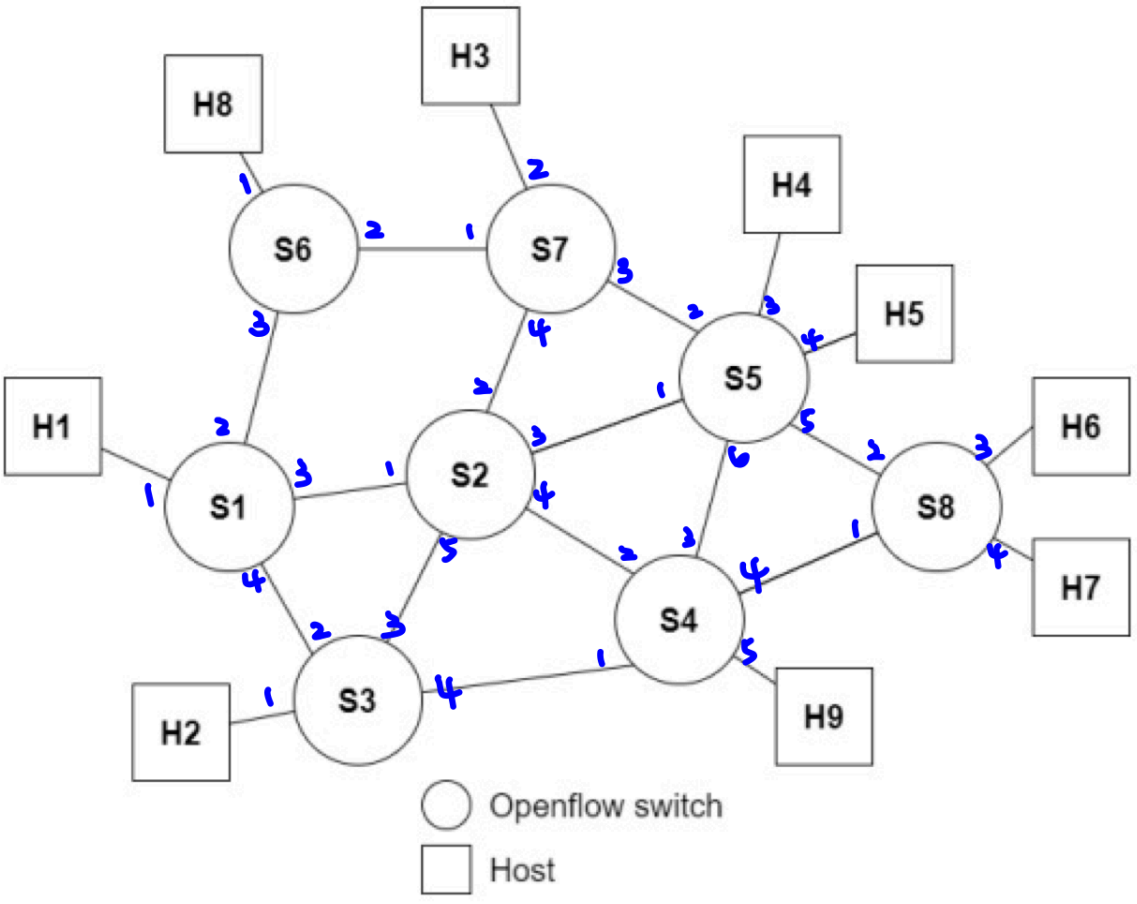
NFV-dual-path routing

概念陳述:

1. 讀取 source host 和 target host(限制只能輸入 H1, H2 到 H9)
2. 建立網路圖, 並用 networkx 工具找出 source 到 target 的兩條最短路徑。
    - 只有一條路徑的情況(兩條最短路徑相同的情況)
    - 第一條最短路徑必包含在第二條最短路徑中時, 若第一條路徑不通, 則第二條一定也不通, 所以沒必要找更長的路徑;也可以說兩條最短路徑是相同的, 因為找不到更少相同 switch 的路徑
    - 兩個 host 接到相同 switch
    - 兩個host 各自接的 switch 直接連接
    - 簡單來說就是路徑上的 switch 小於等於 2 個時, 路徑只會有一條
3. 根據路徑建立拓樸(為避免 network 啟動時, 因為 link 有迴圈導致封包不斷被傳誦, 所以先建立第一條路徑)
4. 新增 openflow 規則, 使 source host 連到 target host(單向, 避免建立第二條路徑形成迴圈)
5. 建立第二條路徑(現在 openflow 規則已建立, 可以控制封包流向)
6. 建立第二條路徑的 openflow規則(單向)
7. 執行, 完成需求

網路圖:



成果展示：

兩個 **host** 接在相同 **switch** 的情況（**H4** 和 **H5** 都接到 **s5**）：

連通情況：

- H4 ping H5 是通的；H5 ping H4 不通

```
Please input the hosts as H1, H2, ..., H9.
Input source host: H4
Input target host: H5
source: H4  target: H5
path: ['S5']
mininet> H4 ping H5
PING 10.0.0.5 (10.0.0.5) 56(84) bytes of data.
64 bytes from 10.0.0.5: icmp_seq=1 ttl=64 time=1.60 ms
64 bytes from 10.0.0.5: icmp_seq=2 ttl=64 time=1.13 ms
64 bytes from 10.0.0.5: icmp_seq=3 ttl=64 time=0.073 ms
^C
--- 10.0.0.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2006ms
rtt min/avg/max/mdev = 0.073/0.936/1.604/0.640 ms
mininet> H5 ping H4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
^C
--- 10.0.0.4 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2081ms
```

網路連接情況：

- s5 port 3 接到 H4；s5 port 4 接到 H5

```
mininet> net.links
H1 H1-eth0:s1-eth1
H2 H2-eth0:s3-eth1
H3 H3-eth0:s7-eth2
H4 H4-eth0:s5-eth3
H5 H5-eth0:s5-eth4
H6 H6-eth0:s8-eth3
H7 H7-eth0:s8-eth4
H8 H8-eth0:s6-eth1
H9 H9-eth0:s4-eth5
s1 lo:  s1-eth1:H1-eth0
s2 lo:
s3 lo:  s3-eth1:H2-eth0
s4 lo:  s4-eth5:H9-eth0
s5 lo:  s5-eth3:H4-eth0 s5-eth4:H5-eth0
s6 lo:  s6-eth1:H8-eth0
s7 lo:  s7-eth2:H3-eth0
s8 lo:  s8-eth3:H6-eth0 s8-eth4:H7-eth0
c0
```

