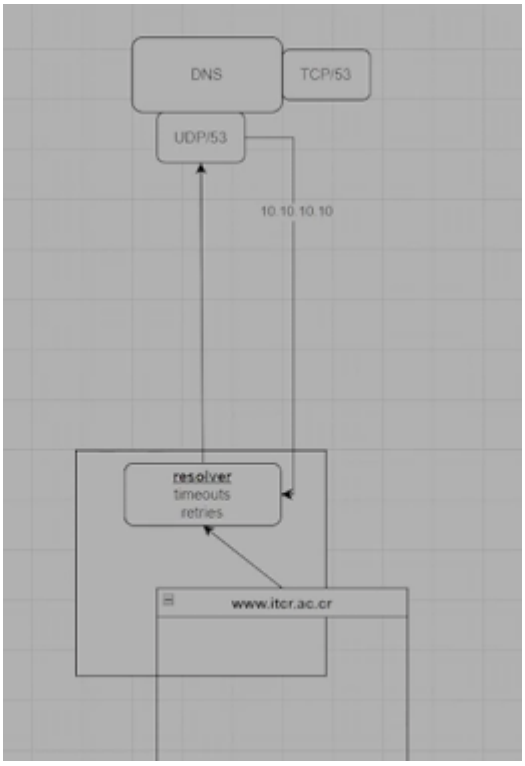
Apunte #5: Clase del 27-10-23

Instituto Tecnológico de Costa Rica. Escuela de Ingeniería en Computación. Redes - IC-7602. Grupo 2. Profesor: Gerardo Nereo Campos Araya Estudiante: Daniel Araya Sambucci - 2020207809 Fecha de entrega: 3/11/23

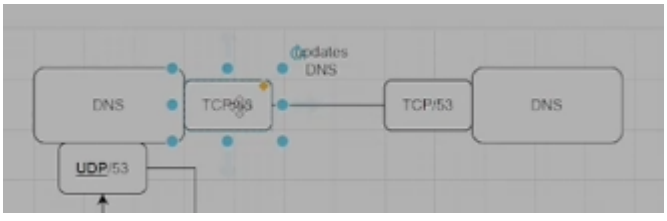
Repaso Clase Pasada

Los sistemas Windows/Linux utilizan un remanente del DNS, donde se podía definir una IP y un dominio (en Windows hosts.txt). Este puede ser sobreescrito para realizar un ataque de Spoofing.

El DNS como protocolo usa dos puertos: el UDP/53 y el TCP/53. Existe el resolver el cual se conecta al puerto UDP/53 y se encarga de manejar los timeouts y retries a nivel de la computadora.

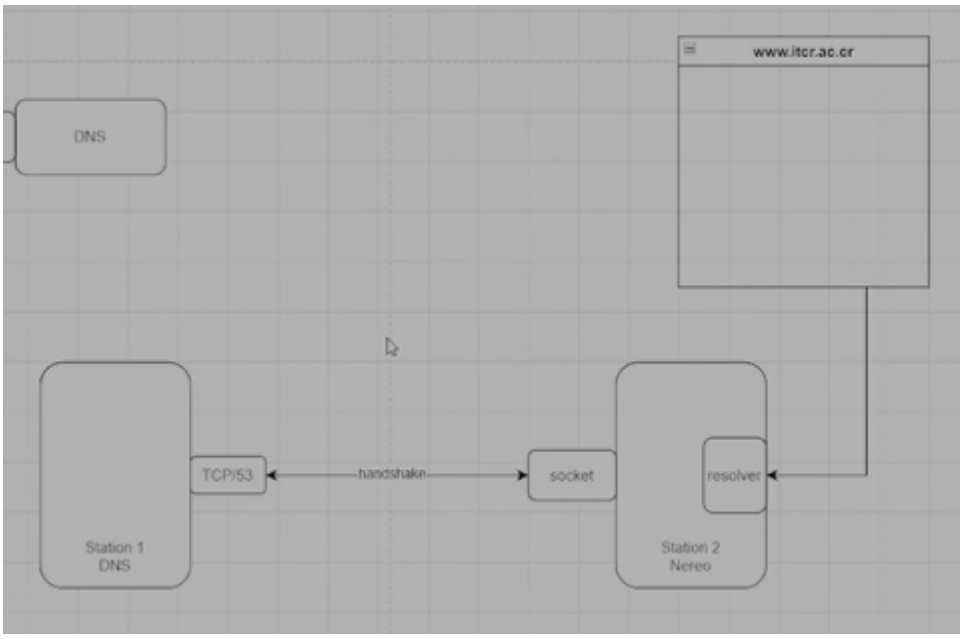


Se le están dando características de un protocolo orientado a conexión a un protocolo no seguro.

