HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION TECHNOLOGY AND COMMUNICATION

---------------------o0o--------------------

Network software architecture report

Topic: Set up internet simulation network system

Supervisor: **Dr. Pham Huy Hoang**

Student Name: **Nguyen Minh Tuan** 20194876
Nguyen Thu Hieu 20194761

Ha Noi, Jan 2024

Table of contents

I) Introduction

The simulated network architecture was constructed with meticulous attention to detail, comprising three Autonomous Systems (AS). Within each AS, we employed Interior Gateway Protocol (IGP) routing to facilitate efficient communication between routers. Bridging the ASs was achieved through the strategic implementation of Border Gateway Protocol (BGP), a key protocol that governs the interconnection of different ASs on the global Internet. This emulation not only required a profound understanding of routing algorithms and protocols but also demanded the orchestration of BGP configurations for seamless communication across the diverse AS landscape.

The intricacies of IP address planning were a focal point in our design, as we navigated the allocation of addresses to ensure not only uniqueness but also optimal routing and scalability. This critical aspect of the project underscored the importance of meticulous planning in real-world networking scenarios, where IP addresses serve as the linchpin for identifying and routing data across a myriad of interconnected devices.

Furthermore, our scope extended to the integration of workstations in Home/Office networks, emphasizing the practicality of our simulation. The connectivity of these workstations within the simulated ASs was a testament to the effectiveness of our IP address planning and routing configurations, mirroring the intricacies faced in actual network deployments.

In exploring the nuanced decision-making process in inter-ISP peering, we confronted the challenge of optimizing connectivity between Internet Service Providers (ISPs) while considering the role of a Tier 1 network. The strategic use of BGP policies emerged as a pivotal aspect, allowing us to articulate preferences for direct peering or routing through the intermediate Tier 1 network.

As the project reached its zenith, considerations extended beyond mere connectivity to encompass advanced networking features. The choice between multicast services or Quality of Service (QoS) for Home/Office networks served as a capstone, showcasing the versatility and real-world applicability of our simulated infrastructure.

In essence, this project was not merely an exercise in network simulation but a comprehensive exploration of the multifaceted challenges inherent in designing and managing a robust Internet backbone. The ensuing report chronicles our journey through the intricacies of network design, implementation, and optimization, providing a holistic view of our endeavors in the realm of advanced networking.

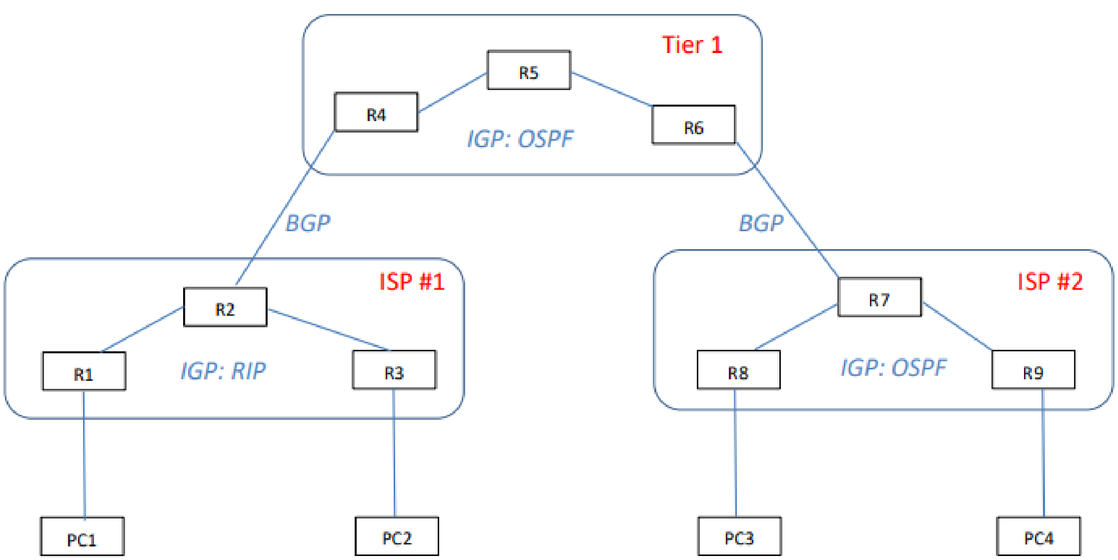
II) Problem

Utilize the knowledge acquired in the course to fulfill the following requirements. Use one of the simulation environments mentioned in the course content.

