ASSIGNMENT 4

Course- CS 549

Yugaank Arun Sharma

Canvas Id – ysharma

Date of Submission: 11/17/2017

In an earlier assignment, you developed a simple peer-to-peer distributed hash table for sharing bindings of keys to values. Each node in the network runs a CLI "shell" that allows bindings to be added and queried in the network. In this assignment, we will add the ability for one node's shell to remotely control another node. The communication to support this will be performed using Web sockets, using a very simple protocol where command lines are sent from a controlling client to a controlled server, and the server sends responses back to the client

1. To simplify how the assignment works, the "proxy shell" on the client just copies lines of input from the console to the server, and displays lines of response from the server on the console.
2. The prompt that the controlling client sees in the user interface is just a string sent back to it to be displayed on the console.
3. This assignment introduces three new operations to the CLI- accept, connect and reject.
4. Connect make a connection request to the node identified by the host name and TCP port number. On the remote node, if a control session is not already pending or in progress, then a new pending control session is registered, and the user at that node is notified that there is a pending control request. The user at the remote node uses one of the two following commands to respond.
5. Reject command reject the pending connection request. The connection to the client is closed, with an explanation that the control request has been terminated
6. Accept command accept the pending connection request. An acknowledgement (the string "ACK") is sent back to the client, to let them know that their control request has been approved.
7. If you want to quit from remote connection, just type quit command and the connection will be closed and you can work on your own machine.
8. The two important abstractions for this assignment are shell and context. Shell accepts the command lines and process these, producing the output. Where as context provides an operation for reading a line of input, and various operations for writing responses and reporting errors.

You can see the testing videos that demonstrates all the operations between 3 instances. There are 3 separate videos that demonstrates, basic accept, connect and reject, connect while connected and accept while connected operations.

In the zip archive, there is README.PDF, dht-remote.zip archive (assignment), videos demonstrating the testing and dht.jar file (server address as localhost)