

# ARQUITECTURAS DE REDES AVANZADAS

## PRÁCTICA 1

### ENERO 2015

José Luis Cánovas Sánchez  
jose Luis.canovas2@um.es  
48636907A

#### Resumen

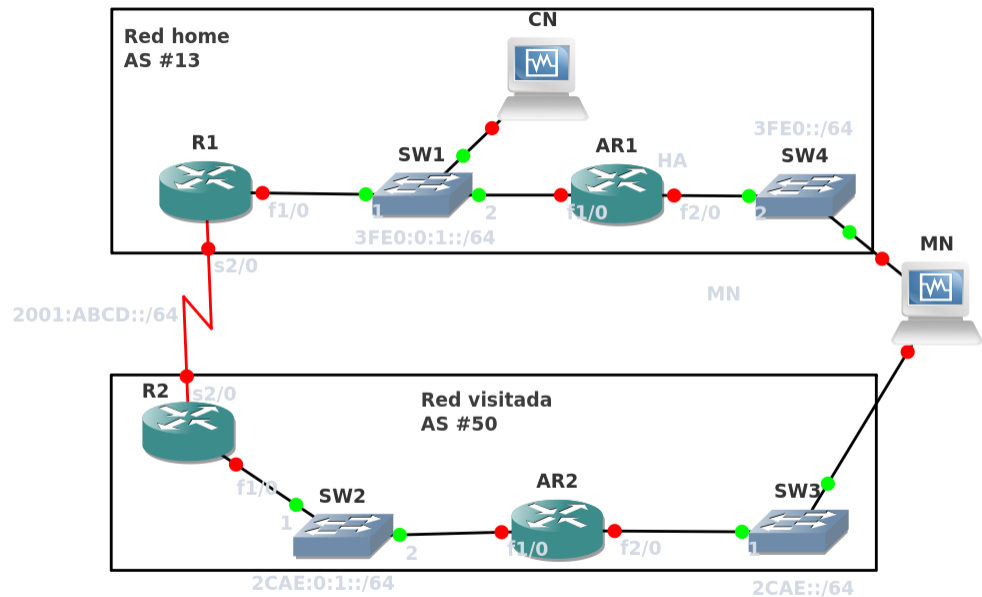
En este informe se redacta el desarrollo usando la herramienta GNS3 para el despliegue del escenario de red IPv6 con movilidad. Se incluyen de manera opcional la configuración de BGP entre R1 y R2, y el análisis de trazas de Neighbour Discovery.

## Índice

<b>1. Topología inicial</b>	<b>1</b>
<b>2. Configuración opcional BGP</b>	<b>3</b>
<b>3. Configuración Home Agent</b>	<b>5</b>
<b>4. Correspondent Node</b>	<b>5</b>
<b>5. Mobile Node: MIP6D</b>	<b>6</b>
<b>6. Ping6 CN→N</b>	<b>6</b>
<b>7. Traceroute6 CN→N</b>	<b>11</b>
<b>8. Análisis opcional Neighbour Discovery</b>	<b>12</b>
<b>A. Ping6 salida por pantalla</b>	<b>14</b>

## 1. Topología inicial

Partimos en este proyecto creando una nueva topología en GNS3 equivalente a la del enunciado y que se puede ver en la imagen inferior. Para los routers se utiliza la ios del modelo C7200 de CISCO, con soporte para IPv6. Para los hosts, una imagen linux sin interfaz gráfica y con soporte de mobile IPv6 por mip6d.



La configuración básica para que todos los dispositivos y redes tengan conectividad IPv6 entre ellas se puede realizar por medio del fichero de configuración del router directamente, y su equivalente en comandos de la terminal suelen ser las mismas líneas aplicadas en el nivel *configure terminal*. En cuanto a los hosts, como utilizan la autoconfiguración de IPv6 no hace falta configurar manualmente nada, a lo sumo en */etc/network/interfaces* que la interfaz tenga en su configuración *inet6 auto*.

Veamos primero la configuración común de todos los routers:

Para activar IPv6 es necesario indicarlo antes de la configuración de las interfaces con *ipv6 unicast-routing*. En cada interfaz que activemos, debemos indicar *ipv6 enable* y la orden *ipv6 address diripv6*, donde *diripv6* es la dirección IPv6 asignada manualmente, o el prefijo de red seguido de *eui-64* para permitir que la interfaz se autoconfigure su dirección.

Con esto tendríamos activado IPv6 en cada router, pero al intentar hacer *ping* a una interfaz de otra red, vemos que no se recibe respuesta. Esto es porque no hay información en las tablas de rutas más que de las redes conectadas directamente a las interfaces de cada router.

Para cada Sistema Autónomo vamos a usar OSPFv3 con soporte de IPv6. El diseño será de una única área 0 para todo el AS, lo que incluye las subredes en AS#13 3FE0:0:1::/64 y 3FE0::/64, y en AS#50 las subredes 2CAE:0:1::/64 y 2CAE::64. Activamos OSPF en las dos interfaces de los routers AR1 y AR2, y en la interfaz FastEthernet1/0 de los routers R1 y R2 que se pueden ver en la figura de la topología más arriba.

Las órdenes para activar OSPF en cada router serían:

Dentro de una interfaz que deba soportar OSPF (está dentro del área 0):

```
ipv6 ospf 100 area 0
```

Y tras la configuración de las interfaces, activamos el proceso de ospf:

```
ipv6 router ospf 100
```

```
router-id a.b.c.d
```

Donde *a.b.c.d* es un valor identificador del router con 32 bits que se puede escribir como una dirección IPv4. Para cada router hemos elegido los identificadores *1.1.1.1*, *2.2.2.2*, *3.3.3.3* y *4.4.4.4* para R1, R2, AR1 y AR2 respectivamente. El valor 100 es indicador del proceso OSPF en el router y puede tomar cualquier otro valor, el tomar 100 es arbitrario.

