COPENHAGUE – Cómo funciona: Fundamentos del DNS
Domingo, 12 de marzo de 2017 – 11:00 a 12:30 CET
ICANN58 | Copenhague, Dinamarca

STEVE CONTE: …pueden quedarse aquí para burlarse de mí o pueden participar de otra sesión. Gracias. Vamos a comenzar. Gracias a todos por venir hoy. Soy Steve Conte, de ICANN. Es una clase de fundamentos del DNS. Si ustedes están en un servidor del DNS, esta es la sala incorrecta. Si nunca conocieron nada del DNS, es la sala correcta. Intentaremos hacer esta sesión lo más rápido posible entre hoy y mañana. Vamos a ver distintas partes de la tecnología que están cubiertas en el sistema de la ICANN. Vamos a tratar los fundamentos del DNS en esta sesión. En otra, trataremos el networking en Internet con Alain Durand, también del equipo y Jeff Houston de APNIC. Tenemos una sesión sobre el uso indebido del DNS, a cargo de John Crain. El uso indebido del DNS es un tema que nos permitirá entender mejor cómo luchar contra este uso indebido. Por último tenemos el comité asesor de servidores raíz, el RSSAC hoy a la tarde, que hablará del sistema de servidores raíz. Es una sesión siempre interesante. Les invito a asistir.

Si no pueden participar en las sesiones de hoy y les resulta un tema de interés, haremos mañana las mismas sesiones. Verán

que mañana están las mismas sesiones. Todo lo que decimos hoy lo diremos mañana. Tienen dos días para asistir.

Pasemos entonces al contenido de la sesión de fundamentos del DNS. El concepto central del DNS es que los números son difíciles de recordar. Ya es suficiente poder recordar dos o tres números de teléfono pero la guía telefónica completa, intentar recordarlos es muy difícil. Lo mismo sucede con las direcciones IP, con miles de millones de direcciones de IP. Además, hay trillones de direcciones en el espacio. Es muy difícil recordar así los números de direcciones IP de servidores específicos. El DNS, el sistema de nombres de dominio, se desarrolló para asociar palabras humanas a estas direcciones numéricas del IP. Usamos nombres porque los recordamos más fácilmente que los números.

Al comienzo, había solo un puñado de servidores en la Internet. Era muy fácil entonces recordarlos. Eran de etiqueta única. No había un punto después. Era un nombre y nada más el que hacía referencia a los nombres de hosts. El concepto concreto del mapeo de los nombres con las direcciones IP se llama resolución de nombre. La resolución de un nombre al comienzo iba a un archivo host. Cada computadora conectada a Internet tenía un archivo host. Este archivo tenía un directorio de nombres con direcciones IP de los otros servidores de la Internet. Era fácil de

localizar, era fácil de buscar, porque eran solo un puñado de servidores. Era fácil de manejar.

Hay un archivo host también hoy día. Sigue estando en el centro del sistema. Ha cambiado un poco el sistema. Se ha desarrollado más. Ahora actúa más como un puntero que como un directorio. Era relativamente fácil de mantener por parte del NIC, que es el Centro de Información de la Red. Básicamente, si se tenía una máquina, se incorporaba a la Internet y se presentaba esa información al SRI, al Stanford Research Institute. El resto lo hacía el centro. El administrador enviaba las actualizaciones por correo electrónico, se hacía una actualización una vez por semana y había un protocolo de transferencia que era el FTP. Esto funcionó durante un tiempo pero la Internet creció y surgieron problemas. Primero el conflicto de nombres. Al comienzo, si uno quería un nombre, lo solicitaba. Yo mi sitio lo voy a llamar Steve y lo conseguía, pero a medida que surgieron nombres hubo varias personas que querían el mismo nombre. Surgieron conflictos de resolución. No había buenos métodos de prevención de duplicación. El concepto del DNS y de la Internet, ahí es donde surge el concepto de identificadores unívocos.

También había problemas de sincronización, porque había que optar activamente para descargar el último archivo actualizado. Como cada uno lo hacía a su tiempo, no había sincronización del archivo. Había posibilidades, entonces, de que los archivos

fueran diferentes en las distintas máquinas clientes. No había sincronización de los datos. Eso significaba que algunas personas podían no haber descargado los archivos. Tenían información antigua. Si los servidores cambiaban o los servidores no tenían el último archivo, no tenían los datos completos. El tráfico y la carga, ahora estamos en una era en que con la fibra óptica y demás tenemos mucho ancho de banda pero en esa época teníamos módem por discado y había 200 BPS de velocidad. Era fantástico tener la Internet pero era muy difícil, muy dificultosa la transmisión.

Había un archivo host que se mantenía centralmente que no tenía o no se escalaba en un único lugar. Se comenzó a hablar, a comienzos de los 80, acerca de remplazarlo. El objetivo era resolver estas cuestiones de escalamiento del host.txt, que era el archivo y simplificar el enrutamiento de los emails. El correo electrónico era algo cada vez más presente en la Internet en los primeros días. Surgieron dificultades para manejar los mecanismos de enrutamiento. Se comenzó a discutir cómo cambiar entonces la metodología de resolución del host. Como resultado, surgió el sistema de nombres de dominio. Esto requirió muchos documentos que se hicieron en la IETF, el grupo de trabajo de ingeniería de la Internet, que produce normas abiertas y lo hace a través de una metodología que se llama RFC que es el estándar de documentación. Tenemos un par, el 799 y

el 819, que fueron las primeras normas que describían cómo era el sistema DNS.

En lugar de tener una única fuente de datos localizada centralmente que era el modelo antiguo en SRI, se dijo que tener una base de datos distribuida era mejor, más manejable, más disponible globalmente, con las distintas zonas para distribuir la resolución. Ponía la ventaja de la gestión en manos del operador de la zona. Así era más fácil hacer el mantenimiento porque requería más de un único punto de fallo. Había distintas personas que monitoreaban las zonas. Se introdujo el concepto del caching, de la caché. Era un diagrama de resolución. Vamos a hablar del caching en un momento. Había replicación que facilitaba la redundancia y la distribución de la carga de los servicios también.

Un vistazo de lo que es el DNS. Tiene distintas cosas. Vamos a comenzar por la derecha hacia la izquierda. Tenemos el servidor de nombres autorizado. Desde la perspectiva del TLD, estamos hablando del .COM, .NET, .UK, etc. El titular del nombre de dominio. Por ejemplo, yo tengo CONTE.NET. Tengo un servidor de Internet que es el servidor de mi nombre autorizado. También tenemos los recursivos que usualmente están con los ISP o en una red empresarial. La red puede tener un servidor recursivo que son los proxy. Son los servidores que actúan en nombre de las personas en la búsqueda de información. Luego tenemos los

resolutores de stub, que puede ser cualquier cosa. Puede ser una laptop, puede ser el teléfono… Cualquier dispositivo que haga una consulta al DNS a través del resolutor tiene que entender cómo hacer la consulta y también tiene que evaluar las respuestas y tiene cierto nivel de caching.

