**Práctica 8**

**Capa de Red - Ruteo**

**Ruteo**

**2. En las redes IP el ruteo puede configurarse en forma estática o en forma dinámica. Indique ventajas y desventajas de cada método.**

Introducción

El router es el equipo más importante de la red, ya que su principal trabajo es dirigir el tráfico hacia otras rutas directamente conexas y/o redes remotas. Por ende podemos afirmar que los routers dirigen a los paquetes en la red.

Por naturaleza va a tener mas de una interfaz de salida, las cuales, cada una, van a estar conectadas a una red.

Tabla de enrutamiento:

* Función: Contiene la información necesaria para que el router tome las decisiones de reenvío de paquetes hacia la red de destino, utilizando la información de asociación entre la red y el siguiente salto. (tiene direcciones de las redes a las que quiero llegar)
* Ubicación: es un archivo de datos que se encuentra en la memoria del router.
* Conceptos:
  + Destino: lugar a donde quiero llegar.
    - Destination 0000 (ruta por defecto). Si un paquete no tiene una ruta específica, se envía por esta ruta por defecto. Va a terminar llegando a otro router, que repite el procedimiento.
  + Gateway: a donde tengo que enviar el datagrama para llegar a “Destino”.
    - Gateway 0000: indica que estoy en la red a la cual quería llegar, y debo mandar el datagrama al dispositivo final.
    - Interfaz de salida: placa de red por la que sale el paquete. En una PC hay una sola (dada por el proveedor de internet). En un router hay varias.

Rutas en una tabla:

* Rutas conectadas directamente: rutas que aparecen en la tabla porque existe una conexión física entre el router y esa red.
* Rutas estáticas: es una ruta que se configura manualmente y que siempre es la misma hacia un destino.
* Rutas dinámicas: es una ruta que puede variar dependiendo de las variables del protocolo que se configuró.

Enrutamiento estático: Se da cuando un ingeniero de red configura manualmente una ruta de envío de paquetes a la red destino. El mantenimiento de la tabla en estos casos debe hacerse manualmente.

Enrutamiento dinámico: los routers comparten información de la tabla de enrutamiento mediante el uso de los protocolos de enrutamiento dinámico. La tabla de enrutamiento prácticamente se mantiene sola.

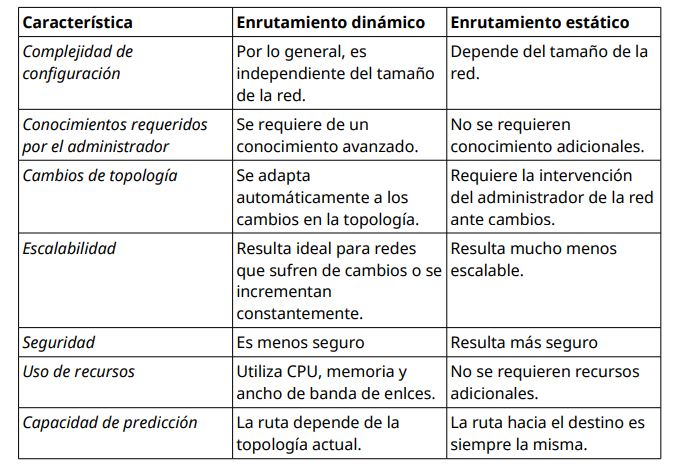
Protocolos de enrutamiento: llenan las tablas de rutas para tomar la decisión de cual seguir.

**Resolución del ejercicio:**

Estática:

* Es viable en redes de una dimensión manejable por un ser humano y donde no haya mucha fluctuación de hosts conectados a la red.
  + No es viable con direcciones móviles. (esto no, es DHCP)
  + Tampoco para direcciones IP temporales. (esto no, es DHCP)
* Ante cada cambio requiere configurar parte (o todos) los dispositivos que comparten

una misma red.

* Tiene un costo de tener un administrador dedicado para esta tarea con los conocimientos necesarios para hacerlo.
* Hay que interactuar con los sistemas de red autónomos para coordinar actualizaciones si hay cambios en las direcciones IP de sus redes públicas.

En resumen, es útil para redes estáticas y de tamaño relativamente reducido.

Dinámica:

Ventajas:

* Escalabilidad.
* Adaptabilidad.

Desventajas:

* Complejidad.

**3. Una máquina conectada a una red pero no a Internet, ¿tiene tabla de ruteo?**

Sí, porque aunque sea una red local se utiliza la tabla de enrutamiento para comunicarse con los demás dispositivos de la red. (por ejemplo, otra pc o una impresora compartida)

Además vimos que en la tabla de enrutamiento hay que tener:

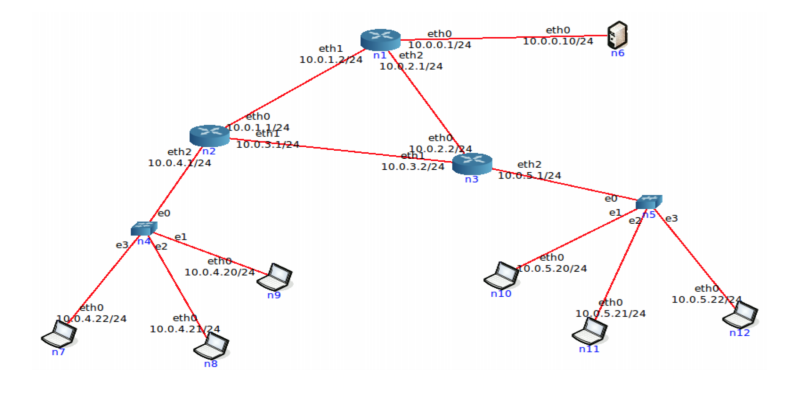
* entrada de tabla con default route, destination 0000 y un gateway particular → a donde mando los paquetes cuyo destino no está en mi tabla de ruteo en ninguna línea
* entrada de la tabla con un detination en particular y el gateway 0000, que indica que ya estoy en la red del destino y no necesito hops intermedios (le mando directamente el datagrama al dispositivo final)

Por ejemplo, a continuación tenemos una tabla de enrutamiento de un host que podría no tener conexión a internet.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Destination | Gateway | Genmask | Flags | Interface |
| 0.0.0.0 | 192.168.5.1 | 0.0.0.0 | UG | eth0 |
| 192.168.5.0 | 0.0.0.0 | 255.255.255.0 | U | eth0 |

**NOTA**: es super importante notar en la parte de destination pongo REDES, no IPS. Me importa saber como llego a ‘x’ red y una vez que llego ahi el router le manda el paquete al host correspondiente.

**4. Utilizando la máquina virtual, se configurará ruteo estático en la red que se muestra en el siguiente gráfico:**

****

**a. Inicie la herramienta CORE y abra el archivo 1-ruteo-estatico.imn.**

**b. Inicie la virtualización de la topología.**

**c. Configure cada uno de los equipos considerando:**

**i. Para entrar a configurar cada equipo (PC o router) debe hacer doble click sobre el mismo, lo cual abre una terminal de comandos.**

**ii. Utilice alguno de los comandos vistos para configurar las direcciones IP de las interfaces según el gráfico. Por ejemplo:**

* **En la PC n6 debe configurar la interfaz eth0 con la IP 10.0.0.10.**
  + hago doble click en n6 y pongo *ifconfig* para ver que dir tiene eth0. Luego, hago *ifconfig eth0 10.0.0.10 netmask 255.255.0.0* para cambiarle la ip, y vuelvo a hacer ifconfig para ver si cambio. NOTA: le cambio la mascara porque dice que es /24 y la anterior era /8
* **En el Router n1 debe configurar la eth0 con la IP 10.0.0.1, la eth1 con la IP 10.0.1.2 y la eth2 con la 10.0.2.1.**
  + hago doble click en n1 y pongo *ifconfig* para ver que dir tiene eth0 eth1 y eth2. Luego, hago *ifconfig eth0 10.0.0.1 netmask 255.255.0.0, ifconfig eth1 10.0.1.2 netmask 255.255.0.0*, y *ifconfig eth2 10.0.2.1* *netmask 255.255.0.0* para cambiarles las ip, y vuelvo a hacer *ifconfig* para ver si cambio.

