Parallel and Distributed Computing: Homework 3

Due April 24, 2020, in pdf format on Canvas.

"A protocol stack (such as OSI), today is usually provided by the operating system (rather than as a separate library, for instance), it is a set of programs that allow processes to communicate over a network using the protocols that the stack implements.

The application programming interface (API) that programs use to communicate with the protocol stack, using network sockets, is called a socket API. Development of application programs that utilize this API is called socket programming or network programming."

`https://en.wikipedia.org/wiki/Network_socket`

Assignment

Create two programs `Client` and `Server` in C, that using **sockets**, send a short string from `Client` to `Server`. Capture and analyze the message sent using Wireshark.

Install Wireshark in one of the computers connected to the local area network (LAN).

Start capturing packets, making sure of selecting the appropiate interface. Services, browser, and the OS constantly send packets to the internet and the local network, therefore, you will need to apply a **capture filter** (not a display filter) such as:

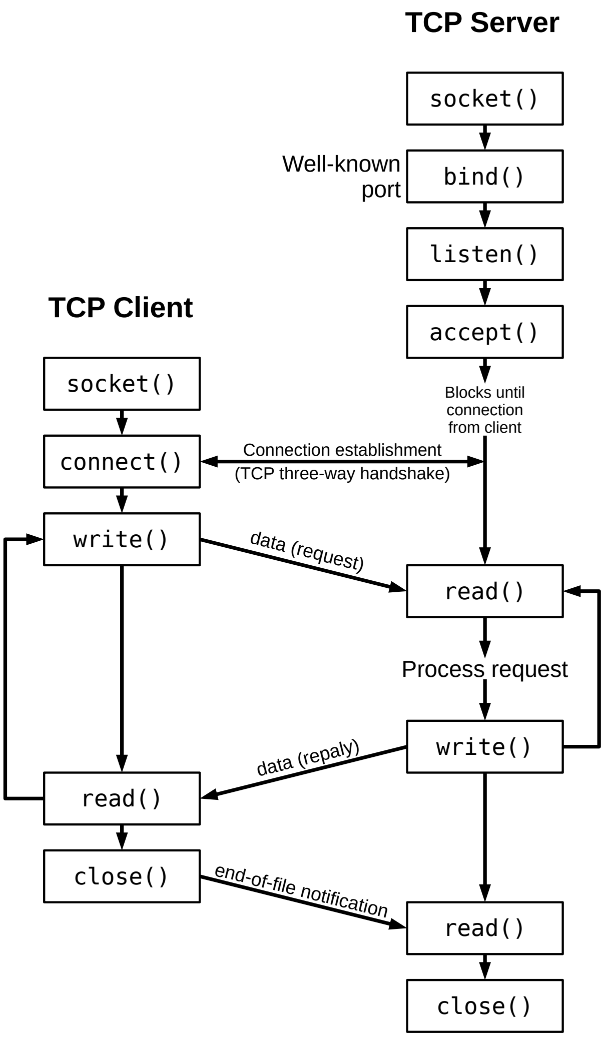
```
ip src XXXX and ip dst YYYY
```

Where `XXXX` is the source IP address, and `YYYY` is the destination IP address.

Run the `Server`.

In the other computer connected to the same network, run the `Client` to send the message.

For verification of feasibility, this exercise was already carried out using a laptop with Ubuntu 17.10, and a MacBook running OS-X. Both computers were connected to the same network, Ubuntu over wire and MacBook over Wifi. (Not tested in MS Windows).



After "Unix Network Programming"
by W. Richard Stevens

Submission

Submit a zip file named `x.zip`, where `x` is your UCInetID, containing 4 files:

- The commented C source code of the client, called `client.c`.
- The commented C source code of the server, called `server.c`.
- The captured packets in `pcapng` format, called `packets.pcapng`. The file must only contain packets transmitted between `Client` and `Server`.
- A digitally produced, individual report of two pages.

Identify the single packet that contains the message sent from `Client`. Based on this packet, answer the following questions in the first page of the report. Note that A-B denotes a range from A to B:

- What is encoded, and what is the purpose of bytes 0-5 and 6-11?
- What is encoded, and what is the relationship between, byte 14 and the two bytes 16,17?
- What is encoded, and what is the purpose of bytes 18-19?
- What is encoded, and what is the purpose of bytes 20-21?
- What is encoded, and what is the purpose of byte 23?
- What is encoded, and what is the purpose of bytes 26-29 and 30-33?
- What is encoded, and what is the purpose of bytes 34-35 and 36-37?
- What is encoded after byte 65?

Obtain a screenshot of the raw block of bits (located at the bottom part of Wireshark) of the single packet that contains the message sent by `Client`.

In the second page of the report, split the block of bits, and clearly associate each sub-block of bits to the layers 2 (Link), 3 (Network), 4 (Transport), and 7 (Application). Indicate the first and last byte of each layer.