Otto-Friedrich-University of Bamberg

Professorship for Computer Science,
Communication Services, Telecommunication
Systems and Computer Networks

Foundation of Internet Communication

Assignment 6
Interdomain Routing Protocols

Submitted by:

Group X

Moktahid Al Faisal
Abdullah Al Mosabbir
Mohammed Mehedi Hasan
Kazi Sayef Shawgat
Sheikh Jumon Ahmed

Supervisor: Prof. Dr. Udo Krieger

Bamberg, July 19, 2020
Summer Term 2020

Contents

List of Figures

1 Exterior Routing with BGP

1. Build the topology depicted in Figure 1 with Kathara

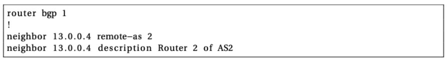
   Soluation: Done

2. Determine the addresses of the Internet DNS root servers

   Soluation: Done

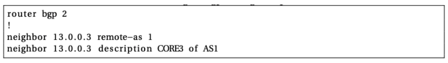
3. Configure the BGP peering between both ASes, where AS 1 is responsiblefortheOSPFnetworkandAS2forRIP

   Solution: To complete the task, we set 'bgpd = yes' in the daemons files of core3 and rip4. And also create a bgpd.conf file for both of the routers.



```
router bgp 1
!
neighbor 13.0.0.4 remote-as 2
neighbor 13.0.0.4 description Router 2 of AS2
```

Figure 1: bgpd.cong of core3

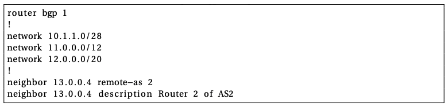


```
router bgp 2
!
neighbor 13.0.0.3 remote-as 1
neighbor 13.0.0.3 description CORE3 of AS1
```

Figure 2: bgpd.cong of rip4

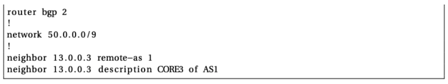
4. Adjust the announcements, so that all hosts of both ASes can reach each other

   Solution: We have modified the bgpd.conf files like below:



```
router bgp 1
!
network 10.1.1.0/28
network 11.0.0.0/12
network 12.0.0.0/20
!
neighbor 13.0.0.4 remote-as 2
neighbor 13.0.0.4 description Router 2 of AS2
```

Figure 3: bgpd.cong of core3



```
router bgp 2
!
network 50.0.0.0/9
!
neighbor 13.0.0.3 remote-as 1
neighbor 13.0.0.3 description CORE3 of AS1
```

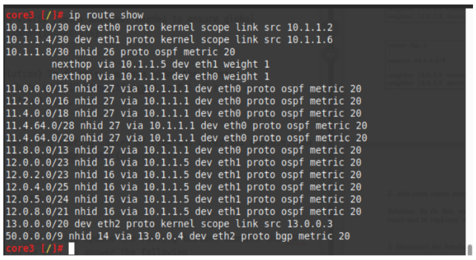
Figure 4: bgpd.cong of rip4

5. Add static routes wherever needed to ensure global connectivity

   Solution: To do this, we have added the command redistribute bgp in 'ospfd.conf ' of core3 and in ripd.conf of rip4.

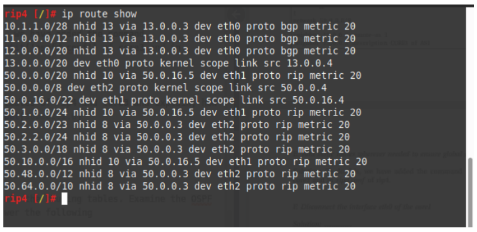
6. Inspect the routing tables of rip4 and core3 and explain, how the peering is done with BGP

