# 一、引言

## 1.1 研究背景

随着互联网技术的快速发展与普及，网络游戏已成为全球范围内广受欢迎的娱乐形式。其中，实时性强、交互性高的联机游戏更是吸引了大量玩家。为了确保这类游戏的顺畅体验，选择一种高效、可靠的网络传输协议至关重要。用户数据报协议（UDP）以其低延迟、高实时性、良好带宽利用率和高度自定义性等特点，逐渐成为游戏联机设计与实现的首选。然而，面对NAT限制、公网IP地址短缺等网络环境挑战，以及数据同步、冲突解决、预测性延迟补偿、丢包处理、性能优化等技术难题，如何基于UDP构建稳定且高效的联机系统仍需深入研究。

## 1.2 研究目的与意义

本研究旨在设计并实现一套基于UDP的游戏联机系统，解决网络环境制约、数据同步、延迟补偿、丢包处理等关键问题，提升玩家游戏体验，确保数据可靠传输，并适应不同网络环境。具体研究意义如下：

1. 低延迟与高实时性：利用UDP的特性降低网络传输延迟，提高游戏实时性，满足竞技类、动作类等对时间敏感游戏类型的需求。

2. 优化资源利用：通过有效利用带宽和减少协议开销，提升网络资源利用率，支持大规模并发联机。

3. 提升游戏体验：通过先进的数据同步和延迟补偿机制，减少操作滞后感，增强玩家沉浸感。

4. 保障数据一致性：设计可靠的丢包处理和重传机制，确保游戏状态在所有参与者间保持一致，避免因数据丢失引发的异常行为。

5. 适应性与普适性：研发适用于多种网络环境的技术方案，使游戏联机功能能够在不同网络条件下稳定运行。

## 1.3 研究内容概述

本研究将涵盖以下几个方面：

- 数据同步与冲突解决机制设计

- 预测性延迟补偿算法开发

- 丢包处理与重传机制研究

- 性能优化与负载均衡策略制定

- 适应不同网络环境的技术探索

# 二、文献综述

## 2.1 网络通信现状分析

本节将详细探讨当前网络通信环境的特点，特别是NAT限制与公网IP地址匮乏对游戏联机功能的影响，分析其对玩家连接、数据传输、服务器架构等方面的挑战。

## 2.2 相关解决方案评估

针对上述网络环境问题，本部分将回顾已有的解决方案，如STUN、TURN、ICE等NAT穿透技术，以及针对IP地址短缺的多玩家共享IP策略等，并对其适用性、优缺点进行评估。

# 三、理论基础

## 3.1 UDP协议基础

UDP（User Datagram Protocol）是一种无连接的传输协议，常用于在计算机网络上发送数据包。与TCP（Transmission Control Protocol）相比，UDP更加简单和轻量，但也具有一些特定的特性和限制。UDP是一种无连接协议，发送端和接收端之间不需要建立持久的连接。UDP是无连接的,每个UDP数据包都是独立的，它们可以按照任意的顺序到达接收端。UDP是快速和轻量的,由于UDP没有建立和维护连接的开销，它具有延迟的延迟和更小的数据包开销。这使得UDP在实时性方面要求较高的应用中的表现。无拥塞控制：UDP不提供拥塞控制机制，因此发送端可以以任意的速率发送数据包。这使得UDP在某些特定场景下更加适用，例如音频视频传输和实时游戏。多播和广播支持：UDP支持将数据包同时发送给多个接收端，这在多播和广播场景中非常有用。UDP使用简单的数据报文格式，每个数据包都包含源端口号和目标端口号，以及数据内容。发送方将数据包发送到接收方的IP地址和端口号，然后接收方则通过监听指定的端口来接收数据包。UDP不保证数据包的可靠传输，也不提供重传机制或流量控制。

## 3.2 P2P架构原理

P2P（Peer-to-Peer）网络架构是一种分散的网络架构，其中所有参与者（对等节点）具有相同的功能和地位，可以充当服务提供者和服务请求者。P2P网络的基本概念是直接连接和共享资源，而不依赖于中央服务器。这种网络架构在游戏连接中有广泛的应用。主要分为两种大类,

1. 纯P2P网络：在纯P2P网络中，对等节点之间直接通信，没有中央服务器的存在。节点可以共享资源、提供服务和请求服务，形成一个去中心化的网络。

2. P2P网络：混合混合P2P网络结合了P2P和客户端服务器（C/S）模式的特点。在这种网络中，一部分节点充当中央服务器的角色，提供服务发现、路由和协调功能，而其他节点则通过这些中央节点进行通信。

在游戏连接中的应用：:P2P网络可以用于游戏的下载和更新，通过直接从其他玩家的计算机上获取游戏文件，减轻游戏服务器的负担。玩家可以同时下载和上传游戏文件，提高下载速度和下载的可用性。游戏资源共享：P2P网络可以供玩家之间共享游戏资源，如自定义地图、模组、皮肤等。玩家可以直接从其他玩家那里获取所需的资源，减少对游戏服务器的依赖。游戏连接通信：P2P网络可以用于玩家之间的实时通信，如语音聊天和消息传递。玩家可以直接建立点对点的连接，实现低延迟的通信，而不需要通过中央服务器转发。

对降低服务器负担、提高连接效率的作用：分担服务器负载：P2P网络可以分担服务器的负载，通过直接从其他玩家获取资源和数据，减少对服务器的请求。这样可以降低服务器的带宽和计算压力，服务器的可扩展性和提高稳定性 加速数据传输：P2P网络可以通过直接点对点的连接传输数据，减少数据传输的路径和中间节点，从而提高传输效率并降低延迟。对于这种实时游戏中需要快速传输玩家操作和状态信息的场景非常有益。提供更好的可用性：P2P网络可以通过从其他玩家获取资源和数据，提供更好的可用性。如果游戏服务器出现故障或网络问题，其他玩家仍然可以通过P2P网络相互通信和共享资源，保持游戏的进行。

