Publicação
ceptro.br

IPv6.br

Curso IPv6 básico

Laboratorio de familiarización con IPv6

cgi.br nic.br

Laboratorio de familiarización con IPv6

Objetivo: Familiarizarnos con las nuevas características del protocolo IPv6, configurarlo en nuestras computadoras portátiles y realizar los primeros ejercicios de laboratorio, tal como la configuración manual de direcciones y rutas. También analizaremos la estructura del protocolo IPv6 a través de la captura de paquetes con el programa Wireshark, donde podremos observar mejor los temas aprendidos durante la clase teórica.

Ejercicio 1: Instalación de aplicaciones

Para realizar los ejercicios propuestos en este laboratorio es necesario instalar algunas aplicaciones en nuestras computadoras portátiles:

- Wireshark
- cliente SSH (putty, por ejemplo).

Puede consultar su buscador preferido.

Ejercicio 0: Configuración nativa de IPv6 en su computadora portátil.

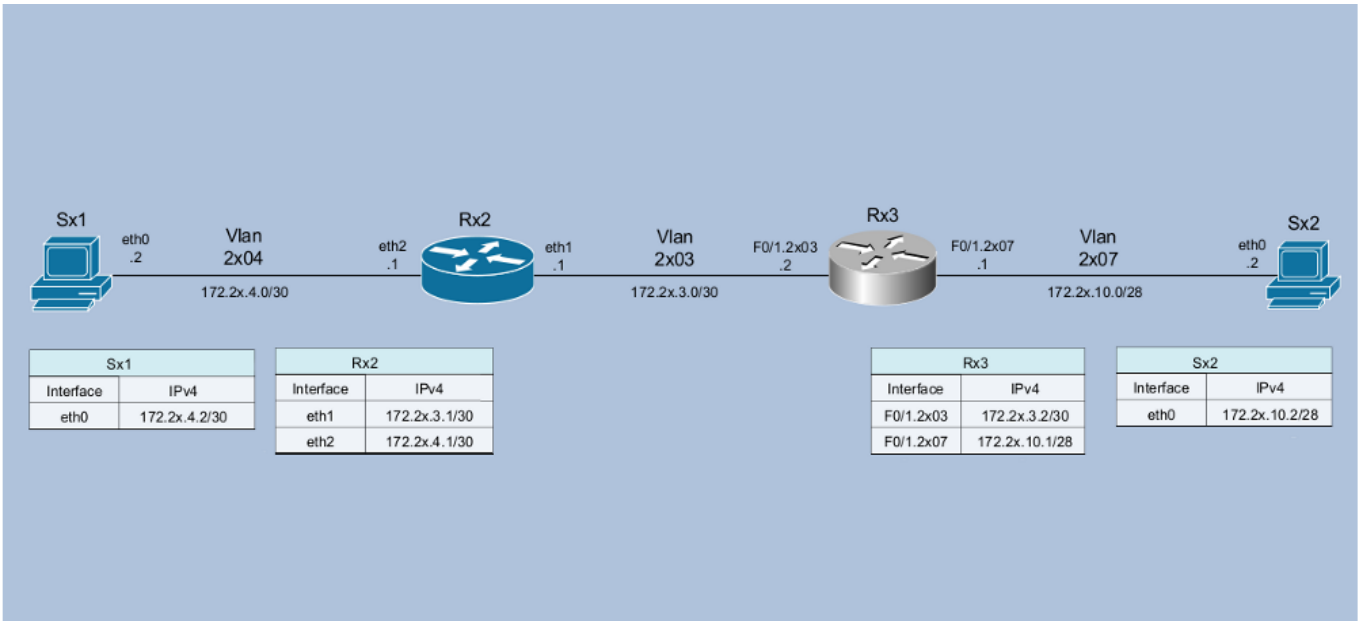
Consulte el artículo “Habilitando IPv6 em Sistemas Operacionais” en el sitio http://ipv6.br. Allí encontrará cómo configurar IPv6 en los principales sistemas operativos.

Configure...

Acceda a sitios ipv6, haga pings y traceroutes en IPv6.

Realice un traceroute a www.google.com.br y a www6.terra.com.br

Ejercicio 1: Acceso a las configuraciones de red.



Sx1	
Interface	IPv4
eth0	172.2x.4.2/30

Rx2	
Interface	IPv4
eth1	172.2x.3.1/30
eth2	172.2x.4.1/30

Rx3	
Interface	IPv4
F0/1.2x03	172.2x.3.2/30
F0/1.2x07	172.2x.10.1/28

Sx2	
Interface	IPv4
eth0	172.2x.10.2/28

A partir de ahora todos los ejercicios de laboratorio seguirán el siguiente escenario:

- La clase se dividirá en grupos y cada grupo representará un AS. Inicialmente dicho AS estará formado por dos servidores y dos routers, un Cisco y un Linux/Quagga.
- Para acceder al router CISCO del laboratorio es necesario realizar una conexión ssh a la dirección xxx.xxx.xxx.xxx o xxxx:xxxx:x:xxxx::xxx, en el puerto 3443, con el usuario “labnicX” y la contraseña “labgrupoX” (sin las comillas), donde X representa el número del grupo.

```
moreiras@atenas:~$ ssh xxxx:xxxx:x:xxxx::xxx -p3443 -llabnicX
The authenticity of host '[xxxx:xxxx:x:xxxx::xxx]:3443
([xxxx:xxxx:x:xxxx::xxx]:3443)' can't be established.
RSA key fingerprint is 7d:af:21:68:6f:9b:13:cd:9d:ce:07:b5:b0:4e:40:e5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[xxxx:xxxx:x:xxxx::xxx]:3443' (RSA) to the list of known
hosts.
labnicX@xxxx:xxxx:x:xxxx::xxx's password: labg
Last login: Mon Jun 15 02:23:42 2009 from atenas.ceptro.br
You are in a limited shell.
Type '?' or 'help' to get the list of allowed commands
labnicX:~$
```

De este modo usted iniciará una sesión en el servidor de administración de nuestro laboratorio. A partir de allí podrá acceder a los servidores y routers disponibles. El usuario labnicX permite acceder a una sesión limitada, en la cual solo es posible ejecutar los comandos para acceder a los componentes del laboratorio:

```
labnicX:~$ help
exit  help  router  server
```

```
labnicX:~$
```

Los comandos “server x1” y “server x2” permiten acceder a los servidores, siendo x el número del grupo. El comando “router x3” permite acceder al router Cisco: utilice el usuario “cisco” y la contraseña “cisco”. Por último, el comando “router x2” permite acceder al router Linux/Quagga. Para los servidores y routers Linux/Quagga utilice la contraseña “labgrupoX”.

El grupo debe probar el acceso a todos ellos.

```
labnicX:~$
labnicX:~$ server x1 (reemplace la X por el número de su grupo)
Password: labgrupoX
entered into CT 110
[root@SX1 /]# exit
logout
exited from CT 110

labnicX:~$
labnicX:~$ router x2 (reemplace la X por el número de su grupo)
entered into CT 112
[root@RX2 /]# exit
logout
exited from CT 112

labnicX:~$
labnicX:~$ router x3 (reemplace la X por el número de su grupo)
Trying 192.168.50.1...
Connected to 192.168.50.1.
Escape character is '^]'.
User Access Verification
Username: cisco
Password: cisco
router-RX3#
router-RX3#exit
Connection closed by foreign host.

labnicX:~$
labnicX:~$ server x2 (reemplace la X por el número de su grupo)
entered into CT 114
[root@SX2 /]# exit
logout
exited from CT 114
labnicX:~$
```

En esta fase de laboratorio solo está configurado IPv4, pero todos los equipos pueden ejecutar IPv6. Inicialmente no se utilizan protocolos de enrutamiento sino solamente rutas estáticas. Verifique las configuraciones de red y realice pruebas de conectividad entre todos los elementos del laboratorio. Use, por ejemplo, comandos como “ip”, “ping”, “traceroute”, “mtr”, etc. Intente comprender cómo están configurados los equipos y las rutas.

Si no hubiera conectividad entre todos los elementos, avise a los instructores e intente descubrir dónde está el problema y solucionarlo.

Ejercicio 2: Captura y análisis de paquetes.

Para la captura de paquetes en los servidores y routers Linux utilizaremos el comando “tcpdump”. Para realizar los análisis utilizaremos el programa “Wireshark”, previamente instalado en las computadoras portátiles de los participantes.

Para familiarizarnos con el uso de las herramientas vamos a monitorear el tráfico en la interfaz eth2 del router Rx2, y a ejecutar un traceroute de Sx1 a Sx2.

- En el router Rx2:

```
labnicX:~$ router x2
Password:
entered into CT 112
[root@RX2 /]#
[root@RX2 /]# tcpdump -i eth2 -s 0 -w /captura/exerc02.pcap
tcpdump: listening on eth2, link-type EN10MB (Ethernet), capture size 65535 bytes
```

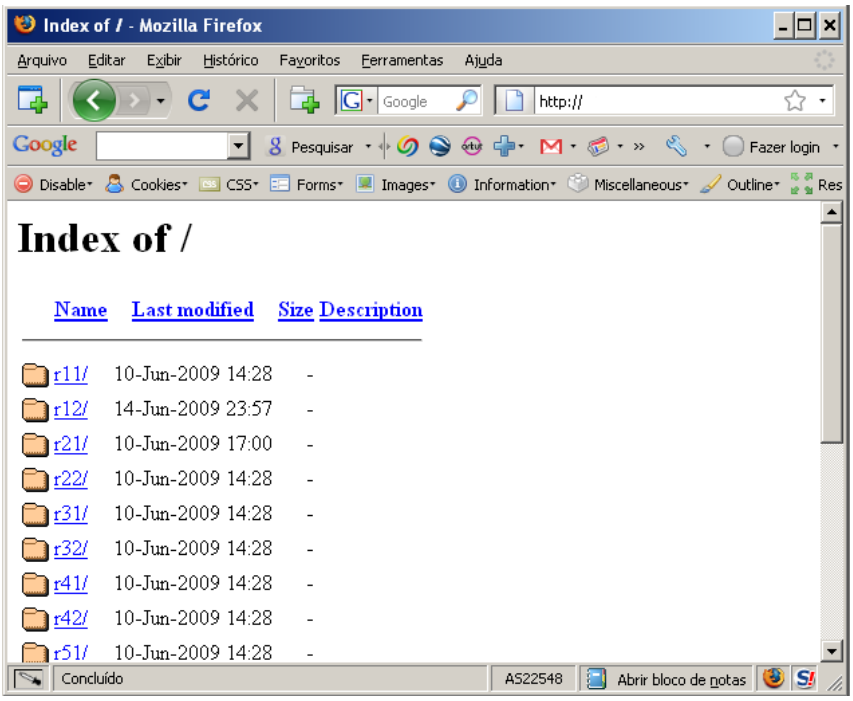
- En el servidor Sx1:

```
labnicX:~$ server x1
Password:
entered into CT 110
[root@SX1 /]# traceroute 172.2X.10.2
traceroute to 172.2X.10.2 (172.2X.10.2), 30 hops max, 46 byte packets
 1  172.2X.4.1 (172.2X.4.1)  3.027 ms  0.026 ms  0.025 ms
 2  172.2X.3.2 (172.2X.3.2)  0.642 ms  0.663 ms  0.651 ms
 3  172.2X.10.2 (172.2X.10.2)  1.851 ms  0.277 ms  0.275 ms
[root@SX1 /]#
```

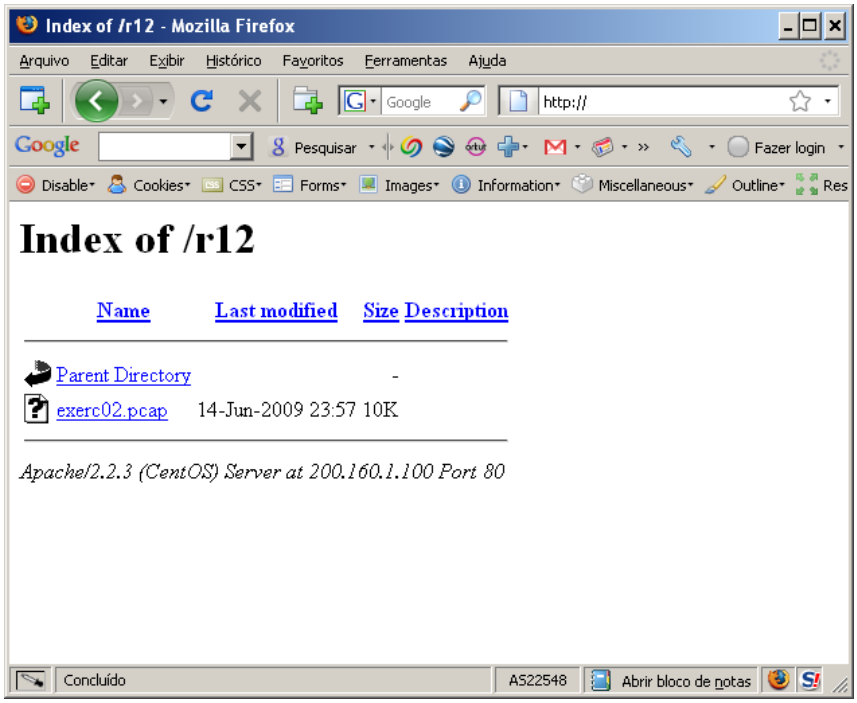
- Nuevamente, en el router Rx2:

```
[root@RX2 /]# tcpdump -i eth2 -s 0 -w /captura/exerc02.pcap
tcpdump: listening on eth2, link-type EN10MB (Ethernet), capture size 65535 bytes
[CTRL + C]
127 packets captured
127 packets received by filter
0 packets dropped by kernel
[root@RX2 /]#
```

Aproximadamente en 1 minuto, como máximo, un script copiará este archivo a un directorio compartido vía web en nuestro servidor de administración. Aguarde unos instantes y, usando un navegador en su computadora portátil, acceda a la dirección http://[xxxx:xxxx:x:xxxx::xxx] (la misma que utilizó con ssh).

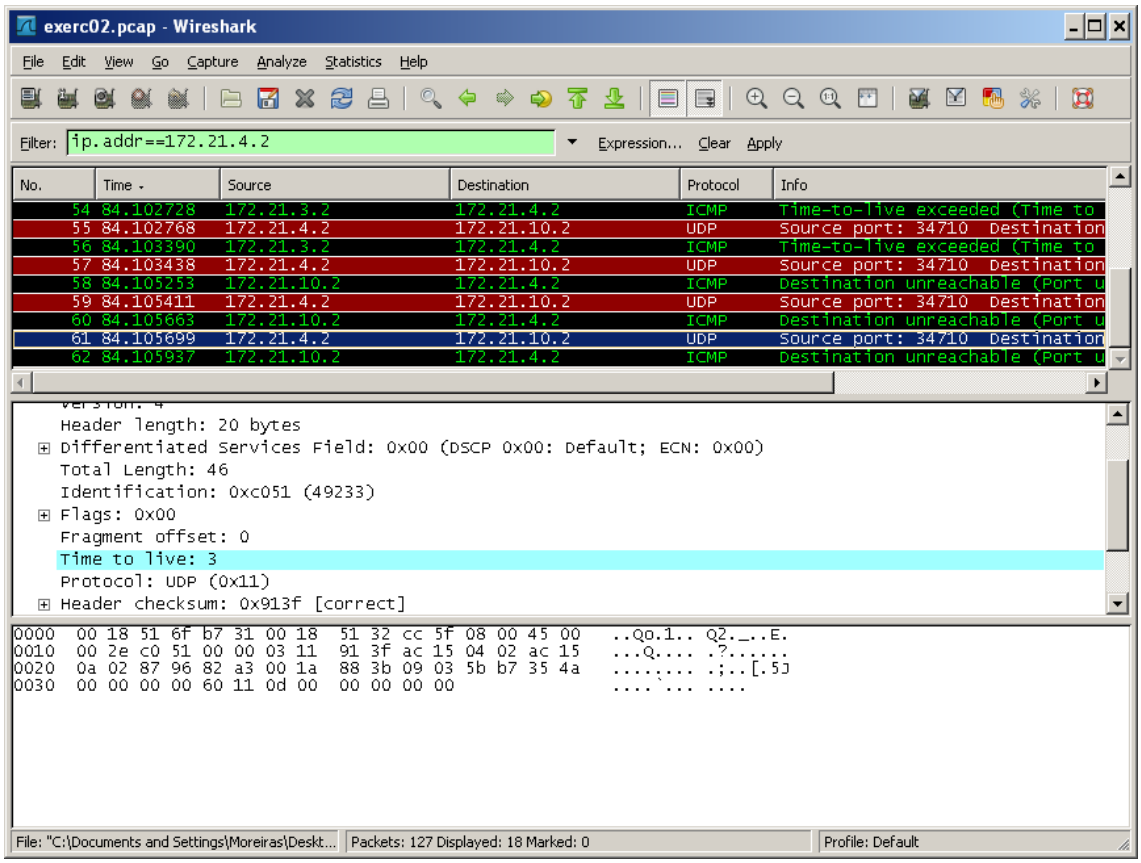


Entre a la carpeta correspondiente al router x2 y guarde el archivo exerc02.pcap en su computadora portátil.



Abra el archivo en Wireshark.
Aplique el filtro ip.addr==''dirección de origen del traceroute", si lo desea, para facilitar la visualización, y responda las 2 preguntas siguientes:
1 – ¿Qué protocolo se utiliza para enviar los mensajes por el origen?

2 – ¿Cuántos paquetes se envían para cada valor de TTL?



Ejercicio 3: IPv6 – Direcciones locales.

Habilite IPv6 en los equipos y verifique que, aunque las direcciones IPv6 aun no han sido configuradas, ya hay direcciones tipo “link-local” en cada uno de ellos. Puede ser que en algunos equipos IPv6 ya esté habilitado.

Ejemplo, en Linux:

```
[root@SX1 /]# ip addr show
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 qlen
1000
    link/ether 00:18:51:32:CC:5F brd ff:ff:ff:ff:ff:ff
    inet 172.2x.4.2/30 brd 172.2x.4.3 scope global eth0
    inet6 fe80::218:51ff:fe32:cc5f/64 scope link
       valid_lft forever preferred_lft forever
```

En este ejemplo, IPv6 ya está habilitado (en las máquinas del laboratorio IPv6 está incluido en el kernel; una alternativa sería usarlo como un módulo, cargado o no por defecto). En el caso que IPv6 haya sido compilado como módulo y no sea cargado por defecto será necesario utilizar el comando “modprobe ipv6”.

Ejemplo, en Cisco:

```
router-RX3#show ipv6 interface FastEthernet 0/1.2x03
router-RX3#
```

En este caso IPv6 no está habilitado. Hay que habilitarlo:

```
router-RX3#
router-RX3#configure terminal
router-RX3(config)#interface FastEthernet 0/1.2x03
router-RX3(config-subif)#ipv6 enable
router-RX3(config-subif)#end
router-RX3#show ipv6 interface FastEthernet 0/1.2x03
FastEthernet0/1.2x03 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::224:97FF:FEC1:C8BD
  No Virtual link-local address(es):
  Description: Conexion-RX2
  No global unicast address is configured
  Joined group address(es):
    FF02::1
    FF02::1:FFC1:C8BD
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 22434)
router-RX3#
```

Verifique que con las direcciones “link-local” ya haya conectividad IPv6 para cada segmento de red. Y que no haya conectividad entre diferentes segmentos.

Ejemplo:

```
[root@SX1 /]# ping6 fe80::218:51ff:fe32:cc5f
64 bytes from fe80::218:51ff:fe32:cc5f: icmp_seq=0 ttl=64 time=2.25 ms
64 bytes from fe80::218:51ff:fe32:cc5f: icmp_seq=1 ttl=64 time=0.079 ms
64 bytes from fe80::218:51ff:fe32:cc5f: icmp_seq=2 ttl=64 time=0.088 ms
```

¿Sus salidas se parecen más a ésta (debajo)? ¿Qué está faltando en el comando?

```
[root@SX1 /]# ping6  fe80::218:51ff:fe32:cc5f
connect: Invalid argument
```

Descubra la dirección física (MAC) de cada interfaz y responda la siguiente pregunta: ¿Cómo se forman las direcciones “link-local” a partir de las direcciones físicas?

Ejercicio 4: IPv6 – Análisis del encabezado de los paquetes.

En este ejercicio vamos a analizar el encabezado del protocolo IPv6 e intentaremos descubrir algunas diferencias con respecto a IPv4.

Vamos a capturar los paquetes de pings, v4 y v6, enviados de Sx1 a Rx2, en el router Rx2.

- En el router Rx2:

```
[root@RX2 ~]# ip addr show eth2
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:18:51:6F:B7:31 brd ff:ff:ff:ff:ff:ff
    inet 172.2x.4.1/30 brd 172.2x.4.3 scope global eth2
    inet6 fe80::218:51ff:fe6f:b731/64 scope link
       valid_lft forever preferred_lft forever

[root@RX2 ~]# tcpdump -i eth2 -s 0 -w /captura/exerc04.pcap
tcpdump: listening on eth2, link-type EN10MB (Ethernet), capture size 65535 bytes
```

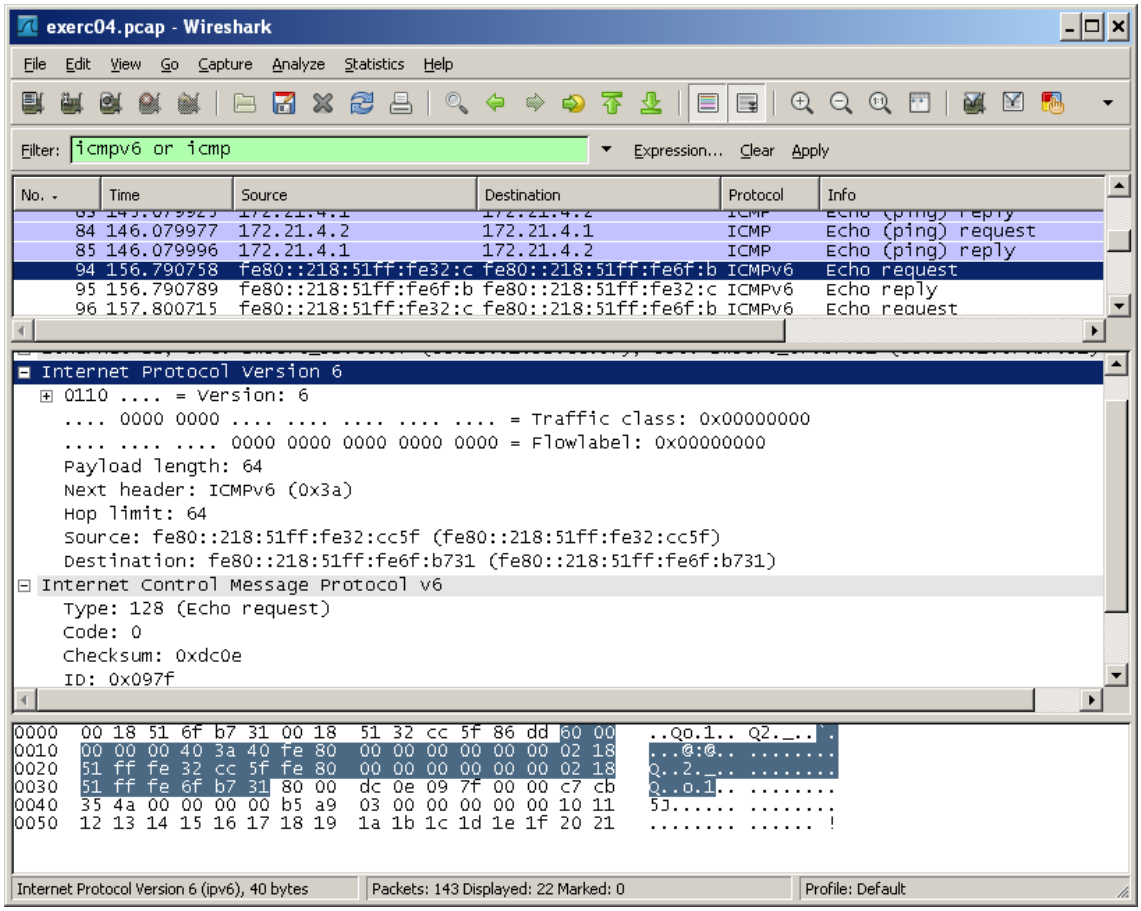
- En el servidor Sx1:

```
[root@SX1 /]# ping -c 5 172.2x.4.1
PING 172.2x.4.1 (172.2x.4.1) 56(84) bytes of data.
64 bytes from 172.2x.4.1: icmp_seq=0 ttl=64 time=2.06 ms
64 bytes from 172.2x.4.1: icmp_seq=1 ttl=64 time=0.031 ms
64 bytes from 172.2x.4.1: icmp_seq=2 ttl=64 time=0.081 ms
64 bytes from 172.2x.4.1: icmp_seq=3 ttl=64 time=0.038 ms
64 bytes from 172.2x.4.1: icmp_seq=4 ttl=64 time=0.044 ms

--- 172.2x.4.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4002ms
rtt min/avg/max/mdev = 0.031/0.451/2.061/0.805 ms, pipe 2
[root@SX1 /]# ping6 -c 5 fe80::218:51ff:fe6f:b731 -I eth0
PING fe80::218:51ff:fe6f:b731(fe80::218:51ff:fe6f:b731) from fe80::218:51ff:fe32:cc5f eth0:
56 data bytes
64 bytes from fe80::218:51ff:fe6f:b731: icmp_seq=0 ttl=64 time=0.061 ms
64 bytes from fe80::218:51ff:fe6f:b731: icmp_seq=1 ttl=64 time=0.085 ms
64 bytes from fe80::218:51ff:fe6f:b731: icmp_seq=2 ttl=64 time=0.035 ms
64 bytes from fe80::218:51ff:fe6f:b731: icmp_seq=3 ttl=64 time=0.089 ms
64 bytes from fe80::218:51ff:fe6f:b731: icmp_seq=4 ttl=64 time=0.036 ms

--- fe80::218:51ff:fe6f:b731 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4010ms
rtt min/avg/max/mdev = 0.035/0.061/0.089/0.023 ms, pipe 2
```

Abra el archivo en Wireshark, en su computadora portátil.
Para facilitar la visualización puede usar el siguiente filtro: “icmp or icmpv6”



Compare los paquetes IPv4 e IPv6... Identifique cada uno de los campos del encabezado IP en los dos casos y observe sus valores. Compare también el ICMP. Responda las siguientes preguntas:

- ¿Cuál es la diferencia de tamaño entre el encabezado IPv4 y el encabezado IPv6?
- ¿Hay también diferencias en el encabezado ICMP? ¿Cuáles?

