TECHNICAL REPORT

COMP 4981 File Transfer

Table of Contents

Executive Summary

The purpose of this technical analysis & report is to observe the types of behavior of TCP and UDP protocols under several different network environments, such as LAN (Local Area Network), WLAN (Wireless Local Area Network), and WAN (Wide Area Network). The statistics and data will be calculated and gathered from Wireshark and the File Transfer application, which provides the minimum constraints to execute the experiment.

Aside from comparing and contrasting the performance of the two protocols, such as outlining their respective data rate and packets loss, the paper also highlights some of the major technical difficulties when designing a network architecture, and observations of sockets programming.

Intro

Protocol Analysis

The File Transfer application demonstrates the main usage between TCP and UDP suite, which is able to generate packets and transfer the data between two workstations. In addition, the main usage of the program is to gather and collect transfer data and calculate its statistics in order to gain a better understanding of the two protocols.

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.

Data and statistics will be gathered using our application and Wireshark.

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

- Starting Time Stamp - nanoseconds
- Ending Time Stamp - nanoseconds
- Delay -nanoseconds
- Packet size - bytes
- Packets Expected
- Packets Received
- Total Bytes Received - bytes
- Data Rate – mb / s

Usage

1. Select client or server using radio buttons
2. Select UDP or TCP using radio buttons

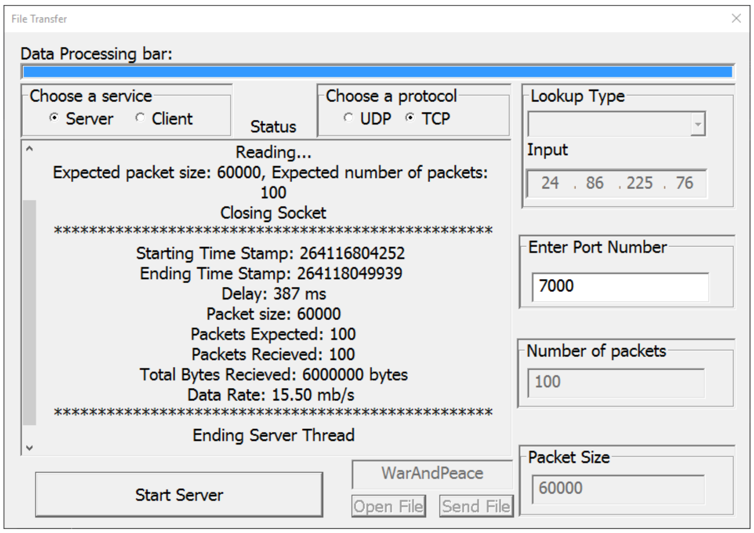
Client	Server
1. Specify IP address of server or host name 2. Specify port number to communicate with 3. Specify packet size and number of times to send 4. Select Send Packets button to send dummy packets, or Send File for sending an opened file	1. Select Connect to start server [Note: Transmission statistics is displayed on the screen and written to file called statistics.txt]

Analysis / Findings

Local Area Network: Gathering Data

TCP: 60k 100 Times

Application result



Graph

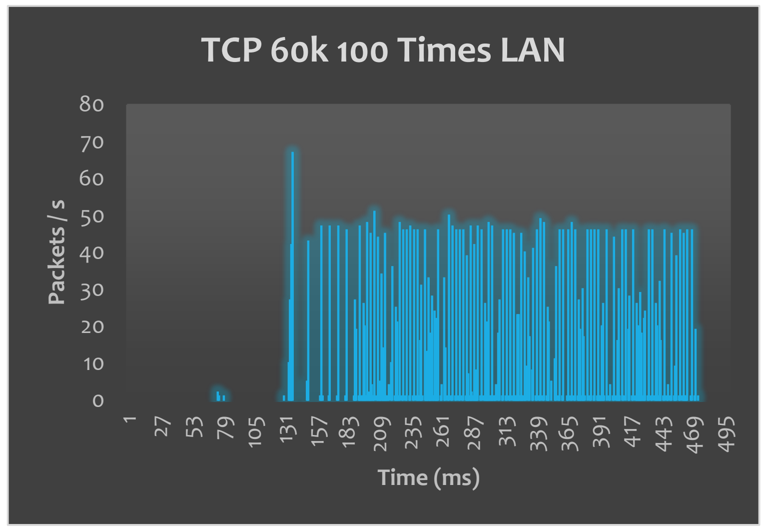


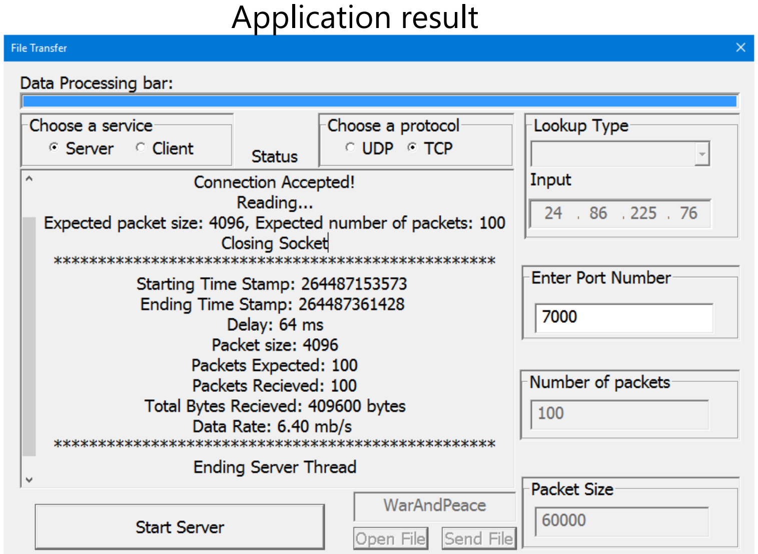
Figure Left: Data gathered from the application

