****

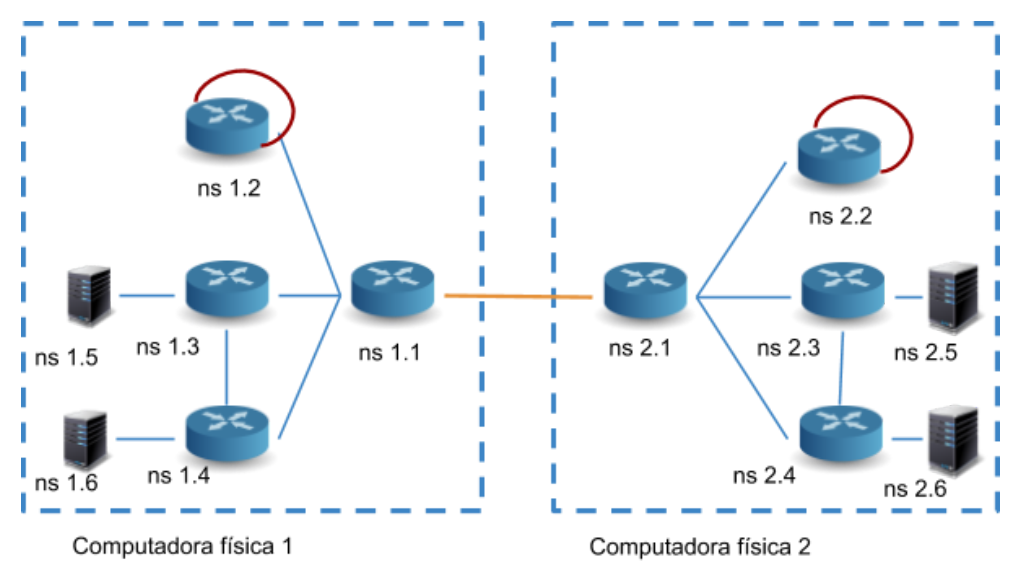
**Redes de Computadoras**

**Práctico 3: Ruteo estático. Fragmentación.**

**Alumnos:**

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

### Ejercicio 1: Ruteo Estático. Fragmentación.

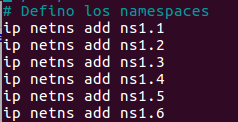
****

Podemos ver la asignación de IPs en la siguiente tabla:

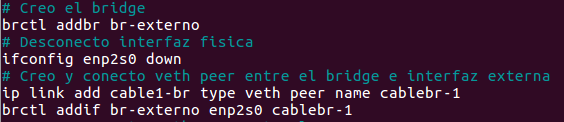
|  |  |  |
| --- | --- | --- |
| Computadora | Interfaz compartida con | Dirección IP |
| ns 1.1 | ns 2.1 | 2001:aaaa:bbbb::1 |
| ns 1.1 | ns 1.2 | 2001:aaaa:aaaa:1::1 |
| ns 1.1 | ns 1.3 | 2001:aaaa:aaaa:2::1 |
| ns 1.1 | ns 1.4 | 2001:aaaa:aaaa:3::1 |
| ns 1.2 | ns 1.1 | 2001:aaaa:aaaa:1::2 |
| ns 1.2 | ns 1.2 | 2001:aaaa:aaaa:7::1 |
| ns 1.3 | ns 1.1 | 2001:aaaa:aaaa:2::2 |
| ns 1.3 | ns 1.4 | 2001:aaaa:aaaa:4::2 |
| ns 1.3 | ns 1.5 | 2001:aaaa:aaaa:5::1 |
| ns 1.4 | ns 1.1 | 2001:aaaa:aaaa:3::2 |
| ns 1.4 | ns 1.3 | 2001:aaaa:aaaa:4::1 |
| ns 1.4 | ns 1.6 | 2001:aaaa:aaaa:6::1 |
| ns 1.5 | ns 1.3 | 2001:aaaa:aaaa:5::2 |
| ns 1.6 | ns 1.4 | 2001:aaaa:aaaa:6::2 |
| ns 2.1 | ns 1.1 | 2001:aaaa:bbbb::2 |
| ns 2.1 | ns 2.2 | 2001:aaaa:bbbb:1::1 |
| ns 2.1 | ns 2.3 | 2001:aaaa:bbbb:2::1 |
| ns 2.1 | ns 2.4 | 2001:aaaa:bbbb:3::1 |
| ns 2.2 | ns 2.1 | 2001:aaaa:bbbb:1::2 |
| ns 2.2 | ns 2.2 | 2001:aaaa:bbbb:7::1 |
| ns 2.3 | ns 2.1 | 2001:aaaa:bbbb:2::2 |
| ns 2.3 | ns 2.4 | 2001:aaaa:bbbb:4::2 |
| ns 2.3 | ns 2.5 | 2001:aaaa:bbbb:5::1 |
| ns 2.4 | ns 2.1 | 2001:aaaa:bbbb:3::2 |
| ns 2.4 | ns 2.3 | 2001:aaaa:bbbb:4::1 |
| ns 2.4 | ns 2.6 | 2001:aaaa:bbbb:6::1 |
| ns 2.5 | ns 2.3 | 2001:aaaa:bbbb:5::2 |
| ns 2.6 | ns 2.4 | 2001:aaaa:bbbb:6::2 |

