TCP vs UDP

Gheorghiţă Butnaru

1 Server Application

1.1 Implementation Details

Supported protocols as parameters: TCP, UDP
Message size: 1 to 65535-28 bytes (-28B because of the UDP client)
Two mechanisms implemented: STREAMING, STOP_AND_WAIT

1.2 How to run the application

Step 1: Compile the Java application

```
javac -d bin src/main/java/uaic/fii/pcd/*.java src/main/java/uaic/fii/pcd/
    monitors/*.java src/main/java/uaic/fii/pcd/servers/*.java src/main/java/
    uaic/fii/pcd/utils/*.java
```

Step 2: Run the Java application

```
java -cp bin uaic.fii.pcd.App
```

1.3 Available Options

To open a TCP Server, run the following command:

```
java -cp bin uaic.fii.pcd.App <PORT> TCP <STREAMING/STOP_AND_WAIT>
```

Replace `<PORT>` with the desired port number, and choose either `STREAMING` or `STOP_AND_WAIT` as the mechanism type.

To open a UDP Server, run the following command:

```
java -cp bin uaic.fii.pcd.App <PORT> UDP <STREAMING/STOP_AND_WAIT>
<TIMEOUT_AFTER_FIRST_RECEIVED_MESSAGE_MILLIS>
```

Replace `<PORT>` with the desired port number, choose either `STREAMING` or `STOP_AND_WAIT` as the mechanism type, and `<TIMEOUT_AFTER_FIRST_RECEIVED_MESSAGE_MILLIS>` with the timeout value in milliseconds. The last parameter is used to close the server if it is not receiving another messages from the client in the specified time, and to be able to display some statistics at the console.

2 Client Application

2.1 Implementation Details

Supported protocols as parameters: TCP, UDP
Message size: 1 to 65535-28 bytes (-28B because of the UDP client)
Two mechanisms implemented: STREAMING, STOP_AND_WAIT

2.2 How to run the application

Step 1: Compile the Java application

```
javac -d bin src/main/java/uaic/fii/pcd/*.java src/main/java/uaic/fii/pcd/
    monitors/*.java src/main/java/uaic/fii/pcd/clients/*.java src/main/java/
    uaic/fii/pcd/utils/*.java
```

Step 2: Run the Java application

```
java -cp bin uaic.fii.pcd.App
```

2.3 Available Options

To open a TCP Client, run the following command:

```
java -cp bin uaic.fii.pcd.App <PORT> TCP <STREAMING/STOP_AND_WAIT> <MB_OF_DATA_TO_SEND>
```

Replace `<PORT>` with the desired port number, `<STREAMING/STOP_AND_WAIT>` with the desired mode, and `<MB_OF_DATA_TO_SEND>` with the amount of data to send in megabytes. The data will be generated first, and then will be sent to the server.

To open a UDP Client, run the following command:

```
java -cp bin uaic.fii.pcd.App <PORT> UDP <STREAMING/STOP_AND_WAIT> <MB_OF_DATA_TO_SEND>
<DELAY_AFTER_SENT_MESSAGE_MILLIS>
```

Replace `<PORT>` with the desired port number, `<STREAMING/STOP_AND_WAIT>` with the desired mode, `<MB_OF_DATA_TO_SEND>` with the amount of data to send in megabytes, and `<DELAY_AFTER_SENT_MESSAGE_MILLIS>` with the delay after sending a message in milliseconds.

3 Localhost Environment Analysis

The following statistics were observed using the following dataset:

Dataset Description: The dataset contains statistics related to the transmission and reception of messages between a client and a server using different protocols and mechanisms.

Data Collection: The data was collected by observing the behavior of the client and server during message transmission.

