Technical Report

COMP 4981 File Transfer

Table of Contents

[*Executive Summary* 2](#_Toc443233435)

[*Intro* 3](#_Toc443233436)

[*Analysis / Findings* 4](#_Toc443233437)

[*Conclusion* 6](#_Toc443233438)

[*References* 7](#_Toc443233439)

# Executive Summary

# Intro

1. **The objective**

**Protocol Analysis**

The File Transfer application demonstrates the main usage between TCP and UDP suite, which is able to generate datagrams or packet streams and transfer the data between two workstations. In addition, the main usage of the program is to gather and collect transfer data and calculate the analytic statistics.

As for the minimum benchmark that will be used to collect data, we will be using two different set of testing parameters on both protocols: sending 4096 bytes 100 times and 60k bytes 100 times.

Transfer statistics will display the following results on the server, which will be written to a file:

* Starting Time Stamp
* Ending Time Stamp
* Delay
* Packet size
* Packets Expected
* Packets Received
* Total Bytes Received
* Data Rate

**Testing parameters**

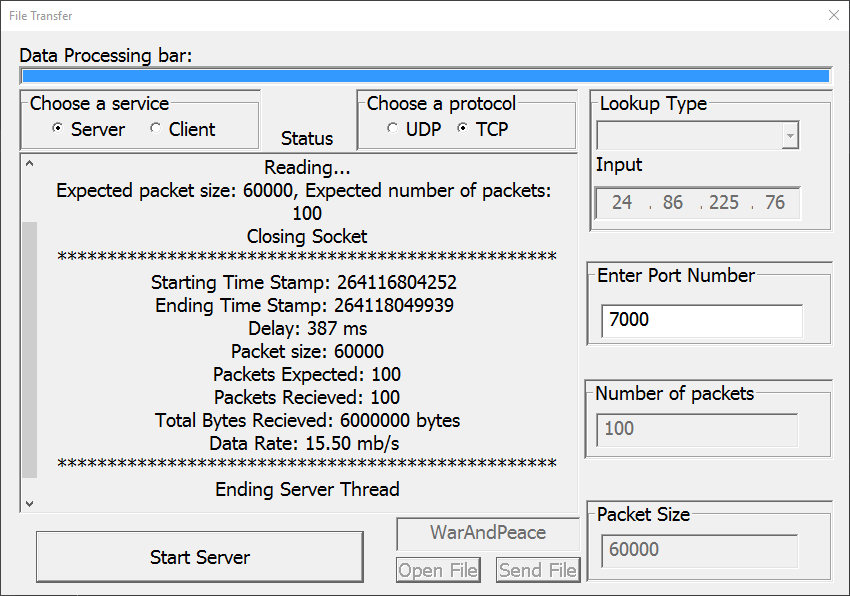
# Analysis / Findings

**LAN**:

**TCP**:

**60k 100 Times**

Application result

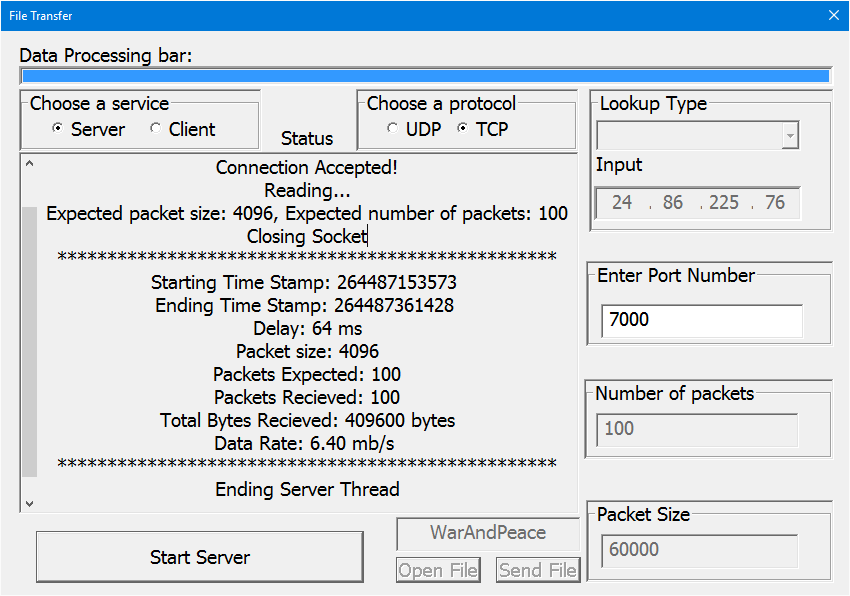


- analysis on speed

- analysis on accuracy

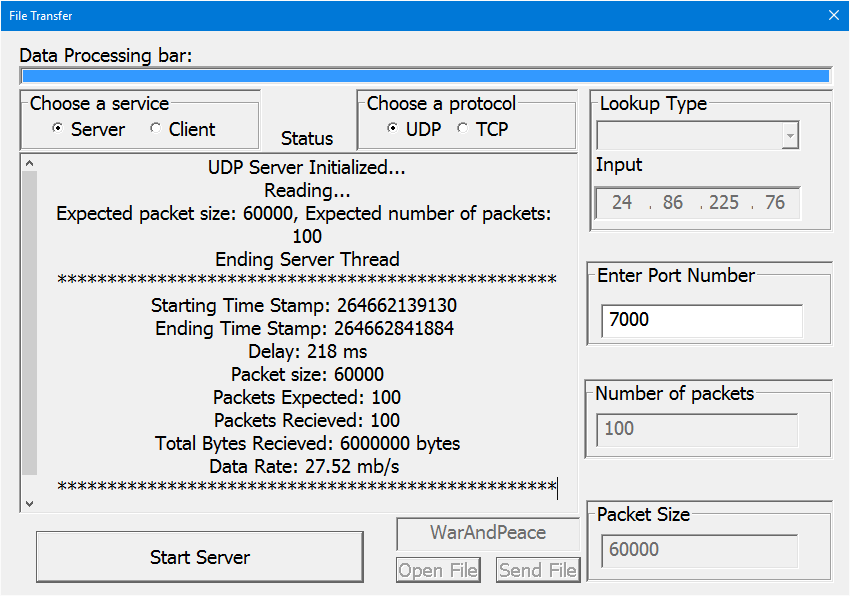
- analysis on reliability

4096 Bytes 100 times

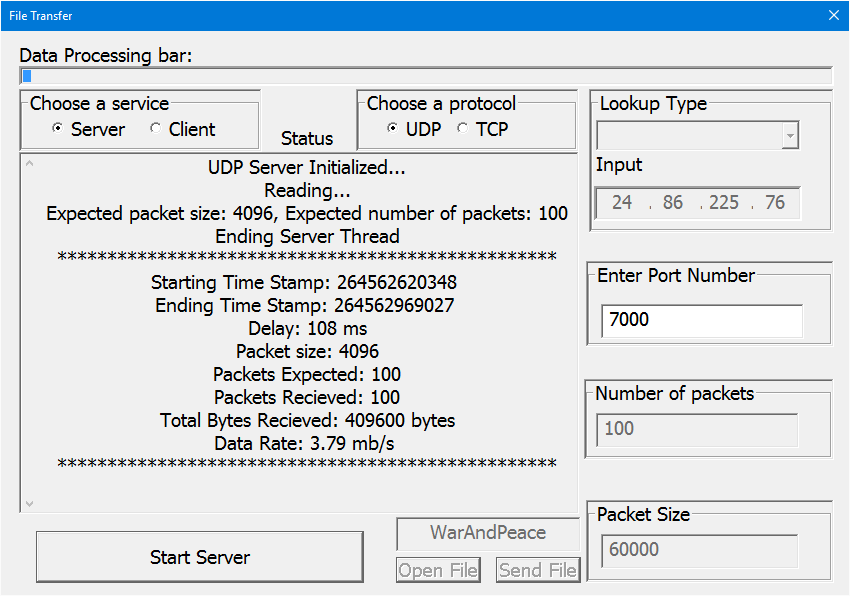


**UDP**:

60k Bytes 100 Times



4096 Bytes 100 Times



- 4096 100 times and 60k 100 times ss

