Assignment #1

CNTR2115-16W – Industrial Application Development

Denys Politiuk 7058977, William Pring 6919765

Date of Submission: 2016-01-18

**Assignment Scope**

This assignments purpose was to create a Client/Server TCP/UDP application that can send 1000, 2000, 5000, and 10000 blocks to be sent between sender and receiver. We had to benchmark the speeds for these blocks and benchmark the different protocols.

The attended audience are developers that are trying to pick a protocol which will be the most beneficial to their application. It gives them details about the performance of each protocol and the differences in using those protocols with different Operating Systems.

In the end, we found out that usually TCP is a lot slower than UDP. We also found out that TCP on Linux usually a lot faster than TCP on Windows, and UDP on Linux usually a lot slower than UDP on Windows [1].

**Testing Methods**

How we carried our test was that we ran release mode executables with the server and the client knowing the amount of block and the amount of blocks being sent. In the Server we have a timer that keep track of the time spend on receiving the data from the client. We performed this test 3 times and took the average speed between these 3 tests so we have a more accurate results.

**Results**

We were measuring speed of data transferring based on applications we developed. The tests were made on the same platforms using two computers on the same network. The results showed that by increasing size of the packages the speed generally goes up, the results depend on the platform they were tested on, and general view of the reliable TCP and fast UDP is correct. However, we noticed that for Linux operating system, in most situations it is better to use TCP, as it provides better data transfer rate than UDP while still maintaining data integrity.

TCP Windows

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Block Size | Blocks Sent | Total Time | Speed | Missing Data | Out of Order Data |
| 1000 | 1000000 | 49 | 162.27 Mbp/s | 0 | 0 |
| 2000 | 1000000 | 73 | 219.17 Mbp/s | 0 | 0 |
| 5000 | 1000000 | 188 | 212.77 Mbp/s | 0 | 0 |
| 10000 | 1000000 | 276 | 289.86 Mbp/s | 0 | 0 |

TCP LINUX

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Block Size | Blocks Sent | Total Time | Speed | Missing Data | Out of Order Data |
| 1000 | 500000 | 14.97 | 267.20 Mbp/s | 0 | 0 |
| 2000 | 500000 | 19.78 | 404.45 Mbp/s | 0 | 0 |
| 5000 | 500000 | 45.09 | 443.56 Mbp/s | 0 | 0 |
| 10000 | 500000 | 93.45 | 428.03 Mbp/s | 0 | 0 |

UDP LINUX

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Block Size | Blocks Sent | Total Time | Speed | Missing Data | Out of Order Data |
| 1000 | 500000 | 22.65 sec | 175 Mbp/s | 32 packs | 2 packs |
| 2000 | 500000 | 38.5 sec | 206 Mbp/s | 237 packs | 15 packs |
| 5000 | 500000 | 68.15 sec | 292 Mbp/s | 5151 packs | 551 packs |
| 10000 | 500000 | 110 sec | 362 Mbp/s | 9185 packs | 1234 packs |

UDP Windows

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Block Size | Blocks Sent | Total Time | Speed | Missing Data | Out of Order Data |
| 1000 | 1000000 | 19 sec | 418 Mbp/s | 5922 packs | 389 packs |
| 2000 | 1000000 | 180 sec | 89 Mbp/s | 0 packs | 0 packs |
| 5000 | 1000000 | 180 sec | 208 Mbp/s | 50 packs | 5 packs |
| 10000 | 1000000 | 180 sec | 430 Mbp/s | 14 packs | 12 pack |

For UDP Windows we have changed the network adapter setting to only use QoS Packet Scheduler and Internet Protocol Version 4. Otherwise when using default settings the client application would not be able to send packages of 2000, 5000, and 10000 bytes. The speed provided for 1000 bytes was the same in both default settings and new settings of the network adapterS

**Theoretical Speeds**

|  |  |  |
| --- | --- | --- |
| Network | TCP | UDP |
| 100 megabit | 1 MB/s [2] | 12 MB/s [3] |
| 1 gigabit | 125 MB/s [2] | 117 MB/s [3] |

