NS Lab2 Report

1. Step 1-1 :

```
h3 doesn't have connectivity to 192.168.3.2
h4 doesn't have connectivity to 192.168.1.65
h4 doesn't have connectivity to 192.168.1.66
h5 doesn't have connectivity to 192.168.1.65
h5 doesn't have connectivity to 192.168.1.66
WRONG ANSWER
mininet> h2 ping h3 -c5
PING 192.168.1.66 (192.168.1.66) 56(84) bytes of data.
64 bytes from 192.168.1.66: icmp_seq=1 ttl=64 time=0.596 ms
64 bytes from 192.168.1.66: icmp_seq=2 ttl=64 time=0.104 ms
64 bytes from 192.168.1.66: icmp_seq=3 ttl=64 time=0.109 ms
64 bytes from 192.168.1.66: icmp_seq=4 ttl=64 time=0.105 ms
64 bytes from 192.168.1.66: icmp_seq=5 ttl=64 time=0.112 ms

--- 192.168.1.66 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4104ms
rtt min/avg/max/mdev = 0.104/0.205/0.596/0.195 ms
mininet> h2 ping h4 -c5
PING 192.168.3.1 (192.168.3.1) 56(84) bytes of data.
From 192.168.1.65 icmp_seq=1 Destination Host Unreachable
From 192.168.1.65 icmp_seq=2 Destination Host Unreachable
From 192.168.1.65 icmp_seq=3 Destination Host Unreachable
From 192.168.1.65 icmp_seq=4 Destination Host Unreachable
From 192.168.1.65 icmp_seq=5 Destination Host Unreachable

--- 192.168.3.1 ping statistics ---
5 packets transmitted, 0 received, +5 errors, 100% packet loss, time 4093ms
pipe 4
mininet>
```

   a. h2 可以 ping 到 h3，因為 h2 和 h3 在同一個 LAN 裡面，不需要經過 router 轉送。

   b. h2 不能 ping 到 h4，因為它們不在同一個 LAN，需要經過 router，而 router 還沒設定好。

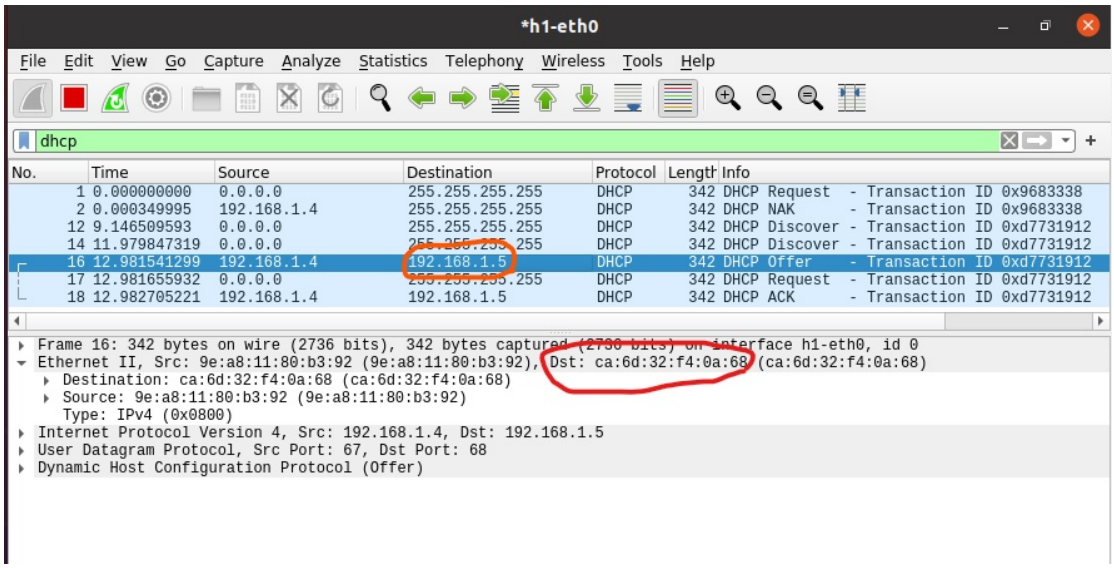
2. Topology except h1 :

```
yucheng@ubuntu:~/Documents/ns_hw2$ sudo python2 topology.py
[sudo] password for yucheng:
h1 doesn't have connectivity to 192.168.1.65
h1 doesn't have connectivity to 192.168.1.66
h1 doesn't have connectivity to 192.168.3.1
h1 doesn't have connectivity to 192.168.3.2
WRONG ANSWER
mininet>
```

