doi 10.3969/j.issn.1671-1122.2014.03.007

基于 P2P 的远程控制系统设计与实现

姚洪波[1]，孟正[2]，李希[2]，文伟平[2]
（1. 中国石油大学计算机与通信工程学院，山东青岛 266580；2. 北京大学软件与微电子学院，北京 102600）

摘 要：近年来，NAT（网络地址转换）技术在很大程度上缓解了 IP 地址资源短缺的问题，但却给互联网上主机之间的通信带来了障碍，使得传统远程控制技术受到限制。文章围绕 P2P 穿越 NAT 设备并实现远程控制的若干问题进行研究，首先分析了当前主流的 NAT 穿越技术，并以此为基础，提出一套远程控制方案，然后使用 JAVA 语言对该方案予以实现。实验结果表明，文章提出的方案能够有效实现远程控制的功能。

关键词：P2P; 远程控制；NAT 穿越
中图分类号：TP309 文献标识码：A 文章编号：1671-1122（2014）03-0032-07

Design and Implementation on Remote Control System based on P2P Network

YAO Hong-Bo[1], MENG Zheng[2], LI Xi[2], WEN Wei-ping[2]
(1.*School of Computer & Communication Engineering, China University of Petroleum, Qingdao ShanDong* 266580*, China*;2.*Department of Information Security, SSM, Peking University, Beijing* 102600*, China*)

Abstract: Recently, NAT(network address translation) technology has largely alleviated the shortage of IP resource problems, but creates a communication barrier between the hosts on the Internet, which limits the traditional remote control technology. This paper studies a number of issues focusing on P2P(peer to peer) traversing NAT equipment and firstly analyses the mainframe NAT traversal technology. On this basis, the remote control program is proposed and the program is achieved by JAVA language. The experimental results show that the program proposed in this paper can achieve remote control effectively.

Key words: peer-to-peer; remote control; communication across Network Address Translators

0 引言

传统的远程控制系统通常具有以下不足：1）维护点较多；2）维护点区域覆盖较广；3）不同用户的计算机操作水平存在很大差距，部分问题仅仅依靠电话沟通是不够的，而去现场指导则费时费力；4）传统的远程控制软件使用中有局限性，如 pcAnyhere 要求被控端拥有公网 IP 地址，这在现实中很难做到，而 QQ 远程控制软件要依附 Internet 才能运行，限制了使用场景[1]。基于 P2P 技术的远程控制系统具有带宽要求低、不需要公有 IP 地址和不依赖于 Internet 的特性，相对于传统远程控制系统，优势明显[2]。

本文对 P2P 穿越 NAT 设备的若干问题进行研究，以当前主流的 NAT 穿越技术为基础，提出了一套远程控制方案，并使用 JAVA 语言对该系统予以实现。

1 相关理论

1.1 IP协议

因特网协议（Internet Protocol,IP）是为了网络互联而设计的。一个 IP 数据报由一个头部和一个正文部分构成。头部包含一个 20 字节的固定长度部分和一个可选任意长度部分，其中包括以下信息：版本（version）、报头长度（header length）、

收稿日期：2013-08-18
基金项目：国家自然科学基金 [61170282]
作者简介：姚洪波（1976-），女，山东，讲师，主要研究方向：软件工程；孟正（1990－），男，河北，硕士研究生，主要研究方向：漏洞分析和漏洞挖掘；李希（1985－），男，江苏，硕士研究生，主要研究方向：软件工程；文伟平（1976－），男，湖南，副教授，博士，主要研究方向：网络攻击与防范、恶意代码研究、信息系统逆向工程和可信计算技术等。

服务类型（type of server）、总长度（total length）、标识符（identifier）、标记字段（flag）、分段偏移（fragment offset）、生命期（TTL）、协议（protocol）、头部检验和（header checksum）以及源地址（source address）和目的地址（destination address）。