1. Simulate an Internet backbone network consisting of 3 AS: a Tier 1 intermediate level connecting 2 ISPs (these 2 ISPs do not connect directly). In each AS, use IGP routing and have a minimum of 3 routers. Connect the ASs using BGP.
2. Connect each ISP to two Home/Office networks. Each Home/Office network should include one workstation.
3. Plan the IP address scheme for the entire system and configure IP addresses for devices, ensuring that workstations can ping each other.
4. The ISPs want to establish direct peering between them and use BGP policy to decide whether to choose peering or go through the Tier 1 network. Provide a solution.
5. Workstations in Home/Office networks require either multicast services or QoS (choose one). Propose a plan for implementation.

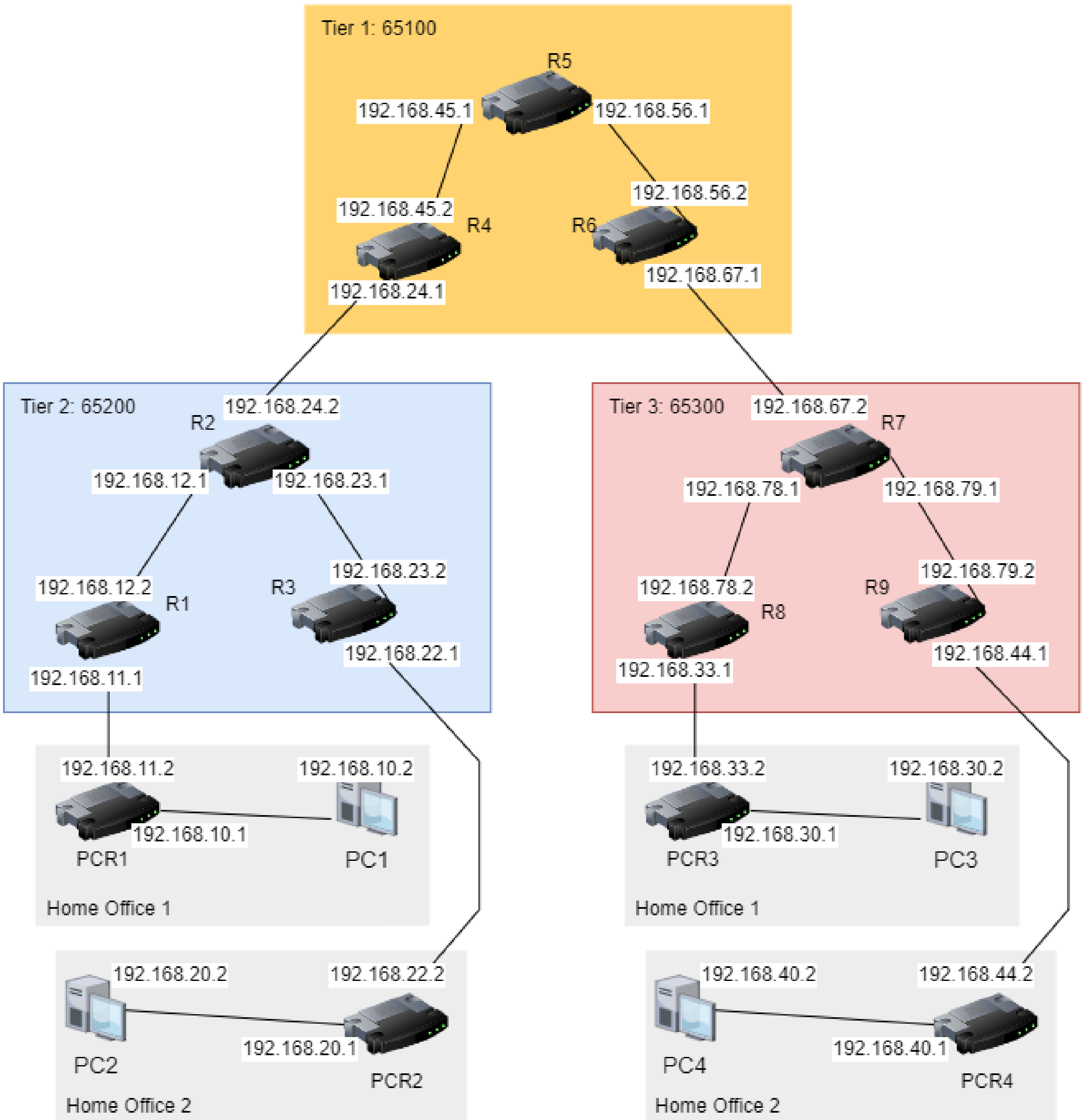
Note: AS = Autonomous System, ISP = Internet Service Provider, IGP = Interior Gateway Protocol, BGP = Border Gateway Protocol, QoS = Quality of Service.

