



# Report

## Assignment 3 *1DV701*

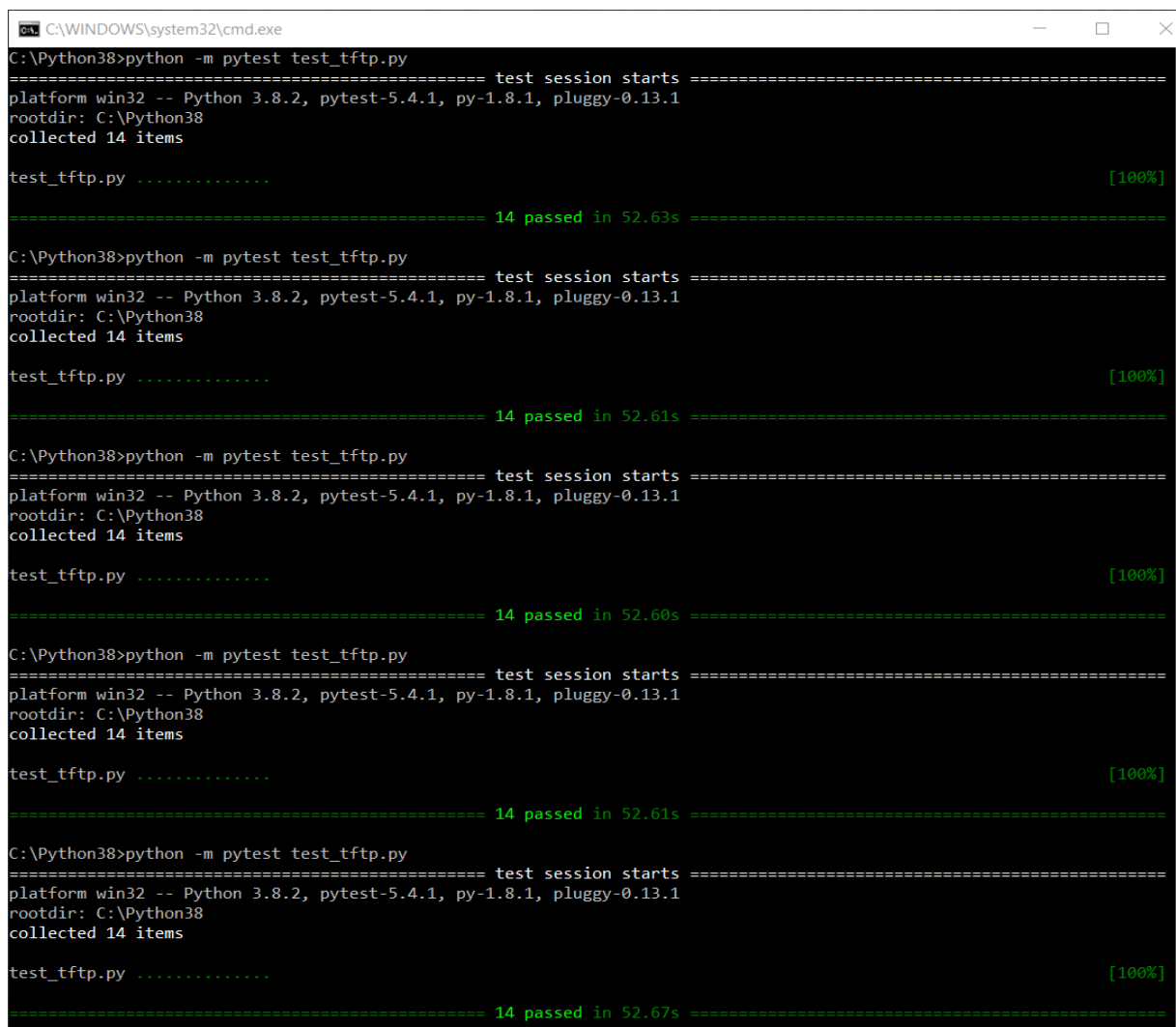


*Author:* Jacob YOUSIF  
*Semester:* VT 2020  
*Area:* Computer Science  
*Course Code:* 1DV701

**Contents**

<b>1</b>	<b>Problem 1</b>	<b>1</b>
1.1	Discussion . . . . .	2
<b>2</b>	<b>Problem 2</b>	<b>3</b>
2.1	Discussion . . . . .	5
2.2	VG-Task 1 . . . . .	6
2.2.1	Discussion . . . . .	17
<b>3</b>	<b>Problem 3</b>	<b>18</b>
3.1	Discussion . . . . .	21
3.2	VG-Task 2 . . . . .	22
3.2.1	Discussion . . . . .	24

# 1 Problem 1



```
C:\WINDOWS\system32\cmd.exe
C:\Python38>python -m pytest test_tftp.py
==== test session starts =====
platform win32 -- Python 3.8.2, pytest-5.4.1, py-1.8.1, pluggy-0.13.1
rootdir: C:\Python38
collected 14 items

test_tftp.py ..... [100%]

===== 14 passed in 52.63s =====

C:\Python38>python -m pytest test_tftp.py
==== test session starts =====
platform win32 -- Python 3.8.2, pytest-5.4.1, py-1.8.1, pluggy-0.13.1
rootdir: C:\Python38
collected 14 items

test_tftp.py ..... [100%]

===== 14 passed in 52.61s =====

C:\Python38>python -m pytest test_tftp.py
==== test session starts =====
platform win32 -- Python 3.8.2, pytest-5.4.1, py-1.8.1, pluggy-0.13.1
rootdir: C:\Python38
collected 14 items

test_tftp.py ..... [100%]

===== 14 passed in 52.60s =====

C:\Python38>python -m pytest test_tftp.py
==== test session starts =====
platform win32 -- Python 3.8.2, pytest-5.4.1, py-1.8.1, pluggy-0.13.1
rootdir: C:\Python38
collected 14 items

test_tftp.py ..... [100%]

===== 14 passed in 52.61s =====

C:\Python38>python -m pytest test_tftp.py
==== test session starts =====
platform win32 -- Python 3.8.2, pytest-5.4.1, py-1.8.1, pluggy-0.13.1
rootdir: C:\Python38
collected 14 items

test_tftp.py ..... [100%]

===== 14 passed in 52.67s =====
```

Figure 1: Using the provided grading aid to test the implementation. It shows the result after running the TEST several times.