Ejercicio 5: IPv6 – Encabezados de extensión.
En este ejercicio vamos a verificar la existencia de los encabezados de extensión, generando la necesidad de fragmentación en el comando ping.

Vamos a capturar los paquetes de pings, v4 y v6, enviados de Sx1 a Rx2, en el router Rx2, como en el ejercicio anterior. También vamos a especificar 2000 bytes como tamaño de paquete.

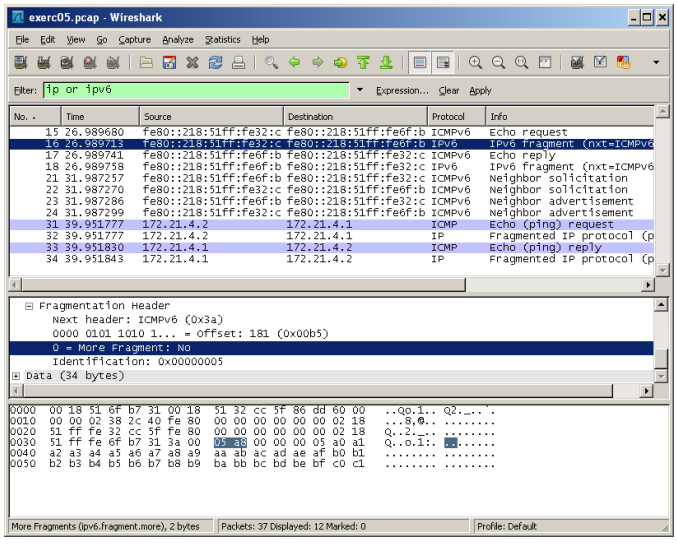
- En el router Rx2:

```
[root@RX2 ~]# tcpdump -i eth2 -s 0 -w /captura/exerc05.pcap
tcpdump: listening on eth2, link-type EN10MB (Ethernet), capture size 65535 bytes
37 packets captured
37 packets received by filter
0 packets dropped by kernel
```

- En el servidor Sx1:

```
[root@SX1 /]# ping6 -c 1 -s 2000 fe80::218:51ff:fe6f:b731 -I eth0
PING fe80::218:51ff:fe6f:b731(fe80::218:51ff:fe6f:b731) from
fe80::218:51ff:fe32:cc5f eth0: 2000 data bytes
2008 bytes from fe80::218:51ff:fe6f:b731: icmp_seq=0 ttl=64 time=0.112 ms
--- fe80::218:51ff:fe6f:b731 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.112/0.112/0.112/0.000 ms, pipe 2
[root@SX1 /]# ping -c 1 -s 2000 172.2x.4.1
PING 172.2x.4.1 (172.2x.4.1) 2000(2028) bytes of data.
2008 bytes from 172.2x.4.1: icmp_seq=0 ttl=64 time=1.38 ms
--- 172.2x.4.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.388/1.388/1.388/0.000 ms, pipe 2
[root@SX1 /]#
```

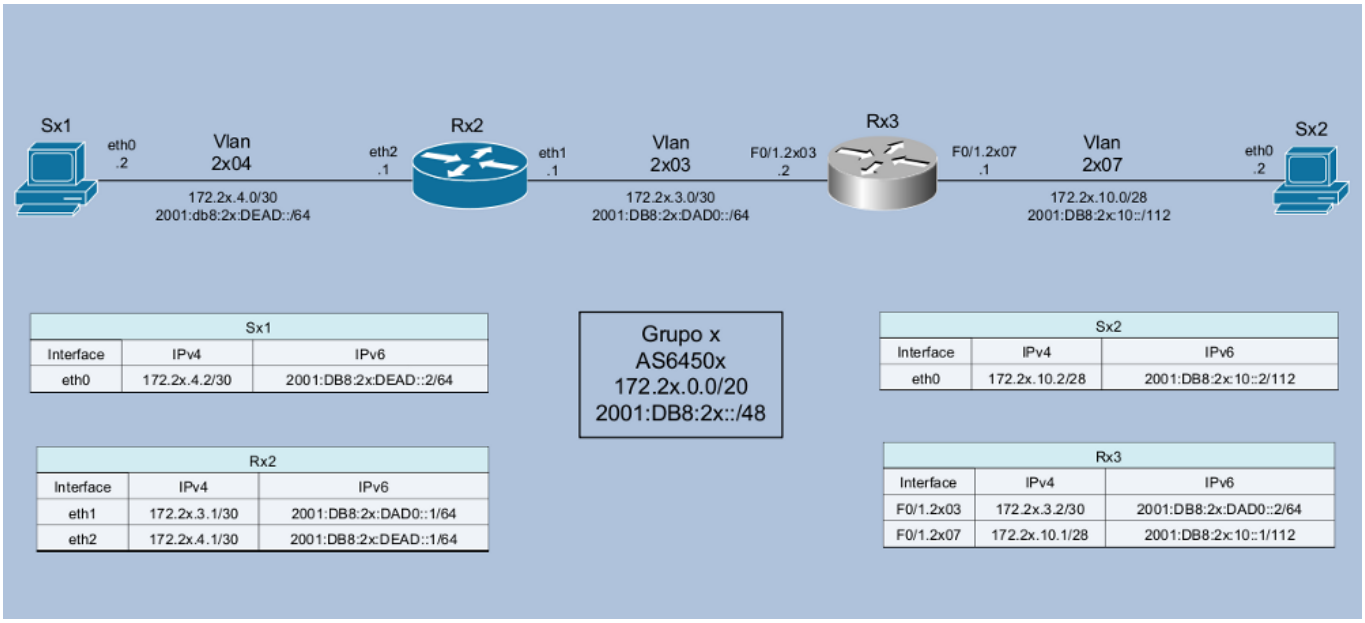
Abra el archivo en Wireshark y utilice el filtro “ipv6 or ip” para facilitar la visualización.



- Verifique la existencia del encabezado de fragmentación y responda las siguientes preguntas:
- ¿Cuál es la diferencia entre el proceso de fragmentación en IPv4 y en IPv6?
- ¿Cuál es el tamaño del encabezado de extensión (fragmentación)?
- ¿Cuál es la diferencia entre el valor del campo Next Header en el encabezado v6 del ejercicio 04 y el de este ejercicio?

Ejercicio 6a: IPv6 – Configuración manual de direcciones.

En este ejercicio vamos a configurar las direcciones de nuestro bloque (2001:db8:2x::/48) de acuerdo con la siguiente figura.



Sx1		
Interface	IPv4	IPv6
eth0	172.2x.4.2/30	2001:DB8:2x:DEAD::2/64

Sx2		
Interface	IPv4	IPv6
eth0	172.2x.10.2/28	2001:DB8:2x:10::2/112

Rx2		
Interface	IPv4	IPv6
eth1	172.2x.3.1/30	2001:DB8:2x:DAD0::1/64
eth2	172.2x.4.1/30	2001:DB8:2x:DEAD::1/64

Rx3		
Interface	IPv4	IPv6
F0/1.2x03	172.2x.3.2/30	2001:DB8:2x:DAD0::2/64
F0/1.2x07	172.2x.10.1/28	2001:DB8:2x:10::1/112

- En el servidor Sx1:

```
[root@SX1 ]# ip -6 addr add 2001:db8:2x:dead::2/64 dev eth0
[root@SX1 ]# ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen
1000
    link/ether 00:18:51:32:CC:5F brd ff:ff:ff:ff:ff:ff
    inet 172.2x.4.2/30 brd 172.2x.4.3 scope global eth0
    inet6 2001:db8:2x:dead::2/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::218:51ff:fe32:cc5f/64 scope link
       valid_lft forever preferred_lft forever
```

- En el servidor Sx2:

```
[root@SX2 /]# ip -6 addr add 2001:db8:2x:10::2/112 dev eth0
[root@SX2 /]# ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen
1000
    link/ether 00:18:51:F2:87:5B brd ff:ff:ff:ff:ff:ff
    inet 172.2x.10.2/28 brd 172.2x.10.15 scope global eth0
    inet6 2001:db8:2x:10::2/112 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::218:51ff:fef2:875b/64 scope link
       valid_lft forever preferred_lft forever
```

- En el router Rx2:

```
[root@RX2 ~]# ip -6 addr add 2001:db8:2x:dad0::1/64 dev eth1
[root@RX2 ~]# ip -6 addr add 2001:db8:2x:dead::1/64 dev eth2
[root@RX2 ~]# ip addr show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen
1000
    link/ether 00:18:51:1D:41:8A brd ff:ff:ff:ff:ff:ff
    inet 172.2x.3.1/30 brd 172.2x.3.3 scope global eth1
    inet6 2001:db8:2x:dad0::1/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::218:51ff:fe1d:418a/64 scope link
       valid_lft forever preferred_lft forever
[root@RX2 ~]# ip addr show eth2
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen
1000
    link/ether 00:18:51:6F:B7:31 brd ff:ff:ff:ff:ff:ff
    inet 172.2x.4.1/30 brd 172.2x.4.3 scope global eth2
    inet6 2001:db8:2x:dead::1/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::218:51ff:fe6f:b731/64 scope link
       valid_lft forever preferred_lft forever
```

- En el router Rx3:

```
router-RX3#
router-RX3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
router-RX3(config)#interface fastEthernet 0/1.2x03
router-RX3(config-subif)#ipv6 address 2001:db8:2x:dad0::2/64
router-RX3(config-subif)#end
router-RX3#show ipv6 interface FastEthernet 0/1.2x03
FastEthernet0/1.2x03 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::224:97FF:FEC1:C8BD
  No Virtual link-local address(es):
  Description: Conexion-RX2
  Global unicast address(es):
    2001:DB8:2x:DAD0::2, subnet is 2001:DB8:2x:DAD0::/64
  Joined group address(es):
    FF02::1
    FF02::1:FF00:2
    FF02::1:FFC1:C8BD
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 22434)
router-RX3#

router-RX3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
router-RX3(config)#interface fastEthernet 0/1.2x07
router-RX3(config-subif)#ipv6 address 2001:db8:2x:10::1/112
router-RX3(config-subif)#end
router-RX3#show ipv6 interface FastEthernet 0/1.2x07
```

```
FastEthernet0/1.2x07 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::224:97FF:FEC1:C8BD
  No Virtual link-local address(es):
  Description: Conexion-SX2
  Global unicast address(es):
    2001:DB8:2x:10::1, subnet is 2001:DB8:2x:10::/112
  Joined group address(es):
    FF02::1
    FF02::1:FF00:1
    FF02::1:FFC1:C8BD
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 26577)
router-R13#
```

Verifique que ahora haya conectividad con las nuevas direcciones, válidas globalmente, y con las direcciones "link-local" preexistentes. Sin embargo, no hay conectividad entre las diferentes redes porque las rutas aun no han sido configuradas.

Observe que es posible agregar otras direcciones IPv6 a las interfaces.

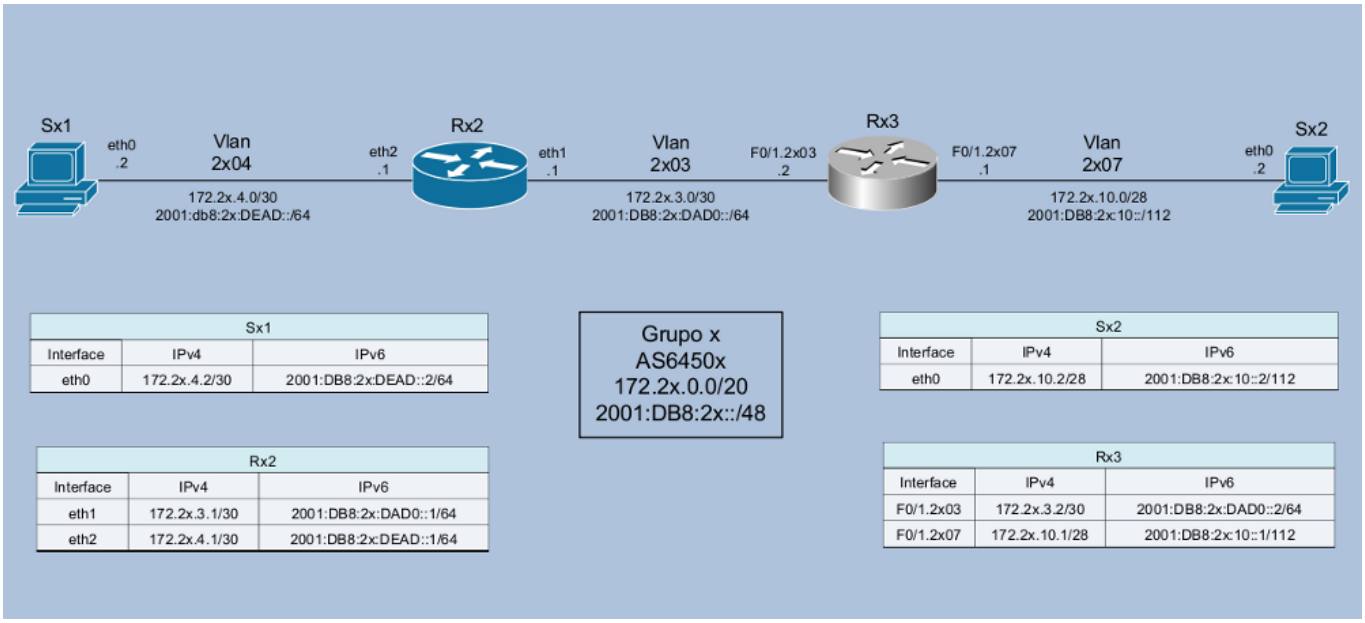
Intente, por ejemplo, agregar nuevas direcciones a Sx1:

```
[root@SX1 ~]# ip -6 addr add 2001:db8:2x:dead::60:61e/64 dev eth0
[root@SX1 ~]# ip -6 addr add 2001:db8:2x:dead::cafe:dad0/64 dev eth0
[root@SX1 ~]# ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:18:51:32:CC:5F brd ff:ff:ff:ff:ff:ff
    inet 172.2x.4.2/30 brd 172.2x.4.3 scope global eth0
    inet6 2001:db8:2x:dead::2/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::218:51ff:fe32:cc5f/64 scope link
       valid_lft forever preferred_lft forever
    inet6 2001:db8:2x:dead::cafe:dad0/64 scope global
       valid_lft forever preferred_lft forever
    inet6 2001:db8:2x:dead::60:61e/64 scope global
       valid_lft forever preferred_lft forever

[root@SX1 ~]# ip -6 addr del 2001:db8:2x:dead::60:61e/64 dev eth0
[root@SX1 ~]# ip -6 addr del 2001:db8:2x:dead::cafe:dad0/64 dev eth0
```

Ejercicio 6b: IPv6 – Configuración de las rutas.

En este ejercicio vamos a configurar las rutas manualmente para tener conectividad v4 y v6 en nuestro laboratorio.



Sx1		
Interface	IPv4	IPv6
eth0	172.2x.4.2/30	2001:DB8:2x:DEAD::2/64

Sx2		
Interface	IPv4	IPv6
eth0	172.2x.10.2/28	2001:DB8:2x:10::2/112

Rx2		
Interface	IPv4	IPv6
eth1	172.2x.3.1/30	2001:DB8:2x:DAD0::1/64
eth2	172.2x.4.1/30	2001:DB8:2x:DEAD::1/64

Rx3		
Interface	IPv4	IPv6
F0/1.2x03	172.2x.3.2/30	2001:DB8:2x:DAD0::2/64
F0/1.2x07	172.2x.10.1/28	2001:DB8:2x:10::1/112

Antes de consultar los siguientes ejemplos, observe la configuración IPv4 e intente “copiarla” al contexto IPv6.

Ejemplos:

- Para Sx1:

```
[root@SX1 ~]# ip route add default via 2001:db8:2x:dead::1
[root@SX1 ~]# ip -6 route show
2001:db8:2x:dead::/64 dev eth0  metric 256  expires 21331916sec mtu 1500 advmss
1440 hoplimit 4294967295
fe80::/64 dev eth0  metric 256  expires 21296278sec mtu 1500 advmss 1440 hoplimit
4294967295
default via 2001:db8:2x:dead::1 dev eth0  metric 1024  expires 21334251sec mtu 1500
advmss 1440 hoplimit 4294967295
unreachable default dev lo  proto none  metric -1  error -101 hoplimit 255
ff00::/8 dev eth0  metric 256  expires 21296278sec mtu 1500 advmss 1440 hoplimit
4294967295
unreachable default dev lo  proto none  metric -1  error -101 hoplimit 255
[root@SX1 ~]#
```

- Para Sx2:

```
[root@SX2 /]# ip route add default via 2001:db8:2x:10::1
[root@SX2 /]# ip -6 route show
2001:db8:2x:10::/112 dev eth0  metric 256  expires 21332280sec mtu 1500 advmss 1440 hoplimit
4294967295
fe80::/64 dev eth0  metric 256  expires 21296188sec mtu 1500 advmss 1440 hoplimit 4294967295
default via 2001:db8:2x:10::1 dev eth0  metric 1024  expires 21334371sec mtu 1500 advmss 1440
hoplimit 4294967295
unreachable default dev lo  proto none  metric -1  error -101 hoplimit 255
ff00::/8 dev eth0  metric 256  expires 21296188sec mtu 1500 advmss 1440 hoplimit 4294967295
unreachable default dev lo  proto none  metric -1  error -101 hoplimit 255
[root@SX2 /]#
```

- Para Rx2:

```
[root@RX2 ~]# ip route add default via 2001:db8:2x:dad0::2
[root@RX2 ~]# ip -6 route show
2001:db8:2x:dad0::/64 dev eth1  metric 256  expires 21334232sec mtu 1500 advmss
1440 hoplimit 4294967295
2001:db8:2x:dead::/64 dev eth2  metric 256  expires 21334251sec mtu 1500 advmss
1440 hoplimit 4294967295
2001:db8:2x:faca::/64 dev eth0  metric 256  expires 21334205sec mtu 1500 advmss
1440 hoplimit 4294967295
fe80::/64 dev eth0  metric 256  expires 21295857sec mtu 1500 advmss 1440 hoplimit
4294967295
fe80::/64 dev eth1  metric 256  expires 21295863sec mtu 1500 advmss 1440 hoplimit
4294967295
fe80::/64 dev eth2  metric 256  expires 21295868sec mtu 1500 advmss 1440 hoplimit
4294967295
unreachable fe80::/64 dev lo  metric 256  expires 21295873sec error -101 mtu 16436
advmss 16376 hoplimit 4294967295
default via 2001:db8:2x:dad0::2 dev eth1  metric 1024  expires 21334364sec mtu 1500
advmss 1440 hoplimit 4294967295
unreachable default dev lo  proto none  metric -1  error -101 hoplimit 255
ff00::/8 dev eth0  metric 256  expires 21295857sec mtu 1500 advmss 1440 hoplimit
4294967295
ff00::/8 dev eth1  metric 256  expires 21295863sec mtu 1500 advmss 1440 hoplimit
4294967295
ff00::/8 dev eth2  metric 256  expires 21295868sec mtu 1500 advmss 1440 hoplimit
4294967295
unreachable ff00::/8 dev lo  metric 256  expires 21295873sec error -101 mtu 16436
advmss 16376 hoplimit 4294967295
unreachable default dev lo  proto none  metric -1  error -101 hoplimit 255
```

- Para Rx3:

```
router-RX3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
router-RX3(config)#ipv6 unicast-routing
router-RX3(config)#ipv6 route ::0/0 2001:db8:2x:dad0::1
router-RX3(config)#end
router-RX3#show ipv6 route
IPv6 Routing Table - Default - 6 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, M - MIPv6, R - RIP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S   ::/0 [1/0]
     via 2001:DB8:2x:DAD0::1
C   2001:DB8:2x:10::/112 [0/0]
     via FastEthernet0/1.2x07, directly connected
L   2001:DB8:2x:10::1/128 [0/0]
     via FastEthernet0/1.2x07, receive
C   2001:DB8:2x:DAD0::/64 [0/0]
     via FastEthernet0/1.2x03, directly connected
L   2001:DB8:2x:DAD0::2/128 [0/0]
     via FastEthernet0/1.2x03, receive
L   FF00::/8 [0/0]
     via Null0, receive
```

Luego de realizar las configuraciones pruebe la conectividad de punta a punta, con ping6 de Sx1 a Sx2.

Ejemplo:

```
[root@SX1~]# ping6 -c 5 2001:db8:2x:10::2
PING 2001:db8:2x:10::2(2001:db8:2x:10::2) 56 data bytes
64 bytes from 2001:db8:2x:10::2: icmp_seq=0 ttl=62 time=0.441 ms
64 bytes from 2001:db8:2x:10::2: icmp_seq=1 ttl=62 time=0.458 ms
64 bytes from 2001:db8:2x:10::2: icmp_seq=2 ttl=62 time=0.454 ms
64 bytes from 2001:db8:2x:10::2: icmp_seq=3 ttl=62 time=0.408 ms
64 bytes from 2001:db8:2x:10::2: icmp_seq=4 ttl=62 time=0.456 ms

--- 2001:db8:2x:10::2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4009ms
rtt min/avg/max/mdev = 0.408/0.443/0.458/0.026 ms, pipe 2
[root@SX1 ~]# traceroute6 2001:db8:2x:10::2
traceroute to 2001:db8:2x:10::2 (2001:db8:2x:10::2) from 2001:db8:2x:dead::2, 30
hops max, 24 byte packets
 1  2001:db8:2x:dead::1 (2001:db8:2x:dead::1)  0.064 ms  0.035 ms  0.031 ms
 2  2001:db8:2x:dad0::2 (2001:db8:2x:dad0::2)  0.748 ms  0.603 ms  0.604 ms
 3  2001:db8:2x:10::2 (2001:db8:2x:10::2)  0.371 ms  0.313 ms  0.339 ms
[root@SX1 ~]#
```

Ejercicio 7: IPv6 – Neighbour Discovery.

En este ejercicio vamos a observar el funcionamiento del protocolo Neighbour Discovery.

En primer lugar vamos a limpiar la tabla de Neighbour Discovery para la interfaz eth2 del router Rx2:

Verificación de la tabla de vecinos de IPv6

- En Linux:

```
[root@RX2]# ip -6 neighbor (esta tabla es similar a la tabla ARP de IPv4)
```

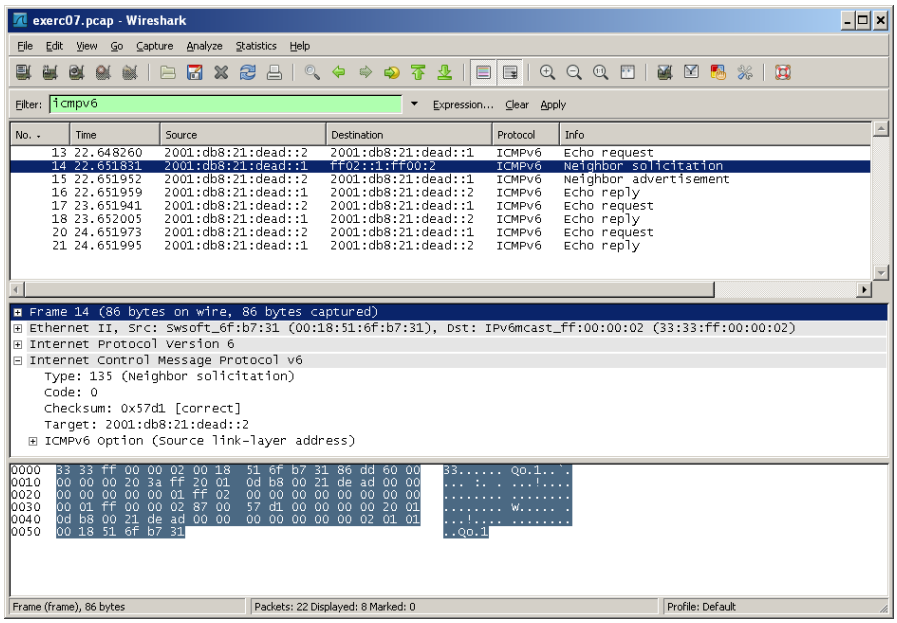
```
[root@RX2 ~]# ip neighbor flush dev eth2.
```

Ahora vamos a comenzar a capturar los paquetes y a hacer ping a la interfaz desde Sx1... Logo após o ping paramos a captura.

```
[root@RX2 ~]# tcpdump -i eth2 -s 0 -w /captura/exerc07.pcap
tcpdump: listening on eth2, link-type EN10MB (Ethernet), capture size 65535 bytes
22 packets captured
22 packets received by filter
0 packets dropped by kernel
```

```
[root@SX1 ~]# ping6 -c 3 2001:db8:2x:dead::1
PING 2001:db8:2x:dead::1(2001:db8:2x:dead::1) 56 data bytes
64 bytes from 2001:db8:2x:dead::1: icmp_seq=0 ttl=64 time=3.72 ms
64 bytes from 2001:db8:2x:dead::1: icmp_seq=1 ttl=64 time=0.114 ms
64 bytes from 2001:db8:2x:dead::1: icmp_seq=2 ttl=64 time=0.040 ms
--- 2001:db8:21:dead::1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.040/1.291/3.721/1.718 ms, pipe 2
```

Abra el archivo en Wireshark y use el “icmpv6”. Verifique la existencia de los mensajes Neighbour Solicitation y Neighbour Advertisement.



