CSE 434

Solutions to LAB1/ Sep 2022

Exercise 1: Changing the IP address of an interface. Use the `ifconfig` command to modify the IP address of the `p2p1` interface of computer D.

1. On computer D, run `ifconfig -a` and save the output.
2. Change the IP address of interface `p2p1` of computer D to 10.0.1.16/24.
3. Run `ifconfig -a` again and save the output.

Include the saved output in your report and explain the fields of the `ifconfig` output.

1.1)

```
[root@hostD ~]# ifconfig -a
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:7524 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7524 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:614856 (600.4 KiB)  TX bytes:614856 (600.4 KiB)

p2p1      Link encap:Ethernet  HWaddr 00:13:3B:11:40:F6
          inet addr:10.0.1.14  Bcast:10.0.1.255  Mask:255.255.255.0
          inet6 addr: fe80::213:3bff:fe11:40f6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:16 errors:0 dropped:0 overruns:0 frame:0
          TX packets:25 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1340 (1.3 KiB)  TX bytes:1972 (1.9 KiB)

p2p2      Link encap:Ethernet  HWaddr 00:13:3B:11:40:F7
          inet addr:10.0.1.24  Bcast:10.0.1.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

p4p1      Link encap:Ethernet  HWaddr D4:BE:D9:E4:FF:BE
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

[root@hostD ~]#
```

CSE 434

Solutions to LAB1/ Sep 2022

1.2 and 1.3) Output of `ifconfig -a` on Computer D after IP address change

```
[root@hostD ~]# ifconfig p2p1 10.0.1.16/24
[root@hostD ~]# ifconfig -a
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:7716 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7716 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:630412 (615.6 KiB)  TX bytes:630412 (615.6 KiB)

p2p1      Link encap:Ethernet  HWaddr 00:13:3B:11:40:E6
          inet addr:10.0.1.16  Bcast:10.0.1.255  Mask:255.255.255.0
          inet6 addr: fe80::213:3bff:fe11:40e6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:16 errors:0 dropped:0 overruns:0 frame:0
          TX packets:25 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1340 (1.3 KiB)  TX bytes:1972 (1.9 KiB)

p2p2      Link encap:Ethernet  HWaddr 00:13:3B:11:40:F7
          inet addr:10.0.1.24  Bcast:10.0.1.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

p4p1      Link encap:Ethernet  HWaddr D4:BE:D9:E4:FF:BE
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

[root@hostD ~]#
```

Students should include the saved output in their report and explain the fields of the `ifconfig`

- `Link encap:Ethernet` - This denotes that the interface is an Ethernet related device.
- `HWaddr 00:70:40:42:8A:60` - This is the hardware address or MAC address
- `inet addr` - indicates the machine IP address
- `Bcast` - denotes the broadcast address
- `Mask` - is the network mask which we passed using the netmask option (see above).
- `UP` - This flag indicates that the kernel modules related to the Ethernet interface has been loaded.
- `BROADCAST` - Denotes that the Ethernet device supports broadcasting - a necessary characteristic to obtain IP address via DHCP.
- `RUNNING` - The interface is ready to accept data.
- `MTU` - short form for Maximum Transmission Unit is the size of each packet received by the Ethernet card.
- `Metric` - This option can take a value of 0,1,2,3... with the lower the value the more leverage it has.
- `RX Packets, TX Packets` - The next two lines show the total number of packets received and transmitted respectively.

CSE 434

Solutions to LAB1/ Sep 2022

Exercise 2: Changing netmasks.

1. Configure the interfaces of the hosts as shown in Table 2.
2. Run Wireshark on computer A and capture the packets for the following `ping` commands (running each on the "from" computer). Save the Wireshark output to a text file (clearing the *Packet details* option), and save the output of the `ping` commands, including any error messages.
   (a) From A to C: `ping -c 3 128.143.137.144`
   (b) From A to B: `ping -c 3 128.143.71.21`
   (c) From A to D: `ping -c 3 128.143.137.32`
   (d) From D to A: `ping -c 3 128.143.71.201`
   (e) From B to D: `ping -c 3 128.143.137.32`
   (f) From B to C: `ping -c 3 128.143.137.144`

Include selected output from each `ping` in your report and briefly explain the `ping` result.

