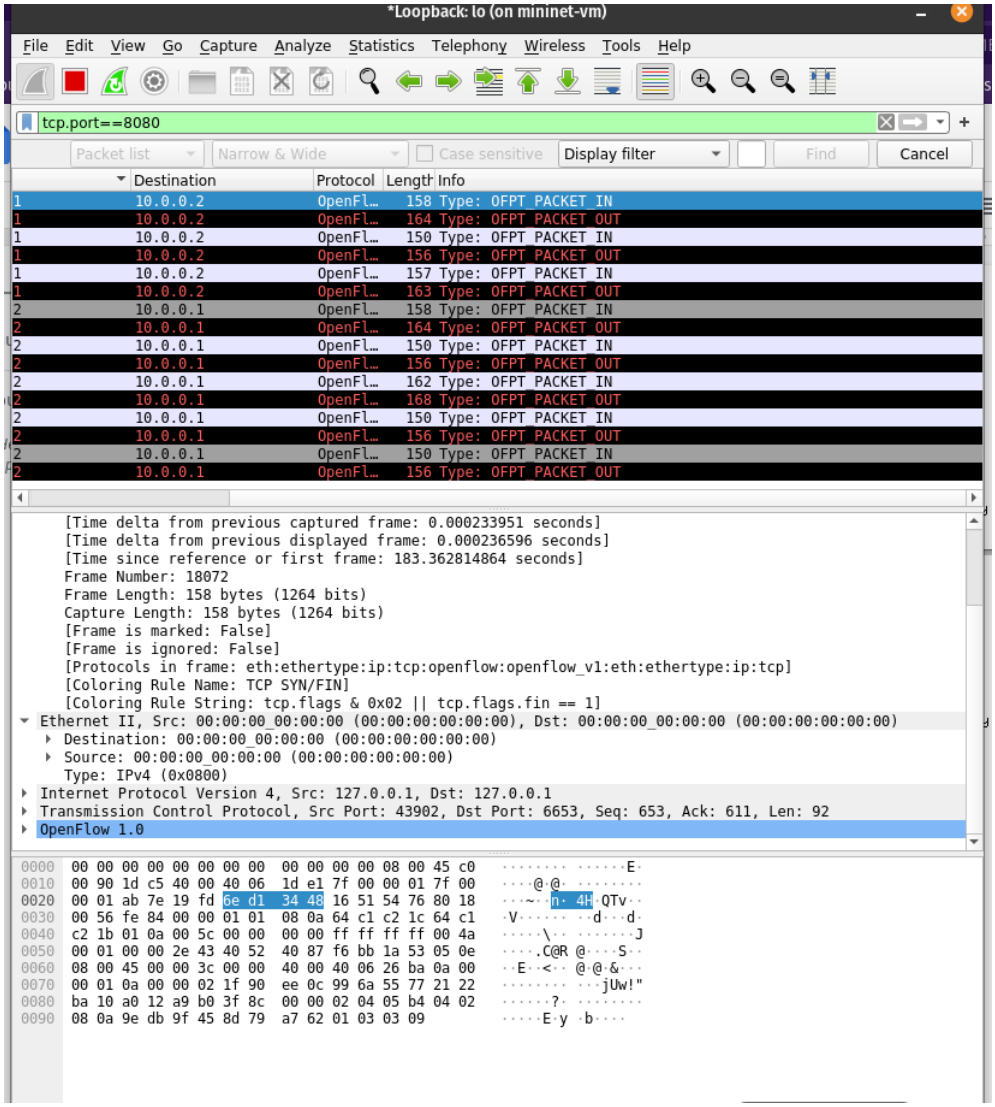
Client and Server:

```
root@mininet-vm:~/cs370# python client.py
Received b'got it!'
root@mininet-vm:~/cs370#
```

```
root@mininet-vm:~/cs370# python server.py
Connected bu ('10.0.0.2', 60940)
root@mininet-vm:~/cs370#
```

Wireshark:



Write up:

One main thing I learned from this lab is how abstracted networks have become for most developers. The difference between what is coded in python and what is actually happening between the two hosts with exchanged frames, handshakes, etc is much more complex than the python code shows. This is a good thing, but for someone who has never taken a closer look at how a client and a server communicate, this was very interesting. My ethernet II section of the frame matches with the structure we discussed in class, and it was interesting to see how the different structures are placed together within a frame.

Within my wireshark, it appears that packets were not fragmented. I believe that the header has options specifying how long the packets stay alive, as well as other information. Interestingly, no checksums were used. Maybe because in reality this is all done locally? I believe there are in and out flows because mininet runs the packets through a switch, as each node believes they are a different device. My only issue during this lab was that my source and destination fields in Ethernet II were both just 00:00:00:00, but I get all the frames, and the two nodes connect and communicate. Not sure if it's something with my specific setup, or maybe ethernet wasn't used, or something else?