Llegados a este punto, tenemos conectividad dentro de cada área, de modo que ahora un ping desde R1 a la interfaz de AR1 en la subred 3FE0::/64 funciona, pero de AR1 a R2 o AR2 no, porque no se conoce la ruta. Ahora mismo sólo R1 y R2 podrían hacer ping a las interfaces de la red 2001:ABCD::/64.

Para solucionarlo la forma más rápida es añadir una ruta estática en R1 y R2 a las redes del otro AS con la línea: *ipv6 route 2CAE::/42 Serial2/0* en R1, y la equivalente *ipv6 route 3FE0::/42 Serial2/0* en R2. Finalmente, en la configuración de OSPF, tras el *router-id* añadimos las líneas, sólo en R1 y R2, *redistribute connected* para anunciar la subred 2001:ABCD::/64, y *redistribute static*, para que comuniquen con LSA5 (rutas fuera de dominio) que para alcanzar el otro AS a través de ellos hay una ruta.

Ahora sí se puede realizar un ping desde cualquier dispositivo a cualquier otro, una vez OSPF ha convergido y compartido todos los mensajes. Pero como la mejora para configurar BGP entre R1 y R2 sobrescribe la configuración estática, mostraré las pruebas en el apartado siguiente.

## 2. Configuración opcional BGP

En los routers R1 y R2 debemos comentar o borrar las líneas de las rutas estáticas y su distribución en OSPF. A continuación, tras la configuración de OSPF podemos escribir la de BGP:

Para R1:

```
1 router bgp 13
2     bgp router-id 1.1.1.1
3     no bgp default ipv4-unicast
4     !--- Without configuring "no bgp default ipv4-unicast" only IPv4 will be !---
5     advertised
6     bgp log-neighbor-changes
7     neighbor 2001:ABCD::2 remote-as 50
8 !
9 address-family ipv6
10     neighbor 2001:ABCD::2 activate
11     network 3FE0:0:1::/64
12     network 3FE0::/64
13 exit-address-family
```

Donde 13 indica el número de Sistema Autónomo y 1.1.1.1 es un identificador equivalente al de OSPF, en este caso elegimos el mismo. La línea *no bgp default ipv4-unicast* es necesaria para permitir las rutas con

IPv6. La primera línea de *neighbor* indica que en la interfaz de dirección 2001:ABCD::2, que corresponde a la IPv6 fija de R2, se encuentra el AS 50. A continuación, en el bloque *address-family* activamos dicho vecino e indicamos que se deben anunciar las redes 3FE0:0:1::/64 y 3FE0::/64, correspondientes a las del AS 13 de la topología dada.

Para R2 la configuración sería equivalente cambiando los valores del 13 a 50 para el AS, el del *router-id*, la dirección de R1 y las subredes a compartir.

Con esto R1 y R2 conocerían las redes de cada AS, pero AR1 y AR2 todavía no, pues no hablan BGP. Sin embargo, sí hablan OSPF y añadiendo la línea en R1 y R2 *redistribute bgp 10* en la configuración de OSPF (donde antes *redistribute static*) se comunicarán las rutas externas con métrica 10 a los routers de cada AS. Podemos observarlo en la tabla de rutas de AR2, por ejemplo:

```

1 Router>show ipv6 route
2 IPv6 Routing Table - 8 entries
3 Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
4 U - Per-user Static route
5 I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
6 O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
7 ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
8 C 2CAE::/64 [0/0]
9 via ::, FastEthernet2/0
10 L 2CAE::C801:1AFF:FE5F:38/128 [0/0]
11 via ::, FastEthernet2/0
12 C 2CAE:0:1::/64 [0/0]
13 via ::, FastEthernet1/0
14 L 2CAE:0:1:0:C801:1AFF:FE5F:1C/128 [0/0]
15 via ::, FastEthernet1/0
16 OE2 3FE0::/64 [110/2]
17 via FE80::C803:1AFF:FE5F:1C, FastEthernet1/0
18 OE2 3FE0:0:1::/64 [110/1]
19 via FE80::C803:1AFF:FE5F:1C, FastEthernet1/0
20 L FE80::/10 [0/0]
21 via ::, Null0
22 L FF00::/8 [0/0]
23 via ::, Null0

```

En las líneas 16 a 19 se puede observar rutas a las redes del AS 13 conocidas por OSPF de rutas externas.

Ahora las pruebas con ping y traceroute:

Ping y traceroute de AR1 a AR2 en la interfaz de la subred 2CAE::/64:

```

1 Router>ping 2CAE::C801:1AFF:FE5F:38
2
3 Type escape sequence to abort.
4 Sending 5, 100-byte ICMP Echos to 2CAE::C801:1AFF:FE5F:38, timeout is 2 seconds:
5 !!!!!
6 Success rate is 100 percent (5/5), round-trip min/avg/max = 60/65/84 ms

```

```
1 Router>traceroute 2CAE::C801:1AFF:FE5F:38
2
3 Type escape sequence to abort.
4 Tracing the route to 2CAE::C801:1AFF:FE5F:38
5
6  1 3FE0:0:1:0:C802:1AFF:FE5F:1C 16 msec 20 msec 20 msec
7  2 2001:ABCD::2 40 msec 40 msec 40 msec
8  3 2CAE::C801:1AFF:FE5F:38 60 msec 60 msec 60 msec
9 Router>
```

Donde 3FE0:0:1:0:C802:1AFF:FE5F:1C es la IP autoconfigurada de R1 en la subred 3FE0:0:1::/64, 2001:ABCD::2 es la IP manual de R2 en la subred 2001:ABCD::/64 y finalmente 2CAE::C801:1AFF:FE5F:38 la IP de AR2.

