





資料庫與網路安全實驗室

Database System and Network Security Lab



# 5G 認證流程

以 *Mininet* 拓樸，*Ryu* 控制

St425

# O *Install*

在 *Virtual Box* 上建立 *Linux* 虛擬環境，*ubuntu-18.04.5-desktop-amd64.iso*，使用 *Mininet* 拓樸，使用 *Ryu* 控制傳輸，模擬 5G 認證過程。

環境：

**Mac(OS):** Big Sur 11.1

**App:** Virtual Box 6.1.16 r140961

環境：

**Win:** Win10

**App:** Virtual Box 6.1.16 r140961

*Virtual Box:*

**RAM:** 8192 MB

**虛擬硬碟大小:** 60.0 GB

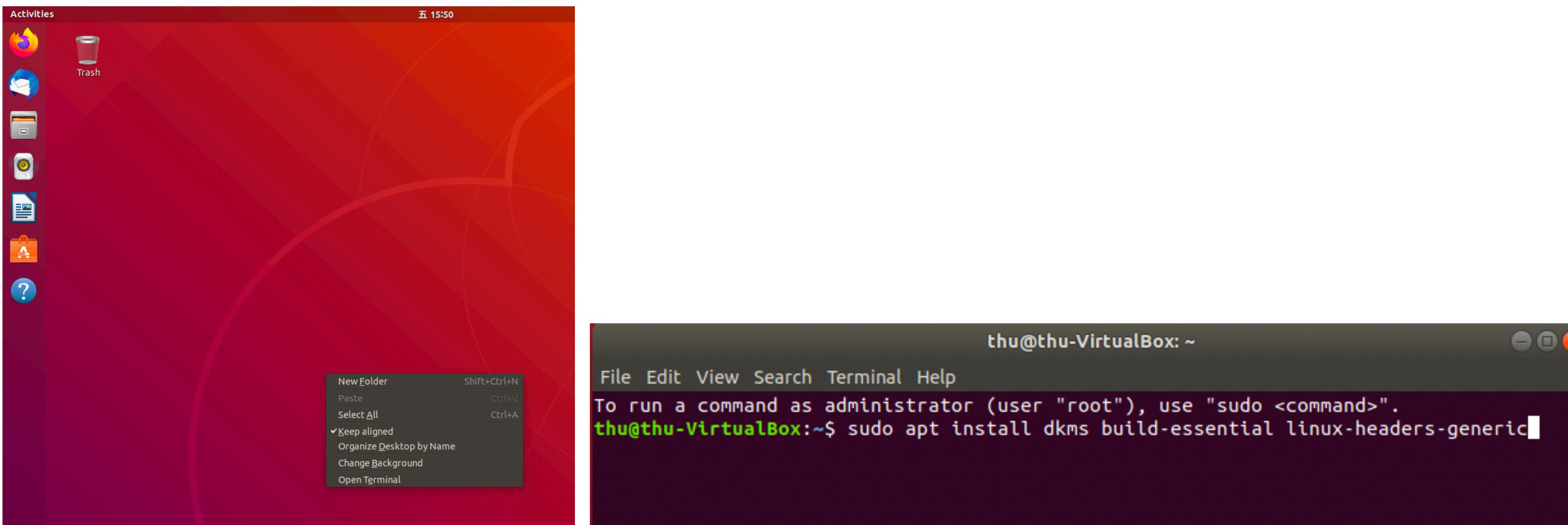
*Virtual Box:*

**RAM:** 16384 MB

**虛擬硬碟大小:** 60.0 GB

# 1 Install Ubuntu 複製貼上

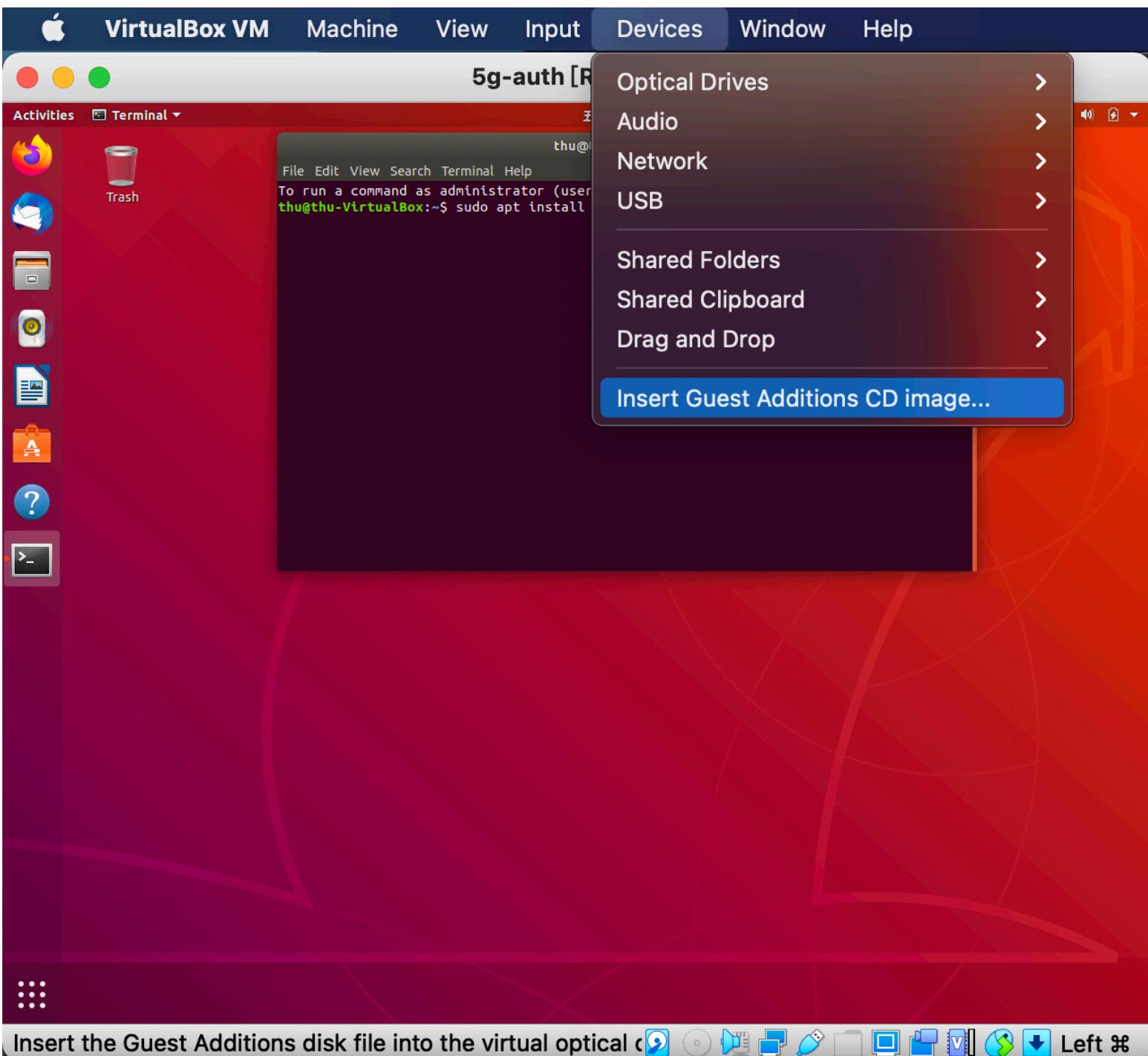
進入 *Ubuntu* 內，為了方便能從 *Mac* 到 *Virtual Box(Ubuntu)* 複製貼上，請進行以下操作。