Figure Right: Average Data gathered from Wireshark from 5 transmissions

- Speed: 15.50 Mb/s
- Total delay from first packet to last packet: 387 Milliseconds
- Analysis: Very stable stream of bytes, overall small influxes, 100% reliability

TCP: 4096 Bytes 100 time

Application result



Graph

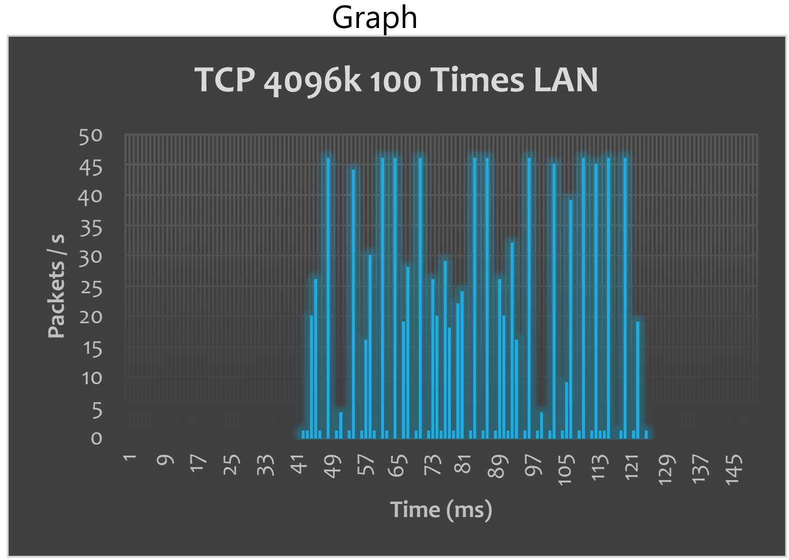


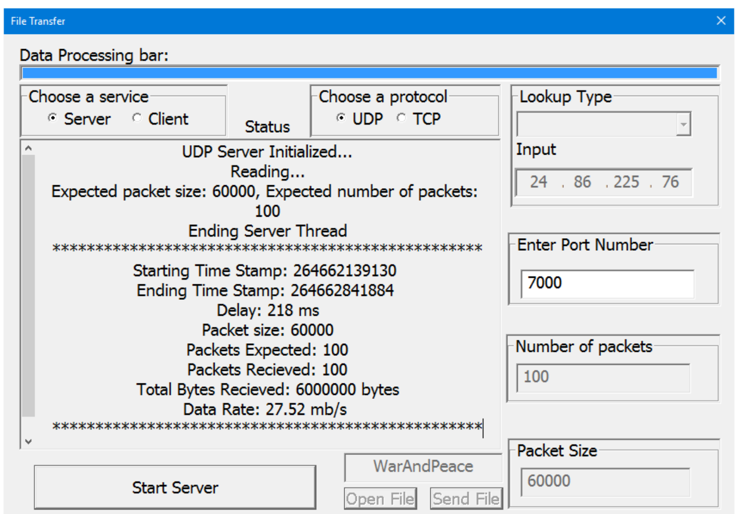
Figure Left: Data gathered from the application

Figure Right: Average Data gathered from Wireshark from 5 transmissions

- Speed: 6.40 Mb/s
- Total delay from first packet to last packet: 64 Milliseconds
- Analysis: Stable stream of bytes, big influxes, 100% reliability

UDP: 60k Bytes 100 Times

Application Result



Graph

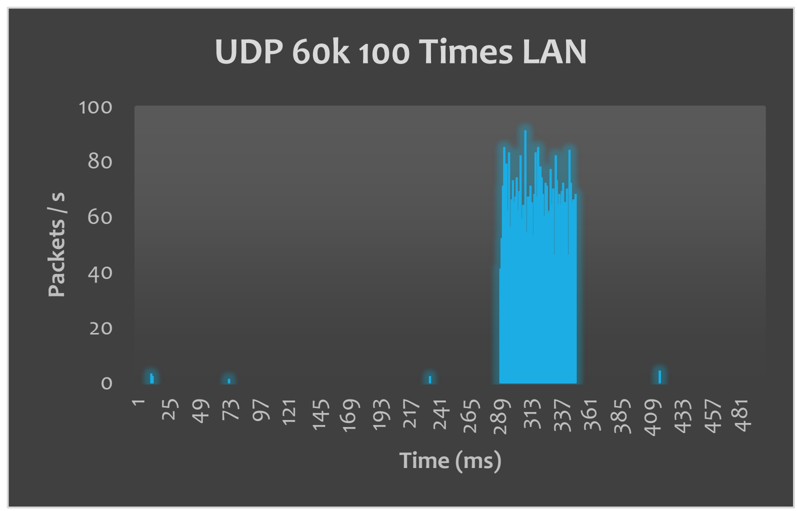


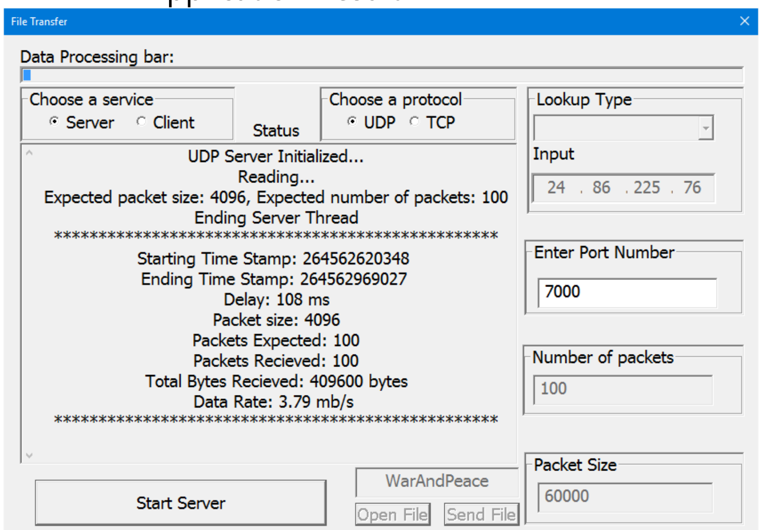
Figure Left: Data gathered from the application

