

Laboratório 03 - iBGP

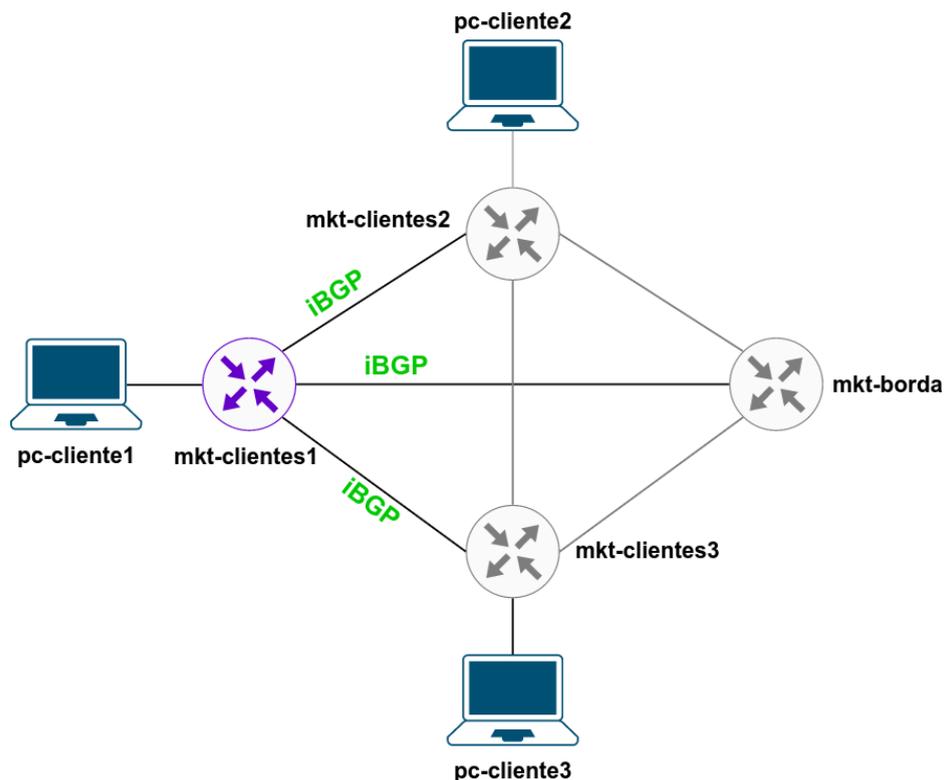
Objetivo

Configurar as sessões de **iBGP** no roteador **mkt-clientes1**, permitindo a troca dinâmica de informações de roteamento referente as redes clientes dos roteadores **mkt-clientes2** e **mkt-clientes3**.

As sessões iBGP também possibilitam que o roteador **mkt-clientes1** anuncie suas rotas para o **mkt-borda**, bem como possibilitam que o **mkt-borda** envie as rotas aprendidas externamente (via eBGP) para todos os roteadores do AS.

Obs: Você realizará as configurações apenas no **mkt-clientes1**, os outros roteadores já estão configurados, e por esse motivo eles aparecem com a cor cinza na topologia abaixo.

As configurações de interfaces de rede e **OSPFv2** e **v3** já foram realizadas no **mkt-clientes1**.



*** É preciso substituir XX nas configurações a seguir pelo número do seu grupo.**

Os equipamentos Mikrotiks estão sem senha, para acessá-los use o usuário **admin.**

Parte 1 - Configuração de Rede dos Clientes

Etapa 1: Configuração de Rede nos PCs

Antes de iniciar os testes, realize as configurações de rede do `pc-cliente1`, `pc-cliente2` e `pc-cliente3`.

Em todos os computadores do laboratório foi desenvolvido um script de configuração, que fará as configurações de rede automaticamente, basta você digitar o número do seu grupo.

1. Acesse o `pc-cliente1`. As credenciais dessa máquina são:

Login: root

Senha: toor

2. Execute o script de configuração com o seguinte comando:

```
./conf-rede-pc-cliente1.sh
```

- a. Digite o número do seu grupo.
- b. Confirme a configuração digitando 'y'.
- c. Ou digite 'n' para inserir o grupo novamente.
- d. Digite o comando abaixo e verifique se as configurações foram aplicadas:

```
ip add
```

Repita o Passo 1 e 2 com o `pc-cliente2` e `3` (não esqueça de mudar o nome do script para `conf-rede-pc-cliente2.sh` e `conf-rede-pc-cliente3.sh`).

