Jacob Malcy

TA: Vipra Gupta

Section: 104

Assignment 8 Project Proposal

**What is your project?**

My project is a simple, yet somewhat secure, person to person messaging system that works over your Local Area Network (LAN). This only works for 2 people at a time. One person is the host, and the other is the client. It works by using C++ sockets over a TCP IP connection, using IPv4 addresses.

**Explanation of the network manager:**

The networkManager class is a class that handles the creating and maintaining the connections, receiving and sending messages, logging messages, and validating IP addresses. It handles both the client side of things and the host side of things.

**How do the connections work:**

I’ll avoid getting really technical about networks. The constructor for the networkManager class takes an array of 3 strings. These are, in order, whether you are the host or not, the IP address that will be used for connecting or hosting, and your user name. Once the appropriate data members are initialized with these values the function “makeSocket()” is called. This function will take different actions depending on if you’re the host or not.

If you are, the function will attempt to create a socket, set some options for the socket that prevent errors about port and address usage, bind the socket to a usable and appropriate IPv4 address, listen for incoming connections, attempt to accept them, and the declares an open that a connection has been made. At each one of these steps, the function will check for errors and print errors accordingly should they be found.

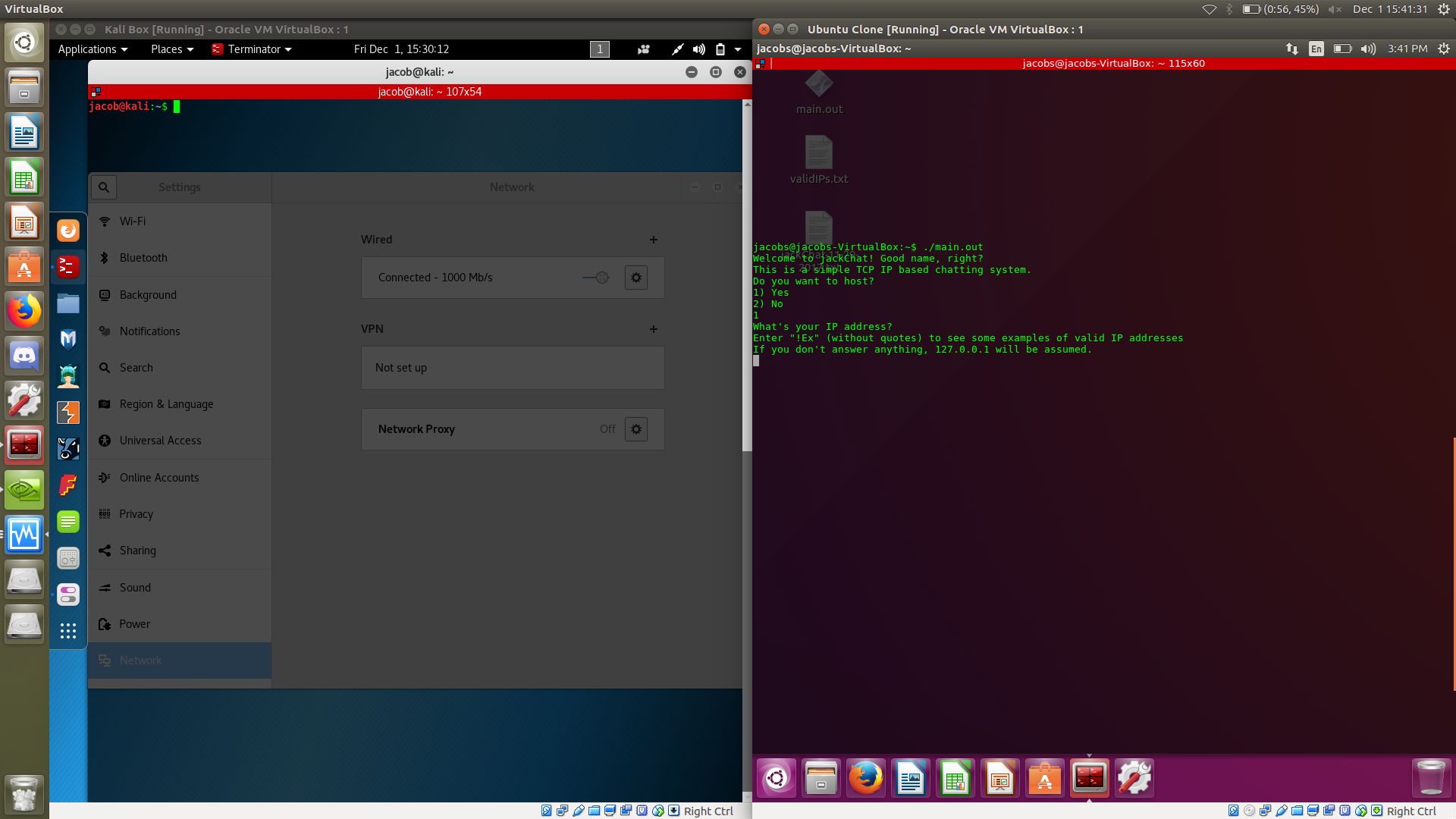
If you’re not the host, the function will create a socket and attempt to connect to the server. If it does, the connection will be declared open and the chat will begin.

