**2nd Lt David Crow**

**ENG/20M**

**CSCE 560**

**Socket Programming Assignment 2**

**Web Server**

**Fall 18**

**Complete server code**

*# Import socket module*

from socket import \* *# pylint: disable=W0614*

*# Create a TCP server socket*

*# (AF\_INET is used for IPv4 protocols)*

*# (SOCK\_STREAM is used for TCP)*

serverSocket = socket(AF\_INET, SOCK\_STREAM)

*# Prepare a server socket*

serverPort = 50000

serverSocket.bind(('', serverPort))

serverSocket.listen(1)

*# Server should be up and running and listening to the incoming connections*

while True:

**print** ('Ready to serve...')

*# Set up a new connection from the client*

    connectionSocket, addr = serverSocket.accept()

*# If an exception occurs during the execution of try clause*

*# the rest of the clause is skipped*

*# If the exception type matches the word after except*

*# the except clause is executed*

    try:

*# Receives the request message from the client*

        message = connectionSocket.recv(1024)

*# Extract the path of the requested object from the message*

*# The path is the second part of HTTP header, identified by [1]*

        filename = message.split()[1]

*# Because the extracted path of the HTTP request includes*

*# a character '/', we read the path from the second character*

        f = **open**(filename[1:])

*# Store the entire content of the requested file in a temporary buffer*

        outputdata = f.read()

*# Send the HTTP response header line to the connection socket*

        connectionSocket.send('HTTP/1.1 200 OK\n'.encode())

*# Send the contents of the requested file to the connection socket*

        connectionSocket.send('Content-Type: text/html\n'.encode())

        for i in **range**(0, **len**(outputdata)):

            connectionSocket.send(outputdata[i].encode())

        connectionSocket.send("\r\n".encode())

*# Close the client connection socket*

        connectionSocket.close()

    except IOError as e:

*# Send HTTP response message for file not found*

        connectionSocket.send('HTTP/1.1 404 Not Found\n'.encode())

        connectionSocket.send('Content-Type: text/html\n'.encode())

*# Get contents of HTML error file*

        f = **open**('customerror.html')

        outputdata = f.read()

*# Send the contents of the error file to the connection socket*

        for i in **range**(0, **len**(outputdata)):

            connectionSocket.send(outputdata[i].encode())

        connectionSocket.send("\r\n".encode())

*# Close the client connection socket*

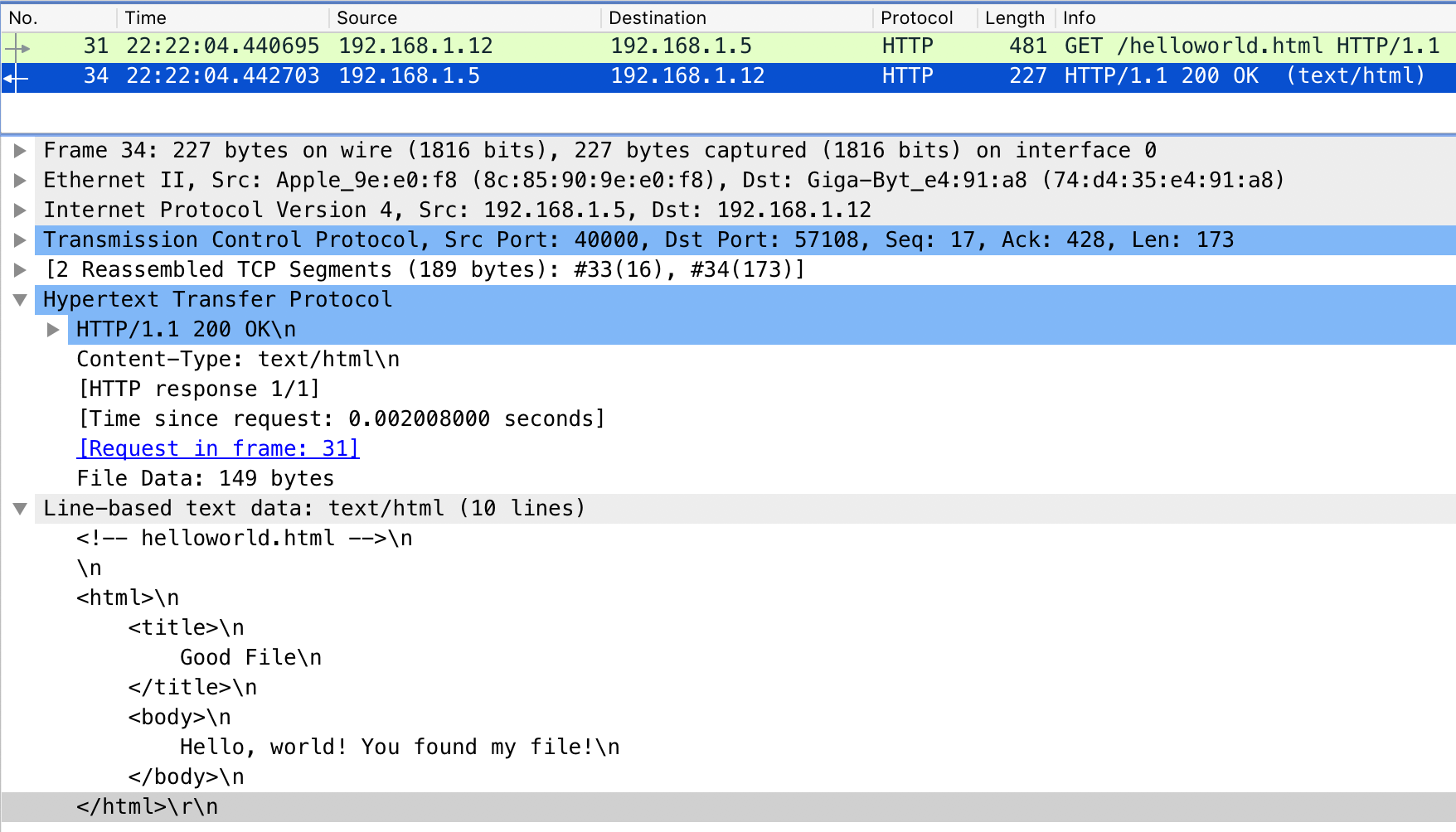
        connectionSocket.close()

serverSocket.close()

**Browser on successful retrieval**

**A screenshot of a cell phone

Description automatically generated**

**Wireshark on successful retrieval**

****

**Browser on unsuccessful retrieval**

****

**Wireshark on unsuccessful retrieval**

****

**HTML code**

*<!-- helloworld.html -->*

<html>

<title>

Good File

</title>

<body>

Hello, world! You found my file!

</body>

</html>

*<!-- customerror.html -->*

<html>

<title>

Bad File

</title>

<body>

Your requested file does not exist!

</body>

</html>