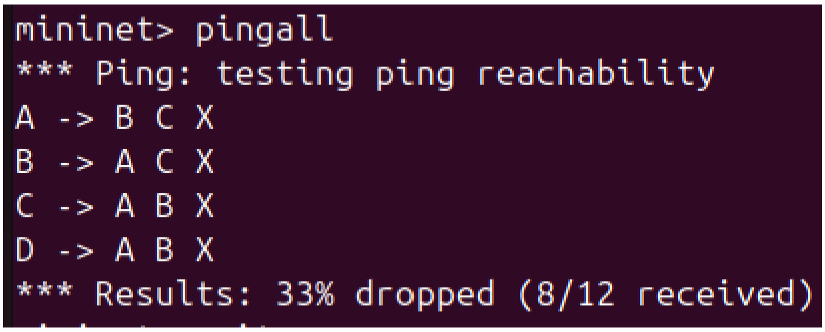
最終結果：



```
mininet> pingall
*** Ping: testing ping reachability
A -> B C X
B -> A C X
C -> A B X
D -> A B X
*** Results: 33% dropped (8/12 received)
```

—--

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 'AIREADY_HANDLED' from 'eventlet.wsgi':

```
pip3 install eventlet==0.30.2
```

```
sudo apt install mininet
sudo mn --version
```

啟動 ryu

```
source venv_37/bin/activate
ryu-manager ryu.app.simple_switch_13
```

啟動 mininet（用另一個 terminal）

```
source venv_37/bin/activate
(sudo mn --topo single,3 --controller=remote --ip=127.0.0.1 --switch=ovsk)
sudo mn topo.py（執行自己的拓樸）
```

拓樸設計步驟如下：

1. 建 Controller

2. 加入 Host（A, B, C, D）
3. 加入 switch（s1, s2, s3）
4. 建立連結

topo.py如下：

```
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

net = Mininet(controller=RemoteController, link=TCLink)

# Add controller
controller = net.addController('c0', controller=RemoteController, ip='127.0.0.1', port=6633)

# Add hosts
A = net.addHost('A', ip='10.0.0.1', mac='00:00:00:00:00:01')
B = net.addHost('B', ip='10.0.0.2', mac='00:00:00:00:00:02')
C = net.addHost('C', ip='10.0.0.3', mac='00:00:00:00:00:03')
D = net.addHost('D', ip='10.0.0.4', mac='00:00:00:00:00:04')

# Add switches
S1 = net.addSwitch('s1')
S2 = net.addSwitch('s2')
S3 = net.addSwitch('s3')

# Create links
net.addLink(A, S3, port2=1)
net.addLink(B, S1, port2=3)
net.addLink(C, S2, port2=3)
net.addLink(D, S2, port2=2)
net.addLink(S1, S2, port1=2, port2=4)
net.addLink(S2, S3, port1=1, port2=2)

# Start the network
net.start()
CLI(net)
net.stop()
```

另外開一個 **terminal** 輸入以下指令來調整 **flow**：

實作步驟說明：

1. 清空現存的 flow
2. 建立 A <-> B、B <-> C、C <-> A 的通道
3. 阻止 C 和 D 通信（把訊息drop掉）

4. 建立 D 和 A、B 的 connection tracking 實現 Three Way Handshake 的概念
    a. D 訪問 A、B 的 22 和 80 端口，commit 連線（目前此連線為untracked, ct_state=-trk）
    b. 允許已建立的連接通過（D -> A, B）（目前此連線為tracked, 且為新連線, ct_state=+trk+new）
    c. 反方向處理 syn-ack (A, B -> D)（ct_state=-trk）
    d. 允許已建立的連接通過（A. B -> D）（連線已建立, ct_state=+trk+est）
    e. 允許傳送流量（D -> A, B）（ct_state=+trk+est）
5. ARP 允許（為了讓主機能夠解析 MAC 地址）