Suggested connection diagram:



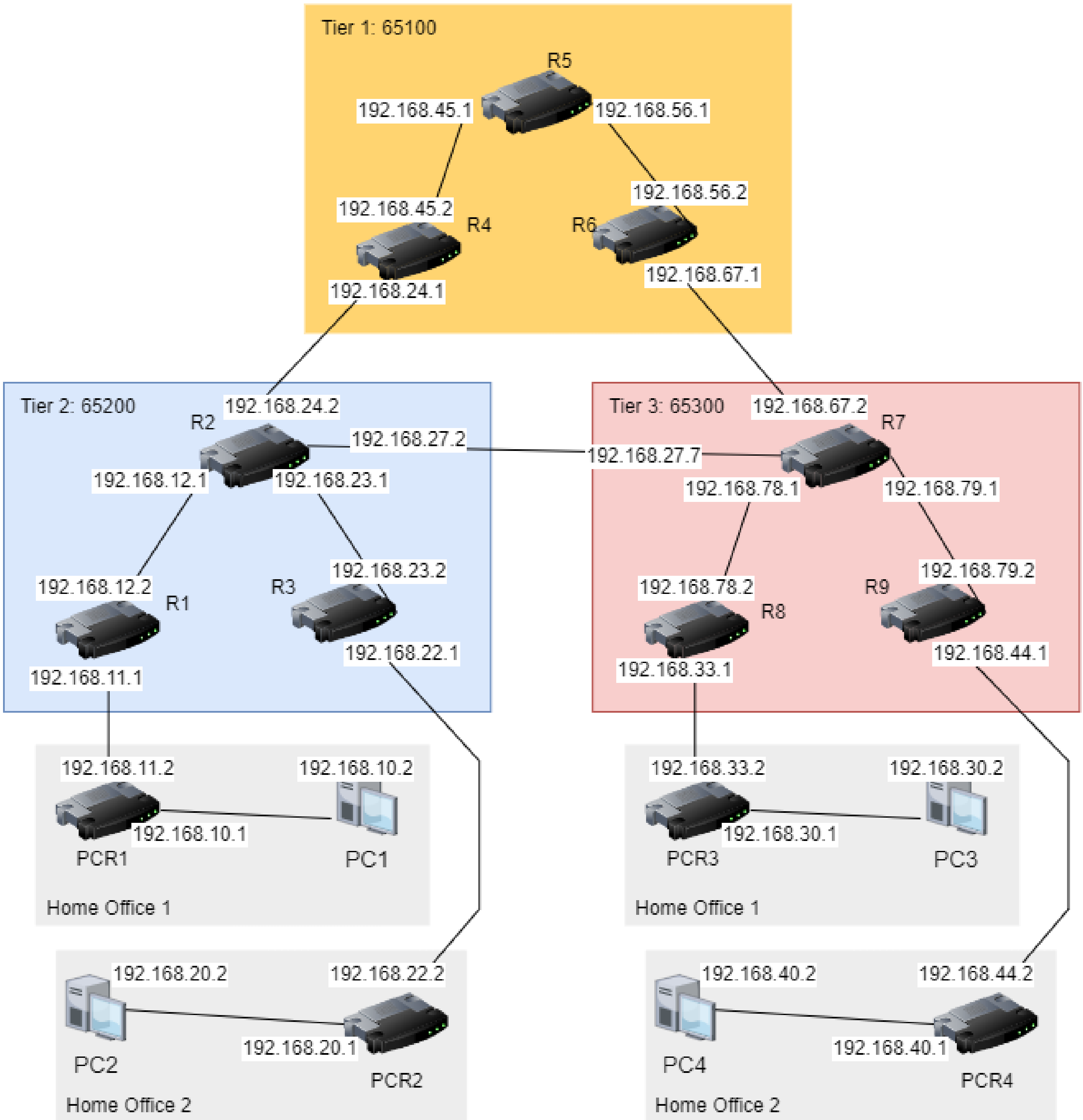
III) Design architecture

- For the first 3 questions, the system should look like this:



For simplicity, we use OSPF in all 3 tiers.

- In question 4, we use a line to connection between R2 and R7, so that tier 2 and tier 3 can communicate with each other:



We also need to modify BGP protocol so that it will prioritize going through that line instead of going through tier 1

- For question 5, we use multicast to broadcast messages between each computer in all home offices.

IV) Implementation

We use the tool FRR to config for every protocol in this demonstration. We also use VM VirtualBox to create virtual ubuntu servers, which is the environment to run. We use Ubuntu version 20 and FRR latest version of Jan 2024 (v8.5.3).

We use OSPF as IGP routing protocol for each AS. We use the pimd protocol for multicast service for the last question.

- First we create all the devices and setup the network using netplan normally, we will setup for 4 first questions, including the network 192.168.27.0

```
hieu@hieu:~$ sudo nano /etc/netplan/00-installer-config.yaml _
```

Settings of R5:

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: false
      addresses: [192.168.45.1/24]
    enp0s9:
      dhcp4: false
      addresses: [192.168.56.1/24]
  version: 2
```

R4:

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: false
      addresses: [192.168.45.2/24]
    enp0s10:
      dhcp4: false
      addresses: [192.168.24.1/24]
  version: 2
```

R6:

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s9:
      dhcp4: false
      addresses: [192.168.56.2/24]
    enp0s10:
      dhcp4: false
      addresses: [192.168.67.1/24]
  version: 2
```

R2:

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: false
      addresses: [192.168.27.2/24]
    enp0s8:
      dhcp4: false
      addresses: [192.168.12.1/24]
    enp0s9:
      dhcp4: false
      addresses: [192.168.23.1/24]
    enp0s10:
      dhcp4: false
      addresses: [192.168.24.2/24]
  version: 2
```

R1:

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: false
      addresses: [192.168.12.2/24]
    enp0s9:
      dhcp4: false
      addresses: [192.168.11.1/24]
  version: 2
```

PRC1:

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: false
      addresses: [192.168.10.1/24]
    enp0s9:
      dhcp4: false
      addresses: [192.168.11.2/24]
      gateway4: 192.168.11.1
  version: 2
```

PC1:

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: false
      addresses: [192.168.10.2/24]
      gateway4: 192.168.10.1
  version: 2
```

R3:

```yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: false
      addresses: [192.168.23.2/24]
    enp0s9:
      dhcp4: false
      addresses: [192.168.22.1/24]
  version: 2
```

PCR2:

```yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: false
      addresses: [192.168.20.1/24]
    enp0s9:
      dhcp4: false
      addresses: [192.168.22.2/24]
      gateway4: 192.168.22.1
  version: 2
```

PC2:

```yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: false
      addresses: [192.168.20.2/24]
      gateway4: 192.168.20.1
  version: 2
```

R7:

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: false
      addresses: [192.168.27.7/24]
    enp0s8:
      dhcp4: false
      addresses: [192.168.78.1/24]
    enp0s9:
      dhcp4: false
      addresses: [192.168.79.1/24]
    enp0s10:
      dhcp4: false
      addresses: [192.168.67.2/24]
  version: 2
```

R8:

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: false
      addresses: [192.168.78.2/24]
    enp0s9:
     dhcp4: false
     addresses: [192.168.33.1/24]
  version: 2
```

PCR3:

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: false
      addresses: [192.168.30.1/24]
    enp0s9:
      dhcp4: false
      addresses: [192.168.33.2/24]
      gateway4: 192.168.33.1
  version: 2
```

PC3:

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: false
      addresses: [192.168.30.2/24]
      gateway4: 192.168.30.1
  version: 2
```

R9:

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: false
      addresses: [192.168.79.2/24]
    enp0s9:
      dhcp4: false
      addresses: [192.168.44.1/24]
  version: 2
```

PCR4:

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: false
      addresses: [192.168.40.1/24]
    enp0s9:
      dhcp4: false
      addresses: [192.168.44.2/24]
      gateway4: 192.168.44.1
  version: 2
```

PC4:

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: false
      addresses: [192.168.40.2/24]
      gateway4: 192.168.40.1
  version: 2
```

Use command to apply the network address:

```
hieu@hieu:~$ sudo netplan apply
```

- Install FRR tool for each server:

The new version of FRR is harder to install in ubuntu. Following these command:
sudo apt update
sudo apt upgrade
echo deb https://deb.frrouting.org/frr $(lsb_release -s -c) frr-stable | sudo tee -a /etc/apt/sources.list.d/frr.list
curl -s https://deb.frrouting.org/frr/keys.asc | sudo apt-key add -
sudo apt update
sudo apt install frr frr-pythontools
Sudo systemctl start frr
sudo systemctl status frr

To install FRR, we need to make sure our router can access the internet by setting up a NAT network for each router.
Fix bug in each router, which can not access internet by running command:
echo “nameserver 8.8.8.8” | sudo tee /etc/resolv.conf > /dev/null

- We setup OSPF and BGP for all routers:

Enable FRR service in all router:

```
hieu@hieu:~$ sudo nano /etc/frr/daemons_
```

```
#
bgpd=yes
ospfd=yes
ospf6d=no
ripd=no
ripngd=no
isisd=no
pimd=yes
pim6d=no
```

R5:

```
!
router ospf
 network 192.168.45.0/24 area 0
 network 192.168.56.0/24 area 0
exit
!
```

R4:

```
!
router bgp 65100
 bgp router-id 192.168.24.1
 no bgp ebgp-requires-policy
 no bgp network import-check
 neighbor 192.168.24.2 remote-as 65200
 !
 address-family ipv4 unicast
  network 192.168.24.0/24
  network 192.168.45.0/24
  redistribute ospf
  neighbor 192.168.24.2 next-hop-self
 exit-address-family
exit
!
router ospf
 redistribute connected
 redistribute bgp
 network 192.168.45.0/24 area 0
exit
!
```

R6:

```
!
router bgp 65100
 bgp router-id 192.168.67.1
 no bgp ebgp-requires-policy
 no bgp network import-check
 neighbor 192.168.67.2 remote-as 65300
 !
 address-family ipv4 unicast
  network 192.168.56.0/24
  network 192.168.67.0/24
  redistribute ospf
  neighbor 192.168.67.2 next-hop-self
 exit-address-family
exit
!
router ospf
 redistribute connected
 redistribute bgp
 network 192.168.56.0/24 area 0
exit
!
```

R1:

```
!
router ospf
 redistribute connected
 network 192.168.12.0/24 area 1
exit
!
```

R3:

```
!
router ospf
 redistribute connected
 network 192.168.23.0/24 area 1
exit
!
```

R2:

```
!
router bgp 65200
 bgp router-id 192.168.24.2
 no bgp ebgp-requires-policy
 no bgp network import-check
 neighbor 192.168.24.1 remote-as 65100
 neighbor 192.168.27.7 remote-as 65300
 !
 address-family ipv4 unicast
  network 192.168.12.0/24
  network 192.168.23.0/24
  redistribute ospf
  neighbor 192.168.24.1 next-hop-self
  neighbor 192.168.27.7 next-hop-self
  neighbor 192.168.27.7 route-map PDP out
 exit-address-family
exit
!
router ospf
 redistribute connected
 redistribute bgp
 network 192.168.12.0/24 area 1
 network 192.168.23.0/24 area 1
exit
!
route-map PDP permit 10
 set local-preference 200
exit
!
```