**iii. Compruebe conectividad utilizando el comando ping.**

Hice un ping desde n1 a n6, de la forma: *root@n1:/tmp/pycore.56201/****n1****.conf#* ***ping 10.0.0.10***

**iv. Configure una ruta por defecto en todas las computadoras.**

Con *netstat -r*  vemos la tabla de ruteo.

Para agregar una ruta por defecto, hacemos *ip route add default via 10.0.0.1.*

*→ root@n6:/tmp/pycore.56201/****n6****.conf# i****p route add default via 10.0.0.1***

*→ root@n6:/tmp/pycore.56201/****n6****.conf#* ***netstat -r***

Al hacer eso, vemos en la tabla se añade la entrada:

Destination Gateway Genmask Flags MSS Window irtt Iface

default 10.0.0.1 0.0.0.0 UG 0 0 0 eth0

10.0.0.0 \* 255.255.0.0 U 0 0 0 eth0

El default va a ser a donde envío los paquetes que no están en mi red local.

**v. Configure en los routers rutas estáticas, considerando:**

* **Router n1 envía todo el tráfico desconocido a Router n2.**

*→ root@n6:/tmp/pycore.56201/****n1****.conf# i****p route add default via 10.0.1.1***

*→ root@n6:/tmp/pycore.56201/****n1****.conf#* ***netstat -r***

Por defecto me conecto desde n1 a n2 (10.0.1.1) por la interfaz de n1 eth1 (10.0.1.2)

* **Router n2 envía todo el tráfico desconocido a Router n3.**

*→ root@n6:/tmp/pycore.56201/****n2****.conf# i****p route add default via 10.0.3.2***

*→ root@n6:/tmp/pycore.56201/****n2****.conf#* ***netstat -r***

Por defecto me conecto desde n2 a n3 (10.0.3.2) por la interfaz de n2 eth1 (10.0.3.1)

* **Router n3 envía todo el tráfico desconocido a Router n1.**

*→ root@n6:/tmp/pycore.56201/****n3****.conf# i****p route add default via 10.0.2.1*** (corresponde a la ip de eth2)

*→ root@n6:/tmp/pycore.56201/****n3****.conf#* ***netstat -r***

Por defecto me conecto desde n3 a n1 (10.0.2.1) por la interfaz de n3 eth0 (10.0.2.2)

NOTA: cuando pongo ‘Por defecto me conecto … ’ no es que lo invente yo, tipo al hacer los comandos se agrego esa linea en la tabla de ruteo con ethX

**vi. Función de ruteo: Un router o una PC puede configurarse para rutear paquetes entre sus placas de red. Lo deseable es que el router tenga esto activado.**

* **Verificar IP\_FORWARD, en los routers y las PCs, obteniendo la configuración con:**

**→ cat /proc/sys/net/ipv4/ip\_forward**

**El valor 0 indica funcionalidad desactivada. 1 indica que está habilitado.**

Forwarding → acción de reenvío; pasar un paquete que me llega de una interfaz a otra

El bit verificado está activado en todos. El hecho de activarlo en una pc sirve para que esa PC sirva como router.

* **Verificar RP\_FILTER. Este parámetro, por seguridad, evita la recepción de paquetes por una interfaz que vengan de una IP de una red que el router no rutearía a través de esa interfaz. Este valor debe deshabilitarse en caso que el enrutamiento circular propuesto cause problemas. Para obtener el valor:**

**→ cat /proc/sys/net/ipv4/conf/all/rp\_filter**

**El valor en 0 deshabilita su funcionalidad. Un 1 lo habilita.**

**Para cambiar el valor:**

**→ echo 0 >/proc/sys/net/ipv4/conf/all/rp\_filter**

Osea si yo NO ruteo a través de vos como gateway para llegar a un destino, no acepto paquetes que vengan de vos, si es que seteo ese flag de rp\_filter en 1. Es como que ‘no te conozco’.

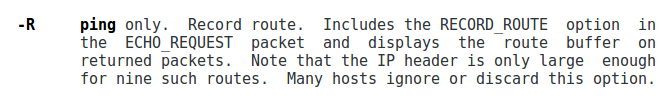
**d. Verifique conectividad entre las PCs, utilizando los comandos ping, traceroute y ping -nR.**

**Mientras realiza ping desde una PC, capture paquetes en un router intermedio y verifique qué paquetes pasan por la interfaz. Por ejemplo, mientras una PC hace ping a otra, analice los paquetes que se visualizan en eth0 y en eth1 de algún router por el que parte o todo el tráfico pasa. La captura de paquetes puede hacerse con el comando tcpdump -i interfaz.**

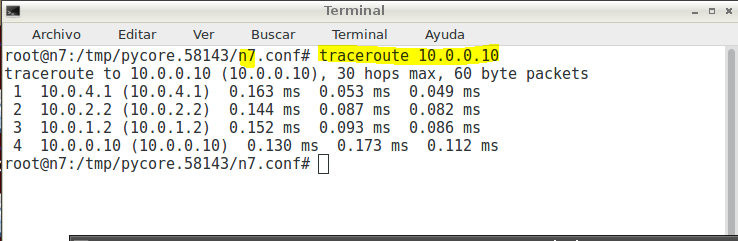
**Alternativamente, botón derecho sobre el router o host ->tcpdump -><interface de red>.**

PING -nR



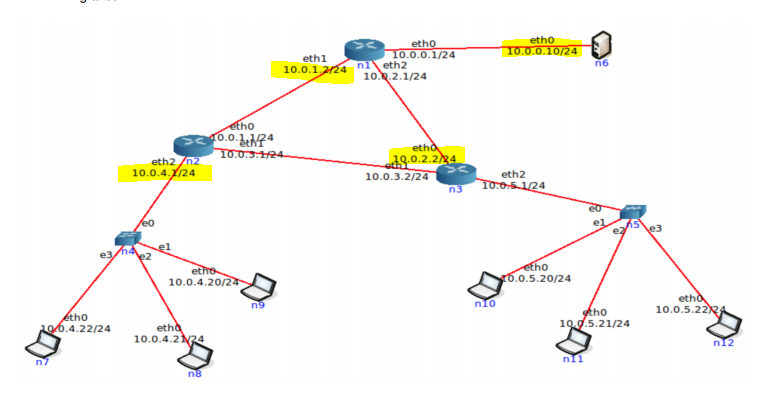


TRACEROUTE DE N7 A N6



n2 - n3 - n1 - n6

No es que lo podes ver asi nomas en el dibujito para saber el traceroute, tenes que ver la interfaz de salida en cada tabla de ruteo para el destination en particular, en este caso ir a n6 cae en la linea destination 0000, y en cada nodo se especifica el gateway (a donde salta, a que interfaz de ‘entrada’ de un router) y por que interfaz tuya de salida te vas.



