Name: Kwa Li Ying
Student ID: 1003833

50.012 Networks Lab 6

Q1:

```
mininet> net
extGW extGW-eth0:extS1-eth3 extGW-eth1:intGW-eth1
extH1 extH1-eth0:extS1-eth1
extH2 extH2-eth0:extS1-eth2
h0 h0-eth0:s2-eth3
h1 h1-eth0:s2-eth4
h2 h2-eth0:s2-eth5
h3 h3-eth0:s2-eth6
h4 h4-eth0:s2-eth7
intGW intGW-eth0:s2-eth2 intGW-eth1:extGW-eth1
srv1 srv1-eth0:s1-eth1
srv2 srv2-eth0:s1-eth2
extS1 lo:   extS1-eth1:extH1-eth0 extS1-eth2:extH2-eth0 extS1-eth3:extGW-eth0
s1 lo:   s1-eth1:srv1-eth0 s1-eth2:srv2-eth0 s1-eth3:s2-eth1
s2 lo:   s2-eth1:s1-eth3 s2-eth2:intGW-eth0 s2-eth3:h0-eth0 s2-eth4:h1-eth0 s2-eth5:h2-eth0 s2-eth6:h3-eth0 s2-eth7:h4-eth0
c0
```

```
Node: intGW
root@bowen-VirtualBox:~/Documents/lab6# ifconfig
intGW-eth0 Link encap:Ethernet  HWaddr 00:00:00:00:00:09
          inet addr:10.0.0.1  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::200:ff:fe00:9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:819 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:149561 (149.5 KB)  TX bytes:906 (906.0 B)

intGW-eth1 Link encap:Ethernet  HWaddr da:0d:f0:1c:3a:23
          inet addr:2.2.2.2  Bcast:2.2.2.255  Mask:255.255.255.0
          inet6 addr: fe80::d80d:f0ff:fe1c:3a23/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:11 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1170 (1.1 KB)  TX bytes:906 (906.0 B)

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:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@bowen-VirtualBox:~/Documents/lab6#
```

From above, the internal gateway router's interface intGW-eth0 is connected to switch s2's interface s2-eth2. From intGW's ifconfig output and from the interface intGW-eth0 section, the IP subnet chosen for the hosts is 10.0.0.0/24.

Q2:

```
Node: srv1
root@bowen-VirtualBox:~/Documents/lab6# ifconfig
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:3 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:228 (228.0 B)  TX bytes:228 (228.0 B)

srv1-eth0 Link encap:Ethernet  HWaddr 00:00:00:00:00:0a
          inet addr:10.0.0.10  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::200:ff:fe00:a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:990 errors:0 dropped:0 overruns:0 frame:0
          TX packets:64 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:140840 (140.8 KB)  TX bytes:9052 (9.0 KB)
```

```
Node: srv2
root@bowen-VirtualBox:~/Documents/lab6# ifconfig
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:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

srv2-eth0 Link encap:Ethernet  HWaddr 00:00:00:00:00:0b
          inet addr:10.0.0.11  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::200:ff:fe00:b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1136 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:190387 (190.3 KB)  TX bytes:1338 (1.3 KB)
```
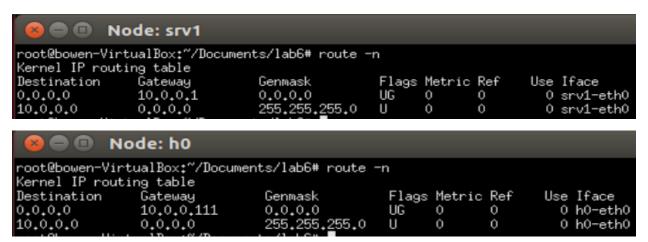
Yes, servers srv1 and srv2 are in the same subnet. Their IP addresses are 10.0.0.10 and 10.0.0.11 respectively, and their IP masks are both 255.255.255.0, meaning they are both in the subnet 10.0.0.0/24.

Name: Kwa Li Ying
Student ID: 1003833

Q3:

```
mininet> h1 tracepath -n srv1
 1?: [LOCALHOST]                                        pmtu 1500
 1:   10.0.0.10                                          2.089ms reached
 1:   10.0.0.10                                          1.090ms reached
      Resume: pmtu 1500 hops 1 back 1
```

The tracepath command only traces router hops and does not take into consideration link-layer switches.

