

1

HỆ THỐNG NHÚNG
MẠNG KHÔNG DÂY

Tìm hiểu WLAN 802.11

Biên soạn: Phan Trung Phát
Liên hệ: phatpt@uit.edu.vn

Lưu hành nội bộ

A. TỔNG QUAN

1. Mục tiêu

- Hiểu về cách vận hành của IEEE MAC 802.11 và cấu trúc gói tin.
- Tìm hiểu cách cài đặt NS-3 và xây dựng các kịch bản mô phỏng mạng máy tính đơn giản trên NS-3.
- Phân tích, tìm hiểu cách vận hành của IEEE 802.11 trong thực tế.

2. Môi trường, thiết bị thực hành

- Máy ảo Ubuntu 20.04 đã cài đặt NS-3.
- Phần mềm Wireshark.
- USB WiFi, card mạng rời.

B. GIỚI THIỆU NS-3 SIMULATOR

NS-3 simulator là một chương trình mô phỏng mã nguồn mở, hỗ trợ mô phỏng các sự kiện truyền thông rời rạc (*discrete event simulation*) của nhiều giao thức mạng khác nhau như WLAN, AODV, OSLR,... NS-3 được viết bằng ngôn ngữ C++, tuy nhiên người sử dụng chỉ cần nắm ngôn ngữ C++ ở mức cơ bản để có thể hiểu và xây dựng các kịch bản mô phỏng mong muốn.

Một số khái niệm cần quan tâm trong NS-3:

1. Node

Trong NS-3, chúng ta sử dụng một khái niệm chung thường được sử dụng bởi các trình mô phỏng mạng khác là Node để biểu diễn cho một thiết bị tính toán cơ bản. Khái niệm được biểu diễn bằng lớp Node của ngôn ngữ C++ trong NS-3. Có thể xem một Node như một máy tính mà chúng ta có thể thêm các chức năng cho nó bằng việc cài thêm ứng dụng dụng mạng, thêm các tầng giao thức, gắn thêm các card WiFi, Bluetooth với các driver tương thích.