**Comparison of theoretical and measured results**

* There is a speed difference when you are sending block of data between 2 Windows PC’s compared to 2 Linux. The speed difference is that TCP in Linux sends a lot faster than in Windows. However, in UDP Linux sends a lot slower compare to Windows.
* By definition TCP should not lose any bytes no matter of the number of bytes per second that can be sent. This was shown in our test in both Windows and Linux. In UDP it is known for losing packages and not being as reliable as TCP. This was showed in our test that both in Windows and Linux UDP is losing packages. However, our additional tests showed that sending small packages, such as 1000 bytes, small amount of time (less than 10 000 times) can prevent the lose of packages. As a result, it is possible to decrease the chance of losing data by choosing specific values for size of a package and total number of packages.
* In both Windows and Linux TCP application there were no lost packages and all of them were in order. For UDP, the packages were usually not in order and there were lost packages when using both Windows and Linux. Also, in UDP, there is direct relation between number of packages being send and number of lost packages.
* Yes, TCP does have significant overhead in its send and receive compare to UDP in Windows. However, In Linux our data showed that UDP is comparably slower then TCP. This is the result of TCP being more optimized in the Linux kernel compare to UDP.

**Conclusions**

Each protocols has its own strength and weakness and depends on the situation you are using it for. We would use TCP when we want something to be guaranteed to be delivered and we are able to sacrifice speed and handle the major overhead that comes along with TCP. We would use UDP only if we don’t care about reliability and if we prioritize speed and less overhead over reliability. Another point to take in consideration is the platform in which your application will run on. Our results, shows that TCP on Linux is much faster than Windows. As for UDP, on Windows it is a lot faster than on Linux.

**References**

[1] “Why is UDP slower than TCP on Ubuntu Server”. Retrieved January 17, 2016

<http://serverfault.com/questions/432101/why-is-udp-slower-than-tcp-on-ubuntu-server>

[2] Gabriel Torres. “How Gigabit Ethernet Works”. Retrieved January 17, 2016

<http://www.hardwaresecrets.com/how-gigabit-ethernet-works/3/>

[3] “LAN Ethernet Maximum Rates, Generation, Capturing & Monitoring”. Retrieved January 17, 2016

<http://wiki.networksecuritytoolkit.org/nstwiki/index.php/LAN_Ethernet_Maximum_Rates,_Generation,_Capturing_%26_Monitoring>

**Appendix: Complication and usage Instructions**

UDP Client:

To run application run the executable with command line arguments as IP address, size of package (1000, 2000, 5000, or 10000), number of packages to send (between 1 and 999 999 999). Other feedback is available from the application itself

UDP Server:

To run application run the executable with command line arguments as IP address, size of package (1000, 2000, 5000, or 10000), number of packages to send (between 1 and 999 999 999). Other feedback is available from the application itself

TCP Client:

To run application run the executable with command line arguments as IP address, size of package (1000, 2000, 5000, or 10000), number of packages to send. Other feedback is available from the application itself

TCP Server:

To run application run the executable with command line arguments as IP address, size of package (1000, 2000, 5000, or 10000), number of packages to send. Other feedback is available from the application itself

**Evaluation Form**

|  |  |  |  |
| --- | --- | --- | --- |
| Document | | Self Evaluation | Score |
|  | Completeness | 5/ 5 | / 5 |
|  | Format | 4/ 5 | / 5 |
|  | Clarity / Writing | 4/ 5 | / 5 |
|  | References | 3/ 5 | / 5 |
| Document Total | | | / 20 |
| Demonstration | | Self Evaluation | Score |
|  | Readiness | 4/ 5 | / 5 |
| Technical | | Self Evaluation | Score |
|  | Results | 4/ 5 | / 5 |
|  | Theoretical Calculation | 4/ 5 | / 5 |
|  | UI Quality | 5/ 5 | / 5 |
|  | Bug Free Operation | 4/ 5 | / 5 |
|  | SET Coding Standards | 5/ 5 | / 5 |
| Technical Subtotal | | | / 25 |
|  | Completeness Scale Factor | 5/ 5 | / 5 |
| Technical Total (Technical Subtotal \* Completeness / 5) | | | / 25 |
| Reflection | | Self Evaluation | Score |
|  | Self Evaluation Accuracy | 5/ 5 | / 5 |
| Assignment Total | | | / 55 |