**Tema 1 – DHCP**

DHCP → Dynamic host configuracion protocol → un protocolo para configurar host automáticamente.

Cuando un equipo está en una red necesita tener una ip, podemos hacerlo todo manualmente, pero podemos automatizarlo mediante un DHCP.

Esto es la leche para configurar cosas.

Ahora, cuando yo quiero ofrecer un servicio web, tener una IP dinámica no mola porque no puede gestionarse bien, si yo monto mi web, y mi ip pública del router es X, cuando mi amigo se conecte, la primera vez funciona, ahora mañana será otra, y yo no cambio la ip del dominio, no puede entrar.

Esto se solucionaría con 2 cosas, estableciendo una ip estática (a pagar) o usar cosas como dyndns o no-ip.

Esto son programas que lo que hacen es analizar tu ip publica y anotarla, si el DHCP te cambia la IP, se da cuenta del cambio y configura tu dominio automáticamente a esa nueva IP.

Cuando mi router de ip publica recibe peticiones web lo que hace es redirigimiento de puertos, esa petición me la manda a mi PC con mi IP por el puerto 80.

¿Como funciona DHCP en una red?

Primero tenemos que distinguir DHCP cliente y DHCP server, el servidor es quien proporciona el funcionamiento DHCP a toda la red, y el cliente es el programita que se encarga de comunicarse con ese servidor para solicitar la IP.

Vamos a hacer ejemplito con packet tracer.

1º: El cliente manda una trama broadcast a la red con un DHCP DISCOVER, esta trama lo que busca es un servidor DHCP en la red. Para ello manda un discover a toda la red, es decir, broadcast.

Como el pc esta limpito, no tiene ip, esta trama se manda con IP origen 0,0,0,0: 68, y es una trama de tipo UDP. Se utiliza UDP porque vamos a hacer una solicitud muy breve, rápida y ligera, es decir, no merece la pena establecer una comunicación TCP por todo el proceso lento.

El destino de esta trama será broadcast es decir 255,255,255,255:67

Si el CLIENTE manda un discover y no obtiene respuestas, se repite todo el proceso y vuelve a solicitar un DHCP.

2º: Dentro de la red, todos los servidores DHCP escucharan ese discover, y por tanto, ofrecerán su servicio al cliente, es decir, cada uno de ellos mandará una trama DHCP offer, en esta trama le dan todo lo necesario, pues su ip, su mascara, dns, etc. Si solo hay un server, pues solo habrá 1 oferta.

3º: El cliente elige una de las ofertas, o coge la que es única si es única, y esa elección debe ser comunicada a los servidores, así como a los otros clientes de la red (aunque estos descartan la trama porque no va con ellos el asunto). Por tanto, se manda una nueva trama, DHCP Request, también de tipo broadcast.

En esta trama se especifica la ID del DHCP que ha sido elegida. DE esta forma, el DHCP sabe que ha sido elegido.

4º: Se produce una trama DHCPACK (un paquete ack, acknowlogment, es decir, acuse de recibo), es decir, el servidor dice que recibido y que muchas gracias por elegir mi servicio.

Así que ahora con la teoría, vamos a hacer una práctica con máquinas virtuales, concretamente vamos a instalar un servidor DHCP, concretamente ISC (tenemos link con los pasos en el aula)

Practica con 2 máquinas:

1º instalamos el servicio

OJO, ahora tenemos que quitar el modo NAT de la maquina y ponernos en red interna o no furula

2º configuramos la tarjeta-elegir cual vamos a usar básicamente

3º configuramos el archivo de config con el rango de la red que queremos y la propia red

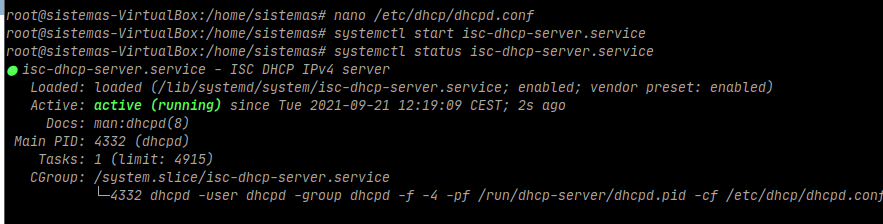
4º nos asignamos en la máquina que hace de servidor una IP ESTATICA de esa red

NOTA: Aquí lo primero, comprobar que tu IP no está en modo automático.

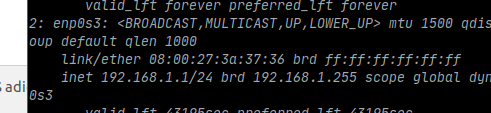
NOTA2: Comprobar que la IP está bien puesta con ip a s /ipconfig, si está bien puesta, nos debería salir la ip, y si no está bien puesta, desconectamos el cable de red y lo conectamos, para que se actualice.

5º hacemos el enable servicio, start servicio y status servicio. Si esta verde, estamos guay.

6º conectaos con otra máquina virtual, TAMBIEN EN RED INTERNA, y le hacemos un ip a s o un ipconfig /renew si es windows, y deberíamos tener una IP

Esto es en el server

Esto es en el cliente

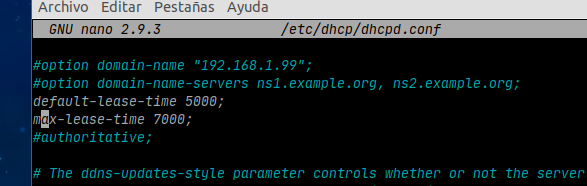


Hoy vamos a proseguir un poco con el servidor dhcp de ayer.

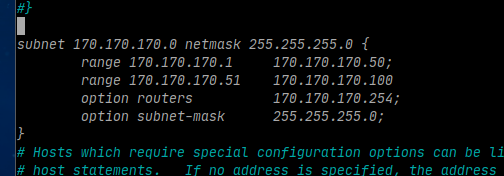
Vamos a probar diferentes rangos de ips, lo que se llama lease o leasing en inglés, algo como préstamo de ips, y el dns.

Luego lo subimos al github. Hoy voy a intentar hacer fotitos, que ayer no hice, y las ha pedido.

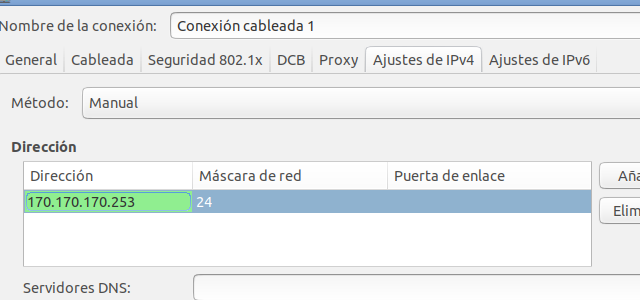
Aquí tenemos cambiado el LEASE time (era 3000 no sé cuántos y 8000 o algo así, además estaba comentado)



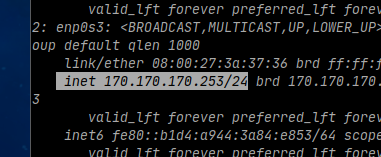
Aquí cambio el rango de Ips, ayer tenía solo 1 rango, y era en la 192,168,1,0



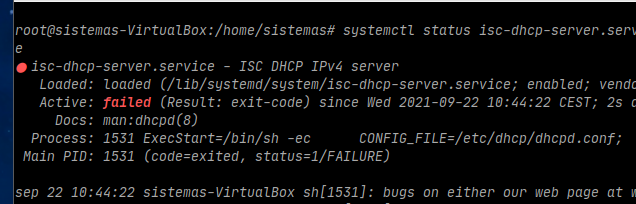
Cambio la IP del servidor



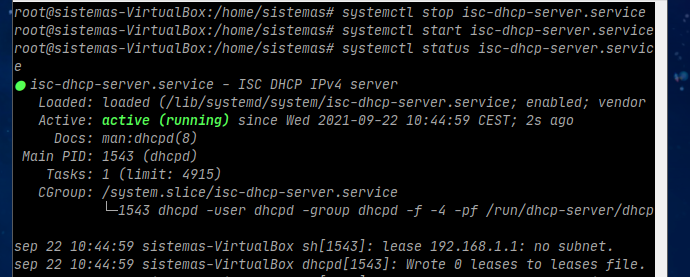
Y compruebo que se ha cambiado



RECUERDA poner las malditas ; y } en el doc de confi, que si no te pasan estas cosas

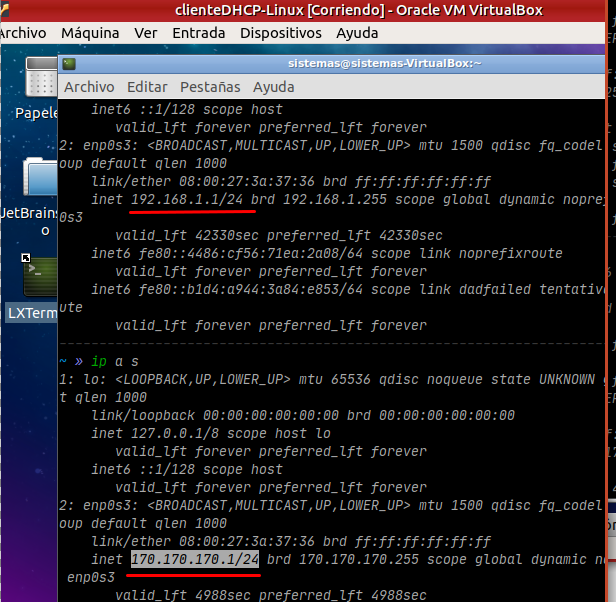


Ale, solucionado



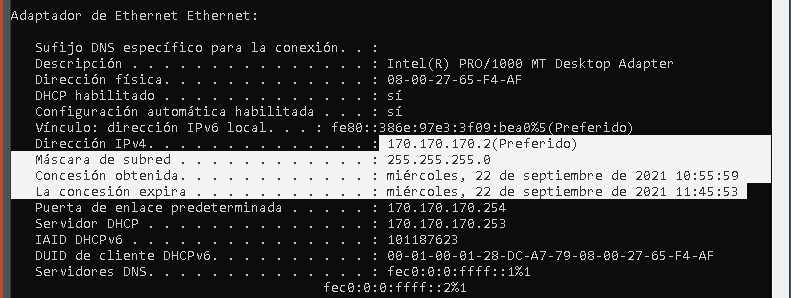
Nos vamos al cliente

Estábamos en la red anterior, guardamos los cambios, y solicitamos la nueva ip. Ahora lo mismo, pero en windows bonito.