1.2 UDP协议

用户数据报协议（User Data Protocol，UDP）是无连接的传输协议。它封装了一个原始IP数据报，发送时无需建立连接。一个UDP数据段包括一个8字节的头部和数据部分，头部包括源端口号（source port number）、目的端口号（destination port number）、UDP长度（UDP length）以及UDP校验和（UDP checksum）。

1.3 TCP协议

传输控制协议（Transmission Control Protocol，TCP）是专门设计用于在不可靠的因特网上提供可靠的、端到端的字节流通信的协议。因特网不同于一个单独的网络，不同的部分可能具有不同的拓扑结构、带宽、延迟、分组大小以及其他特性。TCP被设计成能动态满足互联网的要求，并且能面对多种出错。TCP数据段均以固定的20字节的头部开始。固定的头部后面可能是一些可选项。可选项后面最多（如果存在）有65535-20-20=65495字节数据，第一个20指20字节的IP头部，第二个20指20字节的TCP头部。不带任何数据的数据段也是合法的，一般用于确认报文和控制报文。

TCP的头部包括以下信息：源端口号（source port number）、目的端口号（destination port number）、顺序号（sequence number）、确认号（acknowledgement number）、头长（header length）、URG（urgent pointer field significant）、ACK（acknowledgement field significant）、PSH（push function）、RST（reset the connection）、SYN（synchronize sequence numbers）、FIN（no more data from sender）、窗口（window）、校验和（checksum）、紧急指针（urgent pointer）以及选项（options）。

1.4 NAT

NAT（Network Address Translation）是IETF标准，它通过将局域网内的主机IP地址映射为Internet上有效的公网IP地址实现IP地址的复用。由于UDP协议具有无状态的特性，当前对其予以实现的NAT可分为Symmetric NAT、Restricted Cone NAT、Full Cone NAT和Port Restricted Cone NAT四种[3,4]。

1.4.1 Full Cone NAT

Full Cone NAT也被称为一对一的NAT。这种类型的NAT设备一旦建立了内部地址（LocalIP: LocalPort）和公网地址（PublicIP:PublicPort）的映射关系，以后用这个内部地址（LocalIP:LocalPort）向外面的任何主机发送数据都将使用这个公网地址（PublicIP:PublicPort）。在知道这个公网地址的前提下，任何外部主机（HostIP:AnyPort）都可以发送数据给公网地址（PublicIP:PublicPort），此时，内网的主机就可以收到这个数据包。

1.4.2 Restricted Cone NAT

它是Full Cone NAT的受限版本。一旦建立了内部地址（LocalIP:LocalPort）和公网地址（PublicIP:PublicPort）的映射关系，以后这个内部地址（LocalIP:LocalPort）都将使用公网地址（PublicIP:PublicPort）向外面的任何主机发送数据。借助于公网地址（PublicIP:PublicPort），只要内部地址（LocalIP:LocalPort）向外部主机（HostIP:AnyPort）发送数据包，外部主机（HostIP:AnyPort）就能够向公网地址（PublicIP:PublicPort）发送数据，内网的主机就可以收到这个数据包。

1.4.3 Port Restricted Cone NAT

它是在Restricted Cone NAT的基础上对端口做了进一步的限制。一旦建立了内部地址（LocalIP:LocalPort）和公网地址（PublicIP:PublicPort）的映射关系，这个内部地址（LocalIP:LocalPort）将使用公网地址（PublicIP:PublicPort）向外面的任何主机发送数据。这时外部主机（HostIP:HostPort）不能通过公网地址（PublicIP:PublicPort）和内部地址（LocalIP:LocalPort）进行通信，除非内部地址（LocalIP:LocalPort）的主机先前已经向这个外部主机（HostIP:HostPort）发送过数据了。

1.4.4 Symmetric NAT

当内部地址（LocalIP:LocalPort）第一次向外部主机1发送数据时，NAT为其提供一个映射（PublicIP-1，Port-1），之后内网主机向外部主机1发送的全部数据都是用（PublicIP-1，Port-1）。此时，如果内网主机（LocalIP:LocalPort）又发送数据给外部主机2，第一次发

