Group Practice 1: Introduction

Introduction
The objective of these practices is to create a realistic simulator of the TCP/IP stack. For this purpose, you will implement each layer as an independent thread. In this first practice, you have to prepare the environment and to test that all works fine. You will also create a good basis for the simulator so that it can grow properly.

Installing the environment
First of all, you will need to install the environment that will be used in the practices.

1. Install the last version of the Java SDK (32 bits).
2. Install Wireshark **with WinPcap**.
3. Go to the PDU and download Jpcap 0.7. Follow the install instructions in the documentation.
4. Install the last version of Eclipse to develop programs in Java (32 bits).

Learning how to manage Jpcap
Your first task is to try to work with this library. For that, follow the instructions that the tutorial gives you and try to create a simple program to receive and send packages. Do not submit this program for the practice. You don't need to learn how to save or read packets from files or how to set up filters. You will create your own filters and we won't save or read packets from files.

Starting with the TCP/IP stack
To create your TCP/IP stack, first you have to model it. Think that your application is going to have a main thread that will manage all the other threads. Each individual layer will be a thread. Using this information you can create an abstract class called *Layer* that extends *Thread*. All the layers will extend this class. All the common functions will be coded only once in this abstract *Layer*.

Each *Layer* at least needs the following methods:

- Configuration
- Run

They will be different in each layer. We can define these functions in *Layer* and then fulfill the code in each class that extends it.

At the beginning of the program, the main thread will execute the configuration method for each layer and will launch them after the configuration.

Layer 1
Layer 1 is the physical layer. In our case, this is equivalent to our NIC. For this practice you have to implement the Layer 1 class. This layer will be very simple:

- Configuration: You have to select the interface you want to use.
- Run: In this practice this layer will be very simple. It will be a repeater. Every time the NIC gets a packet, it will be sent again using the same interface. The program will do that for 10 packets.

Optional extension (up to 2 points)
The behavior of this repeater is not very realistic. Our repeater is sending the packets to the same segment while the desired behavior is to send them to another segment of the network.

To get extra points with this optional extension, you have to use two Layer 1 objects: one for the Ethernet NIC and another for the Wi-Fi one. If a Layer 1 receives a packet, it will be passed to the other Layer 1, which will send the packet to its segment.

The passing of packets from one layer to another will be studied in the next practice, but you should propose an approximation to this action.