lacnic25

2/6 mayo - la habana, cuba

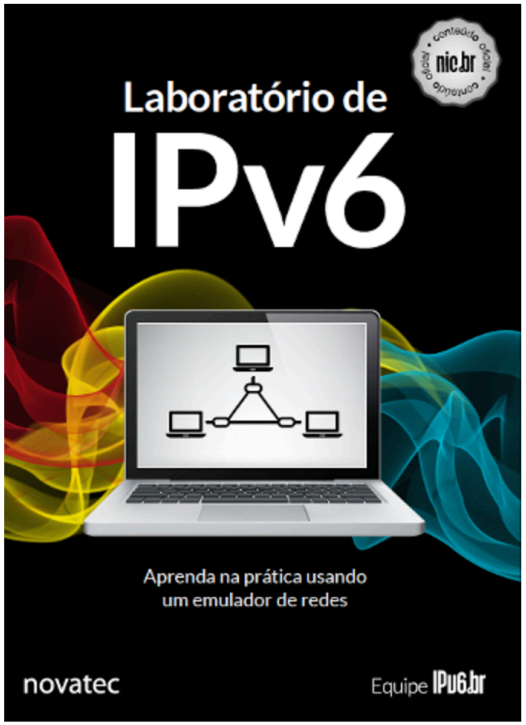
Tutorial IPv6 Básico
Prácticas de Laboratorio

Alejandro Acosta & Ariel Weher

alejandro@lacnic.net | ariel@weher.net

Materiales a usar:

http://ipv6.br/pagina/livro-ipv6/



Laboratório de IPv6

Aprenda na prática usando um emulador de redes

Equipe IPv6

- Antonio Marcos Moreiras
- Rodrigo Regis dos Santos
- Alexandre Yukio Harano
- Edwin Santos Cordeiro
- Tiago Jun Nakamura
- Eduardo Barasal Morales
- Heitor de Souza Ganzeli
- Rodrigo Matos Carnier
- Gustavo Borges Lugoboni

Ficha Técnica

ISBN: 978-85-7522-418-2
Editora: NovaTec
Ano: 2015
Número de Páginas: 416

Download .PDF

Compre online

VM para Laboratórios

Download VM (.ova)

Disponemos de copias locales de los archivos. Todo el que tenga algún pendrive de al menos 4 gigabytes, que se acerque para que le hagamos una copia

AVISO IMPORTANTE:

Demostración de uso de la VM de ipv6.br

- VM:
  - Importar la VM en el virtualizador.
  - Iniciar la VM.
  - Cambiar el idioma a inglés.
- CORE:
  - Cargar un archivo de simulación.
  - Iniciar y detener la simulación.
  - Creación de laboratorios customizados.
    - Edición de links, nodos y servicios.
- Wireshark:
  - Capturar tráfico de uno de los componentes de la simulación.
  - Analizar el tráfico.

Práctica #1:

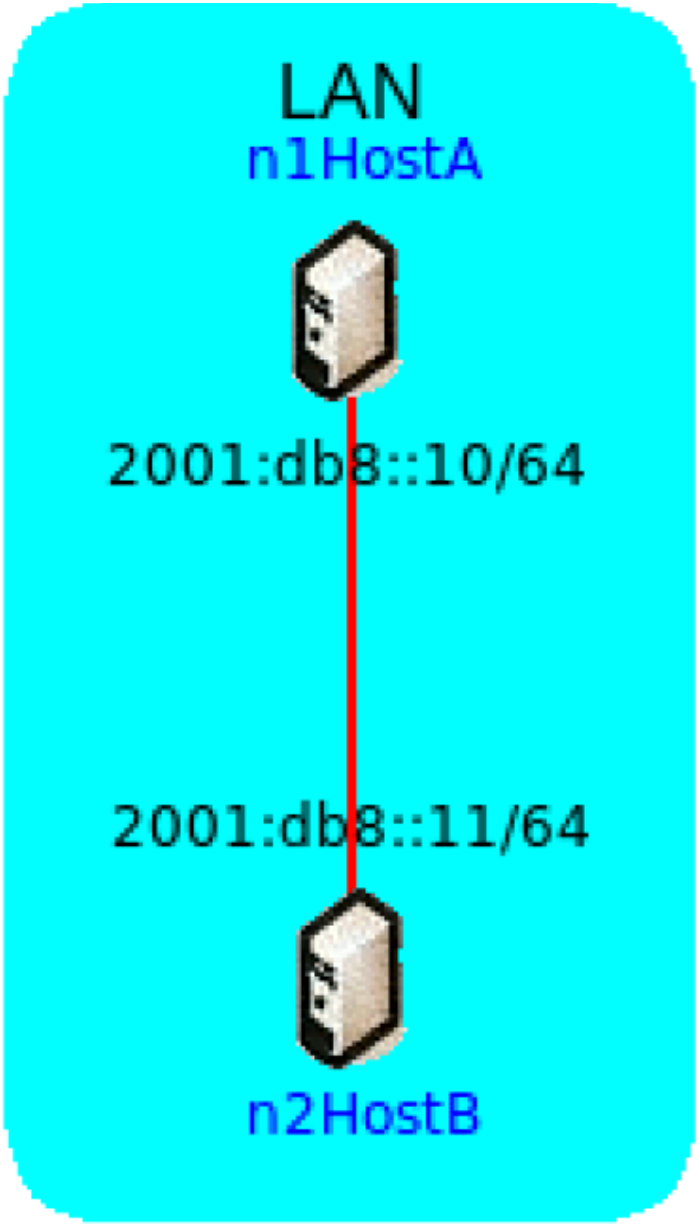
Análisis de NS & NA

Objetivo:

Capturar y reconocer paquetes de Neighbor Solicitation y Neighbor Advertisement en una red simple.

