

Report

學號：111065510

一、Topology 說明

A. Topology 建立流程

建構 Topology，其實就如同我們現實世界中實際建立連線相同，需要有電腦、網卡、用來傳輸的線路、傳輸協議、應用程序等等的工具一起幫忙，因此使用 NS-3 建立 Topology 也相同，需要有 node（電腦）、channel（用來傳輸的線路）、Application（應用程序）、Net Device（網卡）、各種 Topology Helpers 等等的物件進行建構。

以下會詳細說明該 Project 的 Topology 建立流程：

- i. include 各種需要的模組，include 的檔案會放在 build 目錄下一個叫 ns3 的目錄當中。接著由於我們使用 C++ 進行編寫，因此使用”using”語句來把 ns3 namespace 引入到當前的全域作用域中。
- ii. 使用 NS_LOG_COMPONENT_DEFINE、LogComponentEnable 實現使用 log 進行紀錄。
- iii. 宣告各種基本物件模型以搭建 Topology：
 1. 使用 NodeContainer 建立 NS-3 節點。
 2. 使用 NetDeviceContainer 來建立 NetDevice。
 3. 使用 Ipv6InterfaceContainer 來建立 Ipv6Interface 將一個 IP 地址同一個網路裝置進行關聯，完成 IP 地址的配置，

4. 使用 **InternetStackHelper**，並透過 function “install”來在節點上安裝網路協定棧，它能夠為 NodeContainer 中的每一個節點安裝網路協議棧。
5. 使用 CsmHelper 來建立 NetDevice，以便後續能夠擷取封包進行觀察。
6. 使用 **Ipv6AddressHelper** 為節點上的裝置設定 IP 地址，能管理 IP 地址的分類。

在以上的步驟中，我們完成了建構 Topology 所需的基礎物件的宣告，並建立好兩個 AP 節點，且安裝好協議在其中，此外也幫 NetDevice 建立好通道。

- iv. 建構 WIFI 環境：由於我們需要模擬 AP 與 STA 使用 WIFI 的環境，因此我們需要使用額外的 NetDevice（網卡）：

WifiNetDevice，而要建立一個 WifiNetDevice 需要以下等等的配置。

1. 配置 WifiPhy：WifiPhy 為物理層的設備，與 WifiChannel 相連接，因此透過 **YansWifiPhyHelper** 類別去創建 YansWifiPhy 的實例，並且夠過 function “SetPcapDataLinkType”來透過 pcap 檔跟蹤記錄。
2. 配置 WifiChannel:用以連接 WifiPhy，讓資料在設備間進行傳

輸，因此使用 **YansWifiChannelHelper** 來建立 WifiChannel，
並讓 WifiPhy 能透過 function “SetChannel”將 WifiPhy 連接至
WifiChannel。

3. 配置 WifiMac:用以設定 WifiNetDevice 的 MAC，透過
WiFiMacHelper 中 function ”SetType” 來為 AP 與 STAs 設定
MAC 與 SSID。

4. 創建 WifiNetDevice: 我們透過 **WifiHelper** 來創建
WifiNetDevice，並將 WifiPhy 與 WifiMac 都安裝到此
WifiNetDevice 上。

5. 配置 mobility:透過 **MobilityHelper** 來設定節點 AP、STAs 的
初始位置 (SetPositionAllocator) 與移動方式
(SetMobilityModel)。

最後建立並安裝好後，將這些物件都放入剛剛所創建的 Container
中，以便等等使用。

v. 架設 Server/Client UDP Application：

1. Server 端：

我們透過使用 **UdpEchoServerHelper** 來建立 server 的 helper
物件，並透過其中的 function ”install” 來為我們之前建立的
節點（從 Container 中拿取）設定 UDP 的應用。

並透過 function “Start”, ”Stop”來設定生效與停止的時間。

2. Client 端

我們透過使用 **UdpEchoClientHelper** 來建立 helper 物件，其中需設定遠端地址為 Server 節點的 IP 地址，與所使用的 port。

並且還需要額外設定 **MaxPackets** 屬性，以告訴 Client 端我們能允許它在模擬期間所能傳送的最大資料；**Interval** 屬性，已告訴 Client 端在兩個資料包之間要等待多長時間；**PacketSize** 屬性，告訴 Client 端他的資料包應該有多少資料。

