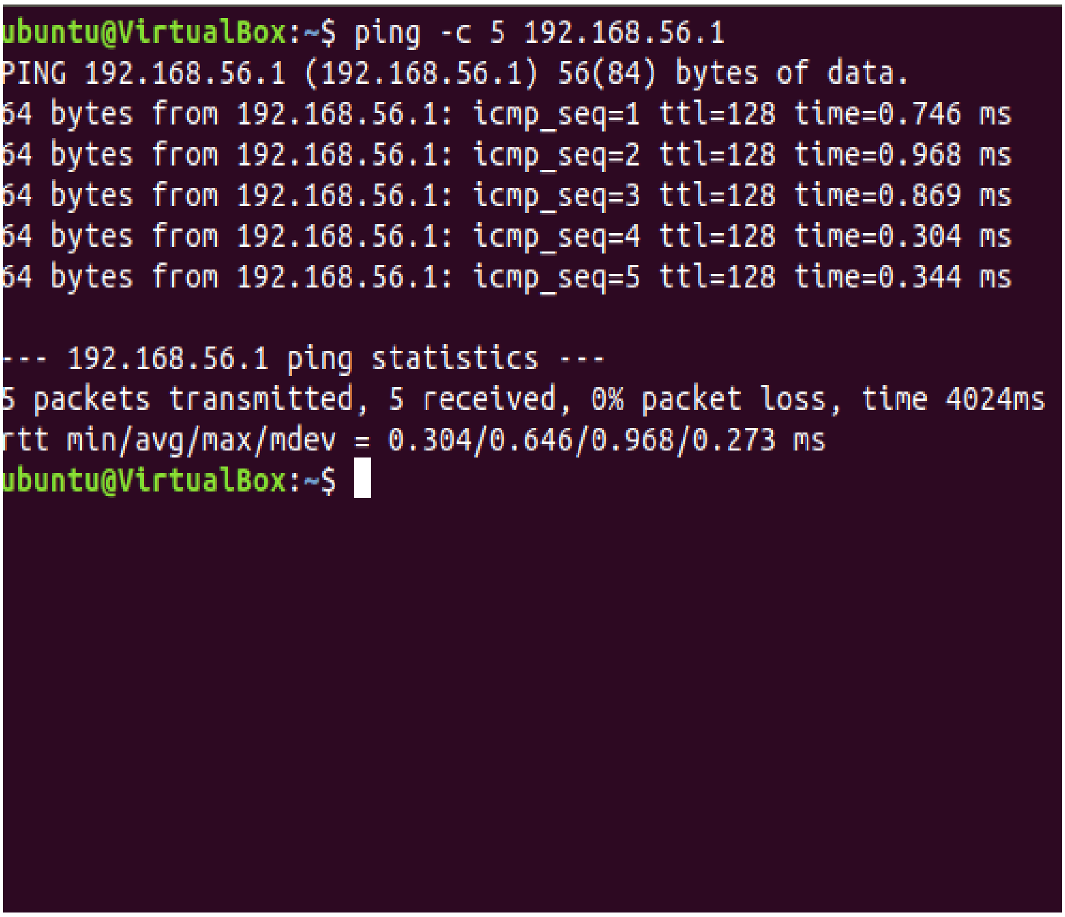
Report

Assignment 1
1DV701

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

Contents

1 Problem 1



```
ubuntu@VirtualBox:~$ ping -c 5 192.168.56.1
PING 192.168.56.1 (192.168.56.1) 56(84) bytes of data.
64 bytes from 192.168.56.1: icmp_seq=1 ttl=128 time=0.746 ms
64 bytes from 192.168.56.1: icmp_seq=2 ttl=128 time=0.968 ms
64 bytes from 192.168.56.1: icmp_seq=3 ttl=128 time=0.869 ms
64 bytes from 192.168.56.1: icmp_seq=4 ttl=128 time=0.304 ms
64 bytes from 192.168.56.1: icmp_seq=5 ttl=128 time=0.344 ms

--- 192.168.56.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4024ms
rtt min/avg/max/mdev = 0.304/0.646/0.968/0.273 ms
ubuntu@VirtualBox:~$
```

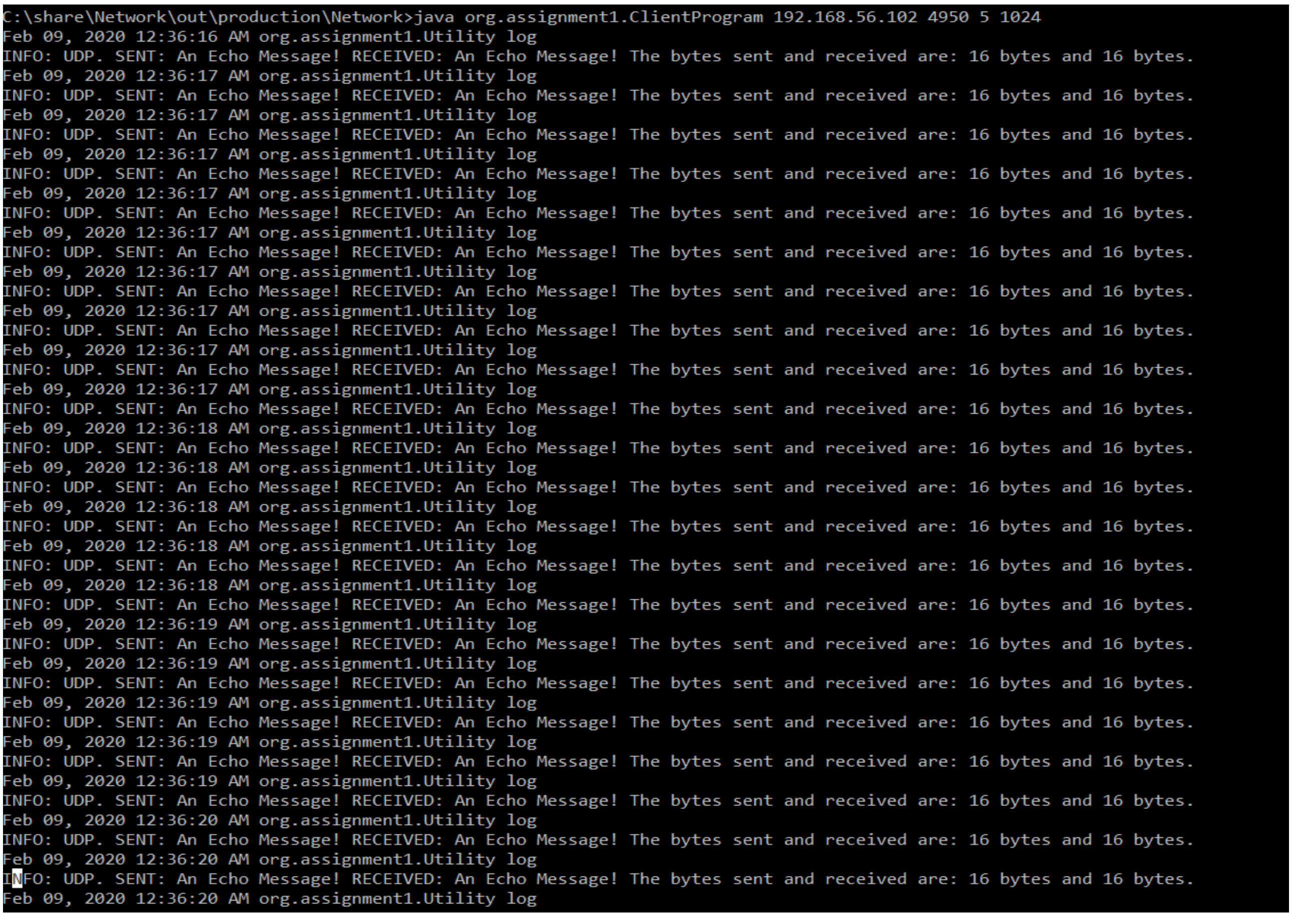
Figure 1: Ping testing the IP address of the other host machine.

1.1 Discussion

It is quite a useful tool to check and test a connection between a machine at one point with another device at another end, by using the IP address of the host. Moreover, it provides the time rate, per seconds and how many packages transmitted, sent and received.

