



IPv6 – Autoconfiguración de Direcciones Stateless

Objetivo

Esta práctica tiene como objetivo presentar el funcionamiento de autoconfiguración Stateless de direcciones ipv6 a través del intercambio de mensajes *Router Advertisement* (RA) enviados por el software Quagga, una plataforma de ruteo para servidores UNIX, y por RADVD, un daemon para sistemas operativos Linux responsable de implementar el envío de este tipo de mensajes.

Para la realización del presente ejercicio será utilizada la topología presentada en el archivo: **Auto_conf-E1.imn**

Introducción Teórica

Autoconfiguración Stateless, a través de mensajes *Router Advertisement*, es un procedimiento utilizado por los routers para transmitir la información necesaria de autoconfiguración a los otros nodos de la red.

Esas características pueden tanto ser solicitadas por los dispositivos, con el mensaje *Router Solicitation*, o enviadas periódicamente por los routers. Independientemente del modo, luego de recibir el mensaje *Router Advertisement*, se inicia el proceso denominado *Duplicate Address Detection* para evitar la creación de direcciones duplicadas en el enlace.

Este protocolo es denominado *stateless*, dado que el dispositivo que ofrece información de configuración no mantiene el registro de estado y características del destinatario de esta información, el nodo destino se encarga de su propia configuración.

Detalle de la Práctica

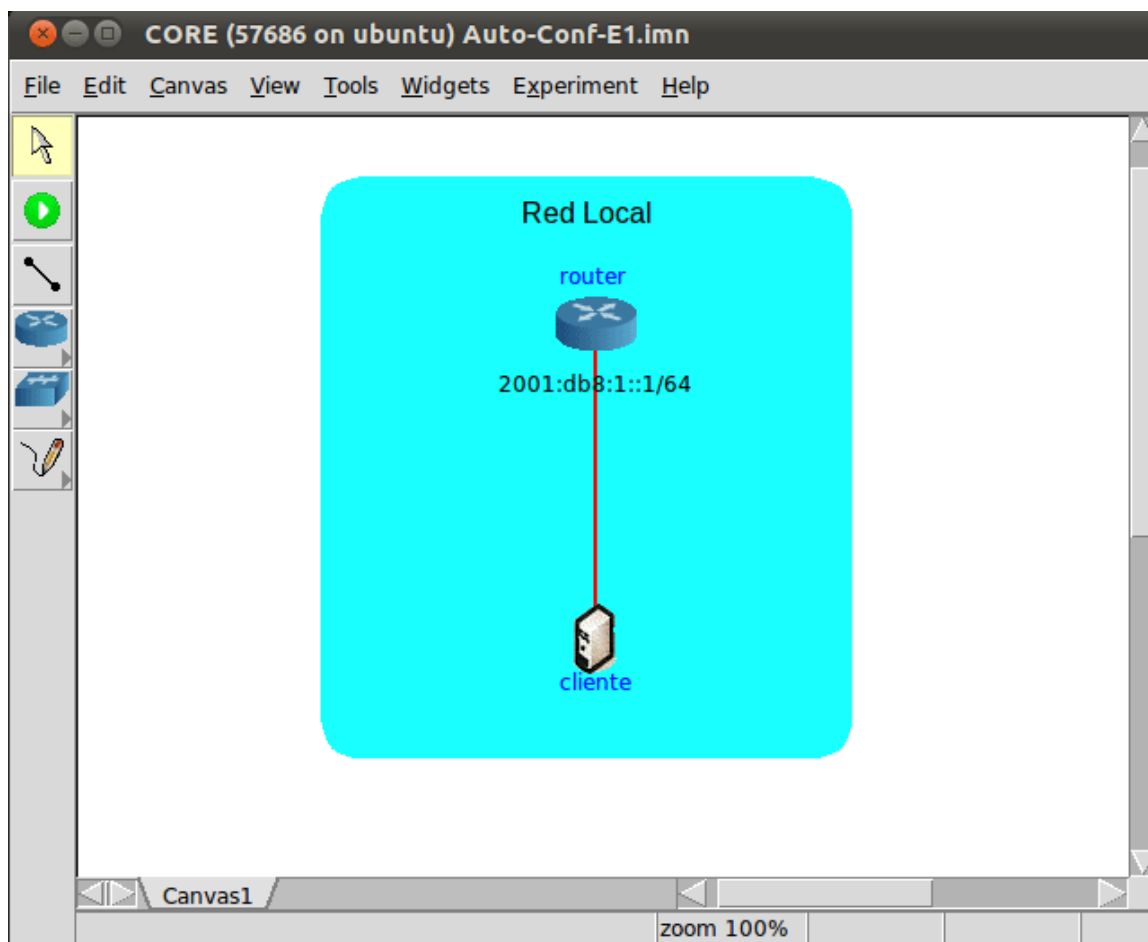
Práctica 1 - Quagga - Router Advertisement