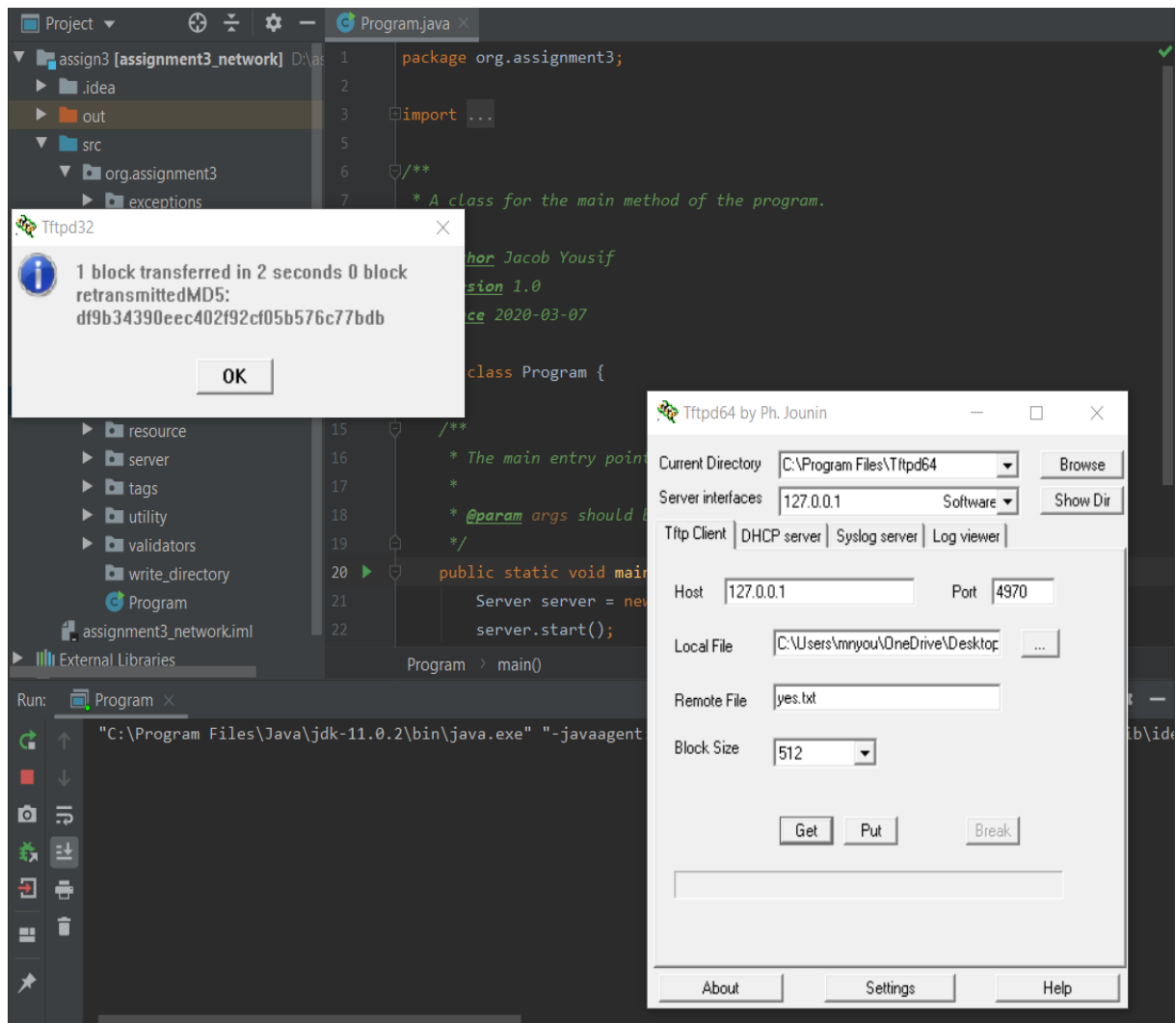


Figure 2: A GET request - file smaller than 512 bytes.

## 1.1 Discussion

Figure 2 shows the file transmission between the client and the server after the client made a GET request to download a file that is smaller than 512 bytes on a local machine. The request was executed successfully.

When it comes to the reason for having two sockets; socket and sendSocket, the reason is that socket is for binding connections with clients and the sendSocket is for exchanging traffic with one client.

## 2 Problem 2

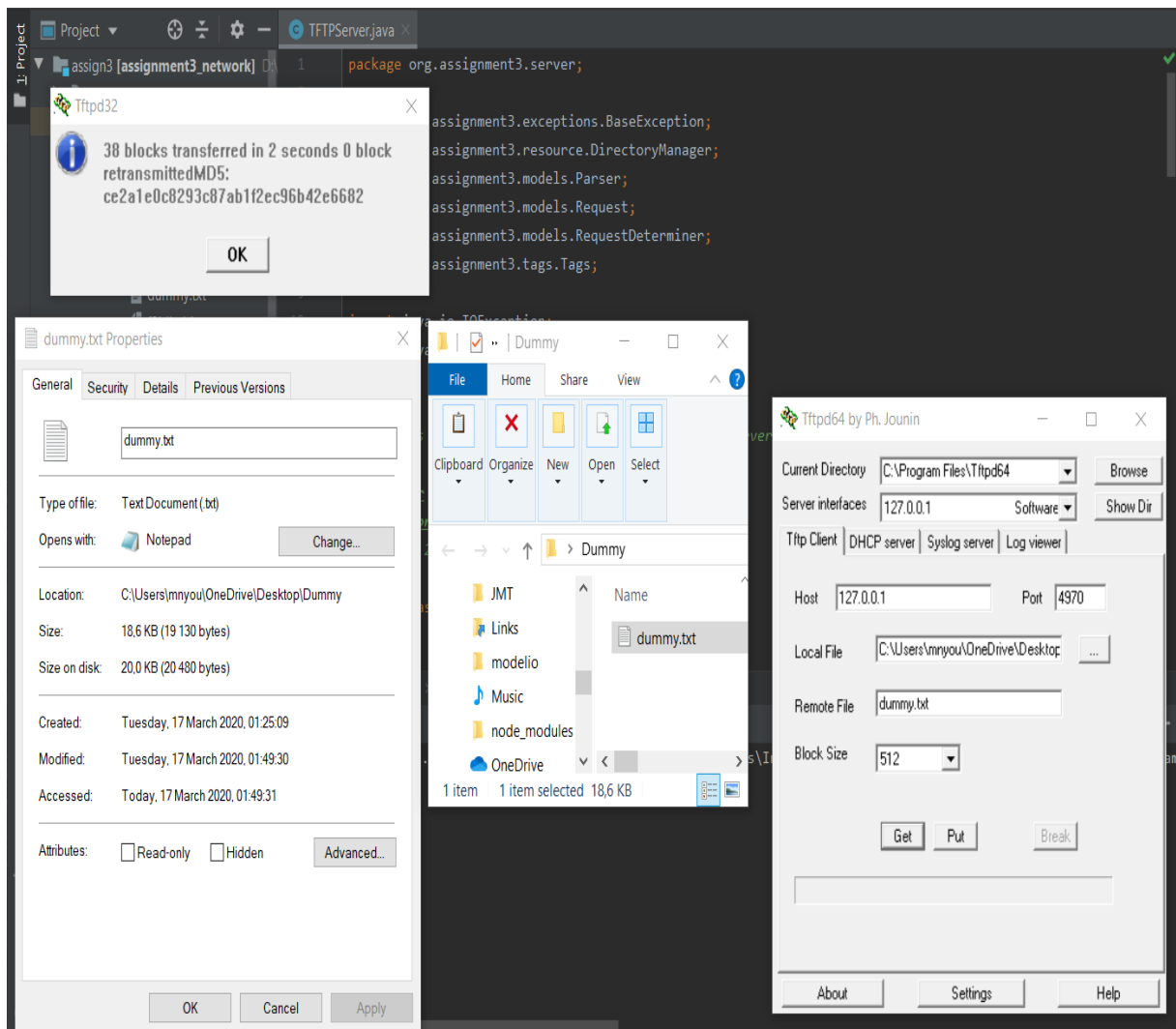


Figure 3: A GET request - file larger than 512 bytes.

