Client and Server:

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

Hoodings you add to the document w///
root@mininet-vm:~/cs370# python server.py
Connected bu ('10.0.0.2', 60940)
root@mininet-vm:~/cs370# []
```

Wireshark:

```
*Loopback: lo (on mininet-vm)
<u>F</u>ile <u>E</u>dit <u>V</u>iew <u>G</u>o <u>C</u>apture <u>A</u>nalyze <u>S</u>tatistics Telephony <u>W</u>ireless <u>T</u>ools <u>H</u>elp
 A 📕 🔏 🔞 🖮 🖺 🔯 🌀 | 🤇 🧽 📦 壁 🍒 👲 🕎
                                                                                         QQQ
tcp.port==8080
                                                                                                                            X → ▼ +
       Packet list 

Narrow & Wide 

Case sensitive Display filter
                                                                                                    ▼ Find Cancel
              ▼ Destination
                                         Protocol Length Info
                10.0.0.2
                                                        150 Type: OFPT_PACKET_IN
                10.0.0.2
                                          OpenFl...
                                                      157 Type: OFPT_PACKET_IN
                                                     163 Type: OFPT PACKET OUT
158 Type: OFPT PACKET IN
                10.0.0.1
                                                       164 Type: OFPT PACKET OUT
                                                        150 Type: OFPT_PACKET_IN
                                           OpenFl...
                 10.0.0.1
                                                       156 Type: OFPT PACKET OU
                                                        162 Type: OFPT_PACKET_IN
                                           OpenFl...
                 10.0.0.1
                                          OpenFl... 168 Type: OFPT PACKET OUT
OpenFl... 150 Type: OFPT PACKET IN
                10.0.0.1
                                          OpenFl... 156 Type: OFPT PACKET OUT
OpenFl... 150 Type: OFPT PACKET IN
OpenFl... 156 Type: OFPT PACKET OUT
                10.0.0.1
      [Time delta from previous captured frame: 0.000233951 seconds]
      [Time delta from previous displayed frame: 0.000236596 seconds]
[Time since reference or first frame: 183.362814864 seconds]
      Frame Number: 18072
Frame Length: 158 bytes (1264 bits)
Capture Length: 158 bytes (1264 bits)
[Frame is marked: False]
[Frame is ignored: False]
      [Protocols in frame: eth:ethertype:ip:tcp:openflow:openflow_v1:eth:ethertype:ip:tcp]
[Coloring Rule Name: TCP SYN/FIN]
[Coloring Rule String: tcp.flags & 0x02 || tcp.flags.fin == 1]

Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:0), Dst: 00:00:00_00:00:00 (00:00:00:00:0)
     Destination: 00:00:00_00:00:00 (00:00:00:00:00:00)
   Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 43902, Dst Port: 6653, Seq: 653, Ack: 611, Len: 92
OpenFlow 1.0
··E··<·· @·@·&·
·····jUw
                                                                      · · · · · E · y · b · · · ·
```

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?