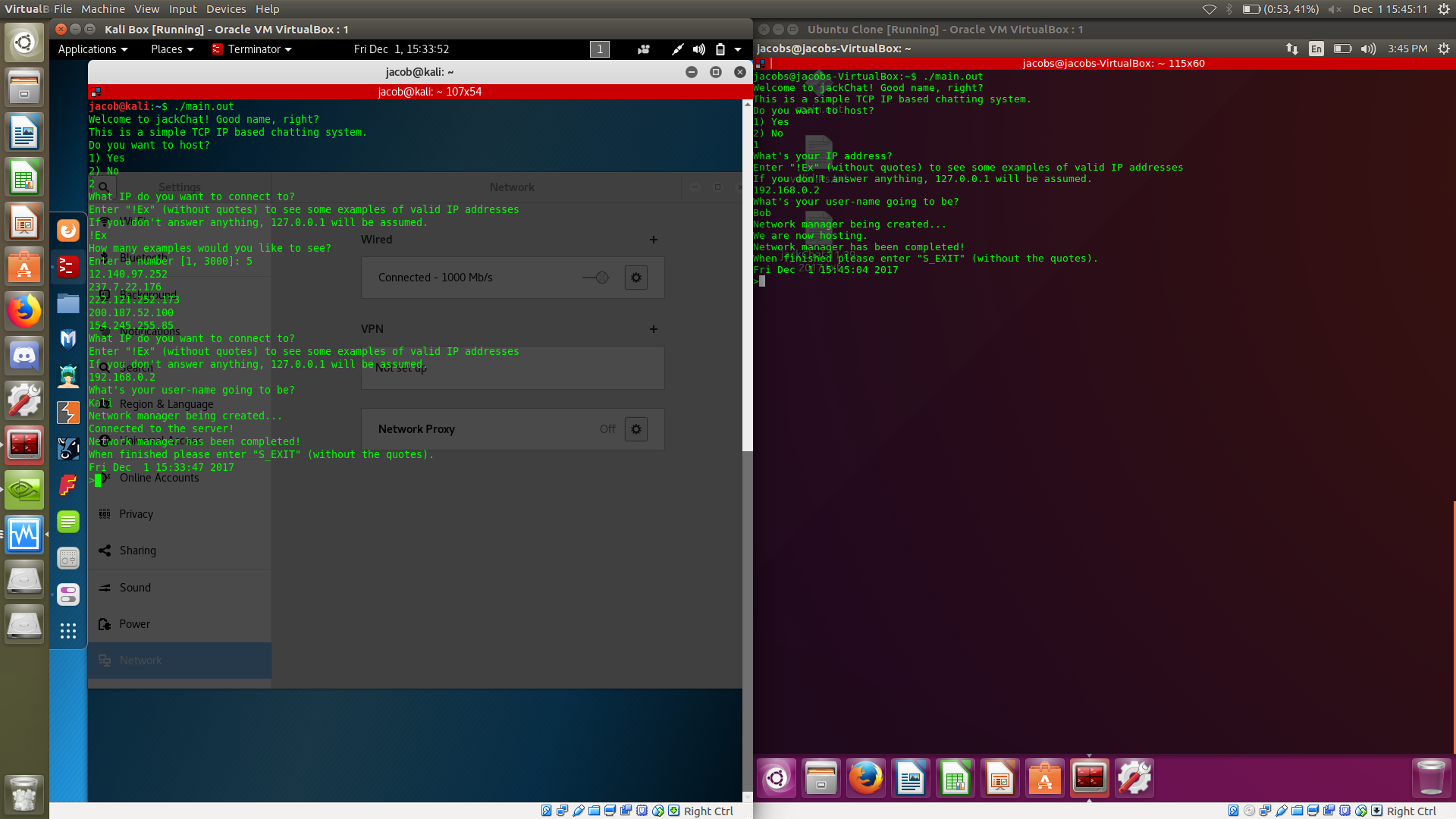
**How do we send and receive messages:**