Q4:

```
⊗⊖⊡   Node: srv1

root@bowen-VirtualBox:~/Documents/lab6# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.0.0.1        0.0.0.0         UG    0      0        0 srv1-eth0
10.0.0.0        0.0.0.0         255.255.255.0   U     0      0        0 srv1-eth0
```

```
⊗⊖⊡   Node: h0

root@bowen-VirtualBox:~/Documents/lab6# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.0.0.111      0.0.0.0         UG    0      0        0 h0-eth0
10.0.0.0        0.0.0.0         255.255.255.0   U     0      0        0 h0-eth0
```

The gateway for srv1 and srv2 is 10.0.0.1, and the gateway for h0 to h4 is 10.0.0.111.

Q5:

```
⊗⊖⊡   Node: h1

root@bowen-VirtualBox:~/Documents/lab6# ping 8.8.8.2
PING 8.8.8.2 (8.8.8.2) 56(84) bytes of data.
From 10.0.0.105 icmp_seq=1 Destination Host Unreachable
From 10.0.0.105 icmp_seq=2 Destination Host Unreachable
From 10.0.0.105 icmp_seq=3 Destination Host Unreachable
^C
--- 8.8.8.2 ping statistics ---
5 packets transmitted, 0 received, +3 errors, 100% packet loss, time 4001ms
pipe 3
```

No, h1 cannot ping test.net (8.8.8.2). The gateway IP specified for h1 (as well as h0, h2, h3, h4) is wrong, since 10.0.0.111 does not exist as an IP address for any interface in the network.

Q6:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.0.105	10.0.0.10	DHCP	342	DHCP Release - Transaction ID 0x42410e45
2	0.911693746	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xf0f49a6c
3	0.912939594	10.0.0.10	10.0.0.105	DHCP	342	DHCP Offer - Transaction ID 0xf0f49a6c
4	0.913227295	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0xf0f49a6c
5	0.914835165	10.0.0.10	10.0.0.105	DHCP	356	DHCP ACK - Transaction ID 0xf0f49a6c

Yes, a DHCP server is running in the local network, on srv1. This can be seen from the Wireshark screenshot, which is captured during the time period where the commands `h1 dhclient -r h1-eth0` and `h1 dhclient h1-eth0` are executed. The DHCP ACK packets sent to h1 come from 10.0.0.10, which is the IP address of srv1.

Q7:

Yes, the third line '`dhcp-option=3,10.0.0.111`' has a wrong gateway IP specified. 10.0.0.111 should be changed to 10.0.0.1. After modifying, the conf file is as shown:

```
bowen@bowen-VirtualBox: ~/Documents/lab6
  GNU nano 2.2.6              File: srv1DHCP.conf

interface=srv1-eth0
dhcp-range=srv1-eth0,10.0.0.100,10.0.0.200,255.255.255.0,12h
dhcp-option=3,10.0.0.1
dhcp-option=option:dns-server,0.0.0.0,8.8.8.8
dhcp-authoritative
```

Q8:

```
Node: h1
root@bowen-VirtualBox:~/Documents/lab6# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=62 time=3.14 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=62 time=5.04 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=62 time=0.251 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=62 time=0.144 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 0.144/2.147/5.045/2.062 ms
```

Yes, h1 can reach Google (8.8.8.8) now.

Name: Kwa Li Ying
Student ID: 1003833

Q9:

Before the changes were applied, h1 could ping test.net.

```
mininet> h1 ping test.net
PING test.net (8.8.8.2) 56(84) bytes of data.
64 bytes from 8.8.8.2: icmp_seq=1 ttl=62 time=3.39 ms
64 bytes from 8.8.8.2: icmp_seq=2 ttl=62 time=0.797 ms
64 bytes from 8.8.8.2: icmp_seq=3 ttl=62 time=0.157 ms
^C
--- test.net ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.157/1.449/3.395/1.400 ms
```

This is because when mininet is started at the very beginning, dnsmasq configures /etc/resolv.conf to have the DNS server at 8.8.8.8, so h1 could ping test.net.