1-9)

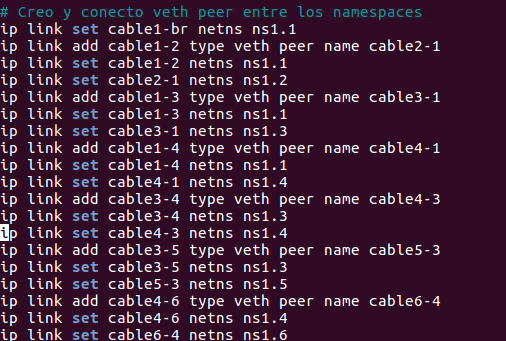
En primer lugar lo que realizamos fue crear los namespaces, para ello utilizamos los siguientes comandos:

**

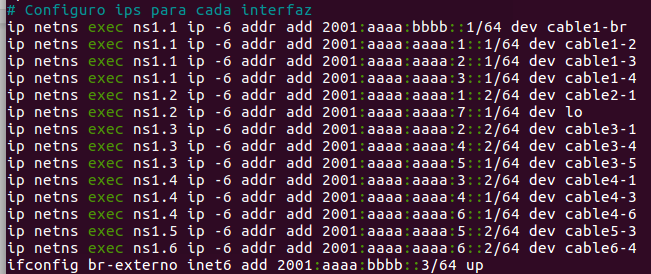
Creamos un bridge virtual y lo conectamos con la interfaz externa de nuestra pc:

**

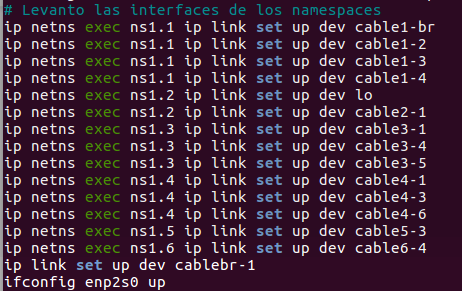
Realizamos la conexión de los distintos namespaces utilizando veth peer:



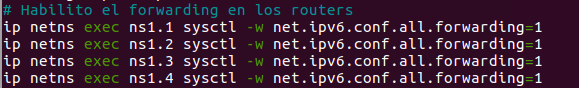
Configuramos las direcciones de IPv6 de cada interfaz:



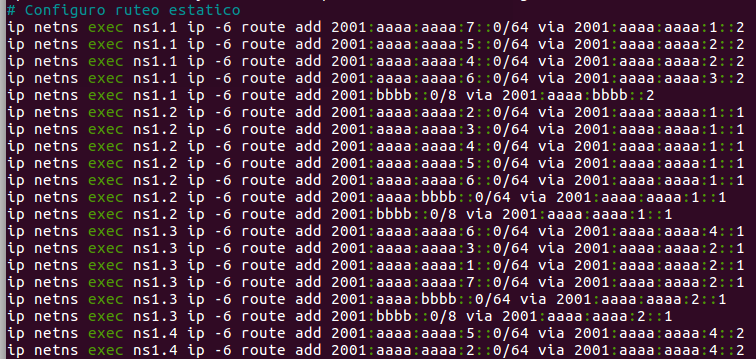
Levantamos las interfaces de los distintos namespaces y también la interfaz física de nuestra pc:

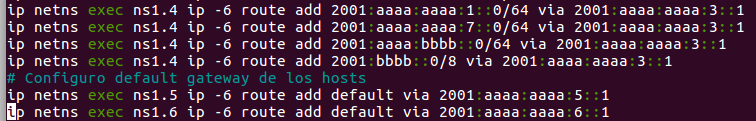


Habilitamos el forwarding para los namespaces 1.1, 1.2, 1.3, 1.4, 2.1, 2.2, 2.3 y 2.4 que son los namespaces que se tienen que comportar como routers:



Configuramos el ruteo estático en los routers mencionados anteriormente y además configuramos el default gateway para los namespaces 1.5, 1.6, 2.5 y 2.6 que son los que simulan hosts.