Time (MS)	Sent Messages	Sent Bytes	Sent MB	Read Messages	Read Bytes	Read MB	Protocol	Mechanism
2	17	1048576	1	17	1048576	1	tcp	streaming
8	81	5242880	5	81	5242880	5	tcp	streaming
14	161	10485760	10	161	10485760	10	tcp	streaming
24	321	20971520	20	321	20971520	20	tcp	streaming
43	801	52428800	50	801	52428800	50	tcp	streaming
74	1601	104857600	100	1601	104857600	100	tcp	streaming
153	3202	209715200	200	3202	209715200	200	tcp	streaming
224	4002	262144000	250	4002	262144000	250	tcp	streaming
281	6403	419430400	400	6403	419430400	400	tcp	streaming
343	8004	524288000	500	8004	524288000	500	tcp	streaming
456	9605	629145600	600	9605	629145600	600	tcp	streaming
560	12806	838860800	800	12806	838860800	800	tcp	streaming
712	16392	1073741824	1024	16392	1073741824	1024	tcp	streaming
1068	24587	1610612736	1536	24587	1610612736	1536	tcp	streaming
4	17	1048576	1	17	1048576	1	tcp	stop_and_wait
13	81	5242880	5	81	5242880	5	tcp	stop_and_wait
19	161	10485760	10	161	10485760	10	tcp	stop_and_wait
33	321	10485760	20	321	10485760	20	tcp	stop_and_wait
64	801	52428800	50	801	52428800	50	tcp	stop_and_wait
90	1601	104857600	100	1601	104857600	100	tcp	stop_and_wait
206	3202	209715200	200	3202	209715200	200	tcp	stop_and_wait
295	4002	262144000	250	4002	262144000	250	tcp	stop_and_wait
337	6403	419430400	400	6403	419430400	400	tcp	stop_and_wait
441	8004	524288000	500	8004	524288000	500	tcp	stop_and_wait
492	9605	629145600	600	9605	629145600	600	tcp	stop_and_wait
676	12806	838860800	800	12806	838860800	800	tcp	stop_and_wait
837	16392	1073741824	1024	16392	1073741824	1024	tcp	stop_and_wait
1301	24587	1610612736	1536	24587	1610612736	1536	tcp	stop_and_wait
3	17	1048576	1	3	196521	0	udp	streaming
4	81	5242880	5	20	1310140	1	udp	streaming
13	161	10485760	10	60	3930420	3	udp	streaming
28	321	10485760	20	172	11267204	10	udp	streaming
47	801	52428800	50	450	29435843	28	udp	streaming
79	1601	104857600	100	912	59742384	56	udp	streaming
149	3202	209715200	200	2205	144442935	137	udp	streaming
222	4002	262144000	250	3400	228735430	218	udp	streaming
251	6403	419430400	400	3508	229783542	219	udp	streaming
303	8004	524288000	500	5312	347943156	331	udp	streaming
380	9605	629145600	600	7720	505664905	482	udp	streaming
534	12806	838860800	800	11402	746888972	712	udp	streaming
640	16392	1073741824	1024	14189	929429903	886	udp	streaming
1005	24587	1610612736	1536	22513	1474751218	1406	udp	streaming
12	17	1048576	1	17	1048576	1	udp	stop_and_wait
26	81	5242880	5	81	5242880	5	udp	stop_and_wait
44	161	10485760	10	161	10485760	10	udp	stop_and_wait
90	321	10485760	20	321	10485760	20	udp	stop_and_wait
143	801	52428800	50	801	52428800	50	udp	stop_and_wait
235	1601	104857600	100	1601	104857600	100	udp	stop_and_wait
333	3202	209715200	200	3202	209715200	200	udp	stop_and_wait
373	4002	262144000	250	4002	262144000	250	udp	stop_and_wait
736	6403	419430400	400	6403	419430400	400	udp	stop_and_wait

