Chat Application - Topology

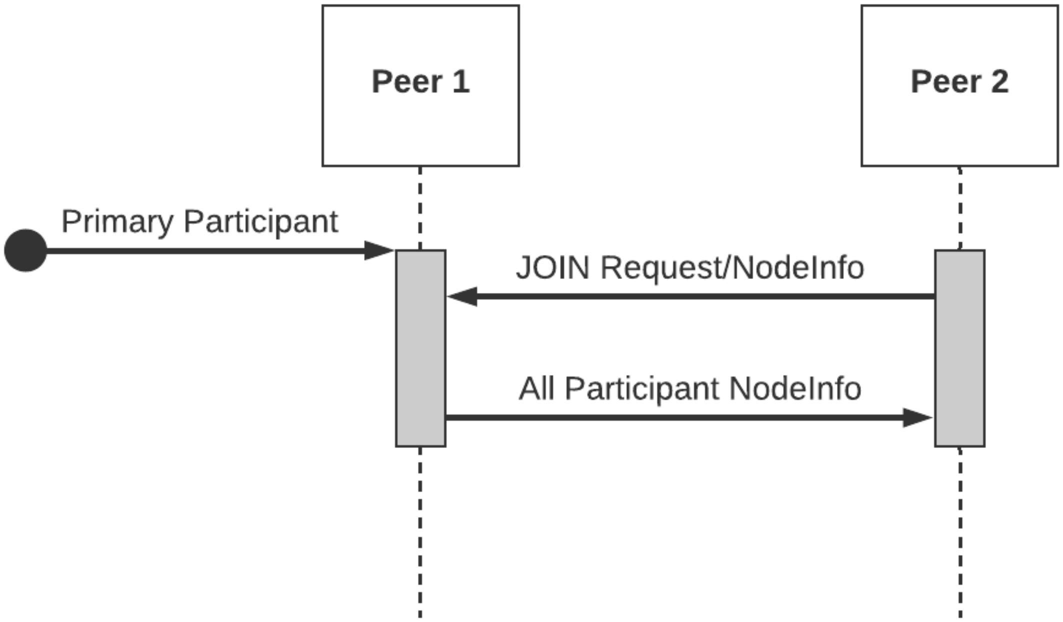
Joe Domabyl V
Daniel Rydberg
Nick Nannen

Overview

Like our previous project, there are a number of users (called peers), any of whom can join the chat at any time using a join request. Unlike our previous project, each user is both a client and a server, allowing them to directly connect to one another without using a server as a middleman, making this a true peer-to-peer chat. There is no central server. Once a user is connected to another, they can directly message each other and any other connected users. Such a network of interlinked clients is called a mesh topology, while one using a central server is called a star topology. In our peer-to-peer network, messages will be distributed from a single peer (the master) using a MessageQueue. Each peer will maintain records of who is connected via NodeInfo objects within an ArrayList. From there, peers will be able to interact with each other via a series of different message types, described below.