## 3.3 NAT等级

对于NAT来说,我们一般把NAT分为四个等级,分别为

NAT1：Full Cone NAT（完全圆锥型，一对一）

NAT2：Restricted Cone NAT（地址限制圆锥型）

NAT3：Port Restricted Cone NAT（端口限制圆锥型）

NAT4：Symmetric NAT（对称型）

## 3.3 NAT穿透技术概述

常用的NAT穿透技术旨在解决网络地址转换（NAT）所引起的直接通信难题，以便在私有网络中的主机之间建立直接的通信连接。以下是几种常见的NAT穿透技术及其工作原理：

1. 端口映射（Port Mapping）：通过在NAT设备上配置端口映射规则，将外部网络的请求转发到内部网络的特定主机和端口。这样，可以实现从外部网络直接访问内部网络主机的能力。

2. UPnP（Universal Plug and Play）：UPnP允许设备在网络中自动发现和配置，包括NAT设备。通过使用UPnP协议，应用程序可以与NAT设备通信，并动态地配置端口映射规则，以便在NAT后面的主机之间建立直接通信。

3. STUN（Session Traversal Utilities for NAT）：STUN是一种基于UDP的协议，用于发现和绕过NAT设备。客户端向STUN服务器发送请求，以获取其在NAT后面的公共IP地址和端口。然后，客户端可以将这些地址和端口信息交给其他客户端，以建立直接通信连接。

4. TURN（Traversal Using Relays around NAT）：TURN是一种在NAT后面建立直接通信连接的技术。它使用中继服务器作为中间人，转发数据包以实现通信。当直接通信不可行时，客户端可以将数据发送到中继服务器，然后由中继服务器将数据转发给目标客户端。

游戏联机实现:

1. 端口映射：游戏客户端可以通过配置端口映射规则，使得其他玩家可以直接从外部网络访问到其所在的客户端。这样，玩家可以通过直接的IP地址和端口进行通信。

2. UPnP：游戏客户端可以利用UPnP协议与NAT设备通信，动态配置端口映射规则。通过与NAT设备交互，客户端可以获取到其在NAT后面的公共IP地址和端口，从而使其他玩家可以直接与其建立连接。

3. STUN：游戏客户端可以使用STUN协议与STUN服务器通信，获取其在NAT后面的公共IP地址和端口。然后，客户端可以将这些信息交给其他玩家，以便建立直接通信连接。

4. TURN：如果无法直接建立通信连接，游戏客户端可以使用TURN技术，通过中继服务器进行数据转发。客户端将数据发送到中继服务器，然后由中继服务器转发给目标客户端。

3.4 网络安全原则与加密技术

1. 加密通信：使用加密算法对游戏联机通信进行加密可以保护玩家数据的隐私和完整性。通信双方可以使用对称加密算法（如AES）或非对称加密算法（如RSA）来加密和解密数据。这样，即使有人截获了通信数据，也无法解密和窃取敏感信息。

1. 数字证书：使用数字证书可以验证通信双方的身份和确保通信的可信度。数字证书包含了公钥和相关的身份验证信息，并由可信的第三方证书机构（CA）签名。游戏客户端可以验证服务器的数字证书，以确保与合法服务器建立安全连接。

2. 安全协议：安全协议是一种规定了通信双方之间安全通信规则的协议。例如，SSL/TLS（Secure Sockets Layer/Transport Layer Security）协议用于在网络层上提供安全的通信连接。SSL/TLS使用加密和身份验证机制，防止数据被篡改或窃取。

3. 防止作弊和外挂：游戏联机中，为了保护游戏公平性，需要采取措施防止作弊和外挂。这可以包括使用反作弊技术，如行为分析、数据校验和防外挂软件。此外，服务器端还可以使用加密技术来保护游戏逻辑和敏感数据，以防止外挂程序的攻击。

4. 安全存储：玩家数据的安全存储是保护玩家隐私的重要方面。游戏开发者应采取措施，如加密存储、访问控制和备份策略，以确保玩家数据的保密性和完整性。此外，还需要定期更新和修复潜在的安全漏洞。

5. DDOS防护：分布式拒绝服务（DDoS）攻击可能是游戏联机面临的一个主要威胁。为了应对DDoS攻击，游戏服务器可以采用流量过滤、负载均衡和云防护等措施，以减轻攻击的影响并保持服务的可用性。

# 4.1 **UPnP协议**

Universal Plug and Play（通用即插即用）,先说这个之前,先了解pnp(即插即用),在计算机发展初期，安装新的硬件设备需要手动配置中断和I/O端口，这对普通用户来说是一项复杂的任务。为了解决这个问题，计算机产业引入了即插即用（Plug and Play，简称PnP）技术。PnP技术的目标是使硬件设备的安装和配置过程更加简化和自动化，减少用户的操作和配置工作。通过PnP技术，用户可以简单地将新的硬件设备插入计算机的插槽或接口中，而不需要手动分配中断和I/O端口。PnP可以实现自动设备检测,资源分配,设备驱动程序管理等功能.通用即插即用（Universal Plug and Play，简称UPnP）是在PnP技术基础上的进一步发展。UPnP是一种网络协议，旨在使智能设备和计算机网络更加易于配置和使用。它允许设备在网络中自动发现彼此，并建立通信和共享功能，无需复杂的手动配置。通过UPnP，用户可以轻松地连接和使用诸如打印机、摄像头、网络存储设备等智能设备，而无需手动配置网络设置或安装繁琐的驱动程序。但UPnp不同于常规的端口映射,常规的端口映射,我们需要每次手动调整设备的端口来保证不冲突,如NAT设备下有两个设备都开放了80端口,就不能都占用NAT设备的80端口,且手动配置有点麻烦,这个时候UPnp就可以识别这两个位于NAT下面开放的端口,如果都是80的话,将会自动分配到NAT设备上进行映射,如80映射到80,80自动映射到10000.这样通过UPnP，主机就能够像公网主机一样被网络中的其他主机访问，而无需手动配置端口映射规则。这极大地简化了用户的操作，提供了更便利的网络体验。