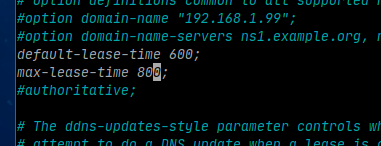


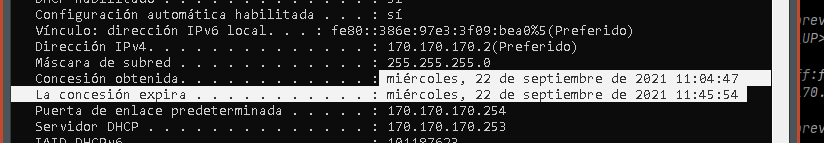
En el

windows me ha dado la dirección 2, porque la 1 esta cogida ahora mismo por el linux, y como tengo puesto 3000 de LEASE time, que son 50 minutos, pues la concesión me dura 50 minutos.



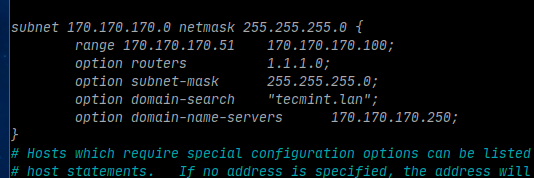
Voy a bajarlo a 600 y hacer un renew, a ver qué pasa. /RELEASE para lliberarte y /renew para pillar otra nueva, mas eficaz que solo renew.



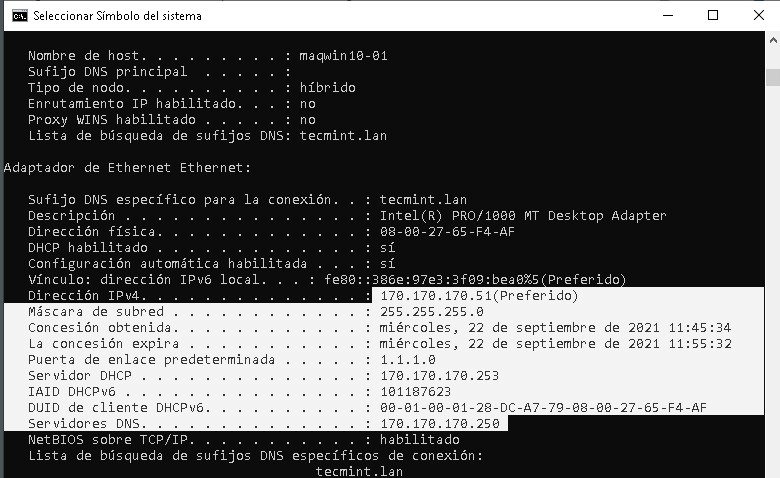


La IP NO CAMBIA, pero si me ha cambiado el tiempo de concesión, aunque son 40 minutos, y en teoría el máximo son 13. Explicación a posteriori: Entre que son máquinas virtuales y fallan un poco, y que lo hice mal para conseguir el /renew, pues pasan cosas, cambiar el lease funciona, haz primero el /reléase y luego el /renew o no se aplican correctamente.

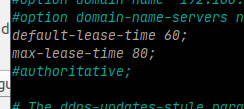
Como quiere 3 cambios, pues vamos a cambiar cosas: Quito uno de los rangos, ahora tiene que darme una ip diferente, le he puesto un router diferente, y le he puesto DNS, a ver qué pasa. Esto lo tengo en un documento aparte, mejor explicado.

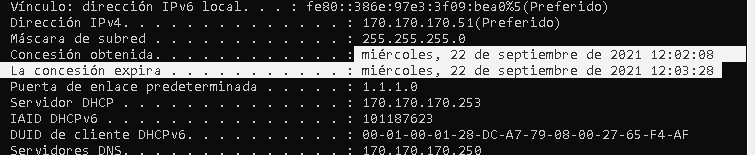


aquí podemos ver que tenemos el servidor DNS 250, el router es el 1,1,1,0, y que la IP es del rango 50-100. Y abajo del todo sale también el DNS especifico. Buen trabajo chicos, ha funcionado.



En el windows, cuando queramos hacer cambios de ips y cosas para comprobar, haz primero un ipconfig /release y luego el /renew, es decir, abandonas tu ip, y pides una nueva, en lugar de estar haciendo movidas de cambiar la red o reiniciar el pc.



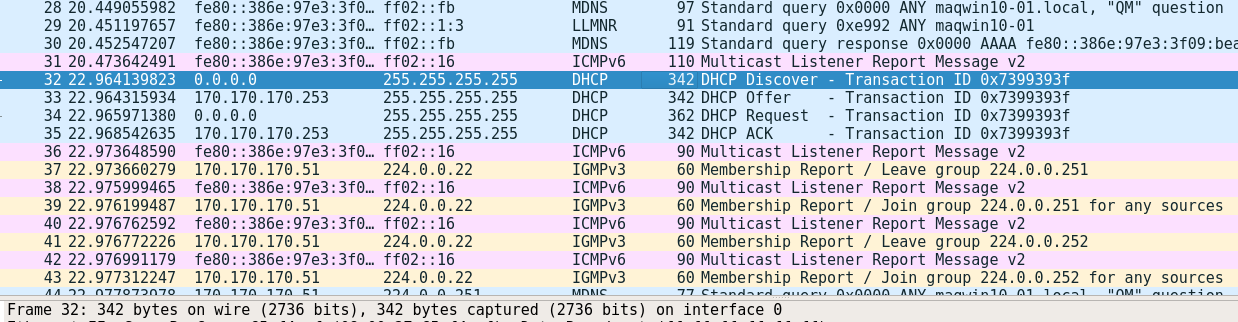


Ahora vamos a instalar wireshark, lo primero cambiamos a nat porque estamos en red interna, lo instalamos.

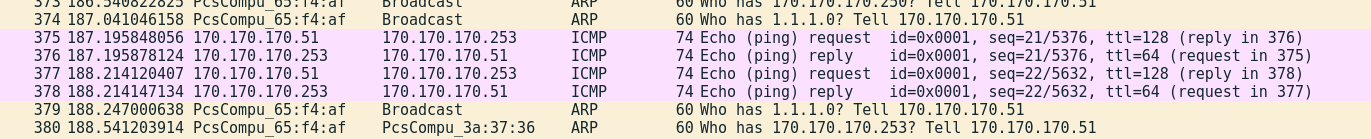
IMPORTANTE, para abrirlo, o lo abrimos como sudo, o tenemos que darle permiso al os usuarios, porque no tenemos permiso, para ello sudo chmod +x *usr/*bin/dumpcat y listo

Ya podemos capturar tramas.

Si en el cliente me quito la ip y le pido una nueva, podemos ver todo el proceso de tramas.

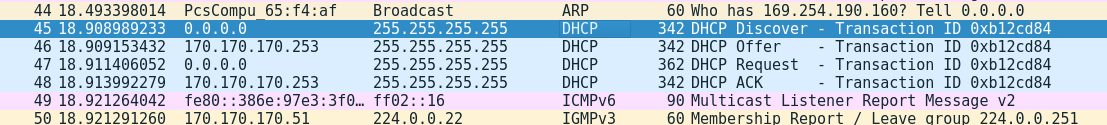


Si pruebo a hacer un ping vemos un icmp



Vamos a ejecutar wireshark para ver las tramas

En el equipo CLIENTE, ahora mismo no tengo IP, asi que solicito una ip con /renew. El wireshark está analizando toda la red y capta esto

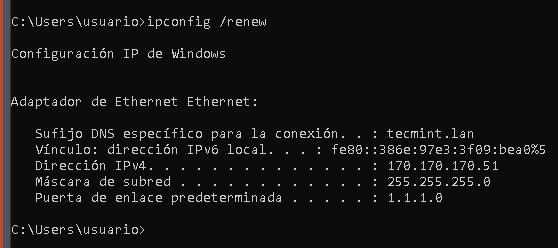


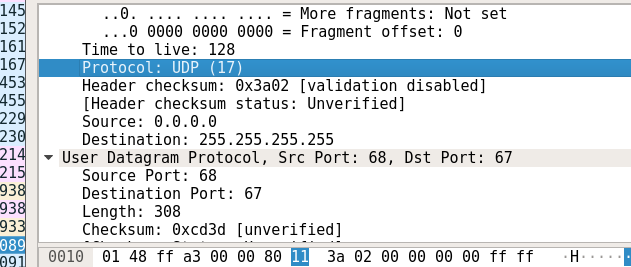
Como vimos ayer, hay alguien sin ip (0,0,0,0) mandando un DHCP discover

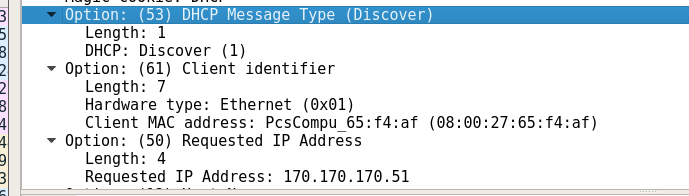
Entonces el servidor DHCP le responde con un Offer y los parámetros correspondientes.

Como solo tenemos 1 DHCP, el cliente le manda un request indicando que acepta la IP

El servidor DHCP le dice que ok.

