Java Version Design

Thursday, April 25, 2019 1:20 PM

using DS skeleton

peer server design — Tree Struture

(server proj)
- → server (init)
- → connection
- → control
- → listener
- → settings (basis of info)

△ Roles:

* Listener: listening on the port assigned
creates 'connection' instances & adds to server connection list

* Control: Msg Processing
HeartBeat implemented
Body of server — Queue may implement here
Manipulating on connections — init, create, close

* Connection: Definition of 'connection' in this proj.
Contents: socket
socket-relevant buffers

△ Flows:

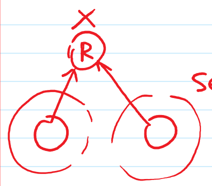
$$\begin{tikzcd}
\text{init} \arrow[d] & & \\
\boxed{\text{server}} \text{ extract cmd / settings} \arrow[d, "\text{thread}"] & & \\
\boxed{\text{control}} \arrow[rr, "\text{activate}"] \arrow[d] & & \boxed{\text{Listener}} \\
\boxed{\text{run()}} \text{ heart beat implementation} & &
\end{tikzcd}$$

△ MSG Format:

* using other msg to detect server status may fail
situation: no client input detected.

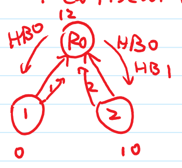
* abstract basic MSG: *Type { SYNC, CLIENT, *HB ... }
*src-host, *dest-host,
*src-port, *dest-port,
get ×4
set ×4
*Body
get, set

△ Server Isolation



Server crashed

* HeartBeat MSG

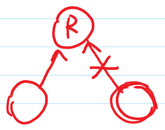


Forwarding another peer's HBx to any other peer

HB: Body msg → SocAddr

connection interrupted

HB MSG { srcHost
srcPort

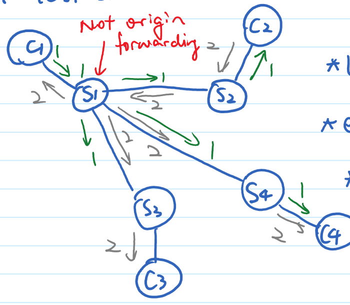


connection interrupted

VM - test

HB MSG {
- srcHost
- srcPort
- destHost
- destPort

client test suite



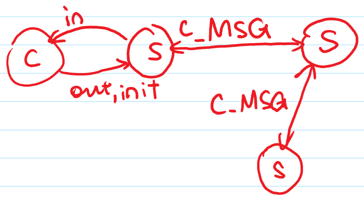
* let clients continuously send msg to server ✓

* each server has an ArrayList Buffer area — holding client msg received.

* each client has an ArrayList Buff, holding msg from other clients

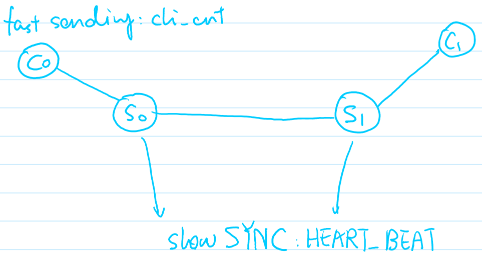
+ client_id: 'client_ip : client_port'

+ CLIENT_INIT : convey client id.



Test Case: 2 clients — 2 servers

fast sending: cli_cnt



try to maintain single client msg #

+ late clients should have full msg.