Parte 2 - Criação de Interfaces de Loopback

Etapa 1: Criar Interface de Loopback no `mkt-clientes1`

Você utilizará uma interface de *loopback*, que será utilizada para as sessões iBGP e também servirá como Router-ID do equipamento (mesmo procedimento adotado no laboratório de OSPF).

1. No `mkt-clientes1`, crie um interface de *loopback* com o seguinte comando:

```
/interface bridge  
add name=lo20 auto-mac=no admin-mac=1A:B0:06:01:XX:20
```

2. Configure os endereços IPv4 e IPv6 na Interface `lo10`:

```
/ip address  
add address=10.XX.0.248/32 interface=lo20 comment=IBGP
```

```
/ipv6 address
add address=4D0C:XX::248/128 interface=lo20 comment=IBGP
```

Parte 3 - Inserir interface de Loopback no OSFP

Etapa 1: Inserir interface de Loopback no OSPFv2 e v3 no mkt-clientes1

Para estabelecer a sessão de iBGP através da interface de Loopback, faz-se necessário que ela seja propagada via OSPF para os outros roteadores.

1. Adicione a interface de loopback no OSPFv2 com o seguinte comando:

```
/routing ospf network
add area=backbone network=10.XX.0.248/32
```

2. Adicione a interface de loopback no OSPFv3 com o seguinte comando:

```
/routing ospf-v3 interface
add area=backbone interface=lo20 passive=yes
```

3. Verifique se os outros roteadores (**mkt-clientes2**, **mkt-clientes3** e **mkt-borda**) conseguem se comunicar com o **mkt-clientes1** com os seguintes comandos:

```
/ping 10.XX.0.248
/ping 4d0c:XX::248
```

Parte 4 - Estabelecer as sessões iBGP

Etapa 1: Criar a instância iBGP no mkt-clientes1

1. Para iniciar a instância iBGP, realize o seguinte comando:

```
/routing bgp instance
set default as=655XX router-id=10.XX.0.248
```

Etapa 2: Configurar as sessões iBGP com os outros roteadores

Para funcionar corretamente e evitar loops de roteamento, o iBGP deve rodar em uma rede *full-mesh*. Isso exige que as sessões de iBGP sejam fechadas entre todos os roteadores do AS. Dessa forma, será necessário fechar três sessões de iBGP (uma vez que temos três roteadores vizinhos no AS):

1. Crie as sessões de iBGP (IPv4 e IPv6) com o roteador **mkt-clientes2**:

```
/routing bgp peer
add address-families=ip name=IBGP-IPV4-MKT-CLIENTES2 \
update-source=lo20 nexthop-choice=force-self \
```

```
remote-address=10.XX.0.249 remote-as=655XX

add address-families=ipv6 name=IBGP-IPV6-MKT-CLIENTES2 \
update-source=lo20 nexthop-choice=force-self \
remote-address=4D0C:XX::249 remote-as=655XX
```

2. Crie as sessões de iBGP (IPv4 e IPv6) com o roteador **mkt-clientes3**:

```
/routing bgp peer

add address-families=ip name=IBGP-IPV4-MKT-CLIENTES3 \
update-source=lo20 nexthop-choice=force-self \
remote-address=10.XX.0.250 remote-as=655XX

add address-families=ipv6 name=IBGP-IPV6-MKT-CLIENTES3 \
update-source=lo20 nexthop-choice=force-self \
remote-address=4D0C:XX::250 remote-as=655XX
```

3. Crie as sessões de iBGP (IPv4 e IPv6) com o roteador **mkt-borda**:

```
/routing bgp peer

add address-families=ip name=IBGP-IPV4-MKT-BORDA \
update-source=lo20 nexthop-choice=force-self \
remote-address=10.XX.0.251 remote-as=655XX
add address-families=ipv6 name=IBGP-IPV6-MKT-BORDA \
update-source=lo20 nexthop-choice=force-self \
remote-address=4D0C:XX::251 remote-as=655XX
```

4. Verifique se as sessões iBGP foram estabelecidas (deve aparecer a letra E no canto esquerdo de cada sessão) utilizando o seguinte comando:

```
/routing bgp peer print
```

5. Tente realizar um teste de ping utilizando o **pc-cliente1** para qualquer um dos outros computadores:

```
#Ping para o pc-cliente2
ping6 4d0c:XX:400::10

#Ping para o pc-cliente3
ping6 4d0c:XX:c00::10
```

O teste de conectividade deu certo? Se não, por que ele não funcionou?