*Right Click & Select: open Terminal*

\$ sudo apt install dkms build-essential linux-headers-generic

# 1 Install Ubuntu 複製貼上

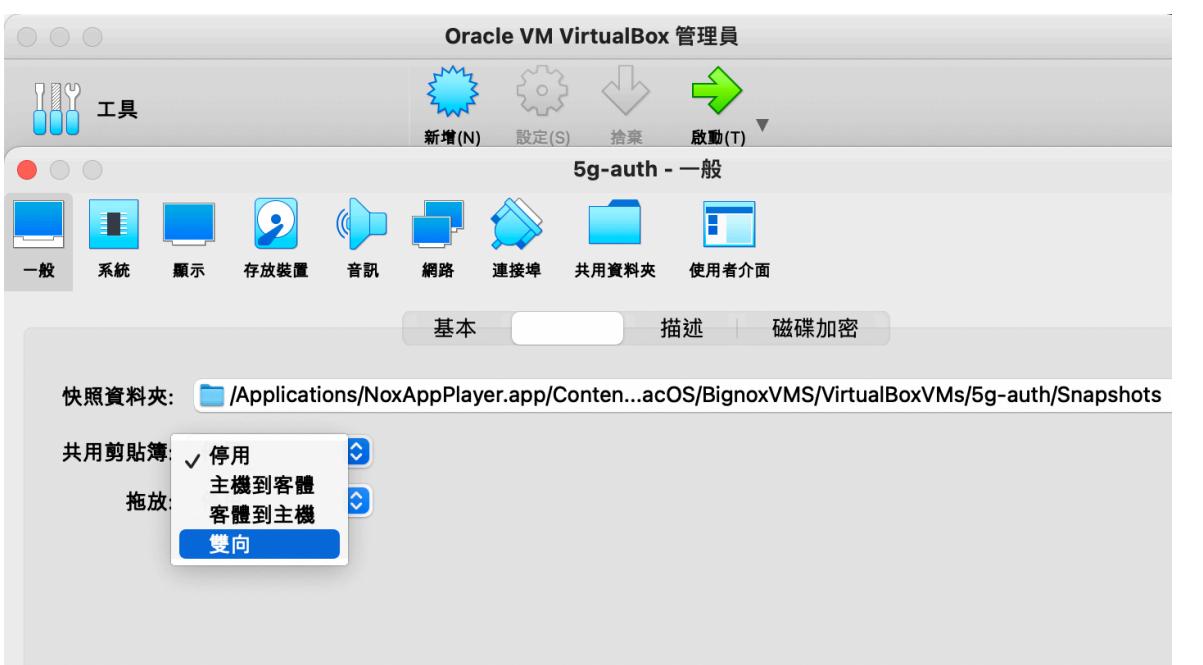


*VirtualBox Menu > Devices > Insert Guest Additions CD images ...*

# 1 Install Ubuntu 複製貼上

\$ shutdown -f now

VirtualBox > 設定 > 一般 > 進階 > 共用剪貼簿 > 雙向



—Restart Ubuntu—

即可從 Mac 到 Virtual Box(Ubuntu) 複製貼上

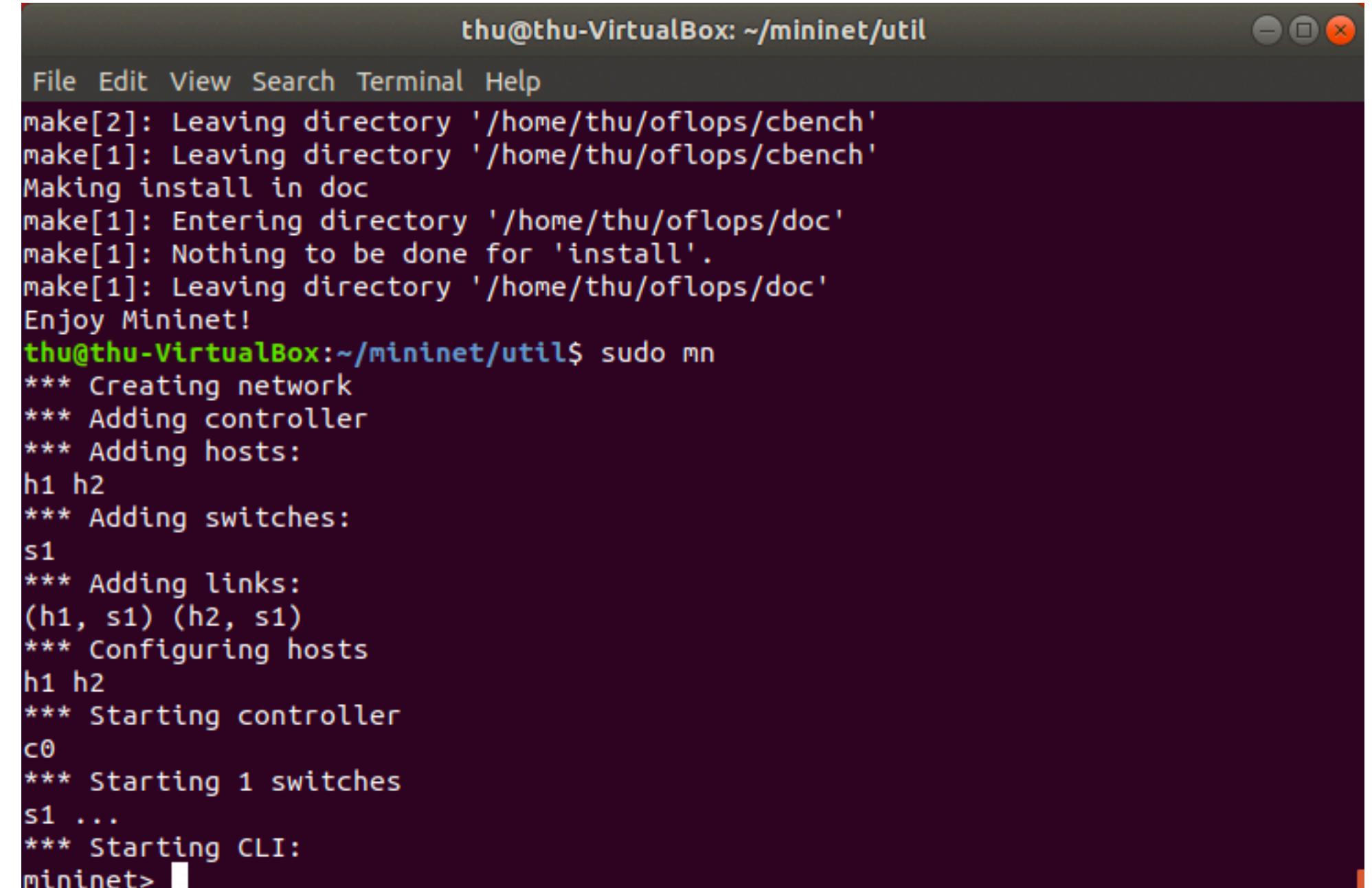
# 2

# Mininet + Ryu

## Mininet install

```
$ sudo apt-get update  
$ sudo apt-get upgrade  
$ sudo apt-get install -y git  
  
$ git clone git://github.com/mininet/mininet  
$ cd mininet/util  
$ sudo ./install.sh -a  
  
$ sudo mn
```

