

学步园

现在的位置: [首页](#) > 综合 > 正文
[RSS](#)

NAT与NAT穿透(三)

2018年08月31日/综合/共 1766字/字号 [小](#) [中](#) [大](#)/评论关闭

nat 穿透原理

一直以来，说起NAT穿透，很多人都会被告知使用UDP打孔这个技术，基本上没有人会告诉你如何使用TCP协议去穿透（甚至有的人会直接告诉你TCP协议是无法实现穿透的）。但是，众所周知的是，UDP是一个无连接的数据报协议，使用它就必须自己维护收发数据包的完整性，这常常会大大增加程序的复杂度，而且一些程序由于某些原因，必须使用TCP协议，这样就常常令一些开发TCP网络程序的人员“谈穿透色变”。那么，使用TCP协议是不是就不能实现穿透呢？答案当然是否定的：TCP协议不仅能实现NAT穿透，而且实现起来比UDP穿透甚至还简单一些。

要了解如何使用TCP穿透NAT，就要首先看看如何使用UDP穿透NAT。

我们假设在两个不同的局域网后面分别有2台客户机A和 B，AB所在的局域网都分别通过一个路由器接入互联网。互联网上有一台服务器S。

现在AB是无法直接和对方发送信息的，AB都不知道对方在互联网上真正的IP和端口，AB所在的局域网的路由器只允许内部向外主动发送的信息通过。对于B直接发送给A的路由器的消息，路由会认为其“不被信任”而直接丢弃。



了解网络社区对消费者行为的影响

网络社区、虚拟社区、网络社区对消费者行为的影响、网络社区对消费者行为的影响、网络社区对消费者行为的影响。

人力资源系统

商标转让

PS 培训班

电商软件

免费 云服务器

VR公司



可视化工具

要实现 AB直接的通讯，就必须进行以下3步：A首先连接互联网上的服务器S并发送一条消息（对于UDP这种无连接的协议其实直接初始会话发送消息即可），这样S就获取了A在互联网上的实际终端（发送消息的IP和端口号）。接着 B也进行同样的步骤，S就知道了AB在互联网上的终端（这就是“打洞”）。接着S分别告诉A和B对方客户端在互联网上的实际终端，也即S告诉A客户B的会话终端，S告诉B客户A的会话终端。这样，在AB都知道了对方的实际终端之后，就可以直接通过实际终端发送消息了（因为先前双方都向外发送过消息，路由上已经有允许数据进出的消息通道）。

用UDP来实现以上3步不存在什么理论上的问题，因为UDP是无连接的协议，它允许socket进行“多对一”的通讯（即几个具有不同IP和端口号的socket向一个接收socket发送消息）。但是使用TCP就出现了问题：在一般情况下，TCP socket不允许在已经建立连接的端口上再进行监听和使用该本地端口。换句话说，当AB连接上服务器S后，S将AB的实际终端告诉对方，下一步本该是AB利用对方的实际终端进行直连，但这时你会发现对方的实际终端已经被占用了（就是各自连接到服务器S的会话占用了终端），无法同时listen和connect。于是很多人得出结论：TCP无法实现NAT穿透。

于是问题的关键变成了如何复用一個TCP连接的本地终端，这其实不是协议的问题，而是一个API的问题。幸运的是，所有主流操作系统都支持一个特定的TCP套接字选项——SO_REUSEADDR。这个选项允许将多个socket绑定到同一个本地终端。我们建立socket的时候只要加上这么一行：

```
setsockopt(socket, SOL_SOCKET, SO_REUSEADDR, &flag, len); //C++就这么做
```

```
_Client.SetSocketOption(SocketOptionLevel.Socket, SocketOptionName.ReuseAddress, True) '这是vb.net 更加简单
```

知道上面的知识就很好办了，下面我来说说TCP协议的穿透流程：

机器布局还是和上面使用UDP的一样。现在假设客户A想和客户B建立TCP连接。

首先还是 AB分别和服务端S分别建立连接，S记录AB的互联网实际终端。然后S分别向AB发送对方的实际终端。接着，从A和B向S连接时使用的端口，AB都异步调用connect函数连接对方的实际终端（就是S告诉的终端），同时，AB双方都在同一个本地端口监听到来的连接（也可以先监听，再connect更好）。由于双方都向对方发送了connect请求（假设各自的SYN封包已经穿过了自己的NAT），因此在对方connect请求到达本地的监听端口时，路由器会认为这个请求是刚刚那个connect会话的一部分，是已经被许可的，本地监听端口就会用SYN-ACK响应，同意连接。这样，TCP穿透NAT的点对点连接就成功了。

<http://blog.163.com/hanyinlong@126/blog/static/99751486201142114129838/>

[返回](#)

【上篇】 [TCP三次握手/四次挥手及其状态分析](#)

【下篇】 [NAT与NAT穿透\(一\)](#)

作者: [HiexiaTodeHot](#)

• 该日志由 HiexiaTodeHot 于1年前发表在综合分类下，最后更新于 2018年08月31日。

- 转载请注明: [NAT与NAT穿透\(三\)| 学步园 +复制链接](#)

-

抱歉!评论已关闭.

书签

- [176互娱](#)
- [aaa培训机构](#)
- [乐折网](#)
- [仟渔网](#)
- [代刷网](#)
- [传奇私服](#)
- [公司起名](#)
- [古茗奶茶加盟费多少](#)
- [名城](#)
- [广州房产网](#)
- [庆余年全集](#)
- [庆余年在线观看](#)
- [找商网](#)
- [招生](#)
- [探索网](#)
- [搜好货](#)
- [机器人加盟](#)
- [柚子视频](#)
- [浙江食品网](#)
- [白云飘飘网](#)
- [航班查询时刻表](#)
- [贷小秘贷款平台](#)
- [阿斯达](#)

-

-

-

- [青岛房产网](#)

最新文章

- [go语言映射\(map\)要点总结](#) [php的扩展写法总结](#) [详解GO语言，一文告诉你PHP程序员](#) [Go语言的9大优势和3大缺点](#) [Docker容器监控方案怎么选？看看](#) [百万级高并发mongodb集群性能数十](#) [从零开始学YII2框架（六）高级应](#) [图解Go里面的WaitGroup了解编程语](#) [关于go语言中gc的初步研究](#)

本站推荐

- [作业的提交和监控（二）](#)
- [作业的提交和监控（一）](#)
- [Boost – Function 分析](#)
- [奇技淫巧 – C/C++ 宏自身](#)
- [模板的 SFINAE 原则](#)
- [web前端](#)
- [数据库](#)
- [编程语言](#)
- [搜索技术](#)
- [关于本站](#)

[返回首页](#)

Copyright © 2013-2018 学步园 保留所有权利.

软文销售 QQ客服: 2517270004  QQ交谈 (其他合作也可洽谈)

-
-
-