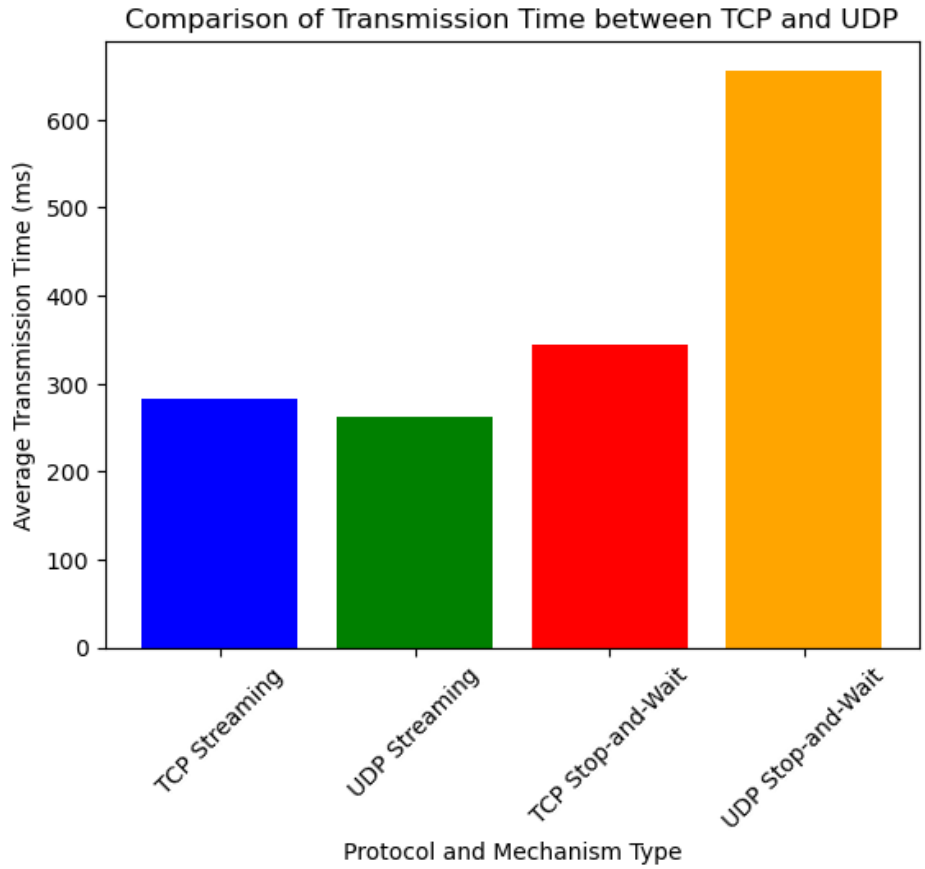
Continued on next page

Table 1 – Continued from previous page

Time (MS)	Sent Messages	Sent Bytes	Sent MB	Read Messages	Read Bytes	Read MB	Protocol	Mechanism
892	8004	524288000	500	8004	524288000	500	udp	stop_and_wait
990	9605	629145600	600	9605	629145600	600	udp	stop_and_wait
1289	12806	838860800	800	12806	838860800	800	udp	stop_and_wait
1624	16392	1073741824	1024	16392	1073741824	1024	udp	stop_and_wait
2401	24587	1610612736	1536	24587	1610612736	1536	udp	stop_and_wait

Table 1: Dataset1

3.1 Comparison of Transmission Time between TCP and UDP using the two mechanism types

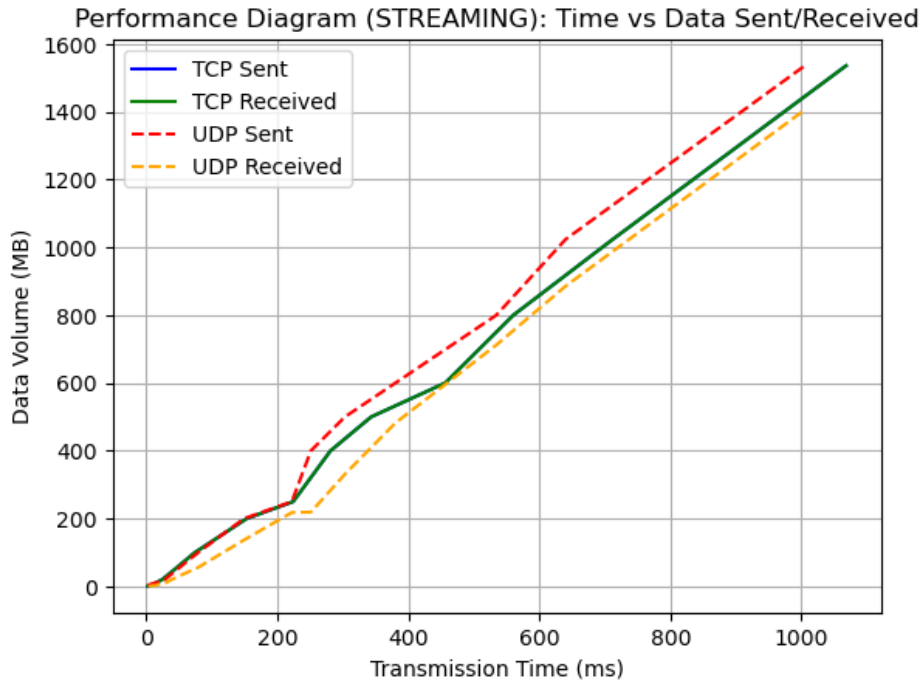


The best performance in terms of time is UDP using STREAMING, followed by TCP using STREAMING.