1. En caso de que no este utilizando la maquina virtual provista por LACNIC es preciso, antes de comenzar con el experimento, instalar algún software adicional para la realización de esta practica (caso contrario continúe desde el paso 2). Siga las instrucciones siguientes para realizar la instalación:
 - Para realizar algunas verificaciones durante el experimento será necesaria la utilización del programa **Wireshark**, que mejora la visualización de paquetes transmitidos en la red. En la maquina virtual, utilice una terminal para ejecutar el comando:

```
$ sudo apt-get install wireshark
```

Antes de la instalación será solicitada una contraseña, digite “core” para continuar con la instalación.

2. Inicie CORE y abra el archivo “**Auto_Conf-E1.imn**” ubicado en el Desktop “Funcionalidades/AutoStateless”, de la maquina virtual de LACNIC. La siguiente topología de red debe aparecer:



3. Verifique la configuración de los nodos de la topología.
 - Inicie la simulación realizando uno de los siguientes pasos:
 - Pulse el boton  ; o
 - Utilice el menú Experiment > Start.
 - Espere hasta que CORE termine la inicialización de la simulación y abra el terminal del ‘cliente’ haciendo doble-click.
 - Observe la configuración del ‘cliente’ con el siguiente comando:

```
# ip addr
```

El resultado debe ser:

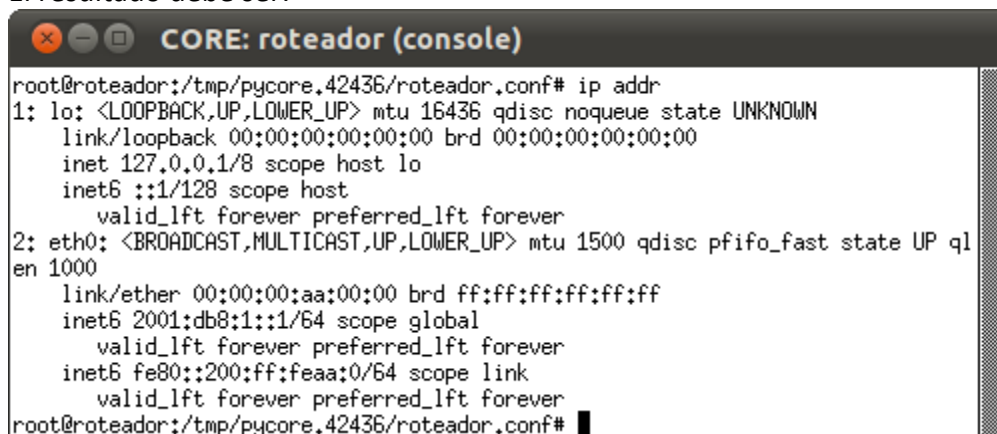


```
root@cliente:/tmp/pycore.42436/cliente.conf# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP ql
en 1000
    link/ether 00:00:00:aa:00:01 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::200:ff:feaa:1/64 scope link
        valid_lft forever preferred_lft forever
root@cliente:/tmp/pycore.42436/cliente.conf# █
```

***Obs:** Con este comando es posible ver las direcciones de las interfaces.

- Observe la configuración del 'router' con el mismo comando.