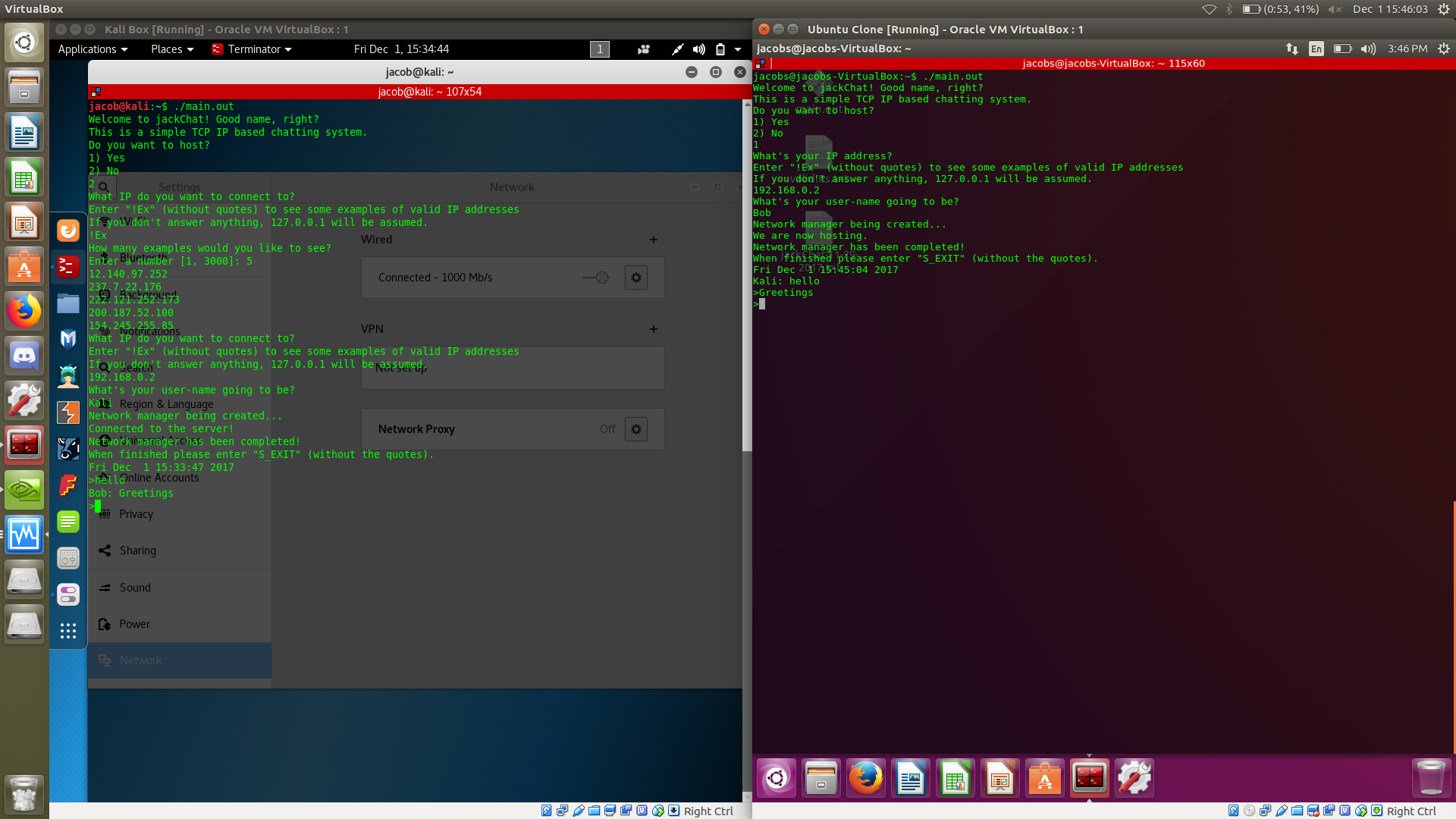
Once a connection has been opened, two threads are created. One will manage sending messages to the other person, and the other will manage receiving them. They run along side each other so that you can send and receive messages at the same time. They both access the socket(s) that were made during the initial boot up.

