

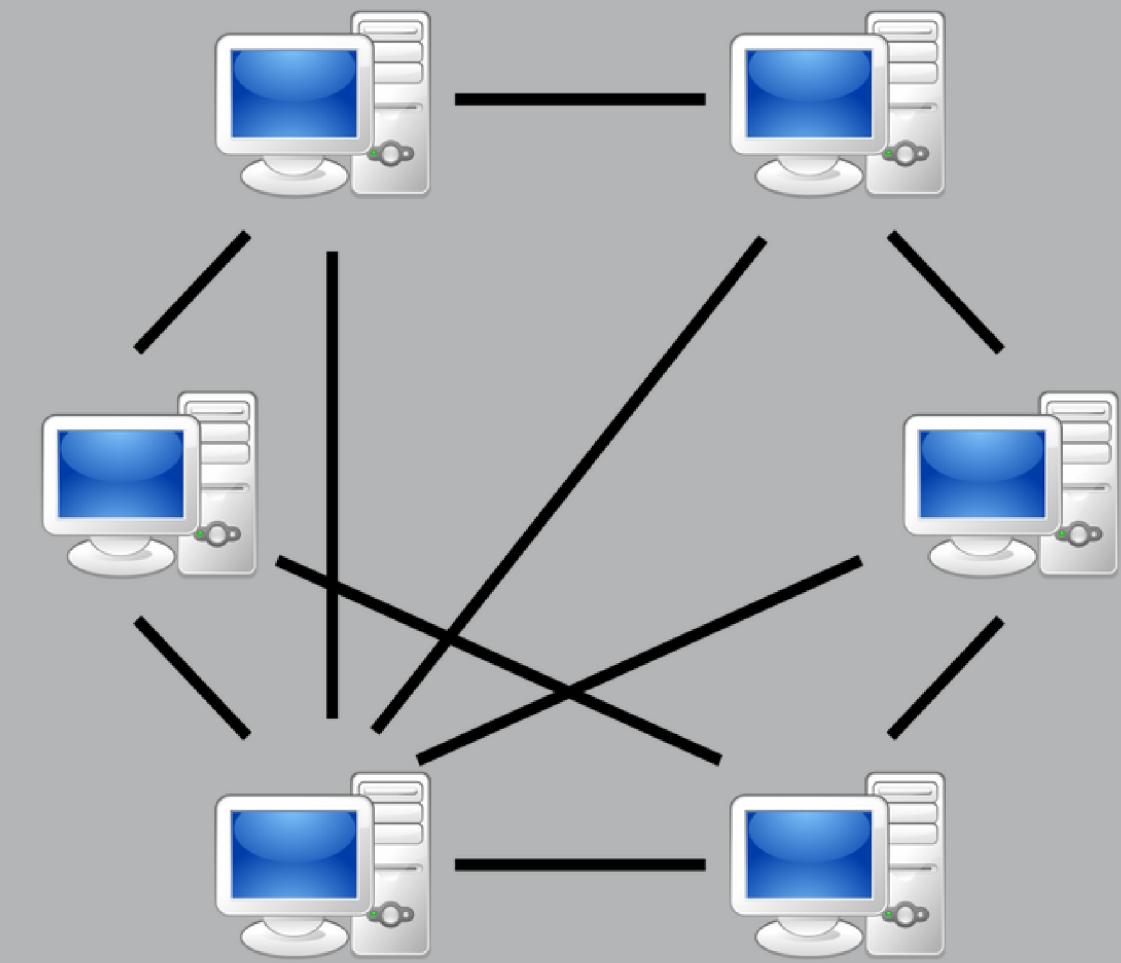
12230040

RUZIMATOVA DIYORAKHON

# FINAL PROJECT

P2P SYSTEM USING TCP PROTOCOL

**THE PROPOSED PROJECT AIMS TO DEVELOP A PEER-TO-PEER (P2P) SYSTEM USING PYTHON 3 AND THE TCP PROTOCOL. THE SYSTEM WILL ALLOW CLIENTS TO SEND AND VISUALIZE IMAGE, CSV, AND JSON FILES ON OTHER CLIENT SIDES. IT WILL ALSO PROVIDE THE CAPABILITY TO BROADCAST ANY FILE TO MULTIPLE CLIENTS. THE SYSTEM WILL BE IMPLEMENTED WITH A MENU-BASED INTERFACE FOR SELECTING DIFFERENT CHOICES. THE PROJECT WILL BE WELL-COMMENTED TO ENSURE CLARITY AND UNDERSTANDING OF THE CODE. ADDITIONALLY, EXPERIMENTAL RESULTS AND ANALYSIS WILL BE CONDUCTED USING WIRESHARK TO MEASURE END-TO-END LATENCY FOR EACH SCENARIO, AND A TIMEOUT MECHANISM WILL BE IMPLEMENTED FOR SENDING FILES**



```
        "min": "0.0",
        "max": "833.789",
        "avg": "285.5177",
        "values": "[344.336,306.0544999999996,269.922,233.0,
        "start_time": "None",
        "start_value": "None"
    },
    {
        "id": 38421212540,
        "sensor_id": 12944473,
        "value": "6771.2902",
        "date": "2015-02-04",
        "min": "0.0",
        "max": "828.9065",
        "avg": "282.1371",
        "values": "[333.0075,297.0705,253.3204999999998,213.0,
        "start_time": "None".
```

```
pythonProject4 ~/Test_project-/pythonProject4
5ce426d9a4a48f75aee6f062_what-is-a-comp
Client.py
electronic-card-transactions-april-2023-csv-tables.csv
File
Final_exam.json
Server.py
External Libraries
Scratches and Consoles

140     filtered_data = [entry for entry in json_data if entry.get('date') == date]
141     print(filtered_data)
142     else:
143         print("Invalid choice. Please try again.")
144     else:
145         print("Unsupported file type.")
146
147     # Function to display the menu and handle user choices
148     def show_menu(client_socket=None):
149         print("1. Receive a file from the server")
