# Homework 1

## Client

### Implementation reasoning

Aside from the Program class, which acts as the start-up object, the client has 3 classes: **BaseClient, ClientTcp** and **ClientUdp**.

The BaseClient is an abstract class that denotes some methods, along with some fields and properties that are used by both TCP and UDP clients, such as counting the messages and bytes sent, resetting these counters and so on.

Having a base class for the two also facilitates writing the code that allows for the program to support different protocols without much code repetition. A variable is first declared as being of the BaseClient type and then, based on the input parameters, it is initialized as being either ClientTcp or ClientUdp.

### Options

The **client.cs** file can be compiled with **MonoDevelop**, using the command **mcs client.cs** and then the resulting .exe can be run using **mono client.exe**.

Once the program starts running, lines of text will appear in the console with instructions as to what the user should provide for input. For all these input values, some basic verification has been added to check if they are in expected values.

The values that we can input for the client are:

- Address
- Port
- Communication type: stream or stop-and-wait
- Message size - how many bytes should a message have
- Connection type - TCP or UDP

Once it has finished sending all the contents of the input file, the user has the choice to send the info again or to stop. If he chooses to send again, he may also choose to change the size of the messages that are being sent to a different value.

### Statistics

The messages for the messages sent and the bytes sent are incremented right after the **Write/SendTo** operations. The message counter is always increased by 1 and the bytes counter is increased by the number of bytes that were read from the input file and placed into the buffer. To this extent, the values displayed at the end of a session are always the expected ones:

- Number of bytes equal with file size
- Number of messages approximately equal with file size divided by message size

# Server

## Implementation reasoning

Similarly to the client side, we have 3 classes aside from the Program class: **BaseServer, TcpServer** and **UdpServer.** The approach is similar to the client side so I will not waste much time here.

When talking about the options, you will notice that the message size is not present as an input parameter. As a trade-off for the fact that you can change the message size during execution of the client, without having to restart it, I made the buffer for the server to have the maximum possible size, 65535.

## Options

Again, much like the client, once the server is running, lines of text will appear with instructions as to what the user should input to start the server. These values are:

- Address
- Port
- Communication type: stream or stop-and-wait
- Connection type - TCP or UDP

## Statistics

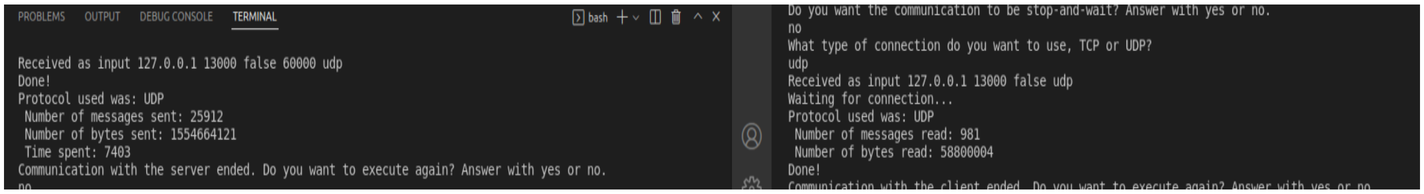
Here is where we see differences from the client. For stop-and-wait communication the output is what you would expect out of a normal execution. The values displayed on the server for messages and bytes read are the same as the ones displayed on the client for those that are sent. Differences however, appear when we get rid of the stop-and-wait verifications.

In the case of TCP, although the number of bytes received is the same as the number of bytes sent, the message numbers differ. Most likely this is due to the client sending multiple messages faster than the server reads them. This is also tied to the fact that I chose to leave the size of the server's buffer to a fixed size.

If the message size is increased from the client's side, we can see that the values start to get closer and closer in value and if the value is the maximum size, then the number of messages match.

As for UDP without stop-and-wait, here we start to see package losses. The greater the message size, the higher the loss.

