Scott Holley
Project 2, CS470
Write-up report

General Outline and Plan:

The outline of the project is to implement both UDP and TCP as protocols while demonstrating both unicast and multicast. The given example for this assignment is a projector. This is still essentially just a client and server program, with the server constantly “streaming” if using multicast, or consistently pushing to one client if unicast. In this write up, the server is the projector where the client is the mobile device. I just chose to think of them differently

The Planned Implementation:

Essentially, with what we are to demonstrate in mind, I’m going to implement the option to test either a UDP or TCP communication with the server. My reasoning for this is to illustrate a cast/protocol relationship. For example, a multicast is using UDP under the hood to deliver their message. So by using a Datagram, we can efficiently and effectively display use of both multicast and UDP.

On the other side of the spectrum, since we know that TCP by design is unicast oriented we can also combine these two attributes to get the most out of using the protocols. Thus, by letting the user decide between UDP and TCP, we’re demonstrating all four options by using the design of the protocol to our favor. The Server broadcasts the IP associated so that the client can look and connect on the same network.

The End Product Expectations:

The user should start up the server, and should be made aware that the server is up.
At this point, the receiver program can be started. They will connect with each other, and the user will be given a choice in the protocol to use. If the user chooses to test the UDP connection via multicast, then they will be sent a message while the server is active (i.e. Movie Data) to simulate streaming. It is at this point, another client could be opened and connect to the server to have data sent to it (the client). If the user chooses to use TCP via unicast, then only one client will be sent a message from the server, as only one device can be on this path at one time. Given the options, this seemed like the most efficient way to test TCP, UDP, multi, and unicast. Especially since we can assume that client and server are on the same network.

Scott Holley
Project 2, CS470
Write-up report