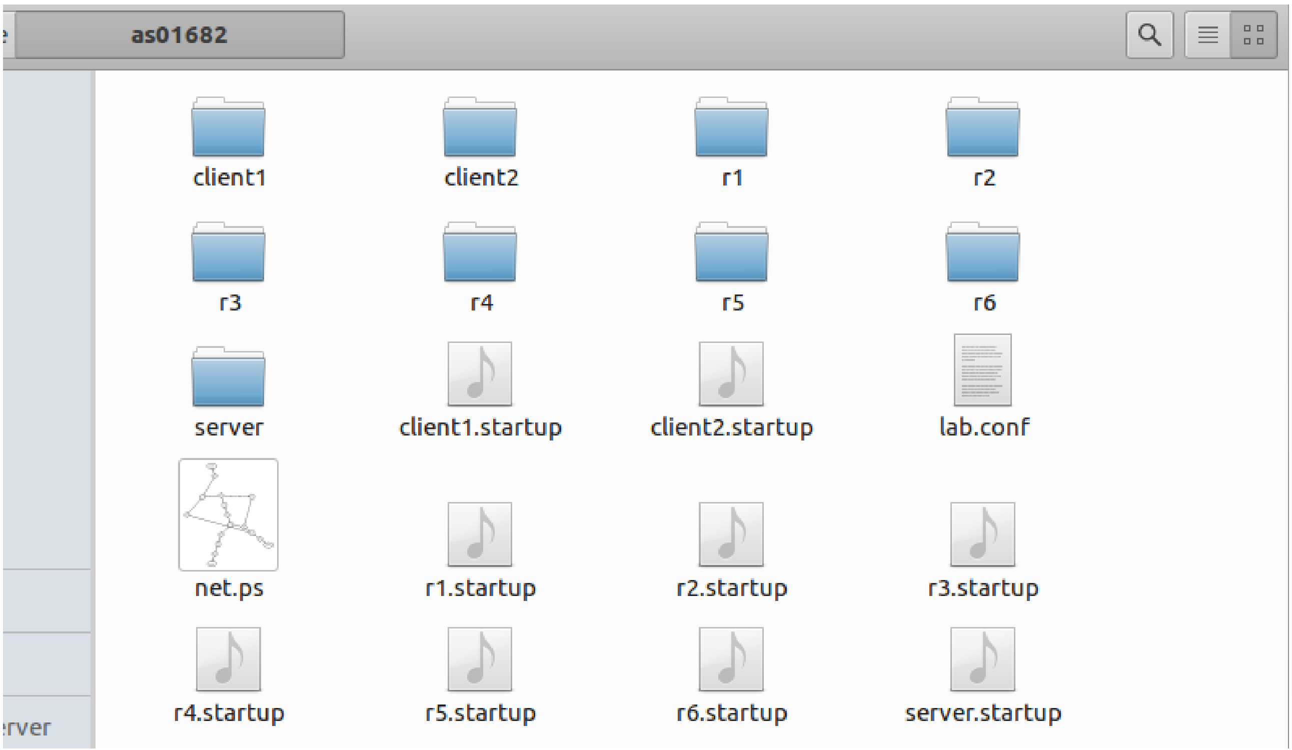
COM2022 as01682 6436270 Report
Task A Network Topology Implementation in Netkit [32 marks]

1.Create a new folder (use your username as the name of the folder) and keep your project solution in there. I will be referring to that folder as the “working folder”. Your topology should contain all of the VMs mentioned in the diagram (client1 , r1 , etc.).

The below screenshot shows the directory with all the VM’s mentioned in the diagram.



2. For each startup configuration file write the necessary commands to setup the VMs based on the details in the given network diagram and the coursework requirements.
T-A1: 9 marks (1 for each VM)

The below start-up files are configured to the specifications of the topology given. The eth# corresponds to the adjacent ethernet port of the VM and the ip address is calculated from the collision domain the eth connects to. The netmask and broadcast IP is worked out via the end number of the ip attached to the collision domain.

r1.startup

```
ifconfig eth0 151.0.82.1 netmask 255.255.255.0 up
ifconfig eth1 151.5.82.1 netmask 255.255.255.0 up
/etc/init.d/zebra start
```

r2.startup

```
ifconfig eth0 151.0.82.2 netmask 255.255.255.0 up
ifconfig eth1 151.1.82.1 netmask 255.255.255.0 up
ifconfig eth2 110.2.82.2 netmask 255.255.255.252 up
/etc/init.d/zebra start
```

r3.startup

```
ifconfig eth0 151.5.82.3 netmask 255.255.255.0 up
ifconfig eth1 151.1.82.3 netmask 255.255.255.0 up
/etc/init.d/zebra start
```

r4.startup

```
ifconfig eth0 151.5.82.2 netmask 255.255.255.0 up
ifconfig eth1 151.4.82.1 netmask 255.255.255.0 up
ifconfig eth2 151.3.82.1 netmask 255.255.255.0 up
/etc/init.d/zebra start
```

r5.startup

```
ifconfig eth0 151.1.82.2 netmask 255.255.255.0 up
ifconfig eth1 151.3.82.2 netmask 255.255.255.0 up
ifconfig eth2 151.2.82.1 netmask 255.255.255.0 up
/etc/init.d/zebra start
```

r6.startup

```
ifconfig eth0 210.2.82.2 netmask 255.255.255.252 up
ifconfig eth1 110.2.82.1 netmask 255.255.255.252 up
/etc/init.d/zebra start
```

server.startup

```
ifconfig eth0 210.2.82.1 netmask 255.255.255.252 up
/etc/init.d/zebra start
```

client1.startup

```
ifconfig eth0 151.4.82.2 netmask 255.255.255.0 up
/etc/init.d/zebra start
```

client2.startup

```
ifconfig eth0 151.2.82.2 netmask 255.255.255.0 up
/etc/init.d/zebra start
```

3. In the lab configuration file describe how the VMs are interconnected. Add the necessary commands to define the VM's interface and which collision domain is connected to.
T-A2: 2 marks

The lab.conf file shown below describes what eth# of the VM is connected to which collision domain(subnet). With the eth# being described by the number within the square brackets and the letter corresponding to the collision domain.

```
r1[0]=A
r1[1]=B

r2[0]=A
r2[1]=C
r2[2]=G

r3[0]=B
r3[1]=C

r4[0]=B
r4[1]=D
r4[2]=E

r5[0]=C
r5[1]=E
r5[2]=F

r6[0]=H
r6[1]=G

client1[0]=D

client2[0]=F

server[0]=H
```

4. Make sure to setup the routing service and daemons on all routing VMs, including the technique, interface cost, router and network area, as necessary. Be aware that the interface and cost might differ for each VM!
T-A3: 12 marks (2 for each VM)

The daemons file below is the same for all VM's so has only been shown once. It sets the ospfd file and zebra on. The ospfd files allow the VM's to connect to any router across the network and are shown below for each VM.

Daemons file for all VM's

```
# This file tells the zebra package
# which daemons to start.
# Entries are in the format: <daemon>=(yes|no|priority)
# where 'yes' is equivalent to infinitely low priority, and
# lower numbers mean higher priority. Read
# /usr/doc/zebra/README.Debian for details.
# Daemons are: bgpd zebra ospfd ospf6d ripd ripngd
zebra=yes
bgpd=no
ospfd=yes
ospf6d=no
ripd=no
ripngd=no
```

client1/ospfd.conf

```
!
hostname ospfd
password zebra
enable password zebra
!
router ospf
! Speak OSPF on all interfaces falling in the listed subnets
network 151.0.0.0/8 area 0.0.0.0

redistribute connected
!
log file /var/log/zebra/ospfd.log
!
```

client2/ospfd.conf

```
!
hostname ospfd
password zebra
enable password zebra
!
router ospf
! Speak OSPF on all interfaces falling in the listed subnets
network 151.0.0.0/8 area 0.0.0.0

redistribute connected
!
log file /var/log/zebra/ospfd.log
!
```

r1/ospfd.conf

```
!
hostname ospfd
password zebra
enable password zebra
!
interface eth1
ospf cost 15
!
router ospf
! Speak OSPF on all interfaces falling in the listed subnets
network 151.0.0.0/8 area 0.0.0.0

redistribute connected
!
log file /var/log/zebra/ospfd.log
!
```

r2/ospfd.conf

```
!
hostname ospfd
password zebra
enable password zebra
!
interface eth1
ospf cost 50
!
router ospf
! Speak OSPF on all interfaces falling in the listed subnets
network 151.0.0.0/8 area 0.0.0.0
network 110.0.0.0/30 area 1.1.1.1
area 1.1.1.1 stub

redistribute connected
!
log file /var/log/zebra/ospfd.log
!
```

r3/ospfd.conf

```
!
hostname ospfd
password zebra
enable password zebra
!
interface eth0
ospf cost 82
interface eth1
ospf cost 20
!
router ospf
! Speak OSPF on all interfaces falling in the listed subnets
network 151.0.0.0/8 area 0.0.0.0

redistribute connected
!
log file /var/log/zebra/ospfd.log
!
```

r4/ospfd.conf

```
!
hostname ospfd
password zebra
enable password zebra
!
interface eth2
ospf cost 50
!
router ospf
! Speak OSPF on all interfaces falling in the listed subnets
network 151.0.0.0/8 area 0.0.0.0

redistribute connected
!
log file /var/log/zebra/ospfd.log
!
```

r5/ospfd.conf

```
!
hostname ospfd
password zebra
enable password zebra
!
interface eth0
ospf cost 82
!
router ospf
! Speak OSPF on all interfaces falling in the listed subnets
network 151.0.0.0/8 area 0.0.0.0

redistribute connected
!
log file /var/log/zebra/ospfd.log
!
```

r6/ospfd.conf

```
!
hostname ospfd
password zebra
enable password zebra
!
router ospf
! Speak OSPF on all interfaces falling in the listed subnets
network 210.0.0.0/30 area 1.1.1.1
network 110.0.0.0/30 area 1.1.1.1
area 1.1.1.1 stub

redistribute connected
!
log file /var/log/zebra/ospfd.log
!
```

server/ospfd.conf

```
!
hostname ospfd
password zebra
enable password zebra
!
router ospf
! Speak OSPF on all interfaces falling in the listed subnets
network 210.0.0.0/30 area 1.1.1.1
area 1.1.1.1 stub

redistribute connected
!
log file /var/log/zebra/ospfd.log
!
```

