

### Exercício 3 - iBGP

**Objetivo:** Configurar as sessões iBGP com o intuito de repassar as rotas aprendidas externamente para todos os roteadores do AS e também para divulgar as rotas para as redes dos clientes.

**Cenário inicial:** Protocolo de roteamento interno configurado (OSPFv2 e OSPFv3).

1. Crie uma interface de *loopback* em cada roteador que será utilizada para estabelecer as sessões iBGP.

- No roteador **MikrotikClientes** utilize os seguintes comandos:

```
/interface bridge
  add name=lo20 auto-mac=no admin-mac=1A:B0:06:02:XX:20
/ip address
  add address=10.XX.0.251/32 interface=lo20 comment=IBGP
/ipv6 address
  add address=4D0C:XX::251/128 interface=lo20 comment=IBGP
```

- No roteador **MikrotikBorda** utilize os seguintes comandos:

```
/interface bridge
  add name=lo20 auto-mac=no admin-mac=1A:B0:06:01:XX:20
/ip address
  add address=10.XX.0.254/32 interface=lo20 comment=IBGP
/ipv6 address
  add address=4D0C:XX::254/128 interface=lo20 comment=IBGP
```

2. O passo seguinte é adicionar as interfaces de *loopback* criadas à área 0 do OSPFv2 e do OSPFv3.

- No roteador **MikrotikClientes** utilize os seguintes comandos:

```
/routing ospf network
  add area=backbone disabled=no network=10.XX.0.251/32
/routing ospf-v3 interface
  add area=backbone interface=lo20 passive=yes
```

- No roteador **MikrotikBorda** utilize os seguintes comandos:

```
/routing ospf network
  add area=backbone disabled=no network=10.XX.0.254/32
/routing ospf-v3 interface
  add area=backbone interface=lo20 passive=yes
```

3. Teste a conectividade entre as *loopback* criadas, tanto IPv4 quanto IPv6. No **MikotikClientes**

```
/ping 10.XX.0.254
/ping 4D0C:XX::254
```

4. No **MikotikBorda**

```
/ping 10.XX.0.251
/ping 4D0C:XX::251
```

5. No passo seguinte, configure as sessões iBGP entre os roteadores. Com isso, quando as sessões eBGP com os provedores de trânsito estiverem configuradas, a rede do AS estará preparada para divulgar as redes dos clientes e propagar internamente o que for aprendido da Internet. A configuração será feita em full mesh, ou seja, todos os roteadores terão sessões iBGP com os demais dentro do próprio AS.

- No roteador **MikrotikClientes** utilize os seguintes comandos:

```
/routing bgp instance
  set default as=655XX router-id=10.XX.0.251
/routing bgp peer
  add address-families=ip name=IBGP-IPV4-MKT-BORDA update-source=lo20 \
nexthop-choice=force-self remote-address=10.XX.0.254 remote-as=655XX
  add address-families=ipv6 name=IBGP-IPV6-MKT-BORDA update-source=lo20 \
nexthop-choice=force-self remote-address=4D0C:XX::254 remote-as=655XX
```

- No roteador **MikrotikBorda** utilize os seguintes comandos:

```
/routing bgp instance
  set default as=655XX router-id=10.XX.0.254
/routing bgp peer
  add address-families=ip name=IBGP-IPV4-MKT-CLIENTES remote-as=655XX \
nexthop-choice=force-self remote-address=10.XX.0.251 update-source=lo20
  add address-families=ipv6 name=IBGP-IPV6-MKT-CLIENTES \
update-source=lo20 nexthop-choice=force-self \
remote-address=4D0C:XX::251 remote-as=655XX
```

6. Para verificar se todas as sessões foram estabelecidas corretamente, utilize os comandos listados abaixo. Atente ao estado e à quantidade de prefixos aprendidos de cada vizinho. No **MikrotikClientes**

```
/routing bgp peer print
```

```
Flags: X - disabled, E - established
```

#	INSTANCE	REMOTE-ADDRESS	REMOTE-AS
0	E default	10.XX.0.254	655XX
1	E default	4D0C:XX::254	655XX

7. Para verificar se todas as sessões foram estabelecidas corretamente, utilize os comandos listados abaixo. Atente ao estado e à quantidade de prefixos aprendidos de cada vizinho. No **MikrotikBorda**

```
/routing bgp peer print
```

```
Flags: X - disabled, E - established
```

#	INSTANCE	REMOTE-ADDRESS	REMOTE-AS
0	E default	10.XX.0.251	655XX
1	E default	4D0C:XX::251	655XX

8. Para verificar as informações com mais detalhes é possível utilizar o comando

```
/routing bgp peer print detail status
```

9. Com as sessões estabelecidas, podemos agora configurar o iBGP para divulgar as redes dos clientes. Isso será feito através da declaração das redes através do comando *network* no **MikrotikClientes**.

```
/routing bgp network
add disabled=no network=10.XX.1.0/24 synchronize=no
add disabled=no network=10.XX.2.0/23 synchronize=no
add disabled=no network=4D0C:XX:4000::/40 synchronize=no
add disabled=no network=4D0C:XX:C000::/40 synchronize=no
```

10. Verifique as rotas recebidas em **MikrotikBorda**

```
/ip route print where bgp
/ipv6 route print where bgp
```

Todas as rotas foram consideradas válidas? Para verificar com mais detalhes utilize o comando

```
/ip route print detail where bgp
/ipv6 route print detail where bgp
```

11. Verifique a conectividade do **MikrotikBorda** para os clientes.

```
/ping 10.XX.2.100
/ping 4D0C:XX:C000::100
/ping 10.XX.1.100
/ping 4D0C:XX:4000::100
```

O resultado está condizente com o esperado? Todos os clientes foram alcançados?

**O Mikrotik possui um bug no iBGP em IPv6. Ele espera que o nexthop esteja diretamente conectado e como estamos usando um nexthop de loopback ele não consegue ativar a tabela de rotas. Isso é um bug porque ele deveria reconhecer isso sem problemas, assim como feito no IPv4. Para corrigir isso precisamos alterar o nexthop para um IP diretamente conectado ao Mikrotik.**

12. Aplique o patch do bug no **MikrotikClientes**

```
/routing filter
  add chain=BUG-IBGP-MKT-IPV6 set-in-nexthop-ipv6=4D0C:XX:0:1::1
/routing bgp peer
  set IBGP-IPV6-MKT-BORDA in-filter=BUG-IBGP-MKT-IPV6
```

13. Aplique o patch do bug no **MikrotikBorda**

```
/routing filter
  add chain=BUG-IBGP-MKT-IPV6 set-in-nexthop-ipv6=4D0C:XX:0:1::2
/routing bgp peer
  set IBGP-IPV6-MKT-CLIENTES in-filter=BUG-IBGP-MKT-IPV6
```

14. Verifique novamente as rotas recebidas em **MikrotikBorda**

```
/ip route print where bgp
/ipv6 route print where bgp
```

15. Verifique novamente a conectividade do **MikrotikBorda** para os clientes.

```
/ping 10.XX.2.100
/ping 4D0C:XX:C000::100
/ping 10.XX.1.100
/ping 4D0C:XX:4000::100
```