Report

學號：111065510

1. Topology說明
   1. 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牤，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：

* + 1. 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．
    2. 使用NS\_LOG\_COMPONENT\_DEFINE、LogComponentEnable實現使用log進行紀錄．
    3. 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CsmaHelper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．

* + 1. 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．

* + 1. 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．

* + - 1. 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．

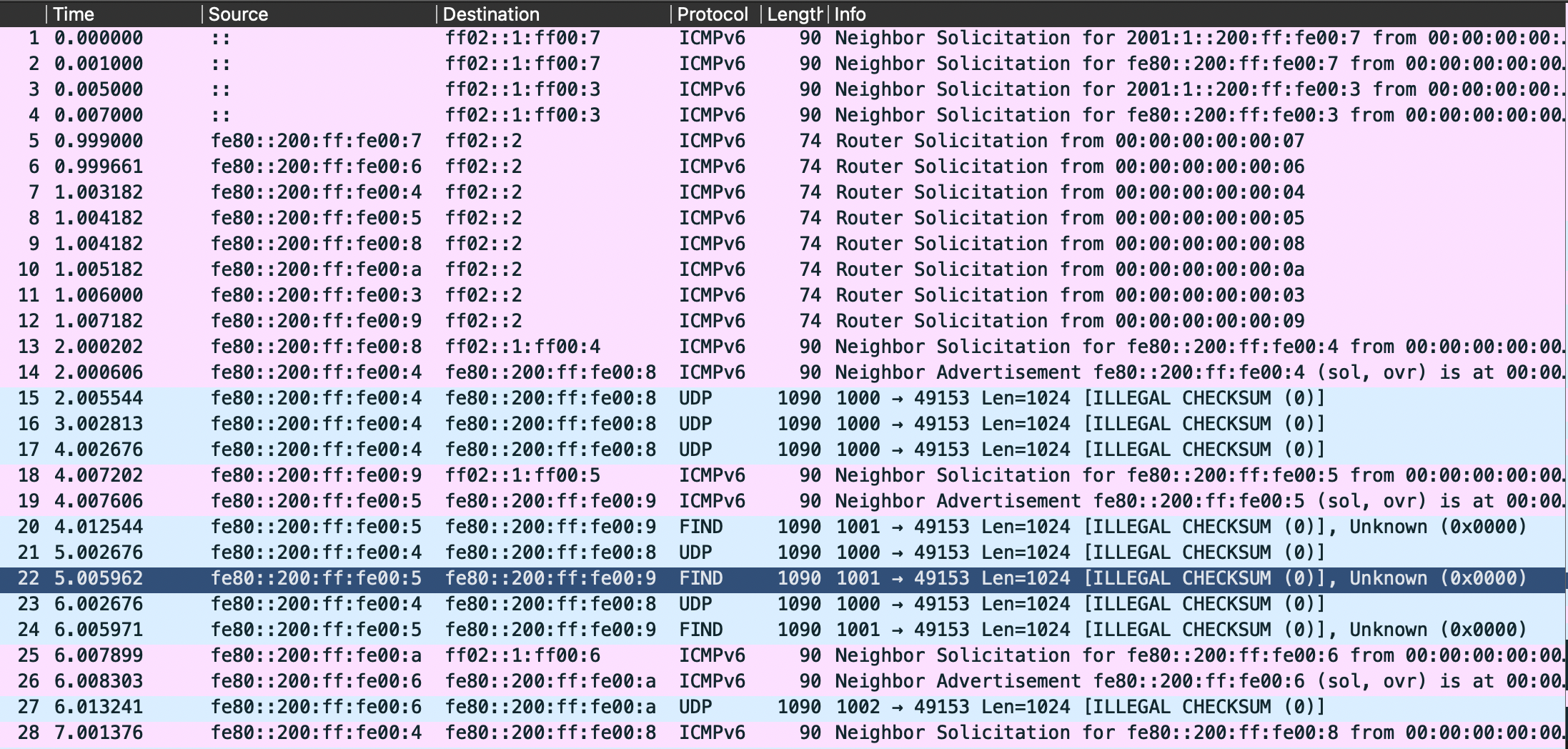
同時也須要設定生效與停止時間．

* + - 1. 注意事項
         1. Server與Client端所設定的port需一致．
         2. Server與Client端設定的停止時間最好一致．
    1. 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．