3. Run DHCP :

```
yucheng@ubuntu:~/Documents/ns_hw2$ sudo python2 topology.py
[+] Run DHCP server
h1 doesn't have connectivity to 192.168.1.65
h1 doesn't have connectivity to 192.168.1.66
h1 doesn't have connectivity to 192.168.3.1
h1 doesn't have connectivity to 192.168.3.2
WRONG ANSWER
mininet> h1 dhclient h1-eth0
mininet> exit
ACCEPT
[-] Killing DHCP server
yucheng@ubuntu:~/Documents/ns_hw2$
```

Capture DHCP messages and show IPs and MACs :



h1 的 IP 為橘色圈圈，192.168.1.5；MAC 為紅色圈圈，

ca:6d:32:f4:0a:68。接下來輸入 h1 ifconfig 做檢查:

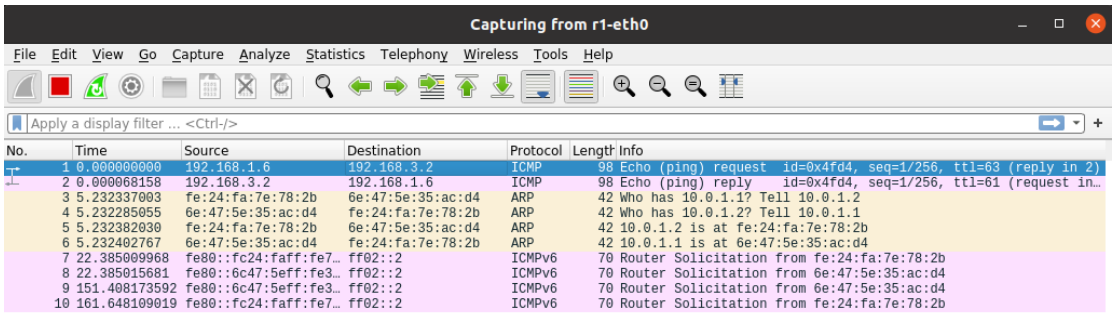
```
yucheng@ubuntu:~/Documents/ns_hw2$ sudo python2 topology.py
[+] Run DHCP server
h1 doesn't have connectivity to 192.168.1.65
h1 doesn't have connectivity to 192.168.1.66
h1 doesn't have connectivity to 192.168.3.1
h1 doesn't have connectivity to 192.168.3.2
WRONG ANSWER
mininet> h1 wireshark &
mininet> h1 dhclient h1-eth0
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
mininet> h1 ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.5  netmask 255.255.255.192  broadcast 192.168.1.63
        inet6 fe80::c86d:32ff:fef4:a68  prefixlen 64  scopeid 0x20<link>
        ether ca:6d:32:f4:0a:68  txqueuelen 1000  (Ethernet)
        RX packets 58  bytes 7105 (7.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 24  bytes 2660 (2.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

發現結果皆無錯誤。

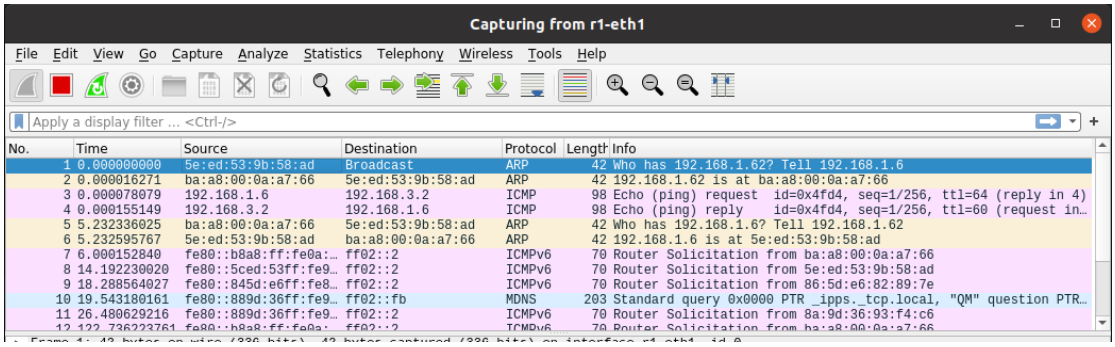
4. 不行，因為只有 h1 和 DHCP server 在同一個 LAN。要是不在同一個 LAN，DHCP 相關封包將無法傳到。

5. h1 ping h5 :

r1-eth0 :



r1-eth1 :

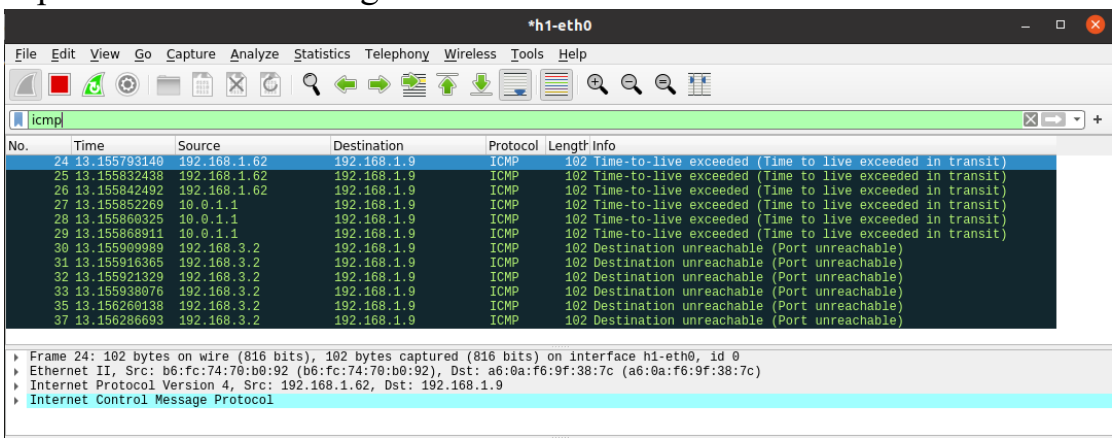


h1 ping 了 h5 之後，h1 會發出一個 ping request，r1-eth1 收到(下圖 No.3)，再從 r1-eth0 送出(上圖 No.1)，最後會送到 h5，h5 收到後會回應一個 ping reply，並會由 r1-eth0 收到(上圖 No.2)，r1-eth1 送回給 h1(下圖 No.4)。

6. traceroute :

