# 内网穿透详解

目录

## 一、前言

阅读本文前需要先搞懂NAT、PAT、端口映射几个概念，前面我有写了一篇关于这几个概念的博文。根据之前的博文我们已经知道，内网宽带中的主机可以访问公网宽带主机，反之不可以访问；公网宽带主机可以和公网宽带主机双向访问；内网宽带中的主机和内网宽带中的主机互相无法访问。那么内网宽带中的客户机和公网宽带中的客户机如何访问另一个内网宽带中的服务器呢？这里就需要用到内网穿透技术。

## 二、应用场景

家里或公司是运营商内网ipv4宽带或公网IP在防火墙后，且需要在外访问家里或公司的电脑、NAS、树莓派、摄像头等网络设备或远程控制等情况下。

## 三、详细原理

### 3.1 传统内网穿透（服务器中转数据穿透）原理

对于在NAT之后的服务器来说，其不是不能主动访问公网端口，而是不能反过来有效的被公网访问。所以可以在中间架设一个公网服务器，让在NAT之后的服务器持续主动访问这个拥有公网IP地址的服务器，，这样内网服务器就成功与公网中转服务器建立了一个连接通道。然后当有任何其他NAT后的客户端主动连接公网中转服务器时，公网服务器接收到连接请求之后马上把这连接请求通过先前建立好的隧道转发到内网服务器，内网服务器将响应数据包再原路转发回去，最终到达公网中转服务器，然后返回给其他NAT后的客户端。

### 3.2 点对点穿透原理

在内网穿透传输大量数据时如果都经过服务器中转的话，这样会对服务器端带宽压力比较大。只要是数据量很大，而一般利用中转服务器又需要一定规模投入的应用，我们都可以考虑用P2P技术。
1.UDP打洞技术
最为常见的实现P2P的方式是采用UDP打洞技术，UDP打洞技术是通过中间服务器的协助在各自的NAT网关上建立相关的表项，使P2P连接的双方发送的报文能够直接穿透对方的NAT网关，从而实现P2P客户端互连。如果两台位于NAT设备后面的P2P客户端希望在自己的NAT网关上打个洞，那么他们需要一个协助者——集中服务器，并且还需要一种用于打洞的Session建立机制。
Session建立机制：
假定客户端A要发起对客户端B的直接连接，具体的“打洞”过程如下：
（1）A最初不知道如何向客户端B发起连接，于是A向集中服务器（本质上是一台被设置在公网上的服务器，建立P2P的双方都可以直接访问到这台服务器。位于NAT网关后面的客户端A和B都可以与一台已知的集中服务器建立连接，并通过这台集中服务器了解对方的信息并中转各自的信息）发送消息，请求集中服务器帮助建立与客户端B的UDP连接。
（2）集中服务器将含有B的外网和内网的地址二元组发给A，同时，集中服务器将包含有A的外网和内网的地址二元组信息的消息也发给B。这样一来， A与B就都知道对方外网和内网的地址二元组信息了。
（3）当A收到由集中服务器发来的包含B的外网和内网的地址二元组信息后，A开始向B的地址二元组发送UDP数据包，并且A会自动锁定第一个给出响应的B的地址二元组。同理，当B收到由集中服务器发来的A的外网和内网地址二元组信息后，也会开始向A的外网和内网的地址二元组发送UDP数据包，并且自动锁定第一个得到A回应的地址二元组。一旦A与B都向对方的NAT设备在外网上的地址二元组发送了数据包，就打开了A与B之间的“洞”，A与B向对方的外网地址发送数据，等效为向对方的客户端直接发送UDP数据包了。一旦应用程序确认已经可以通过往对方的外网地址发送数据包的方式让数据包到达NAT后面的目的应用程序，程序会自动停止继续发送用于“打洞”的数据包，转而开始真正的P2P数据传输。
当然，UDP转换协议提供的“洞”不是绝对可靠的，多数NAT设备内部都有一个UDP转换的空闲状态计时器，如果在一段时间内没有UDP数据通信，NAT设备会关掉由“打洞”过程打出来的“洞”。如果P2P应用程序希望“洞”的存活时间不受NAT网关的限制，就最好在穿越NAT以后设定一个穿越的有效期。
2.TCP打洞技术
从现在的主流应用的角度上来看，基于TCP的P2P应用显然不如基于UDP的应用那么广泛，但是也存在打洞的需求。TCP相对于UDP而言要复杂的多，TCP连接的建立要依赖于三次握手的交互，所以NAT网关在处理TCP连接的时候，需要更多的开销。但是，由于TCP协议完备的状态机机制，TCP反而比UDP更能精确的获取某个Session的生命期。
一种新的代理类型 XTCP 能解决这个问题，实现方式可以是采用搭建FRP服务器的方式，在传输数据的两端都部署上FRP客户端用于建立直接的连接。

