de forma local con el software Bind9. Luego los diferentes casos fueron recreados en él. Los datos fueron creados por el autor, ya que solamente se necesitaban que los datos fueran consistentes, y no su contenido en sí.

El experimento consistió en hacer consultas DNS al *name server* con los dos *resolvers* implementados para el experimento de tiempo de respuesta. Dado que la diferencia de los algoritmos reside en la información que guardan en el caché, primero se hizo una consulta para que los *resolvers* guardaran la información de delegación. Cabe destacar que esta consulta no se tomó en cuenta para los resultados del experimento. Luego, se hicieron 5 consultas al servidor, para así obtener las respuestas de los *resolvers*. La cantidad de consultas a realizar

por caso es mucho más baja que el análisis anterior, debido a que *name servers* y *resolvers* fueron ejecutados en la misma máquina de forma local, disminuyendo considerablemente los problemas y errores de red. De este experimento se recolectaron las respuestas DNS y se compararon. Si el nuevo *resolver* no logra encontrar la respuesta, significa que este fallará en el caso que se está estudiando. Finalmente el experimento se hizo por cada uno de los posibles casos a fallar encontrados.

3.2.3. Casos encontrados

Luego de realizar el experimento, se encontró que los casos problemáticos son el caso 3, el caso 4, y el caso 5. Estos casos no permitirán al resolver devolverles la respuesta esperada a sus clientes (por ejemplo una IP), por lo que los usuarios no podrán acceder a los servicios de los dominios alojados en esta zona. Cabe destacar que esto perdurará solamente hasta que la información de delegación sea revalidada.

Para aquellos casos donde el dominio no posea *glue records*, habrá que consultar por la dirección IP del name server. Luego de obtener esta dirección IP, esta se guarda en el caché, y el proceso continua al igual que los casos donde existe el glue record. Destacar que dentro de esta consulta puede ocurrir cualquiera de los casos antes mencionados.

3.3. Análisis de los dominios de la zona .cl

Luego de verificar que alguno de los casos presentados anteriormente producirá una falla, se procedió a investigar cuales son aquellos dominios en nuestro país que cumplen con los casos estipulados. Para lograrlo se compararon los diferentes registros de delegación entre la zona padre y la zona del dominio.

3.3.1. Procedimiento

Lo primero fue obtener el archivo de zona de '.cl', para así conocer los dominios que se encuentran en ella, y también sus registros de delegación. Este archivo fue obtenido directamente de la entidad administradora de los dominios en Chile. A continuación se procedió a consultar los registros de tipo NS para cada dominio, y así obtener estos registros desde sus propias zonas. Luego con los registros provenientes desde la zona padre y la zona del dominio, se comenzaron a comparar los registros para así encontrar los casos estudiados en el capitulo 3.2. Finalmente para aquellos casos en donde no fue posible asegurar su funcionamiento solo con la comparación de los registros, se realizaron consultas a sus zonas con los *resolvers* utilizados en los experimentos de la sección anterior.

3.3.2. Resultados

Del total de dominios analizados, se encontró que el 19,03 % corresponden a dominios que poseen diferentes registros en su zona en comparación a su zona padre. De estos, el 61,12 % tuvieron que ser chequeados con los *resolvers* implementados. Con respecto a la cantidad total de dominios de la zona, se encontró que el 10,33 % pertenecen al caso 1, un 7,95 % pertenecen al caso 2, un 0,57 % pertenecen al caso 3, y un 0,16 % pertenecen al caso 4. Además, no fueron encontrado dominios correspondientes al caso 5.

Finalmente del 100 % de los dominios en la zona '.cl', un 0.74 % se verían afectados en caso de que se comience a utilizar el nuevo algoritmo. Estos resultados se deben a que en Chile existe una gran cantidad de dominios que no coinciden sus registros NS en la zona padre, con respecto a los de la zona hija.

Por otro lado, se confirmó que todos los dominios que serán afectados no funcionan actualmente con el algoritmo nuevo, pero si lo hacen con el algoritmo actual. Cabe destacar que los nombres de los dominios clasificados pueden ser encontrados en un documento adjunto en el Anexo B.

Capítulo 4

Discusión y Conclusión

A través de la presente investigación se logró encontrar los diferentes casos de mala configuración en los cuales el algoritmo analizado podría causar que una zona de un dominio se vuelva inalcanzable por un periodo de tiempo. Estos casos son los siguientes:

- En la zona hija, ninguno de los servidores de los registros NS responde. Este caso se refiere a que ninguno de los servidores responde consultas DNS, por lo tanto al preguntarles, no se recibe respuesta alguna.
- En la zona hija, no existen Resource Records de tipo NS.
- El servidor responde con NXDOMAIN cualquier consulta por registros NS, pero responde correctamente a consultas de otros tipos de registro.

Por otro parte, se logró comprobar que el tiempo de respuesta no es afectado significativamente por la implementación del nuevo algoritmo, gracias a que la diferencia del tiempo de respuesta entre ambos algoritmos es mínimo.

En cuanto al análisis de los dominios de la zona 'cl', de los 550.825 (100 %) dominios, 104.847 (19.03 % del total) corresponden a dominios que difieren en sus registros NS de la zona padre y de la zona hija. Destacar que esta cantidad es cerca del 8 % para zonas como 'com', 'org' y 'net', según el trabajo *'When parents and children disagree: Diving into DNS delegation inconsistency'* [15]. Esto es preocupante, ya que según Sommese et al. (2020) podría traer consecuencias con respecto del balance de carga de los servidores, pudiendo llegar a aumentar el tiempo de respuesta de las consultas y la cantidad de servidores que no responden.

