Otto-Friedrich-University Bamberg

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

Foundation of Internet Communication

Assignment-06 :IInterdomain Routing Protocols

Submitted by:
Group J

Reem Eslam Mohamed Mekky Khalil
Shivasharan Reddy
Azar Ghadami
Reema Miranda
Manjunath B Marigoudar

Supervisor: Prof. Dr. Udo Krieger

Bamberg, July 19, 2020
Summer Term 2020

Contents

List of Figures

Chapter 1

Border Gateway Protocol (BGP)

1.1 Exterior Routing with BGP

- The Border Gateway Protocol (BGP) is the protocol backing the core routing decisions.

- On the Internet. It maintains a table of IP networks or prefixes which designate network reachability among autonomous systems (AS). It is described as a path vector protocol.

- BGP does not use traditional Interior Gateway Protocol (IGP) metrics, but makes routing decisions based on path, network policies and/or rule sets

1.2 Build the topology with Kathara.

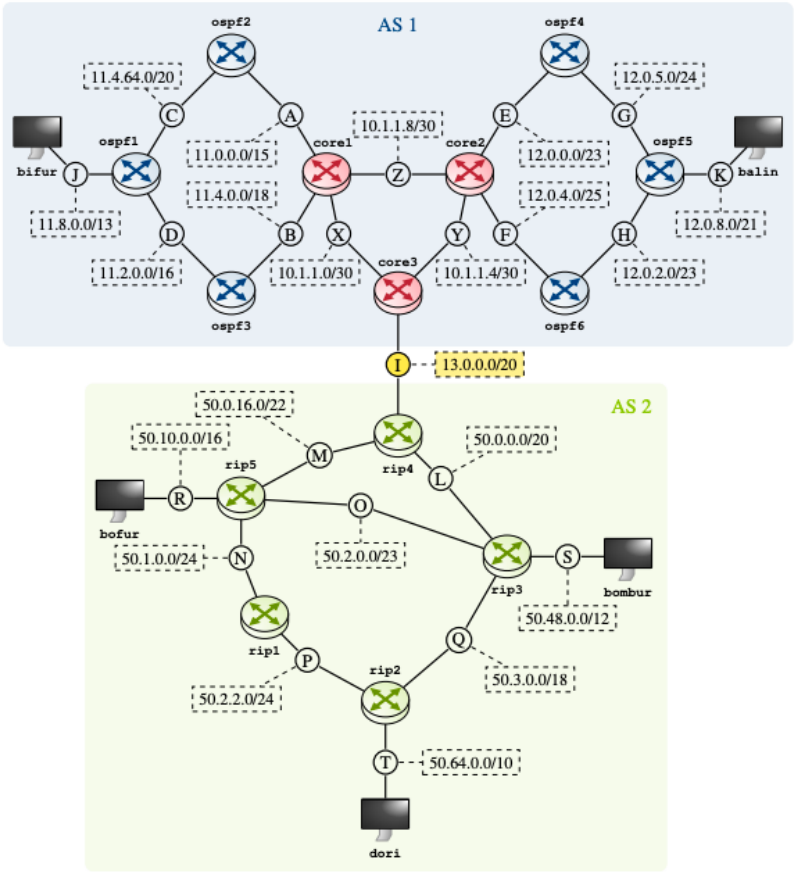

Figure 1.1: BGP Topology

1.3 Building the topology for BGP network

1.3.1 Step 1. Remove kili at the edge of each network and connect core3 with rip4 over CD I

- Combine the two lab.conf files of previous networks into a single lab file and remove killi in the lab.conf file.

1.3.2 Choose IP address of the network 13.0.0.0/20 and add CD I to rip4

- IP address in rip4.startup file which connects to core3 over collision domain 13.0.0.0/20
- ip addr add 13.0.0.4/20 brd + dev eth0

Figure 1.2: rip4.startup file

- IP address in core3.startup file which connects to core3 over collision domain 13.0.0.0/20
- ip addr add 13.0.0.3/20 brd + dev eth2