### 3.3 延伸一个内网渗透知识

为了安全起见，通常会在网络中加入防火墙，防火墙有入站规则和出站规则。如果不是非常严格的安全管控，通常是不会设置出站规则的，但是入站规则一般都会设置的，比如说外部可以通过80端口传入内网的WEB服务器访问网页，但是不能通过3389端口登陆内网的远程桌面。而在内网渗透的过程中碰到这种情况，我们也可以借助上面内网传统的方式实现穿透防火墙的入站规则。因为防火墙通常只拦截了入站，没有拦截出站，那么我们可以让内网服务器主动出站（主动连接到黑客的服务器），与黑客自己的服务器打通隧道，最终绕过防火墙连上3389远程桌面。
还有一种情况就是我们已经拿下了内网其中一台并没有做任何防火墙规则的白名单服务器，但是我们想连上内网另一台做了入站规则的目标服务器，那么我们可以让这台白名单服务器作为一个跳板，让他先监听自身任意一个端口，然后在有任何用户连上这个端口之后，白名单服务器就主动连上内网的目标服务器，然后借助这台白名单服务器打通黑客和目标服务器的连接隧道。而在黑客工具中大名鼎鼎的lcx原理也就是如此，前者的实现是lcx的listen和slave命令，后者的实现是lcx的tran命令。

### 3.4 穿透原理总结

内网穿透传统方式会出现服务器和客户机之间的数据传输全部经过中转服务器，传输速度将受制于中转服务器的上下行带宽，不过稳定性很好，对于自己要购置的云主机要求就是大带宽，一般这种云主机按流量计费，传输的数据量越大价格自然越贵。所以点对点穿透便能解决流量带来的困扰，点对点可以实现服务器和客户机之间打洞直接进行数据通信，这种方式一般用于udp协议的传输，比如应用于远程NAS看视频听歌，但这种方式需要服务器和客户机都安装穿透工具，对用户访问端来说不够方便，而且这种方式受复杂网络环境影响较大，不能100%实现，稳定性欠缺。

## 四、方法实现

### 4.1 几款主流工具实现

1.frp（开源）：

FRP使用 Go 语言开发，可以支持 Windows、Linux、macOS、ARM 等多平台部署。frp内网穿透无需多复杂的配置就可以达到比较好的穿透效果，具有较强的扩展性，支持tcp, udp, http, https 协议，并且 web 服务支持根据域名进行路由转发。此外，FRP 提供了一种新的代理类型 XTCP ，可以在传输大量数据时让流量不经过服务器中转，用于实现点对点穿透。当然，此功能并不能保证在你的网络环境 100% 可用，成功率较低，而且还要求访问端也得运行 FRP 客户端 (目前手机端ios没有frp客户端软件)。由于实现条件较多，所以有文件传输需求的朋友但必须用frp的朋友还是建议买带宽稍大一点的 VPS 会比较省心。frp适合有linux基础的个人，且需要自购一个云主机做中转，企业追求稳定可以考虑其他方式。

2.ngrok（开源）：

ngrok是一个反向代理，通过在公共的端点和本地运行的Web服务器之间建立一个安全的通道。ngrok可捕获和分析所有通道上的流量，便于后期分析与响应。主要用途是给网站或者安装开发提供部署环境，但是也同样可以用来进行端口转发。ngrok适合有linux基础的个人，和frp类似，但比frp配置稍复杂，且需要自购一个云主机做中转，企业追求稳定可以考虑其他方式。

3.zerotier（开源/商业）：

ZeroTier 官方解释为将整个世界转变为单个数据中心或云区域，将所有设备，虚拟机和应用程序联接起来，就像在同一个交换机接入所有设备一样。zerotier类似VPN,为内网服务器所在NAT网络和客户机所在NAT网络各虚拟出一个VLAN,这样俩个VLAN就可以通过ZeroTier 建立连接，在通过UDP打洞技术就可以实现内网服务器和内网客户机直接交互数据。目前ZeroTier 提供的点对点穿透技术成功率还是比较高的，可以应付大多数网络，国内长城宽带无法点对点穿透。而zerotier因为是国外的项目，官方提供的中转服务器在国外，国内连接延迟高甚至连不上，所以官方给出了moons概念，可以自己搭建中转服务器。另外ZeroTier 支持 Windows、macOS、Linux 三大主流平台，iOS、Android 两大移动平台，以及 QNAP（威连通）、Synology（群晖）、Western Digital MyCloud NAS（西部数据）三个 NAS 平台，还支持 LEDE 开源路由器项目，支持客户端真的很多。zerotier适合有linux基础的个人，一般用于NAS,为了稳定性需要自购一个云主机做中转，服务器和客户机都需要安装zerotier，企业追求稳定可以考虑其他方式。

