***Trabajo Práctico*** *3*

*Ruteo estático. Fragmentación*

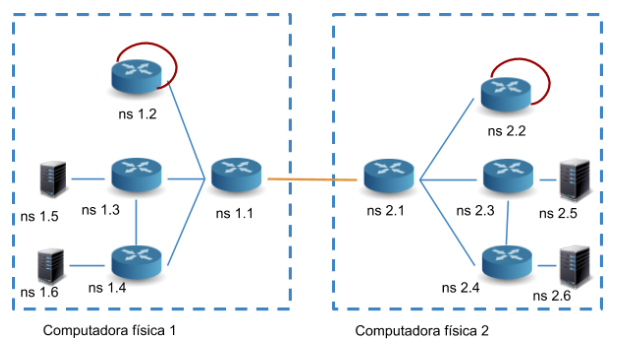
*Autores*

*Esteban Nicolas Andrada:*

[*nico86gnr@gmail.com*](mailto:nico86gnr@gmail.com)

*Federico Sepulveda:* [*federico.sepulveda@alumnos.unc.edu.ar*](mailto:federico.sepulveda@alumnos.unc.edu.ar)

# Diagrama



# Tabla de asignación de direcciones IPv6

# Crear la tabla de asignación de direcciones IP

|  |  |  |
| --- | --- | --- |
| **Computadora** | **Interfaz de Red** | **Dirección IP** |
| PC 1 | enp0s3 | 2001:1111:1111:1111::1 |
| ns 1.1 | tap1.1.2 | 2001:1111:1111:2222::10 |
| tap1.1.3 | 2001:1111:1111:3333::10 |
| tap1.1.4 | 2001:1111:1111:4444::10 |
| tap1.1.br | 2001:aaaa:bbbb:cccc::10 |
| ns 1.2 | tap1.2.1 | 2001:1111:1111:2222::20 |
| tap1.2.a | 2001:1111:2222:aabb::aaaa |
| tap1.2.b | 2001:1111:2222:aabb::bbbb |
| ns 1.3 | tap1.3.1 | 2001:1111:1111:3333::30 |
| tap1.3.4 | 2001:1111:3333:4444::30 |
| tap1.3.5 | 2001:1111:3333:5555::30 |
| ns 1.4 | tap1.4.1 | 2001:1111:1111:4444::40 |
| tap1.4.3 | 2001:1111:3333:4444::40 |
| tap1.4.6 | 2001:1111:4444:6666::40 |
| ns 1.5 | tap1.5.3 | 2001:1111:3333:5555::50 |
| ns 1.6 | tap1.6.4 | 2001:1111:3333:5555::60 |
| PC 2 | eth0 | 2001:2222:1111:1111::1 |
| ns 2.1 | tap2.1.2 | 2001:2222:1111:2222::10 |
| tap2.1.3 | 2001:2222:1111:3333::10 |
| tap2.1.4 | 2001:2222:1111:4444::10 |
| tap2.1.br | 2001:aaaa:bbbb:cccc::20 |
| ns 2.2 | tap2.2.1 | 2001:2222:1111:2222::20 |
| tap2.2.a | 2001:2222:2222:aabb::aaaa |
| tap2.2.b | 2001:2222:2222:aabb::bbbb |
| ns 2.3 | tap2.3.1 | 2001:2222:1111:3333::30 |
| tap2.3.4 | 2001:2222:3333:4444::30 |
| tap2.3.5 | 2001:2222:3333:5555::30 |
| ns 2.4 | tap2.4.1 | 2001:2222:1111:4444::40 |
| tap2.4.3 | 2001:2222:3333:4444::40 |
| tap2.4.6 | 2001:2222:4444:6666::40 |
| ns 2.5 | tap2.5.3 | 2001:2222:3333:5555::50 |
| ns 2.6 | tap2.6.4 | 2001:2222:3333:5555::60 |