Figure Right: Average Data gathered from Wireshark from 5 transmissions

- Speed: 27.520 Mb/s
- Total delay from first packet to last packet: 64 Milliseconds
- Analysis: About twice as fast as TCP: 60k 100times, around 95% reliability, huge burst of bytes in short amount of time

UDP: 4096 Bytes 100 Times

Application Result



Graph

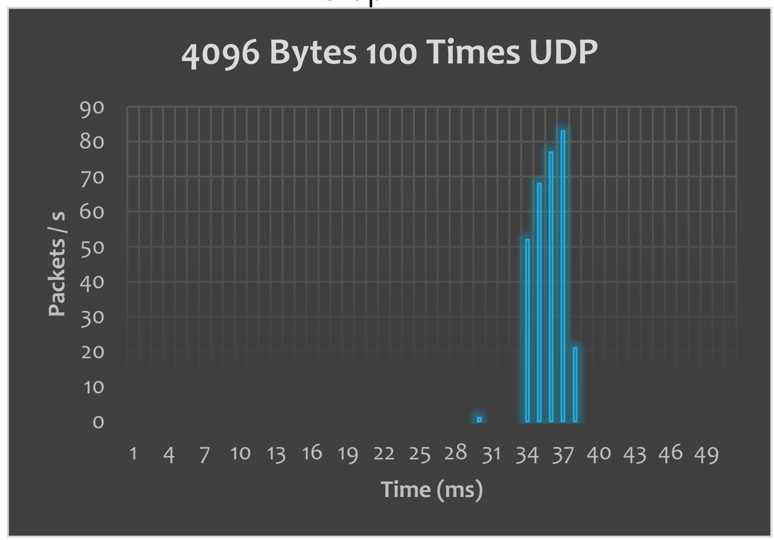


Figure Left: Data gathered from the application

Figure Right: Average Data gathered from Wireshark from 5 transmissions

- Speed: 3.790 Mb/s
- Total delay from first packet to last packet: 64 Milliseconds
- Analysis: Slower than TCP: 4096 100 times, and has a higher influx of bytes in the duration

Local Area Network: Analysis

Given the data provided above, in a reliable Local Area Network (BCIT Secure), TCP and UDP both has the same reliability of around 100% packets received.

If we compare sending 60k bytes and 100 packets between, there is a noticeable visual difference that TCP has a steadier stream of incoming bytes compare to UDP, and UDP has a higher burst in traffic

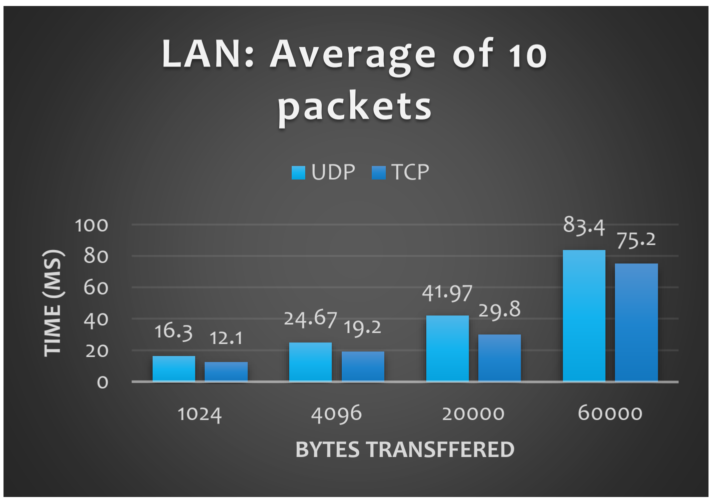


Figure Left: Average data rate of 10 packets

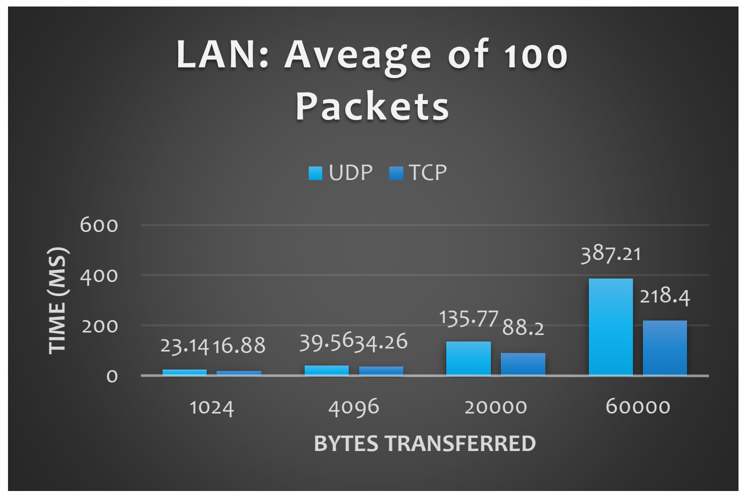


Figure Right: Average data rate of 100 packets

Figure on the left and right has around 99% packets capture rate on both TCP and UDP packets, sending 10 and 100 times in a very short duration. The result shows that UDP is much slower than TCP even when testing environment is under a fairly stable network.

