Publicação

ceptro.br

IPv6.br

Curso IPv6 básico

Laboratório: Familiarizando-se com o IPv6

cgi.br nic.br

Laboratório – Familiarizando-se com o IPv6

Objetivo: Familiarizarmos com as novas características do protocolo IPv6, configurando-o em nossos notebooks e realizando os primeiros exercícios do laboratório, como configuração manual de endereços e rotas. Também analisaremos a estrutura do protocolo IPv6 através da captura de pacotes com o programa Wireshark, onde poderemos observar melhor os tópicos aprendidos durante a aula teórica.

Exercício - 1: Instalação de aplicativos

Para realizarmos os exercícios propostos neste laboratório, será necessário a instalação de alguns aplicativos em nossos nossos noteboks, como:

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

Você pode consultar seu buscador preferido.

Exercício 0: Configuração nativa de IPv6 em seu notebook.

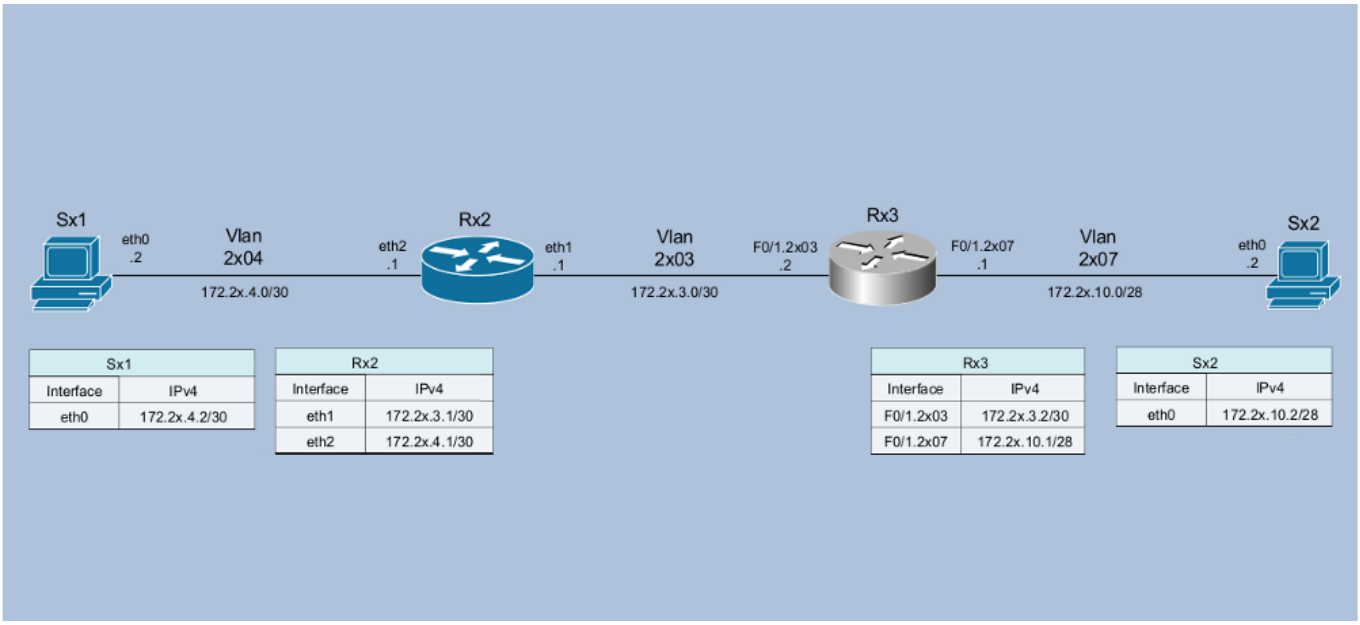
Consulte no site http://ipv6.br o artigo "Habilitando IPv6 em Sistemas Operacionais". Nele você encontrará como configurar o IPv6 nos principais Sistemas Operacionais.

Configure...

Acesse sites ipv6, faça pings e traceroutes em IPv6.

Dê um traceroute para www.google.com.br e para www6.terra.com.br

Exercício 1: Acesso e configurações de rede.



A partir de agora, todos os exercícios do laboratório seguirão o seguinte cenário:

- A turma será dividida em grupos, onde cada grupo representara um AS. Inicialmente esse AS será composto por dois servidores e dois roteadores, um Cisco e um Linux/Quagga.
- Para acessar o roteador CISCO do laboratório, é necessário fazer um ssh para o endereço xxx.xxx.xxx.xxx, ou xxxx:xxxx:x:xxxx::xxx, na porta 3443, com usuário “labnicX” e senha “labgrupoX” (sem as aspas), onde X representa o número do 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:~$
```

Com isso você estará logado no servidor de administração de nosso laboratório. À partir dele, você poderá acessar os servidores e roteadores disponíveis. O usuário labnicX dá acesso à uma sessão limitada, onde é possível executar apenas os comandos para acessar os componentes do laboratório:

```
labnicX:~$ help
console  exit  help  juniper  router  rx1  rx2  rx3  rx4  server  sx1  sx2  sx3
labnicX:~$
```

Os comandos “server x1” e “server x2” dão acesso aos servidores, onde x é o número do grupo. O comando “router x3” dá acesso ao roteador Cisco: use o usuário “cisco” e senha “cisco”. Por fim, o comando “router x2” dá acesso ao roteador Linux/Quagga. Use a senha “labgrupoX” para os servidores e roteadores Linux/Quagga.

O grupo deve testar o acesso a todos eles.

```
labnicX:~$
labnicX:~$ server x1 (troque o x pelo número do seu grupo)
Senha: labgrupoX
entered into CT 110
[root@SX1 /]# exit
logout
exited from CT 110

labnicX:~$
labnicX:~$ router x2 (troque o x pelo número do seu grupo)
entered into CT 112
[root@RX2 /]# exit
logout
exited from CT 112

labnicX:~$
labnicX:~$ router x3 (troque o x pelo número do seu 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 (troque o x pelo número do seu grupo)
entered into CT 114
[root@SX2 /]# exit
logout
exited from CT 114
labnicX:~$
```

Nessa fase do laboratório, apenas o IPv4 está configurado, embora todos os equipamentos sejam capazes de executar IPv6. Não são usados protocolos de roteamento inicialmente, apenas rotas estáticas. Verifique as configurações de rede e execute testes de conectividade entre todos os elementos do laboratório. Use, por exemplo, comandos como “ip”, “ping”, “traceroute”, “mtr”, etc. Procure entender como estão configurados os equipamentos e as rotas.

Se não houver conectividade entre todos os elementos, avise aos instrutores e tente descobrir onde está o problema e resolvê-lo.

Exercício 2: Captura e análise de pacotes.

Para a captura de pacotes nos servidores e roteadores Linux, será utilizado o comando “tcpdump”. Para as análises será utilizado o programa “Wireshark”, previamente instalado nos notebook dos participantes do curso.

Para nos habituarmos ao uso das ferramentas, vamos monitorar o tráfego na interface eth2, do roteador Rx2, e executar um traceroute, de Sx1 para Sx2.

- No router Rx2:

```
labnicX:~$ router x2
Senha:
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
```

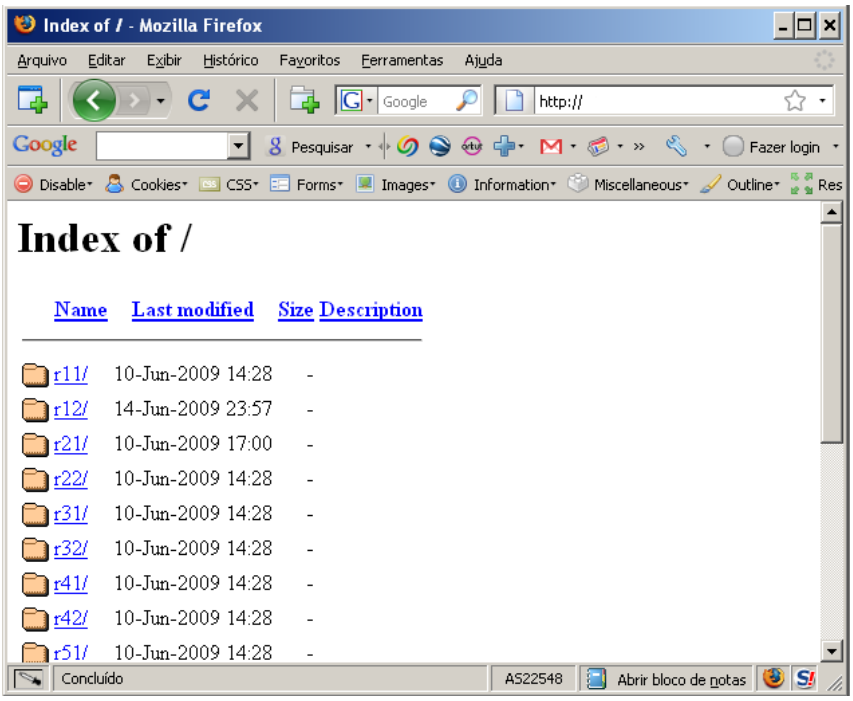
- No servidor Sx1:

```
labnicX:~$ server x1
Senha:
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 /]#
```

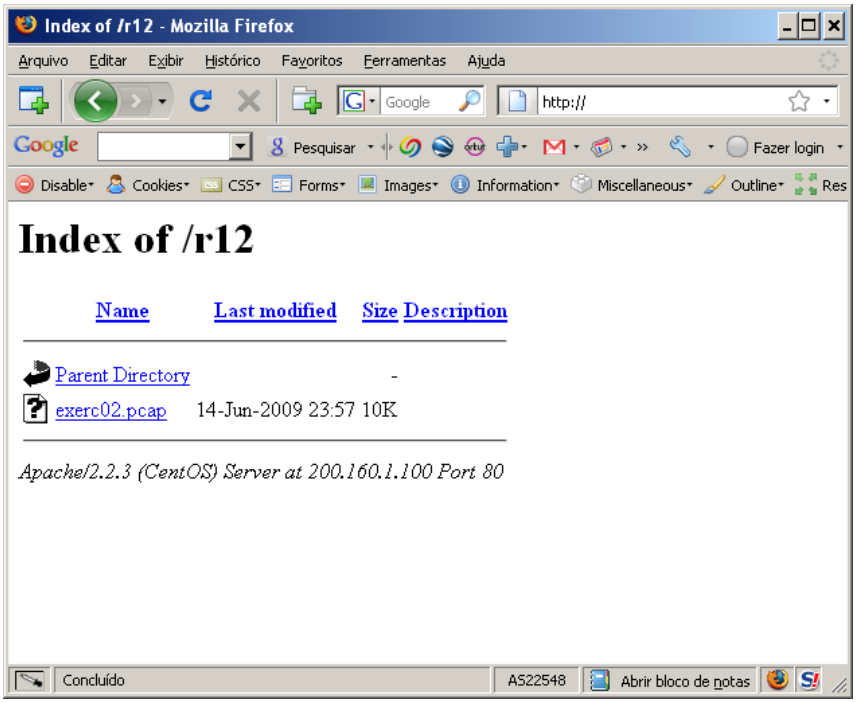
- Novamente, no router Rx2:

```
[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 /]#
```

Em cerca de 1 minuto, no máximo, um script copiará este arquivo para um diretório compartilhado via Web em nosso servidor de gerenciamento. Aguarde alguns instantes e, usando um navegador em seu notebook, acesse o endereço http://[xxxx:xxxx:x:xxxx::xxx]/captura/ (o mesmo utilizado com o ssh).



Entre na pasta correspondente ao router x2 e salve o arquivo exerc02.pcap em seu notebook.

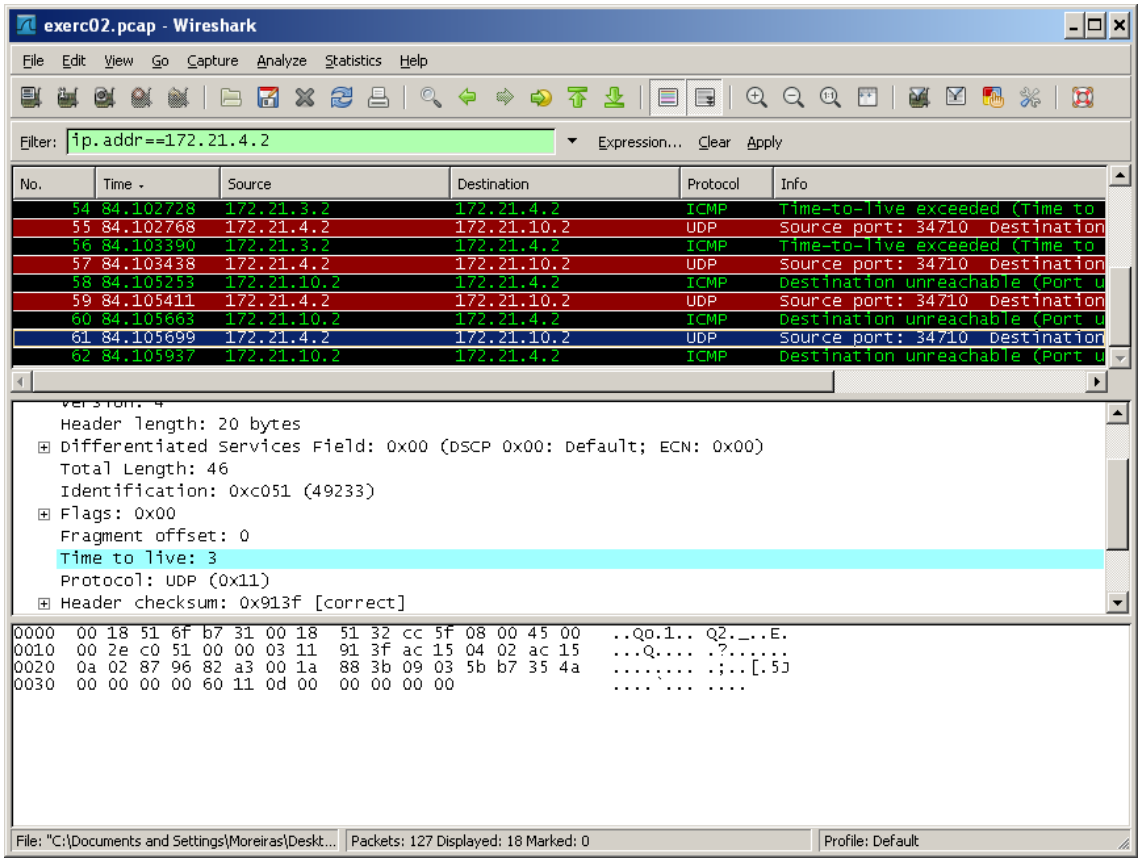


Abra o arquivo no Wireshark.

Aplique o filtro ip.addr==''endereço de origem do traceroute", se quiser, para facilitar a visualização, e responda as seguintes 2 questões:

1 – Qual protocolo é utilizado para o envio das mensagens pela origem?

2 – Quantos pacotes são enviados para cada valor de TTL?


exerc02.pcap - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: ip.addr==172.21.4.2
Expression... Clear Apply
No. Time Source Destination Protocol Info
54 84.102728 172.21.3.2 172.21.4.2 ICMP Time-to-live exceeded (Time to
55 84.102768 172.21.4.2 172.21.10.2 UDP Source port: 34710 Destination
56 84.103390 172.21.3.2 172.21.4.2 ICMP Time-to-live exceeded (Time to
57 84.103438 172.21.4.2 172.21.10.2 UDP Source port: 34710 Destination
58 84.105253 172.21.10.2 172.21.4.2 ICMP Destination unreachable (Port u
59 84.105411 172.21.4.2 172.21.10.2 UDP Source port: 34710 Destination
60 84.105663 172.21.10.2 172.21.4.2 ICMP Destination unreachable (Port u
61 84.105699 172.21.4.2 172.21.10.2 UDP Source port: 34710 Destination
62 84.105937 172.21.10.2 172.21.4.2 ICMP Destination unreachable (Port u
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
Total Length: 46
Identification: 0xc051 (49233)
Flags: 0x00
Fragment offset: 0
Time to live: 3
Protocol: UDP (0x11)
Header checksum: 0x913f [correct]
0000 00 18 51 6f b7 31 00 18 51 32 cc 5f 08 00 45 00 ..Qo.1.. Q2._..E.
0010 00 2e c0 51 00 00 03 11 91 3f ac 15 04 02 ac 15 ...Q.... .?......
0020 0a 02 87 96 82 a3 00 1a 88 3b 09 03 5b b7 35 4a ....;..[.5J
0030 00 00 00 00 60 11 0d 00 00 00 00 00 ....`... ....
File: "C:\Documents and Settings\Moreiras\Deskt... Packets: 127 Displayed: 18 Marked: 0 Profile: Default

Exercício 3: IPv6 – endereços locais.

Habilite o IPv6 nos equipamentos e verifique que, mesmo que os endereços IPv6 ainda não tenha sido configurados, já há endereços do tipo "link-local" em cada um deles. Pode ser que para alguns dos equipamentos, o IPv6 já esteja habilitado.

Exemplo, no 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
```

Nesse exemplo, o IPv6 já está habilitado (nas máquinas do laboratório, o IPv6 foi incluído no kernel; uma alternativa seria usá-lo como um módulo, carregado ou não por padrão). No caso do ipv6 ter sido compilado como módulo e não ser carregado por padrão, é necessário usar o comando "modprobe ipv6".