# 4.2 **UPnP的应用场景**

UPnP为软件（如BitTorrent、eMule、IPFS、Ethereum等）带来了极大的便利。这些软件通常需要在本地计算机上监听特定的端口，以便其他用户可以直接连接并共享数据。启用UPnP功能后，这些软件可以自动请求NAT设备将所需的端口映射到公网地址上，使得公网上的用户能够直接连接到局域网内的主机。这样，用户之间可以更快速地共享文件、进行区块链交易、下载资源等。实现 UPnP 必须同时满足三个条件：NAT 网关设备必须支持 UPnP 功能；这是因为它需要扮演控制点（239.255.255.250:1900）的角色，控制点提供的是 SSDP 服务。操作系统必须支持 UPnP 功能；比如 Windows 系列操作系统；应用程序必须支持 UPnP 功能；比如 Bt、eMule、IPFS, Ethereum 等。以上三个条件必须同时满足，缺一不可。



# 4.2 SSDP协议

SSDP（Simple Service Discovery Protocol）是UPnP（Universal Plug and Play）的核心协议之一。它定义了一组协议框架，用于实现设备之间的发现和通信。SSDP使用HTTPU（HTTP over UDP）和HTTPMU（HTTP Multicast）规范，基于UDP端口进行通信。SSDP的作用是允许UPnP设备通过与根设备（通常是用户设备）进行交互，自动发现和注册可用的服务。根设备充当了UPnP网络中的控制点，它负责管理和控制网络中的设备。SSDP使用UDP端口（通常是端口号1900）进行通信。设备可以通过监听和发送SSDP消息来实现自动发现和通信。当设备加入网络或有新的服务可用时，它会发送SSDP广播消息进行通知。其他设备可以通过监听这些消息来发现和获取设备的信息。总结起来，SSDP作为UPnP的核心协议，通过使用HTTPU和HTTPMU规范，基于UDP端口进行通信，实现了设备之间的自动发现和通信。它允许设备通过广播消息进行发现、通知和描述，以及进行生命周期管理，为UPnP设备的互联互通提供了基础框架。SSDP 使用一个固定的组播地址 239.255.255.250 和 UDP 端口号 1900 来监听其他设备的请求。