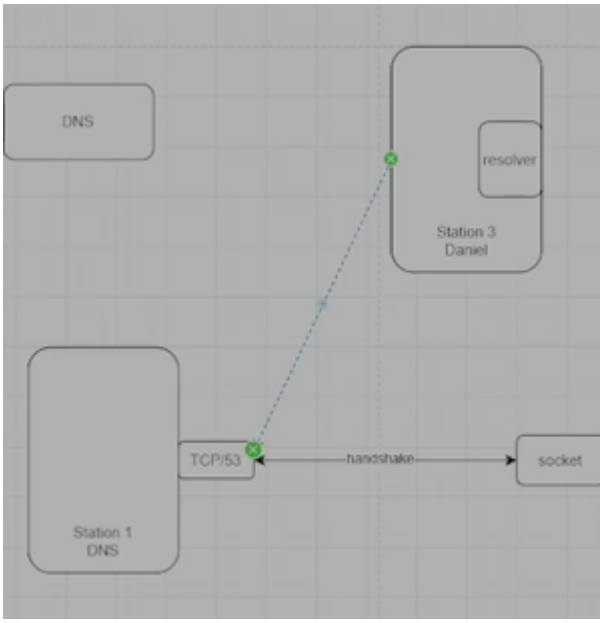


El puerto TCP/53 es más comúnmente utilizado para establecer conexiones. Se mantiene un socket abierto y se cambian updates de DNS. Estas fluyen de lado a lado en la conexión establecida Esto permite poder mantener la conexión abierta.

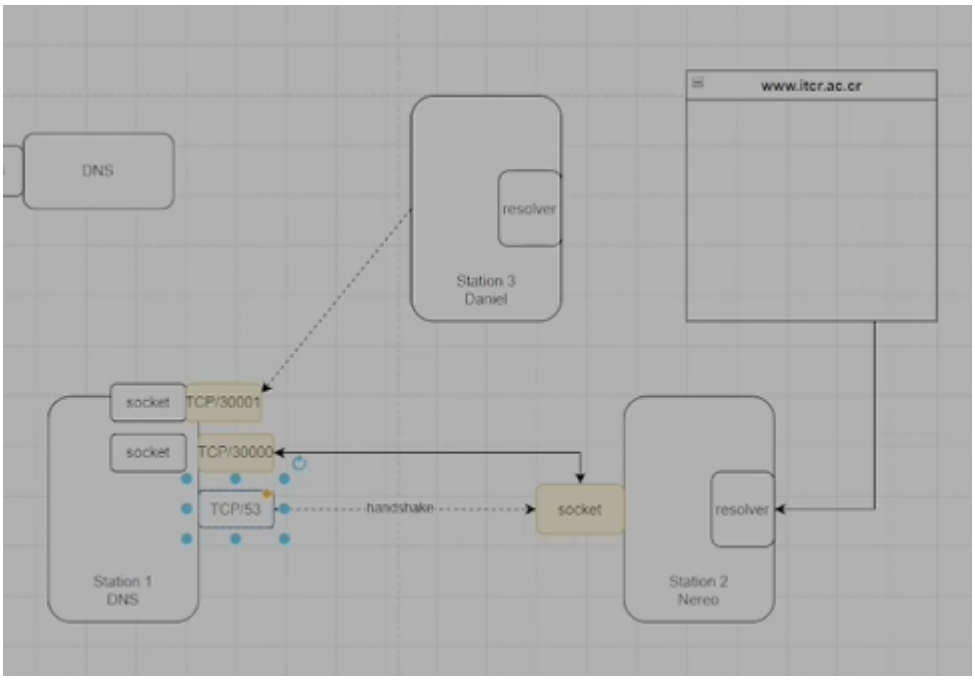
TCP Funciona de la siguiente forma:



La compu de Nereo va a realizar una solicitud a la dirección dada. Entonces el resolver se encarga de realizar esa conexión. Crea un socket para establecer un canal orientado a conexión. Entonces se realiza un handshake y se ocupa el puerto.