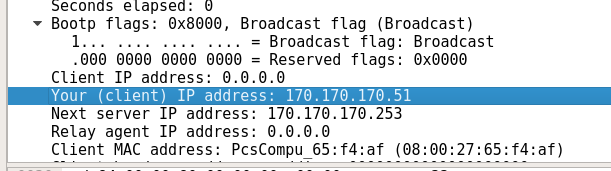


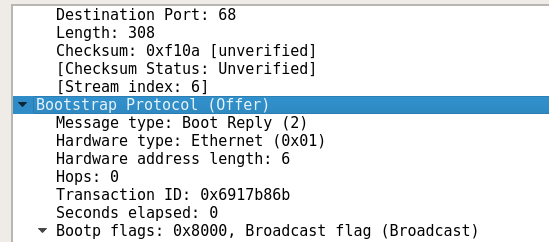
Podemos ir mirando 1 a 1 las 4 tramas para ver su contenido.

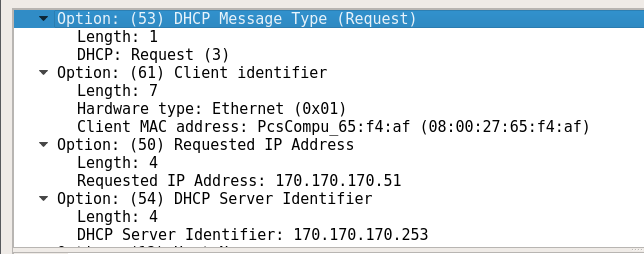


En la trama discover podemos ver un poco todo el contenido: Protocolo empleado, IP origen, IP destino (broadcast), puertos, el TTL, etc.

Si miramos la trama DHCP OFFER, podemos ver como el servidor ve que el cliente no tiene IP, y le manda una IP para él solito, con todas sus demás opciones.



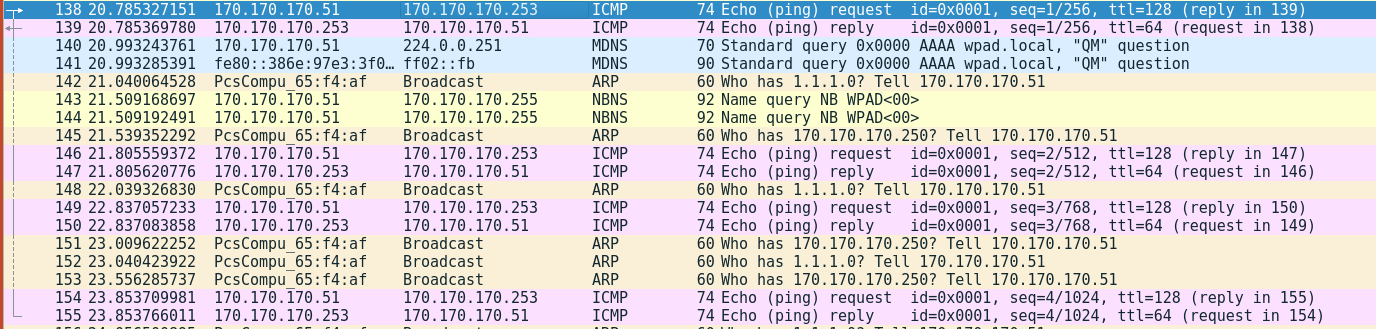




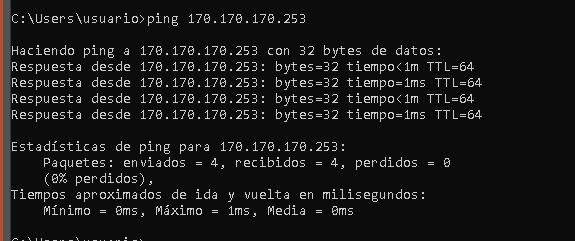
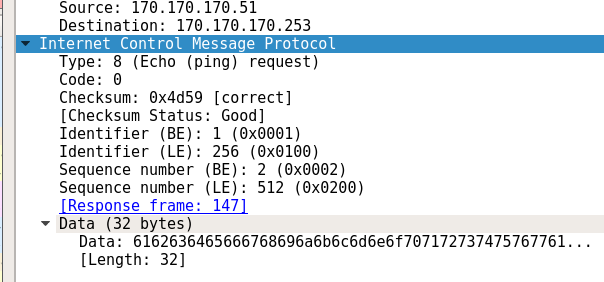


Podemos probar a mandar un PING desde el cliente hasta el servidor DNS

Se mandan el correspondiente ICMP reply y un REQUEST, y comienza el envio del PING



Si lo comparamos al cliente, aquí solo podemos ver el TTL y los bytes, mientras que en wireshark podemos ver TODOS los protocolos que se involucran, y los contenidos de las tramas.

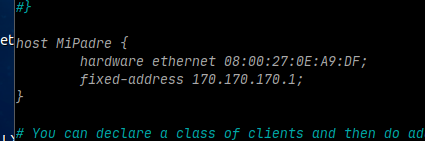


Dato curioso: En la CMD de windows sale TTL 64 todo el rato, mientras que en wireshark salen 128 y 64. Si nos fijamos con detalle, las que aparecen con TTL=64 en wireshark son las tramas REPLY, es decir, las que manda el servidor, y las que vemos en la CMD son las tramas que ha recibido del servidor, osease, es correcto.

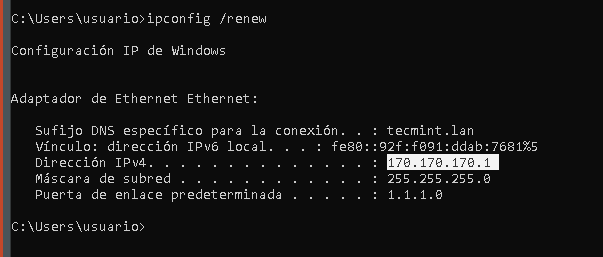
El ping manda 32 bytes según la cmd, y si abrimos la trama, podemos ver la DATA del ping, y efectivamente, hay 32 bits de datos.

A veces puede darse el caso, de que el servidor DHCP mande una trama DHCP NACK → significa no ack, es decir, NO te dejo usar la ip que te he dado.

Podemos asignar Ips estáticas en un servidor DHCP, para ello tenemos que configurar en el archivo de config otras líneas.

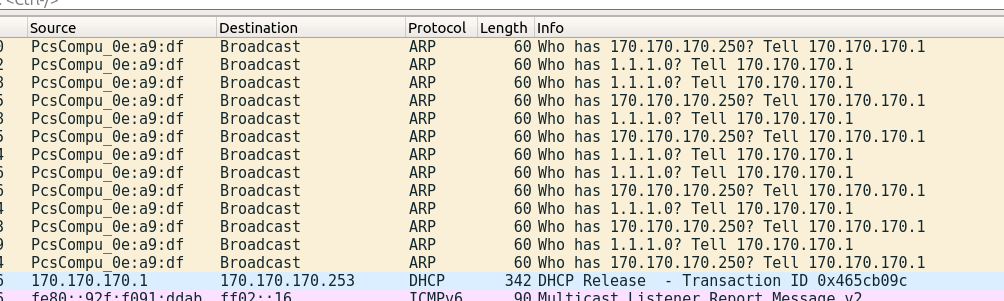


Importante que pongas los 2 puntos, porque los guiones de windows no funcionan.



Estoy dentro de la misma red, y tengo la IP estática asignada.

Con el protocolo ARP se busca, por un lado, conocer las Ips de todos los otros equipos de la red, y otro uso es preguntar quien tiene una IP X y así sabes si la tiene alguien, o si está libre, y por tanto puedes solicitarla



Cuando hacemos un /release seguido de un /renew, se ejecutan los 4 protocolos DHCP que hemos visto, y adicionalmente, posterior a esto, se ejecuta un ARP Gratuitous



Se hace después del DHCP para comprobar que la Ip que te han asignado esta libre, si aparece alguien que ya la tenía, entonces es que tu estas duplicando, y por tanto la soltarás y pedirás una nueva.

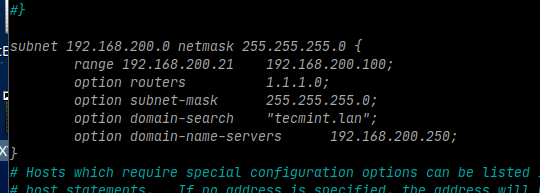
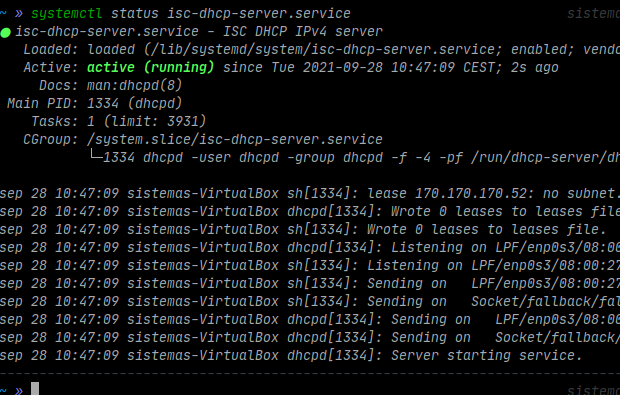
Escenario 1 para hoy

Tener 1 servidor DHCP + 2 clientes corriendo a la vez, para que se hagan peticiones y no tenga 1 cliente solo 1 servidor dhcp para él solito, que comparta. (W10+linux)

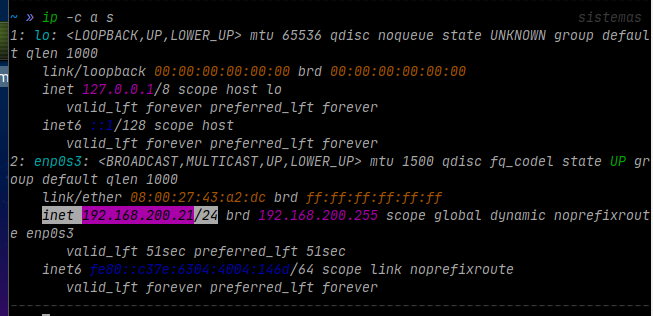
Escenario 2:

Tener 2 servidores DHCP y un cliente, para que el cliente haga lo de elegir un servidor.

