****

**Redes de Computadoras**

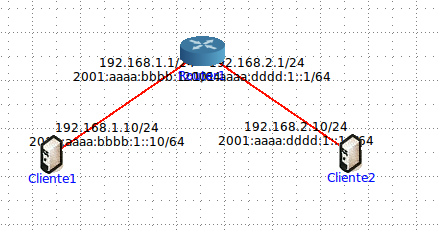
**Práctico 1: Tráfico en capa de enlace relacionado con IPv4 e IPv6.**

**Alumnos:**

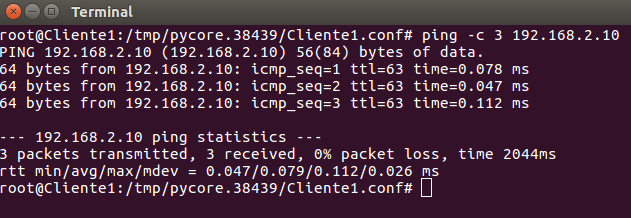
* **Heredia, Marco.**
* **Yepez Hinostroza, Franz.**

**Ejercicio 1: Tráfico IPv4 e IPv6 con CORE**

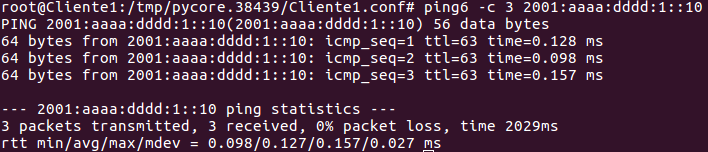
1)



2)

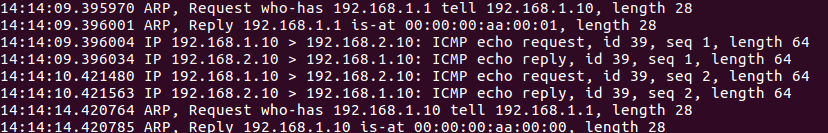


3)

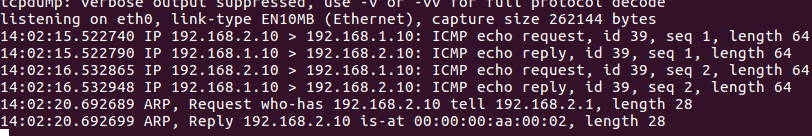


4)

Para el caso de ping desde el cliente1 a cliente2



Para el caso de ping desde cliente2 a cliente1



4.1) Hay envío de paquetes ARP entre el cliente1 y la interfaz 1 del router, donde en un mensaje de solicitud ARP, el cliente1 pregunta mediante un mensaje de broadcast cuál es la dirección MAC de la interfaz del router, cuya dirección ip la tiene como default gateway.Aunque todos los conectados a esa red reciben la solicitud, sólo el router responde, de forma unicast, enviando la dirección MAC de la interfaz del router que está en la misma red dentro de un mensaje de respuesta ARP.

Por otro lado, en la segunda red se observa que no fue necesario enviar solicitudes de ARP ya que se guardaron las direcciones MAC al momento de hacer el ping de cliente1 a cliente2.

4.2) Se observan que en el caso de ping desde cliente1 a cliente2, la dirección ip de origen es la 192.168.1.10, y la 192.168.2.10 como destino, para el caso del request.

Para el caso de reply, conmutan los valores de de destino y origen.

La dirección ip tanto de destino como origen no cambia durante el envío del paquete desde un cliente a otro.

Caso análogo pasa con el ping desde cliente2 a cliente1.

4.3) El router conoce a ambas redes ya que tiene definida para una interfaz una dirección ip correspondiente a cada red, por lo tanto, el router está directamente conectada a ambas. Cuando le llega un paquete dirigido a una de esas redes, se fija en la tabla de ruteo propia y lo envía por la interfaz correspondiente a la red de destino.

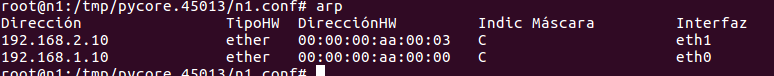
4.4) Ya que cada red cuenta con un único host, no es necesario compartir el medio.