送时，NAT 将分配一个（PublicIP-2，Port-2），之后内网主机向外部主机 2 发送的全部数据都是通过这个(PublicIP-2，Port-2)。如果 NAT 拥有多个公网 IP，那么 PublicIP-1 和 PublicIP-2 可能是不同的 ；如果 NAT 仅仅拥有一个公网 IP，则 Port-1 与 Port-2 必然是不同的，也就是说 PublicIP-1 等于 PublicIP-2 且 Port-1 等于 Port-2 一定是不成立的。另外，假如某一个外部主机想要向内网主机发送数据，那么它应该收到来自内网主机的数据，接下来才能向回发送，否则即使内网主机的（Public IP，Port）是已知的，外部主机也不能向内网主机发送数据。

1.5 远程控制技术

远程控制是指一台终端（通常是一个具有输入输出功能的设备）通过网络（包括互联网、局域网、电话系统、无线通信系统、通信卫星等）连接到另一终端，在本地终端操作远程终端的一项技术。

远程控制有很大的优越性，它可以跨越地理位置的限制，实现网络管理、网络教学等应用。它可以将指令传送到一些人类无法到达的现场，如进行深海研究、地下救援等。远程控制系统一般可以划分成三大部分 ：控制端系统、被控制端系统以及数据传输系统 [5,6]，如图 1 所示。各个部分协同工作，最终实现远程控制设备的目标。

图1 远程控制系统模型

控制端系统可以对设备进行远端监测和控制，它主要包括控制命令和参数的输入、终端显示远程设备状态、对状态数据和命令参数进行必要的处理，以及其他操作。

数据传输系统包括应用软件、传输协议以及网际设备（如路由器等）。作为远程控制的信息传输通道，数据传输系统负责对各类管理数据、控制数据以及状态数据进行传输。传输系统主要有两个目的，一是尽快将现场的设备状态信息传送到监控端，从而使操作人员对现场设备的工作状态有充分了解，以决定接下来的控制措施 ；二是向现场控制主机传输监控端的控制信息，从而对设备进行控制。

被控制端通过远程监控终端的控制数据完成对设备的控制，同时对设备状态进行监测，并进行分析，然后再通过传输通道将这些状态反馈到远程监控端。

2 NAT 穿越技术

NAT 技术 [7] 通过将局域网内的主机 IP 地址映射为 Internet上有效的公网 IP 地址，从而实现了网络地址的复用。NAT 技术虽然能够有效缓解 IP 地址匮乏的现状，却为互联网上主机之间的通信带来障碍。NAT 之所以阻碍主机之间进行通信，是因为它不允许外网主机对内网主机进行主动访问，这就造成了互联网上具有公网 IP 地址的主机不能对 NAT 之后的主机进行主动访问，而处于不同 NAT 之后的主机之间则更是不能直接通信。因此，为了在当前网络环境下进行通信，我们必须研究穿越 NAT 的方案。NAT 穿越技术是建立在互联网协议基础之上的，它能够穿越 NAT 网关并连接两个终端。

2.1 中继

中继是在当前 NAT 设备坏境中最可靠但也是效率最低的建立 P2P 通信的一种方法。利用中继实现 P2P 通信就是把 P2P 通信网络看作是通过中继的 C/S 通信。当两个客户端都在服务器上连接时，这种方式就是有效的，这也是中继的优势。位于中间的 NAT 设备不需要是 EIM-NAT。中继的缺点是需要服务器处理并占用带宽，即使服务器拥有的 I/O 带宽是足够的，客户端之间的通信也会产生延迟。TURN[8] 实现了一种相对安全的中继方法。虽然中继方式自身存在缺陷，但在对系统进行设计时应将其作为最基本的连接方式。当其他方式都无效时，中继仍然能够对最基本的通信予以保证。而当其他方式都有效时，中继也可以作为一种并用的连接方法，从而使传输速度有所增加。