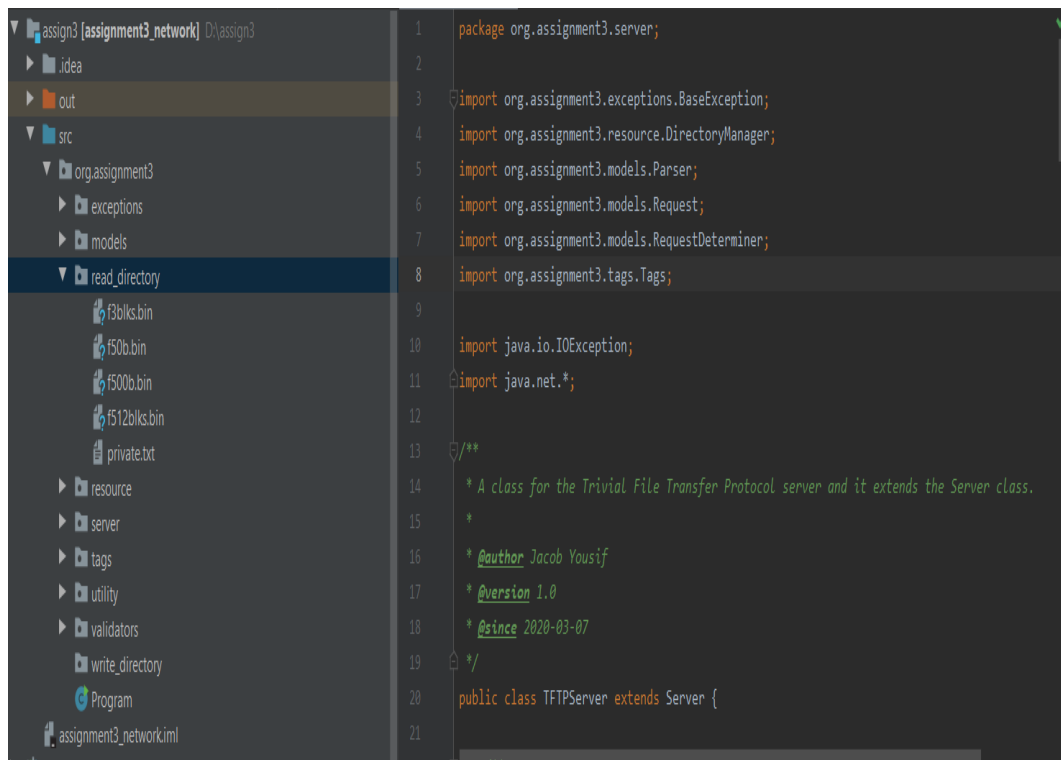


Figure 4: Before a PUT request.

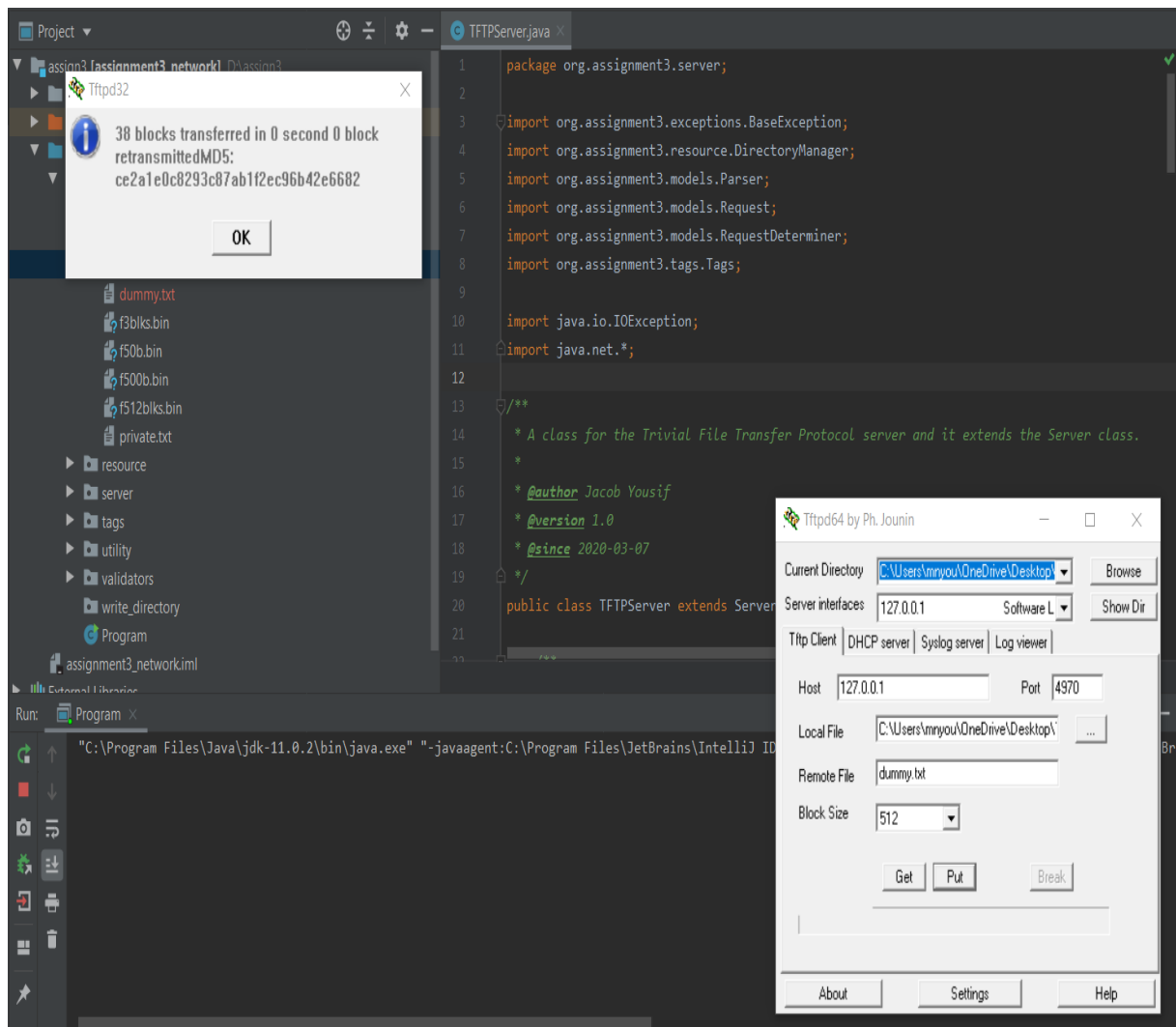


Figure 5: A PUT request.

## 2.1 Discussion

Figure 3 shows the file transmission between the client and the server after the client made a GET request to download a file that is larger than 512 bytes on a local machine. The request was executed successfully.

When it comes to PUT request, Figure 4 displays the state of the directory before the request, and Figure 5 shows the transmission was carried out successfully where the file was uploaded to the directory.

As for re-transmissions, the grading aids for Assignment 3 that were provided by the teacher was used to test this implementation. In this grading aid, some test cases that check re-transmissions through time-outs. The aid was used to test the application, and it was executed several times, and the implementation passed all these tests positively, see Figure 1.

## 2.2 VG-Task 1

14	131.725490	192.168.1.4	192.168.1.4	UDP	73 53105 → 4970 Len=41
15	131.726594	192.168.1.4	192.168.1.4	UDP	548 53106 → 53105 Len=516
16	131.728793	192.168.1.4	192.168.1.4	UDP	36 53105 → 53106 Len=4
17	131.728936	192.168.1.4	192.168.1.4	UDP	440 53106 → 53105 Len=408
18	131.729199	192.168.1.4	192.168.1.4	UDP	36 53105 → 53106 Len=4

Figure 6: Wireshark: A GET request.

```
> Frame 14: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface \Device\NPF_{Loopback}, id 0
▼ Null/Loopback
  Family: IP (2)
▼ Internet Protocol Version 4, Src: 192.168.1.4, Dst: 192.168.1.4
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 69
    Identification: 0x2aa8 (10920)
  > Flags: 0x0000
    ...0 0000 0000 0000 = Fragment offset: 0
    Time to live: 128
    Protocol: UDP (17)
    Header checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.4
    Destination: 192.168.1.4
▼ User Datagram Protocol, Src Port: 53105, Dst Port: 4970
  Source Port: 53105
  Destination Port: 4970
  Length: 49
  Checksum: 0x1afc [unverified]
  [Checksum Status: Unverified]
  [Stream index: 4]
  > [Timestamps]
▼ Data (41 bytes)
  Data: 0001736fd6546696c652e747874006f6374657400626c6b...
  [Length: 41]
```