R8:

```
!
router ospf
 redistribute connected
 network 192.168.78.0/24 area 1
exit
!
```

R9:

```
!
router ospf
 redistribute connected
 network 192.168.79.0/24 area 1
exit
!
```

R7:

```
!
router bgp 65300
 bgp router-id 192.168.67.2
 no bgp ebgp-requires-policy
 no bgp network import-check
 neighbor 192.168.27.2 remote-as 65200
 neighbor 192.168.67.1 remote-as 65100
 !
 address-family ipv4 unicast
  network 192.168.78.0/24
  network 192.168.79.0/24
  redistribute ospf
  neighbor 192.168.27.2 next-hop-self
  neighbor 192.168.27.2 route-map PDP out
  neighbor 192.168.67.1 next-hop-self
 exit-address-family
exit
!
router ospf
 redistribute connected
 redistribute bgp
 network 192.168.78.0/24 area 1
 network 192.168.79.0/24 area 1
exit
!
route-map PDP permit 10
 set local-preference 200
exit
!
```

Here in router R2 and R7, we use router-map to prioritize the connection over the path through tier 1. We did it here for question 4 already.

Make sure to enable rule for all router:

```
hieu@hieu:~$ sudo sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

In PCR1, PCR2, PCR3, PCR4 we need to enable rule to connect home office with outside network:

```
hieu@hieu:~$ sudo iptables -t nat -A POSTROUTING -j MASQUERADE_
```

=> Then all machines can ping each other, the routing table of every router is auto updated.

- We use multicast by protocol pimd in PCR1, R1, R2, PRC2, R7, R8, R9, PCR3, PCR4. In PC1, PC2, PC3, PC4 we only need to install the iperf tool.

Here we set up so that PC2 is the source emitting signal, other PCs will receive the signal. We use group 226.96.1.1 to multicast.

PCR1:

```
!
interface enp0s8
 ip igmp
 ip pim
exit
!
interface enp0s9
 ip igmp
 ip pim
exit
!
interface enp0s9
  ip mroute enp0s8 226.96.1.1
exit
!
```

R1:

```
!
interface enp0s8
 ip igmp
 ip pim
exit
!
interface enp0s9
 ip igmp
 ip pim
exit
!
router ospf
 redistribute connected
 network 192.168.12.0/24 area 1
exit
!
interface enp0s8
  ip mroute enp0s9 226.96.1.1
exit
!
```

R2:

```
!
interface enp0s8
 ip igmp
 ip pim
exit
!
interface enp0s9
 ip igmp
 ip pim
exit
!
interface enp0s3
 ip igmp
 ip pim
exit
!
```

```
!
interface enp0s9
 ip mroute enp0s8 226.96.1.1
 ip mroute enp0s3 226.96.1.1
exit
!
interface enp0s3
 ip mroute enp0s8 226.96.1.1
exit
!
```

R3:

```
!
interface enp0s8
 ip igmp
 ip pim
exit
!
interface enp0s9
 ip igmp
 ip pim
exit
!
router ospf
 redistribute connected
 network 192.168.23.0/24 area 1
exit
!
interface enp0s9
 ip mroute enp0s8 226.96.1.1
exit
!
```

PCR2:

```
!
interface enp0s8
 ip igmp
 ip pim
exit
!
interface enp0s9
 ip igmp
 ip pim
exit
!
router ospf
 network 192.168.23.0/24 area 1
exit
!
interface enp0s8
  ip mroute enp0s9 226.96.1.1
exit
!
```

R7:

```
!
interface enp0s3
 ip igmp
 ip pim
exit
!
interface enp0s8
 ip igmp
 ip pim
exit
!
interface enp0s9
 ip igmp
 ip pim
exit
!
```

```
!
route-map PDP permit 10
 set local-preference 200
exit
!
interface enp0s3
 ip mroute enp0s8 226.96.1.1
 ip mroute enp0s9 226.96.1.1
exit
!
```

R8:

```
!
interface enp0s8
 ip igmp
 ip pim
exit
!
interface enp0s9
 ip igmp
 ip pim
exit
!
router ospf
 redistribute connected
 network 192.168.78.0/24 area 1
exit
!
interface enp0s8
 ip mroute enp0s9 226.96.1.1
exit
!
interface enp0s9
 ip mroute enp0s8 226.96.1.2
exit
!
```

R9:

```
!
interface enp0s8
 ip igmp
 ip pim
exit
!
interface enp0s9
 ip igmp
 ip pim
exit
!
router ospf
 redistribute connected
 network 192.168.79.0/24 area 1
exit
!
interface enp0s8
 ip mroute enp0s9 226.96.1.1
 ip mroute enp0s9 226.96.1.2
exit
!
```

PCR3:

```
!
interface enp0s8
 ip igmp
 ip pim
exit
!
interface enp0s9
 ip igmp
 ip pim
exit
!
router ospf
 network 192.168.23.0/24 area 1
exit
!
interface enp0s9
  ip mroute enp0s8 226.96.1.1
exit
!
interface enp0s8
  ip mroute enp0s9 226.96.1.2
exit
!
```

PCR4:

```
!
interface enp0s8
 ip igmp
 ip pim
exit
!
interface enp0s9
 ip igmp
 ip pim
exit
!
router ospf
 network 192.168.23.0/24 area 1
exit
!
interface enp0s9
  ip mroute enp0s8 226.96.1.1
  ip mroute enp0s8 226.96.1.2
exit
!
```

After changing the file we need to restart FRR services, run:

```
hieu@hieu:~$ sudo systemctl restart frr
```

- Test multicast:

PC2:

```
hieu@hieu:~$ iperf -c 226.96.1.1 -u -T 10
------------------------------------------------------------
Client connecting to 226.96.1.1, UDP port 5001
Sending 1470 byte datagrams, IPG target: 11215.21 us (kalman adjust)
Setting multicast TTL to 10
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 192.168.20.2 port 38178 connected with 226.96.1.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec
[  3] Sent 893 datagrams
hieu@hieu:~$ _
```

PC1 receives signal:

```
hieu@hieu:~$ iperf -s -u -B 226.96.1.1
------------------------------------------------------------
Server listening on UDP port 5001
Binding to local address 226.96.1.1
Joining multicast group  226.96.1.1
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 226.96.1.1 port 5001 connected with 192.168.10.1 port 38178
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec   0.328 ms    0/  893 (0%)
```

PC3 receives signal:

```
hieu@hieu:~$ iperf -s -u -B 226.96.1.1
------------------------------------------------------------
Server listening on UDP port 5001
Binding to local address 226.96.1.1
Joining multicast group  226.96.1.1
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 226.96.1.1 port 5001 connected with 192.168.10.1 port 38178
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec   0.328 ms    0/  893 (0%)
```

PC4 receives signal:

```
hieu@hieu:~$ iperf -s -u -B 226.96.1.1
------------------------------------------------------------
Server listening on UDP port 5001
Binding to local address 226.96.1.1
Joining multicast group  226.96.1.1
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 226.96.1.1 port 5001 connected with 192.168.40.1 port 38178
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec   0.312 ms    0/  893 (0%)
```

V) Conclusion

In conclusion, this project provided hands-on experience in designing and configuring a sophisticated network infrastructure. Through the simulation, we successfully implemented IGP routing within each AS, established BGP connections between ASs, devised an IP address scheme, and ensured seamless connectivity among workstations. Additionally, the project addressed the nuanced decision-making process involved in direct peering between ISPs and the use of BGP policies. The consideration of multicast services or QoS for Home/Office networks further enriched our understanding of real-world networking scenarios. Overall, this endeavor not only reinforced theoretical knowledge but also honed practical skills crucial for network professionals in deploying and managing complex infrastructures.

.

REFERENCE

https://soict.daotao.ai/courses/course-v1:SoICT+IT4152E+2023-1/course/

https://docs.frrouting.org/en/latest/pim.html#pim-interface-configuration

https://github.com/FRRouting/frr/blob/master/pimd/COMMANDS