2.2 逆向连接

逆向连接技术适用于一个客户拥有公网 IP 地址，另一个客户位于 NAT 之后的情况。当前各种 P2P 程序都实现了这种技术。该技术的最大限制在于只能允许连接双方中的一方在 NAT 后面。如果 NAT 设备是 EIM-NAT，拥有公网 IP 地址的客户端就可以通过服务器来确定是否可以接收来自另一客户端的连接。假如 NAT 设备不是 EIM-NAT，拥有公网 IP 的客户端就难以判断哪些 endpoint（终端）能够接收由客户端发起的连接。

当两个 endpoint 都处于 NAT 后面时，这种逆向连接技术就会失败。因此，逆向连接技术对于解决 P2P 连接问题并不是通用的。当“ 前向 ”或“ 逆向 ”都不可以建立连接时，程序只能使用中继的机制。

2.3 UDP挖洞

UDP 挖洞技术对 EIM-NAT 的一些特性形成依赖，允许 P2P 程序在 NAT 设备上“ 挖洞 ”，从而建立双方的连接，即使两个客户端同时处于 NAT 之后这种方法也是适用的。当客户端所处的 NAT 设备不是 EIM-NAT 时，对等客户端就难以确定和哪个已经映射的 endpoint 发起连接。

下面对两个特定的场景分别予以考虑 ：第一个场景代表最常见的情况，两个客户端分别位于不同的 NAT 设备后面，它们直接建立 P2P 通信。第二个场景是两个客户端位于同一个 NAT 后面，但它们自身并不知道。

2.3.1 位于不同的NAT后面

客户端 A 和 B 分别处于各自的局域网中，它们处于不同的 NAT 设备之后，并拥有各自的私有 IP 地址。服务器 S 拥有公网 IP 地址，它是可以寻址的，能够用来注册、发现和有限制地中继。NAT 之后的客户端通过服务器 S 来注册他们的公共 endpoint。对等客户端通过服务器 S 来发现 NAT 后面的客户端的公有 endpoint。和中继不一样，服务器 S 只用来向客户端传递连接初始控制信息，而不是直接发送端到端的信息。

UDP 挖洞技术有很多特性是有用的。当两个位于 NAT 设备后面的对等客户端建立了 P2P 的 UDP 连接，连接的双方都可以充当“ 中介 ”为其他对等客户端建立 P2P 连接提供帮助，从而减少服务器 S 的负载，程序也就不需要检测它前面的 NAT 设备的类型。对于上述过程，当任一个客户端或两个客户端都不位于 NAT 设备之后时，也可以创建 P2P 连接。在多级 NAT 情况下，UDP 挖洞技术也可以自动地工作。在这种情况下，当一个或两个客户端到达公有 Internet 网络时，它们经过了两级或更多的 NAT 设备。

2.3.2 位于同一个NAT后面

假设客户端 A 和 B 处于同一个局域网中，他们处在同一个 EMI-NAT 后面，拥有相同的地址空间，但它们自己并不知道。服务器 S 拥有公网 IP 地址（可寻址），可以用来注册、发现和有限制地中继。通过服务器 S，对等客户端可以发现 NAT 之后的其他对等客户端。和位于不同的 NAT 后面的情况一样，客户端仅使用服务器 S 传递连接初始控制信息，而不是直接使用服务器 S 发送端到端的信息。

当客户端 A 和 B 通过服务器 S 初始化注册 endpoint 时，服务器 S 保存客户端的私有 endpoint 以及它所看到的客户端的公有 endpoint。客户端使用第一个连接成功的地址发送数据包至它所得知的对方的任一个 endpoint。假如两个客户端位于同一个 NAT 的后面，那么直接发送给客户端私有 endpoint 的数据包有可能较早到达，从而实现不需要 NAT 参与的直接通信。

2.4 TCP挖洞

2.4.1 套接字和TCP端口复用

为了能让 TCP 挖洞正常工作，需要使用一个单独的本地的 TCP 端口来监听向内的 TCP 连接，并且同时初始化复用这个端口的外向 TCP 连接。所有的主流操作系统都对一个特殊的 TCP 套接字选项提供支持，即 SO_REUSEADDR，它允许应用程序把多个套接字绑定到本地相同的 endpoint 上，我们需要为所有涉及到的套接字设置该选项。通过引进 SO_REUSEPORT，BSD 系统将端口复用和地址复用进行分离。

