

ASSIGNMENT 2: TCP/IP AND UDP SOCKET PROGRAMMING

Kha Phan

7783970

Industrial application development

CNTR2115-19W

Ed Barsalou

January 21, 2019

SECTION 1: ABSTRACT / ASSIGNMENT SCOPE

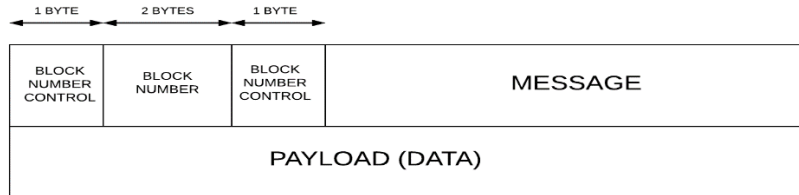
Two to three paragraphs, stating the nature of assignment, who the intended audience is, and other relevant background information, as well as a summary of results.

- Nowadays, we are living in the world of networking, every pieces of data should be transferred between 2 parties which are Client and Server efficiency and fast. From dynamic websites to Android or iOS applications, software many companies has been designing their own protocol to transfer the data. Depending on how good of the protocol, the connection and transformation will be happened proficiency lead to efficiency in their software products. Because of that, knowing the procedure of how to design a socket for communication between the 2 parties as well as understanding how we can customize the protocol to increase the productivity of the software.
- Beside TCP/IP socket communication provide us opportunities to create our own protocol, UDP has come and became more popular nowadays. They both build on top of the internet protocol. Since the TCP is preferred as it guarantees that all the packets of data will be sent to the other ends. The TCP clients firstly initialize the connection while the server is sitting there, listen the incoming packet and established the connection once the server received the first packet of data and they can talk back and forth after that. On the other hand, UDP is a diagram which has higher speed of sending and receiving data because all the packet are just sent to the server without waiting to make sure all data are sent. As a matter of fact, TCP/IP communication protocol is more secure while UDP protocol has a higher speed. Depending on the design of choice, TCP/IP is more preferable in term of websites communication and UDP is mostly used for live broadcasts and online games (Akesson. O).
- There are many protocols are customized and built based on TCP/IP , UDP protocol. To make the custom protocol work, both ends should has a version of the application and talks to the protocol. One end should send the first message and behave as a client, the other will receive the first greeting message and act as server.

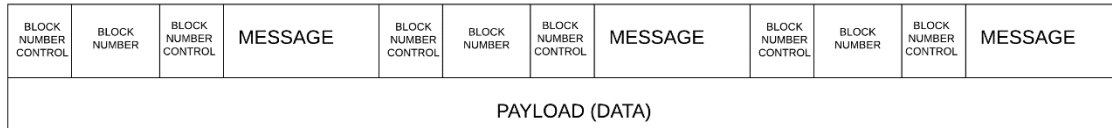
FRAME FORMAT

Provide (in chart format) the layout of the data frames you used, along with a description of each field.

SEND



RCV



In this protocol, every packet is sent with 4 supporting bytes. The first 2 bytes are block number control which are ‘.’ In this protocol. This indicates the block number which is occupied 2 bytes (maximum 65535) which sit between the 2 block number controls. At the end of the packets is the data to be sent. The client will send each block of data in a while loop. It will send until there is no more data to be sent.

In the second data frame, the receiver is ready the maximum data that it can read, those data are merge to each together in a while loop. It will read until no more data in the socket to be read.

TESTING METHOD

Describe how you carried out the tests

- How did you (attempt to) accurately measure speed?
In order to attempt to measure the speed precisely, I measure the time of send and read data in millisecond. Those steps are done in a while loop running forever until there is no more data in the socket to be read. Each time the loop loops, I catch up the return data represent for the byte reading from the socket. In case of UDP, if the recvfrom() function cannot read enough the data, I set up a timer for 5 seconds, if there is no data to be read. I will break the loop and then the total time will be minus 1.
- How did you (attempt to) detect missing blocks?
Firstly, I will merge all the data ready to be sent in client side in a buffer. I use malloc to allocate enough memory to hold all data. In the server side, I use a loop to break all data into blocks because each block are made by a format [.][block number][.][data]. Then I calculate the number of block. Then the number of block missing
- How did you (attempt to) detect out of order blocks?
In this case, I use the block number in each block to compare. For each block split, I will get the block number and assign to an array call FRIST. Then when the next block is split, I assign the block number to an array called SECOND. So if the first+1 is not equal the second, I increase the number of disorder.

RESULTS

Include your result tables here. Be sure to introduce the section with a sentence describing what you observed, especially noting any strange or unusual results.