Etapa 3: Anunciar o prefixo dos clientes

Como vimos no passo anterior, mesmo com as sessões iBGP estabelecidas, os clientes da Região 1 não conseguem se comunicar com os outros clientes das Regiões 2 e 3. Isso porque o protocolo BGP não é como o OSPF, que propaga as suas rotas automaticamente após ser ativado nas sub-redes/interfaces.

Para habilitar a comunicação entre os clientes, é necessário anunciar as rotas dos clientes que estão conectados ao roteador **mkt-clientes1**.

Antes de realizar as configurações, verifique quais rotas estão sendo anunciadas via iBGP no AS.

1. Verifique as rotas aprendidas via iBGP utilizando o seguinte comando em **todos os roteadores**:

```
#Verificar as rotas IPv4 aprendidas via BGP
/ip route print where bgp

#Verificar as rotas IPv6 aprendidas via BGP
/ipv6 route print where bgp
```

Quantas e quais rotas estão sendo aprendidas pelos roteadores?

2. Anuncie as rotas IPv4 e IPv6 disponíveis no **mkt-clientes1**:

```
/routing bgp network
add disabled=no network=10.XX.1.0/24 synchronize=no
add disabled=no network=4d0c:XX:800::/40 synchronize=no
```

3. Repita o Passo 1 e verifique se as rotas foram apreendidas pelos outros roteadores. Todas as rotas estão ativas e válidas?
4. Efetue novamente o teste de ping no **pc-clientes1**:

```
#Ping para o pc-cliente2
ping6 4d0c:XX:400::10

#Ping para o pc-cliente3
ping6 4d0c:XX:c00::10
```

Os testes retornaram resultados positivos?

5. Tente realizar os testes em IPv4:

```
#Ping para o pc-cliente2
ping 10.XX.2.10

#Ping para o pc-cliente3
```

```
ping 10.XX.3.10
```

Os testes retornaram resultados positivos?

Etapa 4: Aplicar *patch* para o *bug* do IPv6 no Mikrotik

Como vimos nos teste anteriores, os pacotes IPv4 funcionam normalmente, porém os pacotes IPv6 não funcionam.

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

Para corrigir esse *bug*, vamos criar três filtros e aplicá-los nas três sessões de iBGP IPv6 (uma sessão para cada roteador do AS) do roteador **mkt-clientes1**.

1. Crie os filtro alterando o *nexthop* de cada umas das três sessões:

```
/routing filter
add chain=BUG-IBGP-MKT-CLIENTES2-IPV6 \
set-in-nexthop-ipv6=4D0C:XX::2

add chain=BUG-IBGP-MKT-CLIENTES3-IPV6 \
set-in-nexthop-ipv6=4D0C:XX::6

add chain=BUG-IBGP-MKT-BORDA-IPV6 \
set-in-nexthop-ipv6=4D0C:XX::a
```

2. Aplique os filtros em cada um das sessão de iBGP:

```
/routing bgp peer
set IBGP-IPV6-MKT-CLIENTES2 \
in-filter=BUG-IBGP-MKT-CLIENTES2-IPV6

set IBGP-IPV6-MKT-CLIENTES3 \
in-filter=BUG-IBGP-MKT-CLIENTES3-IPV6

set IBGP-IPV6-MKT-BORDA in-filter=BUG-IBGP-MKT-BORDA-IPV6
```

3. Realize novamente o teste de conectividade IPv6 no **pc-clientes1**:

```
#Ping para o pc-cliente2
ping6 4d0c:XX:400::10

#Ping para o pc-cliente3
ping6 4d0c:XX:c00::10
```

Os testes retornaram resultados positivos?

Etapas 5: Testes com o iBGP

1. Desative a interface **ether2** do roteador **mkt-clientes1**.

```
/interface disable ether2
```

2. Verifique as sessões de BGP, mudou alguma coisa?

```
/routing bgp peer print
```

3. Agora desative a interface **ether3** do roteador **mkt-clientes1**.

```
/interface disable ether3
```

4. Verifique as sessões de BGP, mudou alguma coisa?

```
/routing bgp peer print
```

- Algumas das sessões iBGP foram desativadas?
- Existe alguma diferença entre a saída desse comando e o realizado no **Passo 4 da Etapa 2**?
- Por que mesmo com duas interfaces desativadas as sessões de iBGP continuam funcionando?