4.花生壳（商业）：

花生壳既是内网穿透软件、端口映射软件。功能比较齐全，比较简单，也是大家比较耳熟了，支持tcp,udp.https.http,socket5 应用。流量还是得走传说中的rootnode，所以是限流1G，分的等级比较多商业版、旗舰版、铂金版，需要根据不同需求付费开通相关功能。安装使用门槛低，无需自购云主机做中转，企业商用追求稳定可以选择花生壳付费版本。

5.Nat123（商业）：

nat123是内网端口映射与动态域名解析（DDNS）软件，在内网启动映射后，可在外网访问连接内网网站等应用。安装使用门槛低，无需自购云主机做中转，企业商用追求稳定可以选择此方式付费版。

6.NATAPP（商业）：

natapp是基于ngrok的国内收费内网穿透工具，类似花生壳，有免费版本，比花生壳好。免费版本：提供http,https,tcp全隧道穿透，随机域名/TCP端口，不定时强制更换域名/端口，自定义本地端口。安装使用门槛低，无需自购云主机做中转，企业商用追求稳定可以选择此方式付费版。

7.向日葵、teamviewer、anydesk（远程桌面）：

这三款比较特殊，都是用来远程桌面使用，原理就是基于点对点穿透+传统穿透，三款软件都有免费版本，经本人长期工作使用发现，teamviewer免费版稳定性和连通性可以说是最好的，但“怀疑商业用途”不时弹出加上近年来爆出的漏洞，而anydesk使用过程中会经常断连，所以现在很多国内用户都转用了免费版的向日葵，本人使用中发现移动网络环境下使用向日葵很卡。

8.VPN(加密穿透):

原理类似zerotier，利用组建虚拟网络，局域网VLAN通信的方式，不同的是不打洞，流量均走中间服务器，内网服务端和客户端均需要安装vpn软件，但是它大量使用了OpenSSL加密库中的SSLv3/TLSv1协议函数库，数据通信全程加密，非常适用于企业内网资料供员工在外网访问，一般使用开源的openvpn，因相关规定这里就不做过多描述和搭建介绍了。对了，VPN还有个作用就是用来正向代理，游戏加速器就是基于VPN正向代理原理，有兴趣的童鞋可以自行查找相关资料，而正向代理典型的工具是S..，不能说了，再说下去这篇博文就要被删了...

### 4.2 方法实现总结

商业付费版暂不具体阐述，只能说花钱的就是好，有专业团队维护，且使用简单方便，但要注意由于数据包会流经第三方，因此对数据安全也是一大隐患。那么对于个人有动手能力的用户或者中小型企业小规模使用可以选择开源方式，综合软件配置复杂度、穿透能力等因素，个人觉得目前最好的选择就是frp用于对流量转发不大的场景，而需要大量udp传输的场景选择zerotier。

## 五、案例配置

经过前面的介绍，相信大家对内网穿透的几个实现方法有了大概的了解，那么接下来就针对frp和zerotier分别做下安装配置的讲解。

### 5.1 frp安装配置

1)中转服务器frps安装配置：
FRP 安装非常容易，只需下载对应系统平台的软件包并解压就可用了。这里以centos7系统为例：
安装脚本：

```
#!/bin/bash
#0.33.0是frp的版本号 (截稿为止最新版本)
export FRP_VERSION=0.33.0
sudo mkdir -p /etc/frp
cd /etc/frp
sudo wget "https://github.com/fatedier/frp/releases/download/v${FRP_VERSION}/frp_${FRP_VERSION}_linux_amd64.tar.gz"
sudo tar xzvf frp_${FRP_VERSION}_linux_amd64.tar.gz
sudo mv frp_${FRP_VERSION}_linux_amd64/* /etc/frp
```