Figure 7: Wireshark: A GET request.



```

> Frame 15: 548 bytes on wire (4384 bits), 548 bytes captured (4384 bits) on interface \Device\NPF_{Loopback, id
  Null/Loopback
    Family: IP (2)
  Internet Protocol Version 4, Src: 192.168.1.4, Dst: 192.168.1.4
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      Total Length: 544
      Identification: 0x2aa9 (10921)
    > Flags: 0x0000
      ...0 0000 0000 0000 = Fragment offset: 0
      Time to live: 128
      Protocol: UDP (17)
      Header checksum: 0x0000 [validation disabled]
      [Header checksum status: Unverified]
      Source: 192.168.1.4
      Destination: 192.168.1.4
  User Datagram Protocol, Src Port: 53106, Dst Port: 53105
    Source Port: 53106
    Destination Port: 53105
    Length: 524
    Checksum: 0x004c [unverified]
    [Checksum Status: Unverified]
    [Stream index: 5]
    > [Timestamps]
  Data (516 bytes)
    Data: 000300014c6f72656d20697073756d2064666c6f72207369...
    [Length: 516]

```

Figure 8: Wireshark: A GET request.

```

> Frame 16: 36 bytes on wire (288 bits), 36 bytes captured (288 bits) on interface \Device\NPF_{Loopback}, id 0
  Null/Loopback
    Family: IP (2)
  Internet Protocol Version 4, Src: 192.168.1.4, Dst: 192.168.1.4
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 32
    Identification: 0x2aaa (10922)
    > Flags: 0x0000
    ...0 0000 0000 0000 = Fragment offset: 0
    Time to live: 128
    Protocol: UDP (17)
    Header checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.4
    Destination: 192.168.1.4
  User Datagram Protocol, Src Port: 53105, Dst Port: 53106
    Source Port: 53105
    Destination Port: 53106
    Length: 12
    Checksum: 0xdd93 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 5]
    > [Timestamps]
  Data (4 bytes)
    Data: 00040001
    [Length: 4]

```

Figure 9: Wireshark: A GET request.

```

> Frame 17: 440 bytes on wire (3520 bits), 440 bytes captured (3520 bits) on interface \Device\NPF_{Loopback, id
  Null/Loopback
    Family: IP (2)
  Internet Protocol Version 4, Src: 192.168.1.4, Dst: 192.168.1.4
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      Total Length: 436
      Identification: 0x2aab (10923)
    > Flags: 0x0000
      ...0 0000 0000 0000 = Fragment offset: 0
      Time to live: 128
      Protocol: UDP (17)
      Header checksum: 0x0000 [validation disabled]
      [Header checksum status: Unverified]
      Source: 192.168.1.4
      Destination: 192.168.1.4
  User Datagram Protocol, Src Port: 53106, Dst Port: 53105
    Source Port: 53106
    Destination Port: 53105
    Length: 416
    Checksum: 0x6be5 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 5]
    > [Timestamps]
  Data (408 bytes)
    Data: 000300020a566976616d757320656c656d656e74756d2073...
    [Length: 408]

```

Figure 10: Wireshark: A GET request.

>	Frame 18: 36 bytes on wire (288 bits), 36 bytes captured (288 bits) on interface \Device\NPF_{Loopback}, id 0
▼	Null/Loopback
	Family: IP (2)
▼	Internet Protocol Version 4, Src: 192.168.1.4, Dst: 192.168.1.4
	0100 .... = Version: 4
	.... 0101 = Header Length: 20 bytes (5)
>	Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
	Total Length: 32
	Identification: 0x2aac (10924)
>	Flags: 0x0000
	...0 0000 0000 0000 = Fragment offset: 0
	Time to live: 128
	Protocol: UDP (17)
	Header checksum: 0x0000 [validation disabled]
	[Header checksum status: Unverified]
	Source: 192.168.1.4
	Destination: 192.168.1.4
▼	User Datagram Protocol, Src Port: 53105, Dst Port: 53106
	Source Port: 53105
	Destination Port: 53106
	Length: 12
	Checksum: 0xdd92 [unverified]
	[Checksum Status: Unverified]
	[Stream index: 5]
>	[Timestamps]
▼	Data (4 bytes)
	Data: 00040002
	[Length: 4]

Figure 11: Wireshark: A GET request.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.4	192.168.1.4	UDP	70	60821 → 4970 Len=38
2	0.020658	192.168.1.4	192.168.1.4	UDP	36	60822 → 60821 Len=4
3	0.020912	192.168.1.4	192.168.1.4	UDP	548	60821 → 60822 Len=516
4	0.022316	192.168.1.4	192.168.1.4	UDP	36	60822 → 60821 Len=4
5	0.022438	192.168.1.4	192.168.1.4	UDP	440	60821 → 60822 Len=408
6	0.023436	192.168.1.4	192.168.1.4	UDP	36	60822 → 60821 Len=4

Figure 12: Wireshark: A PUT request.

```

> Frame 1: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface \Device\NPF_{Loopback}, id 0
> Null/Loopback
v Internet Protocol Version 4, Src: 192.168.1.4, Dst: 192.168.1.4
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 66
    Identification: 0x2abb (10939)
  > Flags: 0x0000
    ...0 0000 0000 0000 = Fragment offset: 0
    Time to live: 128
    Protocol: UDP (17)
    Header checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.4
    Destination: 192.168.1.4
v User Datagram Protocol, Src Port: 60821, Dst Port: 4970
  Source Port: 60821
  Destination Port: 4970
  Length: 46
  Checksum: 0x98b5 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 0]
  > [Timestamps]
v Data (38 bytes)
  Data: 00027965732e747874006f6374657400626c6b73697a6500...
  [Length: 38]

```

Figure 13: Wireshark: A PUT request.

```
> Frame 2: 36 bytes on wire (288 bits), 36 bytes captured (288 bits) on interface \Device\NPF_{Loopback}, id 0
> Null/Loopback
▼ Internet Protocol Version 4, Src: 192.168.1.4, Dst: 192.168.1.4
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 32
    Identification: 0x2abc (10940)
  > Flags: 0x0000
    ...0 0000 0000 0000 = Fragment offset: 0
    Time to live: 128
    Protocol: UDP (17)
    Header checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.4
    Destination: 192.168.1.4
  ▼ User Datagram Protocol, Src Port: 60822, Dst Port: 60821
    Source Port: 60822
    Destination Port: 60821
    Length: 12
    Checksum: 0xa14c [unverified]
    [Checksum Status: Unverified]
    [Stream index: 1]
  > [Timestamps]
  ▼ Data (4 bytes)
    Data: 00040000
    [Length: 4]
```

Figure 14: Wireshark: A PUT request.

```

> Frame 3: 548 bytes on wire (4384 bits), 548 bytes captured (4384 bits) on interface \Device\NPF_{Loopback}, id
> Null/Loopback
▼ Internet Protocol Version 4, Src: 192.168.1.4, Dst: 192.168.1.4
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
        Total Length: 544
        Identification: 0x2abd (10941)
    > Flags: 0x0000
        ...0 0000 0000 0000 = Fragment offset: 0
        Time to live: 128
        Protocol: UDP (17)
        Header checksum: 0x0000 [validation disabled]
        [Header checksum status: Unverified]
        Source: 192.168.1.4
        Destination: 192.168.1.4
▼ User Datagram Protocol, Src Port: 60821, Dst Port: 60822
    Source Port: 60821
    Destination Port: 60822
    Length: 524
    Checksum: 0xc403 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 1]
    > [Timestamps]
▼ Data (516 bytes)
    Data: 000300014c6f72656d20697073756d2064666c6f72207369...
    [Length: 516]