openflow規則設定：

- s5 規則

```
(venv_37) shawn@shawn-VirtualBox:~/Desktop/Lab2$ sudo ovs-ofctl dump-flows s5
[sudo] password for shawn:
 cookie=0x0, duration=1347.046s, table=0, n_packets=0, n_bytes=0, priority=200,ct_state=-trk,ip,nw_src=10.0.0.4,nw_dst=10.0.0.5 actions=ct(table=0)
 cookie=0x0, duration=1346.859s, table=0, n_packets=0, n_bytes=0, priority=200,ct_state=-trk,ip,nw_src=10.0.0.5,nw_dst=10.0.0.4 actions=ct(table=0)
 cookie=0x0, duration=1346.993s, table=0, n_packets=0, n_bytes=0, priority=200,ct_state=+new+trk,ip,nw_src=10.0.0.4,nw_dst=10.0.0.5 actions=ct(commit),output:"s5-eth4"
 cookie=0x0, duration=1346.937s, table=0, n_packets=0, n_bytes=0, priority=200,ct_state=+est+trk,ip,nw_src=10.0.0.4,nw_dst=10.0.0.5 actions=output:"s5-eth4"
 cookie=0x0, duration=1346.793s, table=0, n_packets=0, n_bytes=0, priority=200,ct_state=+est+trk,ip,nw_src=10.0.0.5,nw_dst=10.0.0.4 actions=output:"s5-eth3"
 cookie=0x0, duration=1346.488s, table=0, n_packets=0, n_bytes=0, priority=200,arp actions=FLOOD
```

路徑上的 switch 只有兩個的情況（H1 接到 s1 、H2 接到 s3，而 s1 和 s3 直接相連 ）：

連通情況：

- H1 ping H2 是通的；H2 ping H1 不通

```
Please input the hosts as H1, H2, ..., H9.
Input source host: H1
Input target host: H2
source: H1  target: H2
path: ['S1', 'S3']
mininet> H1 ping H2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=2.31 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.634 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.080 ms
^C
--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.080/1.009/2.314/0.949 ms
mininet> H2 ping H1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
^C
--- 10.0.0.1 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2082ms
```

網路連接情況：

- s1 port 1 接到 H1；s1 port 4 接到 s3
- s3 port 1 接到 H2；s3 port 2 接到 s1

```
mininet> net.links
H1 H1-eth0:s1-eth1
H2 H2-eth0:s3-eth1
H3 H3-eth0:s7-eth2
H4 H4-eth0:s5-eth3
H5 H5-eth0:s5-eth4
H6 H6-eth0:s8-eth3
H7 H7-eth0:s8-eth4
H8 H8-eth0:s6-eth1
H9 H9-eth0:s4-eth5
s1 lo:  s1-eth1:H1-eth0 s1-eth4:s3-eth2
s2 lo:
s3 lo:  s3-eth1:H2-eth0 s3-eth2:s1-eth4
s4 lo:  s4-eth5:H9-eth0
s5 lo:  s5-eth3:H4-eth0 s5-eth4:H5-eth0
s6 lo:  s6-eth1:H8-eth0
s7 lo:  s7-eth2:H3-eth0
s8 lo:  s8-eth3:H6-eth0 s8-eth4:H7-eth0
c0
```

openflow規則設定：

- s1 規則

```
(venv_37) shawn@shawn-VirtualBox:~/Desktop/Lab2$ sudo ovs-ofctl dump-flows s1
[sudo] password for shawn:
 cookie=0x0, duration=301.354s, table=0, n_packets=3, n_bytes=294, priority=200,
ct_state=-trk,ip,nw_src=10.0.0.1,nw_dst=10.0.0.2 actions=ct(table=0)
 cookie=0x0, duration=301.066s, table=0, n_packets=3, n_bytes=294, priority=200,
ct_state=-trk,ip,nw_src=10.0.0.2,nw_dst=10.0.0.1 actions=ct(table=0)
 cookie=0x0, duration=301.305s, table=0, n_packets=1, n_bytes=98, priority=200,c
t_state=+new+trk,ip,nw_src=10.0.0.1,nw_dst=10.0.0.2 actions=ct(commit),output:"s
1-eth4"
 cookie=0x0, duration=301.262s, table=0, n_packets=2, n_bytes=196, priority=200,
ct_state=+est+trk,ip,nw_src=10.0.0.1,nw_dst=10.0.0.2 actions=output:"s1-eth4"
 cookie=0x0, duration=301.006s, table=0, n_packets=3, n_bytes=294, priority=200,
ct_state=+est+trk,ip,nw_src=10.0.0.2,nw_dst=10.0.0.1 actions=output:"s1-eth1"
 cookie=0x0, duration=300.872s, table=0, n_packets=4, n_bytes=168, priority=200,
arp actions=FLOOD
```

- s3 規則

```
(venv_37) shawn@shawn-VirtualBox:~/Desktop/Lab2$ sudo ovs-ofctl dump-flows s3
 cookie=0x0, duration=332.511s, table=0, n_packets=3, n_bytes=294, priority=200,
ct_state=-trk,ip,nw_src=10.0.0.1,nw_dst=10.0.0.2 actions=ct(table=0)
 cookie=0x0, duration=332.249s, table=0, n_packets=6, n_bytes=588, priority=200,
ct_state=-trk,ip,nw_src=10.0.0.2,nw_dst=10.0.0.1 actions=ct(table=0)
 cookie=0x0, duration=332.461s, table=0, n_packets=1, n_bytes=98, priority=200,c
t_state=+new+trk,ip,nw_src=10.0.0.1,nw_dst=10.0.0.2 actions=ct(commit),output:"s
3-eth1"
 cookie=0x0, duration=332.414s, table=0, n_packets=2, n_bytes=196, priority=200,
ct_state=+est+trk,ip,nw_src=10.0.0.1,nw_dst=10.0.0.2 actions=output:"s3-eth1"
 cookie=0x0, duration=332.205s, table=0, n_packets=3, n_bytes=294, priority=200,
ct_state=+est+trk,ip,nw_src=10.0.0.2,nw_dst=10.0.0.1 actions=output:"s3-eth2"
 cookie=0x0, duration=332.065s, table=0, n_packets=4, n_bytes=168, priority=200,
arp actions=FLOOD
```

路徑上的 **switch** 有 **3** 個的情況（**H2 -> H4**）：

連通情況：

- H2 ping H4 是通的；H4 ping H2 不通

```
Please input the hosts as H1, H2, ..., H9.
Input source host: H2
Input target host: H4
source: H2  target: H4
path1: ['S3', 'S2', 'S5']
path2: ['S3', 'S4', 'S5']
mininet> H2 ping H4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=4.10 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=1.72 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=0.093 ms
^C
--- 10.0.0.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 0.093/1.972/4.102/1.646 ms
mininet> H4 ping H2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
^C
--- 10.0.0.2 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2031ms
```

