

COMPUTER NETWORKS PROJECT

IP MESSENGER

IP Messaging, variously referred to as "realtime", "native", "in-app", or "over-the-top" messaging, describes the general use of TCP/IP (the Internet protocol) to provide messaging capabilities to mobile applications (typically via GSM / 3G / LTE or WiFi connectivity). In mobile development, IP Messaging is closely associated with instant messaging, geolocation tracking, mobile / push notifications, and IoT automation.

It is the primary form of user-to-user and application-to-user messaging in modern mobile applications and has several distinct advantages over technology like SMS / MMS. These advantages include:

1. Any device with an IP address may send and receive messages to anywhere globally over the Internet. In today's world, it's not difficult to see how far this idea went. Per-message fees do not apply. As such, IP Messaging is technically free apart from the indirect, and often negligible, cost of bandwidth usage. The term, "over-the-top", originates from this fact, as IP Messaging uses public Internet routing and skips over the fees imposed by telecommunication carriers.
2. Developers have complete control over messaging functionality and appearance as it can be embedded within the application UI (i.e. "in-app"). Similarly, users can enjoy the benefit of not having to leave the app to receive messages or notifications that would otherwise be served via SMS.
3. Any application – whether web, mobile, desktop, or embedded – can make use of IP Messaging, while the specific methods of implementation and technologies used will vary widely between its almost unlimited use cases ranging from basic chat apps to the next Uber. In short: anything the Internet can do, IP Messaging can do for your app, too.

P2P communication

Peer-to-peer (P2P) computing or networking is a distributed application architecture that partitions tasks or workloads between peers. Peers are equally privileged, equipotent participants in the application. They are said to form a peer-to-peer network of nodes.

Peers make a portion of their resources, such as processing power, disk storage or network bandwidth, directly available to other network participants, without the need for central coordination by servers or stable hosts. Peers are both suppliers and consumers of resources, in contrast to the traditional client-server model in which the consumption and supply of resources is divided. Emerging collaborative P2P systems are going beyond the era of peers doing similar things while sharing resources, and are looking for diverse peers that can bring in unique resources and capabilities to a virtual community thereby empowering it to engage in greater tasks beyond those that can be accomplished by individual peers, yet that are beneficial to all the peers.

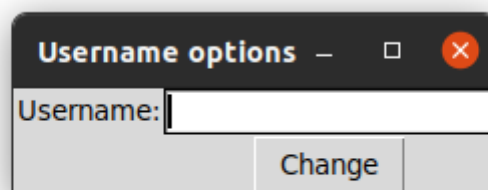
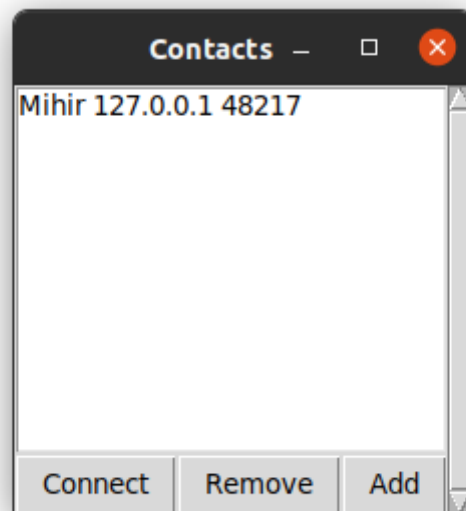
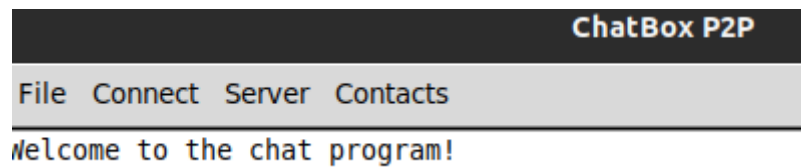
While P2P systems had previously been used in many application domains, the architecture was popularized by the file sharing system Napster, originally released in 1999. The concept has inspired new structures and philosophies in many areas of human interaction.