Table 2: IP addresses for Exercise 2

Linux PC	IP Address of Ethernet Interface *p2p1*	Network Mask
Computer A	128.143.71.201/16	255.255.0.0
Computer B	128.143.71.21/24	255.255.255.0
Computer C	128.143.137.144/26	255.255.255.192
Computer D	128.143.137.32/26	255.255.255.192

Computer A	128.143.71.201/16	11111111.11111111.01000111.11001001
Computer B	128.143.71.21/24	11111111.11111111.11111111.00010101
Computer C	128.143.137.144/26	11111111.11111111.11111111.11010000
Computer D	128.143.137.32/26	11111111.11111111.11111111.11100000

2.a. Ping A to C. Unsuccessful ping.

A assumes that C is on the same subnet as it (since first 16 bits match). A sends a ICMP echo request to C, and an ARP request for C. But C determines that the A does not belong to the same subnet, hence it does not respond to the ARP request, or send an ICMP echo reply. A repeats the ARP request twice more, then gives up.

Terminal output:

```
[root@hostA ~]# ping -c 3 128.143.137.144
PING 128.143.137.144 (128.143.137.144) 56(84) bytes of data.
From 128.143.71.201 icmp_seq=1 Destination Host Unreachable
From 128.143.71.201 icmp_seq=2 Destination Host Unreachable
From 128.143.71.201 icmp_seq=3 Destination Host Unreachable
--- 128.143.137.144 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 3000ms
```

Wireshark output:

```
No. Time Source Destination Protocol Length Info
1 0.000000000 00:13:3b:21:78:4f ff:ff:ff:ff:ff:ff ARP 42 Who has 128.143.137.144? Tell 128.143.71.201
2 1.000024784 00:13:3b:21:78:4f ff:ff:ff:ff:ff:ff ARP 42 Who has 128.143.137.144? Tell 128.143.71.201
3 2.000021692 00:13:3b:21:78:4f ff:ff:ff:ff:ff:ff ARP 42 Who has 128.143.137.144? Tell 128.143.71.201
```

2.b. Ping A to B. Successful ping.

While the subnet mask of A is smaller than that of B, both the machines are on the same network from A's perspective so it sends an ICMP echo request to B. The high order 24 bits of B and A match (though their masks are of different length) and B thinks A belongs to its subnet and sends the ICMP echo reply. Thus A can successfully ping B

Terminal output:

```
[root@hostA ~]# ping -c 3 128.143.71.21
PING 128.143.71.21 (128.143.71.21) 56(84) bytes of data.
64 bytes from 128.143.71.21: icmp_seq=1 ttl=64 time=1.98 ms
64 bytes from 128.143.71.21: icmp_seq=2 ttl=64 time=0.262 ms
```

```
64 bytes from 128.143.71.21: icmp_seq=3 ttl=64 time=0.255 ms
--- 128.143.71.21 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.255/0.835/1.988/0.815 ms
```

Wireshark output:

```
No. Time Source Destination Protocol Length Info
1 0.000000000 00:13:3b:21:78:4f ff:ff:ff:ff:ff:ff ARP 42 Who has 128.143.71.21? Tell 128.143.71.201
2 0.000191885 00:13:3b:21:78:53 00:13:3b:21:78:4f ARP 60 128.143.71.21 is at 00:13:3b:21:78:53
3 0.000202675 128.143.71.201 128.143.71.21 ICMP 98 Echo (ping) request id=0x7c23, seq=1/256, ttl=64
4 0.000373540 128.143.71.21 128.143.71.201 ICMP 98 Echo (ping) reply id=0x7c23, seq=1/256, ttl=64
5 0.999531632 128.143.71.201 128.143.71.21 ICMP 98 Echo (ping) request id=0x7c23, seq=2/512, ttl=64
6 0.999769627 128.143.71.21 128.143.71.201 ICMP 98 Echo (ping) reply id=0x7c23, seq=2/512, ttl=64
7 1.999077510 128.143.71.201 128.143.71.21 ICMP 98 Echo (ping) request id=0x7c23, seq=3/768, ttl=64
8 1.999310763 128.143.71.21 128.143.71.201 ICMP 98 Echo (ping) reply id=0x7c23, seq=3/768, ttl=64
9 5.000286516 00:13:3b:21:78:53 00:13:3b:21:78:4f ARP 60 Who has 128.143.71.201? Tell 128.143.71.21
10 5.000298927 00:13:3b:21:78:4f 00:13:3b:21:78:53 ARP 42 128.143.71.201 is at 00:13:3b:21:78:4f
```