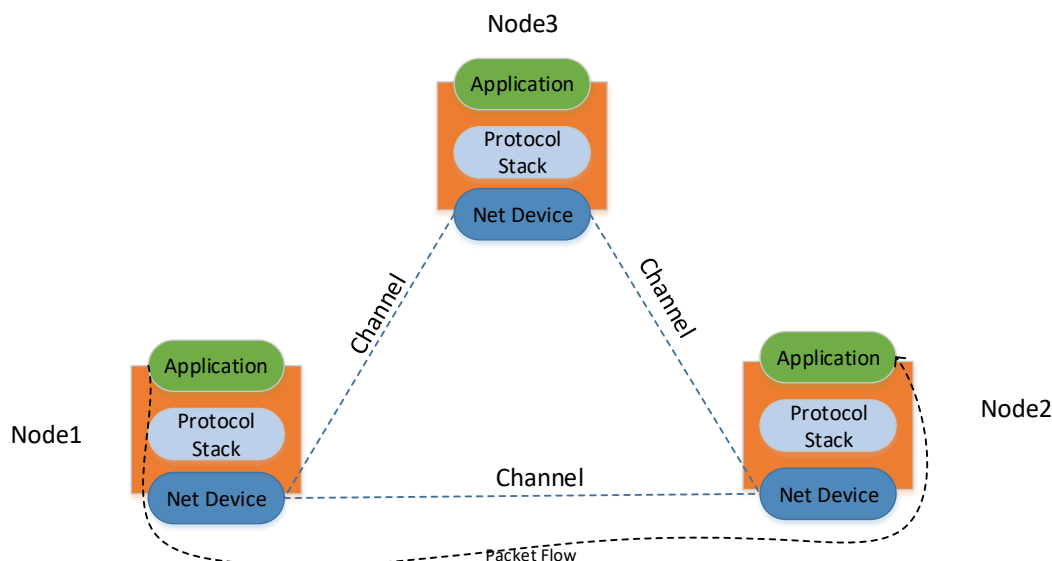
2. Application

Trong NS-3, chúng ta không cần quan tâm đến các Node sử dụng hệ điều hành gì hoặc cấu trúc lệnh hệ thống để lập trình mô phỏng giao thức mạng mà mình mong muốn. Tuy nhiên, trong NS-3, vẫn tồn tại khái niệm Application được biểu diễn với lớp cùng tên trong C++. Lớp này cung cấp các phương thức để hỗ trợ xây dựng được những tính năng mô phỏng ở tầng ứng dụng mong muốn (Ví dụ: FTP, HTTP,...). Các nhà phát triển có thể mở rộng lớp Application để tạo ra ứng dụng cụ thể hơn phục vụ cho việc giải lập.

3. Channel

Trong thực tế, để kết nối một máy tính vào mạng, thông thường dữ liệu truyền dẫn qua mạng thông qua một kênh truyền – Channel. Ví dụ khi chúng ta kết nối cáp Ethernet vào ổ cắm trên tường, chúng ta đang kết nối vào một kênh giao tiếp Ethernet. Trong trình mô phỏng NS-3, chúng ta sẽ kết nối Node tới một đối tượng biểu diễn cho kênh truyền – Channel. Lớp Channel cung cấp phương thức để quản lý, lập trình những cách thức truyền thông thông giữa các Node. Những lớp Channel cụ thể có thể đặc tả được những mô hình phức tạp như Ethernet Switch, hoặc môi trường truyền dẫn mạng không dây trong thực tế (không gian 3 chiều).

Một số lớp thông dụng mô tả các kênh truyền thông trong thực tế như *CsmaChannel*, *PointToPointChannel* và *WifiChannel*. Ví dụ *CsmaChannel*, mô hình một phiên bản của môi trường mạng không dây mà chúng ta có thể can thiệp vào điều chỉnh các thông số liên quan đến tính năng Carrier Sense Multiple Access / Collision Avoidance phục vụ cho mô phỏng.



Hình 1. Mô tả các khái niệm trong NS-3

4. NetDevice

Trong NS-3, *NetDevice* là một mô hình dùng để mô phỏng cho các card mạng (NIC) mô phỏng cả thông số card và software driver. Một *NetDevice* được cài đặt lên một Node để nó có thể truyền thông được với Node khác trong môi trường mô phỏng thông qua các kênh (Channel).

NS-3 cung cấp lớp *NetDevice* với các phương thức để quản lý các kết nối đến các đối tượng Node và Channel; và được đặc tả cụ thể hơn bởi nhà phát triển trong khi lập trình. Trong nội dung thực hành, chúng ta sẽ sử dụng nhiều phiên bản khác nhau của *NetDevice* như *CsmaNetDevice*, *PointToPointNetDevice* và *WifiNetDevice*. Lưu ý: *NetDevice* nào thì sử dụng Channel tương thích, Ethernet NIC được thiết kế để sử dụng với Ethernet Network, *CsmaNetDevice* làm việc với *CsmaChannel*, *PointToPointNetDevice* thì làm việc với *PointToPointChannel* và *WifiNetDevice* tương thích với *WifiChannel*.

5. Topology Helpers

Trong NS-3, chúng ta sẽ thấy Node sẽ gắn liền với *NetDevice*. Trong những mô phỏng mạng lớn, chúng ta cần sắp đặt rất nhiều kết nối giữa các Node, *NetDevice* và Channel.