如上在/etc/frp目录下关注4个文件，分别是frpc、frpc.ini和frps、frps.ini，前者两个文件是客户端所关注文件，后者两个文件是服务端所关注两个文件。这里配置服务端（公网服务器），首先删掉frpc、frpc.ini两个文件，然后再进行配置，vi ./frps.ini，FRP 默认提供了2个服务端配置文件，一个是简化版的frps.ini，包含了基础配置，另一个是完整版的frps_full.ini，包含了frp其他功能配置，这里不做详细介绍，感兴趣的小伙伴可以自行去查看，初学者只需用简版配置即可，在简版frps.ini 配置文件里，默认设置了监听端口为 7000，你可以按需修改它。来看看简化版的：

```
> $ vi frps.ini
[common]
bind_port = 7000     #隧道端口，默认配置中监听的是 7000 端口，可根据自己实际情况修改。
subdomain_host = frp.abc.top  # 配置中转服务器域名绑定,可选配置，如果域名多可以设置一个
authentication_method = token  # 启用token验证，frpc也需要加此参数
token = changeit              #token密码
vhost_http_port = 80                #内网服务器http映射的端口，内网可以有多个网站使用这个端口
vhost_https_port = 443              #内网服务器https映射的端口，内网可以有多个网站使用这个端口
dashboard_port = 7500   # 配置Dashboard 监控frp状态的端口，后面可以直接访问http://frp.abc.top:7500/或http://中转服务器公网IP:7500/
dashboard_user = admin  # 监控访问账户
dashboard_pwd = admin   # 监控访问密码
```

前台启动frps

```
./frps -c ./frps.ini
```

配置自动启动：首先

```
sudo vi /lib/systemd/system/frps.service
```

在frps.service里写入以下内容

```
[Unit]
Description=fraps service
After=network.target network-online.target syslog.target
Wants=network.target network-online.target

[Service]
Type=simple

#启动服务的命令（此处写你的frps的实际安装目录）
ExecStart=/etc/frp/frps -c /etc/frp/frps.ini

[Install]
WantedBy=multi-user.target
```

这时候就可以用centos7的systemctl命令了，设置开机启动

```
sudo systemctl enable frps
```

请一定要记住，你需要将服务器的系统防火土啬，以及阿里云、腾讯云后台里找到"安全组策略"的相关配置，设置 7000 7500 或你修改过的对应端口的「允许入站和出站」，否则会一直连接不上的哦这个切记！！
如服务器使用 Win 系统，假设解压到 c:\frp 文件夹，那么只需这样启动：

```
c:\frp\frps.exe -c c:\frp\frps.ini
```

这里可以为frp添加进程守护，防止访问因意外导致frp进程终止，我们写个脚本来监控frp运行

```
#!/bin/bash
#添加本地执行路径
export LD_LIBRARY_PATH=/etc/frp/
while true; do
        server=`ps aux | grep frpc | grep -v grep`
        if [ ! "$server" ]; then
            nohup ./frpc -c ./frpc.ini &
        fi
        sleep 60
done
```

