**浙江工商大学计算机与信息工程学院**

**实验报告**

**课程名称： 密码货币与区块链技术 姓 名： 梁宇航 沈林杰 黄尧**

**学 号：2212190506 2212190519 2212190512**

**指导教师：邵俊 班 级： 安全2201 日 期： 2024年 11月 5日**

**【一】实验内容及要求**

**实验名称：**P2P 系统实现

**实验目的：**实现一个nat 穿透的 p2p 系统

实验环境

· **操作系统**：Linux / Window / Mac

· **开发工具**：Rust

实验内容

1. Nat 穿透
2. P2P 实现

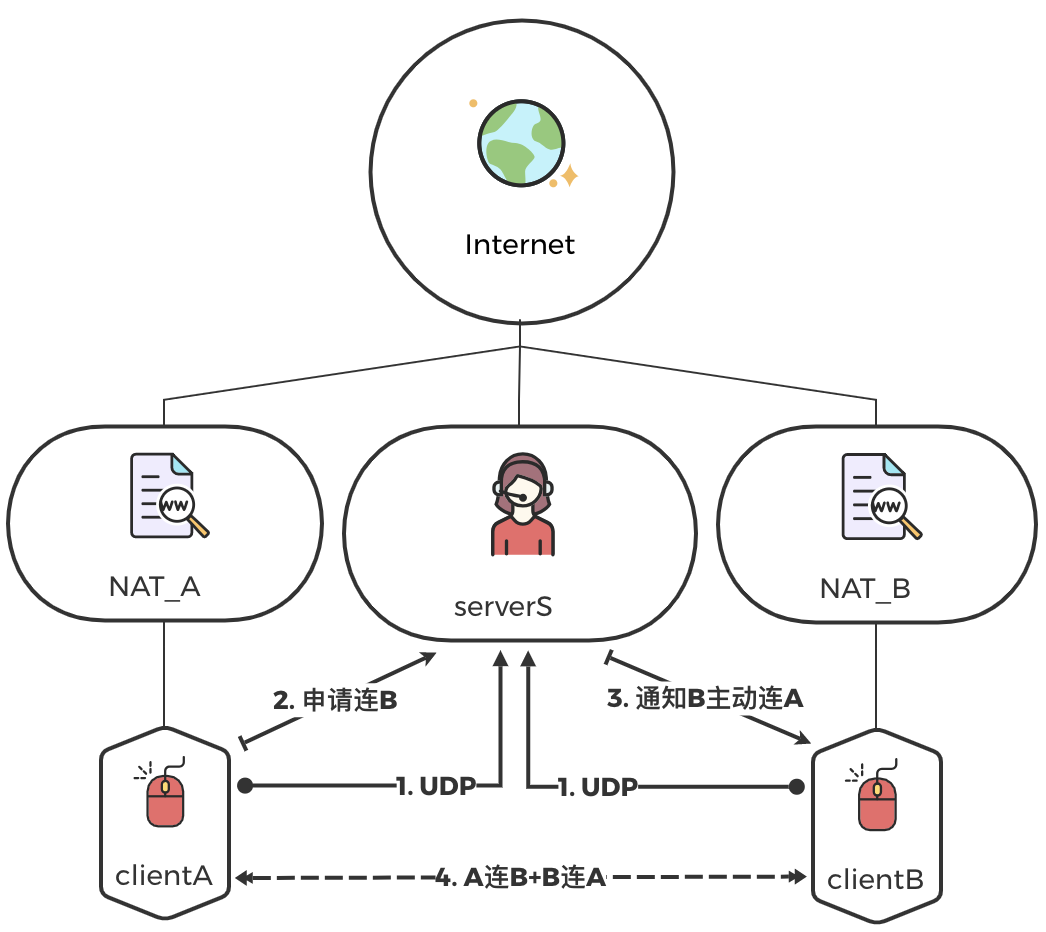
**【二】实验过程及结果**

**实验内容**

### 概述

该代码实现了一个简单的P2P（点对点）系统，使用UDP协议进行NAT 穿透后通信。系统包含服务器和客户端两部分，服务器负责维护会话列表，客户端可以向服务器请求会话信息并与其他客户端通信。