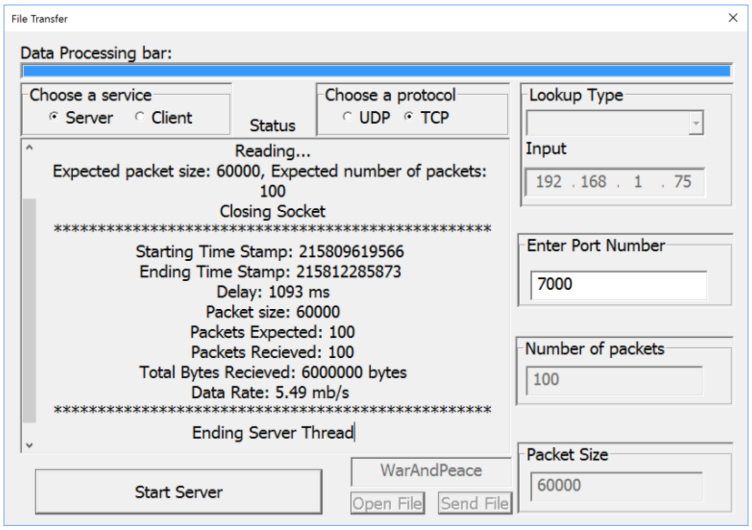
Although the above figures present and contradicts the fact that UDP would have a higher data rate in any circumstances, TCP has a better throughput in the case when its doing lots of small writes relative to the MTU (Maximum Transmission Unity) size.

In our scenario, TCP buffer the data received and fill up the entire network space, which is efficiently using the available bandwidth available. On the other hand, UDP doesn't provide that kind of service, all it does is causing congestion in the network by sending the datagrams to the network immediately.

Wireless Local Area Network: Gathering Data

TCP: 60k 100 Times

Application result



Graph

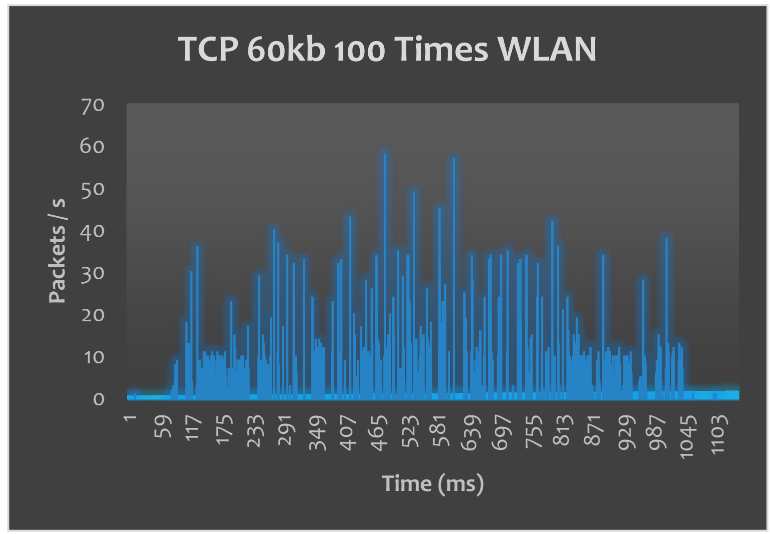


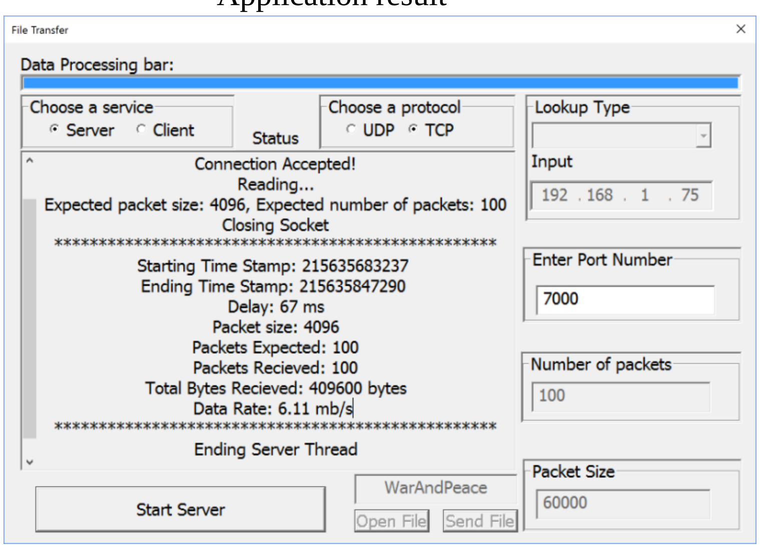
Figure Left: Data gathered from the application

Figure Right: Average Data gathered from Wireshark from 5 transmissions

- Speed: 5.40 Mb/s
- Total delay from first packet to last packet: 1093 Milliseconds
- Analysis: 100% packets received, very reliable, and relatively stable traffic.