Cuando inicies el segundo servidor DHCP, acuérdate de cambiar las ips y el rango en el archivo de confi, que es un clon, ergo es idéntico al primero.



Servidor iniciado y operativo, con esa configuración, arranco los clientes.



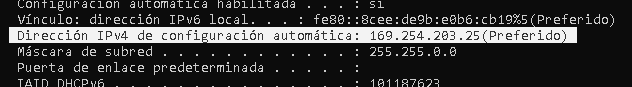
En el cliente linux tengo ya conexión, me ha dado una ip.

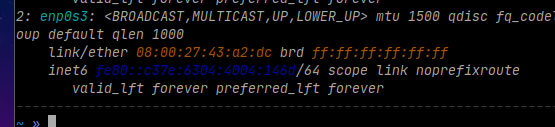


En el cliente windows también tengo conexión, me ha dado una ip.

Vamos a verlo todo ahora en wireshark

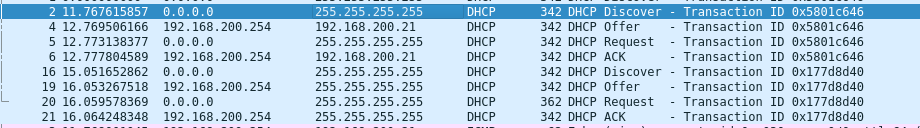
1º → Hacemos ipconfig /release y sudo dhclient -v para quedarnos sin ips en ambos clientes.



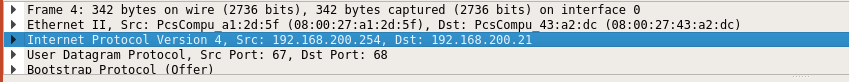
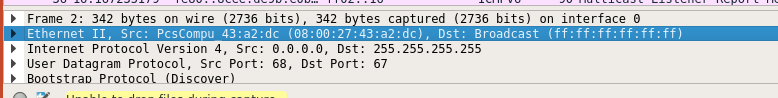


2º → Abrimos wireshark, capturamos trama y conectamos los clientes. Aparece 2 veces el proceso de DHCP protocol.

Si nos fijamos en el contenido de la trama vemos que cada uno es para un cliente.



El primer DHCP es para el cliente linux, le otorga la dirección correspondiente.

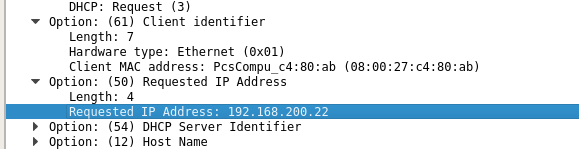
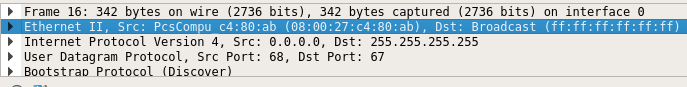


Le ofrece la .21, acepta la trama y se hace el ACK.

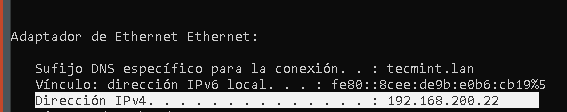


Comprobamos que efectivamente la tiene.

Vamos al siguiente

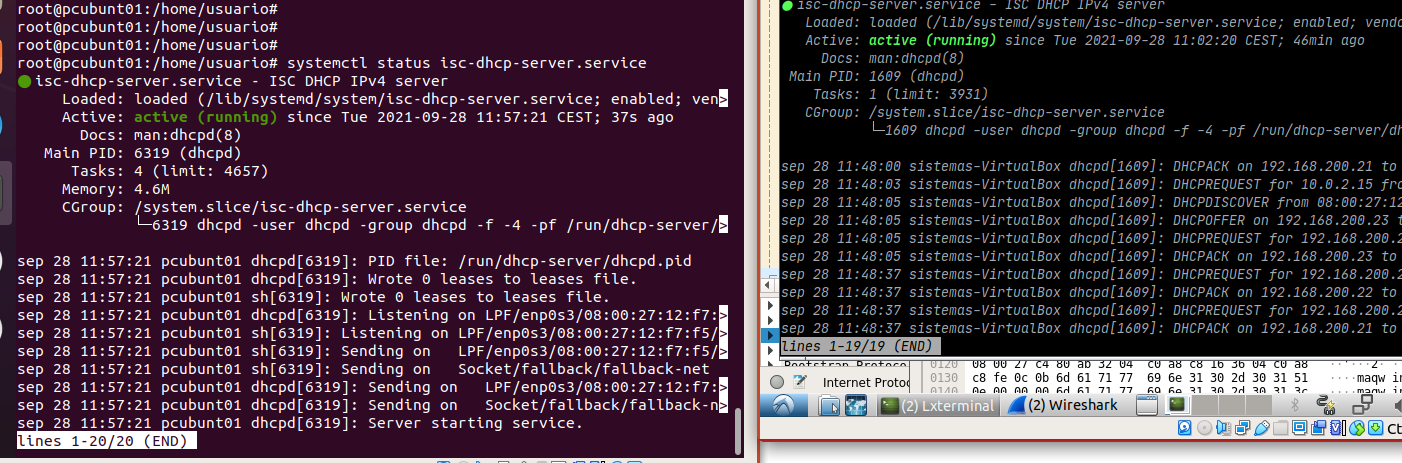


Aquí tenemos la IP del cliente2.



Pasamos al ejercicio 2.

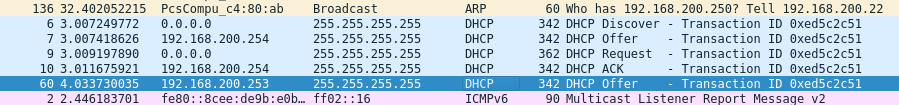
1º → Configuro un nuevo servidor DHCP dentro de la misma red



El primer DHCP es el de antes, tiene la misma configuración. Este nuevo DHCP le he asignado otro rango de direcciones.



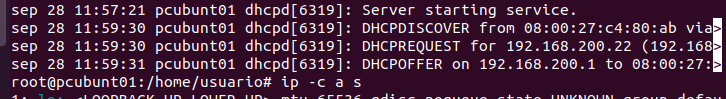
2º → Le quitamos al cliente windows la IP para hacer todo el proceso, y solicitamos una IP a la red.



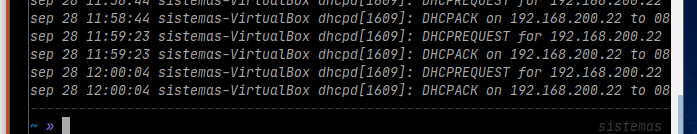
Aquí podemos apreciar que hay 2 DHCP offer, el primero es del servidor 254, que es quien realiza todo el proceso, y luego más tarde llega el servidor 253.

Según tengo entendido, por defecto, está configurado para que los clientes cojan la primera IP que les llegue. Y para evitar que los servidores descartados bombardeen con el DHCP offer a la red, el Request del cliente informa a todos los servidores que ha escogido una red, para que no manden ese broadcast innecesario.

De hecho, me acabo de percatar, en el SYSTEMCTL status, podemos ver los paquetes DHCP, interesante porque podemos ver las peticiones sin necesidad de wireshark, pero no podemos ver el contenido de la trama, asi que es útil pero no mucho.

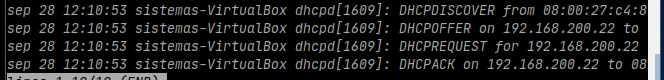


Aquí se puede apreciar como el servidor Ubuntu recibe un discover, y acto seguido recibe un Request del mismo cliente diciendo que ya tiene una IP, justo antes de que el propio servidor le mandase su offer, es decir, se corresponde con lo que vemos en wireshark en la imagen anterior.



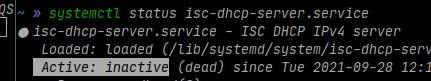
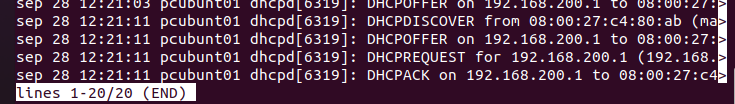
Mientras tanto, el otro servidor está renovando la IP del cliente.

Si vuelvo a liberar la IP, puedo ver aquí también los paquetes, nuevamente siendo el servidor 1 el más rápido.

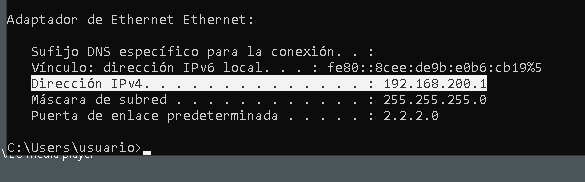


Ahora, dado que ya me funciona todo con los 2 servidores, voy a complicarme yo solito la vida y ver qué pasa, voy a probar a desactivar este DHCP, para que el cliente coja una IP del otro servidor.

El servidor 2 recibe el offer y hace todo el proceso, exactamente igual que antes.



El otro server esta apagado.

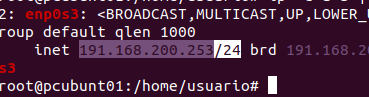


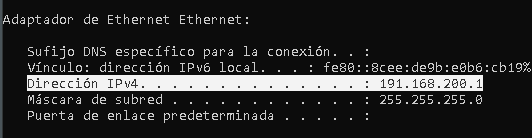
El cliente recibe una IP del otro servidor, genial, funciona.