測試 Mininet 安裝是否成功



The terminal window shows the following output:

```
thu@thu-VirtualBox: ~/mininet/util  
File Edit View Search Terminal Help  
make[2]: Leaving directory '/home/thu/oflops/cbench'  
make[1]: Leaving directory '/home/thu/oflops/cbench'  
Making install in doc  
make[1]: Entering directory '/home/thu/oflops/doc'  
make[1]: Nothing to be done for 'install'.  
make[1]: Leaving directory '/home/thu/oflops/doc'  
Enjoy Mininet!  
thu@thu-VirtualBox:~/mininet/util$ sudo mn  
*** Creating network  
*** Adding controller  
*** Adding hosts:  
h1 h2  
*** Adding switches:  
s1  
*** Adding links:  
(h1, s1) (h2, s1)  
*** Configuring hosts  
h1 h2  
*** Starting controller  
c0  
*** Starting 1 switches  
s1 ...  
*** Starting CLI:  
mininet> █
```

mininet> exit  
即可退出

# 2

# *Mininet + Ryu*

## *Ryu install*

```
$ cd ~  
$ sudo apt-get update  
$ sudo apt-get install libxml2-dev libxslt1-dev python-pip python-eventlet python-routes  
python-webob python-paramiko  
$ sudo pip install msgpack-python eventlet==0.15.2  
$ sudo pip install six --upgrade  
$ sudo pip install oslo.config q --upgrade  
  
$ sudo apt install python3-pip
```