2.4.2 建立P2P的TCP连接流程

建立 P2P 的 TCP 连接主要有以下 5 个步骤 ：1）客户端 A 通过与服务器 S 的 TCP 连接向 S 发出连接客户端 B 的请求 ；2）服务器 S 将客户端 A 的私有和公有 endpoint 信息发给客户端 B ；3）使用在服务器 S 上注册的 TCP 端口，客户端 A 和 B 分别异步向从服务器 S 那里获得的对方私有和公有 endpoint 发起外连请求，同时在本地 TCP 端口上监听向内的连接 ；4）客户端 A 和 B 等待外连请求确认信号以及外部传来的连接请求。当一方的外连请求由于客户端无法到达、连接重置等网络故障失败时，主机将于短时间内进行重试，直至达到预先设定的连接超时时间 ；5）当 TCP 连接建立时，主机将验证连接进入的另一主机的身份，假如验证失败，客户端就会关闭连接，并等待来自其他客户的连接。

2.4.3 TCP同时打开

假如 NAT 设备认为一个 SYN 数据包的目的 endpoint 和源 endpoint 与一个活动的 TCP 连接有关，那么当这个

SYN 数据包到达 NAT 设备时，NAT 设备将允许这个包通过。如果一个 NAT 设备通过相同的端口号和地址转换了一个向外的连接，那么它会认为这个会话是活动的，并且允许向内的 SYN 包通过。假如客户端 A 和 B 各自发起向外的 TCP 连接，其中另一方超时了，那么在到达对方的 NAT 设备之前，各个客户端的 SYN 包都会通过自己本地的 NAT 设备，此时一个 TCP 连接就会建立起来。

大多数 NAT 设备（超过 50%）都对 Endpoint-Independent 映射提供支持，且不发送 RST 包或 ICMP 错误对未获同意的向内连接的 SYN 包予以响应。对于大多数的 TCP 连接尝试，在穿越 NAT 设备时，TCP 同时打开技术并不会有效。

2.5 UDP端口预测

当一些实现 Endpoint-Dependent 映射的 NAT 存在时，UDP 挖洞将会失败，然而，仍然有一种变种的 UDP 挖洞技术能够对 P2P 的 UDP 连接进行创建。这种方法称为“ N+1 ”技术，“ Symmetric NAT Traversal using STUN ”[3] 对这种方法进行了详细讨论。这种方法对 NAT 的行为进行分析并对在将来的会话中给客户端分配的公网端口号进行预测。一般来说，分配的公网端口号是可预测的，因为大多数 NAT 映射端口是按顺序来分配的。

如果要让应用程序在即使有一个 NAT 使用了 Endpoint-Independent 时仍能够运行，程序就需要预先知道每一个端点所涉及的 NAT 类型，并且对它们的行为进行相应的修改，这就提升了算法的复杂度，也增大了网络的脆弱性。当每一个客户端都处在两级或者更多级的 NAT 设备之后时，端口预测方法很少能够工作。

2.6 TCP端口预测

为了创建穿越 Address-Dependent 映射 NAT 的 P2P 的 TCP 直接会话，有一种“ TCP 挖洞技术 ”的变种方法可以采用。然而这种方法对时序更加敏感，比之前介绍的 UDP 端口预测方法更为脆弱。这种方法预测到的 NAT 分配端口的正确性难以保证。如果任何一个客户端的 SYN 包到达对方 NAT 设备的速度太快，那么远端的 NAT 设备就可能发送一个 RST 包或丢掉一个 SYN 包，导致本地的 NAT 设备对新的会话予以关闭，并通过相同的端口继续进行失败的 SYN 重传。