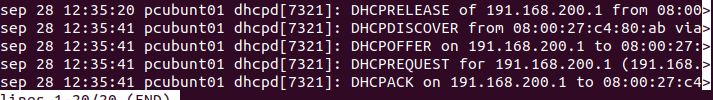
Adicionalmente, voy a probar otra cosa, voy a poner CADA SERVIDOR en una RED DIFERENTE.

El original se queda en la 192, y el segundo, lo pongo en la 191.

1º → Voy a probar que funcione solo 1 servidor, para ver que no hay problemas raros con las ips y las redes.

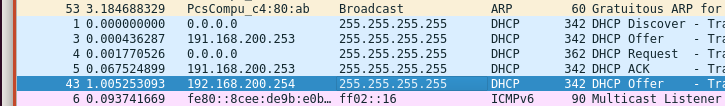






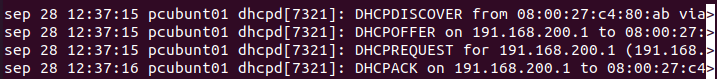
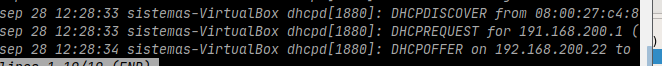
El cliente recibe una IP correctamente de otro rango de red.

Ahora voy a levantar los 2 servidores y voy a ver como se pegan entre ellos por darle una ip al cliente.



Si miramos en wireshark, nuevamente hay 2 offers diferentes, el de la 191 y el de la 192.

El cliente vuelve a coger el de la 191, manda el request y lo recibe el servidor 192.



Al igual que antes, los 2 servidores reciben el DHCP discover, mandan su offer, y uno de los dos se quedará sin poder otorgar direccionamiento al cliente. Con la particularidad de que ahora el ejercicio utiliza DHCPs de diferentes redes.

Hoy vamos a repetir el escenario 2 de ayer, pero con una variación.

Vamos a hacer que los 2 servidores estén sincronizados y trabajen como una pareja peer.

Tenemos todo en la web que nos ha puesto en el aula

El failover es básicamente un servidor de repuesto, tenemos 2 servidores, el primario que es el que trabaja y el segundo que guarda la info. Failover es básicamente que, si muere el primario, el secundario espera X tiempo a ver si vuelve, y si no vuelve, asume las funciones del primario durante un tiempo, por eso registra lo que hace el primero.

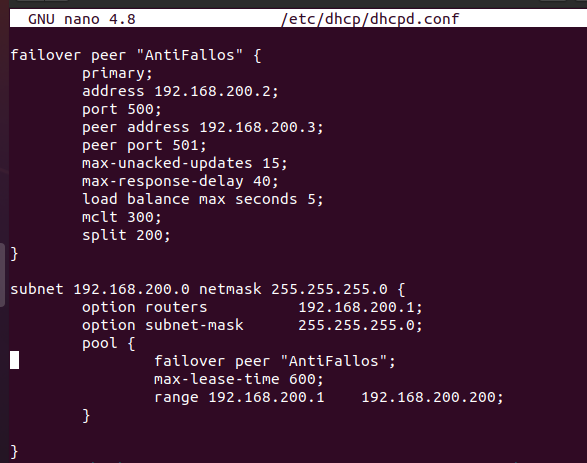
Además, podemos configurar un secundario con balanceo de carga, es decir, en toda la red, tenemos muchos equipos, pues podemos configurar un primario con cosas, y poner balanceos de carga para que cada uno trabaje con X capacidad total, por ejemplo 50-50

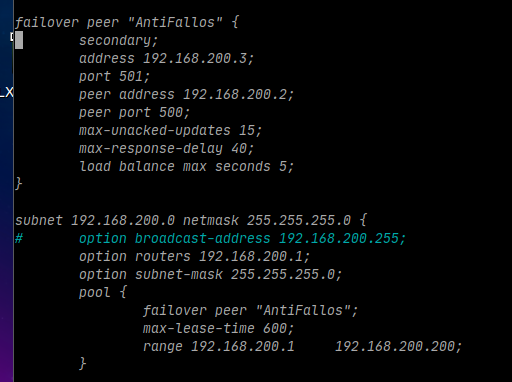
El split va de 0 a 255, es decir, hay 256. Solo se configura en el primario. Si el primario tiene split X, pues el secundario se encarga del resto.

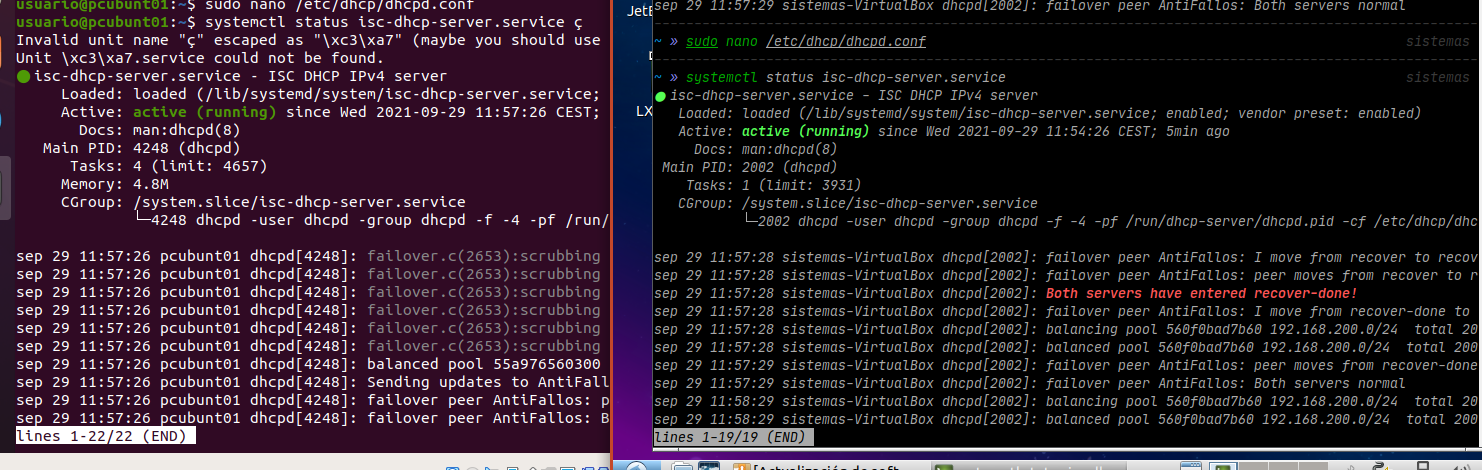
A diferencia de ayer, al ser failover, DEBEN ESTAR EN AL MISMA RED, es la gracia de que sean panitas.

Ejercicio de FAILOVER

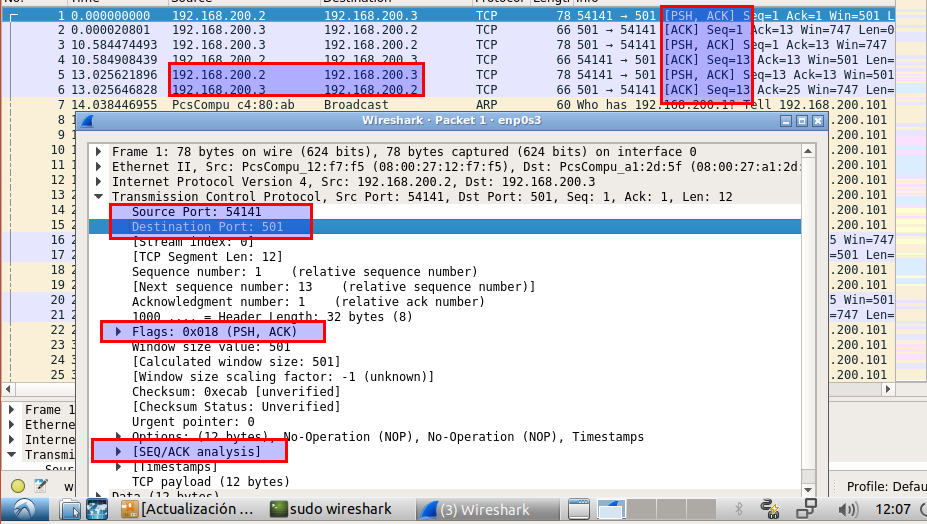
1º → Configurar los archivos DHCP de cada servidor







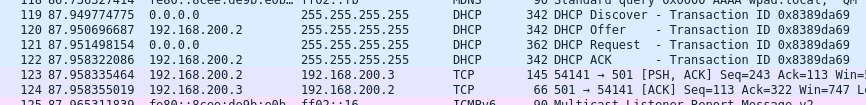
Ambos servidores operativos.

2º → Abrimos un cliente, vamos a solicitar una dirección IP, con wireshark analizaremos lo que sale.

Lo primero que vemos es que los 2 servidores se están comunicando entre ellos vía TCP y mandándose mensajes ACK.

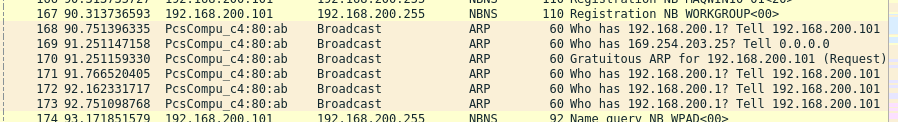
Nota: Al parecer el puerto 500 está ocupado, porque el servidor primario tiene el 54,141.

Liberamos la IP de windows para solicitar una nueva.



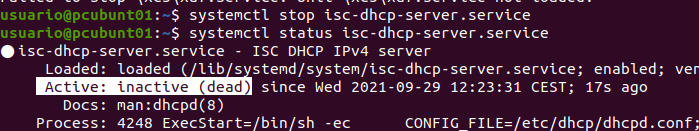
Se realiza todo el proceso DHCP y acto seguido los servidores se comunican, para informar de que ha asignado una nueva IP.

El cliente está a su rollo mirando a ver quién hay en la red.