Khi kết nối NetDevice với Nodes, NetDevices với Channels, gán địa chỉ IP,..., là một trong những tác vụ phổ biến trong NS-3, topology helpers được cung cấp để đơn giản hóa các công việc trên. Ví dụ, chúng ta có thể mất nhiều thao tác NS-3 cơ bản như vừa nêu để tạo một NetDevice, thêm một MAC address, cài đặt NetDevice lên Node, cấu hình các tầng giao thức rồi kết nối NetDevice tới một Channel. Và còn cần nhiều thời gian hơn để kết nối những thiết bị này tới nhiều điểm hoặc kết nối các mạng này lại với nhau, hoặc tạo thành một internetwork. NS-3 cung cấp các lớp topology helper như *NodeContainer*, *NetDeviceContainer*, *PointToPointHelper* hay *InternetStackHelper* giúp thực hiện các tác vụ trên một cách nhanh chóng và tiện lợi hơn.

C. THỰC HÀNH

1. Cài đặt NS-3

Cập nhật lại các Ubuntu Repo và các ứng dụng hiện có:

```
$ cd ~  
$ sudo apt update  
$ sudo apt -y upgrade
```

Cài đặt các Core Dependencies để build và biên dịch:

```
$ sudo apt install build-essential libsqlite3-dev libboost-all-dev libssl-dev git python3-  
setuptools castxml
```

Cài đặt các Dependencies cho các liên kết NS-3 Python:

```
$ sudo apt install gir1.2-goocanvas-2.0 gir1.2-gtk-3.0 libgirepository1.0-dev python3-  
dev python3-gi python3-gi-cairo python3-pip python3-pygraphviz python3-pygccxml
```

Cài đặt các Dependencies cho NS-3:

```
$ sudo apt install g++ pkg-config sqlite3 qt5-default mercurial ipython3 openmpi-bin  
openmpi-common openmpi-doc libopenmpi-dev autoconf cvs bzip2 unrar gdb valgrind  
uncrustify doxygen graphviz imagemagick python3-sphinx dia tcpdump libxml2  
libxml2-dev cmake libc6-dev libc6-dev-i386 libclang-6.0-dev llvm-6.0-dev automake
```

Cài đặt các gói Python bổ sung:

```
$ sudo su  
  
$ cd  
  
$ pip3 install kiwi  
  
$ exit  
  
$ cd
```

Tải và cài đặt NS-3:

```
$ wget -c https://www.nsnam.org/releases/ns-allinone-3.33.tar.bz2  
  
# extract tar.bz2  
$ tar -xvjf ns-allinone-3.33.tar.bz2  
  
# go back to home folder  
$ cd  
  
# navigate to ns-3 directory (not the NS-3 all in one)  
$ cd ns-allinone-3.33/ns-3.33/  
  
# Configure the installation  
$ ./waf configure --enable-examples  
  
# Build ns-3 installation  
$ ./waf  
  
# to check whether installation was a success  
$ ./waf --run hello-simulator
```

```
phatpt@ubuntu: ~/ns-allinone-3.33/ns-3.33
phatpt@ubuntu:~/ns-allinone-3.33/ns-3.33$ ./waf --run hello-simulator
Waf: Entering directory `/home/phatpt/ns-allinone-3.33/ns-3.33/build'
Waf: Leaving directory `/home/phatpt/ns-allinone-3.33/ns-3.33/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.716s)
Hello Simulator
phatpt@ubuntu:~/ns-allinone-3.33/ns-3.33$
```

2. Chạy một kịch bản mô phỏng mẫu đơn giản

Chuẩn bị code và tiến hành build kịch bản:

```
$ cd ns-3.33
$ cp examples/tutorial/first.cc scratch/myfirst.cc
$ ./waf
```