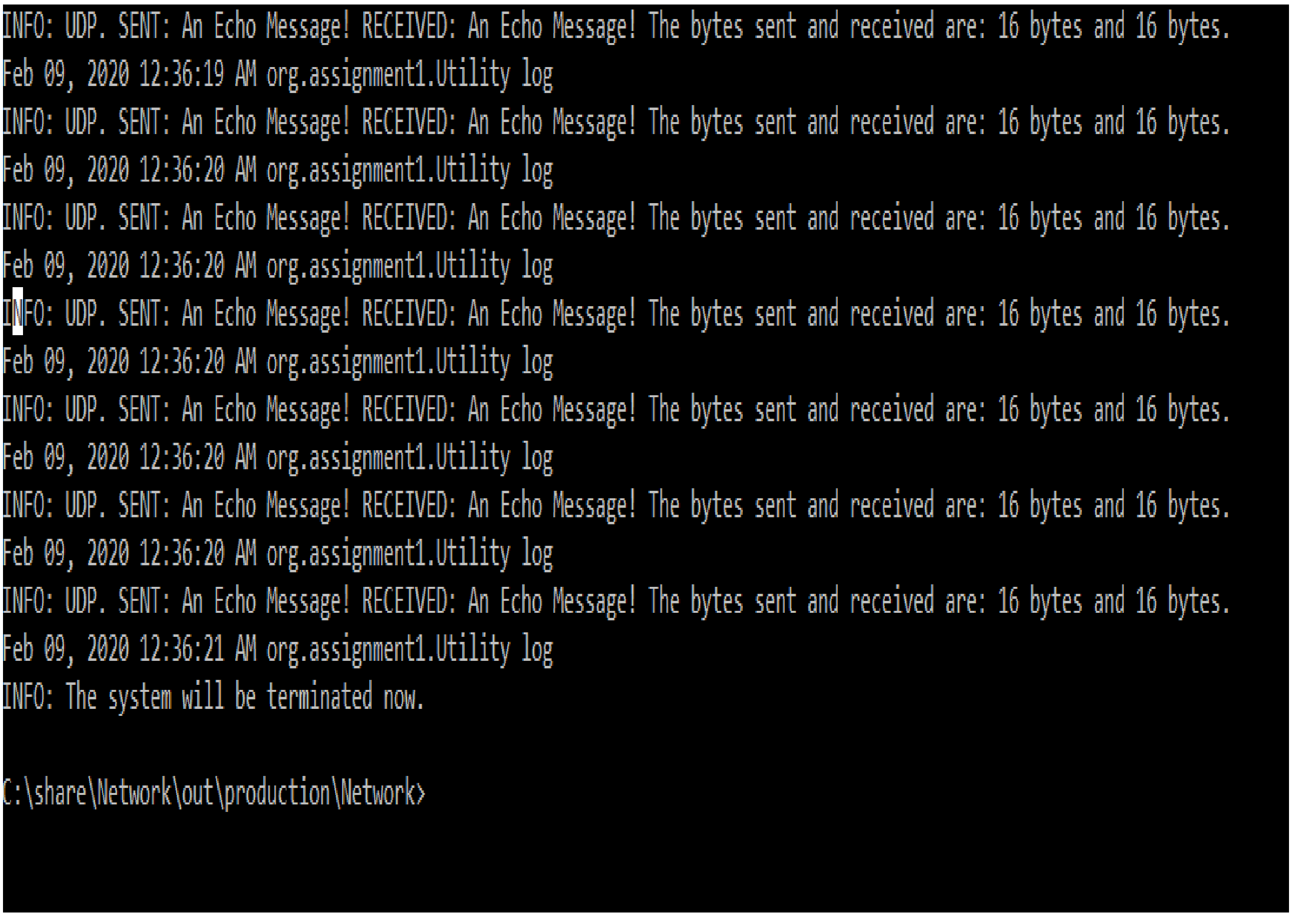
2 Problem 2

2.1 Problem 2.1



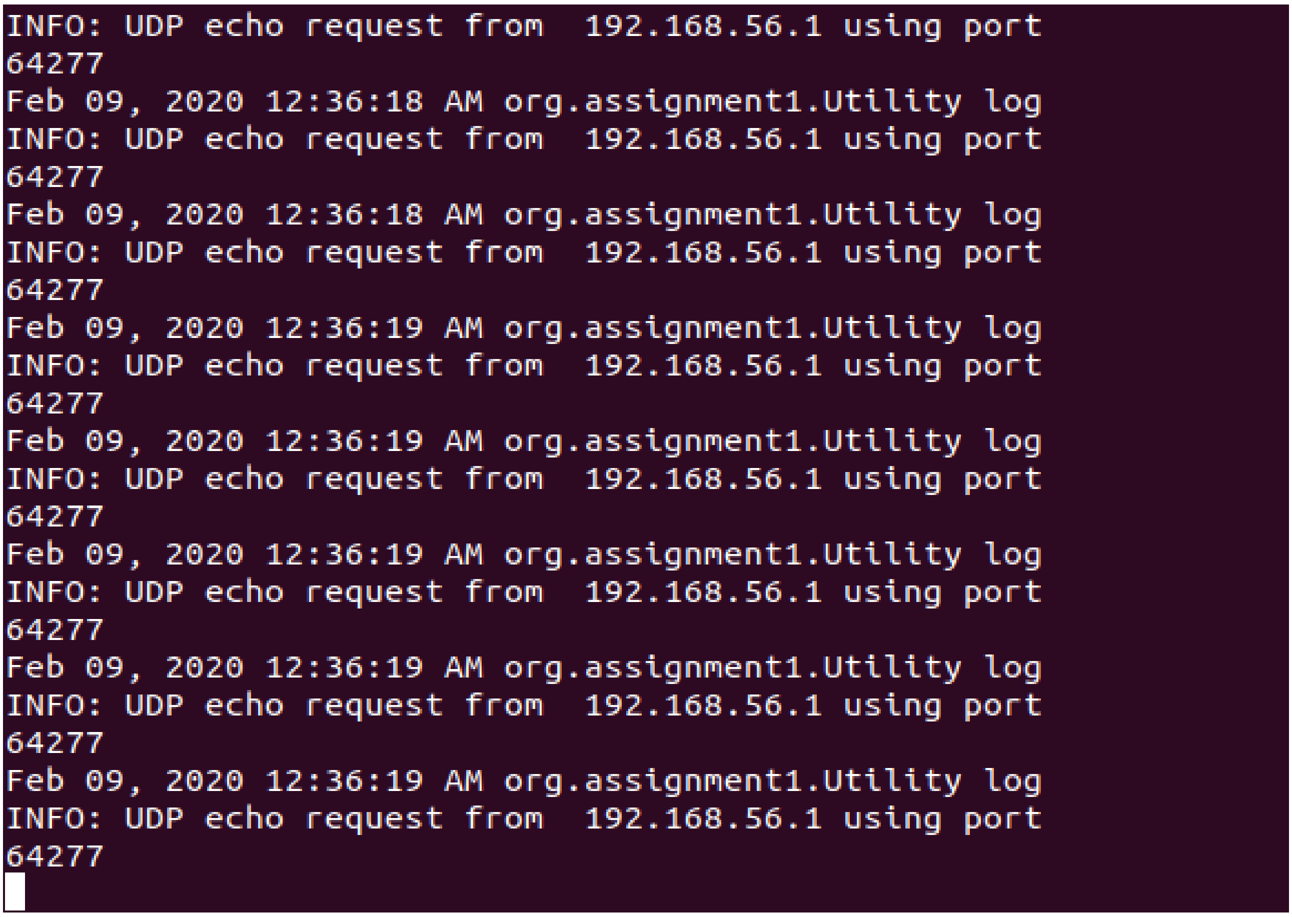
```
C:\share\Network\out\production\Network>java org.assignment1.ClientProgram 192.168.56.102 4950 5 1024
Feb 09, 2020 12:36:16 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:17 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:17 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:17 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:17 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:17 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:17 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:17 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:17 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:17 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:18 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:18 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:18 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:18 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:18 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:19 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:19 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:19 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:19 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:19 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:20 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:20 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:20 AM org.assignment1.Utility log
```

Figure 2: UDP: The host sends five messages per second to the receiver.



```
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:19 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:20 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:20 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:20 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:20 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:20 AM org.assignment1.Utility log
INFO: UDP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:36:21 AM org.assignment1.Utility log
INFO: The system will be terminated now.

C:\share\Network\out\production\Network>
```

Figure 3: UDP: The host sends five messages per second to the receiver.



```
INFO: UDP echo request from  192.168.56.1 using port
64277
Feb 09, 2020 12:36:18 AM org.assignment1.Utility log
INFO: UDP echo request from  192.168.56.1 using port
64277
Feb 09, 2020 12:36:18 AM org.assignment1.Utility log
INFO: UDP echo request from  192.168.56.1 using port
64277
Feb 09, 2020 12:36:19 AM org.assignment1.Utility log
INFO: UDP echo request from  192.168.56.1 using port
64277
Feb 09, 2020 12:36:19 AM org.assignment1.Utility log
INFO: UDP echo request from  192.168.56.1 using port
64277
Feb 09, 2020 12:36:19 AM org.assignment1.Utility log
INFO: UDP echo request from  192.168.56.1 using port
64277
Feb 09, 2020 12:36:19 AM org.assignment1.Utility log
INFO: UDP echo request from  192.168.56.1 using port
64277
Feb 09, 2020 12:36:19 AM org.assignment1.Utility log
INFO: UDP echo request from  192.168.56.1 using port
64277
```

Figure 4: UDP: Segment from the server command control: Echoing the messages back.

2.1.1 Discussion

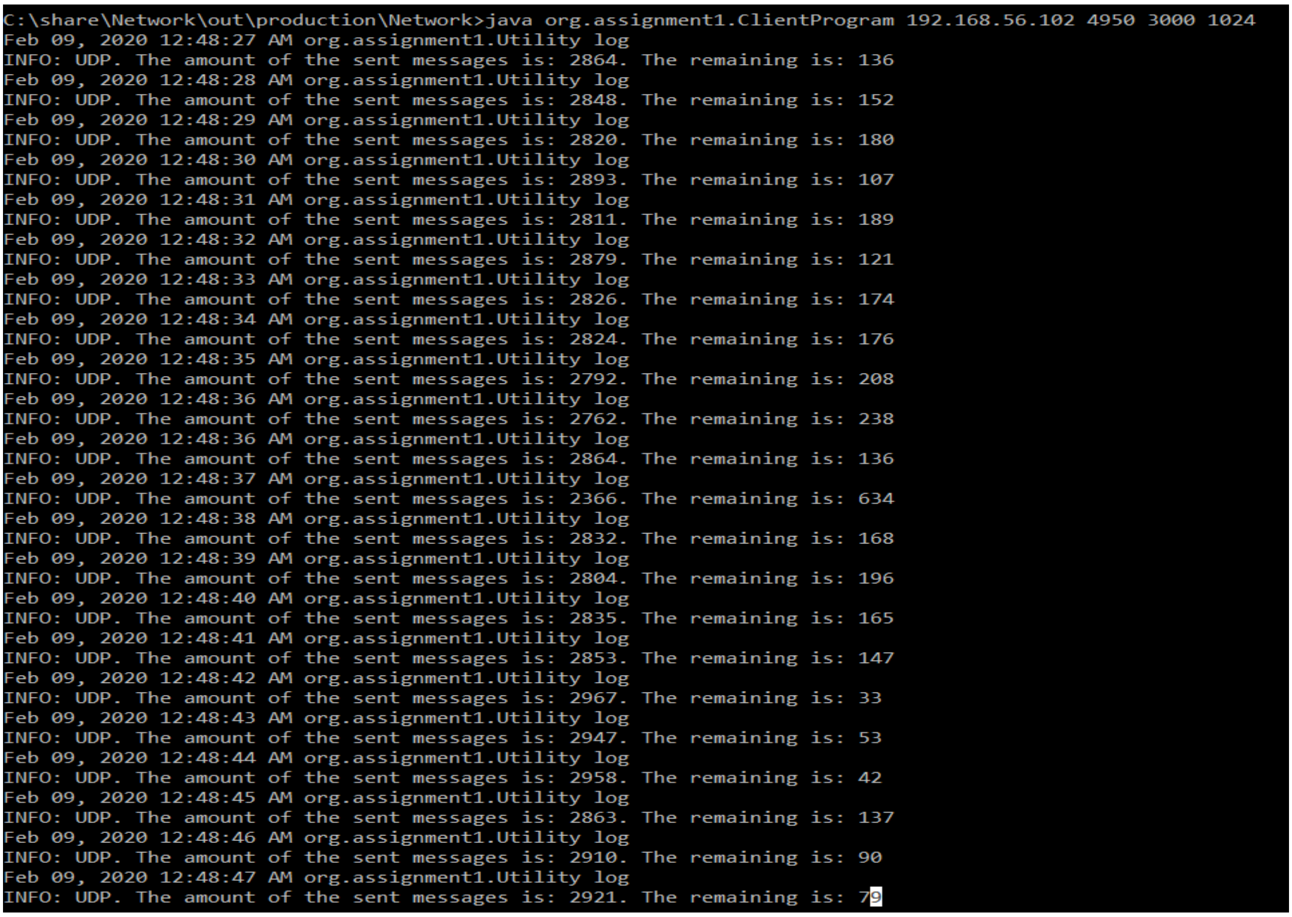
The above Figures 2, 3 and 4 show that the client sends fives message per second to the server and the server echoes the messages back to the client. The duration of the operation is approximately five seconds.

When it comes to handling exceptions, the arguments should be validated when the user provides them to ensure the connection occurs in the best way possible. Without valid values, the connection will not occur, therefore, the exceptions below are necessary when it comes to establishing the connection gracefully and to warn the user of any invalid value or error in case there is an issue.

To make the connection efficient, then the application should not wait unnecessarily for the source to sort its issues or waste unnecessary resources with no positive result. Therefore the values should be correct from the start to establish valuable networking.

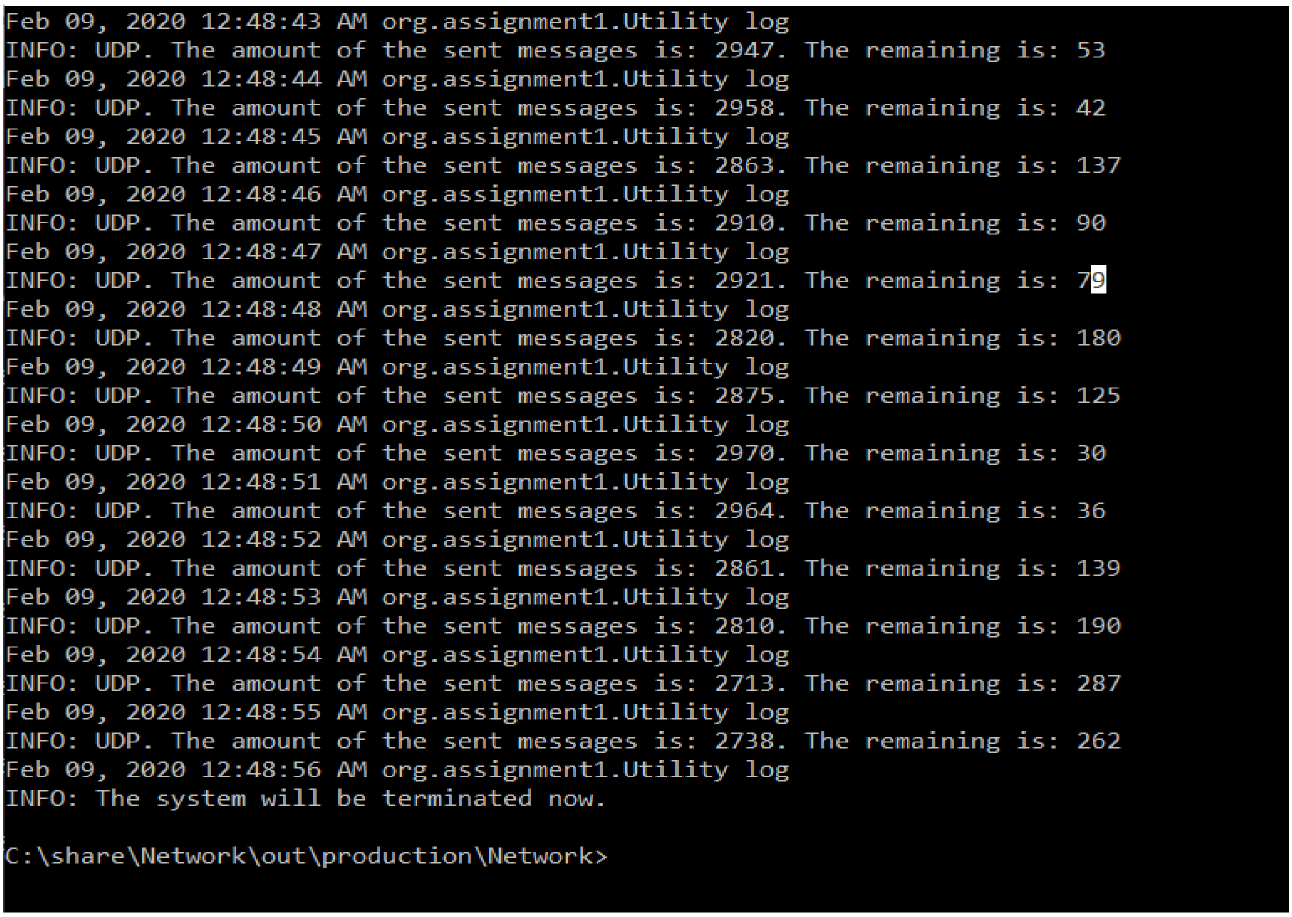
Exception	**Motivation**
Arguments are null	To ensure that the array of the arguments is valid.
IP address is invalid	To ensure that the provided address is valid.
Illegal amount of arguments	To ensure the right amount of arguments is provided, which is four arguments.
The size of the buffer is invalid	To ensure the buffer size does not exceeds the size of the memory of the machine and it is with acceptable range.
The format of the arguments are invalid	To ensure the format is valid.
The port number is invalid	To ensure the port number does exist and valid.
Negative value	To ensure there is no invalid value, since all the arguments do not support negative values.
Illegal package limit	To ensure the package limit is valid.
IO Exception	To handle input and out put errors.
Interruption Exception	To handle any interruption when the thread is on sleep.
Socket Exception	To handle any error related to sockets in terms of failing to connect, close, send and receive.