# Consignas

# Creación y conexión de namespaces ns 1.1 y ns 2.1

# 1.- Crear los namespaces ns 1.1 y ns 2.1 en las correspondientes máquinas virtuales

El namespace para la pc1 se crea con el comando:

ip netns add ns1.1

El namespace para la pc2 se crea con el comando:

ip netns add ns2.1

# 2.- Crear un bridge llamado br-externo en cada máquina virtual

Los bridge para cada pc se crearon con:

brctl addbr br-externo

# 3.- Conectar cada namespace con el bridge creado en el punto anterior.

Conexión con el ns1.1:

ip netns exec ns1.1 ip link set up dev tap1.1.br

Conexión con el ns2.1:

ip netns exec ns2.1 ip link set up dev tap2.1.br

# 4.- Conectar la interfaz externa de la máquina virtual al bridge creado en el punto anterior.

Se creó la variable INTERFAZ\_FISICA en la cual se carga el nombre de la interfaz que posteriormente se utilizara para la conexión con el bridge:

para PC1:

INTERFAZ\_FISICA=enp3s0f1

para PC2:

INTERFAZ\_FISICA=eth0

Para conectar la interfaz con el bridge se utilizó el siguiente comando en ambas pc:

brctl addif br-externo ${INTERFAZ\_FISICA}

No se le asigna dirección ipv6 a la interfaz física, solo se deja la link local.

# 5.- Configurar direcciones de red a los namespaces.

Asignación de direcciones IPv6 para cada interfaz de n1.1:

ip netns exec ns1.1 ip -6 addr add ${IPV6\_TAP112}/64 dev tap1.1.2

ip netns exec ns1.1 ip -6 addr add ${IPV6\_TAP113}/64 dev tap1.1.3

ip netns exec ns1.1 ip -6 addr add ${IPV6\_TAP114}/64 dev tap1.1.4

ip netns exec ns1.1 ip -6 addr add ${IPV6\_TAP11br}/64 dev tap1.1.br

Asignación de direcciones IPv6 para cada interfaz de n2.1:

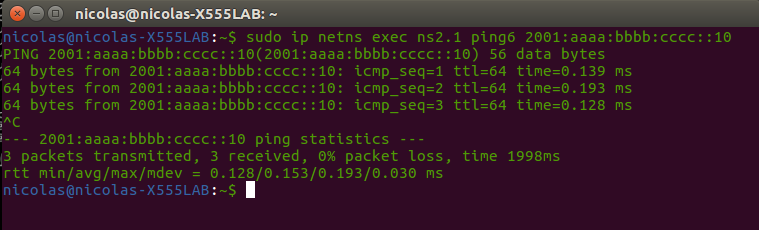
ip netns exec ns2.1 ip -6 addr add ${IPV6\_TAP212}/64 dev tap2.1.2

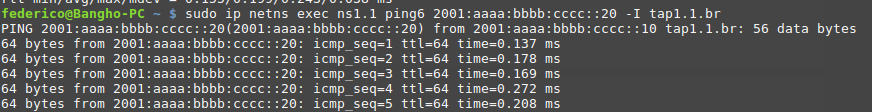
ip netns exec ns2.1 ip -6 addr add ${IPV6\_TAP213}/64 dev tap2.1.3

ip netns exec ns2.1 ip -6 addr add ${IPV6\_TAP214}/64 dev tap2.1.4

ip netns exec ns2.1 ip -6 addr add ${IPV6\_TAP21br}/64 dev tap2.1.br

# 6.- Probar conectividad entre las dos puntas





# 7.- Configurar MTU a 500 para estas interfaces en los namespaces. No cambiar MTU en

# ningún otro punto de la red.

Para la configuración del MTU se utilizó el siguiente comando en el enlace entre ns1.2 y ns2.1:

ip netns exec ns1.1 ip link set tap1.1.br mtu 1280

ip netns exec ns2.1 ip link set tap2.1.br mtu 1280

# Creación de redes y ruteo

# 8.- Crear los routers y host en cada computadora.

Routers PC1:

creación de los routers:

ip netns add ns1.2

ip netns add ns1.3

ip netns add ns1.4

Asignación de direcciones IPv6 para cada interfaz de n1.2:

ip netns exec ns1.2 ip -6 addr add ${IPV6\_TAP121}/64 dev tap1.2.1

ip netns exec ns1.2 ip -6 addr add ${IPV6\_TAP12a}/64 dev tap1.2.a

ip netns exec ns1.2 ip -6 addr add ${IPV6\_TAP12b}/64 dev tap1.2.b

Asignación de direcciones IPv6 para cada interfaz de n1.3:

ip netns exec ns1.3 ip -6 addr add ${IPV6\_TAP131}/64 dev tap1.3.1

ip netns exec ns1.3 ip -6 addr add ${IPV6\_TAP134}/64 dev tap1.3.4

ip netns exec ns1.3 ip -6 addr add ${IPV6\_TAP135}/64 dev tap1.3.5

Asignación de direcciones IPv6 para cada interfaz de n1.4:

ip netns exec ns1.4 ip -6 addr add ${IPV6\_TAP141}/64 dev tap1.4.1

ip netns exec ns1.4 ip -6 addr add ${IPV6\_TAP143}/64 dev tap1.4.3

ip netns exec ns1.4 ip -6 addr add ${IPV6\_TAP146}/64 dev tap1.4.6

Hosts PC1:

creación de los hosts:

ip netns add ns1.5

ip netns add ns1.6

Asignación de direcciones IPv6 para cada interfaz de n1.5:

ip netns exec ns1.5 ip -6 addr add ${IPV6\_TAP153}/64 dev tap1.5.3

Asignación de direcciones IPv6 para cada interfaz de n1.6:

ip netns exec ns1.6 ip -6 addr add ${IPV6\_TAP164}/64 dev tap1.6.4

Routers PC2:

creación de los routers:

ip netns add ns2.2

ip netns add ns2.3

ip netns add ns2.4

Asignación de direcciones IPv6 para cada interfaz de n2.2:

ip netns exec ns2.2 ip -6 addr add ${IPV6\_TAP221}/64 dev tap2.2.1

ip netns exec ns2.2 ip -6 addr add ${IPV6\_TAP22a}/64 dev tap2.2.a

ip netns exec ns2.2 ip -6 addr add ${IPV6\_TAP22b}/64 dev tap2.2.b

Asignación de direcciones IPv6 para cada interfaz de n1.3:

ip netns exec ns2.3 ip -6 addr add ${IPV6\_TAP231}/64 dev tap2.3.1

ip netns exec ns2.3 ip -6 addr add ${IPV6\_TAP234}/64 dev tap2.3.4

ip netns exec ns2.3 ip -6 addr add ${IPV6\_TAP235}/64 dev tap2.3.5

Asignación de direcciones IPv6 para cada interfaz de n1.4:

ip netns exec ns2.4 ip -6 addr add ${IPV6\_TAP241}/64 dev tap2.4.1

ip netns exec ns2.4 ip -6 addr add ${IPV6\_TAP243}/64 dev tap2.4.3

ip netns exec ns2.4 ip -6 addr add ${IPV6\_TAP246}/64 dev tap2.4.6

Hosts PC2:

creación de los hosts:

ip netns add ns2.5

ip netns add ns2.6

Asignación de direcciones IPv6 para cada interfaz de n2.5:

ip netns exec ns2.5 ip -6 addr add ${IPV6\_TAP253}/64 dev tap2.5.3

Asignación de direcciones IPv6 para cada interfaz de n2.6:

ip netns exec ns2.6 ip -6 addr add ${IPV6\_TAP264}/64 dev tap2.6.4

Para cada uno de los routers se habilitó el forwarding mediante el siguiente comando:

ip netns exec <namespace> sysctl -w net.ipv6.conf.all.forwarding=1

# 9.- Configurar ruteo estático para que todos los hosts desde todas las interfaces sean

# alcanzables unos a otros

PC1:

Configuración del default gateway para las interfaces de ns1.5 y ns1.6:

ip netns exec ns1.5 ip -6 route add default dev tap1.5.3 via ${IPV6\_TAP135}

ip netns exec ns1.6 ip -6 route add default dev tap1.6.4 via ${IPV6\_TAP146}

Ruteo estático interno ns1.1:

ip netns exec ns1.1 ip -6 route add ${RED135}/64 via ${IPV6\_TAP131} dev tap1.1.3

ip netns exec ns1.1 ip -6 route add ${RED146}/64 via ${IPV6\_TAP141} dev tap1.1.4

ip netns exec ns1.1 ip -6 route add ${RED134}/64 via ${IPV6\_TAP131} dev tap1.1.3

ip netns exec ns1.1 ip -6 route add ${RED12AB}/64 via ${IPV6\_TAP121} dev tap1.1.2