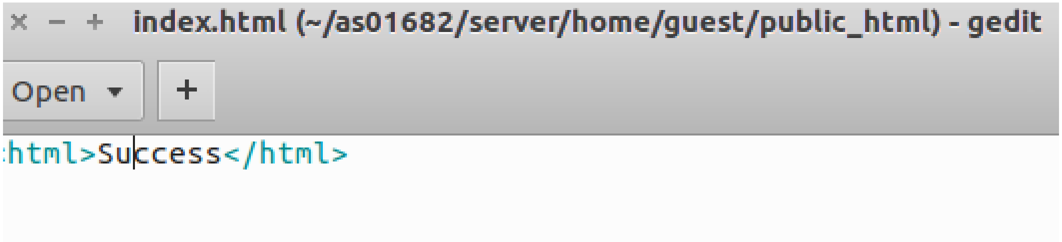
5. When you setup the Web server (Apache), edit the server startup file and add the necessary command to instruct Apache to start.
T-A4: 3 marks

The following bottom two lines of code were added to the server.startup file so that apache will run.

```
ifconfig eth0 210.2.82.1 netmask 255.255.255.252 up
/etc/init.d/zebra start
a2enmod userdir
/etc/init.d/apache2 start
```

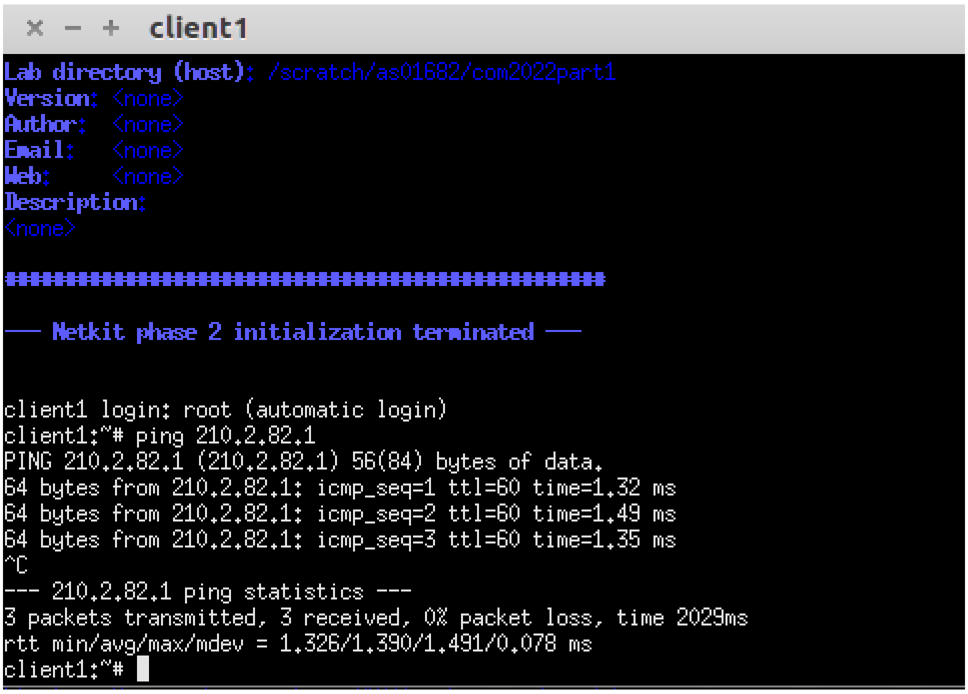
6. Create a simple HTML file (index.html) for the Apache server that would be requested by the clients for testing. The contents of the file can be anything like:
<html>success!!!</html>
Place that file in server/home/guest/public_html/
T-A5: 2 marks

The following html file can be requested by clients and is stored in server/home/guest/public_html/

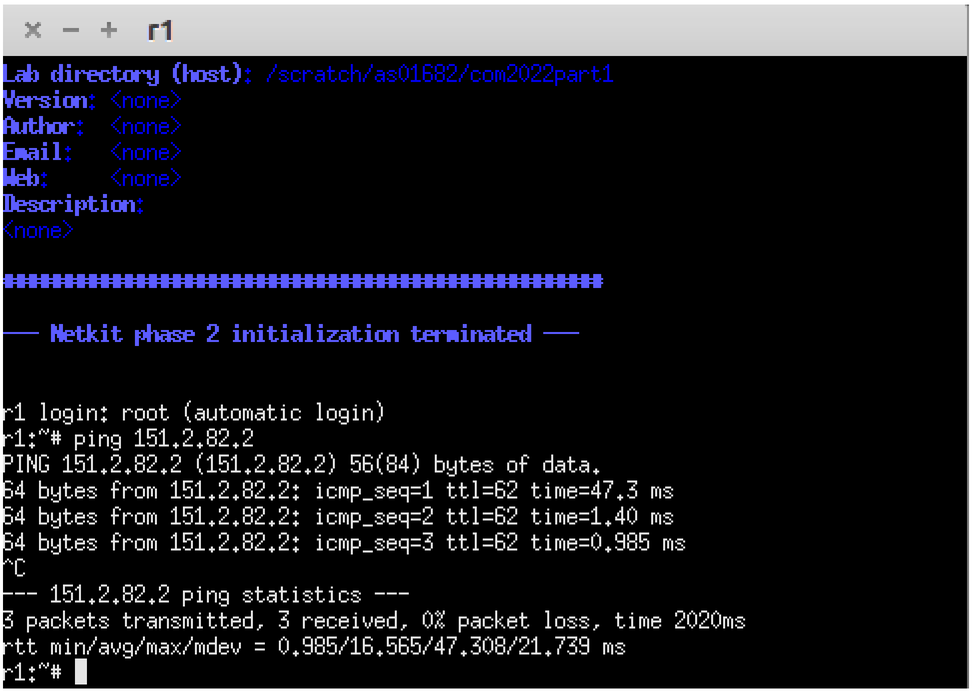


7. Make sure you try the network and ensure that it all works correctly by trying to ping from one node to another. Show the result of pinging 2 different node pairs, one from two different network areas and one from the same network area.
T-A6: 2 marks

To show the network is working correctly I have chosen to ping client1 which is in area 0.0.0.0 to server which is in 1.1.1.1

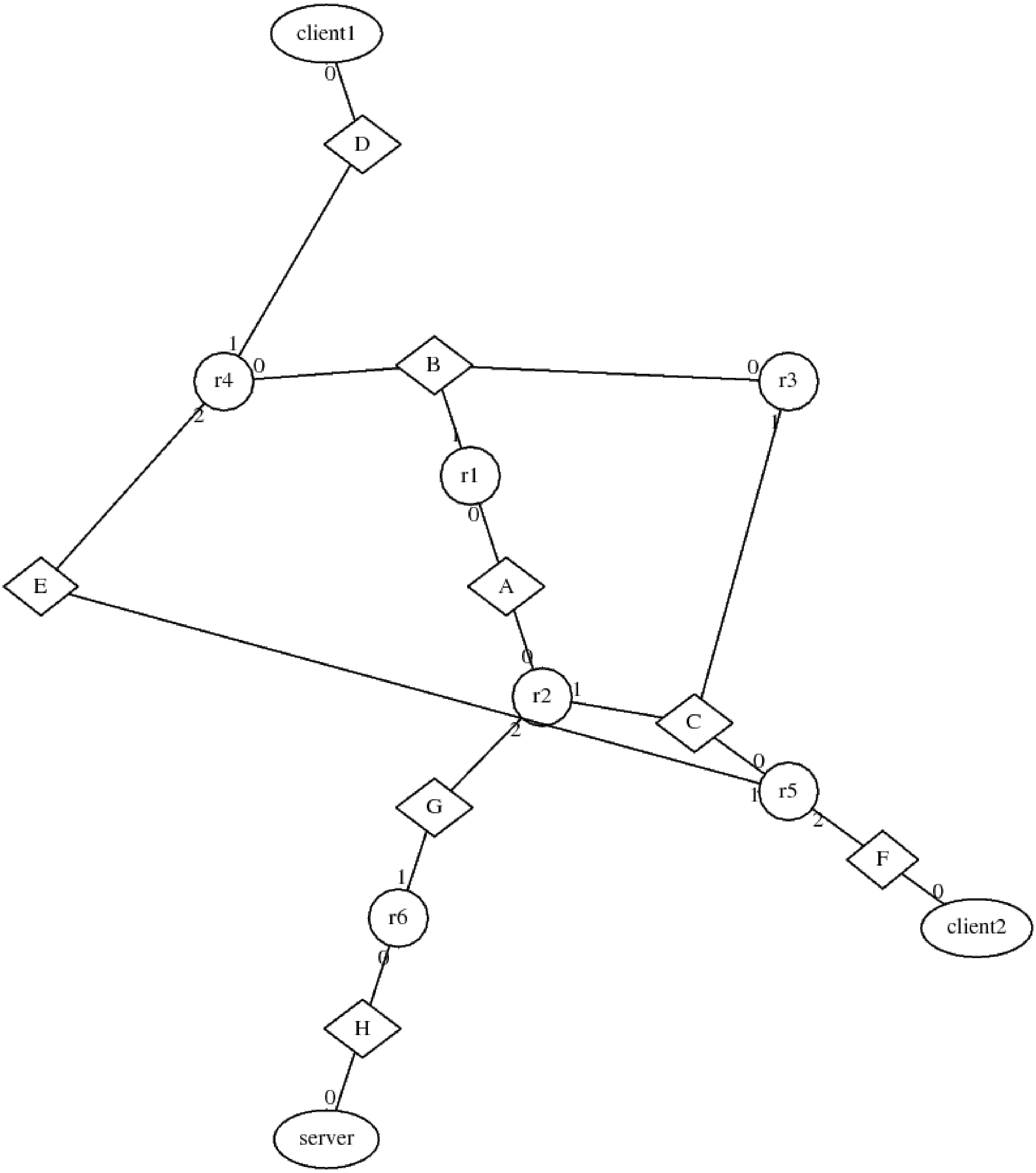


The second pair I chose was r1 to client 2. As there was a 0% packet loss it can be seen as successul.



