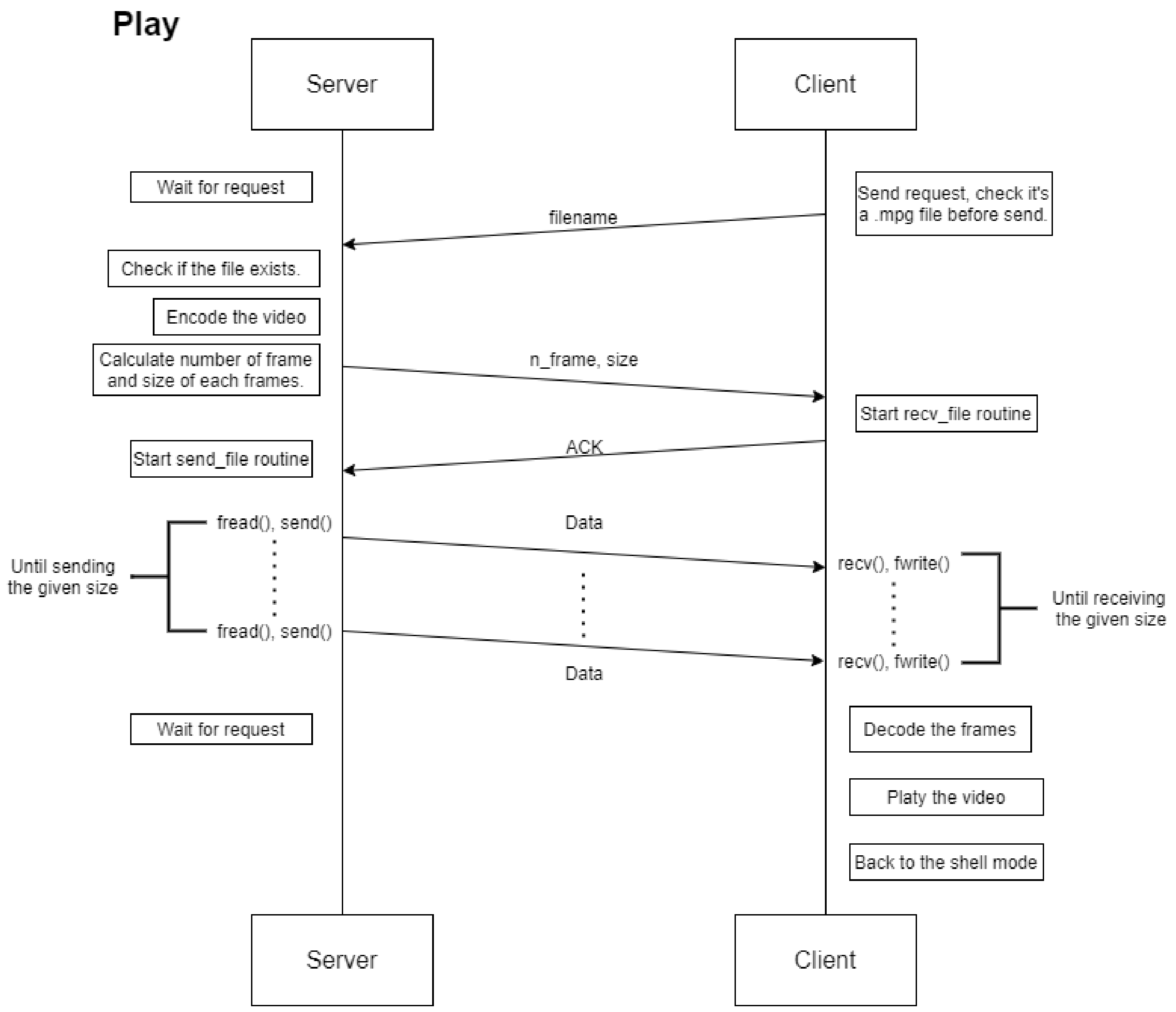
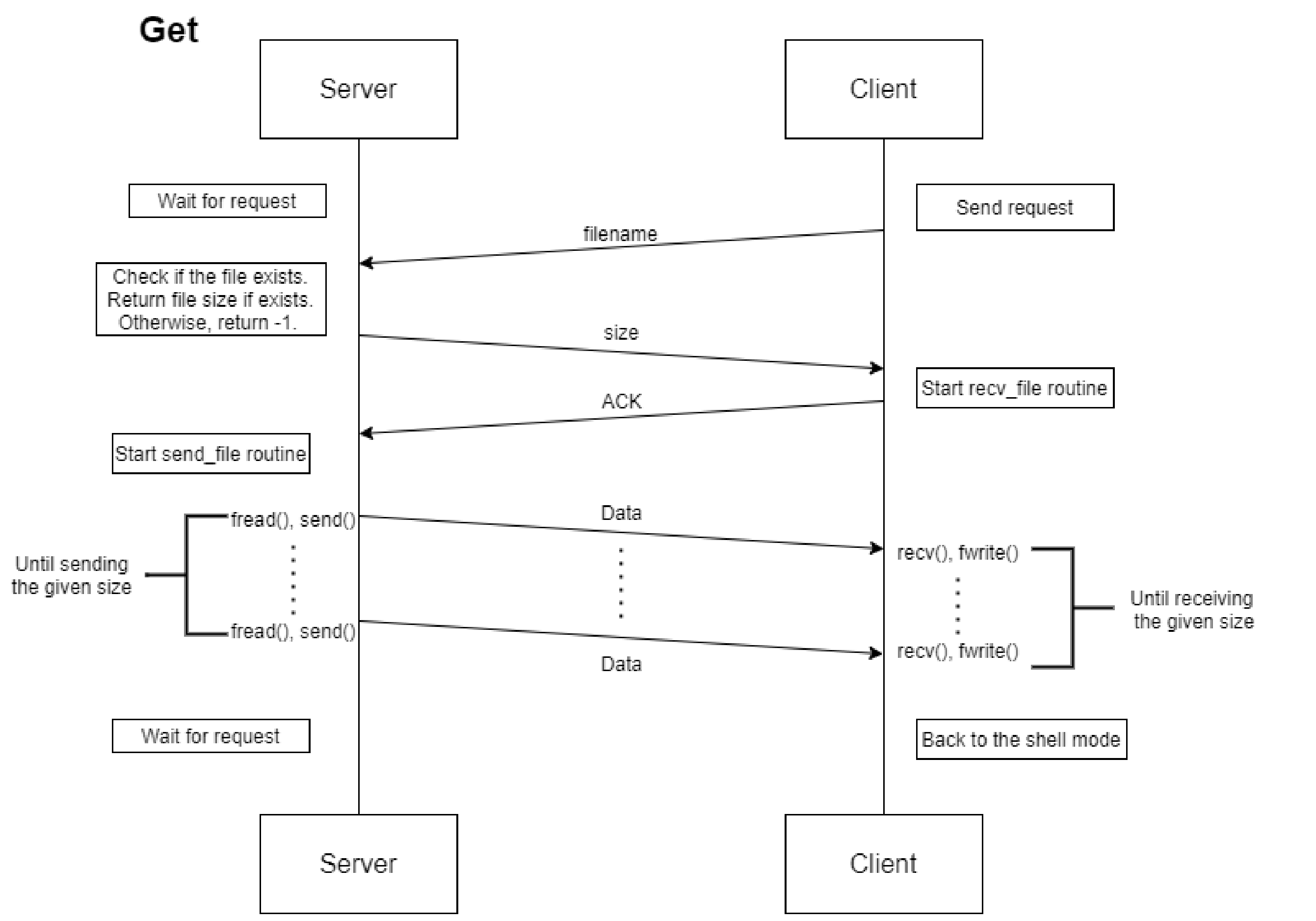
CN-2020 Homework2 Report B07902078 資工三 沈韋辰

