Protocol

My protocol consisted of 3 functions: send(), recv(), recv\_packets().

1)send() - data sending function. it receives two variables: socket and request.

1. socket - name of the socket, to which we make sending.
2. request - data we want to send.

At first, function turns request into bytes, so request is able to be sent. Then function packs request’s length, using bytes type, so receive function can unpack request’s length and so get request. At the end function send all these together (packed length and request in bytes).

2)recv() - function for reading received data. it receives only one variable: socket.

socket - name of the socket, from which we receive data.

At first, function receives sent data, using function recv\_packets(), but this data is packed. It unpacks data’s length and then returns decoded data, which it received from recv\_packets() function, because it knows what is length of request now.

3)recv\_packets() - helper function that returns decoded data specified length. it receives two variables: socket and n.

1. socket - name of the socket, to which we make sending.
2. n - number of bytes, function needs to read and return.

At first function creates variable in type byte to fill it with received data. Then function makes loop until it gets a piece of data of the required length. It reads a section of length no more than what it lacks in required length. If it can’t get anything as it returns nothing in other cases function returns our variable filled with all received data.