



Android Suite Chat

James Senior

CPSC 353, Data Comm/Computer Networks

Department of Computer Science, Chapman University, Orange, CA



Introduction

For this project, I wanted to implement a topic we covered in class through an outside deliverable. After considering various methods and executions, I decided to create an application for a mobile device. Whether this would include a tablet, phone, watch, etc., I figured it would be interesting to take a data communication method covered in class, and apply it to a device. I find it fascinating to be able to take what I have learned in this class and be able to create programs that have the capability to talk to each other via various methods. Across the many communication methods we covered in class, one of the main methods was the TCP protocol (Transmission Control Protocol), which is what I ended up implementing in this release.

Hypothesis

I originally strived to create an iPhone application that would have the ability to connect to a server and communicate with other instances of that application on the same server. I figured this would be a great idea due to the fact that many people have iPhones and the technology would be somewhat easy to distribute and would not have to cater to many different derivatives of the same software or devices.

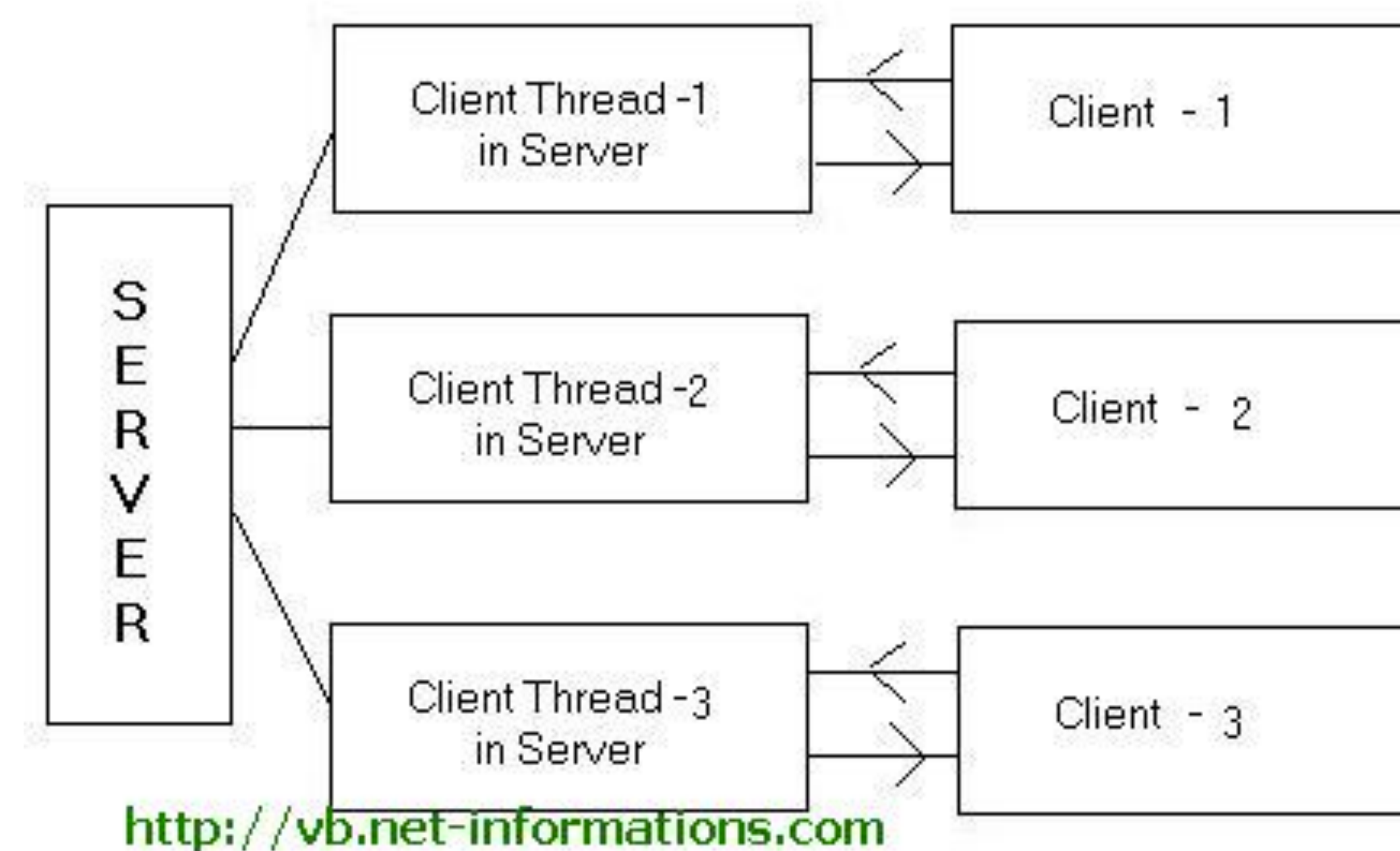
Experimental Method

Firstly, I began attempting to code a client for iPhone that would have the ability to connect to a server written in either Java or Python. I soon ran into trouble due to my little experience in iPhone programming combined with the realization that distributing/developing applications for iOS is much more complicated than other operating systems. I then turned to Android due to the Java support along with the simplicity of loading applications onto multiple devices.