```

Figure 15: Wireshark: A PUT request.

```
> Frame 4: 36 bytes on wire (288 bits), 36 bytes captured (288 bits) on interface \Device\NPF_{Loopback}, id 0
> Null/Loopback
▼ Internet Protocol Version 4, Src: 192.168.1.4, Dst: 192.168.1.4
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 32
    Identification: 0x2abe (10942)
  > Flags: 0x0000
    ...0 0000 0000 0000 = Fragment offset: 0
    Time to live: 128
    Protocol: UDP (17)
    Header checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.4
    Destination: 192.168.1.4
▼ User Datagram Protocol, Src Port: 60822, Dst Port: 60821
  Source Port: 60822
  Destination Port: 60821
  Length: 12
  Checksum: 0xa14b [unverified]
  [Checksum Status: Unverified]
  [Stream index: 1]
  > [Timestamps]
▼ Data (4 bytes)
  Data: 00040001
  [Length: 4]
```

Figure 16: Wireshark: A PUT request.



```

> Frame 5: 440 bytes on wire (3520 bits), 440 bytes captured (3520 bits) on interface \Device\NPF_{Loopback, id
> Null/Loopback
v Internet Protocol Version 4, Src: 192.168.1.4, Dst: 192.168.1.4
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 436
    Identification: 0x2abf (10943)
  > Flags: 0x0000
    ...0 0000 0000 0000 = Fragment offset: 0
    Time to live: 128
    Protocol: UDP (17)
    Header checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.4
    Destination: 192.168.1.4
v User Datagram Protocol, Src Port: 60821, Dst Port: 60822
  Source Port: 60821
  Destination Port: 60822
  Length: 416
  Checksum: 0x2f9d [unverified]
  [Checksum Status: Unverified]
  [Stream index: 1]
  > [Timestamps]
v Data (408 bytes)
  Data: 000300020a566976616d757320656c656d656e74756d2073...
  [Length: 408]

```

Figure 17: Wireshark: A PUT request.

```

> Frame 6: 36 bytes on wire (288 bits), 36 bytes captured (288 bits) on interface \Device\NPF_{Loopback}, id 0
> Null/Loopback
▼ Internet Protocol Version 4, Src: 192.168.1.4, Dst: 192.168.1.4
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 32
    Identification: 0x2ac0 (10944)
  > Flags: 0x0000
    ...0 0000 0000 0000 = Fragment offset: 0
    Time to live: 128
    Protocol: UDP (17)
    Header checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.4
    Destination: 192.168.1.4
▼ User Datagram Protocol, Src Port: 60822, Dst Port: 60821
  Source Port: 60822
  Destination Port: 60821
  Length: 12
  Checksum: 0xa14a [unverified]
  [Checksum Status: Unverified]
  [Stream index: 1]
  > [Timestamps]
▼ Data (4 bytes)
  Data: 00040002
  [Length: 4]

```

Figure 18: Wireshark: A PUT request.

```

> Frame 7: 36 bytes on wire (288 bits), 36 bytes captured (288 bits) on interface \Device\NPF_{Loopback}, id 0
> Null/Loopback
v Internet Protocol Version 4, Src: 192.168.1.4, Dst: 192.168.1.4
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 32
  Identification: 0x2ac1 (10945)
> Flags: 0x0000
  ...0 0000 0000 0000 = Fragment offset: 0
  Time to live: 128
  Protocol: UDP (17)
  Header checksum: 0x0000 [validation disabled]
  [Header checksum status: Unverified]
  Source: 192.168.1.4
  Destination: 192.168.1.4
v User Datagram Protocol, Src Port: 60822, Dst Port: 60821
  Source Port: 60822
  Destination Port: 60821
  Length: 12
  Checksum: 0xa14b [unverified]
  [Checksum Status: Unverified]
  [Stream index: 1]
> [Timestamps]
v Data (4 bytes)
  Data: 00050000
  [Length: 4]

```

Figure 19: Wireshark: A PUT request.

### 2.2.1 Discussion

From Figure 6 - Figure 11, the traffic between the client and the server after the client made a GET request. The client requested to download a file on the local machine.

The first line in Figure 6, the client sent the request to the server where its size is 41, and that is the opcode, filename, and the mode. In the second line, the server sent data to the client. The size is 516, where 4 bytes just the opcode and the block number, and the rest is the raw data. In the third line, the client acknowledged the transmission, and in the fourth line, the server sends the next blocks, and in the last line, the client acknowledges it again.

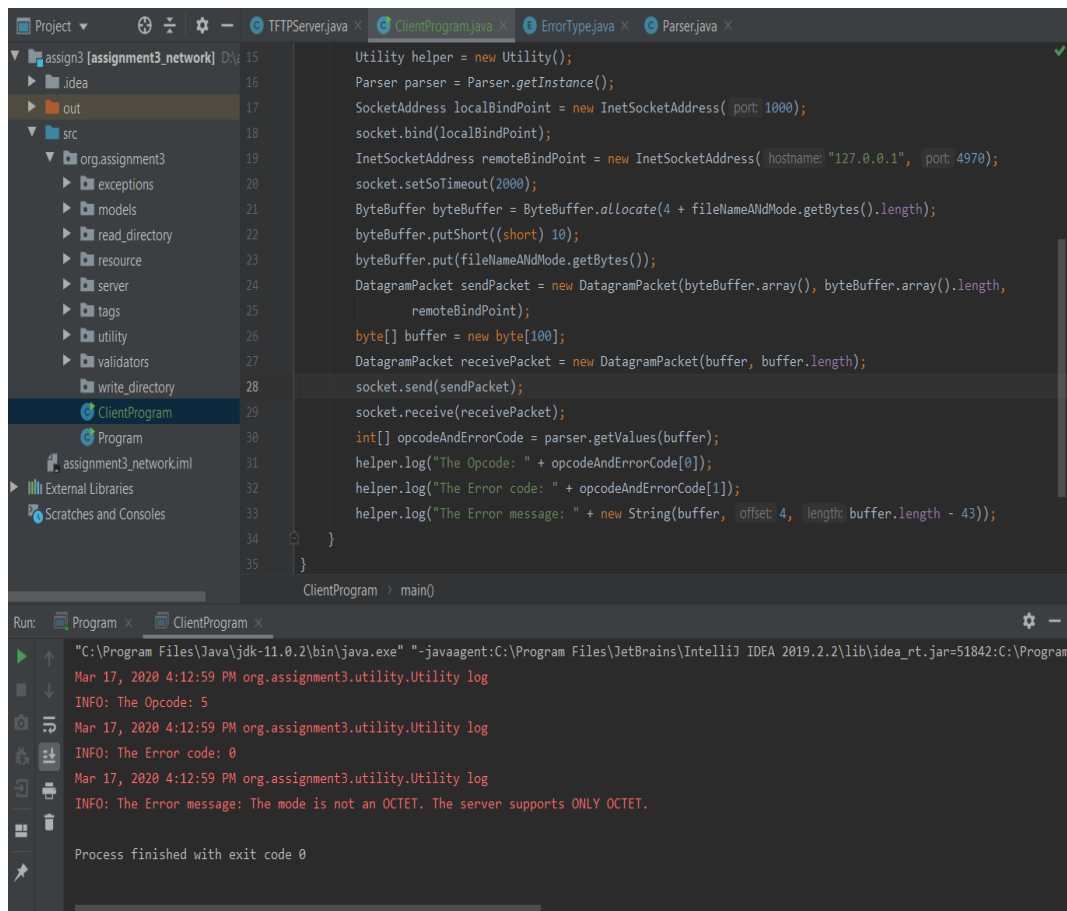
From Figure 6 - Figure 11, the information about each transaction that occurred during this transmission. The information contains the address of the source, the address of the destination, the port number of the source, the port number of the destination, checksum, index number, data length and the raw data.

From Figure 12 - Figure 19, the traffic between the client and the server after the client made a PUT request. The client requested to upload a file to the server. It is a similar process. The difference is the client who sends data in this operation and the server is the one who acknowledges the data.

