

# EECS 678 - Lab 04

Theodore Lindsey

April 24, 2015

<http://people.eecs.ku.edu/~vvivekan/lab/sockets/sockets.html>

1. *Briefly describe the design of the program. How does your program control when the client runs and when the server runs?*

First, both the server and the client need to know where to look for each other. The server starts by waiting for the client to initiate contact (and so, the server must be started first). Once the client starts running (after the server has already been started), it reaches out to the server and initiates communication. The server and client then handshake to arrange for an actual channel of communication and then begin communicating.

Once the communication channel is established, the server and the client take turns waiting for and then broadcasting a message. First, the client send a message to the waiting server. Then, the server processes the message and broadcasts it to the waiting client. The client knows that there will be only 3 messages and the server just waits for the connection to close before finishing itself.

2. *What is the purpose of the handshake socket? Why not have the server create and bind session sockets that clients may connect to directly?*

The purpose of the handshake socket is to listen for and set up incoming connections. The handshake then sets up the direct line of communication. This allows for versatility in the server. The handshake socket listens for incoming connections. The communication socket handles bi-directional data communication.

3. *For the simple / client server program, we chose to use sockets instead of pipes to send messages between the client and server. Why are sockets preferred over pipes for this program? Give at least two reasons.*

Pipes are designed for single-directional communication. Sockets are designed for bi-directional communication. Synchronization between a pipe flowing one way and a pipe flowing the other way would be much more difficult than writing to a pipe or waiting to receive a message from a pipe.

Sockets are easily implementable for local communication on a single machine or even via network traffic between multiple machines. Changing the setup for networked communication requires very few changes and would be easy to perform. If the sockets were set up and pointed to some given address, then simply pointing to localhost would replicate the functionality of the current program in that it would work fine locally. Pipes are useless for network communication.