Exemplo, no Cisco:

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

Nesse caso, o IPv6 não está habilitado. É necessário habilitá-lo:

```
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: Conexao-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 com os endereços “link-local” já há conectividade IPv6 para cada segmento de rede. E que não há conectividade entre diferentes seguimentos.

Exemplo:

```
[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
```

Suas saídas se parecem mais com esta (abaixo)? O que está faltando no comando?

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

Descubra o endereço físico (MAC) de cada interface e responda a seguinte questão: Como os endereços “link-local” são formados à partir dos endereços físicos?

Exercício 4: IPv6 – analisando o cabeçalho dos pacotes.

Neste exercício, vamos analisar o cabeçalho do protocolo IPv6, e tentar descobrir algumas diferenças em relação ao IPv4.

Vamos capturar os pacotes de pings, v4 e v6, enviados de Sx1 para Rx2, no roteador Rx2.

- No 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
```

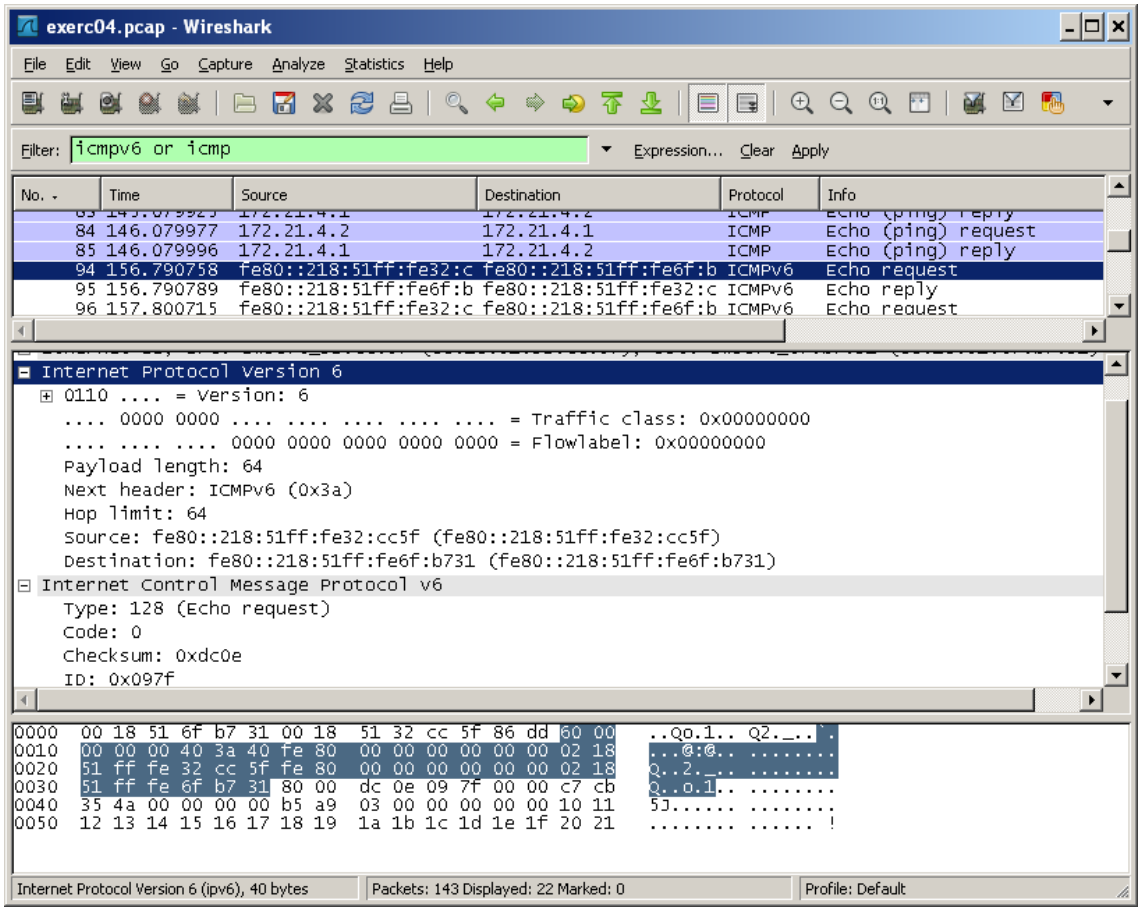
- No servidor Sx1:

```
[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 o arquivo no Wireshark, em seu notebook.
Você pode utilizar o seguinte filtro, para facilitar a visualização: “icmp or icmpv6”



Compare os pacotes IPv4 e IPv6... Identifique cada um dos campos do cabeçalho IP nos dois casos, e observe seus valores. Compare também o ICMP. Responda às seguintes questões?

- Qual a diferença de tamanho entre o cabeçalho IPv4 e o cabeçalho IPv6?
- Há diferenças também no cabeçalho ICMP? Quais?

Exercício 5: IPv6 – cabeçalhos de extensão.
Neste exercício, vamos verificar a existência dos cabeçalhos de extensão, gerando a necessidade de fragmentação no comando ping.

Vamos capturar os pacotes de pings, v4 e v6, enviados de Sx1 para Rx2, no roteador Rx2, como no exercício anterior. Mas vamos especificar 2000 bytes para o tamanho do pacote.

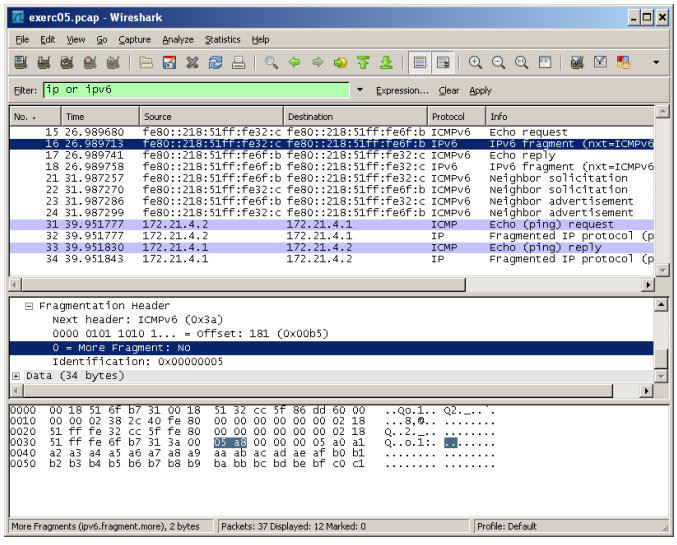
- No router Rx2:

```
[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
```

- No 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 /]#
```

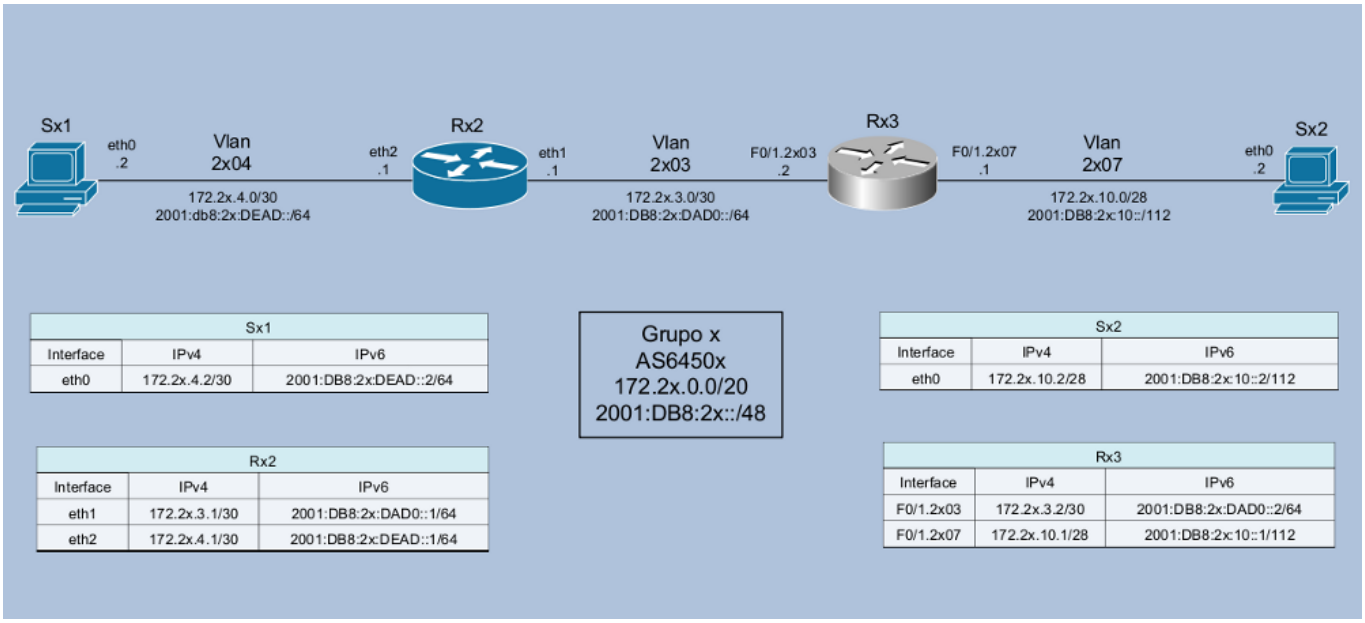
Abra o arquivo no Wireshark, e utilize o filtro “ipv6 or ip” para facilitar a visualização.



- Verifique a existência do cabeçalho de fragmentação e responda às seguintes questões?
- Qual a diferença entre o processo de fragmentação no IPv4 e no IPv6?
- Qual o tamanho do cabeçalho de extensão (fragmentação)?
- Qual a diferença do valor do campo Next Header no cabeçalho v6 do exercício 04 para este exercício?

Exercício 6a: IPv6 – configuração manual dos endereços.

Neste exercício, vamos configurar os endereços de nosso bloco (2001:db8:2x::/48), conforme a figura à seguir.



- No 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
```

- No 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
```

- No roteador 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
```

- No roteador 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: Conexao-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: Conexao-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 agora há conectividade local, através dos novos endereços, válidos globalmente, bem como dos pré-existentes endereços “link-local”. Não há conectividade entre as diferentes redes, no entanto, porque as rotas ainda não foram configuradas.

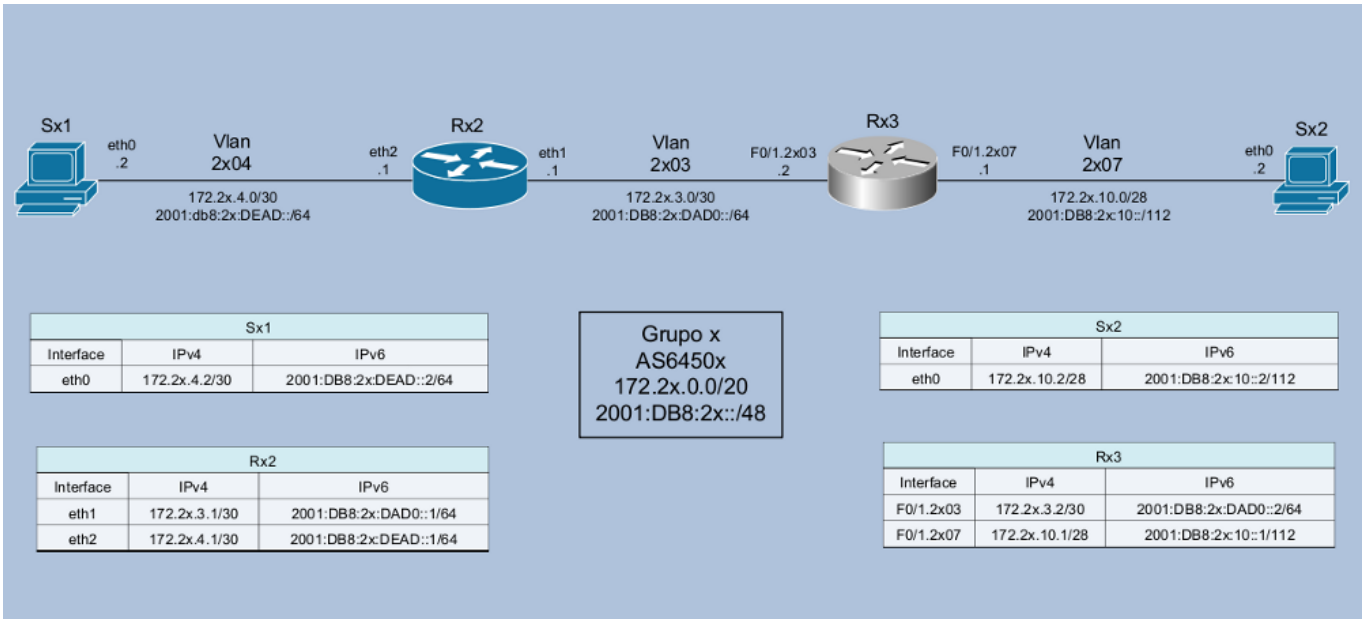
Veja que é possível adicionar outros endereços IPv6 às interfaces.

Experimente, por exemplo, adicionar novos endereços à 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
```

Exercício 6b: IPv6 – configuração manual das rotas.
Neste exercício, vamos configurar as rotas, manualmente, de forma a ter conectividade v4 e v6 em nosso laboratório.



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

Grupo x
AS6450x
172.2x.0.0/20
2001:DB8:2x::/48

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

Procure observar a configuração IPv4 e "copiá-la" para o contexto IPv6, antes de consultar os exemplos que seguem.

Exs:

- 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
```

Após as configurações, teste a conectividade ponta a ponta, com ping6 de Sx1 para Sx2.

Ex:

```
[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 ~]#
```

Exercício 7: IPv6 – Neighbour Discovery.

Neste exercício, vamos observar o funcionamento do protocolo Neighbour Discovery.

Vamos, primeiramente, limpar a tabela do Neighbour Discovery para a interface eth2 do roteador Rx2:

Verificação da tabela de vizinhos do IPv6

- No Linux:

```
[root@RX2]# ip -6 neighbor (esta tabela e similar a tabela ARP do IPv4)
```

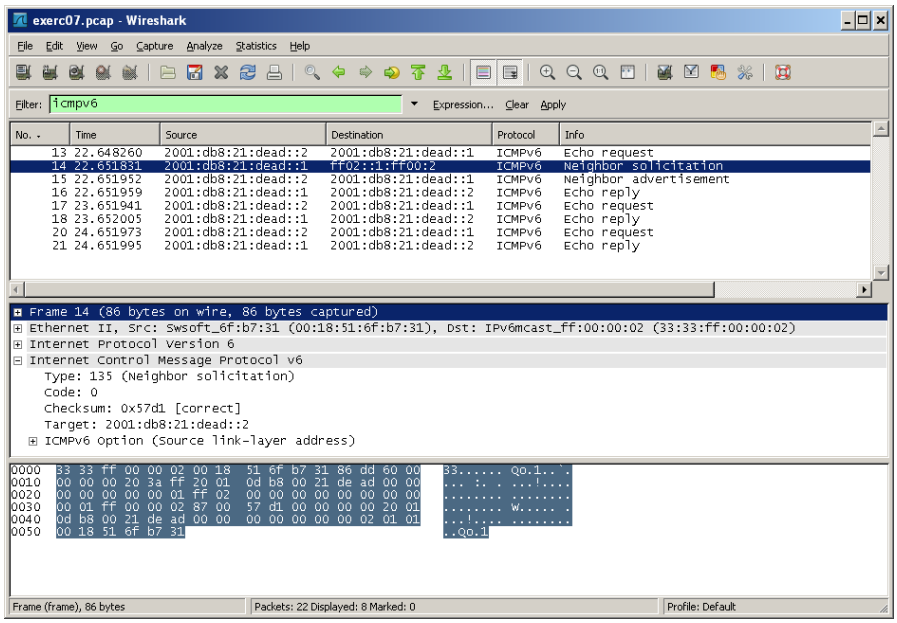
```
[root@RX2 ~]# ip neighbor flush dev eth2.
```

Vamos começar agora a capturar os pacotes, e pingar a interface à partir de 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 o arquivo no Wireshark e use o filtro “icmpv6”. Verifique a existência das mensagens Neighbour Solicitation e Neighbour Advertisement.



Exercício 8: IPv6 – Path MTU Discovery.

Neste exercício, vamos observar o funcionamento do protocolo Path MTU Discovery.
Primeiramente, vamos diminuir "artificialmente" o MTU de uma das interface eth1 de Rx2, e ativar o tcpdump na 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
```

Vamos, então, executar um ping, de Sx1 para Sx2, com tamanho de pacote maior que o MTU especificado, e observar o que ocorre.

```
[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
```

Paramos então a captura e analisamos os dados no Wireshark. Procure identificar a mensagem icmp de "Packet too big" e observar que tipo de informação extra ela traz. Responda à seguinte pergunta: qual o protocolo dessa mensagem, e o que você acha que ocorre se ela for bloqueada por um firewall?

