[首页](https://www.oschina.net/)

[开源项目](https://www.oschina.net/project)

[问答](https://www.oschina.net/question)

[动弹](https://www.oschina.net/tweets) [博客](https://www.oschina.net/blog) [翻译](https://www.oschina.net/translate) [资讯](https://www.oschina.net/news)

[专题](https://www.oschina.net/android)

[活动](https://www.oschina.net/event) [招聘](https://job.oschina.net/)

[ [登录](https://www.oschina.net/home/login?goto_page=https%3A%2F%2Fmy.oschina.net%2Fu%2F660063%2Fblog%2F201187) | [注册](https://www.oschina.net/home/reg) ]

[博客专区](https://www.oschina.net/blog) > [李秀杰的博客](https://my.oschina.net/u/660063/blog) > 博客详情

socks代理服务器协议的说明

[李秀杰](https://my.oschina.net/u/660063/home) 发表于3年前

转socks代理服务器协议的说明

[收藏](javascript:void(0);)

[李秀杰](https://my.oschina.net/u/660063/home)

* 发表于 3年前

* 阅读 5429

* 收藏 13

* 点赞 1

* [评论 0](https://my.oschina.net/u/660063/blog/201187#comment-list)

[**【粉丝福利】-《web 前端基础到实战系列课程》免费在线直播教学>>>**](http://web.tanzhouedu.com/index/osc.html)  https://my.oschina.net/img/hot3.png

socks代理里面，有socks4 socks4a socks5，也有叫sock4 sock4a sock5的，都是一回事。

socks4 socks4a和socks5的tcp部分极其简单。

这又出现一个怪现象了，网文中除了一个人说道udp比tcp复杂，其他的全都说udp简单，只说tcp部分，把udp都一笔带过了。

而我就明显感到udp比tcp复杂很多。

socks的rfc是rfc1928和rfc1929，别看了，打个比方，某人看到一本书，当他看了目录，准备往后翻看内容时，发现已经到底了，原来这不是“目录”它就是内容，这就是rfc1928，话说rfc是干什么的啊，写成这样就算交活了啊，还有王法吗，还有法律吗。中文翻译的就像是机器翻译一样，不是人类语言。所以造成网文上全都是连蒙加猜，没有一个对的，这两天我连续翻看那些网文，又用现成的带socks的程序然后sniffer，算是知道了个大概，其实用程序编写出来很简单，关键就是rfc的协议说的不明不白的，所以造成本来很简单的事情，弄得大家全都一知半解。

socks4和socks4a只代理tcp。

socks4

比如ie可以直接用socks代理，但是ie只能用sock4代理，即使我现在用最新的ie9仍然如此。比如ie要用sock4访问百度，那么ie就先用本地的dns把百度转成ip比如202.108.22.5，然后向sock4服务器发送

04 01 00 50 CA 6C 16 05  41 64 6D 69 6E 69 73 74 72 61 74 6F 72 00

1、其中开头04 01是固定的

2、00 50是ie要连接站点的端口号，这里是80端口，十六进制就是00 50

3、CA 6C 16 05就是202.108.22.5的十六进制

4、41 64 6D 69 6E 69 73 74 72 61 74 6F 72是Administrator的acsii码

5、最后一个00也是固定的

这就是socks4的固定格式04 01+端口2字节+ip4字节+id+00，其中的id，这里是Administrator是可有也可以没有，因为我现在win7用的账户是Administrator所以ie就把账号名给发过去了，我试了一下最新的火狐4.0，他发送MOZ，有的就直接省略掉了

04 01 00 50 CA 6C 16 05 00 所以发成这样也行，随便写什么都行，只要最后一个是0中间没有0就行了。

如果代理服务器允许代理，比如验证一下那个id，比如你想只允许id是administrator后，允许代理了，就发回8个字节

04 5A 00 50 CA 6C 16 05

看到了吗，区别就是第二个字节是5A而且最后没有00，那么就是允许代理。其实根据协议5A这里可以换其他的反馈字节，但是我们就不要管了，如果你做的代理程序不允许他用，那么直接就在程序中断开连接就行了。

如果允许连接了，然后就把客户端的所有数据原封不动的发到远程主机，把收到远程主机的内容原封不动发回客户端就行了，以下我就简称交换交换就行了。

socks4a

细心的人会发现一个问题，在sock4中，客户端比如ie要用sock4代理访问百度，ie会在本地dns解析出百度的ip202.108.22.5，而不是通过sock4代理的那个机器的dns解析的，这样就会出现一个问题，比如你的那个机器是个受限制的网络dns不给你解析域名，只让你以ip方式访问网络，那么怎么办呢？就没办法了对吧，你知道百度的ip没用啊，网络上都以域名形式存在的，其他的你都能知道吗对吧。那么ie为什么要在本地dns解析而不再sock4代理机器上解析呢？因为sock4协议就这样的，否则就不叫sock4协议了。