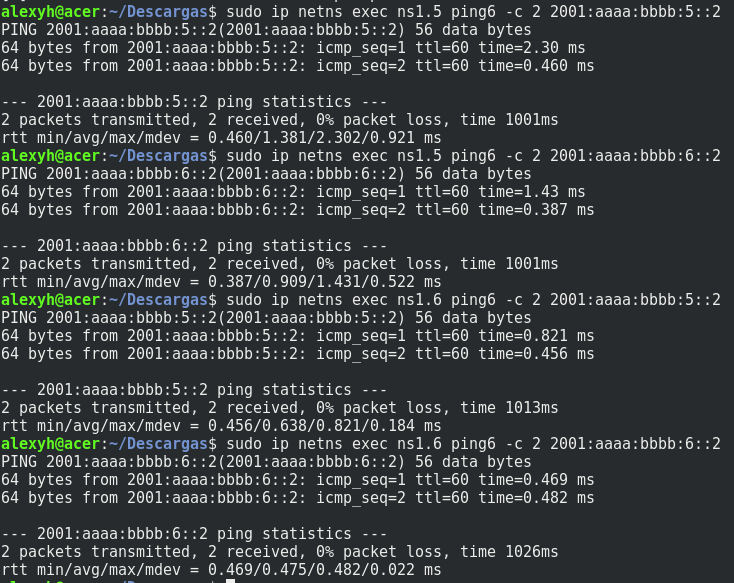




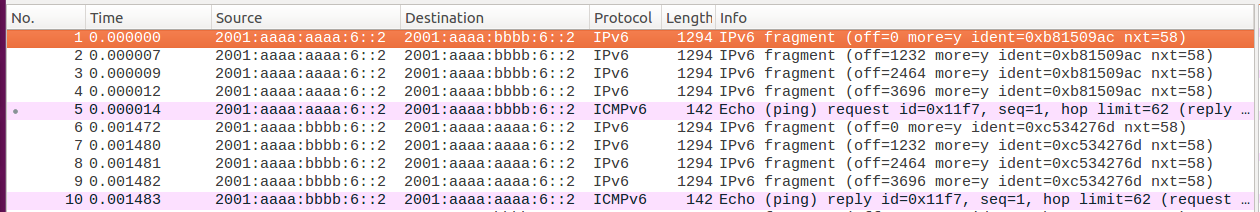
Al momento de querer configurar el MTU de una interfaz del namespace 1.1 y 2.1 a 500 nos generaba problema ya que se borraba la dirección IP de esa interfaz y esto evitaba que haya comunicación entre las 2 computadoras. Según el rfc 2460, sección 5, no se pueden configurar MTU inferiores a 1280, por lo que se configuró con ese valor:



Podemos verificar que hay conectividad entre los distintos namespaces realizando pings entre cualquier punto de la red con otro:



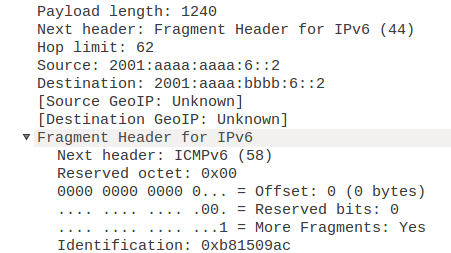
10) Para analizar la fragmentación de paquetes, vamos a enviar un paquete ICMPv6 de un tamaño mayor al MTU del enlace entre las 2 computadoras, para eso enviamos ping agregando al final del mismo la opción -s 5000 y podemos analizar con wireshark los distintos paquetes.



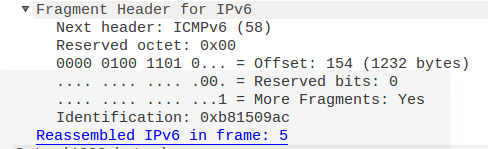
Aca podemos ver que que el paquete de 5000 bytes lo divide en 5 paquetes.