Publicação
ceptro.br

IPv6.br

Curso IPv6 básico

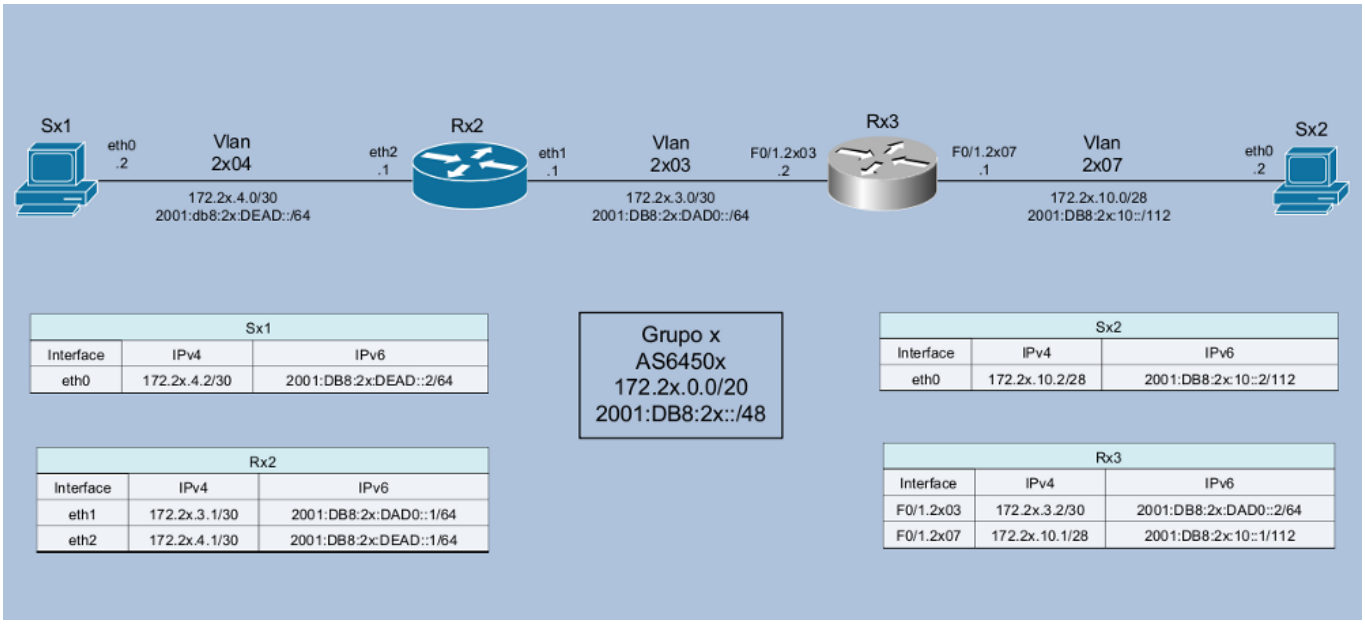
Laboratório: Firewall IPv6

cgi.br nic.br

Laboratório – Firewall IPv6

Objetivo: Implementar um firewall simples nos servidores do AS, com suporte nativo a IPv6, utilizando ip6tables.

Cenário inicial: Cada AS possui acesso a um roteador Cisco, um roteador Linux/Quagga, e dois servidores Linux. Não há políticas de roteamento externo ou protocolo de roteamento interno (IGP) implementados, nem para IPv4 nem IPv6. Há apenas as configurações de endereçamento estático IPv4 e IPv6. O grupo deve testar a comunicação dentro do próprio AS (use mtr, ping e traceroute IPv4 e IPv6, por exemplo).



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

Exercício 1 - Configurando ip6tables.

É preciso ter uma atenção maior na utilização de firewall's em redes IPv6, visto que, ao contrário da maioria das redes IPv4, a rede interna não é mais “protegida” pela utilização de endereços IP privados (RFC 1918). Com a adoção do protocolo IPv6 todos os hosts podem utilizar endereços válidos com conectividade direta a Internet e alcance a todos os hosts da rede interna que tenham IPv6 habilitado.

Vamos utilizar o seguinte script com regras para o ip6tables.

```
#!/bin/sh

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

# caminho do iptables
iptables="/sbin/ip6tables"

# Meus IPs
# Acrescentar os IPs v6 do servidor aqui
ips_locais="2001:DB8:XX:DEAD::2/128 FE80::XXXX:XXFF:FEXX:XXXX/128 FF02::1:FF00:0/104
FF02::1/128"

start () {
        echo "Iniciando o filtro de pacotes: ip6tables..."

        # A politica padrao eh recusar todos os pacotes
        echo "Configurando a politica padrao para recusar todos os pacotes"
        $iptables -F
        $iptables -P INPUT DROP
        $iptables -P OUTPUT DROP
        $iptables -P FORWARD DROP

        # Permitir trafego ilimitado para o localhost
        echo "Permitindo trafego ilimitado para o localhost"
        $iptables -A INPUT  -i lo -j ACCEPT
        $iptables -A OUTPUT -o lo -j ACCEPT

        # Conexoes permitidas de entrada e saida para este servidor
        for ip in $ips_locais
          do
            echo -n "Permitindo algumas conexoes de entrada para o este servidor (IP $ip)..."

            # Abrindo o 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

            # Trafego 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

            # Permitindo 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

            # Permitindo o envio de mensagens ICMPv6
            echo -n "icmp out "
            $iptables -A OUTPUT -p icmpv6 -s $ip -j ACCEPT

            ###### RFC 4890 #####
            ###### Trafego ICMPv6 que NAO DEVE ser DESCARTADO ######
            echo -n "icmp in "
            # ECHO REQUESTS E RESPONSES (Type 128 e 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

            ##### Trafego ICMPv6 que NORMALMENTE NAO DEVE ser DESCARTADO #####
            # Mobilidade IPv6 ### Apenas as habilite se o noh for um Noh Movel ###
            # ===============
```

```
            # 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 especificos ######
            ## Algumas mensagens não precisam de tratamento:
            #  - Router Renumbering (Type 138): Devem ser autenticadas com IPSec
            #
            #
            ## Algumas mensagens precisam de politicas especificas:
            #  - Redirect (Type 137): Podem oferecer riscos a segurança. Sua
            # utilização deve ser estudada caso a caso.
            #
            #
            ## Mensagens ainda nao definidas pela a IANA ou de uso experimental
            # devem ser sempre descartadas.
            ## A nao ser que exista um caso muito especifico na rede e que elas
            # sejam utilizadas.
           echo .

        done

        # Descartando tudo mais
        echo "Descartando todos os demais pacotes... "
        $iptables -A INPUT -s ::/0 -j DROP
        $iptables -A OUTPUT -d ::/0 -j DROP
}

stop () {
        echo "Parando o filtro de pacotes: 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 as regras e cadeias estao limpas."
        echo "Tome cuidado... Isso eh perigoso!!"
        echo "Execute: ** /etc/init.d/ip6tables start ** assim que possivel."
}

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
```

Faça o download desse script no endereço

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

Basicamente, as regras de firewall aqui aplicadas tem como política padrão descartar todos os tipos de pacotes, permitindo apenas o acesso ao nó via ssh na porta 22, acesso via http na porta 80, a rastreabilidade do nó via traceroute e a utilização das mensagens ICMPv6 de acordo com as recomendações da RFC 4890.

Agora vamos aplicar essas regras em um dos servidores do AS. No servidor Sx1, crie o arquivo ip6tables no diretório /etc/init.d/ e adicione o conteúdo do script baixado anteriormente.

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

Altere os endereços contidos na linha 10 do arquivo de acordo com a numeração do servidor, salve o arquivo e reinicie o serviço do ip6tables:

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

Exercício 2 - Testando as regras do firewall

Seu firewall IPv6 já deve estar funcionando normalmente. Agora vamos realizar alguns testes para analisarmos sua configuração e se há diferenças na definição das regras para o iptables (IPv4) e o ip6tables (IPv6).

A partir do servidor Sx2, acesse o servidor Sx1 via 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:
```

Acesse também o serviço de http do sevidor Sx1 utilizando o *browser* elinks:

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

Após a realização dos dois acessos, altere o script do ip6tables no servidor Sx1 para bloquear esses dois serviços, comentando as linhas correspondentes:

```
...
# Abrindo o 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


# Trafego 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 o serviço do ip6tables e tente realizar o acesso a esses serviços novamente, a partir do servidor Sx2.

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

Também testaremos a utilização do comando traceroute6 traçando a rota a partir do servidor Sx2 até o 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
```

Altere novamente o script do ip6tables no Sx1, comentando as linhas que permitem a utilização do comando traceroute:

```
...
# Permitindo 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 o serviço do ip6tables e tente traçar novamente a rota para o servidor Sx1 a partir do servidor Sx2.

Também podemos testar a conectividade IPv6 entre os dois servidores do As, através de pings de um servidor para ou 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
...
```

Vamos agora alterar o script do ip6tables do servidor Sx1, comentando as linhas que permitem o recebimento de mensagens ICMPv6 echo-request e echo-reply:

```
# ECHO REQUESTS E RESPONSES (Type 128 e 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 o serviço do ip6tables e realize novamente os testes de conectividade entre os servidores Sx1 e Sx2.

A RFC 4890 recomenda que não se bloqueie a utilização de pings, dizendo que está prática de segurança é desnecessária, visto que a realização de varredura de endereços em uma rede IPv6 é praticamente impossível. O que você acha desta recomendação? O bloqueio de pings é importante ou não?

Exercício 3 – Bloqueando mensagens ICMPv6

Vamos testar agora as regras que permitem o envio e o recebimento das mensagens ICMPv6 utilizadas pelo protocolo de Descoberta de Vizinhança.

Primeiro, habilite o serviço radvd no roteador Rx2, editando ou criando o arquivo /etc/radvd.conf com o seguinte conteúdo:

- No roteador 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 o Radvd

- No roteador Rx2:

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

Caso ocorra algum erro ao se iniciar o processo do Radvd, verifique o arquivo de logs do roteador Rx2:

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

Verifique se o servidor Sx1 recebeu um endereço IPv6 Unicast Global através do mecanismo de autoconfiguração stateless.

Agora, vamos comentar as linhas que permitem o recebimento das mensagens RA, RS, NA e 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 o serviço do ip6tables e verifique se o servidor Sx1 continua recebendo um endereço via autoconfiguração stateless. Note que o endereço atribuído anteriormente pode demorar alguns minutos para deixar de ser utilizado pela interface.

Agora descomente apenas as linhas necessárias para que esse serviço volte a funcionar normalmente. Quais mensagens são necessárias para que a autoconfiguração stateless funcione?

Também é possível melhorar esse script permitindo que os endereços FF02::1:FF00:0/104 e FF02::1/128 recebam apenas as mensagens que realmente são destinadas a eles. Consulte na apostila teórica quais são essas mensagens e faça as alterações necessárias no script. Em seguida teste para verificar se a autoconfiguração stateless continua funcionado normalmente.