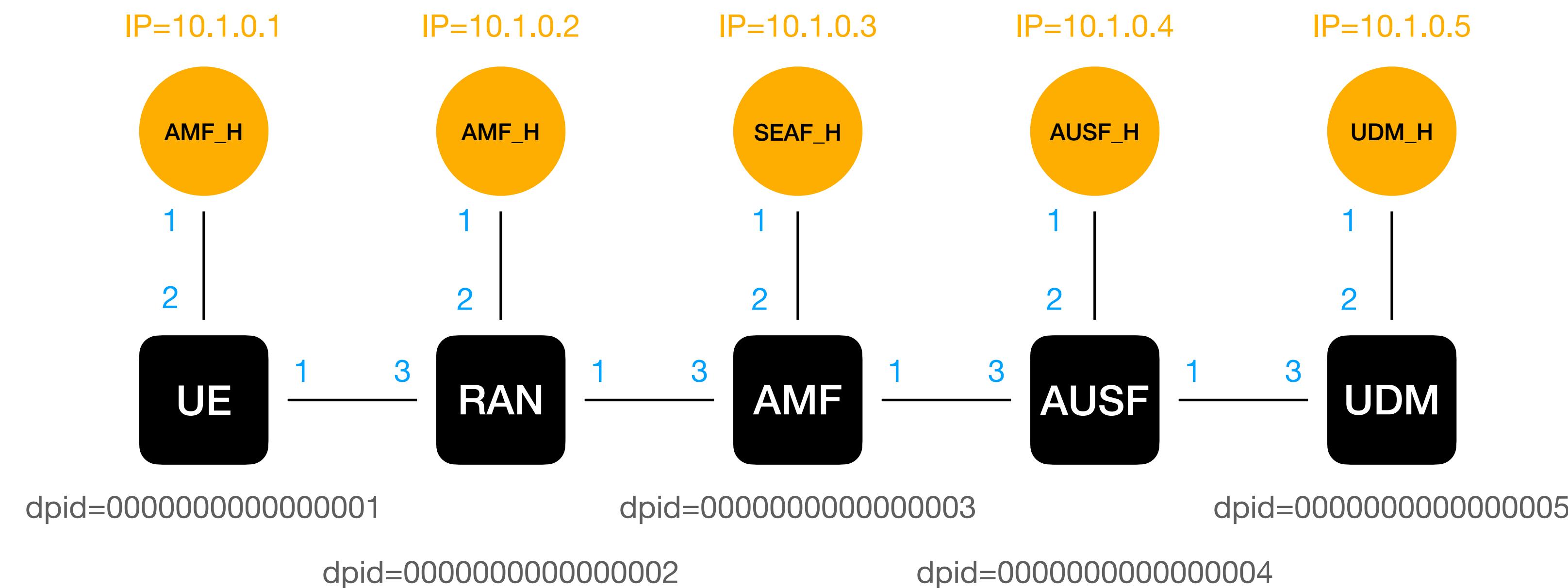
# 2

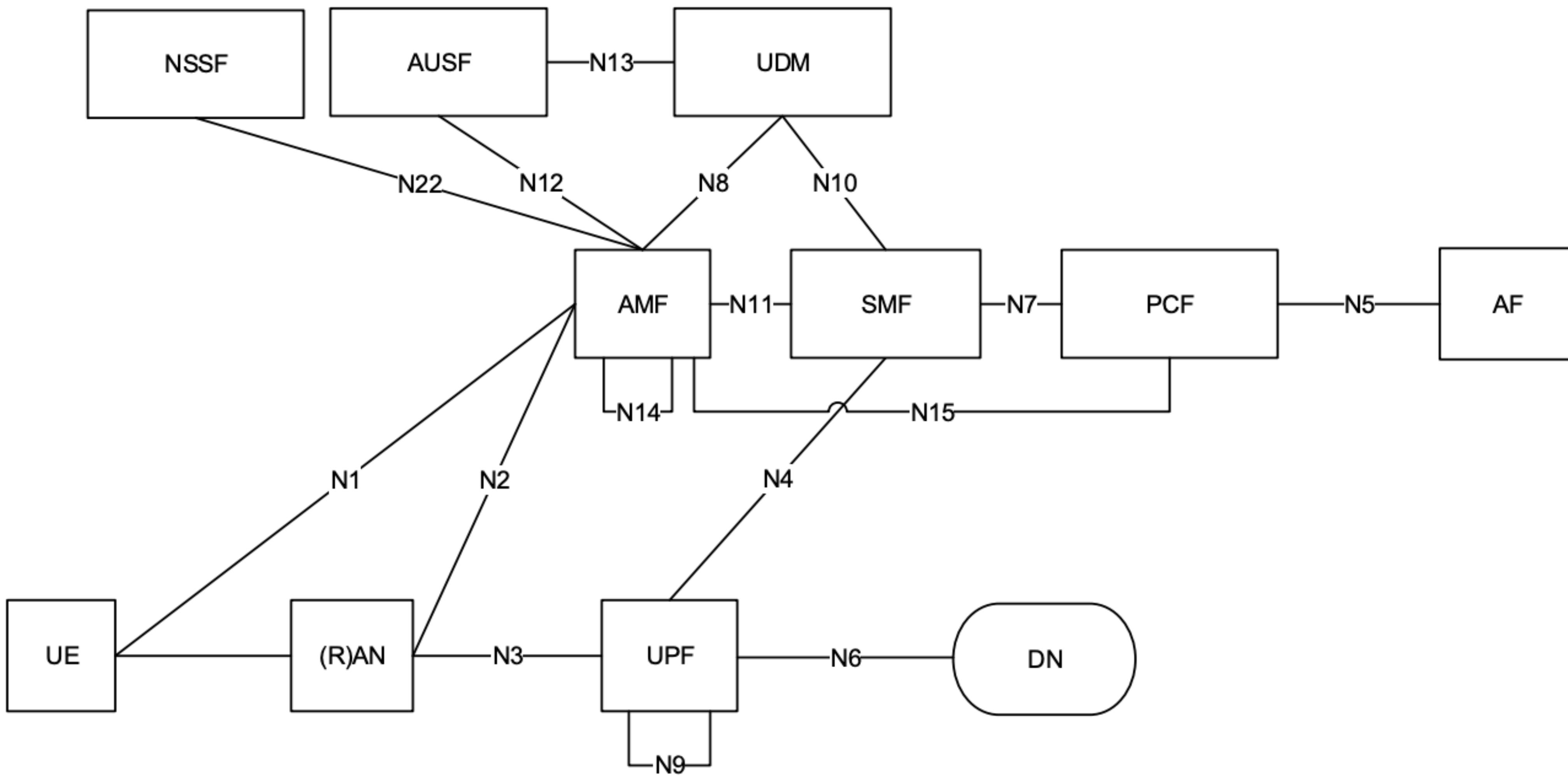
# *Mininet + Ryu*

## *Ryu install*

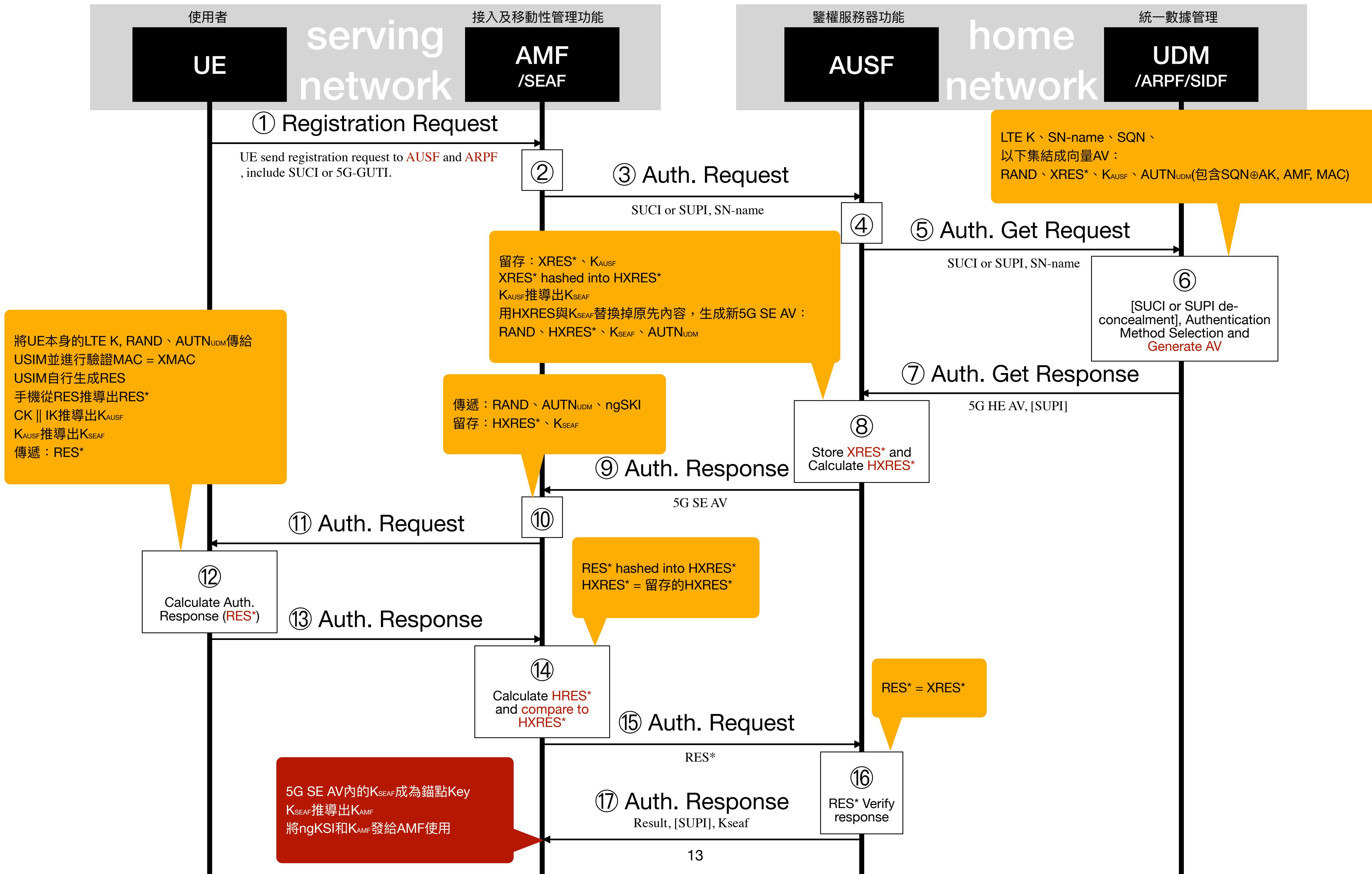
```
$ cd ~  