TCP: 4096 100 Times

Application result



Graph

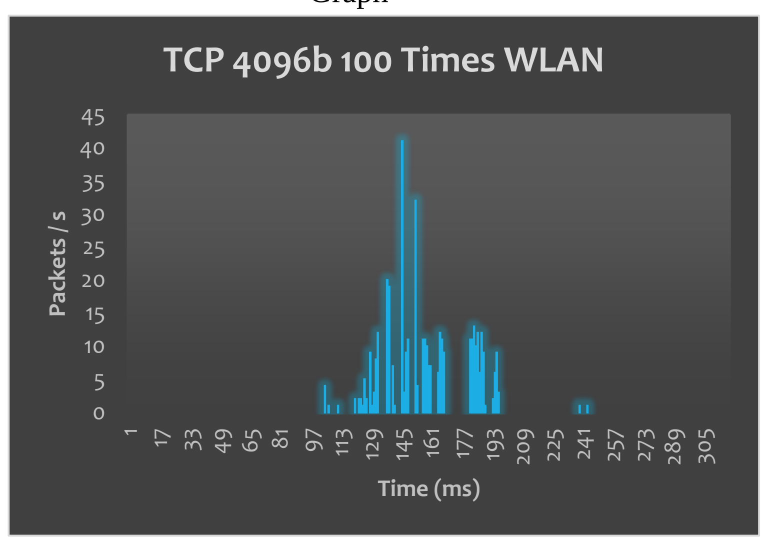


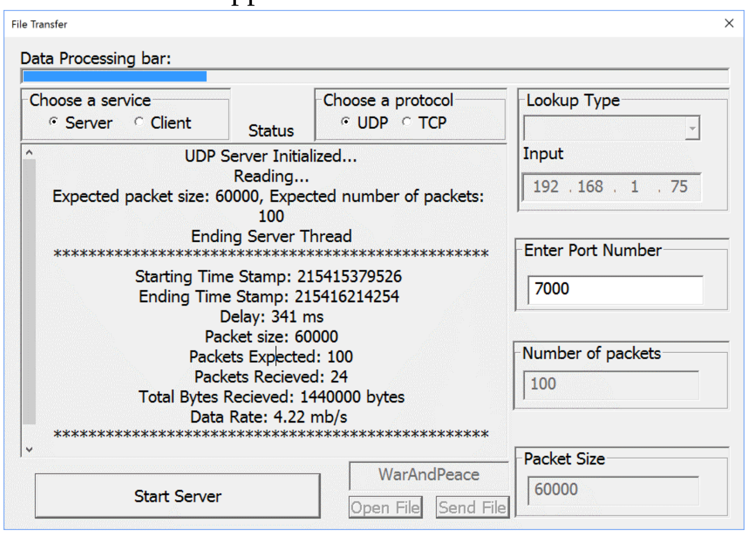
Figure Left: Data gathered from the application

Figure Right: Average Data gathered from Wireshark from 5 transmissions

- Speed: 6.11 Mb/s
- Total delay from first packet to last packet: 67 Milliseconds
- Analysis: 100% packets received, very reliable, and small influx in traffic

UDP: 60k 100 Times

Application result



Graph

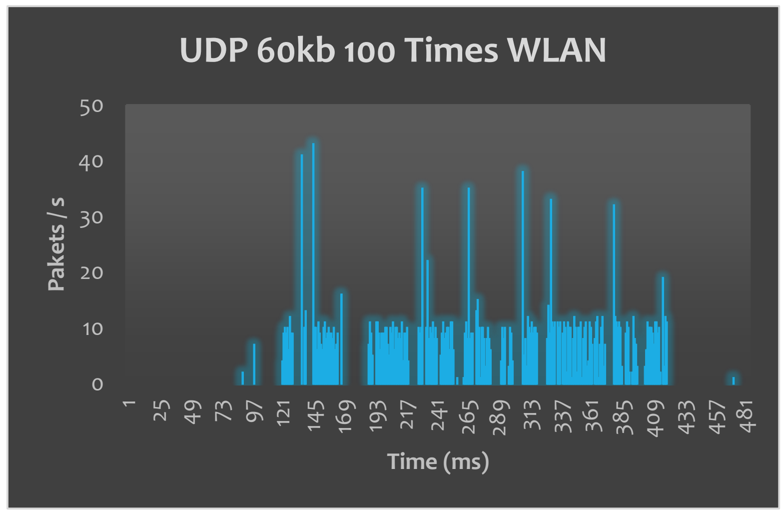


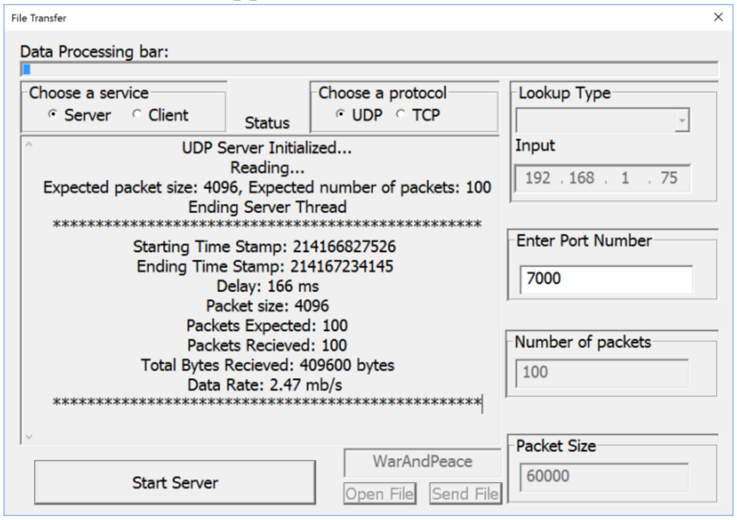
Figure Left: Data gathered from the application

Figure Right: Average Data gathered from Wireshark from 5 transmissions

- Speed: 4.11 Mb/s
- Total delay from first packet to last packet: 341 Milliseconds
- Analysis: Only ~20% accuracy, not reliable, big influx in traffic and empty gaps between bursts (packet loss)