Si analizamos la cabecera de cada uno de estos fragmentos vamos a ver 2 campos interesantes, uno de esos es el campo Offset el cual nos indica cuando bytes ya se enviaron en fragmentos anteriores, si el Offset fuera 0 nos indicaría que es el primer fragmento del paquete enviado. El segundo campo importante es el More Fragments, que dirá Yes indicando que vienen más fragmentos del mismo paquete, o No indicando que es el último fragmento del paquete.

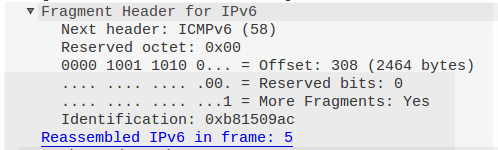
Cabecera del primer fragmento:



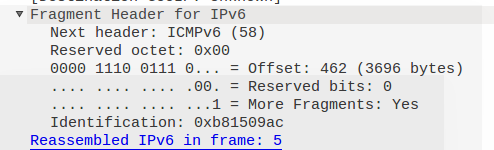
Del segundo fragmento:



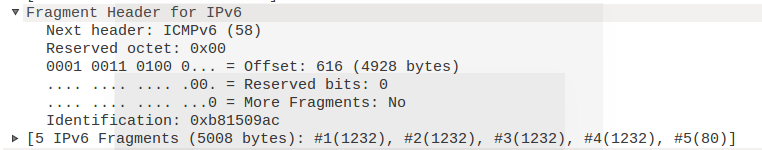
Del tercer fragmento:



Del cuarto fragmento:



Del último fragmento:

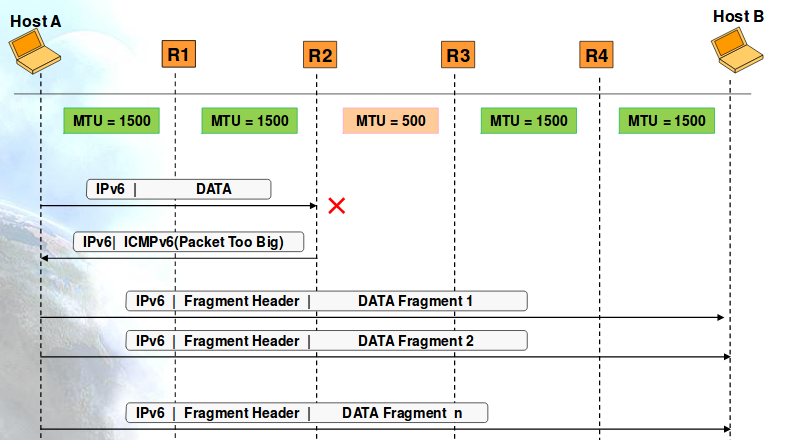


En donde efectivamente, el More fragments está en No indicando que no vienen más fragmentos.

Esto mismo ocurre para el echo reply.

11) La fragmentación de paquetes IP es un mecanismo para separar un paquete de IP grande en varios paquetes de menor tamaño. En IPv4 la fragmentación se puede dar en un host o en un router del camino que recorre el paquete entre su origen y destino cuando el tamaño del paquete es mayor al MTU del enlace.

En cambio, en IPv6 los routers no fragmentan los paquetes, por lo que toda la fragmentación se debe generar en los hosts. Cuando un paquete llega a un router con un enlace que tiene un MTU menor al tamaño del paquete, se envía un mensaje ICMPv6 al host de origen para que fragmente el paquete y lo vuelva a enviar, como se puede ver en la siguiente imagen:



El reensamblado se realiza en el host de destino, usando el campo offset para ordenar los distintos fragmentos del paquete original.

**REFERENCIAS:**

<http://www.tutorialspoint.com/unix_commands/ping6.htm>

<https://www.youtube.com/watch?v=_WgUwUf1d34>

<http://man7.org/linux/man-pages/man4/veth.4.html>

<https://wiki.debian.org/BridgeNetworkConnections>

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

<https://es.wikipedia.org/wiki/Cabecera_IP>

<http://www.6deploy.eu/workshops2/20111010_guayaquil_ecuador/DIA1-1-2-Consulintel_Curso-IPv6_WALC2011.pdf>