$ git clone https://github.com/faucetsdn/ryu.git  
$ cd ryu  
$ pip3 install .  
$ sudo apt install python-ryu  
  
$ ryu-manager --verbose ryu.app.ofctl_rest
```

# 3 5G Topo Architecture





非漫遊 5G 架構



計算消息認證碼(MAC) :  $MAC = F^1_K(SQN \parallel RAND \parallel AMF)$  ;

計算期望的認證應答(XRES) :  $XRES = F^2_K(RAND)$  ;

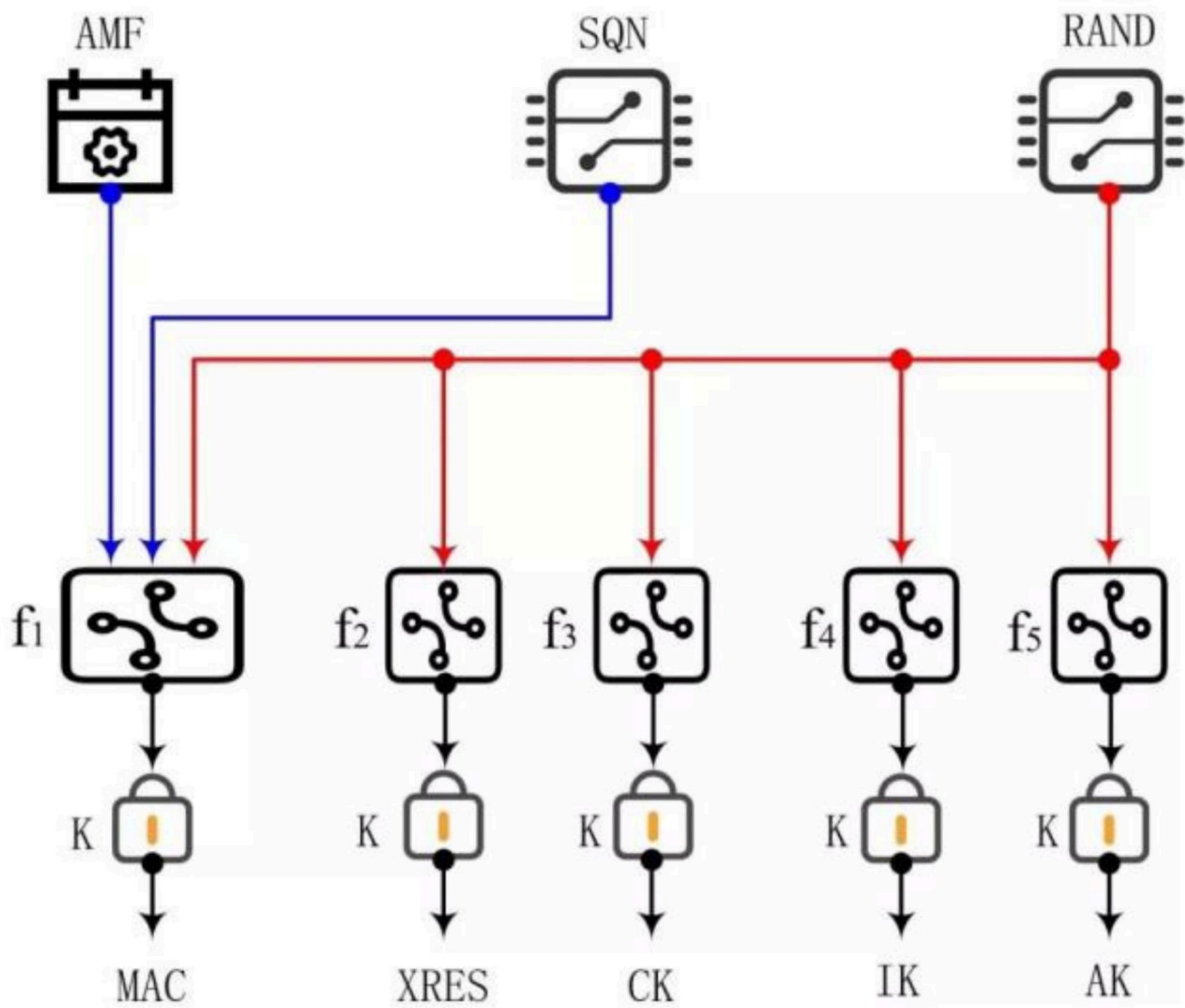
計算保密性密鑰(CK) :  $CK = F^3_K(RAND)$  ;