150         print("2. Visualize an image, CSV, or JSON file")
151         print("3. Send a file or data to the server")
152         print("4. Exit")
153
154         choice = input("Enter your choice: ")
155
send_data() > if os.path.isfile(file_path)

Run Client x Server x
/Users/diyorakhon/PycharmProjects/pythonProject1/new/bin/python /Users/diyorakhon/Test_project-/pythonProject4/Client.py
Connected to the server.
1. Receive a file from the server
2. Visualize an image, CSV, or JSON file
3. Send a file or data to the server
4. Exit
Enter your choice: 2
pythonProject4 > Client.py
17:55 LF UTF-8 4 spaces Python 3.10 (new)
```

```
pythonProject4 ~/master
Project v
pythonProject4 ~/Test_project-/pythonProject4
5ce426d9a4a48f75aee6f062_what-is-a-comp
Client.py
electronic-card-transactions-april-2023-csv-tables.csv
File
Final_exam.json
Server.py
External Libraries
Scratches and Consoles

13     file_data = file.read()
14
15     # Get the file size
16     file_size = os.path.getsize(file_path)
17
18     # Send the file size to the server
19     server_socket.send(str(file_size).encode())
20
21     # Wait for the server to acknowledge the file size
22     server_socket.recv(1024)
23
24     # Start measuring the time
25     start_time = time.time()
26
27     # Send the file data to the server
28     server_socket.sendall(file_data)
29
30     # Wait for the server to acknowledge the file receipt
send_data() > if os.path.isfile(file_path)

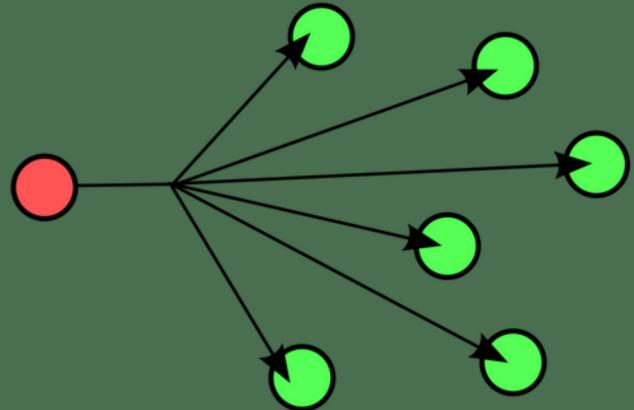
Run Client x Server x
/Users/diyorakhon/PycharmProjects/pythonProject1/new/bin/python /Users/diyorakhon/Test_project-/pythonProject4/Client.py
Connected to the server.
1. Receive a file from the server
2. Visualize an image, CSV, or JSON file
3. Send a file or data to the server
4. Exit
Enter your choice: 2
pythonProject4 > Client.py
17:55 LF UTF-8 4 spaces Python 3.10 (new)
```