El resultado debe ser:



```
root@roteador:/tmp/pycore.42436/roteador.conf# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP ql
en 1000
    link/ether 00:00:00:aa:00:00 brd ff:ff:ff:ff:ff:ff
    inet6 2001:db8:1::1/64 scope global
        valid_lft forever preferred_lft forever
    inet6 fe80::200:ff:feaa:0/64 scope link
        valid_lft forever preferred_lft forever
root@roteador:/tmp/pycore.42436/roteador.conf# █
```

***Obs:** Con este comando es posible ver las direcciones de las interfaces.

4. Edite las configuraciones de Quagga, para que pueda enviar mensajes *Router Advertisement* que contengan información de configuración para el cliente.

- Abra una terminal en el cliente con un doble-click.
- Utilice el siguiente comando para iniciar una captura de paquetes provenientes del router:

```
# tcpdump -i eth0 -s 0 -w /tmp/captura_auto_conf_e1.pcap
```

El resultado debe ser:

```

CORE: cliente (console)
root@cliente:/tmp/pycore.42436/cliente.conf# tcpdump -i eth0 -s 0 -w /tmp/captur
a_auto_conf_e1.pcap
tcpdump: WARNING: eth0: no IPv4 address assigned
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 byte
$

```

***Obs:** No cierre esta terminal hasta el final del experimento ya que esto ocasionaría la interrupción del comando “tcpdump” y perjudicaría el resultado de la experiencia.

- Abra una terminal en el ‘router’ con un doble-click.
- Sustituya el contenido del archivo *Quagga.conf* ubicado en el directorio

```
/usr/local/etc/quagga
```

por el siguiente:

```

interface eth0
  ipv6 address 2001:db8:1::1/64
  no ipv6 nd suppress-ra
  ipv6 nd ra-interval 5
  ipv6 nd prefix 2001:db8:1::/64
!

```

***Obs:** En la versión actual de Quagga instalado en el CORE, no es posible enviar la(s) direcciones del DNS via *Router Advertisement*. Mas información sobre esa configuración puede ser encontrada en:

<http://www.nongnu.org/quagga/docs/docs-info.html#SEC140>

El resultado debe ser:

```

CORE: router (console)
root@router:/usr/local/etc/quagga# cat /usr/local/etc/quagga/Quagga.conf
interface eth0
  ipv6 address 2001:db8:1::1/64
  no ipv6 nd suppress-ra
  ipv6 nd ra-interval 5
  ipv6 nd prefix 2001:db8:1::/64
!
root@router:/usr/local/etc/quagga#

```

***Obs:** un editor presente en la máquina virtual que puede ser utilizado es el **nano**.

Para utilizarlo digite en la terminal:

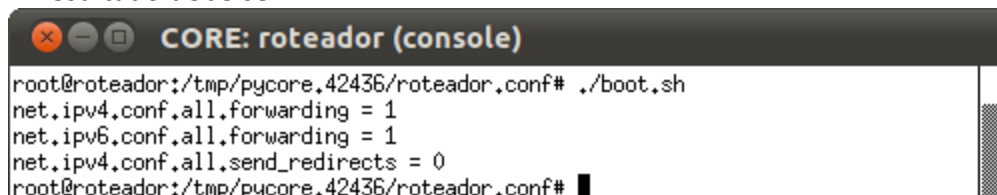
```
# nano /usr/local/etc/quagga/Quagga.conf
```

En *nano*, la secuencia utilizada para guardar el archivo es CTRL-O y para salir del editor es CTRL-X.

- Luego ejecute el script que re-inicia el servicio de Quagga. Este script se encuentra en la carpeta del 'router' (carpeta inicial cuando se abre la terminal):

```
# ./boot.sh
```