網路連接情況：

- (input port) switch name (output port)
- path1: H2 -> (1)S3(3) -> (5)S2(3) - > (1)S5(3) - > H4
- path2: H2 -> (1)S3(4) -> (1)S4(3) - > (6)S5(3) -> H4

```
mininet> net.links
H1 H1-eth0:s1-eth1
H2 H2-eth0:s3-eth1
H3 H3-eth0:s7-eth2
H4 H4-eth0:s5-eth3
H5 H5-eth0:s5-eth4
H6 H6-eth0:s8-eth3
H7 H7-eth0:s8-eth4
H8 H8-eth0:s6-eth1
H9 H9-eth0:s4-eth5
s1 lo:  s1-eth1:H1-eth0
s2 lo:  s2-eth3:s5-eth1 s2-eth5:s3-eth3
s3 lo:  s3-eth1:H2-eth0 s3-eth3:s2-eth5 s3-eth4:s4-eth1
s4 lo:  s4-eth1:s3-eth4 s4-eth3:s5-eth6 s4-eth5:H9-eth0
s5 lo:  s5-eth1:s2-eth3 s5-eth3:H4-eth0 s5-eth4:H5-eth0 s5-eth6:s4-eth3
s6 lo:  s6-eth1:H8-eth0
s7 lo:  s7-eth2:H3-eth0
s8 lo:  s8-eth3:H6-eth0 s8-eth4:H7-eth0
c0
```

openflow規則設定:

- s2 規則

```
(venv_37) shawn@shawn-VirtualBox:~/Desktop/Lab2$ sudo ovs-ofctl dump-flows s2
[sudo] password for shawn:
 cookie=0x0, duration=778.989s, table=0, n_packets=3, n_bytes=294, priority=200,
ct_state=-trk,ip,nw_src=10.0.0.2,nw_dst=10.0.0.4 actions=ct(table=0)
 cookie=0x0, duration=778.591s, table=0, n_packets=3, n_bytes=294, priority=200,
ct_state=-trk,ip,nw_src=10.0.0.4,nw_dst=10.0.0.2 actions=ct(table=0)
 cookie=0x0, duration=778.942s, table=0, n_packets=1, n_bytes=98, priority=200,c
t_state=+new+trk,ip,nw_src=10.0.0.2,nw_dst=10.0.0.4 actions=ct(commit),output:"s
2-eth3"
 cookie=0x0, duration=778.893s, table=0, n_packets=2, n_bytes=196, priority=200,
ct_state=+est+trk,ip,nw_src=10.0.0.2,nw_dst=10.0.0.4 actions=output:"s2-eth3"
 cookie=0x0, duration=778.548s, table=0, n_packets=3, n_bytes=294, priority=200,
ct_state=+est+trk,ip,nw_src=10.0.0.4,nw_dst=10.0.0.2 actions=output:"s2-eth5"
 cookie=0x0, duration=777.399s, table=0, n_packets=4, n_bytes=168, priority=200,
arp actions=FLOOD
```

- s3 規則

```
(venv_37) shawn@shawn-VirtualBox:~/Desktop/Lab2$ sudo ovs-ofctl dump-flows s3
 cookie=0x0, duration=805.872s, table=0, n_packets=3, n_bytes=294, priority=200,
ct_state=-trk,ip,nw_src=10.0.0.2,nw_dst=10.0.0.4 actions=ct(table=0)
 cookie=0x0, duration=805.431s, table=0, n_packets=3, n_bytes=294, priority=200,
ct_state=-trk,ip,nw_src=10.0.0.4,nw_dst=10.0.0.2 actions=ct(table=0)
 cookie=0x0, duration=804.937s, table=0, n_packets=0, n_bytes=0, priority=100,ct
_state=-trk,ip,nw_src=10.0.0.2,nw_dst=10.0.0.4 actions=ct(table=1)
 cookie=0x0, duration=804.475s, table=0, n_packets=0, n_bytes=0, priority=100,ct
_state=-trk,ip,nw_src=10.0.0.4,nw_dst=10.0.0.2 actions=ct(table=1)
 cookie=0x0, duration=805.823s, table=0, n_packets=1, n_bytes=98, priority=200,c
t_state=+new+trk,ip,nw_src=10.0.0.2,nw_dst=10.0.0.4 actions=ct(commit),output:"s
3-eth3"
 cookie=0x0, duration=805.778s, table=0, n_packets=2, n_bytes=196, priority=200,
ct_state=+est+trk,ip,nw_src=10.0.0.2,nw_dst=10.0.0.4 actions=output:"s3-eth3"
 cookie=0x0, duration=805.379s, table=0, n_packets=3, n_bytes=294, priority=200,
ct_state=+est+trk,ip,nw_src=10.0.0.4,nw_dst=10.0.0.2 actions=output:"s3-eth1"
 cookie=0x0, duration=804.093s, table=0, n_packets=4, n_bytes=168, priority=200,
arp actions=FLOOD
 cookie=0x0, duration=804.889s, table=1, n_packets=0, n_bytes=0, priority=100,ct
_state=+new+trk,ip,nw_src=10.0.0.2,nw_dst=10.0.0.4 actions=ct(commit),output:4
 cookie=0x0, duration=804.831s, table=1, n_packets=0, n_bytes=0, priority=100,ct
_state=+est+trk,ip,nw_src=10.0.0.2,nw_dst=10.0.0.4 actions=output:4
 cookie=0x0, duration=804.433s, table=1, n_packets=0, n_bytes=0, priority=100,ct
_state=+est+trk,ip,nw_src=10.0.0.4,nw_dst=10.0.0.2 actions=output:"s3-eth1"
```

- s4 規則

```
(venv_37) shawn@shawn-VirtualBox:~/Desktop/Lab2$ sudo ovs-ofctl dump-flows s4
 cookie=0x0, duration=829.484s, table=0, n_packets=0, n_bytes=0, priority=200,ar
p actions=FLOOD
 cookie=0x0, duration=830.219s, table=0, n_packets=0, n_bytes=0, priority=100,ct
_state=-trk,ip,nw_src=10.0.0.2,nw_dst=10.0.0.4 actions=ct(table=1)
 cookie=0x0, duration=829.818s, table=0, n_packets=0, n_bytes=0, priority=100,ct
_state=-trk,ip,nw_src=10.0.0.4,nw_dst=10.0.0.2 actions=ct(table=1)
 cookie=0x0, duration=830.167s, table=1, n_packets=0, n_bytes=0, priority=100,ct
_state=+new+trk,ip,nw_src=10.0.0.2,nw_dst=10.0.0.4 actions=ct(commit),output:3
 cookie=0x0, duration=830.119s, table=1, n_packets=0, n_bytes=0, priority=100,ct
_state=+est+trk,ip,nw_src=10.0.0.2,nw_dst=10.0.0.4 actions=output:3
 cookie=0x0, duration=829.754s, table=1, n_packets=0, n_bytes=0, priority=100,ct
_state=+est+trk,ip,nw_src=10.0.0.4,nw_dst=10.0.0.2 actions=output:1
```