Figure 1.3: Core3.startup file

1.4 Configure the BGP peering between both Areas, where AS 1 is responsible for the OSPF network and AS 2 for RIP

1.5 BGP configuration in core3

- We must create bgpd.conf file in core3

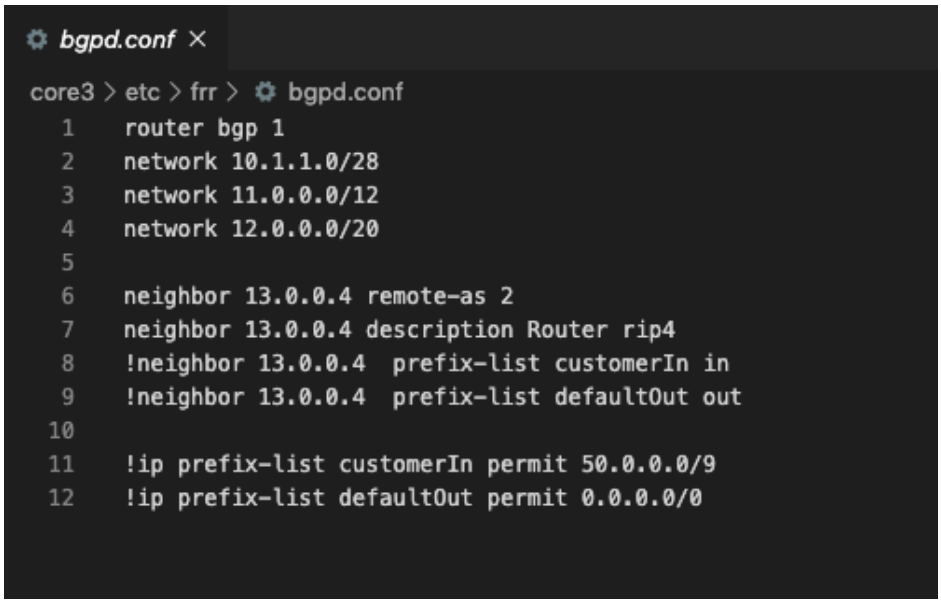

Figure 1.4: BGPD.conf file of Core3

1.5.1 router bgp 1 : this specifies that the core3 is in area one . That is OSPF network is completely considered area 1

1.5.2 IP Addresses of OSPF network which includes IP address of all 3 areas(networks) of the OSPF network

- network 10.1.1.0/28 : Backbone area 3.3.3.3(core 1 core2 core3)
- network 11.0.0.0/12 : Area 1.1.1.1 (bgp1 bgp2 bgp3)
- network 12.0.0.0/20 : Area 2.2.2.2 (bgp4 bgp5 bgp6)

1.5.3 Neighbour of core3

- neighbor 13.0.0.4 remote-as 2 : Specifies the ip of rip4 and considers its area as 2
- neighbor 13.0.0.4 description Router rip4 : Specifies the name of the neighbour that is rip4

1.6 BGP configuration in rip4

- We must create bgpd.conf file in rip4

Figure 1.5: BGPD.conf file of rip4

1.6.1 router bgp 2 : this specifies that the rip4 is in area 2. That is RIP network is completely considered as area 2

1.6.2 IP Addresses of RIP network which includes IP address of all areas(networks) of the RIP network

- network 50.0.0.0/8 : Includes all the RIP network routers and host

1.6.3 Neighbour of rip4

- neighbor 13.0.0.3 remote-as 1 : Specifies the ip of core3 and considers its area as 1

- neighbor 13.0.0.3 description Router core3 : Specifies the name of the neighbour that is core3

1.7 Adjust the announcements, so that all hosts of both Areas can reach each other

1.7.1 Declare BGPD in both daemon file

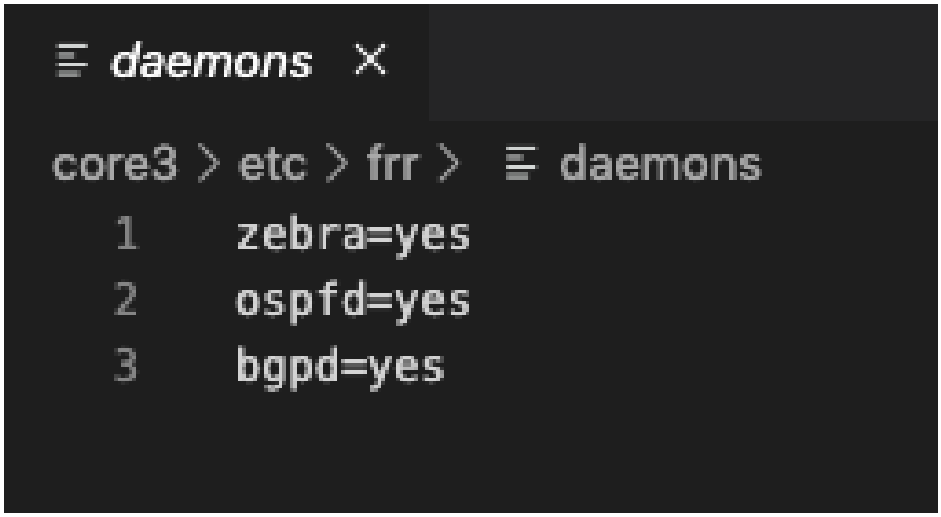

Figure 1.6: Deamon file

1.8 Adding static routes wherever needed to ensure global connectivity.

- In OSPFD file of Core3 remove interfaces and areas as it is already declared in bgpd file

