CLIENT ALGORITHM

class EchoClient(protocol.Protocol)

def \_\_init\_\_(self,echofactory,id,filename):

generating client handlers with the same id using echofactory;

filename is taken as input;

def connectionMade(self):

This module is used for creating the first connection made by the client after sending the request;

if(self.id == None)then

filename is taken as input;

Blockcreator().createinit() is invoked using self.transport.write;

else

the existing client instance is going to create another handler for the client by callbacks of createBlockForClient(REINIT,filename) in BlockCreater;

def dataRecieved(self,data):

this module is called after the client's request is processed by the server.

if(d[DS.CONTENT\_TYPE]==DS.ACK and[DS.OPERATION]==DS.INIT) then

id = d[DS.ID]

Call forOperation(("GET " +str(filename)) from the BlockCreator class.

if(self.d[DS.CONTENT\_TYPE]==DS.DATA):

if(self.d[DS.OPERATION]==DS.REINIT) then

creates client handler upon the acknowledgement from the server depending on the size of the data that is transferred.

ef.getMessageFromClient(id,filename)

recieveBlock(self.d)

elif(d[CONTENT\_TYPE]==EOF and d[OPERATION]==GET):

only the last block is received by the client.