2.2 Problem VG-1



```
C:\share\Network\out\production\Network>java org.assignment1.ClientProgram 192.168.56.102 4950 3000 1024
Feb 09, 2020 12:48:27 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2864. The remaining is: 136
Feb 09, 2020 12:48:28 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2848. The remaining is: 152
Feb 09, 2020 12:48:29 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2820. The remaining is: 180
Feb 09, 2020 12:48:30 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2893. The remaining is: 107
Feb 09, 2020 12:48:31 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2811. The remaining is: 189
Feb 09, 2020 12:48:32 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2879. The remaining is: 121
Feb 09, 2020 12:48:33 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2826. The remaining is: 174
Feb 09, 2020 12:48:34 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2824. The remaining is: 176
Feb 09, 2020 12:48:35 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2792. The remaining is: 208
Feb 09, 2020 12:48:36 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2762. The remaining is: 238
Feb 09, 2020 12:48:36 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2864. The remaining is: 136
Feb 09, 2020 12:48:37 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2366. The remaining is: 634
Feb 09, 2020 12:48:38 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2832. The remaining is: 168
Feb 09, 2020 12:48:39 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2804. The remaining is: 196
Feb 09, 2020 12:48:40 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2835. The remaining is: 165
Feb 09, 2020 12:48:41 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2853. The remaining is: 147
Feb 09, 2020 12:48:42 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2967. The remaining is: 33
Feb 09, 2020 12:48:43 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2947. The remaining is: 53
Feb 09, 2020 12:48:44 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2958. The remaining is: 42
Feb 09, 2020 12:48:45 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2863. The remaining is: 137
Feb 09, 2020 12:48:46 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2910. The remaining is: 90
Feb 09, 2020 12:48:47 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2921. The remaining is: 79
```

Figure 5: UDP: The host sends 3000 messages per second to the receiver.



```
Feb 09, 2020 12:48:43 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2947. The remaining is: 53
Feb 09, 2020 12:48:44 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2958. The remaining is: 42
Feb 09, 2020 12:48:45 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2863. The remaining is: 137
Feb 09, 2020 12:48:46 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2910. The remaining is: 90
Feb 09, 2020 12:48:47 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2921. The remaining is: 79
Feb 09, 2020 12:48:48 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2820. The remaining is: 180
Feb 09, 2020 12:48:49 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2875. The remaining is: 125
Feb 09, 2020 12:48:50 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2970. The remaining is: 30
Feb 09, 2020 12:48:51 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2964. The remaining is: 36
Feb 09, 2020 12:48:52 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2861. The remaining is: 139
Feb 09, 2020 12:48:53 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2810. The remaining is: 190
Feb 09, 2020 12:48:54 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2713. The remaining is: 287
Feb 09, 2020 12:48:55 AM org.assignment1.Utility log
INFO: UDP. The amount of the sent messages is: 2738. The remaining is: 262
Feb 09, 2020 12:48:56 AM org.assignment1.Utility log
INFO: The system will be terminated now.

C:\share\Network\out\production\Network>
```

Figure 6: UDP: The host sends 3000 messages per second to the receiver.

2.2.1 Discussion

In Figure 5 and 6, the echo messages are not displayed to avoid making the screenshots be unnecessarily sizeable.

Figure 5 and 6 show the host sends 3000 messages to the receiver for approximately 30 seconds.

The result demonstrates that from one second to another, the amount of the sent message varies. The UDP protocol does not establish any connection to communicate with the server socket. The size of the package is limited.

It is fair to assume the host and the server do not coordinate in terms of agreeing on the transmission or ensuring and confirming that the entire data is received or sent. It can also mean the server is communicating with numerous hosts at the same time, and that can affect the data and its integrity.

Moreover, from the results, it can mean; there is no data is tracking, and sending a massive load of data over a short period can cause a stand, where the server cannot keep up to receive them since it can be that the server is sending and receiving data with the other host at the same time. In this case, it is not likely that the server was sending and receiving from another host, and that can lead to the assumption that sending a high load of data can lead to data loss since there is no coordination in terms of agreeing how the data will be transferred and protected.

2.3 Problem VG-2

2.3.1 Discussion

The abstract class *Client* is the base class for both the UDP and TCP echo classes. Both UDP and TCP classes have standard parameters: IP address, destination port number, the transfer rate per second, and the buffer size. These arguments have to be validated in both classes. Therefore it was decided to validate the arguments in the abstract class.

As far as the requirements were understood, the clients classes should accept the arguments via the command control and as for the server classes, the requirements did not specify the same condition for them and that implies the server classes were implemented in that way, where they do not accept arguments via the command control.

The rquirements were understod that when the rate is zero, then the host should send a single message and terminates afterwards. When the rate is higher than zero, then it should send the corresponding rate over the second and restarts the same operation when the second elapses, and the next one starts. That implies the host should operate infinitely. However, in this implementation, the host can function infinitely or finitely, depends on the objective of the user. Both client classes UDP and TCP run endlessly by default unless the user explicitly defines the run time in seconds.

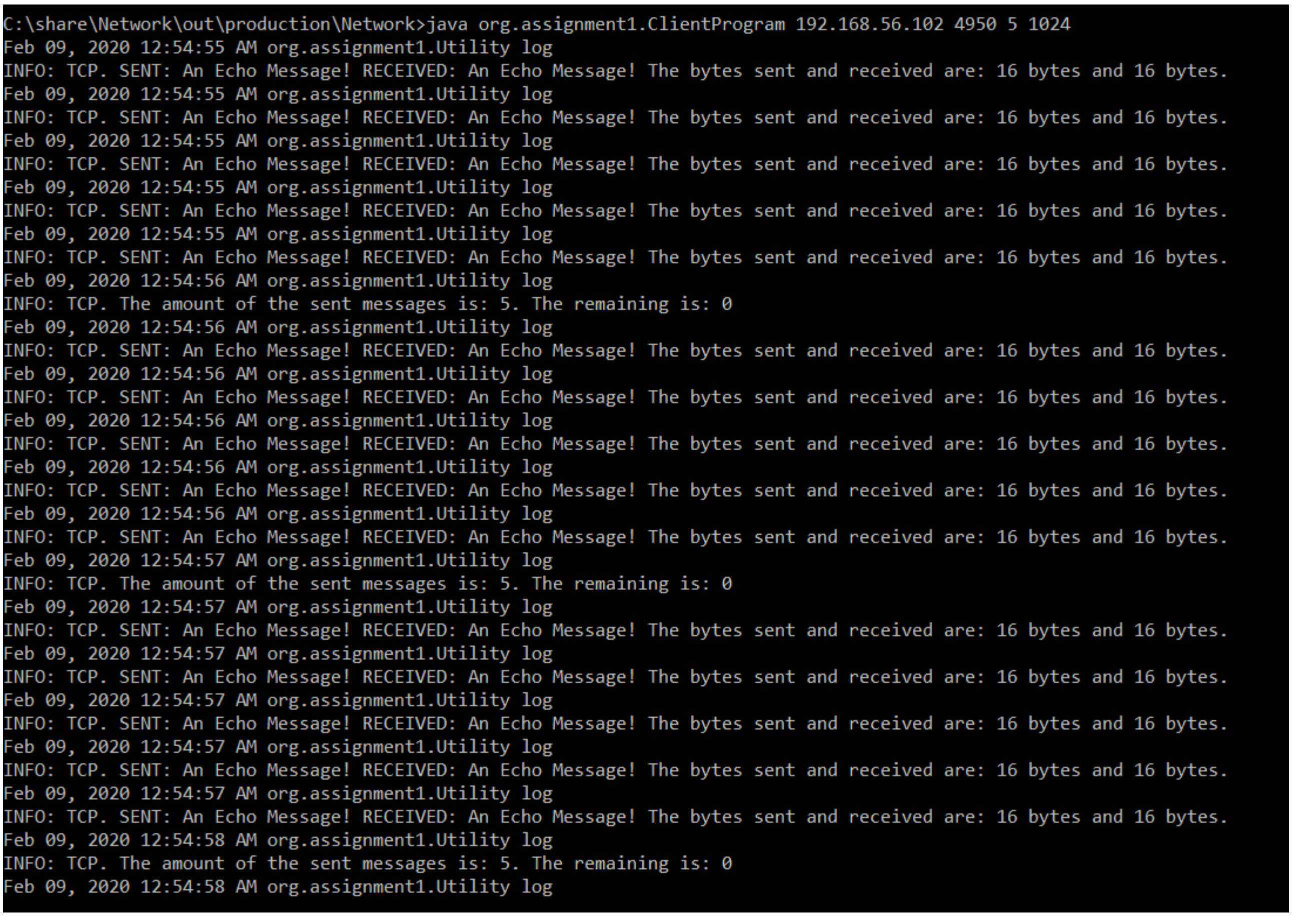
With the explanation above, that led to make the abstract class has a standard method to start the function, which is the start() method, it is a skeleton method with default validation methods, the default methods to validate the arguments, and initializes the essential values for both classes. In this method, there is an abstract method called run(), the run() method should be implemented by the child class depending on its objective. It is responsible for establishing the connection, maintaining it and running it.

The functions that are responsible for checking the status of the transfer rate, and whether the target time has elapsed, are in the abstract class since they are useful for both classes.

The method that conducts comparison to check whether the received message equals the sent message is in the abstract class. Moreover, the abstract class provide useful method for both classes such as method counts the elapsing time, method to terminate the program when an error occurs and logger method.

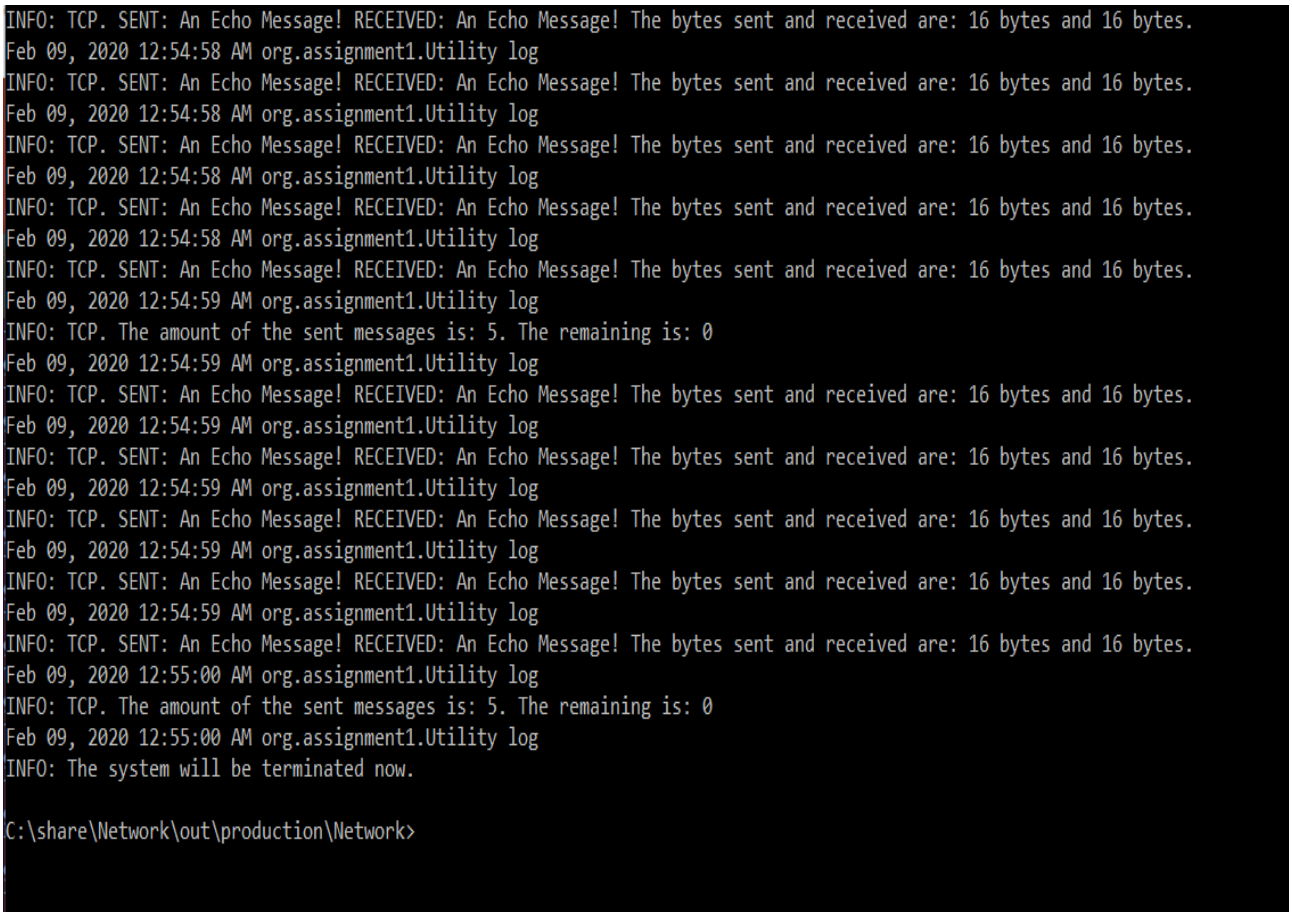
The abstract class uses logger; it is useful in terms of documenting the time and the date, and since this assignment is about establishing connections and transmitting data over a specified rate.