Ruteo estático externo ns1.1:

ip netns exec ns1.1 ip -6 route add ${RED212}/64 via ${IPV6\_TAP21br} dev tap1.1.br

ip netns exec ns1.1 ip -6 route add ${RED213}/64 via ${IPV6\_TAP21br} dev tap1.1.br

ip netns exec ns1.1 ip -6 route add ${RED214}/64 via ${IPV6\_TAP21br} dev tap1.1.br

ip netns exec ns1.1 ip -6 route add ${RED22AB}/64 via ${IPV6\_TAP21br} dev tap1.1.br

ip netns exec ns1.1 ip -6 route add ${RED234}/64 via ${IPV6\_TAP21br} dev tap1.1.br

ip netns exec ns1.1 ip -6 route add ${RED235}/64 via ${IPV6\_TAP21br} dev tap1.1.br

ip netns exec ns1.1 ip -6 route add ${RED246}/64 via ${IPV6\_TAP21br} dev tap1.1.br

Ruteo estático interno ns1.2:

ip netns exec ns1.2 ip -6 route add ${RED113}/64 via ${IPV6\_TAP112} dev tap1.2.1

ip netns exec ns1.2 ip -6 route add ${RED114}/64 via ${IPV6\_TAP112} dev tap1.2.1

ip netns exec ns1.2 ip -6 route add ${RED135}/64 via ${IPV6\_TAP112} dev tap1.2.1

ip netns exec ns1.2 ip -6 route add ${RED146}/64 via ${IPV6\_TAP112} dev tap1.2.1

ip netns exec ns1.2 ip -6 route add ${RED134}/64 via ${IPV6\_TAP112} dev tap1.2.1

Ruteo estático externo ns1.2:

ip netns exec ns1.2 ip -6 route add ${REDABC}/64 via ${IPV6\_TAP112} dev tap1.2.1

ip netns exec ns1.2 ip -6 route add ${RED212}/64 via ${IPV6\_TAP112} dev tap1.2.1

ip netns exec ns1.2 ip -6 route add ${RED213}/64 via ${IPV6\_TAP112} dev tap1.2.1

ip netns exec ns1.2 ip -6 route add ${RED214}/64 via ${IPV6\_TAP112} dev tap1.2.1

ip netns exec ns1.2 ip -6 route add ${RED22AB}/64 via ${IPV6\_TAP112} dev tap1.2.1

ip netns exec ns1.2 ip -6 route add ${RED234}/64 via ${IPV6\_TAP112} dev tap1.2.1

ip netns exec ns1.2 ip -6 route add ${RED235}/64 via ${IPV6\_TAP112} dev tap1.2.1

ip netns exec ns1.2 ip -6 route add ${RED246}/64 via ${IPV6\_TAP112} dev tap1.2.1

Ruteo estático interno ns1.3:

ip netns exec ns1.3 ip -6 route add ${RED146}/64 via ${IPV6\_TAP143} dev tap1.3.4

ip netns exec ns1.3 ip -6 route add ${RED114}/64 via ${IPV6\_TAP143} dev tap1.3.4

ip netns exec ns1.3 ip -6 route add ${RED112}/64 via ${IPV6\_TAP113} dev tap1.3.1

ip netns exec ns1.3 ip -6 route add ${RED12AB}/64 via ${IPV6\_TAP113} dev tap1.3.1

Ruteo estático externo ns1.3:

ip netns exec ns1.3 ip -6 route add ${REDABC}/64 via ${IPV6\_TAP113} dev tap1.3.1

ip netns exec ns1.3 ip -6 route add ${RED212}/64 via ${IPV6\_TAP113} dev tap1.3.1

ip netns exec ns1.3 ip -6 route add ${RED213}/64 via ${IPV6\_TAP113} dev tap1.3.1

ip netns exec ns1.3 ip -6 route add ${RED214}/64 via ${IPV6\_TAP113} dev tap1.3.1

ip netns exec ns1.3 ip -6 route add ${RED22AB}/64 via ${IPV6\_TAP113} dev tap1.3.1

ip netns exec ns1.3 ip -6 route add ${RED234}/64 via ${IPV6\_TAP113} dev tap1.3.1

ip netns exec ns1.3 ip -6 route add ${RED235}/64 via ${IPV6\_TAP113} dev tap1.3.1

ip netns exec ns1.3 ip -6 route add ${RED246}/64 via ${IPV6\_TAP113} dev tap1.3.1

Ruteo estático interno ns1.4:

ip netns exec ns1.4 ip -6 route add ${RED135}/64 via ${IPV6\_TAP134} dev tap1.4.3

ip netns exec ns1.4 ip -6 route add ${RED113}/64 via ${IPV6\_TAP134} dev tap1.4.3

ip netns exec ns1.4 ip -6 route add ${RED112}/64 via ${IPV6\_TAP114} dev tap1.4.1

ip netns exec ns1.4 ip -6 route add ${RED12AB}/64 via ${IPV6\_TAP114} dev tap1.4.1

