Building a Sound Network: Implementing Reliable Sound Transmission with Java

Zhiyi Ye
ShanghaiTech University
Shanghai, China
yezhy1@shanghaitech.edu.cn

Kangjie Xia
ShanghaiTech University
Shanghai, China
xiakj@shanghaitech.edu.cn

Abstract

This project utilizes Java's ASIO interface and sound library to implement a sound network, with the assistance of pcap4j for internet transmission. We started from scratch to build the sound network. Initially, we used ASIO to implement the recording and playback functions for sound. Building upon this foundation, we combined PSK modulation methods to implement the physical layer of the sound network. Next, we designed the MAC layer based on the ACK protocol to achieve reliable sound transmission. Subsequently, we utilized pcap4j to implement the IP layer, including the transmission of information between routers in different network modules. Finally, we implemented website access functionality based on UDP, TCP, DNS, and HTTP protocols.

1 Introduction

The rapid advancement of technology has led to the development of various networks, including voice networks that can transmit audio data over the internet. In this project, we leverage the powerful capabilities of Java's ASIO interface and sound libraries, along with pcap4j for internet transmission, to build a voice network from scratch.

Our main objective is to create a robust and efficient voice network that enables basic network functionality. To achieve this goal, we first implement the fundamental functionalities of recording and playback using ASIO. Building upon this foundation, we delve into the physical layer domain by incorporating PSK modulation methods to ensure reliable transmission within the voice network.

Moving up the protocol stack, we design a Media Access Control (MAC) layer based on an ACK protocol. This layer plays a crucial role in maintaining the integrity and reliability of voice transmission. With the MAC layer in place, we can ensure accurate and lossless delivery of voice data.

The project utilizes pcap4j to implement the Internet Protocol (IP) layer, further expanding its capabilities. This enables the voice network to handle information transfer between routers across different network modules. By integrating IP functionality, we enhance the scalability and versatility of the network.

Lastly, we combine widely used protocols such as UDP, TCP, DNS, and HTTP to achieve practical applications, especially seamless website access within the sound network. This integration allows users to browse websites seamlessly, leveraging the underlying robust network infrastructure.

In summary, the project aims to create a comprehensive voice network that encompasses various protocol layers, ensuring reliable voice transmission and enabling practical internet-based functionalities.

2 Environment Dependence

Our project depends on the following components in our environment:

1. JDK 11.0.6 version of Java environment
2. ASIO library
3. Java's sound library
4. pcap4j library

3 Project 1

3.1 Recording and Playback

We can easily interact with audio devices by utilizing the interfaces provided by the ASIO library. We have designed a user-friendly UI interface to facilitate convenient sound recording and playback.

3.2 Generating Sound Waves of Arbitrary Frequencies

We have generated a time sequence sample of a sound wave using specific frequencies. For example, generating a sound wave $f(t) = \sin(2\pi 1000t) + \sin(2\pi 10000t)$: 1. First, we need to calculate the time 't', which is obtained by dividing the current sample index 'sampleIndex' by the sample rate 'sampleRate'. This can be represented as:

$$t = \frac{\text{sampleIndex}}{\text{sampleRate}}$$

2. Then, we use the sine function to generate two sine waves. The frequency of the first sine wave is 1000Hz, and the frequency of the second sine wave is 10000Hz. This can be represented as:

$$f(t) = \sin(2\pi \times 1000 \times t) + \sin(2\pi \times 10000 \times t)$$

3. Finally, we add these two sine waves together and divide by 2 for normalization.

3.3 File Transfer

1. **File Sending:**
   (a) We first implement file reading using the Java File library, converting it into a string.
   (b) Then, we modulate it into a carrier wave based on PSK.
   (c) Add a physical layer header.
   (d) Finally, we insert a silent interval between each packet.
2. **File Receiving:**
   (a) When a predetermined fixed form of sound signal (preamble) is detected in the audio network, it enters the receiving state.
   (b) Demodulate the carrier wave.
   (c) Use a smoothing function to detect the phase of the sound for decoding.

3.3.1 Modulation

$$\text{signal} = \begin{cases} \sin(2\pi f t) & \text{if data bit} = 0 \\ \sin(2\pi f t + \pi) & \text{if data bit} = 1 \end{cases}$$

where f is the carrier frequency, t is the time.

3.4 Error Correction

Due to the transmission of sound in an open environment, errors can easily occur, and we have not introduced an ACK mechanism. Therefore, when errors occur during transmission, we need to use additional information for error correction. We have employed Hamming codes for error correction, specifically single error correction and double error detection. As we have adopted error correction codes, we have omitted the CRC part.

4 Project 2

4.1 Wired High-Speed Transmission

Audio cables have signal fidelity. Audio cables are specially designed cables for transmitting audio signals, featuring excellent shielding and insulation performance. They effectively reduce the impact of interference and noise, thereby maintaining signal integrity and accuracy. In contrast, wireless transmission is susceptible to environmental noise, electromagnetic interference, and signal attenuation, leading to a decrease in signal quality.

Referring to Task 3 of Project 1, we made no changes to the code but only increased the bit rate. After connecting the audio cable, it is capable of supporting high-speed transmission.

4.2 File Exchange

In order to achieve higher-speed file exchange, we did not send an ACK after each received packet. Instead, we initiate a new transmission after successfully receiving a certain number of packets. We prearranged the number of transmissions to achieve a simple file exchange process.