* De n7 va a 10.0.4.1/24 tipo porque lo veo en el dibujo. → ACA PONGO A DONDE SE CONECTA EL 1º CON EL 2º
* De n2 a n3 va por 10.0.3.1/24 pero NO se ve en el traceroute.(interfaz de salida a un destination cualquiera)
* De n3 a n1 va por 10.0.2.2/24 y se ve en el traceroute. (interfaz de salida a un destination cualquiera) → POR QUE INTERFAZ SALGO PARA IR AL SIGUIENTE
* De n1 a n6 va por 10.0.1.2/24 y se ve en el tracerout. (interfaz de salida a un destination cualquiera) → POR QUE INTERFAZ SALGO PARA IR AL SIGUIENTE
* Se ve el destino → IP DE N6

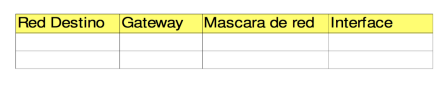
Pongo como llego de n7 a n2 pero no como llego de n2 a n3 ni de n3 a n1 ni de n1 a n6 ….

**NOTA**: Pensar la idea de TTL en paquetes. El nº ese que se decrementa en cada salto, en cada pasaje por router para no entrar en loop si no se puede alcanzar una red. Que pasa, si n7 mandara un paquete a una red que no existe (8.8.8.8), se irian pasando el paquete entre n1 n2 y n3 infinitamente. Con ttl, cuando el ttl del paquete llega a 0 se descarta y tipo se avisa al host origen.

Si el ttl fuera 1, cuando llega a n2 responde este por IMCP ttl exceeded, por la interfaz 10.0.4.1 a n7. Luego, si el ttl es 2, n3 le manda a n1 por 10.0.2.2 ttl exceeded, y n2 a n7. Despuès, si el ttl es 3, cuando el paquete llega a n1, este le manda ttl exceded a n2 por 10.0.1.2 y n2 a n7.

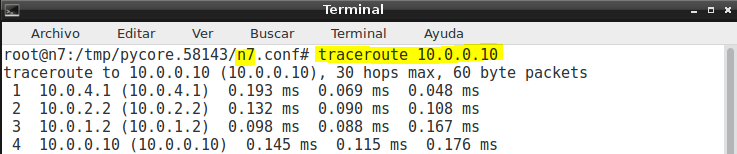
Es como que marco las interfaces de salida en el trace route.

**e. Relevamiento: Utilizando el comando route -n, netstat -nr o ip route ls releve la configuración de las tablas de rutas tanto de los routers como de las PCs completando para cada dispositivo una tabla como la siguiente:**

****

**Si la estación PC n7 le envía un ping a la estación PC n6:**

**• ¿Cuál es el camino por el que viaja el requerimiento?**



Secuencia de nodos: n2, n3, n1, n6

**• ¿Cuál es el camino por el que viaja la respuesta?**

n1 - n2 - n7

**Mantenimiento de red - Suponiendo que en el Router n1 se agregó una interfaz de red con la dirección IP 163.10.10.1/24:**

**• Identifique los cambios que serían necesarios aplicar al ruteo estático para que todas las PCs puedan comunicarse con hosts en la nueva red.**

Habría que configurar el router n1 para que tenga una entrada en la tabla de enrutamiento para esa red.

Habría que configurar el resto de los routers (2 y 3) para que tengan una entrada en la tabla de enrutamiento con esa red (si es que no se resuelve por default) y se resuelva mediante el router n1.

**• En base a lo anterior, ¿qué puede decir respecto del mantenimiento del ruteo en una**

**red que tiene más de 20 routers?**

Con la configuración estática sería difícil y costoso de mantener, porque, además de la complejidad de conocer las IPs de todas las redes, hay que configurar cada router por cada nueva interfaz de red. Resultaría poco escalable si hubiera 20 routers.

**ICMP y RUTEO.**

**• Desde la PC n6, realice un ping a la dirección IP 5.5.5.5. ¿Qué indica el mensaje de error recibido? ¿Quién lo envía?**

Si se realiza un ping a la dirección 5.5.5.5, se va a recibir un mensaje "Time to Live

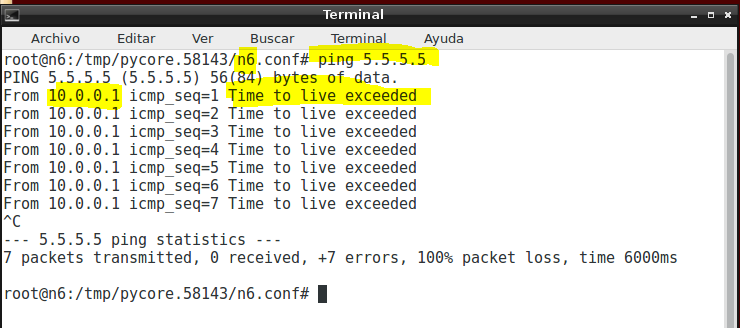
exceeded", esto significa que posiblemente el paquete quedó en loop infinito entre

los routers y como no pertenecía a ninguna de las subredes directamente conectadas

a estos routers o en sus tablas de ruteo, quedó dando vueltas hasta que acabó

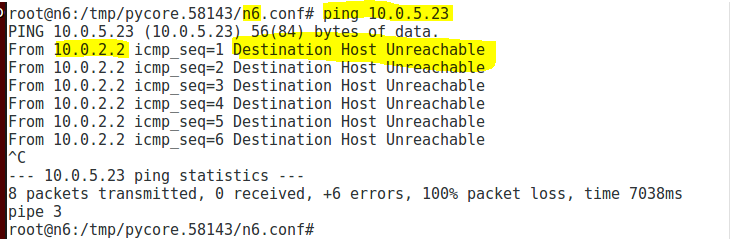
"muriendo".

Sabemos cada interfaz de un router, cada ‘ethX’ representa a una red digamos.



Indica que se agotó el TTL y se lo envía el router n1 (a través de la interface 10.0.0.1, lo vemos en el from)

**• Desde la PC n6, realice un ping a la dirección IP 10.0.5.23. ¿Qué indica el mensaje de error recibido? ¿Quién lo envía?**



Que no se encuentra el host y se lo envía el router n3 a través de la interfaz 10.0.2.2.

Si se realiza un ping a un host desconocido dentro de una red conocida, el que responderá con host unreachable en ICMP será un router conectado directamente a la red. En cambio, si se hace un ping a una subred desconocida, el paquete se quedará dando vueltas hasta que "muera" (es lo que pasa en el caso anterior).

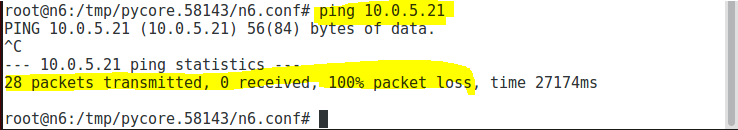
El tema aca es que el nodo 3 tiene la interfaz con ip 10.0.5.1/24. Se esta representando en ella a la red 10.0.5.1/24. La ip a la que quiere llegar n6 corresponde a esa red, ya que es 10.0.5.23 . Entonces, el nodo 1 se comunica con el 2, el 2 con el 3, y es el nodo 3 que seria como el q ‘esta conectado directamente a esa red’ quien busca esa ip y como no existe dice ‘Host Unreachable’.