Nombre del archivo:

1-01-NSMA.imn



Actividad #1:

- Iniciar la simulación.
  - Archivo 1-01-NSMA.imn
- Capturar tráfico en la interface eth0 de n1HostA usando wireshark.
- Probar conectividad IPv6 entre n1HostA y n2HostB usando *ping6*.
    - `# ping6 2001:db8::10`
    - `# ping6 2001:db8::11`
- Analizar el tráfico capturado.

```
1 0.000000 2001:db8::10 ff02::1:ff00:11 ICMPv6 86 Neighbor Solicitation for 2001:db8::11 from 00:00:00:aa:00:00
▶ Frame 1: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
▼ Ethernet II, Src: 00:00:00_aa:00:00 (00:00:00:aa:00:00), Dst: IPv6mcast_ff:00:00:11 (33:33:ff:00:00:11)
  ▶ Destination: IPv6mcast_ff:00:00:11 (33:33:ff:00:00:11)
  ▶ Source: 00:00:00_aa:00:00 (00:00:00:aa:00:00)
    Type: IPv6 (0x86dd)
▼ Internet Protocol Version 6, Src: 2001:db8::10 (2001:db8::10), Dst: ff02::1:ff00:11 (ff02::1:ff00:11)
  ▶ 0110 .... = Version: 6
  ▶ .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00000000
    .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 32
    Next header: ICMPv6 (0x3a)
    Hop limit: 255
    Source: 2001:db8::10 (2001:db8::10)
    Destination: ff02::1:ff00:11 (ff02::1:ff00:11)
▼ Internet Control Message Protocol v6
    Type: Neighbor Solicitation (135)
    Code: 0
    Checksum: 0x1d51 [correct]
    Reserved: 00000000
    Target Address: 2001:db8::11 (2001:db8::11)
  ▼ ICMPv6 Option (Source link-layer address : 00:00:00:aa:00:00)
      Type: Source link-layer address (1)
      Length: 1 (8 bytes)
      Link-layer address: 00:00:00_aa:00:00 (00:00:00:aa:00:00)

0030  00 01 ff 00 00 11 87 00  1d 51 00 00 00 00 20 01   ........ .Q.... .
0040  0d b8 00 00 00 00 00 00  00 00 00 00 00 11 01 01   ........ ........
0050  00 00 00 aa 00 00                                  ......
```

Análisis de Paquete NS

```
2 0.000087 2001:db8::11 2001:db8::10 ICMPv6 86 Neighbor Advertisement 2001:db8::11 (sol, ovr) is at 00:00:00:aa:00:01
▶ Frame 2: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
▼ Ethernet II, Src: 00:00:00_aa:00:01 (00:00:00:aa:00:01), Dst: 00:00:00_aa:00:00 (00:00:00:aa:00:00)
  ▶ Destination: 00:00:00_aa:00:00 (00:00:00:aa:00:00)
  ▶ Source: 00:00:00_aa:00:01 (00:00:00:aa:00:01)
    Type: IPv6 (0x86dd)
▼ Internet Protocol Version 6, Src: 2001:db8::11 (2001:db8::11), Dst: 2001:db8::10 (2001:db8::10)
  ▶ 0110 .... = Version: 6
  ▶ .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00000000
    .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 32
    Next header: ICMPv6 (0x3a)
    Hop limit: 255
    Source: 2001:db8::11 (2001:db8::11)
    Destination: 2001:db8::10 (2001:db8::10)
▼ Internet Control Message Protocol v6
    Type: Neighbor Advertisement (136)
    Code: 0
    Checksum: 0x8b9b [correct]
  ▶ Flags: 0x60000000
    Target Address: 2001:db8::11 (2001:db8::11)
  ▼ ICMPv6 Option (Target link-layer address : 00:00:00:aa:00:01)
      Type: Target link-layer address (2)
      Length: 1 (8 bytes)
      Link-layer address: 00:00:00_aa:00:01 (00:00:00:aa:00:01)

0030  00 00 00 00 00 10 88 00  8b 9b 60 00 00 00 20 01   ......`. ..`... .
0040  0d b8 00 00 00 00 00 00  00 00 00 00 00 11 02 01   ........ ........
0050  00 00 00 aa 00 01                                  ......
```

Análisis de Paquete NA

Práctica #2:

Análisis de RS & RA

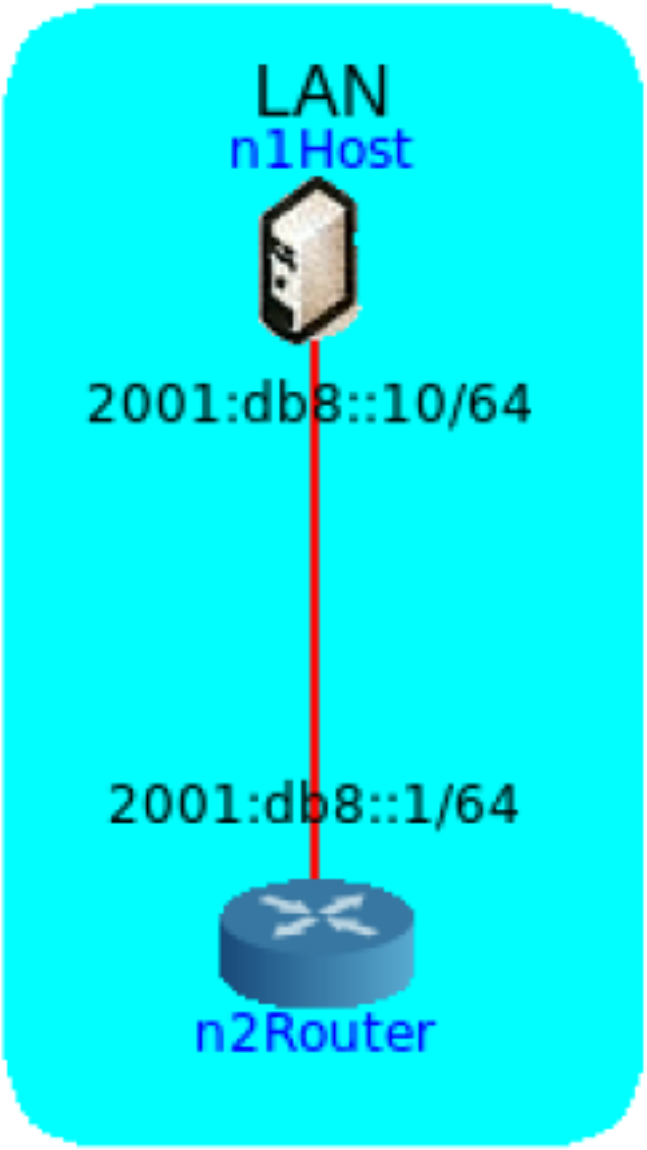
Objetivo:

Capturar y reconocer paquetes de Router Solicitation y Router Advertisement en una red simple.