3 Problem 3



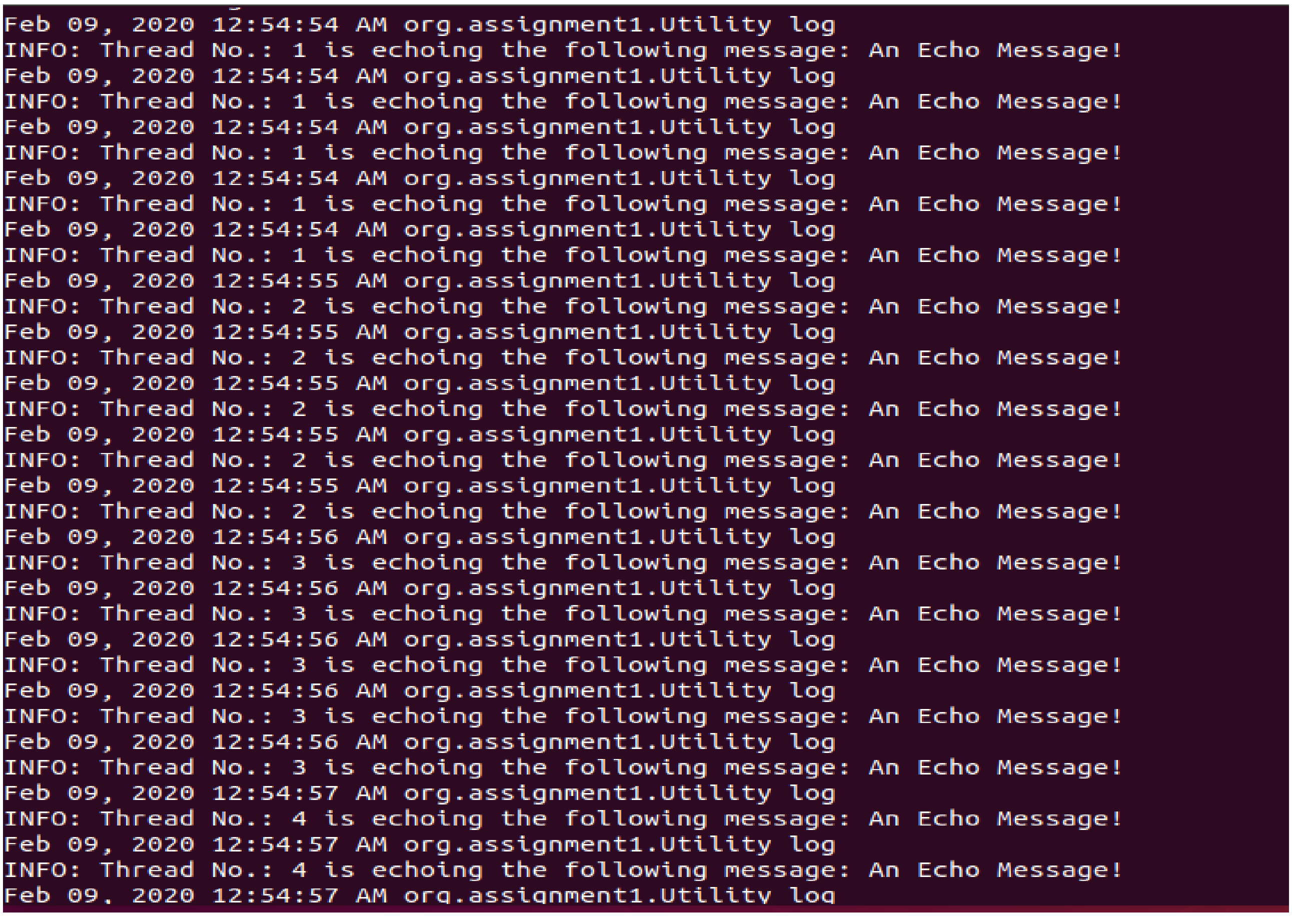
```
C:\share\Network\out\production\Network>java org.assignment1.ClientProgram 192.168.56.102 4950 5 1024
Feb 09, 2020 12:54:55 AM org.assignment1.Utility log
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:54:55 AM org.assignment1.Utility log
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:54:55 AM org.assignment1.Utility log
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:54:55 AM org.assignment1.Utility log
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:54:55 AM org.assignment1.Utility log
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:54:56 AM org.assignment1.Utility log
INFO: TCP. The amount of the sent messages is: 5. The remaining is: 0
Feb 09, 2020 12:54:56 AM org.assignment1.Utility log
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:54:56 AM org.assignment1.Utility log
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:54:56 AM org.assignment1.Utility log
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:54:56 AM org.assignment1.Utility log
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:54:56 AM org.assignment1.Utility log
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:54:57 AM org.assignment1.Utility log
INFO: TCP. The amount of the sent messages is: 5. The remaining is: 0
Feb 09, 2020 12:54:57 AM org.assignment1.Utility log
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:54:57 AM org.assignment1.Utility log
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:54:57 AM org.assignment1.Utility log
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:54:57 AM org.assignment1.Utility log
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:54:57 AM org.assignment1.Utility log
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:54:58 AM org.assignment1.Utility log
INFO: TCP. The amount of the sent messages is: 5. The remaining is: 0
Feb 09, 2020 12:54:58 AM org.assignment1.Utility log
```

Figure 7: TCP: The host sends five message per second.



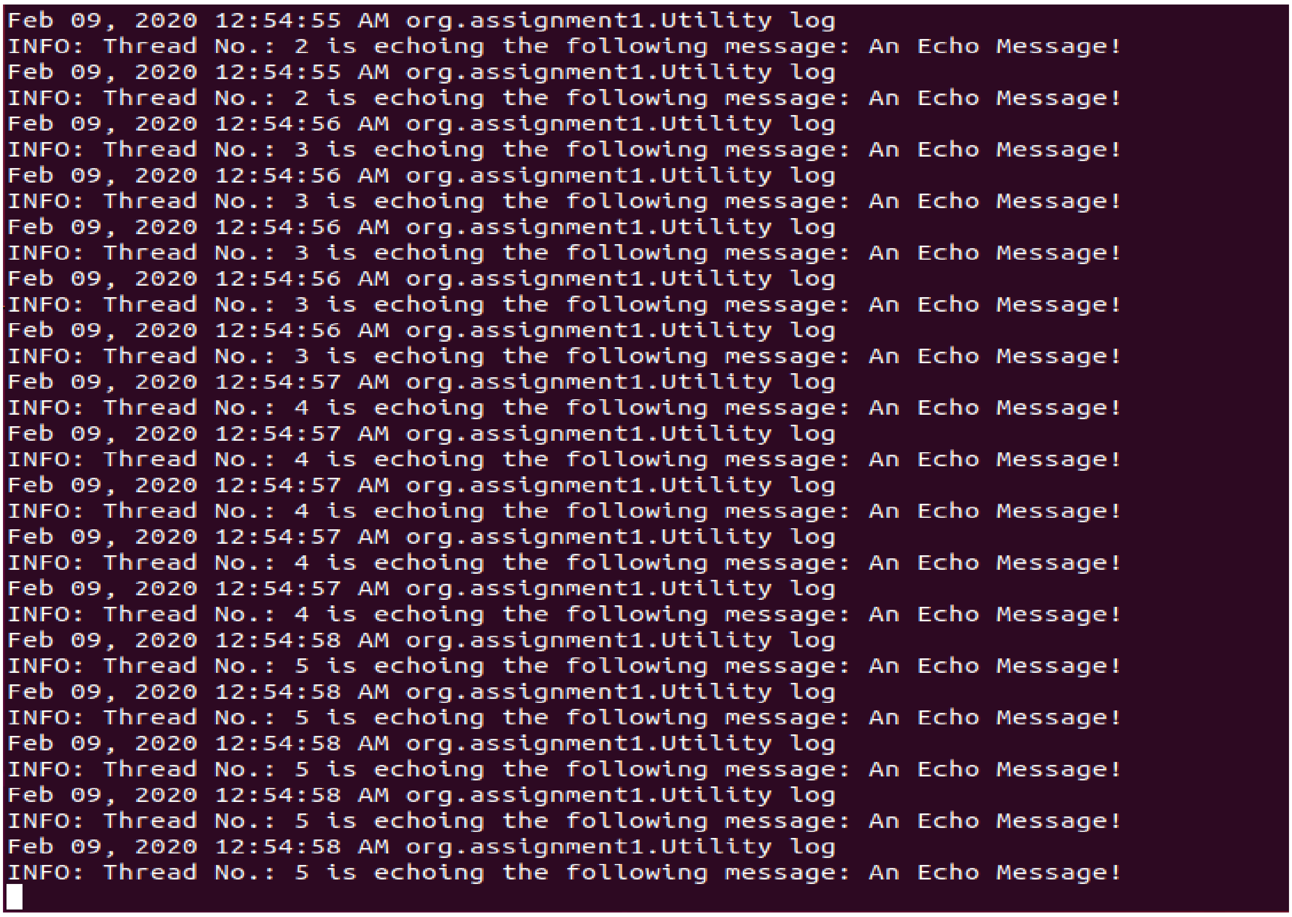
```
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:54:58 AM org.assignment1.Utility log
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:54:58 AM org.assignment1.Utility log
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:54:58 AM org.assignment1.Utility log
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:54:58 AM org.assignment1.Utility log
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:54:59 AM org.assignment1.Utility log
INFO: TCP. The amount of the sent messages is: 5. The remaining is: 0
Feb 09, 2020 12:54:59 AM org.assignment1.Utility log
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:54:59 AM org.assignment1.Utility log
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:54:59 AM org.assignment1.Utility log
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:54:59 AM org.assignment1.Utility log
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:54:59 AM org.assignment1.Utility log
INFO: TCP. SENT: An Echo Message! RECEIVED: An Echo Message! The bytes sent and received are: 16 bytes and 16 bytes.
Feb 09, 2020 12:55:00 AM org.assignment1.Utility log
INFO: TCP. The amount of the sent messages is: 5. The remaining is: 0
Feb 09, 2020 12:55:00 AM org.assignment1.Utility log
INFO: The system will be terminated now.

C:\share\Network\out\production\Network>
```

Figure 8: TCP: The host sends five message per second.



```
Feb 09, 2020 12:54:54 AM org.assignment1.Utility log
INFO: Thread No.: 1 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:54 AM org.assignment1.Utility log
INFO: Thread No.: 1 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:54 AM org.assignment1.Utility log
INFO: Thread No.: 1 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:54 AM org.assignment1.Utility log
INFO: Thread No.: 1 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:54 AM org.assignment1.Utility log
INFO: Thread No.: 1 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:55 AM org.assignment1.Utility log
INFO: Thread No.: 2 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:55 AM org.assignment1.Utility log
INFO: Thread No.: 2 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:55 AM org.assignment1.Utility log
INFO: Thread No.: 2 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:55 AM org.assignment1.Utility log
INFO: Thread No.: 2 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:55 AM org.assignment1.Utility log
INFO: Thread No.: 2 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:56 AM org.assignment1.Utility log
INFO: Thread No.: 3 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:56 AM org.assignment1.Utility log
INFO: Thread No.: 3 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:56 AM org.assignment1.Utility log
INFO: Thread No.: 3 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:56 AM org.assignment1.Utility log
INFO: Thread No.: 3 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:56 AM org.assignment1.Utility log
INFO: Thread No.: 3 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:57 AM org.assignment1.Utility log
INFO: Thread No.: 4 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:57 AM org.assignment1.Utility log
INFO: Thread No.: 4 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:57 AM org.assignment1.Utility log
```

Figure 9: TCP: The server handles each connection in separate thread.



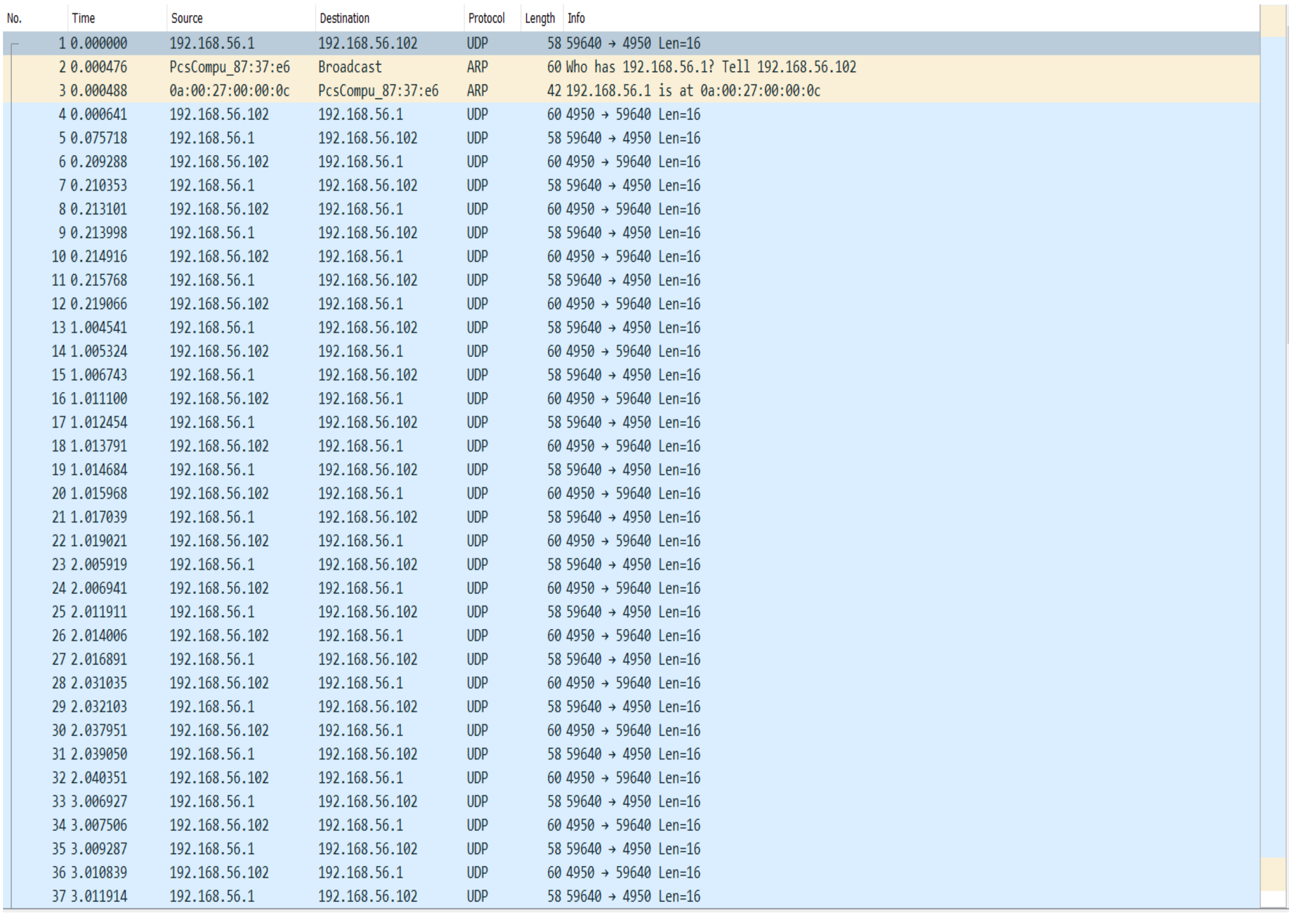
```
Feb 09, 2020 12:54:55 AM org.assignment1.Utility log
INFO: Thread No.: 2 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:55 AM org.assignment1.Utility log
INFO: Thread No.: 2 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:56 AM org.assignment1.Utility log
INFO: Thread No.: 3 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:56 AM org.assignment1.Utility log
INFO: Thread No.: 3 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:56 AM org.assignment1.Utility log
INFO: Thread No.: 3 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:56 AM org.assignment1.Utility log
INFO: Thread No.: 3 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:56 AM org.assignment1.Utility log
INFO: Thread No.: 3 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:57 AM org.assignment1.Utility log
INFO: Thread No.: 4 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:57 AM org.assignment1.Utility log
INFO: Thread No.: 4 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:57 AM org.assignment1.Utility log
INFO: Thread No.: 4 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:57 AM org.assignment1.Utility log
INFO: Thread No.: 4 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:57 AM org.assignment1.Utility log
INFO: Thread No.: 4 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:58 AM org.assignment1.Utility log
INFO: Thread No.: 5 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:58 AM org.assignment1.Utility log
INFO: Thread No.: 5 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:58 AM org.assignment1.Utility log
INFO: Thread No.: 5 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:58 AM org.assignment1.Utility log
INFO: Thread No.: 5 is echoing the following message: An Echo Message!
Feb 09, 2020 12:54:58 AM org.assignment1.Utility log
INFO: Thread No.: 5 is echoing the following message: An Echo Message!
```