**• Desde la PC n6, haga un ping a la IP 10.0.5.21 y verifique que el ping responda correctamente.**

Llega porque tipo n3 maneja esa red y hay una pc conectada directamente a n3 con la ip 10.0.5.21

**NOTA**: NO es que se confundieron al escribir el enunciado y a la ip no le pusieron la mask; NO VA!! Lo que va a pasar es que se va a buscar en la tabla de ruteo aplicandole qué mask obtengo la misma dir de red que en la ip dada, osea busco con que mask hay coincidencia,

**Luego, en el router n1 habilite RP\_FILTER y vuelva a intentar hacer el ping. ¿Qué ocurre?**



100% packets loss. En realidad llegan los paquetes al destino pero nunca llegan las respuestas ya que n1 al no tener en una entrada de su tabla de ruteo una gateway que sea igual a 10.0.5.21, no aceptara paquetes que provengan de la red que representa esa ip. Es por el bit de seguridad que está activo.

**Agregue en n1 una ruta a la red 10.0.5.0 a través de la 10.0.2.2 y vuelva**

**a hacer el ping. ¿Qué ocurre ahora? ¿Puede explicar por qué?**

Ahora si se dejan pasar los paquetes ya que al tener en su tabla la red 10.0.5.0 n1 ‘la conoce’ y acepta la respuesta que viene de esta.

El paquete ICMP que manda ping sigue la ruta n1 -> n2 -> n3. Pero la respuesta es n3 -> n1, como ahora n1 enrutaría (por la nueva entrada) el paquete directamente por n3 (n1 -> n3) el RP\_FILTER deja pasar el paquete.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**DHCP y NAT**

**Introducción:**

**↠** Qué es DHCP?

DHCP (Dynamic Host Configuration Protocol, protocolo de configuración de host dinámico) es un protocolo que permite que un equipo conectado a una red pueda obtener su configuración (principalmente, su configuración de red) en forma dinámica (es decir, sin una intervención especial). Solo tienes que especificarle al equipo, mediante DHCP, que encuentre una dirección IP de manera independiente. El objetivo principal es simplificar la administración de la red.

**↠** Cómo funciona el protocolo DHCP?

Primero, se necesita un servidor DHCP que distribuya las direcciones IP. Este equipo será la base para todas las solicitudes DHCP por lo cual debe tener una dirección IP fija. Por lo tanto, en una red puede tener solo un equipo con una dirección IP fija: el servidor DHCP.

El sistema básico de comunicación es BOOTP (con la trama UDP). Cuando un equipo se inicia no tiene información sobre su configuración de red y no hay nada especial que el usuario deba hacer para obtener una dirección IP. Para esto, la técnica que se usa es la transmisión: para encontrar y comunicarse con un servidor DHCP, el equipo simplemente enviará un paquete especial de transmisión (transmisión en 255.255.255.255 con información adicional como el tipo de solicitud, los puertos de conexión, etc.) a través de la red local. Cuando el DHCP recibe el paquete de transmisión, contestará con otro paquete de transmisión (no olvide que el cliente no tiene una dirección IP y, por lo tanto, no es posible conectar directamente con él) que contiene toda la información solicitada por el cliente.

Se podría suponer que un único paquete es suficiente para que el protocolo funcione. En realidad, hay varios tipos de paquetes DHCP que pueden emitirse tanto desde el cliente hacia el servidor o servidores, como desde los servidores hacia un cliente:

* DHCP**DISCOVER** (para ubicar servidores DHCP disponibles)
* DHCP**OFFER** (respuesta del servidor a un paquete DHCPDISCOVER, que contiene los parámetros iniciales)
* DHCP**REQUEST** (Si tenés más de un servidor DHCP, todos contestan. El cliente con el request elige con cual se queda.)
* DHCP**ACK** (respuesta del servidor que contiene los parámetros y la dirección IP del cliente)
* DHCP**NAK** (respuesta del servidor para indicarle al cliente que su concesión ha vencido o si el cliente anuncia una configuración de red errónea)
* DHCP**DECLINE** (el cliente le anuncia al servidor que la dirección ya está en uso)
* DHCP**RELEASE** (el cliente libera su dirección IP)
* DHCP**INFORM** (el cliente solicita parámetros locales, ya tiene su dirección IP)

El primer paquete emitido por el cliente es un paquete del tipo DHCPDISCOVER. El servidor responde con un paquete DHCPOFFER, fundamentalmente para luego (con DHCPACK) enviarle una dirección IP al cliente. El cliente establece su configuración y luego realiza un DHCPREQUEST para validar su dirección IP (una solicitud de transmisión ya que DHCPOFFER no contiene la dirección IP). El servidor simplemente responde con un DHCPACK con la dirección IP para confirmar la asignación. Normalmente, esto es suficiente para que el cliente obtenga una configuración de red efectiva, pero puede tardar más o menos en función de que el cliente acepte o no la dirección IP.

**↠** Qué es una 'concesión' en DHCP

Para optimizar los recursos de red, las direcciones IP se asignan con una fecha de inicio y de vencimiento para su validez. Esto es lo que se conoce como "concesión". Un cliente que detecta que su concesión está a punto de vencer, puede solicitarle al servidor una extensión de la misma por medio de un DHCPREQUEST. Del mismo modo, cuando el servidor detecta que una concesión va a vencer, enviará un DCHPNAK para consultarle al cliente si desea extenderla. Si el servidor no recibe una respuesta válida, convertirá la dirección IP en una dirección disponible.

Esta es la efectividad de DHCP: se puede optimizar la asignación de direcciones IP planificando la duración de las concesiones. El problema es que si no se liberan direcciones, en un momento determinado no se podrá cumplir con nuevas solicitudes DHCP debido a que faltarán direcciones que puedan distribuirse.

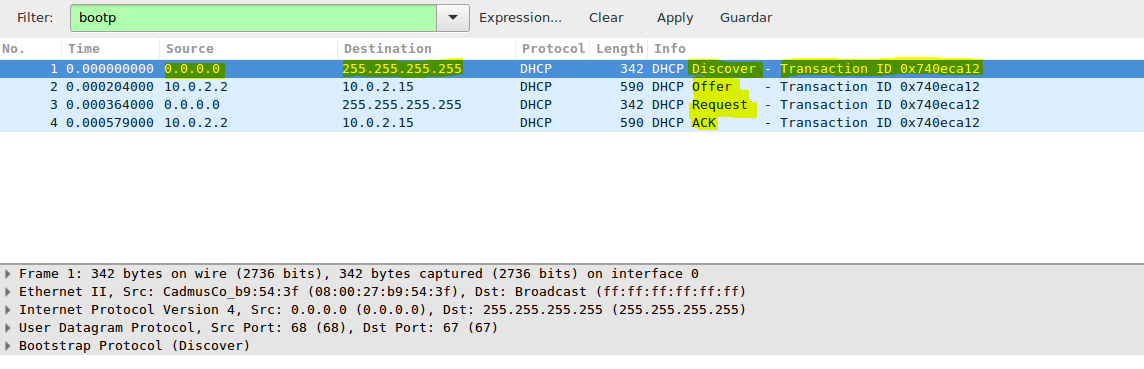
En una red en la cual muchos equipos se conectan y desconectan permanentemente (redes de escuelas o de oficinas de ventas, por ejemplo), es aconsejable ofrecer concesiones por períodos cortos. En cambio, para una red compuesta principalmente por equipos fijos que se reinician rara vez, las concesiones por períodos largos son más que suficientes. No se olvide que DHCP trabaja principalmente por transmisión y que puede ocupar ancho de banda en redes pequeñas con alta demanda.

**↠** Una de las principales innovaciones en esta versión es la posibilidad de actualizar en forma dinámica un DNS de acuerdo a las direcciones IP suministradas por el servidor DHCP.