2.c. A pings D. Unsuccessful ping.

A assumes that D is on the same subnet as it (since first 16 bits match). A sends an ICMP echo request to D. Computer D, which has a subnet mask of 26, compares its high order 26 bits with those of A and determines that the A does not belong to the same subnet, hence it does not respond to the ICMP echo request. A repeats the ARP request twice more, then gives up.

Terminal output:

```
[root@hostA ~]# ping -c 3 128.143.137.32
PING 128.143.137.32 (128.143.137.32) 56(84) bytes of data.
From 128.143.71.201 icmp_seq=1 Destination Host Unreachable
From 128.143.71.201 icmp_seq=2 Destination Host Unreachable
From 128.143.71.201 icmp_seq=3 Destination Host Unreachable
--- 128.143.137.32 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 3000ms
```

Wireshark output:

```
No. Time Source Destination Protocol Length Info
1 0.000000000 00:13:3b:21:78:4f ff:ff:ff:ff:ff:ff ARP 42 Who has 128.143.137.32? Tell 128.143.71.201
2 1.000016719 00:13:3b:21:78:4f ff:ff:ff:ff:ff:ff ARP 42 Who has 128.143.137.32? Tell 128.143.71.201
3 2.000007955 00:13:3b:21:78:4f ff:ff:ff:ff:ff:ff ARP 42 Who has 128.143.137.32? Tell 128.143.71.201
```

2.d. D pings A. Unsuccessful ping.

D compares the IP addresses using its /26 subnet mask and determines that A does not belong to its subnet because there is not a match. Hence it displays the error message “Network is unreachable,” and Wireshark shows that no packets were sent, i.e., it does not even send the ICMP echo request.

Terminal output:

```
[root@hostD ~]# ping -c 3 128.143.71.201
connect: Network is unreachable
```

Wireshark output:

```
No output captured in the wireshark at hostA
```

2.e. B pings D. Unsuccessful ping.

B compares the IP addresses using its /24 subnet mask and determines that D does not belong to its subnet (no match on the high order 24 bits). Hence it displays the error message “Network is unreachable,” and Wireshark shows that no packets were sent, i.e., it does not even send the ICMP echo request out.

Terminal output:

```
[root@hostB ~]# ping -c 3 128.143.137.32
connect: Network is unreachable
```

Wireshark output:

```
No output captured in the wireshark at hostA
```

CSE 434

Solutions to LAB1/ Sep 2022

2.f. B pings C. Unsuccessful ping.

B compares the IP addresses using its /24 subnet mask and determines that C does not belong to its subnet. Hence it displays the error message "Network is unreachable," and Wireshark shows that no packets were sent, i.e., the ICMP echo request message is never sent.

Terminal output:

```
[root@hostB ~]# ping -c 3 128.143.137.144
connect: Network is unreachable
```