4.5) La tabla ARP del host de origen contiene la ip de la interfaz 1 del router con su respectiva dirección MAC, el índice de la máscara y el tipo de conexión que en este caso es mediante ethernet, 

4.6) La tabla ARP del host de destino es similar a la de origen, contiene la ip de la interfaz 2 del router con su MAC, el índice de la máscara y el tipo de conexión que en este caso es mediante ethernet.

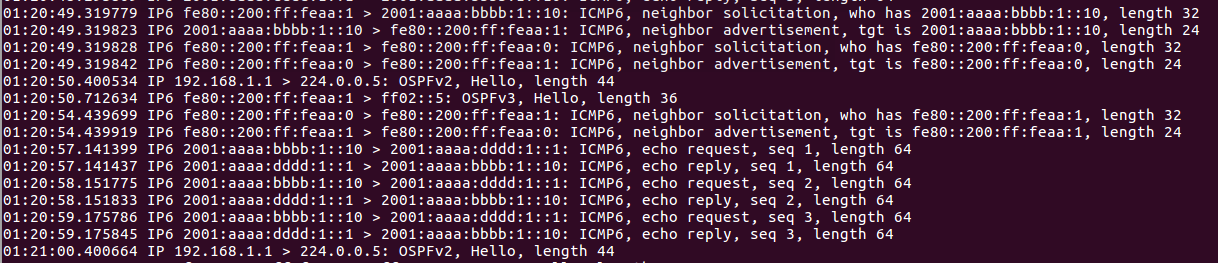


4.7) La tabla ARP del router contiene las direcciones ip de los hosts con sus respectivas direcciones MAC, el índice del tipo de máscara e indica la interfaz física donde está conectado cada host.



4.8) Las direcciones de broadcast en IPv4, son direcciones ip especiales utilizadas para la difusión de datos dentro de una red, donde un paquete enviado por el emisor llegará a todos los hosts de la misma red. Estas direcciones tienen siempre el valor más alto del rango de la red. Se suelen utilizar en los algoritmos de enrutamiento para descubrir vecinos y verificar si los mismos siguen conectados o no.

4.9) Las direcciones multicast en IPv4, son direcciones donde se pueden enviar datos a múltiples hosts de múltiples redes. No es obligatorio que una red tenga una dirección multicast en IPv4. Se podría llegar a usar en videoconferencias y en transmisiones de radio y televisión por internet o para OSPF.

5) Analizar el tráfico con “tcpdump” sobre el host 1 podemos ver: 

Si lo analizamos en el host2 vemos lo siguiente:



5.1) Observamos 3 comunicaciones:

* Neighbor Solicitation.
* Neighbor Advertisement.
* Router Solicitation.

5.2) No, NDP no reemplaza a ARP.

5.3) ARP es un protocolo de IPv4, mientras que NDP es un protocolo de IPv6. NDP incluye las funcionalidades de ARP pero ademas agrega funcionalidades como ICMP Router Discovery y ICMP Redirect que no estan incluidas en ARP.

NDP es utilizado para configuración automática de SLAAC.

5.4) Las funcionalidades de NDP son:

* Resolución de direcciones. El principio es muy similar al del protocolo ARP de IPv4. La principal diferencia consiste en el uso de mensajes ICMPv6 estándar en lugar de definir otro protocolo de capa 3. Con esto se consigue una mayor flexibilidad de uso, especialmente en las redes que no soportan la difusión de mensajes. Al igual que en IPv4, el protocolo se encarga de construir tablas de correspondencia entre las direcciones IPv6 y las físicas.
* Detección de inaccesibilidad de vecinos (NUD, *Neighbor Unreachability Detection*). Esta función, que no existe en IPv4, permite eliminar de las tablas de configuración de un equipo, los vecinos que se han vuelto inaccesibles (por fallas, cambio de domicilio, ...). Si un enrutador deja de estar disponible, la tabla de enrutamiento puede ser modificada para considerar otra ruta.
* Se implementan varias funcionalidades del protocolo de descubrimiento de vecinos:
  + Descubrimiento de enrutadores. Este protocolo permite que los dispositivos identifiquen a los enrutadores que se encuentran en su enlace físico. En IPv4, esta función la proporciona el protocolo *ICMP Router Discovery*.
  + Descubrimiento de prefijos. El dispositivo conoce el o los prefijos de red de acuerdo a los anuncios realizados por los enrutadores. Agregando su identificador de interfaz, el dispositivo construye su o sus direcciones IPv6. No hay un mecanismo equivalente para las direcciones IPv4 ya que son demasiado cortas para realizar la auto-configuración.
  + Detección de direcciones duplicadas. Dado que las direcciones se configuran de forma automática, existen riesgos de error en caso de tener equipos con identificadores repetidos. Este protocolo verifica que ningún otro equipo en el enlace tiene la misma dirección IPv6. Esta función es una evolución del ARP gratuito en IPv4 emitido durante la inicialización de la interfaz.
  + Descubrimiento de parámetros. Este protocolo permite que los dispositivos conozcan los distintos parámetros del enlace físico, por ejemplo, el tamaño de MTU, el número máximo de saltos permitido, si la configuración automática con estado está activa, etc. No existe un equivalente de esta funcionalidad en IPv4.
* Indicación de redirección. Este mensaje se utiliza cuando un enrutador conoce una trayectoria mejor (medida en número de saltos) para alcanzar un destino.