3 系统设计

3.1 穿越NAT设备

在穿越 NAT 设备时，UDP 挖洞技术具有良好的稳定性，并且被当前很多 NAT 设备所支持，因此本文使用 UDP 挖洞技术实现一个简易的 P2P 通信示例。程序分为两部分 ：客户端和服务端。服务端程序运行在具有全球唯一可寻址的公网 IP 地址的主机上，可以用来注册、发现和中继。客户端程序则运行在处于 NAT 后面等待 P2P 连接的主机上。

3.1.1 服务器端程序执行流程

服务器端程序负责为每一个登录的客户端分配唯一的 ID，同时维护着一个保存了当前所有在线客户端信息的列表，包括客户端的 ID、私有和公有 endpoint 等。程序主体执行一个 while(true) 循环，不断接受来自客户端的各类消息并做出响应。执行流程如图 2 所示。

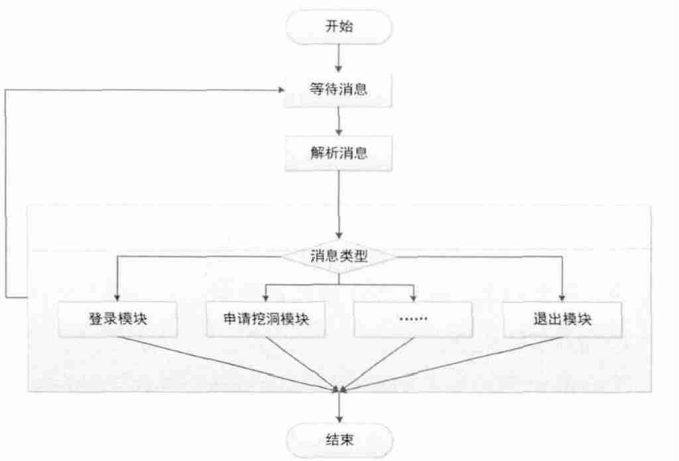

图2 服务器端程序执行流程

具体步骤描述如下：

1) 登录。当服务器端监听到登录请求时，服务器端为客户端分配一个 ID，将客户端发送的私有 endpoint 信息和自己观察到的客户端的公有 endpoint 信息保存起来，并将分配的 ID 和公有 endpoint 信息回传给客户端。

2) 退出。当服务器端监听到退出请求时，服务器将该客户端信息从维护的用户列表中删除。

3) 查询所有客户端。当服务器端监听到查询所有客户端请求时，服务器将维护的用户列表发给申请方。

4) 挖洞预处理。当服务器端监听到挖洞预处理请求时，服务器从申请方那里获知打洞的目的地址 ID，然后遍历用

户列表，将目的地的 endpoint 信息发送给申请方。

5）挖洞。当服务器端监听到挖洞请求时，服务器向打洞的目的客户端发送一个命令。

3.1.2 客户端程序执行流程

客户端要同时监听服务器发来的消息和用户的输入，而等待用户输入和接受服务器发来的消息都会造成客户端程序阻塞，所以客户端创建了两个线程分别监听消息和用户输入。客户端程序的执行流程如图 3 所示。