- Declare redistribute bgp

Figure 1.7: Deamon file

- In RIPD file of rip4 mention the network 50.0.0.0/20 and network 50.0.16.0/22

- Declare redistribute bgp

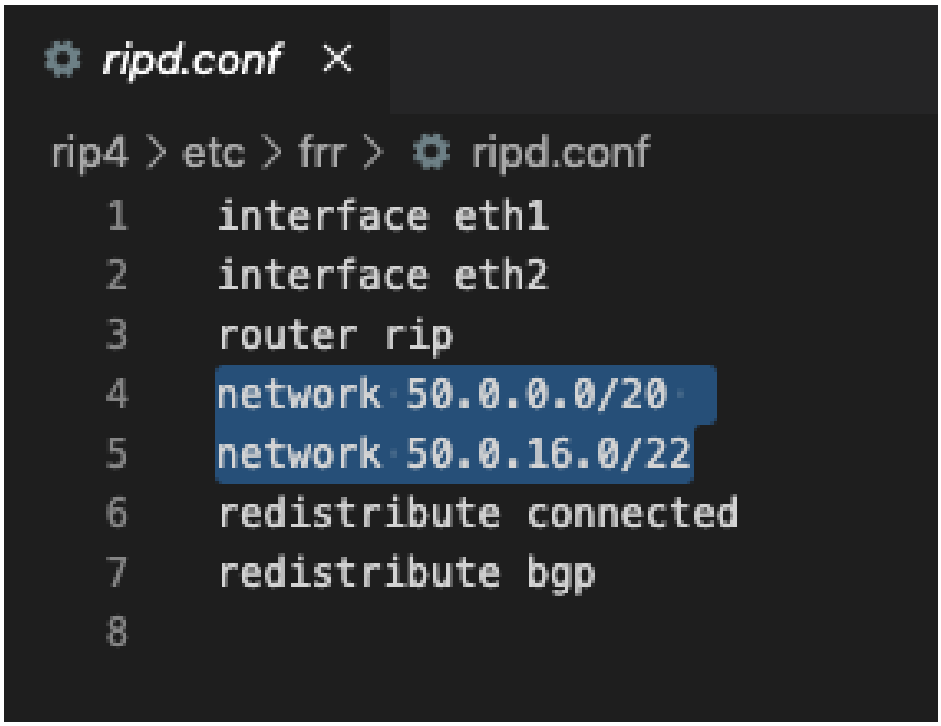

Figure 1.8: RIPD file

1.9 Checking connectivity among all hosts of both Areas

- Checking connectivity between Bifur and Bombur (Bifur belongs to OSPFD and Bombur belongs to RIPD networks)

```
/app # ping 50.48.0.20
PING 50.48.0.20 (50.48.0.20): 56 data bytes
64 bytes from 50.48.0.20: seq=15 ttl=58 time=0.697 ms
64 bytes from 50.48.0.20: seq=16 ttl=58 time=0.409 ms
64 bytes from 50.48.0.20: seq=17 ttl=58 time=0.373 ms
64 bytes from 50.48.0.20: seq=18 ttl=58 time=0.467 ms
64 bytes from 50.48.0.20: seq=19 ttl=58 time=0.425 ms
64 bytes from 50.48.0.20: seq=20 ttl=58 time=0.311 ms
^C
--- 50.48.0.20 ping statistics ---
21 packets transmitted, 6 packets received, 71% packet loss
round-trip min/avg/max = 0.311/0.447/0.697 ms
/app #
```

Figure 1.9: Connectivity between Bifur and Bombur

- Checking connectivity between Balin and Dori

```
Reemas-MacBook-Air:Config Files BGPpart1 reemamiranda$ kathara connect balin
/app # ping 50.64.0.30
PING 50.64.0.30 (50.64.0.30): 56 data bytes
64 bytes from 50.64.0.30: seq=0 ttl=57 time=0.907 ms
64 bytes from 50.64.0.30: seq=1 ttl=57 time=0.817 ms
64 bytes from 50.64.0.30: seq=2 ttl=57 time=0.399 ms
^C
--- 50.64.0.30 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.399/0.707/0.907 ms
/app #
```

Figure 1.10: Connectivity between Balin and Dori

- Checking connectivity between Balin and Bombur

```
Reemas-MacBook-Air:Config Files BGPpart1 reemamiranda$ kathara connect balin
/app # ping 50.48.0.20
PING 50.48.0.20 (50.48.0.20): 56 data bytes
64 bytes from 50.48.0.20: seq=0 ttl=58 time=0.297 ms
64 bytes from 50.48.0.20: seq=1 ttl=58 time=0.444 ms
64 bytes from 50.48.0.20: seq=2 ttl=58 time=0.470 ms
^C
--- 50.48.0.20 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.297/0.403/0.470 ms
/app #
```