Considerando que para llegar de AR1 a AR2 con ping y traceroute todos los routers deben conocer el camino en un sentido u otro, con esta única prueba podemos concluir que la configuración de la topología básica con OSPF y BGP es correcta.

### 3. Configuración Home Agent

Según mi topología, el router AR1 por la interfaz FastEthernet2/0 debería actuar como Home Agent para la subred 3FE0::/64. Para activar el proceso del router que actúe como Home Agent basta incluir la siguiente línea en la configuración de la interfaz: `ipv6 mobile home-agent`

Además, incluimos las siguientes configuraciones para el neighbor discovery del router por dicha interfaz, para mejorar la movilidad, al incrementar los anuncios en la subred del router, de modo que el Mobile Node detecta antes los cambios:

```
ipv6 nd prefix 3FE0::/64
ipv6 nd ra-interval 3
```

La primera línea es obvia, indica el prefijo a anunciar, el de la subred. La segunda línea indica el intervalo del Router Advertisement en segundos, en este caso con 3 segundos parece razonable. Para indicarlo en milisegundos habría que añadir `msec` en la línea.

Como el Mobile Node en la prueba pasará a la subred de AR2, incrementamos también el intervalo del Router Advertisement en su interfaz FastEthernet2/0 (la de la red visitada).

### 4. Correspondent Node

El CN no es más que una máquina cualquiera conectada en la red 3FE0:0:1::/64, entre R1 y AR1. La IP que en la ejecución de las pruebas se ha autoconfigurado (por EUI64) es 3fe0:0:1:0:a00:27ff:fef1:1d6d.

No necesita configuración especial para la red. Como máquina virtual de Virtual Box en GNS3, se ha configurado con la interfaz eth0 para el NAT con salida a internet, y eth1 conectado al switch de la topología.

En `/etc/network/interfaces` se definen ambas interfaces, y a `eth1` se le añade la línea `iface eth1 inet6 auto` de modo que se configura automáticamente una dirección IPv6.

## 5. Mobile Node: MIP6D

El MN es otra máquina virtual conectada a GNS3 con `eth0` para el NAT y conectividad a internet, como el CN, y añadimos `eth1` y `eth2` con `inet6 auto` y conectadas respectivamente a las subredes `3FE0::/64` y `2CAE::/64`. La configuración del demonio de movilidad se realiza con el fichero `/usr/local/etc/mip6d.conf` que se muestra a continuación:

```
1 NodeConfig MN;
2 DebugLevel 1;
3 OptimisticHandoff enabled;
4 DoRouteOptimizationMN disabled;
5 MnMaxHaBindingLife 60;
6 Interface "eth1" {
7     MnIfPreference 1;
8 }
9 Interface "eth2" {
10     MnIfPreference 2;
11 }
12 MnHomeLink "eth1" {
13     HomeAgentAddress 3FE0::1;
14     HomeAddress 3fe0::a00:27ff:feed:7e79/64;
15 }
16 UseMnHaIPsec disabled;
17 KeyMngMobCapability disabled;
```

La primera línea indica al demonio que debe actuar como Mobile Node, mostrando mensajes de debug (0 para no mostrar ninguno), sin optimización de ruta y un tiempo de vida de un binding con el Home Agent de 60 segundos.

A continuación se indican las interfaces que formarán parte de mobile ipv6, en este caso `eth1` y `eth2`, con preferencias de a menor valor se elige antes, de modo que de estar disponibles ambas interfaces, se elige antes `eth1` (la conectada directamente a la red del HA).

Indicamos también la dirección IPv6 del router AR1 por su interfaz `FastEthernet2/0` (IP fija) y la dirección Home Address del MN.

Finalmente indicamos que no haya túnel IPsec entre el Mobile Node y Home Agent.

Para iniciar el demonio basta hacer `sudo mip6d -c /usr/local/etc/mip6d.conf`.

## 6. Ping6 CN→N

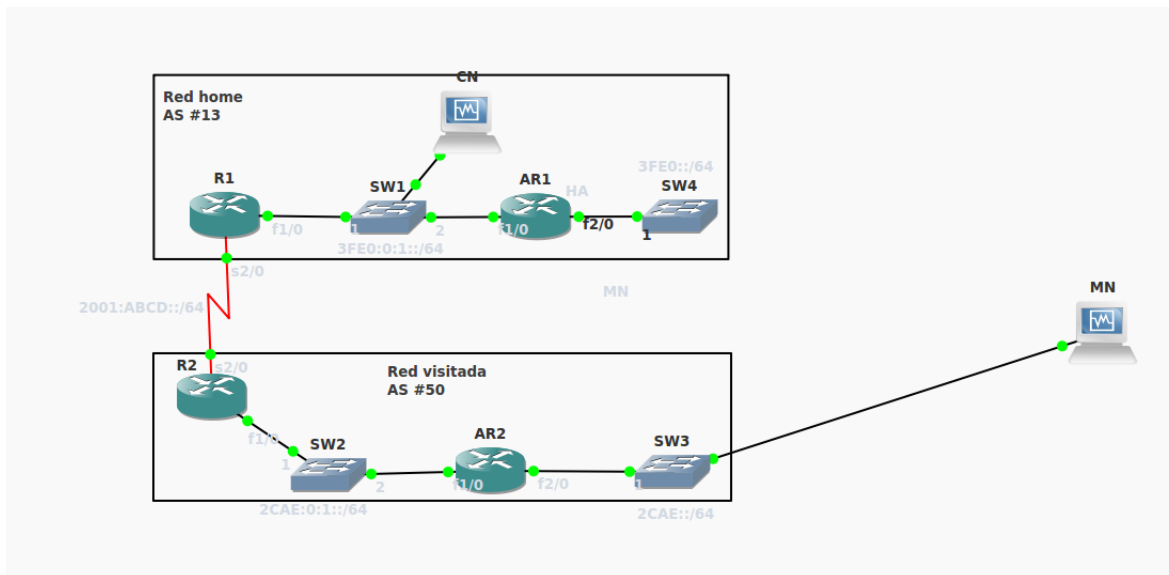
Ejecutamos la orden en la máquina del CN:

```
ping6 -I eth1 3fe0::a00:27ff:feed:7e79
```