所以sock4a协议就是为了弥补这个问题的，可以让客户把域名发到sock4a代理机器上，用这个代理机器上的dns解析出ip，这样就克服了客户端不方便解析域名的情况，而且有的网站比如google在本地解析和在代理服务器解析域名得到的ip是不一样的，在代理服务器解析域名得到的ip更好，一般连接速度更快也更像就是从代理服务器发起的。

socks4a是这样的，截取我用flashfxp通过socks4a代理登录sourceforge的sftp

04 01 00 16 00 00 00 01  6C 61 6F 74 73 65 00 65 62 2E 73 6F 75 72 63  65 66 6F 72 67 65 2E 6E  65 74 00

1、04 01是固定的

2、00 16是端口22的十六进制

3、00 00 00 01是原本ip地址的地方，这里前3个字节必须为0，最后一个必须不能为0，一般都是1

4、6C 61 6F 74 73 65是laotse的ascii，我设的flashfxp代理的命名，可以没有，和sock4一样

5、00 固定的，和sock4一样，如果第四步的id没有，这个00也不能省略

6、65 62 2E 73 6F 75 72 63 65 66 6F 72 67 65 2E 6E 65 74 就是域名web.sourceforge.net的ascii了

7、00 固定的。

这样就可以看出，就是把sock4的ip那地方换成00 00 00 xx这样的假ip，然后再sock4后面多了域名+00

这样socks4a代理首先把web.sourceforge.net解析成216.34.181.70（D8 22 B5 46），并且和216.34.181.70连接上，再向客户端发送

04 5A 00 16 D8 22 B5 46

和sock4一样的，依然是04 5A+端口+IP共8字节。

然后就和sock4一样就是转发转发了，如果不允许代理直接断开就行了也不用发送什么反馈了。

所以明显socks4a比socks4要好，这样避免了本地对域名解析带来的问题，话说如果你要上些不和谐的站呢，总是用本地dns解析那些域名……是吧。所以能用socks4a就不要用socks4，但是杯具的是ie只能用sock4，ie9也如此，火狐也是。

看到了吧socks4和socks4a就是这么简单，简单的不能再简单了。

socks5

socks5代理和socks4 socks4a比，多了一个验证功能和udp代理的功能。

socks5的tcp代理几乎和socks4 socks4a一样简单，但是udp却比较复杂一点，但是再复杂也没有http代理那么复杂。

首先说验证，不管是要代理tcp还是udp，开始验证都是要一个tcp连接的，而且验证过程一样，验证完了才分tcp和udp的实际代理过程，有好几种验证，别管了，其他的都用不到太小众化了，那些协议是什么都没听说过，也没处试验去，信我的，就记住3种就行

如果一个客户想通过socks5代理，那么开始他会发送以下3种内容

05 01 00 共3字节，这种是要求匿名代理

05 01 02 共3字节，这种是要求以用户名密码方式验证代理

05 02 00 02 共4字节，这种是要求以匿名或者用户名密码方式代理

如果socks5代理允许匿名那么就返回05 00两个字节，如果要求验证就返回05 02两个字节。

这里匿名等一下，先说要求密码验证的，因为要求验证就比匿名多了一步而已，后头是一样的，所以先说密码验证。

当上面socks5返回05 02两个字节后

客户端发送01 06 6C 61 6F  74 73 65 06 36 36 36 38 38 38

1、01固定的

2、06这一个字节这是指明用户名长度，说明后面紧跟的6个字节就是用户名

3、6C 61 6F 74 73 65这就是那6个是用户名，是laotse的ascii

4、又一个06共1个字节指明密码长度，说明后面紧跟的6个字节就是密码

5、36 36 36 38 38 38就是这6个是密码，666888的ascii。

6、假如这后面还有字节，一律无视。

这时socks5代理就验证了用户名laotse密码666888对不对啊，如果不对直接关闭连接就可不用反馈了。

如果这个用户名和密码通过了，可以进行代理，那么就发送01 00给客户端。那么下面就和匿名是一样的了，匿名就是省略了这一步而已。

这时无论匿名或者通过了密码验证的客户端向socks5发送下列三种方式

先说tcp的

第一种

05 01 00 03 13 77  65 62 2E 73 6F 75 72 63  65 66 6F 72 67 65 2E 6E  65 74 00 16

1、05固定

2、01说明是tcp

3、00固定

4、03说明后面跟着的是域名而不是ip地址，由socks5服务器进行dns解析

5、13前面指明了是域名，那么0x13（19字节）是域名字符长度

6、77 65 62 2E 73 6F 75 72 63 65 66 6F 72 67 65 2E 6E 65 74 就这19个是域名web.sourceforge.net的ascii。