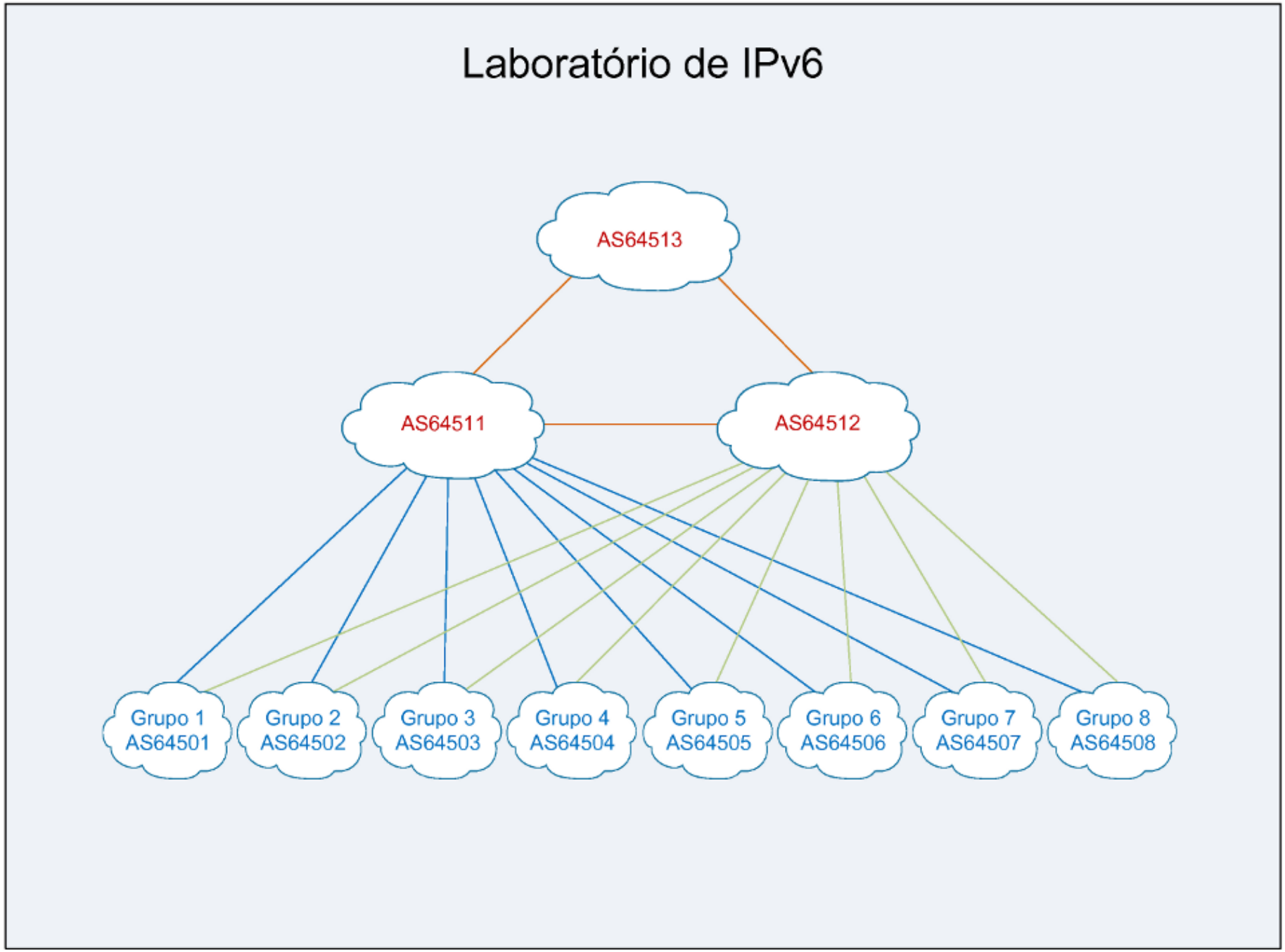
Publicação
ceptro.br

IPv6.br

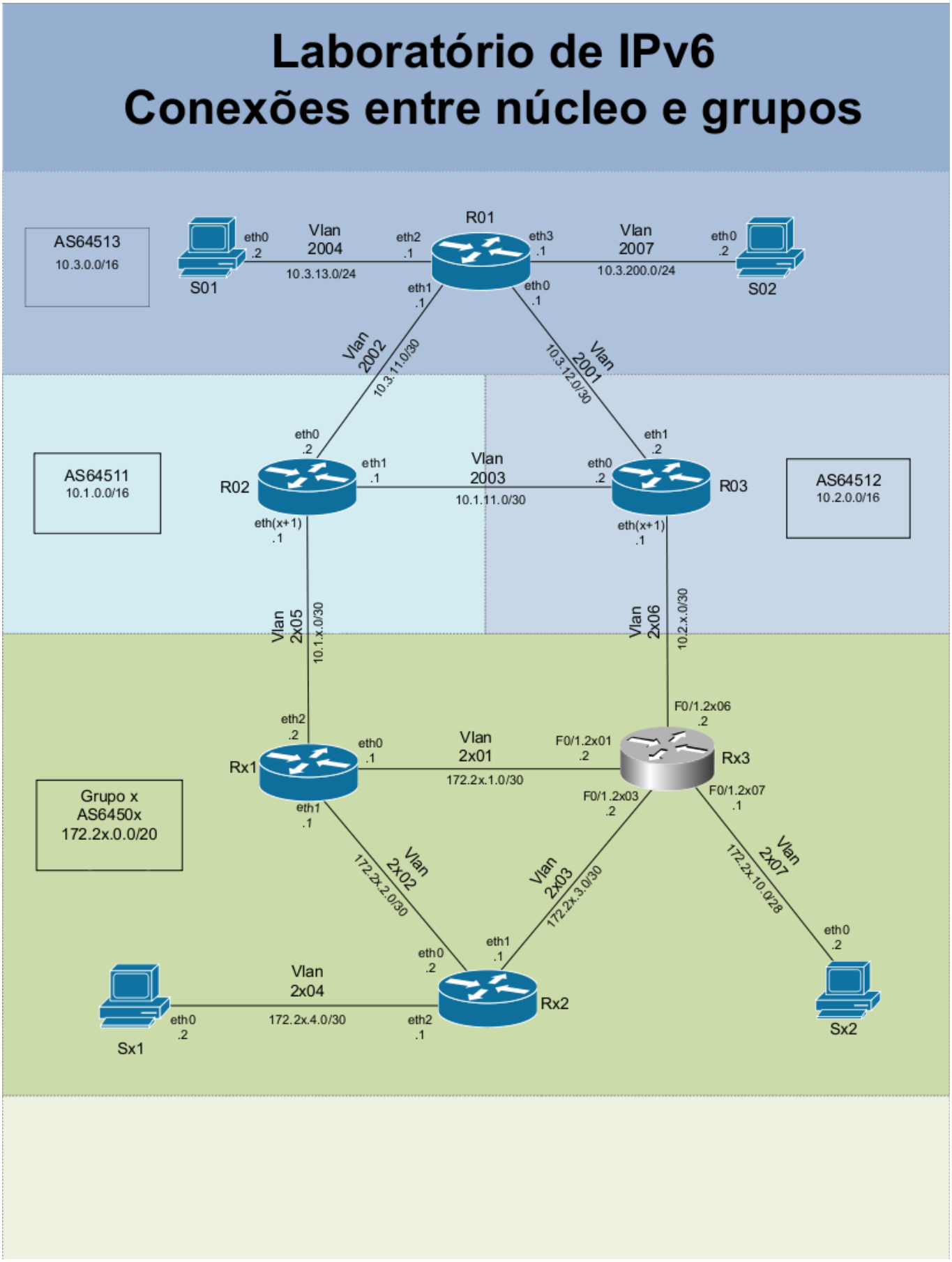
Curso IPv6 básico

Laboratório: Túneis 6to4

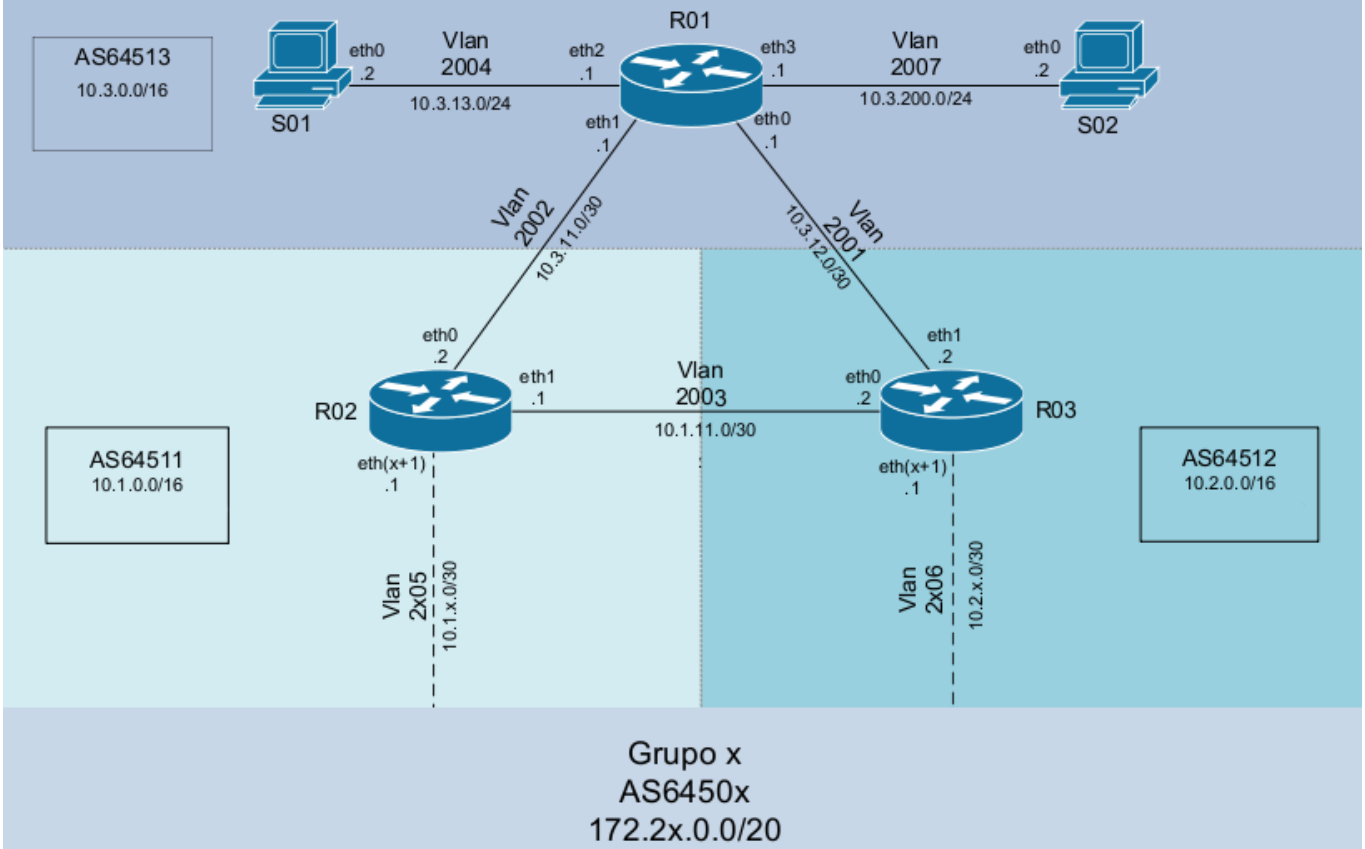
cgi.br nic.br


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	

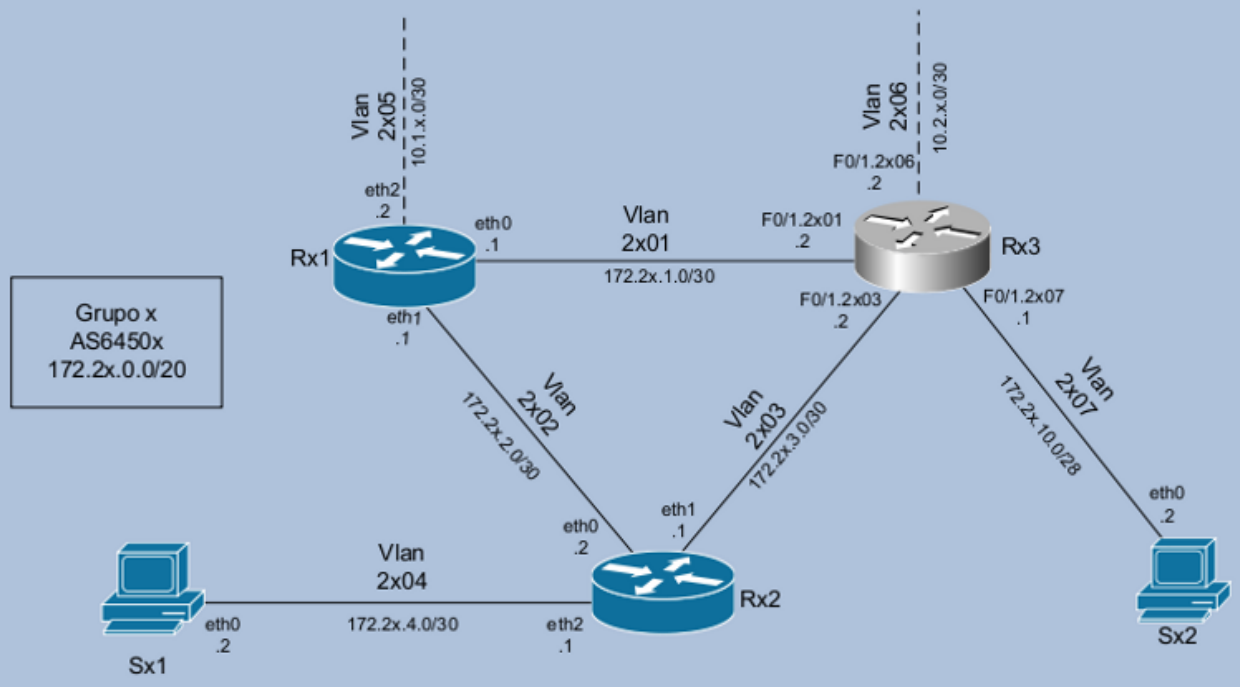
S02		
Interface	IPv4	IPv6
eth0	10.3.200.2/24	

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

Laboratório – Túneis 6to4

Objetivo: Possibilitar conectividade IPv6 ao AS através de um túnel 6to4. Para isso, o Cisco será configurado como o roteador 6to4 de nosso AS, e a partir do bloco de endereço IPv6 atribuído a ele, iremos numerar os servidores e roteadores do AS com endereços 6to4.
Com a utilização do comando tcpdump e do programa Wireshark analisaremos a estrutura de um pacote IPv6 encapsulado em um pacote IPv4.

Cenário inicial: Nessa fase, cada grupo representa um AS distinto com conexão para 2 provedores de transito.

Cada AS possui acesso a um roteador Cisco, dois roteadores Linux/Quagga, e dois servidores Linux. A política de roteamento externo e o protocolo de roteamento interno (IGP), neste caso o OSPF, já estão implementados para IPv4. O grupo deve testar a comunicação dentro do próprio AS e com os demais ASs (use mtr, ping e traceroute IPv4, por exemplo).

Exercício 1: Configurando Túneis 6to4.

Em primeiro lugar vamos calcular o endereço 6to4 local, à partir do endereço IPv4. O comando a seguir lhe ajudará a converter o endereço da interface FastEthernet 0/1.2X01 para hexadecimal:

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

Obs1.: as 4 primeiras letras 'x' não devem ser trocadas pelo número do grupo, pois fazem parte do comando printf. Apenas a última letra deve ser substituída, no trecho '172 2X 1 2'.

Obs2.: no endereço 6to4 gerado, o trecho 'acZZ:0102' corresponde ao endereço IPv4 da interface FastEthernet 0/1.2X01 convertido para hexadecimal.

Agora, ative o túnel com os seguintes 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:: (rota default para 192.88.99.1 – relay 6to4 público)
```

Exercício 2: Testando a conectividade através do Túnel 6to4.

Com as configurações do túnel já realizadas, vamos agora testar a conectividade com os grupos vizinhos.

Confirme com os grupos ao lado qual o endereço 6to4 de seus roteadores Cisco e teste a conectividade, a partir do roteador RX3, através do comando traceroute.

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

Quantos saltos o pacote deu para alcançar o o AS vizinho? Dê um traceroute no endereço IPv4 do roteador Cisco do AS vizinho e compare o número de saltos.

Exercício 3: Analisando os pacotes 6to4.

Acesse o roteador Rx1, inicie um tcpdump na interface eth0.

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

No roteador Rx3 (Cisco) dê pings para os endereços 6to4 dos ASs vizinhos, capture os pacotes e verifique como é o tráfego encapsulado.

Observe a estrutura do pacote capturado. Analise os campos “Protocol”, “Source” e “Destination”do cabeçalho IPv4.

Exercício 4: Numerando a rede com endereços 6to4.

Um túnel 6to4 proporciona um bloco /48 IPv6 para cada endereço IPv4 válido. Vamos utilizar o bloco / 48 obtido pelo roteador Cisco, para configurar em todos os servidores e roteadores do AS um endereço IPv6 (6to4).

Primeiro vamos atribuir um endereço 6to4 ao servidor Sx2 através do mecanismo de autoconfiguração. Para isso, devemos habilitar na interface FastEthernet0/1.2X07 do roteador Rx3 o envio de mensagens Router Advertisement (RA), para que o prefixo 6to4 seja anunciado ao 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
```

O que houve com o anúncio do prefixo? Como resolver o problema?

Defina com seu grupo qual a melhor forma de segmentar a rede do AS. Este bloco pode ser sub-dividido em quaisquer blocos mais específicos (/49, /56, /64, etc.), no entanto, para que o servidor Sx2 obtenha um endereço através do mecanismo de autoconfiguração, o bloco anunciado pelo roteador Rx3 deve ser um /64.

Ex.:

```
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 se o servidor Sx2 recebeu o endereço 6to4 corretamente.

Do mesmo modo como foi feito com a interface FastEthernet0/1.2X07, habilite na interface 0/1.2X03 um endereço 6to4 e o anúncio de mensagens RA. Verifique se o roteador Rx2 recebeu o endereço 6to4 via autoconfiguração stateless. Caso isso não tenha ocorrido, onde está o problema?

De acordo com a estrutura de endereçamento definida pelo grupo, vamos agora configurar o AS para que todos os servidores tenham um endereço IPv6 (6to4). Configure um endereço estático nas interfaces eth1 e eth2 do roteador Rx2, e na interface eth0 do servidor Sx1.

Ex.:

- No roteador 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
```

- No servidor Sx1:

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

Agora vamos configurar as rotas manualmente, de forma a termos conectividade através dos endereços 6to4.

- No servidor Sx1:

`[root@SX1 ~]# ip route add default via 2002:ACZZ:0102:YYYY::YYYY` (endereço 6to4 do roteador Rx2)

- No servidor Sx2:

`[root@SX2 ~]# ip route add default via 2002:ACZZ:0102:YYYY::YYYY` (endereço 6to4 do roteador Rx3)

- No roteador Rx2:

`[root@RX2 ~]# ip route add default via 2002:ACZZ:0102:YYYY::YYYY` (endereço 6to4 do roteador Rx3)