3.2 Comparison of Data Sent by Client and Data Received by Server

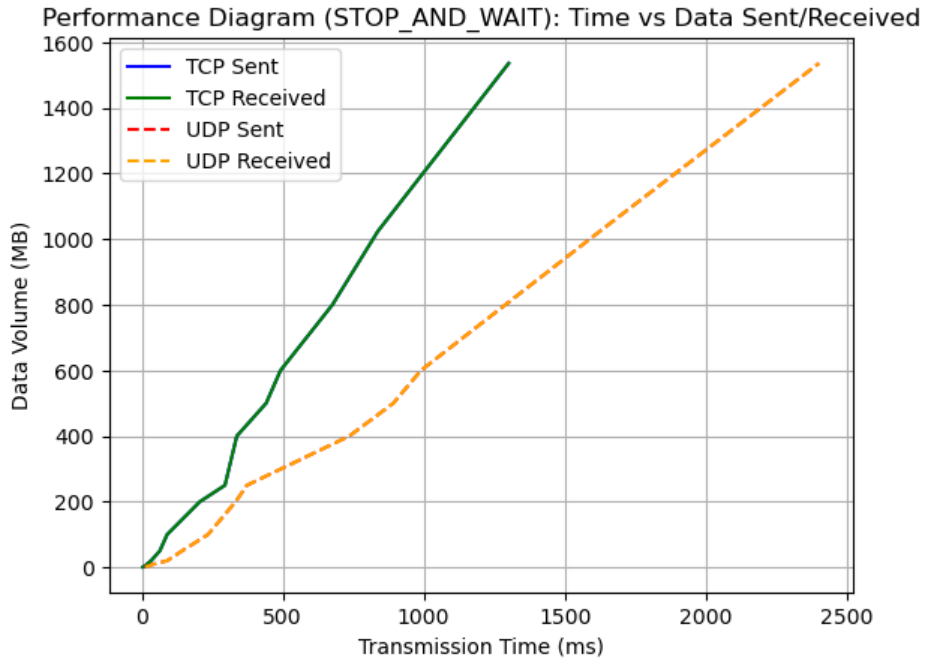
In terms of the data sent by the client and received by the server, UDP using the STREAMING mechanism did not read all the data sent by the client. However, the other combinations (TCP with both STREAMING and STOP_AND_WAIT mechanisms, and UDP with the STOP_AND_WAIT mechanism) did not result in any data loss.

3.3 Performance Diagram (STREAMING): Time vs Data Sent/Received



For the STREAMING mechanism, TCP managed to retrieve all the data transmitted by the client, whereas UDP experienced data loss. Although UDP offered faster transmission times, it suffered from this drawback. In contrast, TCP, while slower, ensured the complete retrieval of client-sent data.

3.4 Performance Diagram (STOP_AND_WAIT): Time vs Data Sent/Received



For the STOP_AND_WAIT mechanism, both TCP and UDP successfully retrieved all the data sent by the client. More than that, it can be observed that TCP is faster than UDP.

4 EC2 Instance Deployment and Analysis

For this statistics, an EC2 instance of type t3.micro was used:

Table 2: Specifications of t3.micro Instance Type

Specification	**Value**
Instance Type	t3.micro
Number of vCPUs	2
Architecture	x86_64
Cores	1
Valid cores	1
Threads per core	2
Valid threads per core	1,2
Sustained clock speed (GHz)	2.5
Memory (GiB)	1
Free-Tier eligible	true
Free Trial available	false
Bare metal	false
Current generation	true

The new dataset based on observations is:

Time (MS)	Sent Messages	Sent Bytes	Sent MB	Read Messages	Read Bytes	Read MB	Protocol	Mechanism
425	17	1048576	1	17	1048576	1	tcp	streaming
600	81	5242880	5	81	5242880	5	tcp	streaming
866	161	10485760	10	161	10485760	10	tcp	streaming
1404	321	20971520	20	321	20971520	20	tcp	streaming
3435	801	52428800	50	801	52428800	50	tcp	streaming
5688	1601	104857600	100	1601	104857600	100	tcp	streaming
10974	3202	209715200	200	3202	209715200	200	tcp	streaming
13138	4002	262144000	250	4002	262144000	250	tcp	streaming
21014	6403	419430400	400	6403	419430400	400	tcp	streaming
23072	8004	524288000	500	8004	524288000	500	tcp	streaming
26423	9605	629145600	600	9605	629145600	600	tcp	streaming
32074	12806	838860800	800	12806	838860800	800	tcp	streaming
50832	16392	1073741824	1024	16392	1073741824	1024	tcp	streaming
80909	24587	1610612736	1536	24587	1610612736	1536	tcp	streaming
348	17	1048576	1	17	1048576	1	tcp	stop_and_wait
556	81	5242880	5	81	5242880	5	tcp	stop_and_wait
897	161	10485760	10	161	10485760	10	tcp	stop_and_wait
1254	321	10485760	20	321	10485760	20	tcp	stop_and_wait
2932	801	52428800	50	801	52428800	50	tcp	stop_and_wait
5917	1601	104857600	100	1601	104857600	100	tcp	stop_and_wait
9686	3202	209715200	200	3202	209715200	200	tcp	stop_and_wait
15285	4002	262144000	250	4002	262144000	250	tcp	stop_and_wait
16809	6403	419430400	400	6403	419430400	400	tcp	stop_and_wait
22438	8004	524288000	500	8004	524288000	500	tcp	stop_and_wait
25976	9605	629145600	600	9605	629145600	600	tcp	stop_and_wait
41778	12806	838860800	800	12806	838860800	800	tcp	stop_and_wait
52172	16392	1073741824	1024	16392	1073741824	1024	tcp	stop_and_wait
69590	24587	1610612736	1536	24587	1610612736	1536	tcp	stop_and_wait
55	17	1048576	1	17	1048576	1	udp	streaming
310	81	5242880	5	81	5242880	5	udp	streaming
630	161	10485760	10	161	10485760	10	udp	streaming
1171	321	10485760	20	321	10485760	20	udp	streaming
2859	801	52428800	50	801	52428800	50	udp	streaming
5122	1601	104857600	100	1601	104857600	100	udp	streaming
8793	3202	209715200	200	3202	209715200	200	udp	streaming
14534	4002	262144000	250	4002	262144000	250	udp	streaming
19559	6403	419430400	400	6403	419430400	400	udp	streaming
22748	8004	524288000	500	7998	523894958	499.63	udp	streaming
26053	9605	629145600	600	9602	628949079	599.81	udp	streaming
36104	12806	838860800	800	12802	838598772	799.75	udp	streaming
45519	16392	1073741824	1024	16388	1073479796	1023.75	udp	streaming
69411	24587	1610612736	1536	24585	1610481722	1535.88	udp	streaming
1098	17	1048576	1	17	1048576	1	udp	stop_and_wait
4957	81	5242880	5	81	5242880	5	udp	stop_and_wait
9866	161	10485760	10	161	10485760	10	udp	stop_and_wait
21091	321	10485760	20	321	10485760	20	udp	stop_and_wait
49625	801	52428800	50	801	52428800	50	udp	stop_and_wait
104625	1601	104857600	100	1601	104857600	100	udp	stop_and_wait
106192	3202	209715200	200	3202	209715200	200	udp	stop_and_wait

