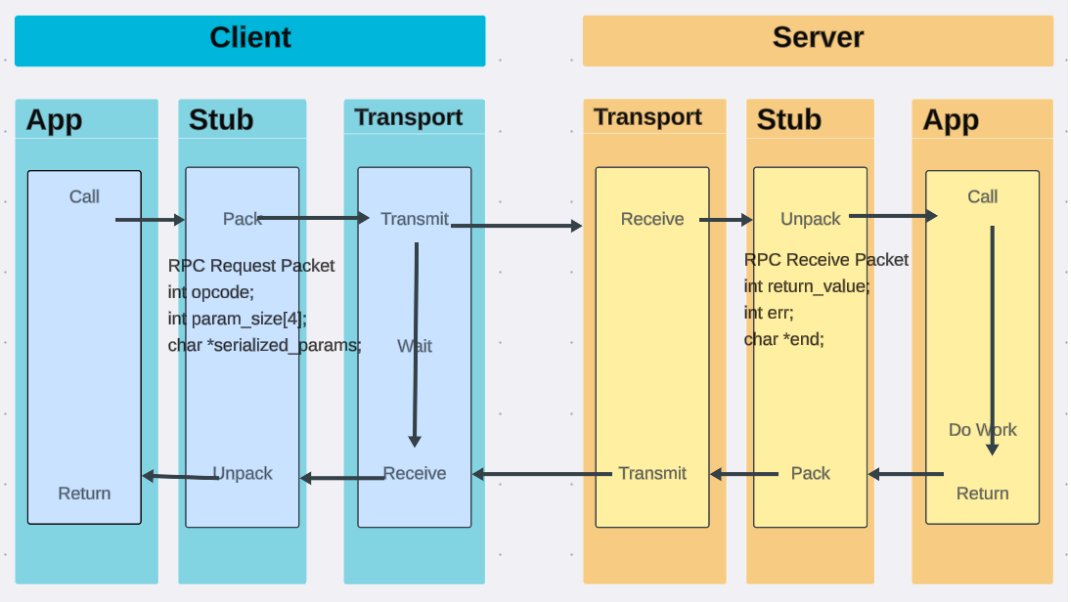
15-640 Project 1 Design Document

Siqi (Edward) Guo, siqiguo@andrew.cmu.edu, Feb 1, 2024

Serialization protocol

- How/what is sent/received? And how directory trees are handled?



- I implemented two packets for both RPC requesting and receiving. In RPCRequestPkt, there is an opcode to indicate the command, four variables to the size of four parameters, and marshalled data as a char* variable. In RPCReceivedPkt, there is a return value, an errno number, also a marshalled message as a char* variable. My sending and receiving strategies are sending or receiving the length of the following packet first, and continuing sending or receiving the whole packet.
- Specifically, the directory trees would be recursively traversed (pre-order) and the file names in this tree would be separated by the number of the file name's length and the number of its children. In this way, the pre-order traversal is only.

Handling concurrency

- I handled concurrency by fork a child process for each client. The client would build a socket connection, holding it for its lifetime. The fork() in the server would copy a current file descriptor table from its parent, which could avoid conflicts of the file descriptors among the server and clients. After forking, the child process would deal with one client's RPC in the background. Besides, I added OFFSET 1005 to the file descriptors in the server, so the client could distinguish between local calls and RPCs.