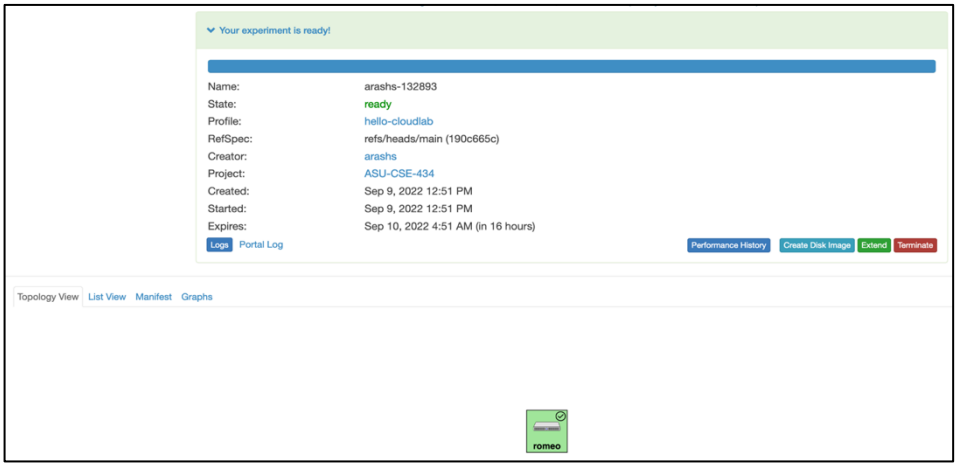
Wireshark output:

```
No output captured in the wireshark at hostA
```

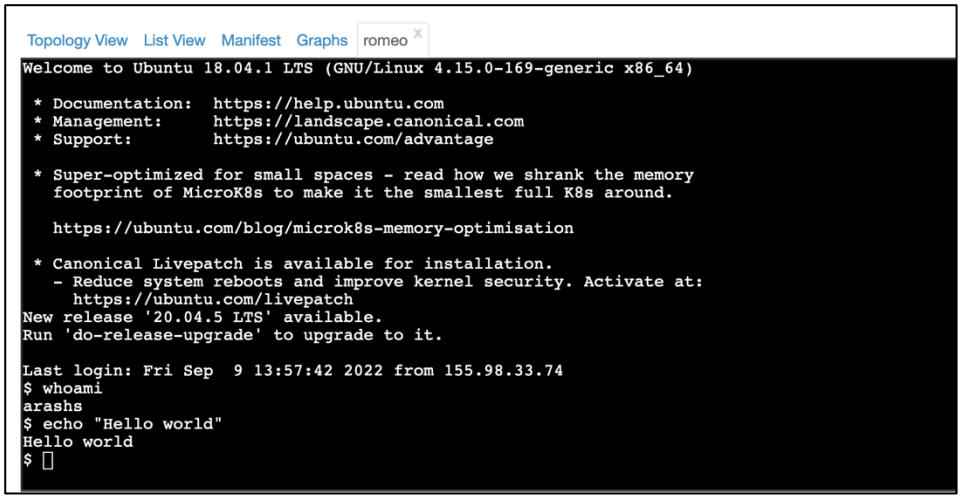
Exercise 3: Testing `ssh` keys. *As stated in §1.2, be sure that all screen shots include the ASUrite id of a group member, i.e., set your login prompt when you* **`ssh`** *into a VM.*

1. Once your topology is ready (*i.e.*, in the Topology View, your VM will be green and have a "⊙" icon in the top right corner), take a screen shot of your topology and include it in your report.
2. Continue to follow the instructions, logging into your VM using the terminal in the CloudLab web portal. Take a screen shot of this terminal shell after executing the `echo` command.
3. You should also try using your own terminal application, and using VNC.
4. IMPORTANT: Follow the instructions in the section *Release resources* to return the resources used by your VM in this slice when you are finished.