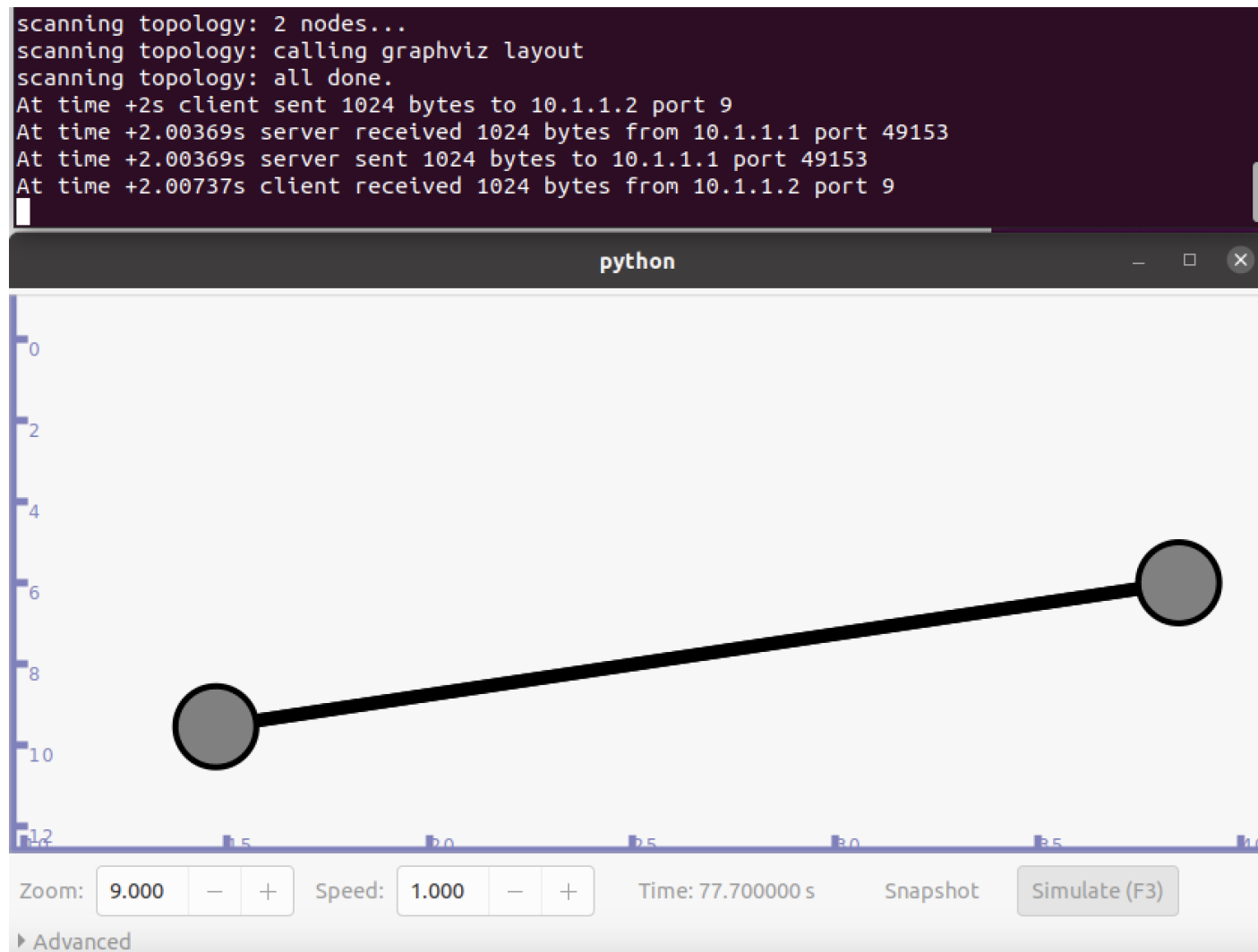
```
phatpt@ubuntu:~/ns-allinone-3.33/ns-3.33$ ./waf
Waf: Entering directory `/home/phatpt/ns-allinone-3.33/ns-3.33/build'
[2530/2603] Compiling scratch/myfirst.cc
[2561/2603] Compiling scratch/subdir/scratch-simulator-subdir.cc
[2562/2603] Linking build/scratch/subdir/subdir
[2563/2603] Linking build/scratch/myfirst
Waf: Leaving directory `/home/phatpt/ns-allinone-3.33/ns-3.33/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (3.787s)

Modules built:
antenna          aodv             applications
bridge           buildings        config-store
core             csma             csma-layout
dsdv             dsr              energy
fd-net-device    flow-monitor     internet
internet-apps    lr-wpan          lte
mesh             mobility         netanim
network          nix-vector-routing
point-to-point   point-to-point-layout
sixlowpan        spectrum         stats
tap-bridge       test (no Python) topology-read
traffic-control  uan             virtual-net-device
visualizer       wave            wifi
wimax

Modules not built (see ns-3 tutorial for explanation):
brite            click            dpdk-net-device
mpi              openflow
```