Finalmente, los dominios afectados que quedarían inalcanzables si el nuevo algoritmo es implementado son 4.077 (0.74 % del total). Esto quiere decir que se logra estandarizar el algoritmo en caso de delegación afectando solamente el 0.74 % de los dominios nacionales. Además, se desarrolló una aplicación web que permite consultar si un dominio se verá afectado o no con esta problemática, para más información consulte el Anexo C.

Cabe mencionar que al momento de la escritura de la presente investigación, una nueva versión del 'draft' fue publicada [11]. En ella los autores indican que al realizar la consulta para

la obtención de los registros NS desde el hijo, solamente se guardarán en caché las respuestas positivas, y no las negativas como sucedía en la versión anterior del 'draft'. Esto produce que solamente el primer caso mencionado cause que un dominio se vuelva inalcanzable. Dada esta circunstancia, solo 3.147 (0.57 %) dominios nacionales se verían afectados.

Finalmente, se recomienda encarecidamente arreglar la mal configuración de los registros NS, para así evitar problemas de balance de carga, y también para que los dominios no sean afectados si el nuevo algoritmo llegase a implementarse prontamente. Para esto se recomienda a los dominios afectados actualizar los registros NS de su zona, y confirmar de que sean los mismos que se encuentran en la zona *.cl*.

Bibliografía

[1] Domain names - concepts and facilities. RFC 1034, November 1987.

[2] Domain names - implementation and specification. RFC 1035, November 1987.

[3] Abhishta Abhishta, Roland van Rijswijk-Deij, and Lambert JM Nieuwenhuis. Measuring the impact of a successful ddos attack on the customer behaviour of managed dns service providers. *ACM SIGCOMM Computer Communication Review*, 48(5):70–76, 2019.

[4] Mark P. Andrews. Negative Caching of DNS Queries (DNS NCACHE). RFC 2308, March 1998.

[5] Stéphane Bortzmeyer and Shumon Huque. NXDOMAIN: There Really Is Nothing Underneath. RFC 8020, November 2016.

[6] Juan Camilo Cardona, Stefano Vissicchio, Paolo Lucente, and Pierre Francois. "i can't get no satisfaction": Helping autonomous systems identify their unsatisfied interdomain interests. *IEEE Transactions on Network and Service Management*, 13(1):43–57, 2016.

[7] Quang-Vinh Dang. Detecting the attacks to dns. In *International Conference on Comprehensible Science*, pages 173–179. Springer, 2021.

[8] Robert Elz and Randy Bush. Clarifications to the DNS Specification. RFC 2181, July 1997.

[9] Amir Herzberg. Dns-based email sender authentication mechanisms: A critical review. *Computers & security*, 28(8):731–742, 2009.

[10] Shumon Huque, Paul A. Vixie, and Ralph Dolmans. Delegation Revalidation by DNS Resolvers. Internet-Draft draft-ietf-dnsop-ns-revalidation-02, Internet Engineering Task Force, March 2022. Work in Progress.

[11] Shumon Huque, Paul A. Vixie, and Ralph Dolmans. Delegation Revalidation by DNS Resolvers. Internet-Draft draft-ietf-dnsop-ns-revalidation-03, Internet Engineering Task Force, September 2022. Work in Progress.

[12] Richard Liston, Sridhar Srinivasan, and Ellen Zegura. Diversity in dns performance measures. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 19–31, 2002.

[13] Giovane CM Moura, John Heidemann, Ricardo de O Schmidt, and Wes Hardaker. Cache me if you can: Effects of dns time-to-live. In *Proceedings of the Internet Measurement Conference*, pages 101–115, 2019.

[14] Semrush. Top 100: The most visited websites in the us, 2022. [Web; accedido el 14-10-2022].

[15] Raffaele Sommese, Giovane Moura, Mattijs Jonker, Roland van Rijswijk-Deij, Alberto Dainotti, Kimberly C Claffy, and Anna Sperotto. When parents and children disagree: Diving into dns delegation inconsistency. In *International Conference on Passive and Active Network Measurement*, pages 175–189. Springer, 2020.

[16] Nikhil Tripathi, Mayank Swarnkar, and Neminath Hubballi. Dns spoofing in local networks made easy. In *2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–6. IEEE, 2017.

[17] vixie. Dve-2018-0004: fiservapps.com front end load balancer is returning nxdomain for qtype=ns, 2018. [Web; accedido el 18-08-2022].

ANEXOS

Anexo A

Tiempos de respuesta

A continuación se listan los diferentes promedios de tiempo de respuesta de los dominios consultados para realizar el respectivo experimento, para ambos algoritmos usados.

A.1. Primer proveedor

A.1.1. Algoritmo Actual

Nombre del dominio	Promedio tiempo de respuesta [ms]
google.com	477.2
youtube.com	477.5
facebook.com	490.3
apple.com	392.2
reddit.com	399.9
instagram.com	492.5
duckduckgo.com	505.1
twitter.com	800.0
fandom.com	789.3
linkedin.com	796.9
twitch.tv	804.1
quora.com	401.3
ebay.com	500.3
tiktok.com	398.7