After the changes were applied, h1 could not ping test.net.

```
mininet> h1 ping test.net
ping: unknown host test.net
```

The hostname test.net cannot be resolved into its IP address (8.8.8.2). The issue is confirmed to be a DNS issue because h1 can ping 8.8.8.2

```
mininet> h1 ping 8.8.8.2
PING 8.8.8.2 (8.8.8.2) 56(84) bytes of data.
64 bytes from 8.8.8.2: icmp_seq=1 ttl=62 time=2.17 ms
^C
--- 8.8.8.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.178/2.178/2.178/0.000 ms
```

and that dig and nslookup cannot resolve its hostname.

```
mininet> h1 dig test.net

; <<>> DiG 9.9.5-3-Ubuntu <<>> test.net
;; global options: +cmd
;; connection timed out; no servers could be reached
```

```
mininet> h1 nslookup test.net
;; connection timed out; no servers could be reached
```

This happens because when restart the network manager service, the contents of the /etc/resolv.conf file will be automatically written back to 127.0.0.1, and overwrite whatever we wrote to it when the network manager service was stopped.

The IP of test.net is 8.8.8.2.

Q10:

This node that provides NAT for the hosts with private IP addresses is intGW.

This is a Wireshark screenshot in intGW when the command `h1 ping 8.8.8.8` is executed.

No.	Time	Source	Destination	Protocol	Length Info
91	43.064124800	2.2.2.2	8.8.8.8	DNS	70 Standard query 0x6a0b A test.net
92	43.066505964	8.8.8.8	2.2.2.2	DNS	86 Standard query response 0x6a0b A test.net A 8...
93	43.066520252	8.8.8.8	10.0.0.105	DNS	86 Standard query response 0x6a0b A test.net A 8...
94	43.079123960	10.0.0.105	8.8.8.2	ICMP	100 Echo (ping) request id=0x103d, seq=1/256, ttl...
95	43.079150709	2.2.2.2	8.8.8.2	ICMP	100 Echo (ping) request id=0x103d, seq=1/256, ttl...
96	43.084449511	8.8.8.2	2.2.2.2	ICMP	100 Echo (ping) reply id=0x103d, seq=1/256, ttl...
97	43.084462555	8.8.8.2	10.0.0.105	ICMP	100 Echo (ping) reply id=0x103d, seq=1/256, ttl...
98	43.091066983	10.0.0.105	8.8.8.8	DNS	82 Standard query 0x218b PTR 2.8.8.8.in-addr.arpa
99	43.091090201	2.2.2.2	8.8.8.8	DNS	82 Standard query 0x218b PTR 2.8.8.8.in-addr.arpa
100	43.096339148	8.8.8.8	2.2.2.2	DNS	82 Standard query response 0x218b Refused PTR 2.8...
101	43.096351045	8.8.8.8	10.0.0.105	DNS	82 Standard query response 0x218b Refused PTR 2.8...
102	43.098360089	10.0.0.105	8.8.8.8	DNS	82 Standard query 0x218b PTR 2.8.8.8.in-addr.arpa
103	43.098403963	2.2.2.2	8.8.8.8	DNS	82 Standard query 0x218b PTR 2.8.8.8.in-addr.arpa
104	43.101865181	8.8.8.8	2.2.2.2	DNS	82 Standard query response 0x218b Refused PTR 2.8...

In the enableNAT(net, 'intGW') function in net.py, IP masquerading and forwarding is set up in intGW. The results can be seen in the Wireshark screenshot above, where packets coming from 10.0.0.x will have their source IP translated by NAT from 10.0.0.x to 2.2.2.2 and packets coming from 8.8.8.x will have their destination IP translated by NAT from 2.2.2.2 to 10.0.0.x.

Q11:

To block all traffic from srv2 specifically to intGW, the command executed on intGW is
`iptables -I FORWARD -s 10.0.0.11 -j DROP`

```
  ⊗ ⊖ ▢   Node: intGW
root@bowen-VirtualBox:~/Documents/lab6# iptables -I FORWARD -s 10.0.0.11 -j DROP
root@bowen-VirtualBox:~/Documents/lab6# iptables -L -v
Chain INPUT (policy ACCEPT 2 packets, 656 bytes)
 pkts bytes target     prot opt in      out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in      out     source               destination
    0     0 DROP       all  --  any     any     10.0.0.11            anywhere
   26  1744 ACCEPT     all  --  intGW-eth0 intGW-eth1  10.0.0.0/24            anywhere             ctstate NEW
   60  4664 ACCEPT     all  --  any     any     anywhere             anywhere             ctstate RELATED,ESTABLISHED

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in      out     source               destination
```

After this is done, srv2 can no longer ping 8.8.8.8.

```
mininet> srv2 ping 8.8.8.8
^CPING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.

--- 8.8.8.8 ping statistics ---
12 packets transmitted, 0 received, 100% packet loss, time 10999ms
```