CS120 Project Report

Anonymous Authors

1 Introduction

In the course project, I built a network with sound as the transmission medium. The project progressed step by step, from using air to propagate sound to using audio cables for transmission, from only send to send with confirmation (ACK), and from local interconnection to interaction with the Internet. Throughout the project, despite facing numerous challenges, I learned a lot. It served as excellent hands-on practice for the content learned in class, providing me with a more direct understanding of protocol design details.

2 Project 0

Project 0 took place during the course's add/drop period at the beginning of the semester. The objective was to familiarize ourselves with controlling the sound interface using programming. I utilized Java and ASIO4ALL for this project. It also served as a reminder to fellow students, urging us to assess whether we could successfully complete subsequent projects. To be honest, my self-assessment was incorrect. I later discovered that I disliked dealing with hardware. Even though Project 0 seemed very user-friendly in hindsight, during its execution, my computer consistently failed to correctly use ASIO4ALL. Due to a lack of knowledge about sound interfaces, I initially couldn't determine whether the issue lay with my code, ASIO4ALL settings, or if my computer's sound card was incompatible. This consumed a significant amount of my time (only to find out it was a sound card compatibility issue—quite sad). This realization led me to understand that I disliked working with hardware because, if it was a hardware problem, I had no means of resolving it (even identifying and locating the issue required a lot of effort).

Of course, achieving success in recording and playback was ultimately satisfying, even though the sound quality left something to be desired. Regarding this project, I believe its existence is necessary. As far as I know, many students dropped this course during the add/drop period, indicating that the warning from Project 0 was effective. The remaining students either had confidence or were well-prepared to tackle the challenges head-on (myself for example). In any case, it's not easy to face a mid-term breakdown (although I almost considered dropping the course while working on Project 2).

The only aspect I feel needs improvement is the communication platform. WeChat groups are not an ideal discussion platform due to the format of messages. I believe platforms like Piazza are more suitable; they classify questions more clearly and are convenient for searching (e.g., separate categories for Project 1, Project 2, etc.). Piazza also supports anonymous posting (sometimes I'm afraid of asking something too silly, so I refrain from asking; I know this should be corrected, but maybe I need time). For classmates with similar or derivative questions, they can find answers/discussions under the same post on Piazza. In contrast, WeChat group messages can become chaotic, with updates potentially pushing discussions about the same issue out of view. After the mid-term evaluation, when the teacher posted screenshots in the group, there was suddenly a lot of discussion (compared to the sparse messages before this). I believe this indicates that students had things to say, but there wasn't a suitable platform. Based on my previous experience in other courses that used Piazza, discussions on Piazza were more active. I assume the course's use of a WeChat group is due to the consideration of timely replies. However, I feel it's counterproductive if there's no ongoing discussion, and Piazza can also be configured to send email reminders for new posts every few hours. A delay of a few hours is not an issue for me, at least.

3 Project 1

Project 1 combines elements from Project 0 and serves as the first formal project. In this project, I initially designed a packet, consisting of a header and data. Upon completing this project, the audio network could transmit data between two hosts using sound propagation through the air. The modulation scheme for binary data was Phase Shift Keying (PSK), and the preamble utilized a chirp signal.In this project, I designed each packet to consist of a preamble of length 16 and data of length 100, with both measured in symbols. The transmission parameters were set as follows: a sampling rate of 48000Hz, a carrier frequency of 11000Hz, a symbol rate of 1000 (meaning each symbol comprises 48 samples), and a chirp signal with a minimum frequency of 2000Hz and a maximum frequency of 10000Hz.

Initially, I attempted to modulate and demodulate based on the example code's instructions. However, there were persistent issues in some details. Fortunately, with the help of the example code and a step-by-step comparison of results, I identified the problem (a bug in the demodulation calculation). The primary challenges I encountered were related to parameter adjustments and environmental influences. These parameters included packet length, recognition threshold, and computer volume. Although the project documentation mentioned that the packet should not be too long, I had no concept of what "long range" meant in this context. As a result, the initially set packet length was too short. Determining the recognition threshold and computer volume involved trial and error. Due to the difficulty of controlling variables (as I was simultaneously investigating other issues), this consumed a significant amount of time. Additionally, the computer volume should not be too high, as it drastically affects sound quality, making testing a painful experience.

Moreover, I find one aspect of this project somewhat problematic (although I don't have a clear solution): it has environmental requirements. Noise has a significant impact on transmission effectiveness (for instance, someone dragging a chair upstairs or voices from the adjacent room while in a library discussion room). This led to many instances where I had to wait for the right conditions while testing the project.

4 Porject 2

Project 2 shifted the audio transmission from air propagation to transmission through audio cables. This significantly improved the accuracy and stability of the transmission. In this project, a medium-awareness module was introduced, enabling two hosts to symmetrically transmit data. The MAC frame I used consisted of a 60-sample preamble (still a chirp signal, hardcoded and stored directly in the Config file), a 4-bit offset, a 2-bit type, a 10-bit index, a 2-bit destination, 300 bits of data, and an 8-bit CRC, making a total of 326 bits (excluding the preamble). As audio cables are more reliable than air propagation, the bit rate could be increased. I employed line coding with a symbol rate set at 12000Hz (1 bit corresponding to 4 samples, where bit 1 corresponds to 4 samples of 1.0f, and bit 0 corresponds to 4 samples of -1.0f). During decoding, I only used the sum of the middle two samples, considering the significant influence of hardware transitions on the first and last samples.