Ejercicio 8: IPv6 – Path MTU Discovery.

En este ejercicio vamos a observar el funcionamiento del protocolo Path MTU Discovery.
Primero vamos a disminuir "artificialmente" la MTU de una de las interface eth1 de Rx2 y activar el tcpdump en eth2:

```
[root@RX2 ~]# ip link set eth1 mtu 1280
[root@RX2 ~]# ip addr show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1280 qdisc pfifo_fast state UP qlen
1000
    link/ether 00:18:51:1D:41:8A brd ff:ff:ff:ff:ff:ff
    inet 172.2x.3.1/30 brd 172.2x.3.3 scope global eth1
    inet6 2001:db8:2x:dad0::1/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::218:51ff:fe1d:418a/64 scope link
       valid_lft forever preferred_lft forever
[root@RX2 ~]# tcpdump -i eth2 -s 0 -w /captura/exerc08.pcap
tcpdump: listening on eth1, link-type EN10MB (Ethernet), capture size 65535 bytes
```

A continuación vamos a ejecutar un ping de Sx1 a Sx2 con un tamaño de paquete mayor que la MTU especificada y observaremos qué ocurre.

```
[root@SX1 ~]# ping6 -c 5 -s 1500 2001:db8:2x:10::2
PING 2001:db8:2x:10::2(2001:db8:2x:10::2) 1500 data bytes
From 2001:db8:2x:dead::1 icmp_seq=0 Packet too big: mtu=1280
1508 bytes from 2001:db8:2x:10::2: icmp_seq=1 ttl=62 time=1.39 ms
1508 bytes from 2001:db8:2x:10::2: icmp_seq=2 ttl=62 time=1.29 ms
1508 bytes from 2001:db8:2x:10::2: icmp_seq=3 ttl=62 time=1.32 ms
1508 bytes from 2001:db8:2x:10::2: icmp_seq=4 ttl=62 time=1.33 ms
```

Luego detenemos la captura y analizamos los datos en Wireshark. Intente identificar el mensaje icmp "Packet too big" y obsere qué tipo de información trae. Responda la siguiente pregunta: ¿Cuál es el protocolo de ese mensaje y qué piensa que ocurriría si fuera bloqueado por un firewall?

Publicação
ceptro.br

IPv6.br

Curso IPv6 básico

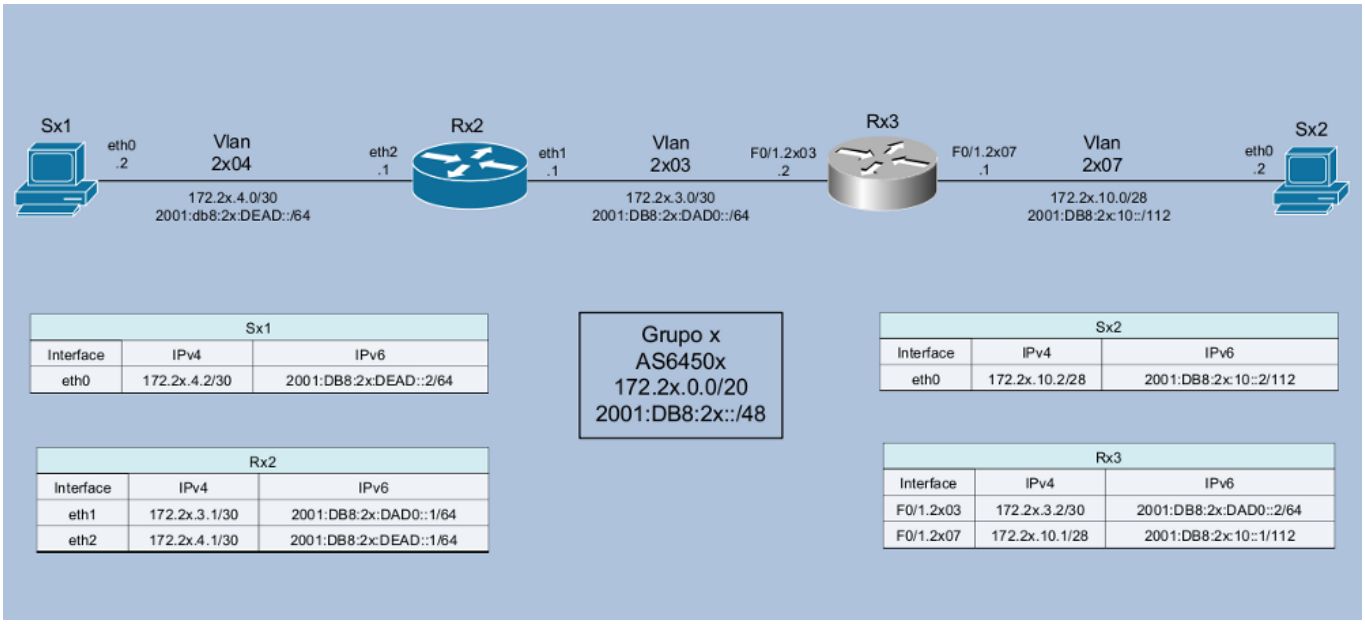
Laboratório: Firewall IPv6

cgi.br nic.br

Laboratorio – Firewall IPv6

Objetivo: Implementar un firewall simple en los servidores del AS, con soporte nativo para IPv6, utilizando ip6tables.

Escenario inicial: Cada AS tiene acceso a un router Cisco, un router Linux/Quagga y dos servidores Linux. No hay políticas de enrutamiento externo ni protocolo de enrutamiento interno (IGP) implementados, ni para IPv4 ni para IPv6. Solo se han realizado las configuraciones de direccionamiento estático IPv4 y IPv6. El grupo debe probar la comunicación dentro del propio AS AS (usar, por ejemplo, mtr, ping y traceroute IPv4 e IPv6).



Sx1		
Interface	IPv4	IPv6
eth0	172.2x.4.2/30	2001:DB8:2x:DEAD::2/64

Rx2		
Interface	IPv4	IPv6
eth1	172.2x.3.1/30	2001:DB8:2x:DAD0::1/64
eth2	172.2x.4.1/30	2001:DB8:2x:DEAD::1/64

Sx2		
Interface	IPv4	IPv6
eth0	172.2x.10.2/28	2001:DB8:2x:10::2/112

Rx3		
Interface	IPv4	IPv6
F0/1.2x03	172.2x.3.2/30	2001:DB8:2x:DAD0::2/64
F0/1.2x07	172.2x.10.1/28	2001:DB8:2x:10::1/112

Ejercicio 1 - Configuración de ip6tables.

Es necesario ser más cuidadosos al utilizar firewalls en redes IPv6 ya que, al contrario de lo que ocurre en la mayoría de las redes IPv4, la red interna no está “protegida” por el uso de direcciones IP privadas (RFC 1918). Con la adopción del protocolo IPv6 todos los host pueden utilizar direcciones válidas con conectividad directa a Internet y alcance a todos los host de la red interna que tengan IPv6 habilitado.

Utilizaremos el siguiente script con reglas para ip6tables.

```
#!/bin/sh

PATH=/sbin:/bin:/usr/sbin:/usr/bin

# camino de iptables
iptables="/sbin/ip6tables"

# Mis IPs
# Agregar las IPs v6 del servidor aquí
ips_locais="2001:DB8:XX:DEAD::2/128 FE80::XXXX:XXFF:FEXX:XXXX/128 FF02::1:FF00:0/104
FF02::1/128"

start () {
        echo "Iniciar el filtro de paquetes: ip6tables..."

        # La política por defecto es descartar todos los paquetes
        echo "Configuración de la política por defecto para descartar todos los paquetes"
        $iptables -F
        $iptables -P INPUT DROP
        $iptables -P OUTPUT DROP
        $iptables -P FORWARD DROP

        # Permitir tráfico ilimitado al localhost
        echo "Permitiendo tráfico ilimitado al localhost"
        $iptables -A INPUT  -i lo -j ACCEPT
        $iptables -A OUTPUT -o lo -j ACCEPT

        # Conexiones de entrada y salida permitidas para este servidor
        for ip in $ips_locais
          do
            echo -n "Permitir algunas conexiones de entrada para este servidor (IP $ip)..."

            # Abrir ssh para todos
            echo -n "ssh "
            $iptables -A INPUT -p tcp -s ::/0 --sport 513:65535 -d $ip --dport 22 -j ACCEPT
            $iptables -A OUTPUT -p tcp -d ::/0 --dport 513:65535 -s $ip --sport 22 -j ACCEPT

            # Tráfico HTTP
            echo -n "http "
            $iptables -A INPUT -p tcp -d $ip --dport 80 -j ACCEPT
            $iptables -A OUTPUT -p tcp -s $ip --sport 80 -j ACCEPT

            # Permitir Traceroute
            $iptables -A INPUT -p udp --dport 33434:65535 -d $ip -j ACCEPT
            $iptables -A OUTPUT -p udp --dport 33434:65535 -s $ip -j ACCEPT

            # Permitir el envío de mensajes ICMPv6
            echo -n "icmp out "
            $iptables -A OUTPUT -p icmpv6 -s $ip -j ACCEPT

            ###### RFC 4890 #####
            ###### Tráfico ICMPv6 que NO DEBE ser DESCARTADO ######
            echo -n "icmp in "
            # ECHO REQUESTS y RESPONSES (Type 128 y 129)
            # =========================
            $iptables -A INPUT -p icmpv6 --icmpv6-type echo-request -d $ip -j ACCEPT
```

```
            $iptables -A INPUT -p icmpv6 --icmpv6-type echo-reply -d $ip -j ACCEPT

            # DESTINATION UNREACHABLE (Type 1)
            # ========================
            $iptables -A INPUT -p icmpv6 --icmpv6-type destination-unreachable -d $ip -j
ACCEPT

            # PACKET TOO BIG (Type 2)
            # ===============
            $iptables -A INPUT -p icmpv6 --icmpv6-type packet-too-big -d $ip -j ACCEPT

            # TIME EXCEEDED (Type 3)
            # ==============
            $iptables -A INPUT -p icmpv6 --icmpv6-type ttl-zero-during-transit -d $ip -j
ACCEPT
            $iptables -A INPUT -p icmpv6 --icmpv6-type ttl-zero-during-reassembly -d $ip -j
ACCEPT

            # PARAMETER PROBLEM (Type 4)
            # ==================
            $iptables -A INPUT -p icmpv6 --icmpv6-type unknown-option -d $ip -j ACCEPT
            $iptables -A INPUT -p icmpv6 --icmpv6-type unknown-header-type -d $ip -j ACCEPT
            $iptables -A INPUT -p icmpv6 --icmpv6-type bad-header -d $ip -j ACCEPT

            # NEIGHBOR DISCOVERY
            # ==================
            # RS (Type 133)
            $iptables -A INPUT -p icmpv6 --icmpv6-type 133 -d $ip -j ACCEPT
            # RA (Type 134)
            $iptables -A INPUT -p icmpv6 --icmpv6-type 134 -d $ip -j ACCEPT
            # NS (Type 135)
            $iptables -A INPUT -p icmpv6 --icmpv6-type 135 -d $ip -j ACCEPT
            # NA (Type 136)
            $iptables -A INPUT -p icmpv6 --icmpv6-type 136 -d $ip -j ACCEPT
            # Inverse Neighbor Discovery Solicitation (Type 141)
            $iptables -A INPUT -p icmpv6 --icmpv6-type 141 -d $ip -j ACCEPT
            # Inverse Neighbor Discovery Advertisement (Type 142)
            $iptables -A INPUT -p icmpv6 --icmpv6-type 142 -d $ip -j ACCEPT

            # MLD
            # ===
            # Listener Query (Type 130)
            $iptables -A INPUT -p icmpv6 --icmpv6-type 130 -d $ip -j ACCEPT
            # Listener Report (Type 131)
            $iptables -A INPUT -p icmpv6 --icmpv6-type 131 -d $ip -j ACCEPT
            # Listener Done (Type 132)
            $iptables -A INPUT -p icmpv6 --icmpv6-type 132 -d $ip -j ACCEPT
            # Listener Report v2 (Type 143)
            $iptables -A INPUT -p icmpv6 --icmpv6-type 143 -d $ip -j ACCEPT

            # SEND
            # ====
            # Certificate Path Solicitation (Type 148)
            $iptables -A INPUT -p icmpv6 --icmpv6-type 148 -d $ip -j ACCEPT
            # Certificate Path Advertisement (Type 149)
            $iptables -A INPUT -p icmpv6 --icmpv6-type 149 -d $ip -j ACCEPT

            # Multicast Router Discovery
            # ==========================
            # Multicast Router Advertisement (Type 151)
            $iptables -A INPUT -p icmpv6 --icmpv6-type 151 -d $ip -j ACCEPT
            # Multicast Router Solicitation (Type 152)
            $iptables -A INPUT -p icmpv6 --icmpv6-type 152 -d $ip -j ACCEPT
            # Multicast Router Termination (Type 153)
            $iptables -A INPUT -p icmpv6 --icmpv6-type 153 -d $ip -j ACCEPT

            ##### Tráfico ICMPv6 que NORMALMENTE NO DEBE ser DESCARTADO #####
            # Movilidad IPv6 ### Solo habilitar si el nodo es un Nodo Móvil ###
            # ===============
```

```
            # Home Agent Address Discovery Request (Type 144)
            # $iptables -A INPUT -p icmpv6 --icmpv6-type 144 -d $ip -j ACCEPT
            # Home Agent Address Discovery Reply (Type 145)
            # $iptables -A INPUT -p icmpv6 --icmpv6-type 145 -d $ip -j ACCEPT
            # Mobile Prefix Solicitation (Type 146)
            # $iptables -A INPUT -p icmpv6 --icmpv6-type 146 -d $ip -j ACCEPT
            # Mobile Prefix Advertisement (Type 147)
            # $iptables -A INPUT -p icmpv6 --icmpv6-type 147 -d $ip -j ACCEPT

            ###### Casos específicos ######
            ## Algunos mensajes pueden no requerir tratamiento:
            #  - Router Renumbering (Type 138): Deben ser autenticados con IPSec
            #
            #
            ## Algunos mensajes pueden requerir políticas específicas:
            #  - Redirect (Type 137): Pueden presentar riesgos para la seguridad. Su
            # utilización debe ser analizada caso a caso.
            #
            #
            ## Los mensajes aun no definidos por la IANA o de uso experimental
            # siempre deben ser descartados.
            ## A no ser que exista un caso muy específico en la red y que
            # sean utilizados.
           echo .

        done

        # Descartar todo lo demás
        echo "Descartar todos los demás paquetes... "
        $iptables -A INPUT -s ::/0 -j DROP
        $iptables -A OUTPUT -d ::/0 -j DROP
}

stop () {
        echo "Detener el filtro de paquetes: ip6tables..."
        $iptables -P INPUT ACCEPT
        $iptables -F INPUT
        $iptables -P OUTPUT ACCEPT
        $iptables -F OUTPUT
        $iptables -P FORWARD ACCEPT
        $iptables -F FORWARD
        $iptables -F LOGDROP
        $iptables -X LOGDROP
        echo "Todas las reglas y cadenas están limpias."
        echo "Tenga cuidado... ¡¡Esto es peligroso!!"
        echo "Ejecute: ** /etc/init.d/ip6tables start ** tan pronto como sea posible."
}

status () {
       $iptables --list -v
}
case "$1" in
    start)
        start
    ;;
    stop)
        stop
    ;;
    try|test)
        start
        sleep 10
        stop
    ;;

    restart|reload|force-reload)
        stop
        sleep 2
        start
    ;;
```

```
    status)
        status
    ;;
    *)
        echo "Uso: /etc/init.d/ip6tables {start|stop|restart|status|try}" >&2
        exit 1
    ;;
esac

exit
```

Descarge este escrip de la dirección

http://[xxxx:xxxx:x:xxxx::xxx]/ipt6tables.txt

Básicamente, la política por defecto de las reglas de firewall aquí aplicadas consiste en descartar todos los tipos de paquetes, solo permitiendo el acceso al nodo vía ssh en el puerto 22, acceso vía http en el puerto 80, la trazabilidad del nodo vía traceroute y la utilización de mensajes ICMPv6 de acuerdo con las recomendaciones de la RFC 4890.

Ahora aplicaremos estas reglas a uno de los servidores del AS. En el servidor Sx1, cree el archivo ip6tables dentro del directorio /etc/init.d/ y agregue el contenido del script que bajó anteriormente.

```
[root@SX1 /]# cd /etc/init.d/
[root@SX1 /]# cat -> ip6tables
[Ctrl+V]
[Ctrl+D]
```

Modifique las direcciones de la línea 10 del archivo de acuerdo con la numeración del servidor, guarde el archivo y reinicie el servicio ip6tables:

```
[root@SX1 /]# /etc/init.d/ip6tables restart
```

Ejercicio 2 - Prueba de las reglas del firewall

Su firewall IPv6 ya debe estar funcionando normalmente. Ahora realizaremos algunas pruebas para analizar su configuración y si hay diferencias en la definición de las reglas para iptables (IPv4) e ip6tables (IPv6).

Desde el servidor Sx2, acceda al servidor Sx1 vía ssh:

```
[root@SX2 /]# ssh root@2001:DB8:2X:DEAD::2
The authenticity of host '2001:db8:2X:dead::2 (2001:db8:2X:dead::2)' can't be
established.
RSA key fingerprint is f9:66:86:4b:d6:81:1b:c7:79:27:1f:54:76:00:ba:d9.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '2001:db8:2X:dead::2' (RSA) to the list of known hosts.
root@2001:db8:2X:dead::2's password:
```

También acceda al servicio http del sevidor Sx1 utilizando el *browser* elinks:

```
[root@SX2 /]# elinks 2001:DB8:2X:DEAD::2
```

¿Qué está faltando en el comando anterior para que funcione correctamente?

Luego modifique el script ip6tables en el servidor Sx1 para bloquear estos dos servicios, comentando las líneas correspondientes:

```
...
# Abrir ssh para todos
#echo -n "ssh "
#$iptables -A INPUT -p tcp -s ::/0 --sport 513:65535 -d $ip --dport 22 -j ACCEPT
#$iptables -A OUTPUT -p tcp -d ::/0 --dport 513:65535 -s $ip --sport 22 -j ACCEPT


# Tráfico HTTP
#echo -n "http "
#$iptables -A INPUT -p tcp -d $ip --dport 80 -j ACCEPT
#$iptables -A OUTPUT -p tcp -s $ip --sport 80 -j ACCEPT
...
```

Reinicie el servicio ip6tables e intente acceder a estos servicios nuevamente desde el servidor Sx2.

```
[root@SX1 /]# /etc/init.d/ip6tables restart
```

También probaremos el uso del comando traceroute6 trazando la ruta desde el servidor Sx2 hasta el servidor Sx1:

```
[root@SX2 /]# traceroute6 2001:db8:2X:DEAD::2
traceroute to 2001:db8:2X:DEAD::2 (2001:db8:2X:dead::2) from 2001:db8:2X:10::2,
      30 hops max, 24 byte packets
 1  2001:db8:2X:10::1 (2001:db8:2X:10::1)  0.745 ms  0.597 ms  0.624 ms
 2  2001:db8:2X:dad0::1 (2001:db8:2X:dad0::1)  0.43 ms  0.411 ms  0.323 ms
 3  2001:db8:2X:dead::2 (2001:db8:2X:dead::2)  1.468 ms  0.4 ms  0.337 ms
```

Modifique nuevamenate el script ip6tables en Sx1, comentando las líneas que permiten el uso del comando traceroute:

```
...
# Permitir Traceroute
#$iptables -A INPUT -p udp --dport 33434:65535 -d $ip -j ACCEPT
#$iptables -A OUTPUT -p udp --dport 33434:65535 -s $ip -j ACCEPT
...
```

Reinicie el servicio ip6tables e intente trazar nuevamente la ruta al servidor Sx1 desde el servidor Sx2.

También podemos probar la conectividad IPv6 entre los dos servidores del AS realizando pings de un servidor a outro.

```
[root@SX1 /]# ping6 2001:db8:2X:10::1
PING 2001:db8:2X:10::1(2001:db8:2X:10::1) 56 data bytes
64 bytes from 2001:db8:2X:10::1: icmp_seq=0 ttl=63 time=0.658 ms
64 bytes from 2001:db8:2X:10::1: icmp_seq=1 ttl=63 time=0.647 ms
64 bytes from 2001:db8:2X:10::1: icmp_seq=2 ttl=63 time=0.659 ms
...


[root@SX2 /]# ping6 2001:db8:2X:dead::2
PING 2001:db8:2X:dead::2(2001:db8:2X:dead::2) 56 data bytes
64 bytes from 2001:db8:2X:dead::2: icmp_seq=0 ttl=62 time=1.61 ms
64 bytes from 2001:db8:2X:dead::2: icmp_seq=1 ttl=62 time=0.427 ms
64 bytes from 2001:db8:2X:dead::2: icmp_seq=2 ttl=62 time=0.431 ms
...
```

Ahora vamos a modificar el script ip6tables del servidor Sx1, comentando las líneas que permiten recibir mensajes ICMPv6 echo-request y echo-reply:

```
# ECHO REQUESTS y RESPONSES (Type 128 y 129)
# ========================
#$iptables -A INPUT -p icmpv6 --icmpv6-type echo-request -d $ip -j ACCEPT
#$iptables -A INPUT -p icmpv6 --icmpv6-type echo-reply -d $ip -j ACCEPT
```

Reinicie el servicio ip6tables y realice nuevamente las pruebas de conectividad entre los servidores Sx1 y Sx2.

La RFC 4890 recomienda no bloquear el uso de pings y sostiene que esta práctica de seguridad es innecesaria dado que realizar un barrido de direcciones en una red IPv6 es prácticamente imposible ¿Usted que piensa acerca de esta recomendación? ¿Es importante bloquear los pings o no?

Ejercicio 3 – Bloque ode mensajes ICMPv6

Ahora vamos a probar las reglas que permiten enviar y recibir los mensajes ICMPv6 que utiliza el protocolo de Descubrimiento de Vecinos.

Primero, habilite el servicio radvd en el router Rx2 editando o creando el archivo /etc/radvd.conf con el siguiente contenido:

- En el router Rx2:

```
interface eth2 {
      AdvSendAdvert on;
      MinRtrAdvInterval 3;
      MaxRtrAdvInterval 30;
      AdvLinkMTU 1500;
      prefix 2001:DB8:2X:DEAD::/64 {
              AdvOnLink on;
              AdvAutonomous on;
              AdvRouterAddr on;
              AdvPreferredLifetime 90;
              AdvValidLifetime 120;
      };
 };
```

Inicie Radvd

- En el router Rx2:

```
[root@RX2 /]# /etc/init.d/radvd start
```

Si al iniciar el proceso Radvd ocurre algún error, revise el archivo de logs el router Rx2:

```
[root@RX2 /]#tail /var/log/messages
```

Verifique si el servidor Sx1 recibió una dirección IPv6 Unicast Global a través del mecanismo de autoconfiguración stateless.

Ahora comentaremos las líneas que permiten recibir mensajes RA, RS, NA y NS:

```
# NEIGHBOR DISCOVERY
# ==================
# RS (Type 133)
#$iptables -A INPUT -p icmpv6 --icmpv6-type 133 -d $ip -j ACCEPT
# RA (Type 134)
#$iptables -A INPUT -p icmpv6 --icmpv6-type 134 -d $ip -j ACCEPT
# NS (Type 135)
#$iptables -A INPUT -p icmpv6 --icmpv6-type 135 -d $ip -j ACCEPT
# NA (Type 136)
#$iptables -A INPUT -p icmpv6 --icmpv6-type 136 -d $ip -j ACCEPT
```

Reinicie el servicio ip6tables y verifique si el servidor Sx1 continúa recibiendo una dirección mediante autoconfiguración stateless. Observe que la dirección asignada anteriormente puede demorar algunos minutos en dejar de ser utilizada por la interfaz.

Ahora descomente solo las líneas necesarias para que este servicio vuelva a funcionar normalmente. ¿Qué mensajes son necesarios para que funcione la autoconfiguración stateless?

También se puede mejorar este script permitiendo que las direcciones FF02::1:FF00:0/104 y FF02::1/128 solamente reciban los mensajes que realmente están destinados a las mismas. Consulte cuáles son estos mensajes en los apuntes teóricos y realice los cambios necesarios en el script. Luego verifique que la autoconfiguración stateless continúe funcionando normalmente.