Nombre del dominio	Promedio tiempo de respuesta [ms]
nytimes.com	402.1
redd.it	947.5
etsy.com	394.1
usps.com	538.4
cnn.com	396.2
netflix.com	394.0
office.com	538.9
affirm.com	394.0
discord.com	375.7
spotify.com	508.5
espn.com	395.5
foxnews.com	514.3
evergage.com	395.6
adobe.com	747.7
airbnb.com	782.4
onlyfans.com	397.4
github.com	397.5
force.com	426.4
dropbox.com	394.4
imgur.com	883.8
samsung.com	683.4
wordpress.com	433.6
patreon.com	374.1
blogspot.com	478.2
verizon.com	978.5
myshopify.com	376.4
buzzfeed.com	403.8
steamcommunity.com	1053.2
nike.com	407.8
mcdonalds.com	402.5
fedex.com	552.2
wikipedia.org	363.3
zoom.us	788.7
stackoverflow.com	374.9
chase.com	499.6
shein.com	403.8
sii.cl	319.8
paris.cl	323.6
meganoticias.cl	317.5
wom.cl	320.1
uchile.cl	319.1
uc.cl	317.5
jumbo.cl	318.7

Nombre del dominio	Promedio tiempo de respuesta [ms]
u-cursos.cl	318.6
mundo.cl	319.4
lacrianza.cl	319.0

Tabla A.1: Promedio de tiempo de respuesta con el algoritmo actual para los dominios utilizados en el experimento. Primer proveedor de internet.

A.1.2. Algoritmo Nuevo

Nombre del dominio	Promedio tiempo de respuesta [ms]
google.com	476.5
youtube.com	475.9
facebook.com	489.1
apple.com	392.6
reddit.com	398.9
instagram.com	491.6
duckduckgo.com	508.8
twitter.com	801.9
fandom.com	786.7
linkedin.com	797.0
twitch.tv	806.4
quora.com	400.8
ebay.com	499.8
tiktok.com	396.9
nytimes.com	401.2
redd.it	959.1
etsy.com	393.6
usps.com	535.9
cnn.com	396.0
netflix.com	393.1
office.com	540.3
affirm.com	393.8
discord.com	379.2
spotify.com	509.0
espn.com	395.8
foxnews.com	528.6
evergage.com	396.7
adobe.com	755.2
airbnb.com	798.2
onlyfans.com	395.6
github.com	399.8
force.com	433.2
dropbox.com	397.5
imgur.com	880.5

Nombre del dominio	Promedio tiempo de respuesta [ms]
samsung.com	690.0
wordpress.com	434.8
patreon.com	374.3
blogspot.com	478.6
verizon.com	984.6
myshopify.com	378.0
buzzfeed.com	404.4
steamcommunity.com	1055.6
nike.com	409.0
mcdonalds.com	402.7
fedex.com	551.7
wikipedia.org	363.5
zoom.us	802.3
stackoverflow.com	378.1
chase.com	505.1
shein.com	405.4
sii.cl	319.5
paris.cl	324.1
meganoticias.cl	317.7
wom.cl	320.1
uchile.cl	319.3
uc.cl	317.6
jumbo.cl	317.7
u-cursos.cl	318.5
mundo.cl	318.5
lacrianza.cl	319.3

Tabla A.2: Promedio de tiempo de respuesta con el algoritmo nuevo para los dominios utilizados en el experimento. Primer proveedor de internet.

A.2. Segundo proveedor

A.2.1. Algoritmo Actual

Nombre del dominio	Promedio tiempo de respuesta [ms]
google.com	463.2
youtube.com	462.3
facebook.com	552.8
apple.com	379.0
reddit.com	469.0
instagram.com	552.5
duckduckgo.com	640.7
twitter.com	801.2

Nombre del dominio	Promedio tiempo de respuesta [ms]
fandom.com	798.9
linkedin.com	815.8
twitch.tv	1045.0
quora.com	470.4
ebay.com	625.2
tiktok.com	469.2
nytimes.com	472.2
redd.it	1146.9
etsy.com	469.1
usps.com	488.2
cnn.com	456.9
netflix.com	460.4
office.com	480.1
affirm.com	470.9
discord.com	367.3
spotify.com	490.1
espn.com	469.5
foxnews.com	627.1
evergage.com	467.6
adobe.com	670.0
airbnb.com	819.0
onlyfans.com	470.8
github.com	470.9
force.com	588.5
dropbox.com	467.6
imgur.com	1181.4
samsung.com	707.5
wordpress.com	498.8
patreon.com	363.9
blogspot.com	461.8
verizon.com	1021.2
myshopify.com	367.4
buzzfeed.com	469.3
steamcommunity.com	1312.4
nike.com	501.6
mcdonalds.com	468.4
fedex.com	627.9
wikipedia.org	477.7
zoom.us	1058.4
stackoverflow.com	472.0
chase.com	510.5
shein.com	469.6
sii.cl	313.8

Nombre del dominio	Promedio tiempo de respuesta [ms]
paris.cl	319.3
meganoticias.cl	316.5
wom.cl	318.6
uchile.cl	317.8
uc.cl	320.3
jumbo.cl	317.0
u-cursos.cl	318.3
mundo.cl	321.9
lacrianza.cl	317.1

Tabla A.3: Promedio de tiempo de respuesta con el algoritmo actual para los dominios utilizados en el experimento. Segundo proveedor de internet.