The state machine in the documentation was clear and helpful for implementing the MAC layer. However, following this state machine, my program couldn't consistently achieve the task of completing the transmission within 20 seconds for Task 2. Based on the timestamp, I speculate that decoding takes some time, combined with the inherently long Round-Trip Time (RTT). While using a sliding window efficiently improves efficiency, since the task instructions did not explicitly require sliding window implementation, I believe it might be more appropriate to extend the checkpoint deadline.

In Task 3, achieving medium awareness to allow the two nodes to transmit without conflicts proved to be challenging. I did not successfully complete this task consistently; success was occasional. I attribute this to the long RTT discovered in the previous task. Additionally, the echo ring in the circuit might be a possible reason for the failure of medium sensing.

A sentence in the documentation might need modification for clarity (the part in parentheses): "When the BACKOFF timer reaches zero, sense the medium (generally, at least longer than the frame interval between a DATA and the corresponding ACK)."

This was the most challenging project for me, and I contemplated dropping or withdrawing from the course multiple times. However, it also provided me with valuable learning experiences. A

side note is that, based on the reminder from Project 0, I initially thought multithreading would be complicated, but during the actual writing, as long as the state machine was correct and attention was paid to resource usage, there were no other issues. Furthermore, in this project, I encountered a problem speculated to be due to hardware reasons: after a sequence of 1s (0s), if there is a 0 (1), it is recognized correctly. However, if a 1 (0) follows, the energy of the sample for this 1 (0) is lower (higher), leading to it being identified as 0 (1). This issue also arose during the checking process.

4.1 Project 3

Project 3 involved connecting the previously established Ethernet to the internet, confirming reachability and Round-Trip Time (RTT) through ICMP echo, with the core being the implementation of routing. It required distinguishing between different source and destination IP addresses for forwarding (receiving). The main challenge I encountered was how to use APIs to encapsulate packets.

I used Jpcap and referred to online code to construct packets. I mimicked the packets from terminal pings by capturing them through Wireshark to create identical packets (originally, I filled in the wrong destination MAC, so I couldn't receive a reply). Additionally, I encountered a strange issue with one of my computers. When using Jpcap to capture packets, it identified packets with a sequence number greater than 127 as negative, while Wireshark captured them correctly. Hence, I suspected that Jpcap assumed the sequence field to be only 8 bits, considering Jpcap is quite old. I haven't tried other libpcaps.

5 Project4

For Project 4, I only completed Task 1-DNS. This task aimed to make the Ethernet more user-friendly by enabling it to resolve the correct IP address from somedotname.com. Regarding its implementation, I have some questions. The documentation states: "Configure the local DNS server on NODE1 to NODE2-3.IP (need to implement a local DNS server on NODE2), 10.15.44.11, or 1.1.1.1, so that NODE1 can resolve domain names." My understanding of this sentence is that the DNS server on Node1 needs to be configured with one of three options: NODE2-3.IP, 10.15.44.11, or 1.1.1.1. What I find confusing is that Node1 is not connected to Wi-Fi, so it cannot configure its local DNS server. I believe it can only resolve DNS by using Node2 connected through an audio cable. Therefore, my approach is to have Node1 encode somedotname.com into binary data through ASCII encoding, send this packet to Node2 via Ethernet, and let Node2, connected to Wi-Fi, handle the DNS resolution (I can configure Node2's DNS server as 10.15.44.11 or 1.1.1.1). Node2 then encodes the resolved IP into binary data and sends it back to Node1 through Ethernet, allowing Node1 to store this domain name-IP pair. If my approach is correct, perhaps the wording in the documentation needs to be clarified, as it was quite confusing for me.

6 Conclusion

In summary, despite my continued dislike for hardware-related projects (due to their challenging debugging nature!), this course project has been the most interesting one I encountered at ShanghaiTech. Building a network from scratch and achieving a sense of accomplishment and fulfillment upon completion, even with the presence of numerous bugs, has been a rewarding experience. My only suggestion is to change the discussion platform (as the WeChat group sees minimal discussion, the BB discussion is rarely used, and GitHub's wiki is practically unused). Especially in the initial projects like Proj1 and Proj2, unfamiliarity with hardware makes everything seem mysterious (for instance, I initially had no idea that even with an audio cable connection, excessive volume could cause interference due to vibrations - I couldn't even think in that direction without guidance, thus having no idea about how to ask questions), so having an active discussion environment is essential.

In conclusion, I am very grateful that the university offers such a course. It has been enjoyable and has taught me a lot. This project provided me with valuable engineering experience, not just considering algorithm performance and bugs arising from various boundary conditions, but actually designing how to make it successfully fulfill its intended functionality. The content taught by the instructor is also clear and straightforward, providing a clear train of thought during learning. I can appreciate the development journey of computer networks, from identifying a problem, finding solutions, making

further improvements, and updating with each generation, all with logical clarity. I wish this course continued to improve!