2)内网服务器frpc安装配置：
设置好中转服务器上 Frp 服务端后，我们就需要在内网服务器上安装 Frp 的客户端了。 这里安装方式和中转服务器一样，此处省略。当然内网服务器可以是Windows 电脑、Linux 设备 (比如树莓派) 或者 NAS，甚至部分路由器等设备上。Linux 客户端的安装和启动与服务器端没有太多区别，只是对应运行程序是frpc 而不是 frps。前面介绍了Linux安装，这里就以内网服务器为 Windows 电脑来安装 Frp 客户端，因为 Frp 是绿色程序，下载软件包回来解压后，启动 frpc.exe 即可。但在启动前，我们需要先修改配置文件，首先删掉frps、frps.ini、frps_full.ini几个文件。假设你的 FRP 中转服务端所在的 VPS 公网 IP 为 1.2.3.4 或为域名 frp.abc.top， 而客户端是 Win 电脑，我们来修改 frpc.ini 配置文件：

```
[common]
server_addr = 1.2.3.4 或者frp.abc.top   #公网服务器ip或域名
server_port = 7000            #隧道端口，与服务端bind_port 一致
authentication_method = token  # 启用token验证，frps也需要加此参数
token = changeit              #token密码

[ssh]           #公网通过ssh访问内部服务器（这一种ssh是最简单暴露公网不太安全的映射，如果需要安全的ssh，可以看后面的配置[secret_ssh]）
type = tcp              #连接协议tcp
local_ip = 192.168.1.30  # 内网服务器在局域网中的 IP (如是本机，也可使用 127.0.0.1)
local_port = 22         #内网服务器ssh端口号
remote_port = 6000      #自定义的访问中转服务器映射的ssh端口号，客户机访问中转服务器IP:6000即可到达本机ssh

[secret_ssh]          #公网通过ssh加密访问内部服务器，注意这种方式还需要在客户机                              [secret_ssh_visitor]
                  #上安装配置frpc.ini（客户机配置请看右边[secret_ssh_visitor]）                          type = stcp
type = stcp           #连接协议stcp                                                  role = visitor
sk = abcdefg          #客户端连接ssh的秘钥                                             server_name = secret_ssh
local_ip = 127.0.0.1                                                          sk = abcdefg
local_port = 22                                                               bind_addr = 127.0.0.1
remote_port = 6000                                                            bind_port = 6000

[web]           #公网访问内部web服务器以http方式
type = http         #访问协议
local_port = 80   #内网web服务的端口号
custom_domains = repo.iwi.com   # 所绑定的中转服务器IP的域名，一级、二级域名都可以，此域名会映射给中转服务端，客户机访问http://repo.iwi.com:8000即可到达本机web
remote_port = 8000          #自定义的访问中转服务器映射的web端口号，客户机访问中转服务器IP:8000或域名:8000即可到达本机web。如果中转服务器配置了vhost_http_port = 80和vhost_https_port = 443，这里可以省略，不省略则访问端口以这里配置为准。

[RDP]           #Windows远程桌面控制
type = tcp
local_ip = 192.168.1.30    # 内网服务器在局域网中的 IP (如是本机，也可使用 127.0.0.1)
local_port = 3389
remote_port = 7002

# 如果内网服务器是https，还需要做以下配置（frp对于https的映射代理原理在后面会有工作流程图介绍）
[web_https]                     #Https服务，映射的是服务端https443端口
type = https                   服务方式
local_ip = 127.0.0.1             #内网服务器在局域网中的 IP (如是本机，也可使用 127.0.0.1)
local_port = 443                 #内网web服务的端口号
custom_domains = test.test.com          #所绑定的中转服务器IP的域名，一级、二级域名都可以，此域名会映射给中转服务端，客户机访问https://test.test.com:443即可到达本机web

#配置https插件部分（这里都配置在了内网服务器端，即frp客户端，实际是有缺陷的，后面https工作原理介绍会提到，还可以在中转服务器，即frp服务端结合nginx配置，我个人推荐选择中转服务器端结合nginx做配置，后面会有详细配置介绍）
plugin = https2http             #配置插件，将https请求转换成http请求后再发送给本地服务
plugin_local_addr = 127.0.0.1:80        #转换http后的端口，这里端口号为前面设置[web]的local_port, 即内网服务器http的web服务的端口号
remote_port = 8001               #映射http配置，这一句可选加

#内网服务器证书相关配置
plugin_crt_path = C:\Users\Administrator\Desktop\test.test.com\fullchain1.crt  #linux 下生成的证书为fullchain.pem 格式，复制到Windows上改成.crt后缀即可
plugin_key_path = C:\Users\Administrator\Desktop\test.test.com\privkey1.key    #linux 下生成的证书为privkey.pem 格式，复制到Windows上改成.key后缀即可
plugin_host_header_rewrite = 127.0.0.1                  #这里必须写成 127.0.0.1
plugin_header_X-From-Where = frp                        #指定代理方式为 frp
```

这样就在本地上新增了"ssh"和"web"以及"RDP""https"几个可供公网访问的服务了 (它们名称可以自己取)。如果你需要添加更多的设备和服务供外网访问，那么只需要照葫芦画瓢，指定正确的IP地址和端口号即可。
注意放行端口:每个服务的 remote_port 是远程访问时要用到的端口号，注意这些端口号也要在服务器的防火土啬和安全组里放行才能顺利访问的，如上面的6000、7002、8000。

```
server
 {
 listen 443 ssl;                                        # https对应端口，注意因为有nginx也有frps，以nginx这里的端口为准，frps中的vhost_http**s**_port不用配置
   ssl_certificate /etc/letsencrypt/live/test.test.com/fullchain.pem;       # 证书存放位置
   ssl_certificate_key /etc/letsencrypt/live/test.test.com/privkey.pem;     # 证书存放位置
       server_name *.test.test.com;                                 # ip或域名，我这里以泛域名举例，毕竟是做反向代理，http就不用配置了
       rewrite ^(.*) https://$server_name$1 permanent;                      #可以做个http强制转换为https
                                                        # https可以做其他安全配置，需要的去看其他文章
    location / {
       proxy_pass  http://127.0.0.1:12369;                          #映射的frp服务端frps.ini的 vhost_http_port端口
       proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
       proxy_set_header Host $http_host;
       proxy_set_header X-NginX-Proxy true;
       proxy_http_version 1.1;
       proxy_set_header Upgrade $http_upgrade;
       proxy_set_header Connection "upgrade";
       proxy_max_temp_file_size 0;
       proxy_redirect off;
       proxy_read_timeout 240s;
    }

       error_page   500 502 503 504  /50x.html;
    location = /50x.html {
       root   /usr/share/nginx/html;
    }
 }
```