Figure 10: TCP: The server handles each connection in separate thread.

3.1 Discussion

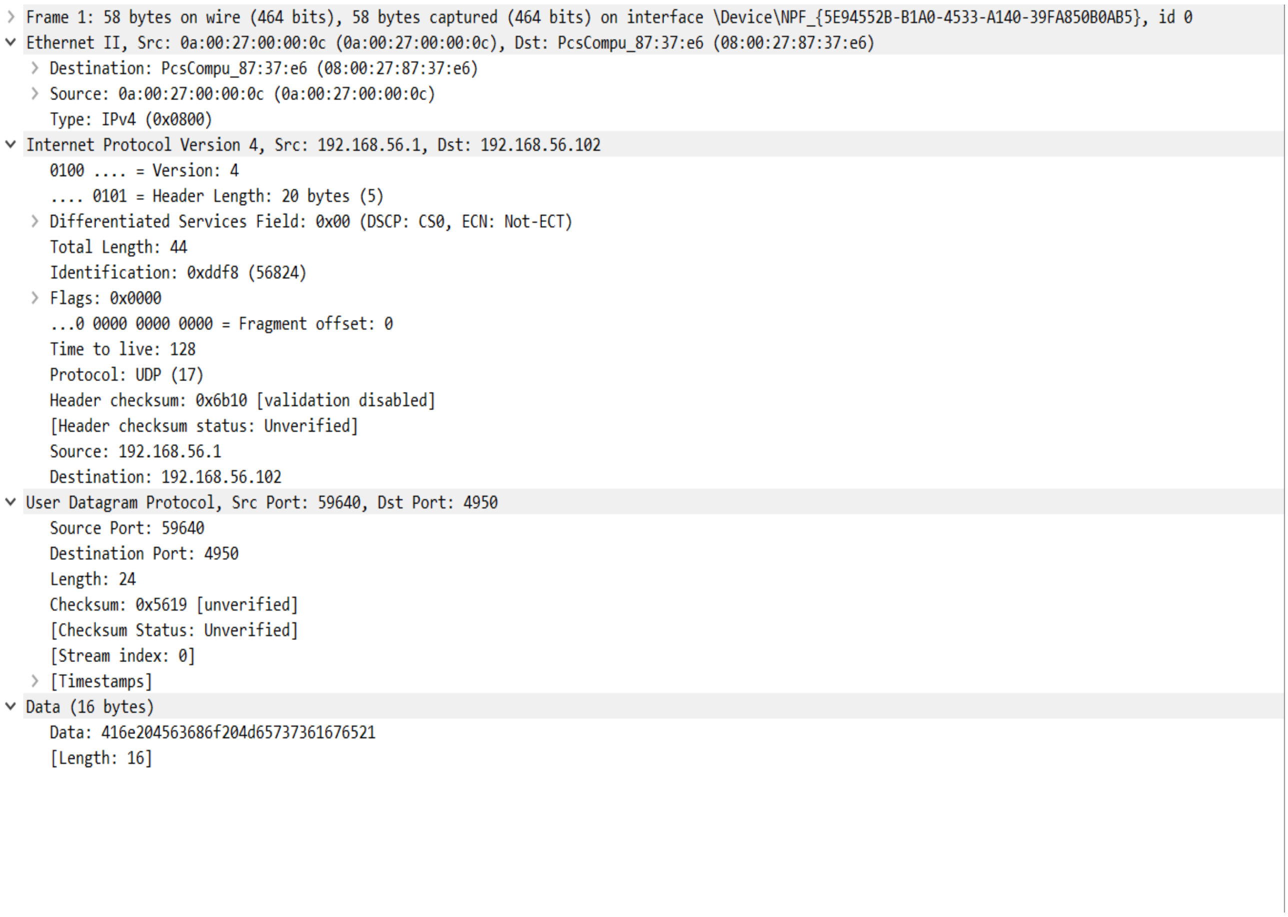
Figures 7, 8, 9, and 10 show the TCP echo class sends five messages per second to the TCP server class, and the server handles each connection in a separate thread. where it echos back the messages. The operation was conducted for approximately five seconds.

4 Problem 4



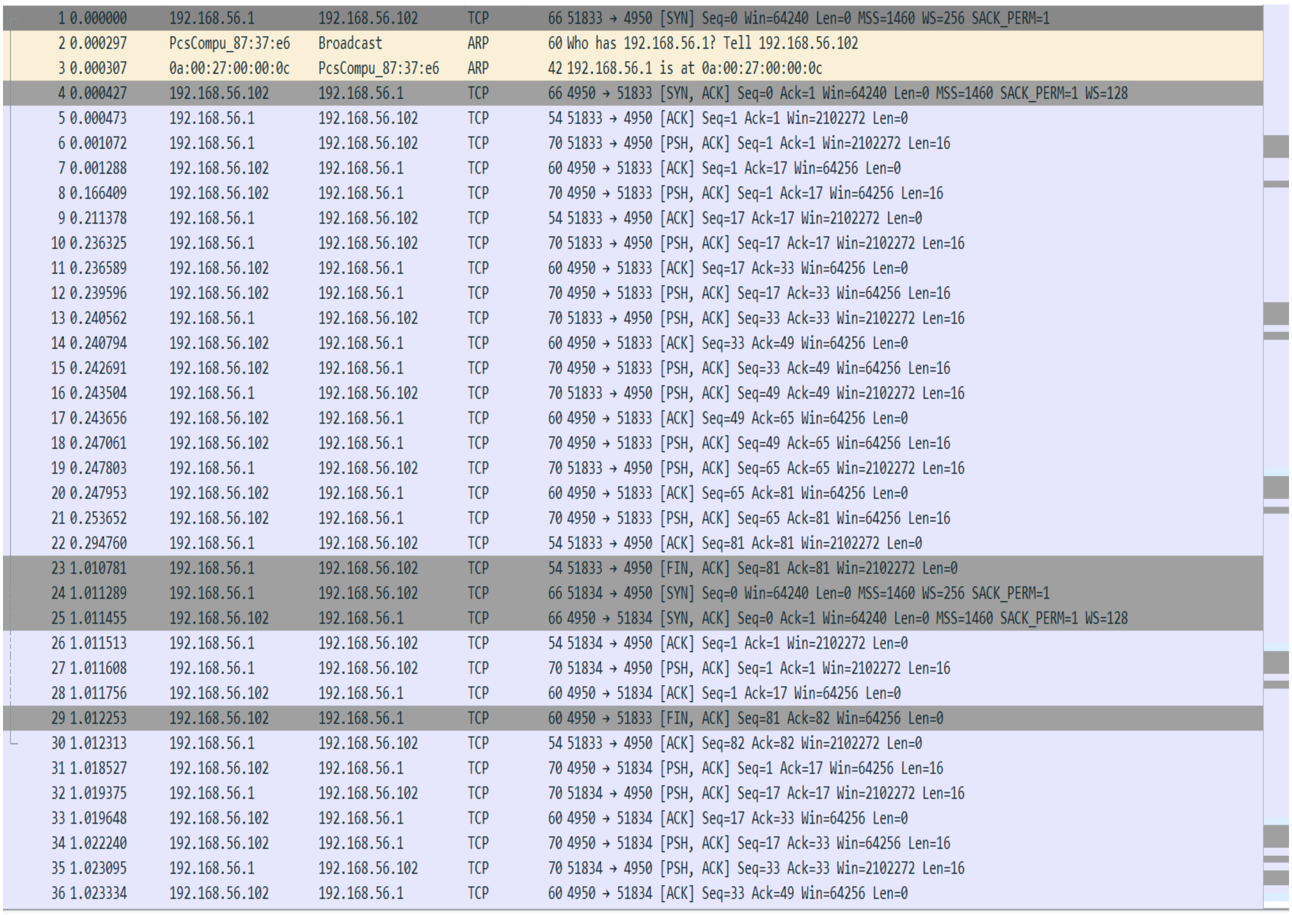
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.56.1	192.168.56.102	UDP	58	59640 → 4950 Len=16
2	0.000476	PcsCompu_87:37:e6	Broadcast	ARP	60	Who has 192.168.56.1? Tell 192.168.56.102
3	0.000488	0a:00:27:00:00:0c	PcsCompu_87:37:e6	ARP	42	192.168.56.1 is at 0a:00:27:00:00:0c
4	0.000641	192.168.56.102	192.168.56.1	UDP	60	4950 → 59640 Len=16
5	0.075718	192.168.56.1	192.168.56.102	UDP	58	59640 → 4950 Len=16
6	0.209288	192.168.56.102	192.168.56.1	UDP	60	4950 → 59640 Len=16
7	0.210353	192.168.56.1	192.168.56.102	UDP	58	59640 → 4950 Len=16
8	0.213101	192.168.56.102	192.168.56.1	UDP	60	4950 → 59640 Len=16
9	0.213998	192.168.56.1	192.168.56.102	UDP	58	59640 → 4950 Len=16
10	0.214916	192.168.56.102	192.168.56.1	UDP	60	4950 → 59640 Len=16
11	0.215768	192.168.56.1	192.168.56.102	UDP	58	59640 → 4950 Len=16
12	0.219066	192.168.56.102	192.168.56.1	UDP	60	4950 → 59640 Len=16
13	1.004541	192.168.56.1	192.168.56.102	UDP	58	59640 → 4950 Len=16
14	1.005324	192.168.56.102	192.168.56.1	UDP	60	4950 → 59640 Len=16
15	1.006743	192.168.56.1	192.168.56.102	UDP	58	59640 → 4950 Len=16
16	1.011100	192.168.56.102	192.168.56.1	UDP	60	4950 → 59640 Len=16
17	1.012454	192.168.56.1	192.168.56.102	UDP	58	59640 → 4950 Len=16
18	1.013791	192.168.56.102	192.168.56.1	UDP	60	4950 → 59640 Len=16
19	1.014684	192.168.56.1	192.168.56.102	UDP	58	59640 → 4950 Len=16
20	1.015968	192.168.56.102	192.168.56.1	UDP	60	4950 → 59640 Len=16
21	1.017039	192.168.56.1	192.168.56.102	UDP	58	59640 → 4950 Len=16
22	1.019021	192.168.56.102	192.168.56.1	UDP	60	4950 → 59640 Len=16
23	2.005919	192.168.56.1	192.168.56.102	UDP	58	59640 → 4950 Len=16
24	2.006941	192.168.56.102	192.168.56.1	UDP	60	4950 → 59640 Len=16
25	2.011911	192.168.56.1	192.168.56.102	UDP	58	59640 → 4950 Len=16
26	2.014006	192.168.56.102	192.168.56.1	UDP	60	4950 → 59640 Len=16
27	2.016891	192.168.56.1	192.168.56.102	UDP	58	59640 → 4950 Len=16
28	2.031035	192.168.56.102	192.168.56.1	UDP	60	4950 → 59640 Len=16
29	2.032103	192.168.56.1	192.168.56.102	UDP	58	59640 → 4950 Len=16
30	2.037951	192.168.56.102	192.168.56.1	UDP	60	4950 → 59640 Len=16
31	2.039050	192.168.56.1	192.168.56.102	UDP	58	59640 → 4950 Len=16
32	2.040351	192.168.56.102	192.168.56.1	UDP	60	4950 → 59640 Len=16
33	3.006927	192.168.56.1	192.168.56.102	UDP	58	59640 → 4950 Len=16
34	3.007506	192.168.56.102	192.168.56.1	UDP	60	4950 → 59640 Len=16
35	3.009287	192.168.56.1	192.168.56.102	UDP	58	59640 → 4950 Len=16
36	3.010839	192.168.56.102	192.168.56.1	UDP	60	4950 → 59640 Len=16
37	3.011914	192.168.56.1	192.168.56.102	UDP	58	59640 → 4950 Len=16

Figure 11: UDP transfer.