7、00 16端口，即为22端口。

第二种

05 01 00 01 CA 6C 16 05 00 50

1、05固定

2、01说明tcp

3、00固定

4、01说明是ip地址

5、CA 6C 16 05就是202.108.22.5了，百度ip

6、00 50端口，即为80端口

看到了吗，tcp的可以本地解析出ip来，只让socks5代理去连，也可以发过域名去让socks5去用它的dns去解析ip再连接

这时代理服务器收到了上面的请求后，如果是域名的，先解析出ip来连接，如果直接是ip的就用一个tcp连接去连接那个ip和端口，如果和远程主机连接成功了，就向客户发送什么呢

05 00 00 01 C0 A8  00 08 16 CE共10个字节

无论上面两种哪一种都是这样

1、05 00 00 01固定的

2、后面8个字节可以全是00，也可以发送socks5服务器连接远程主机用到的ip地址和端口，比如这里C0 A8 00 08，就是192.168.0.8，这是我socks5服务器的ip地址，16 CE即5838端口，即是socks5服务器用5838端口去连接百度的80端口。也可以05 00 00 01 00 00 00 00 00 00，只告知客户连接成功不告诉他细节，但是0不要省略。

后面就是在客户和远程主机之间转发转发啊的，是不是很容易啊，比http代理简单太多了。

然后说udp的，udp的要复杂不少。首先要说下原理，udp和tcp不一样，不是一个连接中一口气下来的，上面说了不管tcp还是udp上面的那个tcp协商部分都是一样的，而且如果是udp的话会占用socks5代理一个tcp连接一个udp。

第三种udp的

客户端如qq发送（仍在刚才的tcp中发送）

05 03 00 01 00  00 00 00 E5 F0

1、05固定

2、03说明是要代理udp

3、00固定

4、01固定，只能制定后面跟着的是ip地址

5、00 00 00 00这里可以由客户端如qq发送客户的ip，也可全是0，因为这个ip地址没用。

6、E5 F0最重要的一条，客户端比如qq，向socks5代理说明它预备开放的udp端口，这里是58864。

那么socks5怎么回答呢？如果不同意代理直接关闭连接就可以了不用反馈了。如果同意的话，socks5要这么做，首先准备一个ip和一个udp端口，比如我用192.168.0.8这个ip上开放58865udp端口给客户转发用。然后返回

05 00 00 01 C0 A8  00 08 E5 F1

1、05 00 00 01固定

2、C0 A8 00 08预备开放udp端口开放给客户的ip，这里是192.168.0.8，如果多ip机器，那么返回下面开放udp端口绑定的那个ip。

3、E5 F1返回给客户说明预备开放哪个端口，这里是58865。

好了tcp协商部分完成了，注意这个tcp连接一定不要关闭，要一直开着，虽然再也不会发送和接受数据了，但是要一直开着，如果这个连接一关，那么客户就认为连接被断开了，因为这就是socks5协议，所以说socsk5转发udp不但要占用一个udp还要占用一个tcp连接。麻烦吧。

上面那个tcp不要关，下面就是客户和socks5的2个udp端口之间进行数据交换了，这里udp麻烦的一方面又体现出来了，它就是无连接的，它不像tcp那样因为tcp协议部分就保证了数据的可靠性，这么说吧比如tcp连接qq发送abcdef，这些数据太大了一次发送不完，那么就会分片，那么socks5可能会收到ab第二次c第三次def这样的，虽然不知道能一次收多少，但是只要连起来还是abcdef的顺序，而且不会出现数据丢失而发送方不知道的情况。udp就不一样，可能第一次就接到了def，第二次才接到a，bc可能直接就丢了还不知道，所以要保证udp数据的完整，不能靠udp协议这一层了，得自己手动指定，那么在socks5里就有一个udp分包的概念，就是在头几个字节指定这是1号包还是2号包，socks5程序必须自己弄个排序，比如第一次接到标记为3号的包，那么先存起来等着1号2号来了把3号放后面再发，所以说是很麻烦的，而且rfc也说了应用程序尽量不要弄这种分包，而且rfc说了，socks5程序可以选择拒绝这种分包方式接到后直接丢弃而不通知客户端，所以既然那么麻烦，咱也不用去实现这用不大上的功能，因为即使你这socks5程序实现了，对于应用程序比如qq来说还是不可靠的，而使用udp的应用程序在它应用程序本身就有个完整性和排序的功能，比如丢包了，qq之间自己就知道了，qq之间自己会去想办法重发还是排序的什么的，所以我们就不用去管分包了，让应用程序自己去解决吧，我们只实现转发不分包的那种就行了。