修改完nginx配置后重启nginx

```
sudo nginx -s reload
```

修改中转服务器frps.ini配置

```
[common]
.....................    其他配置省略
.....................
vhost_http_port = 12369          #客户端http映射的端口，就是上面Nginx的proxy_pass对应端口
.....................
.....................
```

配置完之后还是运行服务端测试一下Linux：

```
/etc/frp/frps -c /etc/frp/frps.ini
```

内网服务器，即frp客户端frpc.ini配置:

```
.....................
.....................
[test_http]                      #Http服务，映射的是服务端http80端口
type = http                      服务方式
local_ip = 127.0.0.1                 #服务端ip，可写本地，局域网等做反向代理的ip
local_port = 80                  服务端端口
#custom_domains = test.test.com          #需要反向代理的域名，就是服务端要代理的域名，因为我们只需要https穿透，在nginx中已经做了https的域名配置，这一句可以不用添加
```

配置完之后运行，Linux：

```
/etc/frp/frps -c /etc/frp/frps.ini
```

启动frpc，windows:

```
c:\frp\frpc.exe -c c:\frp\frpc.ini
```

3）进行远程访问:
按照上面的配置，我们想要远程桌面连接到家里的Windows 电脑，那么打开“微软远程桌面客户端”后，在地址栏里填入中转服务器公网IP:7002 即可连接。同理其他连接根据中转服务器IP:设置的端口号来访问就可以了。由此，借助Frp，你就能轻松地为本地局域网内网的设备提供公网直接访问的能力了，你可以用Frp 来转发包括但不限于ssh、http、https、转发 Unix 域套接字等服务。
上面只是最基础的教程，Frp 还有很多很多高级功能，比如给Web 增加密码保护、点对点内网穿透、设置端口白名单等等，Frp 官网上也提供了很详细的文档，感兴趣的朋友可以去研究一下。

## 5.2 zerotier安装配置

免费版支持客户端多。连入同一个网络的客户端不超过100个就都免费。所以我们自用NAS远程访问完全够了，这里以安装配置免费版为例:
ZeroTier相当于软交换机，需要在内网服务器和客户机上安装ZeroTier one 软件，然后在官方给的中转服务端或者自己搭建的云主机中转服务器（moon）中配置VLAN和路由，这样就把俩个VLAN连在了一起，里面的设备便可以互相访问。
正是因为如此，zerotier不支持域名和ddns，也就是说必须以虚拟IP互相访问，moon节点仅支持固定公网IP,且未来2.0版本依然维持此设定，如果需要用域名访问内网web服务器，此方法不可用，这也是目前我想到的这个方法唯一一个不满足的场景。
官方中转服务端或moon配置:
1.ZeroTier账号并登陆(此处省略)
2.登录后配置网络（这里默认是官方免费中转服务器，可以免费创建多个network，每个network实现一个局域网）

这样就进入了网络设置界面,网络设置界面可以配置很多东西，但是这里只介绍需要用到的部分，就是NetworkID 和下面的 Members。NetworkID 是在连入新设备时需要用到的标识符，每个设备连接这个虚拟的局域网时都要输入这个ID。Members 则是连入网络的设备列表，按照默认的设定，当有一个新设备接入网络时，用户需要在这个页面进行授权。这里其实可以不用自己做额外配置，默认就好，下面还是把其中一些重要参数截图说明下:

下面插入 一个概念：moon。
先简单提一下 UDP 打洞的原理，UDP 打洞的本质是让 NAT 后面的机器 A 和 B 先连接一个有公网 IP 的中间服务器，然后中间服务器经过一番操作之后让A 和 B 直接互联，这样之后的数据传输就是 A 和 B 之间直接传输，不再通过中间服务器。但是出于节省资源和增强健壮性的角度，A 和 B 服务器直接打出来的「洞」并不是永久的，而是维持一段时间后释放，下次连接时重新打洞。
这样就带来一个问题，Zerotier 是一款国外的项目，他提供的中间服务器到大陆的网络状况并不好，有很高的延迟，并且很可能丢包，这就导致我们每次「打洞」都有很高的延迟。为了解决这个问题，Zerotier 提供了一个 moons 的概念，中提供了详细的解释和配置方法。
如果需要moon，则要做另外配置,这里再说明下如何搭建配置中转机器(moon)centos为例:
给自己的中转主机安装 zerotier-one 这个软件包:

```
curl -s https://install.zerotier.com/ | sudo bash
```