Figure 1.11: Connectivity between Balin and Bombur

1.10 Inspecting the routing tables of rip4 and core3 and explain, how the peering is done with BGP

1.10.1 Routing Table of Rip4

```
rip4 [/]# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.1.1.0        13.0.0.3        255.255.255.240 UG    20     0        0 eth0
11.0.0.0        13.0.0.3        255.240.0.0     UG    20     0        0 eth0
12.0.0.0        13.0.0.3        255.255.240.0   UG    20     0        0 eth0
13.0.0.0        *               255.255.240.0   U     0      0        0 eth0
50.0.0.0        *               255.255.240.0   U     0      0        0 eth2
50.0.16.0       *               255.255.252.0   U     0      0        0 eth1
50.1.0.0        50.0.0.3        255.255.255.0   UG    20     0        0 eth2
50.2.0.0        50.0.0.3        255.255.254.0   UG    20     0        0 eth2
50.2.2.0        50.0.0.3        255.255.255.0   UG    20     0        0 eth2
50.3.0.0        50.0.0.3        255.255.192.0   UG    20     0        0 eth2
50.48.0.0       50.0.0.3        255.240.0.0     UG    20     0        0 eth2
50.64.0.0       50.0.0.3        255.192.0.0     UG    20     0        0 eth2
```

Figure 1.12: Routing Table of Rip4

- In Rip4 routing table the OSPFD network in connected from core3 which is the bgpd router
- RIP4 is connecting to 10.1.1.0 ip address which covers all the ip address of the backbone area (core1,core2,core3) of OSPfD network through the gateway 13.0.0.3 which is the ip address of Core3

```
rip4 [/]# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.1.1.0        13.0.0.3        255.255.255.240 UG    20     0        0 eth0
11.0.0.0        13.0.0.3        255.240.0.0     UG    20     0        0 eth0
12.0.0.0        13.0.0.3        255.255.240.0   UG    20     0        0 eth0
```

Figure 1.13: Connecting from rip4 to core3 network

- Similarly RIP4 is connecting to 11.0.0.0 and 12.0.0.0 ip address which covers all the ip address of Bifur, Ospf1, Ospf2 ,Ospf3 ,core1 and Core2,

Ospf4, Ospf5, Ospf6, Balin of OSPFD network through the gateway 13.0.0.3 which is the ip address of Core3

- In Rip4 routing table the RIP network in connected from rip4 which is the bgpd router via rip3 as gateway.
- RIP4 is connecting to all the collision domain ip address which covers all the ip address of RiP network via rip3(50.0.0.3).

```
50.1.0.0        50.0.0.3        255.255.255.0   UG    20     0        0 eth2
50.2.0.0        50.0.0.3        255.255.254.0   UG    20     0        0 eth2
50.2.2.0        50.0.0.3        255.255.255.0   UG    20     0        0 eth2
50.3.0.0        50.0.0.3        255.255.192.0   UG    20     0        0 eth2
50.48.0.0       50.0.0.3        255.240.0.0     UG    20     0        0 eth2
50.64.0.0       50.0.0.3        255.192.0.0     UG    20     0        0 eth2
rip4 [/]#
```

Figure 1.14: Connecting from rip4 to RIPD network

1.10.2 Routing Table of Core3

```
core3 [/]# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.1.1.0        *               255.255.255.252 U     0      0        0 eth0
10.1.1.4        *               255.255.255.252 U     0      0        0 eth1
10.1.1.8        10.1.1.1        255.255.255.252 UG    20     0        0 eth0
11.0.0.0        10.1.1.1        255.254.0.0     UG    20     0        0 eth0
11.2.0.0        10.1.1.1        255.255.0.0     UG    20     0        0 eth0
11.4.0.0        10.1.1.1        255.255.192.0   UG    20     0        0 eth0
11.4.64.0       10.1.1.1        255.255.240.0   UG    20     0        0 eth0
11.8.0.0        10.1.1.1        255.248.0.0     UG    20     0        0 eth0
12.0.0.0        10.1.1.5        255.255.254.0   UG    20     0        0 eth1
12.0.2.0        10.1.1.5        255.255.254.0   UG    20     0        0 eth1
12.0.4.0        10.1.1.5        255.255.255.128 UG    20     0        0 eth1
12.0.5.0        10.1.1.5        255.255.255.0   UG    20     0        0 eth1
12.0.8.0        10.1.1.5        255.255.248.0   UG    20     0        0 eth1
13.0.0.0        *               255.255.240.0   U     0      0        0 eth2
50.0.0.0        13.0.0.4        255.0.0.0       UG    20     0        0 eth2
core3 [/]#
```

Figure 1.15: Routing Table of Core3

- In Core3 routing table the RIP network is connected to Bgpd network(50.0.0.0) from rip4 which is the Bgpd router via rip4(13.0.0.4) as gateway.

```
50.0.0.0        13.0.0.4        255.0.0.0       UG    20     0        0 eth2
core3 [/]#
```

