

What did we do

In the project1, we build a communication link between our devices through acoustic signals. The key part of project1 is to choose how to modulate and demodulate your signal. From very beginning, we find that it is hard to correctly detect out frame header which leads to the data demodulated from signal is almost all wrong. To solve this, we design our unique frame header and pick a proper magic number to detect the frame header. It also takes our much effort to think how to implement our system in a further view. Cause how you implement project1 really has influence on the following projects.

In addition, another challenge we meet is how to balance the speed and correctness. We first design a really big frame so that all data can be translate by just one frame but soon we find that this design will crush all data if just one bit go wrong.

From project2, the biggest difference is moving from wireless link to wired link. The advantage of this is to reduce the heterogeneity and complexity in acoustic hardware. Although CSMA Protocol is discussed in the lecture, it is really difficult to implement the CSMA and ACK state machine. But implement this state machine by hands can help us deeply understand the idea of "listen before talk".

Compared with project1 and project2, project3 and project4 are focused on socket programming. For those who has never had contact with socket program, it may take some time to get familiar with it. And the experience of using Wireshark to capture Ping packets not only give us a interesting learning process but also make us understand the Ping more preciously. And File Transfer Protocol in project4 give us a different view from lecture in class about transfer between clients and servers. The main challenge we meet in project3 and project4 is how to use specific API to do what we really want to do; it takes our much time to read the document about API and read code from sample code about API.

Generally speaking, the project in CS120 is the most reasonable and most rewardable project from the courses I have took. Firstly, the difficulty is proper, many difficult parts are put in optional part. It is also the worthiest part to praise. It enables us to understand the basic part of project and for those who want to get deeper challenge, they can finish the optional part. Secondly, the knowledge which are required to finish the project are almost mentioned in the lecture or class.

Technical detail about error detection

From project2, our first significant improvement is to bring in the CRC error detection. Although we have learned a brief error detection in the lecture, it is a little bit difficult for us to implement crc32 by us, after search in the wiki, we found that there exist an official CRC lookup table and after read the official document and combine with slides, we implement out crc32 error detection, which helps us a lot in project2 and project3. Still, project of CS120 not only help us understand what we have learned in class but also give us opportunity to explore the knowledge that is just mentioned in the class.

Some suggestions

First thing first, for the physical lawyer part, the physical part (hardware) is really important! For project1, it takes us a lot of time to debug about the hardware, but find it impossible to deal with it in software but change our computer disappointedly. It will take those people who does not realize that it is the hardware that obstruct them from the correct answer much time to debug.

From our perspective, if you choose Java to implement project1 and MacOS and Linux are highly NOT recommended to use in this project. Also, in speed perspective, Python is not recommended in project1.

Another suggestion about project is how to grade, we think it is better to set a upper bound of the grade you can get from the whole project. For example, for now you can get 45 points from compulsory part and 30 points from bonus part. But when some students are too hard-working and do all bonus so other students may be panic and the purpose of bonus may get lost. So maybe we can set the whole project have 75 points composed by 45+30 but you can only get maybe 60 points from the project so when a student gets 30+30 and a student gets 45 + 20, they are all brilliant and they all get full score 60 points. This may stop students from malicious competition and let bonus part plays true role in this project.

Every End is a New Beginning

Actually, project in CS120 is more engineering-oriented instead of code-oriented, which means it is our first time to build such a big project to implement what we have learned in the class by hands from empty. And one thing I learn from CS120 project and never get touch before is trade off. From our pervious coding experience, we barely need to consider trade off between bandwidth and speed. An impressive example is when we choose magic number (the number to decide whether it is frame header). When we set the magic number too big, it can more correctly detect the data frame header but it also has more opportunity to lost the header. But if we choose a small magic number, it is more likely to detect a frame header when there is just a noise. Another interesting trade off example is to determine the size of your frame, if it is too big, it may have more opportunity to lost a bit but if it is too small, it may take more time to transfer the data. There is no answer what size you should set and how big the magic number you should choose, it is all about trade off.

CS120 is a great course to have a acknowledge about computer network, and CS120 project is also a great project where we can truly understand the computer network.