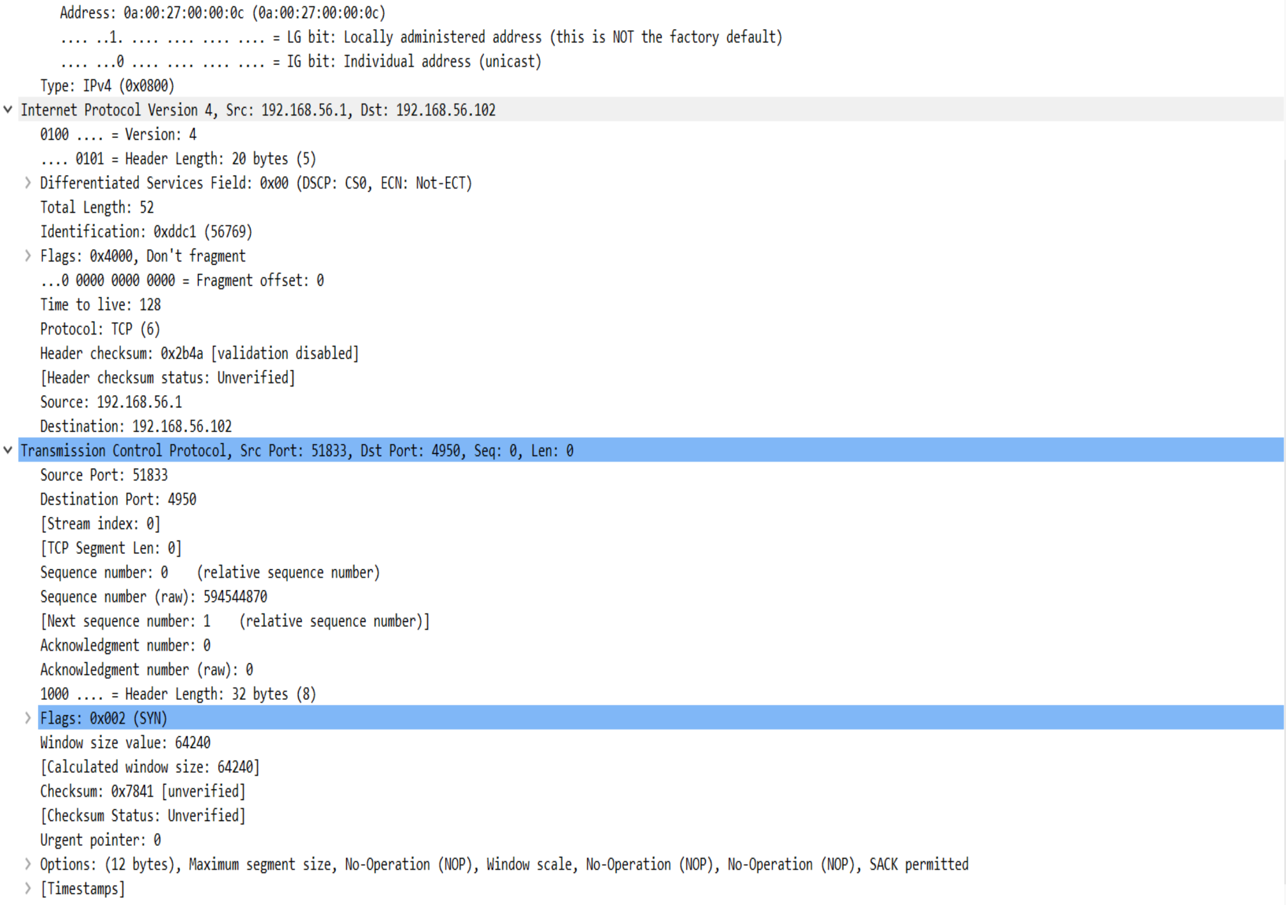
```
Frame 1: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface \Device\NPF_{5E94552B-B1A0-4533-A140-39FA850B0AB5}, id 0
Ethernet II, Src: 0a:00:27:00:00:0c (0a:00:27:00:00:0c), Dst: PcsCompu_87:37:e6 (08:00:27:87:37:e6)
  Destination: PcsCompu_87:37:e6 (08:00:27:87:37:e6)
  Source: 0a:00:27:00:00:0c (0a:00:27:00:00:0c)
  Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.102
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 44
  Identification: 0xddf8 (56824)
  Flags: 0x0000
  ...0 0000 0000 0000 = Fragment offset: 0
  Time to live: 128
  Protocol: UDP (17)
  Header checksum: 0x6b10 [validation disabled]
  [Header checksum status: Unverified]
  Source: 192.168.56.1
  Destination: 192.168.56.102
User Datagram Protocol, Src Port: 59640, Dst Port: 4950
  Source Port: 59640
  Destination Port: 4950
  Length: 24
  Checksum: 0x5619 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 0]
  [Timestamps]
Data (16 bytes)
  Data: 416e204563686f204d65737361676521
  [Length: 16]
```

Figure 12: UDP transfer.



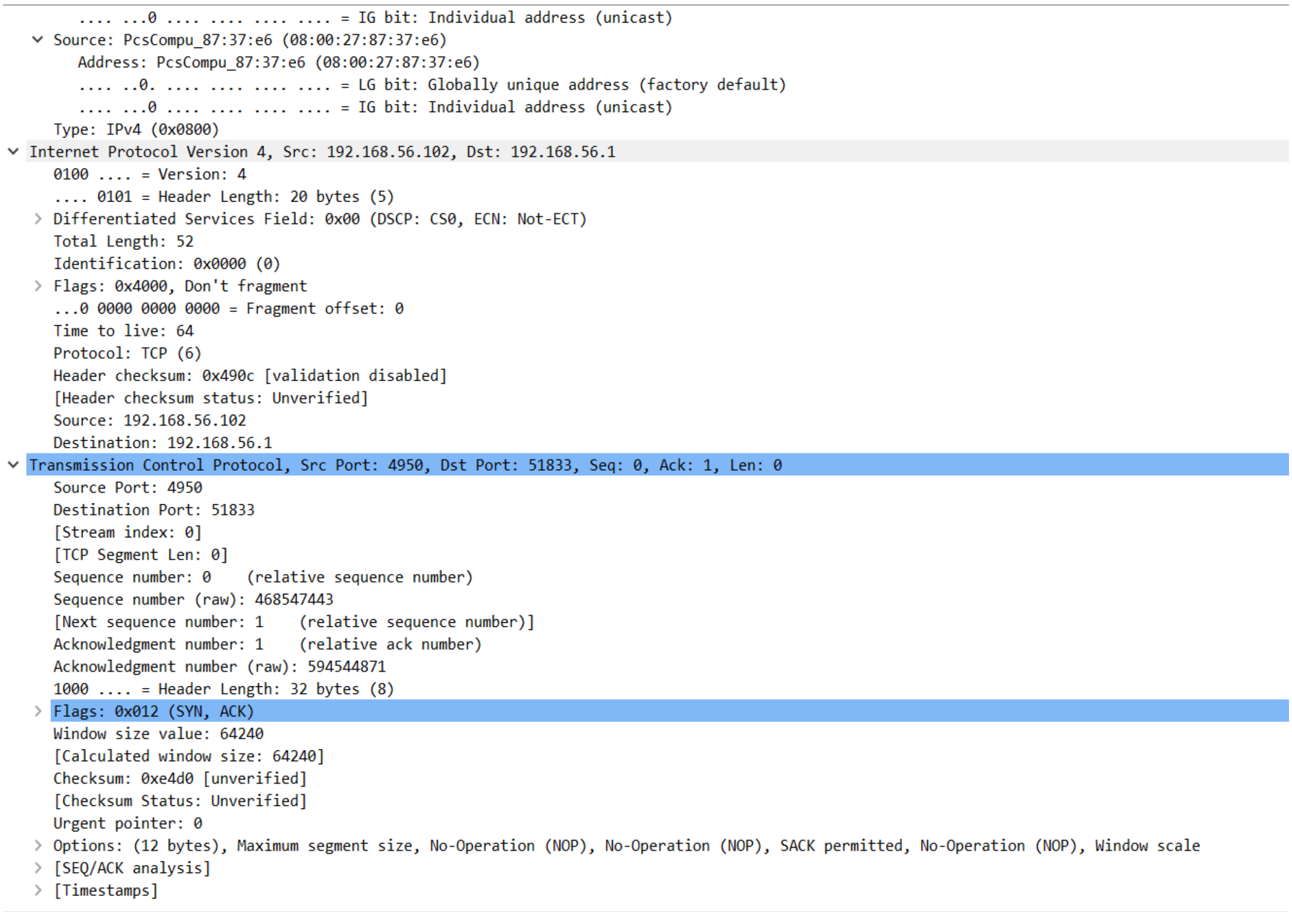
```
 1 0.000000   192.168.56.1       192.168.56.102     TCP   66 51833 → 4950 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
 2 0.000297   PcsCompu_87:37:e6  Broadcast          ARP   60 Who has 192.168.56.1? Tell 192.168.56.102
 3 0.000307   0a:00:27:00:00:0c  PcsCompu_87:37:e6  ARP   42 192.168.56.1 is at 0a:00:27:00:00:0c
 4 0.000427   192.168.56.102     192.168.56.1       TCP   66 4950 → 51833 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
 5 0.000473   192.168.56.1       192.168.56.102     TCP   54 51833 → 4950 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
 6 0.001072   192.168.56.1       192.168.56.102     TCP   70 51833 → 4950 [PSH, ACK] Seq=1 Ack=1 Win=2102272 Len=16
 7 0.001288   192.168.56.102     192.168.56.1       TCP   60 4950 → 51833 [ACK] Seq=1 Ack=17 Win=64256 Len=0
 8 0.166409   192.168.56.102     192.168.56.1       TCP   70 4950 → 51833 [PSH, ACK] Seq=1 Ack=17 Win=64256 Len=16
 9 0.211378   192.168.56.1       192.168.56.102     TCP   54 51833 → 4950 [ACK] Seq=17 Ack=17 Win=2102272 Len=0
10 0.236325   192.168.56.1       192.168.56.102     TCP   70 51833 → 4950 [PSH, ACK] Seq=17 Ack=17 Win=2102272 Len=16
11 0.236589   192.168.56.102     192.168.56.1       TCP   60 4950 → 51833 [ACK] Seq=17 Ack=33 Win=64256 Len=0
12 0.239596   192.168.56.102     192.168.56.1       TCP   70 4950 → 51833 [PSH, ACK] Seq=17 Ack=33 Win=64256 Len=16
13 0.240562   192.168.56.1       192.168.56.102     TCP   70 51833 → 4950 [PSH, ACK] Seq=33 Ack=33 Win=2102272 Len=16
14 0.240794   192.168.56.102     192.168.56.1       TCP   60 4950 → 51833 [ACK] Seq=33 Ack=49 Win=64256 Len=0
15 0.242691   192.168.56.102     192.168.56.1       TCP   70 4950 → 51833 [PSH, ACK] Seq=33 Ack=49 Win=64256 Len=16
16 0.243504   192.168.56.1       192.168.56.102     TCP   70 51833 → 4950 [PSH, ACK] Seq=49 Ack=49 Win=2102272 Len=16
17 0.243656   192.168.56.102     192.168.56.1       TCP   60 4950 → 51833 [ACK] Seq=49 Ack=65 Win=64256 Len=0
18 0.247061   192.168.56.102     192.168.56.1       TCP   70 4950 → 51833 [PSH, ACK] Seq=49 Ack=65 Win=64256 Len=16
19 0.247803   192.168.56.1       192.168.56.102     TCP   70 51833 → 4950 [PSH, ACK] Seq=65 Ack=65 Win=2102272 Len=16
20 0.247953   192.168.56.102     192.168.56.1       TCP   60 4950 → 51833 [ACK] Seq=65 Ack=81 Win=64256 Len=0
21 0.253652   192.168.56.102     192.168.56.1       TCP   70 4950 → 51833 [PSH, ACK] Seq=65 Ack=81 Win=64256 Len=16
22 0.294760   192.168.56.1       192.168.56.102     TCP   54 51833 → 4950 [ACK] Seq=81 Ack=81 Win=2102272 Len=0
23 1.010781   192.168.56.1       192.168.56.102     TCP   54 51833 → 4950 [FIN, ACK] Seq=81 Ack=81 Win=2102272 Len=0
24 1.011289   192.168.56.1       192.168.56.102     TCP   66 51834 → 4950 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
25 1.011455   192.168.56.102     192.168.56.1       TCP   66 4950 → 51834 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
26 1.011513   192.168.56.1       192.168.56.102     TCP   54 51834 → 4950 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
27 1.011608   192.168.56.1       192.168.56.102     TCP   70 51834 → 4950 [PSH, ACK] Seq=1 Ack=1 Win=2102272 Len=16
28 1.011756   192.168.56.102     192.168.56.1       TCP   60 4950 → 51834 [ACK] Seq=1 Ack=17 Win=64256 Len=0
29 1.012253   192.168.56.102     192.168.56.1       TCP   60 4950 → 51833 [FIN, ACK] Seq=81 Ack=82 Win=64256 Len=0
30 1.012313   192.168.56.1       192.168.56.102     TCP   54 51833 → 4950 [ACK] Seq=82 Ack=82 Win=2102272 Len=0
31 1.018527   192.168.56.102     192.168.56.1       TCP   70 4950 → 51834 [PSH, ACK] Seq=1 Ack=17 Win=64256 Len=16
32 1.019375   192.168.56.1       192.168.56.102     TCP   70 51834 → 4950 [PSH, ACK] Seq=17 Ack=17 Win=2102272 Len=16
33 1.019648   192.168.56.102     192.168.56.1       TCP   60 4950 → 51834 [ACK] Seq=17 Ack=33 Win=64256 Len=0
34 1.022240   192.168.56.102     192.168.56.1       TCP   70 4950 → 51834 [PSH, ACK] Seq=17 Ack=33 Win=64256 Len=16
35 1.023095   192.168.56.1       192.168.56.102     TCP   70 51834 → 4950 [PSH, ACK] Seq=33 Ack=33 Win=2102272 Len=16
36 1.023334   192.168.56.102     192.168.56.1       TCP   60 4950 → 51834 [ACK] Seq=33 Ack=49 Win=64256 Len=0
```