**5. Con la máquina virtual con acceso a Internet realice las siguientes observaciones respecto de la autoconfiguración IP vía DHCP:**

**a. Inicie una captura de tráfico Wireshark utilizando el filtro bootp para visualizar únicamente tráfico de DHCP.**

**b. En una terminal de root, ejecute el comando sudo /sbin/dhclient eth0 y analice el intercambio de paquetes capturado.**

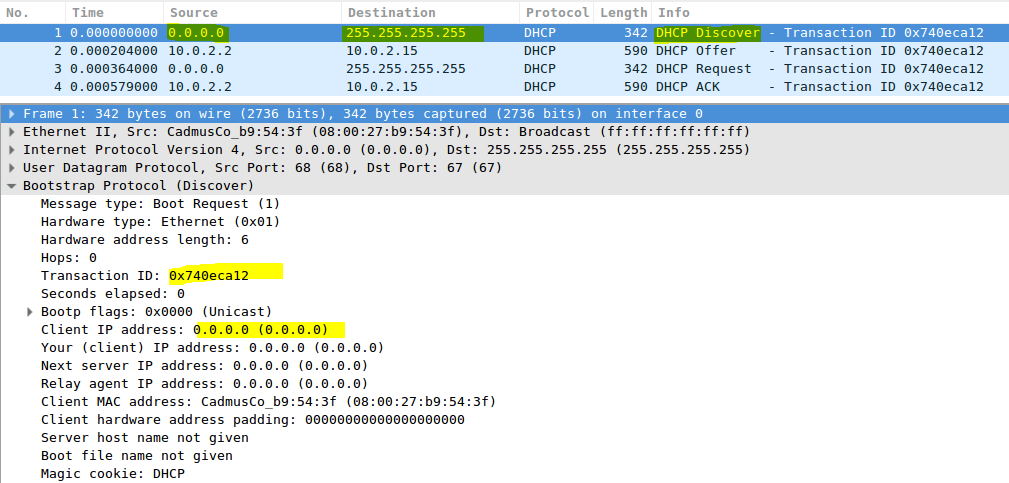


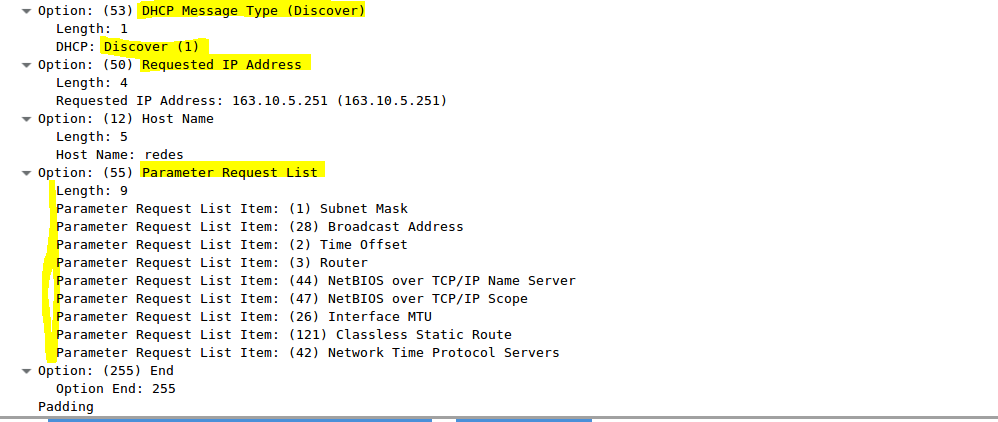
Se intercambian 4 paquetes con el mismo **ID de transacción**

1. Del cliente se ejecuta un mensaje por broadcast (IP 255.255.255.255) con un tipo **DHCP** **Discover**
   1. Puerto de origen **68 (CLIENTE)**
   2. Puerto destino **67 (SERVIDOR)**
2. Del servidor DHCP se responde un mensaje de tipo **DHCP** **Offer**.
3. El cliente vuelve a mandar un broadcast con un mensaje tipo **DHCP** **Request**
4. El servidor DHCP responde con un **DHCP** **ACK**

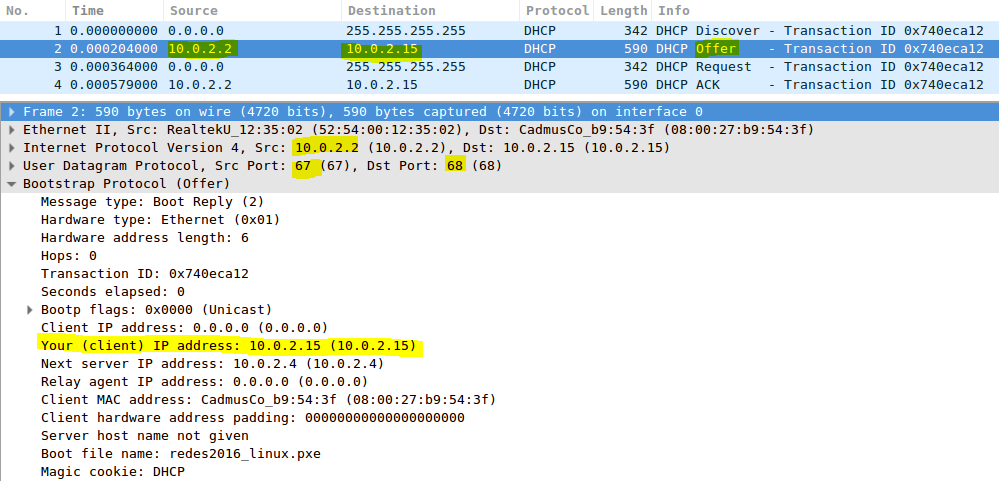
**NOTA**: La dirección f.f.f.f es un broadcast global. La PC que quiere una dirección IP (pública o privada) hace un broadcast con esta dirección porque no sabe a qué red pertenece. Cuando tenga la red, directamente hace el broadcast en su red: id\_de\_red + [todos unos]. El servidor DHCP responde a la PC que pidió (ya entrando en capa de enlace) mediante su dirección MAC (la cual es única).