   Solution: In both routers core3 and rip4, bgp sends the corresponding connected networks to its neighbor ASes. We can see the updated routing tables below.



```
core3 [/]# ip route show
10.1.1.0/30 dev eth0 proto kernel scope link src 10.1.1.2
10.1.1.4/30 dev eth1 proto kernel scope link src 10.1.1.6
10.1.1.8/30 nhid 26 proto ospf metric 20
        nexthop via 10.1.1.5 dev eth1 weight 1
        nexthop via 10.1.1.1 dev eth0 weight 1
11.0.0.0/15 nhid 27 via 10.1.1.1 dev eth0 proto ospf metric 20
11.2.0.0/16 nhid 27 via 10.1.1.1 dev eth0 proto ospf metric 20
11.4.0.0/18 nhid 27 via 10.1.1.1 dev eth0 proto ospf metric 20
11.4.64.0/28 nhid 27 via 10.1.1.1 dev eth0 proto ospf metric 20
11.4.64.0/20 nhid 27 via 10.1.1.1 dev eth0 proto ospf metric 20
11.8.0.0/13 nhid 27 via 10.1.1.1 dev eth0 proto ospf metric 20
12.0.0.0/23 nhid 16 via 10.1.1.5 dev eth1 proto ospf metric 20
12.0.2.0/23 nhid 16 via 10.1.1.5 dev eth1 proto ospf metric 20
12.0.4.0/25 nhid 16 via 10.1.1.5 dev eth1 proto ospf metric 20
12.0.5.0/24 nhid 16 via 10.1.1.5 dev eth1 proto ospf metric 20
12.0.8.0/21 nhid 16 via 10.1.1.5 dev eth1 proto ospf metric 20
13.0.0.0/20 dev eth2 proto kernel scope link src 13.0.0.3
50.0.0.0/9 nhid 14 via 13.0.0.4 dev eth2 proto bgp metric 20
core3 [/]#
```

Figure 5: Routing table of core3



```
rip4 [/]# ip route show
10.1.1.0/28 nhid 13 via 13.0.0.3 dev eth0 proto bgp metric 20
11.0.0.0/12 nhid 13 via 13.0.0.3 dev eth0 proto bgp metric 20
12.0.0.0/20 nhid 13 via 13.0.0.3 dev eth0 proto bgp metric 20
13.0.0.0/20 dev eth0 proto kernel scope link src 13.0.0.4
50.0.0.0/20 nhid 10 via 50.0.16.5 dev eth1 proto rip metric 20
50.0.0.0/8 dev eth2 proto kernel scope link src 50.0.0.4
50.0.16.0/22 dev eth1 proto kernel scope link src 50.0.16.4
50.1.0.0/24 nhid 10 via 50.0.16.5 dev eth1 proto rip metric 20
50.2.0.0/23 nhid 8 via 50.0.0.3 dev eth2 proto rip metric 20
50.2.2.0/24 nhid 8 via 50.0.0.3 dev eth2 proto rip metric 20
50.3.0.0/18 nhid 8 via 50.0.0.3 dev eth2 proto rip metric 20
50.10.0.0/16 nhid 10 via 50.0.16.5 dev eth1 proto rip metric 20
50.48.0.0/12 nhid 8 via 50.0.0.3 dev eth2 proto rip metric 20
50.64.0.0/10 nhid 8 via 50.0.0.3 dev eth2 proto rip metric 20
rip4 [/]#
```

Figure 6: ”Routing table of rip4

2 Multihoming and Redundancy

1. Add AS3 with a new BGP router as3

   Soluation: Done as asked. Added the following commands in the lab.conf file

```
as3[0]="U"
as3[1]="V"
as3[2]="W"
as3[image]="unibaktr/alpine:frr"
as3[exec]="ifconfig eth0 13.0.16.2 netmask 255.255.248.0 up"
as3[exec]="ifconfig eth1 100.0.0.2 netmask 255.255.240.0 up"
as3[exec]="ifconfig eth2 13.0.24.2 netmask 255.255.252.0 up"
```

Figure 7: configurationf for as3 router

2. Attach kili over 100.0.0.0/20 to it. Choose the IP addresses accordingly

   Solution: Added the following commands in the lab.conf file

```
kili[0]="V"
kili[image]="unibaktr/alpine:whoami"
kili[exec]="ifconfig eth0 100.0.0.1 netmask 255.255.240.0 up"
kili[exec]="ip route add default via 100.0.0.2
```

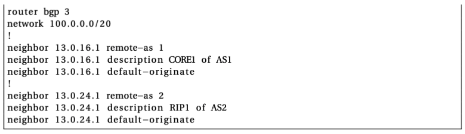
Figure 8: configuration for kili

3. Create CD U to connect as3 with core1 over the network 13.0.16.0/21

4. In the same fashion, create CD W to interconnect as3 with rip1 over the network 13.0.24.0/22. *ru*

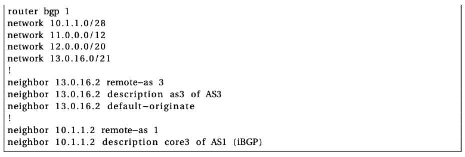
   Solution: Done as asked. See the Figure 7