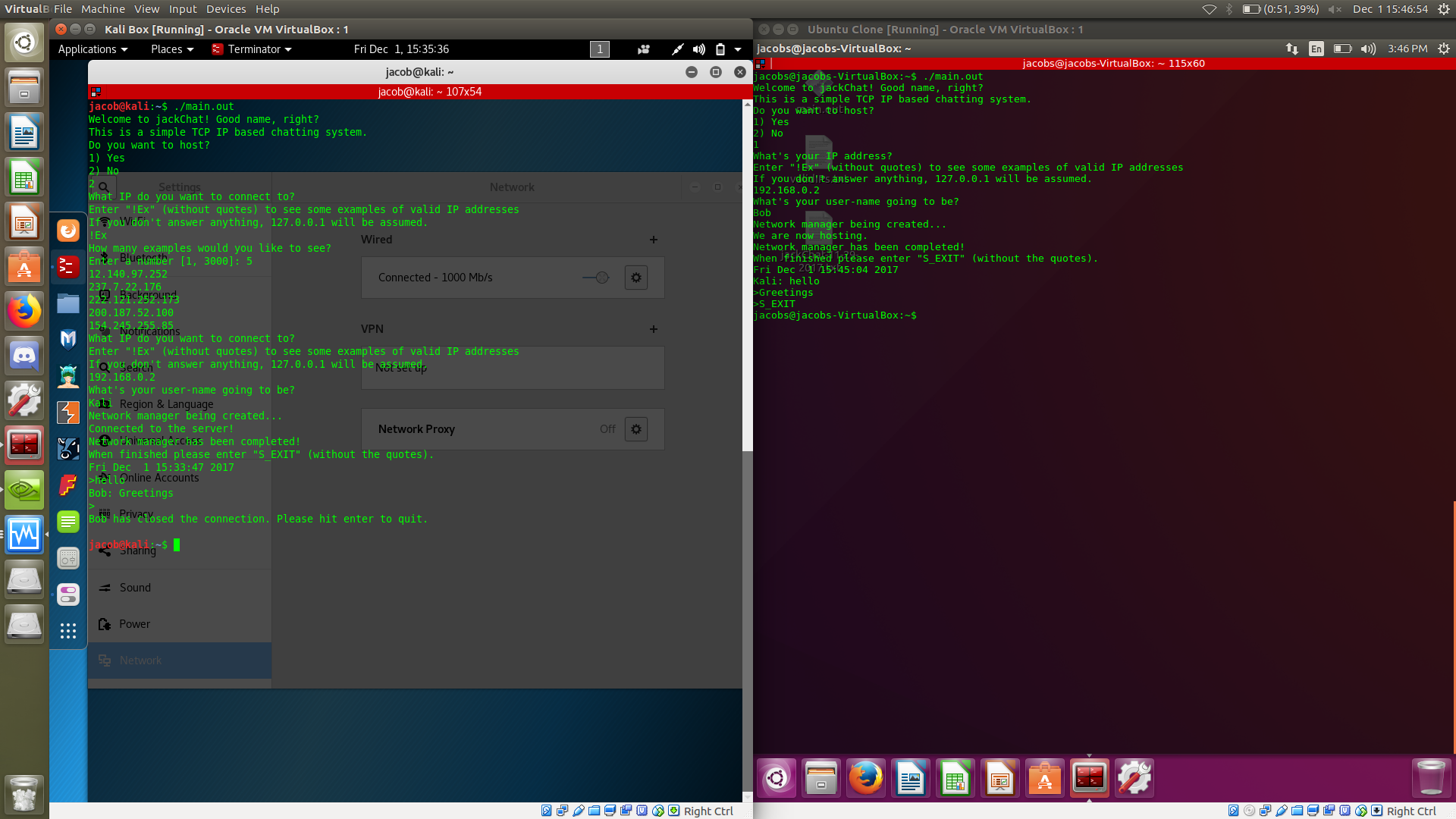
**Screenshots**:

Description of the screenshots: These are photos of me running the same program on two different virtual machines that are connected over a host-only network, running through a VyOS router that I built from scratch.

****

****

****

****