- s5 規則

```
(venv_37) shawn@shawn-VirtualBox:~/Desktop/Lab2$ sudo ovs-ofctl dump-flows s5
 cookie=0x0, duration=859.438s, table=0, n_packets=3, n_bytes=294, priority=200,
ct_state=-trk,ip,nw_src=10.0.0.2,nw_dst=10.0.0.4 actions=ct(table=0)
 cookie=0x0, duration=859.090s, table=0, n_packets=6, n_bytes=588, priority=200,
ct_state=-trk,ip,nw_src=10.0.0.4,nw_dst=10.0.0.2 actions=ct(table=0)
 cookie=0x0, duration=858.478s, table=0, n_packets=0, n_bytes=0, priority=100,ct
_state=-trk,ip,nw_src=10.0.0.2,nw_dst=10.0.0.4 actions=ct(table=1)
 cookie=0x0, duration=858.119s, table=0, n_packets=0, n_bytes=0, priority=100,ct
_state=-trk,ip,nw_src=10.0.0.4,nw_dst=10.0.0.2 actions=ct(table=1)
 cookie=0x0, duration=859.384s, table=0, n_packets=1, n_bytes=98, priority=200,c
t_state=+new+trk,ip,nw_src=10.0.0.2,nw_dst=10.0.0.4 actions=ct(commit),output:"s
5-eth3"
 cookie=0x0, duration=859.331s, table=0, n_packets=2, n_bytes=196, priority=200,
ct_state=+est+trk,ip,nw_src=10.0.0.2,nw_dst=10.0.0.4 actions=output:"s5-eth3"
 cookie=0x0, duration=859.046s, table=0, n_packets=3, n_bytes=294, priority=200,
ct_state=+est+trk,ip,nw_src=10.0.0.4,nw_dst=10.0.0.2 actions=output:"s5-eth1"
 cookie=0x0, duration=857.845s, table=0, n_packets=4, n_bytes=168, priority=200,
arp actions=FLOOD
 cookie=0x0, duration=858.430s, table=1, n_packets=0, n_bytes=0, priority=100,ct
_state=+new+trk,ip,nw_src=10.0.0.2,nw_dst=10.0.0.4 actions=ct(commit),output:"s5
-eth3"
 cookie=0x0, duration=858.377s, table=1, n_packets=0, n_bytes=0, priority=100,ct
_state=+est+trk,ip,nw_src=10.0.0.2,nw_dst=10.0.0.4 actions=output:"s5-eth3"
 cookie=0x0, duration=858.076s, table=1, n_packets=0, n_bytes=0, priority=100,ct
_state=+est+trk,ip,nw_src=10.0.0.4,nw_dst=10.0.0.2 actions=output:6
```

路徑上的 switch 有 4、5 個的情況（H8 -> H7）：

連通情況：

- H8 ping H7 是通的;H7 ping H8 不通

```
(venv_37) shawn@shawn-VirtualBox:~/Desktop/Lab2$ sudo /home/shawn/Desktop/venv/v
env_37/bin/python3 Lab2.py
[sudo] password for shawn:
Sorry, try again.
[sudo] password for shawn:
Please input the hosts as H1, H2, ..., H9.
Input source host: H8
Input target host: H7
source: H8  target: H7
path1: ['S6', 'S7', 'S5', 'S8']
path2: ['S6', 'S1', 'S2', 'S4', 'S8']
mininet> H8 ping H7
PING 10.0.0.7 (10.0.0.7) 56(84) bytes of data.
64 bytes from 10.0.0.7: icmp_seq=1 ttl=64 time=1.39 ms
64 bytes from 10.0.0.7: icmp_seq=2 ttl=64 time=0.467 ms
64 bytes from 10.0.0.7: icmp_seq=3 ttl=64 time=0.110 ms
^C
--- 10.0.0.7 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.110/0.657/1.394/0.541 ms
mininet> H7 ping H8
PING 10.0.0.8 (10.0.0.8) 56(84) bytes of data.
^C
--- 10.0.0.8 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1030ms
```

網路連接情況:

- path1: H8 -> (1)S6(2) -> (1)S7(3) - > (2)S5(5) - > (2)S8(4) - > H7
- path2: H8 -> (1)S6(3) -> (2)S1(3) - > (1)S2(4) - > (2)S4(4) - > (1)S8(4) -> H7

```
mininet> net.links
H1 H1-eth0:s1-eth1
H2 H2-eth0:s3-eth1
H3 H3-eth0:s7-eth2
H4 H4-eth0:s5-eth3
H5 H5-eth0:s5-eth4
H6 H6-eth0:s8-eth3
H7 H7-eth0:s8-eth4
H8 H8-eth0:s6-eth1
H9 H9-eth0:s4-eth5
s1 lo:  s1-eth1:H1-eth0 s1-eth2:s6-eth3 s1-eth3:s2-eth1
s2 lo:  s2-eth1:s1-eth3 s2-eth4:s4-eth2
s3 lo:  s3-eth1:H2-eth0
s4 lo:  s4-eth2:s2-eth4 s4-eth4:s8-eth1 s4-eth5:H9-eth0
s5 lo:  s5-eth2:s7-eth3 s5-eth3:H4-eth0 s5-eth4:H5-eth0 s5-eth5:s8-eth2
s6 lo:  s6-eth1:H8-eth0 s6-eth2:s7-eth1 s6-eth3:s1-eth2
s7 lo:  s7-eth1:s6-eth2 s7-eth2:H3-eth0 s7-eth3:s5-eth2
s8 lo:  s8-eth1:s4-eth4 s8-eth2:s5-eth5 s8-eth3:H6-eth0 s8-eth4:H7-eth0
c0
```

openflow規則設定:

- s1 規則

```
(venv_37) shawn@shawn-VirtualBox:~/Desktop/Lab2$ sudo ovs-ofctl dump-flows s1
 cookie=0x0, duration=557.717s, table=0, n_packets=0, n_bytes=0, priority=200,ar
p actions=FLOOD
 cookie=0x0, duration=558.829s, table=0, n_packets=0, n_bytes=0, priority=100,ct
_state=-trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=ct(table=1)
 cookie=0x0, duration=558.135s, table=0, n_packets=0, n_bytes=0, priority=100,ct
_state=-trk,ip,nw_src=10.0.0.7,nw_dst=10.0.0.8 actions=ct(table=1)
 cookie=0x0, duration=558.775s, table=1, n_packets=0, n_bytes=0, priority=100,ct
_state=+new+trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=ct(commit),output:3
 cookie=0x0, duration=558.729s, table=1, n_packets=0, n_bytes=0, priority=100,ct
_state=+est+trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=output:3
 cookie=0x0, duration=558.088s, table=1, n_packets=0, n_bytes=0, priority=100,ct
_state=+est+trk,ip,nw_src=10.0.0.7,nw_dst=10.0.0.8 actions=output:2
```