The difference between the two requests are:

- The value of the Opcode.
- GET: The server who sends data and the client who acknowledges the data.
- PUT: The client who sends data and the server who acknowledges the data.

### 3 Problem 3



The screenshot displays an IDE with the following components:

- Project Explorer:** Shows a project named 'assignment3\_network' with a 'src' directory containing 'org.assignment3' and 'ClientProgram'.
- Code Editor:** Displays the code for 'ClientProgram.java'. The code includes imports for 'Utility', 'Parser', 'SocketAddress', 'InetSocketAddress', 'ByteBuffer', 'DatagramPacket', and 'String'. It defines a 'main' method that sets up a socket, sends a packet, and receives a response. The received data is parsed and logged.
- Run Console:** Shows the output of the program. It includes the command used to run the program, the class name, and the logs generated by the 'Utility' class. The logs show the opcode as 5, the error code as 0, and the error message as 'The mode is not an OCTET. The server supports ONLY OCTET.'.

```
Utility helper = new Utility();
Parser parser = Parser.getInstance();
SocketAddress localBindPoint = new InetSocketAddress( port: 1000);
socket.bind(localBindPoint);
InetSocketAddress remoteBindPoint = new InetSocketAddress( hostname: "127.0.0.1", port: 4970);
socket.setSoTimeout(2000);
ByteBuffer byteBuffer = ByteBuffer.allocate(4 + fileNameAndMode.getBytes().length);
byteBuffer.putShort((short) 10);
byteBuffer.put(fileNameAndMode.getBytes());
DatagramPacket sendPacket = new DatagramPacket(byteBuffer.array(), byteBuffer.array().length,
remoteBindPoint);
byte[] buffer = new byte[100];
DatagramPacket receivePacket = new DatagramPacket(buffer, buffer.length);
socket.send(sendPacket);
socket.receive(receivePacket);
int[] opcodeAndErrorCode = parser.getValues(buffer);
helper.log("The Opcode: " + opcodeAndErrorCode[0]);
helper.log("The Error code: " + opcodeAndErrorCode[1]);
helper.log("The Error message: " + new String(buffer, offset: 4, length: buffer.length - 43));
}
```

Run: Program x ClientProgram x

```
"C:\Program Files\Java\jdk-11.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2019.2.2\lib\idea_rt.jar=51842:C:\Program
Mar 17, 2020 4:12:59 PM org.assignment3.utility.Utility log
INFO: The Opcode: 5
Mar 17, 2020 4:12:59 PM org.assignment3.utility.Utility log
INFO: The Error code: 0
Mar 17, 2020 4:12:59 PM org.assignment3.utility.Utility log
INFO: The Error message: The mode is not an OCTET. The server supports ONLY OCTET.
Process finished with exit code 0
```