实现原理如下：



图表 1 P2P 系统实现原理

因本程序仅作为demo实现，省略了步骤（2）（3）

Nat 穿透部分基于 UDP socket 复用机制实现，基于 UDP 协议的不稳定性，引入指数退避算法在消息未取得确认报文时重发。具体过程如下：

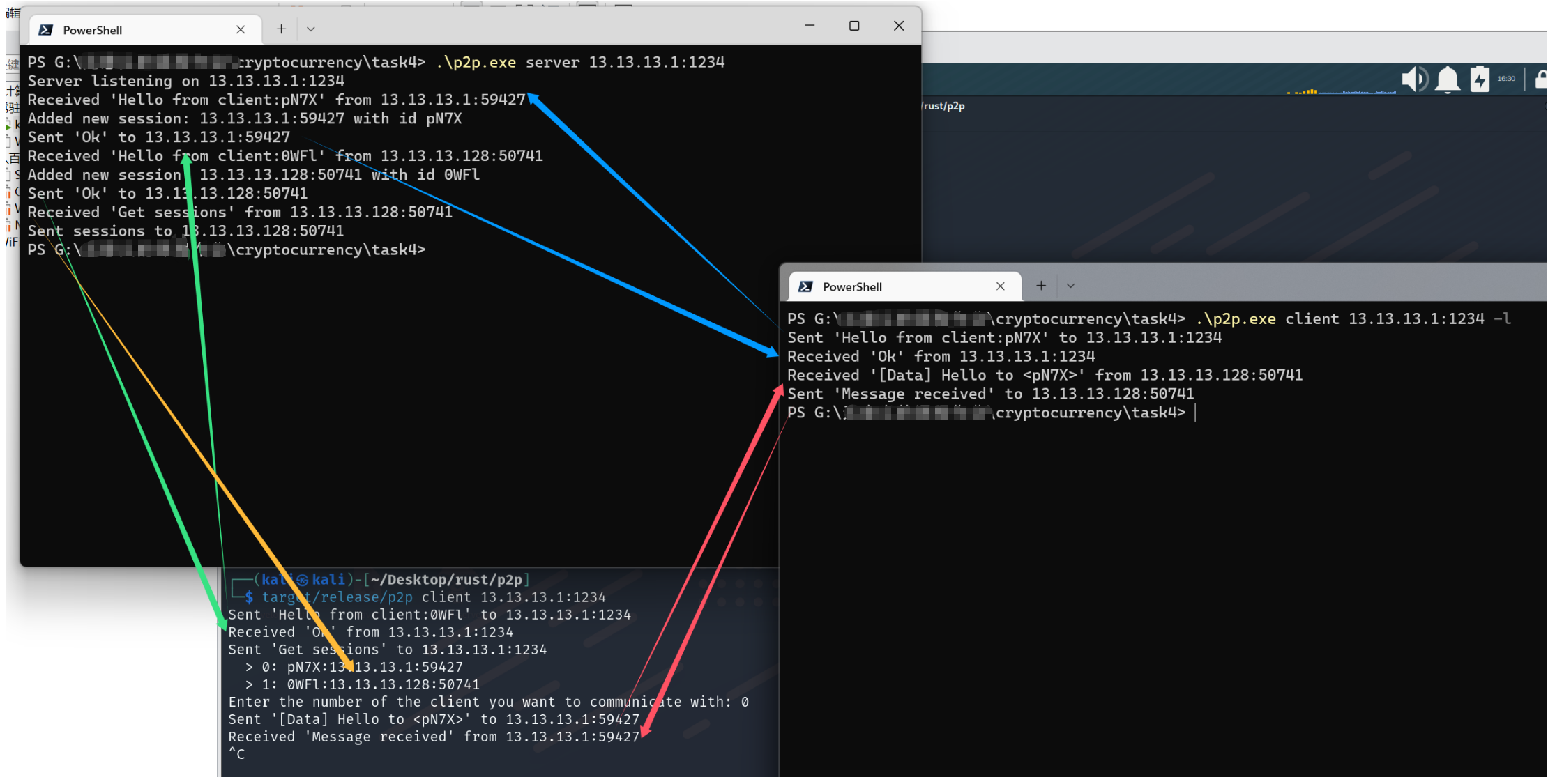
ClientA 首先先向 Server （以下简称 S）发送 hello 包——向服务器注册自己，Client 信息在服务器端通过如下结构体存储

|  |
| --- |
| *struct* Session {      id: String,      addr: SocketAddr,  } |

其中，id 为 4为随机字符串，由 Client 生成

服务器再收到 hello 包后，会向ClientB 返回当前注册过的 sessions 列表，收到列表的 ClientB 可以根据 Session 中的公网映射地址来直接与 ClientA 通信

### 演示



### 使用方法

1. **运行服务器**：

cargo run -- server <port>

1. **运行客户端**：

cargo run -- client <server\_addr> <client\_addr>

### 关键代码

指数退避算法：

|  |
| --- |
| while attempts < max\_attempts {  socket  .send\_to(msg, server\_addr)  .expect("Couldn't send data");  println!("Sent '{}' to {}", str::from\_utf8(msg).unwrap(), server\_addr);  let mut buf = [0; 1024];  socket  .set\_read\_timeout(Some(Duration::from\_secs(1 << attempts)))  .expect("Couldn't set read timeout");  match socket.recv\_from(&mut buf) {  Ok((amt, src)) => {  let data = str::from\_utf8(&buf[..amt]).expect("Couldn't parse data");  if data == "Ok" {  println!("Received '{}' from {}", data, src);  break;  }  }  Err(e) => {  println!("Attempt {} failed: {}", attempts + 1, e);  }  }  attempts += 1;  } |

Session 池维护：

|  |
| --- |
| *let* *mut* sessions = session\_list.lock().unwrap();  *if* data.starts\_with("Hello from client:") {  *let* id = &data["Hello from client:".len()..];  *if* *let* Some(session) = sessions.iter\_mut().find(|*s*| *s*.id == id) {                  session.addr = src;                  println!("Updated session: {} with new address {}", id, src);              } *else* {                  sessions.push(Session {                      id: id.to\_string(),                      addr: src,                  });                  println!("Added new session: {} with id {}", src, id);              }          } |

全部代码参见文档所附工程文件