指令:

```
# Clear all existing flows
sudo ovs-ofctl del-flows s1
sudo ovs-ofctl del-flows s2
sudo ovs-ofctl del-flows s3

# A 和 B 的雙向通信
# A -> B
sudo ovs-ofctl add-flow s3
"in_port=1,dl_type=0x0800,nw_src=10.0.0.1,nw_dst=10.0.0.2,priority=100,actions=output:2"
sudo ovs-ofctl add-flow s2
"in_port=1,dl_type=0x0800,nw_src=10.0.0.1,nw_dst=10.0.0.2,priority=100,actions=output:4"
sudo ovs-ofctl add-flow s1
"in_port=2,dl_type=0x0800,nw_src=10.0.0.1,nw_dst=10.0.0.2,priority=100,actions=output:3"

# B -> A
sudo ovs-ofctl add-flow s1
"in_port=3,dl_type=0x0800,nw_src=10.0.0.2,nw_dst=10.0.0.1,priority=100,actions=output:2"
sudo ovs-ofctl add-flow s2
"in_port=4,dl_type=0x0800,nw_src=10.0.0.2,nw_dst=10.0.0.1,priority=100,actions=output:1"
sudo ovs-ofctl add-flow s3
"in_port=2,dl_type=0x0800,nw_src=10.0.0.2,nw_dst=10.0.0.1,priority=100,actions=output:1"

# B 和 C 的雙向通信
# B -> C
sudo ovs-ofctl add-flow s1
"in_port=3,dl_type=0x0800,nw_src=10.0.0.2,nw_dst=10.0.0.3,priority=100,actions=output:2"
sudo ovs-ofctl add-flow s2
"in_port=4,dl_type=0x0800,nw_src=10.0.0.2,nw_dst=10.0.0.3,priority=100,actions=output:3"

# C -> B
sudo ovs-ofctl add-flow s2
"in_port=3,dl_type=0x0800,nw_src=10.0.0.3,nw_dst=10.0.0.2,priority=100,actions=output:4"
sudo ovs-ofctl add-flow s1
"in_port=2,dl_type=0x0800,nw_src=10.0.0.3,nw_dst=10.0.0.2,priority=100,actions=output:3"
```

```
# A 和 C 的雙向通信
# A -> C
sudo ovs-ofctl add-flow s3
"in_port=1,dl_type=0x0800,nw_src=10.0.0.1,nw_dst=10.0.0.3,priority=100,actions=output:2"
sudo ovs-ofctl add-flow s2
"in_port=1,dl_type=0x0800,nw_src=10.0.0.1,nw_dst=10.0.0.3,priority=100,actions=output:3"

# C -> A
sudo ovs-ofctl add-flow s2
"in_port=3,dl_type=0x0800,nw_src=10.0.0.3,nw_dst=10.0.0.1,priority=100,actions=output:1"
sudo ovs-ofctl add-flow s3
"in_port=2,dl_type=0x0800,nw_src=10.0.0.3,nw_dst=10.0.0.1,priority=100,actions=output:1"

# 阻止 D 和 C 的互相通信
sudo ovs-ofctl add-flow s2
"in_port=2,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.3,priority=500,actions=drop"
sudo ovs-ofctl add-flow s2
"in_port=3,dl_type=0x0800,nw_src=10.0.0.3,nw_dst=10.0.0.4,priority=500,actions=drop"

# D 訪問 A、B 的 22 和 80 端口 (使用 connection tracking)
# D -> A
sudo ovs-ofctl add-flow s2
"priority=200,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.1,tcp_dst=22,ct_state=-trk,act
ions=ct(table=1)"
sudo ovs-ofctl add-flow s2
"priority=200,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.1,tcp_dst=80,ct_state=-trk,act
ions=ct(table=1)"
sudo ovs-ofctl add-flow s3
"priority=200,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.1,tcp_dst=22,ct_state=-trk,act
ions=ct(table=1)"
sudo ovs-ofctl add-flow s3
"priority=200,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.1,tcp_dst=80,ct_state=-trk,act
ions=ct(table=1)"

# D -> B
sudo ovs-ofctl add-flow s2
"priority=200,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.2,tcp_dst=22,ct_state=-trk,act
ions=ct(table=1)"
sudo ovs-ofctl add-flow s2
"priority=200,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.2,tcp_dst=80,ct_state=-trk,act
ions=ct(table=1)"
sudo ovs-ofctl add-flow s1
"priority=200,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.2,tcp_dst=22,ct_state=-trk,act
ions=ct(table=1)"
sudo ovs-ofctl add-flow s1