Ruteo estático externo ns1.4:

ip netns exec ns1.4 ip -6 route add ${REDABC}/64 via ${IPV6\_TAP114} dev tap1.4.1

ip netns exec ns1.4 ip -6 route add ${RED212}/64 via ${IPV6\_TAP114} dev tap1.4.1

ip netns exec ns1.4 ip -6 route add ${RED213}/64 via ${IPV6\_TAP114} dev tap1.4.1

ip netns exec ns1.4 ip -6 route add ${RED214}/64 via ${IPV6\_TAP114} dev tap1.4.1

ip netns exec ns1.4 ip -6 route add ${RED22AB}/64 via ${IPV6\_TAP114} dev tap1.4.1

ip netns exec ns1.4 ip -6 route add ${RED234}/64 via ${IPV6\_TAP114} dev tap1.4.1

ip netns exec ns1.4 ip -6 route add ${RED235}/64 via ${IPV6\_TAP114} dev tap1.4.1

ip netns exec ns1.4 ip -6 route add ${RED246}/64 via ${IPV6\_TAP114} dev tap1.4.1

PC2:

Configuración del default gateway para las interfaces de ns2.5 y ns2.6:

ip netns exec ns2.5 ip -6 route add default dev tap2.5.3 via ${IPV6\_TAP235}

ip netns exec ns2.6 ip -6 route add default dev tap2.6.4 via ${IPV6\_TAP246}

Ruteo estático interno ns2.1:

ip netns exec ns2.1 ip -6 route add ${RED235}/64 via ${IPV6\_TAP231}

ip netns exec ns2.1 ip -6 route add ${RED246}/64 via ${IPV6\_TAP241}

ip netns exec ns2.1 ip -6 route add ${RED234}/64 via ${IPV6\_TAP231}

ip netns exec ns2.1 ip -6 route add ${RED22AB}/64 via ${IPV6\_TAP221}

Ruteo estático externo ns2.1:

ip netns exec ns2.1 ip -6 route add ${RED112}/64 via ${IPV6\_TAP11br} dev tap2.1.br

ip netns exec ns2.1 ip -6 route add ${RED113}/64 via ${IPV6\_TAP11br} dev tap2.1.br

ip netns exec ns2.1 ip -6 route add ${RED114}/64 via ${IPV6\_TAP11br} dev tap2.1.br

ip netns exec ns2.1 ip -6 route add ${RED12AB}/64 via ${IPV6\_TAP11br} dev tap2.1.br

ip netns exec ns2.1 ip -6 route add ${RED134}/64 via ${IPV6\_TAP11br} dev tap2.1.br

ip netns exec ns2.1 ip -6 route add ${RED135}/64 via ${IPV6\_TAP11br} dev tap2.1.br

ip netns exec ns2.1 ip -6 route add ${RED146}/64 via ${IPV6\_TAP11br} dev tap2.1.br

Ruteo estático interno ns2.3:

ip netns exec ns2.3 ip -6 route add ${RED246}/64 via ${IPV6\_TAP243} dev tap2.3.4

ip netns exec ns2.3 ip -6 route add ${RED214}/64 via ${IPV6\_TAP243} dev tap2.3.4

ip netns exec ns2.3 ip -6 route add ${RED212}/64 via ${IPV6\_TAP213} dev tap2.3.1

ip netns exec ns2.3 ip -6 route add ${RED22AB}/64 via ${IPV6\_TAP213} dev tap2.3.1

Ruteo estático externo ns2.3:

ip netns exec ns2.3 ip -6 route add ${REDABC}/64 via ${IPV6\_TAP213} dev tap2.3.1

ip netns exec ns2.3 ip -6 route add ${RED112}/64 via ${IPV6\_TAP213} dev tap2.3.1

ip netns exec ns2.3 ip -6 route add ${RED113}/64 via ${IPV6\_TAP213} dev tap2.3.1

ip netns exec ns2.3 ip -6 route add ${RED114}/64 via ${IPV6\_TAP213} dev tap2.3.1

ip netns exec ns2.3 ip -6 route add ${RED12AB}/64 via ${IPV6\_TAP213} dev tap2.3.1

ip netns exec ns2.3 ip -6 route add ${RED134}/64 via ${IPV6\_TAP213} dev tap2.3.1

ip netns exec ns2.3 ip -6 route add ${RED135}/64 via ${IPV6\_TAP213} dev tap2.3.1

ip netns exec ns2.3 ip -6 route add ${RED146}/64 via ${IPV6\_TAP213} dev tap2.3.1