8. Save an image (net.ps) of the graph using the linfo command, add it to your report and include any description or clarifications needed (only if is not clear).
$ linfo -m net.ps
T-A7: 2 marks

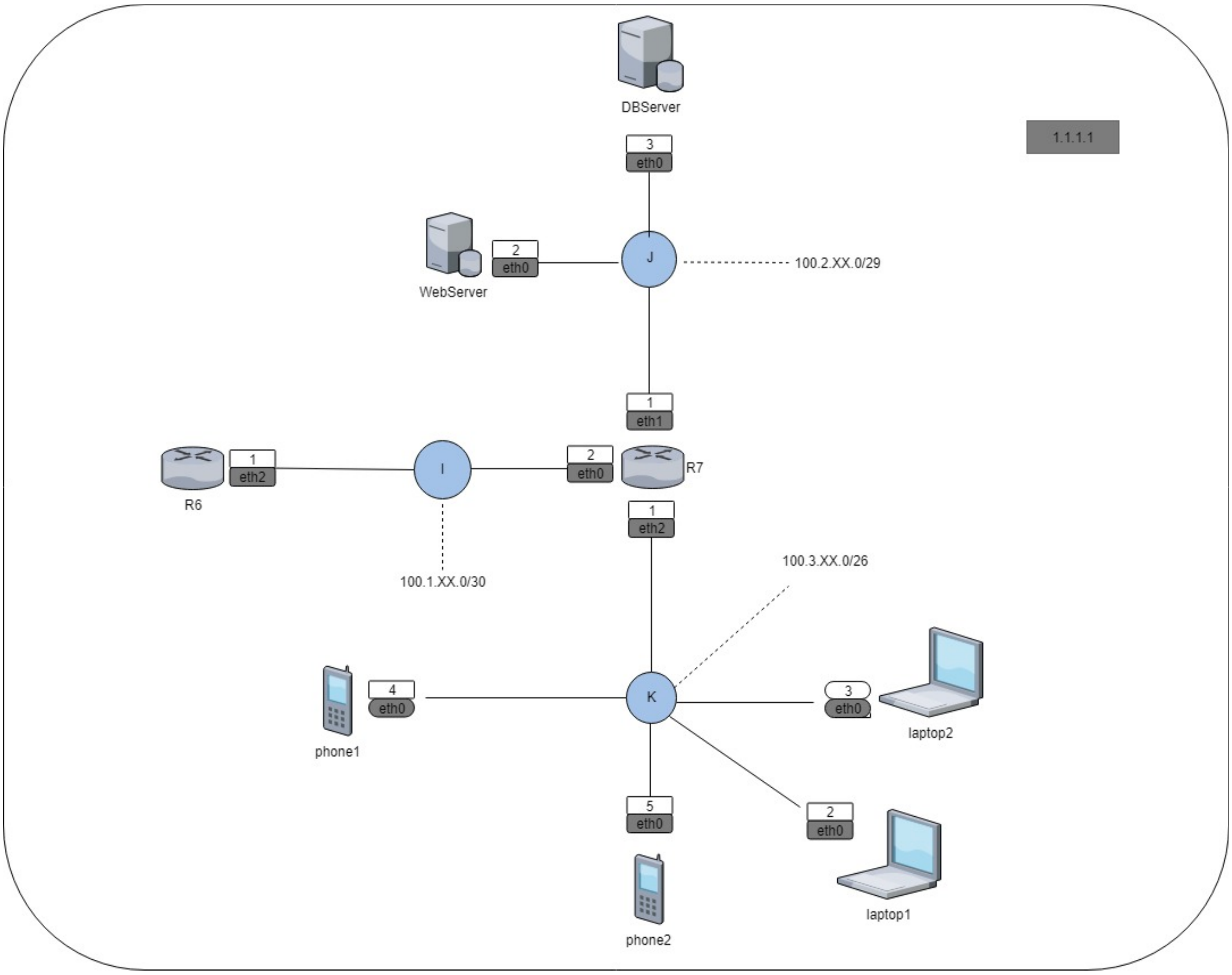
The image below shows the graph produced using the linfo command.



Task B: Network extension [32 marks]

1. Draw (either using draw.io on the web or your preferred drawing software) the network diagram and include this in your report. Give a short description too (max 1 page for all).
T-B1: 4 marks (2 for diagram and 2 for description)

I created an extension of the network in Draw.io which is visible below. In order to meet the requirements of the extension I decided to use router7 which connects to 3 subnets. One which links back to router6, one which links to the 2 servers and one which links to the 4 new devices.



2. Also respond to the following questions:
A) how and why you chose the network architecture (1 paragraph)

I chose this architecture as it was the most efficient way of meeting the requirements. I created a subnet with the IP address 100.1.82.0/30 as given, then I decided to use 2 subnets off of the router r7 to separate the servers and devices as specified in the business requirements. As we want up to 50 devices I chose a subnetmask of 26 for K to minimise hosts and a subnet mask of 29 for J as we want several servers, although several is an undefined number I decided to use 29 so that a few more servers could be added to the 2 asked for if desired.

B)how did you calculate the IPs? Show your working (half page)

I calculated the IP's based off of the initial requirements. As the given IP was 100.1.XX.0/30 and knew that as I would need 2 other servers and up to 50 other devices I would need 2 other subnets. I connected r6 to r7 and then made collision domains J and K. I went with the subnet mask of 29 for J as it would be sufficient to allow for the 2 servers and give room for extension for further VM's in the future. To support up to 50 different devices on the second subnet I had to adjust my subnet mask to allow for sufficient IP addresses. I decided on /26 as my subnet mask as it would allow for up to 62 device IP's. I worked this out as there are 32 bits to the IP address and so 32-26 = 6 bits. The maximum number that can be made from 6 bits is 63. If I were to do 5 bits and thus /27 would only allow for 31 IP's which wouldn't be enough for 50 devices.

C) Based on the given IP and subnet mask, could you implement three subnets and how? (a paragraph)

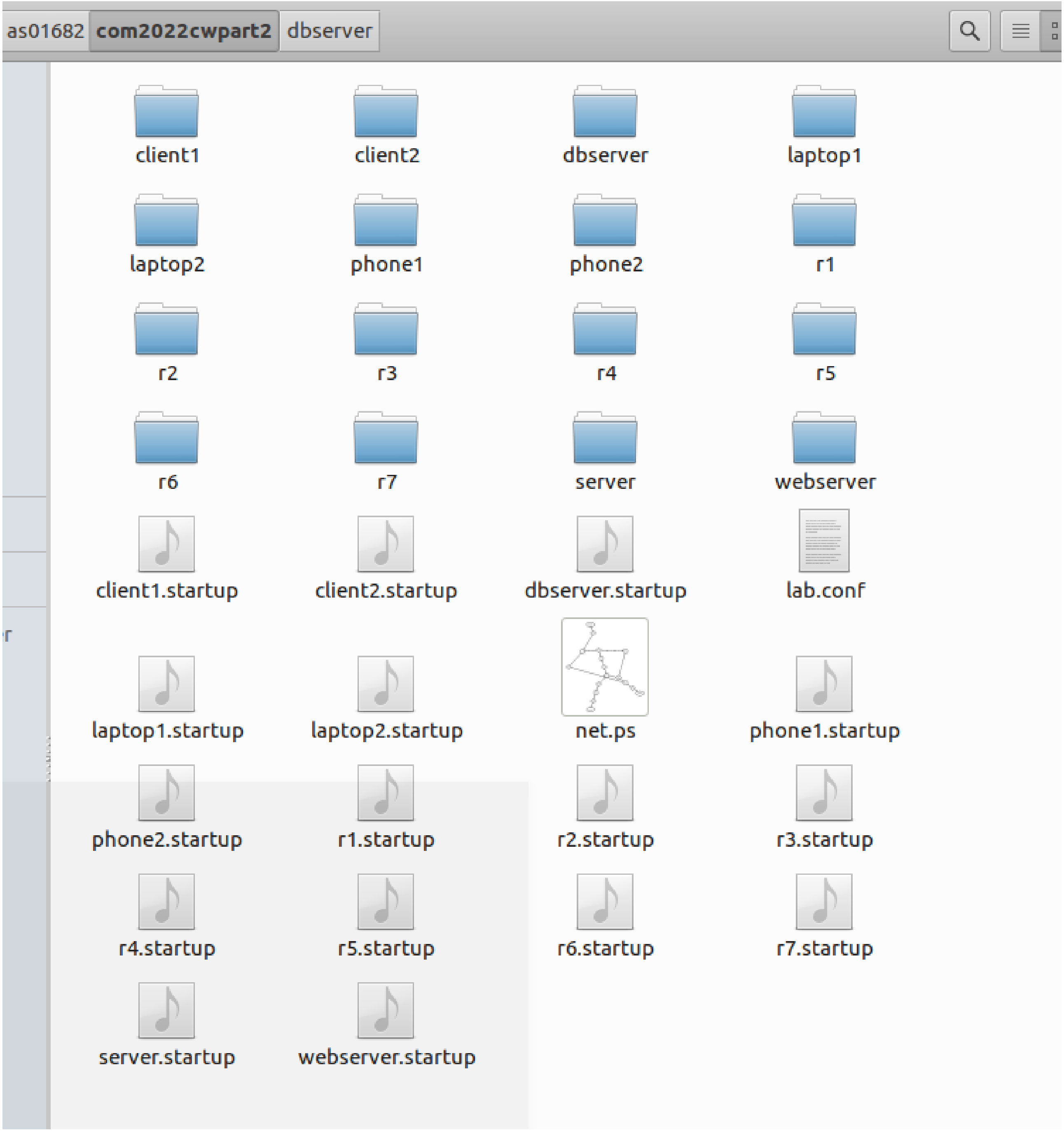
I believe it would be easy to implement a 3rd subnet as it could be extended off of r7 on a new eth3. The Ip would be 100.4.XX.0/XX.

