

# Project Report

## SHAREKARO:A data sharing platform

**Subject: PCS-II**

**Course Code: CSL2090**

**Contributors:**

- **Atharva Date (B22AI045)**
- **Saumitra Agarwal (B22AI054)**

## Overview

The project aims to build a software that fulfills various user requirements of sharing messages, file transfers and monitoring Network traffic all over your local Wifi network.

Note-This project itself extracts your IP address. If you want to change it please change in host\_ip\_adress variable in main function.

## LIBRARIES USED:

- OS
- SOCKET
- PSUTIL
- TIME
- THREADING
- SHUTIL

## Features

1. Peer-to-Peer Messaging: Users can send and receive texts in real time with peer-to-peer messaging. UDP sockets are used to send and receive messages, which makes connection light and quick.
2. File Transfer: Using TCP links to send and receive files over local wifi network. You can use this function to send files to other devices on the same network. It also incorporates acknowledgement, thus making it reliable to use and efficient.
3. Network Monitoring: Monitors the Wi-Fi interface for anomalies in network traffic. It stores bytes sent and received, packets sent and received, and so on. A variable threshold based on previous traffic is used to find anomalies. Also all the data can be real time monitored from a File name Network.csv attached with the project.

```
SHAREKARO:Next Generation Sharing Platform
Choose the action:
1. Peer to Peer Message Transfer
2. Send Files
3. Receive Files
4. Monitor Network over your WIFI!!
5. Exit
Enter choice (1-5):
```

## Implementation Details

Peer-to-Peer Messaging: This feature is for real time 2 way messaging between two peers in the same local network.

- **Functionality**: The `start_peer` function initializes a UDP socket and binds it to a specified host and port. It then starts two threads: one for sending messages and another for receiving messages. The `send_messages` function allows users to input messages to send, while the `receive_messages` function listens for incoming messages and prints them to the console.
- **Key Features**:
  - UDP-based communication for lightweight messaging through Datagrams.
  - No complexity of sending and receiving Acknowledgements.
  - Threading for concurrent message sending and receiving.
  - Disconnection handling with a "quit" command.
  - After each disconnection the user is brought back to home screen if wants to further use any feature or leave out.

```
SHAREKARO:Next Generation Sharing Platform
Choose the action:
1. Peer to Peer Message Transfer
2. Send Files
3. Receive Files
4. Monitor Network over your WIFI!!
5. Exit
Enter choice (1-5): 1
Enter your port number: 1234
Enter server IP address: 172.31.19.251
Enter target port number: 1233
Peer started, listening on 172.31.34.100:1234
Enter message to send (or 'quit' to exit): hi
Message sent to ('172.31.19.251', 1233): hi
Enter message to send (or 'quit' to exit): hello
Message sent to ('172.31.19.251', 1233): hello
Enter message to send (or 'quit' to exit): Received from ('172.31.19.251', 1233): hi man atharva here
quit
Messaging session ended. Returning to main menu.
SHAREKARO:Next Generation Sharing Platform
Choose the action:
1. Peer to Peer Message Transfer
2. Send Files
3. Receive Files
4. Monitor Network over your WIFI!!
5. Exit
Enter choice (1-5):
```

## File Transfer- It has two options to Send files and to Receive files.

- **Functionality:** The `get_my_ip` function automatically fetches your ip address and then the `send_files` function establishes a TCP connection to a server and sends a list of files. The `receive_files` function listens for incoming file transfers and saves them to the local filesystem. There is acknowledgement after every transfer.
- **Requirements:**

For Sending file: Sender's available Port number, Receiver's IP address and receiving port number.

For receiving file: Sender's available port number.