5. Adjust the BGP configuration in a way, that eBGP (external BGP) is used for CDs I, U and W



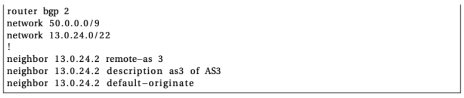
```
router bgp 3
network 100.0.0.0/20
!
neighbor 13.0.16.1 remote-as 1
neighbor 13.0.16.1 description CORE1 of AS1
neighbor 13.0.16.1 default-originate
!
neighbor 13.0.24.1 remote-as 2
neighbor 13.0.24.1 description RIP1 of AS2
neighbor 13.0.24.1 default-originate
```

Figure 9: bgpd.conf configuration for router as3



```
router bgp 1
network 10.1.1.0/28
network 11.0.0.0/12
network 12.0.0.0/20
network 13.0.16.0/21
!
neighbor 13.0.16.2 remote-as 3
neighbor 13.0.16.2 description as3 of AS3
neighbor 13.0.16.2 default-originate
!
neighbor 10.1.1.2 remote-as 1
neighbor 10.1.1.2 description core3 of AS1 (iBGP)
```

Figure 10: bgpd.conf configuration for router core1



```
router bgp 2
network 50.0.0.0/9
network 13.0.24.0/22
!
neighbor 13.0.24.2 remote-as 3
neighbor 13.0.24.2 description as3 of AS3
neighbor 13.0.24.2 default-originate
```

Figure 11: bgpd.conf configuration for router rip1

6. Additionally, use iBGP (internal BGP) to propagate learned routes inside the networks of AS 1 and AS 2 *ru*

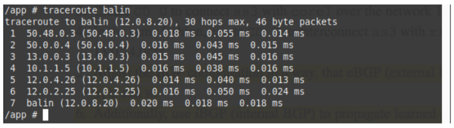
   Soluation: Done

7. Ensure connectivity of all hosts.

   Soluation: Done

8. Now, let us start a short evaluation on the path from bombur to balin.

   1) Determine the path between both nodes with traceroute.:



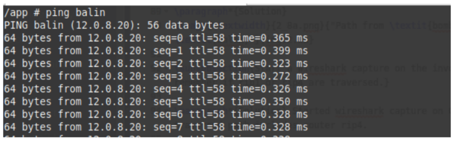
```
/app # traceroute balin
traceroute to balin (12.0.8.20), 30 hops max, 46 byte packets
 1  50.48.0.3 (50.48.0.3)  0.018 ms  0.055 ms  0.014 ms
 2  50.0.0.4 (50.0.0.4)  0.016 ms  0.043 ms  0.015 ms
 3  13.0.0.3 (13.0.0.3)  0.015 ms  0.045 ms  0.016 ms
 4  10.1.1.5 (10.1.1.5)  0.016 ms  0.038 ms  0.016 ms
 5  12.0.4.26 (12.0.4.26)  0.014 ms  0.040 ms  0.013 ms
 6  12.0.2.25 (12.0.2.25)  0.016 ms  0.050 ms  0.024 ms
 7  balin (12.0.8.20)  0.020 ms  0.018 ms  0.018 ms
/app # 
```

Figure 12:   Path from bombur to balin

2) Start a Wireshark capture on the involved CD I or W and remember if rip1 or rip4 are traversed

Solution: Started wireshark capture on CD I, as the traversed path includes router rip4.

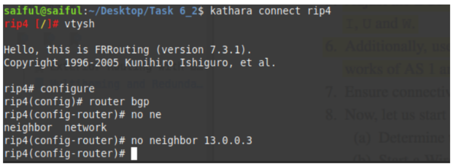
3) Start to continuously ping balin from bombur



```
/app # ping balin
PING balin (12.0.8.20): 56 data bytes
64 bytes from 12.0.8.20: seq=0 ttl=58 time=0.365 ms
64 bytes from 12.0.8.20: seq=1 ttl=58 time=0.399 ms
64 bytes from 12.0.8.20: seq=2 ttl=58 time=0.323 ms
64 bytes from 12.0.8.20: seq=3 ttl=58 time=0.272 ms
64 bytes from 12.0.8.20: seq=4 ttl=58 time=0.326 ms
64 bytes from 12.0.8.20: seq=5 ttl=58 time=0.350 ms
64 bytes from 12.0.8.20: seq=6 ttl=58 time=0.328 ms
64 bytes from 12.0.8.20: seq=7 ttl=58 time=0.328 ms
```

Figure 13:   Path from bombur to balin

4) Open a vtysh terminal on the involved RIP router and temporarily remove the BGP neighbor