5.5) No, en IPv6 no existen direcciones broadcast, su funcionalidad ha sido mejorada por las direcciones multicast, las cuales le envían el paquete a todas las interfaces de cierto grupo y se evita enviar paquetes a interfaces que no pertenezcan a ese grupo.

5.6) ***Las direcciones Link-Local*** son el equivalente a las direcciones IP privadas en IPv4.

El prefijo de estas direcciones es FE80::/10. Estas direcciones no pueden ser encaminadas a través de los Routers fuera del segmento local, de ahí deriva su nombre.

Una dirección IPv6 Link-Local comienza con el prefijo FE80::/10 (los primeros 10 bits), luego los bits del 11 hasta 64 (los siguientes 54 bits) se configuran con valores de ceros (0000).

La porción de nodo, que son los últimos 64 bits, se forma con el formato EUI-64. El formato EUI-64 toma los 48 bits de la dirección MAC de la tarjeta Ethernet y le coloca 16 bits adicionales predefinidos por el protocolo IPv6 (FFFE). Ejemplo: FE80::211:21FF:FE6C:C86B

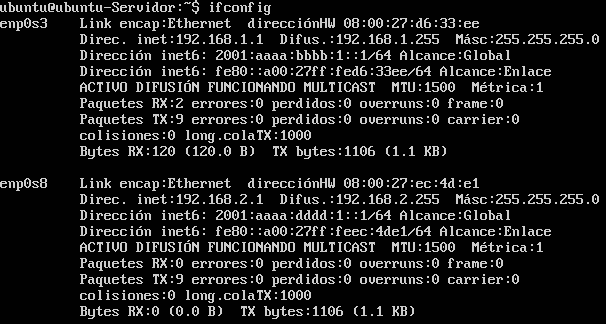
***Las direcciones Site-Loca***l son también el equivalente a las direcciones IP privadas en IPv4. A diferencias de las direcciones Link-Local, estas pueden ser encaminadas fuera del segmento local, es decir, podemos enviar paquetes entre diferentes segmentos de la red pero no hacia el Internet.

En las direcciones Site-Local, los primeros 10 bits se establecen con los valores 1111111011, por lo tanto, el prefijo de estás direcciones tendrá un valor en hexadecimal de FEC0 :: /10. Los siguientes 54 bits están compuestos por el ID de red. Los últimos 64 bits son el identificador de la interfaz o nodo, y estos se configuran de la misma forma que las direcciones Link-Local, tomando 48 bits de la dirección MAC y luego agregando 16 bits con los valores FFFE. Ejemplo: FEC0::CE00:3BFF:FE85:0

***Las direcciones Global*** son el equivalente de las direcciones IP públicas en IPv4. Estas direcciones pueden ser encaminadas a través de la Internet. Los primeros 3 bits están compuestos por los valores 001 (en notación binaria), por lo tanto, el prefijo de estás direcciones IP tendrá un valor en hexadecimal de 2000 con una máscara /3. Ejemplo: 2001:0db8:3c4d:0015::1a2f:1a2b