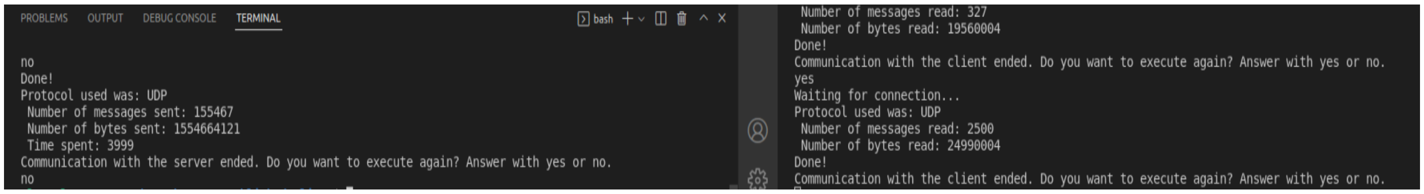
Message size: 60000 bytes, file size: 1.55 GB



```
Received as input 127.0.0.1 13000 false 60000 udp
Done!
Protocol used was: UDP
 Number of messages sent: 25912
 Number of bytes sent: 1554664121
 Time spent: 7403
Communication with the server ended. Do you want to execute again? Answer with yes or no.
no
```

```
Do you want the communication to be stop-and-wait? Answer with yes or no.
no
What type of connection do you want to use, TCP or UDP?
udp
Received as input 127.0.0.1 13000 false udp
Waiting for connection...
Protocol used was: UDP
 Number of messages read: 981
 Number of bytes read: 58800004
Done!
Communication with the client ended. Do you want to execute again? Answer with yes or no.
```

Message size: 10000 bytes, file size: 1.55 GB

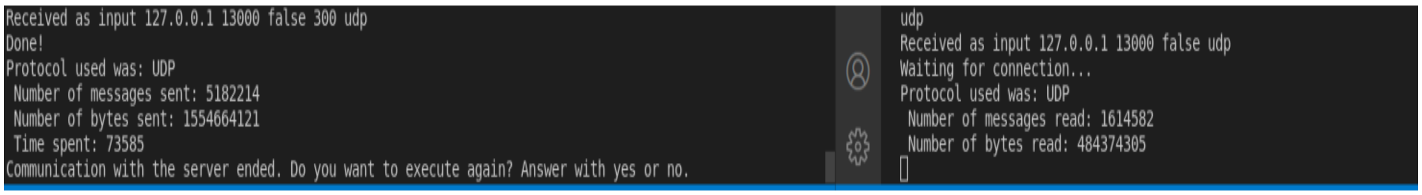


```
no
Done!
Protocol used was: UDP
 Number of messages sent: 155467
 Number of bytes sent: 1554664121
 Time spent: 3999
Communication with the server ended. Do you want to execute again? Answer with yes or no.
no
```

```
 Number of messages read: 327
 Number of bytes read: 19560004
Done!
Communication with the client ended. Do you want to execute again? Answer with yes or no.
yes
Waiting for connection...
Protocol used was: UDP
 Number of messages read: 2500
 Number of bytes read: 24990004
Done!
Communication with the client ended. Do you want to execute again? Answer with yes or no.
```

However, as the package size decreases, we start to get closer to the number of bytes that we sent. Even if we go low enough, below the maximum safe UDP payload of 508 bytes, the loss is still big enough so that our information may not get across.