UDP: 4096k 100 Times

Application result



Graph

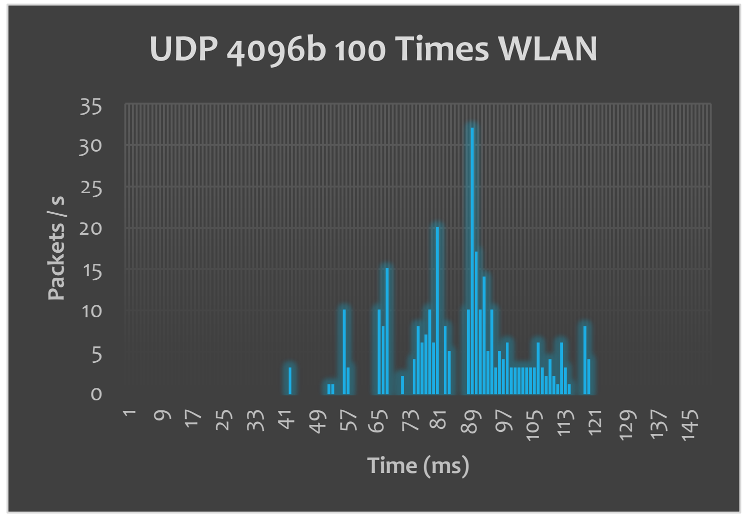


Figure Left: Data gathered from the application

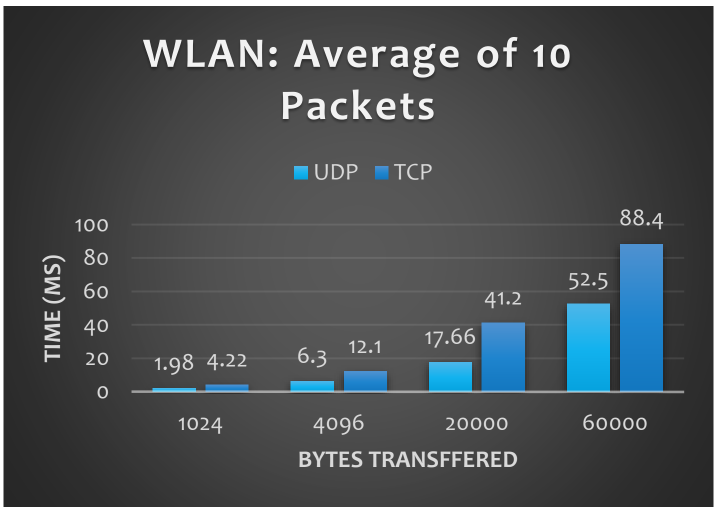
Figure Right: Average Data gathered from Wireshark from 5 transmissions

- Speed: 2.47 Mb/s
- Total delay from first packet to last packet: 166 Milliseconds
- Analysis: High accuracy, big influx in traffic, low data rate.

Wireless Local Area Network: Analysis

Given the data provided above, in a reliable Wireless Local Area Network (home, desktop-laptop), TCP has a huge advantage in packets capture for large data compared to UDP.

TCP's sending / receiving time has increased significantly when we are sending 4096 bytes and 60k packets, whereas UDP steadily gains speed as long as it is not dropping any packets along the way.



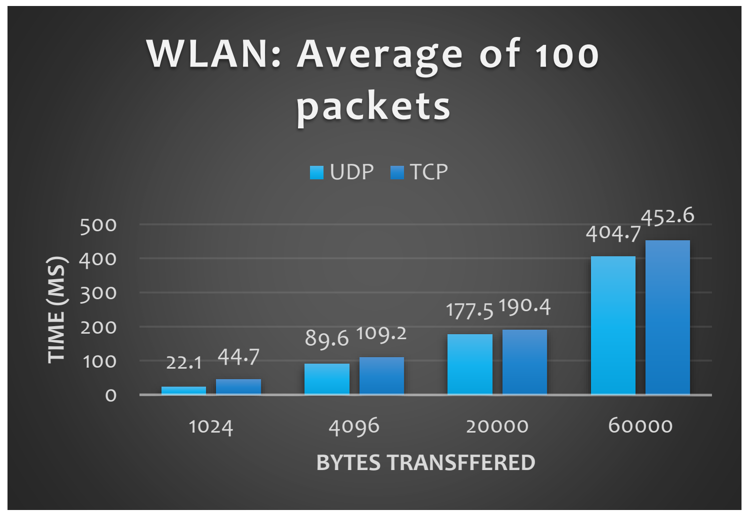


Figure Left: Average data rate of 10 packets

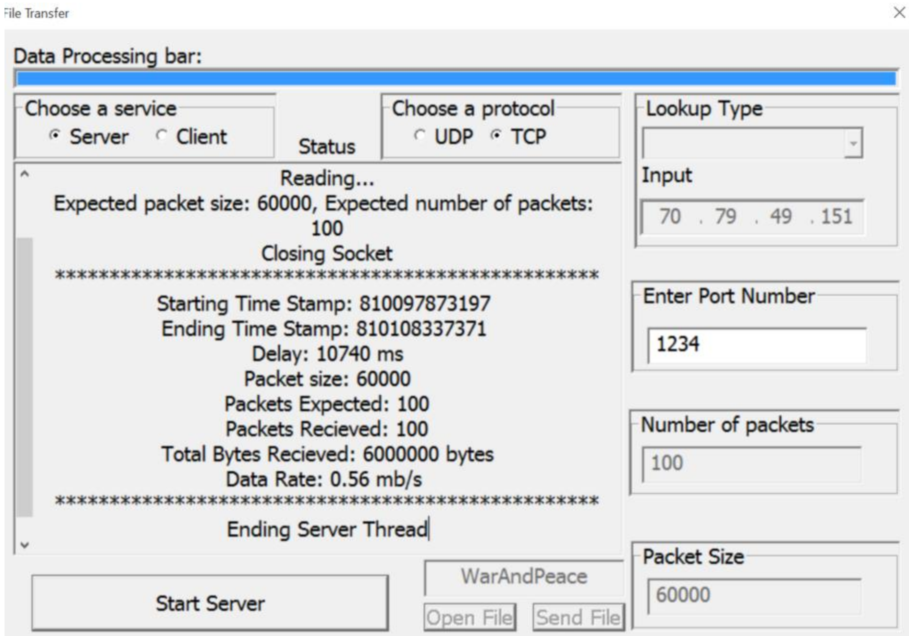
Figure Right: Average data rate of 100 packets

The two figures show the result only of UDP when it is under an ideal network, meaning that it is not dropping any packets. If we look back to the previous data provided above, sending 60k UDP packets 100 times, it's obvious that the data rate will be extremely low because of the amount of packets that are being dropped.