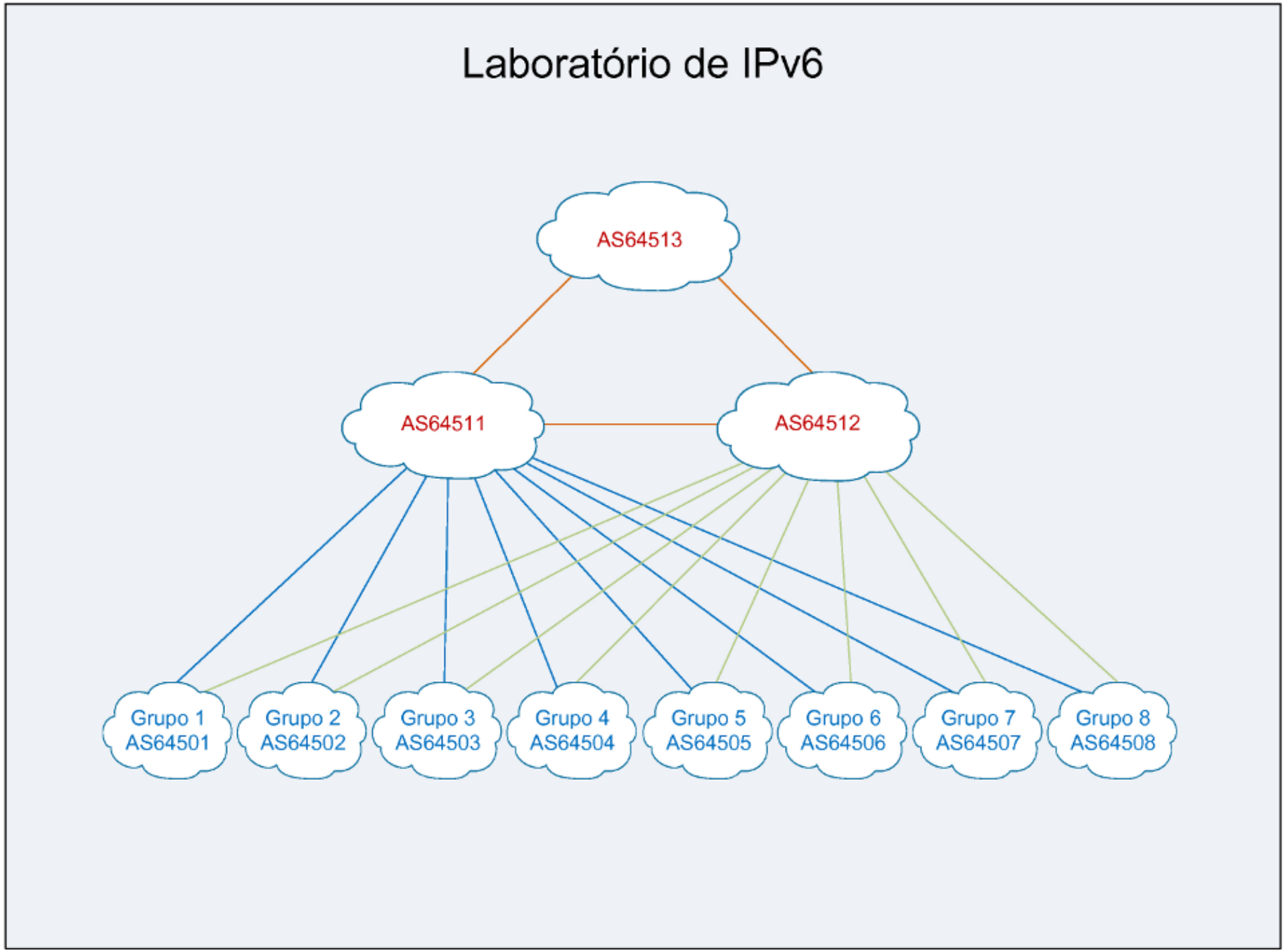
Publicação
ceptro.br

IPv6.br

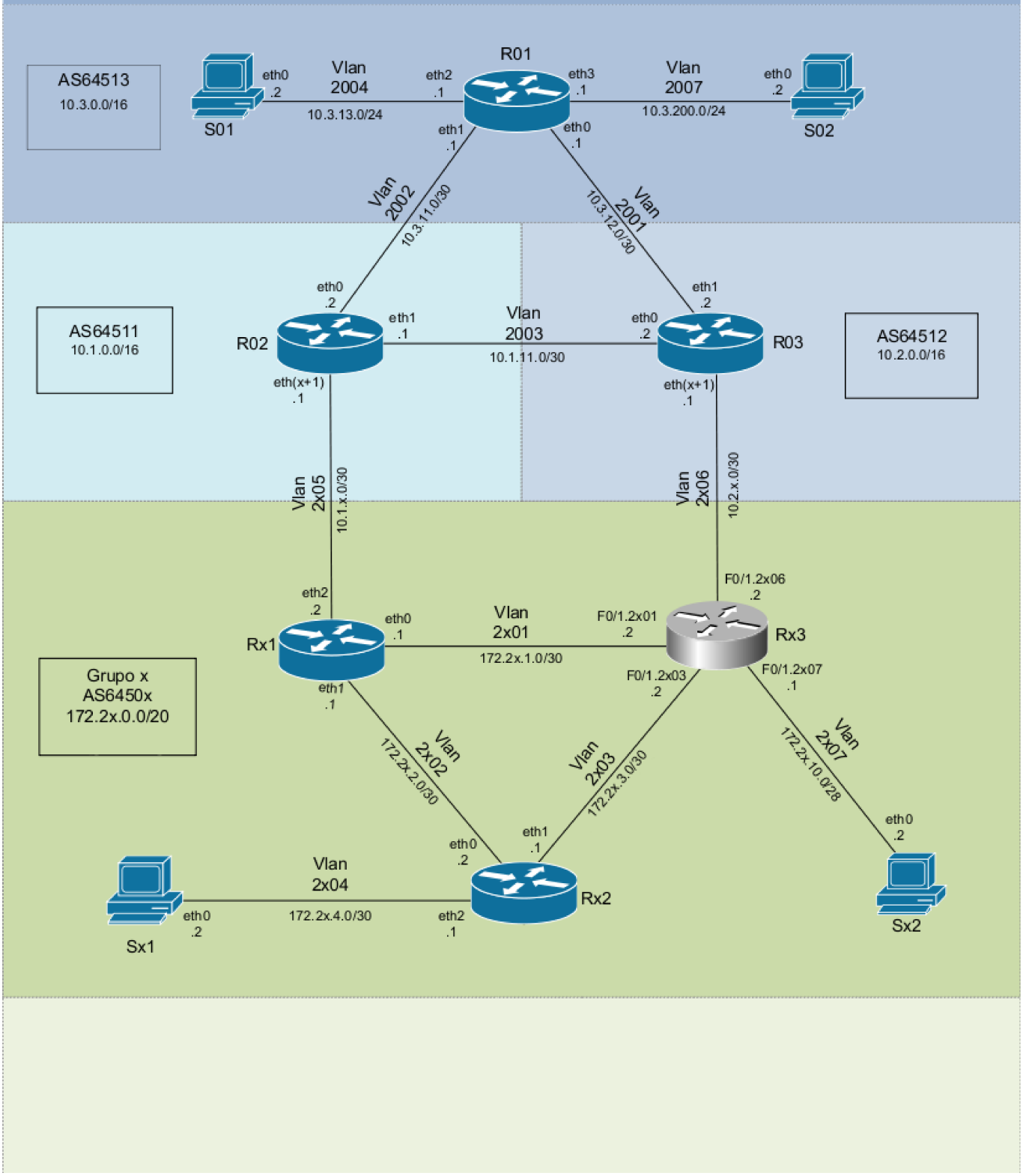
Curso IPv6 básico

Laboratório: Túneis 6to4

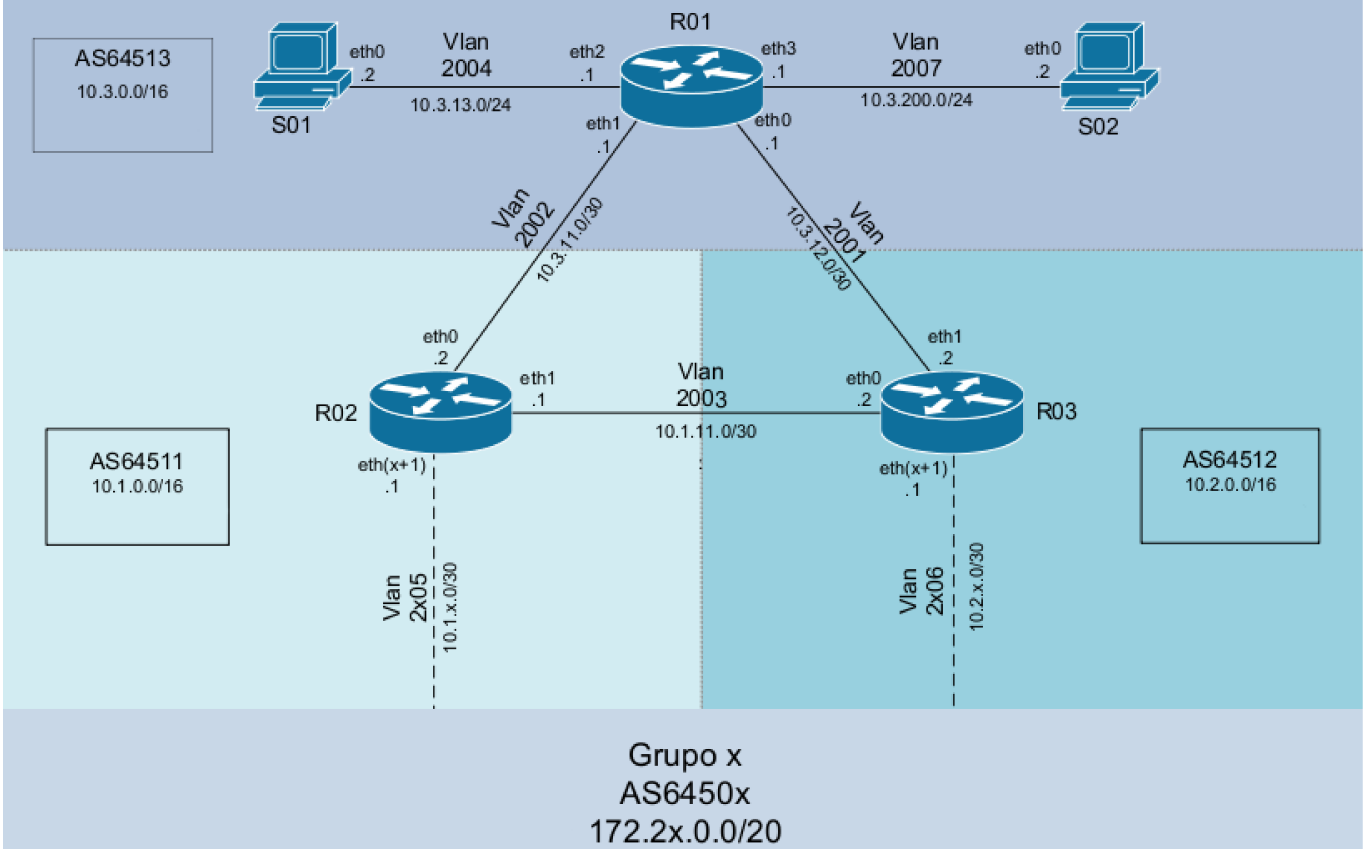
cgi.br nic.br


Laboratório de IPv6
AS64513
AS64511
AS64512
Grupo 1
AS64501
Grupo 2
AS64502
Grupo 3
AS64503
Grupo 4
AS64504
Grupo 5
AS64505
Grupo 6
AS64506
Grupo 7
AS64507
Grupo 8
AS64508

Laboratório de IPv6
Conexões entre núcleo e grupos



Laboratório de IPv6
Núcleo



Grupo x
AS6450x
172.2x.0.0/20

S01		
Interface	IPv4	IPv6
eth0	10.3.13.2/24	

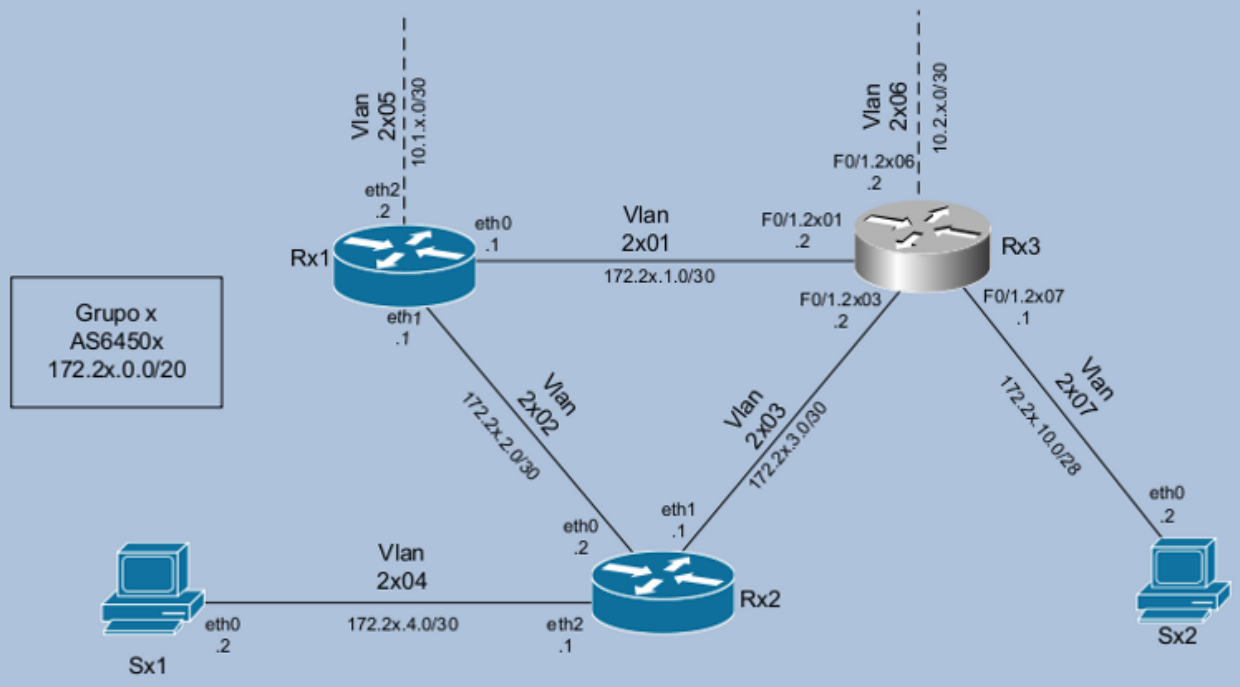
S02		
Interface	IPv4	IPv6
eth0	10.3.200.2/24	

R01		
Interface	IPv4	IPv6
eth0	10.3.12.1/30	
eth1	10.3.11.1/30	
eth2	10.3.13.1/30	
eth3	10.3.200.1/24	
lo	10.3.255.255/32	

R02		
Interface	IPv4	IPv6
eth0	10.3.11.2/30	
eth1	10.1.11.1/30	
ethx	10.1.x.1/30	
lo	10.1.255.255/32	

R03		
Interface	IPv4	IPv6
eth0	10.1.11.2/30	
eth1	10.3.12.2/30	
ethx	10.2.x.1/30	
lo	10.2.255.255/32	

Laboratório de IPv6
Grupos



Sx1	
Interface	IPv4
eth0	172.2x.4.2/30

Sx2	
Interface	IPv4
eth0	172.2x.10.2/28

Rx1	
Interface	IPv4
eth0	172.2x.1.1/30
eth1	172.2x.2.1/30
eth2	10.1.x.2/30
lo	172.2x.15.255/32

Rx2	
Interface	IPv4
eth0	172.2x.2.2/30
eth1	172.2x.3.1/30
eth2	172.2x.4.1/30
lo	172.2x.15.254/32
lo	172.2x.15.250/32

Rx3	
Interface	IPv4
F0/1.2x01	172.2x.1.2/30
F0/1.2x03	172.2x.3.2/30
F0/1.2x06	10.2.x.2/30
F0/1.2x07	172.2x.10.1/28
loopback10	172.2x.15.253/32
loopback20	172.2x.15.252/32
loopback30	172.2x.15.251/32

Laboratorio – Túneles 6to4

Objetivo: Proporcionar conectividad IPv6 al AS a través de un túnel 6to4. Para ello, configuraremos el router Cisco como el router 6to4 de nuestro AS y a partir del bloque de direcciones IPv6 asignado al mismo numeraremos los servidores y routers del AS con direcciones 6to4.
Usando el comando tcpdump y el programa Wireshark analizaremos la estructura de un paquete IPv6 encapsulado en un paquete IPv4.

Escenario inicial: En esta fase cada grupo representa un AS diferente con conexión para 2 proveedores de tránsito.

Cada AS tiene acceso a un router Cisco, dos routers Linux/Quagga y dos servidores Linux. La política de enrutamiento externo y el protocolo de enrutamiento interno (IGP), en este caso OSPF, ya están implementados para IPv4. El grupo debe probar la comunicación dentro del propio AS y con los demás AS (usar, por ejemplo, mtr, ping y traceroute IPv4).

Ejercicio 1: Configuración de túneles 6to4.

Primero calcularemos la dirección 6to4 local a partir de la dirección IPv4. El siguiente comando le ayudará a convertir la dirección de la interfaz FastEthernet 0/1.2X01 a formato hexadecimal:

```
printf "2002:%02x%02x:%02x%02x::1" 172 2X 1 2
2002:acZZ:0102::1
```

Nota 1.: las 4 primeras letras 'x' no se deben reemplazar por el número de grupo, ya que forman parte del comando printf. Solo se debe reemplazar la última letra, en el segmento '172 2X 1 2'.
Nota 2.: En la dirección 6to4 generada, el segmento 'acZZ:0102' corresponde a la dirección IPv4 de la interfaz FastEthernet 0/1.2X01 convertida a formato hexadecimal.

Ahora active el túnel usando los siguientes comandos:

```
Username: cisco
Password: cisco
router-RX3# configure terminal
router-RX3(config)# ipv6 unicast-routing
router-RX3(config)# interface Tunnel2002
router-RX3(config-if)# description tunel 6to4
router-RX3(config-if)# no ip address
router-RX3(config-if)# no ip redirects
router-RX3(config-if)# ipv6 address 2002:acZZ:0102::1/128
router-RX3(config-if)# tunnel source FastEthernet 0/1.2X01
router-RX3(config-if)# tunnel mode ipv6ip 6to4
router-RX3(config-if)# exit
router-RX3(config)# ipv6 route 2002::/16 Tunnel2002
router-RX3(config)# ipv6 route ::/0 2002:c058:6301:: (ruta por defecto para 192.88.99.1 – relay 6to4 público)
```

Ejercicio 2: Probar la conectividad a través del Túnel 6to4.

Ahora que ya hemos realizado las configuraciones del túnel vamos a probar la conectividad con otros grupos vecinos.

Confirme con los otros grupos cuál es la dirección 6to4 de sus routers Cisco y pruebe la conectividad desde el router RX3 usando el comando traceroute.

```
router-RX3#traceroute 2002:acZZ:0102::1
```

¿Cuántos saltos dio el paquete para llegar al AS vecino? Realice un traceroute a la dirección IPv4 del router Cisco del AS vecino y compare el número de saltos.

Ejercicio 3: Análisis de los paquetes 6to4:

Acceda al router Rx1, inicie un tcpdump en la interfaz eth0.

```
[root@RX1 /]# tcpdump -i eth0 -s 0 -w /captura/exerc6to4.pcap
```

En el router Rx3 (Cisco) haga pings a las direcciones 6to4 de los AS vecinos, capture los paquetes y verifique cómo se encapsula el tráfico.

Observe la estructura del paquete capturado. Analice los campos “Protocol”, “Source” y “Destination” del encabezado IPv4.

Ejercicio 4: Numeración de la red con direcciones 6to4.

Un túnel 6to4 proporciona un bloque /48 IPv6 para cada dirección IPv4 válida. Utilizaremos el bloque / 48 obtenido por el router Cisco para configurar en todos los servidores y routers del AS una dirección IPv6 (6to4).

Primero vamos a asignar una dirección 6to4 al servidor Sx2 a través del mecanismo de autoconfiguración. Para ello debemos habilitar en la interfaz FastEthernet0/1.2X07 del router Rx3 el envío de mensajes Router Advertisement (RA) para que el prefijo 6to4 sea anunciado al servidor Sx2:

```
router-RX3# configure terminal
router-RX3(config)# interface Fastethernet 0/1.2X07
router-RX3(config-if)# ipv6 nd prefix 2002:ACZZ:0102::/48
```

¿Qué pasó con el anuncio del prefijo? ¿Cómo se resuelve el problema?

Defina con su grupo cuál es la mejor manera de segmentar la red del AS. Este bloque se puede subdividir en bloques más específicos (/49, /56, /64, etc.), pero para que el servidor Sx2 obtenga una dirección a través del mecanismo de autoconfiguración el bloque anunciado por el router Rx3 debe ser un /64.

Ejemplo:

```
router-RX3# configure terminal
router-RX3(config)# interface Fastethernet 0/1.2X07
router-RX3(config-if)# ipv6 address 2002:ACZZ:0102:1000::1/56
router-RX3(config-if)# ipv6 nd prefix 2002:ACZZ:0102:1000::/64
router-RX3(config-if)# ipv6 nd ra interval 10
router-RX3(config-if)# end
```

Verifique si el servidor Sx2 recebió la dirección 6to4 correctamente.

De la misma manera que lo hizo en la interfaz FastEthernet0/1.2X07, habilite una dirección 6to4 y el anuncio de mensajes RA en la interfaz 0/1.2X03. Verifique si el router Rx2 recibió la dirección 6to4 mediante autoconfiguración stateless. Si no fue así, ¿dónde se encuentra el problema?

De acuerdo con la estructura de direccionamiento definida por el grupo, ahora vamos a configurar el AS para que todos los servidores tengan una dirección IPv6 (6to4). Configure una dirección estática en las interfaces eth1 y eth2 del router Rx2 y en la interfaz eth0 del servidor Sx1.

Ejemplo:

- En el router Rx2:

```
[root@RX2 ~]# ip -6 addr add 2002:ACZZ:0102:YYYY::YYYY/YY dev eth1
[root@RX2 ~]# ip -6 addr add 2002:ACZZ:0102:YYYY::YYYY/YY dev eth2
```

- En el servidor Sx1:

```
[root@SX1 /]# ip -6 addr add 2002:ACZZ:0102:YYYY::YYYY/YY dev eth0
```

Ahora vamos a configurar las rutas manualmente para tener conectividad a través de las direcciones 6to4.

- En el servidor Sx1:

`[root@SX1 ~]# ip route add default via 2002:ACZZ:0102:YYYY::YYYY` (dirección 6to4 del router Rx2)

- En el servidor Sx2:

`[root@SX2 ~]# ip route add default via 2002:ACZZ:0102:YYYY::YYYY` (dirección 6to4 del router Rx3)

- En el router Rx2:

`[root@RX2 ~]# ip route add default via 2002:ACZZ:0102:YYYY::YYYY` (dirección 6to4 del router Rx3)

- En el router Rx3:

```
router-RX3#configure terminal
router-RX3(config)# ipv6 unicast-routing
router-RX3(config)# ipv6 route 2002:ACZZ:102:YYYY::/YY 2002:ACZZ:0102:YYYY::YYYY
```
(dirección 6to4 del router Rx2)

Luego de realizar las configuraciones pruebe la conectividad de punta a punta, con ping6 de Sx1 a Sx2.

También pruebe la conectividad con otros grupos.

Ahora configure nuevamente el servidor Sx1 para que éste obtenga una dirección 6to4 mediante autoconfiguración stateless. Para ello use el Radvd instalado en Rx2 para realizar el anuncio de los mensajes RA.

Configure el Radvd editando o creando el archivo /etc/radvd.conf con el siguiente contenido:

- En el router Rx2:

```
interface eth2 {
      AdvSendAdvert on;
      MinRtrAdvInterval 3;
      MaxRtrAdvInterval 30;
      AdvLinkMTU 1400;
      prefix 2002:ACZZ:0102:YYYY::/64 {
              AdvOnLink off;
              AdvAutonomous on;
              AdvRouterAddr on;
              AdvPreferredLifetime 90;
              AdvValidLifetime 120;
      };
 };
```

Inicie Radvd

- En el router Rx2:

```
[root@RX2 /]# /etc/init.d/radvd start
```

Si al iniciar el proceso Radvd ocurre algún error, revise el archivo de logs el router Rx2:

```
[root@RX2 /]#tail /var/log/messages
```

Verifique si el servidor Sx1 recibió una dirección 6to4 correctamente y pruebe la conectividad internamente y con los otros grupos (entre los servidores Sx1 y Sx2).

