Assignment 6

• You should have already installed and become familiar with the **Ubuntu Mate 20.04 Operating System(OS)**. You can do this either:

(1) with dual boot i.e. if you already have an OS such as MS Windows, MAC OS, or another linux distribution, just install Ubuntu Mate 20.04 as a secondary OS, or,
(2) by installing first a Virtual machine (e.g. the Oracle VM VirtualBox is freely available) in your existing OS, and then within the VirtualBox installing the Ubuntu Mate 20.04, or,
(3) by installing Ubuntu Mate 20.04 as your primary OS.
• If you prefer install the Ubuntu 20.04 OS instead of the Ubuntu Mate.
• You should have already installed the gcc compiler:
sudo apt update
sudo apt install build-essential
• Develop you C code using one of your favorite editors such as gedit, pluma or vi.

Monitoring the network traffic using the Packet Capture library

In this assignment, you will get familiar with the Packet Capture library (libpcap, it is installed by default if you are using Ubuntu Mate 20.04 Operating System) . This assignment assumes background knowledge in network protocols and familiarity with the C programming language.
For more information about the packet capture library, visit the following websites: https://linux.die.net/man/3/pcap, https://www.tcpdump.org.

You are expected to read a pcap file (test_pcap_5mins.pcap), and you will process the incoming/outcoming TCP and UDP packets. Do not use pcap_compile or pcap_setfilter. For a quick review about TCP protocol have a look at:
https://www.tutorialspoint.com/data_communication_computer_network/transmission_control_protocol.htm

More specifically, you are expected to do the following:

1. Select the pcap file name.
2. Start reading packets.
3. Decode each received TCP or UDP packet
4. Skip any packet that is not TCP or UDP.
5. Print the packet's source and destination IPv4 addresses.
6. Print the packet's source and destination port numbers.
7. Print the packet's protocol (please include higher level protocols e.g http)
8. Print the packet's TCP/UDP header length and TCP/UDP payload length in bytes.

9. Can you tell if an incoming TCP packet is a retransmission? If yes, how? If not, why?
10. Can you tell if an incoming UDP packet is a retransmission? If yes, how? If not, why?
11. In your program (when possible), mark each retransmitted packet as "Retransmitted".
12. On exit, your program must print the following statistics:
    a. Total number of network flows captured. Be careful what is and what is not a network flow using this definition: A network flow is a 5-tuple consisted of : {source IPv4 address, source port, destination IPv4 address, destination port, protocol}.
    b. Number of TCP network flows captured.
    c. Number of UDP network flows captured.
    d. Total number of packets received (include the packets you skipped, that weren't TCP or UDP packets.).
    e. Total number of TCP packets received.
    f. Total number of UDP packets received.
    g. Total bytes of TCP packets received.
    h. Total bytes of UDP packets received.

Tool Specification

Your tool will receive the following arguments from the command line upon execution.
Options:

-r Packet capture file name (e.g. test.pcap)

-h Help message

Notes

1. The options defined in the "Tool specification" section must remain as-is.
2. If no appropriate option was given, your program has to print the appropriate error message.
3. You need to create a `Makefile` to compile your library and programs (you must submit it with your source code).
4. You are provided with a sample packet capture to test your program. Its duration is 5 minutes.
5. You need to submit all the source code of your tool: a **"Makefile"**, "**monitor.c",** and a **"README.txt"** file that explains briefly your implementation and what you didn't implement and why. Place all these files in a folder named <yourlastnameAM>_assign6, and then compress it as a .zip file that you will upload to eclass. For example: christodoulou2018123456_assign6.zip.
6. The README.txt file is important to submit.
7. Very important: execute the command gcc –-version and write whatever the output is into your README.txt file, e.g. "gcc (Ubuntu 9.3.0-10ubuntu2~20.04)"