El resultado debe ser:



```

CORE: roteador (console)
root@roteador:/tmp/pycore.42436/roteador.conf# ./boot.sh
net.ipv4.conf.all.forwarding = 1
net.ipv6.conf.all.forwarding = 1
net.ipv4.conf.all.send_redirects = 0
root@roteador:/tmp/pycore.42436/roteador.conf#

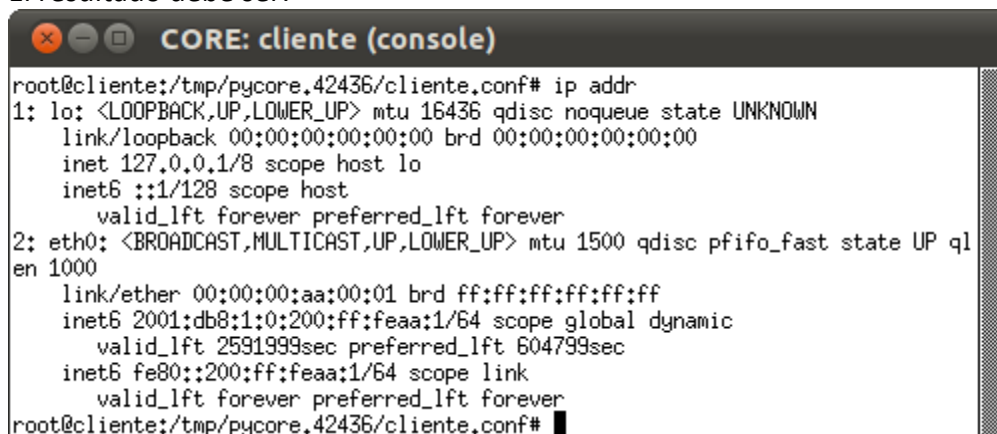
```

5. Pruebe la conectividad IPv6 desde un nodo a otro.

- Abra una terminal en el 'cliente' con un doble-click.
- Espere algunos segundos y digite el siguiente comando para observar la dirección obtenida:

```
# ip addr
```

El resultado debe ser:



```

CORE: cliente (console)
root@cliente:/tmp/pycore.42436/cliente.conf# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP ql
en 1000
    link/ether 00:00:00:aa:00:01 brd ff:ff:ff:ff:ff:ff
    inet6 2001:db8:1:0:200:ff:feaa:1/64 scope global dynamic
        valid_lft 2591999sec preferred_lft 604799sec
    inet6 fe80::200:ff:feaa:1/64 scope link
        valid_lft forever preferred_lft forever
root@cliente:/tmp/pycore.42436/cliente.conf#

```

***Obs:** Note la existencia de una dirección de alcance global en la interface eth0

- Abra una terminal en el 'router' con un doble-click
- Inicie el envío de paquetes para el nuevo ip, para chequear la conectividad a través del siguiente comando:

```
# ping6 -c 4 2001:db8:1:0:200:ff:feaa:1
```

El resultado debe ser:

```
CORE: router (console)
root@router:/tmp/pycore.57686/router.conf# ping6 -c 4 2001:db8:1:0:200:ff:feaa:1
PING 2001:db8:1:0:200:ff:feaa:1(2001:db8:1:0:200:ff:feaa:1) 56 data bytes
64 bytes from 2001:db8:1:0:200:ff:feaa:1: icmp_seq=1 ttl=64 time=2.64 ms
64 bytes from 2001:db8:1:0:200:ff:feaa:1: icmp_seq=2 ttl=64 time=0.198 ms
64 bytes from 2001:db8:1:0:200:ff:feaa:1: icmp_seq=3 ttl=64 time=0.195 ms
64 bytes from 2001:db8:1:0:200:ff:feaa:1: icmp_seq=4 ttl=64 time=0.246 ms

--- 2001:db8:1:0:200:ff:feaa:1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 0.195/0.821/2.648/1.055 ms
root@router:/tmp/pycore.57686/router.conf#
```

***Obs:** el IP debe ser el del cliente, obtenido por el comando anterior.

- En la terminal del cliente, cierre la captura de paquetes con la secuencia Ctrl+C.

El resultado debe ser:

```
CORE: cliente (console)
root@cliente:/tmp/pycore.42436/cliente.conf# tcpdump -i eth0 -s 0 -w /tmp/captura_auto_conf_e1.pcap
tcpdump: WARNING: eth0: no IPv4 address assigned
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
^C82 packets captured
82 packets received by filter
0 packets dropped by kernel
root@cliente:/tmp/pycore.42436/cliente.conf#
```