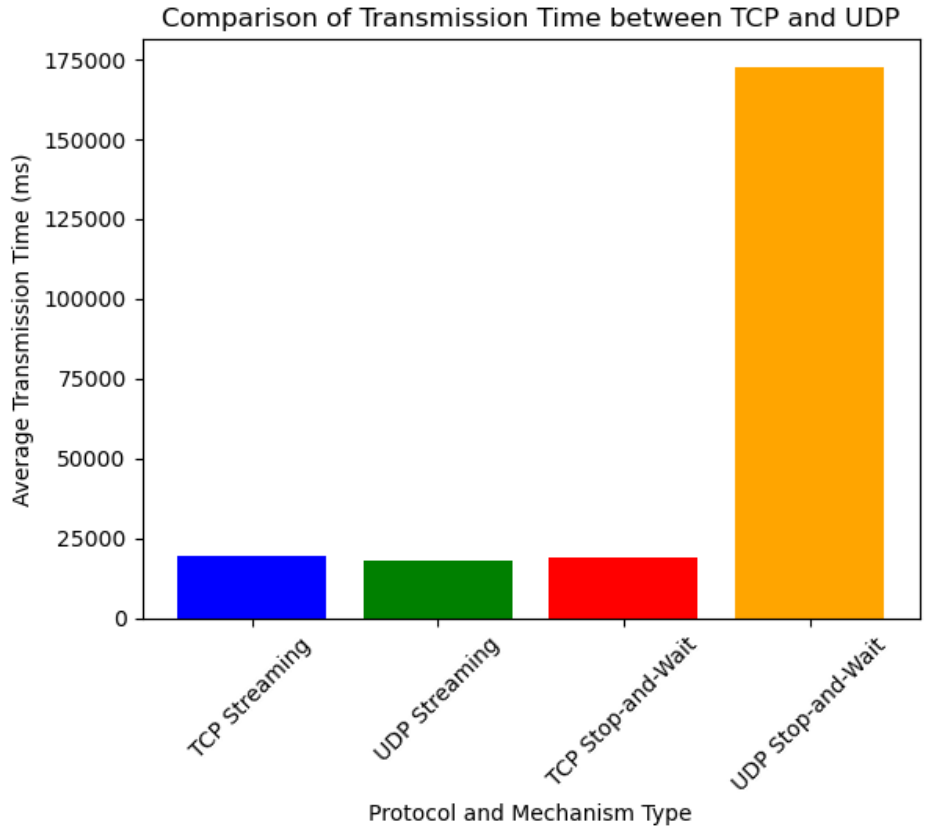
Continued on next page

Table 3 – Continued from previous page

Time (MS)	Sent Messages	Sent Bytes	Sent MB	Read Messages	Read Bytes	Read MB	Protocol	Mechanism
220346	4002	262144000	250	4002	262144000	250	udp	stop_and_wait
222352	6403	419430400	400	6403	419430400	400	udp	stop_and_wait
264467	8004	524288000	500	8004	524288000	500	udp	stop_and_wait
164273	9605	629145600	600	9605	629145600	600	udp	stop_and_wait
178688	12806	838860800	800	12806	838860800	800	udp	stop_and_wait
280456	16392	1073741824	1024	16392	1073741824	1024	udp	stop_and_wait
788275	24587	1610612736	1536	24587	1610612736	1536	udp	stop_and_wait

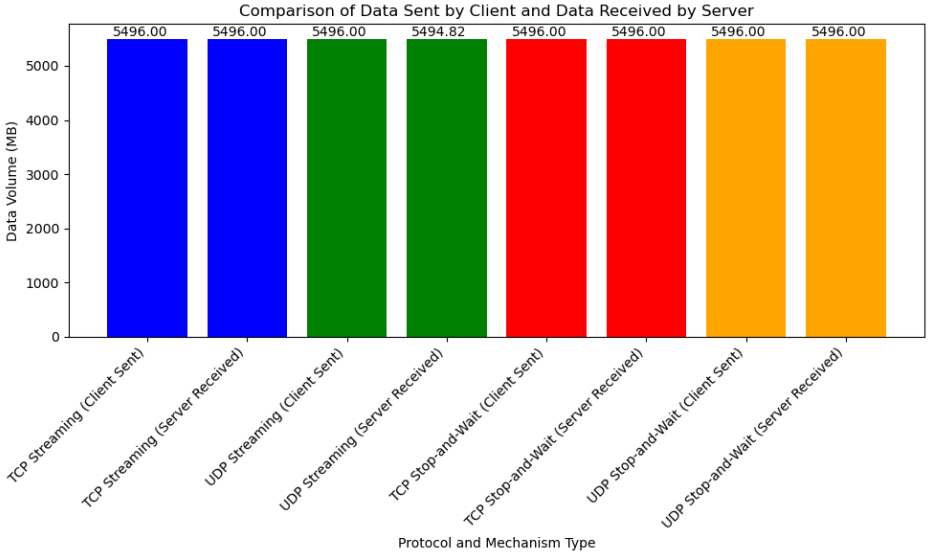
Table 3: Dataset2

4.1 Comparison of Transmission Time between TCP and UDP using the two mechanism types



The best performance in terms of time is UDP using STREAMING, followed by TCP using STREAMING.

