One-page project proposal

Group 02: Ning Ding (3610727), Xin Pang (3111543), and Yang Fu(3595130)

1 Introduction

The goal is to build a distributed chat system. It allows users send messages to each other or in form of a group chat. The whole system are consists of servers and clients, it has to fulfill some properties such that the communication and organisation among the clients and servers is not ambiguous. e.g. the same message is only sent once. At beginning the components has to send broadcast messages to dynamically discover other components in the system.

2 Project requirements analysis

2.1 Architecture model

Client-Server model is used for the system. There are several servers and clients in the system. Each client is connected to one server and each server can have many clients. There is one leader among the servers, which takes care of the new participants and redistributes the servers to clients if some server fails. Clients are the users who participate in chat, servers are responsible to transfer the messages to their receivers i.e. other clients. Clients can be reconnected to the server with best performance by the leader at any time.

2.2 Dynamic discovery

The system is not restricted to a fixed number of clients/servers. Participants addresses should not be hard-coded. New clients and servers should be able to join the system properly. The first server initiates system configuration. When a new client or server comes online, they initially broadcast their address to other components and wait for reply.

2.3 Fault tolerance

If the leader fails, a new unique leader should be elected by the servers. If other server fails, the corresponding message flow will be assigned to other servers by the leader. If one message is missing, the sender has to discover that by using e.g. timeout mechanism, and resend that message to the corresponding receiver until it gets the corresponding acknowledgment. Every component creates a local log file that it updates on a regular basis. The log file could be used to restore a component in the event of a crash.

2.4 Voting

A leader among servers is chosen using an algorithm, and the leader is in charge of granting access to new components and updating the topology of network by exchanging "hello" messages to other components.

2.5 Ordered reliable multicast

The user receives the messages in the order, in which they had been sent by other users. One user can sent one message to different users. Besides that the casual ordering and the total ordering have to be fulfilled as well.

3 Architecture diagram

For a client to join the system, it has to register namely login by one of the server.
In one to one chat, the client can sent to a other client message, via a server, which is assigned by the leader of the servers.
In group chat, a group of clients share the same messages and broadcast messages within the group.

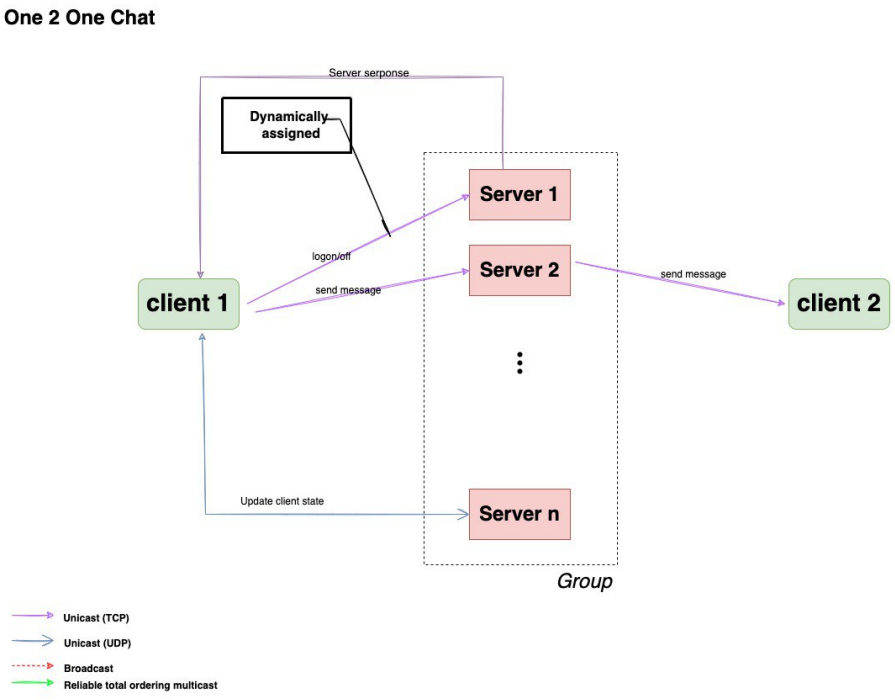

Fig. 1. Architecture diagram of one to one chat.

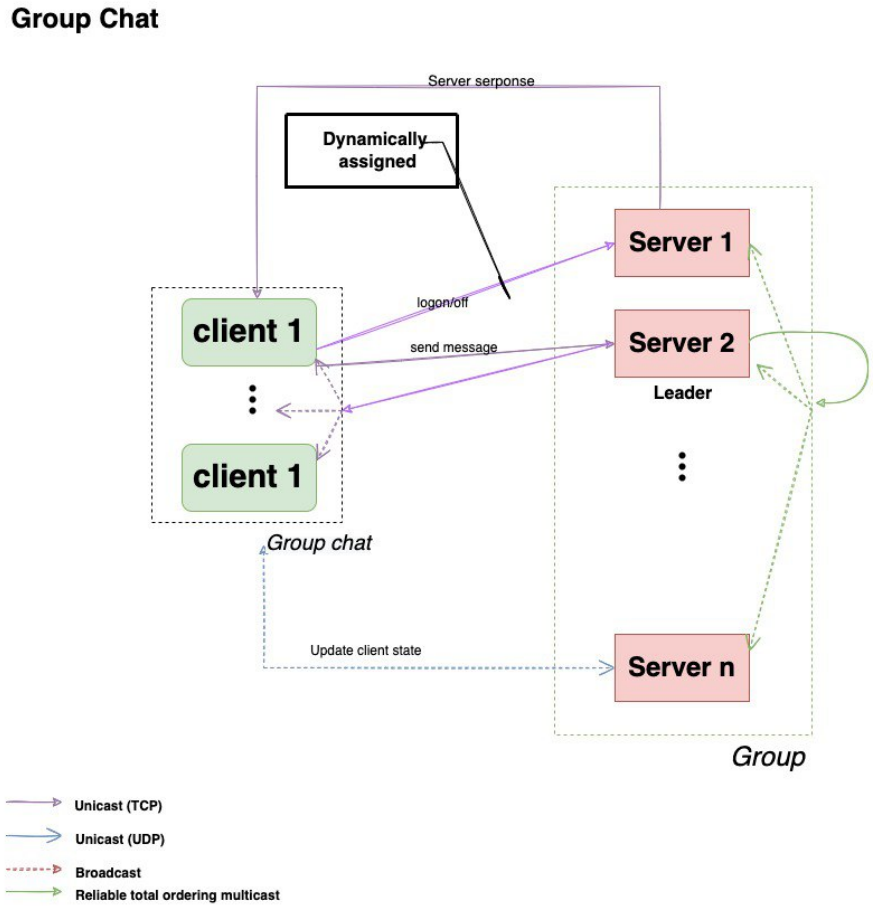

Fig. 2. Architecture diagram of group chat.