計算完整性密鑰(IK) :  $IK = F^4_K(RAND)$  ;

匿名密鑰(AK) :  $AK = F^5_K(RAND)$  ;

網絡認證令牌(AUTN) :  $AUTN = SQN \oplus AK \parallel AMF \parallel MAC$  ;

AV :  $AV = RAND \parallel XRES \parallel CK \parallel IK \parallel AUTN$  ;



AK用來隱藏SQN，因為SQN可能會暴露用戶的位置信息。如果不需要隱藏SQN，那麼AK被設置為0。

# 3

# 5G Topo

## 建立 *5gtopo.py* 在 *mininet/custom/5gAuth* 內

```
from mininet.topo import Topo

class Topo1(Topo):
    def __init__(self):
        #Initialize topology
        Topo.__init__(self)

        #add Host and Switch
        #UE
        ue=self.addHost('UE',ip='10.1.0.1')

        ues=self.addSwitch('UEs',dpid='0000000000000001')

        #RAN
        ran=self.addHost('RAN',ip='10.1.0.2')

        rans=self.addSwitch('RANs',dpid='0000000000000002')
        #AMF
        amf=self.addHost('AMF',ip='10.1.0.3')

        amfs=self.addSwitch('AMFs',dpid='0000000000000003')
        #AUSF
        ausf=self.addHost('AUSF',ip='10.1.0.4')

        ausfs=self.addSwitch('AUSFs',dpid='0000000000000004')
        #UDM
        udm=self.addHost('UDM',ip='10.1.0.5')

        udms=self.addSwitch('UDMs',dpid='0000000000000005')

        #add Links
        self.addLink(ue,ues,port1=1,port2=2)
        self.addLink(ues,rans,port1=1,port2=3)
        self.addLink(rans,ran,port1=2,port2=1)
        self.addLink(rans,amfs,port1=1,port2=3)
        self.addLink(amfs,amf,port1=2,port2=1)
        self.addLink(amfs,ausfs,port1=1,port2=3)
        self.addLink(ausfs,ausf,port1=2,port2=1)
        self.addLink(ausfs,udms,port1=1,port2=3)
        self.addLink(udms,udm,port1=2,port2=1)

    topo={'mytopo':(lambda:Topo1())}
```

# 3 5G Topo *Ryu + Mininet*

開啟 第一個 *Terminal (Ryu)*

```
$ ryu-manager ryu.app.simple_switch_13
```

開啟 第二個 *Terminal (Mininet)*

```
$ cd ~/mininet/custom/5gAuth
$ sudo mn --custom 5gtopo.py --topo
mytopo --
controller=remote,ip=127.0.0.1,port=6633
--switch ovs,protocols=OpenFlow13
$ mininet > pingall
$ mininet > xterm UE RAN
```

# 3 5G Topo Client

在 ~/mininet/custom/5gAuth 寫入 Client.py

```
$ import socket  
  
$ HOST = '10.1.0.2'  
  
$ PORT = 1  
  
$ s = socket.socket(socket.AF_INET,  
socket.SOCK_STREAM)  
  
$ s.connect((HOST, PORT))  
  
$ while True:  
  
$     cmd = raw_input("Please input msg:")  
  
$     s.send(cmd)
```

```
$     data = s.recv(1024)
```

```
$     print "server send : %s " % (data)
```