A.2.2. Algoritmo Nuevo

Nombre del dominio	Promedio tiempo de respuesta [ms]
google.com	451.1
youtube.com	450.1
facebook.com	535.5
apple.com	366.6
reddit.com	452.1
instagram.com	537.0
duckduckgo.com	623.4
twitter.com	783.4
fandom.com	780.4
linkedin.com	796.4
twitch.tv	1013.7
quora.com	455.3
ebay.com	603.1
tiktok.com	452.6
nytimes.com	453.5
redd.it	1110.2
etsy.com	451.8
usps.com	485.8
cnn.com	452.6
netflix.com	437.1
office.com	456.9
affirm.com	451.6
discord.com	353.5
spotify.com	472.0
espn.com	453.6
foxnews.com	616.4
evergage.com	453.8
adobe.com	654.0

Nombre del dominio	Promedio tiempo de respuesta [ms]
airbnb.com	793.9
onlyfans.com	453.5
github.com	454.1
force.com	573.9
dropbox.com	452.7
imgur.com	1183.4
samsung.com	693.1
wordpress.com	483.6
patreon.com	354.3
blogspot.com	445.9
verizon.com	1008.2
myshopify.com	356.2
buzzfeed.com	453.7
steamcommunity.com	1287.4
nike.com	486.0
mcdonalds.com	451.5
fedex.com	615.8
wikipedia.org	453.4
zoom.us	1032.6
stackoverflow.com	456.7
chase.com	509.4
shein.com	453.0
sii.cl	314.6
paris.cl	319.8
meganoticias.cl	316.9
wom.cl	319.0
uchile.cl	318.2
uc.cl	320.6
jumbo.cl	317.5
u-cursos.cl	318.8
mundo.cl	323.6
lacrianza.cl	316.3

Tabla A.4: Promedio de tiempo de respuesta con el nuevo algoritmo para los dominios utilizados en el experimento. Segundo proveedor de internet.

A.3. Tercer proveedor

A.3.1. Algoritmo Actual

Nombre del dominio	Promedio tiempo de respuesta [ms]
google.com	385.8
youtube.com	405.6

Nombre del dominio	Promedio tiempo de respuesta [ms]
facebook.com	359.9
apple.com	314.3
reddit.com	325.7
instagram.com	359.3
duckduckgo.com	436.5
twitter.com	692.3
fandom.com	687.9
linkedin.com	692.5
twitch.tv	719.2
quora.com	326.9
ebay.com	437.9
tiktok.com	326.7
nytimes.com	327.4
redd.it	844.6
etsy.com	324.9
usps.com	446.2
cnn.com	325.0
netflix.com	324.3
office.com	418.8
affirm.com	325.1
discord.com	303.4
spotify.com	429.1
espn.com	325.1
foxnews.com	451.2
evergage.com	325.9
adobe.com	620.4
airbnb.com	685.3
onlyfans.com	326.0
github.com	326.5
force.com	360.3
dropbox.com	323.6
imgur.com	752.8
samsung.com	624.8
wordpress.com	351.4
patreon.com	303.5
blogspot.com	403.3
verizon.com	844.9
myshopify.com	306.2
buzzfeed.com	327.7
steamcommunity.com	934.3
nike.com	361.0
mcdonalds.com	324.1
fedex.com	486.6

Nombre del dominio	Promedio tiempo de respuesta [ms]
wikipedia.org	379.3
zoom.us	673.3
stackoverflow.com	412.9
chase.com	423.1
shein.com	326.4
sii.cl	164.2
paris.cl	167.8
meganoticias.cl	167.3
wom.cl	167.9
uchile.cl	169.8
uc.cl	166.9
jumbo.cl	170.4
u-cursos.cl	168.9
mundo.cl	182.2
lacrianza.cl	168.0

Tabla A.5: Promedio de tiempo de respuesta con el algoritmo actual para los dominios utilizados en el experimento. Tercer proveedor de internet.

A.3.2. Algoritmo Nuevo

Nombre del dominio	Promedio tiempo de respuesta [ms]
google.com	392.4
youtube.com	408.8
facebook.com	361.5
apple.com	319.0
reddit.com	330.5
instagram.com	363.3
duckduckgo.com	436.7
twitter.com	704.4
fandom.com	696.0
linkedin.com	704.0
twitch.tv	719.6
quora.com	332.6
ebay.com	439.5
tiktok.com	331.7
nytimes.com	330.8
redd.it	843.2
etsy.com	328.3
usps.com	447.4
cnn.com	329.0
netflix.com	330.0
office.com	420.2

Nombre del dominio	Promedio tiempo de respuesta [ms]
affirm.com	331.0
discord.com	308.2
spotify.com	432.6
espn.com	329.6
foxnews.com	452.3
evergage.com	331.0
adobe.com	622.3
airbnb.com	698.4
onlyfans.com	331.8
github.com	330.5
force.com	368.0
dropbox.com	327.5
imgur.com	741.0
samsung.com	622.8
wordpress.com	354.3
patreon.com	312.9
blogspot.com	411.0
verizon.com	863.6
myshopify.com	312.7
buzzfeed.com	331.8
steamcommunity.com	936.1
nike.com	363.8
mcdonalds.com	328.8
fedex.com	489.4
wikipedia.org	378.7
zoom.us	681.1
stackoverflow.com	413.4
chase.com	453.6
shein.com	329.8
sii.cl	164.2
paris.cl	169.1
meganoticias.cl	167.8
wom.cl	167.6
uchile.cl	168.9
uc.cl	167.5
jumbo.cl	171.9
u-cursos.cl	170.1
mundo.cl	172.5
lacrianza.cl	170.4

Tabla A.6: Promedio de tiempo de respuesta con el nuevo algoritmo para los dominios utilizados en el experimento. Tercer proveedor de internet.

Anexo B

Dominios Afectados

Dada la gran cantidad de dominios que fueron clasificados, es imposible listarlos directamente en el trabajo, es por esto que estos se encuentran en archivos en Google Drive. Cabe destacar que esta clasificación fue hecha con información del mes de Julio-Agosto del año 2022, por lo que algunos dominios podrían haber sufrido cambios en su comportamiento.