***Obs:** La cantidad de paquetes puede variar de acuerdo al tiempo que se demore en dar el comando Ctrl+C.

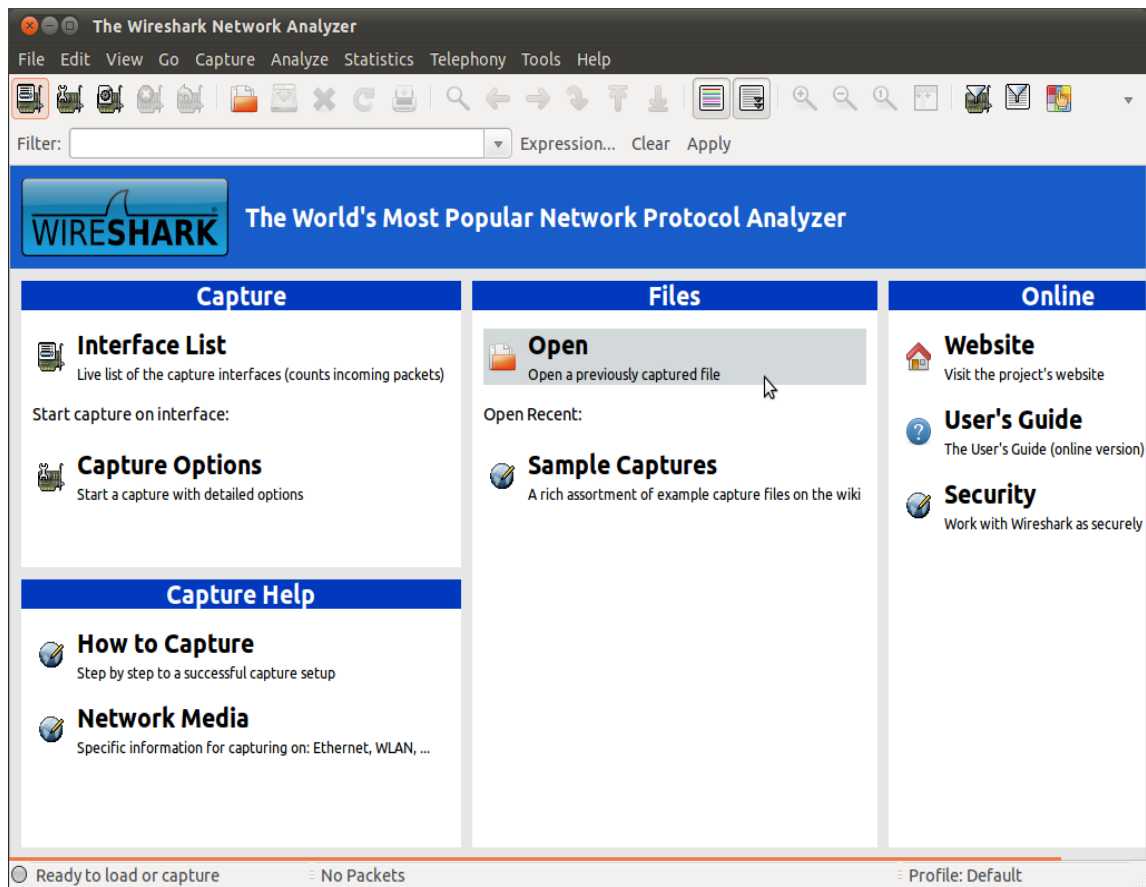
6. Cierre la simulación con una de las siguientes acciones:

- Pulse el botón  ; o
- Utilice el menú Experiment > Stop

7. La verificación de los paquetes capturados será realizada con el programa Wireshark. Para iniciarlo ejecute el siguiente comando en la maquina virtual:

```
$ wireshark
```

Si utiliza la maquina virtual provista por LACNIC puede iniciar la aplicación haciendo doble-click sobre el ícono de la misma ubicado en el Escritorio.



- Abra el archivo **/tmp/captura_auto_conf_e1.pcap** con el menú *File>Open*:
- Busque un paquete de *Router Advertisement* que contenga el protocolo opcional *Prefix Information*. Analice y vea los datos contenidos en el paquete y compare con lo que fue visto en la teoría.

