SSH隧道翻墙的原理和实现

跟朋友闲聊说到翻墙和SSH隧道，刚开始我对SSH的理解就是远程连接，然而SSH隧道并非那么简单，利用SSH隧道可以干很多事，翻墙就是其中之一。

VPN vs SSH

VPN和SSH隧道翻墙有如下区别：

1. VPN的设置是全局的，即电脑一旦挂上VPN，所有的联网程序都将自动使用VPN；而建立好SSH隧道后，需要程序设定使用隧道才会使用隧道联网
2. 如果使用商用的VPN一般要花钱，如果自己搭，VPN的搭建难度大于SSH隧道。因为SSH隧道只要一台可以SSH的服务器即可，根本不需要在服务器上配置任何东西

SSH动态绑定

这是SSH翻墙的基本原理：利用SSH动态绑定的功能。那么何谓SSH动态绑定呢？动态绑定是SSH端口转发功能的一种形式，借用一张图：

1. 首先，墙内的客户机跟墙外的代理服务器，建立好SSH连接，并设定动态绑定
2. 此时墙内客户机上的SSH会监听本地的一个端口7001
3. 客户机上的程序，将对www.youtube.com:80的请求告知7001端口的SSH，SSH将此请求通过SSH加密连接发送到墙外服务器的SSH上
4. 由于建立的动态绑定，服务器会将www.youtube.com:80的请求发送给www.youtube.com上的80端口，并在收到回复后，通过原路返回给客户机的SSH，客户机的SSH返回给应用程序

在这里SSH客户端已经不仅仅是个客户端了，它同时打开了7001端口侦听本机应用程序的请求。这是SSH跟传统用法最大的区别。而服务端的SSH也不仅仅是处理客户端的请求，而是将请求转发到对应的主机和端口，这里的动态二字体现在服务端的SSH的转发目标是不固定的，是根据客户端的请求而

定的。

那么如何让应用程序知道应该把请求发送给本机的7001端口呢?

SOCKS代理

答案就是SOCKS代理。

理解SOCKS代理其实非常简单。HTTP代理都用过吧，浏览器其实也是支持SOCKS代理的，玩法几乎一样，只是SOCKS代理通常不限制端口，所谓来者不拒。

实际上SOCKS代理普遍被许多应用程序支持：QQ、浏览器、MSN…

所以在上述的模型中，客户机的SSH实际上就是实现了一个SOCKS代理的角色，这个SOCKS代理侦听了7001端口，并将所有的请求都代理给服务器的SSH，并利用SSH动态绑定，让服务器进一步转发请求。

SSH隧道的搭建

那么我们来看看搭建一个SSH隧道翻墙，究竟有多简单。首先你需要有一台支持SSH的墙外服务器，此服务器啥都不需要，只要能SSH连接即可。

客户端SSH执行如下命令:

```
ssh -D 7001 username@remote-host
```

上述命令中`-D`表示动态绑定，`7001`表示本地SOCKS代理的侦听端口，可以改成别的，后面的`username@remote-host`就是你登录远程服务器的用户名和主机。当然，这个命令后会提示输入密码，就是`username`这个用户的密码（除非你配置了SSH公钥认证，可以不输入密码）

这样隧道就打通了！是不是超级简单。

最后在浏览器或者其他应用程序上设置SOCKS代理(设置v4的SOCKS就可以了，v5的SOCKS增加了鉴权功能)，代理指向`127.0.0.1`，端口`7001`即可，这样免费的翻墙就做好了。最后再盗一张图:

SSH相当强大，还有很多好用的功能，可以参阅：

[实战 SSH 端口转发]