En este caso el host es el que solicitará la información al router.

Nombre del archivo:

1-02-RS.imn



Actividad #2:

- Iniciar la simulación.
  - Archivo 1-02-RS.imn
- Comenzar captura tráfico en la interface eth0 de n2Router.
- Verificar las direcciones IP asignadas en el router y en el cliente.
    - `# ip -6 addr show dev eth0`
- Chequear conectividad usando *ping6*.
- Hacer caer la interface eth0 en n1Host.
    - `# ip link set eth0 down`
    - `# ip link set eth0 up`

Actividad #2:

- Verificar conectividad en n1Host.
  - `ip -6 route show`

```
n1Host
root@n1Host:/tmp/pycore.46757/n1Host.conf# ip -6 route show
fe80::/64 dev eth0  proto kernel  metric 256
default via fe80::200:ff:feaa:1 dev eth0  proto kernel  metric 1024  expires 11sec
root@n1Host:/tmp/pycore.46757/n1Host.conf#
```

- Analizar los paquetes capturados.

```
20 82.512941 fe80::200:ff:feaa:0 ff02::2 ICMPv6 70 Router Solicitation from 00:00:00:aa:00:00
▶ Frame 20: 70 bytes on wire (560 bits), 70 bytes captured (560 bits)
▼ Ethernet II, Src: 00:00:00_aa:00:00 (00:00:00:aa:00:00), Dst: IPv6mcast_00:00:00:02 (33:33:00:00:00:02)
  ▶ Destination: IPv6mcast_00:00:00:02 (33:33:00:00:00:02)
  ▶ Source: 00:00:00_aa:00:00 (00:00:00:aa:00:00)
    Type: IPv6 (0x86dd)
▼ Internet Protocol Version 6, Src: fe80::200:ff:feaa:0 (fe80::200:ff:feaa:0), Dst: ff02::2 (ff02::2)
  ▶ 0110 .... = Version: 6
  ▶ .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00000000
    .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 16
    Next header: ICMPv6 (0x3a)
    Hop limit: 255
    Source: fe80::200:ff:feaa:0 (fe80::200:ff:feaa:0)
    [Source SA MAC: 00:00:00_aa:00:00 (00:00:00:aa:00:00)]
    Destination: ff02::2 (ff02::2)
▼ Internet Control Message Protocol v6
    Type: Router Solicitation (133)
    Code: 0
    Checksum: 0x79da [correct]
    Reserved: 00000000
  ▶ ICMPv6 Option (Source link-layer address : 00:00:00:aa:00:00)

0020  00 ff fe aa 00 00 ff 02  00 00 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 02 85 00  79 da 00 00 00 00 01 01   ........ y.......
0040  00 00 00 aa 00 00                                  ......
```

Análisis de Paquete RS

```
21 82.513586 fe80::200:ff:feaa:1 ff02::1 ICMPv6 78 Router Advertisement from 00:00:00:aa:00:01
▶ Frame 21: 78 bytes on wire (624 bits), 78 bytes captured (624 bits)
▶ Ethernet II, Src: 00:00:00_aa:00:01 (00:00:00:aa:00:01), Dst: IPv6mcast_00:00:00:01 (33:33:00:00:00:01)
▼ Internet Protocol Version 6, Src: fe80::200:ff:feaa:1 (fe80::200:ff:feaa:1), Dst: ff02::1 (ff02::1)
  ▶ 0110 .... = Version: 6
  ▶ .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00000000
    .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 24
    Next header: ICMPv6 (0x3a)
    Hop limit: 255
    Source: fe80::200:ff:feaa:1 (fe80::200:ff:feaa:1)
    [Source SA MAC: 00:00:00_aa:00:01 (00:00:00:aa:00:01)]
    Destination: ff02::1 (ff02::1)
▼ Internet Control Message Protocol v6
    Type: Router Advertisement (134)
    Code: 0
    Checksum: 0x38c2 [correct]
    Cur hop limit: 64
  ▼ Flags: 0x00
      0... .... = Managed address configuration: Not set
      .0.. .... = Other configuration: Not set
      ..0. .... = Home Agent: Not set
      ...0 0... = Prf (Default Router Preference): Medium (0)
      .... .0.. = Proxy: Not set
      .... ..0. = Reserved: 0
    Router lifetime (s): 15
    Reachable time (ms): 0
    Retrans timer (ms): 0
  ▼ ICMPv6 Option (Source link-layer address : 00:00:00:aa:00:01)
      Type: Source link-layer address (1)
      Length: 1 (8 bytes)
      Link-layer address: 00:00:00_aa:00:01 (00:00:00:aa:00:01)

0020  00 ff fe aa 00 01 ff 02  00 00 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 01 86 00  38 c2 40 00 00 0f 00 00   ........ 8.@.....
0040  00 00 00 00 00 00 01 01  00 00 00 aa 00 01         ........ ......
```

Análisis de Paquete RA

Práctica #3:

Análisis de RS & RA

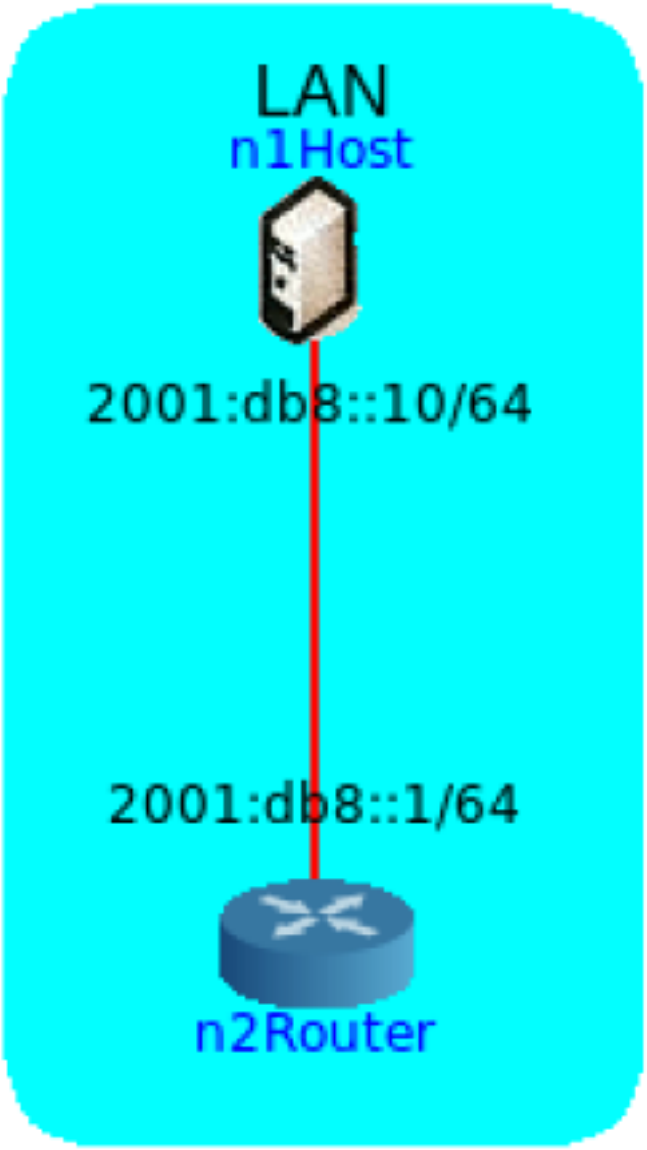
Objetivo:

Capturar y reconocer paquetes de Router Solicitation y Router Advertisement en una red simple.

En este caso el host recibirá los anuncios periódicos del router. El router enviará información de RA mediante la configuración del proceso Quagga.

Nombre del archivo:

1-03-RA.imn



Actividad #3:

- Iniciar la simulación.
  - Archivo 1-03-RA.imn
- Verificar las direcciones IP asignadas.
    - `# ip -6 addr show dev eth0`
- Chequear conectividad entre n1Host y n2Router usando *ping6*.
- Configurar n2Router para que empiece a enviar mensajes de RA.

Actividad #3:

- Abrir una terminal de n2Router y editar el archivo */usr/local/etc/quagga/Quagga.conf*:
  - `# nano /usr/local/etc/quagga/Quagga.conf`
- Incluír las siguientes tres líneas en el archivo:

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

Agregar esto debajo de la definición de eth0

Actividad #3:

- Capturar tráfico con wireshark en la interface eth0 de n1Host.
- En n2Router iniciar el proceso de router IPv6 con el siguiente comando:
    - `# ./boot.sh`

CORE: n2Router (console)

```
root@n2Router:/tmp/pycore.34135/n2Router.conf# ./boot.sh
net.ipv4.conf.all.forwarding = 1
net.ipv6.conf.all.forwarding = 1
net.ipv4.conf.all.send_redirects = 0
root@n2Router:/tmp/pycore.34135/n2Router.conf#
```

- Luego de un par de minutos, analizar los paquetes capturados.

```
1 0.000000 fe80::200:ff:feaa:1 ff02::1 ICMPv6 78 Router Advertisement from 00:00:00:aa:00:01
▶ Frame 1: 78 bytes on wire (624 bits), 78 bytes captured (624 bits)
▼ Ethernet II, Src: 00:00:00_aa:00:01 (00:00:00:aa:00:01), Dst: IPv6mcast_00:00:00:01 (33:33:00:00:00:01)
  ▶ Destination: IPv6mcast_00:00:00:01 (33:33:00:00:00:01)
  ▶ Source: 00:00:00_aa:00:01 (00:00:00:aa:00:01)
    Type: IPv6 (0x86dd)
▼ Internet Protocol Version 6, Src: fe80::200:ff:feaa:1 (fe80::200:ff:feaa:1), Dst: ff02::1 (ff02::1)
  ▶ 0110 .... = Version: 6
  ▶ .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00000000
    .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 24
    Next header: ICMPv6 (0x3a)
    Hop limit: 255
    Source: fe80::200:ff:feaa:1 (fe80::200:ff:feaa:1)
    [Source SA MAC: 00:00:00_aa:00:01 (00:00:00:aa:00:01)]
    Destination: ff02::1 (ff02::1)
▼ Internet Control Message Protocol v6
    Type: Router Advertisement (134)
    Code: 0
    Checksum: 0x31c9 [correct]
    Cur hop limit: 64
  ▶ Flags: 0x00
    Router lifetime (s): 1800
    Reachable time (ms): 0
    Retrans timer (ms): 0
  ▼ ICMPv6 Option (Source link-layer address : 00:00:00:aa:00:01)
      Type: Source link-layer address (1)
      Length: 1 (8 bytes)
      Link-layer address: 00:00:00_aa:00:01 (00:00:00:aa:00:01)

0020  00 ff fe aa 00 01 ff 02  00 00 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 01 86 00  31 c9 40 00 07 08 00 00   ........ 1.@.....
0040  00 00 00 00 00 00 01 01  00 00 00 aa 00 01         ........ ......
```

Análisis de Paquete RA

Práctica #4:

Detección de Direcciones Duplicadas

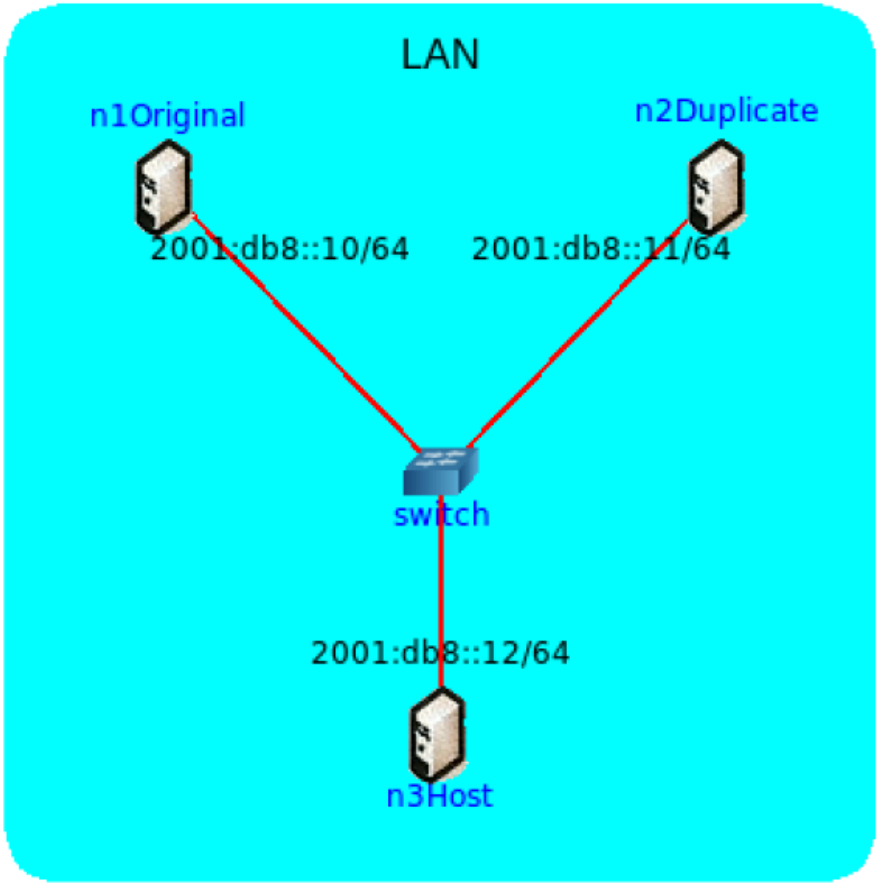
Objetivo:

Capturar y reconocer paquetes de Neighbor Discovery Protocol; Analizar el funcionamiento del mecanismo de detección de direcciones duplicadas.

Uno de los comportamientos más interesantes de DAD tiene que ver con que el nodo que originalmente tenía la dirección que se va a duplicar no se ve afectado por el problema.

Nombre del archivo:

1-04-DAD.imn



Actividad #4:

- Iniciar la simulación.
  - Archivo 1-04-DAD.imn
- Verificar las direcciones IP asignadas.
    - `# ip -6 addr show dev eth0`
- Chequear conectividad entre todos los hosts usando *ping6*.
- Comenzar una captura de paquetes en la interface eth0 de n1Original y n2Duplicate.

Actividad #4:

- Abrir una terminal en n2Duplicate:
  - Borrar la IPv6 asignada.
  - Configurar la dirección IPv6 que ya está siendo usada por n1Original:
    - # ip addr del 2001:db8::11/64 dev eth0
    - # ip addr add 2001:db8::10/64 dev eth0

Actividad #4:

- Chequear el resultado de la asignación:
  - Usando *ifconfig* **no** se ve la asignación fallida:

```
CORE: n2Duplicate (console)
root@n2Duplicate:/tmp/pycore.34135/n2Duplicate.conf# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:00:00:aa:00:01
          inet6 addr: 2001:db8::10/64 Scope:Global
          inet6 addr: fe80::200:ff:feaa:1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:43 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7972 (7.9 KB)  TX bytes:1064 (1.0 KB)

root@n2Duplicate:/tmp/pycore.34135/n2Duplicate.conf#
```

Actividad #4:

- Chequear el resultado de la asignación:
  - Usando *ip* **sí** se ve la asignación fallida:

```
CORE: n2Duplicate (console)
root@n2Duplicate:/tmp/pycore.34135/n2Duplicate.conf# ip addr show dev eth0
39: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP q
len 1000
    link/ether 00:00:00:aa:00:01 brd ff:ff:ff:ff:ff:ff
    inet6 2001:db8::10/64 scope global tentative dadfailed
       valid_lft forever preferred_lft forever
    inet6 fe80::200:ff:feaa:1/64 scope link
       valid_lft forever preferred_lft forever
root@n2Duplicate:/tmp/pycore.34135/n2Duplicate.conf#
```

- Chequear las capturas de paquetes.