Block Size (Bytes)	Number of Blocks Sent	Total Time	Speed (Bytes / Second)	Missing Data (Blocks)	Out of Order Data (Blocks)
1000	1000	226 milliseconds	4425000	0	0
2000	1000	366 milliseconds	5465000	0	0
5000	800	805 milliseconds	4968750	0	0
10000	500	759 milliseconds	6587500	0	0

Testing TCP/IP in Windows

Block Size (Bytes)	Number of Blocks Sent	Total Time	Speed (Bytes / Second)	Missing Data (Blocks)	Out of Order Data (Blocks)
1000	1000	190 milliseconds	5263750	0	0
2000	1000	357 milliseconds	5602500	0	0
5000	800	664 milliseconds	7530000	0	0
10000	500	617 milliseconds	7698750	25	6

Testing UDP in Windows

Block Size (Bytes)	Number of Blocks Sent	Total Time	Speed (Bytes / Second)	Missing Data (Blocks)	Out of Order Data (Blocks)
1000	1000	136.35 milliseconds	7333750	0	0
2000	1000	670.33 milliseconds	2983750	0	0

5000	800	499 milliseconds	8013750	0	0
10000	500	601 milliseconds	8307500	0	0

Testing TCP/IP in Linux

Block Size (Bytes)	Number of Blocks Sent	Total Time	Speed (Bytes / Second)	Missing Data (Blocks)	Out of Order Data (Blocks)
1000	1000	109.92 milliseconds	6377500	299	6
2000	1000	159.99 milliseconds	3510000	720	8
5000	800	255.45 milliseconds	4286250	581	16
10000	500	192.48 milliseconds	3273750	437	10

Testing UDP in Linux

The first 3 table really represents the data receiving speed. But the fourth table data is not recallable because the missing is big and disorder is big. So it take times for waiting and break the loop. So the time is not reliable.

THEORETICAL SPEEDS

State how you determined / calculated the maximum theoretical speeds you expected to achieve.

COMPARISON OF THEORETICAL AND MEASURED RESULTS

Examine how your results for each combination varied with respect to each other, as well as in comparison to theoretical results. Make sure to indicate:

- Is there a speed difference between sending blocks of data between 2 Windows PCs versus 2 Linux PCs?
There is a big speed difference sending between 2 PCs in both OS. On Window, the UDP communication is faster compared to TCP but TCP guarantee the number of block sending from client (UDP has missing orders in the fourth times). On Linux, the missing is more real as in UDP.
- Number of bytes per second that can be reliably sent
It is 4425000 bytes/second
- Were all blocks obtained in the order that they were sent

In term of TCP/IP, all block are sent. But UDP, when we send a huge amount of data, block is missed and the order is mixed

- Does TCP have any appreciable overhead in send/receive vs UDP

Yes, the server need to wait for the connection from client. And that create an overhead in sending and receiving.

CONCLUSION

Given your observations, when would you use TCP, and when would you use UDP? Consider your references as well as your results.

- It is depending on the purpose using the protocol. If I am dealing an average amount of data, and if the data is lost, it can or cannot be resent. This sometime is not a big deal because the packet loss is not a big amount (in sending 500 block of 10000 bytes using UDP in windows in the table above, we only lost 25 blocks). But the advantages of it is huge because its speed help some app run efficiency. For example if I want to create an app having a function that people can talk, call or video streaming, data should be back and forth very fast.
- However, TCP is more popular nowadays because it helps to send and received huge amount of data without losing. In the article “TCP vs UDP: Understanding the Difference”, the author said the TCP is more common as it is used widely in web development because it make sure all data of the response page is received and sent.

WORK ESTIMATE AND BREAKDOWN

For each team member, provide an estimate of the hours worked and tasks completed in the table format shown below.

Table 1 - Team Member Work Breakdown

Team Member	Hours	Task

REFERENCES

Akesson.O. 2019. What is the difference between TCP and UDP? *Holm Security*. Retrieved from:
<https://support.holmsecurty.com/hc/en-us/articles/212963869-What-is-the-difference-between-TCP-and-UDP->

Hochstadt Ariel.2018. TCP vs UDP: Understanding the Difference. Retrieved from:
<https://www.vpnmentor.com/blog/tcp-vs-udp/>

APPENDIX: COMPILATION AND USAGE INSTRUCTIONS

EVALUATION

Please include the evaluation form as a separate last page in your report.

Document		Self Evaluation	Score
	Completeness	4/ 5	/ 5
	Format	5/ 5	/ 5
	Clarity / Writing	5/ 5	/ 5
	References	5/ 5	/ 5
Document Total			/ 20
Demonstration		Self Evaluation	Score
	Readiness	5/ 5	/ 5
Technical		Self Evaluation	Score
	Results	4/ 5	/ 5
	Theoretical Calculation	0/ 5	/ 5
	UI / Scripting	5/ 5	/ 5
	Bug Free Operation	5/ 5	/ 5
	SET Coding Standards	5/ 5	/ 5
Technical Subtotal			/ 25
	Completeness Scale Factor	4/ 5	/ 5
Technical Total (Technical Subtotal * Completeness / 5)			/ 25
Reflection		Self Evaluation	Score
	Self Evaluation Accuracy	4/ 5	/ 5
Assignment Total			/ 55