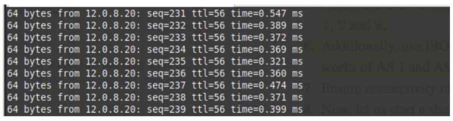
```
saiful@saiful:~/Desktop/Task 6_2$ kathara connect rip4
rip4 [/]# vtysh

Hello, this is FRRouting (version 7.3.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

rip4# configure
rip4(config)# router bgp
rip4(config-router)# no ne
neighbor  network
rip4(config-router)# no neighbor 13.0.0.3
rip4(config-router)# 
```

Figure 14:   Removing BGP neighbor from rip4

5) Wait until the ttl value of the ping changes

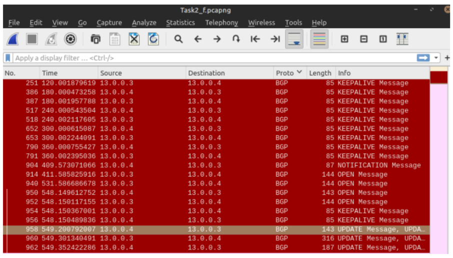


```
64 bytes from 12.0.8.20: seq=231 ttl=56 time=0.547 ms
64 bytes from 12.0.8.20: seq=232 ttl=56 time=0.389 ms
64 bytes from 12.0.8.20: seq=233 ttl=56 time=0.372 ms
64 bytes from 12.0.8.20: seq=234 ttl=56 time=0.369 ms
64 bytes from 12.0.8.20: seq=235 ttl=56 time=0.321 ms
64 bytes from 12.0.8.20: seq=236 ttl=56 time=0.360 ms
64 bytes from 12.0.8.20: seq=237 ttl=56 time=0.474 ms
64 bytes from 12.0.8.20: seq=238 ttl=56 time=0.371 ms
64 bytes from 12.0.8.20: seq=239 ttl=56 time=0.399 ms
```

Figure 15:   ttl changes after removing BGP neighbor

6) Enable the BGP neighbor again, wait for the ping to change and stop the Wireshark capture

I. Explain BGP's update mechanism based on the Wireshark capture. Solution: It can be seen from the wireshark capture, that the rip4 and core3 exchanges 'KEEPALIVE Message' between them to maintain connecivity. After removing BGP neighbor from rip4, it sends core3 a 'NOTIFICATION Message'. Then they keep exchanging 'OPEN Message'. After reattaching the neighbor, the rip4 sends 'UPDATE Message' to core3. Then core3 updates its routing information and returns an updated 'UPDATE Message' to rip4, after which rip4 does the same thing again.


Task2_f.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Proto	Length	Info
251	120.001879619	13.0.0.3	13.0.0.4	BGP	85	KEEPALIVE Message
386	180.000473258	13.0.0.4	13.0.0.3	BGP	85	KEEPALIVE Message
387	180.001957788	13.0.0.3	13.0.0.4	BGP	85	KEEPALIVE Message
517	240.000543504	13.0.0.4	13.0.0.3	BGP	85	KEEPALIVE Message
518	240.002117605	13.0.0.3	13.0.0.4	BGP	85	KEEPALIVE Message
652	300.000615087	13.0.0.4	13.0.0.3	BGP	85	KEEPALIVE Message
653	300.002244091	13.0.0.3	13.0.0.4	BGP	85	KEEPALIVE Message
790	360.000755427	13.0.0.4	13.0.0.3	BGP	85	KEEPALIVE Message
791	360.002395036	13.0.0.3	13.0.0.4	BGP	85	KEEPALIVE Message
904	409.573071066	13.0.0.4	13.0.0.3	BGP	87	NOTIFICATION Message
914	411.585825916	13.0.0.3	13.0.0.4	BGP	144	OPEN Message
940	531.586686678	13.0.0.3	13.0.0.4	BGP	144	OPEN Message
950	548.149612752	13.0.0.4	13.0.0.3	BGP	143	OPEN Message
952	548.150117155	13.0.0.3	13.0.0.4	BGP	144	OPEN Message
954	548.150367001	13.0.0.3	13.0.0.4	BGP	85	KEEPALIVE Message
956	548.150489836	13.0.0.4	13.0.0.3	BGP	85	KEEPALIVE Message
958	549.200792007	13.0.0.4	13.0.0.3	BGP	143	UPDATE Message, UPDA…
960	549.301340491	13.0.0.3	13.0.0.4	BGP	316	UPDATE Message, UPDA…
962	549.352422286	13.0.0.4	13.0.0.3	BGP	187	UPDATE Message, UPDA…

Figure 16: Wireshark capture on CD I

References