D)what would you have to do to support 100 laptops on subnet 2? What if you had 300 laptops on subnet 2? (two paragraphs)

If you were to support 100 laptops on subnet 2 the subnet mask would have to be adjusted to /25 as this would allow for 7 bits and a maximum of 127 devices. If 300 devices were to be connected then a netmask of 23 as this would allow 510 hosts.

3. Implement the solution in NetKit and make sure you follow a similar process to the one described in T-A1 to T-A3, so make sure to give a short description and screenshots of your thinking (like you did in T-A1 to T-A3).

In order to reduce repetition in this report I will only include screen shots of the startup files of the new VM's added to the network and any VMS's which have been changed in order to support the network extension. Below is a screen shot of the new directory with new VM's.



For each startup configuration file write the necessary commands to setup the VMs based on the details in the given network diagram and the coursework requirements.
T-A1: 9 marks (1 for each VM)

Below are the startup files, r6 has been adjusted to include the eth2 extension.

R6.startup

```
ifconfig eth0 210.2.82.2 netmask 255.255.255.252 up
ifconfig eth1 110.2.82.1 netmask 255.255.255.252 up
ifconfig eth2 100.1.82.1 netmask 255.255.255.252 up
/etc/init.d/zebra start
```

r7.startup

```
ifconfig eth0 100.1.82.2 netmask 255.255.255.252 up
ifconfig eth1 100.2.82.1 netmask 255.255.255.248 up
ifconfig eth2 100.3.82.1 netmask 255.255.255.192 up
/etc/init.d/zebra start
```

dbserver.startup

```
ifconfig eth0 100.2.82.3 netmask 255.255.255.248 up
/etc/init.d/zebra start
```

webserver.startup

```
ifconfig eth0 100.2.82.2 netmask 255.255.255.248 up
/etc/init.d/zebra start
```

laptop1.startup

```
ifconfig eth0 100.3.82.2 netmask 255.255.255.192 up
/etc/init.d/zebra start
```

laptop2.startup

```
ifconfig eth0 100.3.82.3 netmask 255.255.255.192 up
/etc/init.d/zebra start
```

phone1.startup

```
ifconfig eth0 100.3.82.4 netmask 255.255.255.192 up
/etc/init.d/zebra start
```

phone2.startup

```
ifconfig eth0 100.3.82.5 netmask 255.255.255.192 up
/etc/init.d/zebra start
```

In the lab configuration file describe how the VMs are interconnected. Add the necessary commands to define the VM's interface and which collision domain is connected to.
T-A2: 2 marks

The lab.conf file has changed to include the new VM's and r6 has included the connection to I.

```
r1[0]=A
r1[1]=B

r2[0]=A
r2[1]=C
r2[2]=G

r3[0]=B
r3[1]=C

r4[0]=B
r4[1]=D
r4[2]=E

r5[0]=C
r5[1]=E
r5[2]=F

r6[0]=H
r6[1]=G
r6[2]=I

r7[0]=I
r7[1]=J
r7[2]=K

client1[0]=D
client2[0]=F

laptop1[0]=K
laptop2[0]=K
phone1[0]=K
phone2[0]=K

server[0]=H
webserver[0]=J
dbserver[0]=J
```

Make sure to setup the routing service and daemons on all routing VMs, including the technique, interface cost, router and network area, as necessary. Be aware that the interface and cost might differ for each VM!
T-A3: 12 marks (2 for each VM)

Below are the new ospfd files to allow the VM's to connect to each other. R6 is the only router that has changed from the original network so is the only one shown. Ithas been adjusted to include the network extension and all other files in the extension have also access to 100.0.0.0/8.

r6.ospf

```
!
hostname ospfd
password zebra
enable password zebra
!
router ospf
! Speak OSPF on all interfaces falling in the listed subnets
network 210.0.0.0/30 area 1.1.1.1
network 110.0.0.0/30 area 1.1.1.1
network 100.0.0.0/8 area 1.1.1.1
area 1.1.1.1 stub

redistribute connected
!
log file /var/log/zebra/ospfd.log
!
```

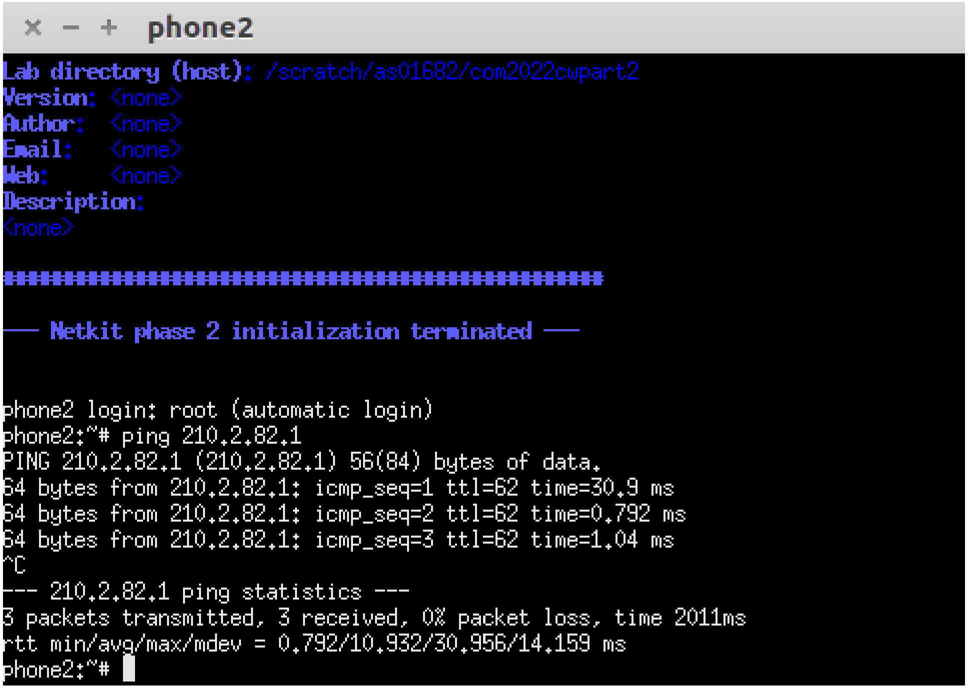
r7.ospf, dbserver.ospf, webserver.ospf, phone1.ospf, phone2.ospf, laptop1.ospf, laptop2.ospf

```
!
hostname ospfd
password zebra
enable password zebra
!
router ospf
! Speak OSPF on all interfaces falling in the listed subnets
network 100.0.0.0/8 area 1.1.1.1
area 1.1.1.1 stub

redistribute connected
!
log file /var/log/zebra/ospfd.log
!
```

4. Test and make sure is all working appropriately. Document your process (about half page and screenshots as you did for T-A6).
NOTE: You will be marked on your approach to test the network.
T-B4: 2 marks

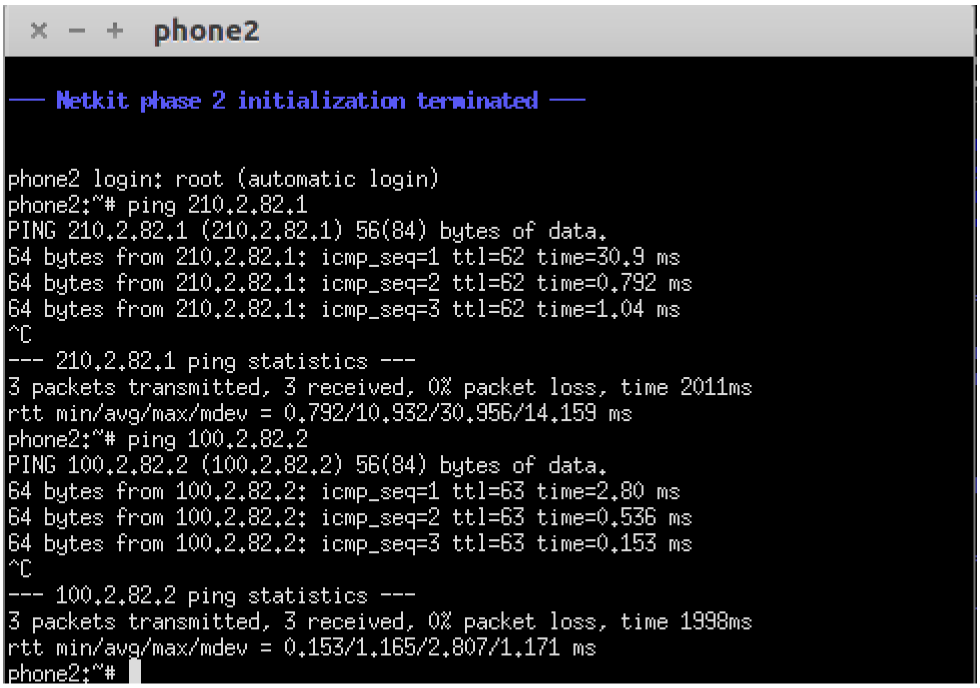
To test that the network is working correctly I have attempted to ping phone 2 to server and also phone 2 to webserver. I chose phone2 to webserver as it is trying to ping from the extension to the original network and phone2 to webserver is within the extension but different subnets.

