# Experiment 6b - Roles of BGP Neighbors

DE-CIX Academy

Version 1.0

## 1 Network Setup

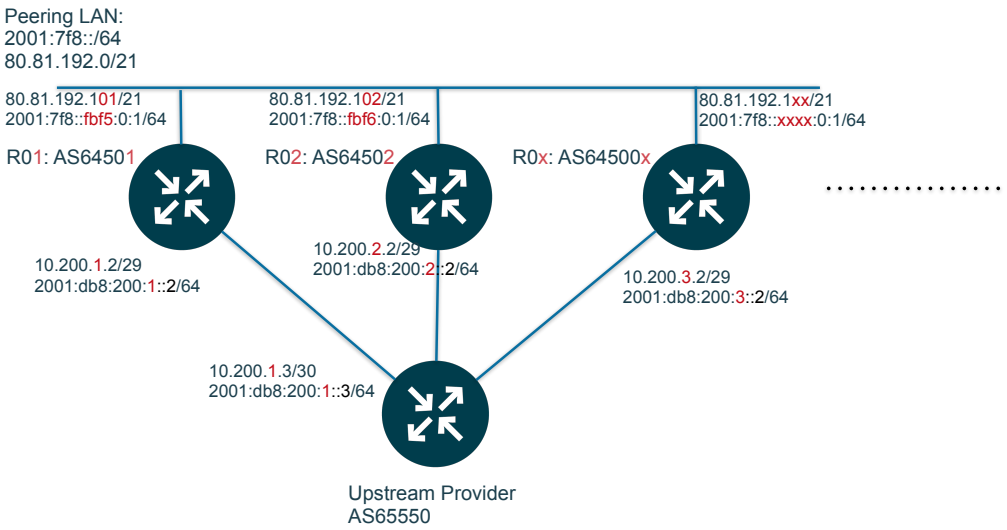



Figure 1: Network Setup

Figure 1 shows the network topology for this experiment. All students' routers connect to one upstream AS65550 (also running FRRouting in a Docker container). Additionally all studens peer with each other.

## 2 Introduction

Building correct prefix filters in BGP always has been a challenge. Often filters are either too restricted, causing prefixes to be missing from announcements or to open, causing flooding of a full BGP table to where it does not belong.

[**rfc9234**] solves this problem by assigning roles to BGP neighbors and automatic filtering according to these roles.

## 3 Task: Configure your local role for each BGP neighbor

- Check what BGP neighbors you have
- Add a statement to each BGP peer group do define your role.
-

# 4 Config statements

## 4.1 Roles

**neighbor *a.b.c.d* local-role *<role>* [ strict-mode** ] or

**neighbor *<peer-group-name>* local-role *<role>* [ strict-mode** ] sets your role in peering relationship with either one neighbor or with all neighbors in that peer group. The following roles are possible:

**customer** - you receive upstream connectivity from your neighbor. Your neighbor must have role *provider*.

**provider** - you give upstream connectivity to your neighbor. Your neighbor must have role *customer*.

**peer** - you are peering with your neighbor. Your neighbor also must have role *peer*.

**rs-client** - you are connecting to a route server as neighbor. Your neighbor must hafe role *rs-server*.

**rs-server** - you are a route server and your neighbor must have role *rs-client*.

The BGP session will not come up in case of a mismatch of roles. The session will come up if one side has no role defined *unless* the *strict* keyword is present, then both sides *must* have matching roles configured.

# 5 Show commands

## 5.1 Neighbors

**show bgp neighbor *a.b.c.d*** or

**show bgp neighbor *x:x::x*** shows all details about that neighbor and the local and remote role at the very top.

# A Upstream configuration

Configuration of the “upstream provider”, note the absense of any filtering:

```
hostname upstream
!
ip route 198.51.100.0/24 Null0
!
interface dummy0
 ip address 198.51.100.48/31
exit
!
router bgp 65550
 no bgp default ipv4-unicast
 bgp disable-ebgp-connected-route-check
 neighbor customer peer-group
 neighbor customer local-role provider
 neighbor customer-v6 peer-group
 neighbor 10.200.1.2 remote-as 64501
 neighbor 10.200.1.2 peer-group customer
 neighbor 10.200.2.2 remote-as 64502
 neighbor 10.200.2.2 peer-group customer
 neighbor 10.200.3.2 remote-as 64503
 neighbor 10.200.3.2 peer-group customer
 neighbor 10.200.4.2 remote-as 64504
 neighbor 10.200.4.2 peer-group customer
 neighbor 10.200.5.2 remote-as 64505
 neighbor 10.200.5.2 peer-group customer
 neighbor 2001:db8:200:1::2 remote-as 64501
 neighbor 2001:db8:200:1::2 peer-group customer-v6
 neighbor 2001:db8:200:2::2 remote-as 64502
 neighbor 2001:db8:200:2::2 peer-group customer-v6
```

```
 neighbor 2001:db8:200:3::2 remote-as 64503
 neighbor 2001:db8:200:3::2 peer-group customer-v6
 neighbor 2001:db8:200:4::2 remote-as 64504
 neighbor 2001:db8:200:4::2 peer-group customer-v6
 neighbor 2001:db8:200:5::2 remote-as 64505
 neighbor 2001:db8:200:5::2 peer-group customer-v6
 !
 address-family ipv4 unicast
  network 198.51.100.0/24
  neighbor customer activate
  neighbor customer soft-reconfiguration inbound
  neighbor customer route-map customer-in in
  neighbor customer route-map customer-out out
 exit-address-family
 !
 address-family ipv6 unicast
  neighbor customer-v6 activate
  neighbor customer-v6 soft-reconfiguration inbound
  neighbor customer-v6 route-map customer-in in
  neighbor customer-v6 route-map customer-out out
 exit-address-family
exit
!
ip prefix-list my-networks seq 5 permit 198.51.100.0/24 le 32
!
route-map customer-in permit 100
exit
!
route-map customer-out permit 100
exit
!
end
```

# B Slides

# Have a look at your BGP table without filtering in place

- just do a **show bgp ipv4**

What a mess in the BGP table:

- Network 198.51.100.0/24 from your upstream, you also receive via peering (and even prefer it).
- Networks originated from your peers (172.16.x.0/24) you receive from *all* peers.
- You announce everything to everybody (no filters!).

Some other show commands:

- **show bgp ipv4 neighbor 10.200.x.3** - check the neighbor role displayed
- **show bgp ipv6 neighbor 2001:db8:200:x::3**
- **sho bgp ipv4 unicast neighbors 10.200.x.3 advertised-routes** - you announce everything

# Configure Roles

- Start with a role on your upstream sessions

```
conf t

router bgp 645xx
neighbor upstream local-role customer
neighbor upstream-v6 local-role customer
```

Check the BGP table again:

- **show bgp ipv4** - once all have configured this: notice any difference?
- **show bgp neighbor 10.200.xx.3** - check the roles of yourself and your neighbor
- **show bgp ipv4 198.51.100.0/24** - do you see any attribute you have not seen before?

# Configure roles to your peers

- Now add a role to the session to your peers

```
conf t

router bgp 645xx
  neighbor peering local-role peer
  neighbor peering-v6 local-role peer
```

Check the BGP table again:

- **show bgp ipv4** - notice any difference?
- **show bgp neighbor 80.81.192.1xx** - check the roles of yourself and your neighbor
- **show bgp ipv4 172.16.xx.0/24** - do you see any attribute you have not seen before?