- No roteador 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
```
(endereço 6to4 do roteador Rx2)

Após as configurações, teste a conectividade ponta a ponta, com ping6 de Sx1 para Sx2.

Teste também a conectividade com os outros grupos.

Agora, reconfigure o servidor Sx1 para que este obtenha um endereço 6to4 através de autoconfiguração stateless. Para isso, use o Radvd instalado em Rx2 para fazer anúncio das mensagens RA.

Configure o Radvd editando ou criando o arquivo /etc/radvd.conf com o seguinte conteúdo:

- No roteador 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 o Radvd

- No roteador Rx2:

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

Caso ocorra algum erro ao se iniciar o processo do Radvd, verifique o arquivo de logs do roteador Rx2:

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

Verifique se o servidor Sx1 recebeu um endereço 6to4 corretamente e teste novamente a conectividade internamente e com os outros grupos (entre os servidores Sx1 e Sx2).

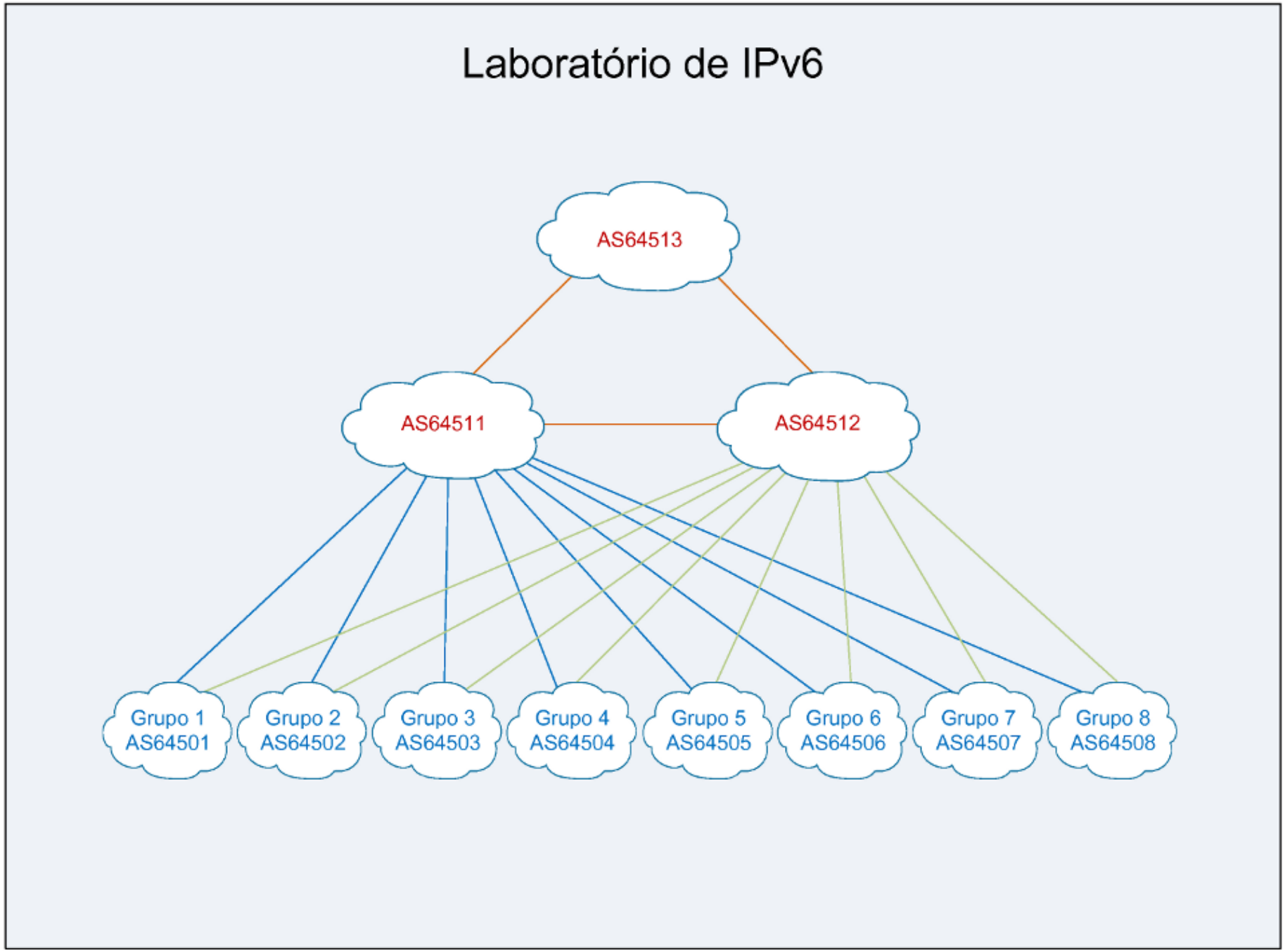
Publicação
ceptro.br

IPv6.br

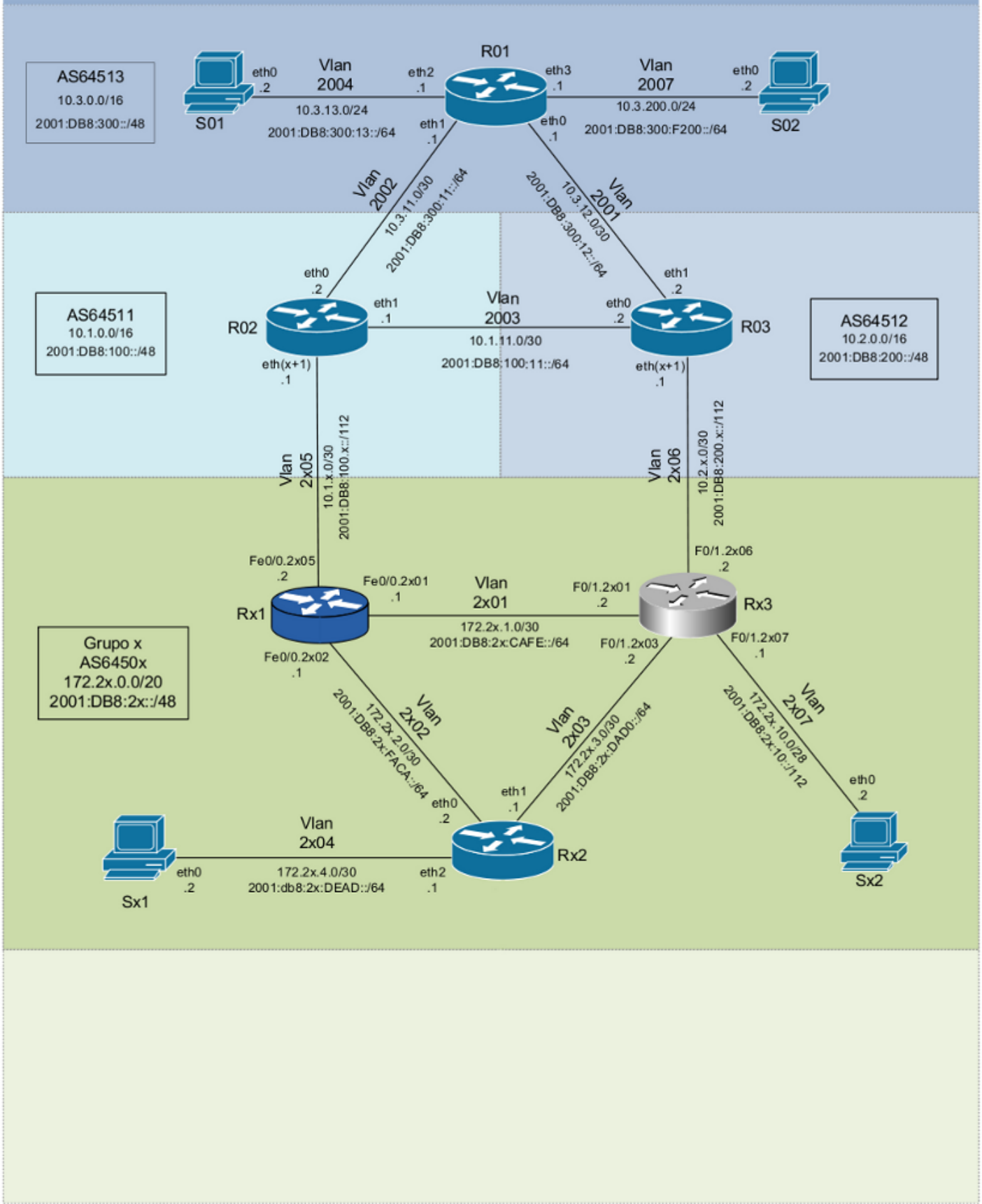
Curso IPv6 básico

Laboratório: Roteamento IPv6

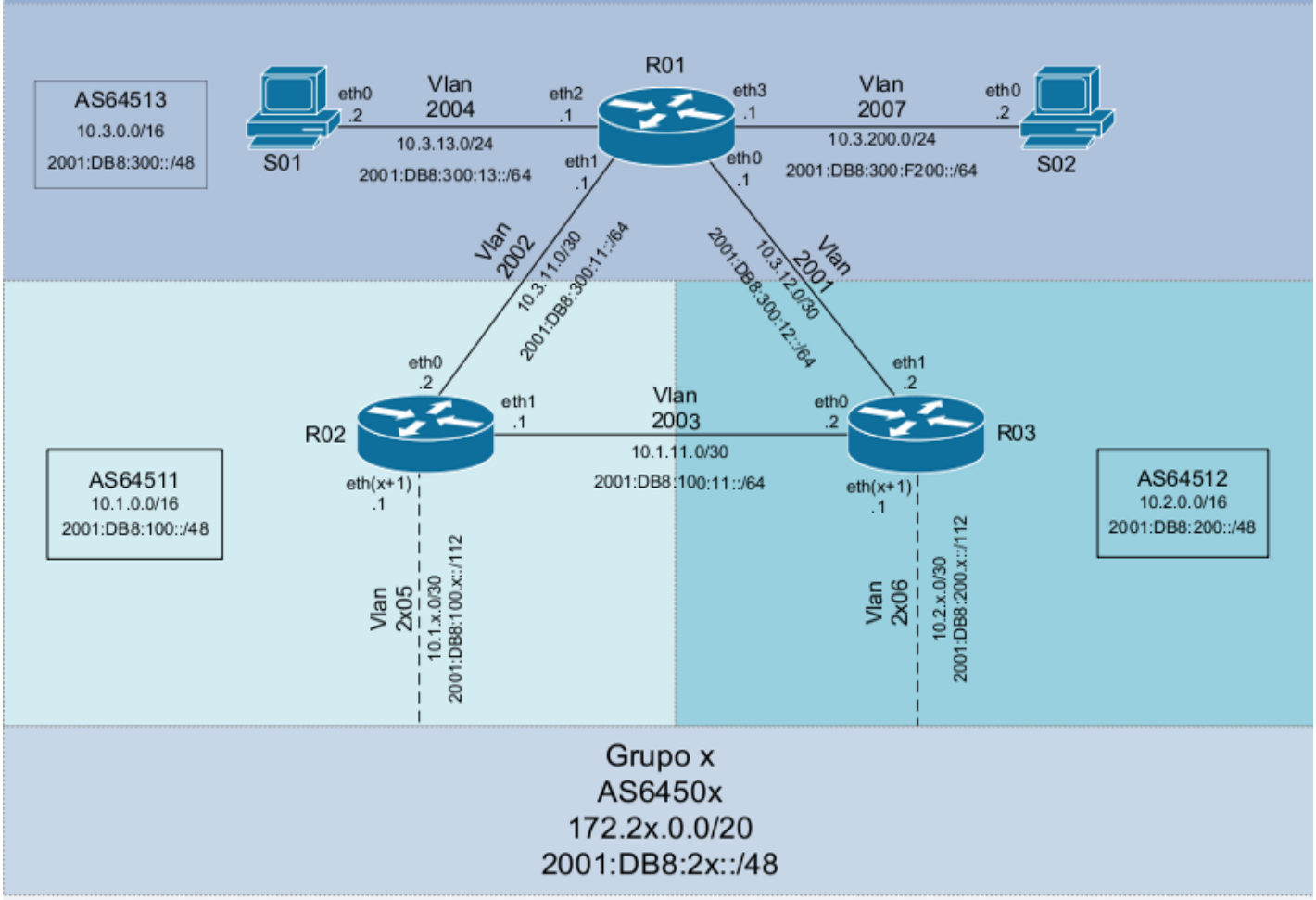
cgi.br nic.br


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



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

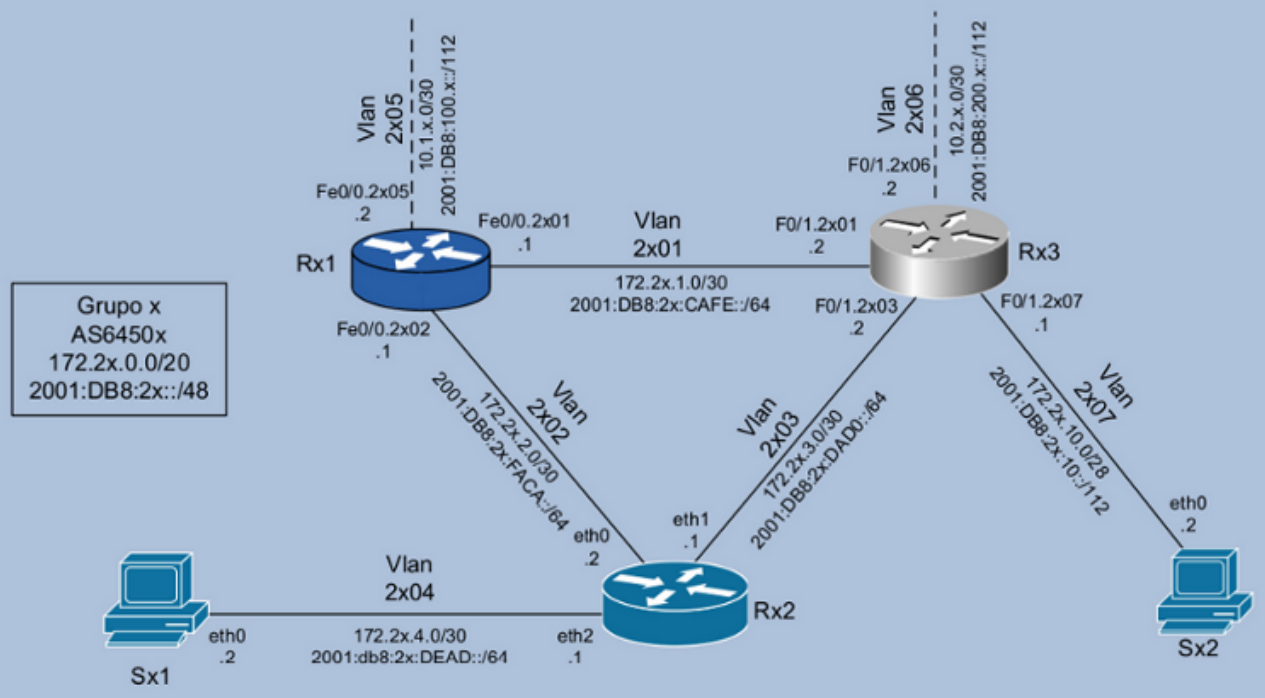
S02		
Interface	IPv4	IPv6
eth0	10.3.200.2/24	2001:DB8:300:F200::2/64

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

Laboratório – Roteamento IPv6

Objetivo: Implementar para o IPv6 uma política de roteamento externo e o protocolo de roteamento interno (IGP), neste caso o OSPF, semelhante a implementação já existente para, IPv4. Para realizarmos esta tarefa, iniciaremos revisando a configuração do IPv4, seguido da configuração do endereçamento IPv6 nas interfaces dos roteadores e dos servidores, do protocolo de roteamento interno OSPFv3, da configuração do iBGP (interna) e do eBGP (externo – operadoras), e por fim, testaremos a conectividade IPv4 / IPv6.

Cenário inicial: Nessa fase, cada grupo representa um AS distinto com conexão para 2 provedores de transito. Os links externos são utilizados para balanceamento de carga e redundância, ou seja, cada link individualmente tem que suportar todo o tráfego do AS, porém em situação normal cada link deverá suportar apenas metade deste tráfego (entrante e sainte). Ex.: suponha que o AS possui 100Mbits de banda total e os links contratados utilizam contratos sob demanda com franquia de 50Mbps e capacidade de 100Mbps (95th percentile).

Cada AS possui acesso a um roteador Cisco, um roteador Linux/Quagga, dois servidores Linux, e a partir de agora, há também um roteador Juniper.

Para acessar o roteador Juniper utilize o seguinte comando:

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

RX1 (ttyp0)

login: juniper
Password: Juniper (ATENÇÃO! A senha começa com "J" maiúsculo)

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

A política de roteamento externo e o protocolo de roteamento interno (IGP), neste caso o OSPF, já estão implementados para IPv4, seguindo as condições acima. O grupo deve testar a comunicação dentro do próprio AS e com os demais ASs (use mtr, ping e traceroute IPv4, por exemplo).

O roteador Linux utiliza a aplicação Quagga para prover os serviços de roteamento. É importante destacar que, diferente das principais implementações de roteadores (ex. Cisco e Juniper), que utilizam uma única CLI (Command Line Interface) para realizar todas as suas configurações, o Quagga baseia-se em uma CLI associada a cada daemon. No Quagga existe um daemon específico para cada protocolo de roteamento, tratado como um processo separado.

Neste laboratório utilizaremos os daemons: ospfd, ospf6d, bgpd e o zebra. Para editar as configurações de cada processo você pode agir de duas maneiras:

1ª – Parando o processo a ser configurado:

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

Edite o arquivo de configuração

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

Reinicie o processo

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

Obs.: Está opção é recomendada apenas quando for criado um arquivo novo, para evitar inconsistência entre as informações do processo em execução com as armazenadas no arquivo.

2ª – Acessando via telnet o terminal de configuração (CLI) de cada daemon com ele ainda em funcionamento. Deste modo, todas as atualizações feitas entrarão em funcionamento imediatamente, sem a necessidade de se reiniciar o serviço.

Cada terminal é acessado através de uma porta TCP específica:

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

Ex.:

```
# 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
```

Obs.: Está opção é recomendada para sistemas em operação.

Neste laboratório, as senhas para acessar e configurar todos os daemons do Quagga são “zebra”.

Mais informações sobre as sintaxes dos comandos nos roteadores Cisco e Quagga podem ser obtidas em:

- 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

Exercício 1 - Verificando a conectividade IPv4

Inicialmente, apenas o os protocolos de roteamento IPv4 estão configurados. Vamos testar a comunicação dentro do próprio ASN, com o núcleo e com os demais ASNs (usando mtr, ping e traceroute IPv4, por exemplo).

Olhe também as configuração de roteamento:

- No roteador 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 (ATENÇÃO! A senha começa com "J" maiúsculo)

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

- No roteador 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
```

- Ainda no roteador Linux/Quagga RX2:

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

- No roteador 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
```

Exercício 2: Configurando as interfaces de rede

Seguindo o planejamento de implantação do protocolo IPv6, já realizamos experimentos internos com o novo protocolo e já testamos a conectividade entre os ASs através de túneis 6to4, o que nos permitiu avaliar o suporte a IPv6 em todos os dispositivos. Agora vamos iniciar a implantação de IPv6 nativo em nosso AS.

Após recebermos a alocação de um bloco de endereços IPv6 /48, devemos planejar de que forma esses recursos serão distribuídos, decidindo quais faixas de endereços serão utilizadas na numeração interna da rede, quais serão utilizadas para serviços, etc.

Iniciaremos agora a configuração básica de endereçamento IPv6 em todas as interfaces de rede dos servidores e roteadores que serão utilizadas no roteamento interno do AS (OSPF), de acordo com a tabela de endereços do diagrama "Laboratório IPv6 - Grupos".

Acesse os servidores SX1 e SX2 e edite o arquivo /etc/sysconfig/network, adicionando as linhas:

```
NETWORKING_IPV6=yes
IPV6_DEFAULTGW=2001:DB8:2X:YYYY::1 (adicione o endereço do roteador mais próximo como gateway)
```

Edite também o arquivo /etc/sysconfig/network-scripts/ifcfg-eth0 adicionando as linhas:

```
IPV6INIT=yes
IPV6ADDR=2001:DB8:2X:YYYY::YYYY/YYY (confira na tabela de endereços do diagrama "Laboratório IPv6 – Grupos", qual o endereço de cada servidor)
```

Feito isso, reinicie as interfaces de rede:

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

Vamos agora, configurar no roteador Juniper os endereços das interfaces ge-0/0/0.2X01 e ge-0/0/0.2X02.

- No roteador 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
```

Agora vamos configurar as interfaces do roteador Linux/Quagga RX2 editando o arquivo /etc/sysconfig/network adicionando a linha:

```
NETWORKING_IPV6=yes
```

Edite também os arquivos /etc/sysconfig/network-scripts/ifcfg-eth_ (para eth0, eth1 e eth2), adicionando as linhas:

```
IPV6INIT=yes
IPV6ADDR=2001:DB8:2X:YYYY::YYYY/YYY (confira na tabela de endereços do diagrama "Laboratório IPv6 – Grupos", qual o endereço de cada interface do roteador)
```

Feito isso, reinicie as interfaces de rede:

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

Vamos agora configurar o endereço da Loopback do roteador Linux/Quagga:

- No roteador 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
```

No roteador Cisco, vamos configurar os endereços IPv6 das interfaces FastEthernet0/1.2X01, 1.2X03 e 1.2X07 e da Loopback10, além de desabilitar o anuncio de mensagens Router Advertisement do protocolo de Descoberta de Vizinhança.

- No roteador 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
```

Exercício 3: OSPFv3

Com todas as interfaces e loopbaks configuradas, já podemos habilitar e configurar o protocolo de roteamento interno OSPFv3 nos roteadores Juniper, Linux/Quagga e Cisco.