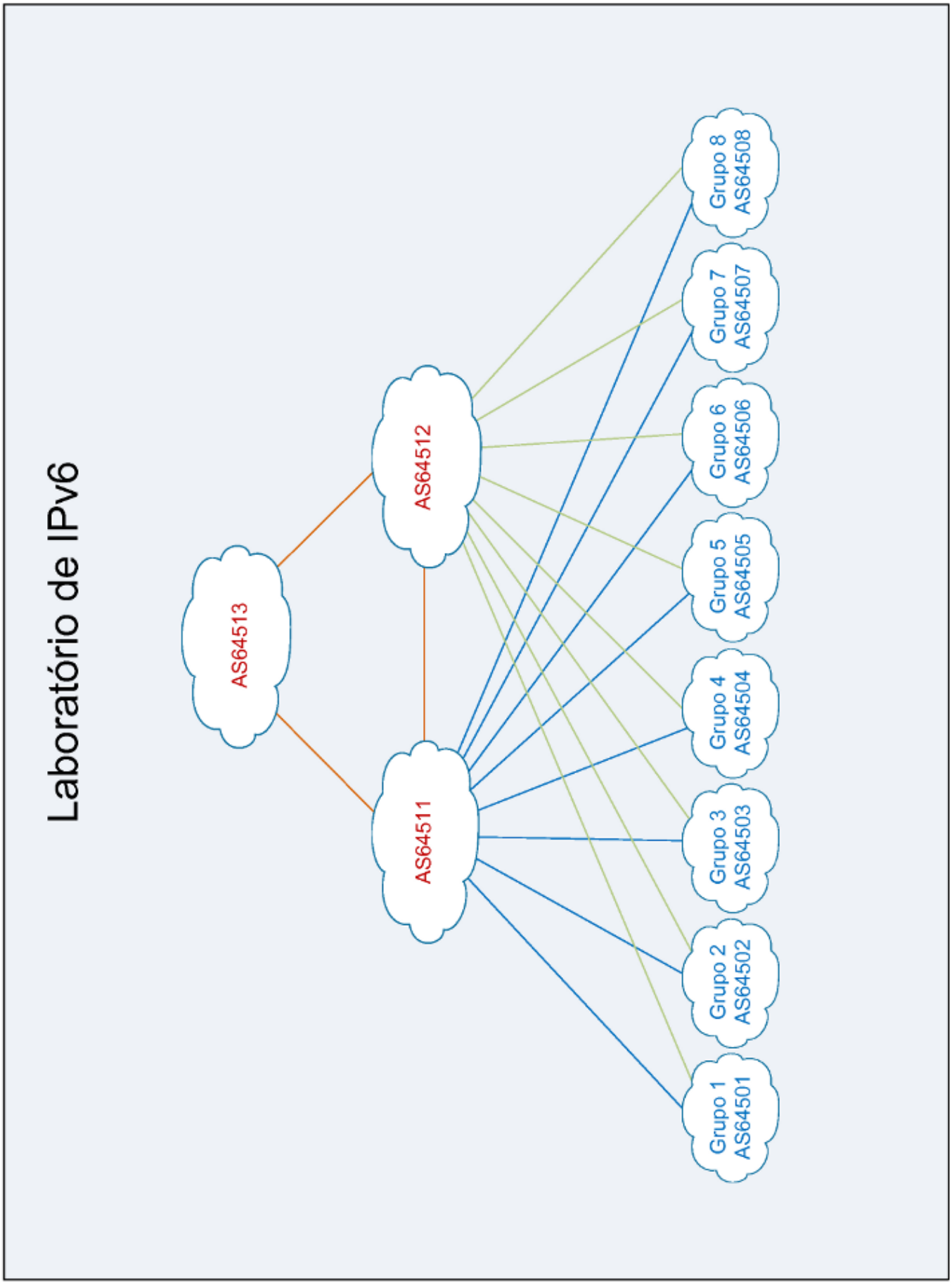
Publicação
ceptro.br

IPv6.br

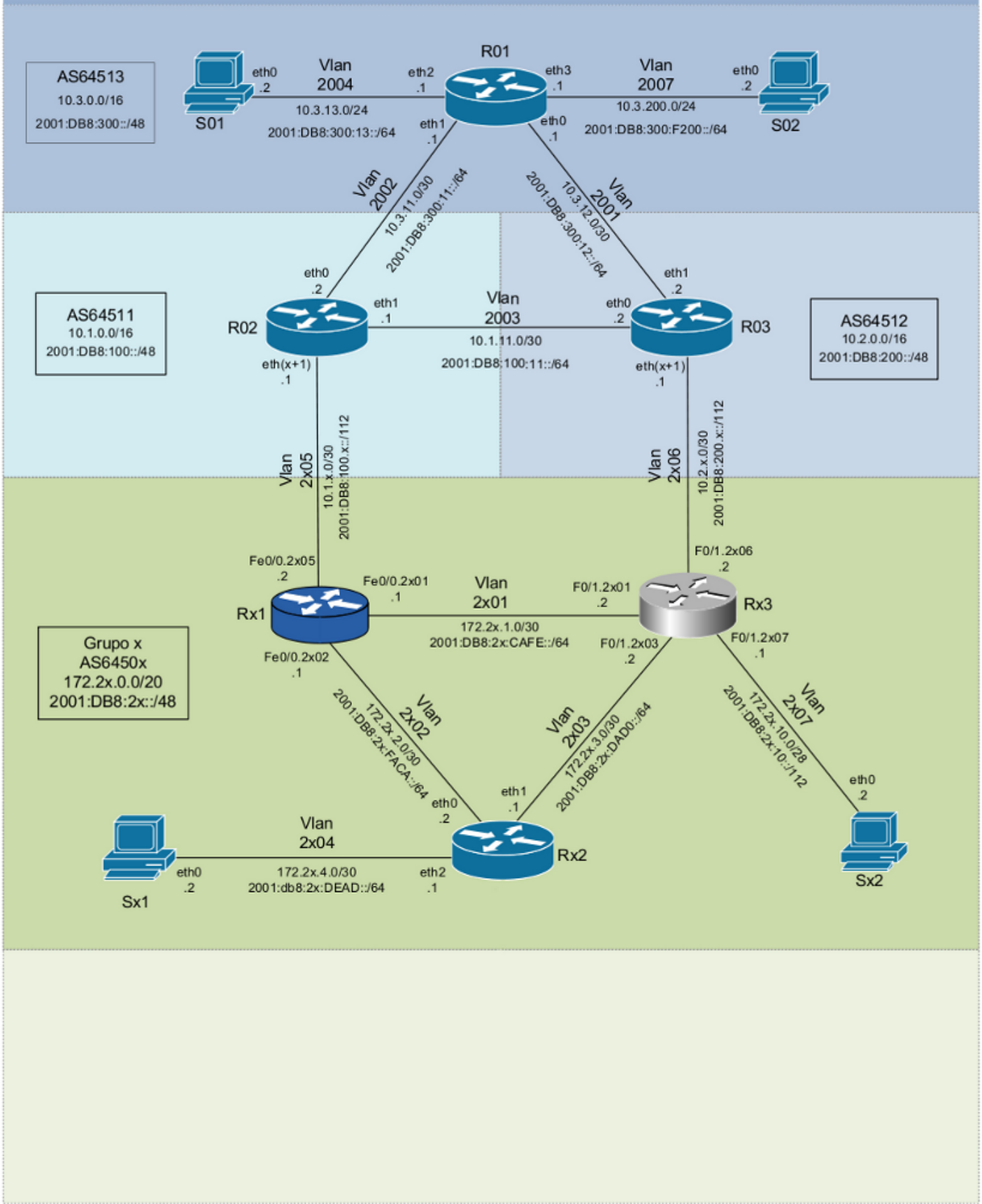
Curso IPv6 básico

Laboratório: Roteamento IPv6

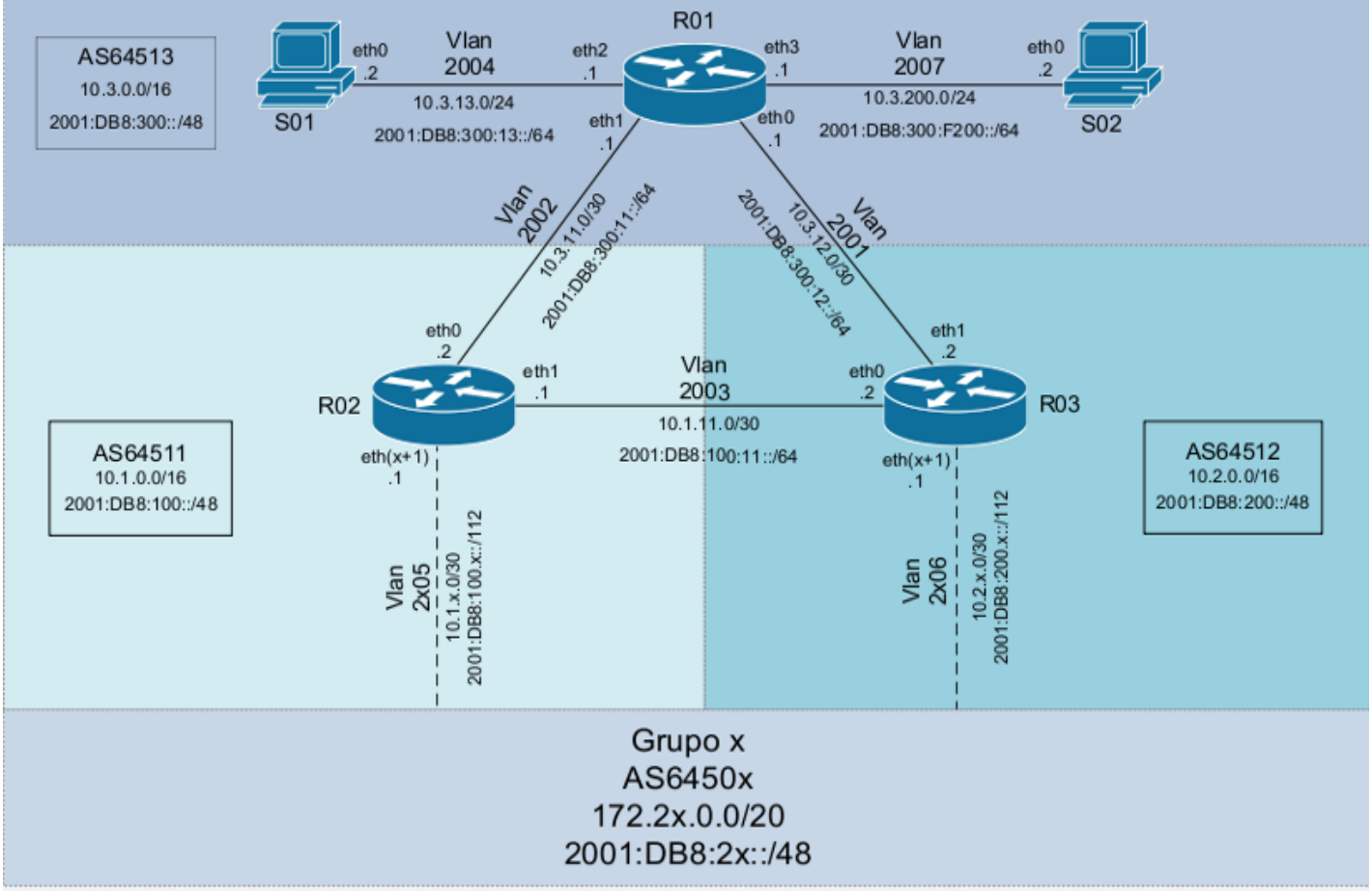
cgi.br nic.br


Laboratório de IPv6
AS64513
AS64512
AS64511
Grupo 1
AS64501
Grupo 2
AS64502
Grupo 3
AS64503
Grupo 4
AS64504
Grupo 5
AS64505
Grupo 6
AS64506
Grupo 7
AS64507
Grupo 8
AS64508

Laboratório de IPv6
Conexões entre núcleo e grupos



Laboratório de IPv6
Núcleo



S01		
Interface	IPv4	IPv6
eth0	10.3.13.2/24	2001:DB8:300:13::2/64

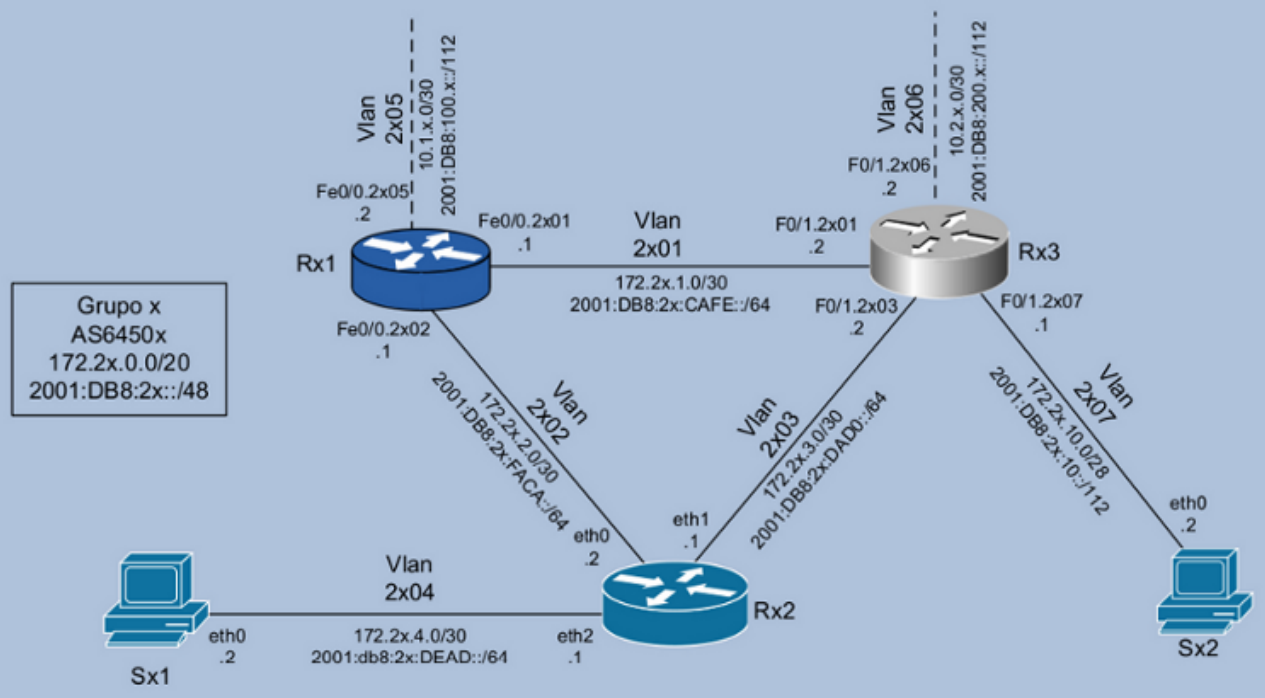
S02		
Interface	IPv4	IPv6
eth0	10.3.200.2/24	2001:DB8:300:F200::2/64

R01		
Interface	IPv4	IPv6
eth0	10.3.12.1/30	2001:DB8:300:12::1/64
eth1	10.3.11.1/30	2001:DB8:300:11::1/64
eth2	10.3.13.1/30	2001:DB8:300:13::1/64
eth3	10.3.200.1/24	2001:DB8:300:F200::1/64
lo	10.3.255.255/32	2001:DB8:300:FFFF::255/128

R02		
Interface	IPv4	IPv6
eth0	10.3.11.2/30	2001:DB8:300:11::2/64
eth1	10.1.11.1/30	2001:DB8:100:11::1/64
ethx	10.1.x.1/30	2001:DB8:100:x::1/112
lo	10.1.255.255/32	2001:DB8:100:FFFF::255/128

R03		
Interface	IPv4	IPv6
eth0	10.1.11.2/30	2001:DB8:100:11::2/64
eth1	10.3.12.2/30	2001:DB8:300:12::2/64
ethx	10.2.x.1/30	2001:DB8:200:x::1/112
lo	10.2.255.255/32	2001:DB8:200:FFFF::255/128

Laboratório de IPv6
Roteamento básico



Sx1		
Interface	IPv4	IPv6
eth0	172.2x.4.2/30	2001:DB8:2x:DEAD::2/64

Sx2		
Interface	IPv4	IPv6
eth0	172.2x.10.2/28	2001:DB8:2x:10::2/112

Rx1		
Interface	IPv4	IPv6
Fe0/0.2x01	172.2x.1.1/30	2001:DB8:2x:CAFE::1/64
Fe0/0.2x02	172.2x.2.1/30	2001:DB8:2x:FACA::1/64
Fe0/0.2x05	10.1.x.2/30	2001:DB8:100:x::2/112
lo0	172.2x.15.255/32	2001:DB8:2x:FFFF::255/128

Rx2			
Interface	IPv4	IPv6	Obs.
eth0	172.2x.2.2/30	2001:DB8:2x:FACA::2/64	
eth1	172.2x.3.1/30	2001:DB8:2x:DAD0::1/64	
eth2	172.2x.4.1/30	2001:DB8:2x:DEAD::1/64	
lo	172.2x.15.254/32	2001:DB8:2x:FFFF::254/128	iBGP

Rx3			
Interface	IPv4	IPv6	Obs.
F0/1.2x01	172.2x.1.2/30	2001:DB8:2x:CAFE::2/64	
F0/1.2x03	172.2x.3.2/30	2001:DB8:2x:DAD0::2/64	
F0/1.2x06	10.2.x.2/30	2001:DB8:200:x::2/112	
F0/1.2x07	172.2x.10.1/28	2001:DB8:2x:10::1/112	
loopback10	172.2x.15.253/32	2001:DB8:2x:FFFF::253/128	Router ID
loopback20	172.2x.15.252/32	2001:DB8:2x:FFFF::252/128	iBGP
loopback30	172.2x.15.251/32	2001:DB8:2x:FFFF::251/128	eBGP

Laboratorio – Enrutamiento IPv6

Objetivo: Implementar para IPv6 una política de enrutamiento externo y un protocolo de enrutamiento interno (IGP), en este caso OSPF, semejante a la implementación para IPv4 ya existente. Para realizar esta tarea, empezaremos revisando la configuración de IPv4, luego pasaremos a la configuración del direccionamiento IPv6 en las interfaces de los routers y los servidores, el protocolo de enrutamiento interno OSPFv3, la configuración del iBGP (interno) y del eBGP (externo – operadoras), y por último probaremos la conectividad IPv4 / IPv6.

Escenario inicial: En esta fase cada grupo representa un AS diferente con conexión para 2 proveedores de tránsito. Los links externos son utilizados para balanceo de carga y redundancia, es decir, cada link individualmente tiene que soportar todo el tráfico del AS, aunque en situaciones normales cada link deberá soportar solo la mitad de este tráfico (entrante y saliente). Ejemplo: Supongamos que el AS tiene 100Mbits de tráfico total y los links contratados utilizan contratos a demanda con franquicias de 50Mbps y capacidades de 100Mbps (percentil 95).

Cada AS tiene acceso a un router Cisco, un router Linux/Quagga, dos servidores Linux y a partir de ahora también hay un router Juniper.

Para acceder al router Juniper utilice el siguiente comando:

```
labnicX:~$juniper X
Trying 192.168.50.201...
Connected to 192.168.50.201.
Escape character is '^]'.

RX1 (ttyp0)

login: juniper
Password: Juniper (¡ATENCIÓN! La contraseña comienza con "J" mayúscula)

--- JUNOS 8.5R4.3 built 2008-08-12 23:14:39 UTC
juniper@RX1>
```

La política de enrutamiento externo y el protocolo de enrutamiento interno (IGP), en este caso OSPF, ya están implementados para IPv4 según las condiciones anteriores. El grupo debe probar la comunicación dentro del propio AS y con los demás AS (usar, por ejemplo, mtr, ping y traceroute IPv4).

El router Linux utiliza la aplicación Quagga para proveer los servicios de enrutamiento. Es importante destacar que, a diferencia de la implementación de los principales routers (por ejemplo Cisco y Juniper) que utilizan una única CLI (Command Line Interface) para realizar todas sus configuraciones, Quagga se basa en una CLI asociada a cada daemon. En Quagga existe un daemon específico para cada protocolo de enrutamiento, los cuales son tratados como procesos separados.

En este laboratorio utilizaremos los siguientes daemons: ospfd, ospf6d, bgpd y zebra. Las configuraciones de cada proceso se pueden editar de dos maneras diferentes:

1ª – Deteniendo el proceso a configurar:

```
[root@RX2]# /etc/init.d/"nome do daemon" stop
```

Edite el archivo de configuración

```
[root@RX2]# /etc/quagga/"nome do daemon".conf
```

Reinicie el proceso

```
[root@RX2]# /etc/init.d/"nome do daemon" start
```

Nota: Esta opción se recomienda solamente cuando se crea un archivo nuevo, para evitar inconsistencias entre la información del proceso en ejecución y la almacenada en el archivo.

2ª – Accediendo via telnet o a través de la terminal de configuración (CLI) de cada daemon mientras aun está en funcionamiento. De este modo todas las actualizaciones realizadas entrarán en funcionamiento de inmediato, sin necesidad de reiniciar el servicio.

A cada terminal se accede a través de un puerto TCP específico:

- zebra → 2601
- ospfd → 2604
- bgpd → 2605
- ospf6d → 2606

Ejemplo:

```
# telnet localhost 2601
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 0.99.4)
Copyright © 1999-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password: XXXXX
Router> ?
  enable            Turn on privileged commands
  exit              Exit current mode and down to previous mode
  help              Description of the interactive help system
  list              Print command list
  show              Show running system information
  who               Display who is on a vty
Router> enable
Password: XXXXX
Router# configure terminal
Router(config)# interface eth0
Router(config-if)# ip address 10.0.0.1/8
Router(config-if)# exit
```

Nota: Esta opción se recomienda para los sistemas que están en funcionamiento.

En este laboratorio las contraseñas para acceder y configurar todos los daemos de Quagga son “zebra”.

Puede obtener más información sobre la sintaxis de los comandos para los routers Cisco y Quagga en los siguientes enlaces:

- http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html
- http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-ospf.html
- http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-mptcl_bgp.html
- http://www.quagga.net/docs/docs-info.php
- http://www.juniper.net/techpubs/software/junos-security/junos-security10.0/junos-security-cli-reference/junos-security-cli-reference-IX.html

Ejercicio 1 - Verificar la conectividad IPv4

Inicialmente solo están configurados los protocolos de enrutamiento IPv4. Vamos a probar la comunicación dentro del propio ASN – con el núcleo y con los demás ASN (usando, por ejemplo, mtr, ping y traceroute IPv4).

Observe también la configuración del enrutamiento:

- En el router Juniper RX1:

```
labnicX:~$juniper X
Trying 192.168.50.201...
Connected to 192.168.50.201.
Escape character is '^]'.

RX1 (ttyp0)

login: juniper
Password: Juniper (¡ATENCIÓN! La contraseña comienza con "J" mayúscula)

--- JUNOS 8.5R4.3 built 2008-08-12 23:14:39 UTC
juniper@RX1> show bgp summary
juniper@RX1> show bgp group brief
```

- En el router Linux/Quagga RX2:

```
labnicX:~$ router X2
entered into CT 1X2
[root@RX2 /]# telnet localhost 2604
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 0.98.6).
Copyright 1996-2005 Kunihiro Ishiguro, et al.


User Access Verification

Password: zebra
ospfd-RX2# show ip ospf interface
ospfd-RX2# show ip ospf neighbor
```

- En el router Linux/Quagga RX2:

```
[root@RX2 /]# telnet localhost 2605
...
Password: zebra
bgpd-RX2# show ip bgp summary
bgpd-RX2# show ip bgp
```

- En el router Cisco RX3:

```
labnicX:~$ router X3
Trying 192.168.50.2...
Connected to 192.168.50.2 (192.168.50.2).
Escape character is '^]'.


User Access Verification

Username: cisco
Password: cisco
router-RX3# show ip int br
router-RX3# show ip proto
router-RX3# show ip ospf interface
router-RX3# show ip ospf neighbor
router-RX3# show ip bgp summary
router-RX3# show ip bgp
```

Ejercicio 2: Configuración de las interfaces de red

Siguiendo el plan de implementacion del protocolo IPv6, ya hemos realizado experimentos internos con el nuevo protocolo y ya hemos probado la conectividad entre los AS a través de túneles 6to4, lo que nos permitió evaluar que todos los dispositivos soportan IPv6. Ahora vamos a iniciar la implementación de IPv6 nativo en nuestro AS.

Después de recibir la distribución de un bloque de direcciones IPv6 /48 debemos planificar de qué forma se distribuirán esos recursos, decidiendo cuáles rangos de direcciones se utilizarán en la numeración interna de la red, cuáles se utilizarán para servicios, etc.

Ahora iniciaremos la configuración básica del direccionamiento IPv6 en todas las interfaces de red de los servidores y routers que se utilizarán en el enrutamiento interno del AS (OSPF), de acuerdo con la tabla de direcciones del diagrama "Laboratorio IPv6 - Grupos".

Acceda a los servidores SX1 y SX2 y edite el archivo /etc/sysconfig/network, agregando las siguientes líneas:

```
NETWORKING_IPV6=yes
IPV6_DEFAULTGW=2001:DB8:2X:YYYY::1 (agregue la dirección del router más próximo como gateway)
```

Edite también el archivo /etc/sysconfig/network-scripts/ifcfg-eth0 agregando las siguientes líneas:

```
IPV6INIT=yes
IPV6ADDR=2001:DB8:2X:YYYY::YYYY/YYY (consulte la dirección de cada servidor en la tabla de direcciones del diagrama "Laboratorio IPv6 – Grupos")
```

Luego reinicie las interfaces de red:

```
# /etc/init.d/network restart
```

Ahora vamos a configurar en el router Juniper las direcciones de las interfaces ge-0/0/0.2X01 y ge-0/0/0.2X02.

- En el router Juniper RX1:

```
juniper@RX1# edit
Entering configuration mode
Users currently editing the configuration:
  juniper terminal d0 (pid 17076) on since 2009-10-21 19:03:41 UTC, idle 01:00:07
      [edit]

[edit]
juniper@RX1# set interfaces ge-0/0/0 unit 2X01 family inet6 address
2001:db8:2X:cafe::1/64

[edit]
juniper@RX1# set interfaces ge-0/0/0 unit 2X02 family inet6 address
2001:db8:2X:faca::1/64

[edit]
juniper@RX1# commit
```

```
commit complete
```

Ahora vamos a configurar las interfaces del router Linux/Quagga RX2 editando el archivo /etc/sysconfig/network para agregar la siguiente línea:

```
NETWORKING_IPV6=yes
```

También edite los archivos /etc/sysconfig/network-scripts/ifcfg-eth_ (para eth0, eth1 e eth2), agregando las siguientes líneas:

```
IPV6INIT=yes
IPV6ADDR=2001:DB8:2X:YYYY::YYYY/YYY (consulte la dirección de cada interfaz del router en la tabla de direcciones del diagrama "Laboratorio IPv6 – Grupos"
```

Luego reinicie las interfaces de red:

```
[root@RX2]# /etc/init.d/network restart
```

Ahora vamos a configurar la dirección de loopback del router Linux/Quagga:

- En el router Linux/Quagga RX2:

```
[root@RX2]# telnet localhost 2601
Password: zebra
Router-RX2> enable
Password: zebra
Router-RX2# configure terminal

Router-RX2(config)# interface lo
Router-RX2(config-if)# ipv6 address 2001:DB8:2X:FFFF::254/128
Router-RX2(config-if)# exit
Router-RX2(config)# exit
Router-RX2 copy running-config startup-config
Router-RX2 exit
```

En el router Cisco, vamos a configurar las direcciones IPv6 de las interfaces FastEthernet0/1.2X01, 1.2X03 y 1.2X07 y de la interfaz Loopback10, además de deshabilitar el anuncio de mensajes Router Advertisement del Protocolo de Descubrimiento de Vecinos (Neighbor Discovery Protocol).