Figure 13: TCP transfer.



```
        Address: 0a:00:27:00:00:0c (0a:00:27:00:00:0c)
        .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NOT the factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
     Type: IPv4 (0x0800)
˅ Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.102
     0100 .... = Version: 4
     .... 0101 = Header Length: 20 bytes (5)
   › Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
     Total Length: 52
     Identification: 0xddc1 (56769)
   › Flags: 0x4000, Don't fragment
     ...0 0000 0000 0000 = Fragment offset: 0
     Time to live: 128
     Protocol: TCP (6)
     Header checksum: 0x2b4a [validation disabled]
     [Header checksum status: Unverified]
     Source: 192.168.56.1
     Destination: 192.168.56.102
˅ Transmission Control Protocol, Src Port: 51833, Dst Port: 4950, Seq: 0, Len: 0
     Source Port: 51833
     Destination Port: 4950
     [Stream index: 0]
     [TCP Segment Len: 0]
     Sequence number: 0    (relative sequence number)
     Sequence number (raw): 594544870
     [Next sequence number: 1    (relative sequence number)]
     Acknowledgment number: 0
     Acknowledgment number (raw): 0
     1000 .... = Header Length: 32 bytes (8)
   › Flags: 0x002 (SYN)
     Window size value: 64240
     [Calculated window size: 64240]
     Checksum: 0x7841 [unverified]
     [Checksum Status: Unverified]
     Urgent pointer: 0
   › Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted
   › [Timestamps]
```

Figure 14: TCP transfer.



```
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ˅ Source: PcsCompu_87:37:e6 (08:00:27:87:37:e6)
        Address: PcsCompu_87:37:e6 (08:00:27:87:37:e6)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
˅ Internet Protocol Version 4, Src: 192.168.56.102, Dst: 192.168.56.1
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  › Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 52
    Identification: 0x0000 (0)
  › Flags: 0x4000, Don't fragment
    ...0 0000 0000 0000 = Fragment offset: 0
    Time to live: 64
    Protocol: TCP (6)
    Header checksum: 0x490c [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.56.102
    Destination: 192.168.56.1
˅ Transmission Control Protocol, Src Port: 4950, Dst Port: 51833, Seq: 0, Ack: 1, Len: 0
    Source Port: 4950
    Destination Port: 51833
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    Sequence number (raw): 468547443
    [Next sequence number: 1    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    Acknowledgment number (raw): 594544871
    1000 .... = Header Length: 32 bytes (8)
  › Flags: 0x012 (SYN, ACK)
    Window size value: 64240
    [Calculated window size: 64240]
    Checksum: 0xe4d0 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  › Options: (12 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted, No-Operation (NOP), Window scale
  › [SEQ/ACK analysis]
  › [Timestamps]
```

Figure 15: TCP transfer.

4.1 Discussion

Figure 11 show the transmission between the host and the server. The transmission shows different information about the connection. It shows the time stamp, the IP address for the source and the destination, the length of the data, and the transfer protocol which is in this case is UDP. Line number one shows that the host sends data to the destination, and in line 4, the server echos back the data to the host.

In Figure 12, There are more details about the transmission such as the destination port number, source port number, header length, the data size, checksum status and more.

As for TCP, Figure 13 demonstrates the connection between the host and the server.

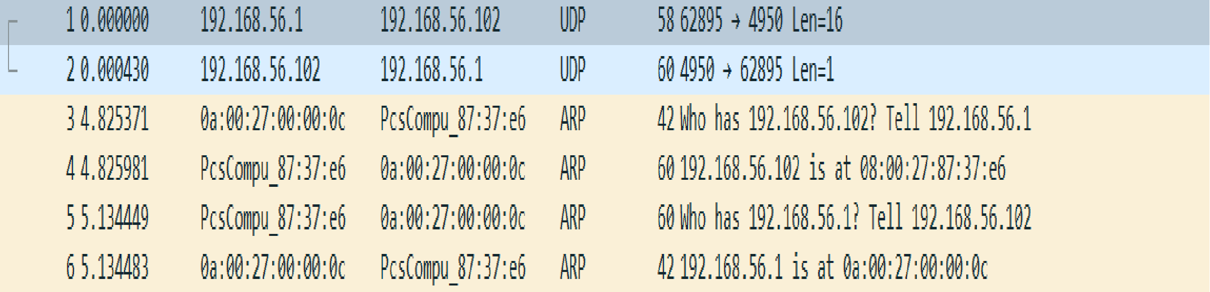
Line number one shows that the host communicates with the server by sending a synchronize message (SYN) to the server, and the sequence is zero. Line four demonstrates that the server sends a synchronize message (SYN) to the host and acknowledges (ACK) its connection by setting the value to one. The host responds again with acknowledgment, and the amount of sequences is 1, which is the sequence of the data segment. The server recognizes it and it sets the flag to push (PSH), which means the host can push its data. When the flag is set, the host sends the data overs. The host sent the data over to the server as one segment.

Figure 15 shows more details about the transmission, such as the sequence number, the flag, port detestation, the source port, length and protocol type. As it can be seen that the server is acknowledging the SYN message of the host, the sequence number is zero since it did not send anything, since in this image the flag has not been set to PSH yet.

The below table shows the difference between UDP and TCP:

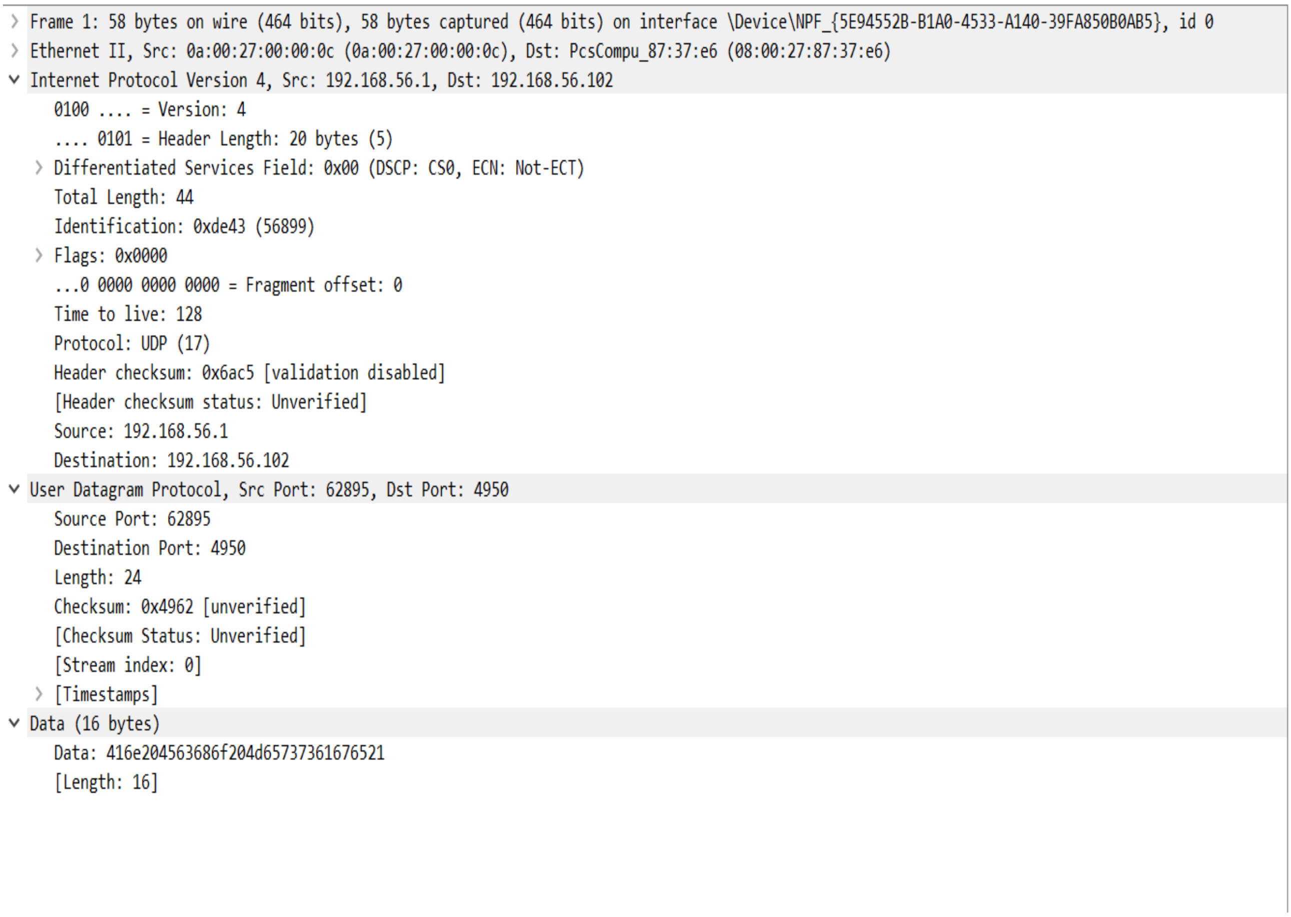
UDP	**TCP**
Connection-less	Connection-Oriented
Unreliable when it comes to data integrity	Reliable when it comes to data integrity
Limited package size	Arbitrary size
Fast	Slow
The order of the data is arbitrary	The order of the data is ordered
Supports many-to-many communication	Supports one-to-one communication

5 Problem 5



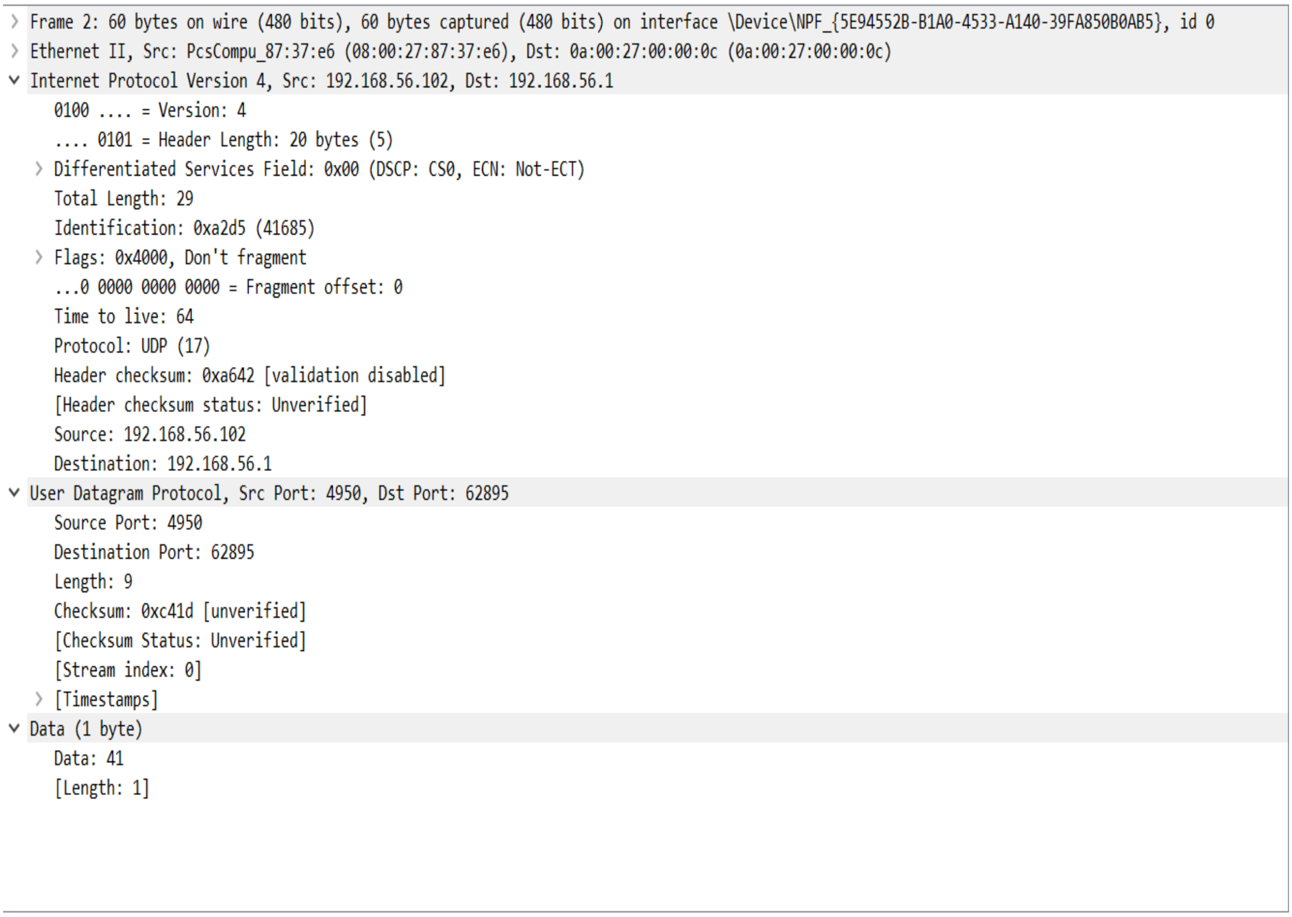
```
1 0.000000  192.168.56.1       192.168.56.102     UDP  58 62895 → 4950 Len=16
2 0.000430  192.168.56.102     192.168.56.1       UDP  60 4950 → 62895 Len=1
3 4.825371  0a:00:27:00:00:0c  PcsCompu_87:37:e6  ARP  42 Who has 192.168.56.102? Tell 192.168.56.1
4 4.825981  PcsCompu_87:37:e6  0a:00:27:00:00:0c  ARP  60 192.168.56.102 is at 08:00:27:87:37:e6
5 5.134449  PcsCompu_87:37:e6  0a:00:27:00:00:0c  ARP  60 Who has 192.168.56.1? Tell 192.168.56.102
6 5.134483  0a:00:27:00:00:0c  PcsCompu_87:37:e6  ARP  42 192.168.56.1 is at 0a:00:27:00:00:0c
```