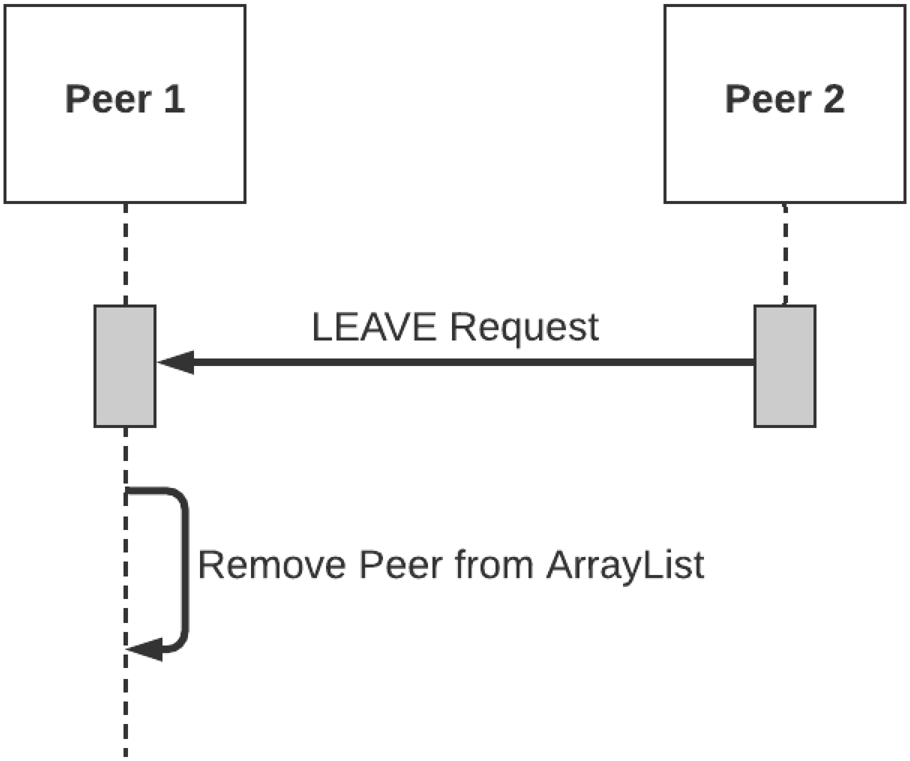
Scenarios

- JOIN



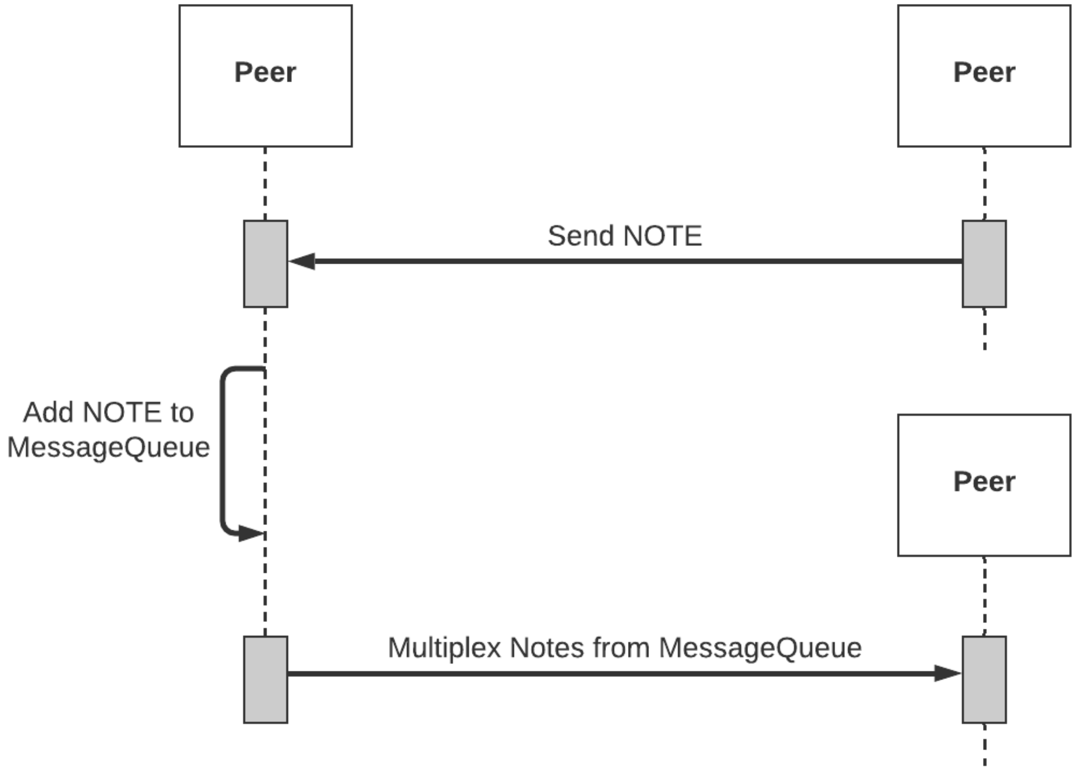
When a peer receives a JOIN request, that peer's NodeInfo is added to the ArrayList of active participants. Upon the addition, the receiving peer sends the JOINing peer its list of total active participants (which includes who is identified as the master peer), so they are kept up to date with who is connected.