A CLIENT CAN SEND AND  
VISUALIZE THE IMAGE, CSV  
AND JSON FILES IN OTHER  
CLIENT SIDE

```
1 usage  
  
def broadcast_file(file_path, client_sockets):  
    # Iterate over all the client sockets  
    for client_socket in client_sockets:  
        # Send the file to each client  
        send_file(file_path, client_socket)  
  
# Function to handle the visualization of different file types
```

```
1 usage
```

```
1. EXIT  
Enter your choice: 3  
Enter the path of the file to broadcast: /Users/diyorakhon/Test_project-/pythonProject4/Final_exam.json  
File broadcasted successfully.  
1. Send a file to another client
```



**A CLIENT CAN BROADCAST ANY FILE TO MULTIPLE  
CLIENTS**



PNG

Period	Data_Value	Suppressed	Status	Units	Magnitude
2001.03	2462.5		F	Dollars	6
2002.03	17177.2		F	Dollars	6
2003.03	22530.5		F	Dollars	6
2004.03	28005.1		F	Dollars	6
2005.03	30629.6		F	Dollars	6
2006.03	33317.4		F	Dollars	6
2007.03	36422		F	Dollars	6
2008.03	39198		F	Dollars	6
2009.03	40629.4		F	Dollars	6
2010.03	41815.4		F	Dollars	6
2011.03	43848.6		F	Dollars	6
2012.03	47004.9		F	Dollars	6
2013.03	49188.8		F	Dollars	6
2014.03	52108.2		F	Dollars	6
2015.03	54410.5		F	Dollars	6
2016.03	57389.5		F	Dollars	6
2017.03	60616.6		F	Dollars	6
2018.03	64238.6		F	Dollars	6

CSV

## JSON

```
Run Client × Server ×
[Run] [Close] [More]
/Users/diyorakhon/PycharmProjects/pythonProject1/new/bin/python /Users/diyorakhon/PycharmProjects/pythonProject1/main.py
Connected to the server.
1. Receive a file from the server
2. Visualize an image, CSV, or JSON file
3. Send a file or data to the server
4. Exit
Enter your choice: 2
Enter the path of the file to visualize: /Users/diyorakhon/Test_project-/pytho...
Enter '1' to display full data or '2' to display data from a specific date:
Enter the date (YYYY-MM-DD): 2021-06-09
[{'id': 38421163212, 'sensor_id': 12944473, 'value': '1234.5678', 'date': '2021-06-09T12:00:00'}
1. Receive a file from the server
2. Visualize an image, CSV, or JSON file
3. Send a file or data to the server
4. Exit
Enter your choice: 3
Enter the file path or data to send: /Users/diyorakhon/Test_project-/python...
File sent successfully.
End-to-end latency: 0.000049 seconds
1. Receive a file from the server
2. Visualize an image, CSV, or JSON file
3. Send a file or data to the server
4. Exit
```

```
Run Client × Server ×

⟳ ⌂ : 

/Users/diyorakhon/PycharmProjects/pythonProject1/ne
Server is listening on 127.0.0.1:8004
New connection from 127.0.0.1:63724
1. Send a file to another client
2. Visualize an image, CSV, or JSON file
3. Broadcast a file to multiple clients
4. Exit
Enter your choice: 3
Enter the path of the file to broadcast: /Users/diy
File broadcasted successfully.
1. Send a file to another client
2. Visualize an image, CSV, or JSON file
3. Broadcast a file to multiple clients
4. Exit
```