Tanto el stub como el recursivo tienen cierto nivel de caché. Gran parte del caché se da a nivel del recursivo y después hablaremos de por qué. Estos son los tipos en general de servidores que vamos a ver en la estructura del DNS. Bien. La base de datos del DNS es un árbol invertido. Si lo damos vuelta, vemos que está la raíz, que es la rama principal, de la cual se ramifican los demás niveles. Esto es así a propósito. Cada sección de este nivel se llama nodo y cada nodo es específico. Lo hacemos jerárquicamente así porque nos aleja de lo que es la distribución única. En estos nodos estamos restringidos a lo que es LDH: letras, dígitos y guión. Son los únicos caracteres ASCII permitidos. Son los únicos caracteres que pueden resolver o residir dentro del espacio del DNS. Aun cuando hoy día tenemos los IDN, los nombres de dominio internacionalizados, en lo que hace al protocolo específico del DNS es el formato LDH.

Aquí tenemos a la izquierda un ejemplo de un IDN xn-j6w193g, que es un mecanismo de traducción que Punycode e IDN utilizan. Es un lenguaje distinto del ASCII que asigna un nombre de dominio. No recuerdo cuál es en este momento pero es un

IDN. Si buscan un IDN, un nombre en árabe o en otro código de escritura que no sea ASCII, lo vemos así. Las etiquetas tienen una longitud máxima de 63 caracteres, no más de eso. Desde la perspectiva humana no es necesario tener más porque el propósito del DNS es que sea algo que pueda memorizarse. Si uno quiere memorizar algo, no tiene que tener más de 63 caracteres.

Dentro de cada nodo tenemos un nombre de dominio. Un ejemplo. En las distintas etiquetas, empezando desde arriba… Perdón, empezamos desde abajo a construir la etiqueta. Tenemos www.example.com. Hablaremos del punto oculto o del calificador al final. Cada dominio en la Internet, por ejemplo, example.com, si lo damos vuelta vemos la estructura exactamente igual. Podemos ver la estructura y cómo se relaciona con la jerarquía del árbol de dominios simplemente dándolo vuelta o poniéndolo de costado. Ahí reconocemos la estructura.

Un nombre de dominio totalmente calificado significa que identifica sin ambigüedad un nodo. En este ejemplo tenemos www.example.com. Nadie pone el punto al final pero en este ejemplo sí lo ponemos. Eso significa, para las máquinas, que está totalmente calificado. Es decir, no quiero ninguna incertidumbre acerca de qué representa esto. El sistema de nombres de dominio reconoce que ese punto es la raíz, que está totalmente

calificado, que no puede haber nada más después de lo que tipeamos aquí. La mayoría en general no son ambiguos. Por eso solemos no poner el punto. Así es como el sistema reconoce que no hay ambigüedad.

Un dominio es un nodo y todo lo que hay debajo. Por ejemplo, .COM, que es un nombre de dominio superior, todo lo que está debajo está en el vértice del dominio. La parte superior de ese nombre, de la parte superior del nodo, son punteros que gestionan el resto. Quizá no sea la gestión completa del nodo pero da indicaciones acerca de dónde está el siguiente paso o el siguiente nivel. Cuando veamos un ejemplo de resolución lo vamos a ver mejor. Si lo llevamos hacia el nivel superior, a la raíz, que es el ápice o el vértice del sistema de nombres de dominio, en un ccTLD, por ejemplo .UK, hay dominios debajo de ese. Entonces lo toma también.

Dividimos el espacio de nombres en un modelo distribuido que permite a entidades separadas manejar el archivo de zonas o los nodos. Los nodos los llamamos zonas. Esta distribución permite reducir a un único punto de gestión. También permite tener una rápida respuesta de edición. Por ejemplo, una organización .COM, si ponemos varios servidores en lugar de enviar un email al registro, ustedes son los administradores por ejemplo del .COM. Quieren hacer un cambio de la zona y eso se distribuye rápidamente en la Internet. La entidad que delega el dominio se

llama padre. A los fines de CONTE.NET, la parte delegante o delegadora en esta relación es .NET. La parte del hijo en esa relación soy yo, que manejo el nombre de dominio CONTE.NET. Siempre habrá una relación padre-hijo hasta la raíz porque la raíz es el padre y no hay ningún otro padre superior a la raíz.

En el espacio de nombres tenemos las fronteras administrativas. En el modelo distribuido, cada nivel tiene un punto de distribución diferentes respecto de quién puede gestionar los dominios o las zonas. Comenzando por arriba tenemos la zona raíz. La IANA y la PTI manejan la zona raíz en colaboración con los administradores de zona raíz. VeriSign también. Tenemos los nombres de dominio superior, los TLD, que pueden ser de país; genéricos, gTLD, y cualquier otra entidad del tipo TLD. Por debajo en este ejemplo tenemos [“example foo”] y “bar”. Estos son dominios que son administrados por el registratario de esos nombres. Por debajo tenemos otros que pueden ser un servidor, un subdominio, pero siguen siendo todos administrados dentro de la frontera administrativa de esos dominios.

Acá vemos la relación padre-hijo y la delegación. Los servidores de nombres responden consultas. Una consulta proviene de una aplicación o de otro servidor de nombre o de otro resolutor. Ya vimos distintos resolutores, servidores de nombres caché. Un servidor de nombres puede ser un resolutor y también puede ser un servidor de nombres. Un servidor de nombres autorizado

tiene información de zona correspondiente a ese dominio. Es la fuente autorizada de ese dominio. Por lo tanto, tiene información total sobre esa zona. El administrador de zona va a poner los datos allí. Puede dar una respuesta definitiva a las consultas. Las zonas tienen la mayor cantidad de servidores autorizados. Esto brinda redundancia y luego extiende además la carga de consultas. Cuantos más servidores tenemos, hasta un punto para una zona, y cuanto más distribuidos estén desde el punto de vista topográfico en la red, mejor será la distribución de la carga que pueda manejar.

En esta forma, en general tenemos un servidor de nombres primario que se llamaba servidor maestro. Tenemos servidores secundarios o esclavos. El servidor primario en general es el lugar donde se hacen los cambios a los datos en la zona raíz. Si yo agrego un dominio, si cambio una dirección de IP, si hago ese tipo de cosas, voy a hacer esos cambios y se los voy a enviar al servidor primario. Mis servidores secundarios, por otra parte, si bien replican exactamente lo que hace el primario, lo hacen a través de un proceso de consulta y respuesta. Los servidores secundarios pasan por el primario para recibir las actualizaciones y cuando los resolutores les hacen consultas, encuentran la misma información y la hacen llegar. Una vez que los datos de zona están distribuidos a través de distintos servidores autorizados, son los mismos datos. El resolutor no

sabe o no le importa cuál es el servidor de nombres primario porque todos son servidores autorizados.

La transferencia de zona es iniciada por los secundarios. Se programa, se autoriza a nivel primario, se autoriza a un servidor, se lo identifica y se dice: “Este es un servidor secundario autorizado”. A nivel de servidor secundario, apuntamos al servidor primario y todos los que buscan algo van a hacer una consulta y van a ver si hay alguna actualización. Si hay alguna actualización se va a iniciar una sincronización. De lo contrario, simplemente no va a recibir ninguna actualización y va a continuar funcionando normalmente.