最後將這些設定安裝到 Client 節點上。

同時也須要設定生效與停止時間。

3. 注意事項

a、Server 與 Client 端所設定的 port 需一致。

b、Server 與 Client 端設定的停止時間最好一致。

vi. 擷取 CSMA 封包

透過一開始創建的 **CsmaHelper** 的 helper 類中的 function

“**EnablePcap**”來擷取 CSMA 封包。

vii. 透過產出.xml 檔案以使用 **NetAnim** 來查看模擬動畫

二、從.pcap 檔中的觀察

A. p1-0-1

Time	Source	Destination	Protocol	Length	Info
1 0.000000	::	ff02::1:ff00:7	ICMPv6	90	Neighbor Solicitation for 2001:1::200:ff:fe00:7 from 00:00:00:00:00:00
2 0.001000	::	ff02::1:ff00:7	ICMPv6	90	Neighbor Solicitation for fe80::200:ff:fe00:7 from 00:00:00:00:00:00
3 0.005000	::	ff02::1:ff00:3	ICMPv6	90	Neighbor Solicitation for 2001:1::200:ff:fe00:3 from 00:00:00:00:00:00
4 0.007000	::	ff02::1:ff00:3	ICMPv6	90	Neighbor Solicitation for fe80::200:ff:fe00:3 from 00:00:00:00:00:00
5 0.999000	fe80::200:ff:fe00:7	ff02::2	ICMPv6	74	Router Solicitation from 00:00:00:00:00:07
6 0.999661	fe80::200:ff:fe00:6	ff02::2	ICMPv6	74	Router Solicitation from 00:00:00:00:00:06
7 1.003182	fe80::200:ff:fe00:4	ff02::2	ICMPv6	74	Router Solicitation from 00:00:00:00:00:04
8 1.004182	fe80::200:ff:fe00:5	ff02::2	ICMPv6	74	Router Solicitation from 00:00:00:00:00:05
9 1.004182	fe80::200:ff:fe00:8	ff02::2	ICMPv6	74	Router Solicitation from 00:00:00:00:00:08
10 1.005182	fe80::200:ff:fe00:a	ff02::2	ICMPv6	74	Router Solicitation from 00:00:00:00:00:0a
11 1.006000	fe80::200:ff:fe00:3	ff02::2	ICMPv6	74	Router Solicitation from 00:00:00:00:00:03
12 1.007182	fe80::200:ff:fe00:9	ff02::2	ICMPv6	74	Router Solicitation from 00:00:00:00:00:09
13 2.000202	fe80::200:ff:fe00:8	ff02::1:ff00:4	ICMPv6	90	Neighbor Solicitation for fe80::200:ff:fe00:4 from 00:00:00:00:00:00
14 2.000606	fe80::200:ff:fe00:4	fe80::200:ff:fe00:8	ICMPv6	90	Neighbor Advertisement fe80::200:ff:fe00:4 (sol, ovr) is at 00:00:00:00:00:00
15 2.005544	fe80::200:ff:fe00:4	fe80::200:ff:fe00:8	UDP	1090	1000 → 49153 Len=1024 [ILLEGAL CHECKSUM (0)]
16 3.002813	fe80::200:ff:fe00:4	fe80::200:ff:fe00:8	UDP	1090	1000 → 49153 Len=1024 [ILLEGAL CHECKSUM (0)]
17 4.002676	fe80::200:ff:fe00:4	fe80::200:ff:fe00:8	UDP	1090	1000 → 49153 Len=1024 [ILLEGAL CHECKSUM (0)]
18 4.007202	fe80::200:ff:fe00:9	ff02::1:ff00:5	ICMPv6	90	Neighbor Solicitation for fe80::200:ff:fe00:5 from 00:00:00:00:00:00
19 4.007606	fe80::200:ff:fe00:5	fe80::200:ff:fe00:9	ICMPv6	90	Neighbor Advertisement fe80::200:ff:fe00:5 (sol, ovr) is at 00:00:00:00:00:00