Figure 1.16: Connecting between OSPFD network and RIPD network

- In Core3 routing table the all the OSFD routers and host are connected as the collision domain ip address is taken of the entire network via 10.1.1.1 (core1) and 10.1.1.5(core5) as gateway respectively.
- Core1 Ospfd1 ospfd2 ospfd3 and Bifur are connected via 10.1.1.1 as gatway.
- Core2 Ospfd4,Ospfd5,Ospfd6,Balin are connected via 10.1.1.5 as gateway.

```
core3 [/]# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.1.1.0        *               255.255.255.252 U     0      0        0 eth0
10.1.1.4        *               255.255.255.252 U     0      0        0 eth1
10.1.1.8        10.1.1.1        255.255.255.252 UG    20     0        0 eth0
11.0.0.0        10.1.1.1        255.254.0.0     UG    20     0        0 eth0
11.2.0.0        10.1.1.1        255.255.0.0     UG    20     0        0 eth0
11.4.0.0        10.1.1.1        255.255.192.0   UG    20     0        0 eth0
11.4.64.0       10.1.1.1        255.255.240.0   UG    20     0        0 eth0
11.8.0.0        10.1.1.1        255.248.0.0     UG    20     0        0 eth0
12.0.0.0        10.1.1.5        255.255.254.0   UG    20     0        0 eth1
12.0.2.0        10.1.1.5        255.255.254.0   UG    20     0        0 eth1
12.0.4.0        10.1.1.5        255.255.255.128 UG    20     0        0 eth1
12.0.5.0        10.1.1.5        255.255.255.0   UG    20     0        0 eth1
12.0.8.0        10.1.1.5        255.255.248.0   UG    20     0        0 eth1
13.0.0.0        *               255.255.240.0   U     0      0        0 eth2
```

Figure 1.17: Connecting Core3 to OSPFD network

- And this is how the bgpd routers helps to connect the two networks and hence we can ping from any host of Ospfd network to host of RIPD network

Chapter 2

Multihoming and Redundancy

In Section one we have created two different networks and connected them through collision domain I. It worked fine however integrating that in real life most probably will cause a lot of problems as we will have a single point of failure. In this section we will work on this problem by introducing third Area called AS 3 to have another route in case failure occurs at collision domain I.

Our goal will be 2.1

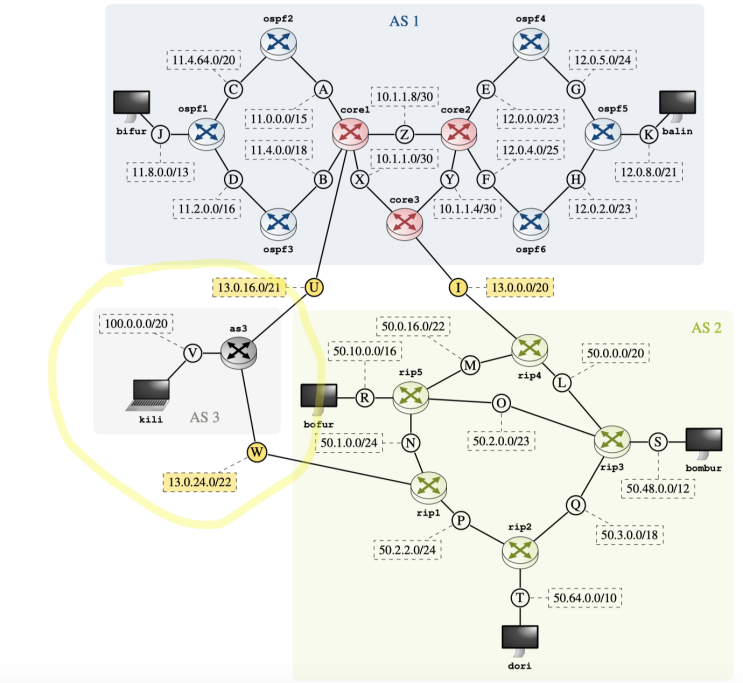

Figure 2.1: Our Network with extra route

2.1 Structuring AS3 network to offer another route

Lets Now Go through steps to create the extra area in figure 2.1

2.1.1 Configure BGB router as3 and Connect as3 with core1 through collision domain U and Connect as3 with rip1 through collision domain W

```
as3[0]="V"
as3[1]="U"
as3[2]="W"
as3[image]="unibaktr/alpine:frr"
```

Figure 2.2: Adding As3 in labConf file. and connect it with three collison domains V, U and W

as3.startup

```
1   ip addr add 100.0.0.2/20 brd + dev eth0
2   ip addr add 13.0.16.3/21 brd + dev eth1
3   ip addr add 13.0.24.3/22 brd + dev eth2
4   frrinit.sh start
```

Figure 2.3: As3 startup file and configure address for each collision domain.

bgpd.conf ×

as3 > etc > frr > bgpd.conf

```
1   router bgp 3
2   network 100.0.0.0/20
3   neighbor 13.0.16.1 remote-as 1
4   neighbor 13.0.16.1 description Router core1
5   neighbor 13.0.24.2 remote-as 2
6   neighbor 13.0.24.2 description Router rip1
```

Figure 2.4: Setuping as3 router as BGP router.