```
mininet> h1 dhclient h1-eth0
mininet> h1 traceroute h5
traceroute to 192.168.3.2 (192.168.3.2), 30 hops max, 60 byte packets
 1  192.168.1.62 (192.168.1.62)  0.684 ms  0.677 ms  0.685 ms
 2  10.0.1.1 (10.0.1.1)  0.692 ms  0.697 ms  0.704 ms
 3  * * *
 4  * * *
 5  192.168.3.2 (192.168.3.2)  0.721 ms  0.723 ms  0.725 ms
mininet>
```

capture all ICMP messages :



traceroute 是利用 router 和 host 不會接受 TTL 為零的封包的機制。封包每經過一個 router，TTL 就會減一，而 router 收到 TTL 為零的封包就會將其丟棄，並回傳一個 Time-to-live exceeded 的錯誤訊息給原 host，而那個錯誤訊息就包含了 router 的 IP。而 host 收到錯誤訊息後只要將原 TTL 再加一送出去，就能達到更遠的 router。traceroute 就是這樣慢慢找到所有 router 的 IP，如上圖中的 No.24 到 No.29。而到達 Destination 的時候，封包會被接收，不會回傳 TTL。因此 traceroute 故意用大於 30000 的 port，而 UDP 規定 port 必須小於 30000，因此就會回傳 Destination Port unreachable(上圖 No.30-37)，其中也包含了 Destination 的 IP，至此 traceroute 結束。

對於題目中只能解析 r1、r2、h5 的 traceroute 訊息，我先將完整的路

徑 trace 出來:

```
mininet> h1 dhclient h1-eth0
mininet> h1 traceroute h5
traceroute to 192.168.3.2 (192.168.3.2), 30 hops max, 60 byte packets
 1  192.168.1.62 (192.168.1.62)  0.608 ms  0.596 ms  0.599 ms
 2  10.0.1.1 (10.0.1.1)  0.604 ms  0.608 ms  0.611 ms
 3  10.0.0.2 (10.0.0.2)  0.611 ms  0.608 ms  0.606 ms
 4  10.0.2.3 (10.0.2.3)  0.605 ms  0.604 ms  0.600 ms
 5  192.168.3.2 (192.168.3.2)  0.632 ms  0.633 ms  0.634 ms
```

接下來註解掉下圖中第一條 routing rule:

```
#r1-r2
#routers['r1'].cmd('route add -net 10.0.0.0/24 gw 10.0.1.1')
#r1-right
routers['r1'].cmd('route add -net 192.168.3.0/24 gw 10.0.1.1')
#r1-(r3-r4) bonus
routers['r1'].cmd('route add -net 10.0.2.0/24 gw 10.0.1.1')
```

得到下圖結果:

```
mininet> h1 traceroute h5
traceroute to 192.168.3.2 (192.168.3.2), 30 hops max, 60 byte packets
 1  192.168.1.62 (192.168.1.62)  0.642 ms  0.580 ms  0.581 ms
 2  10.0.1.1 (10.0.1.1)  0.586 ms  0.589 ms  0.591 ms
 3  * * *
 4  10.0.2.3 (10.0.2.3)  0.561 ms  0.559 ms  0.558 ms
 5  192.168.3.2 (192.168.3.2)  0.584 ms  0.585 ms  0.585 ms
```

第三條路徑成了星號。接著註解掉下圖中第三條 routing rule:

```
#r1-r2
routers['r1'].cmd('route add -net 10.0.0.0/24 gw 10.0.1.1')
#r1-right
routers['r1'].cmd('route add -net 192.168.3.0/24 gw 10.0.1.1')
#r1-(r3-r4) bonus
#routers['r1'].cmd('route add -net 10.0.2.0/24 gw 10.0.1.1')
```

得到下圖結果:

```
mininet> h1 traceroute h5
traceroute to 192.168.3.2 (192.168.3.2), 30 hops max, 60 byte packets
 1  192.168.1.62 (192.168.1.62)  0.653 ms  0.643 ms  0.646 ms
 2  10.0.1.1 (10.0.1.1)  0.651 ms  0.654 ms  0.659 ms
 3  10.0.0.2 (10.0.0.2)  0.658 ms  0.655 ms  0.653 ms
 4  * * *
 5  192.168.3.2 (192.168.3.2)  0.674 ms  0.675 ms  0.675 ms
```

這次是第四條路徑變成星號。只要將這兩條路徑都註解掉，就會得到三和四都是星號的結果:

```
mininet> h1 dhclient h1-eth0
mininet> h1 traceroute h5
traceroute to 192.168.3.2 (192.168.3.2), 30 hops max, 60 byte packets
 1  192.168.1.62 (192.168.1.62)  0.684 ms  0.677 ms  0.685 ms
 2  10.0.1.1 (10.0.1.1)  0.692 ms  0.697 ms  0.704 ms
 3  * * *
 4  * * *
 5  192.168.3.2 (192.168.3.2)  0.721 ms  0.723 ms  0.725 ms
mininet>
```

再對照一下拓樸環境圖，我們可以發現只要 r1 的 routing table 沒有 r3 的資訊，就無法將 r3 回傳的 TTL 訊息送回 h1，因此得到第三條路徑的星號，r4 也是同理，因此第四條路徑也會是星號。而只要將所有路徑都完整的寫出來(共 12 條 routing rules)，就能得到 bonus 的結果:

```
mininet> h1 dhclient h1-eth0
mininet> h1 traceroute h5
traceroute to 192.168.3.2 (192.168.3.2), 30 hops max, 60 byte packets
 1  192.168.1.62 (192.168.1.62)  0.608 ms  0.596 ms  0.599 ms
 2  10.0.1.1 (10.0.1.1)  0.604 ms  0.608 ms  0.611 ms
 3  10.0.0.2 (10.0.0.2)  0.611 ms  0.608 ms  0.606 ms
 4  10.0.2.3 (10.0.2.3)  0.605 ms  0.604 ms  0.600 ms
 5  192.168.3.2 (192.168.3.2)  0.632 ms  0.633 ms  0.634 ms
```

7. 1st hop:
   type: Time-to-live exceeded (type 11)
   sender: 192.168.1.62 (r1-eth1)
   2nd hop:
   type: Time-to-live exceeded (type 11)
   sender: 10.0.1.1 (r2-eth1)

8. 5th hop:
   type: Destination unreachable (type 3)
   sender: 192.168.3.2 (h5)