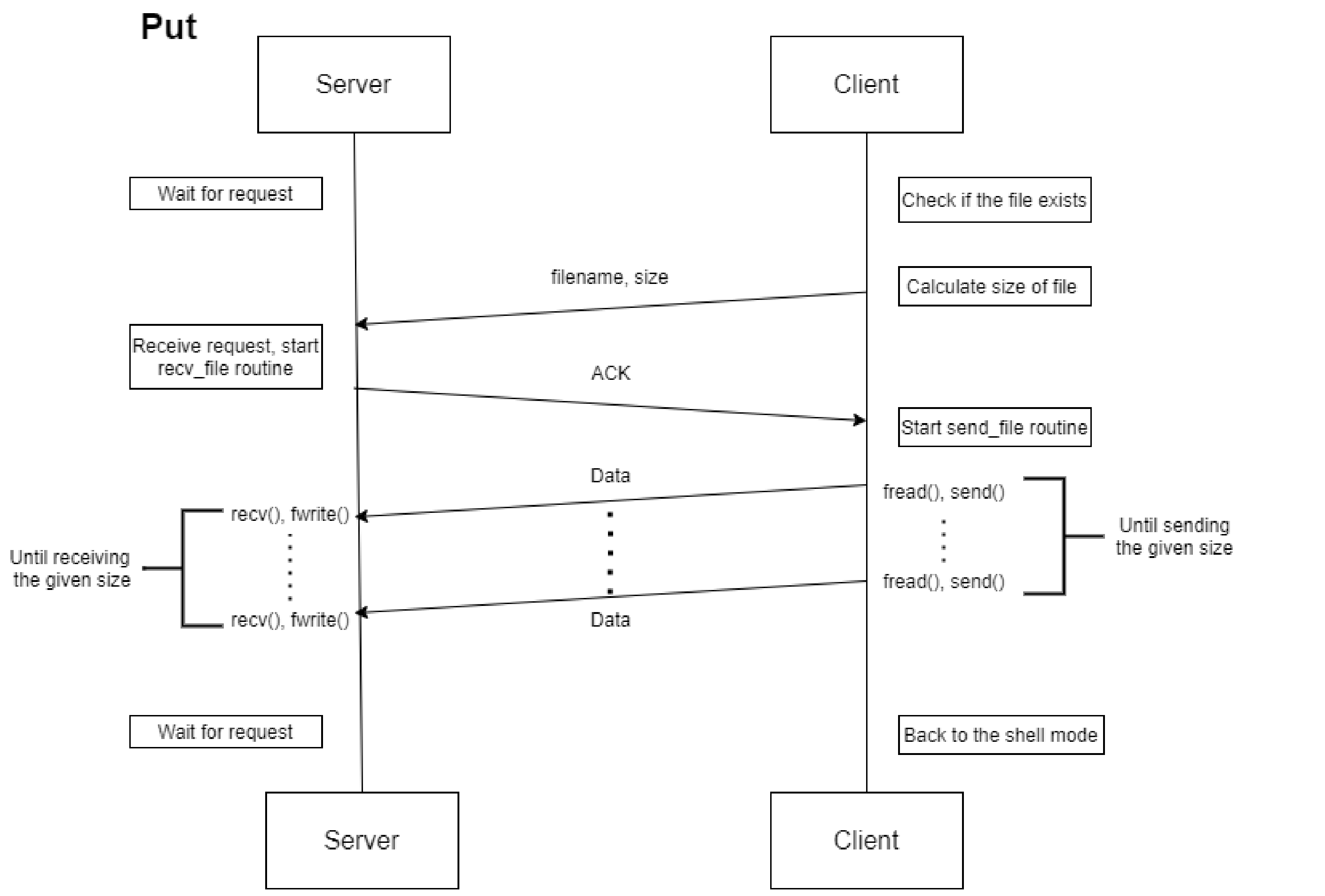
Draw a flowchart of the video streaming and explains how it works in detail.



In this flow chart, buffering doesn't be implemented because I think it is too hard and I don't have enough time to do it.

Draw a flowchart of the file transferring and explains how it works in detail.





What is SIGPIPE? It is possible to happen to your code? If so, how do you handle it?

SIGPIPE generates when reader of a pipe or socket connection is terminated. The default action of SIGPIPE is terminate.

In my code, when the client initiatively closes the socket to the server (in general, the connection should be close by server), server will receive the SIGPIPE and terminates by default. To solve this problem, I use the system call “signal()” and call a signal handler to do the SIGPIPE routine. The routine includes closing the file descriptor, FD_CLR(), etc.

Is blocking I/O equal to synchronized I/O? Please give me some examples to explain it.

No.

Blocking I/O means that the process is blocked when it is **waiting data** AND **copying data from kernel space**. Ex. read(), write().

Synchronized I/O means that the process is blocked when it is **copying data from kernel**. Ex. blocking I/O, nonblocking I/O, I/O multiplexing (select() / poll()).

So, **Blocking I/O** is a kind of **synchronized I/O**, they are not equal.