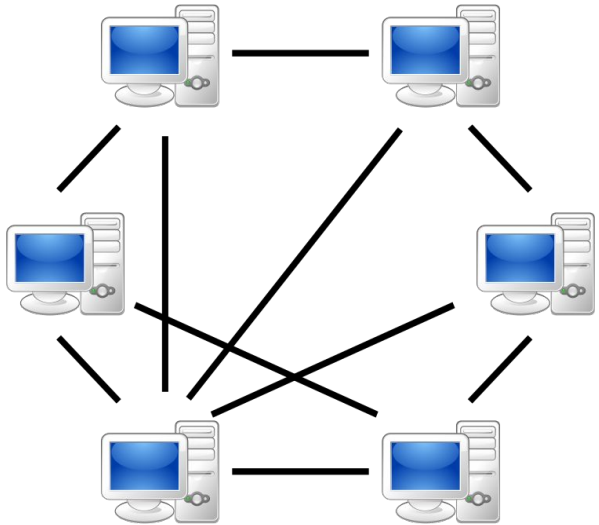
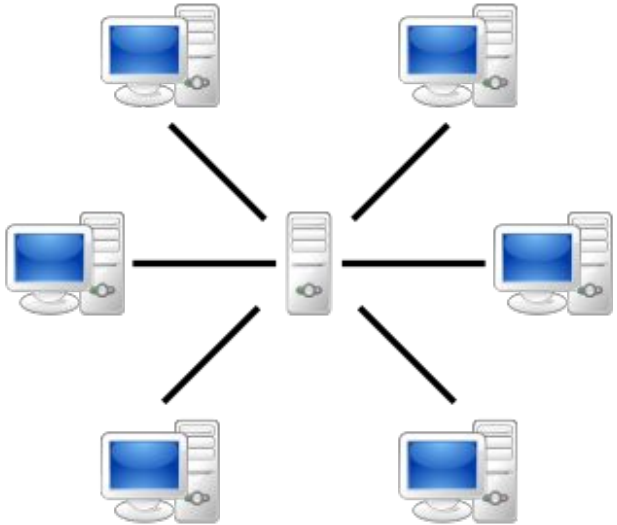
Secure Peer to Peer Messaging Hackathon

Peer to Peer Network vs Client Server Architecture





Applications

- Communications (Voice, video and Chat)
- Resource exchanges (e.g., files)
- IoT devices
- Blockchain

References

Peer to Peer Based Social Networks: A Comprehensive Survey

Highlights

- As a team, you can choose:
  - Application: WEB, python application, or mobile application
- No Chat data will be stored in a server
- Class is a working environment
- Create your own GitHub

Major elements to Consider

System we will work on

- Data will be distributed. No Data will be stored in a central Server.
- All local data is stored on clients.
- All Data stored on the device will be secure.
- The key is different between different chats.

Discovery

One research topic for final project is to investigate different peer to peer systems and implement one

- Users can be discovered via a centralized server. This is to update IP address for peer to peer communications.
- The client needs to make sure the IP address in discovery server is correct. One way of doing it is to have a keep alive packet. If it is not received, the user is not available.
- When a user is discovered, clients can send them any updates or new chats. (Synchronization)

Session Initiation

- To meet a new user, you send a message.
- A user can block other users. This means they become not available to you.
- If you are connected to any user, you will be receiving messages from the user.
- You can mute a user for period of time. In that case, you will nto see their messages.

Communication and Synchronization

- When a user is discovered, other users can chat with that person
- When a user is discovered, content waiting to be sent from other clients are sent to that user. (Synchronization)

MVP (red is nice to have)

- Discovery
  - You register by making yourself findable by others (as in Telegram or Whatsapp)
- Session Initiation
  - Send a message to users who you want to connect with
- Communication and Synchronization
  - You can connect with any of your friends who is available now.
  - For users who are not available (offline), you can write messages, which are stored on your own client.
  - The data for offline users will be synchronized when both users are discoverable.
- Security (optional)
  - All is hashed

Nice Extra (can be part of final project)

- Multi-person (more than 2).
- Mute users.
- Share files and media.
- Distributed Hash Tables

Decisions you need to make today

- Architecture
- Local database to store data (e.g., SQLite)
- Tasks
- First Milestone to accomplish
  - Send messages between two clients
  - Design your database
  - Store data and retrieve it from a database
  - Hard code discovery

Working together

- It is an individual assignment
- Work in groups of two (brainstorm, solve problems, code reviews)
- But you have to deliver your own system

Finally

- Messages can be between devices. It does not have to be between people
- Think of the car project and/or healthcare system

Completion date: 4/7/2024
(Submission on Blackboard)

Thank you