5 Project 3

In this project, information is transmitted between the Local Host and the Router via a sound network, between the Mobile Host and the Router via a WiFi hotspot, and between the Router and the outside world via the Internet. The Router acts as a bridge connecting three major networks: the sound network, WiFi hotspot, and the Internet. It supports basic ping functionality.

5.1 Local Host

5.1.1 Ping

To minimize the amount of information transmitted in the slower sound network, we simplified the structure of the Ping packet in the sound network:

Table 1: Packet Structure

Byte Range	Field	Description
1–4	Local IP	IP address of the local host
5–10	Local MAC	MAC address of the local host
11–14	Target IP	IP address of the target host
15–20	Target MAC	MAC address of the target host
21	TTL	Time-to-Live
22–23	ID	Identification number
24–25	Seq	Sequence number

The content of the Ping packet is sent as the payload in the physical layer.

5.1.2 Reply

When the Local Host receives an external ping request forwarded by the Router, it automatically identifies the IP and responds with a Ping response to the Router. Specifically, it means simply swapping the IP address and MAC address.

5.2 Router

5.2.1 Reply

When the destination IP and MAC in the ping packet match those of the Router itself, it behaves similarly to the Local Host Reply.

5.2.2 Forwarding

It implements the functionality of Network Address Translation (NAT) for external networks: forwarding the corresponding content based on requests from different networks.

Specifically:

1. When a Local Host in the Sound Network attempts to ping an external internet IP, the sender's IP and MAC are replaced with the Router's, the TTL is decremented by one, and then it is sent to the external network.
2. When a Local Host in the Sound Network attempts to ping a Mobile Host, the TTL is decremented by one, and then it is sent to the Mobile Host.
3. When a Target Host in the external network attempts to reply to a ping packet, the packet is encapsulated in the structure of the Sound Network, the destination IP and MAC are replaced with that of the Local Host, the TTL is decremented by one, and then it is sent to the Sound Network.
4. When a Mobile Host in the WiFi hotspot network attempts to ping the Local Host, the packet is encapsulated in the structure of the Sound Network, the TTL is decremented by one, and then it is sent to the Sound Network.

6 Project 4

6.1 DNS

Our router uses the IPv6 protocol to access the DNS server based on the information transmitted by the Local Host. It then transfers the queried IP back to the Local Host.

6.2 TCP

Based on the content of ICMP protocol in Project 3, we have added the functionality of TCP. We reconstruct TCP packets based on the TCP request information sent by the Local Host and send them to the external Internet.

To minimize the amount of information transmitted in the slower sound network, we simplified the structure of the Ping packet in the sound network:

Table 2: Packet Structure

Byte Range	Field	Description
1–4	Local IP	IP address of the local host
5–10	Local MAC	MAC address of the local host
11–14	Target IP	IP address of the target host
15–20	Target MAC	MAC address of the target host
21	TTL	Time-to-Live
22-25	Seq	TCP sequence number
26	Flag	packet flag, include DNS, Ping, Curl
27-N	Payload	website data, such as www.example.com

And we have implemented the three-way handshake for TCP, with the most important part being the modification of Sequence Number and Acknowledgment. The rules are as follows:

$$\begin{cases} \text{Seq}' = \text{Ack} \\ \text{Ack}' = \text{Seq} + \text{payload length} \end{cases}$$

In contrast, we did not fully implement the four-way handshake. Instead, we directly send a termination packet with the Set flag set to 1 to forcefully terminate the connection.

6.3 HTTP/1.1

After establishing a TCP connection, we send an HTTP GET request. The content of the GET request is based on a previous browsing request made by a browser, with the website replaced by the one sent by the Local Host. We specify the encoding format as plaintext for transmission.

If the content received from the other party consists of multiple packets, we automatically concatenate the transmitted information and automatically identify the content belonging to the HTML part. Finally, we save the HTML file locally and open the webpage using the default browser.

7 Achievements

We have gained valuable experience and knowledge through the completion of this project. Firstly, we have learned how to build an audio network from scratch and successfully implemented basic network functionalities such as ping and website access. This project provided us with a deep insight into the various layers and protocols of computer networking, while also enhancing our programming skills and teamwork abilities.

Secondly, we have mastered the use of Java's ASIO interface and audio libraries, as well as pcap4j for internet transmission. These tools have enabled us to have a more comprehensive understanding of the practical implementation and operation of computer networks.

Lastly, this project has made us realize the significance and wide-ranging applications of computer networks. We have come to recognize that computer networks are an indispensable part of modern society, supporting numerous activities and communications in our daily lives. We now have a deeper understanding and appreciation for the convenience and benefits brought about by network technology.

8 Course Suggestions

1. The project description is not accurate enough, especially regarding the requirements in the check section. For example, in Project 4, task 5, the part about opening HTTP in a browser specifically mentioned the use of a virtual network card. We believed that opening an HTML file directly in a browser would suffice, but during the check, we were unexpectedly required to implement it using a virtual network card.
2. Some tasks that require a significant amount of work have been assigned very few points and are included in the mandatory section. For instance, Project 2, task 4, and Project 4, task 2. I find this unreasonable as mandatory tasks should be something everyone is willing to do, given the points allocated. Most students are not inclined to spend so much time on tasks that offer minimal points. On the other hand, the bonus section should encourage voluntary completion and the points allocation should not necessarily reflect the workload.