**Ejercicio 2: Ruteo estático IPv4/IPv6 con Linux**

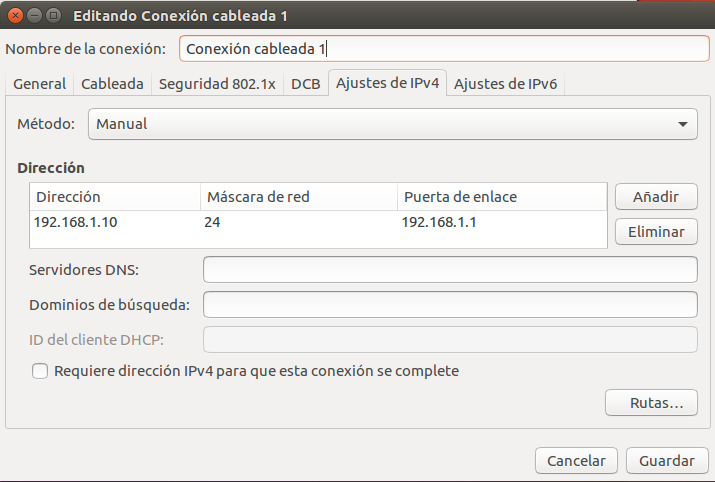
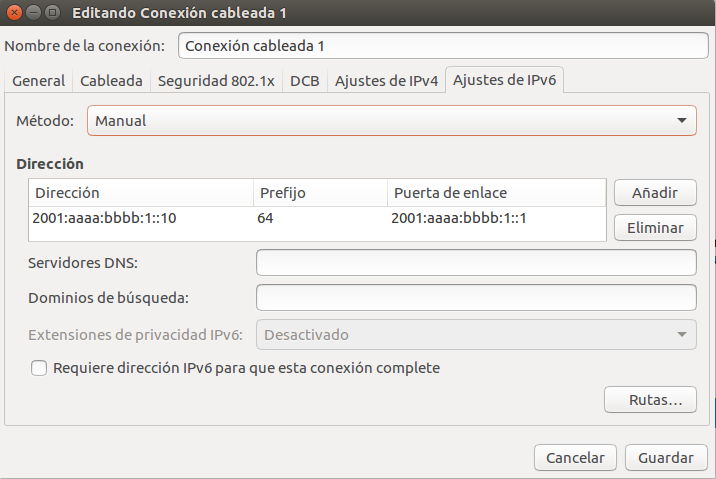
1. Para empezar configuramos de manera permanente las interfaces de red con las direcciones IP correspondientes sobre el router.



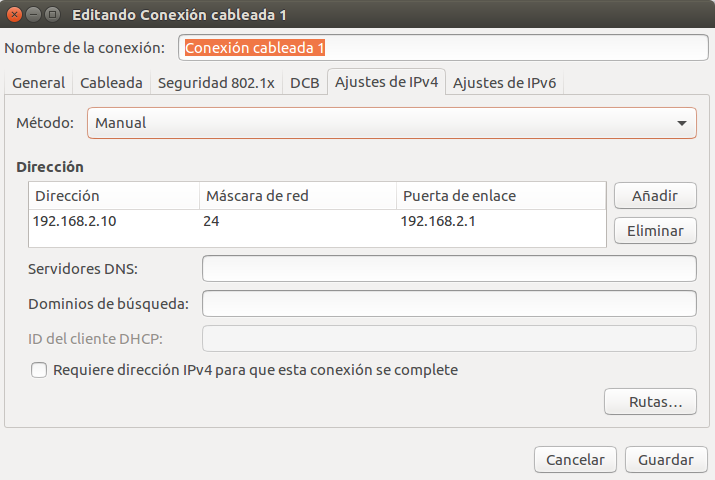
Luego, configuramos para que realice ip\_forwarding de manera permanente, permitiendo que el ubuntu server se comporte como un router virtual.

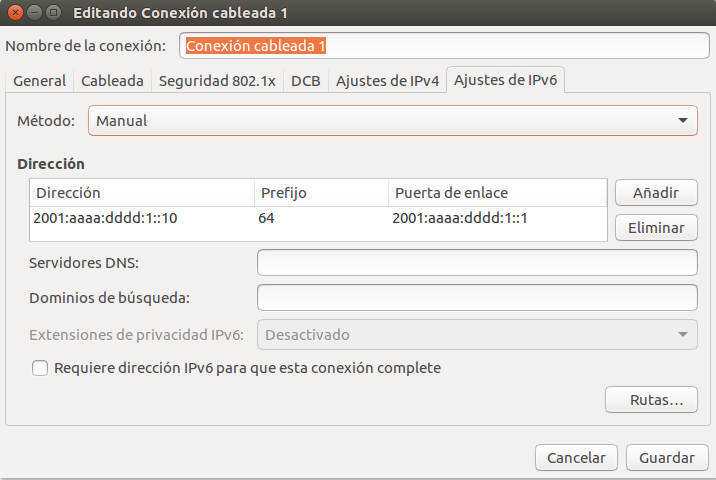
3) Configuramos sobre ambos clientes las direcciones de IP, mascara de red y puerta de enlace de manera permantente utilizando el network manager.