- s2 規則

```
(venv_37) shawn@shawn-VirtualBox:~/Desktop/Lab2$ sudo ovs-ofctl dump-flows s2
 cookie=0x0, duration=587.214s, table=0, n_packets=0, n_bytes=0, priority=200,ar
p actions=FLOOD
 cookie=0x0, duration=588.224s, table=0, n_packets=0, n_bytes=0, priority=100,ct
_state=-trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=ct(table=1)
 cookie=0x0, duration=587.583s, table=0, n_packets=0, n_bytes=0, priority=100,ct
_state=-trk,ip,nw_src=10.0.0.7,nw_dst=10.0.0.8 actions=ct(table=1)
 cookie=0x0, duration=588.175s, table=1, n_packets=0, n_bytes=0, priority=100,ct
_state=+new+trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=ct(commit),output:4
 cookie=0x0, duration=588.134s, table=1, n_packets=0, n_bytes=0, priority=100,ct
_state=+est+trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=output:4
 cookie=0x0, duration=587.530s, table=1, n_packets=0, n_bytes=0, priority=100,ct
_state=+est+trk,ip,nw_src=10.0.0.7,nw_dst=10.0.0.8 actions=output:1
```

- s4 規則

```
(venv_37) shawn@shawn-VirtualBox:~/Desktop/Lab2$ sudo ovs-ofctl dump-flows s4
 cookie=0x0, duration=617.337s, table=0, n_packets=0, n_bytes=0, priority=200,ar
p actions=FLOOD
 cookie=0x0, duration=618.311s, table=0, n_packets=0, n_bytes=0, priority=100,ct
_state=-trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=ct(table=1)
 cookie=0x0, duration=617.687s, table=0, n_packets=0, n_bytes=0, priority=100,ct
_state=-trk,ip,nw_src=10.0.0.7,nw_dst=10.0.0.8 actions=ct(table=1)
 cookie=0x0, duration=618.262s, table=1, n_packets=0, n_bytes=0, priority=100,ct
_state=+new+trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=ct(commit),output:4
 cookie=0x0, duration=618.213s, table=1, n_packets=0, n_bytes=0, priority=100,ct
_state=+est+trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=output:4
 cookie=0x0, duration=617.645s, table=1, n_packets=0, n_bytes=0, priority=100,ct
_state=+est+trk,ip,nw_src=10.0.0.7,nw_dst=10.0.0.8 actions=output:2
```

- s5 規則

```
(venv_37) shawn@shawn-VirtualBox:~/Desktop/Lab2$ sudo ovs-ofctl dump-flows s5
 cookie=0x0, duration=637.417s, table=0, n_packets=3, n_bytes=294, priority=200,
ct_state=-trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=ct(table=0)
 cookie=0x0, duration=636.902s, table=0, n_packets=3, n_bytes=294, priority=200,
ct_state=-trk,ip,nw_src=10.0.0.7,nw_dst=10.0.0.8 actions=ct(table=0)
 cookie=0x0, duration=637.365s, table=0, n_packets=1, n_bytes=98, priority=200,c
t_state=+new+trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=ct(commit),output:"s
5-eth5"
 cookie=0x0, duration=637.313s, table=0, n_packets=2, n_bytes=196, priority=200,
ct_state=+est+trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=output:"s5-eth5"
 cookie=0x0, duration=636.858s, table=0, n_packets=3, n_bytes=294, priority=200,
ct_state=+est+trk,ip,nw_src=10.0.0.7,nw_dst=10.0.0.8 actions=output:"s5-eth2"
 cookie=0x0, duration=634.785s, table=0, n_packets=4, n_bytes=168, priority=200,
arp actions=FLOOD
```

- s6 規則

```
(venv_37) shawn@shawn-VirtualBox:~/Desktop/Lab2$ sudo ovs-ofctl dump-flows s6
 cookie=0x0, duration=700.821s, table=0, n_packets=3, n_bytes=294, priority=200,
ct_state=-trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=ct(table=0)
 cookie=0x0, duration=700.176s, table=0, n_packets=3, n_bytes=294, priority=200,
ct_state=-trk,ip,nw_src=10.0.0.7,nw_dst=10.0.0.8 actions=ct(table=0)
 cookie=0x0, duration=699.331s, table=0, n_packets=0, n_bytes=0, priority=100,ct
_state=-trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=ct(table=1)
 cookie=0x0, duration=698.583s, table=0, n_packets=0, n_bytes=0, priority=100,ct
_state=-trk,ip,nw_src=10.0.0.7,nw_dst=10.0.0.8 actions=ct(table=1)
 cookie=0x0, duration=700.764s, table=0, n_packets=1, n_bytes=98, priority=200,c
t_state=+new+trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=ct(commit),output:"s
6-eth2"
 cookie=0x0, duration=700.697s, table=0, n_packets=2, n_bytes=196, priority=200,
ct_state=+est+trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=output:"s6-eth2"
 cookie=0x0, duration=700.125s, table=0, n_packets=3, n_bytes=294, priority=200,
ct_state=+est+trk,ip,nw_src=10.0.0.7,nw_dst=10.0.0.8 actions=output:"s6-eth1"
 cookie=0x0, duration=697.807s, table=0, n_packets=4, n_bytes=168, priority=200,
arp actions=FLOOD
 cookie=0x0, duration=699.283s, table=1, n_packets=0, n_bytes=0, priority=100,ct
_state=+new+trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=ct(commit),output:3
 cookie=0x0, duration=699.234s, table=1, n_packets=0, n_bytes=0, priority=100,ct
_state=+est+trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=output:3
 cookie=0x0, duration=698.536s, table=1, n_packets=0, n_bytes=0, priority=100,ct
_state=+est+trk,ip,nw_src=10.0.0.7,nw_dst=10.0.0.8 actions=output:"s6-eth1"
```

- s7 規則

```
(venv_37) shawn@shawn-VirtualBox:~/Desktop/Lab2$ sudo ovs-ofctl dump-flows s7
 cookie=0x0, duration=717.172s, table=0, n_packets=3, n_bytes=294, priority=200,
ct_state=-trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=ct(table=0)
 cookie=0x0, duration=716.594s, table=0, n_packets=3, n_bytes=294, priority=200,
ct_state=-trk,ip,nw_src=10.0.0.7,nw_dst=10.0.0.8 actions=ct(table=0)
 cookie=0x0, duration=717.118s, table=0, n_packets=1, n_bytes=98, priority=200,c
t_state=+new+trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=ct(commit),output:"s
7-eth3"
 cookie=0x0, duration=717.067s, table=0, n_packets=2, n_bytes=196, priority=200,
ct_state=+est+trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=output:"s7-eth3"
 cookie=0x0, duration=716.543s, table=0, n_packets=3, n_bytes=294, priority=200,
ct_state=+est+trk,ip,nw_src=10.0.0.7,nw_dst=10.0.0.8 actions=output:"s7-eth1"
 cookie=0x0, duration=714.276s, table=0, n_packets=4, n_bytes=168, priority=200,
arp actions=FLOOD
```