- LEAVE



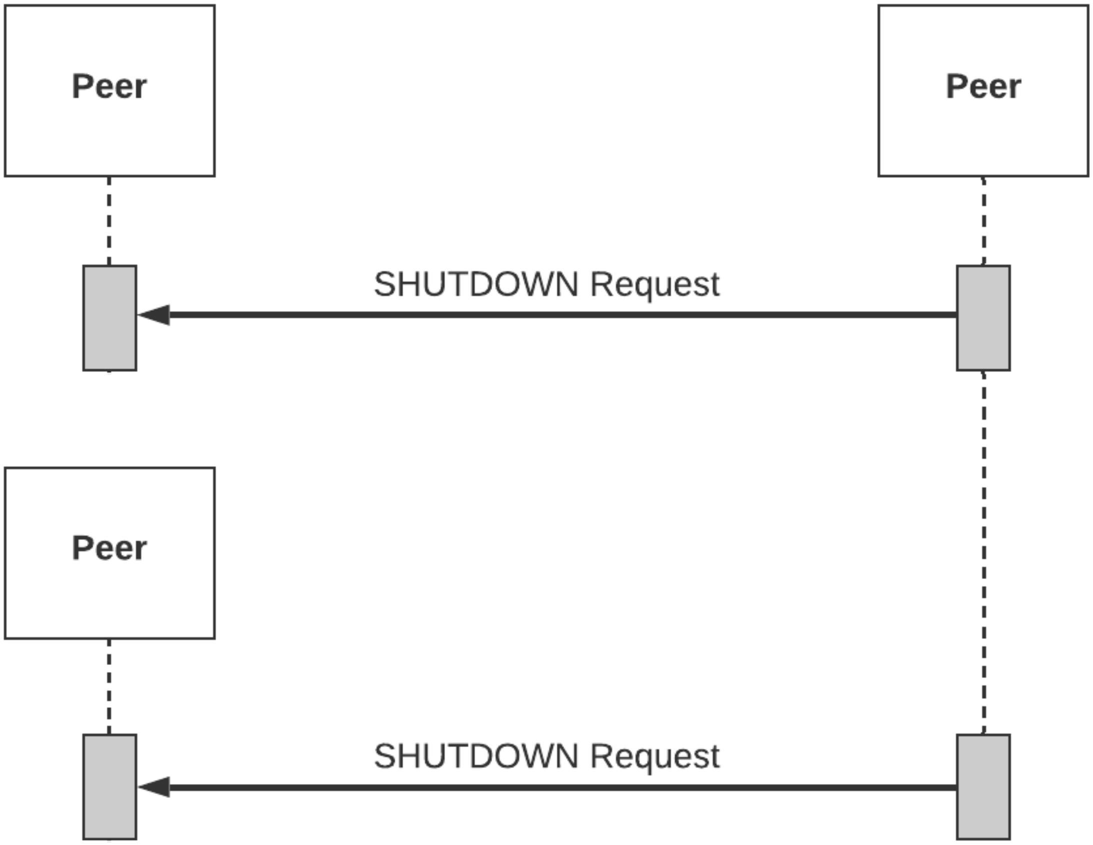
When a peer sends a LEAVE request, it sends it to all peers in the chat. When they receive the LEAVE request, they remove the user from their ArrayList of active participants. The LEAVE request contains NodeInfo data so that each peer knows who to remove from the list.

- Sending/receiving messages (NOTES)



Any JOINed peer can send a message (NOTE) to the rest of the connected peers. However, a peer sending a NOTE will only send it to the master peer. The master peer will store the NOTE in its own MessageQueue and begin multiplexing any stored messages to the remaining peers until the queue is empty.

- SHUTDOWN



Any peer can send a SHUTDOWN request. When a peer sends such a request, it sends it to every peer. The peers then disconnect from the chat and shut down completely, resulting in the entire chat application being shutdown.