3.1



3.2

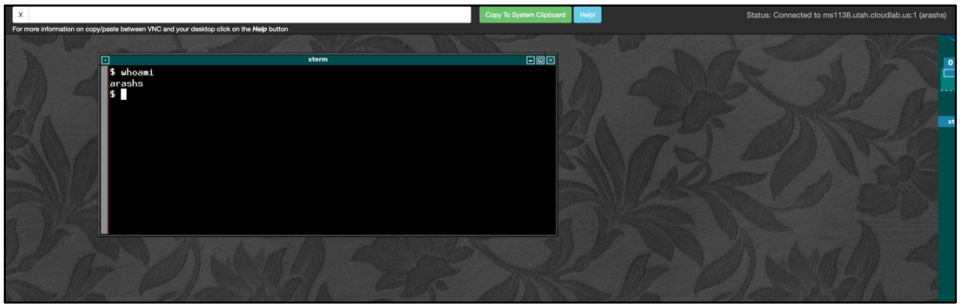


CSE 434

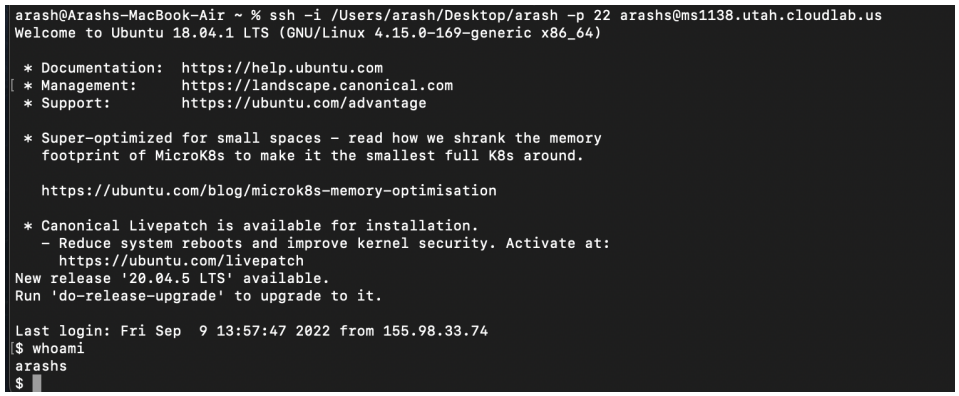
Solutions to LAB1/ Sep 2022

3.3

VNC view:



Terminal view:



3.4 no need to show

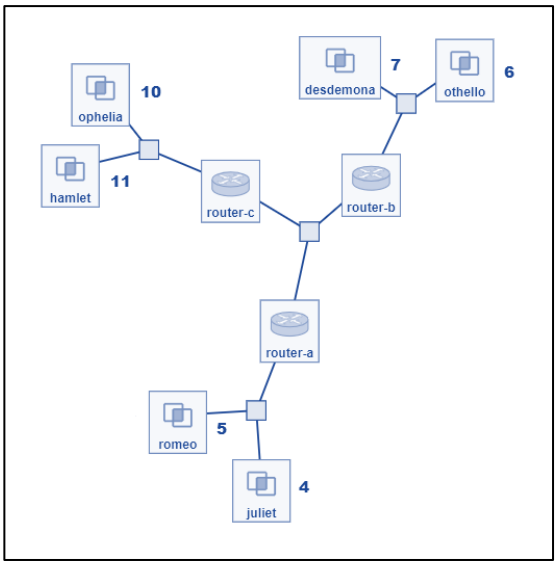
> **Exercise 4: Design Subnets.** Solve the subnet design problem posed in the section **Challenge: Design Subnets** that satisfies the requirements given.
>
> *For each of the three subnets* in the design problem, provide a table in your report giving:
>
> 1. The subnet mask.
> 2. The network address.
> 3. The smallest IPv4 address that may be assigned to a host in the subnet.
> 4. The broadcast address for the subnet.
> 5. The highest IPv4 address that may be assigned to a host in the subnet.
>
> Be sure to follow the conventions on addressing given, *i.e.*, assign the lowest IP address in the subnet to the LAN-facing interface of the router, assign the highest IP address in the subnet to one node, and any other IP address in the subnet to the other node.

4.

LAN	Net Address	Subnet Mask	Smallest IP	Highest IP	Broadcast Address
A Romeo Node 5 Juliet Node 4	10.10.172.128	255.255.255.192	10.10.172.130	10.10.172.190	10.10.172.191
B Othello Node 6 Desdemona Node 7	10.10.172.0	255.255.255.128	10.10.172.2	10.10.172.126	10.10.172.127
C Hamlet Node 11 Ophelia Node 10	10.10.172.192	255.255.255.224	10.10.172.194	10.10.172.222	10.10.172.223

CSE 434

Solutions to LAB1/ Sep 2022



Exercise 5: Implement your Design. *Remember that all screen shots must include the ASUrite id of a group member, i.e., set your login prompt when you* `ssh` *into a VM.*

Configure the IPv4 address and subnet mask of each host in each LAN using your solution to Exercise 4. Recall the instructions on configuring IP addresses from §3.2.