Results

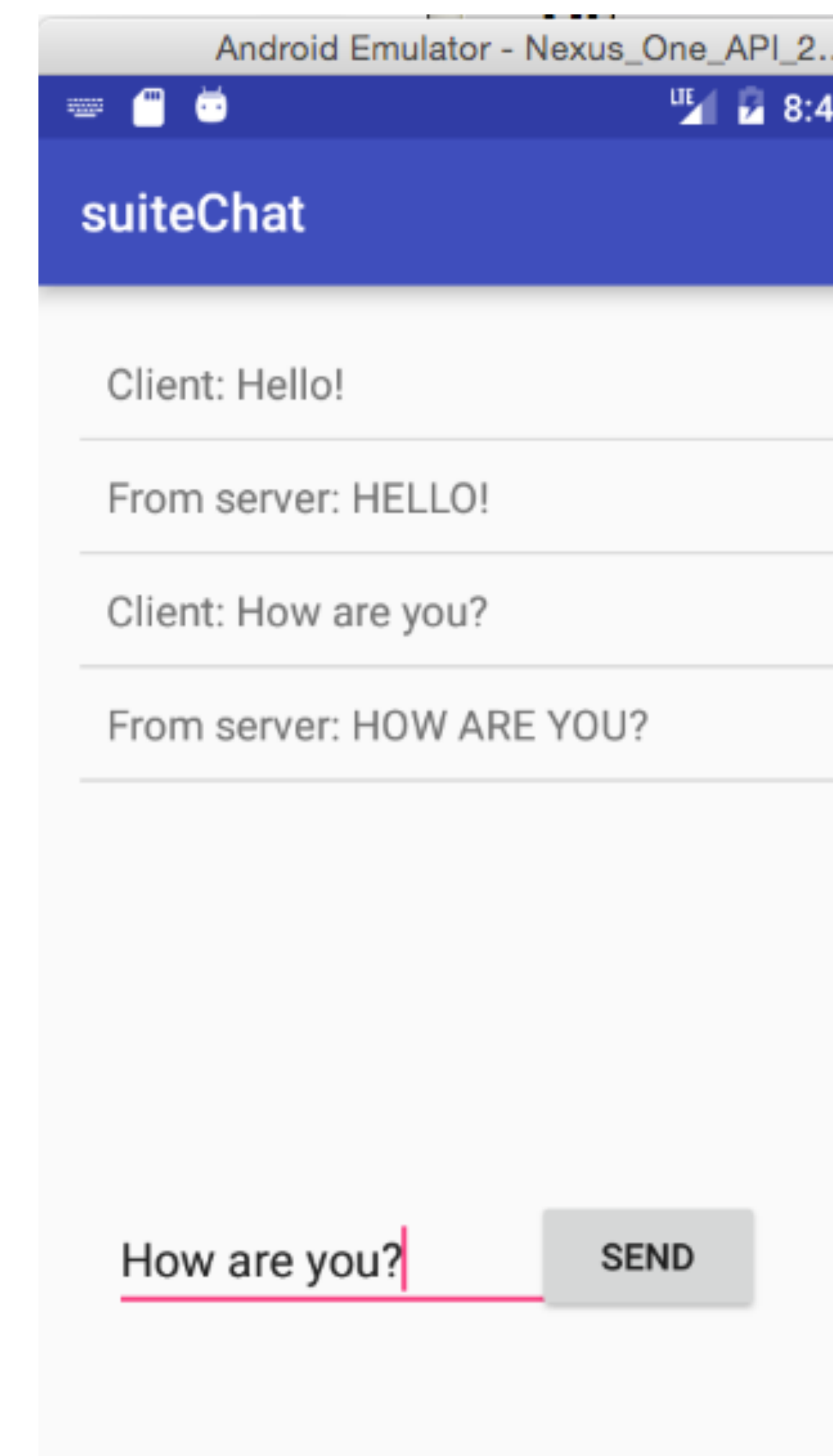
For my finished project, I have created both an application and a server that can communicate back and forth. The server is able to handle more than one application at once, meaning that multiple phones can connect to the single server and hold conversations, similar to a group chat. Furthermore, if two phones are connected, they have the ability to play a simple game such as tic-tac-toe. This ability is made possible via multithreading. Multithreading means that the client has multiple processes running at once. Mainly, one process is running the application and sending messages, while the other process is listening for messages from the server.

The server also implements multithreading by creating a new “process” each time a new application connects. This gives the server the ability to push messages to multiple applications all at the same time, while also listening for messages from the applications simultaneously.



<http://vb.net-informations.com>

```
Pictures — java — 80x24
Last login: Mon Nov 28 08:10:50 on ttys002
James-2:~ James$ cd Pictures
James-2:~ Pictures James$ javac Server.java
James-2:~ Pictures James$ java Server
From client: Hello!
From client: How are you?
```



```
//this class has a main responsibility to hold the socket object.
class tcp
{
    //declare variables for connection
    static String ipAddress;
    static int portNum;
    static Socket clientSocket;
    //constructor for when referring to class
    tcp(String ip, int port)
    {
        ipAddress = ip;
        portNum = port;
    }
    //connect method to establish a TCP connectoin to server
    public static void connect() throws Exception
    {
        clientSocket = new Socket(ipAddress, portNum);
    }
}
```

Conclusions

In conclusion, I am extremely happy with how my project has progressed. I theoretically rebuilt the idea I had for my hypothesis, and decided to create the project on a completely different platform. Developing an Android application made the project much more simple for me due to the fact that I am very familiar with Java opposed to iOS's Objective-C. Additionally, testing the application on an actual mobile device instead of just running an emulator was much easier.

Future Research

This application has a wide array of potential opportunities/abilities. For instance, I could include support for more games, or other ways to use the communication protocol for various purposes, due to the stability and widespread support for the TCP protocol.

References

1. "VB.NET MultiThreaded Socket Programming." VB.NET MultiThreaded Socket Programming. N.p., n.d. Web. 30 Nov. 2016.