Cliente1:



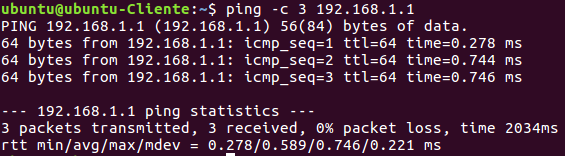
Cliente 2:

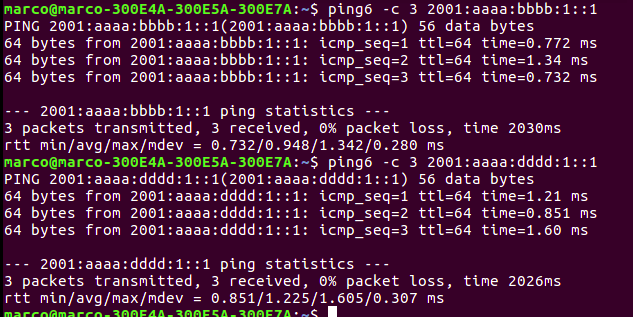




4)

4.1) En las siguientes capturas podemos ver que al realizar ping de un cliente al default gateway, el resultado es exitoso tanto en IPv4 como IPv6.





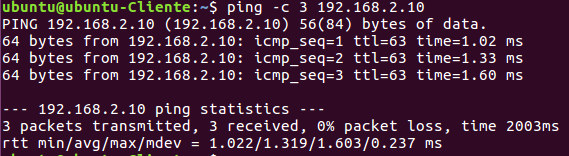
El proceso de comunicación en IPv4 es el siguiente:

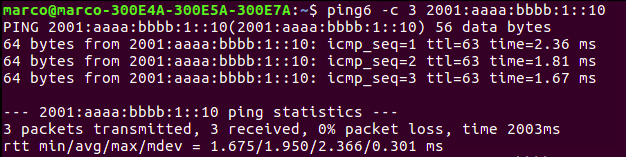
La primera vez que se realiza un ping a una determinada dirección IP, la tabla ARP estara vacia, por lo tanto el primer mensaje será un **ARP request** en broadcast preguntando quién tiene la IP de destino (en este caso la IP del gateway), el ubuntu server que está funcionando como router recibirá ese mensaje y responderá al cliente con un **ARP reply** en la cual informara cual es su dirección MAC. Una vez que el cliente tiene la dirección MAC de destino enviará un **echo request** a la dirección MAC de destino y la misma le responderá con una **echo reply.**

El proceso de comunicación en IPv6 es el siguiente:

La primera vez que el cliente1 intenta realizar un ping6 con el destino tendrá que preguntar por su dirección MAC, para eso realizara enviará un **neighbor solicitation** a su gateway preguntando qué dirección MAC corresponde a la ip de destino, la cual coincide con la dirección del gateway. El router recibe la solicitud por la interfaz que comparte red con el cliente1 y le responde con un **neighbor advertisement**. Ahora el cliente conoce la dirección MAC de destino y puede enviar un **echo request** a la misma. El router contesta con un **echo reply**.

