CNPro

Computer Networks Programming Practice Homework

Parts of this work are based on the textbook "Conceptual Computer Networks" ©2018 by José María Foces Morán and José María Foces Vivancos

· Submission deadline is 11th June 2018 at 23:59
· Include your full name and National ID in all source and documentation files
· Reference and cite the sources you have consulted
· Avoid any form of plagiarism
· All programs must be original
· Pack all files into a zip-compressed archive and send to foces.informatica.unileon@gmail.com with topic "CNPro 2018"
· Organize the response to each question in a separate folder (Question 1, Question 2, etc.). Include all the necessary source files and documentation under their respective folder
· I may contact some of you for appointing a face-to-face examination about the code you have submitted

1. [1 point] Compose a summary about RFC 2516. Use the concepts, structures and terms that we studied in chapter no. 1 about Network Architecture.

Network Programming exercises

2. [1.5 points] Program that send an icmp echo request and expect the icmp reply. Your receiving program must use Libpcap filters. Explain the difficulties you found. You must use libpcap.

3. [6 points] Program a 2-port simple bridge in Linux as described in the practice lab sessions. You can use the base C program included in Lab 4 (Published 3rd/May/2018).

    - Scan the source code for inconsistencies or errors and correct them if any; also, if upon testing the bridge you find some run-time issue, you are expected to resolve it.

    - The basis C program mentioned above uses two global variables that represent the pcap handles of the network interfaces to be used by the bridge service. Discuss whether this programming style is unavoidable in this context.

- It's essential that you document the unit and integrated tests you think should be carried out to make sure the bridge behavior is consistent with the simple bridge concept explained in the lab sessions.

- The bridge should result completely transparent to the Linux **IP host** it runs on and should produce no extraneous traffic.

- Propose some additional improvement of your own and implement it

4. [1 points] Discuss the benefits of programming the bridge in C with Posix Threads, particularly whether you can get rid of the pcap_setnonblock() call. Finally, produce an initial pthreads-based implementation of the bridge.

5. [0.25 points] Explain the usefulness of the Netlink Sockets interface as a complement to the bridge service developed above. You may want to start working this question by skimming the man page about netlink ($ man netlink) under Linux.

6. [0.25 points] If you were required to write the bridge program without using libpcap, discuss how would you proceed.