* + 1. 透過產出.xml 檔案以使用**NetAnim**來查看模擬動畫

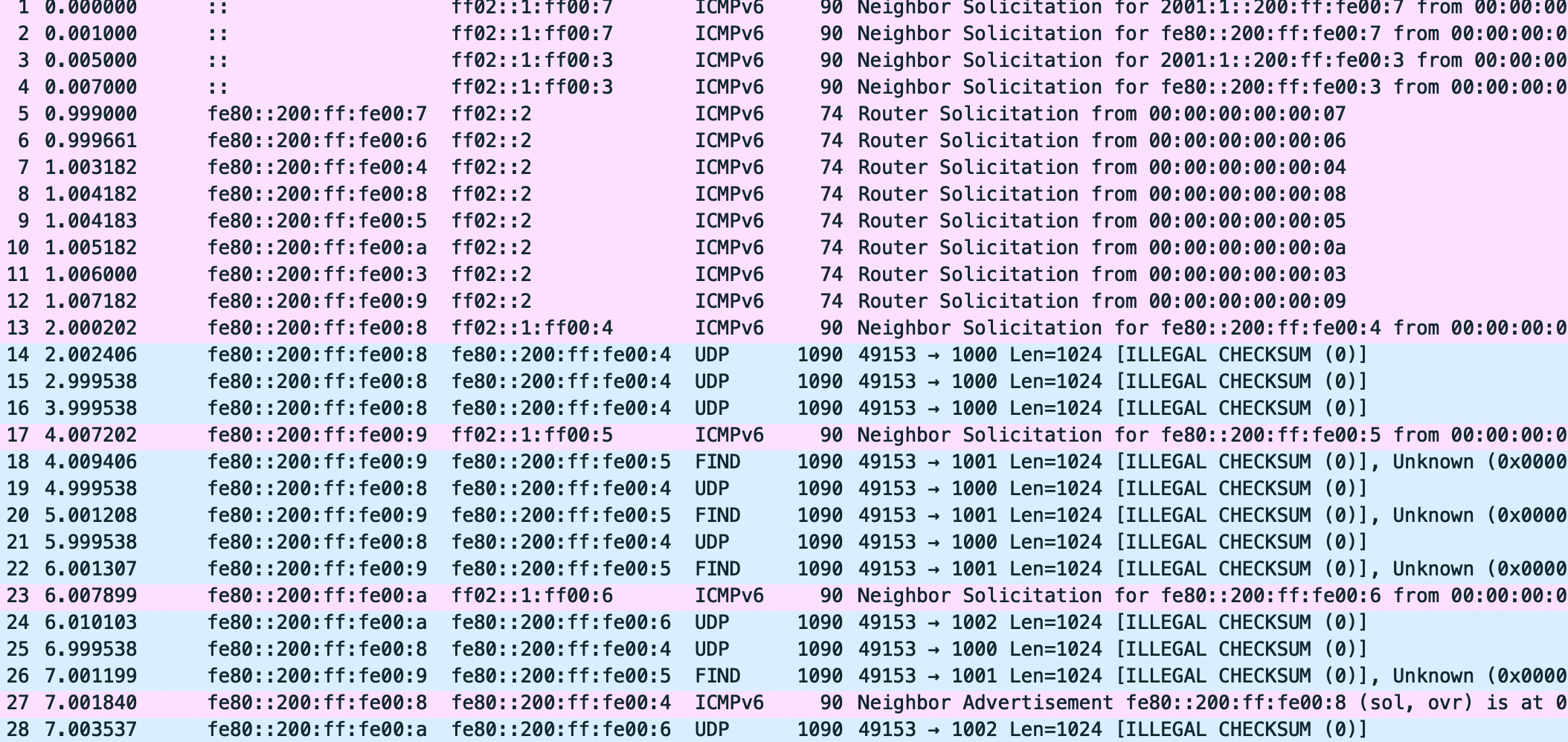
1. 從.pcap檔中的觀察
   1. p1-0-1



一張含有 桌 的圖片

自動產生的描述

* + 1. 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．
    2. 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．
    3. 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．
    4. 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．
  1. p1-1-1



一張含有 桌 的圖片

自動產生的描述

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．

1. 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．