1. Take a screen shot showing the output of the `ping -c 5 IP` command between two hosts in the same LAN, *e.g.*, between `romeo` and `juliet` on LAN A, and include each one in your report. (You should have 3 screen shots, one for each LAN.)
2. Take a screen shot showing the output of the `ping -c 5 IP` between a host in LAN A and a host in LAN B (*e.g.*, between `romeo` and `othello`), between a host in LAN B and a host in LAN C, and between a host in LAN C and a host in LAN A, and include them in your report. (Again, you should have 3 screen shots.)

If your `ping` output is not as expected, you may need to review your solution to the design problem (Exercise 4) and/or check your node configurations.

5.1.

From node 4/Juliet to other node on LAN A

```
PING 10.10.172.190 (10.10.172.190) 56(84) bytes of data.
64 bytes from 10.10.172.190: icmp_seq=1 ttl=64 time=0.036 ms
64 bytes from 10.10.172.190: icmp_seq=2 ttl=64 time=0.050 ms
64 bytes from 10.10.172.190: icmp_seq=3 ttl=64 time=0.040 ms
64 bytes from 10.10.172.190: icmp_seq=4 ttl=64 time=0.052 ms
64 bytes from 10.10.172.190: icmp_seq=5 ttl=64 time=0.048 ms
```

From node 6/Othello to other node on LAN B.

```
PING 10.10.172.126 (10.10.172.126) 56(84) bytes of data.
64 bytes from 10.10.172.126: icmp_seq=1 ttl=64 time=1.28 ms
64 bytes from 10.10.172.126: icmp_seq=2 ttl=64 time=0.532 ms
64 bytes from 10.10.172.126: icmp_seq=3 ttl=64 time=0.545 ms
64 bytes from 10.10.172.126: icmp_seq=4 ttl=64 time=0.524 ms
64 bytes from 10.10.172.126: icmp_seq=5 ttl=64 time=0.473 ms
```

From node 11/Hamlet to other node on LAN C.

```
PING 10.10.172.222 (10.10.172.222) 56(84) bytes of data.
64 bytes from 10.10.172.222: icmp_seq=1 ttl=64 time=1.09 ms
64 bytes from 10.10.172.222: icmp_seq=2 ttl=64 time=0.510 ms
64 bytes from 10.10.172.222: icmp_seq=3 ttl=64 time=0.500 ms
64 bytes from 10.10.172.222: icmp_seq=4 ttl=64 time=0.500 ms
64 bytes from 10.10.172.222: icmp_seq=5 ttl=64 time=0.448 ms
```

5.2.

Ping between host in LAN A to host in LAN B

```
PING 10.10.172.190 (10.10.172.190) 56(84) bytes of data.
From 10.164.0.21 icmp_seq=1 Time to live exceeded
From 10.164.0.20 icmp_seq=2 Time to live exceeded
From 10.164.0.20 icmp_seq=3 Time to live exceeded
From 10.164.0.20 icmp_seq=4 Time to live exceeded
From 10.164.0.20 icmp_seq=5 Time to live exceeded
```

Ping between host in LAN B to host in LAN C

```
PING 10.10.172.2 (10.10.172.2) 56(84) bytes of data.
From 10.164.0.21 icmp_seq=1 Time to live exceeded
From 10.164.0.21 icmp_seq=2 Time to live exceeded
From 10.164.0.20 icmp_seq=3 Time to live exceeded
From 10.164.0.20 icmp_seq=4 Time to live exceeded
From 10.164.0.21 icmp_seq=5 Time to live exceeded
```

Ping between host in LAN C to host in LAN A

```
PING 10.10.172.194 (10.10.172.194) 56(84) bytes of data.
From 10.164.0.20 icmp_seq=1 Time to live exceeded
From 10.164.0.20 icmp_seq=2 Time to live exceeded
From 10.164.0.21 icmp_seq=3 Time to live exceeded
From 10.164.0.20 icmp_seq=4 Time to live exceeded
From 10.164.0.20 icmp_seq=5 Time to live exceeded
```

CSE 434

Solutions to LAB1/ Sep 2022

Exercise 6: Add Routing Rules. Configure the route on each host in each LAN, and also on each router.