Chạy đoạn kịch bản vừa biên dịch ở trên:

```
$ ./waf --run scratch/myfirst --vis  
# --vis to show visualization.
```



3. Thu thập các WLAN frames thực tế

Sử dụng thêm card mạng rồi kết nối vào Ubuntu, chuyển sang chế độ monitor để tiến hành bắt các WLAN frames trong thực tế.

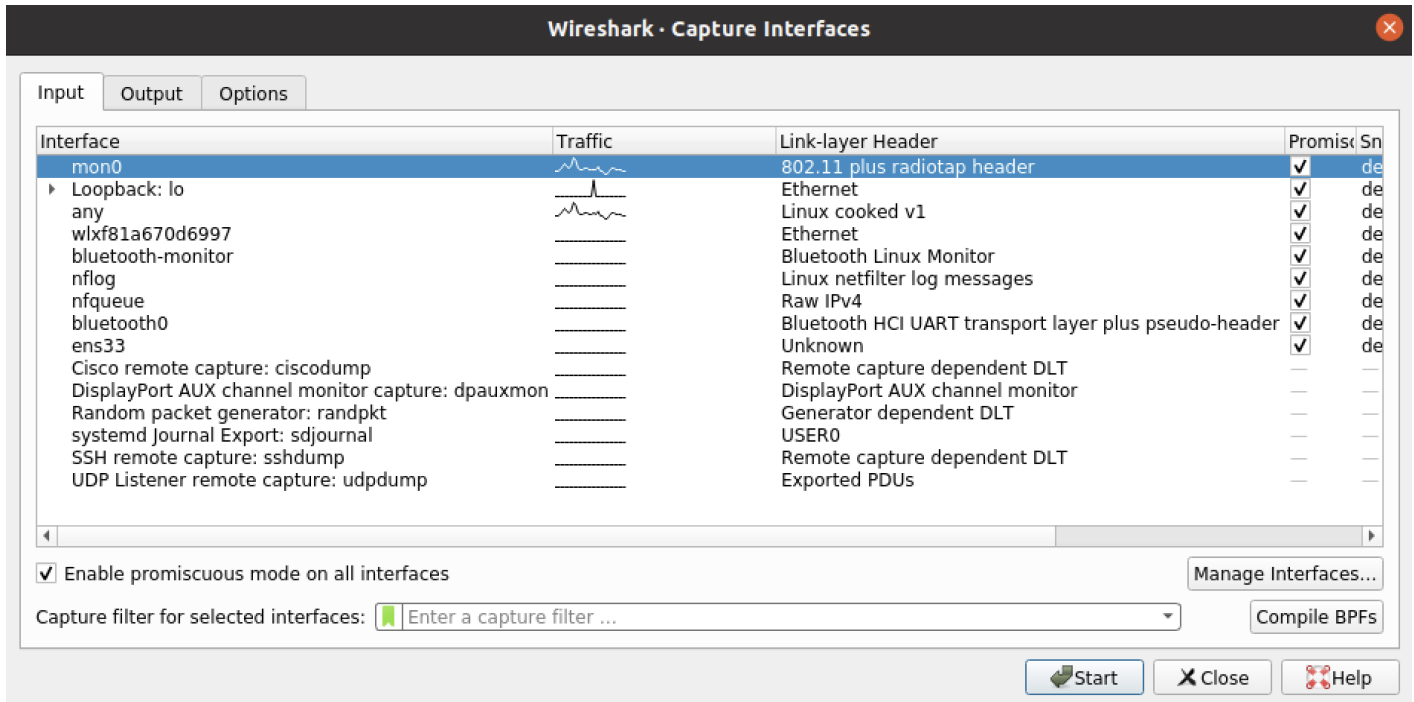
Sử dụng các câu lệnh sau để chuyển card mạng trên sang chế độ monitor:

```
$ iw dev <interface name> interface add <interface monitor> type monitor
```