Para acceder al listado de los dominios pertenecientes al caso 1, ingrese al siguiente enlace. Para acceder al listado de los dominios pertenecientes al caso 2, ingrese al siguiente enlace. Para acceder al listado de los dominios pertenecientes al caso 3, ingrese al siguiente enlace. Para acceder al listado de los dominios pertenecientes al caso 4, ingrese al siguiente enlace.

Anexo C

Aplicación web

Con el fin de ayudar a los dueños de dominios, se creó una aplicación web. Esta aplicación permite consultar si un dominio se verá afectado o no con el nuevo algoritmo del resolver. De esta forma, los dueños de dominios podrán consultar, y así realizar los cambios necesarios para arreglar la mal configuración de los servidores.

Esta aplicación esta construida con el lenguaje de programación Rust, y utiliza los resolvers usados en los experimentos de la investigación. Su funcionamiento es similar al analizar los dominios de la zona '.cl'. Si el dominio responde con el algoritmo actual, pero no lo hace con el nuevo algoritmo, entonces este dominio se verá afectado. Destacar que para que la aplicación web no sea utilizada como medio para realizar ataques *DoS*, es que el número de consultas en paralelo que puede realizar esta limitada por la configuración.

A continuación se muestran algunas imágenes de las vistas de la pagina web. En la imagen C.1 se puede apreciar una sección de consulta de dominios, y también un poco de información sobre la problemática. Por otro lado, en la imagen C.2 se realiza una consulta, en donde se pide al usuario que espere mientras esta se realiza. Finalmente en las imágenes C.3 y C.4 se muestran resultados a una consulta positiva y otra negativa respectivamente.