In our scenario, UDP is much faster than TCP, due to the simple nature of its nonexistent acknowledgement packet that permits a continuous stream of packets, only when the network is enable to process all the congested packets sent to the wire. On the other hand, the reason why TCP became a lot slower is because it has to acknowledges a set of packets, which are calculated through the TCP window size and the round-trip time (RTT).

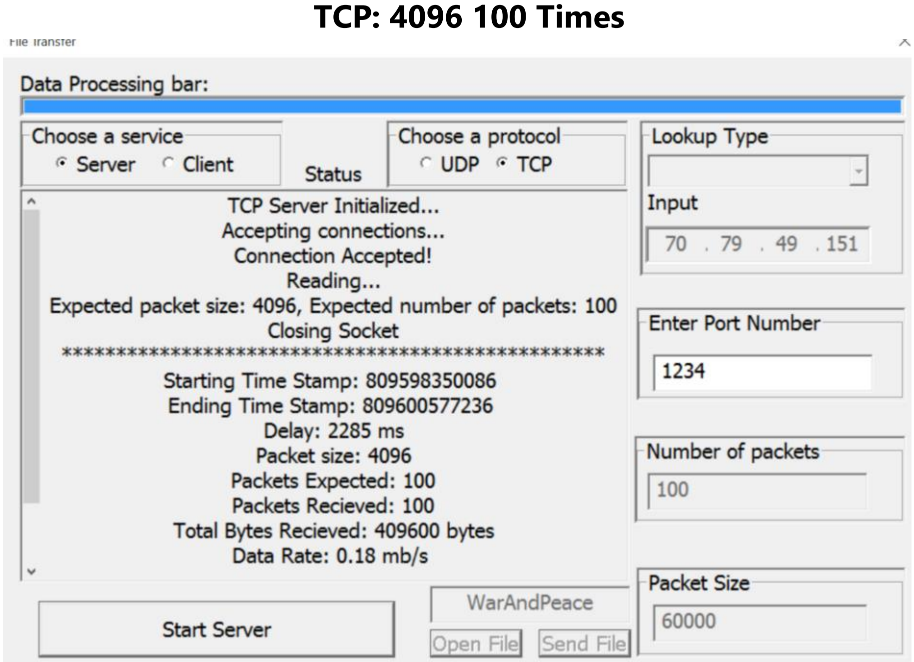
Wide Area Network: Gathering Data

TCP: 60k 100 Times



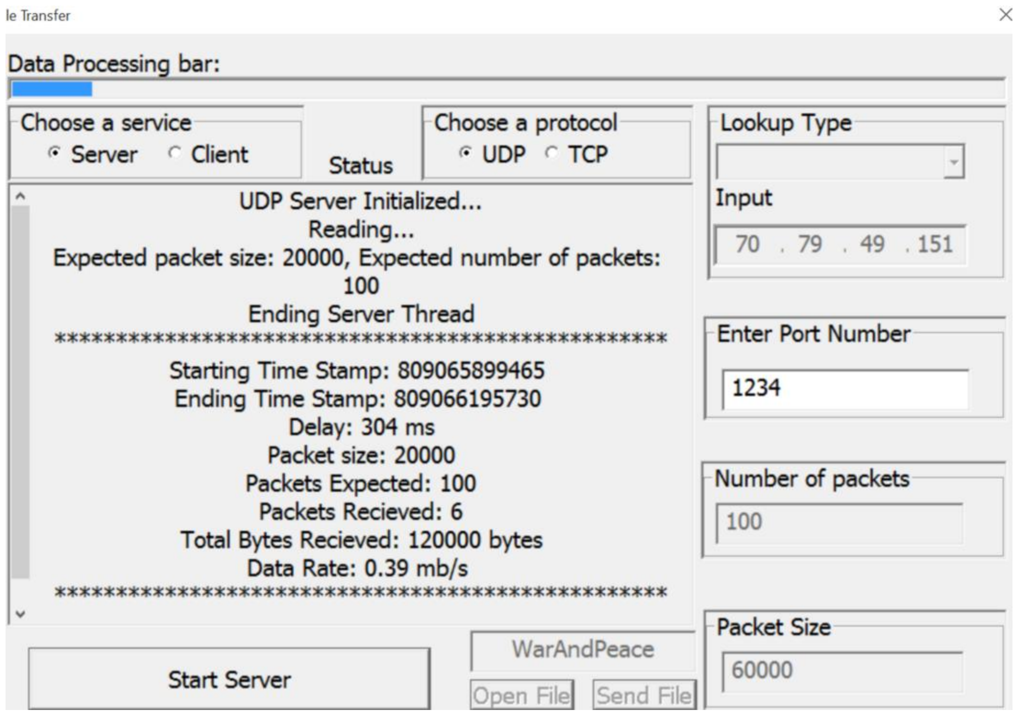
- Speed: 0.56 Mb/s
- Total delay from first packet to last packet: 10.07 Seconds
- Analysis: 100% packets received, very reliable, slow.

TCP: 4096 100 Times



- Speed: 0.18 Mb/s
- Total delay from first packet to last packet: 2.285 Seconds
- Analysis: 100% packets received, very reliable.

UDP: 20k 100 Times

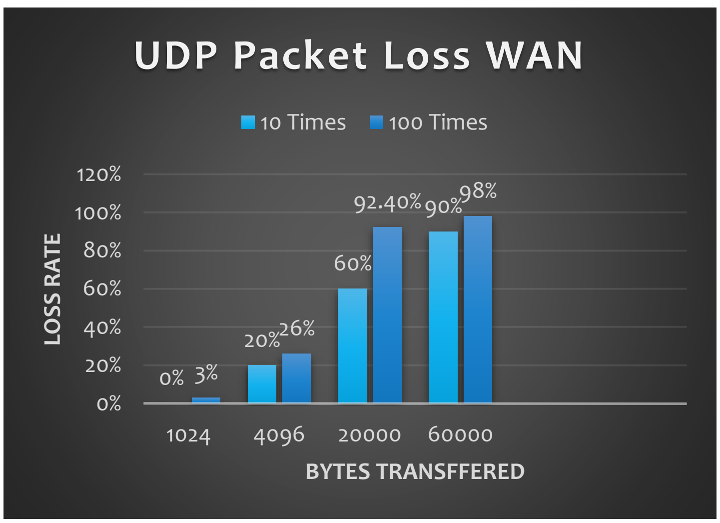


- Speed: 0.39 Mb/s
- Total delay from first packet to last packet: 304 milliseconds
- Analysis: ~95% packets loss, very unreliable, very fast.