- analysis on speed

- analysis on accuracy

-analysis on Reliability

**Comparison**

**Internal**:

**TCP**:

- 4096k 100 times and 60k 100 times ss

- analysis on speed

- analysis on accuracy

- analysis on reliability

**UDP**

- 4096 100 times and 60k 100 times ss

- analysis on speed

- analysis on accuracy

-analysis on Reliability

**Comparison**

**LAN and Internal comparison**

**UDP and TCP analysis:**

**UDP**:

Trouble: Writing too fast, data is recived and stored on the stack before data processing is done

Resolution: Use circular buffer and thread the data processing

Trouble: Client with async writing too fast on the socket, next packet is sent out to socket before the first one is finished

Resolution: use WSAWaitForMultipleEvent after first one is sent if the return error is IO\_PENDING, memset overlapped and WSACreateEvent()

If internal, data rate is super-fast

**TCP**:

Very reliable, pretty much 100% guarentted packets recieved

Almost same speed as UDP under the same network (sending between 2 computers)

**General**:

Socket programming is really fast if done correctly

WIN32 GUI events takes a huge cut in the speed of socket programming, had to thread it

Importance of data structure (Circular Buffer)

Data processing is VERY slow

# Conclusion

Very fun assignment, learned a lot with the extra time from extension

If done right, this could be a very powerful program

Socket programming is good

windows is bad

completion routine is fast

UDP is super fast when sending and reciein

# References