En las trazas capturadas tenemos:

En el fichero de captura de la interfaz FE2/0 de AR1 tenemos desde la traza número 75 con el ping request *icmp\_seq=1* hasta la traza 161 con el ping reply del *icmp\_seq=33*. En la traza 164 vemos el ping request *icmp\_seq=34* pero sin respuesta en el resto de la captura.

Esto último ocurre cuando se corta el cable en la interfaz de GNS3 que une el MN con la subred de AR1:



Observamos en la salida de *ping6* (ver anexo) que en los primeros mensajes tenemos un tiempo cercano a los 20ms de respuesta, y en el mensaje *icmp\_seq=35* pasamos a tiempos de varios cientos, incluso mil milisegundos, y en el proceso hemos perdido el mensaje *icmp\_seq=34*

En el router AR1 podemos ver también el listado de bindings del Home Agent:

```
Router#show ipv6 mobile binding
Mobile IPv6 Binding Cache Entries:

 3FE0::A00:27FF:FEED:7E79
  link local address FE80::A00:27FF:FEED:7E79
  via care-of address 2CAE::A00:27FF:FE7C:6171
  home-agent 3FE0::1
  state ACTIVE, sequence 43521, flags AHLk
  lifetime: remaining 30 (secs), granted 60 (secs), requested 60 (secs)
  interface FastEthernet2/0
  85 tunneled, 61 reversed tunneled
Selection matched 1 bindings
Router#
```

Observamos la dirección del Home Address del MN (la dirección a la que hacemos ping6); la Care-of Address, que pertenece a la subred de AR2 a la que está conectado el MN por la interfaz eth2; la dirección del Home-Agent, la de AR1 por la interfaz FE2/0 que fijamos antes. También notamos cómo tiene en lifetime un tiempo de 60 segundos de vida para cada binding, los configurados en *mip6d.conf*.

Ahora veamos las trazas capturadas en el corte:

En la captura realizada en AR2 por la interfaz FE2/0 (a la que se conecta el MN) filtramos todos los mensajes *ICMPv6* y vemos que el primer ping que se registra es el *icmp\_seq=35* en la traza 94, que le llega al

CN con un ping de 410ms. Ahora filtramos por *mip6* y vemos que en las trazas 90 y 93 están los mensajes de binding update y binding ack correspondientes al corte, justo antes del primer mensaje de ping. Además en las trazas 292 y 295 tenemos otro par de mensajes de binding que coinciden por la marca de tiempo a la caducidad de un minuto de cada binding.

No.	Time	Source	Destination	Protocol	Length	Info
90	185.993263	3fe0::a00:27ff:feed:7e79	3fe0::1	MIPv6	110	Binding Update
93	187.070321	3fe0::1	3fe0::a00:27ff:feed:7e79	MIPv6	94	Binding Acknowledgement
292	243.097319	3fe0::a00:27ff:feed:7e79	3fe0::1	MIPv6	110	Binding Update
295	243.168943	3fe0::1	3fe0::a00:27ff:feed:7e79	MIPv6	94	Binding Acknowledgement

```

+ Frame 90: 110 bytes on wire (880 bits), 110 bytes captured (880 bits)
+ Ethernet II, Src: CadmusCo_7c:61:71 (08:00:27:7c:61:71), Dst: ca:01:44:8d:00:38 (ca:01:44:8d:00:38)
+ Internet Protocol Version 6, Src: 2cae::a00:27ff:fe7c:6171 (2cae::a00:27ff:fe7c:6171), Dst: 3fe0::1 (3fe0::1)
- Mobile IPv6 / Network Mobility
  Payload protocol: IPv6 no next header (59)
  Header length: 3 (32 bytes)
  Mobility Header Type: Binding update (5)
  Reserved: 0x00
  Checksum: 0x05ea
  - Binding Update
    Sequence number: 59271
    1... .. = Acknowledge (A) flag: Binding Acknowledgement requested
    .1.. .. = Home Registration (H) flag: Home Registration
    ..1. .... = Link-Local Compatibility (L) flag: Link-Local Address Compatibility
    ...0 .... = Key Management Compatibility (K) flag: No Key Management Mobility Compatibility
    .... 0... .. = MAP Registration Compatibility (M) flag: No MAP Registration Compatibility
    .... .0.. .... = Mobile Router (R) flag: No Mobile Router Compatibility
    .... ..0. .... = Proxy Registration (P) flag: No Proxy Registration
    .... ...0 .... = Forcing UDP encapsulation (F) flag: No Forcing UDP encapsulation
    .... .... 0... .. = TLV-header format (T) flag: No TLV-header format
    .... .... .0.. .... = Bulk-Binding-Update flag (B): Disable bulk binding update support
    Lifetime: 15 (60 seconds)
  - Mobility options
    - PadN
      Mobility option: PadN (1)
      Length: 0
    - Alternate Care-of Address
      Mobility option: Alternate Care-of Address (3)
      Alternate care-of address: 2cae::a00:27ff:fe7c:6171 (2cae::a00:27ff:fe7c:6171)