Vamos habilitar protocolo OSPF nas interfaces roteador 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
```

No roteador Linux/Quagga RX2, para ativar o daemon ospf6d é preciso primeiro criar o arquivo ospf6d.conf, onde serão armazenadas as configurações do 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 e acesse o daemon ospf6d para configurar o 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 (você pode utilizar o endereço IPv4 da
```

loobback do roteador 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
```

- No roteador Cisco, defina os parâmetros básicos do OSPF e habilite-o nas 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
```

Com o OSPF configurado e ativado em todos os roteadores, vamos conferir a tabela de vizinhos para verificar se todas as rotas internas estão sendo anunciadas corretamente:

- No roteador Juniper RX1:

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

- No roteador Linux/Quagga RX2 (ospf6d):

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

- No roteador Cisco RX3:

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

- Outros comandos para Cisco e Quagga (ospf6d):

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

Teste a conectividade IPv6 dentro do seu AS. Dê pings entre os roteadores e servidores de diferentes segmentos da rede, todos os dispositivos internos já devem estar se “enxergando”.

Exercício 4: BGP

Com a conectividade interna já funcionando, podemos agora iniciar o processo de estabelecimento da conexão com os ASs vizinhos. Para isso, iremos configurar o protocolo BGP nos roteadores de borda RX1 e RX3, para que estes possam se comunicar com os roteadores de borda R02 e R03 dos nossos provedores de transito. A este tipo de relacionamento, entre ASs vizinhos, damos o nome de external BGP (eBGP). No entanto, essa tarefa será dividida em alguns passos como: o estabelecimento de sessões BGP dentro do próprio AS (iBGP), a definição da política de roteamento, e por fim, após a configuração do eBGP, a definição de políticas de fluxo de saída de dados.

Exercício 4a: iBGP

Inicialmente iremos estabelecer sessões BGP entre todos os roteadores internos de nosso AS (iBGP - internal BGP), para manter a consistência de roteamento interno.

As sessões iBGP serão estabelecidas entre interfaces de Loopback, que por serem lógicas, colaboram para aumentar a disponibilidade da rede.

Primeiramente, vamos configurar as Loopbacks utilizadas nessa comunicação. Lembre-se que esta interface já é utilizada para o roteamento IPv4, por isso você não irá criá-la, apenas adicionar o endereço IPv6:

- No roteador 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
```

- No roteador Linux/Quagga RX2 utilizaremos o mesmo endereçamento da Loopback lo.

- No roteador 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
```

Agora vamos configurar os relacionamentos entre os vizinhos:

- No roteador 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
```

- No 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
```

- No roteador 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
```

Confira se os relacionamentos foram estabelecidos entre os vizinhos:

- No roteador Juniper RX1:

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

- No roteador Linux/Quagga RX2:

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

- No roteador Cisco RX3:

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

Também podemos utilizar o comando abaixo para fazer a consulta simultânea às tabelas IPv4 e IPv6:

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

Exercício 4b: Considerações e preparativos para influenciar o tráfego de entrada

Nosso AS recebeu a alocação do bloco de endereço IPv6 2001:0DB8:002X::/48. Para influenciarmos o tráfego de entrada, vamos distribuir os serviços e o consumo de banda entre os dois links (balanceamento de carga). Para isso vamos dividir o bloco /48 em duas partes anunciando cada uma por um único link.

Identifique os dois blocos?

1º - _____

2º - _____

Para redundância será utilizado o prefixo IPv6 correspondente a todo o bloco /48.

Exercício 4c: eBGP

Agora já podemos configurar a relação entre os ASs vizinhos, estabelecendo uma conexão BGP entre nossos roteadores de borda com os roteadores de borda de nossos provedores de transito. Vamos inicialmente configurar a Loopback e as interfaces que serão utilizadas na comunicação eBGP. Lembre-se que elas já estão sendo utilizadas para o roteamento IPv4, por isso você não irá criá-las, apenas adicionar o endereço IPv6:

As sessões eBGP serão estabelecidas de duas formas:

- Entre o roteador Juniper RX1 e o AS 64511 utilizaremos o endereçamento das interfaces físicas entre eles (forma padrão);

- Entre o roteador Cisco RX3 e o AS 64512 utilizaremos o endereçamento das interfaces de Loopback (para aumentar a segurança).

Vamos adicionar os endereços nas interfaces.

- No roteador 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
```

- No roteador 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 configurar as rotas estáticas em nossos roteadores de borda, para gerar os prefixos IPv6 e permitir a conectividade entre as Loopbacks do roteador RX3 e do roteador do AS 64512.

- No roteador 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
```

- No roteador 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
```

No roteador Cisco, também é preciso especificar quais redes do AS serão anunciadas via BGP:

- No roteador Cisco RX3:

```
router-RX3# configure terminal
router-RX3(config)# router bgp 6450X
router-RX3(config-router)# address-family ipv6
router-RX3(config-router-af)# network 2001:DB8:2X::/48
router-RX3(config-router-af)# network 2001:DB8:2X::/49
router-RX3# copy running-config startup-config
```

Com as interfaces já configuradas, vamos estabelecer o relacionamentos entre nosso AS e os ASs vizinhos:

- No roteador Juniper RX1:

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

ATENÇÃO! As configurações de BGP só devem ser aplicadas após estabelecermos as políticas
de fluxo de dados. Portanto, não utilize o comando "commit" ainda.
```

- No roteador 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
```

Exercício 4d: Controle de fluxos de entrada

A aplicação das politicas de anúncios enviados, que vão interferir com o tráfego de entrada (AS-OUT), será dividida em duas funções:

- Redundância:
  - O anuncio do prefixo /48 (equivalente a todo o bloco do AS) deverá ser enviado para todos os ASs externos.

- Balanceamento de carga:
  - o tráfego da faixa 2001:DB8:2X::/49 deve entrar preferencialmente pelo AS 64512;
  - o tráfego da faixa 2001:DB8:2X:8000::/49 deve entrar preferencialmente pelo AS 64511.

Exercício 4e: Controle de fluxos de saída

Para influenciarmos o tráfego de saída (AS-IN), os prefixo recebidos deverão ser preferencialmente distribuídos entre os dois links, de modo que também seja possível conferir a redundância e o balanceamento de carga.

Considere que o tráfego de nosso AS pode ser dividido igualmente para o tráfego do AS64513, de modo que o tráfego com destino ao primeiro prefixo /49 do AS64513 deve sair preferencialmente através de AS64512 e o tráfego com destino ao segundo prefixo /49 do AS64513 deve sair preferencialmente através de AS64511.

Para realizar as duas tarefas acima, você pode se basear nas configurações já existentes para IPv4.

Exercício 4f: Levantando e testando as sessões eBGP

Com as políticas de roteamento já aplicadas, já podemos levantar as sessões eBGP.

- No roteador Juniper RX1:

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

- No roteador Cisco:

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

Agora, confira se os relacionamentos foram estabelecidos entre os ASs vizinhos. Analise o status das conexões BGP:

- No roteador Juniper RX1:

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

- No roteador Cisco:

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

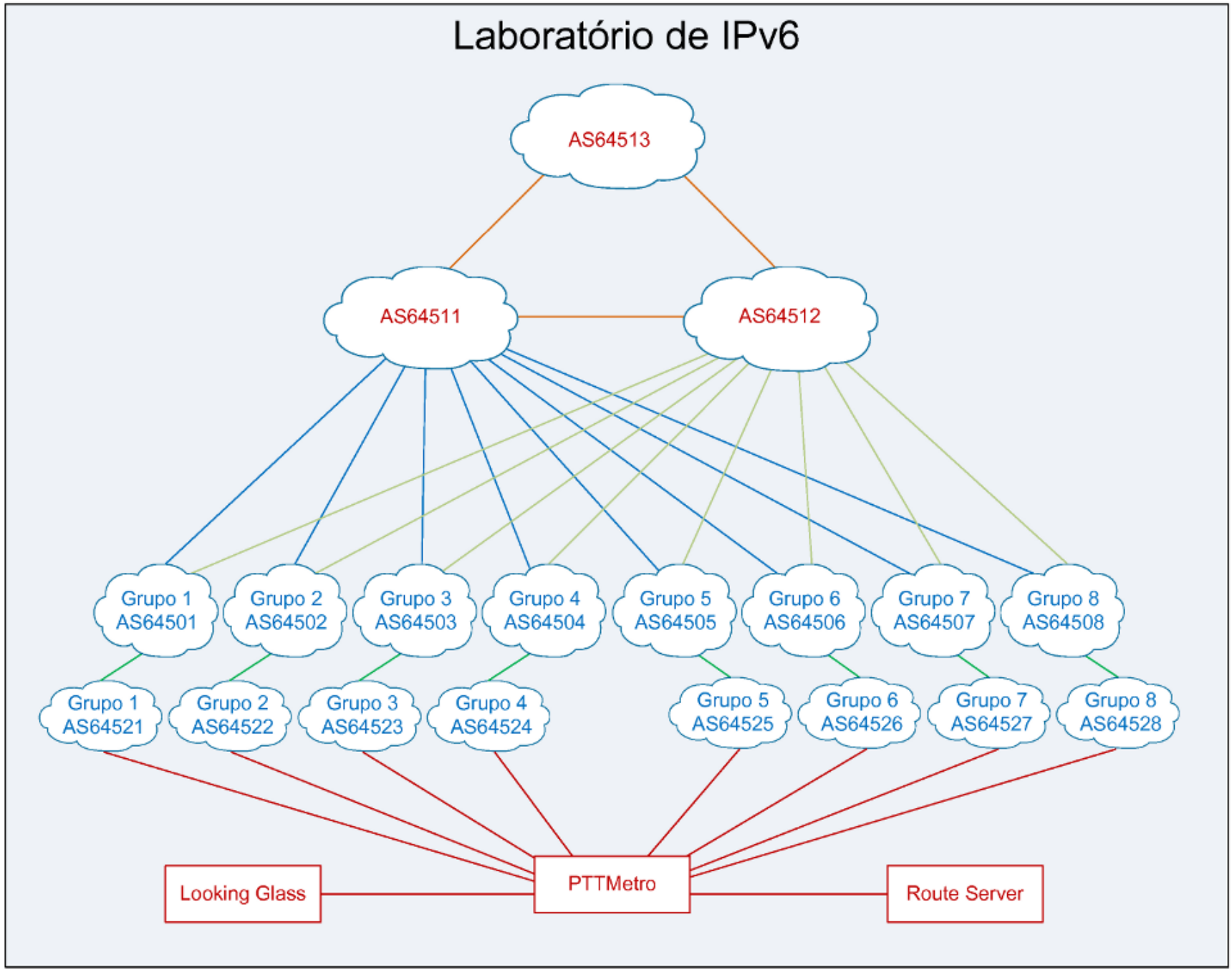
Teste a conectividade entre os roteadores de borda. Dê pings e traceroutes entre os roteadores e servidores do seu AS e os roteadores e servidores dos ASs centrais. A comunicação com os ASs dos outros grupos do laboratório também deve ser possível, desde que estes também já tenham completado esta parte dos exercícios de laboratório. Verifique com os grupos ao lado se eles já completaram o exercício e teste a conectividade entre os ASs.

O As 64513 possui um Looking-Glass configurado. Acesse-o via telnet para verificar se o anuncio das rotas de nosso AS foram configuradas corretamente, seguindo as políticas de roteamento estabelecidas. (O endereço a ser acessado será informado pelo instrutor).

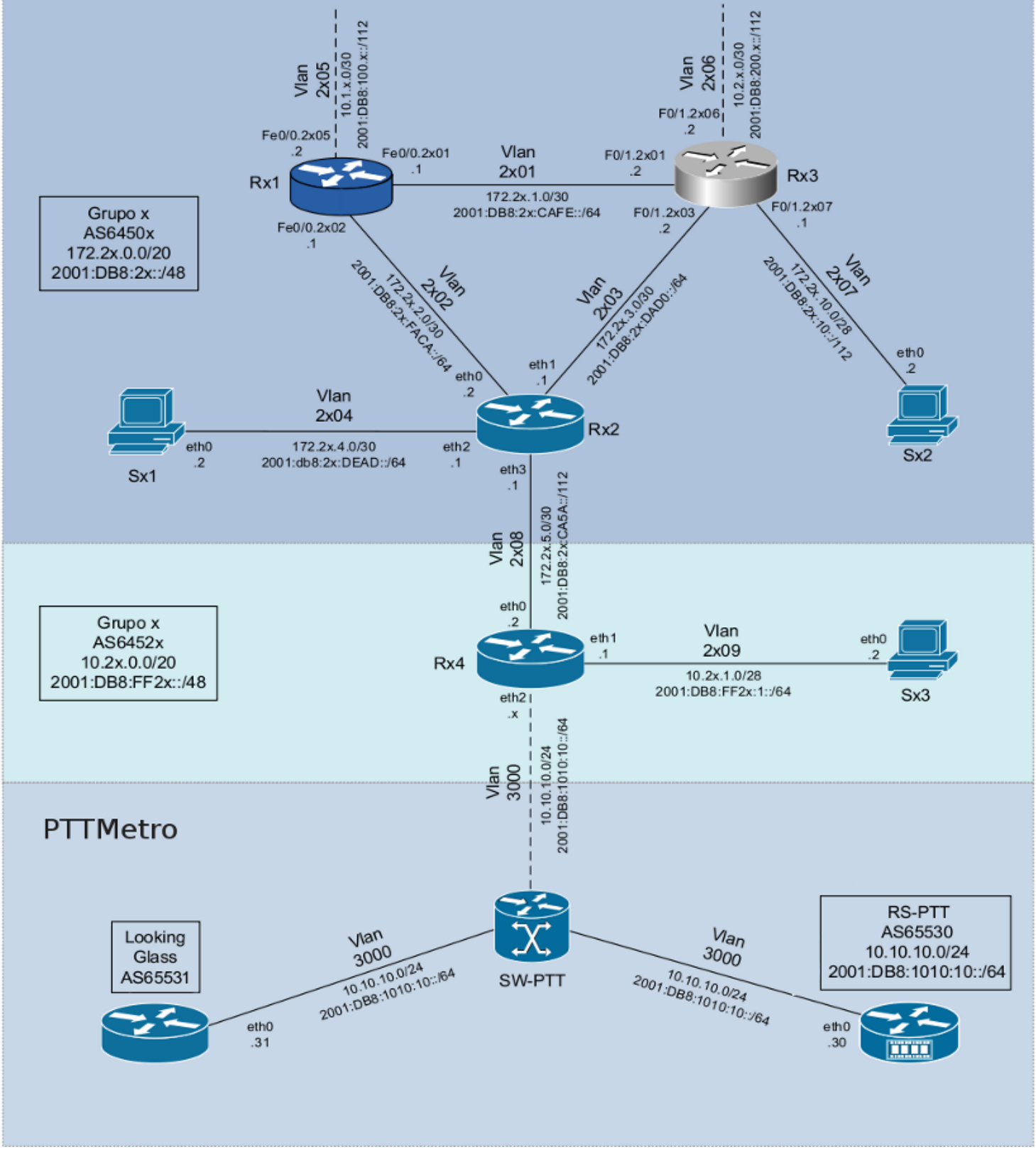
Laboratório – Roteamento IPv6 (Parte 2)
Fornecendo Trânsito e Conectando-se ao PTTMetro

Objetivo: Configurar o AS do grupo para que este forneça trânsito tanto IPv4 quanto IPv6 a um AS cliente, e, em seguida, configurar a conexão entre esse AS cliente a um Ponto de Troca de Tráfego (PTT).