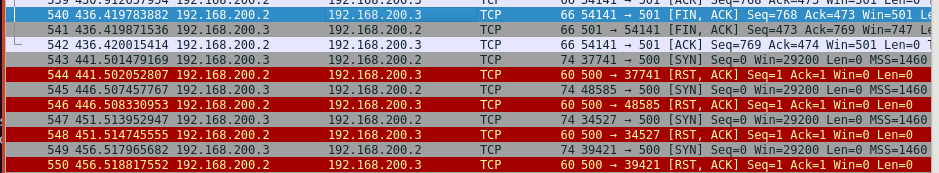


Y a la vez manda una trama broadcast para indicar que ahora tiene una dirección IP (gratuitous).

Si yo desconecto uno de los 2 servidores podremos ver que ocurre en la red.

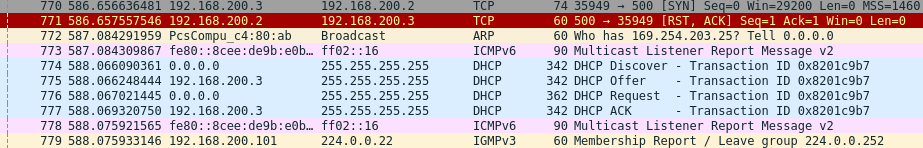


3º → Comprobar la trama wireshark con un servidor muerto



Intenta establecerse comunicación, pero como el servidor 200.2 está muerto, no consiguen mandarse los paquetes.

Ahora pruebo a reiniciar la IP del cliente



Como se puede ver, sigue habiendo problemas de comunicación entre servidores, pero el servidor de repuesto sigue ofreciendo servicios para los clientes.

El ejercicio que hicimos ayer supone fallas de seguridad, pues con los dhcps se podría hacer que un señor maloso se meta en tu red y configure el otro servidor que te redirija a sitios, o se mete en tu red y puede espiarte, usar tus gateways, etc.

Hay que tener bien configurada la seguridad también. Pero eso es de otra asignatura.

DHCP spoofing es eso, un dhcp malicioso 🡪 un cliente se cuela en nuestra red, escucha peticiones DHCP discover y manda respuestas de tipo DHCP falsas, para robar cosas. El cliente se identifica como “default gateway” por lo que el trafico de esos clientes pasara primero por él. Esto es un ataque man in the middle.

dhcp snooping → es un protocolo del switch que se encarga de verificar y comprobar todo el tráfico DHCP, es decir, sirve por un lado para evitar ataques por spoofing (solo los paquetes aceptados por snooping serán validos en la red, por tanto, los paquetes DHCP falsos acaban muriendo en el camino), y, por otro lado, sirve para revisar y comprobar que los equipos se configuran correctamente con el DHCP.

Si abrimos wireshark y capturamos trama, y abrimos un navegador y accedemos a una web, veremos varias cosas

Podemos ver como hacemos un DNS, nuestro pc viajara al router/proxy, haciendo preguntas DNS de tipo query y recibiendo responses.

El protocolo DNS es de capa de aplicación, y trabaja en UDP.

En algún momento aparecerá la query de nuestra petición web y su respuesta.

Hay que matizar que veremos dos querys, una que es de tipo A y otro tipo AAAA

1 a significa ip v 4 → el dns solicita la ipv4 de la web pedida

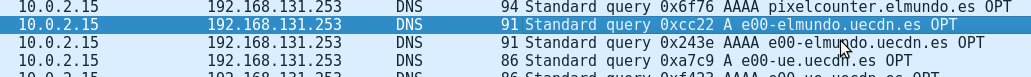
4 a significa ip v 6 →

Si abrimos la trama de wireshark podemos ver en el answer a esa petición las Ips correspondientes

ipv6 repasito → los dobles dos puntos solo se pueden poner una vez para no generar ambigüedad

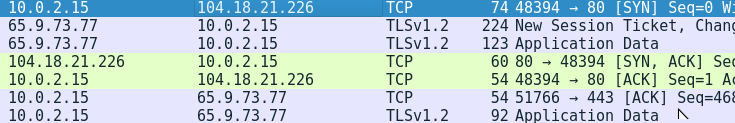
fe::01 bien

fe01::ad1::a8 MAL, cuantos 0 hay en los primeros puntos, y en los segundos? Casca

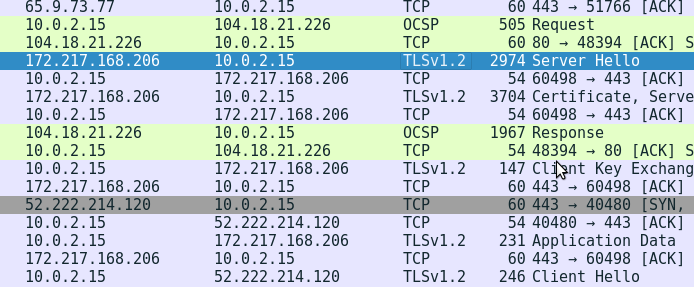


Aquí tengo el query del mundo y los response, con el contenido y la IP del mundo.

Una vez se conocen las ips, comienza el protocolo TCP, con el 3wayhs, y la conexión



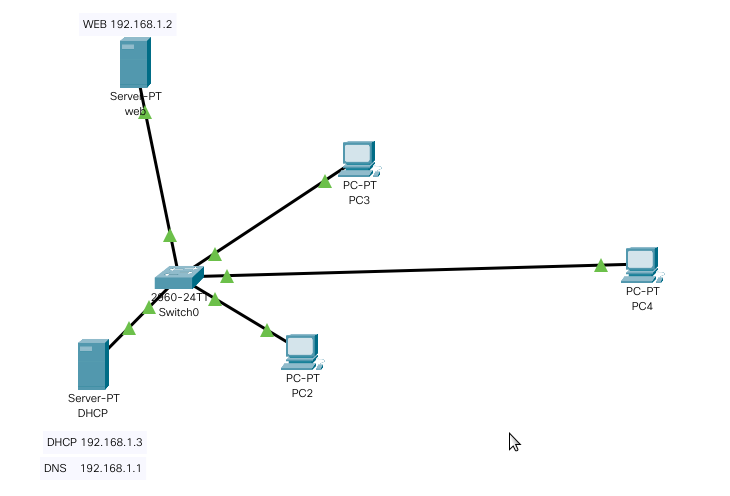
En amarillo tienes el 3wayhs, en este momento, hay conexión entre mi pc y el servidor.



Por un lado, tenemos un request y la respuesta (dame tu página principal)

Y por otro lado tenemos un saludo del servidor y del cliente

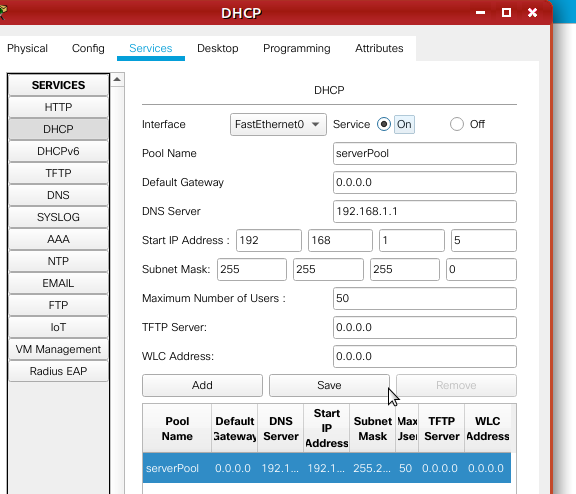
Vamos a hacer una simulación

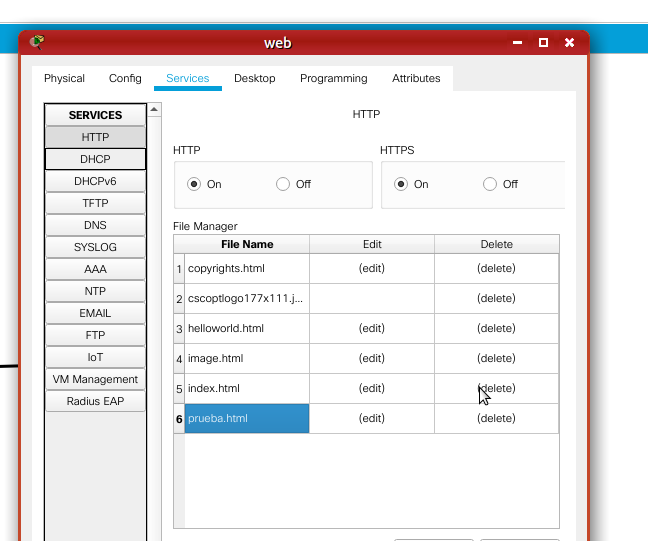


Tenemos este escenario, un servidor con DHCP activado y que además actúa como dns

La idea es configurar un dhcp para asignar direcciones, y tenemos una web con contenido. Necesitamos añadir un dns, o bien aparte, o dentro del dhcp, para que traduzca nombres.