# 四、设计与开发

## 4.1 基于UDP协议的P2P游戏联机功能设计

提出基于UDP协议的P2P游戏联机系统设计方案，包括NAT穿透能力实现、系统架构设计、模块功能介绍等，明确各部分职责与交互方式。

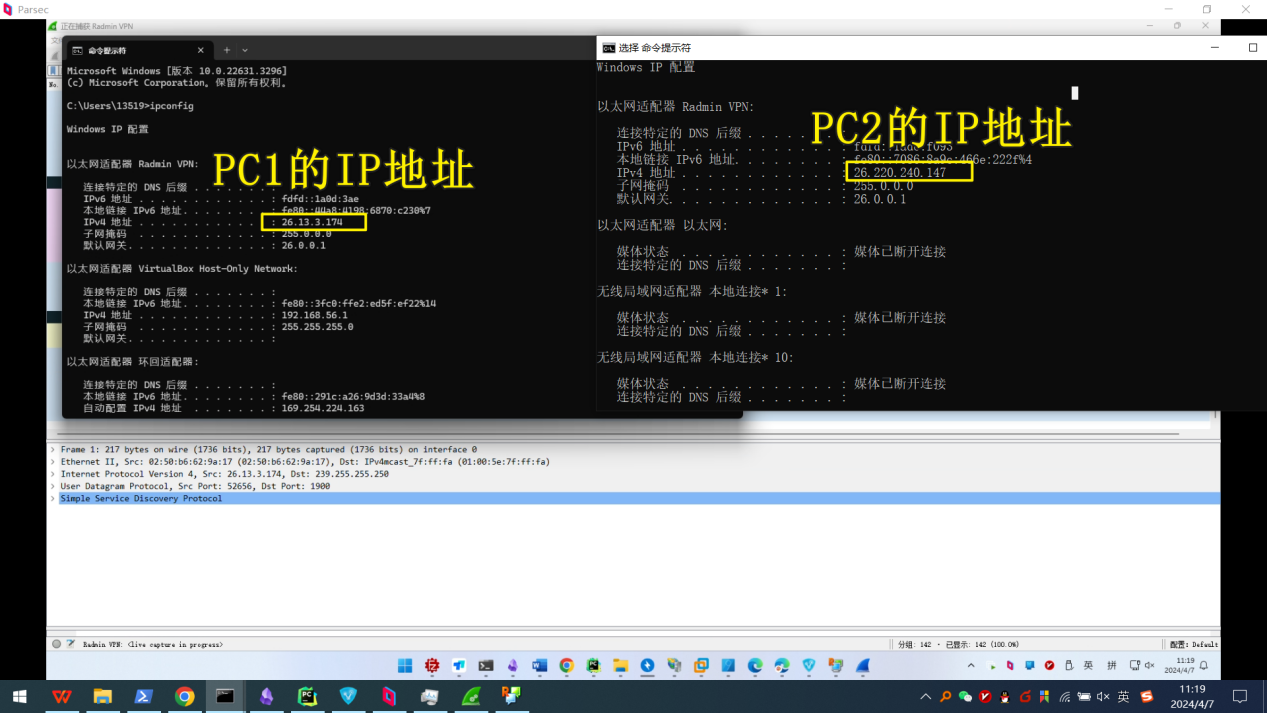
甚至还有限制udp流量的NAT类型

## 4.2 环境搭建与开发工具使用

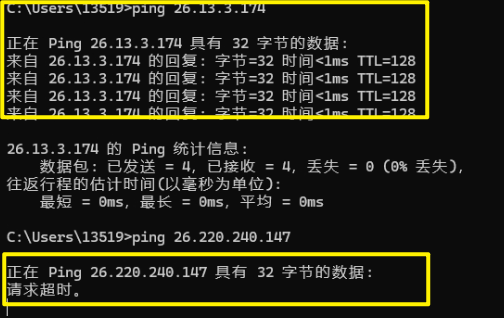
打洞环境:

pc1的ip地址:26.13.3.174

PC2的ip地址:26.220.240.147

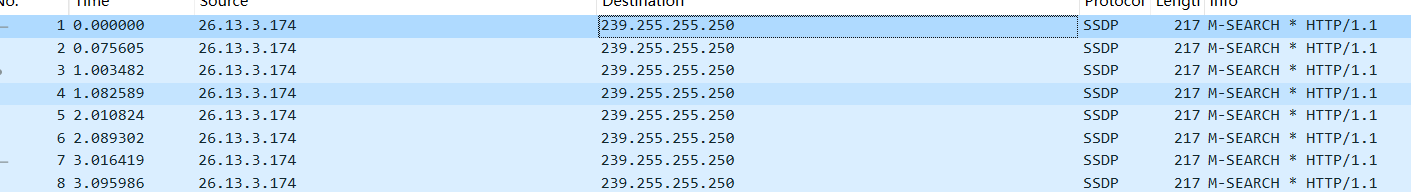


可以看到这两个地址是不能互相ping通的。





使用SSDP协议在局域网内发现和识别网络,简化设备之间的发现和通信。



## 4.3 设备查询

当一个客户端接入网络的时候，它可以向一个特定的多播地址的 SSDP 端口使用 M-SEARCH 方法发送 “ssdp:discover” 消息。当设备监听到这个保留的多播地址上由控制点发送的消息的时候，设备将通过单播的方式直接响应控制点的请求。

相关设备查询请求消息格式：

```

M-SEARCH \* HTTP/1.1

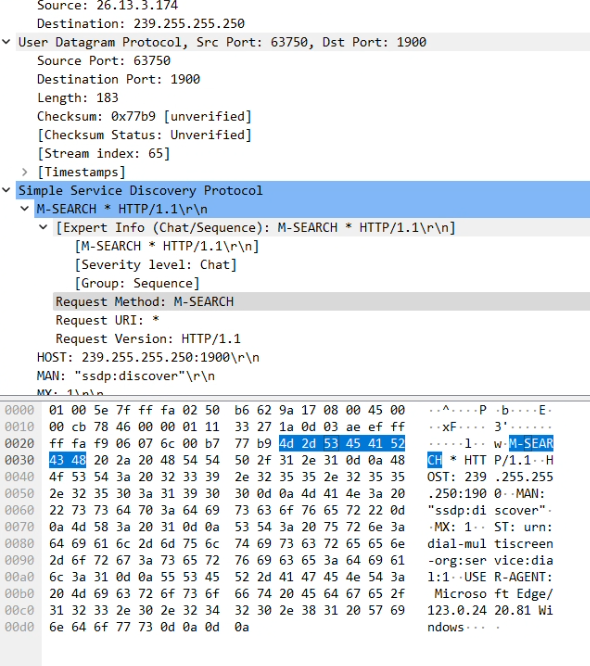
S:uuid:ijklmnop-7dec-11d0-a765-00a0c91e6bf6

Host:239.255.255.250:1900

Man:"ssdp:discover"ST:ge:fridge

MX:3

```











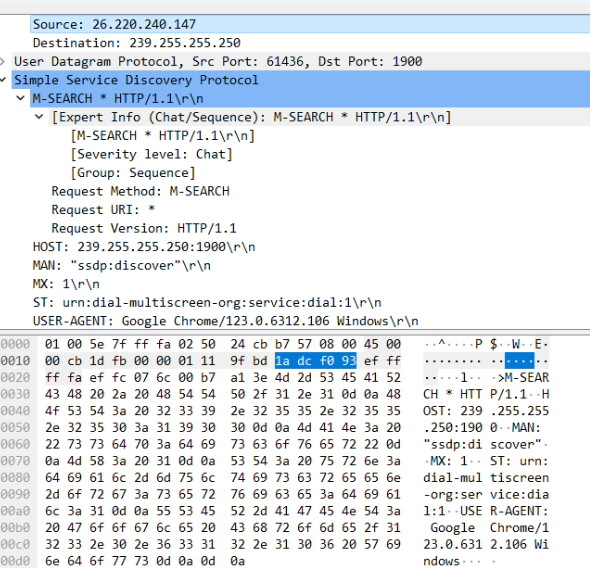
响应消息应该包含服务的位置信息（Location 或AL头），ST和USN头。响应消息应该包含cache控制信息（max-age 或者 Expires头）。

典型的响应消息格式：

```

HTTP/1.1 200 OKCache-Control: max-age= seconds until advertisement expiresS: uuid:ijklmnop-7dec-11d0-a765-00a0c91e6bf6Location: URL for UPnP description for root deviceCache-Control: no-cache="Ext",max-age=5000ST:ge:fridge // search targetUSN: uuid:abcdefgh-7dec-11d0-a765-00a0c91e6bf6 // advertisement UUIDAL: <blender:ixl><http://foo/bar>