20 4.012544	fe80::200:ff:fe00:5	fe80::200:ff:fe00:9	FIN	1090	1001 → 49153 Len=1024 [ILLEGAL CHECKSUM (0)], Unknown (0x0000)
21 5.002676	fe80::200:ff:fe00:4	fe80::200:ff:fe00:8	UDP	1090	1000 → 49153 Len=1024 [ILLEGAL CHECKSUM (0)]
22 5.005962	fe80::200:ff:fe00:5	fe80::200:ff:fe00:9	FIN	1090	1001 → 49153 Len=1024 [ILLEGAL CHECKSUM (0)], Unknown (0x0000)
23 6.002676	fe80::200:ff:fe00:4	fe80::200:ff:fe00:8	UDP	1090	1000 → 49153 Len=1024 [ILLEGAL CHECKSUM (0)]
24 6.005971	fe80::200:ff:fe00:5	fe80::200:ff:fe00:9	FIN	1090	1001 → 49153 Len=1024 [ILLEGAL CHECKSUM (0)], Unknown (0x0000)
25 6.007899	fe80::200:ff:fe00:a	ff02::1:ff00:6	ICMPv6	90	Neighbor Solicitation for fe80::200:ff:fe00:6 from 00:00:00:00:00:00
26 6.008303	fe80::200:ff:fe00:6	fe80::200:ff:fe00:a	ICMPv6	90	Neighbor Advertisement fe80::200:ff:fe00:6 (sol, ovr) is at 00:00:00:00:00:00
27 6.013241	fe80::200:ff:fe00:6	fe80::200:ff:fe00:a	UDP	1090	1002 → 49153 Len=1024 [ILLEGAL CHECKSUM (0)]
28 7.001376	fe80::200:ff:fe00:4	fe80::200:ff:fe00:8	ICMPv6	90	Neighbor Solicitation for fe80::200:ff:fe00:8 from 00:00:00:00:00:00
29 7.004581	fe80::200:ff:fe00:5	fe80::200:ff:fe00:9	FIN	1090	1001 → 49153 Len=1024 [ILLEGAL CHECKSUM (0)], Unknown (0x0000)
30 7.008102	fe80::200:ff:fe00:6	fe80::200:ff:fe00:a	UDP	1090	1002 → 49153 Len=1024 [ILLEGAL CHECKSUM (0)]
31 7.009763	fe80::200:ff:fe00:4	fe80::200:ff:fe00:8	UDP	1090	1000 → 49153 Len=1024 [ILLEGAL CHECKSUM (0)]
32 8.001684	fe80::200:ff:fe00:4	fe80::200:ff:fe00:8	UDP	1090	1000 → 49153 Len=1024 [ILLEGAL CHECKSUM (0)]
33 8.004988	fe80::200:ff:fe00:5	fe80::200:ff:fe00:9	FIN	1090	1001 → 49153 Len=1024 [ILLEGAL CHECKSUM (0)], Unknown (0x0000)
34 8.008229	fe80::200:ff:fe00:6	fe80::200:ff:fe00:a	UDP	1090	1002 → 49153 Len=1024 [ILLEGAL CHECKSUM (0)]
35 9.001684	fe80::200:ff:fe00:4	fe80::200:ff:fe00:8	UDP	1090	1000 → 49153 Len=1024 [ILLEGAL CHECKSUM (0)]
36 9.004889	fe80::200:ff:fe00:6	fe80::200:ff:fe00:a	UDP	1090	1002 → 49153 Len=1024 [ILLEGAL CHECKSUM (0)]
37 9.008121	fe80::200:ff:fe00:5	fe80::200:ff:fe00:9	FIN	1090	1001 → 49153 Len=1024 [ILLEGAL CHECKSUM (0)], Unknown (0x0000)
38 9.008464	fe80::200:ff:fe00:5	fe80::200:ff:fe00:9	ICMPv6	90	Neighbor Solicitation for fe80::200:ff:fe00:9 from 00:00:00:00:00:00
39 10.003402	fe80::200:ff:fe00:5	fe80::200:ff:fe00:9	FIN	1090	1001 → 49153 Len=1024 [ILLEGAL CHECKSUM (0)], Unknown (0x0000)