Router Advertisement:

No.	Time	Source	Destination	Protocol	Info
10	0.000444	fe80::200:ff:feaa:0	ff02::1	ICMPv6	Router advertisement from 00:00:00:aa:00:00
11	1.001133	fe80::200:ff:feaa:0	ff02::1	ICMPv6	Router advertisement from 00:00:00:aa:00:00
12	1.005503	fe80::9c76:9aff:fed7:bdcb	ff02::16	ICMPv6	Multicast Listener Report Message v2
13	1.481517	::	ff02::1:ffd4:bbc0	ICMPv6	Neighbor solicitation for 2001:db8:1:0:4c53:f9ff:f...
14	1.665501	::	ff02::1:ffaa:1	ICMPv6	Neighbor solicitation for 2001:db8:1:0:200:ff:feaa...
16	2.489499	fe80::9c76:9aff:fed7:bdcb	ff02::16	ICMPv6	Multicast Listener Report Message v2
24	6.005873	fe80::200:ff:feaa:0	ff02::1	ICMPv6	Router advertisement from 00:00:00:aa:00:00
26	7.845585	fe80::9c76:9aff:fed7:bdcb	ff02::16	ICMPv6	Multicast Listener Report Message v2

▶ Frame 11: 110 bytes on wire (880 bits), 110 bytes captured (880 bits)
 ▶ Ethernet II, Src: 00:00:00 aa:00:00 (00:00:00:aa:00:00), Dst: IPv6mcast 00:00:00:01 (33:33:00:00:00:01)
 ▶ Internet Protocol Version 6, Src: fe80::200:ff:feaa:0 (fe80::200:ff:feaa:0), Dst: ff02::1 (ff02::1)
 ▼ Internet Control Message Protocol v6
 Type: 134 (Router advertisement)
 Code: 0
 Checksum: 0x8843 [correct]
 Cur hop limit: 64
 ▶ Flags: 0x00
 Router lifetime: 30528
 Reachable time: 0
 Retrans timer: 0
 ▶ ICMPv6 Option (Prefix information)
 ▶ ICMPv6 Option (Source link-layer address)

```

0000  33 33 00 00 00 01 00 00 00 aa 00 00 86 dd 60 00  33.....
0010  00 00 00 38 3a ff fe 80 00 00 00 00 00 02 00  ..8:....
0020  00 ff fe aa 00 00 ff 02 00 00 00 00 00 00 00  ..C@.w@..
0030  00 00 00 00 00 01 86 00 88 43 40 00 77 40 00 00
  
```

*Obs: el filtro icmpv6 puede ser usado para filtrar los mensajes.

Campos importantes:

- **Destination (Ethernet):** el destino es la dirección MAC (33:33:00:00:00:01) siendo que el prefijo 33:33 indica que el mensaje es un multicast en la capa Ethernet y que el sufijo 00:00:00:01 indica los últimos 32 bits de la dirección multicast IPv6 del mensaje.
- **Source (Ethernet):** el origen es la dirección MAC de la interface del router que envió el mensaje (00:00:00:aa:00:00).
- **Type (Ethernet):** indica que el mensaje utiliza el protocolo IPv6 (x86dd).
- **Next Header (IPv6):** indica cual es el próximo cabezal (de extensión IPv6), en este caso, el valor 58(0x3a) se refiere a un mensaje ICMPv6.
- **Source (IPv6):** el origen es la dirección IP del link-local de la interface que envía el mensaje (fe80::200:ff:feaa:0).
- **Destination (IPv6):** el destino es la dirección *Multicast All Nodes* (FF02::1).
- **Type (ICMPv6):** indica que el mensaje es del tipo 134 (Router Advertisement).
- **ICMPv6 Option (ICMPv6):** indica las opciones del paquete ICMPv6:
 - Prefix Information
 - *Type:* contiene el valor 3 que identifica a "Prefix Information".
 - *Autonomous Address-Configuration Flag (A):* indica si el prefijo debe ser utilizado para autoconfiguración stateless (1).



- *Preferred Lifetime*: marca el tiempo, en segundos, que la dirección es preferencial, es decir, una dirección que puede ser utilizada indistintamente. El valor (0xffffffff) indica infinito.
- *Valid Lifetime*: marca el tiempo, en segundos, de expiración de la dirección generada. El valor (0xffffffff) indica infinito.
- *Prefix*: contiene el prefijo de red a ser utilizado (2001:db8:1::).
- *Prefix length*: contiene el tamaño del prefijo de red.
- Source Link Layer Address
 - *Type*: indica el tipo de dato del mensaje ICMPv6. En nuestro caso, este es del tipo “*Source link-layer address*”;
 - *Link-layer address*: indica la dirección MAC de origen del mensaje.