The strategy to solve the problem was to continue from the code from assignment 2a, then start modifying the reliable channel to now use it with UDP sockets. From there, developing Server.py to have the GBN and SR file transfer functionalities.

Since the program must rely on threading to be able to perform its functions, I spend a lot of time researching the threading functionalities needed for my program until finding a way of implementing the Server. After the implementation of the Server, I started implementing Client.py to be able to run my program.

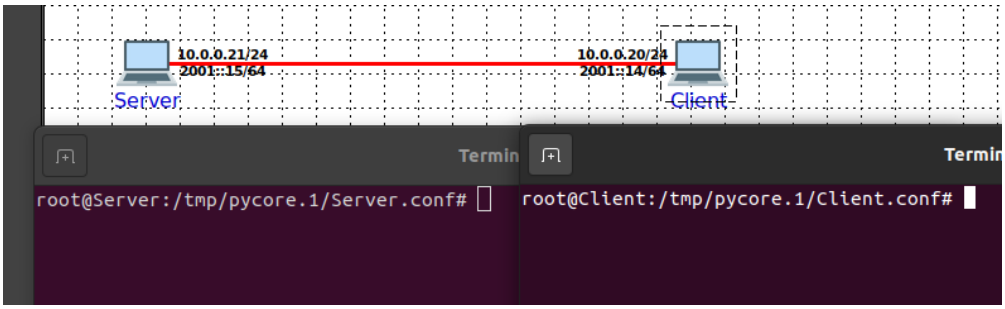
After the update on both files, part of the Client.py file is proven to be working, but unfortunately the sender side still needs further upgrading in order to send the file.

Demonstration of a run of both Server.py and Client.py using CORE:

```
<top/cnetworks/assignment3/Client# python3 Client.py
Provided Server IP: 10.0.0.20
Provided Port #: 24
You are now connected! Enter your commands now.
Asking for file: readme.txt
Traceback (most recent call last):
```

```
<top/cnetworks/assignment3/Server# python3 Server.py
Listening at port #:24
Protocol to use (GBN/SR): GBN
Listening for connection at : 24 ...
Connection accepted from  ('10.0.0.21', 46734)
Sending the file...
```

To run the file using core, first create a basic client – server topology:



, run the topology and enter Server.py/Client.py file location, run the program and enter the information requested.

References used:

[1] Intro to threading: https://realpython.com/intro-to-python-threading/

[2] Further explanation of thread lock: https://www.8bitavenue.com/python-mutex-example/