Figure 20: Error Code: 0

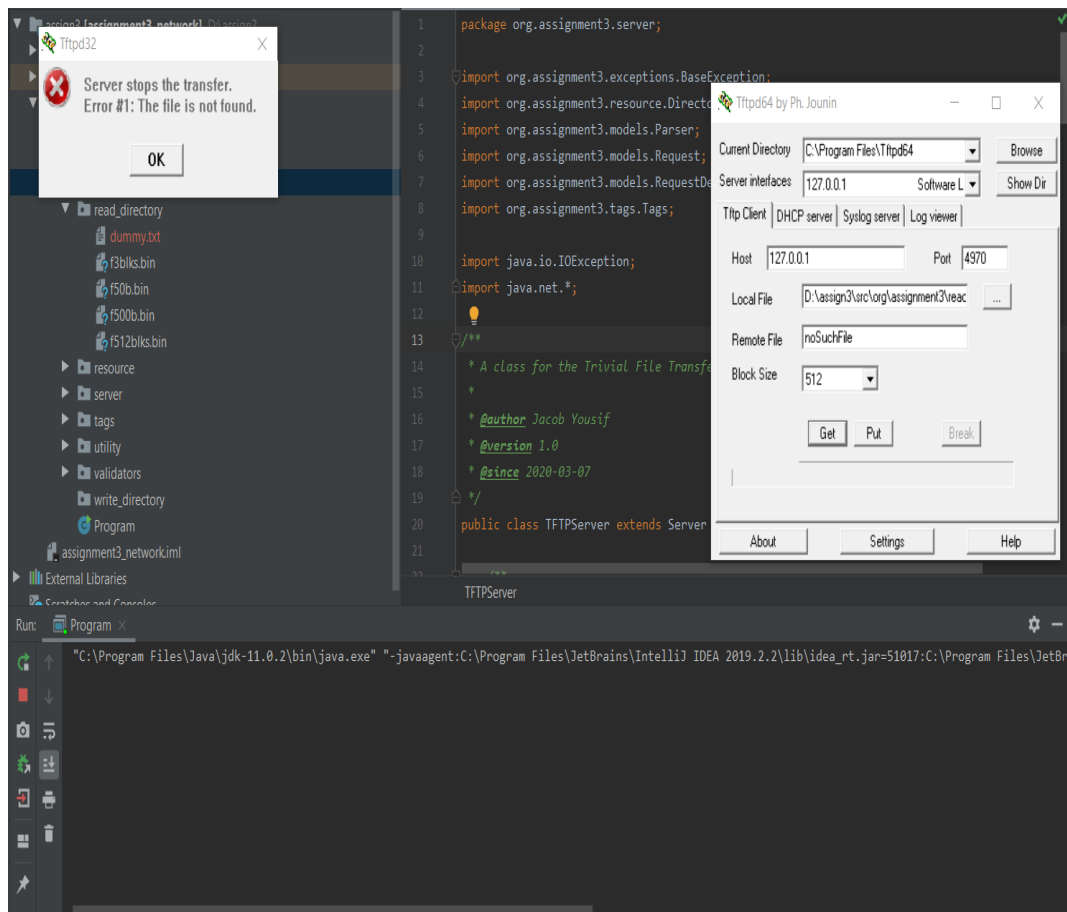


Figure 21: Error Code: 1

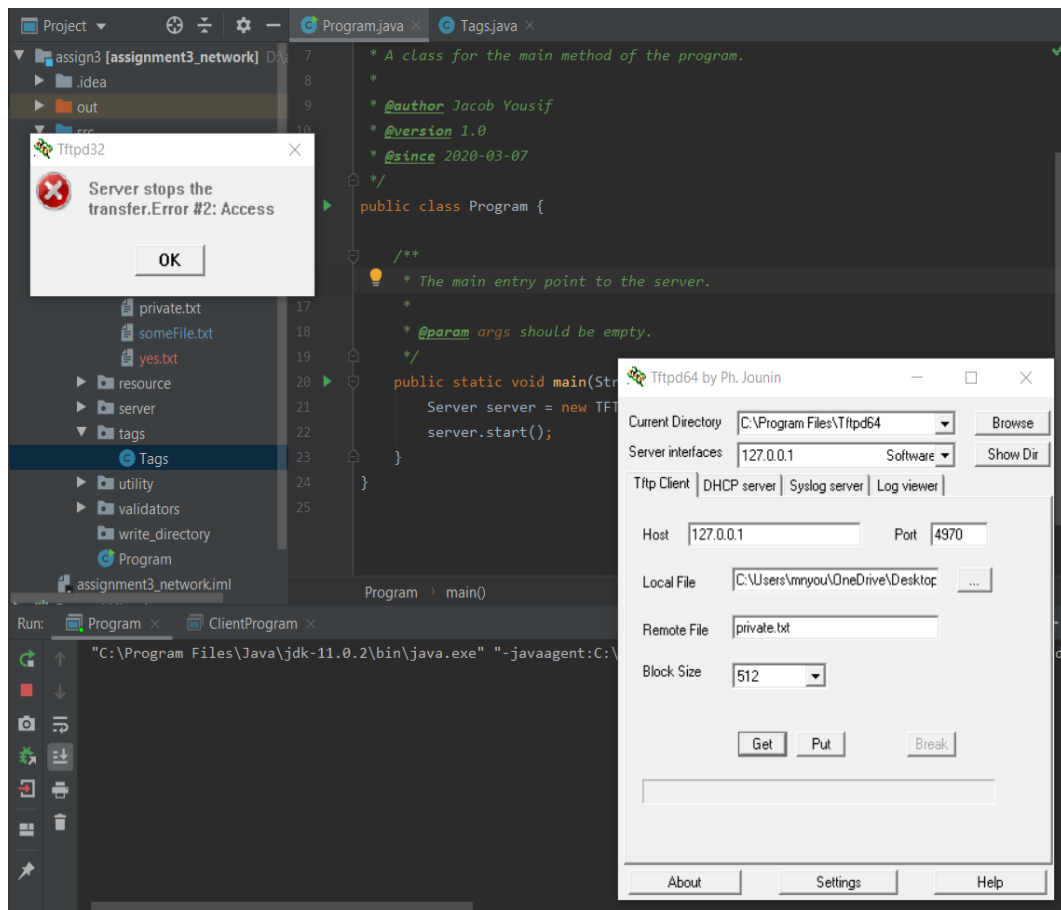


Figure 22: Error Code: 2

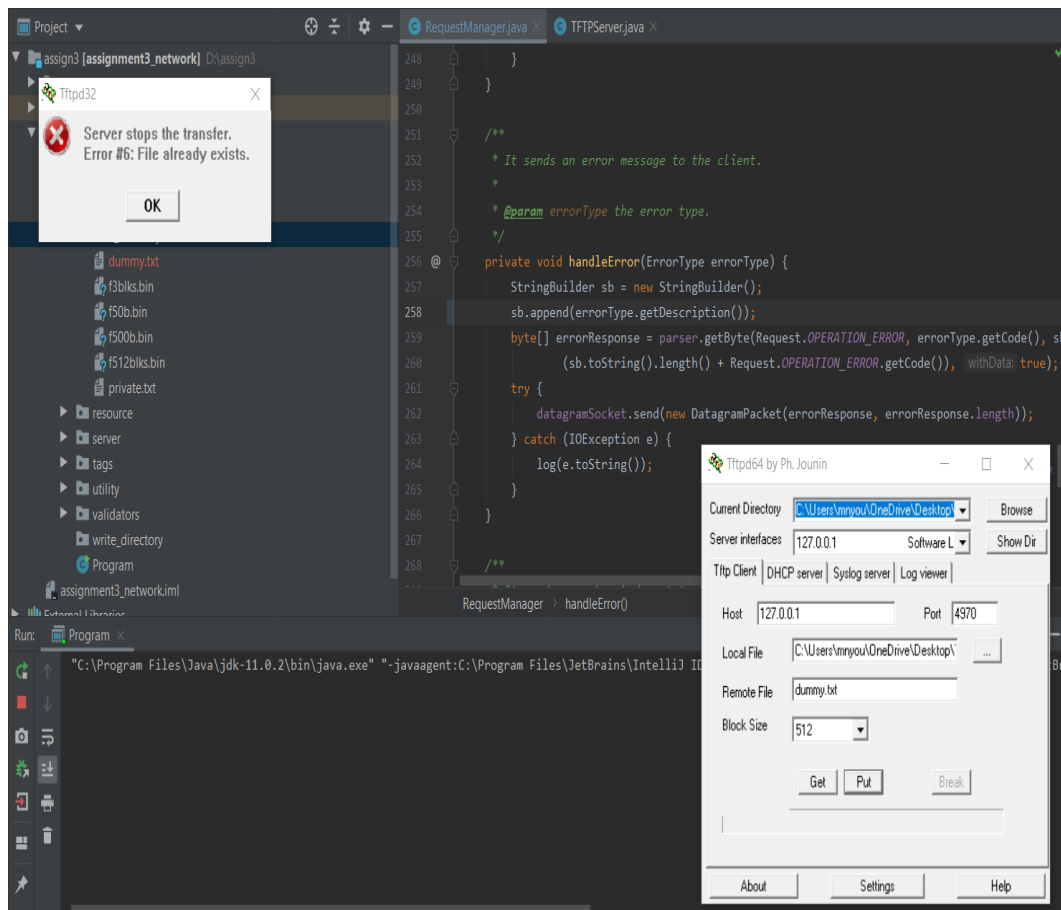


Figure 23: Error Code: 6

### 3.1 Discussion

As for Error Code 1, 2, and 6, they are standard error codes, See Figure 21 - 23. When it comes to Error Code 0, it is undefined, and this code can be used for any error. In this implementation, it was used for invalid mode. The assignment requires only OCTET mode; therefore, any other mode was defined to be invalid in this scope, and the error code applies to it. The Error Code was also assigned for errors that are related to re-transmissions in this application, i. e. when the client exceeds the allowed amount of re-transmissions.

To test this Error Code, a class for the client was implemented to send a request with a mode that is not OCTET, in order to test the response of the server. Figure 20 shows the response of the server after the client sending a request with invalid mode.

### 3.2 VG-Task 2

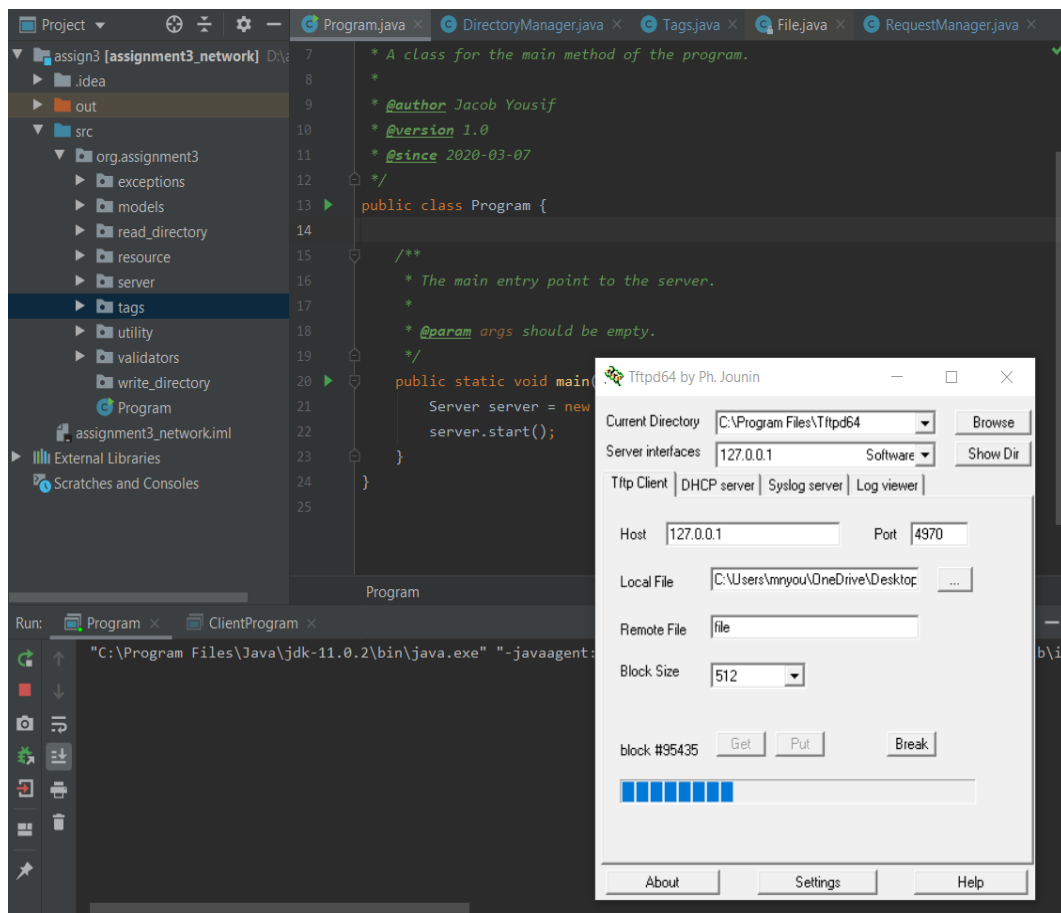


Figure 24: Sending a large file that is larger than the free space of the hard disk.