2.2 Add kili to the new network and connect it to bgp router as3

```
kili[0]="V"
kili[image]="unibaktr/alpine"
```

Figure 2.5: Adding kili to labConf file and attaching it to collision domain V.

```
ip addr add  100.0.0.1/20 brd + dev eth0
ip route add default via 100.0.0.2
```

Figure 2.6: Configuring kili startup file and setting its default router to be as3.

2.3 Connecting network 1 with the new network by connecting core 1 to as3 bgp router through U interface and configuring IBGP in area one for core 1 and core 3

```
1  ip addr add 10.1.1.9/30 brd + dev eth0
2  ip addr add 10.1.1.1/30 brd + dev eth1
3  ip addr add 11.0.0.1/15 brd + dev eth2
4  ip addr add 11.4.0.1/18 brd + dev eth3
5  ip addr add 13.0.16.1/21 brd + dev eth4
6
7  ip addr add 1.0.0.1/32 dev lo:1
8  frrinit.sh start
```

Figure 2.7: Updating Core1 startup file.

```
1  router bgp 1
2  network 10.1.1.0/28
3  network 11.0.0.0/12
4  network 12.0.0.0/20
5
6  neighbor 13.0.16.3 remote-as 3
7  neighbor 13.0.16.3 description Router as3
8  neighbor 1.0.0.3 remote-as 1
9  neighbor 1.0.0.3 description router core3
10 neighbor 1.0.0.3 update-source 1.0.0.1
11
12
```

Figure 2.8: Configuring BGP to core1 and configuring IBGP.

```
1  ip addr add 10.1.1.9/30 brd + dev eth0
2  ip addr add 10.1.1.1/30 brd + dev eth1
3  ip addr add 11.0.0.1/15 brd + dev eth2
4  ip addr add 11.4.0.1/18 brd + dev eth3
5  ip addr add 13.0.16.1/21 brd + dev eth4
6
7  ip addr add 1.0.0.1/32 dev lo:1
8  frrinit.sh start
```

Figure 2.9: Adding iBGP in core3 bgp configurations

2.4 Connecting network 2 with the new network by connecting rip3 to as3 bgp router through W interface.

rip3.startup

```
ip addr add 50.0.0.3/20 brd + dev eth0
ip addr add 50.2.0.3/23 brd + dev eth1
ip addr add 50.3.0.3/18 brd + dev eth2
ip addr add 50.48.0.3/12 brd + dev eth3
frrinit.sh start
```

Figure 2.10: Updating Rip3 startup file.

```
router bgp 2
network 13.0.24.2
neighbor 13.0.24.3 remote-as 1
neighbor 13.0.24.3 description Router as3


neighbor 2.0.0.4 remote-as 2
neighbor 2.0.0.4 description router rip4
neighbor 2.0.0.4 update-source 2.0.0.1
```

Figure 2.11: Updating Rip3 bgp configurations and configuring IBGP

rip4 > etc > frr > bgpd.conf

```
router bgp 2
network 50.0.0.0/8
neighbor 13.0.0.3 remote-as 1
neighbor 13.0.0.3 description Router core3


neighbor 2.0.0.1 remote-as 2
neighbor 2.0.0.1 description router rip1
neighbor 2.0.0.1 update-source 2.0.0.4
```

Figure 2.12: Adding iBGP in rip4 bgp configurations

2.5 Testing connection between neworks

2.5.1 Sending from Area 1 to Area 2 and Area 3

```
/app # ping 50.64.0.30
PING 50.64.0.30 (50.64.0.30): 56 data bytes
64 bytes from 50.64.0.30: seq=0 ttl=57 time=4.922 ms
64 bytes from 50.64.0.30: seq=1 ttl=57 time=0.423 ms
64 bytes from 50.64.0.30: seq=2 ttl=57 time=0.305 ms
64 bytes from 50.64.0.30: seq=3 ttl=57 time=0.371 ms
64 bytes from 50.64.0.30: seq=4 ttl=57 time=0.272 ms
64 bytes from 50.64.0.30: seq=5 ttl=57 time=0.456 ms
64 bytes from 50.64.0.30: seq=6 ttl=57 time=0.326 ms
64 bytes from 50.64.0.30: seq=7 ttl=57 time=0.309 ms
64 bytes from 50.64.0.30: seq=8 ttl=57 time=0.516 ms
64 bytes from 50.64.0.30: seq=9 ttl=57 time=0.261 ms
64 bytes from 50.64.0.30: seq=10 ttl=57 time=0.417 ms
64 bytes from 50.64.0.30: seq=11 ttl=57 time=0.323 ms
64 bytes from 50.64.0.30: seq=12 ttl=57 time=0.298 ms
```

