Wireless internet - WiFi traffic classification.

Luca Ferraro (10748116), Fabio Losavio (10567493)
https://github.com/LucaFerraro/Wireless_internet_project

A.Y. 2019/2020

Contents

List of Figures

Listings

1 Introduction

2 Project description

2.1 Program setup

To launch and use the program, first some pre-requisites have to be meet:

- install pyshark library in order to analyze the capture file
- install numpy library to perform some numerical analysis
- install matplotlib library to visualize plots

The program can run in 3 different modes selected by adding an attribute to the standard python call, using the command `python Traffic_analyzer.py -file "PATH TO A FILE"` the program will scan the file provided after the *-file* parameter. In this mode the program will open the selected capture (in a .pcap/.pcapng format) and start scanning all the packets in order to obtain some information.

The second mode is the live capturing mode in which, using the command `sudo python Traffic_analyzer.py -live "INTERFACE" "DURATION"` the program will first start a capture on the given interface,that has to be enabled to work in monitor mode, for a given amount of time provided with the "DURATION" attribute, will save this capture and will work on it. With this mode administrator privileges are required to allow tshark to start the capture in monitor mode.

The last mode is a default mode, launched using the command `python Traffic_analyzer.py`, which will perform the analysis on a default capture inserted in the code. This mode is mostly used as a debug tool but can also be useful to understand how the output will look like.

2.2 Program structure

To obtain information from a capture, the code will run a for loop on all the packets in the capture file and will analyze only those which can be useful for our purpose. To select the useful the following filter is used:

```
1     (int(packet.wlan.fc_type) == 2) and
2     ((int(packet.wlan.fc_subtype) >= 0 and
3         int(packet.wlan.fc_subtype) <= 3)) or
4     (int(packet.wlan.fc_subtype) >= 8 and
5     int(packet.wlan.fc_subtype) <= 11)
```

Listing 1: Packet filter

it will select all the **data** frames (wlan.fc_type == 2) and among those it will select the ones actually containing data or QoS data excluding null packets or ACKs. From those packets it will extract the destination address and will add it to a dictionary.