This can be seen as successful as there was a 0% packet loss.



```
phone2
Lab directory (host): /scratch/as01682/com2022cwpart2
Version: <none>
Author:  <none>
Email:   <none>
Web:     <none>
Description:
<none>

##################################################

--- Netkit phase 2 initialization terminated ---


phone2 login: root (automatic login)
phone2:~# ping 210.2.82.1
PING 210.2.82.1 (210.2.82.1) 56(84) bytes of data.
64 bytes from 210.2.82.1: icmp_seq=1 ttl=62 time=30.9 ms
64 bytes from 210.2.82.1: icmp_seq=2 ttl=62 time=0.792 ms
64 bytes from 210.2.82.1: icmp_seq=3 ttl=62 time=1.04 ms
^C
--- 210.2.82.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2011ms
rtt min/avg/max/mdev = 0.792/10.932/30.956/14.159 ms
phone2:~#
```

Ping phone2 to webserver. This can be seen as successful as there was a 0% packet loss.



```
phone2
--- Netkit phase 2 initialization terminated ---


phone2 login: root (automatic login)
phone2:~# ping 210.2.82.1
PING 210.2.82.1 (210.2.82.1) 56(84) bytes of data.
64 bytes from 210.2.82.1: icmp_seq=1 ttl=62 time=30.9 ms
64 bytes from 210.2.82.1: icmp_seq=2 ttl=62 time=0.792 ms
64 bytes from 210.2.82.1: icmp_seq=3 ttl=62 time=1.04 ms
^C
--- 210.2.82.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2011ms
rtt min/avg/max/mdev = 0.792/10.932/30.956/14.159 ms
phone2:~# ping 100.2.82.2
PING 100.2.82.2 (100.2.82.2) 56(84) bytes of data.
64 bytes from 100.2.82.2: icmp_seq=1 ttl=63 time=2.80 ms
64 bytes from 100.2.82.2: icmp_seq=2 ttl=63 time=0.536 ms
64 bytes from 100.2.82.2: icmp_seq=3 ttl=63 time=0.153 ms
^C
--- 100.2.82.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.153/1.165/2.807/1.171 ms
phone2:~#
```

Task C: Routing experiments [8 marks]

1. Examine and compare the routing algorithms and see if they have calculated the shortest paths correctly and how they do that (about 1 page). Make use of the show command of the vtysh routing software, but you are basically allowed to use any other tools or commands you know of to examine the routing. You will need to do some research in understanding the two algorithms (OSPF and RIP), in order to explain the differences.
T-C1: 4 marks

Below shows the show route tables for ospf and vtysh of server.

RIP calculates the shortest distance via hop count whereas ospf uses djiktras path and takes into acocunt the interface cost. Rip only keeps track of the closest router for each destination address whereas ospf keeps track of the whole topological tree. Both have calculated their shortest paths correctly for their given methods however they will not always achieve the same result because of this. Rip is best for a small networks as it is poorly scaled to larger ones.

```
server# show ip ospf route
============ OSPF network routing table ============
N IA 0.0.0.0/0          [21] area: 1.1.1.1
                        via 210.2.82.2, eth0
N    100.1.82.0/30      [20] area: 1.1.1.1
                        via 210.2.82.2, eth0
N    100.2.82.0/29      [30] area: 1.1.1.1
                        via 210.2.82.2, eth0
N    100.3.82.0/26      [30] area: 1.1.1.1
                        via 210.2.82.2, eth0
N    110.2.82.0/30      [20] area: 1.1.1.1
                        via 210.2.82.2, eth0
N IA 151.0.82.0/24      [30] area: 1.1.1.1
                        via 210.2.82.2, eth0
N IA 151.1.82.0/24      [65] area: 1.1.1.1
                        via 210.2.82.2, eth0
N IA 151.2.82.0/24      [75] area: 1.1.1.1
                        via 210.2.82.2, eth0
N IA 151.3.82.0/24      [75] area: 1.1.1.1
                        via 210.2.82.2, eth0
N IA 151.4.82.0/24      [55] area: 1.1.1.1
                        via 210.2.82.2, eth0
N IA 151.5.82.0/24      [45] area: 1.1.1.1
                        via 210.2.82.2, eth0
N    210.2.82.0/30      [10] area: 1.1.1.1
                        directly attached to eth0

============ OSPF router routing table =============
R    110.2.82.2         [20] area: 1.1.1.1, ABR
                        via 210.2.82.2, eth0

============ OSPF external routing table ===========
```

```
server:~# vtysh

Hello, this is Quagga (version 0.99.10).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

server# show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

     Network            Next Hop         Metric From            Tag Time
R(n) 100.1.82.0/30      210.2.82.2            2 210.2.82.2        0 02:49
R(n) 100.2.82.0/29      210.2.82.2            3 210.2.82.2        0 02:49
R(n) 100.3.82.0/26      210.2.82.2            3 210.2.82.2        0 02:49
R(n) 110.2.82.0/30      210.2.82.2            2 210.2.82.2        0 02:49
R(n) 151.0.82.0/24      210.2.82.2            3 210.2.82.2        0 02:49
R(n) 151.1.82.0/24      210.2.82.2            3 210.2.82.2        0 02:49
R(n) 151.2.82.0/24      210.2.82.2            4 210.2.82.2        0 02:49
R(n) 151.3.82.0/24      210.2.82.2            4 210.2.82.2        0 02:49
R(n) 151.4.82.0/24      210.2.82.2            5 210.2.82.2        0 02:49
R(n) 151.5.82.0/24      210.2.82.2            4 210.2.82.2        0 02:49
C(i) 210.2.82.0/30      0.0.0.0               1 self              0
server#
```

2. Explain in a few words (about a page including screenshots) the preferred route (separately for OSPF and RIP) in the network when data is exchanged between client2 and the server. Add in your explanation references to the screenshots you take to support your answer.
T-C2: 4 marks

The trace route between client 2 and the server can be seen below for both osfp and rip with the trace route path to 210.2.82.1. Ospf travels from client 2 – r5 – r1 – r2 – r6 – server with a total interface cost of 10 + 10 + 10 + 10 + 10 = 60. RIP on the other hand travels from client 2 to r5 to r2 to r6 to server with a total interface cost of 10 + 82 + 10 + 10 = 112. RIP chooses the more costly path as it works on the shortest number of hops.

ospf

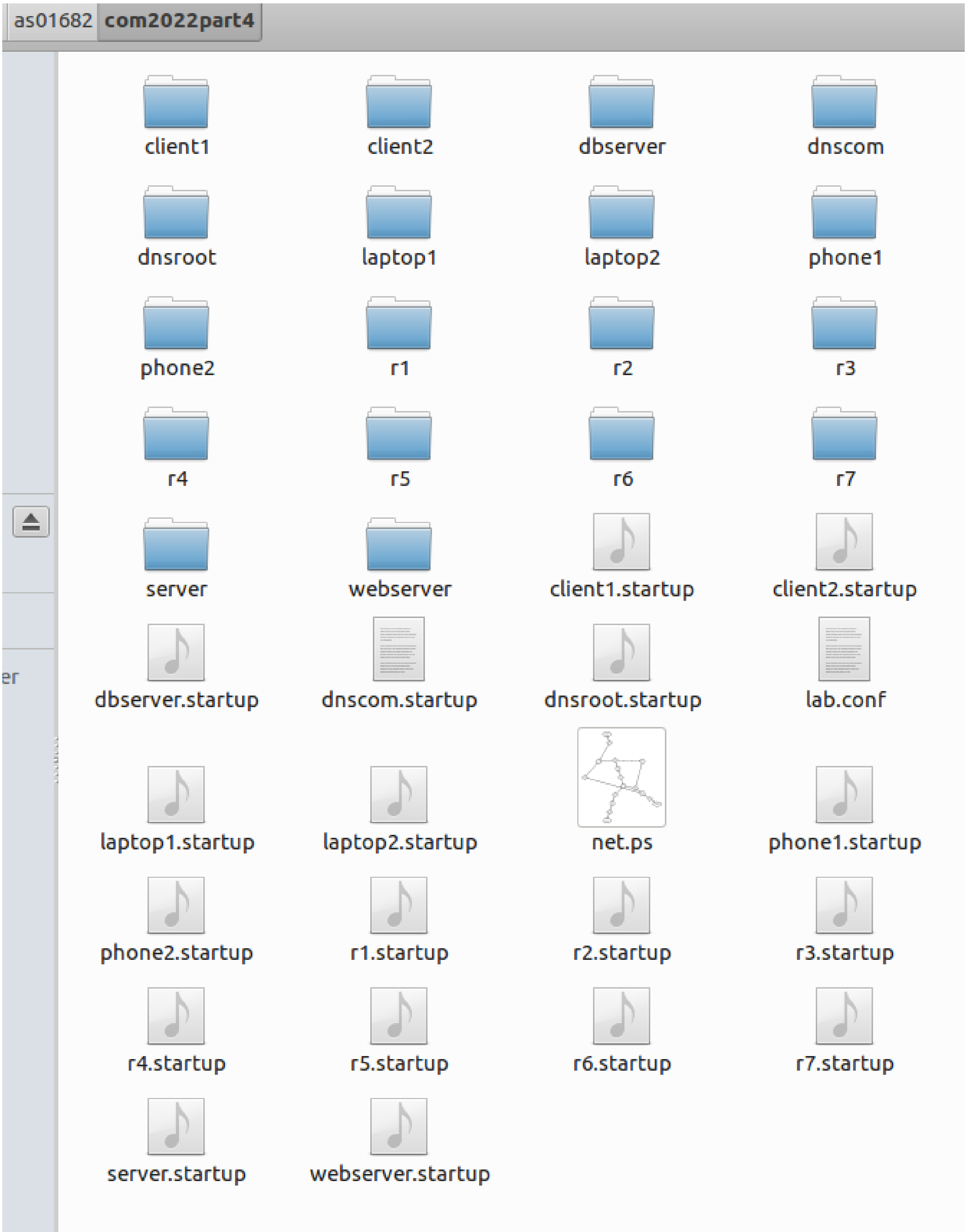
```
× − + client2

Lab directory (host): /scratch/as01682/as01682net/com2022part2
Version: <none>
Author:  <none>
Email:   <none>
Web:     <none>
Description:
<none>

##################################################

--- Netkit phase 2 initialization terminated ---


client2 login: root (automatic login)
client2:~# traceroute 210.2.82.1
traceroute to 210.2.82.1 (210.2.82.1), 64 hops max, 40 byte packets
 1  151.2.82.1 (151.2.82.1)  1 ms  1 ms  0 ms
 2  151.5.82.2 (151.5.82.2)  33 ms  1 ms  4 ms
 3  151.5.82.1 (151.5.82.1)  12 ms  2 ms  1 ms
 4  151.0.82.2 (151.0.82.2)  2 ms  2 ms  1 ms
 5  110.2.82.1 (110.2.82.1)  10 ms  1 ms  2 ms
 6  210.2.82.1 (210.2.82.1)  12 ms  1 ms  1 ms
client2:~#
```