之后就完成了安装，然后我们要启动zerotier，并且让他开机自启动，运行:

```
sudo systemctl start zerotier-one.service
sudo systemctl enable zerotier-one.service
```

进官方网络设置页面，复制这个Network ID

然后在自己的moon中转机器上执行:

```
sudo zerotier-cli join 83048a0632246d2c
```

加入之后就启动了，但是还连不进我们创建的网络，因为我们选择了Private(私有网络)，还需要到官方网络设置页面对接入的机器打勾，拉到Members 这一节，把前面的勾勾选上。这个时候执行一下 ip a 你会发现多了一个叫做 ztuzethlza 或者类似名字的设备，还有IP地址，这就是zerotier组建的局域网的IP地址。
此时自己的中转服务器已经被加入到了zerotier虚拟的局域网下。接下来就是对我们自己的中转服务器进行配置moon。
执行:

```
cd /var/lib/zerotier-one/
sudo zerotier-idtool initmoon identity.public > moon.json
```

此命令会在当前目录下生成一个文件 moon.json，文件内容如下:

```
{
  "id": "deadbeef00",
  "objtype": "world",
  "roots": [
    {
      "identity": "deadbeef00:0:34031483094...",
      "stableEndpoints": []
    }
  ],
  "signingKey": "b324d84cec708d1b51d5ac03e75afba501a12e2124705ec34a614bf8f9b2c800f44d9824ad3ab2e3da1ac52ecb39ac052ce3f54e58d8944b52632eb6d671d0e0",
  "signingKey_SECRET": "ffc5dd0b2baf1c9b220d1c9cb39633f9e2151cf350a6d0e67c913f8952bafaf3671d2226388e1406e7670dc645851bf7d3643da701fd4599fedb9914c3918db3",
  "updatesMustBeSignedBy": "b324d84cec708d1b51d5ac03e75afba501a12e2124705ec34a614bf8f9b2c800f44d9824ad3ab2e3da1ac52ecb39ac052ce3f54e58d8944b52632eb6d671d0e0",
  "worldType": "moon"
}
```

其中 id 为我们自己中转服务器在 ZeroTier 中的 id，本文为 deadbeef00。接下来编辑moon.json，把 "stableEndpoints": [] 这一节里加入我们自己的中转机器的公网IP，例如 "stableEndpoints": ["1.2.3.4/9993"]，其中 9993是默认监听的端口，接下来要把9993端口的防火墙放开(注意是UDP)，如果你的机器外边还有防火墙的话，也要一起放开，例如阿里云的机器就有防火墙规则，要一起把对应端口的UDP流量放行，如：

```
"stableEndpoints": [ "1.2.3.4/9993","2001:abcd:abcd::1/9993" ]
```

若公网机器没有 IPv6 地址，则将其修改为

```
"stableEndpoints": [ "1.2.3.4/9993" ]
```

此后，我们要生成moon的签名文件：

```
sudo zerotier-idtool genmoon moon.json
```

此命令会生成一个签名文件在当前目录下，文件名如000000deadbeef00.moon （我们自己的中转服务器 id 为 deadbeef00)
接下来将 moon 节点加入网络
在我们自己的中转服务器中的 ZeroTier 目录中建立子文件夹 moons.d
不同系统下的 ZeroTier 目录位置：
Windows: C:\ProgramData\ZeroTier\One
Macintosh: /Library/Application Support/ZeroTier/One ( 在 Terminal 中应为 /Library/Application\ Support/ZeroTier/One)
Linux: /var/lib/zerotier-one
FreeBSD/OpenBSD: /var/db/zerotier-one
将前面生成的 000000deadbeef00.moon 拷贝进 moons.d 文件夹中，并重启 ZeroTier （此步好像有些许 bug，重启电脑为佳）
注意： windows需要重启服务，而不是重启程序，按住windows键+r，在弹出的对话框中输入services.msc并回车，找到ZeroTier One ，右键选择重新启动即可

```
sudo systemctl restart zerotier-one
sync && sync && reboot
```