Figura C.1: Vista general sitio web

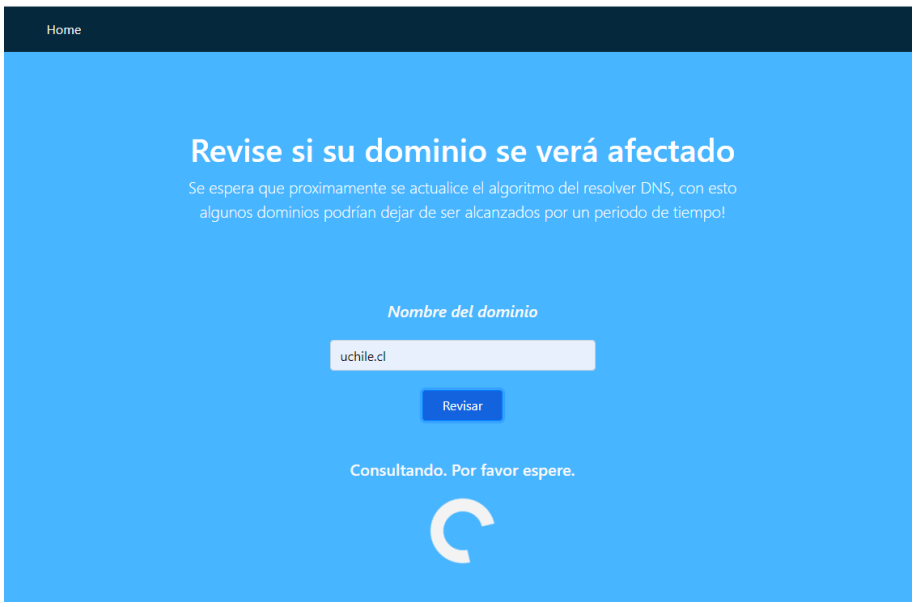

Figura C.2: Realización de una consulta

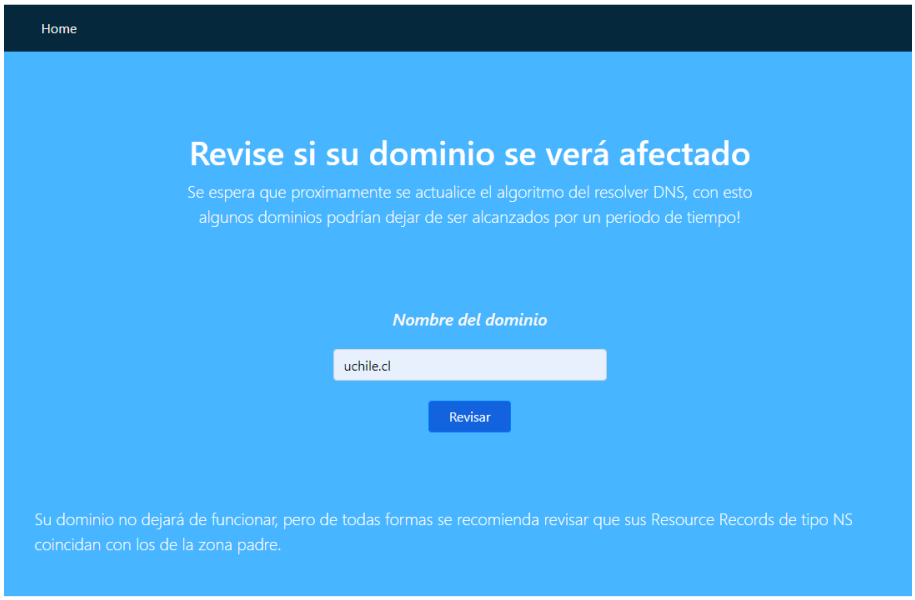

Figura C.3: Respuesta positiva para un dominio

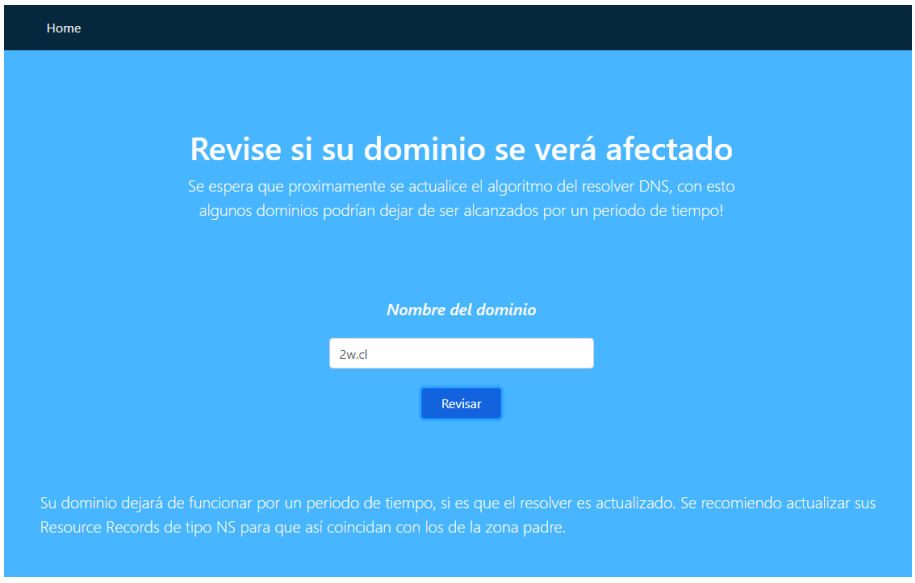

Figura C.4: Respuesta negativa para un dominio

Anexo D

Ejemplos dominios afectados

A continuación, explicaremos algunos de los dominios afectados que se encontraron durante la investigación.

Caso 3

Caso agrimed.cl

Para el siguiente caso, agrimed.cl posee la información mostrada en la figura D.1.

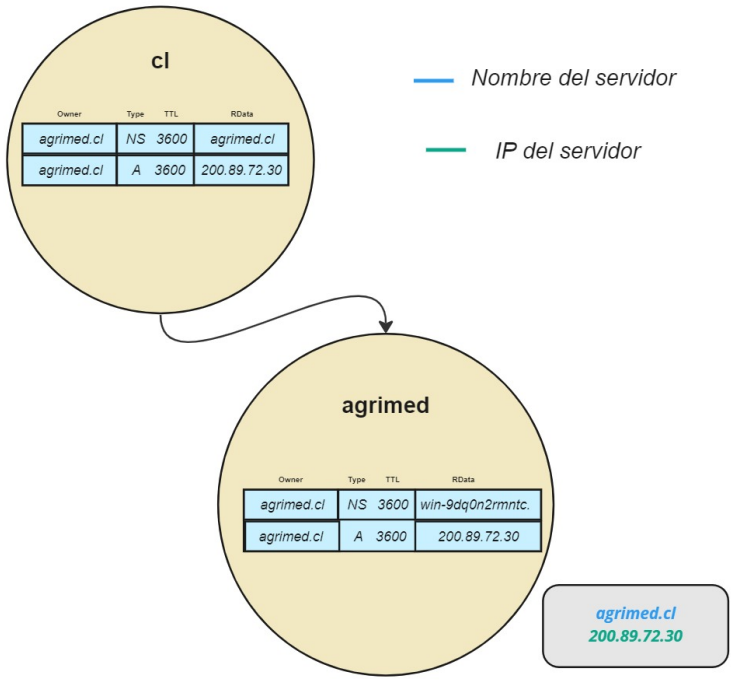

Figura D.1: Información en las zonas de agrimed.cl.

Cuando el resolver pregunte a la zona *.cl* por *agrimed.cl*, esta le devolverá la información de delegación que se muestra en D.1. Con esto, el caché quedaría en el estado de la figura D.2.

	Dominio	Tipo	Dato	Data Ranking
Caché	agrimed.cl	NS	agrimed.cl	4
	agrimed.cl	A	200.89.72.30	5

Figura D.2: Estado del caché luego de recibir la información de delegación desde .cl.

Luego el resolver consultará la información de delegación directamente a la zona de *agrimed.cl*, recibiendo la información presente en la figura D.1. Como resultado, el caché quedaría como se muestra en la figura D.3.

	Dominio	Tipo	Dato	Data Ranking
Caché	agrimed.cl	NS	win-9dq0n2rmntc.	3
	agrimed.cl	A	200.89.72.30	5

Figura D.3: Estado del caché luego de recibir la información de delegación desde agrimed.cl.

Finalmente, como se puede apreciar en la figura D.3, el registro NS de *agrimed.cl* tiene información que no corresponde a un nombre de dominio. Es por esto, que si el resolver quisiera volver a preguntar por otro dominio en aquella zona, entonces el resolver intentaría buscar la IP de un name server no existente. Tal como es de esperar, el resolver no podría conseguir esta IP, y por consecuencia, no podría resolver el nuevo dominio preguntado. Destacar que el dominio *agrimed.cl* funcionará correctamente (gracias al caché), pero no así otros dominios que sean alojados en la zona.

Caso colegiojuanmackenna.cl

La información de este dominio es mostrada en la figura D.4.

Cuando el resolver pregunte a la zona *.cl* por *colegiojuanmackenna.cl*, esta le devolverá la información de delegación que se muestra en D.4. Con esto, el caché quedaría en el estado de la figura D.5.

Luego el resolver consultará la información de delegación directamente a la zona de *colegiojuanmackenna.cl*, recibiendo la información presente en la figura D.4. Como resultado, el caché quedaría como se muestra en la figura D.6.