n1.eth0.90 [Wireshark 1.6.7]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	2001:db8::12	ff02::1:ff00:10	ICMPv6	86	Neighbor Solicitation for 2001:db8::10 from 00:0
2	0.000060	2001:db8::10	2001:db8::12	ICMPv6	86	Neighbor Advertisement 2001:db8::10 (sol, ovr) i
3	0.000075	2001:db8::12	2001:db8::10	ICMPv6	118	Echo (ping) request id=0x0026, seq=1
4	0.000088	2001:db8::10	2001:db8::12	ICMPv6	118	Echo (ping) reply id=0x0026, seq=1
5	1.000876	2001:db8::12	2001:db8::10	ICMPv6	118	Echo (ping) request id=0x0026, seq=2
6	1.000916	2001:db8::10	2001:db8::12	ICMPv6	118	Echo (ping) reply id=0x0026, seq=2
7	1.999918	2001:db8::12	2001:db8::10	ICMPv6	118	Echo (ping) request id=0x0026, seq=3
8	1.999955	2001:db8::10	2001:db8::12	ICMPv6	118	Echo (ping) reply id=0x0026, seq=3
9	3.000030	2001:db8::12	2001:db8::10	ICMPv6	118	Echo (ping) request id=0x0026, seq=4
10	3.000068	2001:db8::10	2001:db8::12	ICMPv6	118	Echo (ping) reply id=0x0026, seq=4

```
▶ Frame 4: 118 bytes on wire (944 bits), 118 bytes captured (944 bits)
▶ Ethernet II, Src: 00:00:00_aa:00:00 (00:00:00:aa:00:00), Dst: 00:00:00_aa:00:02 (00:00:00:aa:00:02)
▼ Internet Protocol Version 6, Src: 2001:db8::10 (2001:db8::10), Dst: 2001:db8::12 (2001:db8::12)
  ▶ 0110 .... = Version: 6
  ▶ .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00000000
    .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 64
    Next header: ICMPv6 (0x3a)
    Hop limit: 64
    Source: 2001:db8::10 (2001:db8::10)
    Destination: 2001:db8::12 (2001:db8::12)
▼ Internet Control Message Protocol v6
    Type: Echo (ping) reply (129)
    Code: 0
```

```
0000  00 00 00 aa 00 02 00 00  00 aa 00 00 86 dd 60 00   ........ ......`.
0010  00 00 00 40 3a 40 20 01  0d b8 00 00 00 00 00 00   ...@:@ . ........
0020  00 00 00 00 00 10 20 01  0d b8 00 00 00 00 00 00   ...... . ........
0030  00 00 00 00 00 12 81 00  76 07 00 26 00 01 ae f7   ........ v..&....
```

File: "/tmp/wireshark_n1.eth0.90_2... Packets: 14 Displayed: 14 Marked: 0 Dropped: 0 Profile: Default

Análisis de Paquete ICMPv6 en n1Original

NDP-4-duplicate.pcap [Wireshark 1.6.7]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	2001:db8::12	ff02::1:ff00:10	ICMPv6	86	Neighbor Solicitation for 2001:db8::10 from 00:00:00:aa:0

```
▶ Frame 1: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
▼ Ethernet II, Src: 00:00:00_aa:00:02 (00:00:00:aa:00:02), Dst: IPv6mcast_ff:00:00:10 (33:33:ff:00:00:10)
  ▶ Destination: IPv6mcast_ff:00:00:10 (33:33:ff:00:00:10)
  ▶ Source: 00:00:00_aa:00:02 (00:00:00:aa:00:02)
    Type: IPv6 (0x86dd)
▼ Internet Protocol Version 6, Src: 2001:db8::12 (2001:db8::12), Dst: ff02::1:ff00:10 (ff02::1:ff00:10)
  ▶ 0110 .... = Version: 6
  ▶ .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00000000
    .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 32
    Next header: ICMPv6 (0x3a)
    Hop limit: 255
```

```
0000  33 33 ff 00 00 10 00 00  00 aa 00 02 86 dd 60 00   33...... ......`.
0010  00 00 00 20 3a ff 20 01  0d b8 00 00 00 00 00 00   ... :. . ........
0020  00 00 00 00 00 12 ff 02  00 00 00 00 00 00 00 00   ........ ........
0030  00 01 ff 00 00 10 87 00  1d 4f 00 00 00 00 20 01   ........ .O.... .
```

File: "/tmp/NDP-4-duplicate.pcap" 1... Packets: 1 Displayed: 1 Marked: 0 Load time: 0:00.000 Profile: Default

Análisis de Paquete ICMPv6 en n2Duplicate

Práctica #5:

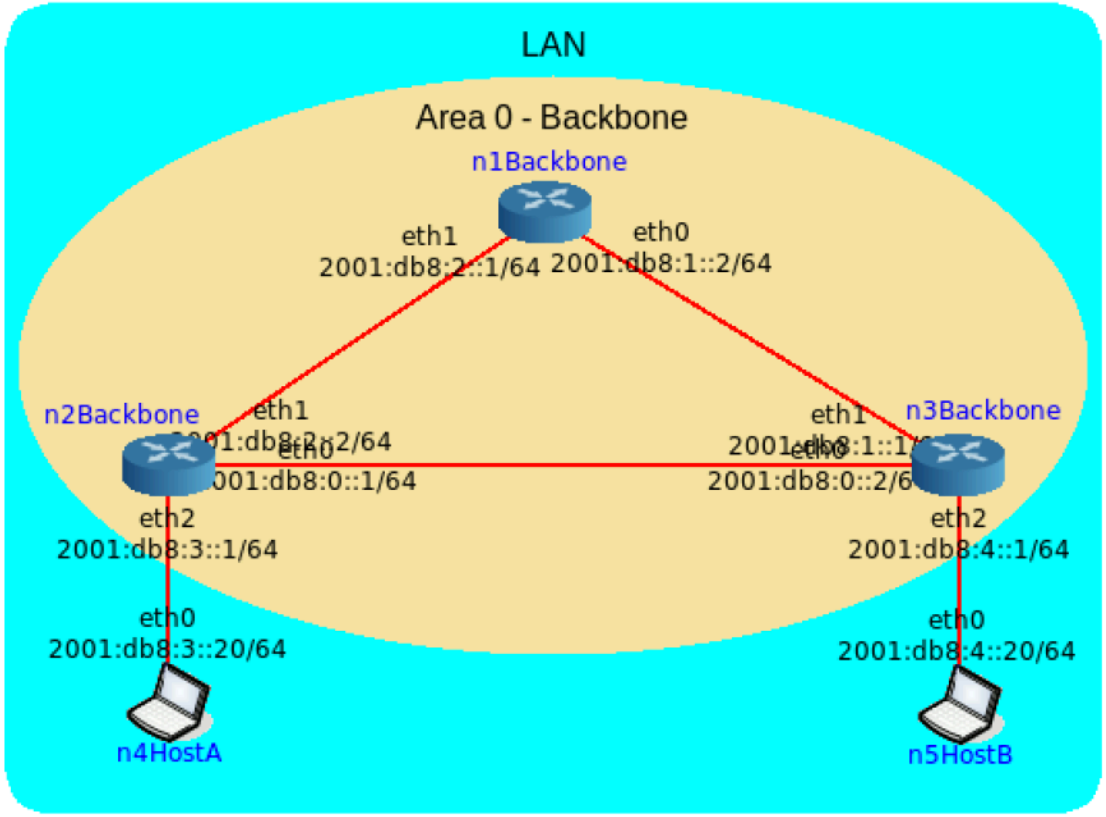
Enrutamiento dinámico con OSPFv3

Objetivo:

Hacer una breve demostración del funcionamiento del protocolo OSPFv3 en una red IPv6.