Figure 16: UDP Transfer



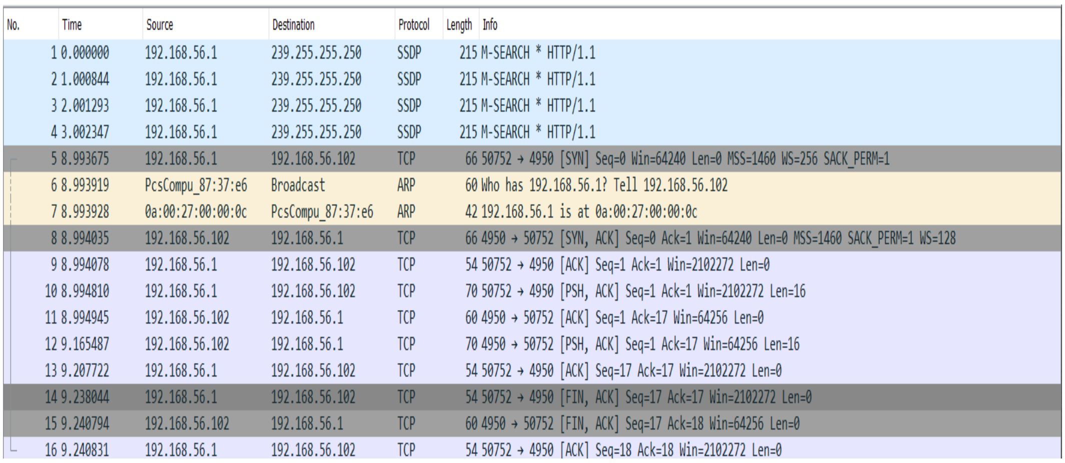
```
> Frame 1: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface \Device\NPF_{5E94552B-B1A0-4533-A140-39FA850B0AB5}, id 0
> Ethernet II, Src: 0a:00:27:00:00:0c (0a:00:27:00:00:0c), Dst: PcsCompu_87:37:e6 (08:00:27:87:37:e6)
v Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.102
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 44
    Identification: 0xde43 (56899)
  > Flags: 0x0000
    ...0 0000 0000 0000 = Fragment offset: 0
    Time to live: 128
    Protocol: UDP (17)
    Header checksum: 0x6ac5 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.56.1
    Destination: 192.168.56.102
v User Datagram Protocol, Src Port: 62895, Dst Port: 4950
    Source Port: 62895
    Destination Port: 4950
    Length: 24
    Checksum: 0x4962 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
  > [Timestamps]
v Data (16 bytes)
    Data: 416e204563686f204d65737361676521
    [Length: 16]
```

Figure 17: UDP Transfer



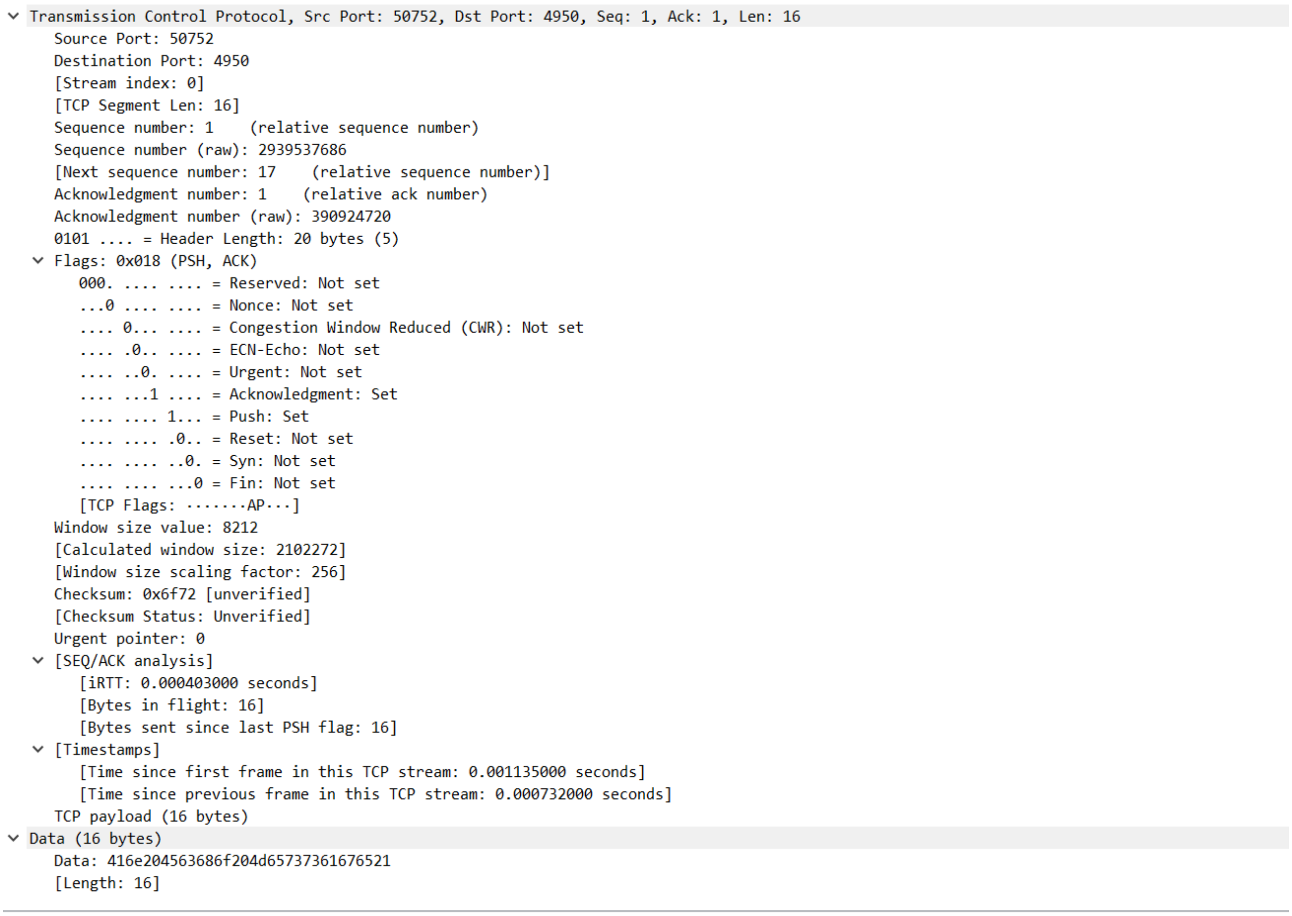
```
> Frame 2: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{5E94552B-B1A0-4533-A140-39FA850B0AB5}, id 0
> Ethernet II, Src: PcsCompu_87:37:e6 (08:00:27:87:37:e6), Dst: 0a:00:27:00:00:0c (0a:00:27:00:00:0c)
v Internet Protocol Version 4, Src: 192.168.56.102, Dst: 192.168.56.1
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 29
    Identification: 0xa2d5 (41685)
  > Flags: 0x4000, Don't fragment
    ...0 0000 0000 0000 = Fragment offset: 0
    Time to live: 64
    Protocol: UDP (17)
    Header checksum: 0xa642 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.56.102
    Destination: 192.168.56.1
v User Datagram Protocol, Src Port: 4950, Dst Port: 62895
    Source Port: 4950
    Destination Port: 62895
    Length: 9
    Checksum: 0xc41d [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
  > [Timestamps]
v Data (1 byte)
    Data: 41
    [Length: 1]
```

Figure 18: UDP Transfer



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.56.1	239.255.255.250	SSDP	215	M-SEARCH * HTTP/1.1
2	1.000844	192.168.56.1	239.255.255.250	SSDP	215	M-SEARCH * HTTP/1.1
3	2.001293	192.168.56.1	239.255.255.250	SSDP	215	M-SEARCH * HTTP/1.1
4	3.002347	192.168.56.1	239.255.255.250	SSDP	215	M-SEARCH * HTTP/1.1
5	8.993675	192.168.56.1	192.168.56.102	TCP	66	50752 → 4950 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
6	8.993919	PcsCompu_87:37:e6	Broadcast	ARP	60	Who has 192.168.56.1? Tell 192.168.56.102
7	8.993928	0a:00:27:00:00:0c	PcsCompu_87:37:e6	ARP	42	192.168.56.1 is at 0a:00:27:00:00:0c
8	8.994035	192.168.56.102	192.168.56.1	TCP	66	4950 → 50752 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
9	8.994078	192.168.56.1	192.168.56.102	TCP	54	50752 → 4950 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
10	8.994810	192.168.56.1	192.168.56.102	TCP	70	50752 → 4950 [PSH, ACK] Seq=1 Ack=1 Win=2102272 Len=16
11	8.994945	192.168.56.102	192.168.56.1	TCP	60	4950 → 50752 [ACK] Seq=1 Ack=17 Win=64256 Len=0
12	9.165487	192.168.56.102	192.168.56.1	TCP	70	4950 → 50752 [PSH, ACK] Seq=1 Ack=17 Win=64256 Len=16
13	9.207722	192.168.56.1	192.168.56.102	TCP	54	50752 → 4950 [ACK] Seq=17 Ack=17 Win=2102272 Len=0
14	9.238044	192.168.56.1	192.168.56.102	TCP	54	50752 → 4950 [FIN, ACK] Seq=17 Ack=17 Win=2102272 Len=0
15	9.240794	192.168.56.102	192.168.56.1	TCP	60	4950 → 50752 [FIN, ACK] Seq=17 Ack=18 Win=64256 Len=0
16	9.240831	192.168.56.1	192.168.56.102	TCP	54	50752 → 4950 [ACK] Seq=18 Ack=18 Win=2102272 Len=0

Figure 19: TCP transfer.



```
v Transmission Control Protocol, Src Port: 50752, Dst Port: 4950, Seq: 1, Ack: 1, Len: 16
    Source Port: 50752
    Destination Port: 4950
    [Stream index: 0]
    [TCP Segment Len: 16]
    Sequence number: 1    (relative sequence number)
    Sequence number (raw): 2939537686
    [Next sequence number: 17    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    Acknowledgment number (raw): 390924720
    0101 .... = Header Length: 20 bytes (5)
  v Flags: 0x018 (PSH, ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 1... = Push: Set
      .... .... .0.. = Reset: Not set
      .... .... ..0. = Syn: Not set
      .... .... ...0 = Fin: Not set
      [TCP Flags: ·······AP···]
    Window size value: 8212
    [Calculated window size: 2102272]
    [Window size scaling factor: 256]
    Checksum: 0x6f72 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  v [SEQ/ACK analysis]
      [iRTT: 0.000403000 seconds]
      [Bytes in flight: 16]
      [Bytes sent since last PSH flag: 16]
  v [Timestamps]
      [Time since first frame in this TCP stream: 0.001135000 seconds]
      [Time since previous frame in this TCP stream: 0.000732000 seconds]
    TCP payload (16 bytes)
v Data (16 bytes)
    Data: 416e204563686f204d65737361676521
    [Length: 16]
```

Figure 20: TCP transfer.

5.1 Discussion

UDP sends whole packages, and TCP sends multiple packets as individual bytes. Figure 16 shows that the hosts send to the server 16 bytes data using UDP protocol. The buffer size of the UDP server is one byte. The server could collect one byte from the data-gram since the package was sent as a whole, which explains why it did send only one byte back to the host.

Figure 17 shows that the host did send 16 bytes of data, and Figure 18 demonstrates that the server echoed back just one byte. That implies the UDP sends the data as a whole package, and the server uses its buffer to collect the data from the package and when the buffer size is less than the size of data that would affect receiving the data as a whole.

When it comes to TCP, the size of the buffer size is one, but it did receive the data from the host as multiple packets. Figure 19 shows the host communicating with server to send data, and the host acknowledges the request by setting the value to 1.

Figure 20 illustrates that the flag is set to push and there is 16 bytes in flight from the last push. Moreover, it shows the sequence number and other details as well. The data is streamed over the network as segments and the sequence n Figure 19 and 20, imply that even when the buffer size is smaller than the data that is streamed over the network, the application that is using the TCP protocol is able to collect the data as whole and in order. A smaller buffer size may cause the application to be slow since it has to collect the data that is being streamed.