"priority=200,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.2,tcp_dst=80,ct_state=-trk,act
ions=ct(table=1)"
```

```
# 允許已建立的連接通過（D -> A, B）
# D -> A (port 22, 80)
sudo ovs-ofctl add-flow s2
"table=1,priority=200,ct_state=+trk+new,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.1,t
cp_dst=22,actions=ct(commit),output:1"
sudo ovs-ofctl add-flow s2
"table=1,priority=200,ct_state=+trk+est,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.1,tc
p_dst=22,actions=output:1"
sudo ovs-ofctl add-flow s3
"table=1,priority=200,ct_state=+trk+new,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.1,t
cp_dst=22,actions=ct(commit),output:1"
sudo ovs-ofctl add-flow s3
"table=1,priority=200,ct_state=+trk+est,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.1,tc
p_dst=22,actions=output:1"

sudo ovs-ofctl add-flow s2
"table=1,priority=200,ct_state=+trk+new,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.1,t
cp_dst=80,actions=ct(commit),output:1"
sudo ovs-ofctl add-flow s2
"table=1,priority=200,ct_state=+trk+est,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.1,tc
p_dst=80,actions=output:1"
sudo ovs-ofctl add-flow s3
"table=1,priority=200,ct_state=+trk+new,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.1,t
cp_dst=80,actions=ct(commit),output:1"
sudo ovs-ofctl add-flow s3
"table=1,priority=200,ct_state=+trk+est,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.1,tc
p_dst=80,actions=output:1"

# D -> B (port 22, 80)
sudo ovs-ofctl add-flow s2
"table=1,priority=200,ct_state=+trk+new,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.2,t
cp_dst=22,actions=ct(commit),output:4"
sudo ovs-ofctl add-flow s2
"table=1,priority=200,ct_state=+trk+est,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.2,tc
p_dst=22,actions=output:4"
sudo ovs-ofctl add-flow s1
"table=1,priority=200,ct_state=+trk+new,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.2,t
cp_dst=22,actions=ct(commit),output:3"
sudo ovs-ofctl add-flow s1
"table=1,priority=200,ct_state=+trk+est,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.2,tc
p_dst=22,actions=output:3"

sudo ovs-ofctl add-flow s2
"table=1,priority=200,ct_state=+trk+new,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.2,t
cp_dst=80,actions=ct(commit),output:4"
sudo ovs-ofctl add-flow s2
"table=1,priority=200,ct_state=+trk+est,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.2,tc
p_dst=80,actions=output:4"
```

```
sudo ovs-ofctl add-flow s1 "table=1,priority=200,ct_state=+trk+new,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.2,tcp_dst=80,actions=ct(commit),output:3"
sudo ovs-ofctl add-flow s1 "table=1,priority=200,ct_state=+trk+est,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.2,tcp_dst=80,actions=output:3"

# 從反方向處理 syn-ack
# A  -> D
sudo ovs-ofctl add-flow s3 "priority=200,dl_type=0x0800,nw_src=10.0.0.1,nw_dst=10.0.0.4,tcp_dst=22,ct_state=-trk,actions=ct(table=1)"
sudo ovs-ofctl add-flow s3 "priority=200,dl_type=0x0800,nw_src=10.0.0.1,nw_dst=10.0.0.4,tcp_dst=80,ct_state=-trk,actions=ct(table=1)"
sudo ovs-ofctl add-flow s2 "priority=200,dl_type=0x0800,nw_src=10.0.0.1,nw_dst=10.0.0.4,tcp_dst=22,ct_state=-trk,actions=ct(table=1)"
sudo ovs-ofctl add-flow s2 "priority=200,dl_type=0x0800,nw_src=10.0.0.1,nw_dst=10.0.0.4,tcp_dst=80,ct_state=-trk,actions=ct(table=1)"