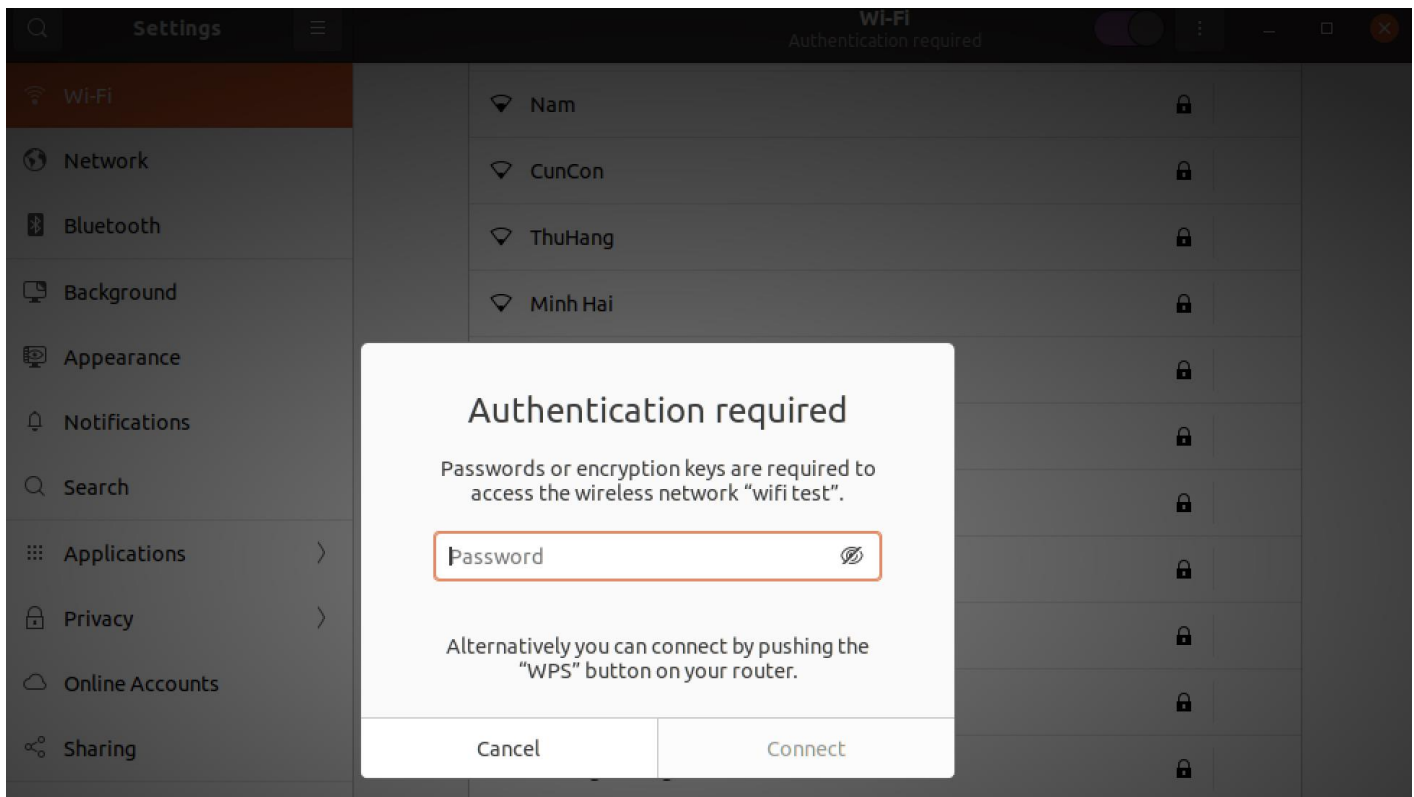


```
$ ip link set <interface monitor> up
```