Figure 2.13: pinging from bifur to dori.

```
(base) Reems-MacBook-Pro:Config Files BGPpart2 reemeslam$ kathara connect bifur
/app # ping 100.0.0.1
PING 100.0.0.1 (100.0.0.1): 56 data bytes
64 bytes from 100.0.0.1: seq=0 ttl=60 time=5.023 ms
64 bytes from 100.0.0.1: seq=1 ttl=60 time=0.363 ms
64 bytes from 100.0.0.1: seq=2 ttl=60 time=0.322 ms
64 bytes from 100.0.0.1: seq=3 ttl=60 time=0.211 ms
64 bytes from 100.0.0.1: seq=4 ttl=60 time=0.194 ms
64 bytes from 100.0.0.1: seq=5 ttl=60 time=0.218 ms
64 bytes from 100.0.0.1: seq=6 ttl=60 time=0.195 ms
64 bytes from 100.0.0.1: seq=7 ttl=60 time=0.345 ms
64 bytes from 100.0.0.1: seq=8 ttl=60 time=0.197 ms
64 bytes from 100.0.0.1: seq=9 ttl=60 time=0.267 ms
^C
--- 100.0.0.1 ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 0.194/0.733/5.023 ms
```

Figure 2.14: pinging from bifur to kili.

2.5.2 Sending from Area 2 to Area 1 and Area 3

```
/app # ping 11.8.0.10
PING 11.8.0.10 (11.8.0.10): 56 data bytes
64 bytes from 11.8.0.10: seq=0 ttl=57 time=2.517 ms
64 bytes from 11.8.0.10: seq=1 ttl=57 time=0.256 ms
64 bytes from 11.8.0.10: seq=2 ttl=57 time=0.406 ms
64 bytes from 11.8.0.10: seq=3 ttl=57 time=2.360 ms
64 bytes from 11.8.0.10: seq=4 ttl=57 time=0.503 ms
64 bytes from 11.8.0.10: seq=5 ttl=57 time=0.383 ms
64 bytes from 11.8.0.10: seq=6 ttl=57 time=0.311 ms
^C
--- 11.8.0.10 ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 0.256/0.962/2.517 ms
/app #
```

Figure 2.15: pinging from dori to bifur.

```
(base) Reems-MacBook-Pro:Config Files BGPpart2 reemeslam$ kathara connect dori
/app # ping 100.0.0.1
PING 100.0.0.1 (100.0.0.1): 56 data bytes
64 bytes from 100.0.0.1: seq=0 ttl=58 time=0.829 ms
64 bytes from 100.0.0.1: seq=1 ttl=58 time=0.385 ms
64 bytes from 100.0.0.1: seq=2 ttl=58 time=0.361 ms
64 bytes from 100.0.0.1: seq=3 ttl=58 time=2.153 ms
64 bytes from 100.0.0.1: seq=4 ttl=58 time=0.376 ms
^C
--- 100.0.0.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.361/0.820/2.153 ms
```

Figure 2.16: pinging from dori to kili.

2.5.3 Sending from Area 3 to Area 2 and Area 1

```
kili [/]# ping 50.64.0.30
PING 50.64.0.30 (50.64.0.30): 56 data bytes
64 bytes from 50.64.0.30: seq=0 ttl=58 time=2.013 ms
64 bytes from 50.64.0.30: seq=1 ttl=58 time=0.487 ms
64 bytes from 50.64.0.30: seq=2 ttl=58 time=0.228 ms
64 bytes from 50.64.0.30: seq=3 ttl=58 time=0.221 ms
64 bytes from 50.64.0.30: seq=4 ttl=58 time=0.282 ms
64 bytes from 50.64.0.30: seq=5 ttl=58 time=0.342 ms
64 bytes from 50.64.0.30: seq=6 ttl=58 time=0.252 ms
```

Figure 2.17: pinging from kili to dori.

```
(base) Reems-MacBook-Pro:Config Files BGPpart2 reemeslam$ kathara connect kili
kili [/]# ping 11.8.0.10
PING 11.8.0.10 (11.8.0.10): 56 data bytes
64 bytes from 11.8.0.10: seq=0 ttl=60 time=0.459 ms
64 bytes from 11.8.0.10: seq=1 ttl=60 time=0.461 ms
64 bytes from 11.8.0.10: seq=2 ttl=60 time=0.363 ms
64 bytes from 11.8.0.10: seq=3 ttl=60 time=0.310 ms
64 bytes from 11.8.0.10: seq=4 ttl=60 time=0.362 ms
64 bytes from 11.8.0.10: seq=5 ttl=60 time=0.275 ms
ç64 bytes from 11.8.0.10: seq=6 ttl=60 time=0.194 ms
```

Figure 2.18: pinging from kili to bifur.