Nombre del archivo:

5-01-OSPFv3.imn



Actividad #5:

- Iniciar la simulación.
  - Archivo 5-01-OSPFv3.imn
- Verificar las direcciones IP asignadas.
    - `# ip -6 addr show dev eth0`
- Chequear conectividad entre todos los hosts y el router directamente conectado usando *ping6*.
- Intentar alcanzar algún equipo remoto usando *ping6*.

Actividad #5:

```
n4HostA
root@n4HostA:/tmp/pycore.53996/n4HostA.conf# ping6 -c 4 2001:db8:3::1
PING 2001:db8:3::1(2001:db8:3::1) 56 data bytes
64 bytes from 2001:db8:3::1: icmp_seq=1 ttl=64 time=0.159 ms
64 bytes from 2001:db8:3::1: icmp_seq=2 ttl=64 time=0.064 ms
64 bytes from 2001:db8:3::1: icmp_seq=3 ttl=64 time=0.161 ms
64 bytes from 2001:db8:3::1: icmp_seq=4 ttl=64 time=0.100 ms

--- 2001:db8:3::1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.064/0.121/0.161/0.041 ms
root@n4HostA:/tmp/pycore.53996/n4HostA.conf#
```

Figura 5.2: *teste de conectividade entre* `n4HostA` *e* `n2Backbone`.

```
n4HostA
root@n4HostA:/tmp/pycore.53997/n4HostA.conf# ping6 -c 4 2001:db8:4::20
PING 2001:db8:4::20(2001:db8:4::20) 56 data bytes
From 2001:db8:3::1 icmp_seq=1 Destination unreachable: No route
From 2001:db8:3::1 icmp_seq=2 Destination unreachable: No route
From 2001:db8:3::1 icmp_seq=3 Destination unreachable: No route
From 2001:db8:3::1 icmp_seq=4 Destination unreachable: No route

--- 2001:db8:4::20 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3016ms

root@n4HostA:/tmp/pycore.53997/n4HostA.conf#
```

Actividad #5:

- Chequear la tabla de rutas de n2Backbone

```
n2Backbone
root@n2Backbone:/tmp/pycore.53997/n2Backbone.conf# ip -6 route show
2001:db8::/64 dev eth0  proto kernel  metric 256
2001:db8:2::/64 dev eth1  proto kernel  metric 256
2001:db8:3::/64 dev eth2  proto kernel  metric 256
fe80::/64 dev eth0  proto kernel  metric 256
fe80::/64 dev eth1  proto kernel  metric 256
fe80::/64 dev eth2  proto kernel  metric 256
root@n2Backbone:/tmp/pycore.53997/n2Backbone.conf#
```

- Una opción sería crear rutas estáticas, pero va a ser mejor utilizar un protocolo de enrutamiento dinámico (OSPFv3).

Actividad #5:

- Configurar OSPFv3 en n1Backbone
  - # vtysh
  - configure terminal
  - router ospfv6
  - router-id 1.1.1.1
  - interface eth0 area 0.0.0.0
  - interface eth1 area 0.0.0.0
  - redistribute connected
  - exit
  - exit
  - exit

Actividad #5:

n1Backbone

```
root@n1Backbone:/tmp/pycore.53997/n1Backbone.conf# vtysh

Hello, this is Quagga (version 0.99.21mr2.2).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

n1Backbone# configure terminal
n1Backbone(config)# router ospf6
n1Backbone(config-ospf6)# router-id 1.1.1.1
n1Backbone(config-ospf6)# interface eth0 area 0.0.0.0
n1Backbone(config-ospf6)# interface eth1 area 0.0.0.0
n1Backbone(config-ospf6)# redistribute connected
n1Backbone(config-ospf6)# exit
n1Backbone(config)# exit
n1Backbone# exit
root@n1Backbone:/tmp/pycore.53997/n1Backbone.conf#
```

Actividad #5:

- De mismo modo, configurar OSPFv3 en n2Backbone
  - # vtysh
  - configure terminal
  - router ospfv6
  - router-id 2.2.2.2
  - interface eth0 area 0.0.0.0
  - interface eth1 area 0.0.0.0
  - redistribute connected
  - exit
  - exit
  - exit

Actividad #5:

- Finalmente, configurar OSPFv3 en n3Backbone
  - # vtysh
  - configure terminal
  - router ospfv6
  - router-id 3.3.3.3
  - interface eth0 area 0.0.0.0
  - interface eth1 area 0.0.0.0
  - redistribute connected
  - exit
  - exit
  - exit

Actividad #5:

- Validar las configuraciones y el funcionamiento de OSPFv3 en cualquiera de los equipos.
    - `# vtysh`
    - `show ipv6 ospf6`
    - `show ipv6 ospf6 neighbor`
    - `show ipv6 route`
    - `exit`

