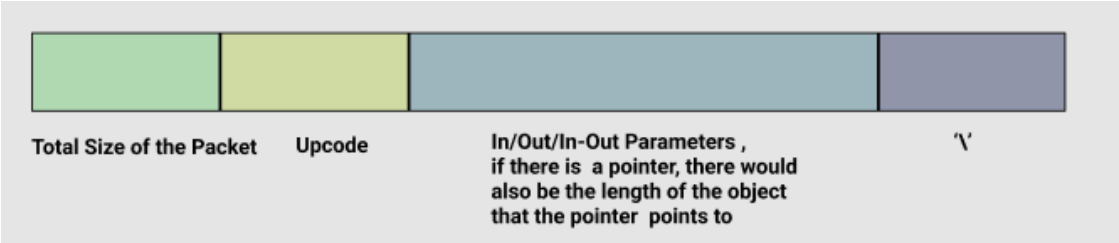
Design of the Serialization:
Packet Design:
All the packet sent to the server would be serialized as below. All the things server needs to know are serialized into a string. The string are stored in the local buffer in the client first, then send to the server as a whole packet. The first part of the packet is the the total size of the packet. The second part is the upcode. The third part is the In/Out/In-Out parameter. If the parameter is a pointer, the length of the object the pointer points to would also be packed in this part. The last part is a NULL which is used to end the string.

Rationale for the design:
TCP cannot guarantee all the content would be send to the server at one time, so sending the total size first would help the server to make sure the bytes sent equal to the bytes received. The upcode is sent in the second place because the upcode can help to switch to different deserialization methods to get all the parameters out.



Other Design Choice:
1. The protocol also distinguishes the file descriptor between the client side and the server side by adding an OFFSET to the server file descriptor so that file descriptor on either side would not mix with the other.

2. Use memcpy() to copy data into the buffer to make sure all bytes are concatenate together.