- En el router Cisco RX3:

```
router-RX3# configure terminal
router-RX3(config)# ipv6 unicast-routing
router-RX3(config)# ipv6 cef
router-RX3(config)# interface FastEthernet0/1

router-RX3(config-if)# interface FastEthernet0/1.2X01
router-RX3(config-subif)# ipv6 address 2001:DB8:2X:CAFE::2/64
```

```
router-RX3(config-subif)# ipv6 nd ra suppress

router-RX3(config-subif)# interface FastEthernet0/1.2X03
router-RX3(config-subif)# ipv6 address 2001:DB8:2X:DAD0::2/64
router-RX3(config-subif)# ipv6 nd ra suppress

router-RX3(config-subif)# interface FastEthernet0/1.2X07
router-RX3(config-subif)# ipv6 address 2001:DB8:2X:10::1/112
router-RX3(config-subif)# ipv6 nd ra suppress
router-RX3(config-subif)# exit

router-RX3(config)# interface Loopback10
router-RX3(config-if)# ipv6 address 2001:DB8:2X:FFFF::253/128
router-RX3(config)# exit
router-RX3# copy running-config startup-config
```

Ejercicio 3: OSPFv3

Ahora que todas las interfaces y loopbacks están configuradas, ya podemos habilitar y configurar el protocolo de enrutamiento interno OSPFv3 en los routers Juniper, Linux/Quagga y Cisco.

Vamos a habilitar el protocolo OSPF en las interfaces del router Juniper RX1:

```
juniper@RX1> edit
Entering configuration mode
Users currently editing the configuration:
  juniper terminal d0 (pid 17076) on since 2009-10-21 19:03:41 UTC, idle 01:00:07
      [edit]

[edit]
juniper@RX1# set protocols ospf3 export ospf-redistributes

[edit]
juniper@RX1# set protocols ospf3 area 0.0.0.0 interface ge-0/0/0.2X01

[edit]
juniper@RX1# set protocols ospf3 area 0.0.0.0 interface ge-0/0/0.2X02

[edit]
juniper@RX1# commit
commit complete
```

En el router Linux/Quagga RX2, para activar el daemon ospf6d , primero es necesario crear el archivo ospf6d.conf donde se almacenarán las configuraciones del protocolo OSPFv3:

```
[root@RX2]# cd /etc/quagga/
[root@RX2 quagga]# cat > ospf6d.conf
!
hostname ospf6d-RX2
password zebra
enable password zebra
log file /var/log/quagga/ospf6d.log
log stdout
!
debug ospf6 lsa unknown
!
line vty
!
[CTRL+D]
[root@RX2 quagga]# chown quagga:quagga ospf6d.conf
```

Inicie y acceda al daemon ospf6d para configurar OSPFv3:

```
[root@RX2 quagga]# /etc/init.d/ospf6d start
[root@RX2 quagga]# telnet ::1 2606
Password: zebra
ospf6d-RX2> enable
Password: zebra
ospf6d-RX2# configure terminal
ospf6d-RX2(config)# router ospf6
ospf6d-RX2(config-ospf6)# router-id 172.2X.15.25Y (puede utilizar la dirección IPv4 de loopback
```

del router como ID)

```
ospf6d-RX2(config-ospf6)# redistribute connected
ospf6d-RX2(config-ospf6)# redistribute static
ospf6d-RX2(config-ospf6)# interface eth0 area 0.0.0.0
ospf6d-RX2(config-ospf6)# interface eth1 area 0.0.0.0
ospf6d-RX2(config-ospf6)# exit
ospf6d-RX2(config)# exit
ospf6d-RX2# copy running-config startup-config
ospf6d-RX2# exit
```

- En el router Cisco, defina los parámetros básicos de OSPF y habilítelo en las interfaces:

```
router-RX3# configure terminal
router-RX3(config)# ipv6 router ospf 200
router-RX3(config-rtr)# redistribute connected
router-RX3(config-rtr)# redistribute static
router-RX3(config-rtr)# exit

router-RX3(config)# interface FastEthernet0/1.2X01
router-RX3(config-subif)# ipv6 ospf 200 area 0

router-RX3(config-subif)# interface FastEthernet0/1.2X03
router-RX3(config-subif)# ipv6 ospf 200 area 0
router-RX3(config-subif)# exit
router-RX3(config)# exit
router-RX3# copy running-config startup-config
```

Ahora que OSPF está configurado y activado en todos los routers, vamos a consultar la tabla de vecinos para verificar que todas las rutas internas estén siendo anunciadas correctamente:

- En el router Juniper RX1:

```
juniper@RX1> show ospf3 interface
juniper@RX1> show ospf3 neighbor
```

- En el router Linux/Quagga RX2 (ospf6d):

```
ospf6d-RX2# show ipv6 ospf neighbor
```

- En el router Cisco RX3:

```
router-RX3# show ipv6 ospf neighbor
```

- Otros comandos para Cisco y Quagga (ospf6d):

```
show ipv6 ospf data
show ipv6 ospf interface
show ipv6 ospf
```

Pruebe la conectividad IPv6 dentro de su AS. Haga pings entre los routers y servidores de diferentes segmentos de la red, todos los dispositivos internos ya deberían estar "viéndose".

Ejercicio 4: BGP

Ahora que la conectividad interna ya está funcionando, podemos iniciar el proceso para establecer la conexión con los AS vecinos. Para ello, vamos a configurar el protocolo BGP en los routers de borde RX1 y RX3, para que éstos puedan comunicarse con los routers de borde R02 y R03 de nuestros proveedores de tránsito. A este tipo de relación –entre AS vecinos– le damos el nombre de external BGP (eBGP). Esta tarea se dividirá en algunos pasos como: establecer sesiones BGP dentro del propio AS (iBGP), definir la política de enrutamiento y, por último, después de configurar eBGP, definir políticas de flujo de salida de datos.

Ejercicio 4a: iBGP

Para comenzar vamos a establecer sesiones BGP entre todos los routers internos de nuestro AS (iBGP - internal BGP) para mantener la consistencia del enrutamiento interno.

Las sesiones iBGP se establecerán entre interfaces loopback que, por ser lógicas, colaboran para aumentar la disponibilidad de la red.

Primero vamos a configurar las loopback utilizadas en esa comunicación. Recuerde que esta interfaz ya se utiliza para el enrutamiento IPv4, por eso no vamos a crearla sino que solo le vamos a agregar la dirección IPv6.

- En el router Juniper RX1:

```
juniper@RX1# edit

[edit]
juniper@RX1# set interfaces lo0 unit 0 family inet6 address
2001:db8:2X:ffff::255/128

[edit]
juniper@RX1# commit
commit complete
```

- En el router Linux/Quagga RX2 utilizaremos el mismo direccionamiento de la loopback lo.

- En el router Cisco RX3:

```
router-RX3# show run int Loopback20
...
router-RX3# configure terminal
router-RX3(config)# interface Loopback20
router-RX3(config-if)# ipv6 address 2001:DB8:2X:FFFF::252/128
```

Ahora vamos a configurar las relaciones entre los vecinos:

- En el router Juniper RX1:

```
juniper@RX1# edit

[edit]
juniper@RX1# set routing-options autonomous-system 6450X

[edit]
juniper@RX1# set protocols bgp group iBGPv6 type internal

[edit]
juniper@RX1# set protocols bgp group iBGPv6 local-address 2001:DB8:2X:FFFF::255

[edit]
juniper@RX1# set protocols bgp group iBGPv6 export next-hop-self

[edit]
juniper@RX1# set protocols bgp group iBGPv6 neighbor 2001:DB8:2X:FFFF::252

[edit]
juniper@RX1# set protocols bgp group iBGPv6 neighbor 2001:DB8:2X:FFFF::254

[edit]
juniper@RX1# commit
commit complete
```

- En el Linux/Quagga RX2:

```
[root@RX2 /]# telnet localhost 2605
Password: zebra
bgpd-RX2> enable
Password:zebra
bgpd-RX2# configure terminal
bgpd-RX2(config)# router bgp 6450X
bgpd-RX2(config-router)# neighbor 2001:DB8:2X:FFFF::252 remote-as 6450X
bgpd-RX2(config-router)# neighbor 2001:DB8:2X:FFFF::252 description RX3
bgpd-RX2(config-router)# neighbor 2001:DB8:2X:FFFF::252 update-source
2001:DB8:2X:FFFF::254
bgpd-RX2(config-router)# address-family ipv6 unicast
bgpd-RX2(config-router-af)# neighbor 2001:DB8:2X:FFFF::252 activate
bgpd-RX2(config-router-af)# neighbor 2001:DB8:2X:FFFF::252 soft-reconfiguration
inbound
bgpd-RX2(config-router-af)# exit


bgpd-RX2(config-router)# neighbor 2001:DB8:2X:FFFF::255 remote-as 6450X
bgpd-RX2(config-router)# neighbor 2001:DB8:2X:FFFF::255 description RX1
bgpd-RX2(config-router)# neighbor 2001:DB8:2X:FFFF::255 update-source
2001:DB8:2X:FFFF::254
bgpd-RX2(config-router)# address-family ipv6 unicast
bgpd-RX2(config-router-af)# neighbor 2001:DB8:2X:FFFF::255 activate
bgpd-RX2(config-router-af)# neighbor 2001:DB8:2X:FFFF::255 soft-reconfiguration
inbound
bgpd-RX2(config-router-af)# exit
bgpd-RX2(config-router)# exit
bgpd-RX2(config)# exit
bgpd-RX2# copy running-config startup-config
```

- En el router Cisco RX3:

```
router-RX3# configure terminal
router-RX3(config)# router bgp 6450X
router-RX3(config-router)# neighbor 2001:DB8:2X:FFFF::254 remote-as 6450X
router-RX3(config-router)# neighbor 2001:DB8:2X:FFFF::254 description RX2
router-RX3(config-router)# neighbor 2001:DB8:2X:FFFF::254 update-source Loopback20
router-RX3(config-router)# neighbor 2001:DB8:2X:FFFF::254 version 4
router-RX3(config-router)# address-family ipv6 unicast
router-RX3(config-router-af)# neighbor 2001:DB8:2X:FFFF::254 activate
router-RX3(config-router-af)# neighbor 2001:DB8:2X:FFFF::254 soft-reconfiguration
inbound
router-RX3(config-router-af)# exit

router-RX3(config-router)# neighbor 2001:DB8:2X:FFFF::255 remote-as 6450X
router-RX3(config-router)# neighbor 2001:DB8:2X:FFFF::255 description RX1
router-RX3(config-router)# neighbor 2001:DB8:2X:FFFF::255 update-source Loopback20
router-RX3(config-router)# neighbor 2001:DB8:2X:FFFF::255 version 4
router-RX3(config-router)# address-family ipv6 unicast
router-RX3(config-router-af)# neighbor 2001:DB8:2X:FFFF::255 activate
router-RX3(config-router-af)# neighbor 2001:DB8:2X:FFFF::255 soft-reconfiguration
inbound
router-RX3(config-router-af)# exit
router-RX3(config-router)# exit
router-RX3(config)# exit
router-RX3#copy running-config startup-config
```

Verifique si se establecieron las relaciones entre los vecinos:

- En el router Juniper RX1:

```
juniper@RX1> show bgp group brief
```

- En el router Linux/Quagga RX2:

```
bgpd-RX2# sh bgp summary
```

- En el router Cisco RX3:

```
Router-RX3# sh bgp ipv6 unicast summary
```

También podemos utilizar el siguiente comando para consultar simultáneamente las tablas IPv4 e IPv6:

```
Router-RX3# sh bgp all summary
```

Ejercicio 4b: Consideraciones y preparativos para influenciar el tráfico de entrada

Nuestro AS recibió la distribución del bloque de direcciones IPv6 2001:0DB8:002X::/48. Para influenciar el tráfico de entrada vamos a distribuir los servicios y el consumo de tráfico entre los dos links (balanceo de carga). Para eso vamos a dividir el bloque /48 en dos partes, anunciando cada una por un único link.

Identifique los dos bloques:

1º - _____

2º - _____

Para lograr redundancia se utilizará el prefijo IPv6 correspondiente a todo el bloque /48.

Ejercicio 4c: eBGP

Ahora ya podemos configurar la relación entre los AS vecinos, estableciendo una conexión BGP entre nuestros routers de borde y los routers de borde de nuestros proveedores de tránsito. Primero vamos a configurar la loopback y las interfaces que se utilizarán en la comunicación eBGP. Recuerde que las interfaces ya están siendo utilizadas para el enrutamiento IPv4, por eso no vamos a crearlas sino que solo les vamos a agregar la dirección IPv6:

Las sesiones eBGP se establecerán de dos formas:

- Entre el router Juniper RX1 y el AS 64511 utilizaremos el direccionamiento de las interfaces físicas entre los mismos (forma estándar);

- Entre el router Cisco RX3 y el AS 64512 utilizaremos el direccionamiento de las interfaces loopback (para aumentar la seguridad).

Vamos a agregar las direcciones en las interfaces.

- En el router Juniper RX1:

```
      [edit]
      juniper@RX1# set interfaces ge-0/0/0 unit 2X05 family inet6 address
2001:db8:100:X::2/112

      [edit]
      juniper@RX1# commit
      commit complete
```

- En el router Cisco RX3:

```
      router-RX3# configure terminal
      router-RX3(config)# interface FastEthernet0/1.2X06
      router-RX3(config-subif)# ipv6 address 2001:DB8:200:X::2/112
```

```
router-RX3(config-subif)# ipv6 nd ra suppress
router-RX3(config-subif)# exit
router-RX3(config-if)# interface Loopback30
router-RX3(config-if)# ipv6 address 2001:DB8:2X:FFFF::251/128
```

Vamos a configurar las rutas estáticas en nuestros routers de borde para generar los prefijos IPv6 y permitir la conectividad entre las loopback del router RX3 y del router del AS 64512.

- En el router Juniper RX1:

```
[edit]
juniper@RX1# set routing-options rib inet6.0 static route ::/0 discard
[edit]
juniper@RX1# set routing-options rib inet6.0 static route 2001:db8:2X:8000::/49
discard

[edit]
juniper@RX1# set routing-options rib inet6.0 static route 2001:db8:2X::/48 discard
```

- En el router Cisco RX3:

```
router-RX3#configure terminal
router-RX3(config)#ipv6 route 2001:DB8:2X::/48 Null0
router-RX3(config)#ipv6 route 2001:DB8:2X::/49 Null0
router-RX3(config)#ipv6 route ::/0 Null0
router-RX3(config)#ipv6 route 2001:DB8:200:FFFF::255/128 2001:DB8:200:X::1
router-RX3(config)# exit
router-RX3#copy running-config startup-config
```

Ahora que las interfaces ya están configuradas, vamos a establecer las relaciones entre nuestro AS y los AS vecinos:

- En el router Juniper RX1:

```
[edit]
juniper@RX1# set protocols bgp group eBGP-AS64511v6 neighbor 2001:db8:100:X::1
peer-as 64511

¡ATENCIÓN! Las configuraciones de BGP solo se deben aplicar después de establecer las
políticas de flujo de datos. Por lo tanto, no utilice el comando "commit" todavía.
```

- En el router Cisco RX3:

```
router-RX3# configure terminal
router-RX3(config)# router bgp 6450X
router-RX3(config-router)# neighbor 2001:DB8:200:FFFF::255 remote-as 64512
router-RX3(config-router)# neighbor 2001:DB8:200:FFFF::255 shutdown
router-RX3(config-router)# neighbor 2001:DB8:200:FFFF::255 description R03
router-RX3(config-router)# neighbor 2001:DB8:200:FFFF::255 ebgp-multihop 2
router-RX3(config-router)# neighbor 2001:DB8:200:FFFF::255 update-source
```

```
Loopback30
      router-RX3(config-router)# neighbor 2001:DB8:200:FFFF::255 version 4
      router-RX3(config-router)# address-family ipv6
      router-RX3(config-router-af)# neighbor 2001:DB8:200:FFFF::255 activate
      router-RX3(config-router-af)# neighbor 2001:DB8:200:FFFF::255 soft-reconfiguration
inbound
      router-RX3(config-router-af)# exit
      router-RX3(config-router)# exit
      router-RX3(config)# exit
      router-RX3# copy running-config startup-config
```

Ejercicio 4d: Control de flujos de entrada

La aplicación de las políticas de anuncios enviados, las cuales van a interferir con el tráfico de entrada (AS-OUT), tendrá dos funciones:

- Redundancia:
  - El anuncio del prefijo /48 (equivalente a todo el bloque del AS) se deberá enviar a todos los AS externos.

- Balanceo de carga:
  - el tráfico del rango 2001:DB8:2X::/49 debe entrar preferentemente por el AS 64512;
  - el tráfico del rango 2001:DB8:2X:8000::/49 debe entrar preferentemente por el AS 64511.

Ejercicio 4e: Control de flujos de salida

Para influenciar el tráfico de salida (AS-IN), los prefijos recibidos se deberán distribuir preferentemente entre los dos links, de modo que también sea posible proveer redundancia y balanceo de carga.

Consideremos que el tráfico de nuestro AS con destino al AS64513 puede ser dividido en partes iguales, de modo que el tráfico con destino al primer prefijo /49 del AS64513 debe salir preferentemente a través de AS64512 y el tráfico con destino al segundo prefijo /49 del AS64513 debe salir preferentemente a través de AS64511.

Para realizar las dos tareas mencionadas usted se puede basar en las configuraciones que ya existen para IPv4.

Ejercicio 4f: Levantar y probar las sesiones eBGP

Ahora que las políticas de enrutamiento ya están aplicadas podemos levantar las sesiones eBGP.

- En el router Juniper RX1:

```
[edit]
juniper@RX1# commit
commit complete
```

- En el router Cisco:

```
router-RX3(config-router)#no neighbor 2001:DB8:200:FFFF::255 shutdown
```

Ahora verifique si se establecieron las relaciones entre los AS vecinos: Analice el estado de las conexiones BGP:

- En el router Juniper RX1:

```
juniper@RX1> show bgp summary
```

- En el router Cisco:

```
router-RX3# show bgp ipv6 unicast summary
```

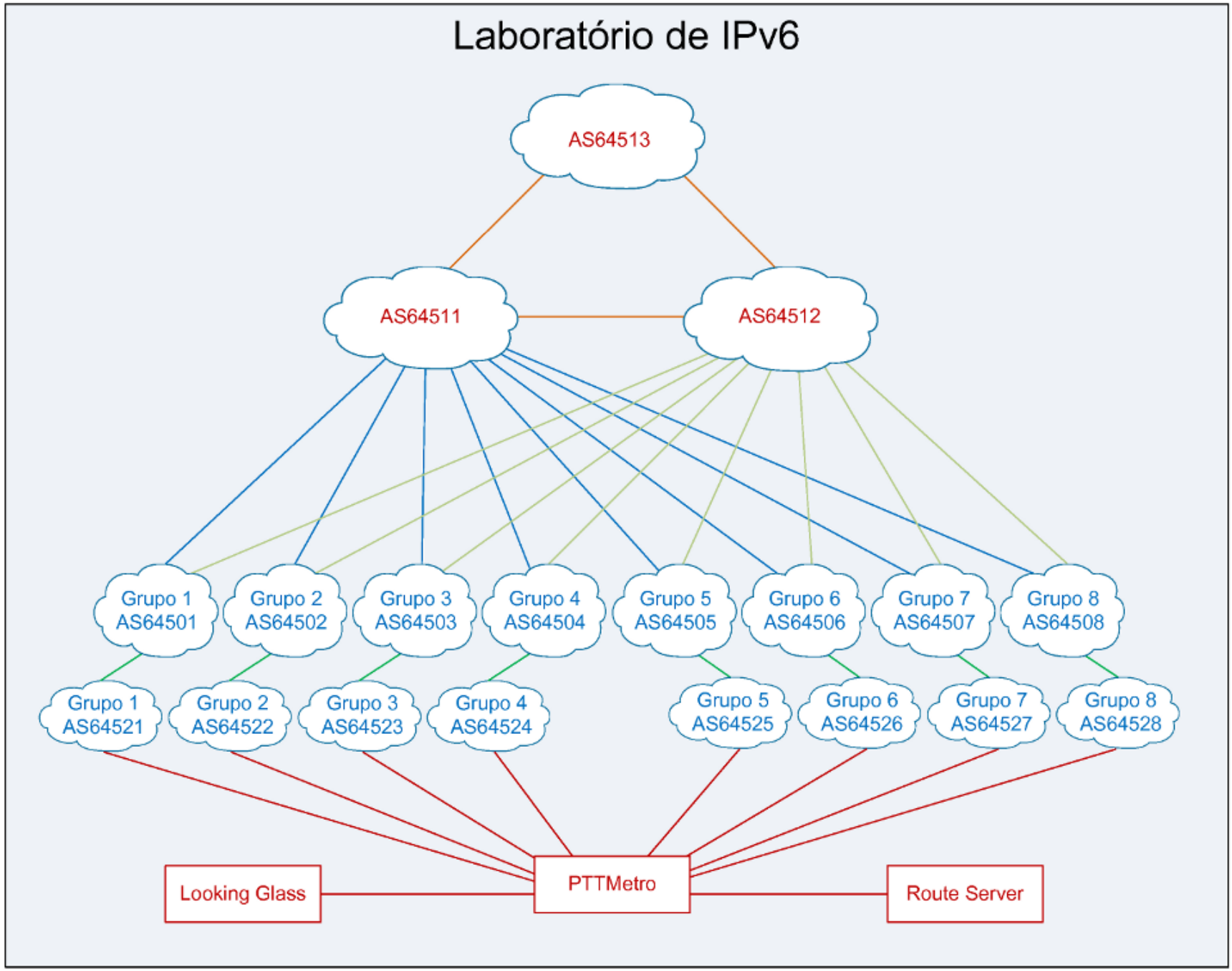
Pruebe la conectividad entre los routers de borde. Haga pings y traceroutes entre los routers y servidores de su AS y los routers y servidores de los AS centrales. También debería ser posible comunicarse con los AS de los otros grupos del laboratorio, ya que también ellos han completado esta parte de los ejercicios. Consulte con los grupos más cercanos para ver si ellos ya han completado el ejercicio y pruebe la conectividad entre los AS.

El AS 64513 tiene un Looking-Glass configurado. Acceda vía telnet para verificar si el anuncio de las rutas de nuestro AS fueron configuradas correctamente, siguiendo las políticas de enrutamiento establecidas. (La dirección a la cual acceder será informada por el instructor).

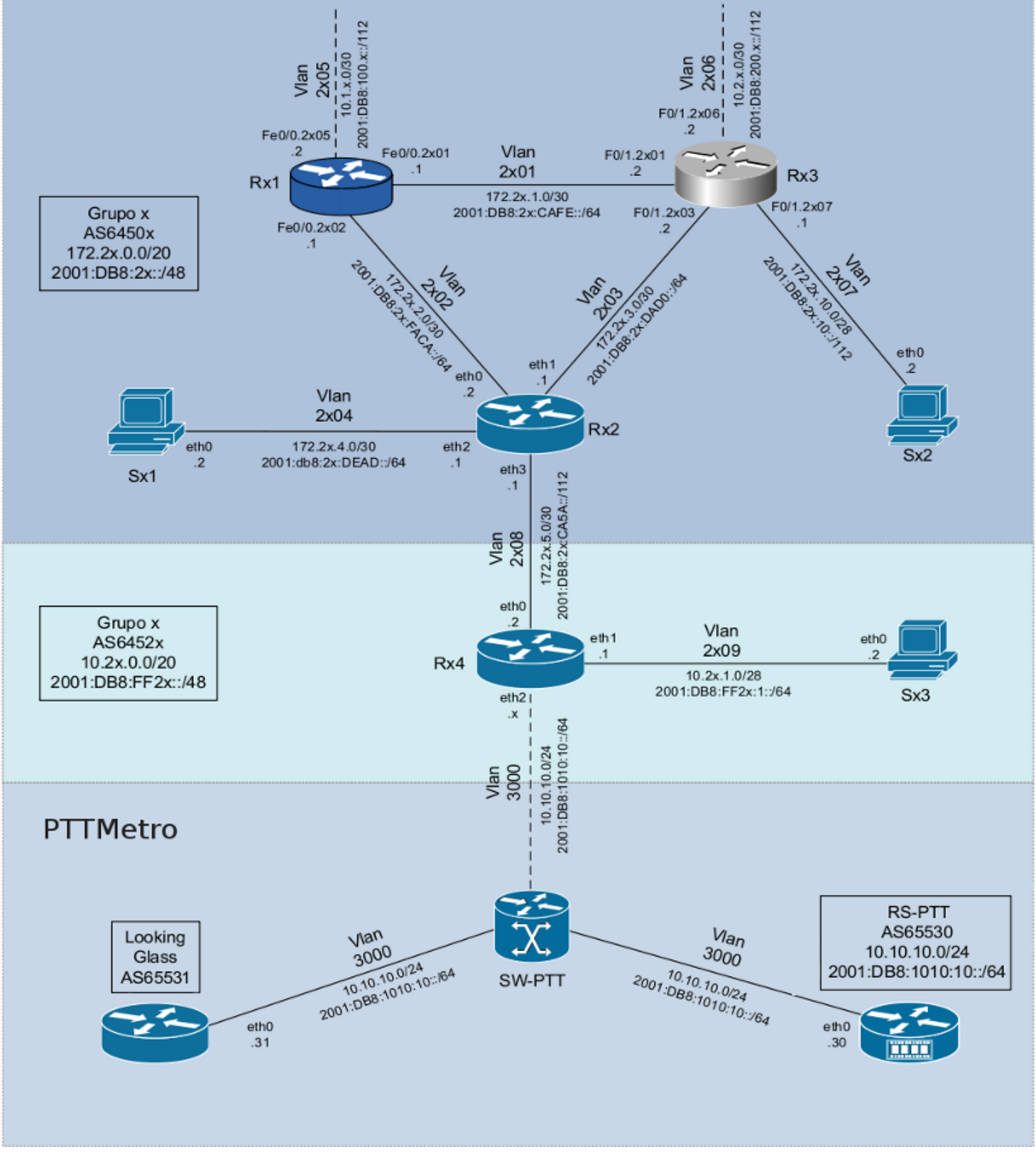
Laboratorio – Enrutamiento IPv6 (Parte 2)
Proveer tránsito y conectarse al PTTMetro

Objetivo: Configurar el AS del grupo para que éste proveea tránsito tanto IPv4 como IPv6 a un AS cliente y luego configurar la conexión entre ese AS cliente y un Punto de Intercambio de Tráfico (PTT).