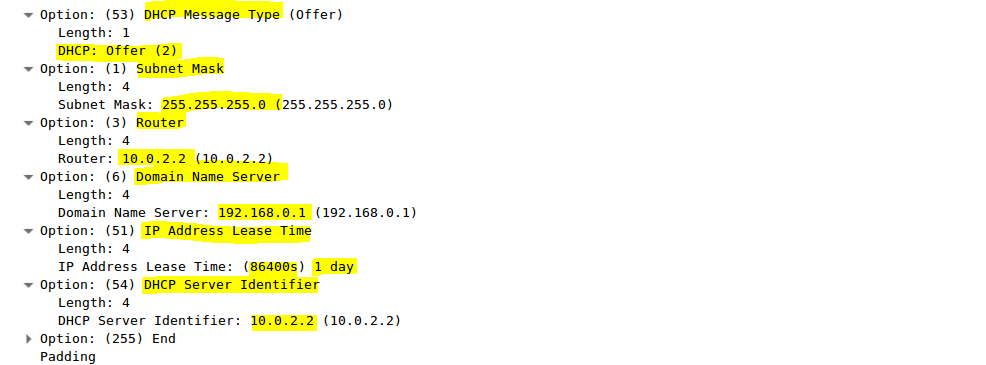
1. DISCOVER





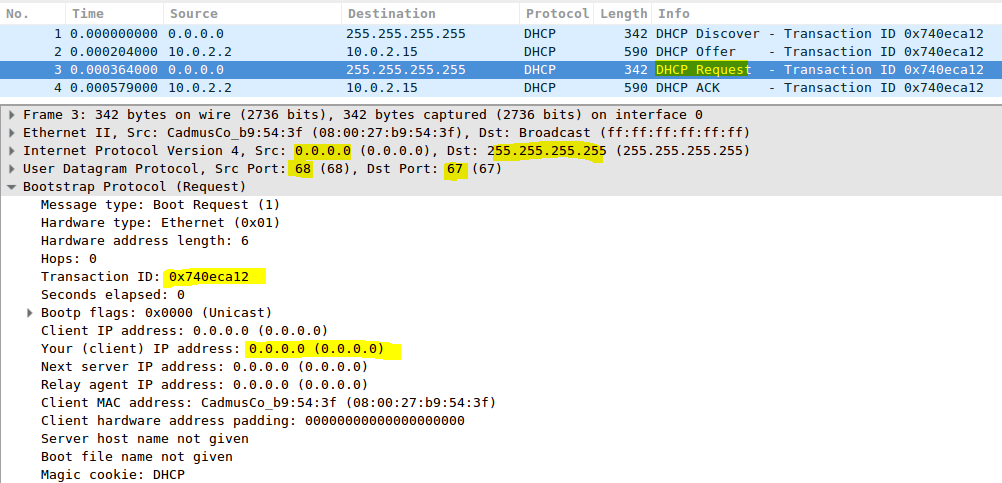
2) OFFER

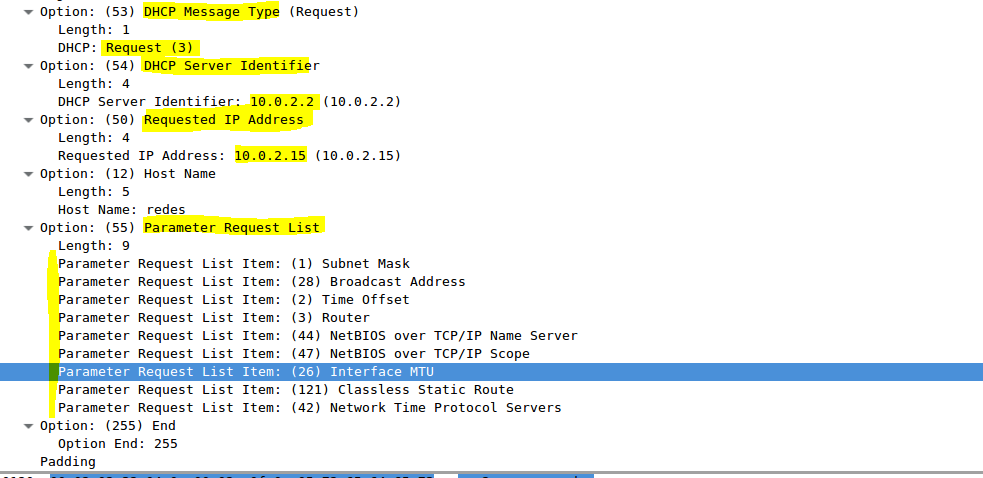




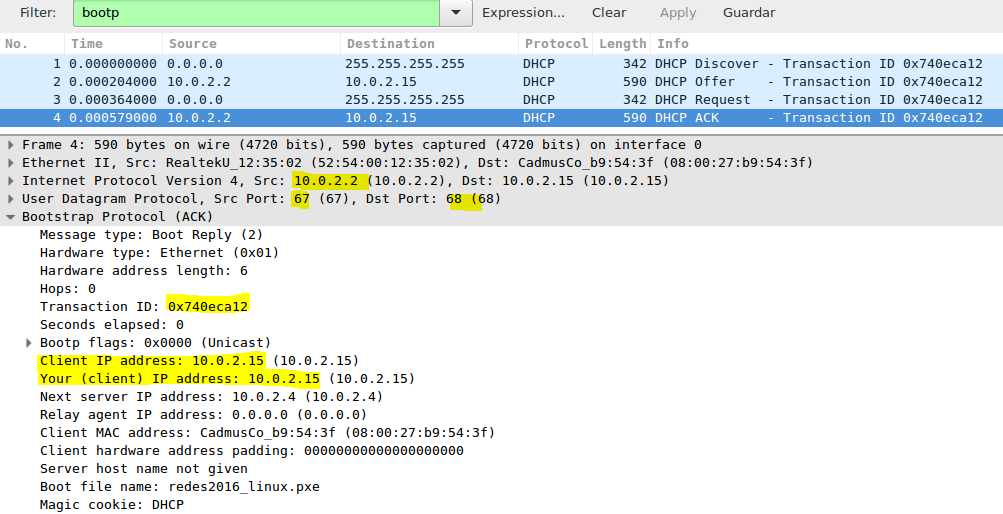
El server le dice la ip que le ofrece pero todavía el cliente tiene que aceptarla.

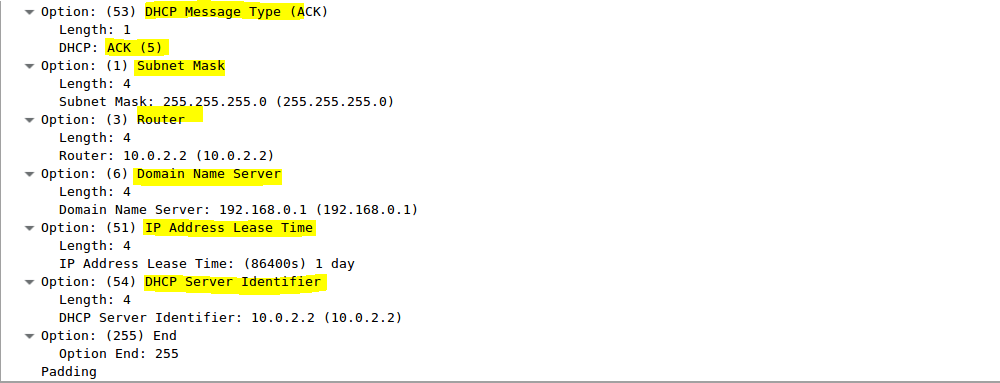
3) REQUEST





4) ACK





Que observamos?

- ID de transacción

- **Dirección IP del cliente**

- Tipo de mensaje DHCP (**5- ACK**)

- La máscara de subred

- La dirección del router (próximo salto); **el default gateway**

- Dirección IP del servidor DNS

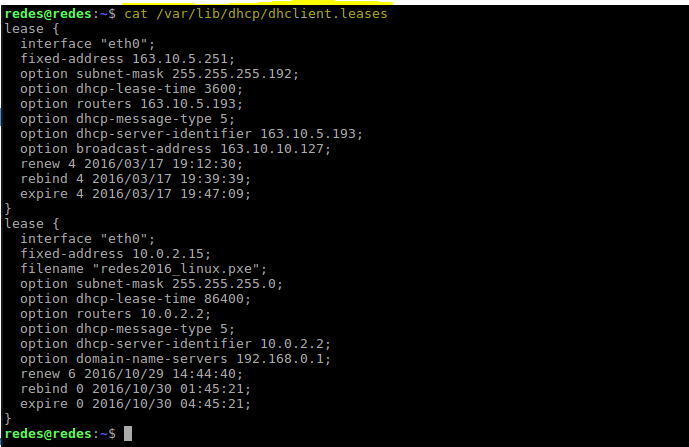
- Tiempo de vida de la IP

- Identificador del servidor DHCP

Aparentemente, el servidor, con el ACK devuelve casi la misma información que en el offer ...

**c. Analice la información registrada en el archivo /var/lib/dhcp/dhclient.leases, ¿cuál parece su función?**

En el caso de GNU/Linux, en el directorio /var/lib/dhcp se guardan los ficheros en los que el cliente DHCP guarda información sobre las concesiones.