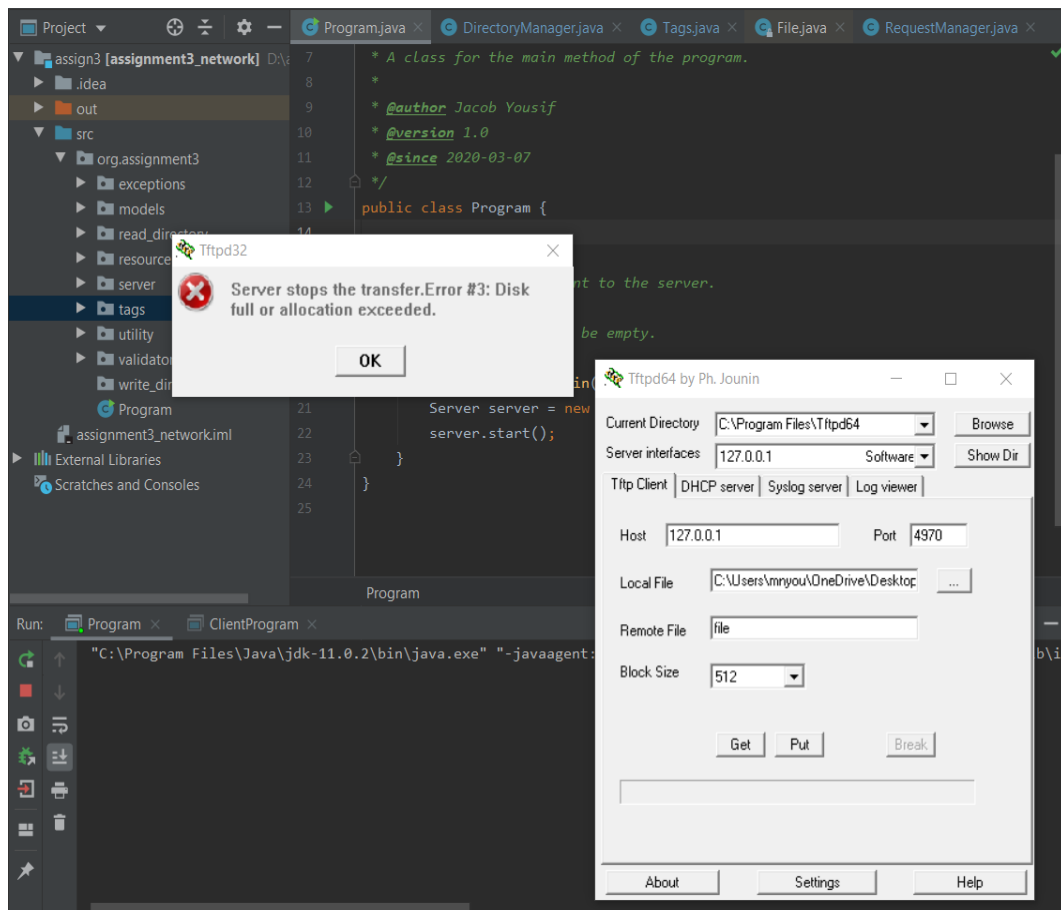


Figure 25: Error Code: 3

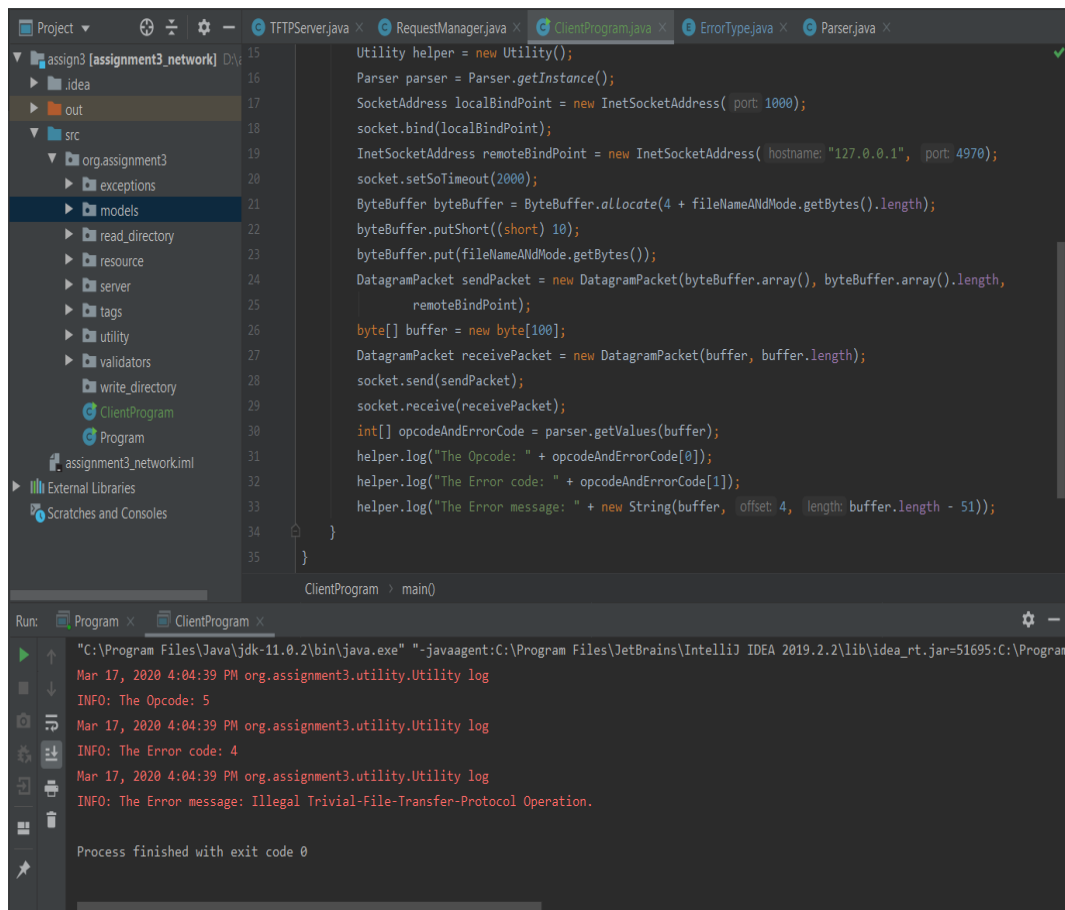


Figure 26: Error Code: 4

### 3.2.1 Discussion

Figure 25 and 26 show the Error Code 3 and 4. As for Error Code 4, a client class was implemented in order to send an opcode that is not defined, i.e. it does not exist in the context of TFTP nor in the RFC specifications. Figure 26 shows the response of the server after the client sends a request with the opcode that is invalid to the server.

When it comes to Error Code 5, it is primarily about identification. The best way to identify whether the server is still communicating with the same client is through the IP address of the client. When a client initiates a connection with the server, the server saves the IP address of the client. The server will continuously check the IP address throughout the transmissions to verify it is still communicating with the same client. In case, it discovers that the IP address is not as same as the one the client used to initiate the connection, then the server will send the error type: Unknown transfer ID and terminates the connection.