About ChatApp.py

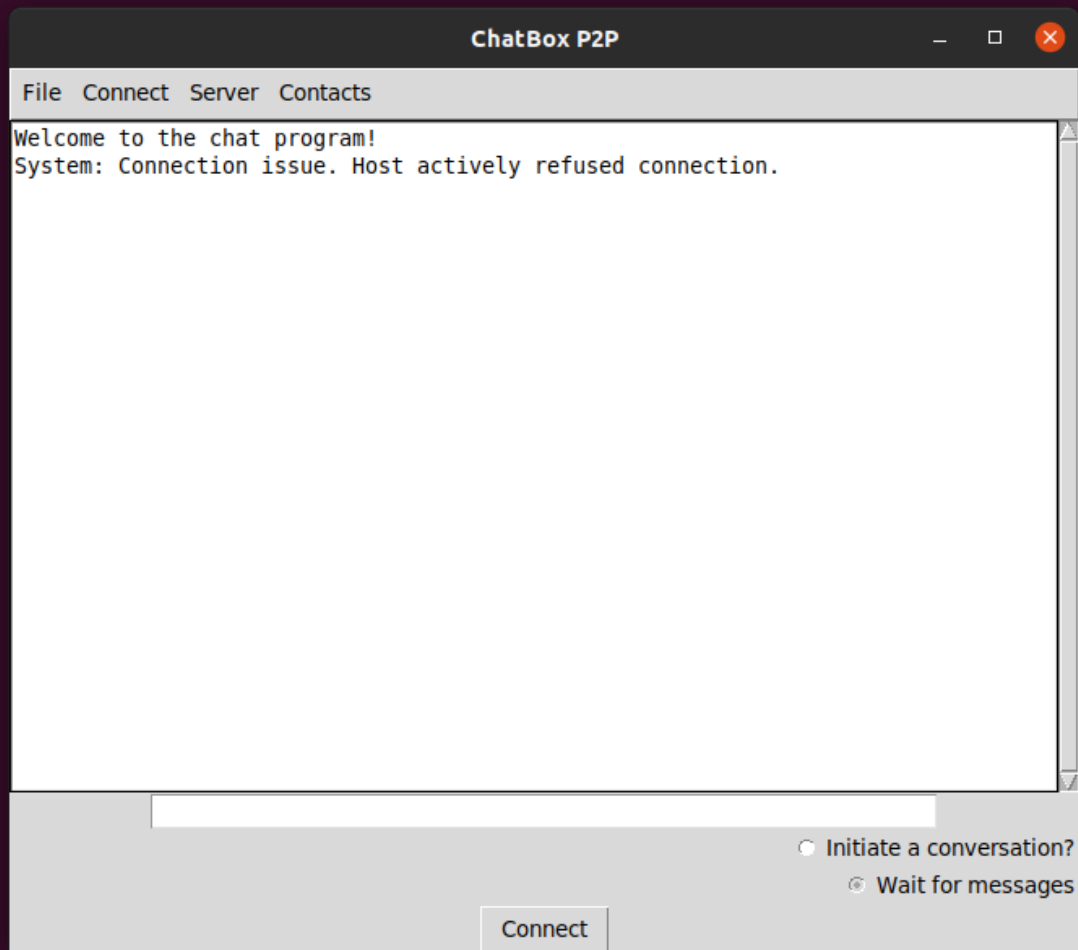
ChatApp is a simple client server architecture which implements a very minimal structure of IP Messenger.

Features:

1. Username specific messaging
2. Requires port only connection
3. Has a contacts.dat file set up on local storage that stores the IP and port of connections.



```
student@student-VirtualBox:~/Documents/Python-Chat-Program-master$ python3 pyChat.py
```



Contribution Analysis:

- 1. SUJOY DATTA(CED18I063)**- Setting up the mainframe serverless architecture through socket programming in a multithreaded fashion for simultaneous execution of app.
- 2. MIHIR SHRI(COE18B064)**- Tkinter GUI implementation for effectiveness and adding the feature of contact adding in contacts.dat file.
- 3. PRATEEK AGRAWAL(CED18I040)**- Setting up runner function to provide additional features like saving chats in a txt file and quick peer communication with subordinate Ips.