# 3 5G Topo Server

在 ~/mininet/custom/5gAuth 寫入 Server.py

```
$ import socket  
  
$ bind_ip = "10.1.0.2"  
  
$ bind_port = 1  
  
$ server =  
    socket.socket(socket.AF_INET,socket.SOCK_STREAM)  
$ server.bind((bind_ip,bind_port))  
$ server.listen(5)  
  
$ print "[*] Listening on %s:%d" % (bind_ip,bind_port)
```

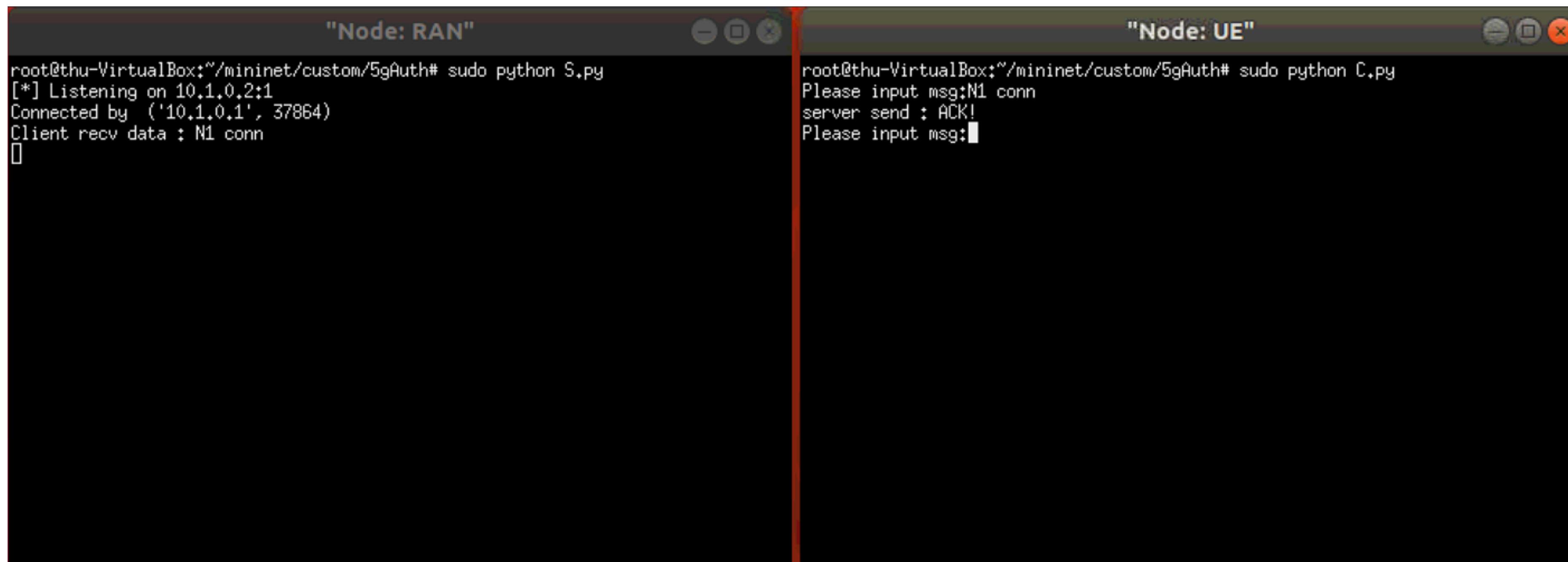
```
$ while True:  
$     client,addr = server.accept()  
$     print 'Connected by ', addr  
$     while True:  
$         data = client.recv(1024)  
$         print "Client recv data : %s " % (data)  
$         client.send("ACK!")
```

# 3 5G Topo

## *UE connect to RAN*

*xterm (RAN) >*

```
$ sudo python Server.py
```



The image shows two terminal windows side-by-side. The left window is titled "Node: RAN" and the right window is titled "Node: UE". Both windows are running on a Linux system (Ubuntu) as indicated by the root prompt.

In the "Node: RAN" window, the command `sudo python Server.py` is run. The output shows:

```
root@thu-VirtualBox:~/mininet/custom/5gAuth# sudo python S.py
[*] Listening on 10.1.0.2:1
Connected by ('10.1.0.1', 37864)
Client recv data : N1 conn
[]
```

In the "Node: UE" window, the command `sudo python Client.py` is run. The output shows:

```
root@thu-VirtualBox:~/mininet/custom/5gAuth# sudo python C.py
Please input msg:N1 conn
server send : ACK!
Please input msg:[]
```

可從 UE 傳遞信息至 RAN

# 4 未完成

把 p.13 「5G 認證流程」，從  $UE \rightarrow RAN \rightarrow AMF \rightarrow AUSF \rightarrow UDF$ ，依序模擬並傳遞信息。