- s8 規則

```
(venv_37) shawn@shawn-VirtualBox:~/Desktop/Lab2$ sudo ovs-ofctl dump-flows s8
 cookie=0x0, duration=741.580s, table=0, n_packets=3, n_bytes=294, priority=200,
ct_state=-trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=ct(table=0)
 cookie=0x0, duration=741.130s, table=0, n_packets=5, n_bytes=490, priority=200,
ct_state=-trk,ip,nw_src=10.0.0.7,nw_dst=10.0.0.8 actions=ct(table=0)
 cookie=0x0, duration=739.993s, table=0, n_packets=0, n_bytes=0, priority=100,ct
_state=-trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=ct(table=1)
 cookie=0x0, duration=739.424s, table=0, n_packets=0, n_bytes=0, priority=100,ct
_state=-trk,ip,nw_src=10.0.0.7,nw_dst=10.0.0.8 actions=ct(table=1)
 cookie=0x0, duration=741.530s, table=0, n_packets=1, n_bytes=98, priority=200,c
t_state=+new+trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=ct(commit),output:"s
8-eth4"
 cookie=0x0, duration=741.479s, table=0, n_packets=2, n_bytes=196, priority=200,
ct_state=+est+trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=output:"s8-eth4"
 cookie=0x0, duration=741.078s, table=0, n_packets=3, n_bytes=294, priority=200,
ct_state=+est+trk,ip,nw_src=10.0.0.7,nw_dst=10.0.0.8 actions=output:"s8-eth2"
 cookie=0x0, duration=738.945s, table=0, n_packets=4, n_bytes=168, priority=200,
arp actions=FLOOD
 cookie=0x0, duration=739.933s, table=1, n_packets=0, n_bytes=0, priority=100,ct
_state=+new+trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=ct(commit),output:"s8
-eth4"
 cookie=0x0, duration=739.881s, table=1, n_packets=0, n_bytes=0, priority=100,ct
_state=+est+trk,ip,nw_src=10.0.0.8,nw_dst=10.0.0.7 actions=output:"s8-eth4"
 cookie=0x0, duration=739.368s, table=1, n_packets=0, n_bytes=0, priority=100,ct
_state=+est+trk,ip,nw_src=10.0.0.7,nw_dst=10.0.0.8 actions=output:1
```

環境設置：安裝 **python**（版本**3.7**）、**ryu**、**mininet**

sudo apt update
sudo apt install python3 python3-pip –y

nano ~/.bashrc
最底下加入 export PATH=$HOME/.local/bin:$PATH
source ~/.bashrc
echo $PATH

sudo apt install python3.7 python3.7-venv python3.7-distutils
python3.7 --version
python3.7 -m venv venv_37
source venv_37/bin/activate
python --version
which python3
which pip3
pip3 install ryu
ryu-manager --version確認是否安裝成功
出現cannot import name 'AlREADY_HANDLED' from 'eventlet.wsgi':
pip3 install eventlet==0.30.2