Ruteo estático interno ns2.4:

ip netns exec ns2.4 ip -6 route add ${RED235}/64 via ${IPV6\_TAP234} dev tap2.4.3

ip netns exec ns2.4 ip -6 route add ${RED213}/64 via ${IPV6\_TAP234} dev tap2.4.3

ip netns exec ns2.4 ip -6 route add ${RED212}/64 via ${IPV6\_TAP214} dev tap2.4.1

ip netns exec ns2.4 ip -6 route add ${RED22AB}/64 via ${IPV6\_TAP214} dev tap2.4.1

Ruteo estático externo ns2.4:

ip netns exec ns2.4 ip -6 route add ${REDABC}/64 via ${IPV6\_TAP214} dev tap2.4.1

ip netns exec ns2.4 ip -6 route add ${RED112}/64 via ${IPV6\_TAP214} dev tap2.4.1

ip netns exec ns2.4 ip -6 route add ${RED113}/64 via ${IPV6\_TAP214} dev tap2.4.1

ip netns exec ns2.4 ip -6 route add ${RED114}/64 via ${IPV6\_TAP214} dev tap2.4.1

ip netns exec ns2.4 ip -6 route add ${RED12AB}/64 via ${IPV6\_TAP214} dev tap2.4.1

ip netns exec ns2.4 ip -6 route add ${RED134}/64 via ${IPV6\_TAP214} dev tap2.4.1

ip netns exec ns2.4 ip -6 route add ${RED135}/64 via ${IPV6\_TAP214} dev tap2.4.1

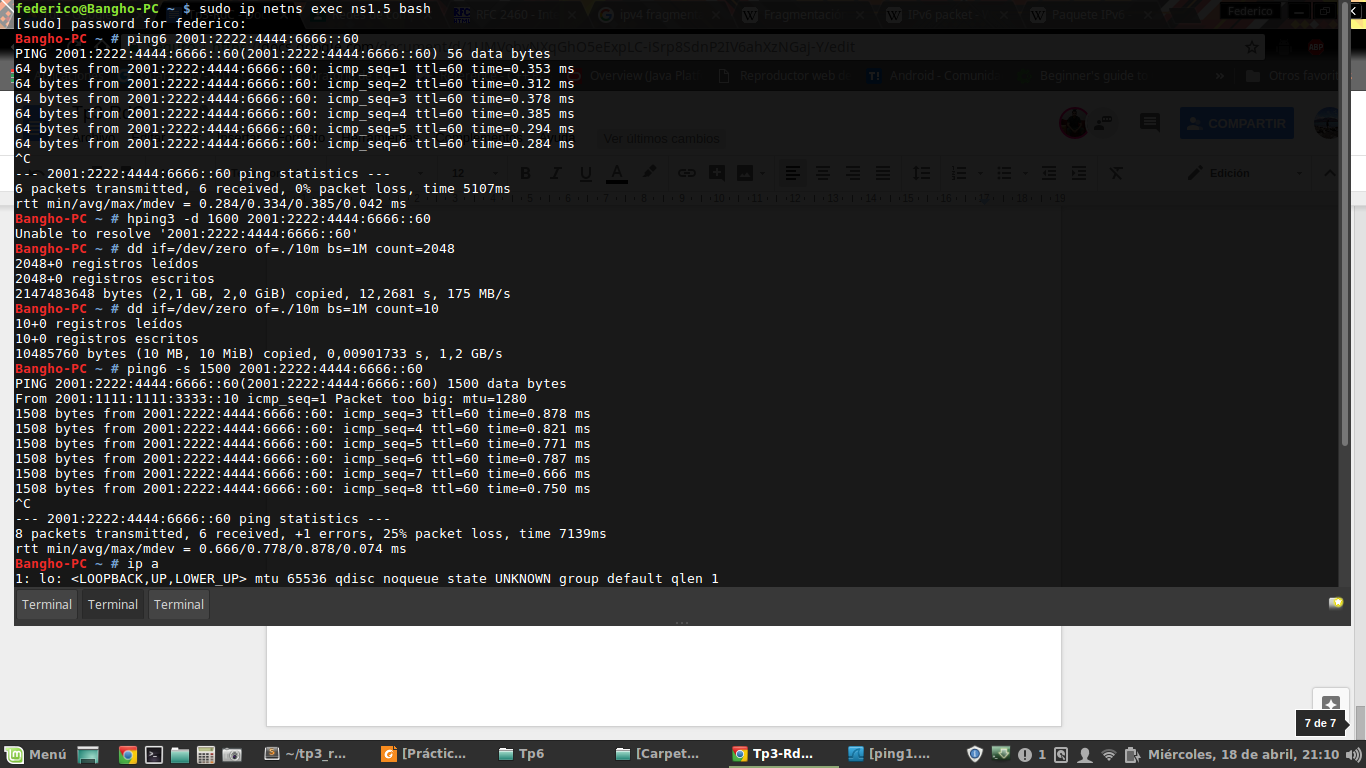
ip netns exec ns2.4 ip -6 route add ${RED146}/64 via ${IPV6\_TAP214} dev tap2.4.1