i. Neighbor Solicitation：可以看到最一開始前四個封包、十三、十

八等等的封包在 info 的欄位都顯示此訊息，此訊息代表的是原節

點 (ex. Node 1) 希望去確定使用同一 link 的其他 node 的地址，

而發出的訊息。

ii. Neighbor Advertisement：可以看出在第十四、十九、二十六的封

包所帶的資訊有所提到，這代表的是當其他節點 (ex. Node 2) 收

到某節點(Node 1)所發出的 Neighbor Solicitation，就會回傳此類型

的封包，而當原節點(Node 1)收到此訊息後，就代表 Node1、

Node2 可以開始交換封包了。因此我們可以發現在一開始有發出

帶有許多 Neighbor Solicitation 訊息的封包，但因為其他節點一直沒有回傳帶 Neighbor Advertisement 的訊息，因此 UDP 封包遲遲無法傳送，一直到第二秒時，也就是第十三、十四個封包完整了傳了一對，地址結尾為四與八的節點才能進行封包交換，因此第十五到第十七個封包才能使用 UDP 進行封包的傳送，後續的也是以此類推。

iii. Router Solicitation: 此訊息代表的是，當系統在啟動時，連結上的節點會向其他的路由器去播送地址，讓節點能夠立即自動配置其 Ipv6 的地址。因此可以看到從第五到第十二個封包（共八個節點）皆是由不同的地址進行此訊息的傳送。而因為其目的就是在配置其位置，因此可以看到其他訊息如 Neighbor Solicitation，也是在此訊息傳送後才能正常傳遞。

iv. UDP: 而因為程式中我們是使用 UDP 當作應用程序，因此這邊便是透過 UDP 協議進行傳送，我們可以看到藍色底的皆是透過 UDP 進行傳送，而因為 UDP 不像是 TCP 需要三次握手跟確保資料不會掉，UDP 就是只要一直傳送就好。