# A MENU FORM SELECTING CHOICES

```
1 import socket # Importing the socket module for network communication
2 import os # Importing the os module for operating system-related tasks
3 import json # Importing the json module for working with JSON data
4 import csv # Importing the csv module for reading and writing CSV file
5 import time # Importing the time module for time-related functions
6
7 # Function to send a file or data to the server
8 usage new *
9 def send_data(file_path, server_socket):
10     # Check if the input is a file path
11     if os.path.isfile(file_path):
12         # Send file to the server
13         with open(file_path, 'rb') as file:
14             file_data = file.read()
```

```
data_size = len(data_bytes)

# Send the data size to the server
server_socket.send(str(data_size).encode())

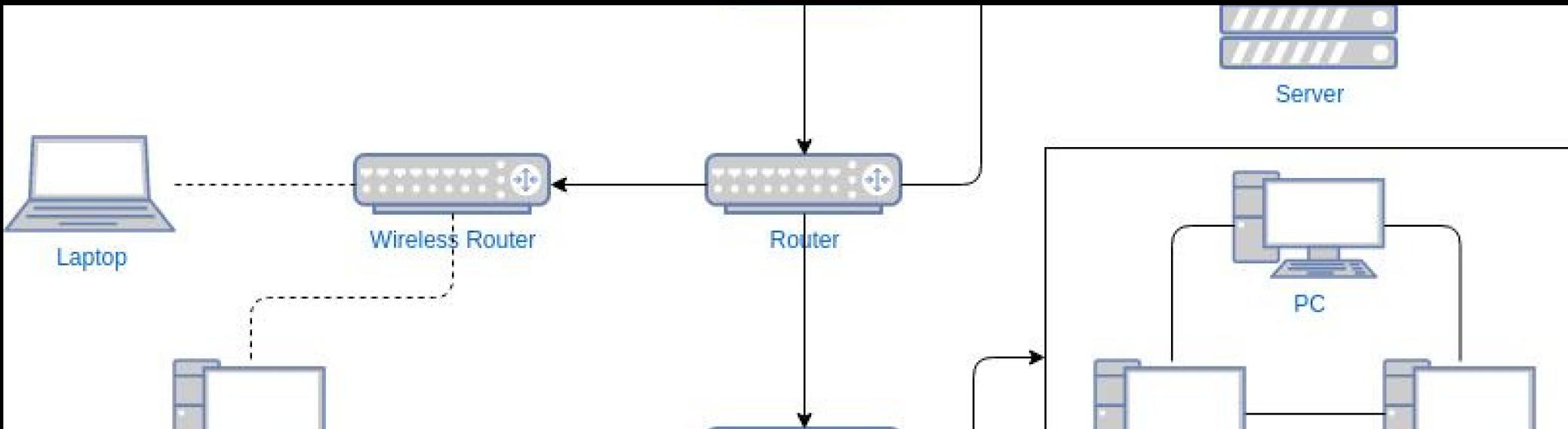
# Wait for the server to acknowledge the size
server_socket.recv(1024)

# Start measuring the time
start_time = time.time()

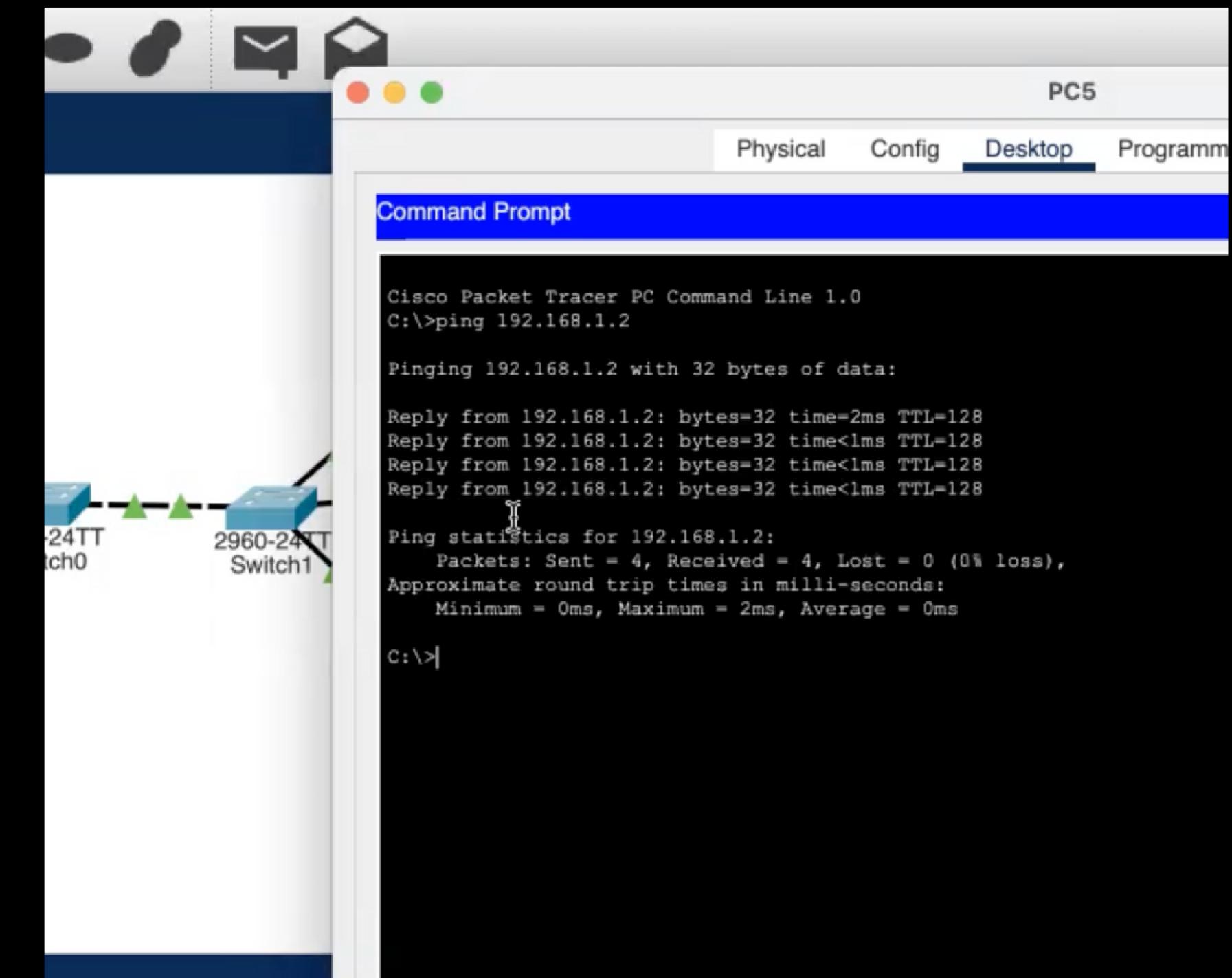
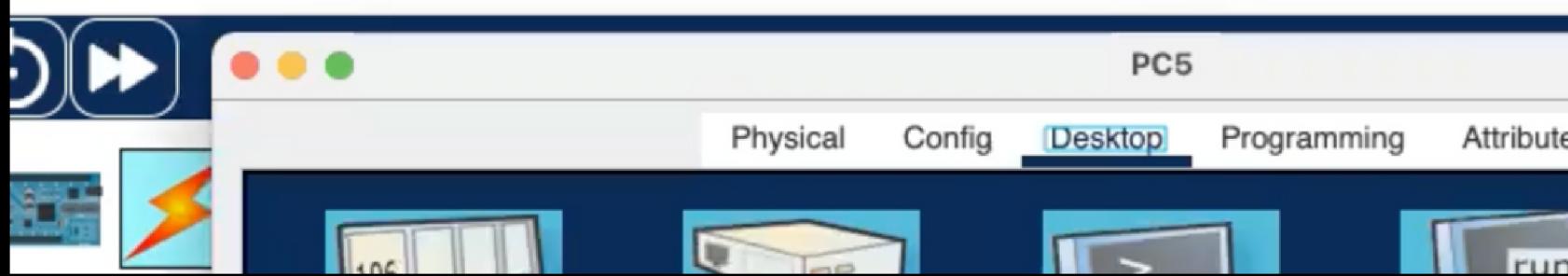
# Send the data to the server
server_socket.sendall(data_bytes)

# Wait for the server to acknowledge the data
server_socket.recv(1024)
```