```





在设备加入网络时，它应当向一个特定的多播地址的 SSDP 端口使用 NOTIFY 方法发送 “ssdp:alive” 消息，以便宣布自己的存在，更新期限信息，更新位置信息。

## 4.4 设备通知信息

在设备加入网络时，它应当向一个特定的多播地址的 SSDP 端口使用 NOTIFY 方法发送 “ssdp:alive” 消息，以便宣布自己的存在，更新期限信息，更新位置信息。由于 UDP 协议是不可信的，设备应该定期发送它的公告消息。在设备加入网络时，它必须用 NOTIFY 方法发送一个多播传送请求。NOTIFY 方法发送的请求没有回应消息。格式一般为:

```

NOTIFY \* HTTP/1.1HOST: 239.255.255.250:1900CACHE-CONTROL: max-age = seconds until advertisement expiresLOCATION: URL for UPnP description for root deviceNT: search targetNTS: ssdp:aliveUSN: advertisement UUID

```

当一个设备计划从网络上卸载的时候，它也应当向一个特S定的多播地址的 SSDP 端口使用 NOTIFY 方法发送 “ssdp:byebye” 消息。但是，即使没有发送 “ssdp:byebye” 消息，控制点也会根据 “ssdp:alive” 消息指定的超时值，将超时并且没有再次收到的 “ssdp:alive” 消息对应的设备认为是失效的设备。格式一般为：

```

NOTIFY \* HTTP/1.1

HOST: 239.255.255.250:1900NT: search target

NTS: ssdp:byebye

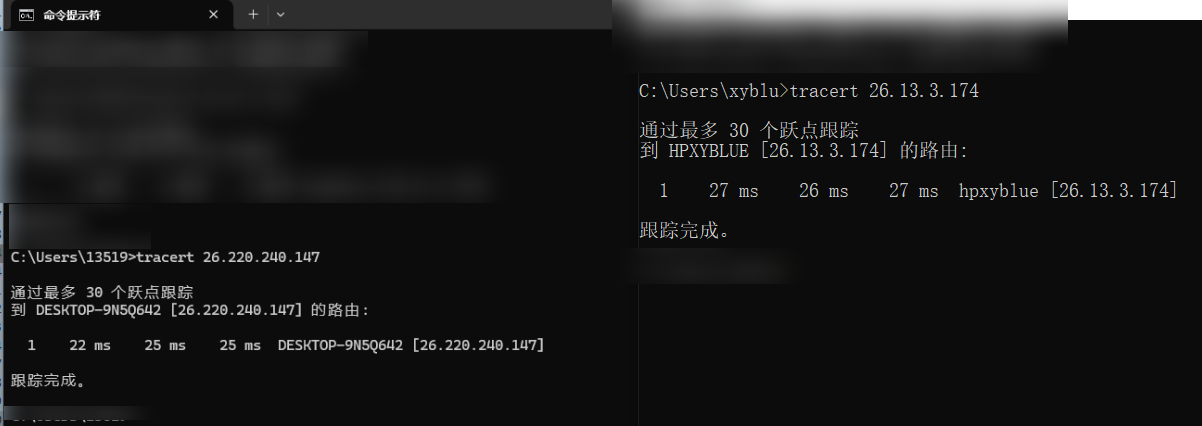
USN: advertisement UUID

```

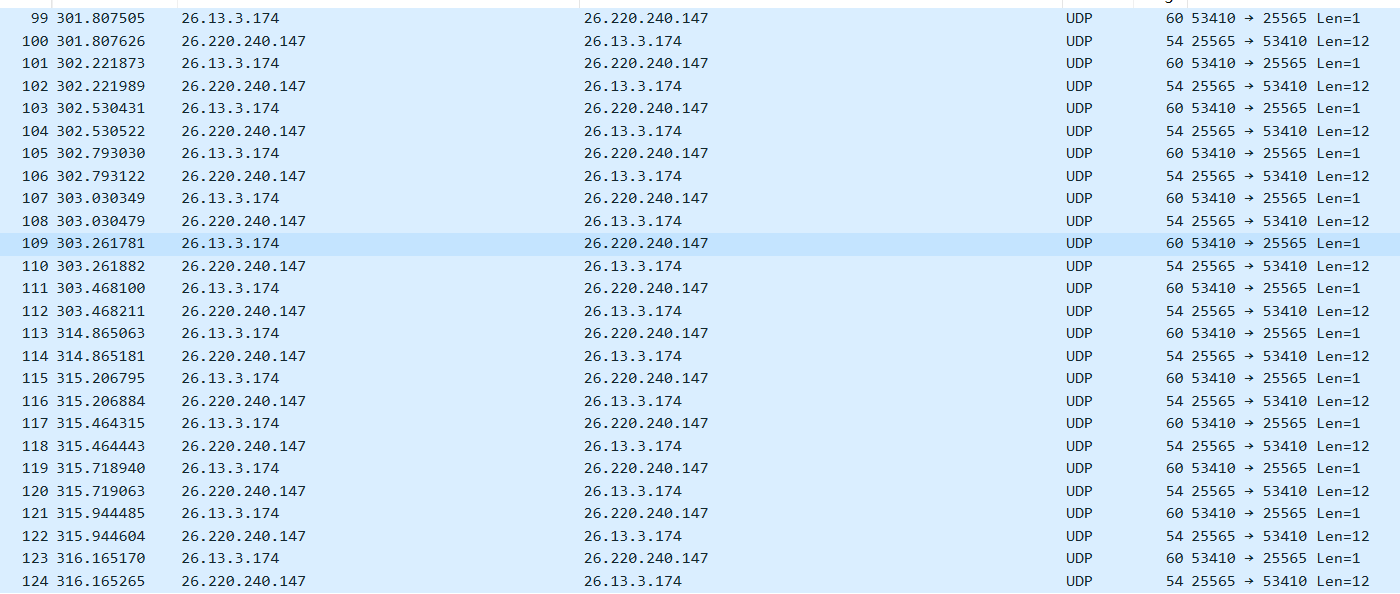
# 五、性能优化与效果评估

## 5.1 延迟与优化方法

建立p2p通讯点对点通讯后，存在25ms左右的延迟。且打洞成功率并不高，对双方环境依赖比较高。需要尽可能降低双方客户端等级来提高打洞的成功率，如家宽的光猫拨号改为路由器桥接来减少一层NAT。