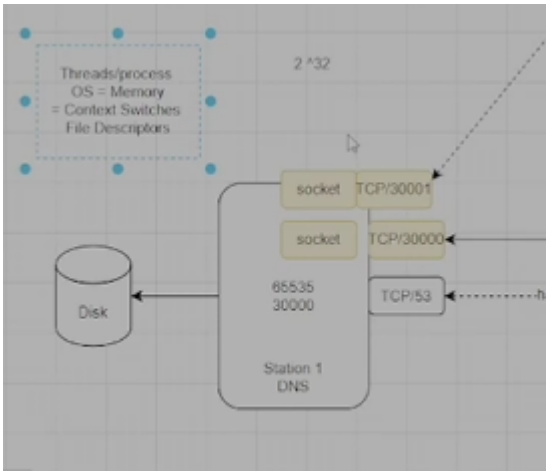


Esta conexión no se puede realizar al estar ocupado el puerto por la compu de Nereo. No se pueden pasar dos canales de datos por el mismo puerto. Entonces TCP, cuando establece la conexión, toma un puerto de un rango de puertos para asignar la conexión de Nereo a ese puerto. El handshake se realiza en el puerto 53, luego se pasa la conexión al nuevo puerto.



Entonces se crea un nuevo socket para manejar el puerto. Entonces la compu de Daniel ya puede conectarse y se genera el mismo proceso para liberar el puerto 53.

Mientras más conexiones se manejan se ocupan más threads, lo que implica mayor consumo de memoria. Esto es un problema para el DNS.



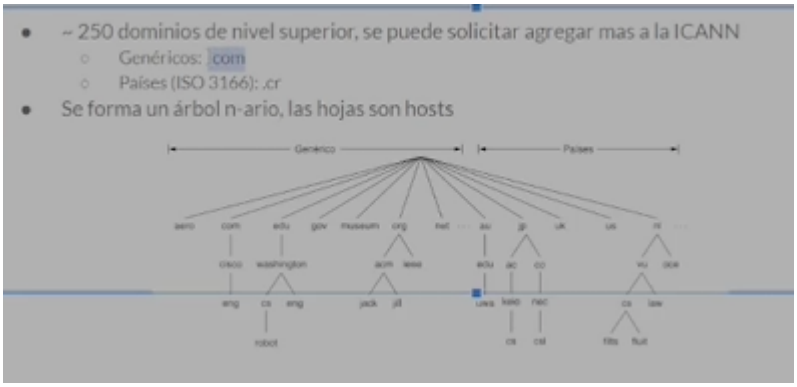
Si tengo que esperar a que se haga el handshake y se de el socket y demás, aumenta el delay. Hay 65534 puertos y el rango utilizado para asignar estos puertos es como de 30000 puertos aproximadamente.

Un servidor DNS no podría servir a un montón de las IPv4. Sería demasiado para manejarlo.

Es virtualmente imposible usar esta arquitectura con TCP, no es viable para el caso de uso del DNS. Con UDP no hay que preocuparse por handshake, el tipo de conexión, crear sockets ni nada de eso. Se usan muy pocos recursos.

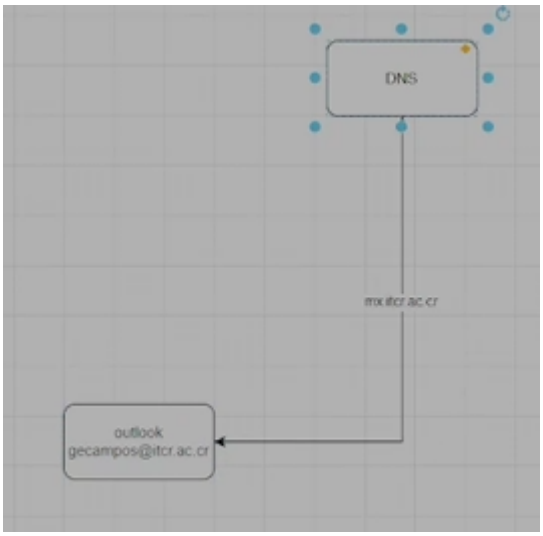
Por otra parte, la conexión entre servidores se prefiere con TCP ya que, al actualizar un DNS, dos servidores puede ocurrir que estén escuchando a valor distintos, por lo que se debe garantizar la conexión entre estos dos. Si los updates fluyen constantemente, es más barato mantener una conexión TCP que estar abriendo constantemente sockets UDP entre los dos servidores, lo que sería mucha carga. Se realiza el handshake 1 vez y se le da a TCP el trabajo de comunicar los sockets.