El estándar DNS especifica el formato de los paquetes de DNS enviados a través de la red. El estándar también especifica una representación basada en texto para los datos DNS que se llama formato de archivo maestro. Si tenemos la capacidad de ver dentro de los servidores propiamente dichos, podemos ver los archivos de zona que son archivos de texto simple. Puede ser un par de renglones. En el caso de mi dominio, yo tengo solo un par de ingresos. Por lo tanto es muy fácil ver la información. O podría ser algo como .COM que tiene millones y millones de entradas debido a los diferentes dominios a los que apunta. Puede ser un archivo muy simple o muy complejo en términos del tamaño del archivo. Puede ser simple en cuanto al servicio que brinda. Después vamos a hablar de eso. Básicamente se refiere también

al tamaño. Debido a esto, es fácil de leer. Es fácil hacer el parsing. Se pasa a este formato durante la etapa de especificaciones, cuando hacemos los estándares para el DNS.

Cada nodo tiene un nombre de dominio y el nombre de dominio puede tener distintas clases de datos asociados. Los llamamos registros de recursos. Vamos a ver un par de registros diferentes en la próxima diapositiva, creo. Una zona puede constar de diferentes clases de registros. Podemos tener punteros que dicen: “Mi dirección de correo electrónico va a estos servidores. WWW va a este servidor. FTP va a aquel servidor”. DNSSEC tiene un registro de recursos. Hay distintas clases de RR, registros de recursos, que les permiten a las máquinas recibir las consultas de manera más específica para saber cómo redirigir y manejar ese tráfico.

Nunca se pueden mezclar los registros de recursos de distintos servidores. Se refieren a un nombre de dominio, a una zona. Si uno maneja varios dominios y tiene el mismo servidor de nombre para estos dominios puede hacerlo pero necesita tener diferentes archivos para cada uno de los dominios. De esa manera puede gestionar mejor los datos.

Clases de registros de recursos. Algunos de los más comunes son los siguientes. El titular, que muestra el nombre del dominio con el que está asociado el titular. El time t olive. Esto tiene que ver

con caché. Después vamos a hablar de caché, después de la resolución. Se puede especificar la vida del dominio. Es decir, la vida de la información que está en ese dominio. Se pueden hacer múltiples cambios en un servidor en particular y mantenemos eso en modo caché para poder propagar los nuevos cambios más rápidamente a la Internet pero si tenemos algo estable, vamos a mantenerlo de forma tal que sea largo pero no demasiado largo como para que los servidores caché no hagan consultas con demasiada frecuencia. Queremos que los datos venzan en algún momento porque en algún punto quizá tengamos un cambio y queremos asegurarnos de que los servidores de nombres y los resolutores busquen ese cambio en algún momento. Después vamos a hablar de los tiempos predeterminados y los tiempos establecidos.

Tenemos clases. Esto no se utiliza hoy en día pero hay distintas clases, distintos tipos de clases. Vemos un archivo que se llama IN Internet Class. También hay otros. Tenemos tipos de registros de recursos. Son distintas formas en las que podemos asociar los datos y tenemos registros A. tenemos distintas clases de registros. Después lo vamos a ver en mayor detalle. Esto indica lo que hay en el archivo de zona. Es el tipo. Ahí es donde se produce la mayor parte de la resolución entre nombres y números, en el tipo de registro de recurso. Después tenemos RData. Son los datos del tipo especificado que tiene el registro.

Esta es la sintaxis en general. Entre paréntesis está lo que podemos dejar afuera. Si enviamos o modificaciones un archivo de zona de dominio, lo que está entre paréntesis puede quedar afuera y va a heredar los valores predeterminados. En general, tenemos titular, tenemos TTL, tenemos clase. Tenemos tipo y después tenemos los datos propiamente dichos. Una vez más, la clase en general es una clase IN en los archivos de zona y TTL, si no lo especificamos, va a heredar lo que figura en el registro maestro de ese archivo.

Siempre es necesario poner tipo y datos porque ahí es donde se produce la resolución. Después les voy a mostrar un ejemplo pero, por ejemplo, podemos tener www como tipo y después los datos serían la dirección IP de ese servidor de nombres.

Tipos de registros de recursos comunes. Como dije, tenemos los registros A, que se llaman ancla. Estos son los que resuelven el nombre para las direcciones en IPv4. Después tenemos los registros de AAAA, que son los que tienen que ver con las direcciones de IPv6. Tenemos NS que es una lista de los nombres de servidores de nombres autorizados que están relacionados con ese dominio. Tenemos SOA, el inicio de autoridad. Este es el primer tipo de registro que van a ver dentro de los archivos de zona. Es un renglón descriptivo que nos dice varias cosas: quién es el operador, el TTL predeterminado. Cumple diversos propósitos.

Tenemos también un CNAME. Este es un alias. Si yo tengo un registro A, tengo www y apunto a 192.168.1.3 y si además quiero que esto sea un servidor FTP, en lugar de poner un registro A, puedo poner un CNAME para FTP y puedo decir CNAME FTP www y esto va a crear un alias también, y va a heredar la dirección IP de ese registro A.

Tengo registros RX que es el nombre de los servidores de intercambio de mail. En general se utilizan cuando queremos enviar un mail a un destino. Antes de ser enviado, ese email tiene que saber cuál es el destino. El servidor de email va a hacer una búsqueda a través de las distintas capas del DNS para ver cuál es el destino de ese servidor de nombres. ¿Cómo lo hace? Pide el registro MX de intercambio de mail. La respuesta frente a eso es una lista de nombres de dominio asociados con ese registro de correo electrónico para ese nombre de dominio.

Tenemos también un registro PTR. En general, en dominio típico tenemos un mapeo de nombres y números. Un registro PTR hace una búsqueda inversa. Relaciona un número con un nombre. Busca un nombre de un host enviando un comando, tipeando la dirección IP y, si está en esa zona, vamos a recibir un nombre de un host como resultado. Se utiliza algo pero en general no lo utilizan los usuarios finales.

En general hay 84 clases de registros desde agosto del año pasado. La mayoría son lo que vimos en la última diapositiva. Estos son archivos de zona que solemos encontrar. Hay muchas clases diferentes, algunos casos especiales que no siempre se utilizan. Estamos empezando a ver nuevos tipos de registros con DNSSEC pero en general tenemos quizá 20 clases de registros que son los que usan los usuarios finales o las aplicaciones comunes. La IANA tiene un registro de tipos de recursos. Acá hay una URL. Estas diapositivas van a estar adjuntas al programa. No hace falta que vayan copiando todo rápidamente. Pueden acceder a esta presentación después.

Un comentario breve. La IANA, como ustedes saben, una de sus funciones es ser el lugar en el que se almacenan y se conservan los identificadores únicos. Si bien el IETF desarrolló el DNS y mantiene los RFC, los parámetros de protocolo correspondientes y los estándares para DNS, tiene que haber un lugar donde se puedan incluir cosas como los tipos de registros de recursos. La IANA tiene muchos registros. Este es uno solo de muchísimos otros.

Este es un ejemplo de la página de la IANA que tiene los tipos de registros de recursos. Abajo pueden ver las distintas clases de las que hablamos. Hablamos de los registros A, NS, no hablamos del MD o del MENOS FAVORECIDAS. Hay 84 clases diferentes. No vamos a verlos todos en detalle. El uso más común del DNS es el

mapeo de un nombre con un número, como dijimos. Esto se hace mapeando el nombre con una dirección v4, en este caso 192.0.2.7 o con una dirección de AAAA, que es IPv6, y obtenemos el resultado que ven en pantalla. Como pueden ver acá, es la misma consulta. Podemos llegar a tener el mismo nombre del host unido a una dirección de v4 y v6 siempre y cuando tengamos un registro A y un registro de AAAA asociados a ese nombre de host.