B. pl-1-1

1	0.000000	::	ff02::1:ff00:7	ICMPv6	90	Neighbor Solicitation	for 2001:1::200:ff:fe00:7 from 00:00:00:00:00:00
2	0.001000	::	ff02::1:ff00:7	ICMPv6	90	Neighbor Solicitation	for fe80::200:ff:fe00:7 from 00:00:00:00:00:00
3	0.005000	::	ff02::1:ff00:3	ICMPv6	90	Neighbor Solicitation	for 2001:1::200:ff:fe00:3 from 00:00:00:00:00:00
4	0.007000	::	ff02::1:ff00:3	ICMPv6	90	Neighbor Solicitation	for fe80::200:ff:fe00:3 from 00:00:00:00:00:00
5	0.999000	fe80::200:ff:fe00:7	ff02::2	ICMPv6	74	Router Solicitation	from 00:00:00:00:00:07
6	0.999661	fe80::200:ff:fe00:6	ff02::2	ICMPv6	74	Router Solicitation	from 00:00:00:00:00:06
7	1.003182	fe80::200:ff:fe00:4	ff02::2	ICMPv6	74	Router Solicitation	from 00:00:00:00:00:04
8	1.004182	fe80::200:ff:fe00:8	ff02::2	ICMPv6	74	Router Solicitation	from 00:00:00:00:00:08
9	1.004183	fe80::200:ff:fe00:5	ff02::2	ICMPv6	74	Router Solicitation	from 00:00:00:00:00:05
10	1.005182	fe80::200:ff:fe00:a	ff02::2	ICMPv6	74	Router Solicitation	from 00:00:00:00:00:0a
11	1.006000	fe80::200:ff:fe00:3	ff02::2	ICMPv6	74	Router Solicitation	from 00:00:00:00:00:03
12	1.007182	fe80::200:ff:fe00:9	ff02::2	ICMPv6	74	Router Solicitation	from 00:00:00:00:00:09
13	2.000202	fe80::200:ff:fe00:8	ff02::1:ff00:4	ICMPv6	90	Neighbor Solicitation	for fe80::200:ff:fe00:4 from 00:00:00:00:00:00
14	2.002406	fe80::200:ff:fe00:8	fe80::200:ff:fe00:4	UDP	1090	49153 → 1000 Len=1024	[ILLEGAL CHECKSUM (0)]
15	2.999538	fe80::200:ff:fe00:8	fe80::200:ff:fe00:4	UDP	1090	49153 → 1000 Len=1024	[ILLEGAL CHECKSUM (0)]
16	3.999538	fe80::200:ff:fe00:8	fe80::200:ff:fe00:4	UDP	1090	49153 → 1000 Len=1024	[ILLEGAL CHECKSUM (0)]
17	4.007202	fe80::200:ff:fe00:9	ff02::1:ff00:5	ICMPv6	90	Neighbor Solicitation	for fe80::200:ff:fe00:5 from 00:00:00:00:00:00
18	4.009406	fe80::200:ff:fe00:9	fe80::200:ff:fe00:5	FIND	1090	49153 → 1001 Len=1024	[ILLEGAL CHECKSUM (0)], Unknown (0x0000)
19	4.999538	fe80::200:ff:fe00:8	fe80::200:ff:fe00:4	UDP	1090	49153 → 1000 Len=1024	[ILLEGAL CHECKSUM (0)]
20	5.001208	fe80::200:ff:fe00:9	fe80::200:ff:fe00:5	FIND	1090	49153 → 1001 Len=1024	[ILLEGAL CHECKSUM (0)], Unknown (0x0000)
21	5.999538	fe80::200:ff:fe00:8	fe80::200:ff:fe00:4	UDP	1090	49153 → 1000 Len=1024	[ILLEGAL CHECKSUM (0)]
22	6.001307	fe80::200:ff:fe00:9	fe80::200:ff:fe00:5	FIND	1090	49153 → 1001 Len=1024	[ILLEGAL CHECKSUM (0)], Unknown (0x0000)
23	6.007899	fe80::200:ff:fe00:a	ff02::1:ff00:6	ICMPv6	90	Neighbor Solicitation	for fe80::200:ff:fe00:6 from 00:00:00:00:00:00
24	6.010103	fe80::200:ff:fe00:a	fe80::200:ff:fe00:6	UDP	1090	49153 → 1002 Len=1024	[ILLEGAL CHECKSUM (0)]
25	6.999538	fe80::200:ff:fe00:8	fe80::200:ff:fe00:4	UDP	1090	49153 → 1000 Len=1024	[ILLEGAL CHECKSUM (0)]
26	7.001199	fe80::200:ff:fe00:9	fe80::200:ff:fe00:5	FIND	1090	49153 → 1001 Len=1024	[ILLEGAL CHECKSUM (0)], Unknown (0x0000)
27	7.001840	fe80::200:ff:fe00:8	fe80::200:ff:fe00:4	ICMPv6	90	Neighbor Advertisement	fe80::200:ff:fe00:8 (sol, ovr) is at 0
28	7.003537	fe80::200:ff:fe00:a	fe80::200:ff:fe00:6	UDP	1090	49153 → 1002 Len=1024	[ILLEGAL CHECKSUM (0)]
29	7.999042	fe80::200:ff:fe00:8	fe80::200:ff:fe00:4	UDP	1090	49153 → 1000 Len=1024	[ILLEGAL CHECKSUM (0)]
30	8.001382	fe80::200:ff:fe00:9	fe80::200:ff:fe00:5	FIND	1090	49153 → 1001 Len=1024	[ILLEGAL CHECKSUM (0)], Unknown (0x0000)
31	8.002998	fe80::200:ff:fe00:a	fe80::200:ff:fe00:6	UDP	1090	49153 → 1002 Len=1024	[ILLEGAL CHECKSUM (0)]
32	8.999042	fe80::200:ff:fe00:8	fe80::200:ff:fe00:4	UDP	1090	49153 → 1000 Len=1024	[ILLEGAL CHECKSUM (0)]
33	9.001247	fe80::200:ff:fe00:a	fe80::200:ff:fe00:6	UDP	1090	49153 → 1002 Len=1024	[ILLEGAL CHECKSUM (0)]
34	9.002863	fe80::200:ff:fe00:9	fe80::200:ff:fe00:5	FIND	1090	49153 → 1001 Len=1024	[ILLEGAL CHECKSUM (0)], Unknown (0x0000)
35	9.010209	fe80::200:ff:fe00:9	fe80::200:ff:fe00:5	ICMPv6	90	Neighbor Advertisement	fe80::200:ff:fe00:9 (sol, ovr) is at 0
36	10.000760	fe80::200:ff:fe00:9	fe80::200:ff:fe00:5	FIND	1090	49153 → 1001 Len=1024	[ILLEGAL CHECKSUM (0)], Unknown (0x0000)
37	10.002412	fe80::200:ff:fe00:a	fe80::200:ff:fe00:6	UDP	1090	49153 → 1002 Len=1024	[ILLEGAL CHECKSUM (0)]
38	11.000598	fe80::200:ff:fe00:9	fe80::200:ff:fe00:5	FIND	1090	49153 → 1001 Len=1024	[ILLEGAL CHECKSUM (0)], Unknown (0x0000)
39	11.002376	fe80::200:ff:fe00:a	fe80::200:ff:fe00:6	UDP	1090	49153 → 1002 Len=1024	[ILLEGAL CHECKSUM (0)]

可以看出基本與剛剛敘述的大致相同，只是在傳送時 source 與 dist 會

相反過來，因此不多加贅述。

三、animation 觀察協定傳輸方式

UDP 不同於 TCP 僅限於一對一的傳送方式，通常當我們需要進行多點傳送

(Multicast)與廣播傳送(Broadcast)時等一對多的傳送方式，便會傾向於使用

UDP，因此在動畫中我們也可以看出 UDP 的此項特性，我們可以看到每當

STA 透過 AP 向對方的 AP 發出訊息後，對方 AP 都會向他的 STAs 一併傳

送。