Sử dụng Wireshark chọn card mạng (Capture → Options) đã chuyển sang chế độ monitor để tiến hành bắt gói tin:



Mở tùy chỉnh Wi-Fi Setting trên Ubuntu và tiến hành đăng nhập vào WiFi mục tiêu:



Kết thúc quá trình bắt gói tin và lưu lại file với tên theo định dạng **NhomXX-Lab1-task2.pcapng**

D. YÊU CẦU & ĐÁNH GIÁ

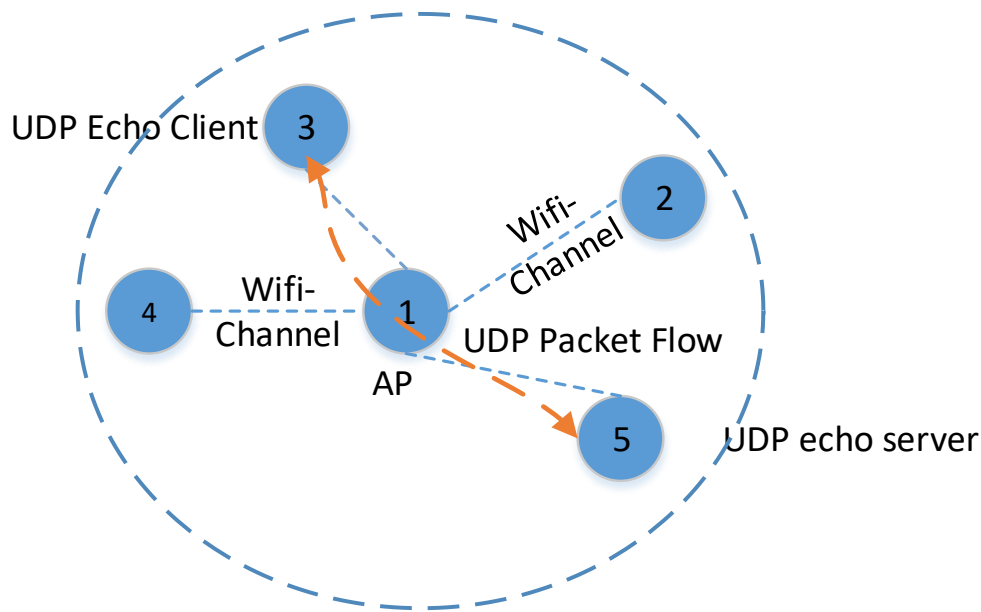
1. Yêu cầu

Đối với nội dung thực hiện giả lập trên NS-3, sinh viên thực hiện các yêu cầu sau:

1. Tìm hiểu nội dung code được cung cấp, trình bày ý tưởng của kịch bản, các bước cài đặt trong code để xây dựng nên kịch bản.

Lưu ý: trình bày các thành phần trọng tâm để xây dựng nên kịch bản.

Gợi ý:



Hình 2. Gợi ý mô hình task 1

2. Tìm hiểu, trình bày và thực hiện log ra 3 file .pcap của 3 node: UDP Client, UDP Server và AP. Lưu lại với tên lần lượt theo định dạng **lab1-task1-client.pcap**, **lab1-task1-server.pcap**, **lab1-task1-ap.pcap** và file sau khi đã chỉnh sửa dưới định dạng **lab1-task1-code.cc**

Dựa trên các file .pcap, sử dụng Wireshark để tìm và trả lời các câu hỏi sau:

3. Các Beacon frames là gì? có ý nghĩa như thế nào? Giải thích. Tìm trên các Beacon frames, SSID của AP là gì?

4. Tìm các gói tin thể hiện quá trình kết nối giữa các node và AP. Các thông điệp này có ý nghĩa gì?

Gợi ý: chú ý cấu trúc giao tiếp giữa lần lượt các node 2, 3, 4, 5 với AP.