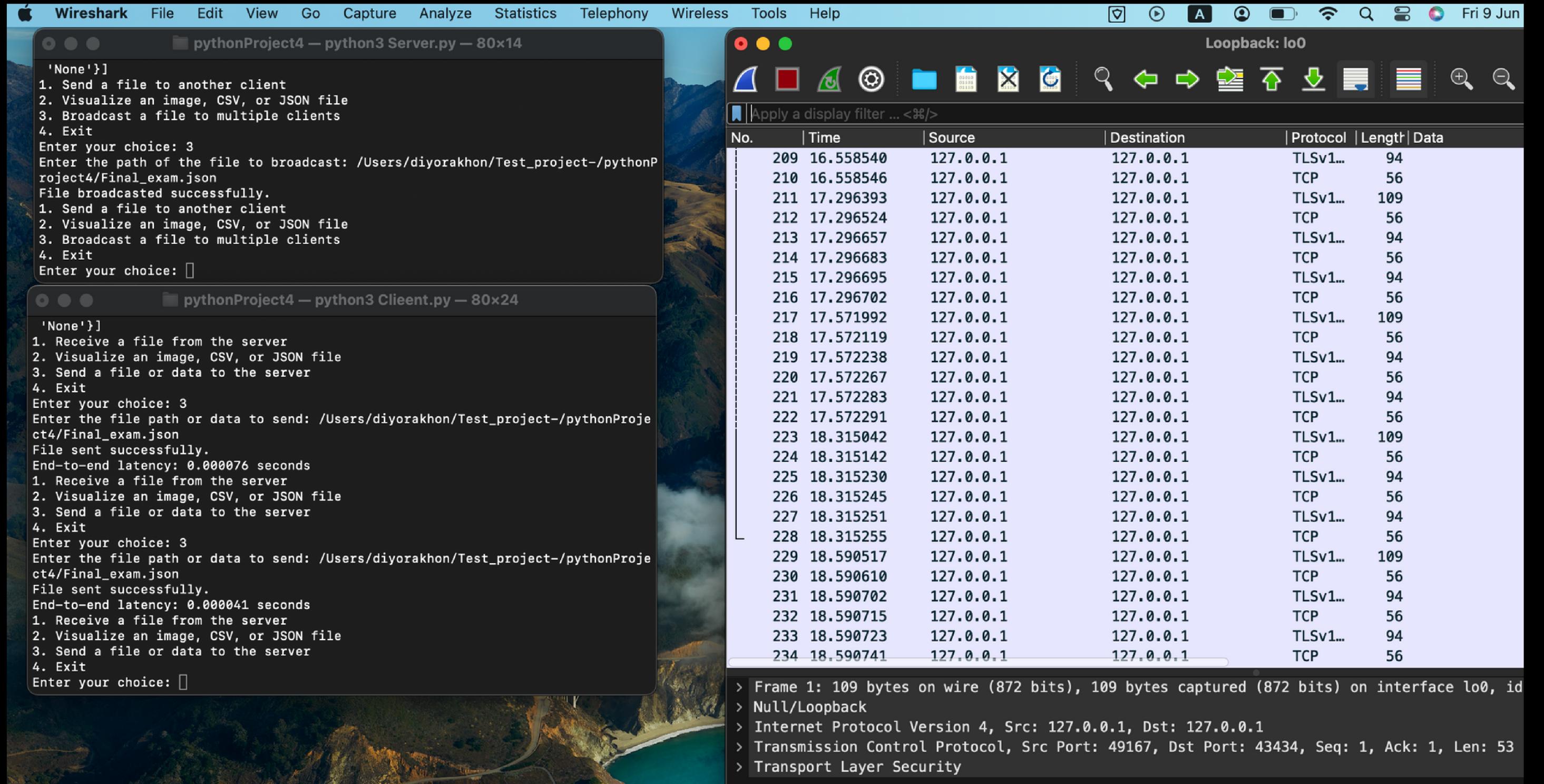
# WELL COMMENTED



Physical x: 633, y: 145



**CISCO PACKET TRACER IMAGE AND FILE FOR THE PROPOSED P2P SYSTEM  
+ VIDEO**



## WIRESHARK EXPERIMENTAL RESULTS, DESCRIPTION AND IMAGES

Enter the path of the file to broadcast: /  
roject4/Final\_exam.json  
File broadcasted successfully.

```
def send_data(file_path, server_socket):
    # Check if the input is a file path
    if os.path.isfile(file_path):
        # Send file to the server
        with open(file_path, 'rb') as file:
            file_data = file.read()
        # Get the file size
        file_size = os.path.getsize(file_path)
        # Send the file size to the server
        server_socket.send(str(file_size).encode())
        # Wait for the server to acknowledge the file
        server_socket.recv(1024)
    # Start measuring the time
    start_time = time.time()

    # Send the file data to the server
    server_socket.sendall(file_data)

    # Wait for the server to acknowledge the file
    server_socket.recv(1024)

    # Calculate the end-to-end latency
```

```
end_time = time.time()
latency = end_time - start_time

print("File sent successfully.")
print(f"End-to-end latency: {latency:.6f} seconds")
else:
    # Send data to the server
    # Convert the data to JSON format
```

```
ENTER THE FILE PATH TO SEND. /USER/S/RIGHT/FILE
File sent successfully.
End-to-end latency: 0.000049 seconds
1. Receive a file from the server
```

## CALCULATING THE END TO END LATENCY FOR EACH SCENARIO AND HAVING A TIMEOUT FOR SENDING A FILE