Servidores de nombre. Es necesario tener un registro para cada zona. Tiene que estar en la zona padre. Vamos a hablar de eso cuando hablemos acerca de la resolución. Como recordarán el ejemplo que les di de .COM, la zona padre es .COM. El hijo es example.com. En el dominio .COM, en zonas subsiguientes, tenemos los registros del servidor de nombre y la información de DNSSEC que pueda necesitar. Eso es todo. Todo lo demás, la consulta, tiene que ir a esa zona para acceder a la información. Si no tiene un puntero del padre para que apunte el hijo, la consulta no va a saber adónde ir para obtener esa información. Es necesario tener registros DNS en el padre y también en la zona hijo para poder mostrar dónde están los demás servidores. Podrán ver acá que no es una dirección IP. El registro NS apunta a otros dominios, a otros nombres de dominio completos. Pueden ver ahí la dirección.

Como dije antes, el padre, en este caso la raíz, tiene registros que apuntan al hijo, en este caso .COM. En la raíz tenemos listas de creo que 13 servidores de gTLD. Tenemos 13 servidores de nombres autorizados asociados a .COM. La raíz apunta a la consulta a que mira a esos 13. Acá está el problema del huevo y la gallina, porque si uno nunca hizo una resolución de nombres y apunta a un nombre de dominio, y nunca hizo una resolución de nombre, al apuntar al nombre de dominio cómo sabemos cómo llegar. También es necesario tener algo que se llama glue (pegamento). Esto asume que todavía no tenemos un nombre de dominio específico. Por tanto, no puede ir a buscar los datos de mapeo correspondientes. Dice: "Es necesario ir a estos servidores de nombres pero como es probable que no sepa cómo acceder, voy a poner este pegamento que son los registros A, los registros ancla en el padre, y ese pegamento nos va a decir cuál es la dirección IP de esos nombres que acabo de mencionar".

Acá vemos que example.com va a ns.example.com. Va girando en círculos en torno a sí mismo de forma tal que no va a poder llegar a la información de example.com. En este caso agregamos este pegamento y hacemos ns1.example.com y ns2.example.com. Esto permite que tenga lugar la resolución y se pueda hacer la consulta sin que necesariamente antes hayamos pasado por ese servidor gracias a este pegamento. El

pegamento puede ser registro A o registro AAAA. Si tenemos servidores de nombres en el espacio de IPv6, es necesario tener pegamento también en el espacio de IPv6. No es necesario tener un servidor de DNS en este espacio para tener un registro de AAAA pero es útil. Si pensamos que la gente va a utilizar v6, si tenemos servidores v6, probablemente deberíamos tener por lo menos una máquina con v6. Por lo tanto, tenemos ahí un registro de AAAA y también un NS.

Hablamos entonces acerca del inicio de la autoridad: SOA. Dijimos que puede ir mostrando cómo es el caso para el resto de los dominios. Tiene todos los datos de la información heredada. Acá tenemos example.com. Vemos el tipo del registro de recurso. Acá vemos el servidor de nombres primario y luego dice hostmaster.example.com. Esta es una dirección de email pero esta arroba (@) significa distintas cosas en distintos lugares. No podemos tener ese @ acá porque significa distintas cosas. Acá ponemos el punto. hostmaster.example.com es en realidad hostmaster@example.com para que pueda llegar a quien maneja ese dominio. Lo ideal es que de esta forma llega al administrador de la zona. Esto no siempre ocurre así hoy en día pero así es como fue diseñado y desarrollado.

Tenemos un número de serie que permite al administrador de zona entender cuándo fue la última vez que se hizo el cambio. En general se hace en el formato de fecha secuencial. En este

ejemplo tenemos 2016, día 1 de mayo. En el formato de fecha que se use, puede ser el 5 de enero también, dependiendo del formato de fecha. El 00 es un número de secuencia. Si se hacen múltiples cambios en el mismo día, el número de serie tiene que ser el más alto, el que refleje el último cambio. Aparece un número de secuencia nuevo. Esa es la fecha incremental. Se va a incrementando por uno.

Cuando entran los servidores secundarios comparan los números de series y el secundario dice: "Este es mi número de serie. ¿Es menor que el número de serie primario?". Si este es mayor, entonces dirá: "Sí, está desactualizada la fecha. Hay que actualizarla". Hace la consulta entonces para hacer la sincronización entre los distintos servidores. Luego hay distintos valores refresh, retry, expire y mínimum. Renovar, reintentar, expirar y mínimo. Son distintas partes. Refresh parece bastante bajo. A ver, lo que están aquí de DNS, ¿cuál es el peor caché por omisión que el número podría tener? Si hay un TTL por omisión, se configura acá, ¿no?

EDWARD LEWIS: El TTL por omisión para datos se estipula en el archivo de zona. El SOA no. esto se confundió. El último número es el valor por omisión para respuestas negativas. Si digo no, es no durante

cinco minutos en este ejemplo pero hay un valor por defecto para una respuesta positiva.

STEVE CONTE: Gracias. Hay un registro SOA por zona que está al comienzo del archivo de la zona. Ahora veamos el tipo CNAME. Hablamos de registro A y de registro AAAA. El CNAME es un nombre canónico. Es un ejemplo similar al registro A. el servidor de intercambio de mail cumple otra función también dentro de la red. Por eso tenemos que darle un nombre de dominio distinto. No le podemos dar otro registro A porque hay un único registro A. queremos asociarlo con un nombre distinto. Lo tenemos que asociar con el CNAME, el nombre canónico. Ponemos primero el registro A, por ejemplo el mail.com. No, al revés, perdón. El [sum] host es el target, el objetivo, el destino. El nombre del mail es el que pusimos en el registro A y el target, el objetivo, es el nombre nuevo. Genera un alias y no hace alias en el otro lado del registro. Pone un alias después de un alias, después de un alias. Hay siempre un único registro asociado y siempre hace alias en relación con ese registro. Aun cuando haya múltiples alias, habrá múltiples líneas, siempre apuntarán al registro A primario, al nombre del servidor primario.

Cuando enviamos un mail, vamos a hablar de mail ahora, los servidores de mail tienen que determinar cómo llegar al servidor

de mails. No solo al nombre de dominio sino al servidor de mails que va a servir a ese servidor de correo. Al principio se hacían búsquedas de direcciones porque había pocas máquinas pero eso no daba flexibilidad. Si queríamos cambiar el servidor de correo o tener más de un servidor, era difícil si no imposible hacerlo. El DNS ofreció más flexibilidad agregando el tipo de registro MX.

¿Para qué sirve? Se puede especificar un servidor de correo dentro del dominio y una preferencia también. En esa diapositiva vemos que hay dos servidores de correo asociados a example.com. Tenemos un registro MX con un valor de 10 que va a mail.example.com y un registro MX con un valor de 20 que va a mailbackup.example.com. Esto permite tener múltiples servidores de correo. Uno lo define con un número de preferencia inferior. En este caso, el valor 10 es el servidor primario. Ahí es donde realmente queremos que vayan los correos pero, si por algún motivo este no está disponible, el servidor puede haberse caído por distintos motivos o por mantenimiento, tenemos el otro. Le establecemos una preferencia de 20 que significa que si no se puede ir al primero, que vaya este en segundo lugar y luego los dos servidores se comunican y se sincronizan.