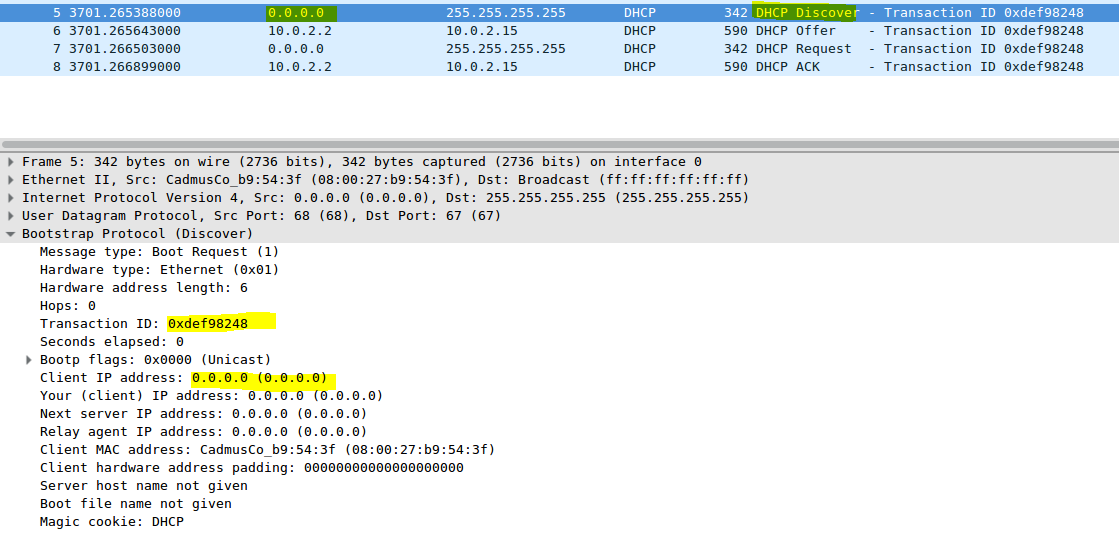


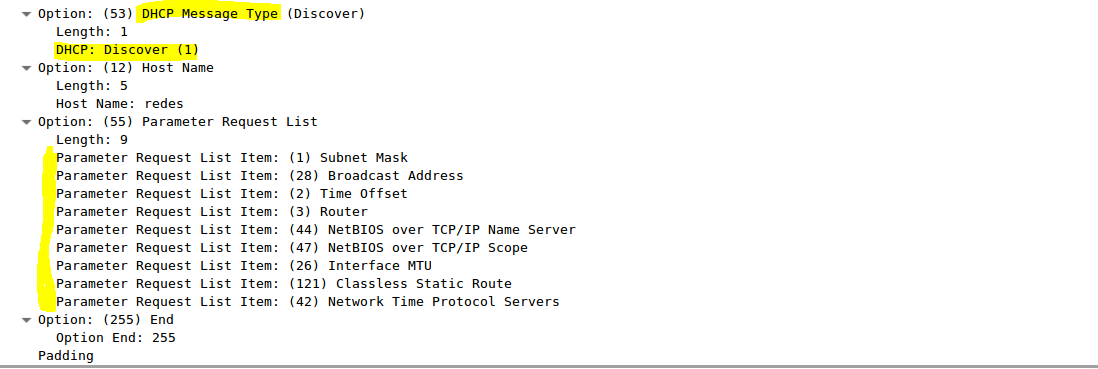
Contiene información cómo:

* Interface
* Dirección IP
* Máscara de subred
* Tiempo de alocamiento de la dirección IP
* Routers (próximo salto?) CONSULTAR, SI, default gateway
* Tipo? CONSULTAR: No importa
* Identificador del servidor DHCP
* Dirección del servidor de DNS
* Time Stamp de actualización y expiración de la información obtenida (o la IP?) CONSULTAR,SI

**d. Ejecute el siguiente comando para eliminar información temporal asignada por el servidor DHCP: rm /var/lib/dhcp/dhclient.leases**

**e. En una terminal de root, vuelva a ejecutar el comando sudo /sbin/dhclient eth0 y analice el intercambio de paquetes capturado nuevamente ¿a que se debió la diferencia con lo observado en el punto “b”?**





* El cliente no mandó el “Requested IP Address” (debería haber sido una de las opciones) que mandó en el Discover previo.
* Al borrar la información previa (del archivo /var/lib/dhcp/dhclient.leases), el cliente no mandó la IP requerida (que tenía en el historial de ese archivo) Esto lo hace para mantener la misma IP una vez que expira. Sirve en situaciones en las cuales no es deseable que la dirección IP cambie “repentinamente” (cuando se expira el tiempo). Esto por ejemplo sirve en caso de usar chats sobre TCP, donde es deseable mantener las conexiones.

**f. Tanto en “b” como en “e”, ¿qué información es brindada al host que realiza la petición DHCP, además de la dirección IP que tiene que utilizar?**

- La máscara de subred

- La dirección del router (próximo salto)

- Dirección IP del servidor DNS

- Duración de la dirección (en segundos)

- Identificador del servidor DHCP

Podría devolver muchas cosas más.

**6. ¿Qué es NAT y para qué sirve? De un ejemplo de su uso y analice cómo funcionaría en ese entorno.**

**Ayuda: analizar el servicio de Internet hogareño en el cual varios dispositivos usan Internet simultáneamente.**

Un router NAT no parece un router a ojos del mundo exterior. En su lugar, el router NAT se comporta de cara al exterior como un único dispositivo con una dirección IP única. Todo el tráfico que sale del router doméstico hacia Internet tiene una dirección IP de origen igual, y todo el tráfico que entra en él tienen que tener la misma dirección de destino única. En resumen, el router NAT oculta los detalles de la red doméstica al mundo exterior. (Como nota al margen, posiblemente se esté preguntando dónde obtienen las computadoras de la red doméstica sus direcciones y dónde obtiene el router su dirección IP única, A menudo, la respuesta a ambas preguntas es la misma: ¡DHCP! El router obtiene su dirección del servidor DHCP del ISP *Internet Service Provider* y el router ejecuta un servidor DHCP para proporcionar direcciones a las computadoras, dentro del espacio de direcciones de la red doméstica controlada por el router NAT-DHCP).

Si todos los datagramas que llegan al router NAT procedentes de la WAN tienen la misma dirección IP de destino (específicamente, la de interfaz WAN del router NAT), entonces ¿Cómo sabe el router a qué host interno debería reenviar un datagrama dado? El truco consiste en utilizar una tabla de traducciones NAT almacenada en el router NAT, e incluir los números de puerto, así como las direcciones IP en las entradas de la tabla.

Suponga que un usuario de la red doméstica que utiliza el host con la dirección 10.0.0.1 solicita una página web almacenada en un servidor web (puerto 80) con la dirección IP 128.119.40.186. El host 10.0.0.1, asigna el número de puerto de origen (arbitrario) 3345 y envía el datagrama a la LAN. El router NAT recibe el datagrama, genera un nuevo número de puerto de origen, 5001, para el datagrama, sustituye la dirección IP de origen por su dirección IP de la red WAN 138.76.29.7, y sustituye el número de puerto de origen original 3345 por el nuevo número de puerto de origen 5001. Al generar un nuevo número de puerto de origen, el router NAT puede seleccionar cualquier número de puerto de origen que actualmente no se encuentre en la tabla de traducciones NAT. (Observe que, puesto que la longitud del campo número de puerto es de 16 bits, el protocolo NAT puede dar soporte a 60.000 conexiones simultáneas utilizando la única dirección IP WAN del router). En el router, NAT también añade una entrada a su tabla de traducciones. El servidor web, que afortunadamente no es consciente de que el datagrama entrante que contiene la solicitud HTTP ha sido manipulado por el router NAT, responde con un datagrama cuya dirección de destino es la dirección IP del router NAT y cuyo número de puerto de destino es 5001. Cuando este datagrama llega al router NAT, éste indexa la tabla de traducciones NAT utilizando la dirección IP de destino (138.76.29.7) y el número de puerto de destino (5001) para obtener la dirección IP (10.0.0.1) y el número de puerto de destino (3345) apropiados para el navegador de la red doméstica.