Domain Name Service



Se tienen dominios de varios niveles. Los dominios propios de países se manejan por ellos mismos. Estos son de primer nivel. Los dominios de segundo nivel se pueden comprar.

Al crear registros, hay distintos tipos. A es para IPv4, AAAA IPv6, CNAME le da un doble nivel para la resoluci[on de nombres. MX se usa para correo electrónico.



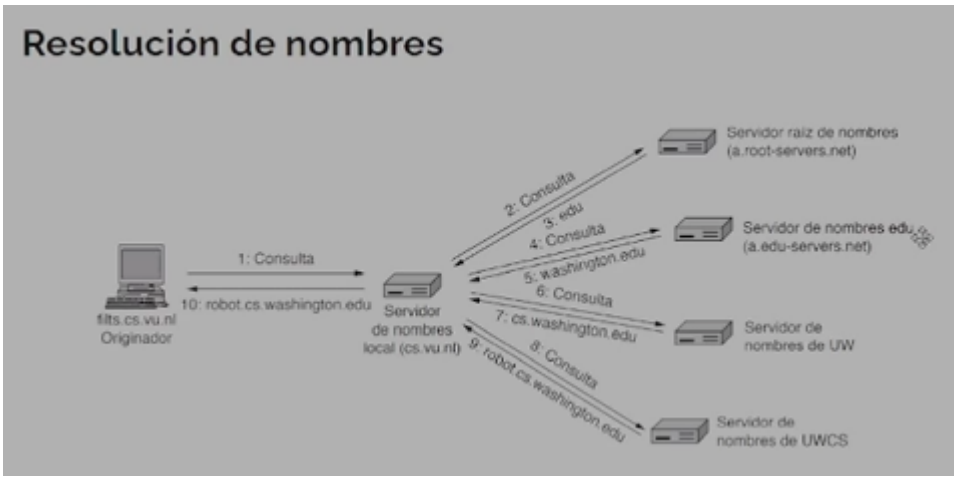
Se manda el mx.dominioCorreo, entonces se recibe una lista de dominios de correo y se van revisando en forma de prioridad hasta encontrar el que se busca.

El tipo TXT es para dar valores textuales en un campo. Se usa para verificar y realizar configuraciones.

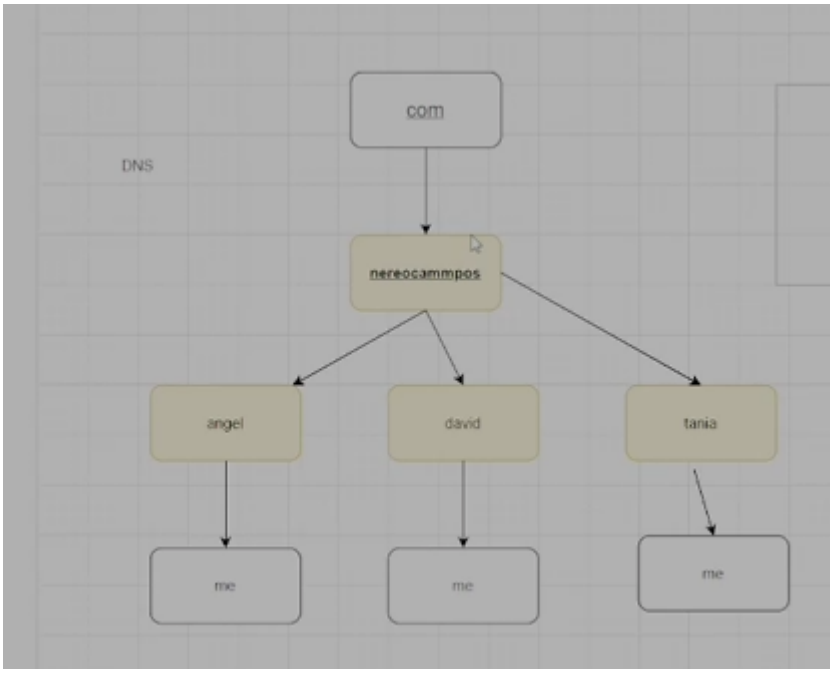
El TTL indica cuánto tiempo se mantiene la conexión en el caché de la computadora. Si se usa uno muy alto, la computadora puede quedarse escuchando una IP vieja. Con los microservicios de los pods, ellos corren por un poco de tiempo y se mueren. Estos cada vez que suben y bajan cambian la IP. Los registros DNS entonces se tienen que actualizar.