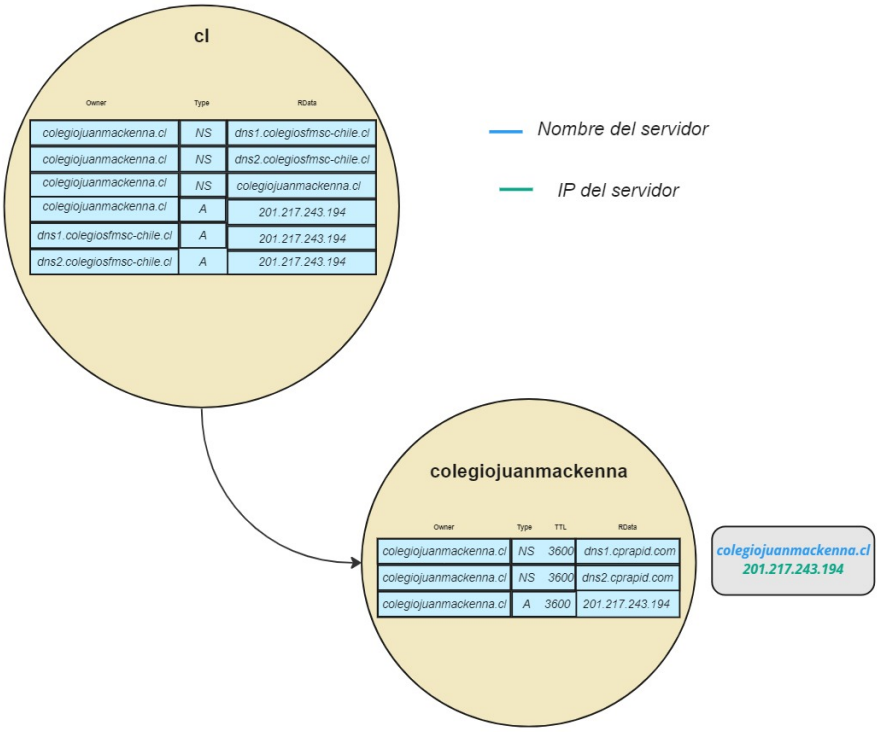

Figura D.4: Información en las zonas de colegiojuanmackenna.cl.

	Dominio	Tipo	Dato	Data Ranking
Caché	colegiojuanmackenna.cl	NS	dns1.colegiosfmsc-chile.cl	4
	colegiojuanmackenna.cl	NS	dns2.colegiosfmsc-chile.cl	4
	colegiojuanmackenna.cl	NS	colegiojuanmackenna.cl	4
	dns1.colegiosfmsc-chile.cl	A	201.217.243.194	5
	dns2.colegiosfmsc-chile.cl	A	201.217.243.194	5
	colegiojuanmackenna.cl	A	201.217.243.194	5

Figura D.5: Estado del caché luego de recibir la información de delegación desde .cl.

Finalmente, al buscar la dirección IP de los nuevos registros, se obtiene que estos dominios no existen (NXDOMAIN). Es por esto, que si el resolver quisiera volver a preguntar por otro dominio en aquella zona, entonces el resolver intentaría buscar la IP de un name server no existente. Tal como es de esperar, el resolver no podría conseguir esta IP, y por consecuencia, no podría resolver el nuevo dominio consultado. Al igual que el caso anterior, el dominio *colegiojuanmackenna.cl* funcionará correctamente, pero no así otros dominios que sean alojados en la zona.

	Dominio	Tipo	Dato	Data Ranking
Caché	colegiojuanmackenna.cl	NS	dns1.cprapid.com	3
	colegiojuanmackenna.cl	NS	dns2.cprapid.com	3
	dns1.colegiosfmsc-chile.cl	A	201.217.243.194	5
	dns2.colegiosfmsc-chile.cl	A	201.217.243.194	5
	colegiojuanmackenna.cl	A	201.217.243.194	5

Figura D.6: Estado del caché luego de recibir la información de delegación desde colegiojuanmackenna.cl.

Caso 4

Caso abogadosrancagua.cl

Para este caso, la información de las zonas corresponde a aquella mostrada en la figura D.7.

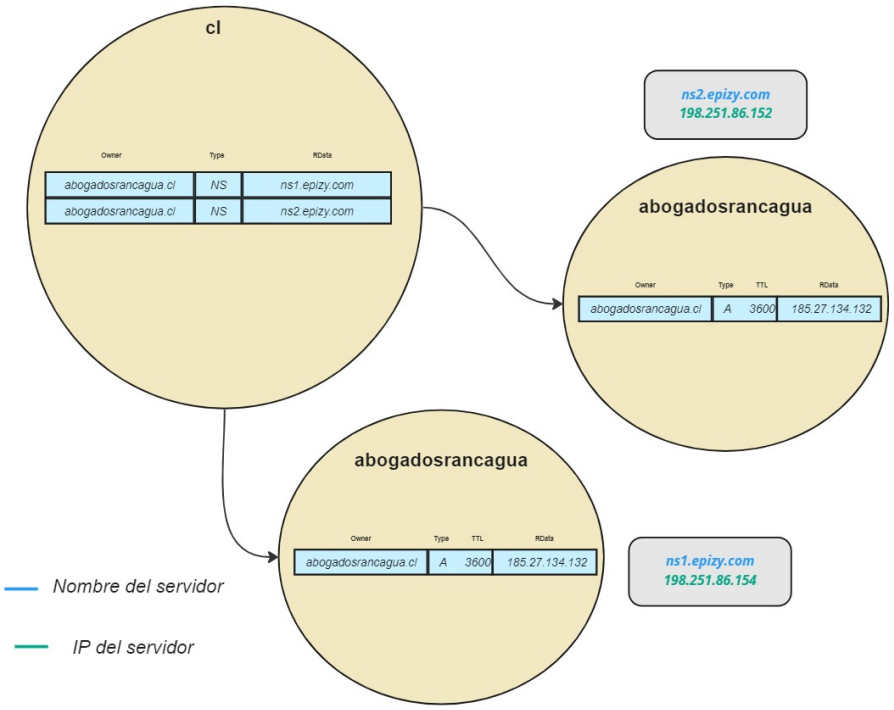

Figura D.7: Información en las zonas de abogadosrancagua.cl.

