Local Peer-to-Peer Chat

System Design

Victoria Kogan

7886506

COMP 3010

Distributed Computing

Rob Guderian

Winter 2023

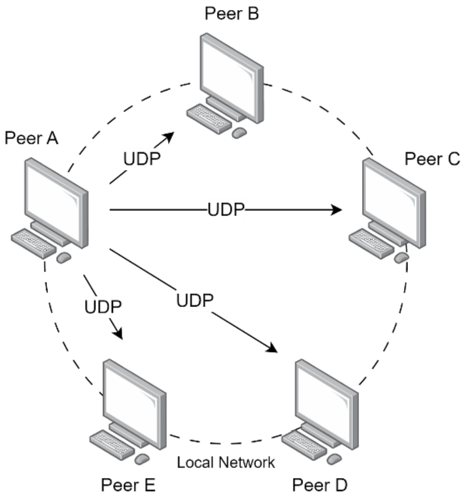
University of Manitoba

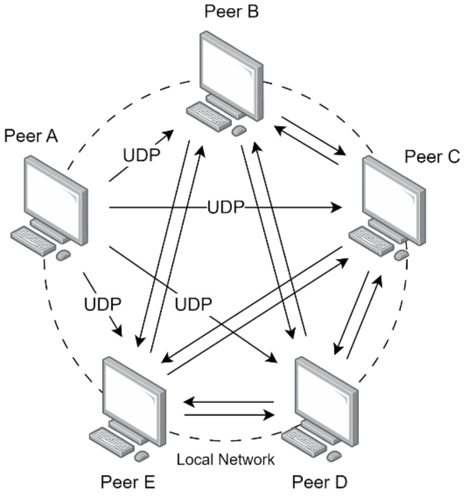
Overview

This peer-to-peer chat service for local networks will be designed to enable users to communicate with other users on the same network. A centralized chat system is not always practical since there is a single point of failure, therefore this system will be decentralized. The service will advertise itself and discover other users that are currently online. When a user decides to have a chat with someone specific online, the service will establish a connection with that user. Once the connection is established, the users can then send each other messages. The chat service will be designed to allow for multiple concurrent chats between users, which will be implemented using threads.

Advertisement and Discovery:

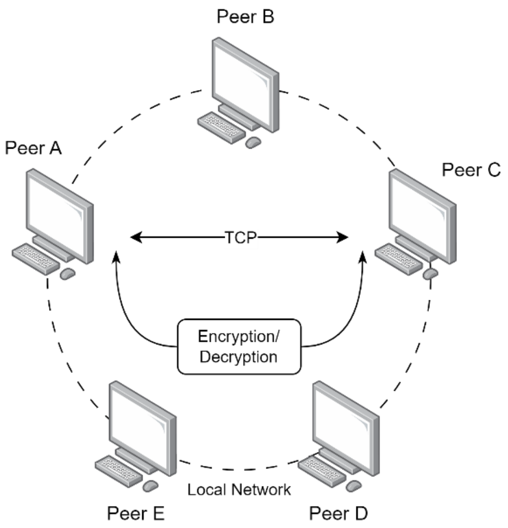
The client application will first advertise itself in the local networks and discover other users currently online in the local network. This will be implemented by multicasting UDP packets. Each user will be multicasting their own packet letting other users in the local network know they are online, as well as listening to other users' packets to discover who is online. As UDP packets can get lost, flooding will be introduced to eliminate that. As flooding can result in low network traffic and potentially cause congestion in larger networks, time to live (TTL) will be added to each packet. The type of TTL is not yet decided (time or number of hops).

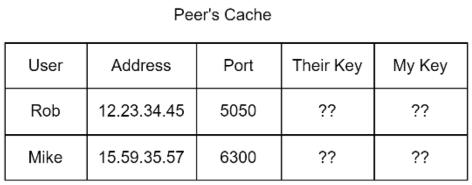




Direct Communication:

Once the user is found and wants to start messaging another user in the network, the service will initiate a TCP connection with the other user's IP and port number. As messages in the system will be encrypted using asymmetric encryption, public RSA keys will also be exchanged and cached during the connection establishment process. Moving forward, the sender will encrypt their message with the receiver's public key, and the receiver will decrypt the message with their own private key that only they know of. This eliminates third-party attacks as even if the attacker manages to obtain the public key for a user, they will not be able to decrypt it using the public key alone. If time permits, messages that have not been acknowledged may be stored in a queue, and once the user is back online or the connection is back, the queue will then be sent to the recipient. This will require more authentication.




Peer's Cache

User	Address	Port	Their Key	My Key
Rob	12.23.34.45	5050	??	??
Mike	15.59.35.57	6300	??	??

Resources:

Decentralised Indexed Based Peer to Peer Chat System by Anirban Kundu

Service Discovery in ADHOC Networks by G.Naga Satish, Ch.V.Raghavendran, G.JoseMoses, Prof.P.Suresh Varma, and SNVSST Murthy