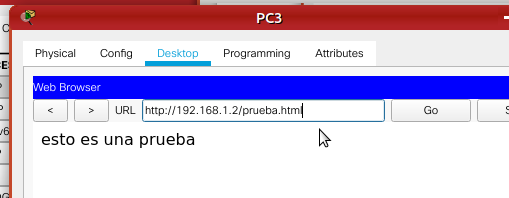
Ponemos los clientes en modo dhcp



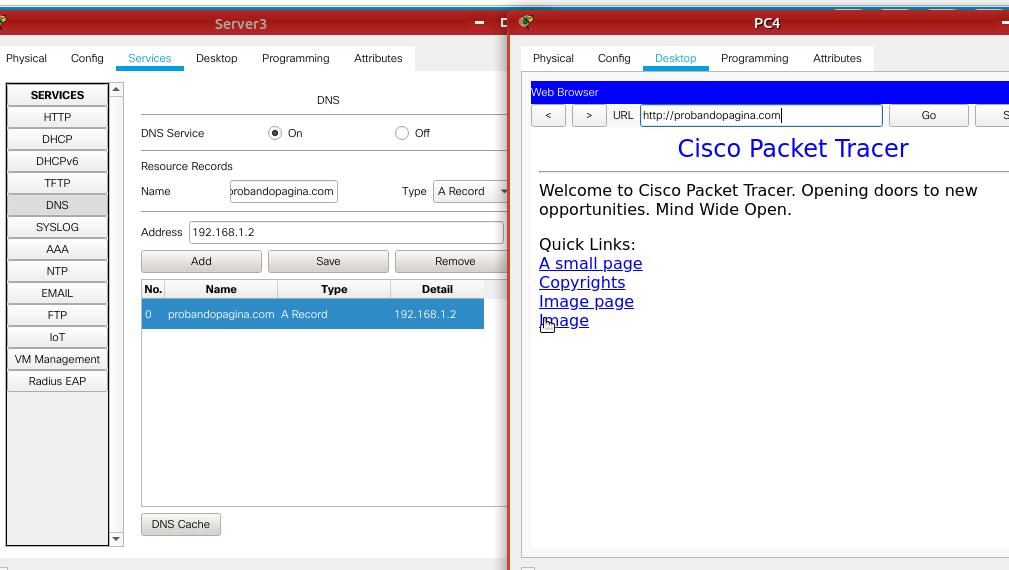


Los pcs se ponen automáticamente en el rango 5-55.

Si accedo a la web vía IP, funciona sin problemas.



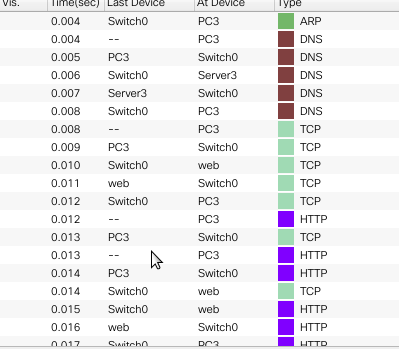
Ahora quiero ponerle nombre de dominio y acceder por dominio para ver cómo actúa el DNS.



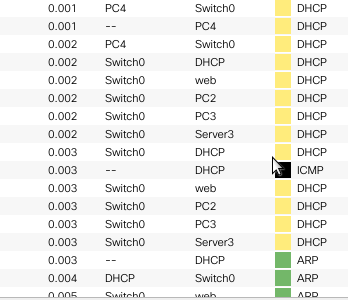
Configuro un TERCER servidor DNS, con la página y la ip asociada, e intento entrar por nombre en lugar de por IP.

Funciona.

Si nos metemos en el modo simulación podemos ver las diferentes tramas enviadas.



NOTA IMPORTANTE: si cambiamos cosas del dhcp o del dns, luego renueva siempre las Ips de los Pcs, para que cojan la confi actual, a Dani no le estaba funcionando porque seguía con una confi anterior.



Si libero un pc de ip y pido una nueva, podemos ver también toda la trama dhcp.

El ultimo día instalamos el servidor dns y tal.

Si miramos el wireshark, podemos ver que al hacer la petición al dns hay 2 tramas:

A(address) record → es una traducción del nombre de dominio a una ip → mipagina.com = 192,168,1,30

NS (name server) record → es el que se encarga de redirigir mi petición web al servidor DNS correspondiente, al dominio correspondiente; como sabes existen muchos dominios, si yo hago una petición .com, el DNS buscará esa web y su ip, ahora, en base a ese dominio, tendrá una IP u otra.

.com = ,30 .es = ,31 por ejemplo

Todos estos registros los podemos configurar en el archivo de confi del servidor DNS. Hay muchos tipos de registros.

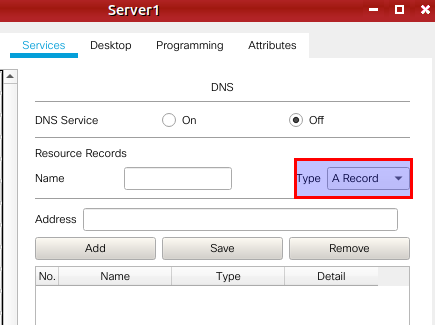
CNAME record → registro canónico, es el que busca por alias, darle X nombres a la misma máquina.

[ftp.miiservidor.com](ftp://ftp.miiservidor.com/) → 192,168,1,10

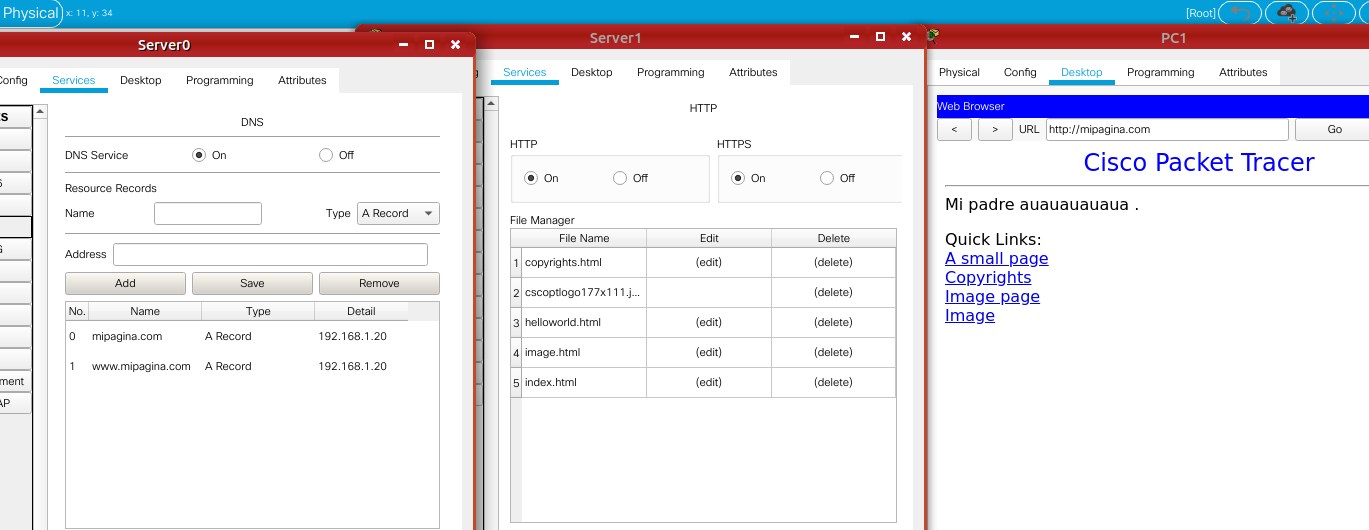
[www.miweb.com](http://www.miweb.com/) → 192,168,10

Ambos son la misma máquina, pero con diferente nombre y tipo de acceso, el registro este lo que hace es anotar con que nombre has entrado a esa máquina.

MX record → se usa en servidores email, registra mensajes enviados-recibidos y tal.

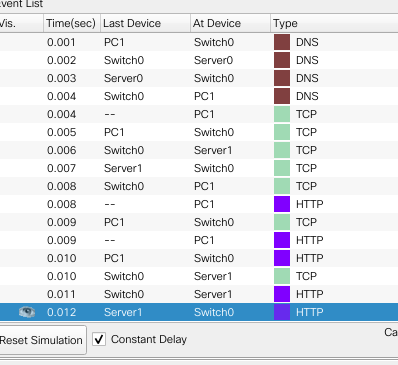


Vamos a hacer un mini escenario sencillo para ver estas tramas en PT

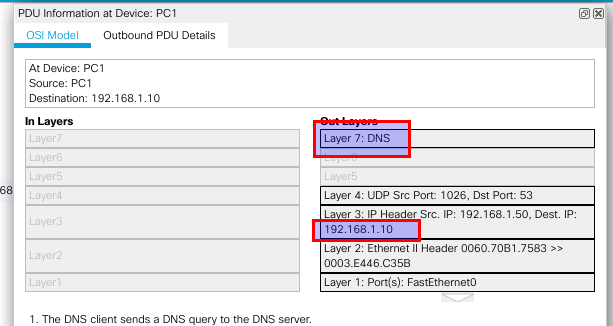


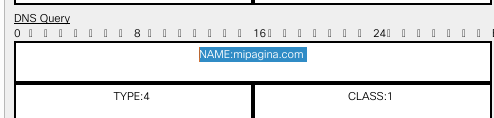
Configuramos todo y ya funciona, hay conectividad entre clientes y servidor.

Vamos a ver las tramas que es donde esta lo interesante:



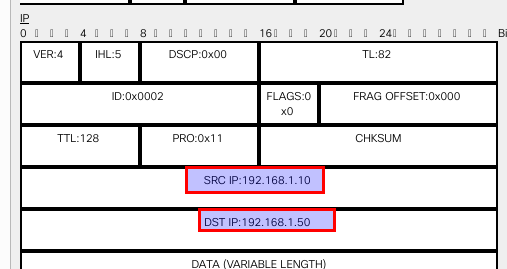
Al lanzar una petición web tenemos el DNS de la página, el TCP y el HTTP. Dentro de cada uno podemos ver sus correspondientes tramas.

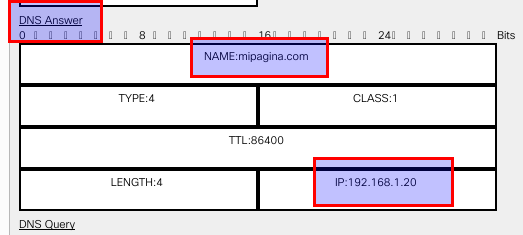


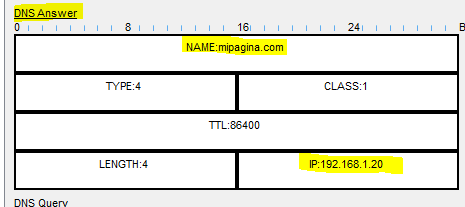


En la trama DNS apuntamos primero al DNS server para resolver el nombre y recibir una IP.