# Análisis de tráfico

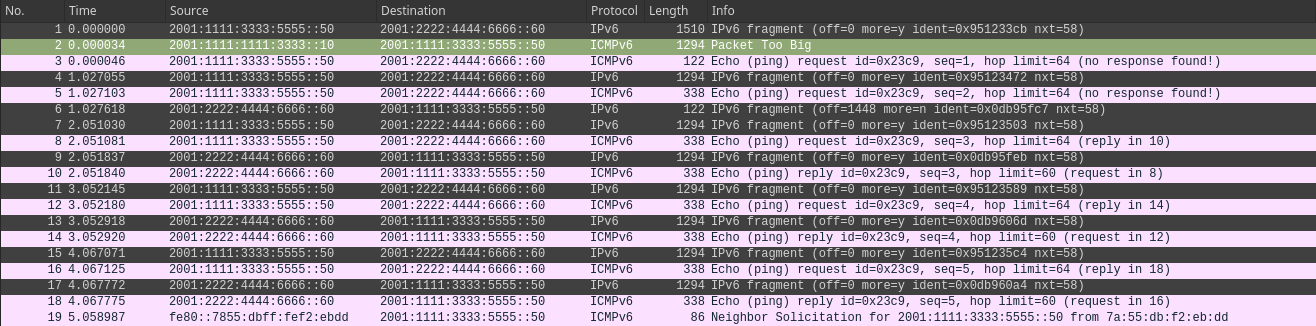
# 10.- Mediante análisis de tráfico, verificar la fragmentación que se se debe producir cuando

# un datagrama IP va desde una máquina virtual a otra.

Se hizo un ping desde en el ns1.5 al ns2.6 con un tamaño de paquete de 1500 bytes mediante la opción -s del ping6



Mientras se realizó el ping se capturó el tráfico con tcpdump para posteriormente analizarlo con wireshark



# 11.- Verificar cómo sucede la fragmentación en IPv6. Explicar cuál es la diferencia con IPv4.

En una primera instancia el host ns1.5 intenta enviar un paquete de 1500 bytes por cada mensaje echo request. Como la conexión entre los routers ns1.1 y ns2.1 se la limitó a paquetes no mayores a 1280 bytes, el router ns1.1 avisa al host ns1.5 con un mensaje “packet too big” (en ICMPv6) que no se puede enviar un paquete mayor a 1280 a bytes. Por lo que el host ns1.5 se ve obligado a fragmentar los paquete en tamaños que se le fue especificado por ns1.1.

Una vez enviado los paquete en fragmentos, estos se ensamblan en el host destino o sea ns2.6. Cuando el host termina de ensamblar envía el mensaje echo reply al origen, los cuales también serán fragmentados por ns2.6 para que puedan llega al origen.

En IPv4 en a diferencia de IPv6 los routers se encargaría de la fragmentación y el host destino se encargará de ensamblar los paquetes.

# Script para ambas pc:

pc1:<https://drive.google.com/open?id=1R1Yr7UUYsYCH7MvsZu5t_n-k6DraBIm_>

pc2:<https://drive.google.com/open?id=1jv5RFJ1LnUN8PcryO7crWxDDV-uHFjYJ>