Todo lo que está a la izquierda es el usuario. A la derecha es el dominio. El mapeo inverso, como decía, no se usa tanto para el

usuario final pero los administradores de redes sí lo usan y tienen aplicaciones diferentes. Hay un archivo de zona que se llama in-adder.arpa que se establece como administrador de zona y se le ponen cosas como estas. Esta entrada, 7.2.0.192.in-adder.arpa. Es un PTR a example.com. Lo que hace es referir directamente a la dirección IP pero en orden inverso. 192.0.2.7, que es la dirección IP asociada a example.com. Si lo hacemos al revés, así es como está añadido al registro PTR. Cada vez que se busca algo, al igual que cualquier otro dominio, in-adder.arpa tiene un delegador. In-adder.arpa es administrado por la IANA. Coadministran partes con los RIR. Ciertas zonas las manejan los RIR que delegan partes del espacio in-adder.arpa a los clientes. Luego los clientes administran esa última parte. Como cualquier otra administración de zona, es para búsquedas inversas. No vamos a entrar en más detalles. Solo sepan que esto existe. Hay uno también para la versión 6 que se llama IP6.arpa que se administra del mismo modo.

La seguridad del DNS. Habrá otra sesión más específica sobre el DNSSEC que les sugiero, si les interesa, que asistan porque ahora solo veremos un resumen de lo que es el DNSSEC. Es como un acertijo. Cuando uno piensa en seguridad, en general uno piensa en encriptado, cosas así, pero DNSSEC no es encriptado de los datos. Es más un proceso de autenticación de los datos, de la fuente y del destino de los datos. No hace ningún tipo de

encriptado. Solo se asegura de que cuando se hace una consulta, la respuesta que se obtiene proviene de la fuente que se estaba buscando efectivamente. Hace pares o distintos niveles de pares de claves o cadenas de claves.

Dentro del DNSSEC hay más tipos de registros. Tenemos DNS key, que es la clave pública de una zona. En lo que es la administración de claves en general tenemos la clave pública y la clave privada. Ustedes, como administradores de un dominio, tendrán la clave privada y publicarán la clave pública. A través de distintos algoritmos y métodos, cuando se hace la resolución, se compara la clave pública y la clave privada para comprobar que lo que está pasando es adonde queremos ir. Como decía, hay una sesión más específica sobre el DNSSEC, no sé si es el martes o el miércoles. Ahí habrá más profundidad sobre cómo funcionan los algoritmos y cómo se hace la comparación de claves si les interesa.

Volviendo a los tipos, tenemos luego el RRSIG que es la firma digital de un conjunto de registros de recursos. Luego NSEC y NSEC3 que son punteros que hacen la denegación de existencia autenticada. En una búsqueda esto indica si un dominio no existe. Si buscas icann58.com, queremos tener una respuesta autorizada tanto de si existe como de si no existe. Previene el spoofing o comportamientos maliciosos. Si no tenemos respuesta, si no hay ninguna respuesta, no tenemos certidumbre

de que un dominio no existe. Esto es una respuesta autorizada que dice que esto no existe. Dice al servidor: “Ni siquiera lo busque. No lo busque más”.

El registro DS es el firmante de delegación que reside en la zona padre que es parte de la cadena de confianza, de la clave. El registro DS está en la zona superior. Por ejemplo, en el example.com tendrá la respuesta desde .COM para empezar a construir desde allí la cadena de confianza. Cuando veamos los algoritmos se determina el modelo de autenticación y esto también es un factor.

Otros tipos de registros de recursos. Tenemos el TXT, URI, TLSA y otros que no vamos a ver. Son casos especiales que se usan con poca frecuencia. Si el usuario los utiliza, los utiliza con propósitos específicos. El TLSA por ejemplo lo usa DANE, que hace la autenticación por DNSSEC de entidades nombradas que son certificados. Este es un archivo que seguramente desde atrás lo pueden ver claramente. No hay mucho que decir aquí más que es un ejemplo de texto de una zona. Es uno muy básico que corresponde a example.com. Arriba vemos el registro SOA y bajando tenemos los registros NS que tenemos que tenerlos, que no pueden ser direcciones IP. Tenemos registros NS relacionados con los nombres de dominio completos. Después tenemos el registro A con una dirección IP después. Luego el registro AAAA, un par de registros MX. Tenemos un alias, un nombre canónico

CNAME y luego tenemos el glue, el pegado, el registro NS con un ancla a una dirección IP.

Este es un formato típico de cómo se ve un nombre de dominio de un registratario. No es nada muy complicado. Es solo cuestión de asociar un par de nombres de host al nombre de dominio: www, podemos tener un FTP o los nombres de host que queramos. Cuanto más alto, cuanto más profundo avancemos en el archivo más extenso será y más específicos serán los tipos de datos.

Ahora vamos a ver el proceso de la resolución. Antes de entrar en el tema quiero preguntarles si tienen ustedes alguna consulta o pregunta sobre estas cosas tan aburridas o quieren pasar a la resolución. ¿Alguien tiene alguna pregunta? Los puse a dormir a todos. Aquí tenemos una pregunta. Es necesario usar el micrófono para los participantes remotos.

ORADOR DESCONOCIDO: Me preguntaba si el archivo de la zona que usted mostró, mostró múltiples zonas. ¿Hay un archivo parecido para cada zona cuando hay múltiples zonas?

STEVE CONTE: En el archivo de zona, ¿podemos mostrarlo, Cathy?

ORADOR DESCONOCIDO: Tenemos la zona raíz, como la zona .COM. ¿Cómo funciona?

STEVE CONTE: Este es el archivo de zona específicamente para .COM. Habrá un archivo de zona para .COM, para example.com, y si hay un subdominio dentro de example.com, no se me ocurre, go.example.com, por ejemplo, habrá la delegación y la administración de otra entidad. Se puede delegar a este nivel y apuntar los registros NS al subdominio y seguir bajando por la cadena. Cathy, ¿hay alguna pregunta en línea?

CATHY PETERSEN: Tenemos una pregunta de Jared. Él quiere saber si vamos a hablar de cuestiones de propiedad intelectual.

STEVE CONTE: No. no lo haremos. Esto es puramente técnico para discutir el proceso de resolución del DNS. Vamos a hablar ahora de la resolución. En el modelo anterior teníamos los resolutores stub, los resolutores recursivos y los servidores de nombres autorizados. Antes de entrar a la resolución diré que hay dos tipos de consultas. Las consultas recursivas primero. Por ejemplo, necesito la respuesta completa o necesito un error. Los

servidores de nombres recursivos envían consultas no recursivas o iterativas. Básicamente, puedo ser parte de la búsqueda y hago una derivación. Es decir, esta es una parte de la resolución. La otra dice: “Usted tiene que ir a buscar a otra parte”. Hace una derivación.

El algoritmo de alto nivel, etc. es una correspondencia exacta con los datos locales si es posible. Si no hay respuesta posible, recorre el espacio de nombres y busca datos locales. Si es una consulta recursiva, enviar la consulta al servidor de nombres de la zona más próxima. Recordamos que teníamos los puntos cerrando la zona y mantiene las derivaciones bajando por el árbol hasta que encuentre la zona con la respuesta.