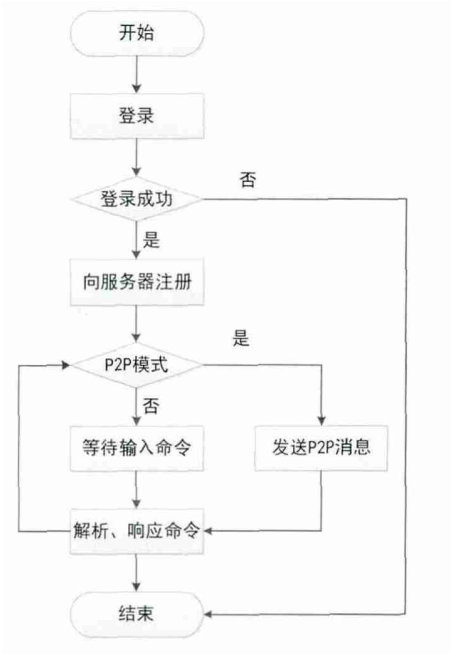

图3 客户端程序执行流程

具体步骤描述如下：

1）登录。登录模块要求用户输入用户名和密码，对客户端进行身份认证。这里并没有真正实现动态的验证，只是采用硬编码的方式模拟登录的过程。程序只是简单地比较用户输入的用户名和密码字符串是否和程序预设的字符串相同。如果相同，客户端是合法的，进入下一个流程；如果不合法，程序则直接退出。客户端通过认证后就和服务器建立连接，将从服务器那里获得的 ID 和服务器观察到的自己的公有 endpoint 信息保存到本地。

2）退出。当客户端退出时，客户端向服务器端发送一个消息，告诉服务器端自己要退出了，通知服务器更新用户列表。同时将线程标志设置为 false，让线程停止，从而退出程序。

3）显示自己。客户端维护了一个对象，包含自己的相关信息。当客户端需要查看自己的相关信息时，只需将其做简单的输出即可。

4）显示所有客户端。客户端向服务器端发送请求信息，并等待接受服务器发来的用户列表信息，然后遍历用户列表，将它显示出来。

5）挖掘预处理。客户端向服务器端发送预处理请求，并接受服务器发来的消息，将消息在本地予以保存，以备在发送 P2P 消息时使用。

6）挖洞。客户端向服务器端发送请求，同时向先前从服务器那里得到的打洞目的方发送一个数据包，并将自己的模式改成 P2P 模式。

7）接到挖洞请求。当客户端被动地接受服务器端发来的挖洞命令时，客户端向申请打洞方发送数据包，同时将自己的通信模式设置成 P2P 模式。

3.2 远程控制

按照传输协议的不同，远程控制可分为 UDP 和 TCP 两种实现方式。如果能够顺利进行 TCP 挖洞程序，则优先使用 TCP 协议，只有当 TCP 挖洞不能正常工作时，才使用 UDP 方式实现远程控制。

3.2.1 被控端

被控端在不需要人工干涉的情况下完成自动启动，等待控制端发来的命令并做出响应，同时将状态信息反馈给控制端，如图 4 所示。

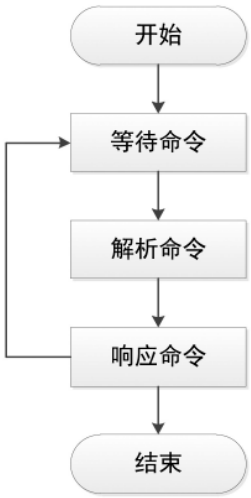

图4 被控端执行流程

3.2.2 控制端

控制端不需要自动启动，它等待用户的命令，将用户命令通过网络发送给被控端，接受被控端回传过来的反馈信息，如图 5 所示。同样这里使用多线程技术来解决阻塞问题。

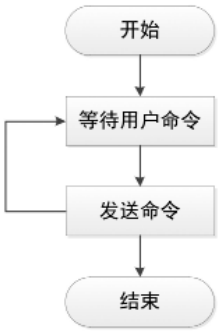

图5 控制端执行流程

4 实验结果测试

为了验证方案的可行性，我们让一个拥有公网IP（61.177.234.219）的主机运行服务器端程序，客户端A和B处于NAT A之后，客户端C处于NAT B之后。服务器运行Windows 2003 Server操作系统，客户端A、B和C分别运行Windows 7、Windows XP和Windows 7操作系统。测试拓扑图如图6所示。