```

```

+ Frame 90: 110 bytes on wire (880 bits), 110 bytes captured (880 bits)
+ Ethernet II, Src: CadmusCo_7c:61:71 (08:00:27:7c:61:71), Dst: ca:01:44:8d:00:38 (ca:01:44:8d:00:38)
+ Internet Protocol Version 6, Src: 2cae::a00:27ff:fe7c:6171 (2cae::a00:27ff:fe7c:6171), Dst: 3fe0::1 (3fe0::1)
  + 0110 .... = Version: 6
  + .... 0000 0000 .... .. = Traffic class: 0x00000000
  .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
  Payload length: 56
  Next header: IPv6 destination option (60)
  Hop limit: 64
  Source: 2cae::a00:27ff:fe7c:6171 (2cae::a00:27ff:fe7c:6171)
  [Source SA MAC: CadmusCo_7c:61:71 (08:00:27:7c:61:71)]
  Destination: 3fe0::1 (3fe0::1)
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]
  - Destination option
    Next header: Mobile IPv6 (135)
    Length: 2 (24 bytes)
  + IPv6 option (PadN)
    - IPv6 Option (Home Address)
      Type: Home Address (201)
      Length: 16
      Home Address: 3fe0::a00:27ff:feed:7e79 (3fe0::a00:27ff:feed:7e79)
  - Mobile IPv6 / Network Mobility

```

La imagen anterior es el primer binding update que manda el MN y vemos que el origen real del paquete



IPv6 es de la subred 2CAE::/64, de AR2, la red visitada, pero tiene una destination option de tipo *home address* que indica al home agent que su dirección Home Address es la 3fe0::a00:27ff:feed:7e79. Así en la lista de trazas la fuente del mensaje parece que es de la subred 3FE0::/64 pero en verdad es de 2CAE::/64.

```

+ Frame 93: 94 bytes on wire (752 bits), 94 bytes captured (752 bits)
+ Ethernet II, Src: ca:01:44:8d:00:38 (ca:01:44:8d:00:38), Dst: CadmusCo_7c:61:71 (08:00:27:7c:61:71)
+ Internet Protocol Version 6, Src: 3fe0::1 (3fe0::1), Dst: 2cae::a00:27ff:fe7c:6171 (2cae::a00:27ff:fe7c:6171)
- Mobile IPv6 / Network Mobility
  Payload protocol: IPv6 no next header (59)
  Header length: 1 (16 bytes)
  Mobility Header Type: Binding Acknowledgement (6)
  Reserved: 0x00
  Checksum: 0xa6a6
- Binding Acknowledgement
  Status: Binding update accepted (0)
  0... .... = Key Management Compatibility (K) flag: No Key Management Mobility Compatibility
  .0.. .... = Mobile Router (R) flag: No Mobile Router Compatibility
  ..0. .... = Proxy Registration (P) flag: No Proxy Registration
  ...0 .... = TLV-header format (T) flag: No TLV-header format
  .... 0... = Bulk-Binding-Update flag (B): Disabled bulk binding update support
  Sequence number: 59271
  Lifetime: 15 (60 seconds)
- Mobility Options
  - PadN
    Mobility option: PadN (1)
    Length: 2
    PadN: 2 bytes

```

```

+ Frame 93: 94 bytes on wire (752 bits), 94 bytes captured (752 bits)
+ Ethernet II, Src: ca:01:44:8d:00:38 (ca:01:44:8d:00:38), Dst: CadmusCo_
+ Internet Protocol Version 6, Src: 3fe0::1 (3fe0::1), Dst: 2cae::a00:27f
  + 0110 .... = Version: 6
  + .... 0000 0000 .... .... .... = Traffic class: 0x00000000
  .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
  Payload length: 40
  Next header: IPv6 routing (43)
  Hop limit: 61
  Source: 3fe0::1 (3fe0::1)
  Destination: 2cae::a00:27ff:fe7c:6171 (2cae::a00:27ff:fe7c:6171)
  [Destination SA MAC: CadmusCo_7c:61:71 (08:00:27:7c:61:71)]
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]
+ Routing Header, Type : Mobile IP (2)
  Next header: Mobile IPv6 (135)
  Length: 2 (24 bytes)
  Type: Mobile IP (2)
  Segments Left: 1
  Home Address: 3fe0::a00:27ff:feed:7e79 (3fe0::a00:27ff:feed:7e79)

```

En las imágenes de la traza 93 vemos que se trata de un Binding ACK exitoso, sin mucha más información interesante por parte del binding ack, pero sí por parte de las cabeceras IPv6 previas, pues le precede una Routing Header de tipo 2, con información de la Home Address, de ahí que en el listado de trazas aparezca como dirección de destino la Home Address, mientras que el paquete IPv6 está realmente dirigido a la Care-of Address en 2CAE::/64.

Finalmente volvemos a unir el cable de eth1 con AR1. Se pierden varios mensajes, casi 50, y en el 154 nos contestan *destino inalcanzable*. Finalmente, en el mensaje 155 volvemos a recibir respuesta y además con un ping cercano a los 20ms iniciales.