1. Take a screen shot of the output of the `route -n` command on each node and on each router and include it in your report. (You should have 9 small screen shots.)
2. Take a screen shot of the output of the `ping -c 5 IP` command between a host in LAN A and a host in LAN B, between a host in LAN B and a host in LAN C, and between a host in LAN C and a host in LAN A and include them in your report. (You should have 3 screen shots.)
3. Follow the instructions to configure the routing rules for the 10.10.100.0/24 network. Now, take a screen shot of the output of the `traceroute IP` command between two nodes in different LANs and include it in your report. (The `traceroute` should be able to trace the route taken.)

Romeo Node 5
Juliet Node 4

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	172.16.0.1	0.0.0.0	UG	1024	0	0	eth0
10.10.172.0	10.10.172.129	255.255.255.128	UG	0	0	0	eth1
10.10.172.128	0.0.0.0	255.255.255.192	U	0	0	0	eth1
10.10.172.192	10.10.172.129	255.255.255.224	UG	0	0	0	eth1
172.16.0.0	0.0.0.0	255.240.0.0	U	0	0	0	eth0
172.16.0.1	0.0.0.0	255.255.255.255	UH	1024	0	0	eth0

Router a

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	172.16.0.1	0.0.0.0	UG	1024	0	0	eth0
10.0.0.0	10.10.100.2	255.0.0.0	UG	0	0	0	eth2
10.10.100.0	0.0.0.0	255.255.255.0	U	0	0	0	eth2
10.10.172.0	10.10.100.2	255.255.255.128	UG	0	0	0	eth2
10.10.172.128	0.0.0.0	255.255.255.192	U	0	0	0	eth1
10.10.172.128	10.10.100.3	255.255.255.128	UG	0	0	0	eth2
10.10.172.192	10.10.100.3	255.255.255.224	UG	0	0	0	eth2
172.16.0.0	0.0.0.0	255.240.0.0	U	0	0	0	eth0
172.16.0.1	0.0.0.0	255.255.255.255	UH	1024	0	0	eth0

Othello Node 6
Desdemona Node 7

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	172.16.0.1	0.0.0.0	UG	1024	0	0	eth0
10.10.172.0	0.0.0.0	255.255.255.128	U	0	0	0	eth1
10.10.172.128	10.10.172.1	255.255.255.192	UG	0	0	0	eth1
10.10.172.192	10.10.172.1	255.255.255.224	UG	0	0	0	eth1
172.16.0.0	0.0.0.0	255.240.0.0	U	0	0	0	eth0
172.16.0.1	0.0.0.0	255.255.255.255	UH	1024	0	0	eth0

Router b

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	172.16.0.1	0.0.0.0	UG	1024	0	0	eth0
10.0.0.0	10.10.100.1	255.0.0.0	UG	0	0	0	eth2
10.10.100.0	0.0.0.0	255.255.255.0	U	0	0	0	eth2
10.10.172.0	0.0.0.0	255.255.255.128	U	0	0	0	eth1
10.10.172.128	10.10.100.1	255.255.255.192	UG	0	0	0	eth2
10.10.172.192	10.10.100.3	255.255.255.224	UG	0	0	0	eth2
10.10.172.192	10.10.100.3	255.255.255.192	UG	0	0	0	eth2
172.16.0.0	0.0.0.0	255.240.0.0	U	0	0	0	eth0
172.16.0.1	0.0.0.0	255.255.255.255	UH	1024	0	0	eth0