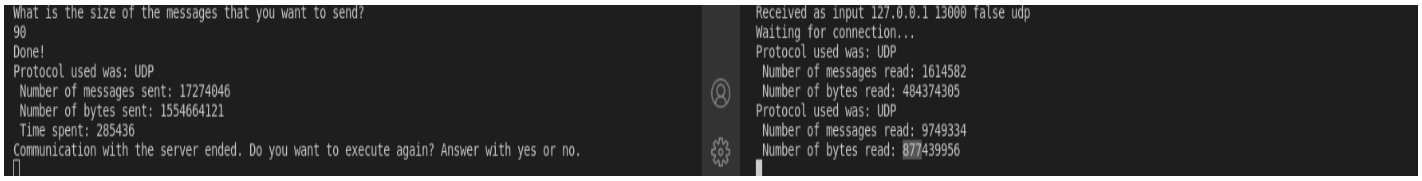
Message size: 300 bytes, file size: 1.55 GB



```
Received as input 127.0.0.1 13000 false 300 udp
Done!
Protocol used was: UDP
 Number of messages sent: 5182214
 Number of bytes sent: 1554664121
 Time spent: 73585
Communication with the server ended. Do you want to execute again? Answer with yes or no.
```

```
udp
Received as input 127.0.0.1 13000 false udp
Waiting for connection...
Protocol used was: UDP
 Number of messages read: 1614582
 Number of bytes read: 484374305
```

Message size: 90 bytes, file size: 1.55 GB



```
What is the size of the messages that you want to send?
90
Done!
Protocol used was: UDP
 Number of messages sent: 17274046
 Number of bytes sent: 1554664121
 Time spent: 285436
Communication with the server ended. Do you want to execute again? Answer with yes or no.
```

```
Received as input 127.0.0.1 13000 false udp
Waiting for connection...
Protocol used was: UDP
 Number of messages read: 1614582
 Number of bytes read: 484374305
Protocol used was: UDP
 Number of messages read: 9749334
 Number of bytes read: 877439956
```

| | | TCP | | UDP | |
|---|---|---|---|---|---|
| | | server | client | | |
| stop-and-wait | messages | 11292 | 11292 | 11292 | 11292 |
| | bytes | 677501115 | 677501115 | 677501115 | 677501115 |
| | time | 3351 ms | | 18536 ms | |
| | block size | 60000 | | 60000 | |
| | file size | 677501115 | | 677501115 | |
| stop-and-wait | messages | 67751 | 67751 | 67751 | 67751 |
| | bytes | 677501115 | 677501115 | 677501115 | 677501115 |
| | time | 4016 ms | | 24048 ms | |
| | block size | 10000 | | 10000 | |
| | file size | 677501115 | | 677501115 | |
| stop-and-wait | messages | 661623 | 661623 | 661623 | 661623 |
| | bytes | 677501115 | 677501115 | 677501115 | 677501115 |
| | time | 18831 ms | | 57859 ms | |
| | block size | 1024 | | 1024 | |
| | file size | 677501115 | | 677501115 | |
| stop-and-wait | messages | 6775012 | 6775012 | 6775012 | 6775012 |
| | bytes | 677501115 | 677501115 | 677501115 | 677501115 |
| | time | 262523 ms | | 260396 ms | |
| | block size | 100 | | 100 | |
| | file size | 677501115 | | 677501115 | |

| | | | | | |
|---|---|---|---|---|---|
| | messages | 11292 | 10339 | 11292 | 395 |
| | bytes | 677501115 | 677501115 | 677501115 | 23640005 |
| | time | 2516 ms | | 336 ms | |
| | block size | 60000 | | 60000 | |
| | file size | 677501115 | | 677501115 | |
| | messages | 67751 | 10339 | 67751 | 7210 |
| | bytes | 677501115 | 677501115 | 677501115 | 72090004 |
| | time | 2503 ms | | 1003 ms | |
| | block size | 10000 | | 10000 | |
| | file size | 677501115 | | 677501115 | |
| | messages | 661623 | 532407 | 661623 | 363249 |
| | bytes | 677501115 | 677501115 | 677501115 | 371965120 |
| | time | 5396 ms | | 5299 ms | |
| | block size | 1024 | | 1024 | |
| | file size | 677501115 | | 677501115 | |
| | messages | 6775012 | 5547533 | 6775012 | 6775012 |
| | bytes | 677501115 | 677501115 | 677501115 | 677501120 |
| | time | 48471 ms | | 53057 ms | |
| | block size | 100 | | 100 | |
| | file size | 677501115 | | 677501115 | |