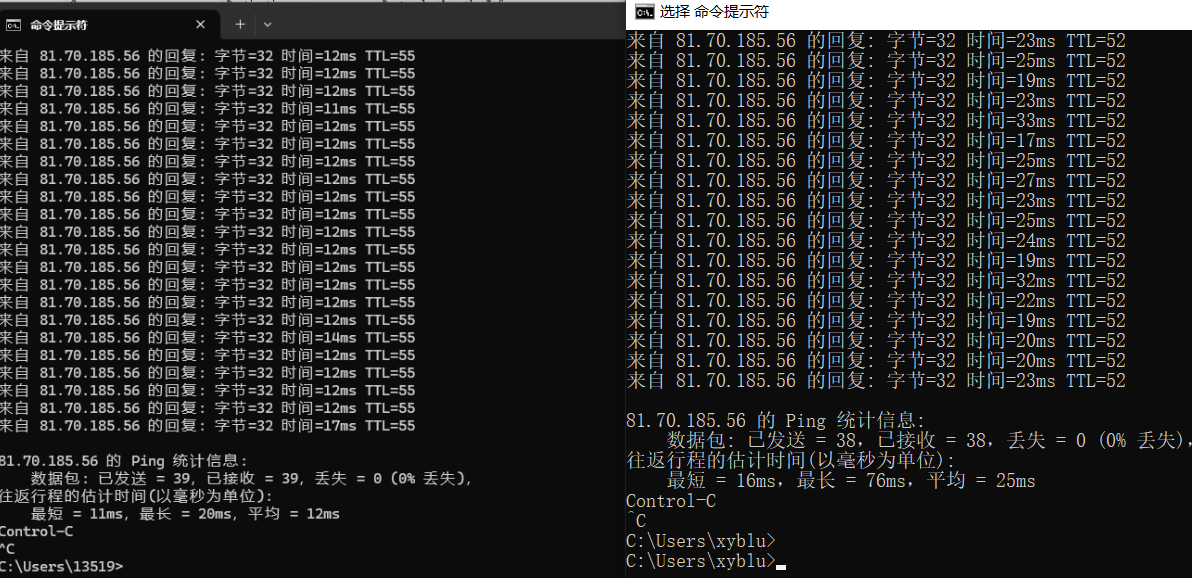


## 5.2 实时性和稳定性测试

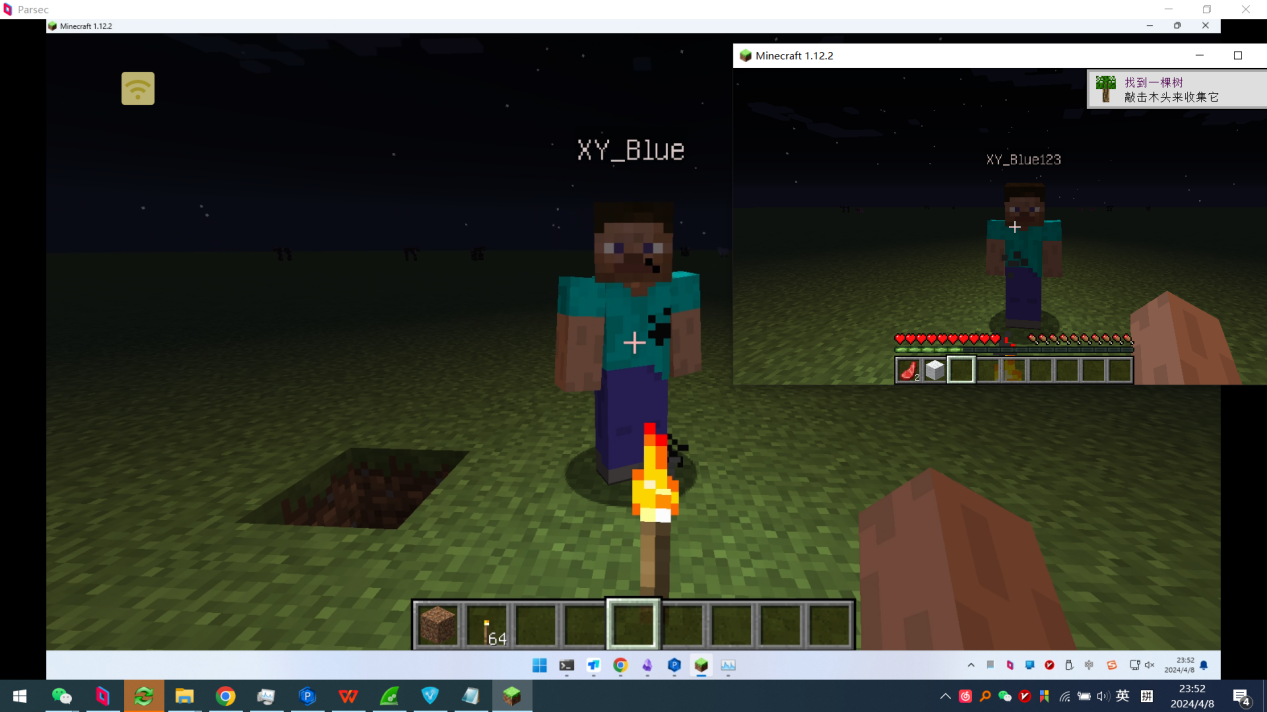


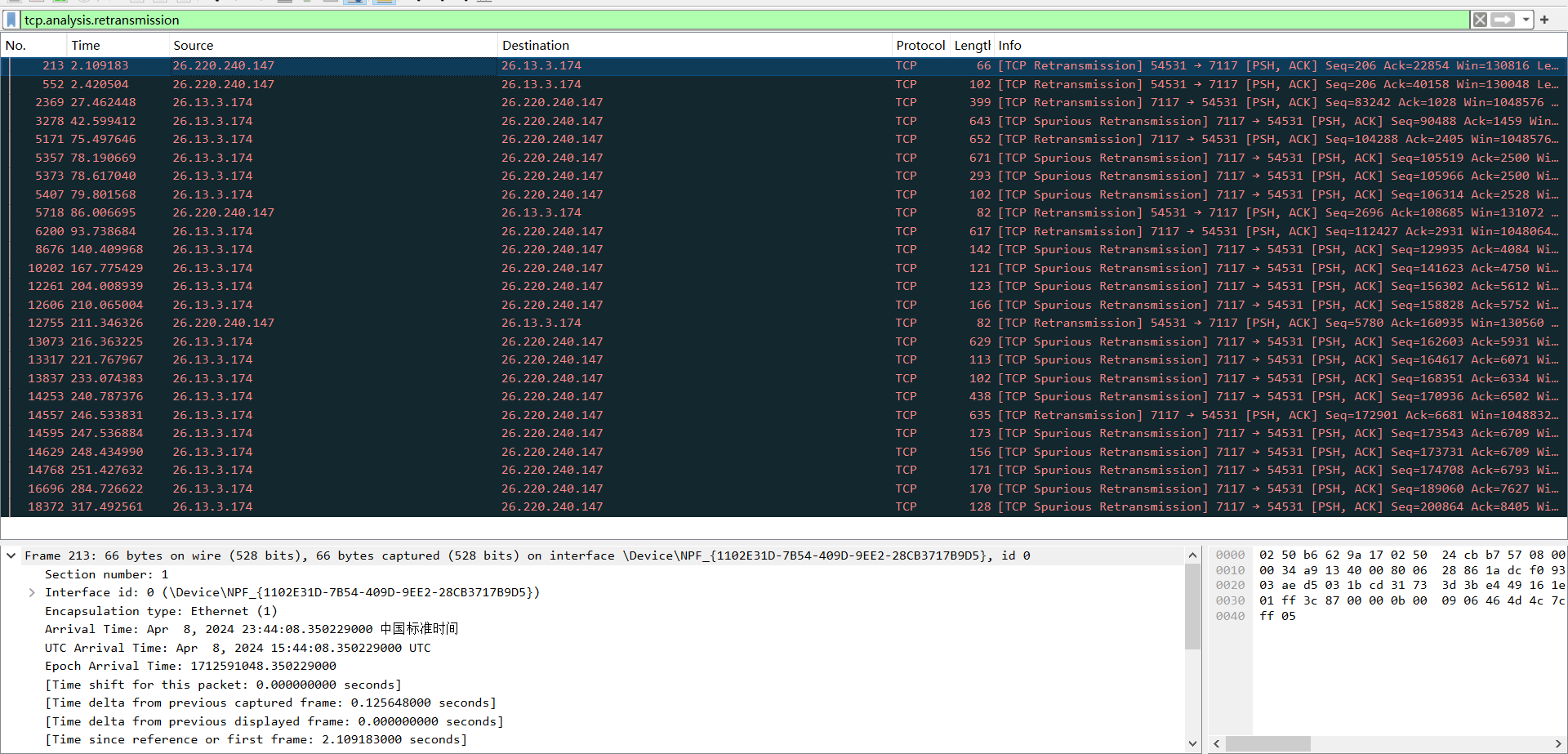
## 5.2.1 性能评估指标

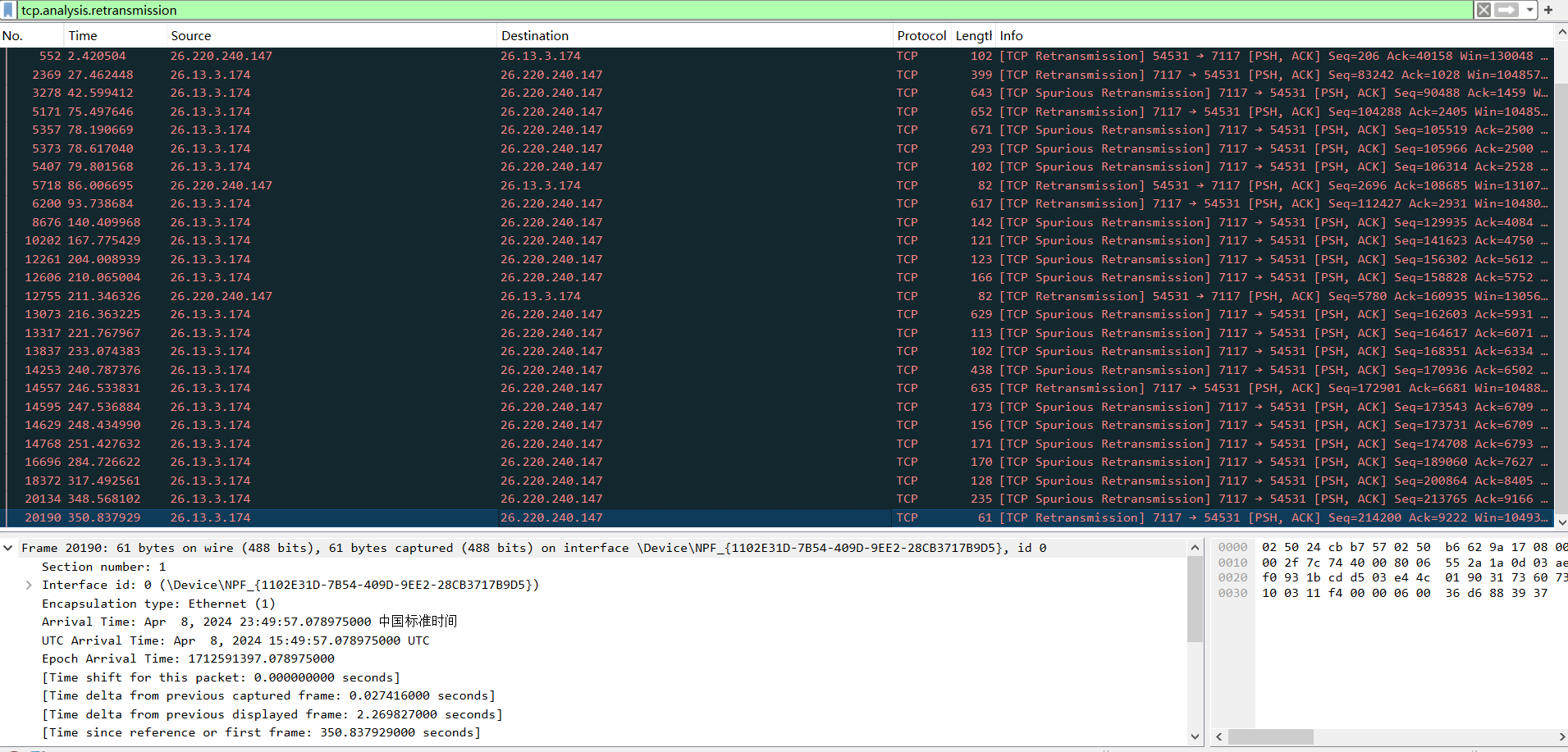
下图为一公网服务器（位于北京）的两客户端延迟分别为12ms和25ms。而如果使用内网穿透进行联机的话，延迟为12+25=37.5ms，延迟要高于p2p打洞的效果。



进行5分的持续建立游戏连接的丢包率测试，是通过Udp打洞的，但是游戏建立是基于TCP的。







我们使用命令tcp.analysis.retransmission 筛选出重传的包，可以看到5分钟一共有25个左右的TCP连接问题，这些的引入会造成网络拥塞：网络拥塞是指网络中的流量过大，导致数据包在传输过程中出现延迟、丢失或重传。当网络拥塞时，TCP Retransmission会增加，因为数据包需要在拥塞的网络中重新发送。网络延迟：高延迟的网络连接可能导致数据包在传输过程中的延迟。如果延迟超过了TCP的重传超时时间，发送端会触发TCP Retransmission。