Hamlet Node 11
Ophelia Node 10

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	172.16.0.1	0.0.0.0	UG	1024	0	0	eth0
10.10.172.0	10.10.172.193	255.255.255.128	UG	0	0	0	eth1
10.10.172.128	10.10.172.193	255.255.255.192	UG	0	0	0	eth1
10.10.172.192	0.0.0.0	255.255.255.224	U	0	0	0	eth1
172.16.0.0	0.0.0.0	255.240.0.0	U	0	0	0	eth0
172.16.0.1	0.0.0.0	255.255.255.255	UH	1024	0	0	eth0

Router c

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	172.16.0.1	0.0.0.0	UG	1024	0	0	eth0
10.0.0.0	10.10.100.1	255.0.0.0	UG	0	0	0	eth2
10.10.100.0	0.0.0.0	255.255.255.0	U	0	0	0	eth2
10.10.172.0	10.10.100.2	255.255.255.128	UG	0	0	0	eth2
10.10.172.128	10.10.100.1	255.255.255.192	UG	0	0	0	eth2
10.10.172.192	0.0.0.0	255.255.255.224	U	0	0	0	eth1
172.16.0.0	0.0.0.0	255.240.0.0	U	0	0	0	eth0
172.16.0.1	0.0.0.0	255.255.255.255	UH	1024	0	0	eth0

Hamlet to Juliet (node 11 to node 4)

```
PING 10.10.172.190 (10.10.172.190) 56(84) bytes of data.
64 bytes from 10.10.172.190: icmp_seq=1 ttl=62 time=1.46 ms
64 bytes from 10.10.172.190: icmp_seq=2 ttl=62 time=1.46 ms
64 bytes from 10.10.172.190: icmp_seq=3 ttl=62 time=1.35 ms
64 bytes from 10.10.172.190: icmp_seq=4 ttl=62 time=1.19 ms
64 bytes from 10.10.172.190: icmp_seq=5 ttl=62 time=1.18 ms

--- 10.10.172.190 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
```

Romeo to Othello (node 5 to node 6)

```
PING 10.10.172.2 (10.10.172.2) 56(84) bytes of data.
64 bytes from 10.10.172.2: icmp_seq=1 ttl=62 time=1.12 ms
64 bytes from 10.10.172.2: icmp_seq=2 ttl=62 time=1.36 ms
64 bytes from 10.10.172.2: icmp_seq=3 ttl=62 time=1.18 ms
64 bytes from 10.10.172.2: icmp_seq=4 ttl=62 time=1.22 ms
64 bytes from 10.10.172.2: icmp_seq=5 ttl=62 time=1.36 ms

--- 10.10.172.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 1.121/1.247/1.364/0.097 ms
```

Othello to Desdemona (node 10 to node 7)

```
PING 10.10.172.126 (10.10.172.126) 56(84) bytes of data.
64 bytes from 10.10.172.126: icmp_seq=1 ttl=62 time=1.45 ms
64 bytes from 10.10.172.126: icmp_seq=2 ttl=62 time=1.36 ms
64 bytes from 10.10.172.126: icmp_seq=3 ttl=62 time=1.33 ms
64 bytes from 10.10.172.126: icmp_seq=4 ttl=62 time=1.18 ms
64 bytes from 10.10.172.126: icmp_seq=5 ttl=62 time=1.06 ms

--- 10.10.172.126 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
```

6.3.

Hamlet to Juliet (node 11 to node 4)

```
traceroute to 10.10.172.190 (10.10.172.190), 30 hops max, 60 byte packets
 1  router-C-link-4 (10.10.172.193)  1.040 ms  0.958 ms  0.887 ms
 2  router-A-link-1 (10.10.100.1)  1.671 ms  1.610 ms  1.546 ms
 3  node-4-link-2 (10.10.172.190)  1.949 ms  1.953 ms  1.884 ms
```

Or

```
traceroute to 10.10.172.126 (10.10.172.126), 30 hops max, 60 byte packets
 1  10.10.172.129 (10.10.172.129)  0.174 ms  0.352 ms  0.338 ms
 2  router-b-link-1 (10.10.100.2)  0.736 ms  0.727 ms  0.716 ms
 3  10.10.172.126 (10.10.172.126)  0.881 ms  0.871 ms  0.857 ms
```