¿Cómo se inicia el proceso de resolución si no hay datos locales? Si es la primera vez, o si se acaba de construir y no hay caché, ¿cómo sabe adónde ir? Hay archivos de indicación en la zona raíz que son punteros a los servidores raíz. Hay una entrada solamente a este archivo de indicaciones. Si no sabe cómo llegar a la raíz, básicamente el pegado. Muestra el nombre del servidor raíz seguido de la dirección IP: a.rootservers.net con un registro A y la dirección IP. Eso lo tiene para las 13 instancias de archivos, incluido el espacio v6 también. Hay una muestra de un archivo hints que viene con cada compilación. El servidor no cambia con mucha frecuencia este archivo.

Tenemos la administración de la zona raíz. La sigla es RZA: root zone administration (administración de zona raíz), que es una función dividida en este momento entre la PTI que es el operador de las funciones de la IANA y VeriSign, que es el mantenedor de la zona raíz. Estos dos en conjunto crean y mantienen la zona raíz. Luego hay 12 organizaciones que operan los archivos de la zona raíz. Tenemos a los operadores RSSAC a las 5:00 de la tarde. En esa sesión se hablará más del rol del operador y cómo interactúa con el mantenedor.

Si bien hay 13 instancias de servidores raíz, hay 12 operadores de servidores. Esto se debe a que VeriSign opera la raíz A y la J. No hay diferencia entre ninguna de ellas. Uno de los mitos era que la raíz A era más autorizada que los demás servidores raíz y no es así. Todos tienen los mismos datos. De hecho, ninguno de ellos es el servidor primario. Todos estos se consideran servidores secundarios. Utilizan la metodología que se llama el hidden master, lo cual significa que hay un servidor que solamente se va a comunicar con las direcciones IP o con las instancias de estos servidores raíz. Todos actúan como secundarios. Todos van a buscar los datos para tener el último número de serie, para tener la última sincronización. El hidden master solo va a hablar con los servidores raíz. Si hay una violación de datos en cualquiera de los servidores, el archivo de zona autorizado no está en la Internet pública. Está protegido entonces. Si se viola alguna de

estas máquinas, no se va a violar todo el sistema de nombres de dominio.

Voy a hablar de la raíz L. si alguien viola la raíz L y dice que hay un TLD que se llama “conte”… No existe todavía pero quién sabe. Si publica datos para .CONTE, solo el servidor de la raíz L va a publicar esos datos porque no es autorizado, no son los datos de zona primaria hasta que no tenga un nuevo número de serie y hasta que el sistema no indique que no debería haber un .CONTE acá, pero cuando hace la próxima búsqueda, va a haber que es un número de serie antiguo y va a ir al archivo maestro para buscar los datos actualizados. Los servidores raíz tienen su propio sitio web: root-servers.org. Este mapa muestra lo que yo llama instancias de servidores raíz. RSSAC lo va a explicar en mayor detalle. Hay más de 13 servidores raíz. Antes era así pero ahora hay una tecnología que se llama Anycast de la que no vamos a hablar en esta sesión que permite básicamente espejar la instancia del servidor raíz utilizando exactamente la misma dirección IP.

Antes teníamos 13 servidores raíz, 13 máquinas en Internet que estaban a cargo de la zona raíz. Ahora tenemos cientos de instancias de servidores raíz que tienen esos datos en todo el mundo. Esto es fantástico porque cumple varios propósitos. Equilibra la carga. En lugar de tener 13 servidores que aceptan la carga de consultas de todo Internet a nivel de la raíz, ahora

tenemos cientos en todo el mundo. Equilibra la carga porque estas instancias tienen diversidad global. Hay puntos de intercambio en todo Internet a través de distintas partes del mundo. También cumplen varios propósitos y es algo muy robusto que además protege frente a ataques de servicio. Debido a la forma en que funciona la tecnología de Anycast es mucho más resiliente, mucho más elástico en cuanto a la forma en que maneja distintas clases de ataques de denegación de servicio.

Es interesante de ver. Pueden entrar en root-servers.org. Van a ver puntos amarillos y verdes. Indican la cantidad de instancias que hay en ese lugar. Vamos a ver el proceso de cambio de la zona raíz. En este caso, el administrador de TLD puede ser cualquiera. Puede ser .COM, .UK. Es un administrador de dominio de alto nivel que puede hacer un cambio en sus datos de servidores de nombres. Para eso es necesario un proceso. Envían la solicitud de cambio a través del operador de las funciones de la IANA. La IANA pasa por un proceso para asegurar que la solicitud provenga de una fuente autorizada de esa zona. Yo no puedo entrar y hacer un cambio en .DK. Por ejemplo, pasan por distintos pasos para verificar que yo realmente tengo la autoridad necesaria para solicitar ese cambio. Pasan por ese proceso utilizando distintos métodos externos e internos.

Una vez que tuvo lugar ese proceso, se solicita la implementación y va al mantenedor de la zona raíz, VeriSign. VeriSign, el mantenedor de la zona raíz, pone esa información en una base de datos. Esa base de datos luego genera un archivo de zona raíz. Ese archivo de zona raíz luego va al hidden master al que se le dan distintos nombres hoy en día. Se le puede llamar también servidor de distribución de raíz. De ese maestro, como dije, los secundarios buscan actualizaciones en el primario creo que cada 24 horas pero podría ser cada 12 horas. Los servidores raíz buscan en el servidor primario si hay nuevas actualizaciones y lo hacen comparando los números de serie. Cuando la base de datos del archivo raíz genera un archivo de zona raíz, aumenta ese número de serie, lo incrementa asegurándose de que sea un número más alto que el que está publicado en Internet.

Veamos ahora cómo es el proceso de resolución. En esta situación se configura un teléfono para enviar consultas a un servidor de nombres recursivos con la dirección 4.2.2.2. Digamos que este teléfono utiliza los servicios de Verizon y que quiere hacer una consulta. Los servicios de Verizon van a operar un servidor de nombres recursivos. Van a actuar a través del ISP, en este caso en el teléfono. En este caso, hace la consulta y quiere llegar a www.example.com pero nunca estuvo ahí. No tiene en la caché. No sabe cómo llegar. Tiene que hacer la pregunta. Tiene un archivo de indicios o de indicaciones. Perdón, no. en su

configuración IP tiene punteros que apuntan a sus servidores DNS. Si alguna vez se fijaron en su laptop, si abrieron la configuración de red van a ver una dirección IP, van a ver el master y van a ver otras direcciones IP de servidores de nombres. Así es como sabe cómo llegar al servidor de nombres recursivos para plantear la pregunta.