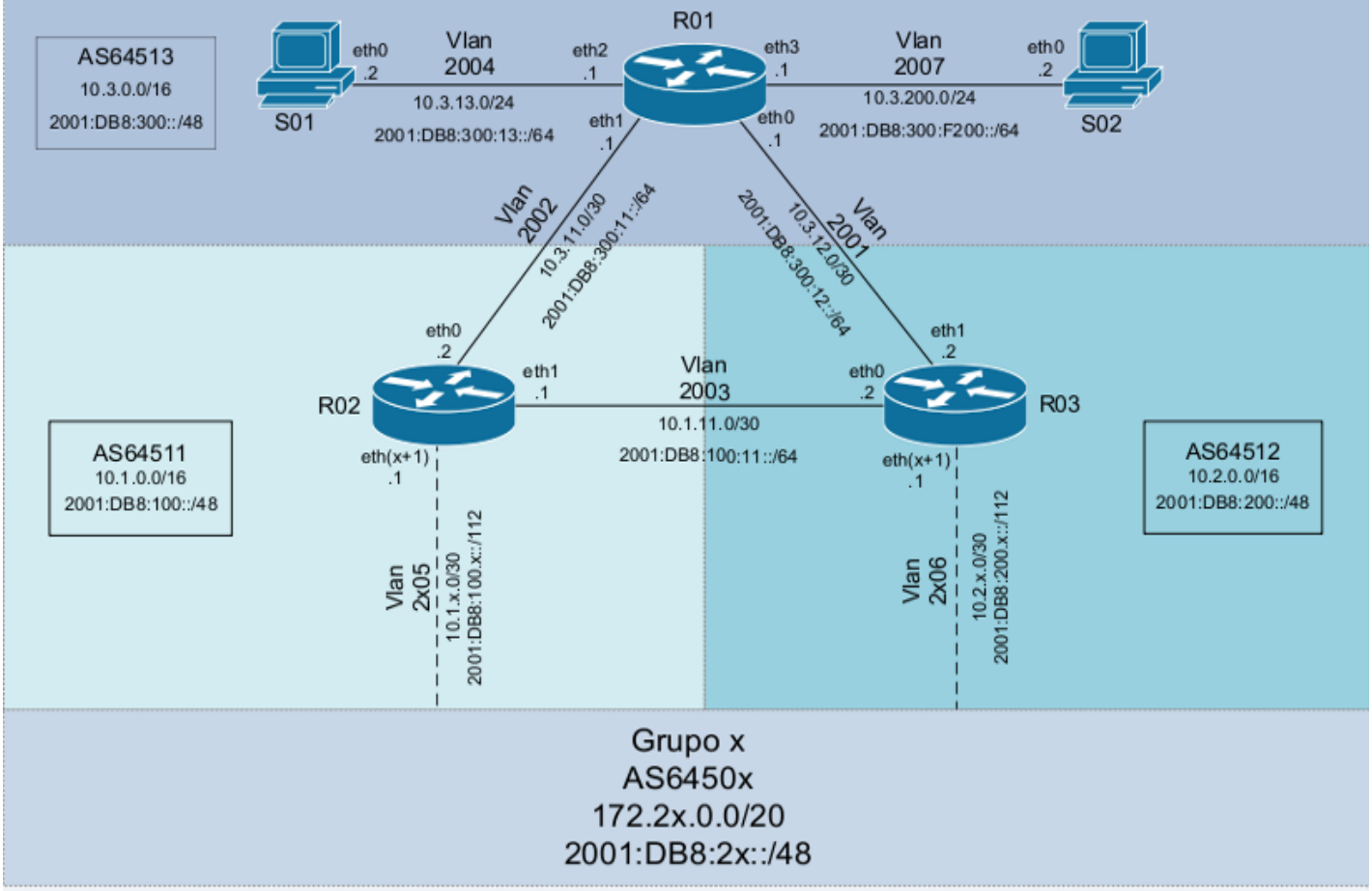
Cenário inicial: Após realizarmos as configurações de roteamento interno e as configurações de roteamento externo com os ASs centrais, nosso AS (AS6450X) já está em operação.
Agora, será adicionado ao nosso cenário, um As cliente, composto por um roteador Linux/Quagga e um servidor Linux. As configurações de endereçamento e roteamento IPv6 do AS cliente ainda não foram estabelecidas. Assim que estas configurações sejam feitas, esse AS cliente irá se conectar ao 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

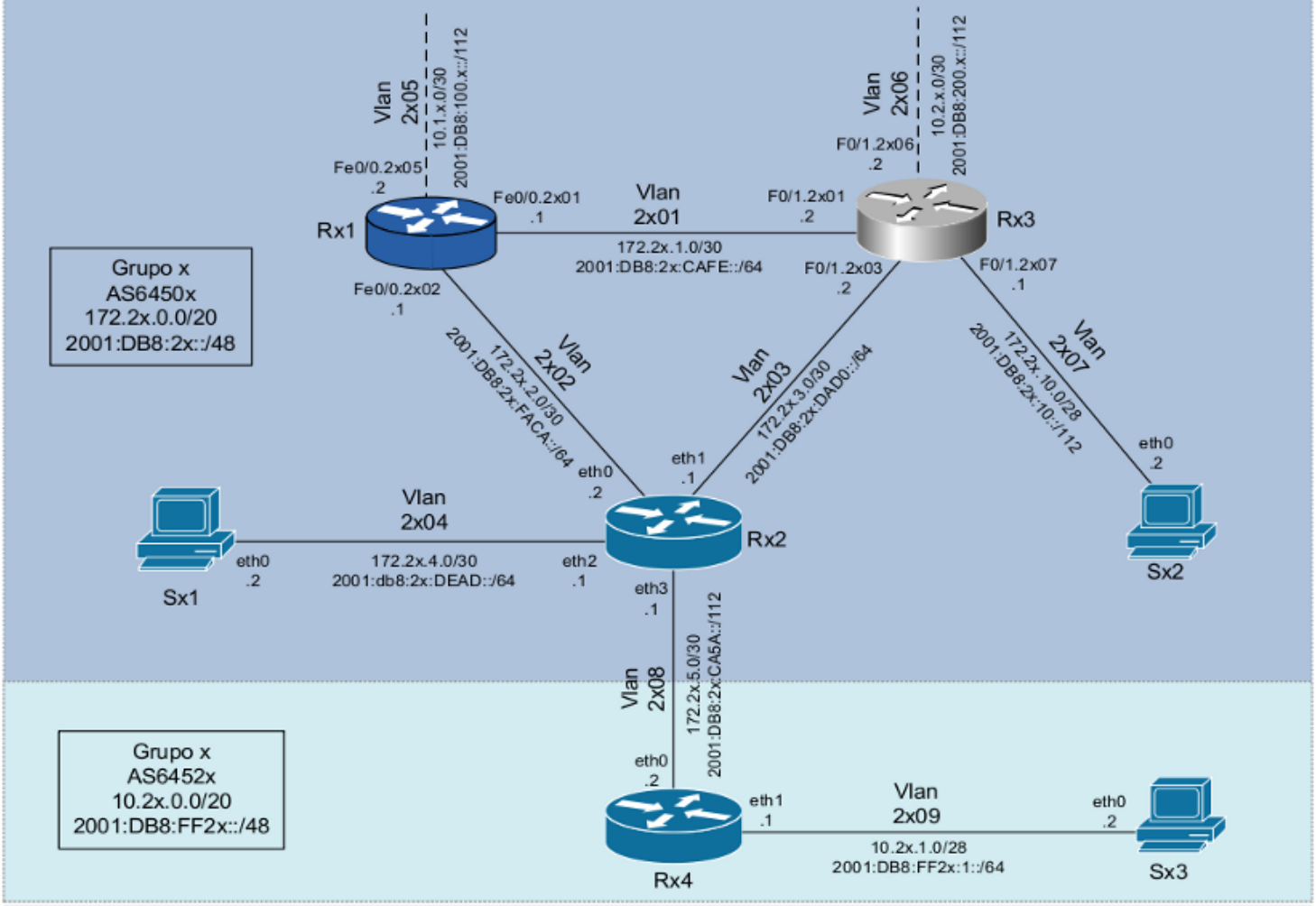
S02		
Interface	IPv4	IPv6
eth0	10.3.200.2/24	2001:DB8:300:F200::2/64

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

Exercício 1: Configurando o AS cliente.

Será conectado ao nosso AS (AS6450X) um AS cliente (AS6452X), ao qual nós forneceremos trânsito tanto IPv4 quanto IPv6. No entanto, para que este serviço seja estabelecido, algumas etapas devem ser cumpridas:

Para acessar o roteador Rx4, primeiramente é preciso estabelecer uma sessão eBGP IPv4 entre os roteadores Rx2 (AS6450X) e Rx4 (AS6452X) utilizando o endereço das interfaces de Loopback.

1º Passo: Endereçamento

- Este AS ainda não possui nenhuma configuração, nem de endereçamento nem de roteamento, definida. Portanto, nosso primeiro passo será configurar toda a parte de endereçamento IPv6 no roteador Rx4 e no servidor Sx3. Consulte o diagrama e a tabela de endereçamento da página anterior para saber quais endereços devem ser adicionados.

Os endereços da interface de Loopback e da interface eth3 do roteador Rx2 também devem ser configurados.

Estas configurações são similares as realizadas no execício 2 da primeira parte de roteamento (Configurando as interfaces de rede).

2º Passo: eBGP

- Com os endereços das interfaces já definidos, podemos agora, configurar a relação entre nosso AS e o AS cliente, estabelecendo uma sessão eBGP entre os roteadores de borda Rx2 (AS6450X) e Rx4 (AS6452X). A sessão eBGP deve ser estabelecida utilizando o endereçamento das interfaces de Loopback.

Por fim, devemos configurar a política de trânsito de nosso AS (AS6450X), evitar o recebimento de anúncios desnecessários vindos do AS cliente.

3º Passo: Testando a conectividade

- Agora, o roteador e o servidor do AS cliente já devem possuir conectividade tanto IPv4 quanto IPv6 com todos os dispositivos do nosso AS e com os dos ASs vizinhos. Teste a comunicação dentro do próprio AS e com os demais ASs usando comandos como mtr, ping e traceroute, por exemplo. Utilize o comando traceroute para verificar as rotas utilizadas na comunicação entre o AS cliente e os ASs vizinhos. Observe-as bem, pois elas serão utilizadas para compararmos os resultados obtidos agora, com os resultados obtidos no próximo exercício.

Exercício 2: Estabelecendo uma conexão com o PTTMetro.

Nesta etapa final do laboratório de roteamento, iremos conectar o AS cliente a um Ponto de Troca de Tráfego (PTT). Para isso, é preciso estabelecer uma conexão eBGP com o Route Server (AS65530). No diagrama "Laboratório IPv6 – Grupo e PTT" você pode verificar os endereços que devem ser utilizados na interface eth2 do roteador Rx4 e no estabelecimento da conexão BGP.

O AS cliente (6452X) deve montar sua política de roteamento para preferir os caminhos de entrada e saída pelo PTTMetro. Isto pode, por exemplo, ser feito da seguinte forma: Para o AS6450X deverá ser anunciado apenas o prefixo /48 IPv6. Para o Route Server (AS65530) deverá ser anunciado os dois prefixos /49 IPv6, além de aumentar o valor do *Local-Preference* para 150 para todos os prefixos aprendidos via PTTMetro.

Para minimizar o impacto de crescimento da Tabela BGP deve-se evitar anúncios desnecessários, mesmo dentro do PTTMetro.

Estabeleça também, uma sessão eBGP com o Looking-Glass (AS65531). Para "alimentar" o Looking-Glass deve-se anunciar todos os prefixos conhecidos pelo AS6452X.

Agora, compare os anúncios do AS cliente (AS6452X) nos Looking-Glasses do PTTMetro e do AS64513.

Isto feito, teste a conectividade entre o AS cliente e os AS dos outros grupos e trace as rotas entre eles. Compare as rotas utilizadas agora, com a a conexão com o PTT já estabelecida, com as rotas que foram utilizadas anteriormente. Qual a principal diferença?