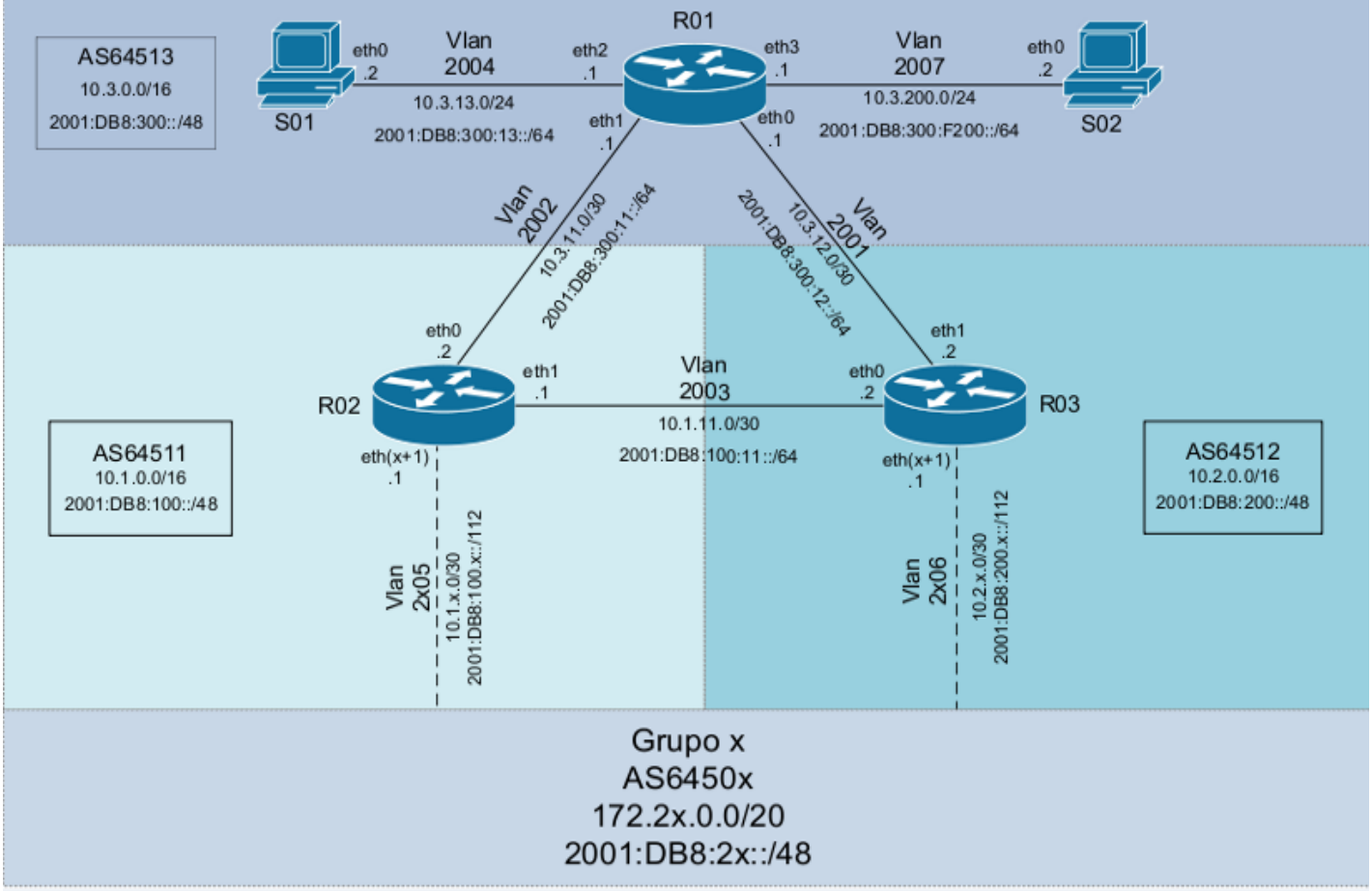
Escenario inicial: Una vez realizadas las configuraciones del enrutamiento interno y las configuraciones del enrutamiento externo con los AS centrales, nuestro AS (AS6450X) ya está en funcionamiento. Ahora agregaremos a nuestro escenario un AS cliente, formado por un router Linux/Quagga y un servidor Linux. Las configuraciones de direccionamiento y enrutamiento IPv6 del AS cliente aun no han sido establecidas. Una vez realizadas estas configuraciones, este AS cliente se conectará al PTTMetro.


Laboratório de IPv6
AS64513
AS64511
AS64512
Grupo 1 AS64501
Grupo 2 AS64502
Grupo 3 AS64503
Grupo 4 AS64504
Grupo 5 AS64505
Grupo 6 AS64506
Grupo 7 AS64507
Grupo 8 AS64508
Grupo 1 AS64521
Grupo 2 AS64522
Grupo 3 AS64523
Grupo 4 AS64524
Grupo 5 AS64525
Grupo 6 AS64526
Grupo 7 AS64527
Grupo 8 AS64528
Looking Glass
PTTMetro
Route Server

Laboratório de IPv6
Grupos e PTT



Laboratório de IPv6
Núcleo



S01		
Interface	IPv4	IPv6
eth0	10.3.13.2/24	2001:DB8:300:13::2/64

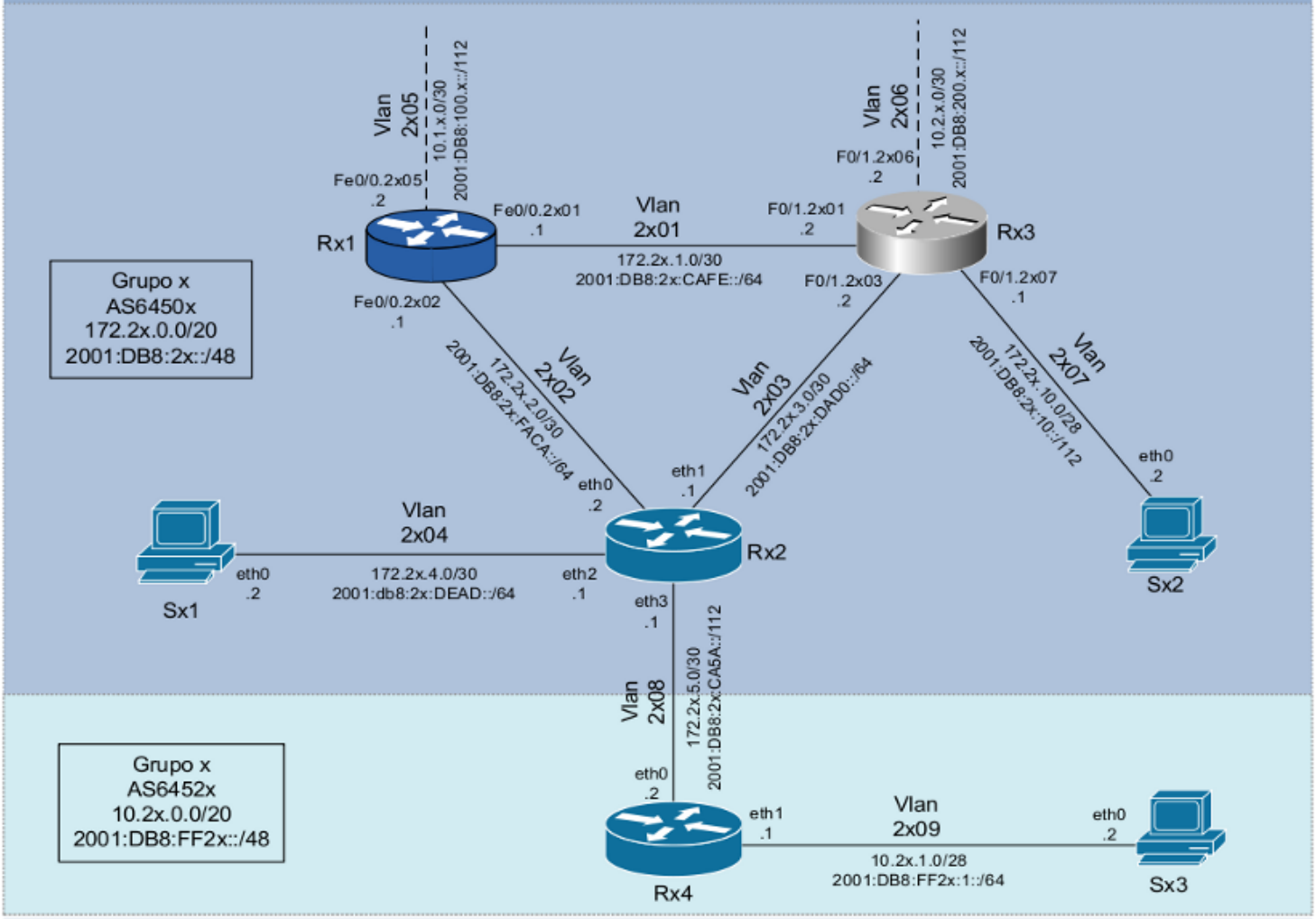
S02		
Interface	IPv4	IPv6
eth0	10.3.200.2/24	2001:DB8:300:F200::2/64

R01		
Interface	IPv4	IPv6
eth0	10.3.12.1/30	2001:DB8:300:12::1/64
eth1	10.3.11.1/30	2001:DB8:300:11::1/64
eth2	10.3.13.1/30	2001:DB8:300:13::1/64
eth3	10.3.200.1/24	2001:DB8:300:F200::1/64
lo	10.3.255.255/32	2001:DB8:300:FFFF::255/128

R02		
Interface	IPv4	IPv6
eth0	10.3.11.2/30	2001:DB8:300:11::2/64
eth1	10.1.11.1/30	2001:DB8:100:11::1/64
ethx	10.1.x.1/30	2001:DB8:100:x::1/112
lo	10.1.255.255/32	2001:DB8:100:FFFF::255/128

R03		
Interface	IPv4	IPv6
eth0	10.1.11.2/30	2001:DB8:100:11::2/64
eth1	10.3.12.2/30	2001:DB8:300:12::2/64
ethx	10.2.x.1/30	2001:DB8:200:x::1/112
lo	10.2.255.255/32	2001:DB8:200:FFFF::255/128

Laboratório de IPv6
Roteamento com AS



Sx1		
Interface	IPv4	IPv6
eth0	172.2x.4.2/30	2001:DB8:2x:DEAD::2/64

Sx2		
Interface	IPv4	IPv6
eth0	172.2x.10.2/28	2001:DB8:2x:10::2/112

Sx3		
Interface	IPv4	IPv6
eth0	10.2x.1.2/28	2001:DB8:FF2x:1::2/64

Rx1		
Interface	IPv4	IPv6
Fe0/0.2x01	172.2x.1.1/30	2001:DB8:2x:CAFE::1/64
Fe0/0.2x02	172.2x.2.1/30	2001:DB8:2x:FACA::1/64
Fe0/0.2x05	10.1.x.2/30	2001:DB8:100:x::2/112
lo0	172.2x.15.255/32	2001:DB8:2x:FFFF::255/128

Rx4		
Interface	IPv4	IPv6
eth0	172.2x.5.2/30	2001:DB8:2x:CA5A::2/112
eth1	10.2x.1.1/28	2001:DB8:FF2x:1::1/64
lo	10.2x.15.255/32	2001:DB8:FF2x:FFFF::255/128

Rx2			
Interface	IPv4	IPv6	Obs.
eth0	172.2x.2.2/30	2001:DB8:2x:FACA::2/64	
eth1	172.2x.3.1/30	2001:DB8:2x:DAD0::1/64	
eth2	172.2x.4.1/30	2001:DB8:2x:DEAD::1/64	
eth3	172.2x.5.1/30	2001:DB8:2x:CA5A::1/112	
lo	172.2x.15.254/32	2001:DB8:2x:FFFF::254/128	iBGP
lo	172.2x.15.250/32	2001:DB8:2x:FFFF::250/128	eBGP

Rx3			
Interface	IPv4	IPv6	Obs.
F0/1.2x01	172.2x.1.2/30	2001:DB8:2x:CAFE::2/64	
F0/1.2x03	172.2x.3.2/30	2001:DB8:2x:DAD0::2/64	
F0/1.2x06	10.2.x.2/30	2001:DB8:200:x::2/112	
F0/1.2x07	172.2x.10.1/28	2001:DB8:2x:10::1/112	
loopback10	172.2x.15.253/32	2001:DB8:2x:FFFF::253/128	Router ID
loopback20	172.2x.15.252/32	2001:DB8:2x:FFFF::252/128	iBGP
loopback30	172.2x.15.251/32	2001:DB8:2x:FFFF::251/128	eBGP

Ejercicio 1: Configuración del AS cliente.

Conectaremos a nuestro AS (AS6450X) un AS cliente (AS6452X), al cual proveeremos tanto tránsito IPv4 como tránsito IPv6. Sin embargo, para establecer este servicio es necesario cumplir algunas etapas:

Para acceder al router Rx4, primero es necesario establecer una sesión eBGP IPv4 entre los routers Rx2 (AS6450X) y Rx4 (AS6452X) utilizando la dirección de las interfaces loopback.

1º Paso: Direccionamiento

- Este AS aun no tiene ninguna configuración definida, ni de direccionamiento ni de enrutamiento. Por lo tanto, nuestro primer paso será configurar todo el direccionamiento IPv6 en el router Rx4 y en el servidor Sx3. Consulte el diagrama y la tabla de direccionamiento en la página anterior para saber cuáles direcciones se deben agregar.

También es necesario configurar las direcciones de la interfaz loopback y de la interfaz eth3 del router Rx2.

Estas configuraciones son similares a las realizadas en el Ejercicio 2 de la primera parte del laboratorio (Configuración de las interfaces de red).

2º Paso: eBGP

- Ahora que las direcciones de las interfaces ya están definidas, podemos configurar la relación entre nuestro AS y el AS cliente, estableciendo una sesión eBGP entre los routers de borde Rx2 (AS6450X) y Rx4 (AS6452X). La sesión eBGP se debe establecer utilizando el direccionamiento de las interfaces loopback.

Por último, debemos configurar la política de tránsito de nuestro AS (AS6450X) y evitar la recepción de anuncios innecesarios provenientes del AS cliente.

3º Paso: Probar la conectividad

- Ahora el router y el servidor del AS cliente ya deben tener conectividad tanto IPv4 como IPv6 con todos los dispositivos de nuestro AS y con los dos AS vecinos. Pruebe la comunicación dentro del propio AS y con los demás AS usando comandos como mtr, ping y traceroute, por ejemplo. Utilice el comando traceroute para verificar las rutas utilizadas en la comunicación entre el AS cliente y los AS vecinos. Obsérvelas bien, ya que las utilizaremos para comparar los resultados obtenidos ahora con los resultados que obtendremos en el próximo ejercicio.

Ejercicio 2: Establecer una conexión con el PTTMetro.

En esta etapa final del laboratorio de enrutamiento vamos a conectar el AS cliente a un Punto de Intercambio de Tráfico (PTT). Para ello es necesario establecer una conexión eBGP con el Route Server (AS65530). Use el diagrama "Laboratorio IPv6 – Grupo y PTT" para verificar las direcciones que se deben utilizar en la interfaz eth2 del router Rx4 y en el establecimiento de la conexión BGP.

El AS cliente (6452X) debe construir su política de enrutamiento de manera que se prefieran los caminos de entrada y salida a través del PTTMetro. Esto se puede realizar, por ejemplo, de la siguiente manera: Para el AS6450X se deberá anunciar solo el prefijo IPv6 /48. Para el Route Server (AS65530) se deberán anunciar los dos prefijos IPv6 /49, además de aumentar el valor de *Local-Preference* a 150 para todos los prefijos aprendidos a través del PTTMetro.

Para minimizar el impacto del crecimiento de la Tabla BGP se deben evitar los anuncios innecesarios, incluso dentro del PTTMetro.

También establezca una sesión eBGP con el Looking-Glass (AS65531). Para "alimentar" el Looking-Glass se deben anunciar todos los prefijos conocidos por el AS6452X.

Ahora compare los anuncios del AS cliente (AS6452X) en los Looking-Glasses del PTTMetro y del AS64513.

Una vez que lo haya hecho, pruebe la conectividad entre el AS cliente y los AS de los otros grupos y trace las rutas entre los mismos. Compare las rutas utilizadas ahora (con la conexión con el PTT ya establecida) con las rutas que fueron utilizadas anteriormente. ¿Cuál es la principal diferencia?

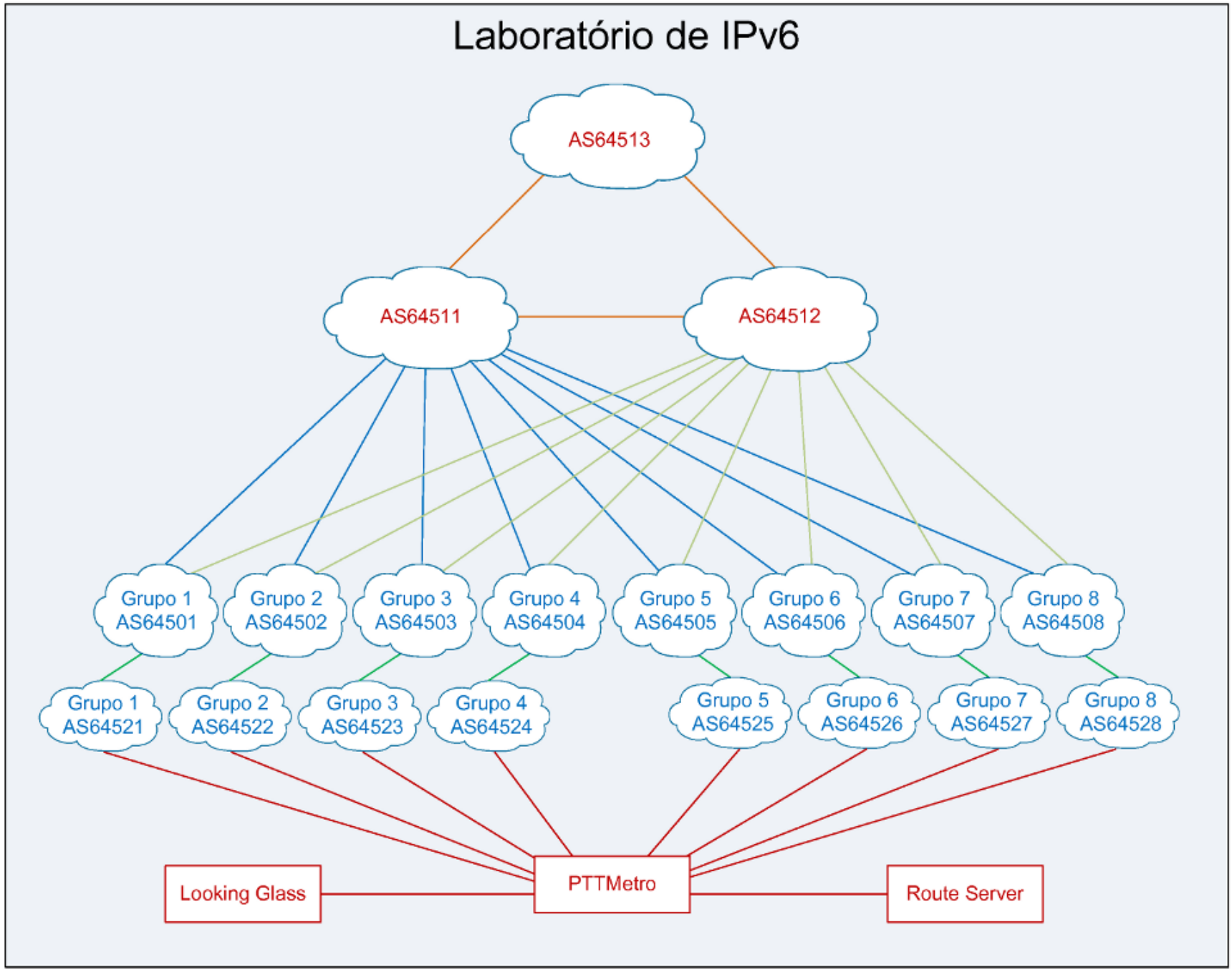
Publicação
ceptro.br

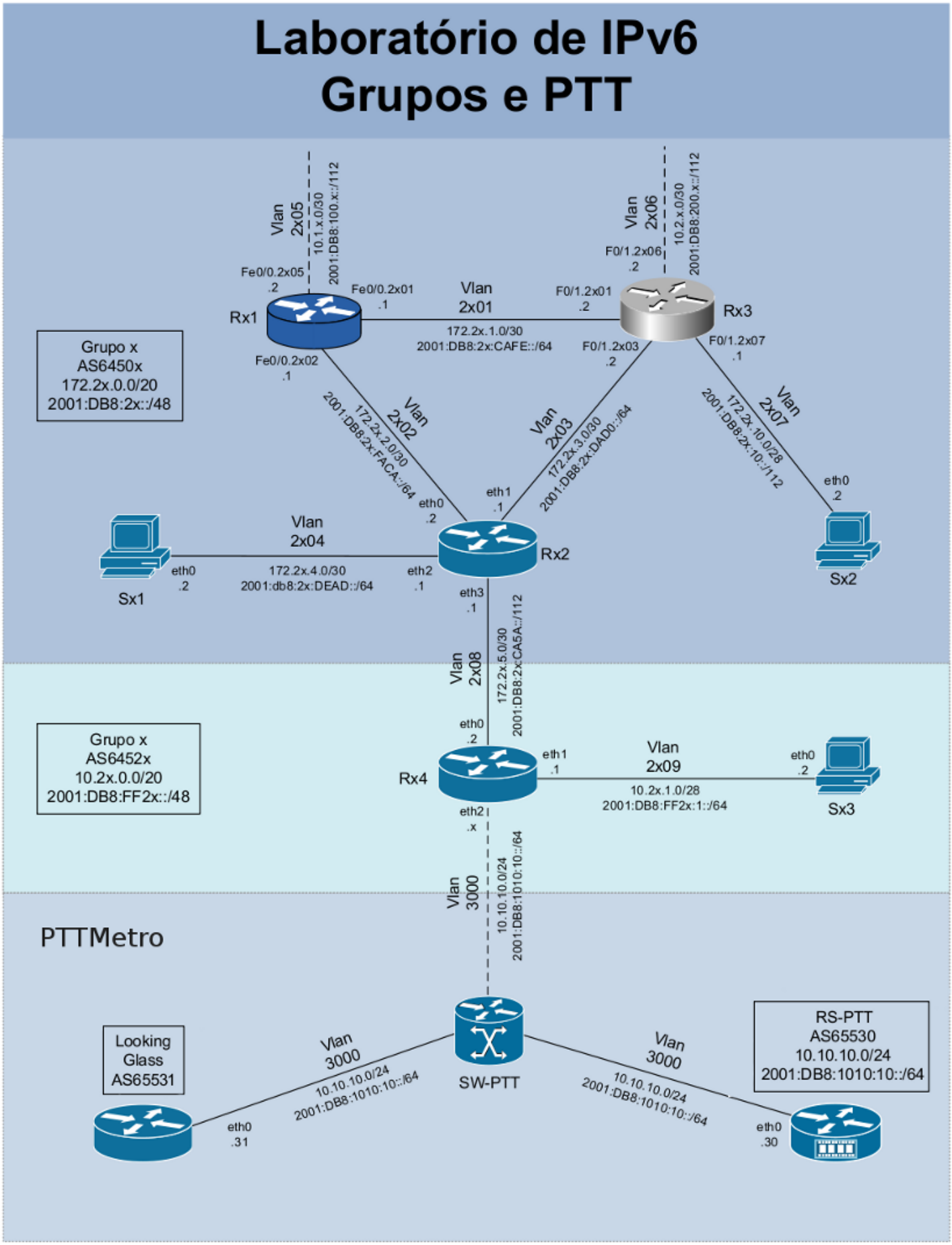
IPv6.br

Curso IPv6 básico

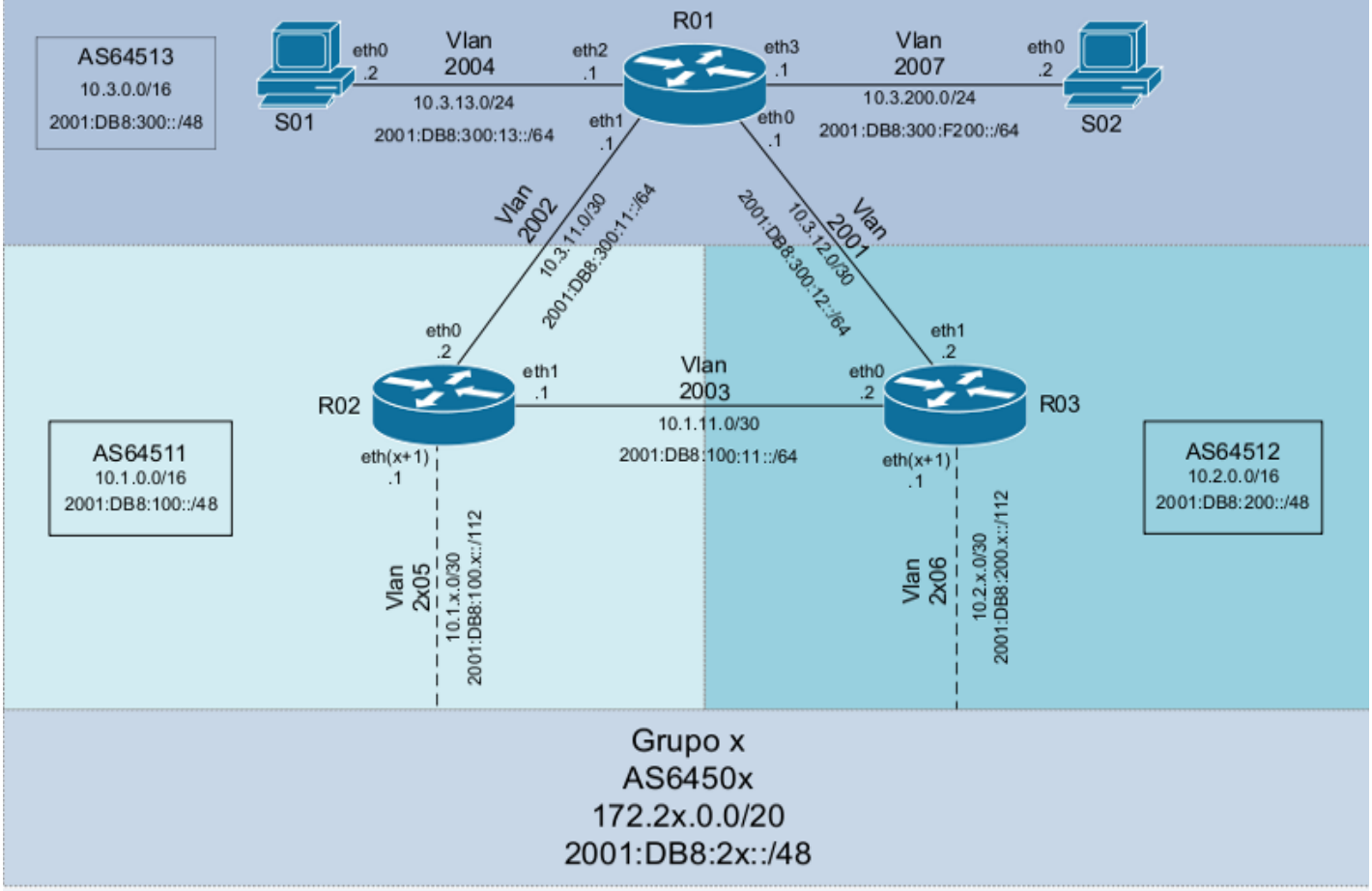
Laboratório: DNS

cgi.br nic.br


Laboratório de IPv6
AS64513
AS64511
AS64512
Grupo 1 AS64501
Grupo 2 AS64502
Grupo 3 AS64503
Grupo 4 AS64504
Grupo 5 AS64505
Grupo 6 AS64506
Grupo 7 AS64507
Grupo 8 AS64508
Grupo 1 AS64521
Grupo 2 AS64522
Grupo 3 AS64523
Grupo 4 AS64524
Grupo 5 AS64525
Grupo 6 AS64526
Grupo 7 AS64527
Grupo 8 AS64528
Looking Glass
PTTMetro
Route Server