# B -> D
sudo ovs-ofctl add-flow s1 "priority=200,dl_type=0x0800,nw_src=10.0.0.2,nw_dst=10.0.0.4,tcp_dst=22,ct_state=-trk,actions=ct(table=1)"
sudo ovs-ofctl add-flow s1 "priority=200,dl_type=0x0800,nw_src=10.0.0.2,nw_dst=10.0.0.4,tcp_dst=80,ct_state=-trk,actions=ct(table=1)"
sudo ovs-ofctl add-flow s2 "priority=200,dl_type=0x0800,nw_src=10.0.0.2,nw_dst=10.0.0.4,tcp_dst=22,ct_state=-trk,actions=ct(table=1)"
sudo ovs-ofctl add-flow s2 "priority=200,dl_type=0x0800,nw_src=10.0.0.2,nw_dst=10.0.0.4,tcp_dst=80,ct_state=-trk,actions=ct(table=1)"

# 允許已建立的連接通過（A. B -> D）
# A -> D (port 22, 80)
sudo ovs-ofctl add-flow s3 "table=1,priority=200,ct_state=+trk+est,dl_type=0x0800,nw_src=10.0.0.1,nw_dst=10.0.0.4,tcp_dst=22,actions=output:2"
sudo ovs-ofctl add-flow s2 "table=1,priority=200,ct_state=+trk+est,dl_type=0x0800,nw_src=10.0.0.1,nw_dst=10.0.0.4,tcp_dst=22,actions=output:2"
sudo ovs-ofctl add-flow s3 "table=1,priority=200,ct_state=+trk+est,dl_type=0x0800,nw_src=10.0.0.1,nw_dst=10.0.0.4,tcp_dst=80,actions=output:2"
sudo ovs-ofctl add-flow s2
"table=1,priority=200,ct_state=+trk+est,dl_type=0x0800,nw_src=10.0.0.1,nw_dst=10.0.0.4,tcp_dst=80,actions=output:2"

# B -> D (port 22, 80)
sudo ovs-ofctl add-flow s1
"table=1,priority=200,ct_state=+trk+est,dl_type=0x0800,nw_src=10.0.0.2,nw_dst=10.0.0.4,tcp_dst=22,actions=output:2"
sudo ovs-ofctl add-flow s2
"table=1,priority=200,ct_state=+trk+est,dl_type=0x0800,nw_src=10.0.0.2,nw_dst=10.0.0.4,tcp_dst=22,actions=output:2"
sudo ovs-ofctl add-flow s1
"table=1,priority=200,ct_state=+trk+est,dl_type=0x0800,nw_src=10.0.0.2,nw_dst=10.0.0.4,tcp_dst=80,actions=output:2"
sudo ovs-ofctl add-flow s2
"table=1,priority=200,ct_state=+trk+est,dl_type=0x0800,nw_src=10.0.0.2,nw_dst=10.0.0.4,tcp_dst=80,actions=output:2"

# 允許相關的返回流量
# D -> A
sudo ovs-ofctl add-flow s2
"priority=200,ct_state=+trk+est,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.1,tcp_src=22,actions=output:1"
sudo ovs-ofctl add-flow s2
"priority=200,ct_state=+trk+est,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.1,tcp_src=80,actions=output:1"
sudo ovs-ofctl add-flow s3
"priority=200,ct_state=+trk+est,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.1,tcp_src=22,actions=output:1"
sudo ovs-ofctl add-flow s3
"priority=200,ct_state=+trk+est,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.1,tcp_src=80,actions=output:1"

# D -> B
sudo ovs-ofctl add-flow s2
"priority=200,ct_state=+trk+est,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.2,tcp_src=22,actions=output:4"
sudo ovs-ofctl add-flow s2
"priority=200,ct_state=+trk+est,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.2,tcp_src=80,actions=output:4"
sudo ovs-ofctl add-flow s1
"priority=200,ct_state=+trk+est,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.2,tcp_src=22,actions=output:3"
sudo ovs-ofctl add-flow s1
"priority=200,ct_state=+trk+est,dl_type=0x0800,nw_src=10.0.0.4,nw_dst=10.0.0.2,tcp_src=80,actions=output:3"

# ARP 允許（為了讓主機能夠解析 MAC 地址）
```

```
sudo ovs-ofctl add-flow s1 "dl_type=0x0806,priority=100,actions=flood"
sudo ovs-ofctl add-flow s2 "dl_type=0x0806,priority=100,actions=flood"
sudo ovs-ofctl add-flow s3 "dl_type=0x0806,priority=100,actions=flood"
```