4.2) En la siguiente captura de pantalla podemos ver que al realizar ping entre los clientes tanto en IPv4 como IPv6 el resultado es exitoso:



****

El proceso de comunicación en IPv4 es el siguiente:

La primera vez que se realiza un ping a una determinada dirección IP, la tabla ARP estara vacia, por lo tanto el primer mensaje será un **ARP request** en broadcast preguntando quién tiene la IP del próximo salto, que en este caso no es la de destino sino la del gateway, el ubuntu server que está funcionando como router recibirá ese mensaje y responderá al cliente con un **ARP reply** en la cual informará cuál es su dirección MAC. Una vez que el cliente tiene la dirección MAC de la interfaz del router, enviará un **echo request** a la dirección MAC de esa interfaz indicandole cual es la IP de destino, que en este caso es la del cliente 2. El router recibe este mensaje y enviará a través de broadcast un nuevo **ARP request** consultando cual es la dirección MAC de la IP del cliente 2. Al recibir este mensaje el cliente 2, le enviará un **ARP reply** al router con su dirección MAC. El router ahora le enviará un **echo request** al cliente 2. El cual responderá con un **echo reply** al router y el router enviará nuevamente un echo reply al cliente 1 para concluir con esta comunicación.

El proceso de comunicación en IPv6 es el siguiente:

Al querer hacer un ping6 al cliente2, el cliente1 reconoce que la ip de destino no se encuentra conectado directamente, por lo que es necesario pasar primero por su default gateway. En el caso que todavía no conozca la dirección MAC del router, el cliente1 enviará un **neighbor solicitation** a su gateway preguntando por su dirección MAC, el router recibe la solicitud por la interfaz que comparte red con el cliente1 y le responde con un **neighbor advertisement**. Ahora el cliente1 le envía un **echo request** al router con destino para el cliente2. El router si desconoce la MAC del cliente 2, enviara un **neighbor solicitation** a través de multicast para obtener la MAC del cliente2. El cliente2 responderá con un **neighbor advertisement**. Ahora el router envía un **echo request** al cliente2, el cual lo contesta con un **echo reply** que llega al router, y por último el router le envía el **echo reply** al cliente1.

**Ejercicio 3: Tecnología namespace**

1.1) Un namespace es una capa de abstracción que hace que parezca que los procesos dentro de un determinado espacio de usuario tengan aislados sus propios recursos hardware. Los cambios en los recursos globales son visibles para procesos miembros del mismo namespace, pero no para procesos desplegados en distintos namespaces. Cuando se necesita aislar un recurso hardware a un grupo de procesos, este dependerá del tipo de namespace. Todos los procesos son asociados con un namespace y solo podrán utilizar los recursos únicamente asociados a ese namespace.

1.2) El Linux Bridge emula un Bridge físico a través de software, esto permite utilizar una máquina virtual como un Bridge. Nos permite conectar 2 o más dispositivos reales o virtuales sin la necesidad de utilizar un dispositivo físico.

1.3) El “Veth Pair” o par virtual Ethernet según su traducción es un par de dispositivos conectados de manera virtual en su capa de enlace.Se utiliza para conectar 2 namespaces distintos y para conectar un contenedor o una máquina virtual a un switch virtual.

1.4) Con los siguientes comandos se pueden crear 2 namespaces y unirlos a traves de un veth pair.

*# creamos los namespaces*

*ip netns add ns1*

*ip netns add ns2*

*# creamos el veth pair*

*ip link add tap1 type veth peer name tap2*

*# movemos las interfaces a los namespaces*

*ip link set tap1 netns ns1*

*ip link set tap2 netns ns2*

*# activamos los enlaces*

*ip netns exec ns1 ip link set dev tap1 up*

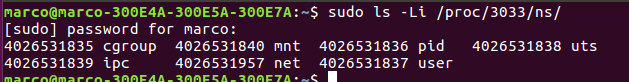
*ip netns exec ns2 ip link set dev tap2 up*

1.5) Para determinar esto lo primero es ejecutar el programa core y ver que PID tiene su proceso.



Podemos ver que el proceso tiene el PID 3033.

Ahora podemos listar todos los namespaces asociados a dicho proceso utilizando el siguiente comando:

****

Como podemos observar hay distintos namespaces asociados al proceso del core, por lo que podemos concluir que si utiliza linux namespaces.

**Bibliografía consultada**

<http://www.yourownlinux.com/2013/07/how-to-configure-ubuntu-as-router.html>

<http://blog.capacityacademy.com/2013/04/16/cisco-ccna-todo-sobre-ipv6-tipos-de-direcciones/>

<http://www.ducea.com/2006/08/01/how-to-enable-ip-forwarding-in-linux/>

<https://blog.scottlowe.org/2013/09/04/introducing-linux-network-namespaces/>

<https://www.opencloudblog.com/?p=66>