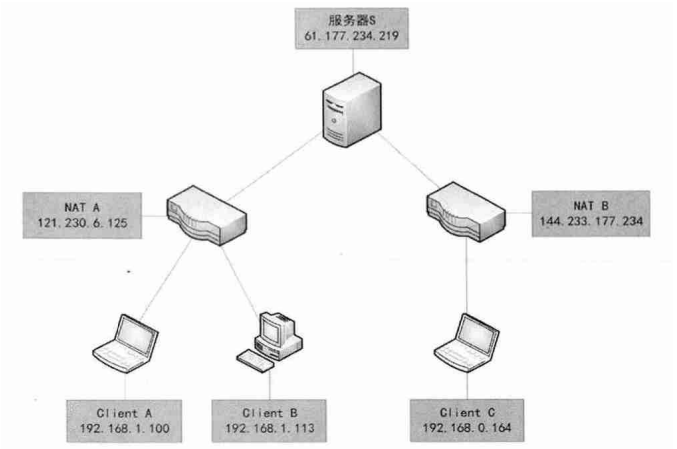

图6 测试拓扑图

4.1 客户端位于不同NAT之后

首先测试客户端处于不同NAT之后的情形，在服务器上运行服务器端程序，客户端A和C运行客户端程序。客户端A向服务器发送登录请求，通过认证后服务器为客户端A产生一个ID（10001），将包含ID和公有endpoint的信息发给A，A收到后将其显示出来，等待用户命令。同理，客户端C也登录到服务器端，服务器为其生成ID（10002）并将自己观察到的C的公有endpoint信息发送给客户端C。当客户端C输入whoami后，客户端将自己维护的ID和endpoint信息显示出来。接着客户端想要查看所有在线客户端的情况。客户端A向服务器发送showall命令，服务器端收到命令后将所有在线用户列表发送给A，A将其显示出来。这时客户端A想要和客户端C建立P2P连接，客户端A就向服务器发送preparpunching命令，告诉服务器它想和谁建立P2P连接，服务器会将对方的信息发送给A，A就可以正式发起挖洞请求了。A向C发送一个数据包，同时服务器通知客户端C，让C也向A发送数据包，这时P2P连接就建立好了。

4.2 客户端位于相同NAT之后

假设客户端X1和X2位于同一个NAT之后，NAT为X2分配了外连的endpoint，如X2':x2'。当X1向X2':x2'发送数据时，它就必须通过NAT中继来达到和X2通信的目的，这种方式称为harpinning。假设客户端A和B都位于NAT A后面，我们来看看hairpinning是否能够奏效。登录到服务器后，由A向B申请挖洞，挖洞成功后，客户端A向B发送消息，此时B能够收到消息；B向A发送消息，A也能收到消息。这就说明hairpinning已经工作了。

5 结束语

本文首先介绍了P2P穿越NAT技术和远程控制系统概况，接着给出了P2P穿越NAT设备及远程控制的实现方法，最后使用可移植的JAVA程序设计语言对UDP挖洞进行了实现，并对结果进行了分析。（责编 马珂）

参考文献
[1] 康治平，向宏，傅鹂等．基于API HOOK技术的特洛伊木马攻防研究[J]. 信息安全与通信保密，2007，(02)：145.
[2] Ting Liu, Xiaohong Guan, Qinghua Zheng, Ke Lu, Yuanfeng Song, Weizhan Zhang. PrototypeDemonstration: Trojan Detection and Defense System[C].In: Consumer Communications and NetworkingConference, 2009. CCNC 2009:1-2.
[3] J. Rosenberg, J. Weinberger, C. Huitema, R. Mahy, STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)[R].Network Working Group, RFC3489, March 2003.
[4] J. Rosenberg, R. Mahy, P. Matthews, D. Wing,Session Traversal Utilities for NAT (STUN) [R]. Network Working Group, RFC5389, October 2008.
[5] 陈克斌，基于因特网远程控制系统的研究与设计[D]. 西安：西北工业大学，2006.
[6] 焦延杰，网络远程访问与控制系统的设计和实现[D]. 北京：北京交通大学，2009.
[7] B. Ford, P. Srisuresh, D. Kegel, Peer-to-Peer (P2P) communication across middleboxes[EB/OL].http://midcom-p2p.sourceforge.net/draft-ford-midcom-p2p-01.txt, October 2003.
[8] P. Matthews, J. Rosenberg, J. Rosenberg, Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN) [R]. Network Working Group, RFC5766, April 2010.