Project #3 – Jenny, I've Got Your Number

Introduction

The purpose of this lab is to write a simple client using UDP transport.

Outcomes Addressed

- understand what networking protocols are and how they are specified
- understand the protocols of the Internet
- be able to write applications using socket connections
- understand the implementation and operation of Internet services

Background

The UDP transport is a thin layer above IP and provides minimal functionality. If features such as reliability, flow control, re-transmission are needed, one would need to implement those features at the application layer, or, choose another transport. Even with the limitations, UDP can be quite useful for certain applications. As a start, in this lab, we will explore a very simple application using UDP.

Activity

You must write an application that will send a message to the specified host (via IP address) and port, collect the response and display to the console. You may use any platform for this assignment, but

Specific requirements are:

1. Your program must accept the IP address of the specified host and port number via command line options.
2. Your program must send a message to the server with the following data in proper network order: an unsigned long integer (32-bit) with the value of 8675309, a char array with the null-terminated string "Jenny", an unsigned char with the value 241, and an unsigned short integer (16-bit) with the value 57005, for a total of 13 bytes. You must deal with these items as proper types (unsigned short, uint16_t, etc) in structure form. Recall that you may need to adjust structure packing in your C program (see example on emerald: https://emerald.msoe.edu/resources/courses:ce-4960:resources:structure_packing).
3. The reply packet will contain a single null-terminated string. You must receive this reply, and display the data received as a null terminated string to the console. If your request was proper, you will receive an inspirational message. If your request was not correct you will either receive a personal attack, or no reply at all. You should time out on your receive after a reasonable amount of time.
4. Close sockets and exit - pat your back, a job well done.

While you may benefit greatly by inspecting packets with WireShark, the above must be achieved programmatically.

A compatible quote server is running on whoami.nebula.msoe.edu at port 22222. Note that this is behind the firewall, so you must be on campus or connected via VPN to access.

Deliverable

For full credit, you must demo at the beginning of the Friday Week 4 lecture meeting. Submit a pdf printout of your source code (pdf only, no zip file). In the text comments for the assignment submission, paste in the text of your favorite quote returned by the server. Submission is due prior to Friday Week 4's lecture meeting.

Resources

Some example UDP programs can be found on emerald: https://emerald.msoe.edu/resources/doku.php?id=courses:ce-4960:resources:start

A popular website with lots of examples is: http://beej.us/guide/bgnet/, UDP examples are here: http://beej.us/guide/bgnet/output/html/multipage/clientserver.html#datagram