No.	Time	Source	Destination	Protocol	Length	Info
386	308.441005	3fe0::a00:27ff:feed:7e79	3fe0:0:1:0:a00:27ff:fe7c:6171	ICMPv6	158	Echo (ping) reply id=0x047d, seq=149, hop limit=64
387	308.441016	3fe0::a00:27ff:feed:7e79	3fe0:0:1:0:a00:27ff:fe7c:6171	ICMPv6	158	Echo (ping) reply id=0x047d, seq=150, hop limit=64
388	308.441022	2cae::a00:27ff:fe7c:6171	3fe0::1	MIPv6	110	Binding Update
389	308.691739	fe80::c800:44ff:fe8d:38	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
390	308.701832	3fe0::1	3fe0::a00:27ff:feed:7e79	MIPv6	94	Binding Acknowledgement
391	308.708774	::	ff02::1:ffed:7e79	ICMPv6	78	Neighbor Solicitation for 3fe0::a00:27ff:feed:7e79

```

Payload length: 56
Next header: IPv6 destination option (60)
Hop limit: 64
Source: 2cae::a00:27ff:fe7c:6171 (2cae::a00:27ff:fe7c:6171)
[Source SA MAC: CadmusCo_7c:61:71 (08:00:27:7c:61:71)]
Destination: 3fe0::1 (3fe0::1)
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]
Destination Option
  Next header: Mobile IPv6 (135)
  Length: 2 (24 bytes)
  IPv6 option (PadN)
    Type: PadN (1)
    Length: 2
    PadN: 0000
  IPv6 option (Home Address)
    Type: Home Address (201)
    Length: 16
    Home Address: 3fe0::a00:27ff:feed:7e79 (3fe0::a00:27ff:feed:7e79)
Mobile IPv6 / Network Mobility
  Payload protocol: IPv6 no next header (59)
  Header length: 3 (32 bytes)
  Mobility Header Type: Binding Update (5)
  Reserved: 0x00
  Checksum: 0x05f6
  Binding Update
    Sequence number: 59274
    1... .. = Acknowledge (A) flag: Binding Acknowledgement requested
    .1. ... = Home Registration (H) flag: Home Registration
    ..1. ... = Link-Local Compatibility (L) flag: Link-Local Address Compatibility
    ...0 ... = Key Management Compatibility (K) flag: No Key Management Mobility Compatibility
    ....0... = MAP Registration Compatibility (M) flag: No MAP Registration Compatibility
    .....0. = Mobile Router (R) flag: No Mobile Router Compatibility
    .....0. = Proxy Registration (P) flag: No Proxy Registration
    .....0. = Forcing UDP encapsulation (F) flag: No Forcing UDP encapsulation
    .....0... = TLV-header format (T) flag: No TLV-header format
    .....0... = Bulk-Binding-update flag (B): Disable bulk binding update support
    Lifetime: 0 (0 seconds)
  Mobility options
    PadN
    Alternate Care-of Address
      Mobility Option: Alternate Care-of Address (3)
      Alternate care-of address: 2cae::a00:27ff:fe7c:6171 (2cae::a00:27ff:fe7c:6171)

```

En las trazas tomadas en AR1 vemos que existen únicamente dos mensajes MIPv6, y justo cuando vuelven a aparecer mensajes de ping con números de secuencia a partir del 149.

En la cabecera IPv6 vemos de nuevo el Destination Option con la Home Address, y en el binding update la información es la misma que en el analizado antes, exceptuando el Lifetime del binding, que es 0, de modo que en el Home Agent se elimina y los paquetes destinados al MN se le envían directamente de nuevo.

## 7. Traceroute6 CN→N

En la prueba de antes hacíamos ping, cortábamos el cable, y volvíamos a unirlo, mientras que el ping6 seguía funcionando. Ahora iniciamos con ping6, cortamos el cable, comprobamos que la movilidad funciona, y ejecutamos sin volver a unir el cable:

```
traceroute6 3fe0::a00:27ff:feed:7e79
```

En la salida del comando observamos que llega un primer salto a la interfaz de AR1 en la subred del CN, y en la segunda entrada AR1 contesta por la misma interfaz que no hay host, es decir, que no está la máquina MN. Ante estos resultados inesperados, he repetido la prueba numerosas veces comprobando que

las MACs de las máquinas virtuales no son las mismas, pues se reinicializaron en la clonación, que no tienen la misma IPv6, y además, ejecutando alternamente las órdenes de ping6 y traceroute6 desde el CN al MN con el cable cortado, la respuesta de traceroute es siempre la misma, y ping6 funciona sin problemas con la movilidad. En los adjuntos estás las trazas con las capturas de dichas pruebas que muestran cómo AR1 devuelve un mensaje de Time Exceeded: hop limit exceeded in transit con el primer mensaje (lo esperado de traceroute para detectar el primer salto), y un Destination Unreachable: Address unreachable, con la contestación al segundo mensaje de hop limit = 2.

En las trazas tomadas en AR2 se observa que al MN le llegan varios bloques de mensajes ping6, que corresponden con los realizados pues al alternar entre ping6 y traceroute6 sólo dejábamos 4 mensajes, y en las trazas van del seq=1 al seq=4 en varios bloques seguidos. Sin embargo, esos mismos bloques de ping en AR1 están intercalados por los mensajes UDP y de error anteriores.

Los mensajes UDP son los request de traceroute, pues la técnica por defecto de la implementación es enviar a puertos que tienen poca probabilidad de estar en escucha por una aplicación, un mensaje con destino el pedido y hop limit cada vez mayor. Para descartar que sea el uso de UDP el responsable de este error, usamos el comando traceroute -6 -I, y traceroute -6 -T que en vez de usar UDP, usan ICMP y TCP respectivamente, pero los resultados son los mismos.