5. Tìm hiểu quá trình khởi tạo kết nối UDP và chỉ ra các gói tin UDP đi từ client đến server và ngược lại. Giải thích sự tương quan giữa code và file .pcap.

Gợi ý: tìm hiểu, trình bày các gói tin ARP để đi đến quá trình kết nối.

6. Khoảng thời gian giữa các lần truyền Beacon frames là bao nhiêu? Hãy chứng minh điều đó.

7. Địa chỉ MAC nguồn, đích, BSSID trong 802.11 frame có giá trị bao nhiêu (dưới dạng thập lục phân)? Chúng có ý nghĩa như thế nào và chiếm bao nhiêu bytes trong cấu trúc gói tin?
8. Trong các Beacon frames quảng bá rằng có thể hỗ trợ nhiều data rates khác nhau, tìm các thông tin này? Cho biết các rates này có ý nghĩa gì?

Đối với nội dung thực tế bắt gói tin, dựa vào file .pcapng bắt được, sinh viên thực hiện các yêu cầu sau:

9. Có tổng cộng bao nhiêu AP xung quanh, SSID của các AP này là gì? Hãy chứng minh điều đó.
10. Thực hiện lại các câu hỏi 6, 7, 8. Các trường / nội dung này có gì đặc biệt, khác nhau giữa 2 kịch bản.
11. Tìm các gói tin Probe Request và Probe Response (từ máy tính của bạn) trong các 802.11 frames. Mục đích của các gói tin này dùng để làm gì? Địa chỉ MAC nguồn, đích, BSSID của chúng như thế nào?
12. Tìm cặp gói tin Authentication được gửi từ client đến AP và ngược lại, cho biết chúng dùng để làm gì? Các trường Authentication Algorithm, Authentication SEQ, Status code trong các gói tin Authentication ở trên thể hiện điều gì?
13. Tìm các gói tin Association được gửi từ client và ngược lại, trường thông tin nào cho biết chúng là một Association Request hay một Association Response? Tìm các thông tin Capabilities Information, SSID, rates supported trong Association Request và cho biết chúng có ý nghĩa như thế nào?
14. Giữa Association Request và Association Response có xuất hiện thêm trường gì đặc biệt? Giá trị của nó là bao nhiêu?
15. Từ các nội dung đã tìm hiểu, trình bày quá trình WLAN 802.11 hoạt động thông qua các gói tin và sự khác nhau giữa kịch bản 1 và kịch bản 2.

2. Đánh giá

- Sinh viên tìm hiểu và thực hành theo hướng dẫn. Thực hiện theo **nhóm**.
- Sinh viên báo cáo kết quả thực hiện và nộp bài bằng file. Trong đó:
 - Trình bày chi tiết quá trình thực hành và trả lời các câu hỏi nếu có (kèm theo các ảnh chụp màn hình tương ứng).
 - Giải thích, tìm hiểu, lý giải các kết quả đạt được.
 - Tải mẫu báo cáo thực hành và trình bày theo mẫu được cung cấp.

Nén **.ZIP** tất cả các file và đặt tên file theo định dạng theo mẫu:

NhomY-LabX_MSSV1_MSSV2

Ví dụ: Nhom1-Lab01_21520001_21520002

- Nộp báo cáo trên theo thời gian đã thống nhất tại website môn học.
- Các bài nộp không tuân theo yêu cầu sẽ **KHÔNG** được chấm điểm.

HẾT

Chúc các bạn hoàn thành tốt bài thực hành!