sudo apt install mininet
sudo mn --version

啟動 **ryu**
啟動環境:source venv_37/bin/activate
ryu-manager ryu.app.simple_switch_13

執行 **python**
啟動環境:source venv_37/bin/activate
執行 py 檔

py檔內容:

```
import networkx as nx
from mininet.net import Mininet
from mininet.node import Controller, RemoteController
from mininet.link import TCLink
from mininet.cli import CLI
from mininet.log import setLogLevel
import subprocess

# return (switch, port_num)
def get_host_switch(host):
   host_switch = {"H1":("S1", 1), "H2":("S3", 1), "H3":("S7", 2),
"H4":("S5", 3), "H5":("S5", 4)
   , "H6":("S8", 3), "H7":("S8", 4), "H8":("S6", 1), "H9":("S4", 5)}
   if host[0] != "H":
       host = ip_to_hostname(host)
   return host_switch[host]

def find_two_shortest_paths(s, t):
```

```
    if s == t:
        return [[s]]

    G = nx.Graph()

    G.add_edge("S1", "S2")
    G.add_edge("S1", "S3")
    G.add_edge("S1", "S6")
    G.add_edge("S2", "S3")
    G.add_edge("S2", "S4")
    G.add_edge("S2", "S5")
    G.add_edge("S2", "S7")
    G.add_edge("S3", "S4")
    G.add_edge("S4", "S5")
    G.add_edge("S4", "S8")
    G.add_edge("S5", "S7")
    G.add_edge("S5", "S8")
    G.add_edge("S6", "S7")

    path1 = nx.shortest_path(G, source = s, target = t)
    # print("path1:", path1)

    middle_switch = path1[1:-1]
    # print(middle_switch)

    for switch in middle_switch:
        G.remove_node(switch)

    path2 = nx.shortest_path(G, source = s, target = t)
    # print("path2:", path2)

    return [path1, path2]

def get_link_port(switch1, switch2):
    switch_port = {"S1S2":(3,1), "S1S3":(4,2), "S1S6":(2,3),
"S2S3":(5,3), "S2S4":(4,2), "S2S5":(3,1), "S2S7":(2,4),
      "S3S4":(4,1), "S4S5":(3,6), "S4S8":(4,1), "S5S7":(2,3),
"S5S8":(5,2), "S6S7":(2,1)}
    if int(switch1[1]) > int(switch2[1]):
        ports = switch_port[switch2.upper()+switch1.upper()]
        return (ports[1], ports[0])
    return switch_port[switch1.upper()+switch2.upper()]
```

```
def get_switch(switch_name):
       switches_dict = {"S1":S1, "S2":S2, "S3":S3, "S4":S4, "S5":S5,
"S6":S6, "S7":S7, "S8":S8}
       return switches_dict[switch_name.upper()]

def network_topo(net, source, target, path):
   for i in range(len(path)-1):
       if not net.linksBetween(net.get(path[i].lower()),
net.get(path[i+1].lower())):
           ports = get_link_port(path[i], path[i+1])
           # print("net.addLink(", path[i], ", ", path[i+1], ",
port1=",ports[0], ", port2=",ports[1], ")", sep="")
           switch1 = get_switch(path[i])
           switch2 = get_switch(path[i+1])
           switch1_port = ports[0]
           switch2_port = ports[1]
           net.addLink(switch1, switch2, port1=switch1_port,
port2=switch2_port)
           switch1.cmd(f'ifconfig {path[i]}-eth{switch1_port} up')
           switch2.cmd(f'ifconfig {path[i+1]}-eth{switch2_port} up')

def hostname_to_ip(hostname):
   host_num = int(hostname[1:])
   return f"10.0.0.{host_num}"

def ip_to_hostname(ip):
   host_num = int(ip.split(".")[-1])
   return f"H{host_num}"

def clear_flows(switch):
   try:
       cmd = f"sudo ovs-ofctl del-flows {switch}"
       subprocess.run(cmd, shell=True, check=True)
   except subprocess.CalledProcessError as e:
       print(f"Failed to add flow rule: {e}")

def allow_ARP(switch, priority):
   try:
       cmd = f"sudo ovs-ofctl add-flow {switch}
\"priority={priority},dl_type=0x0806,action=flood\""
       subprocess.run(cmd, shell=True, check=True)
   except subprocess.CalledProcessError as e:
```

```
        print(f"Failed to add flow rule: {e}")

def add_openflow_rule(switch, rule):
    try:
        cmd = f"sudo ovs-ofctl add-flow {switch} \"{rule}\""
        subprocess.run(cmd, shell=True, check=True)
    except subprocess.CalledProcessError as e:
        print(f"Failed to add flow rule: {e}")

def add_openflow_rules(source_ip, target_ip, path, priority,
table_num):
    if len(path) == 1:
        # source to target
        # untracking
        #
print(f"add_openflow_rule({path[0]},priority={priority},dl_type=0x0800,
nw_src={source_ip},\
        #
nw_dst={target_ip},ct_state=-trk,actions=ct(table={table_num}))")
        add_openflow_rule(path[0],
f"priority={priority},dl_type=0x0800,nw_src={source_ip},\

nw_dst={target_ip},ct_state=-trk,actions=ct(table={table_num})")

        # +trk+new
        #
print(f"add_openflow_rule({path[0]},table={table_num},priority={priorit
y},dl_type=0x0800,nw_src={source_ip},\
        #
nw_dst={target_ip},ct_state=+trk+new,actions=ct(commit),output:{get_hos
t_switch(target_ip)[1]})")
        add_openflow_rule(path[0],
f"table={table_num},priority={priority},dl_type=0x0800,nw_src={source_i
p},\

nw_dst={target_ip},ct_state=+trk+new,actions=ct(commit),output:{get_hos
t_switch(target_ip)[1]}")

        # +trk+est
        #
print(f"add_openflow_rule({path[0]},table={table_num},priority={priorit
y},dl_type=0x0800,nw_src={source_ip},\
```

```
        #
nw_dst={target_ip},ct_state=+trk+est,actions=output:{get_host_switch(ta
rget_ip)[1]})")
        add_openflow_rule(path[0],
f"table={table_num},priority={priority},dl_type=0x0800,nw_src={source_i
p},\

nw_dst={target_ip},ct_state=+trk+est,actions=output:{get_host_switch(ta
rget_ip)[1]}")

        # target to source
        # -trk
        #
print(f"add_openflow_rule({path[0]},priority={priority},dl_type=0x0800,
nw_src={target_ip},\
        #
nw_dst={source_ip},ct_state=-trk,actions=ct(table={table_num}))",
sep="")
        add_openflow_rule(path[0],
f"priority={priority},dl_type=0x0800,nw_src={target_ip},\

nw_dst={source_ip},ct_state=-trk,actions=ct(table={table_num})")

        # +trk+est
        #
print(f"add_openflow_rule({path[0]},table={table_num},priority={priorit
y},dl_type=0x0800,nw_src={target_ip},\
        #
nw_dst={source_ip},ct_state=+trk+est,actions=output:{get_host_switch(so
urce_ip)[1]})")
        add_openflow_rule(path[0],
f"table={table_num},priority={priority},dl_type=0x0800,nw_src={target_i
p},\

nw_dst={source_ip},ct_state=+trk+est,actions=output:{get_host_switch(so
urce_ip)[1]}")

    else:
        # source to target
        for i in range(len(path)):
            if i == len(path)-1:
                # untracking
```

```
                #
print(f"add_openflow_rule({path[i]},priority={priority},dl_type=0x0800,
nw_src={source_ip},\
                #
nw_dst={target_ip},ct_state=-trk,actions=ct(table={table_num}))")
                add_openflow_rule(path[i],
f"priority={priority},dl_type=0x0800,nw_src={source_ip},\

nw_dst={target_ip},ct_state=-trk,actions=ct(table={table_num})")

                # +trk+new
                #
print(f"add_openflow_rule({path[i]},table={table_num},priority={priorit
y},dl_type=0x0800,nw_src={source_ip},\
                #
nw_dst={target_ip},ct_state=+trk+new,actions=ct(commit),output:{get_hos
t_switch(target_ip)[1]})", sep="")
                add_openflow_rule(path[i],
f"table={table_num},priority={priority},dl_type=0x0800,nw_src={source_i
p},\

nw_dst={target_ip},ct_state=+trk+new,actions=ct(commit),output:{get_hos
t_switch(target_ip)[1]}")

                # +trk+est
                #
print(f"add_openflow_rule({path[i]},table={table_num},priority={priorit