Actividad #5:

n2Backbone

```
root@n2Backbone:/tmp/pycore.53997/n2Backbone.conf# vtysh

Hello, this is Quagga (version 0.99.21mr2.2).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

n2Backbone# show ipv6 ospf6
 OSPFv3 Routing Process (0), Instance ID 0, Router-ID 2.2.2.2
 Running 00:54:27
 Number of AS scoped LSAs is 2
 Number of areas in this router is 1
 Area 0.0.0.0
     Number of Area scoped LSAs is 12
     Interface attached to this area: eth0 eth1
n2Backbone# show ipv6 ospf6 neighbor
Neighbor ID     Pri    DeadTime  State/IfState         Duration I/F[State]
3.3.3.3           1    00:00:34   Full/BDR             00:53:11 eth0[DR]
1.1.1.1           1    00:00:32   Full/DR              00:54:28 eth1[BDR]
n2Backbone# show ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       o - OSPF6, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

C>* ::1/128 is directly connected, lo
o   2001:db8::/64 [110/1] is directly connected, eth0, 00:54:54
C>* 2001:db8::/64 is directly connected, eth0
o>* 2001:db8:1::/64 [110/2] via fe80::200:ff:feaa:4, eth1, 00:53:14
o   2001:db8:2::/64 [110/1] is directly connected, eth1, 00:54:42
C>* 2001:db8:2::/64 is directly connected, eth1
C>* 2001:db8:3::/64 is directly connected, eth2
o>* 2001:db8:4::/64 [110/2] via fe80::200:ff:feaa:1, eth0, 00:52:40
C * fe80::/64 is directly connected, eth2
C * fe80::/64 is directly connected, eth1
C>* fe80::/64 is directly connected, eth0
n2Backbone# exit
root@n2Backbone:/tmp/pycore.53997/n2Backbone.conf#
```

Actividad #5:

n2Backbone

```
root@n2Backbone:/tmp/pycore.53997/n2Backbone.conf# ip -6 route show
2001:db8::/64 dev eth0  proto kernel  metric 256
2001:db8:1::/64 via fe80::200:ff:feaa:4 dev eth1  proto zebra  metric 2
2001:db8:2::/64 dev eth1  proto kernel  metric 256
2001:db8:3::/64 dev eth2  proto kernel  metric 256
2001:db8:4::/64 via fe80::200:ff:feaa:1 dev eth0  proto zebra  metric 2
fe80::/64 dev eth0  proto kernel  metric 256
fe80::/64 dev eth1  proto kernel  metric 256
fe80::/64 dev eth2  proto kernel  metric 256
root@n2Backbone:/tmp/pycore.53997/n2Backbone.conf#
```

Actividad #5:

- Finalmente, chequear la conectividad entre todos los hosts usando *ping6*.

n4HostA

```
root@n4HostA:/tmp/pycore.53997/n4HostA.conf# ping6 -c 4 2001:db8:4::20
PING 2001:db8:4::20(2001:db8:4::20) 56 data bytes
64 bytes from 2001:db8:4::20: icmp_seq=1 ttl=62 time=0.225 ms
64 bytes from 2001:db8:4::20: icmp_seq=2 ttl=62 time=0.120 ms
64 bytes from 2001:db8:4::20: icmp_seq=3 ttl=62 time=0.100 ms
64 bytes from 2001:db8:4::20: icmp_seq=4 ttl=62 time=0.106 ms

--- 2001:db8:4::20 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.100/0.137/0.225/0.052 ms
root@n4HostA:/tmp/pycore.53997/n4HostA.conf#
```

Práctica #6:

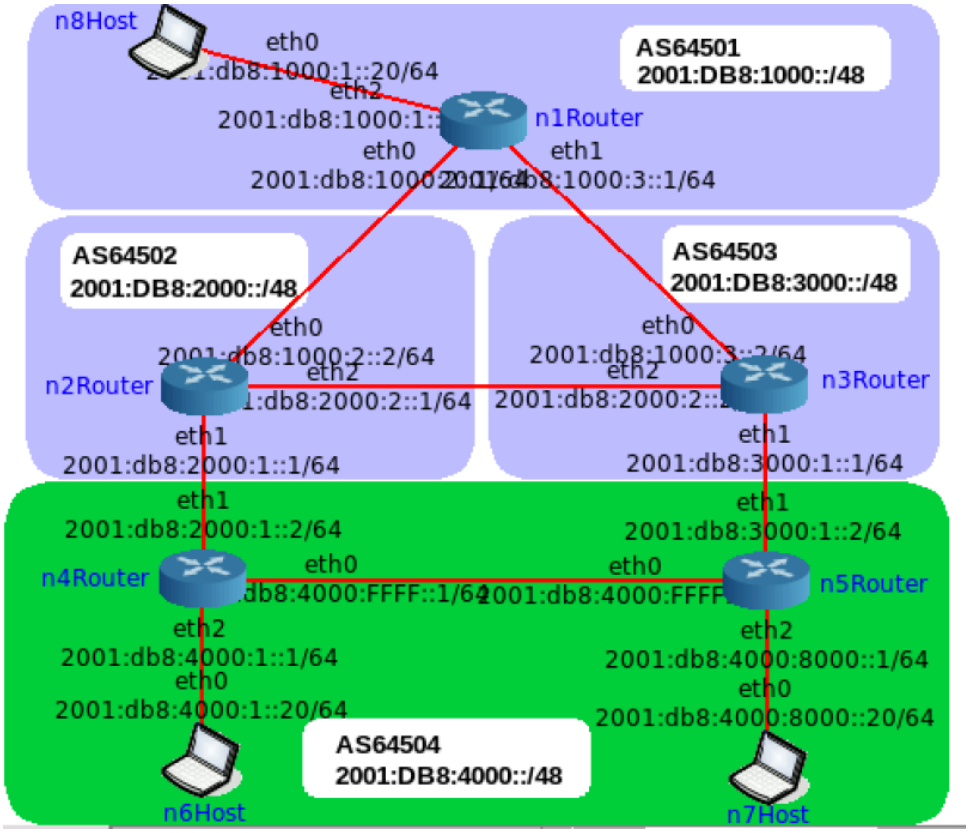
Enrutamiento dinámico con BGP

Objetivo:

Hacer una breve demostración del funcionamiento del protocolo BGP en una red IPv6.

Nombre del archivo:

5-02-BGP.imn



Agradecimientos:

- LACNIC
- IPv6.BR

Contacto:

- alejandro@lacnic.net
- ariel@weher.net