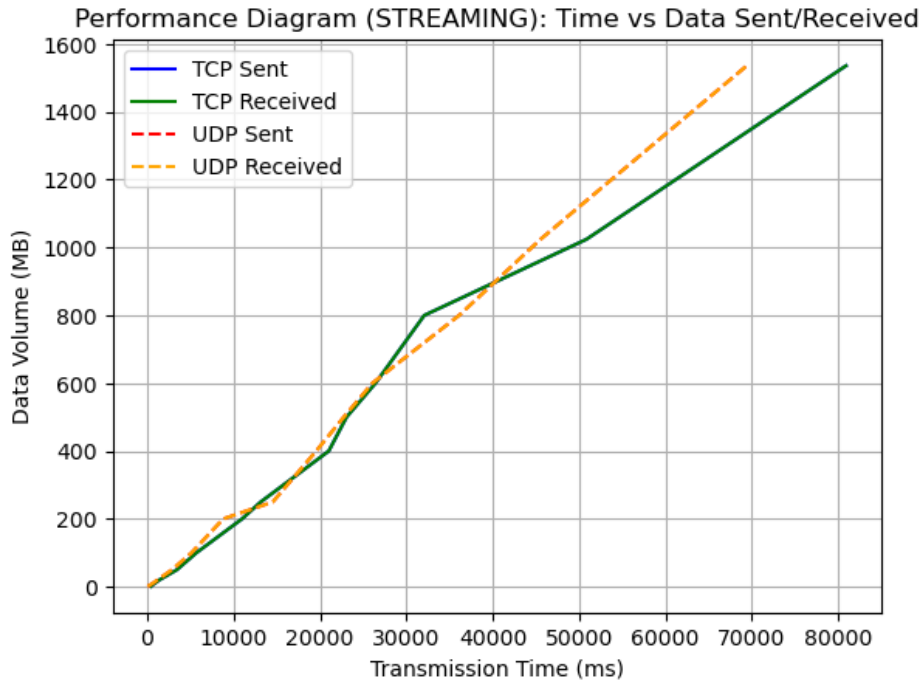
4.2 Comparison of Data Sent by Client and Data Received by Server



In terms of data transmission, UDP, when used with the STREAMING mechanism, experienced a loss of only 1.18 MB out of a total of 5496 MB sent by the client and received by the server.

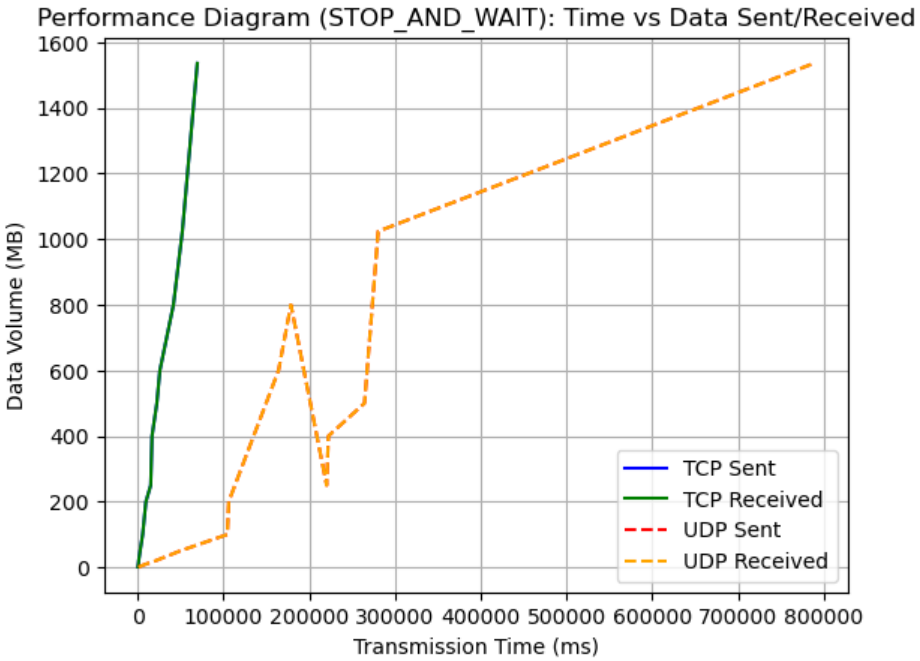
TCP with both STREAMING and STOP_AND_WAIT mechanisms, and UDP with the STOP_AND_WAIT mechanism did not result in any data loss.

4.3 Performance Diagram (STREAMING): Time vs Data Sent/Received



For the STREAMING mechanism, TCP successfully retrieved all transmitted data from the client, while UDP encountered a loss of 1.18 MB out of a total of 5496 MB. However, despite this loss, UDP exhibited slightly faster transmission speeds compared to TCP.

4.4 Performance Diagram (STOP_AND_WAIT): Time vs Data Sent/Received



In the case of the STOP_AND_WAIT mechanism, both TCP and UDP effectively retrieved all data transmitted by the client. However, UDP exhibited significantly slower transmission speeds compared to TCP.