Podemos concluir que existe algún error con el router Cisco usado, donde el ping funciona perfectamente pero el traceroute que llega a un MN con binding falla, pues cuando está conectado por cable el MN a AR1, todo funciona bien.

En el caso de que todo hubiera funcionado, el resultado esperado es que el traceroute mostrara al MN a 2 saltos, CN a AR1, AR1 a MN, debido a que el HA y el MN hacen encapsulación IPv6 a los mensajes, que se puede verificar en los PING capturados en AR2.

Una captura del resultado que se obtiene con el comando en todas las pruebas:

```
alumno@ubuntu:~$ traceroute6 3fe0::a00:27ff:feed:7e79
traceroute to 3fe0::a00:27ff:feed:7e79 (3fe0::a00:27ff:feed:7e79) from 3fe0:0:1:0:a00:27ff:fe1:1d6d, 30 hops max, 16 byte packets
 1  3fe0:0:1:0:c800:9ff:fe2e:1c (3fe0:0:1:0:c800:9ff:fe2e:1c)  10.759 ms  4.957 ms  8.889 ms
 2  3fe0:0:1:0:c800:9ff:fe2e:1c (3fe0:0:1:0:c800:9ff:fe2e:1c)  29.5 ms  !H  26.33 ms  !H  29.012 ms  !H
```

## 8. Análisis opcional Neighbour Discovery

A continuación analizo dos mensajes de neighbour solicitation tomados en la interfaz FE2/0 de AR1:

Neighbor Solicitation:

```

+ Frame 402: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
+ Ethernet II, Src: ca:00:44:8d:00:38 (ca:00:44:8d:00:38), Dst: IPv6mcast_ff:ed:7e:79 (3
- Internet Protocol Version 6, Src: fe80::c800:44ff:fe8d:38 (fe80::c800:44ff:fe8d:38), D
  + 0110 .... = Version: 6
  + .... 1110 0000 .... .... .... .... = Traffic class: 0x000000e0
    .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 32
    Next header: ICMPv6 (58)
    Hop limit: 255
    Source: fe80::c800:44ff:fe8d:38 (fe80::c800:44ff:fe8d:38)
    Destination: ff02::1:ffed:7e79 (ff02::1:ffed:7e79)
    [Source GeoIP: unknown]
    [Destination GeoIP: unknown]
- Internet Control Message Protocol v6
  Type: Neighbor Solicitation (135)
  Code: 0
  Checksum: 0xf1e4 [correct]
  Reserved: 00000000
  Target Address: 3fe0::a00:27ff:feed:7e79 (3fe0::a00:27ff:feed:7e79)
  - ICMPv6 Option (Source link-layer address : ca:00:44:8d:00:38)
    Type: source link-layer address (1)
    Length: 1 (8 bytes)
    Link-layer address: ca:00:44:8d:00:38 (ca:00:44:8d:00:38)

```

En el paquete IPv6 vemos que el next header es el ICMPv6 de Neighbor Solicitation, el hop limit está a 255 para comprobar que no sale del tramo de red en el que se envía, se conoce la IP propia (source) y se pregunta por una dirección multicast, que por terminar en 1:ffed:7e79 es la Solicited Node Multicast Address del MN. El origen del paquete corresponde a la dirección Link-Local de AR1, que se puede comprobar por su MAC en el último mensaje de Binding ACK visto (origen 3FE0::1).

En el mensaje ICMP en sí vemos que tiene el campo de Target Address con la dirección del MN, y como campo option adicional, el Source Link-Layer address, la dirección MAC origen de AR1 que solicita el Neighbor Advertisement.

Neighbor Advertisement:

```

+ Frame 403: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
+ Ethernet II, Src: CadmusCo_ed:7e:79 (08:00:27:ed:7e:79), Dst: ca:00:44:8d:00:3
- Internet Protocol Version 6, Src: 3fe0::a00:27ff:feed:7e79 (3fe0::a00:27ff:fee
  + 0110 .... = Version: 6
  + .... 0000 0000 .... = Traffic class: 0x00000000
  .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
  Payload length: 32
  Next header: ICMPv6 (58)
  Hop limit: 255
  Source: 3fe0::a00:27ff:feed:7e79 (3fe0::a00:27ff:feed:7e79)
  [Source SA MAC: CadmusCo_ed:7e:79 (08:00:27:ed:7e:79)]
  Destination: fe80::c800:44ff:fe8d:38 (fe80::c800:44ff:fe8d:38)
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]
- Internet Control Message Protocol v6
  Type: Neighbor Advertisement (136)
  Code: 0
  Checksum: 0x7e68 [correct]
  Flags: 0x60000000
    0... .. = Router: Not set
    .1.. .. = Solicited: Set
    ..1. .... = Override: Set
    ...0 0000 0000 0000 0000 0000 0000 0000 = Reserved: 0
  Target Address: 3fe0::a00:27ff:feed:7e79 (3fe0::a00:27ff:feed:7e79)
  - ICMPv6 Option (Target link-layer address : 08:00:27:ed:7e:79)
    Type: Target link-layer address (2)
    Length: 1 (8 bytes)
    Link-layer address: CadmusCo_ed:7e:79 (08:00:27:ed:7e:79)

```

Aquí tenemos la contestación al NS anterior. En la cabecera IPv6 vemos que el next-header es un ICMPv6, el hop limit es 255 igual que antes, y las direcciones de fuente y destino son ahora unicast, ya no se usa la multicast de Solicited Node. En concreto la IP fuente es global y la destino es link-local.