Para demostrar el ownership de un dominio, entonces se le pide a Let's encrypt el certificado. Se le crea un registro en el DNS y se le coloca un string indicado por Let's encrypt. Entonces se le avisa y Let's encrypt hace la solicitud al dns y se confirma el dominio. Esto se aplica en TXT.

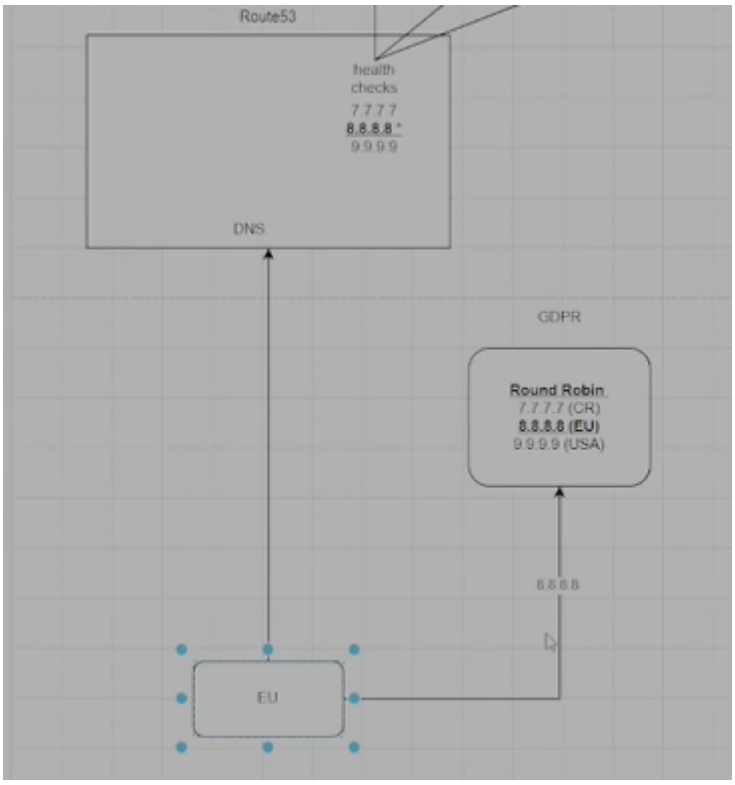
SRV son servicios que puede dar un dominio, como la telefonía. NS es el IP del register que uno va a tener.



En internet se tienen servidores raíz que conocen con quién se puede resolver. Se tiene una jerarquía de servidores Entonces, los servidores raíz solo saben con quién resolver con el primer nivel. Entonces se aplica un Round-Robin. Cada nivel va resolviendo con el siguiente nivel. Es por esto que con el NS hay que indicarle a los DNS quién soy.