Laboratório de IPv6
Grupos e PTT
Vlan 2x05
10.1.x.0/30
2001:DB8:100.x::/112
Vlan 2x06
10.2.x.0/30
2001:DB8:200.x::/112
Fe0/0.2x05
.2
Fe0/0.2x01
.1
Vlan 2x01
F0/1.2x06
.2
F0/1.2x01
.2
Rx1
Rx3
172.2x.1.0/30
2001:DB8:2x:CAFE::/64
Fe0/0.2x02
.1
F0/1.2x03
.2
F0/1.2x07
.1
Grupo x
AS6450x
172.2x.0.0/20
2001:DB8:2x::/48
Vlan 2x02
172.2x.2.0/30
2001:DB8:2x:FACA::/64
Vlan 2x03
172.2x.3.0/30
2001:DB8:2x:DAD0::/64
Vlan 2x07
172.2x.10.0/28
2001:DB8:2x:10::/112
eth1
.1
eth0
.2
eth0
.2
Vlan 2x04
Rx2
Sx2
Sx1
eth0
.2
172.2x.4.0/30
2001:db8:2x:DEAD::/64
eth2
.1
eth3
.1
Vlan 2x08
172.2x.5.0/30
2001:DB8:2x:CA5A::/112
Grupo x
AS6452x
10.2x.0.0/20
2001:DB8:FF2x::/48
eth0
.2
Rx4
eth1
.1
Vlan 2x09
10.2x.1.0/28
2001:DB8:FF2x:1::/64
eth0
.2
Sx3
eth2
.x
Vlan 3000
10.10.10.0/24
2001:DB8:1010:10::/64
PTTMetro
Looking Glass
AS65531
Vlan 3000
10.10.10.0/24
2001:DB8:1010:10::/64
SW-PTT
Vlan 3000
10.10.10.0/24
2001:DB8:1010:10::/64
RS-PTT
AS65530
10.10.10.0/24
2001:DB8:1010:10::/64
eth0
.31
eth0
.30

Laboratório de IPv6
Núcleo



S01		
Interface	IPv4	IPv6
eth0	10.3.13.2/24	2001:DB8:300:13::2/64

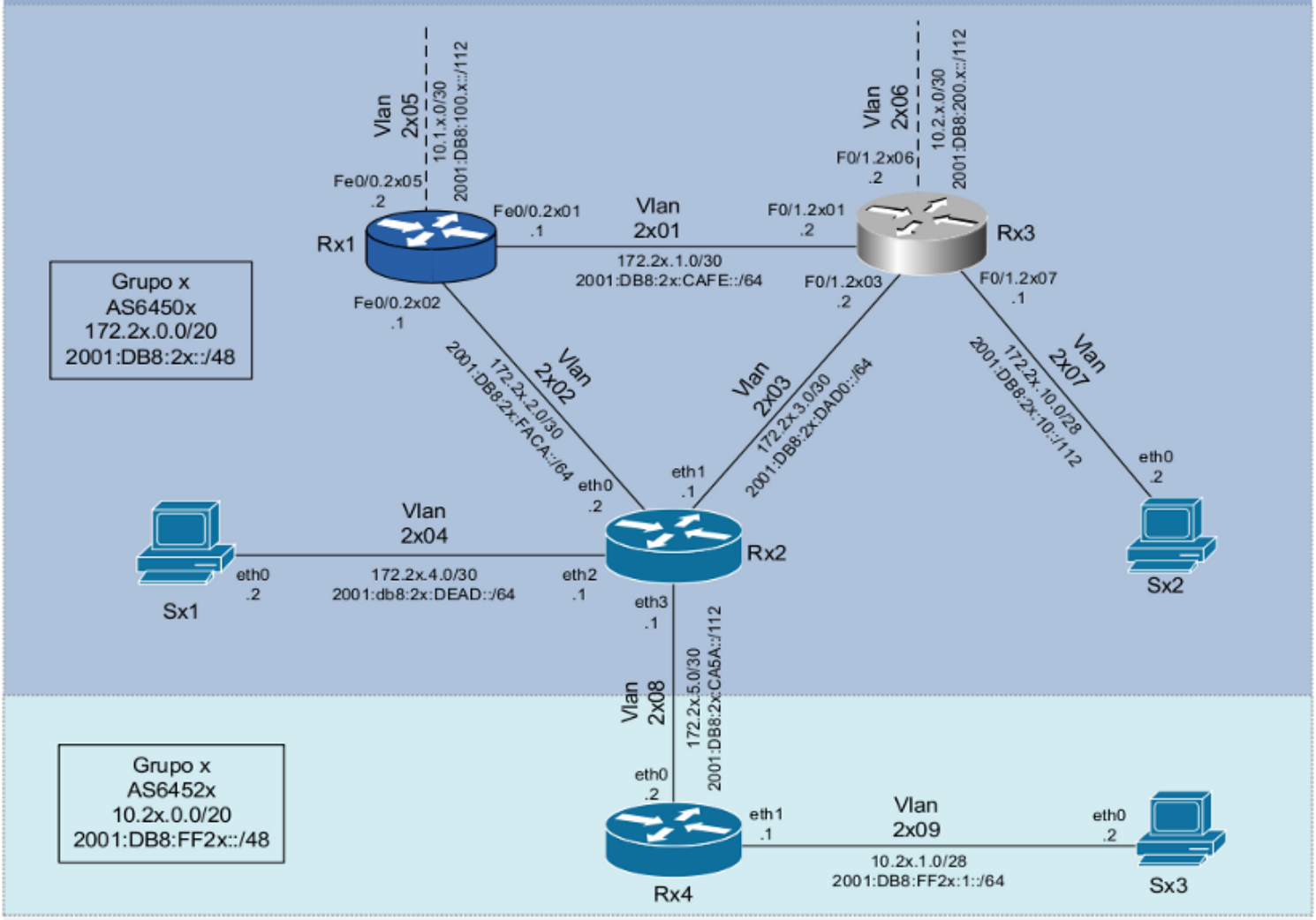
S02		
Interface	IPv4	IPv6
eth0	10.3.200.2/24	2001:DB8:300:F200::2/64

R01		
Interface	IPv4	IPv6
eth0	10.3.12.1/30	2001:DB8:300:12::1/64
eth1	10.3.11.1/30	2001:DB8:300:11::1/64
eth2	10.3.13.1/30	2001:DB8:300:13::1/64
eth3	10.3.200.1/24	2001:DB8:300:F200::1/64
lo	10.3.255.255/32	2001:DB8:300:FFFF::255/128

R02		
Interface	IPv4	IPv6
eth0	10.3.11.2/30	2001:DB8:300:11::2/64
eth1	10.1.11.1/30	2001:DB8:100:11::1/64
ethx	10.1.x.1/30	2001:DB8:100:x::1/112
lo	10.1.255.255/32	2001:DB8:100:FFFF::255/128

R03		
Interface	IPv4	IPv6
eth0	10.1.11.2/30	2001:DB8:100:11::2/64
eth1	10.3.12.2/30	2001:DB8:300:12::2/64
ethx	10.2.x.1/30	2001:DB8:200:x::1/112
lo	10.2.255.255/32	2001:DB8:200:FFFF::255/128

Laboratório de IPv6
Roteamento com AS



Sx1		
Interface	IPv4	IPv6
eth0	172.2x.4.2/30	2001:DB8:2x:DEAD::2/64

Sx2		
Interface	IPv4	IPv6
eth0	172.2x.10.2/28	2001:DB8:2x:10::2/112

Sx3		
Interface	IPv4	IPv6
eth0	10.2x.1.2/28	2001:DB8:FF2x:1::2/64

Rx1		
Interface	IPv4	IPv6
Fe0/0.2x01	172.2x.1.1/30	2001:DB8:2x:CAFE::1/64
Fe0/0.2x02	172.2x.2.1/30	2001:DB8:2x:FACA::1/64
Fe0/0.2x05	10.1.x.2/30	2001:DB8:100:x::2/112
lo0	172.2x.15.255/32	2001:DB8:2x:FFFF::255/128

Rx4		
Interface	IPv4	IPv6
eth0	172.2x.5.2/30	2001:DB8:2x:CA5A::2/112
eth1	10.2x.1.1/28	2001:DB8:FF2x:1::1/64
lo	10.2x.15.255/32	2001:DB8:FF2x:FFFF::255/128

Rx2			
Interface	IPv4	IPv6	Obs.
eth0	172.2x.2.2/30	2001:DB8:2x:FACA::2/64	
eth1	172.2x.3.1/30	2001:DB8:2x:DAD0::1/64	
eth2	172.2x.4.1/30	2001:DB8:2x:DEAD::1/64	
eth3	172.2x.5.1/30	2001:DB8:2x:CA5A::1/112	
lo	172.2x.15.254/32	2001:DB8:2x:FFFF::254/128	iBGP
lo	172.2x.15.250/32	2001:DB8:2x:FFFF::250/128	eBGP

Rx3			
Interface	IPv4	IPv6	Obs.
F0/1.2x01	172.2x.1.2/30	2001:DB8:2x:CAFE::2/64	
F0/1.2x03	172.2x.3.2/30	2001:DB8:2x:DAD0::2/64	
F0/1.2x06	10.2.x.2/30	2001:DB8:200:x::2/112	
F0/1.2x07	172.2x.10.1/28	2001:DB8:2x:10::1/112	
loopback10	172.2x.15.253/32	2001:DB8:2x:FFFF::253/128	Router ID
loopback20	172.2x.15.252/32	2001:DB8:2x:FFFF::252/128	iBGP
loopback30	172.2x.15.251/32	2001:DB8:2x:FFFF::251/128	eBGP

Laboratorio – DNS

Objetivo: Utilizando BIND9, configurar un servidor DNS responsable de responder las consultas realizadas al dominio de primer nivel .gx (la letra 'x' representa el número de grupo). También configuraremos los servidores y routers el AS con
directivas A y AAAA, para que las consultas a sus dominios puedan regresar tanto direcciones IPv6 como direcciones IPv4.

Escenario inicial: En esta fase cada grupo representa un AS diferente que tiene conexión a 2 proveedores de tránsito y provee tránsito a un AS cliente.

Cada AS tiene acceso a un router Cisco, un router Linux/Quagga, un router Juniper y dos servidores Linux. La política de enrutamiento externo y el protocolo de enrutamiento interno (IGP), en este caso OSPF, ya están implementados tanto para IPv4 como para IPv6. El grupo debe probar la comunicación dentro del propio AS y con los demás AS (usar, por ejemplo, mtr, ping y traceroute IPv4 e IPv6).

Ejercicio 1: Configuración de los servidores

En este laboratorio cada AS será responsable por la administración de un dominio de primer nivel, de modo que el grupo 1 será responsable por el dominio .g1, el grupo 2 por el .g2 y así sucesivamente.

El servidor S01 ubicado en el AS64513 será nuestro servidor raíz. Este delegará al servidor DNS de nuestro AS la autoridad sobre el dominio .gx.

Nuestro servidor DNS será el Sx2. Este ya tiene instalado BIND9, por lo que podemos iniciar su configuración editando el archivo named.conf, donde indicaremos la zona a la cual responderá nuestro servidor y también la zona "." de tipo "hint", la raíz de Internet.

- En el servidor Sx2:
  Abra el archivo /etc/named.conf y reemplace su contenido por el siguiente:

```
// Default named.conf generated by install of bind-9.2.4-30.el4_7.2
options {
        directory "/var/named";
        dump-file "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
};
include "/etc/rndc.key";

zone "." {
  type hint;
  file "root.zone";
};

zone "gX" {
  type master;
  file "gX.zone";
};
```

Nota 1: Recuerde reemplazar la 'X' por su número de grupo.
Nota 2: descarge este script de la dirección http://[**********]/named.conf.txt

Nosotros crearemos un domino para cada servidor y para cada router Linux. Para ello editaremos el archivo gX.zone, donde agregaremos la configuración de cada uno.

- En el servidor Sx2:
  Abra el archivo /var/named/gX.zone y reemplace su contenido por el siguiente:

```
$TTL 86400

@       IN      SOA gX. labnic.a.gX. (
                        10 ; serial
                        28800 ; refresh
                        7200 ; retry
                        604800 ; expire
                        86400 ; ttl
                        )
```

```
        IN      NS       ns.gX.

ns.gX.  IN       A       172.2X.10.2
ns.gX.  IN    AAAA       2001:db8:2X:10::2

sX1     IN       A        172.2X.4.2
sX1     IN    AAAA        2001:db8:2X:dead::2

sX2     IN       A       172.2X.10.2
sX2     IN    AAAA       2001:db8:2X:10::2

rX2     IN       A        172.2X.15.254
rX2     IN    AAAA        2001:db8:2X:ffff::254

rX2     IN       A       172.2X.2.1
rX2     IN    AAAA       2001:db8:2X:faca::1

rX2     IN       A       172.2X.3.1
rX2     IN    AAAA       2001:db8:2X:dad0::1

rX2     IN       A       172.2X.4.1
rX2     IN    AAAA       2001:db8:2X:dead::1
```

Nota 1: Recuerde reemplazar la 'X' por su número de grupo.
Nota 2: descargue este script de la dirección http://[**********]/gX.zone.txt

Vamos a indicar a nuestro servidor DNS la dirección del servidor raíz agregando al archivo root.zone la siguiente información:

- En el servidor Sx2:
  Abra el archivo /var/named/root.zone y reemplace su contenido por el siguiente:

```
.       3600000   IN NS  a.g0.
a.g0.   3600000   A      10.3.13.2
a.g0.   3600000   AAAA   2001:db8:300:13::2
```

Con estas configuraciones cada uno de estos dispositivos debería responder por sus respectivos dominios: sX1.gX, sX2.gX y rX2.gX.

Todavía en el servidor Sx2, reinicie el servicio BIND para que las configuraciones tomen efecto.

```
#/etc/init.d/named restart
```

Ahora en cada uno de ellos vamos a agregar el archivo resolv.conf, donde indicaremos que el servidor Sx2 es el servidor DNS de nuestro AS.

- En los servidores Sx1 y Sx2 y en el router Rx2:

Abra el archivo /etc/resolv.conf y reemplace su contenido por el siguiente:

```
nameserver 172.2X.10.2
```

Con las configuraciones realizadas hasta este momento ya estamos en condiciones de probar si el servicio de DNS está funcionado dentro de nuestro AS. Utilice comandos como ping, nslookup y dig para verificar si la consulta por nome está siendo correctamente traducida a la dirección IP correspondiente. Realice estas consultas tanto para direcciones IPv6 como para direcciones IPv4.

```
[root@SX2 /]# ping sX1.gX
PING sX1.gX (172.2X.4.2) 56(84) bytes of data.
64 bytes from 172.2X.4.2: icmp_seq=0 ttl=64 time=2.25 ms
64 bytes from 172.2X.4.2: icmp_seq=1 ttl=61 time=0.412 ms
...

[root@SX2 /]# ping6 sX1.gX
PING sX1.gX(2001:db8:2X:dead::2) 56 data bytes
64 bytes from 2001:db8:2X:dead::2: icmp_seq=0 ttl=61 time=9.85 ms
64 bytes from 2001:db8:2X:dead::2: icmp_seq=1 ttl=61 time=0.396 ms
...

[root@SX2 /]# dig -t A rX2.gX

; <<>> DiG 9.2.4 <<>> -t A rX2.gX
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1890
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 1, ADDITIONAL: 2


;; QUESTION SECTION:
;rX2.gX.                    IN    A


;; ANSWER SECTION:
rX2.gX.               86400 IN    A     172.2X.3.1
rX2.gX.               86400 IN    A     172.2X.4.1
rX2.gX.               86400 IN    A     172.2X.15.254
rX2.gX.               86400 IN    A     172.2X.2.1

;; AUTHORITY SECTION:
gX.             86400 IN    NS    ns.gX.

;; ADDITIONAL SECTION:
ns.gX.                86400 IN    A     172.2X.10.2
ns.gX.                86400 IN    AAAA  2001:db8:2X:10::2

;; Query time: 0 msec
;; SERVER: 172.2X.10.2#53(172.2X.10.2)
;; WHEN: Tue Aug 11 15:25:26 2009
;; MSG SIZE  rcvd: 149
[root@SX2 /]#
[root@SX2 /]#
[root@SX2 /]#
```

```
[root@SX2 /]# dig -t AAAA rX2.gX

; <<>> DiG 9.2.4 <<>> -t AAAA rX2.gX
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 9724
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 1, ADDITIONAL: 2

;; QUESTION SECTION:
;rX2.gX.                    IN    AAAA

;; ANSWER SECTION:
rX2.gX.               86400 IN    AAAA  2001:db8:2X:faca::1
rX2.gX.               86400 IN    AAAA  2001:db8:2X:ffff::254
rX2.gX.               86400 IN    AAAA  2001:db8:2X:dad0::1
rX2.gX.               86400 IN    AAAA  2001:db8:2X:dead::1

;; AUTHORITY SECTION:
gX.              86400 IN    NS    ns.gX.

;; ADDITIONAL SECTION:
ns.gX.                86400 IN    A     172.2X.10.2
ns.gX.                86400 IN    AAAA  2001:db8:2X:10::2

;; Query time: 0 msec
;; SERVER: 172.2X.10.2#53(172.2X.10.2)
;; WHEN: Tue Aug 11 15:25:31 2009
;; MSG SIZE  rcvd: 197
```

Verifique con los otros grupos si ya han finalizados estas tareas y pruebe la conectividad con los otros AS a través de los dominios registrados.

Ejemplo:

```
ping6 s21.g2
dig -t A r42.g4
dig -t AAAA r42.g4
nslookup s51.g5
```

Puede analizar las consultas DNS realizadas al servidor Sx2 utilizando el comando tcpdump.

- En el servidor Sx2:

```
[root@SX2 /]# tcpdump -vv -s 0
```

Realice consultas a los servidores de los AS vecinos desde el servidor Sx1, tanto para direcciones IPv6 como para direcciones IPv4. Observe en la salida del comando tcpdump que ambas consultas se realizan mediante conexión IPv4. Modifique el archivo resolv.conf del servidor Sx1, agregando la dirección IPv6 del servidor DNS y vuelva a realizar esta prueba. ¿Hay algún cambio en los datos obtenidos en las dos consultas?

Ejercicio 2: Resolución reversa

En el servidor DNS Sx2, agregue al archivo `/etc/named.conf` el siguiente contenido:

```
zone "x.2.0.0.8.b.d.0.1.0.0.2.ip6.arpa" {
  type master;
  file "reverso_ipv6.db";
};
```

Nota: Recuerde reemplazar la 'X' por su número de grupo.

De este modo le indicamos al servidor por cuál zona deberá responder autoritativamente.

A continuación cree el archivo `/var/named/reverso_ipv6.db`. Este archivo contendrá los datos de esa zona. Debe tener el mismo nombre indicado en la configuración general del servidor.

El archivo deberá contener la siguiente información:

```
;2001:db8:2x::/48
;
X.2.0.0.8.b.d.0.1.0.0.2.ip6.arpa.       IN      SOA     ns.gX       labnic.ns.gX.   (
                200912100       ; Serial number
                24h             ; Refresh time
                30m             ; Retry time
                2d              ; Expire time
                3d              ; Default TTL
)

X.2.0.0.8.b.d.0.1.0.0.2.ip6.arpa.       IN      NS      ns.gX.

;Subnet #1
$ORIGIN        d.a.e.d.X.2.0.0.8.b.d.0.1.0.0.2.ip6.arpa.

2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0         IN      PTR     sX1.gX.
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0         IN      PTR     rX2.gX.

;Subnet #2
$ORIGIN        a.c.a.f.X.2.0.0.8.b.d.0.1.0.0.2.ip6.arpa.

2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0         IN      PTR     rX2.gX.


;Subnet #3
$ORIGIN        0.d.a.d.X.2.0.0.8.b.d.0.1.0.0.2.ip6.arpa.

1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0         IN      PTR     rX2.gX.


;Subnet #4
$ORIGIN        0.1.0.0.X.2.0.0.8.b.d.0.1.0.0.2.ip6.arpa.

2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0         IN      PTR     sX2.gX.
```

Nota: Recuerde reemplazar la 'X' por su número de grupo.

Las líneas anterioes indican el nombre de la zona que se está configurando y sus RR (*Resources Records*), como por ejemplo el SOA (*Start of Authority*) que indica el servidor autoritativo para esa zona.

Las informaciones entre paréntesis tienen por objetivo organizar la sincronización con servidores secundarios (*slaves*). Luego hay información de los servidores DNS para la zona, que pueden ser más de uno. Por último está la información más interesante en el caso de la resolución reversa, que es la que indica el nombre asociado a cada dirección IP del bloque. El *Resource Record* PTR indica una relación entre la dirección y el nombre

correspondiente. Por ejemplo, la dirección 2001:0DB8:002X:0010:0000:0000:0000:0002, que también se puede representar como 2001:DB8:2X:10::1, tiene asociado el nombre sX2.gX.

Con las configuraciones realizadas hasta este momento ya estamos en condiciones de probar si la resolución reversa está funcionado dentro de nuestro AS. Utilice comandos como `nslookup` y `host` para verificar si la consulta por dirección IPv6 está siendo correctamente traducida al nombre correspondiente.

Ejemplo:

```
[root@SX1 /]# host 2001:db8:2X:dead::1

1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.d.a.e.d.X.2.0.0.8.b.d.0.1.0.0.2.ip6.arpa
domain name pointer rX2.gX.
```

Ejemplo:

```
[root@RX1 /]# nslookup 2001:db8:2X:dead::2

Server:           172.2X.10.2

Address:    172.2X.10.2#53



2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.d.a.e.d.X.2.0.0.8.b.d.0.1.0.0.2.ip6.arpa
      name = sX1.gX.
```

Referencias:

http://www.ceptro.br/pub/CEPTRO/MenuCEPTROPalestrasPapers/DNS.pdf
http://www.lacnic.net/pt/registro/dns/configuracion_ipv6.html
http://www.fccn.pt/files/documents/D2.06.1.PDF
http://tools.ietf.org/html/rfc3363
http://tools.ietf.org/html/rfc3596
http://tools.ietf.org/html/rfc4472