En el mensaje ICMPv6 tenemos en los flags que no es un router (es el MN en concreto), que el Advertisement es solicitado (por eso no lo envía a All-Nodes), y que se deben reescribir las entradas de las cachés con la nueva información.

Añade el Target Address con la dirección pedida del MN, y una Option con el Target Link-Layer Address, la MAC de la interfaz eth1 del MN. Esta información al final del proceso es la equivalente a la que conseguiría con ARP e IPv4.

## A. Ping6 salida por pantalla

```

PING 3fe0::a00:27ff:feed:7e79(3fe0::a00:27ff:feed:7e79)
from 3fe0:0:1:0:a00:27ff:fe1:1d6d eth1: 56 data bytes
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp_seq=1 ttl=63 time=44.3 ms
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp_seq=2 ttl=63 time=17.1 ms
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp_seq=3 ttl=63 time=20.8 ms
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp_seq=4 ttl=63 time=14.8 ms
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp_seq=5 ttl=63 time=18.6 ms
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp_seq=6 ttl=63 time=22.0 ms
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp_seq=7 ttl=63 time=15.4 ms
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp_seq=8 ttl=63 time=19.4 ms

```

64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=9 ttl=63 time=22.8 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=10 ttl=63 time=16.5 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=14 ttl=63 time=17.3 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=15 ttl=63 time=20.5 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=16 ttl=63 time=23.6 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=17 ttl=63 time=15.3 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=18 ttl=63 time=18.0 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=19 ttl=63 time=20.9 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=20 ttl=63 time=22.1 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=21 ttl=63 time=14.0 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=22 ttl=63 time=16.9 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=23 ttl=63 time=18.9 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=24 ttl=63 time=21.5 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=25 ttl=63 time=23.6 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=26 ttl=63 time=16.0 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=27 ttl=63 time=19.3 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=28 ttl=63 time=22.9 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=29 ttl=63 time=15.9 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=30 ttl=63 time=18.0 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=31 ttl=63 time=21.3 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=32 ttl=63 time=13.9 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=33 ttl=63 time=17.2 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=35 ttl=63 time=410 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=36 ttl=63 time=1067 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=37 ttl=63 time=198 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=38 ttl=63 time=202 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=39 ttl=63 time=247 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=40 ttl=63 time=583 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=41 ttl=63 time=1060 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=42 ttl=63 time=1066 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=43 ttl=63 time=1071 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=44 ttl=63 time=1064 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=45 ttl=63 time=807 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=46 ttl=63 time=1063 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=47 ttl=63 time=197 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=48 ttl=63 time=244 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=49 ttl=63 time=1062 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=50 ttl=63 time=845 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=51 ttl=63 time=86.1 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=52 ttl=63 time=1075 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=53 ttl=63 time=1073 ms

64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=54 ttl=63 time=1067 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=55 ttl=63 time=1062 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=56 ttl=63 time=172 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=57 ttl=63 time=177 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=58 ttl=63 time=222 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=59 ttl=63 time=1082 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=60 ttl=63 time=817 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=61 ttl=63 time=1074 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=62 ttl=63 time=1071 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=63 ttl=63 time=1066 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=64 ttl=63 time=1071 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=65 ttl=63 time=975 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=66 ttl=63 time=1072 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=67 ttl=63 time=155 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=68 ttl=63 time=200 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=69 ttl=63 time=1070 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=70 ttl=63 time=194 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=71 ttl=63 time=1074 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=72 ttl=63 time=790 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=73 ttl=63 time=191 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=74 ttl=63 time=1059 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=75 ttl=63 time=984 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=76 ttl=63 time=1080 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=77 ttl=63 time=121 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=78 ttl=63 time=175 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=79 ttl=63 time=1074 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=80 ttl=63 time=769 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=81 ttl=63 time=1075 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=82 ttl=63 time=1074 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=83 ttl=63 time=1070 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=84 ttl=63 time=1067 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=85 ttl=63 time=1072 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=86 ttl=63 time=311 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=87 ttl=63 time=94.3 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=88 ttl=63 time=137 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=89 ttl=63 time=111 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=90 ttl=63 time=97.1 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=91 ttl=63 time=92.0 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=92 ttl=63 time=187 ms  
From 3fe0:0:1:0:c800:44ff:fe8d:1c icmp\_seq=154 Destination unreachable: Address unreachable  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=155 ttl=63 time=21.6 ms



64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=156 ttl=63 time=14.4 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=157 ttl=63 time=18.3 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=158 ttl=63 time=21.4 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=159 ttl=63 time=14.5 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=160 ttl=63 time=17.5 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=161 ttl=63 time=21.1 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=162 ttl=63 time=14.9 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=163 ttl=63 time=17.3 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=164 ttl=63 time=20.2 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=165 ttl=63 time=13.2 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=166 ttl=63 time=16.4 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=167 ttl=63 time=19.9 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=168 ttl=63 time=22.2 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=169 ttl=63 time=15.2 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=170 ttl=63 time=18.1 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=171 ttl=63 time=11.2 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=172 ttl=63 time=14.8 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=173 ttl=63 time=16.6 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=174 ttl=63 time=18.0 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=175 ttl=63 time=21.2 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=176 ttl=63 time=14.8 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=177 ttl=63 time=17.2 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=178 ttl=63 time=20.1 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=179 ttl=63 time=13.3 ms  
64 bytes from 3fe0::a00:27ff:feed:7e79: icmp\_seq=180 ttl=63 time=15.8 ms

## Referencias

- [1] Apuntes de la asignatura y enlaces en las diapositivas de prácticas.
- [2] Documentación online de Cisco. Link ejemplo 1. Link ejemplo 2..
- [3] Manual de traceroute. Link.