- **Key Features:**
  - TCP-based communication for reliable file transfer.
  - Blockages ensure small incoming values like file size to ensure no overloading.
  - File size and name transmission to ensure correct file reconstruction.
  - Acknowledgment mechanism to ensure successful file transfer.

```
SHAREKARO:Next Generation Sharing Platform
Choose the action:
1. Peer to Peer Message Transfer
2. Send Files
3. Receive Files
4. Monitor Network over your WIFI!!
5. Exit
Enter choice (1-5): 3
Enter server port number: 4444
Server started at 172.31.34.100 on port 4444
```

### File Receiving

```
Choose the action:
1. Peer to Peer Message Transfer
2. Send Files
3. Receive Files
4. Monitor Network
5. Exit
Enter choice (1-5): 3
Enter server IP address: 172.31.49.70
Enter server port number: 5555
Server started at 172.31.49.70 on port 5555
Connection from ('172.31.34.100', 51082)
File reciever.py received successfully
Choose the action:
1. Peer to Peer Message Transfer
2. Send Files
3. Receive Files
4. Monitor Network
5. Exit
```

### File Sending

```
Choose the action:
1. Peer to Peer Message Transfer
2. Send Files
3. Receive Files
4. Monitor Network
5. Exit
Enter choice (1-5): 2
Enter file paths, separated by commas: reciever.py
Enter server IP address: 172.31.49.70
Enter server port number: 5555
Connecting to the server...
File reciever.py sent successfully.
Choose the action:
1. Peer to Peer Message Transfer
2. Send Files
3. Receive Files
4. Monitor Network
5. Exit
Enter choice (1-5): █
```

## Network Monitoring

- Functionality:
  - `get_wifi_interface`: Finds the Wi-Fi interface.
  - `get_wifi_stats`: Retrieves network statistics for the Wi-Fi interface.
  - `print_wifi_stats`: Prints the statistics to the console, highlighting anomalies if present.
  - `write_to_csv`: Writes the statistics to “Network.csv” file, truncating it every 5 seconds and refreshing the value..
  - `detect_anomaly`: Compares current and previous statistics to detect anomalies. If any anomaly is detected then that particular entry is made visible in black fonts.
  - `monitor_network`: Main function running an infinite loop to monitor the network, print stats,
- Key Features:
  - Real-time monitoring of Wi-Fi interface statistics.
  - Anomaly detection based on significant changes in rate of bytes sent and received.
  - CSV logging of network statistics for analysis and realtime updation of CSV file to give better understanding to user.
  - Threshold for Anomaly detection in bytes can be self set in code.

```
SHAREKARO:Next Generation Sharing Platform
Choose the action:
1. Peer to Peer Message Transfer
2. Send Files
3. Receive Files
4. Monitor Network over your WIFI!!
5. Exit
Enter choice (1-5): 4
Monitoring Wi-Fi interface: Wi-Fi
To close monitorring press Ctrl+C

For a better view please view the out in Network.csv file
Bytes sent: 26363225
Bytes received: 52532556
Packets sent: 36514
Packets received: 149495
Bytes sent: 26363869
Bytes received: 52535405
Packets sent: 36516
Packets received: 149516
```

```
Time,Bytes_sent,Bytes_received,Packets_sent,Packets_received
22:25:12,26402886,52633977,36589,150142
22:25:13,26403713,52638005,36593,150173
22:25:14,26403755,52640669,36594,150193
22:25:15,26403755,52642987,36594,150211
22:25:16,26403755,52646755,36594,150235
```

Anomaly Detection:

```
Bytes sent: 27087878
Bytes received: 53248577
Packets sent: 37312
Packets received: 154200
```

## Usage

The application provides a simple text-based interface for users to choose between different functionalities:

1. Peer to Peer Message Transfer: Users can start a peer-to-peer messaging session by specifying their host IP, port, target IP, and target port.
2. Send Files: Users can send multiple files to a specified server by providing the file paths and server details.
3. Receive Files: Users can receive files sent to their server by specifying the server IP and port.
4. Monitor Network: Users can monitor their Wi-Fi interface for network anomalies.
5. Exit: Users can exit the application.
6. **Instructions for usage are provided in the README.txt file.**

## Contributions

- Saumitra Agarwal: Responsible for the implementation of UDP messaging functionality and the network analysis component of the project.
- Atharva Date: Responsible for the development of the TCP file transfer feature, synchronization of project features and documentation.