MiniProject 1 姓名:郭紘安 學號:109062578

1. What you do :

Step 1 : Design architecture

Step 2 : Create Topology(.cc)

```
/************************************Topology*************************************/
//建立使用 P2P 連線的兩個網路節點
NodeContainer p2pNodes;
p2pNodes.Create (2);
//設置 datarate 和 delay
PointToPointHelper pointToPoint;
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));
//安裝 P2P 網卡到 P2P 網路節點
NetDeviceContainer p2pDevices;
p2pDevices = pointToPoint.Install (p2pNodes);
//建立 STA 節點
NodeContainer wifiStaNodes1;
wifiStaNodes1.Create (nWifi);
//將 node 0 設為 AP
NodeContainer wifiApNode1 = p2pNodes.Get (0);
//建立 STA 節點
NodeContainer wifiStaNodes2;
wifiStaNodes2.Create (nWifi);
//將 node 1 設為 AP
NodeContainer wifiApNode2 = p2pNodes.Get (1);
//將 channel 初始化
YansWifiChannelHelper channel1 = YansWifiChannelHelper::Default ();
YansWifiPhyHelper phy1 = YansWifiPhyHelper::Default ();
phy1.SetChannel (channel1.Create ());
//將 channel 初始化
YansWifiChannelHelper channel2 = YansWifiChannelHelper::Default ();
YansWifiPhyHelper phy2 = YansWifiPhyHelper::Default ();
phy2.SetChannel (channel2.Create ());

WifiHelper wifi;
wifi.SetRemoteStationManager ("ns3::AarfWifiManager");
//設置 STA MAC
WifiMacHelper mac1;
Ssid ssid1 = Ssid ("ns-3-ssid");
mac1.SetType ("ns3::StaWifiMac",
            "Ssid", SsidValue (ssid1),
            "ActiveProbing", BooleanValue (false));
```

```
//設置 STA MAC
WifiMacHelper mac2;
Ssid ssid2 = Ssid ("ns-3-ssid");
mac2.SetType ("ns3::StaWifiMac",
            "Ssid", SsidValue (ssid2),
            "ActiveProbing", BooleanValue (false));
//安裝網卡到連接 AP 的 STA 節點
NetDeviceContainer staDevices1;
staDevices1 = wifi.Install (phy1, mac1, wifiStaNodes1);
//安裝網卡到連接 AP 的 STA 節點
NetDeviceContainer staDevices2;
staDevices2 = wifi.Install (phy2, mac2, wifiStaNodes2);
//設置 AP MAC
mac1.SetType ("ns3::ApWifiMac",
            "Ssid", SsidValue (ssid1));
//設置 AP MAC
mac2.SetType ("ns3::ApWifiMac",
            "Ssid", SsidValue (ssid2));

//安裝網卡到 AP
NetDeviceContainer apDevices1;
apDevices1 = wifi.Install (phy1, mac1, wifiApNode1);
//安裝網卡到 AP
NetDeviceContainer apDevices2;
apDevices2 = wifi.Install (phy2, mac2, wifiApNode2);
//增加移動模型
MobilityHelper mobility;
mobility.SetPositionAllocator ("ns3::GridPositionAllocator",
                               "MinX", DoubleValue (0.0),
                               "MinY", DoubleValue (0.0),
                               "DeltaX", DoubleValue (5.0),
                               "DeltaY", DoubleValue (10.0),
                               "GridWidth", UintegerValue (3),
                               "LayoutType", StringValue ("RowFirst"));
mobility.SetMobilityModel ("ns3::RandomWalk2dMobilityModel",
                           "Bounds", RectangleValue (Rectangle (-50, 50, -50, 50)));
//將 STA AP 安裝到模型上
mobility.Install (wifiStaNodes1);
mobility.Install (wifiStaNodes2);
mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
mobility.Install (wifiApNode1);
mobility.Install (wifiApNode2);

//安裝網路協議
InternetStackHelper stack;
stack.Install (wifiApNode1);
stack.Install (wifiStaNodes1);
stack.Install (wifiApNode2);
stack.Install (wifiStaNodes2);

Ipv4AddressHelper address;
//為 P2P 設置 IP address
address.SetBase ("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer p2pInterfaces;
p2pInterfaces = address.Assign (p2pDevices);
//為 AP(n0)設置 IP address
address.SetBase ("10.1.2.0", "255.255.255.0");
Ipv4InterfaceContainer staInterfaces;
```

```
staInterfaces = address.Assign (staDevices1);
address.Assign (apDevices1);
//為 AP(n0)設置 IP address
address.SetBase ("10.1.3.0", "255.255.255.0");
address.Assign (staDevices2);
address.Assign (apDevices2);
/****************************************************************************/
```