Tabla de traducciones NAT

|  |  |
| --- | --- |
| **Lado WAN** | **Lado LAN** |
| **138.76.29.7, 5001** | **10.0.0.1,3345** |

A continuación, el router reescribe la dirección de destino y el número de puerto de destino del datagrama y lo reenvía a la red doméstica.

¿Para qué sirve?

* Sirve para ahorrar direcciones IPv4 públicas.
* Para transformar direcciones privadas en direcciones públicas. De esta forma es posible conectar un host de una red privada con otro host de otra red privada (a través del router NAT).
* Se puede utilizar el router como intermediario (da una sensación de “seguridad”, aunque no siempre es verdad).

**7. ¿Qué especifica la RFC 1918 y cómo se relaciona con NAT?**

La RFC-1918 determina que bloques de IP privadas le corresponde a cada clase de red.

La "Autoridad de Números Asignados en Internet", Internet Assigned Numbers Authority (IANA), ha reservado los tres siguientes bloques de direcciones IP para el uso en internets privadas:

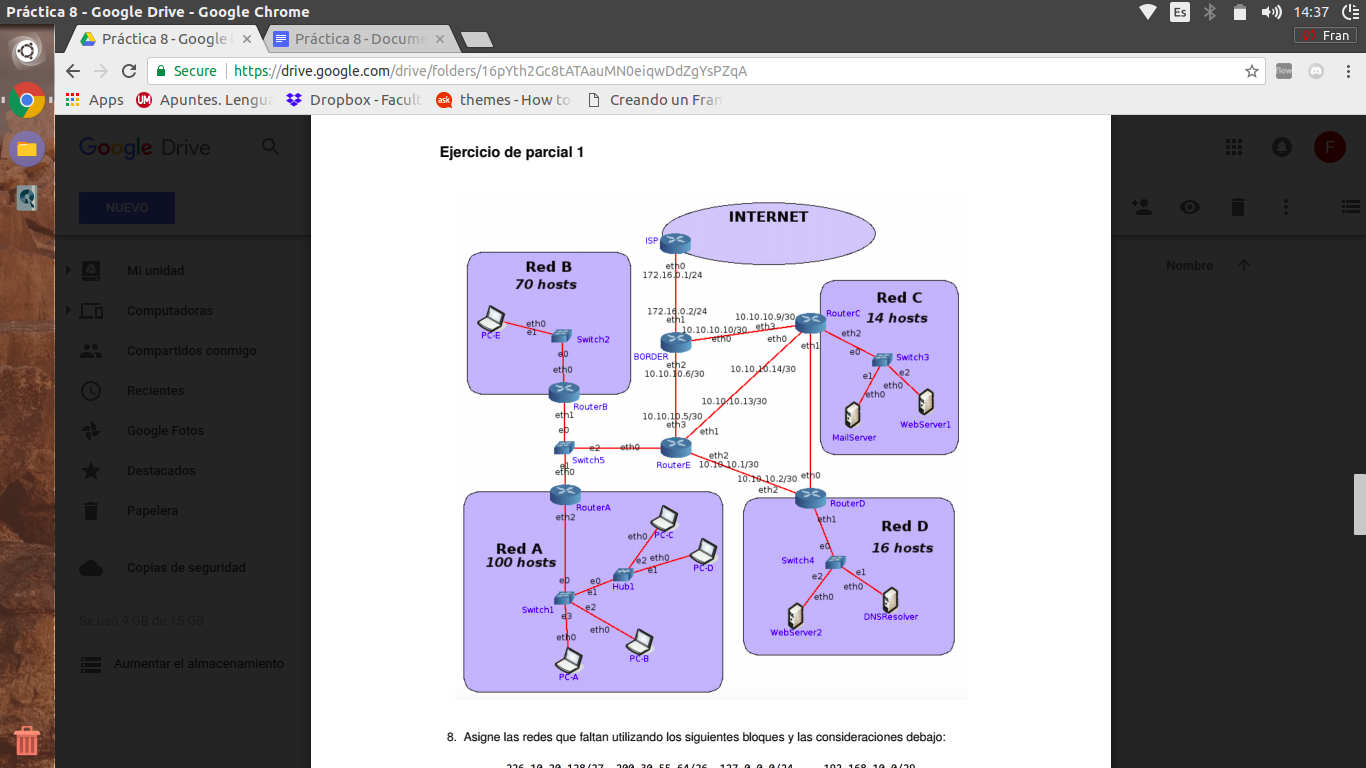
10.0.0.0 - 10.255.255.255 (prefijo 10/8)

172.16.0.0 - 172.31.255.255 (prefijo 172.16/12)

192.168.0.0 - 192.168.255.255 (prefijo 192.168/16)

Se relaciona con NAT, porque este, permite la traslación entre direcciones IP privadas a direcciones públicas. Permitiendo esto, conectar (de forma indirecta, a través de routers NAT) hosts de redes privadas o públicas con hosts de otras redes privadas o públicas. Además de reutilizar IPs públicas partiendo de direcciones IP privadas.

**Ejercicio de parcial 1**



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

eJERCICIO 2

255.255.254.0/23 mascara

necesito 8 bits = 2^8 = 256 - 2 = 254.

me queda 1 bit para redes = puedo referenciar 2 subredes

Para la red B: 160.22.0.0/24 → como ya la asigne no la puedo particionar.

160.22.1.0/24 → se particionan las libres

necesito 7 bits para referenciar mas de 100. 2^7 = 128 - 2 = 126

160.22.1.0/25 → para la red dmz (sobran 26 direcciones)

160.22.1.128/25 → (128 es el bit mas significativo del último octeto). → se divide devuelta

necesito 6 bits para referenciar a mas de 60. 2^6 = 64 - 2 = 62 (sobran 2)

160.22.1.128/26 -> para red A

168.22.1.192/26 -> la divido devuelta

necesito 5 bits para mas de 20. 2^5 = 32 - 2 = 30 (sobran 10)

168.22.1.192/27 → para la red D

168.22.1.224/27 → libre

Inciso 2.

168.22.1.224/27 → libre

no es necesario submarizar xq tengo una sola libre.

Inciso 3.

Le tengo que asignar a los 5 routers

10.0.0.0/8

Necesito 2 bits para host xq 2^2 = 4 - 2 = 2

10.0.0.0/30 → la voy a usar para la interfaz entre R1 y R3

10.0.0.4/30 → la voy a usar para la interfaz entre r1 y r2

10.0.0.8/30 → la voy a usar para la interfaz entre r3 y r2

10.0.0.16/30

10.0.0.32/30

R1 - ETH0 → 10.0.0.1/30

R3 - ETH0 → 10.0.0.2/30

R1 - eth1 → 10.0.0.5/30

r2 - eth0 → 10.0.0.6/30

r3 - eh2 → 10.0.0.9/30

r2 - eth2 → 10.0.0.10/30