WIDE Area Network: Analysis

Given the data provided above, in a Wide Area Network, if we compare the time of TCP sending 60k 100 times on WAN (10740 Ms) to the time of TCP sending 60k 100 times on LAN (364 Ms), it is around 30 times slower. Whereas the same parameters of UDP time on WAN is closely similar to the resulting time on LAN.

Since TCP is all about reliable network, it is guaranteed to receive all the packets sent out, but a matter of time.



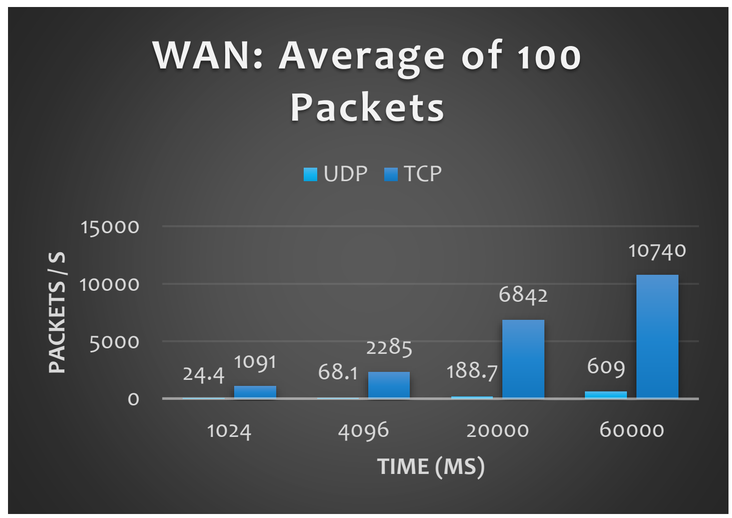


Figure Left: Average packets loss of UDP

Figure Right: Average data rate of 100 packets

Although the speed of UDP on WAN is just as fast as it is on LAN, there is a significant amount of packets that are being dropped, especially packets size that are over 10000 bytes. The explanation is the same as WLAN: where packets are being sent out all at once immediately onto the network, and the protocol has no way of dealing with any control methods, therefore resulting packets being loss on the network, or simply not being processed by the receiver.

Additional Findings:

UDP

- Issue: Client side writing to the socket too fast, data is received and stored on the server stack before data processing is done, therefore popping off packets from the stack, resulting in huge packets loss.
- Resolution: Use circular buffer and thread the data processing

- Issue: Client with async writing too fast on the socket, next packet is sent out to socket before the first one is finished
- Resolution: If the return error of asynchronous send is IO_PENDING, call WSAWaitForMultipleEvents() to ensure that the previous packet is finish processing before sending out another one.

- Very hard to troubleshoot due to random packet losses, had difficult time determining whether if error is due to implementation or simply limited by the network.

- Socket programming using UDP is extremely fast when done using circular buffer sending and receiving on the same workstation. Able to send at maximum speed of ~3 g/s.

General

- WIN32 GUI events takes a huge cut in the speed of socket programming, had to thread multiple GUI events in order to maximize UDP asynchronous send and receive.

Conclusion

With the data collected from experimenting with both UDP and TCP suites, the result confirms the fact that UDP is a very fast, unreliable, connectionless protocol. On the other hand, TCP is generally a bit slower (especially in WLAN), but very connection oriented and reliable in comparison.

In some cases, TCP can be faster than UDP. For example, a wired Local Area Network, where TCP can buffer the data received and fill up the entire network space, which is efficiently using the bandwidth available. Additionally, TCP sends out a much stable stream of packets, which entails that packets are sent and received in an orderly manner. Some of the real life usage of TCP would be a chat program (ensures both sides are recieving the message), E-mails, and file sharing (requires every bit of information for reconstruction).

Although UDP is considered to have a much higher data rate compared to TCP in both Wireless Local Network and Wide Area Network, it is extremely unreliable depending on the size of the packets that are sent. Packets of size over 10000 can be easily loss in the network traffic due to the nature of UDP having zero error handling mechanisms, and can easily lead to network congestions resulting from the large amount of data pumped out from the in a short duration of time. Examples of real life implementation of UDP would be games that doesn't requires every updates to be exact, media streaming (loss frames are fine), and radio stations.

In conclusion, by designing, implementing, and testing this application using sockets, I have gained a much better understanding of both UDP and TCP protocol suites, and how to apply the appropriate one in different fields and under particular types of networks (LAN, WLAN, and WAN). I have also come to the realization that programming in Win32 can really hinder a program that targets for high performance, such as using sockets to send data over the same workstation.

References

TCP and UDP Performance over a Wireless LAN
http://pages.cpsc.ucalgary.ca/~carey/CPSC601.38/archive/2003/papers/tcpudp.pdf

TCP and UDP Performance Tuning
https://www.ualberta.ca/dept/chemeng/AIX-43/share/man/info/C/a_doc_lib/aixbman/prftungd/tcpudpperftun.htm

Characteristics of UDP Packet Loss: Effect of TCP Traffic
https://www.isoc.org/inet97/proceedings/F3/F3_1.HTM

TCP (Transmission Control Protocol)
http://www.linktionary.com/t/tcp.html