丢包：在网络中，数据包可能会由于各种原因而丢失，例如网络故障、设备故障或不稳定的无线连接。如果接收端没有收到发送端发送的数据包，发送端会进行重传。网络错误或故障：网络中的错误或故障，如损坏的网络设备、错误的网络配置或防火墙问题，可能导致无法访问或导致TCP Retransmission增加。

六、资源消耗评估

6.1 网络资源消耗评估

6.1.1 带宽、CPU和内存占用分析

峰值大概在4-5Mbps之间，并不算特别快。但是一个人应该是够用的,其速率也和协议，路由器配置有关。





对于cpu和内存来说，由于双方均承担了一部分网络压力，对比内网穿透来说，可以更为直观的反应两者的关系，但是对于游戏来说，例如存档等信息，只能存储在一个客户端上，也就是说不管使用什么方式来进行开服，作为服务端的客户端都要承受更为大的cpu压力和内存压力。因为其他客户端，读取存档会从开服者的存档中读取的，其中要经过网络-硬盘-内存-cpu的过程。

6.2 设备性能评估

6.2.1 实验设备性能影响分析

分析系统对客户端与服务器设备性能的影响，如对硬件配置的需求、对设备散热、电池续航等方面的影响，为设备选择与优化提供参考。

七、验证与可行性分析

7.1 P2P游戏联机功能原型实现

7.1.1 原型系统功能介绍