rip

```
× − + client2

##################################################

Lab directory (host): /scratch/as01682/com2022pt3
Version: <none>
Author:  <none>
Email:   <none>
Web:     <none>
Description:
<none>

##################################################

--- Netkit phase 2 initialization terminated ---


client2 login: root (automatic login)
client2:~# traceroute 210.2.82.1
traceroute to 210.2.82.1 (210.2.82.1), 64 hops max, 40 byte packets
 1  151.2.82.1 (151.2.82.1)  7 ms  0 ms  0 ms
 2  151.1.82.1 (151.1.82.1)  1 ms  1 ms  1 ms
 3  110.2.82.1 (110.2.82.1)  1 ms  1 ms  1 ms
 4  210.2.82.1 (210.2.82.1)  11 ms  1 ms  1 ms
client2:~#
```

Task D: DNS server [8 marks]

1. I added a DNS server to collision domain H. In order to achieve this I had to change the subnet mask from 30 to 28 in order to make enough space for the dnsroot server and dnscom server. I have provided screenshots below of all the new files which needed to be implemented to make it so that the client can connect to the server using the domain name http://server.com/~guest/.



```
× − + client1
>>> Running client1 specific startup script...
Loading capability module if not yet done.
Starting Quagga daemons (prio:10): zebra ospfd.
>>> End of client1 specific startup script.

##############################################

Lab directory (host): /scratch/as01682/as01682net/com2022part4
Version: <none>
Author:  <none>
Email:   <none>
Web:     <none>
Description:
<none>

##############################################

--- Netkit phase 2 initialization terminated ---

client1 login: root (automatic login)
client1:~# links http://server.com/~guest
```

2. Describe your choice of settings and explain how the DNS server functions for your network (2 paragraphs).
T-D2: 2 marks

Below shows the dnscom files and the dnsroot files. I had to give them IP's off of the same subnet as R6.

```
× − + db.com (/scratch/as01682/com2022part4/dnscom/etc/bind) - gedit
Open ▾  +
$TTL    60000
@               IN      SOA     dnscom.com.     root.dnscom.com. (
                        2006031201 ; serial
                        28800 ; refresh
                        14400 ; retry
                        3600000 ; expire
                        0 ; negative cache ttl
                        )
@                   IN      NS      dnscom.com.
dnscom              IN      A       210.2.82.3
server              IN      A       210.2.82.1
```

```
× − + db.root (/scratch/as01682/com2022part4/dnscom/etc/bind) - gedit
Open ▾  +
.                   IN   NS    ROOT-SERVER.
ROOT-SERVER.        IN   A     210.2.82.4
```

db.210.0.82 (/scratch/as01682/com2022part4/dnsroot/etc/bind) - gedit

```
; reverse lookup database
$TTL    1
@               IN      SOA     4.82.2.210.in-addr.arpa.    nobody.nowhere. (
                        2006031201 ; serial
                        28 ; refresh
                        14 ; retry
                        3600000 ; expire
                        0 ; negative cache ttl
                        )
@                            IN      NS      4.82.2.210.in-addr.arpa.
4.82.2.210.in-addr.arpa.    IN      A       210.2.82.4
4          PTR        .
3          PTR        dnscom.com.
1          PTR        server.com.
```

db.root (/scratch/as01682/com2022part4/dnsroot/etc/bind) - gedit

```
$TTL    60000
@               IN      SOA     ROOT-SERVER.    root.ROOT-SERVER. (
                        2006031201 ; serial
                        28800 ; refresh
                        14400 ; retry
                        3600000 ; expire
                        0 ; negative cache ttl
                        )

@               IN      NS      ROOT-SERVER.
ROOT-SERVER.    IN      A       210.2.82.4

org.            IN      NS      dnsorg.org.
dnscom.com.     IN      A       210.2.82.3
```

Here shows the startup files for dsncom and dnsroot. They have the default gateway 210.2.82.2 which is the r6 router. The resolv.conf files are needed on all clients which would like to access the server.com. I have onl included it in my client1 folder. So server.com/~guest/ can be accessed from here.

resolv.conf (/scratch/as01682/com2022part4/client1/etc) - gedit

```
nameserver 210.2.82.3
search server
```

dnscom.startup (/scratch/as01682/com2022part4) - gedit

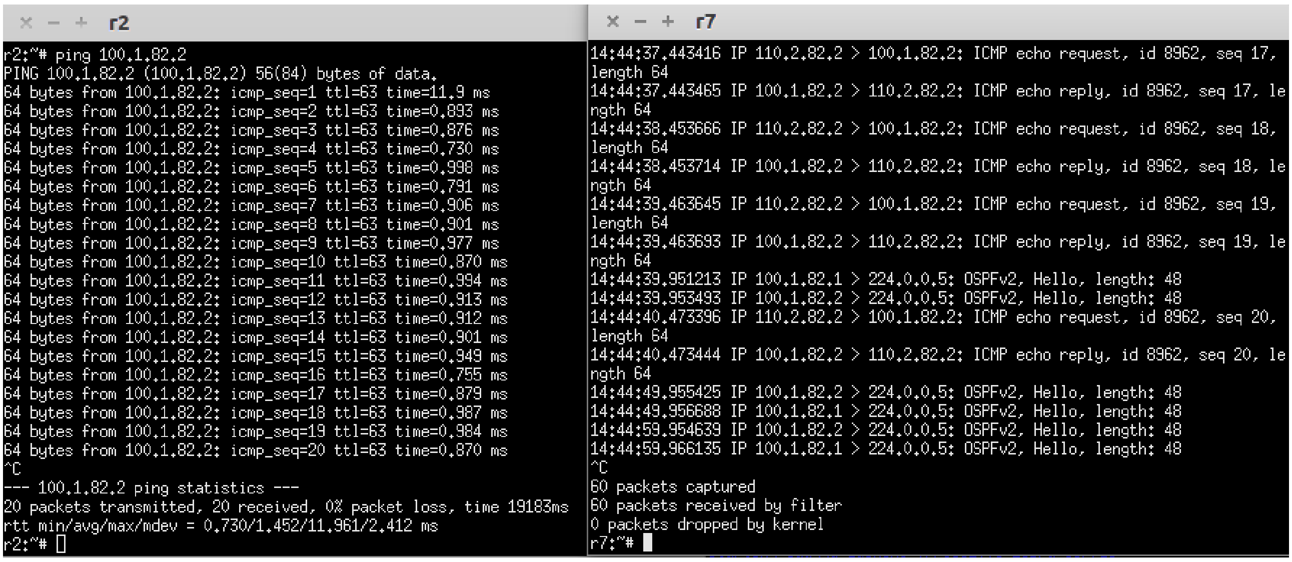
```
/sbin/ifconfig eth0 210.2.82.3 up
route add default gw 210.2.82.2 dev eth0
/etc/init.d/bind start
```

dnsroot.startup (/scratch/as01682/com2022part4) - gedit

```
ifconfig eth0 210.2.82.4 up
route add default gw 210.2.82.2 dev eth0
/etc/init.d/bind start
```