这样我们自己的中转服务器moon配置就完成了，下面还要将内网服务器和内网客户端加入moon，这里我自己的内网服务器用的centos7，需要先安装好zerotier-one并加入网络，和我们自己中转服务端安装方式一样，我的客户机是ios，ios需要用国外的apple账号在applestore下载zerotier one，国内商城这个app被下架了。
这里先查看前面moon.json中的id是多少,下面这个命令可查看并返回结果为“ "id": "xxxxxxxxxx",”：

```
grep id /var/lib/zerotier-one/moon.json | head -n 1
```

接着先说下我的内网centos7服务器如何加入moon节点：
复制前面查到的id，然后在内网centos7机器执行：

```
sudo zerotier-cli orbit xxxxxxxxxx xxxxxxxxxx
```

注意，xxxxxxxxxx 要两遍。此后重启 zerotier one，完毕。等一会儿之后，zerotier局域网内的机器就可以互相访问了，延时非常低：

```
sudo systemctl restart zerotier-one
```

要验证是否moon生效,只需要在客户端zerotier程序目录下,执行以下命令即可:

```
zerotier-cli listpeers
```

若有类似地址,即可证明moon连接成功
200 listpeers abcdefgh00 1.1.1.1/9994;4242;4038 224 1.2.12 MOON
接下来再说下内网客户端ios如何加入moon节点：
　其实这里是个大问题，移动端如何加入moon节点，目前还没研究出来...
内网服务器端和客户端安装配置:
前面介绍moon安装配置的时候已经描述了centos整个安装过程，参考我们自己的中转服务器安装配置过程即可，这里针对Windows做个简短的安装介绍。
进官网下载Windows的msi安装包后直接安装；
安装完后打开，在桌面下方可以看到黄色的图标，鼠标右击选择'join Network"

弹出一个小的会话框，三个选项全部勾选上并在上面填入中转服务器的Network ID后，点击 “join”

稍等片刻，会弹出此窗口，选择“是”即可

最后刷新官网中Network配置页面，在下方的“member”区域中勾选将其注册到网络中

好了,到这里关于zerotier的服务端和客户端安装配置都做完了，zerotier网络组建成功后，相当于里面的设备在同一局域网下或俩个不同VLAN下，所以也就意味着里面的所有设备只要开放了端口互相就能访问，而不需要像frp那样对单独的某个端口最设置，这一点也说明zerotier和vpn相似。

最后针对zerotier，有网友对它点对点穿透模式做过穿透效果测试，这里引用了过来，供参考：
"事实上 Zerotier 的打洞效果好的令我惊讶，我的网络环境 A 是上行 20M 下行 100M 的电信网络，B 则是移动赠送的 50M 上下行对等移动网络。晚高峰时段，在两边都开启 bbr 的情况下，互相下载文件可以跑到网络峰值的 60%~70%。电信拖移动能跑3MB/s，反过来则是 1.4MB/s 左右。延迟的话「打洞」完成之后稳定在 90ms，丢包率在 8% 左右。顺便吐槽一下，一开始看到 90ms 的延迟我还以为打洞出了问题，结果直接从移动 ping 了一下电信的公网 IP（我的电信网络环境是有公网 IP 的，但是我并没有在路由器设置端口转发，因此测试效果仍然是打洞的效果），发现延迟也在 90ms 上下，通过 traceroute 命令追踪路由发现数据包先从隔壁的 H 省进入移动骨干网，然后进入电信骨干，再从另一个方向隔壁的S 省回到本地电信，难怪会有这么高的延迟。同样，8% 也是晚高峰电信和移动互联本身的延迟。换而言之，Zerotier 的打洞非常成功，延迟和丢包率基本都和公网差距不大，但是由于本身移动和电信晚高峰时段的互联效果较差，因此效果并没有想象中完美。从延迟的角度，本地电信到香港阿里云服务器的延迟尚在50ms 以下，到洛杉矶的 CN2 服务器也不过 150ms，然而与本地移动互联延迟竟然高达 90ms，国内 ISP 互联还是任重道远啊。"

# 六、总结

以上基本把内网穿透原理和实现方法很详细的介绍了一遍，还是针对不同实现方法做个总结吧。

1.如果是企业内网资料需要员工远程访问就用VPN。

2.自己家里NAS或远程桌面直接用zerotier，我自己没有额外搭建moon，有点延迟但可以接收，还有远程桌面也优先用zerotier。

3.一些页面访问用frp，比如测试用的web服务、路由器管理界面、NAS管理界面等等。

4.最后如果觉得麻烦，资金允许可以直接花生壳之类的收费穿透。