Entonces va al servidor de nombres recursivos y dice: “No sé cuál es la dirección www.example.com. ¿Usted lo sabe?” En este ejemplo vamos a asumir que es un servidor de nombres nuevo que nunca antes hizo ninguna búsqueda en Internet. No tiene nada en la caché. No tiene más datos que los tiene en su archivo de hints o de indicios. En este caso dice: “No. No sé cómo llegar pero sé cómo llegar al servidor raíz así que preguntémosle al servidor raíz cómo llegar a www.example.com. Yo soy un servidor de nombres recursivos. Voy a actuar en su representación”. Entonces va al servidor raíz, en este caso a la raíz L, y dice: “¿Cuál es la dirección IP para www.example.com?” y el servidor raíz dice: “No sé pero sé cómo llegar a los servidores de .COM. Tengo registros DNS para los servidores .COM así que preguntémosles a ellos”. Pasa la dirección IP de los servidores .COM al recursivo que actúa en su nombre. El servidor de nombres recursivos dice: “Fantástico. Le voy a preguntar a los servidores .COM”. Entonces va: “Servidores .COM, ¿cuál es la dirección IP para www.example.com?” y el servidor de TLD .COM

dice: “No sé pero sé cómo llegar a example.com, a ese servidor de nombres. ¿Por qué no les preguntan a ellos? Esta es la dirección IP para los servidores de nombre de example.com”.

El servidor de nombres recursivos dice: “Fantástico. Voy a tomar esa derivación y voy a ir a los servidores de nombre example.com” y les dice: “Servidores example.com, estoy buscando la dirección www.example.com” y el servidor de nombre example.com dice: “Yo lo conozco, porque yo soy la fuente autorizada de esa información. Le voy a dar la dirección IP correspondiente”. Acá tenemos la dirección IP de todas las direcciones IP asociadas a www.example.com. Puede ser que también tenga algunas direcciones de AAAA. Luego vuelve al servidor recursivo. El servidor recursivo dice: “Muy bien. Se lo voy a pasar a mi usuario final, que es la dirección IP de www.example.com”. Así finalmente llega al stub resolver y a la aplicación que se comunica directamente con ese servidor utilizando una dirección IP.

Es un recorrido bastante largo para un proceso de pregunta y respuesta. Tiene que pasar uno, dos, tres, cuatro, cinco… Por lo menos cinco servidores diferentes para obtener la respuesta. Lo bueno de Internet es que esto puede haber llevado quizá 200 milisegundos. No lleva demasiado tiempo obtener la respuesta y recorrer este proceso pero una de las cosas que ocurre es que tenemos el caching. Caching recuerda algunas de las respuestas

principales a preguntas que ya se hicieron antes. Si vamos a example.com, quizá después buscamos footwear o nike.com, no es necesario ir a pedirles a los servidores raíz cuál es example.com porque ya lo tiene en la memoria caché. Entonces ya conoce esa respuesta. El caché saca pasos del proceso o de lo contrario, en este ejemplo que vimos recién, somos un usuario de un teléfono celular. Dijimos que estaban en un servicio telefónico de Verizon. Tenemos otro usuario de Verizon que también quiere llegar a www.example.com.

El servidor de nombres recursivos mantiene esa información durante un determinado periodo en la caché. Si alguien más quiere llegar a example.com sin el TTL, le va a dar la respuesta directamente. No va a hacer ninguna otra consulta adicional porque dice: “Estos datos son autorizados. Yo los estoy conservando hasta que venza el TTL. Cuando venza el TTL voy a volver a hacer la pregunta”. Si nos fijamos en el proceso de caché, ya estuvimos en www.example.com. Ahora vamos a ir a ftp.example.com. Podemos hacer esto en nuestro navegador. El navegador pasa por un stub resolver, envía los datos a la instancia de nombres y dice: “Estoy tratando de llegar a ftp.example.com”.

Los servidores de nombres recursivos ya estuvieron en la raíz, ya estuvieron en .COM, ya estuvieron en example.com. Estos servidores ya tienen la respuesta en la memoria caché. Lo único

que necesita hacer es llegar a example.com, el servidor de nombres, y decir: "¿Se acuerda de mí? Antes pregunté acerca de www pero ahora quiero saber sobre ftp.example.com"- le va a dar la dirección correspondiente. Se la va a enviar al servidor de nombres recursivos y de ahí va a ir al stub resolver. De ahí va a ir a la aplicación, a Safari. A su vez, ahí se usa esa información para comunicarse con el servidor pero no fue necesario pasar por todos esos otros stubs. No fue necesario hablar con el servidor raíz, con el servidor de TLD. Esto ahorra tiempo al usuario. Ahorra ancho de banda para el operador de TLD o el operador de la raíz. No parece mucho para una consulta. Estamos hablando de bytes pero estamos hablando de Internet, donde las consultas ya no son solamente consultas hechas por seres humanos sino también por máquinas. Son millones y millones y millones por segundo. A nivel de la raíz quizá estemos hablando de cientos de millones o miles de millones de consultas a nivel TLD por segundo.

Cuanto menos sea necesario ir a los servidores para hacer consultas mejor va a ser para los usuarios finales y también para los administradores de la zona. ¿Hay alguna pregunta acerca de la resolución antes de continuar?

ORADOR DESCONOCIDO: Hola. Si yo compro un nuevo dominio, myname.com, ¿va a ir al gTLD y va a agregar para ese dominio, www.myname.com, va a ir a un servidor de nombres a través de la empresa a la que yo le compré el dominio?

STEVE CONTE: Su pregunta es si usted compra un nuevo dominio, ¿qué hace el registro? Muy bien. Este es un buen ejemplo del punto en el que estamos. Usted, como comprador de un nuevo dominio, es el registratario. El registrador es la empresa a la que usted le compró el dominio, a través de la cual compró el dominio. Puede ser GoDaddy, cualquiera. Exactamente. El registro es el dominio de alto nivel en ese espacio. Si hablamos de myname.com, el registro sería .COM, que sería VeriSign en este caso. Para que Internet pueda ver sus datos de zona, en primer lugar es necesario haber configurado, administrado un servidor de nombre autorizado para que aloje myname.com. Es necesario apuntar. Es necesario crear punteros dentro de .COM que apunten a myname.com. Eso se hace en general a través de un registrador. Una vez que tenemos el nombre de dominio, usted ahí puede administrar su espacio de nombre. Si apunta a los servidores en el registrador toma los datos y crea registros NS.

¿Dónde está ese servidor de nombre? Eso depende de usted. Si usted es una empresa o una persona técnica, puede crear sus

propios servidores de nombres autorizados y puede administrarlos por su cuenta. Si usted simplemente quiere utilizar los servicios de otra empresa, hay muchas empresas que no solamente alojan los sitios web sino que también ofrecen servicios de DNS. Se puede hacer una combinación. Pueden utilizar los servidores de nombre de un revendedor y otros de otro siempre y cuando haya un servidor de nombres que apunte a myname.com. Todo lo demás puede manejarse dentro de ese archivo de zona. ¿Entiende lo que digo?

ORADOR DESCONOCIDO: Gracias. De las 12 organizaciones que administran distintos servidores raíz, ¿cuál es la que recibe más consultas y cuál es la que tiene mayor cantidad de servidores distribuidos?

STEVE CONTE: ¿Cuál es la instancia que tiene más tráfico y… cuál era la otra pregunta? ¿Y más instancias? Es una muy buena pregunta, cuál tiene más tráfico. No lo sé. Quizá la pregunta más importante sea cuál recibe más tráfico en una región en particular porque algunas instancias están configuradas como para que respondan estratégicamente más en algunas regiones que están más desatendidas.