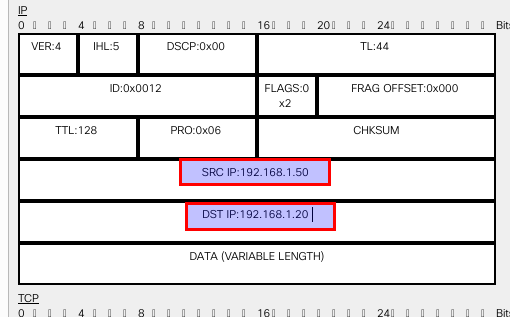
El servidor le responde al cliente, y le manda en la trama de datos la IP del servidor web.



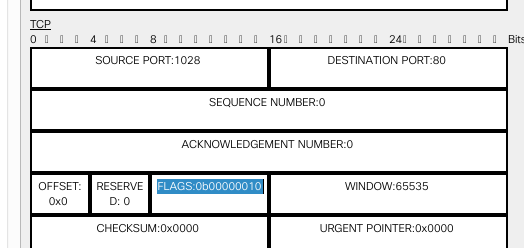




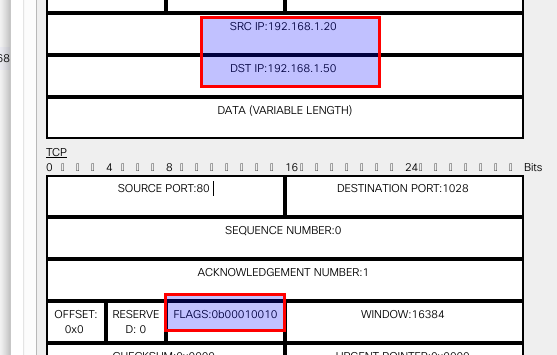
Ahora el cliente ya conoce la IP del servidor y se inicia el protocolo TCP de conexión.



Si miramos el apartado de TCP, podemos comprobar las flags.



Hay una flag, de sincronización con el servidor. Si miramos la respuesta del servidor, podremos ver que hay un cambio de flags.



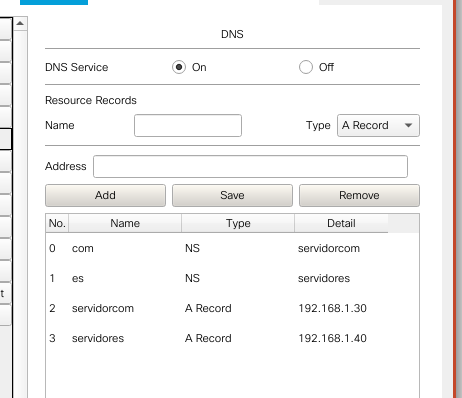
Dicho de otra forma, como estamos en el protocolo TCP, estamos viendo el 3way handshake ahora mismo.

Vale, visto eso, ahora vamos a añadir un SEGUNDO servidor DNS, pero ese va a encargarse solo de los dominios .com y el otro de los .es

Cuando tenemos varios DNS, cada uno para una cosa, hay que tener configurado todo correcto.

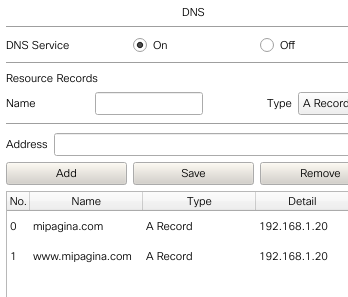
SERVIDOR DNS PRINCIPAL → Por el que sale mi red, en ese indicamos a qué servidores hay que redirigir la petición.

Servidores secundarios 🡪 Tendrán guardadas las diferentes direcciones, y se las mandarán a las peticiones.

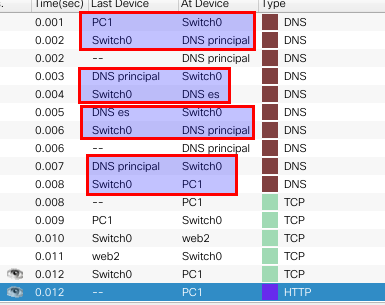


Tenemos que tener esa sintaxis, un A RECORD con el nombre del servidor y la IP, y luego un NS RECORD en el que se indique, para peticiones (NAME), redirígelas al servidor (DETAIL), que tienen la dirección (DETAIL 2, definido en el A RECORD).

Y luego en cada servidor, pues configuraremos las páginas que sean.



Ahora, si analizamos una trama wireshark de una petición web, podremos ver el redirigimiento de DNS



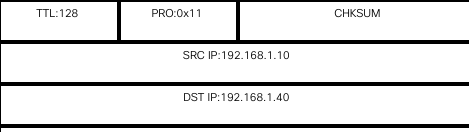
El cliente hace la petición, lo recibe el servidor DNS principal, este no conoce el destino, así que se lo manda al servidor correspondiente en base al nombre que ha recibido, el servidor ES le manda la dirección de esa petición web, y entonces el principal ahora conoce la IP del destino y se lo manda al cliente. Esto es lo que se conoce como DNS recursivo.

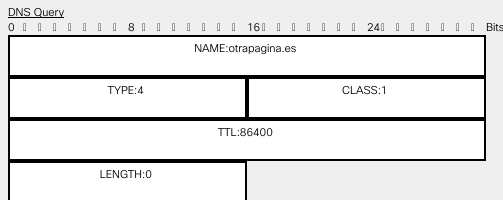
Hay 2 formas de hacer esto: interactiva o recursiva

Recursiva: Yo le pregunto al DNS, este no sabe el destino y se busca la vida, pregunta a un dns diferente, le responde, y se lo manda al cliente.

Interactiva: Yo pregunto al DNS, este no lo sabe y me dice “Mejor pregúntale a Juan, a lo mejor lo sabe”, le pregunto a Juan, y este me responde.

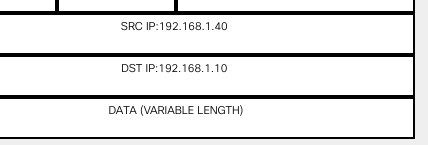
SI analizamos la trama entre DNSs veremos la comunicación

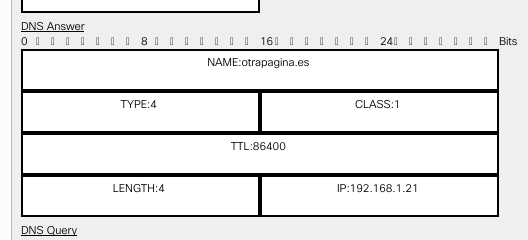




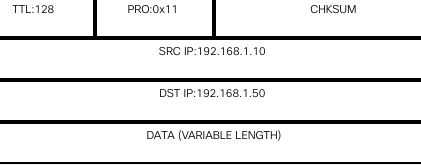
Como ves, falta un campo, es el campo de la IP, porque el servidor DNS principal no lo sabe.

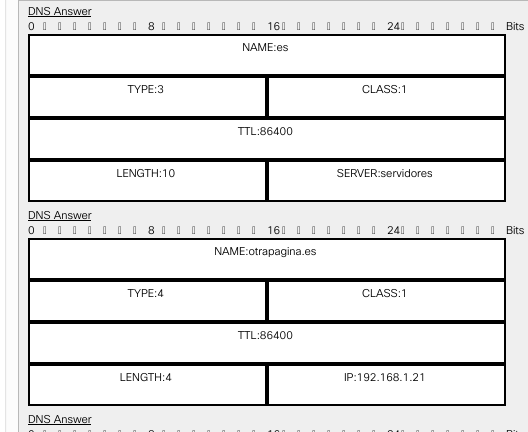
Así que le responde el servidor dedicado. Le estoy preguntando: dame ese campo que me falta.





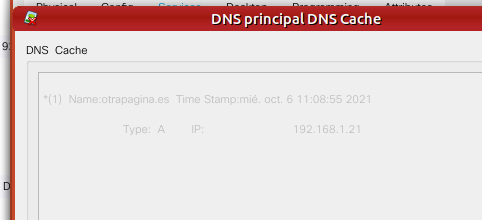
Aquí en la respuesta SI aparece la IP de la página web. El DNS principal aprende la dirección y ahora le responde al cliente a su petición.



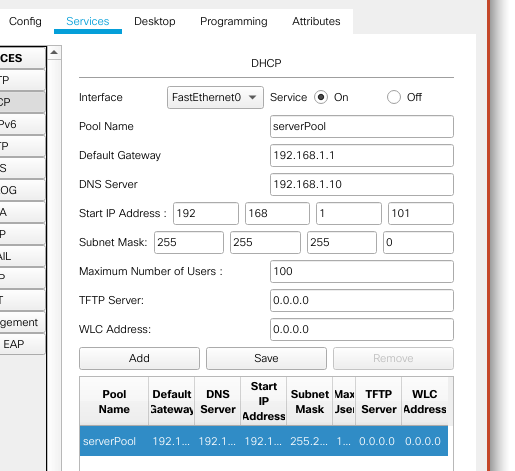


En esta respuesta al cliente aparecen 2 campos en lugar de solo 1, básicamente es “he preguntado a este servidor por tu destino y me ha dado esta dirección”.

IMPORTANTE: Este DNS principal tiene una cache, es decir, almacena durante un tiempo estas direcciones que pregunta. Si yo le vuelvo a solicitar esta web, no va a preguntar, lo tiene en su base de datos y me lo da. Cuando esa cache se vacié (por tiempo o porque la borro a mano), se volverá a hacer el procedimiento.



Podemos añadir un servidor extra DHCP, para que los Pcs se configuren automáticamente con IP y el DNS



Configuramos el rango de Ips y el DNS.





Entonces ahora, funciona exactamente igual que antes el tema de las peticiones y el DNS, pero esta automatizado, no tengo que indicar en cada PC su IP y el DNS al que tienen que conectarse.