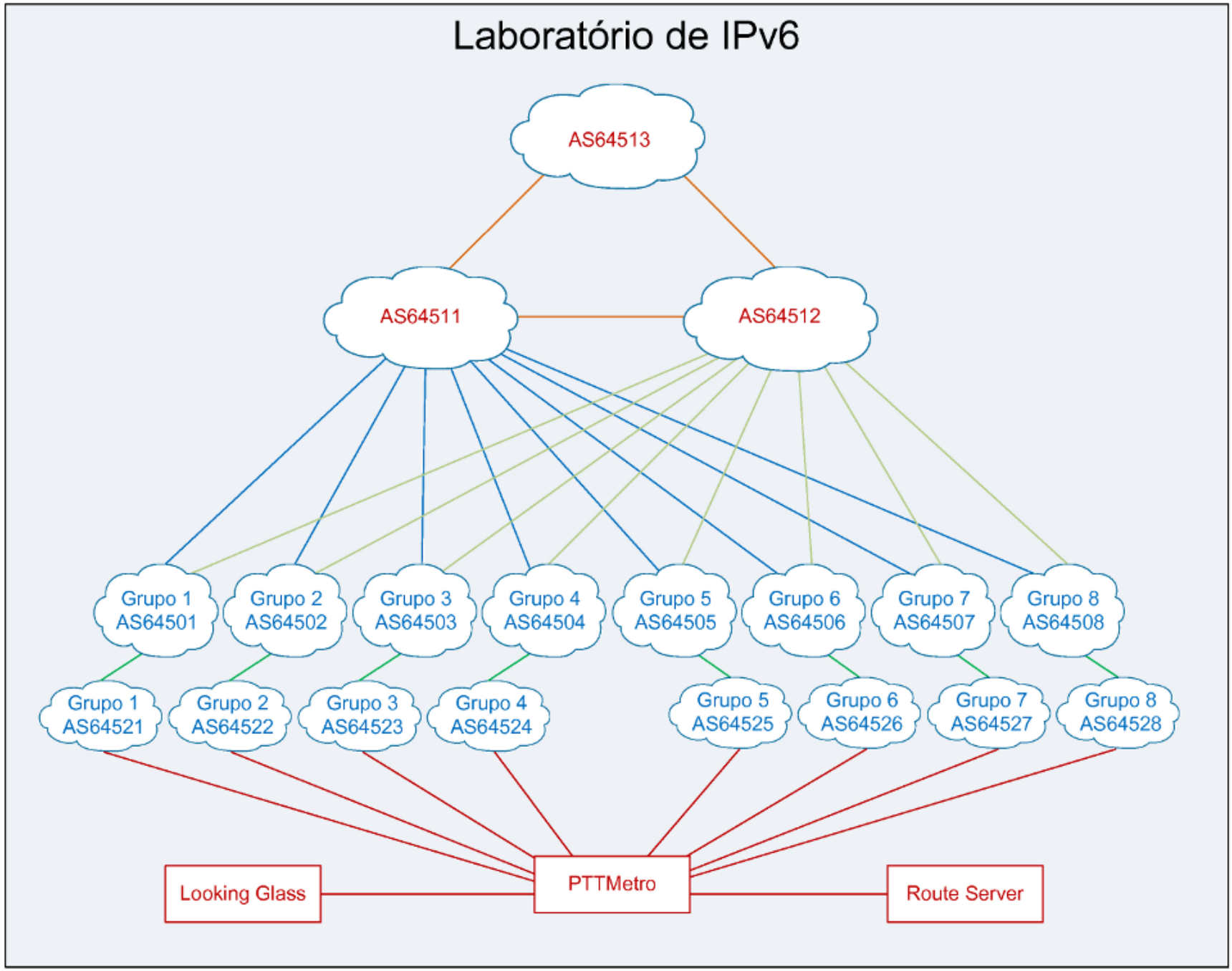
Publicação
ceptro.br

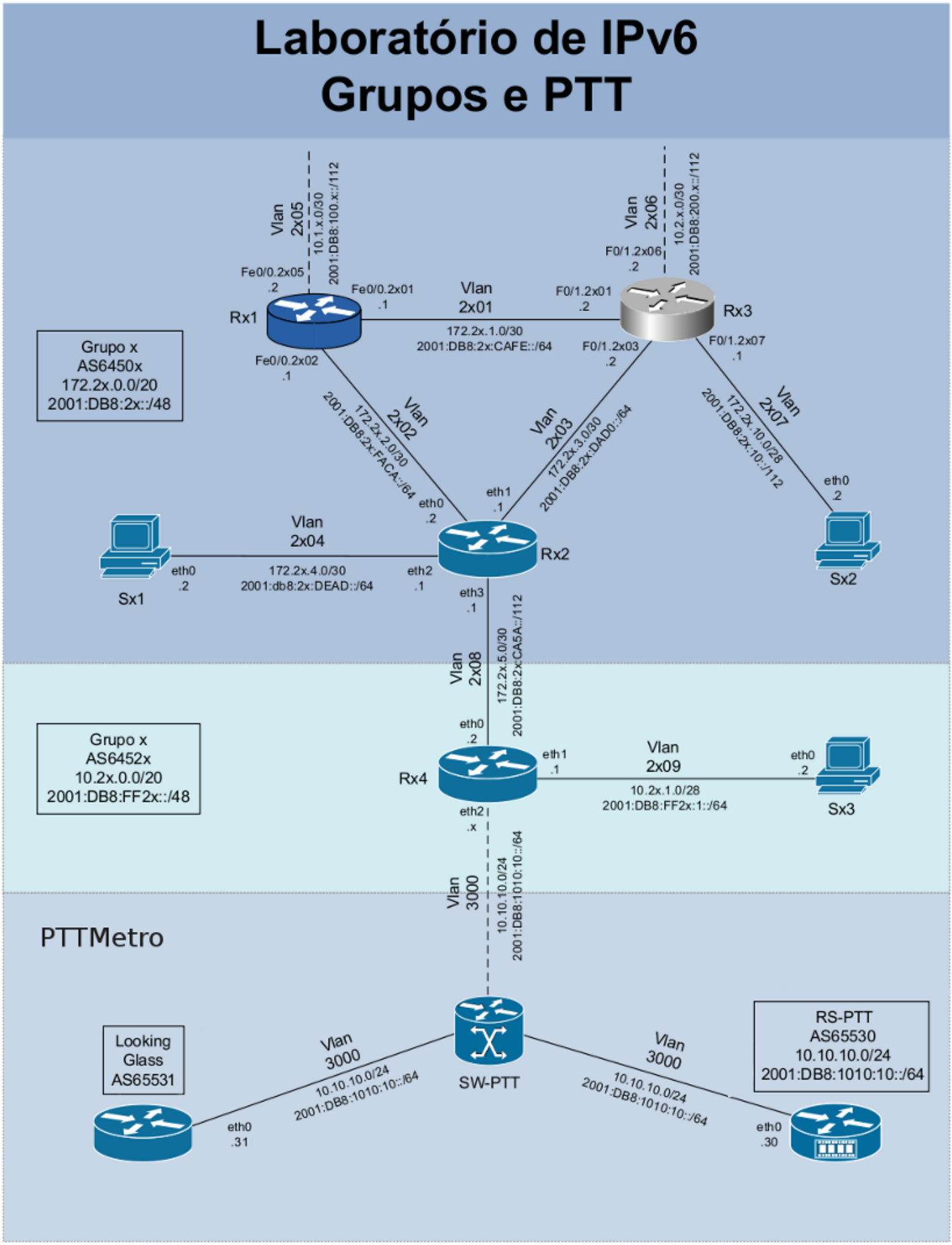
IPv6.br

Curso IPv6 básico

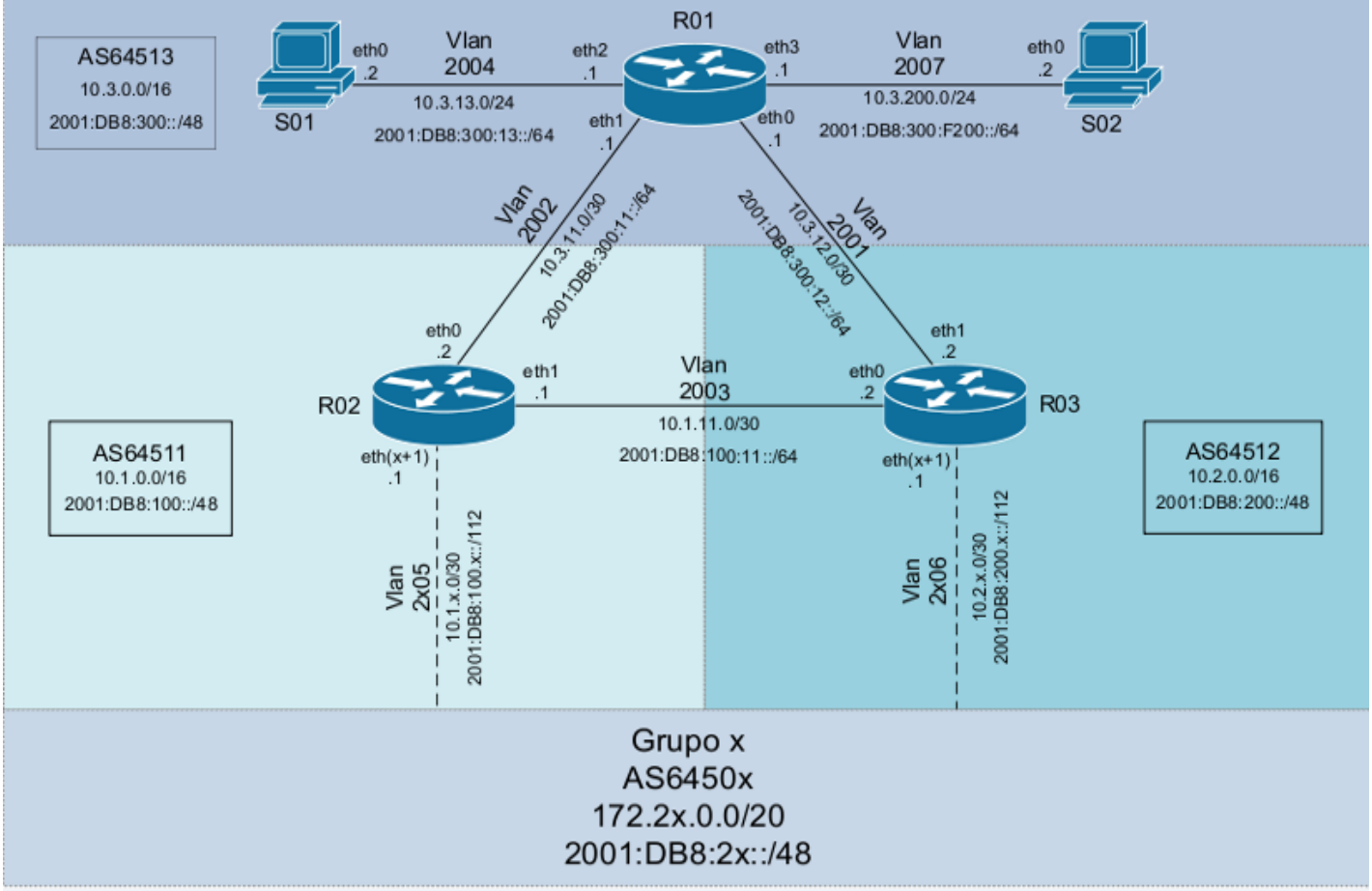
Laboratório: DNS

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

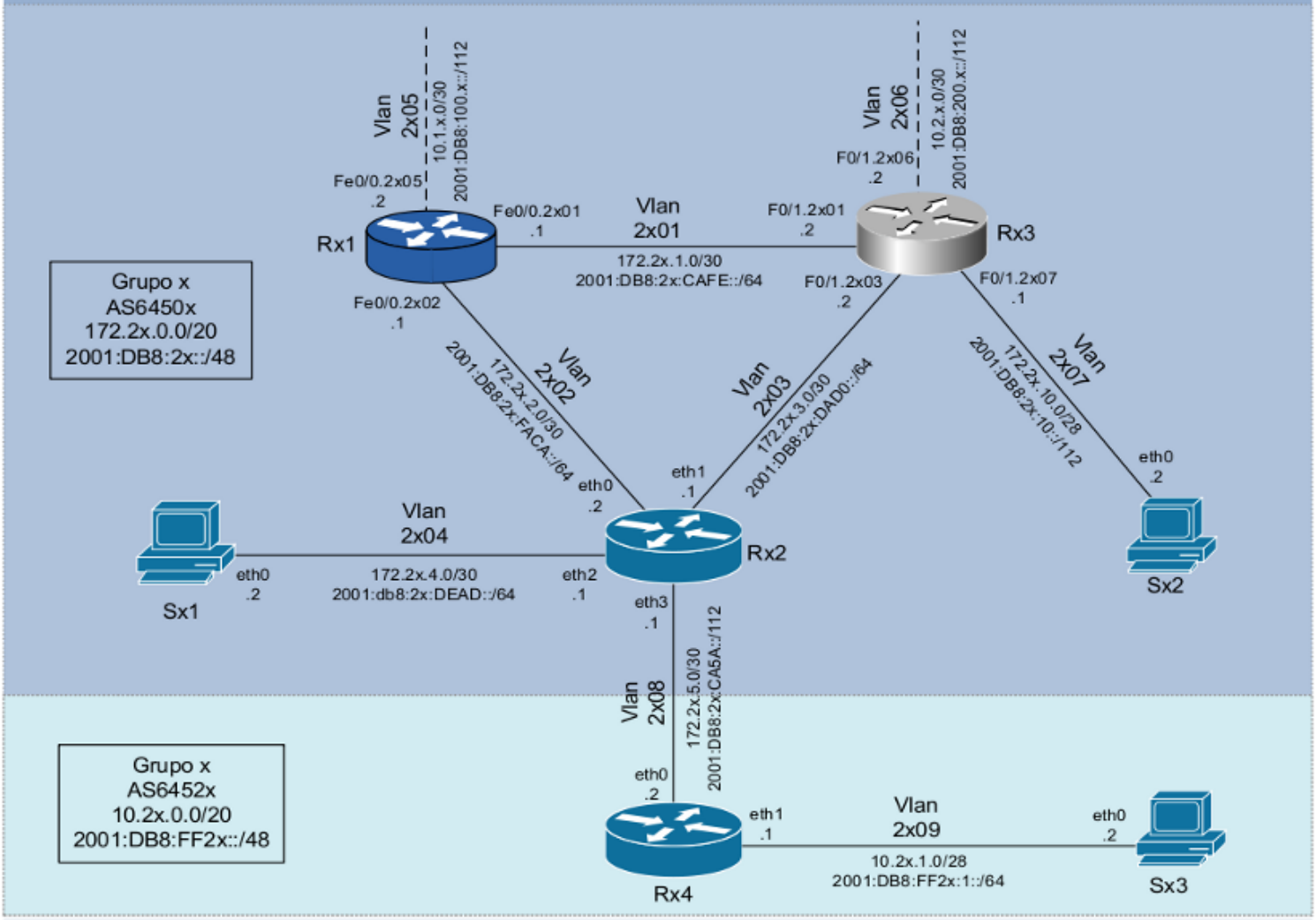
S02		
Interface	IPv4	IPv6
eth0	10.3.200.2/24	2001:DB8:300:F200::2/64

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

Laboratório – DNS

Objetivo: Configurar um servidor de DNS, utilizando BIND9, responsável por responder a requisições feitas ao domínio de primeiro nível .gx (a letra 'x' representa o número do grupo). Também iremos configurar os servidores e roteadores do AS com
diretivas A e AAAA, para que as consultas aos seus domínio possam retornar tanto endereços IPv6 quanto IPv4.

Cenário inicial: Nessa fase, cada grupo representa um AS distinto com conexão para 2 provedores de transito, além de fornecer trânsito a um AS cliente.

Cada AS possui acesso a um roteador Cisco, um roteador Linux/Quagga, um roteador Juniper e dois servidores Linux. A política de roteamento externo e o protocolo de roteamento interno (IGP), neste caso o OSPF, já estão implementados tanto para IPv4 quanto IPv6. O grupo deve testar a comunicação dentro do próprio AS e com os demais ASs (use mtr, ping e traceroute IPv4 e IPv6, por exemplo).

Exercício 1: Configurando o servidores

Neste laboratório, cada AS será responsável pela administração de um domínio de primeiro nível, de modo que o grupo 1 será o responsável pelo domínio .g1, o grupo 2 pelo .g2 e assim sucessivamente.

O servido S01 localizado no AS64513 será nosso servidor raiz. É ele quem delegará ao servidor de DNS do nosso AS a autoridade sobre o domínio .gx.

Nosso servidor de DNS será o Sx2. Ele já possui o BIND9 instalado, portanto, podemos iniciar sua configuração editando o arquivo named.conf, onde indicaremos a zona que nosso servidor irá responder e também a zona "." do tipo "hint", a raiz da Internet.

- No servidor Sx2:

Abra o arquivo /etc/named.conf e substitua seu conteúdo por:

```
// 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";
};
```

Obs. 1: lembre-se de trocar o 'X' pelo número do seu grupo!
Obs. 2: faça o download desse script no endereço
http://[**********]/downloads/named.conf.txt

Nós criaremos um domínio para cada servidor e para cada roteador Linux. Para isso, editaremos o arquivo gX.zone, onde adicionaremos a configuração de cada um.

- No servidor Sx2:

Abra o arquivo /var/named/gX.zone e substitua seu conteúdo por:

```
$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
```

Obs. 1: lembre-se de trocar o 'X' pelo número do seu grupo!
Obs. 2: faça o download desse script no endereço
http://[**********]/downloads/gX.zone.txt

Com essas configurações, cada um desses dispositivos deverá responder pelos seus respectivos domínios: sX1.gX, sX2.gX e rX2.gX.

Vamos indicar para o nosso servidor de DNS o endereço do servidor raiz adicionando ao arquivo root.zone as seguintes informações:

- No servidor Sx2:
  Abra o arquivo /var/named/root.zone e substitua seu conteúdo por:

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

Reinicie o serviço do BIND:

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

Agora, vamos adicionar em cada um deles o arquivo resolv.conf, onde indicaremos o servidor Sx2 como o servidor de DNS de nosso AS.

- Nos servidores Sx1 e Sx2, e no roteador Rx2:

Abra o arquivo /etc/resolv.conf e substitua seu conteúdo por:

```
nameserver 172.2X.10.2
```

Com as configurações realizadas até o momento, já podemos testar se o serviço de DNS está funcionado dentro do nosso AS. Utilize comandos como ping, nslookup e dig para verificar se a consulta por nome está sendo traduzida corretamente para o endereço IP correspondente. Faça essas consultas tanto para endereços IPv6 quanto 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=61 time=4.95 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 com os outros grupos se eles já concluíram essas tarefas e teste a conectividade aos outros ASs através dos domínios cadastrados.

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

Você pode analisar as consultas DNS realizadas ao servidor Sx2 utilizando o comando tcpdump.

- No servidor Sx2:

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

Faça consultas aos servidores dos ASs vizinhos a partir do servidor Sx1 tanto para endereços IPv6 quanto IPv4. Observe na saída do comando tcpdump que ambas as consultas são realizadas via conexão IPv4. Altere o arquivo resolv.conf do servidor Sx1, adicionando o endereço IPv6 do servidor de DNS e refaça este teste. Há alguma alteração nos dados obtidos nas duas consultas?

Exercício 2: Resolução de Reverso

No servidor de DNS Sx2, adicione ao arquivo `/etc/named.conf` o seguinte conteúdo:

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

Obs.: lembre-se de trocar o 'X' pelo número do seu grupo!

Com isso, indicamos ao servidor por qual zona ele deverá responder com autoridade.

Em seguida, crie o arquivo `/var/named/reverso_ipv6.db`. Ele conterá as informações dessa zona. Esse arquivo deve ter o mesmo nome daquele indicado na configuração geral do servidor.

O arquivo deverá conter as seguintes informações:

```
;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.
```

Obs.: lembre-se de trocar o 'X' pelo número do seu grupo!

As linhas acima indicam o nome da zona que se está configurando e seus RR (*Resources Records*), como por exemplo, o SOA (*Start of Authority*), que indica o servidor autoridade para essa zona.

Em seguida, as informações entre parênteses têm o objetivo de organizar a sincronização com servidores secundários (*slaves*). Em seguida, há informação dos servidores DNS para a zona, que podem ser mais do que um. E finalmente, as informações mais interessantes no caso da resolução reversa, que são aquelas indicado o nome associado a cada endereço IP do bloco. O *Resource Record* PTR indica um ponteiro entre o endereço e respectivo

nome. Por exemplo, o endereço 2001:0DB8:002X:0010:0000:0000:0000:0002, que também pode ser representado como 2001:DB8:2X:10::1, tem associado o nome sX2.gX.

Com as configurações realizadas até o momento, já podemos testar se a resolução reversa está funcionado dentro do nosso AS. Utilize comandos como `nslookup` e `host` para verificar se a consulta por endereço IPv6 está sendo traduzida corretamente para o nome correspondente.

Ex.:

```
[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.
```

Ex.:

```
[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.
```

Referências:

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