Hace años, África, América Latina, Asia, algunas partes de Asia, tenían que recurrir a links satelitales para poder acceder a las redes más grandes para poder llegar a los servidores raíz. Ese camino era costoso porque recurre a tráfico satelital y largo porque hay una distancia topográfica y geográfica. Una de las razones por las cuales se utilizan instancias en Anycast es para acercar esos servidores raíz a los usuarios finales, para no tener que hacer esos grandes saltos para acceder a una respuesta. Estratégicamente, los pusieron en puntos de intercambio de Internet en distintos países. Esos puntos de intercambio quizá solamente operen con esa región en particular. Quizá reciban muchas consultas pero solo provenientes de esa región. La pregunta es: ¿Quién tiene más consultas? ¿Quién recibe más consultas? Esa no es una pregunta justa porque se supone que debe ser dinámica. Si se cae un servidor o si el enrutamiento hoy va mejor a un lugar que a otro, entonces la cantidad de consultas va a variar en función de esto.

¿Quién tiene la mayor cantidad de instancias? Diría que hay que hacerle esa pregunta al RSSAC esta tarde. Yo no puedo decírselo. Hay distintos servidores raíz que se esfuerzan mucho por crear. No es una carrera para tener la mayor cantidad de instancias. No es que un servidor raíz sea más servidor que otro. Simplemente tienen los recursos disponibles para distribuir esa instancia en más lugares. Para el usuario final, no marca ninguna diferencia.

No cambia nada. Lo único que le importa es que se le presten servicios, ya sea una raíz L, K o lo que fuera. ¿Responde eso a su pregunta? ¿Hay alguna otra pregunta con respecto a la resolución o cualquier otro tema que hayamos cubierto hasta el momento? Cathy, ¿hay alguna pregunta online?

Como decía entonces, tenemos distintos niveles e interacción humana y agencias dentro del sistema de nombres de dominio. El registratario es típicamente el usuario final. Es mi madre registrando un dominio, o cualquier otra persona. El registratario registra el nombre de dominio ya sea a través de un revendedor o a través de un registrador con quien el revendedor también tiene un acuerdo. Una vez registrado el nombre a través del registrador, el diálogo para obtener la información del servidor en la zona del TLD se da entre el registrador y el registro que es el nombre de nivel superior.

El registro tiene varias funciones que cumplir. La más importante entre todas sus funciones es la de manejar, entregar los datos autorizados de ese espacio de nombres. Tiene todos los registros DNS de todos los dominios hijos debajo. Para minombre.com, .net, example.com. Todos esos son manejados por el registro. Además de manejar los servidores autorizados, cumple otras funciones. La información de WHOIS que puede estar en el registro o en el registrador, dependiendo de cuál fuera el modelo acordado cuando se estipuló el registro. Es la interfaz con la

Internet. También está el servicio RDAP que es otra metodología de búsqueda de la información. El tipo de información puede ser WHOIS pero hay otras.

Luego está la fase pública. El diálogo con el servidor recursivo. El registro tiene dentro una especie de base de datos. Esta base de datos probablemente es un archivo de zonas que toca en algún momento los datos del cliente. Pueden ser varias bases de datos. Tiene algún tipo de API o contacto entre el registrador y el registro. Hay un protocolo EPP de aprovisionamiento extensible que es el método de transporte entre los datos desde el registro hasta el registrador. Luego hay una relación unívoca con el registratario. Si se cae conte.net, yo no contacto necesariamente a .NET que es lo que típicamente sucede cuando hay un problema con un dominio. Muchas veces, a nivel de registratario, si hay algún problema de configuración, recurro al registrador. Como cliente, mi punto de ingreso a este espacio es el registrador. Hablo con el registrador. ¿Qué pasa con mi dominio? Muy rara vez, no se me ocurre una situación específica, el registratario puede contactar al registro directamente pero la mayoría de las veces… Un segundo, tenemos un comentario.

ORADOR DESCONOCIDO: Soy administrador del dominio .DK. Si el registratario tiene un problema, nos contacta a nosotros directamente, no al registrador.

STEVE CONTE: ¿Ustedes actúan también como registradores de los servicios o tienen registradores que venden el espacio .DK?

ORADOR DESCONOCIDO: Tenemos registradores que venden el dominio. No tenemos servicio DNS pero damos soporte si quieren cambiar el proveedor, etc. Tenemos contacto con los registratarios.

STEVE CONTE: Bueno, aquí tenemos un caso. Lo recordaré entonces. Interesante. ¿Tienen muchos registratarios que les contactan por problemas?

ORADOR DESCONOCIDO: *Fuera del micrófono.*

STEVE CONTE: Varios emails. ¿Qué tamaño tiene su registro?

ORADOR DESCONOCIDO: *Fuera del micrófono.*

STEVE CONTE: 1.4 millones. Es bastante el número de interacciones con los registratarios pero es interesante. Gracias por la contribución. ¿Alguna otra pregunta? ¿Cómo vamos con el tiempo? A las 12:30 debemos cerrar. Bueno, acabamos de terminar. Aun cuando no tenemos tiempo, puedo dedicar un par de minutos a saber si tienen alguna consulta. Nuestra próxima sesión no será en esta sala sino en la C1.2 a la 1:45. Tenemos una sesión sobre networking en Internet.

ORADOR DESCONOCIDO: Veo el mapa de root-servers.org. Nosotros no tenemos en mi país, en Afganistán, un servidor raíz sino en países vecinos. Mi pregunta es: ¿cómo se hace un servidor raíz? ¿Qué califica para tener un servidor raíz? Veo que algunos países tienen varios servidores raíz. Quiero saber cómo funciona.

STEVE CONTE: Debemos recordar que la topología de la red no siempre se corresponde con la geografía del mundo real. Por el hecho de que en su país no haya un servidor raíz, no significa que no exista un servidor raíz que esté topográficamente próximo. Es una distinción sutil pero genuina. A nadie aquí le preocupan las

fronteras geográficas. No obstante, respondiendo a su pregunta concreta, si usted considera que su país podría tener un buen servicio con un servidor local, un ISP, puede contactar a los operadores de servidores raíz. Aquí en la reunión hay varios presentes. Asista a la sesión de RSSAC a las 5:00. Ahí va a contactarse con los operadores de servidores raíz. Contácteles directamente y expréseles por qué usted considera que existe la necesidad de tener un servidor raíz en su país. No digamos país sino dentro de su topología o región, por qué es algo importante. Podrá así discutir con ellos distintos modelos de cómo hacer para tener un servidor raíz, su viabilidad, si es lógico, si tiene sentido, si es que uno topográficamente próximo, en un país vecino.

A lo mejor pueden contactarles: "Ya estamos brindando servicio a esa región con estas instancias". Usted puede decir que la tasa de respuesta es baja. Plantéeles el caso a los operadores del servidor raíz y ellos le van a ofrecer versiones de distintos modelos. ¿Alguna otra pregunta? Gracias a todos por asistir. Estas diapositivas van a ser subidas a la agenda en línea. Nuestra próxima sesión, como les decía, sería sobre networking de Internet a cargo de Alain Durand y Jeff Houston. Es en la sala C1.2 o C1.3. C1.2. Gracias. A la 1:45 tenemos networking de Internet. A las 3:15 tenemos uso indebido de la Internet con John Crain, que es el director de seguridad, funcionario de seguridad,

estabilidad y flexibilidad de la ICANN. A las 5:00 tenemos RSSAC. El resto del día va a ser en C1.2. Gracias. Que disfruten el almuerzo.

[FIN DE LA TRANSCRIPCIÓN]