Cuando el resolver pregunte a la zona *.cl* por *abogadosrancagua.cl*, esta le devolverá la información de delegación que se muestra en D.7. Con esto, el caché quedaría en el estado de la figura D.8.

Luego el resolver consultará la información de delegación directamente a la zona de

	Dominio	Tipo	Dato	Data Ranking
Caché	abogadosrancagua.cl	NS	ns1.epizy.com	4
	abogadosrancagua.cl	NS	ns2.epizy.com	4

Figura D.8: Estado del caché luego de recibir la información de delegación desde .cl.

abogadosrancagua.cl, recibiendo la información presente en la figura D.7. Al no poseer registros de tipo NS, el caché quedaría como se muestra en la figura D.9.

	Dominio	Tipo	Dato	Data Ranking
Caché	abogadosrancagua.cl	NS	NODATA	3

Figura D.9: Estado del caché luego de recibir la información de delegación desde abogadosrancagua.cl.

Finalmente, si el resolver quisiera volver a preguntar por otro dominio en aquella zona, entonces el resolver se daría cuenta que no existen name servers para la zona. Esto no le permitirá llegar a la información, y por lo tanto no podría resolver el dominio preguntado.

Caso bgps.cl

Para el último caso, la información de las zonas corresponde a aquella mostrada en la figura D.10.

Cuando el resolver pregunte a la zona *.cl* por *bgps.cl*, esta le devolverá la información de delegación que se muestra en D.10. Con esto, el caché quedaría en el estado de la figura D.11.

Luego el resolver consultará la información de delegación directamente a la zona de *bgps.cl*, recibiendo la información presente en la figura D.10. Al no poseer registros de tipo NS, el caché quedaría como se muestra en la figura D.12.

Finalmente, si el resolver quisiera volver a preguntar por otro dominio en aquella zona, entonces el resolver se daría cuenta que no existen name servers para la zona. Esto no le permitirá llegar a la información, y por lo tanto no podría resolver el dominio consultado.

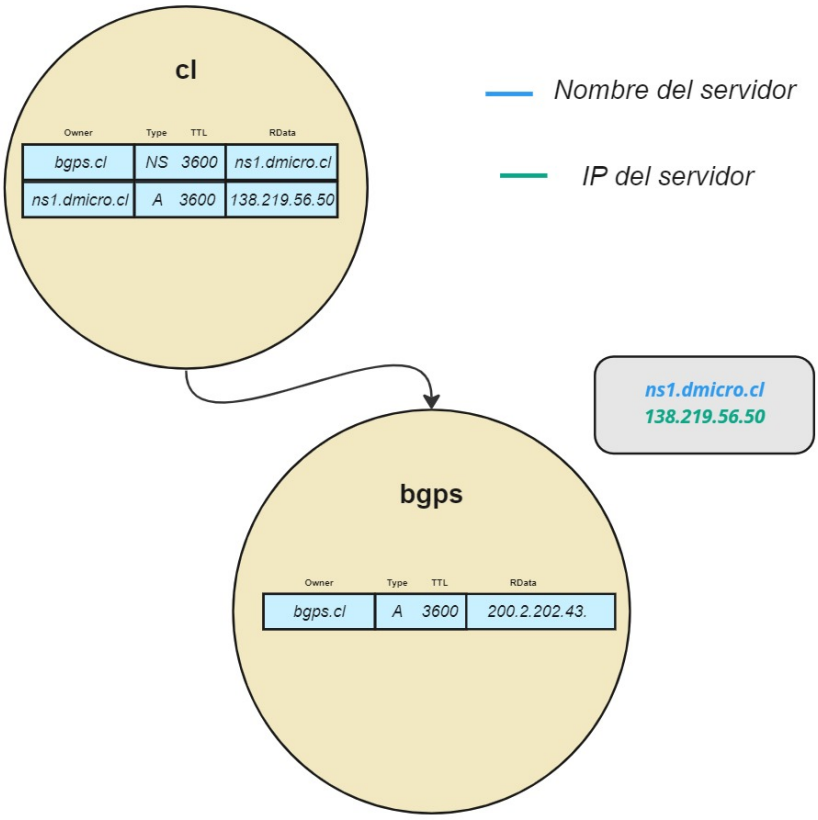

Figura D.10: Información en las zonas de bgps.cl.

	Dominio	Tipo	Dato	Data Ranking
Caché	bgps.cl	NS	ns1.dmicro.cl	4
	ns1.dmicro.cl	A	138.219.56.50	5

Figura D.11: Estado del caché luego de recibir la información de delegación desde .cl.

	Dominio	Tipo	Dato	Data Ranking
Caché	bgps.cl	NS	NODATA	3
	ns1.dmicro.cl	A	138.219.56.50	5

Figura D.12: Estado del caché luego de recibir la información de delegación desde bgps.cl.