详细介绍原型系统的功能实现，包括核心模块的实现细节、接口定义、配置参数等。

7.1.2 实际测试与验证流程

描述原型系统的测试流程，包括测试环境设置、测试用例设计、测试执行记录、问题定位与修复等环节。

7.2 可行性评估

7.2.1 性能、安全性和实用性分析

如果我们的电脑并不需要 UPnP 所提供的功能，比如，我们的电脑并不想要对外直接提供服务，也不运行上述 P2P 软件，那么我们就无需开启 UPnP。因为一旦开启 UPnP，就意味着我们把自己的主机暴露在公网环境中，任何主机都可以向我们的电脑发起连接，NAT 设备会对所有收到的数据包不进行任何 authentication 认证而转发给我们的主机，这样，路由防火墙就会完全失效，我们的主机就很容易受到恶意的网络窥探，感染病毒或者恶意程序的几率也大大增加。

八、总结与展望

8.1 研究成果总结

8.2 研究局限与展望

指出研究存在的局限性，如未解决的问题、潜在改进空间等，并对未来研究方向与可能的应用前景进行展望。

附录

包含实验数据、详细测试报告、源代码片段、相关图表等补充材料。

按照上述提纲撰写论文初稿时，需确保每一部分的内容逻辑清晰、论据充分、数据准确，引用相关文献支持观点，并遵循学术写作规范。在撰写过程中，可根据实际情况对提纲进行微调，以确保论文结构完整、论述严谨。