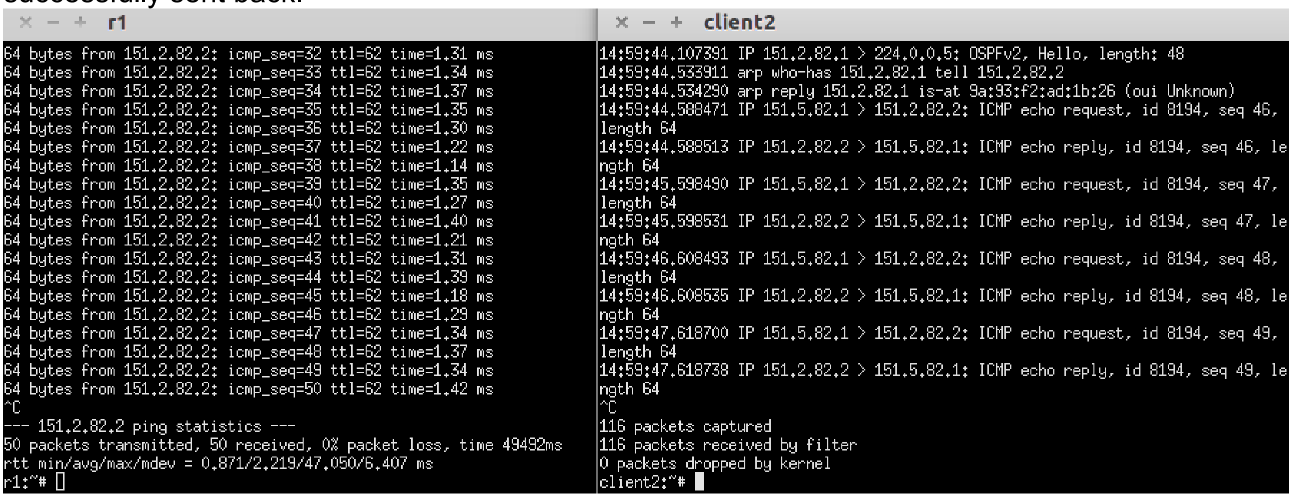
Task E Network Analysis

1. Test the connectivity from a node to another. Choose 2 different node pairs that you think are worth testing (and you have not analysed in previous tasks T-A6 or T-B4) and try to ping the connectivity. Take a screenshot of each output and add them to the report together with a short description (a paragraph for each).
T-E1: 4 marks



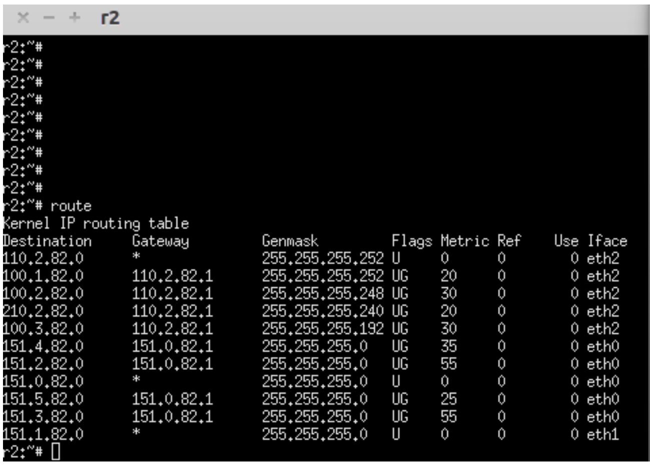
The first pair I chose to test the connectivity of was r2 to r7 as this will test a ping from the backbone area to the new extension. As you can see from the tcpdump all packets were received and an echo reply was successfully sent back.

The second pair I chose was r1 to client2 as this will check the connectivity of the backbone from TaskA. As you can see from the tcpdump all packets were received and an echo reply was successfully sent back.

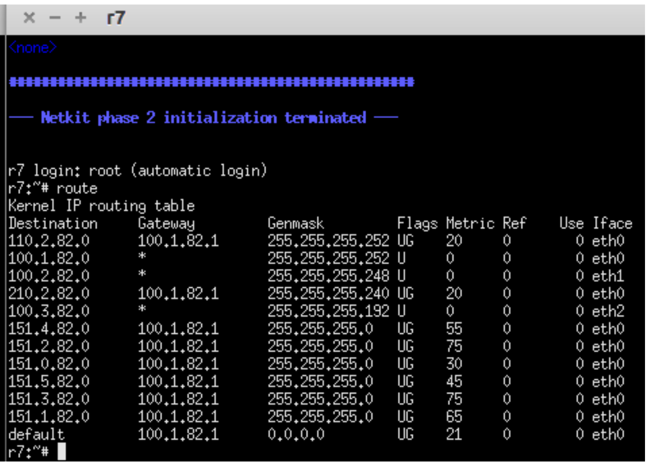


2. Check how the routing works (for the same 2 node pairs) by printing the routing tables. You should also test other commands that you might think are useful for examining the routing. Take a screenshot of each output and add them to the report together with a short description (a paragraph for each).

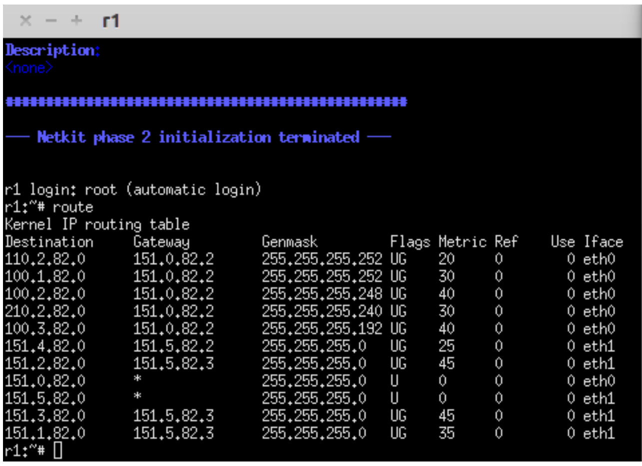
T-E2: 4 marks



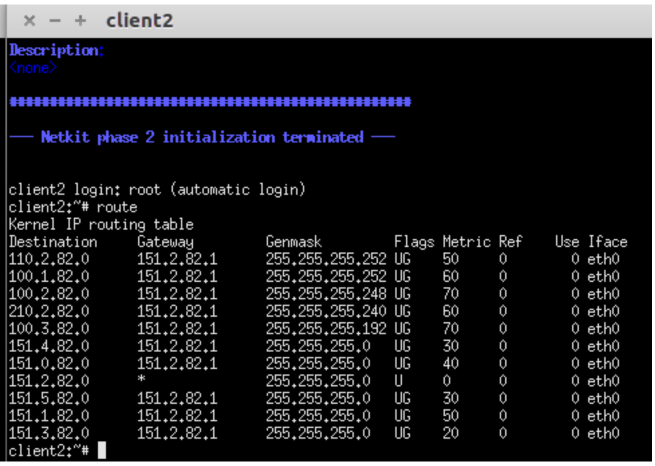
```
r2
r2:~#
r2:~#
r2:~#
r2:~#
r2:~#
r2:~#
r2:~#
r2:~#
r2:~#
r2:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
110.2.82.0      *               255.255.255.252 U     0      0        0 eth2
100.1.82.0      110.2.82.1      255.255.255.252 UG    20     0        0 eth2
100.2.82.0      110.2.82.1      255.255.255.248 UG    30     0        0 eth2
210.2.82.0      110.2.82.1      255.255.255.240 UG    20     0        0 eth2
100.3.82.0      110.2.82.1      255.255.255.192 UG    30     0        0 eth2
151.4.82.0      151.0.82.1      255.255.255.0   UG    35     0        0 eth0
151.2.82.0      151.0.82.1      255.255.255.0   UG    55     0        0 eth0
151.0.82.0      *               255.255.255.0   U     0      0        0 eth0
151.5.82.0      151.0.82.1      255.255.255.0   UG    25     0        0 eth0
151.3.82.0      151.0.82.1      255.255.255.0   UG    55     0        0 eth0
151.1.82.0      *               255.255.255.0   U     0      0        0 eth1
r2:~#
```



```
r7
<none>

################################################

--- Netkit phase 2 initialization terminated ---

r7 login: root (automatic login)
r7:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
110.2.82.0      100.1.82.1      255.255.255.252 UG    20     0        0 eth0
100.1.82.0      *               255.255.255.252 U     0      0        0 eth0
100.2.82.0      *               255.255.255.248 U     0      0        0 eth1
210.2.82.0      100.1.82.1      255.255.255.240 UG    20     0        0 eth0
100.3.82.0      *               255.255.255.192 U     0      0        0 eth2
151.4.82.0      100.1.82.1      255.255.255.0   UG    55     0        0 eth0
151.2.82.0      100.1.82.1      255.255.255.0   UG    75     0        0 eth0
151.0.82.0      100.1.82.1      255.255.255.0   UG    30     0        0 eth0
151.5.82.0      100.1.82.1      255.255.255.0   UG    45     0        0 eth0
151.3.82.0      100.1.82.1      255.255.255.0   UG    75     0        0 eth0
151.1.82.0      100.1.82.1      255.255.255.0   UG    65     0        0 eth0
default         100.1.82.1      0.0.0.0         UG    21     0        0 eth0
r7:~#
```



```
r1
Description:
<none>

################################################

--- Netkit phase 2 initialization terminated ---

r1 login: root (automatic login)
r1:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
110.2.82.0      151.0.82.2      255.255.255.252 UG    20     0        0 eth0
100.1.82.0      151.0.82.2      255.255.255.252 UG    30     0        0 eth0
100.2.82.0      151.0.82.2      255.255.255.248 UG    40     0        0 eth0
210.2.82.0      151.0.82.2      255.255.255.240 UG    30     0        0 eth0
100.3.82.0      151.0.82.2      255.255.255.192 UG    40     0        0 eth0
151.4.82.0      151.5.82.2      255.255.255.0   UG    25     0        0 eth1
151.2.82.0      151.5.82.3      255.255.255.0   UG    45     0        0 eth1
151.0.82.0      *               255.255.255.0   U     0      0        0 eth0
151.5.82.0      *               255.255.255.0   U     0      0        0 eth1
151.3.82.0      151.5.82.3      255.255.255.0   UG    45     0        0 eth1
151.1.82.0      151.5.82.3      255.255.255.0   UG    35     0        0 eth1
r1:~#
```



```
client2
Description:
<none>

################################################

--- Netkit phase 2 initialization terminated ---

client2 login: root (automatic login)
client2:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
110.2.82.0      151.2.82.1      255.255.255.252 UG    50     0        0 eth0
100.1.82.0      151.2.82.1      255.255.255.252 UG    60     0        0 eth0
100.2.82.0      151.2.82.1      255.255.255.248 UG    70     0        0 eth0
210.2.82.0      151.2.82.1      255.255.255.240 UG    60     0        0 eth0
100.3.82.0      151.2.82.1      255.255.255.192 UG    70     0        0 eth0
151.4.82.0      151.2.82.1      255.255.255.0   UG    30     0        0 eth0
151.0.82.0      151.2.82.1      255.255.255.0   UG    40     0        0 eth0
151.2.82.0      *               255.255.255.0   U     0      0        0 eth0
151.5.82.0      151.2.82.1      255.255.255.0   UG    30     0        0 eth0
151.1.82.0      151.2.82.1      255.255.255.0   UG    50     0        0 eth0
151.3.82.0      151.2.82.1      255.255.255.0   UG    20     0        0 eth0
client2:~#
```

The screenshots show that all the VM's have a route to all other VM's on the network. You can see which nodes it will have to go through to get to it's destination via the gateway column. For client 2 all data has to go through 151.2.82.1 (r5).