Step 3 : Create Application(.cc)

```
/*********************************Application*********************************/
UdpEchoServerHelper echoServer (9);

//將 server 安裝在 WIFI 的到數第二個節點上
ApplicationContainer serverApps = echoServer.Install (wifiStaNodes1.Get (nWifi - 1));
serverApps.Start (Seconds (1.0));
serverApps.Stop (Seconds (10.0));

UdpEchoClientHelper echoClient(staInterfaces.GetAddress(nWifi - 1), 9);
echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
echoClient.SetAttribute ("PacketSize", UintegerValue (1024));
//將 client 安裝在 WIFI 的到數第二個節點上
ApplicationContainer clientApps = echoClient.Install (wifiStaNodes2.Get (nWifi - 1));
clientApps.Start (Seconds (2.0));
clientApps.Stop (Seconds (10.0));

Ipv4GlobalRoutingHelper::PopulateRoutingTables ();
Simulator::Stop (Seconds (10.0));
/****************************************************************************/
```

Step 4 : Compile

```
andy@ubuntu:~/ns-allinone-3.26/ns-3.26$ ./waf --run scratch/109062578_project1
Waf: Entering directory `/home/andy/ns-allinone-3.26/ns-3.26/build'
Waf: Leaving directory `/home/andy/ns-allinone-3.26/ns-3.26/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (1.516s)
At time 2s client sent 1024 bytes to 10.1.2.3 port 9
At time 2.01095s server received 1024 bytes from 10.1.3.3 port 49153
At time 2.01095s server sent 1024 bytes to 10.1.3.3 port 49153
At time 2.02702s client received 1024 bytes from 10.1.2.3 port 9
```

2. Your observation :

- ## Use tcpdump -nn -tt -r to observe pcap

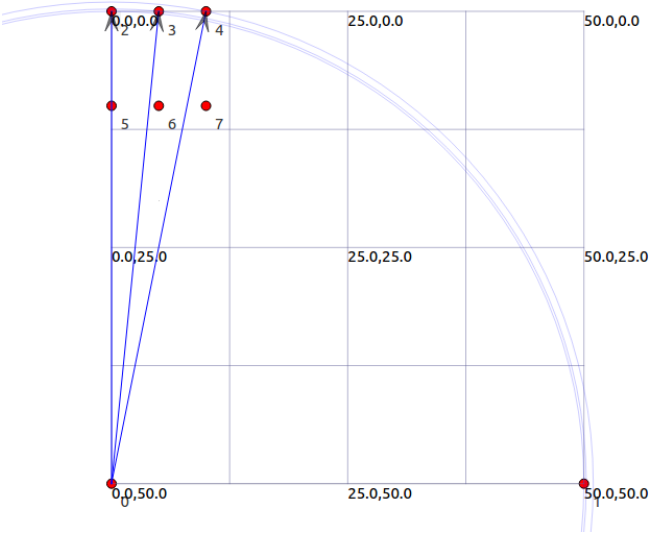
```
if (tracing == true){
	pointToPoint.EnablePcapAll ("P2P");
	phy1.EnablePcap ("AP1", apDevices1.Get (0));
	phy2.EnablePcap ("AP2", apDevices2.Get (0));
}
```

```
andy@ubuntu:~/ns-allinone-3.26/ns-3.26$ tcpdump -nn -tt -r P2P-1-0.pcap
reading from file P2P-1-0.pcap, link-type PPP (PPP)
2.002286 IP 10.1.3.3.49153 > 10.1.2.3.9: UDP, length 1024
2.021936 IP 10.1.2.3.9 > 10.1.3.3.49153: UDP, length 1024
andy@ubuntu:~/ns-allinone-3.26/ns-3.26$ tcpdump -nn -tt -r P2P-0-0.pcap
reading from file P2P-0-0.pcap, link-type PPP (PPP)
2.005972 IP 10.1.3.3.49153 > 10.1.2.3.9: UDP, length 1024
2.018249 IP 10.1.2.3.9 > 10.1.3.3.49153: UDP, length 1024
```