2.6 Evaluation of whole network on the path from bombur to balin.

2.6.1 Determine the path between both nodes with traceroute.

```
(base) Reems-MacBook-Pro:Config Files BGPpart2 reemeslam$ kathara connect bombur
/app # traceroute 12.0.8.20
traceroute to 12.0.8.20 (12.0.8.20), 30 hops max, 46 byte packets
 1  50.48.0.3 (50.48.0.3)  0.033 ms  0.436 ms  0.083 ms
 2  50.0.0.4 (50.0.0.4)  0.068 ms  0.219 ms  0.282 ms
 3  13.0.0.3 (13.0.0.3)  0.066 ms  0.102 ms  0.046 ms
 4  10.1.1.5 (10.1.1.5)  0.032 ms  0.066 ms  0.022 ms
 5  12.0.0.24 (12.0.0.24)  0.020 ms  0.063 ms  0.020 ms
 6  12.0.2.25 (12.0.2.25)  0.026 ms  0.120 ms  0.047 ms
 7  12.0.8.20 (12.0.8.20)  0.037 ms  0.114 ms  0.024 ms
```

Figure 2.19: Traceroute from bombur to balin

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

In this section we are supposed to capture traffic on collision domain I or W. There is a problem in capturing by wireshark in macbook from Kathara lab so I tried to use tcpdump but unfortunately the image doesnot recognise tcpdump. hence we could not attach the required screenshot in this section.

2.8 Checking TTL change

2.8.1 Contentiously ping balin from bombur

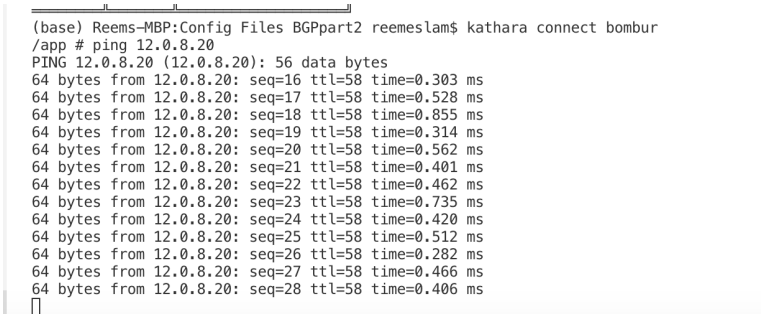

Figure 2.20: Contenious ping from Bombur to Bali

2.8.2 Open Vtysh terminal in Rip4 and shutdown neighbor core3

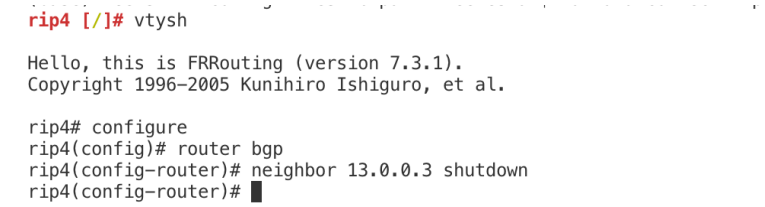

Figure 2.21: Shutdown core3 as neighbor from rip4 BGP

2.8.3 Checking TTL change

Ttl did not change and ping have stopped which should not be the case but I do not know why it stopped

```
64 bytes from 12.0.8.20: seq=69 ttl=58 time=0.670 ms
64 bytes from 12.0.8.20: seq=70 ttl=58 time=0.603 ms
64 bytes from 12.0.8.20: seq=71 ttl=58 time=0.692 ms
64 bytes from 12.0.8.20: seq=72 ttl=58 time=0.220 ms
64 bytes from 12.0.8.20: seq=73 ttl=58 time=1.322 ms
64 bytes from 12.0.8.20: seq=74 ttl=58 time=0.359 ms
64 bytes from 12.0.8.20: seq=75 ttl=58 time=0.255 ms
64 bytes from 12.0.8.20: seq=76 ttl=58 time=0.420 ms
64 bytes from 12.0.8.20: seq=77 ttl=58 time=0.451 ms
```

Figure 2.22: Pinging have stopped instead of Ttl change !

2.8.4 Enable core3 again as neighbor in rip4

```
rip4(config-router)# neighbor 13.0.0.3 remote-as 1
rip4(config-router)# neighbor 13.0.0.3 description Router core3
```

Figure 2.23: Enable core3 as neighbor in rip4

There is a problem in capturing by wireshark in macbook from Kathara lab so I tried to use tcpdump but unfortunately the image doesnot recognise tcpdump. hence we could not attach the required screenshot for wireshark in this section.