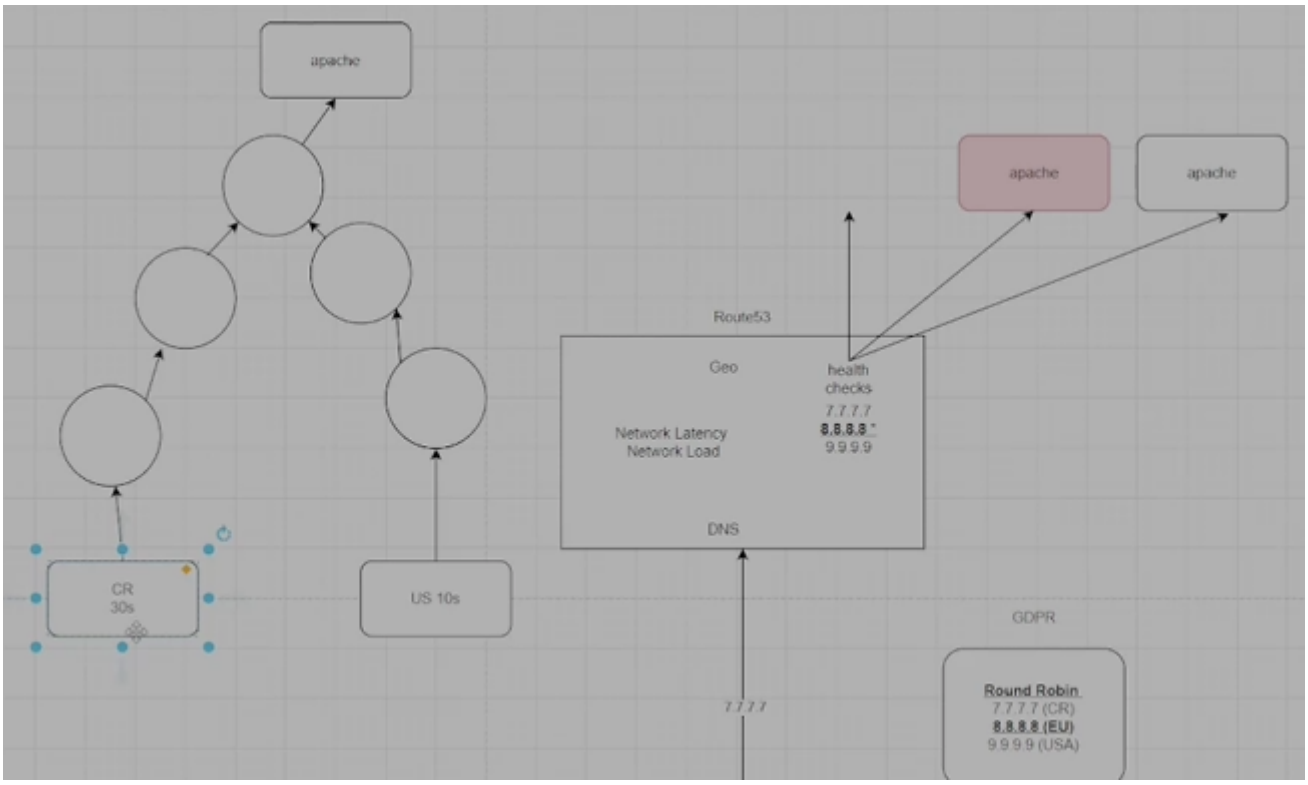
Se tiene el primer nivel, el servidor raíz. Cuando se identifica el dominio .com y se sabe quién puede dar respuesta, se devuelve el IP del servidor que puede decir quién es nereocampos.com. Luego se puede agarrar y buscar a angel, entonces angel va a la zona y jala el dominio me. Los amarillos son zonas. Así se va armando la jerarquía.



El DNS fue pensado para una alta disponibilidad de aplicaciones. Esto causa en un servicio corriente de DNS, cada vez que se recibe una petición, esta va brincando. Entonces se pensó originalmente como un Round Robin. Si una petición tenía un error, entonces daba problemas.

Para solucionar esto, por ejemplo Google creó Route53 que es un DNS pero más potente. Este implementa health checks que empieza a correr pruebas contra los servidores cada cierto tiempo. Este puede comprobar si un handshake está fallando.

Otro problema que podía surgir era que, al pegarse a un servicio, puede ocurrir que se pegue a un servicio muy lejos. Esto afecta a nivel de redes la congestión y otros factores. Si una persona hacía una petición a un servidor pero con la IP de otro país, entonces está violando la ley de protección de datos. Para solucionar esto, Route53 implementa geolocalización. Esta no solo hace el handshake, sino que se revisa la ubicación de la petición para darle el servidor más cercano.



Route53 también implementa Network Latency y Network Load. Si se tiene un servidor Apache, entonces el Route53 empieza a hacerle handshakes. Estos además de revisar que los servidores sean funcionales, están obteniendo métricas.

Aunque geográficamente el servidor esté ubicado más cerca de Costa Rica, el camino es más rápido por US. Esto significa que hay un servidor congestionado por el camino de CR. Además, se puede identificar que no hay carga en la ruta de US. Entonces de esta forma se hace un balanceo de carga.