Distributed Computing Systems: Assignment #3

Due on May 27, 2021 at 11:59pm

Professor: Öznur Özkasap
TA: Waris Gill

Muhammet Soytürk

Implementation and Screenshot

Each peer executes a file named "peer.go" with the following command:

`go run peer.go <IP address/Port Number>`

The first thing that a peer does is to read *group.txt* which contains the information of all peers in the system. It parses the file line by line and add each peer to a string slice if the ID is not equal to itself.

Once a peer gets the information of other peers, it registers the RPC api that can be called by peer processes.

The next step is to try to connect to all other peers. The peer iterates over all peers and try to connect each peer. If it cannot connect for some reason, it tries to connect to the next peer in the slice. It tries to connect a peer every second until all peers are connected.

When a peer connects to all other peers, it waits for the message to be sent to other peers. When a message is entered, it creates a new Message struct which consists of the transcript of the message, sender identifier and sender sequence number of the message. After the message is constructed, the message is sent via a remote procedure call to all peers.

Printing format is the following: <peerID>:<tab><Message Transcript> (<sender sequence number>)
For example, format of the first message of peer 1 (left top command line) is:

172.31.57.62:9090: Hi I am peer 1 this is my first message (1)