y},dl_type=0x0800,nw_src={source_ip},\
                #
nw_dst={target_ip},ct_state=+trk+est,actions=output:{get_host_switch(ta
rget_ip)[1]})", sep="")
                add_openflow_rule(path[i],
f"table={table_num},priority={priority},dl_type=0x0800,nw_src={source_i
p},\

nw_dst={target_ip},ct_state=+trk+est,actions=output:{get_host_switch(ta
rget_ip)[1]}")

            else:
                # untracking
                #
print(f"add_openflow_rule({path[i]},priority={priority},dl_type=0x0800,
nw_src={source_ip},\
```

```
                #
nw_dst={target_ip},ct_state=-trk,actions=ct(table={table_num}))",
sep="")
                add_openflow_rule(path[i],
f"priority={priority},dl_type=0x0800,nw_src={source_ip},\

nw_dst={target_ip},ct_state=-trk,actions=ct(table={table_num})")

                # +trk+new
                #
print(f"add_openflow_rule({path[i]},table={table_num},priority={priorit
y},dl_type=0x0800,nw_src={source_ip},\
                #
nw_dst={target_ip},ct_state=+trk+new,actions=ct(commit),output:{get_lin
k_port(path[i], path[i+1])[0]})", sep="")
                add_openflow_rule(path[i],
f"table={table_num},priority={priority},dl_type=0x0800,nw_src={source_i
p},\

nw_dst={target_ip},ct_state=+trk+new,actions=ct(commit),output:{get_lin
k_port(path[i], path[i+1])[0]}")

                # +trk+est
                #
print(f"add_openflow_rule({path[i]},table={table_num},priority={priorit
y},dl_type=0x0800,nw_src={source_ip},\
                #
nw_dst={target_ip},ct_state=+trk+est,actions=output:{get_link_port(path
[i], path[i+1])[0]})", sep="")
                add_openflow_rule(path[i],
f"table={table_num},priority={priority},dl_type=0x0800,nw_src={source_i
p},\

nw_dst={target_ip},ct_state=+trk+est,actions=output:{get_link_port(path
[i], path[i+1])[0]}")

        # target to source
        for i in range(len(path)):
            if i == 0:
                # -trk
                #
print(f"add_openflow_rule({path[i]},priority={priority},dl_type=0x0800,
nw_src={target_ip},\
```

```
                #
nw_dst={source_ip},ct_state=-trk,actions=ct(table={table_num}))",
sep="")
                add_openflow_rule(path[i],
f"priority={priority},dl_type=0x0800,nw_src={target_ip},\

nw_dst={source_ip},ct_state=-trk,actions=ct(table={table_num})")

                # +trk+est
                #
print(f"add_openflow_rule({path[i]},table={table_num},priority={priorit
y},dl_type=0x0800,nw_src={target_ip},\
                #
nw_dst={source_ip},ct_state=+trk+est,actions=output:{get_host_switch(so
urce_ip)[1]})", sep="")
                add_openflow_rule(path[i],
f"table={table_num},priority={priority},dl_type=0x0800,nw_src={target_i
p},\

nw_dst={source_ip},ct_state=+trk+est,actions=output:{get_host_switch(so
urce_ip)[1]}")

            else:
                # -trk
                #
print(f"add_openflow_rule({path[i]},priority={priority},dl_type=0x0800,
nw_src={target_ip},\
                #
nw_dst={source_ip},ct_state=-trk,actions=ct(table={table_num}))",
sep="")
                add_openflow_rule(path[i],
f"priority={priority},dl_type=0x0800,nw_src={target_ip},\

nw_dst={source_ip},ct_state=-trk,actions=ct(table={table_num})")

                # +trk+est
                #
print(f"add_openflow_rule({path[i]},table={table_num},priority={priorit
y},dl_type=0x0800,nw_src={target_ip},\
                #
nw_dst={source_ip},ct_state=+trk+est,actions=output:{get_link_port(path
[i-1], path[i])[1]})", sep="")
```

```
                add_openflow_rule(path[i],
f"table={table_num},priority={priority},dl_type=0x0800,nw_src={target_i
p},\

nw_dst={source_ip},ct_state=+trk+est,actions=output:{get_link_port(path
[i-1], path[i])[1]}")


"""Input source host and target host"""
print("Please input the hosts as H1, H2, ..., H9.")
source_host = input("Input source host: ")
while source_host[0] != "H" or (not source_host[1].isdigit()) or
int(source_host[1]) == 0 or len(source_host) != 2:
   print("Please input either H1, H2, ..., or H9 as source host")
   source_host = input("Input source host: ")

target_host = input("Input target host: ")
while target_host[0] != "H" or (not target_host[1].isdigit()) or
int(target_host[1]) == 0 or len(target_host) != 2:
   print("Please input either H1, H2, ..., or H9 as target host")
   target_host = input("Input target host: ")


"""Find paths"""
source_switch = get_host_switch(source_host)[0]
target_switch = get_host_switch(target_host)[0]
paths = find_two_shortest_paths(source_switch, target_switch)
# only 2 switches
if len(paths) >= 2 and paths[0] == paths[1]:
   paths = [paths[0]]

print("source:", source_host, " target:", target_host)
# path_num = 0
if len(paths) == 2:
   print("path1:", paths[0], "\npath2:", paths[1])
   # path_num = int(input("Choose one of the paths(1 or 2): "))-1
else:
   print("path:", paths[0])

try:
   """Create net"""
   net = Mininet(controller=RemoteController, link=TCLink,
cleanup=True)
```

```
    controller = net.addController('c0', controller=RemoteController,
ip='127.0.0.1', port=6633)
    # Add hosts
    H1 = net.addHost('H1', ip='10.0.0.1', mac='00:00:00:00:00:01')
    H2 = net.addHost('H2', ip='10.0.0.2', mac='00:00:00:00:00:02')
    H3 = net.addHost('H3', ip='10.0.0.3', mac='00:00:00:00:00:03')
    H4 = net.addHost('H4', ip='10.0.0.4', mac='00:00:00:00:00:04')
    H5 = net.addHost('H5', ip='10.0.0.5', mac='00:00:00:00:00:05')
    H6 = net.addHost('H6', ip='10.0.0.6', mac='00:00:00:00:00:06')
    H7 = net.addHost('H7', ip='10.0.0.7', mac='00:00:00:00:00:07')
    H8 = net.addHost('H8', ip='10.0.0.8', mac='00:00:00:00:00:08')
    H9 = net.addHost('H9', ip='10.0.0.9', mac='00:00:00:00:00:09')
    # Add switches
    S1 = net.addSwitch('s1')
    S2 = net.addSwitch('s2')
    S3 = net.addSwitch('s3')
    S4 = net.addSwitch('s4')
    S5 = net.addSwitch('s5')
    S6 = net.addSwitch('s6')
    S7 = net.addSwitch('s7')
    S8 = net.addSwitch('s8')
    # Create host's link to swtich
    net.addLink(H1, S1, port2=1)
    net.addLink(H2, S3, port2=1)
    net.addLink(H3, S7, port2=2)
    net.addLink(H4, S5, port2=3)
    net.addLink(H5, S5, port2=4)
    net.addLink(H6, S8, port2=3)
    net.addLink(H7, S8, port2=4)
    net.addLink(H8, S6, port2=1)
    net.addLink(H9, S4, port2=5)

    # Add network links (path1)
    network_topo(net, source_host, target_host, paths[0])
    net.start()

    """Flow Control"""
    switches = ["s1", "s2", "s3", "s4", "s5", "s6", "s7", "s8"]
    for switch in switches:
        clear_flows(switch)

    path = [s.lower() for s in paths[0]]
    # print(path)
```

```
    source_ip = hostname_to_ip(source_host)
    target_ip = hostname_to_ip(target_host)

    # for path1
    add_openflow_rules(source_ip, target_ip, path, 200, 0)

    if len(paths) >= 2:
        path = [s.lower() for s in paths[1]]
        # print(path)
        network_topo(net, source_host, target_host, paths[1])
        add_openflow_rules(source_ip, target_ip, path, 100, 1)

    for switch in switches:
        allow_ARP(switch, 200)

    CLI(net)

finally:
    net.stop()
    # if net not cleanup correctly, run "sudo mn -c"
    print()
```