Justin Miles

CS 447 Networking and Data Structures

T. Gamage

Project 1 (protocol)

Introduction

To implement a custom handshake protocol with a client capable of multi-threading with TCP and at least single threading with UDP. I was able to develop a solution that accomplished these things. This project showed me a lot of the inner workings of the application level protocols possibilities for systems.

Objectives

My main goal was to establish connection between client and server over TCP and to relay some messages. I wanted to better understand just how complicated it might be for a application to communicate over a port and potentially with another system.

Design

The design of my application is taking port and address info and setting up a connection to another port on either UDP or TCP. Since I had a implementation started for the client connection over TCP and UDP separately I chose not to check the type of connection and to simply try to connect with TCP and allow it to fail if it was UDP and then it would start a UDP connection and vise versa. Once the connection was made I entered a loop that continued a variable to representing the status of the connection and would close if that status were so set. Within the loop there continues functions to parse input from the client/server and respond accordingly while keeping a state that was used to determine not only if the command was valid at that point in time but, how so to allow progress. In the TCP function a new thread would be spawned for each new connection up to 10 as my arbitrary limit. Given a more robust implementation as for a professional service this would have been configurable or determined by a system's upper limit for the application.

The application can be built via the makefile and the project files are compressed using the tar script.

Implementation

My implementation closely followed the tutorial for socket connections from Beej.us http://beej.us/guide/bgnet/ for connecting and disconnecting to the server and for the server itself to setup open connections for the TCP and UDP protocols. I left these largely in tact to help distinguish where I do venture off from his method, but some customization was required. The port and addresses were externalized as program arguments and threads were used in the server to facilitate using TCP and UDP at the same time. Once the connection was setup the communication between client and server is handed off to one of the ugliest methods I've ever written. So, because of the level of detail in the project specification I left it as compact as possible to verify I handled all cases. My intension, with more

time would be to better encapsulate the parsing and resolution of responses in the server. Encapsulation would have greatly aided testing of this application.

Output

The application follows the project specification of command output as close as I could determine from the document.

Summary