Dylan Schneider, Roark Myers, Josias Polonia

Network Design

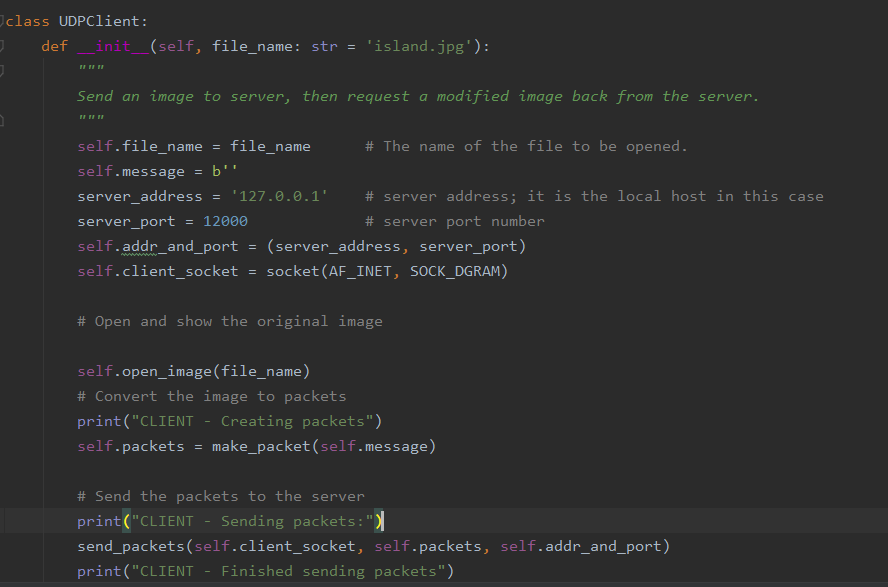
EECE 4830

Vokkarane

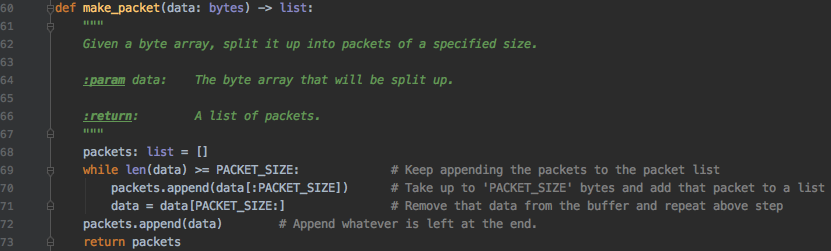
Network Design Project: Phase 2

The purpose of this phase is to transfer an image file such as a bmp between a UDP client and server like phase 1, but this time we provide reliable data transfer. With this method the image file transferred is split into several packets and those packets are sent over one at a time. The server (the first one to receive packets) will continue to read in the packets until it receives a terminator message; after that it converts the image to grayscale then repeats the above process to send the image back to the client. The client will accept packets until getting the terminator the same way the server does.

*Client:*

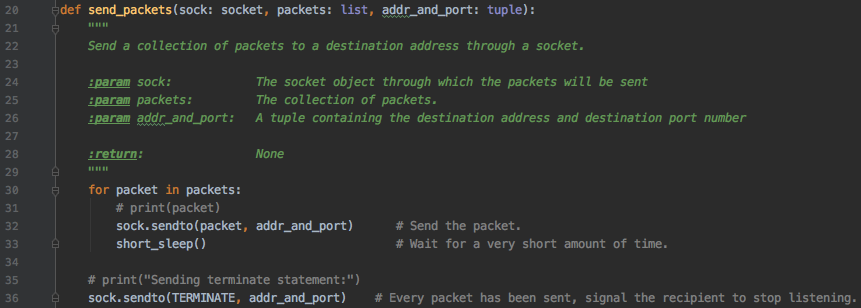


The UDPClient class creates a definition for the image file name prior to sending it to the server. After defining the address and port of the server, self creates access to the class. Line 17 assigns the client socket and address family as well as declares it as a UDP socket equal to the client socket. The image is converted to packets by the make\_packet function:

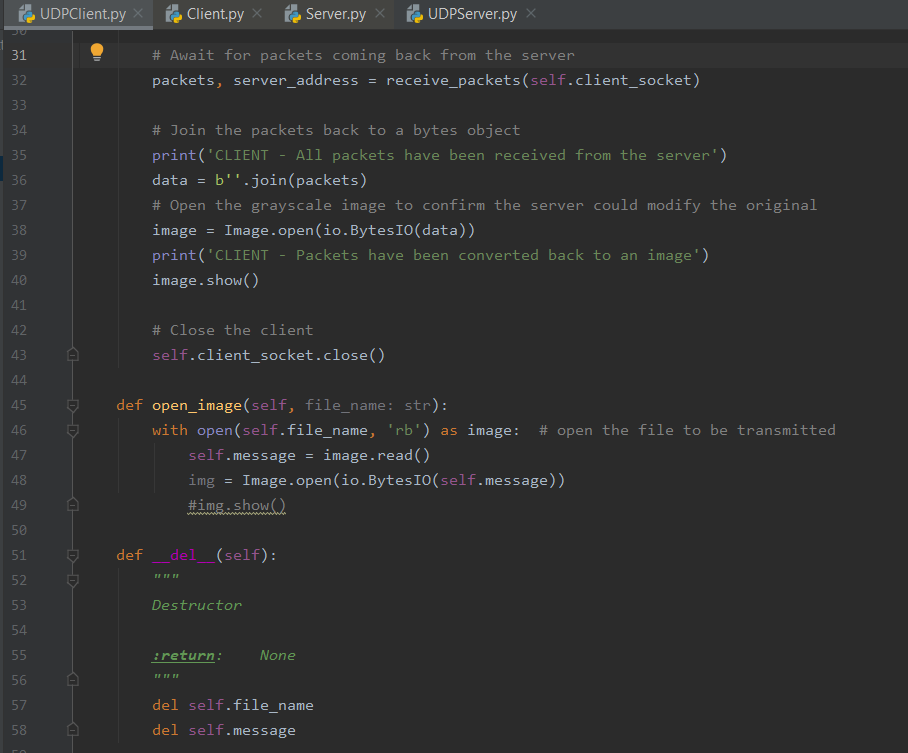


This function takes in the image (as the data parameter) and appends small portions (2048 bytes) as elements of a list of packets. Then the list of packets is returned.

Then the packets are sent over as seen in line 28 of client using the send\_packets function:

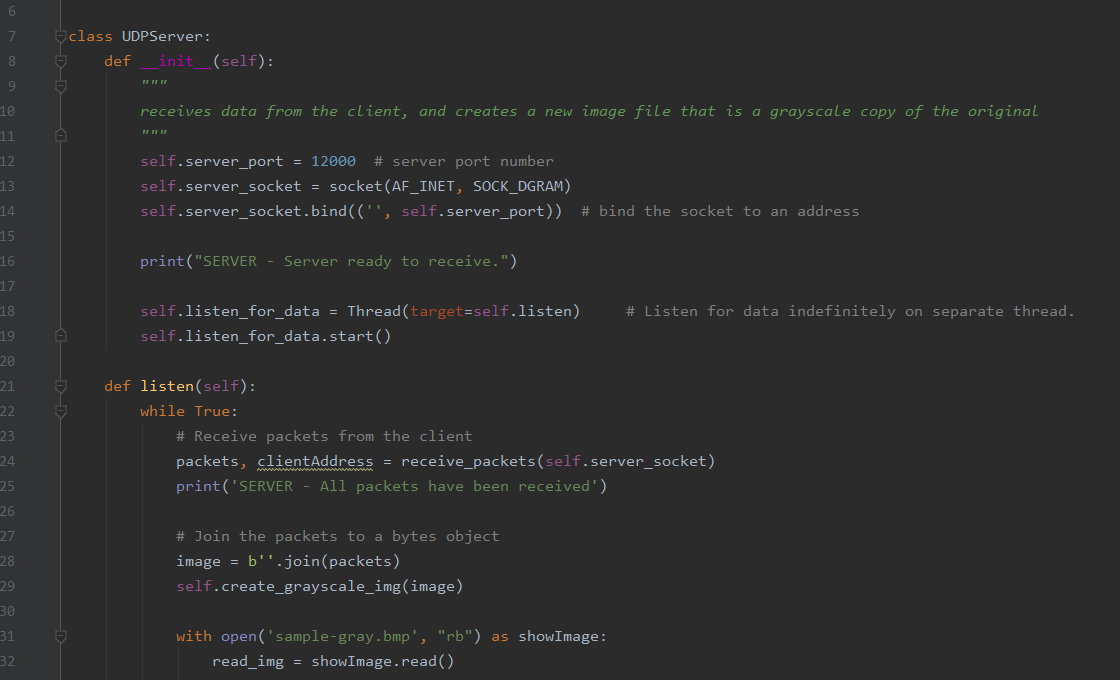


This function iterates through the list of packets sending them over to the target address 1 at a time. There is a short delay between sending each packet so the server may process packets in time before receiving more.

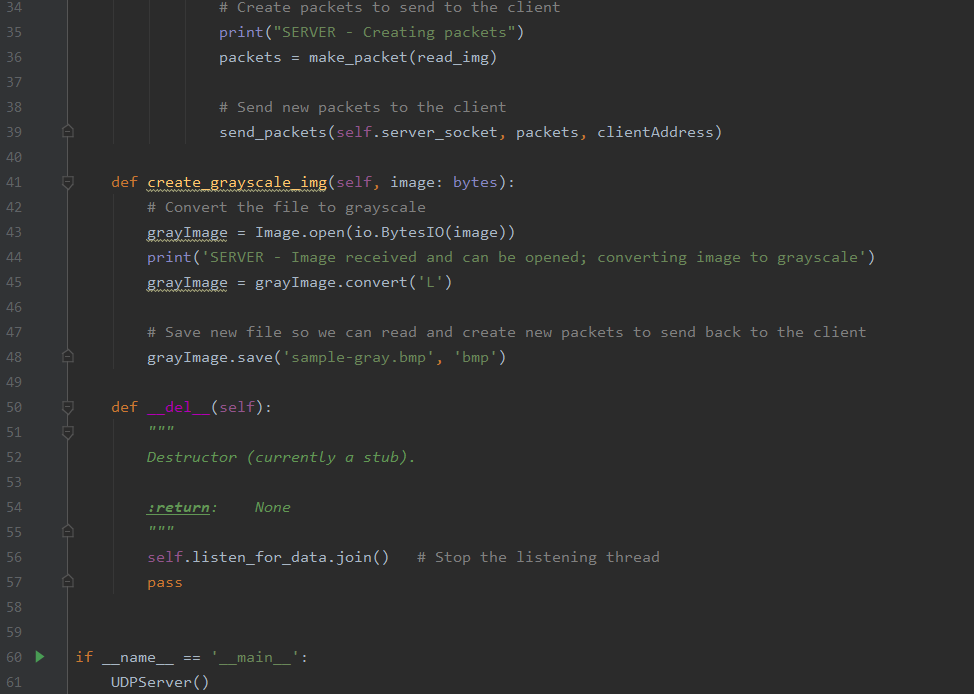


After the server receives and modifies the image, it sends it back to the client. Once it is received by the client, the packets are then stored back in the client as a byte object. The image is then opened and shown (a grayscale version of the original image). Finally, the last function is a destructor that gets called to let the sender know that all the packets have been sent.

*Server:*



The UDPServer class contains the \_init\_(self) function that deals with receiving process from the client. It makes a new image becomes a grayscale from the original. Line 14 connects the other socket to the address. Line 18 lets the listen function run begin to wait for packets. The server continues to read in packets from the client until all the packets have been sent over (determined by a termination statement sent from the client). Packets are then combined into a byte object.



The next function in line 41 converts the image file to a grayscale and then it is saved; we save it for future evaluation and sending it back to the client. After creating the grayscale image, the make\_packet function is called and the packets are then sent back to the client by the send\_packets function.

*Execution Example*

*Client:*



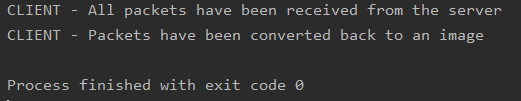


*Server:*





*Client:*



*Images Displayed:*

**

**