3. See what happens if one of the routers in the network that was initially used for a route (any of the 2 previous pairs routes) is not available anymore (i.e. shut it down). Add a short description in the report (screenshot and a paragraph) of the result of your routing analysis that indicates a possible route change.
T-E3: 4 marks

```
× − + r2
r2:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
110.2.82.0      *               255.255.255.252 U     0      0        0 eth2
100.1.82.0      110.2.82.1      255.255.255.252 UG    20     0        0 eth2
100.2.82.0      110.2.82.1      255.255.255.248 UG    30     0        0 eth2
210.2.82.0      110.2.82.1      255.255.255.240 UG    20     0        0 eth2
100.3.82.0      110.2.82.1      255.255.255.192 UG    30     0        0 eth2
151.4.82.0      151.0.82.1      255.255.255.0   UG    35     0        0 eth0
151.2.82.0      151.0.82.1      255.255.255.0   UG    55     0        0 eth0
151.0.82.0      *               255.255.255.0   U     0      0        0 eth0
151.5.82.0      151.0.82.1      255.255.255.0   UG    25     0        0 eth0
151.3.82.0      151.0.82.1      255.255.255.0   UG    55     0        0 eth0
151.1.82.0      *               255.255.255.0   U     0      0        0 eth1
r2:~# ping 100.1.82.2
PING 100.1.82.2 (100.1.82.2) 56(84) bytes of data.
From 110.2.82.1 icmp_seq=2 Destination Host Unreachable
From 110.2.82.1 icmp_seq=3 Destination Host Unreachable
From 110.2.82.1 icmp_seq=4 Destination Host Unreachable
^C
--- 100.1.82.2 ping statistics ---
7 packets transmitted, 0 received, +3 errors, 100% packet loss, time 6061ms
, pipe 3
r2:~#
```

```
× − + r7
################################################

--- Netkit phase 2 initialization terminated ---

r7 login: root (automatic login)
r7:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
110.2.82.0      100.1.82.1      255.255.255.252 UG    20     0        0 eth0
100.1.82.0      *               255.255.255.252 U     0      0        0 eth0
100.2.82.0      *               255.255.255.248 U     0      0        0 eth1
210.2.82.0      100.1.82.1      255.255.255.240 UG    20     0        0 eth0
100.3.82.0      *               255.255.255.192 U     0      0        0 eth2
151.4.82.0      100.1.82.1      255.255.255.0   UG    55     0        0 eth0
151.2.82.0      100.1.82.1      255.255.255.0   UG    75     0        0 eth0
151.0.82.0      100.1.82.1      255.255.255.0   UG    30     0        0 eth0
151.5.82.0      100.1.82.1      255.255.255.0   UG    45     0        0 eth0
151.3.82.0      100.1.82.1      255.255.255.0   UG    75     0        0 eth0
151.1.82.0      100.1.82.1      255.255.255.0   UG    65     0        0 eth0
default         100.1.82.1      0.0.0.0         UG    21     0        0 eth0
r7:~# ifconfig eth0 down
r7:~#
```

By using the command ifconfig eth0 on r7 I have shutdown the router. When r2 then tries to ping r7 it attempts to however returns the message destination host unreachable which is to be expected as it can no longer reach the eth port used to connect to it.

4. See what happens if you change the cost values of the route in the network for any of the 2 pairs you examined previously. Calculate the total route cost before and after the change. Add a short description of that in the report (screenshot and a paragraph).
T-E4: 4 marks
Before

```
r2
r2:~#
r2:~#
r2:~#
r2:~#
r2:~#
r2:~#
r2:~#
r2:~#
r2:~#
r2:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
110.2.82.0      *               255.255.255.252 U     0      0        0 eth2
100.1.82.0      110.2.82.1      255.255.255.252 UG    20     0        0 eth2
100.2.82.0      110.2.82.1      255.255.255.248 UG    30     0        0 eth2
210.2.82.0      110.2.82.1      255.255.255.240 UG    20     0        0 eth2
100.3.82.0      110.2.82.1      255.255.255.192 UG    30     0        0 eth2
151.4.82.0      151.0.82.1      255.255.255.0   UG    35     0        0 eth0
151.2.82.0      151.0.82.1      255.255.255.0   UG    55     0        0 eth0
151.0.82.0      *               255.255.255.0   U     0      0        0 eth0
151.5.82.0      151.0.82.1      255.255.255.0   UG    25     0        0 eth0
151.3.82.0      151.0.82.1      255.255.255.0   UG    55     0        0 eth0
151.1.82.0      *               255.255.255.0   U     0      0        0 eth1
r2:~#
```

```
r7
<none>

################################################

--- Netkit phase 2 initialization terminated ---

r7 login: root (automatic login)
r7:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
110.2.82.0      100.1.82.1      255.255.255.252 UG    20     0        0 eth0
100.1.82.0      *               255.255.255.252 U     0      0        0 eth0
100.2.82.0      *               255.255.255.248 U     0      0        0 eth1
210.2.82.0      100.1.82.1      255.255.255.240 UG    20     0        0 eth0
100.3.82.0      *               255.255.255.192 U     0      0        0 eth2
151.4.82.0      100.1.82.1      255.255.255.0   UG    55     0        0 eth0
151.2.82.0      100.1.82.1      255.255.255.0   UG    75     0        0 eth0
151.0.82.0      100.1.82.1      255.255.255.0   UG    30     0        0 eth0
151.5.82.0      100.1.82.1      255.255.255.0   UG    45     0        0 eth0
151.3.82.0      100.1.82.1      255.255.255.0   UG    75     0        0 eth0
151.1.82.0      100.1.82.1      255.255.255.0   UG    65     0        0 eth0
default         100.1.82.1      0.0.0.0         UG    21     0        0 eth0
r7:~#
```

After

```
r2
Description:
<none>

################################################

--- Netkit phase 2 initialization terminated ---

r2 login: root (automatic login)
r2:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
110.2.82.0      *               255.255.255.252 U     0      0        0 eth2
100.1.82.0      110.2.82.1      255.255.255.252 UG    92     0        0 eth2
100.2.82.0      110.2.82.1      255.255.255.248 UG    102    0        0 eth2
210.2.82.0      110.2.82.1      255.255.255.240 UG    92     0        0 eth2
100.3.82.0      110.2.82.1      255.255.255.192 UG    102    0        0 eth2
151.4.82.0      151.0.82.1      255.255.255.0   UG    35     0        0 eth0
151.2.82.0      151.0.82.1      255.255.255.0   UG    55     0        0 eth0
151.0.82.0      *               255.255.255.0   U     0      0        0 eth0
151.5.82.0      151.0.82.1      255.255.255.0   UG    25     0        0 eth0
151.3.82.0      151.0.82.1      255.255.255.0   UG    55     0        0 eth0
151.1.82.0      *               255.255.255.0   U     0      0        0 eth1
r2:~#
```

```
r7
################################################

--- Netkit phase 2 initialization terminated ---

r7 login: root (automatic login)
Last login: Mon May 14 16:02:02 UTC 2018 on tty1
r7:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
110.2.82.0      100.1.82.1      255.255.255.252 UG    92     0        0 eth0
100.1.82.0      *               255.255.255.252 U     0      0        0 eth0
100.2.82.0      *               255.255.255.248 U     0      0        0 eth1
210.2.82.0      100.1.82.1      255.255.255.240 UG    92     0        0 eth0
100.3.82.0      *               255.255.255.192 U     0      0        0 eth2
151.4.82.0      100.1.82.1      255.255.255.0   UG    127    0        0 eth0
151.2.82.0      100.1.82.1      255.255.255.0   UG    147    0        0 eth0
151.0.82.0      100.1.82.1      255.255.255.0   UG    102    0        0 eth0
151.5.82.0      100.1.82.1      255.255.255.0   UG    117    0        0 eth0
151.3.82.0      100.1.82.1      255.255.255.0   UG    147    0        0 eth0
151.1.82.0      100.1.82.1      255.255.255.0   UG    137    0        0 eth0
default         100.1.82.1      0.0.0.0         UG    93     0        0 eth0
r7:~#
```

I increased the interface cost of r6 from 10 to 82 and this has lead to many of the costs which are visible under the metric column increasing by 72.

5. If the server was also connected to network C via a secondary Ethernet card (eth1) what implications would that have on the network? Provide some evidence (screenshot(s) and a couple of paragraphs) to support your answer.
T-E5: 4 marks

This could lead to there being shorter routes from the clients 1 and 2 to the server depending on the cost at the eth ports. In order to implement this, the lab.conf, server,r2,r3,r5 startup and ospf files would have to be adjusted. Lab.conf would need to add the new connection to C for server, server.startup file would need to add the new eth1 and ip address, r2,r3,r5,server.ospf files would need to be adjusted so that they could connect to server directly