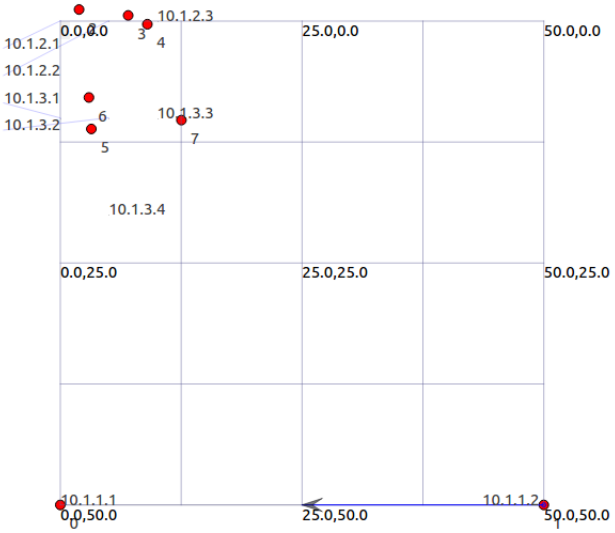
```
andy@ubuntu:~/ns-allinone-3.26/ns-3.26$ tcpdump -nn -tt -r AP1-0-1.pcap
reading from file AP1-0-1.pcap, link-type IEEE802_11 (802.11)
0.000025 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
0.000317 Assoc Request (ns-3-ssid) [6.0 9.0 12.0 18.0 24.0 36.0 48.0 54.0 Mbit]
0.000333 Acknowledgment RA:00:00:00:00:00:05
0.000438 Assoc Response AID(0) :: Successful
0.000582 Acknowledgment RA:00:00:00:00:00:09
0.000730 Assoc Request (ns-3-ssid) [6.0 9.0 12.0 18.0 24.0 36.0 48.0 54.0 Mbit]
0.000746 Acknowledgment RA:00:00:00:00:00:04
0.000948 Assoc Request (ns-3-ssid) [6.0 9.0 12.0 18.0 24.0 36.0 48.0 54.0 Mbit]
0.000964 Acknowledgment RA:00:00:00:00:00:03
0.001051 Assoc Response AID(0) :: Successful
0.001195 Acknowledgment RA:00:00:00:00:00:09
0.001337 Assoc Response AID(0) :: Successful
0.001481 Acknowledgment RA:00:00:00:00:00:09
0.102400 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
0.204800 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
0.307200 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
0.409600 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
0.512000 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
0.614400 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
0.716800 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
0.819200 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
0.921600 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
1.024000 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
1.126400 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
1.228800 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
1.331200 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
1.433600 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
1.536000 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
1.638400 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
1.740800 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
1.843200 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
1.945600 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
2.008972 ARP, Request who-has 10.1.2.3 (ff:ff:ff:ff:ff:ff) tell 10.1.2.4, length 32
2.009276 ARP, Reply 10.1.2.3 is-at 00:00:00:00:00:05, length 32
2.009292 Acknowledgment RA:00:00:00:00:00:05
2.009478 IP 10.1.3.3.49153 > 10.1.2.3.9: UDP, length 1024
2.011014 Acknowledgment RA:00:00:00:00:00:09
2.016066 ARP, Request who-has 10.1.2.4 (ff:ff:ff:ff:ff:ff) tell 10.1.2.3, length 32
2.016082 Acknowledgment RA:00:00:00:00:00:05
2.016232 ARP, Request who-has 10.1.2.4 (ff:ff:ff:ff:ff:ff) tell 10.1.2.3, length 32
2.016495 ARP, Reply 10.1.2.4 is-at 00:00:00:00:00:09, length 32
2.016667 Acknowledgment RA:00:00:00:00:00:09
2.018249 IP 10.1.2.3.9 > 10.1.3.3.49153: UDP, length 1024
2.018265 Acknowledgment RA:00:00:00:00:00:05
2.048000 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
2.150400 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
2.252800 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
2.355200 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
2.457600 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
2.560000 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
2.662400 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
2.764800 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
2.867200 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
2.969600 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
3.072000 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
3.174400 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
3.276800 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
3.379200 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
3.481600 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
3.584000 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
3.686400 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
3.788800 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
3.891200 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
3.993600 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
4.096000 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
4.198400 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
4.300800 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
4.403200 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] IBSS
```

- Observe NetAnim

First: AP send Beacon to Station



Two second later: Send Packet



3. Result :

At 2s client(10.1.3.3) send to server(10.1.2.3)

At 2.01095s server(10.1.2.3) send to client(10.1.3.3)

```
andy@ubuntu:~/ns-allinone-3.26/ns-3.26$ ./waf --run scratch/109062578_project1
Waf: Entering directory `/home/andy/ns-allinone-3.26/ns-3.26/build'
Waf: Leaving directory `/home/andy/ns-allinone-3.26/ns-3.26/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (1.678s)
At time 2s client sent 1024 bytes to 10.1.2.3 port 9
At time 2.01095s server received 1024 bytes from 10.1.3.3 port 49153
At time 2.01095s server sent 1024 bytes to 10.1.3.3 port 49153
At time 2.02702s client received 1024 bytes from 10.1.2.3 port 9
```

4. What you learn :

一開始 client 發封包給 server

server 都接收不到

後來發現是 STA 的 MAC address 用到之前宣告 AP MAC address

的版本

修改了一下之間的順序之後就能夠順利傳封包了