 既然只实现不分包的，那么格式就固定了

客户端qq的58864udp端口向socks5的58865udp端口发送什么呢，仍然是ip+端口或者域名+端口方式

00 00 00 01 70 5F F0 3C  1F 40 +实体数据

比如这个，00 00 00 01开头，那么后面4个就是ip地址70 5F F0 3C即112.95.240.60，1F 40 即端口8000，后面的全都是实体数据了。那么socks5服务器就用58865udp端口向qq的服务器112.95.240.60的8000端口发送后面的实体数据而不要发送前面那些封装内容，那么会受到qq服务器返回58865udp端口的数据，返回的都是实体数据，因为代理嘛，就像是socks5那台机器在用qq一样，所以收到的数据没有前面的封装都是实体数据。那么socks5就要返回给客户端，还不能直接返回，得包装一下

00 00 00 01 70 5F F0 3C  1F 40 +收到远程主机返回的数据

把这个返回给客户端的58864udp端口

是不是前面包装内容都是一样的啊，是一样的，因为客户端已经指明了ip，所以肯定是一样的。

还一种是这样域名的，qq的58864udp端口发送给socks5的58865udp端口

00 00 00 03 12 67 72 6F  75 70 63 6C 69 65 6E 74  2E 71 71 2E 63 6F 6D 23  29+实体数据

00 00 00 03开头说明后面跟的是域名，紧跟着的12说明后面0x12(十进制18)字节就是域名，解出来就是groupclient.qq.com

后面23 29即端口9001。那么socks5服务器就要先把groupclient.qq.com的ip给dns出来58.251.62.164（3A FB 3E A4 ），用58865up端口向58.251.62.164的9001udp发送后面的实体数据，返回来后和上面一样向客户qq的58864udp发送

00 00 00 01 3A FB 3E A4 23 29 +收到远程主机返回的数据，03变01了，直接就ip+端口了

大家注意到了没，客户和socks5tcp协商后，socks5开放的udp端口，既和客户开放的udp端口联系，也和需要连接的远程主机之间联系，都用一个端口所以有点乱，这样就得判断了，如果发现像这个端口如58864udp发送数据的ip和udp端口，是之前协商的那个，就说明是客户的数据，这时就要把客户要发送的远程主机的ip和端口记录下来，比如上例的58.251.62.164的9001udp端口，如果发现是从58.251.62.164的9001udp发送过来的数据，那么说明是远程主机发回的，那么需要转发给协商好的客户，还有一种情况，既不是客户也不在远程主机列表中的机器发过来的数据，就要丢弃，而且比如说客户发过来一条数据是要发送给远程A的a端口，那么发送出去接收到返回给客户，客户又继续发过来一条，这次要发给远程B的b端口，那么就要发出去接收返回给客户，那么这时远程主机就要有个列表了，现在有2条记录，只要接收到的udp在这2条中，就要转发给客户，如果客户又要发给C的c，那么列表就3条记录了，那么可能4条5条。

看出来了吧，socks5代理udp比起tcp来是很麻烦很麻烦的，不但要占用一个tcp一直维持连接，而且还要手动搞这种列表，但是话说回来了，虽然麻烦，但是比起编写http代理去解析http还是要容易多了。

socks5还有一种bind的tcp方式，说是ftp协议中有一种主动模式是一个tcp连上ftp服务器的21后，经过协商，服务器的某端口会反向主动连接客户端的某端口，很早以前好像见过，现在这种模式基本没用，ftp服务器和主机协商有什么用啊，现在的机器要么是在防火墙后，要么是在局域网中，ftp服务器反向连接怎么能连上客户呢是吧，所以现在的ftp几乎都用的是客户主动连接ftp服务器的被动模式，socks5的bind就适用于那种老的主动模式，用处很小很小，所以咱不去考虑。

 ie9只能用sock4，4a和5不行

火狐4可以用sock4, 4a不行，可以用sock5，但是是个半残，不支持用户名和密码验证，而且sock5中我上面说了可以发域名过去让sock5代理去dns解析，但是火狐4却非要在本地解析域名，只用sock5的ip模式，无法达到彻底隐藏的目的，话说如果你想上某些不和谐站，如果被人发现你老是在本地dns解析那些域名……是吧。rfc管这叫dns泄露。

超时怎么做啊？

我觉得，无论4 4a 还是5，对于tcp的代理，如果发现无论远程还是客户只要5分钟内没有数据传输，就把远程和客户的这2条tcp连接断开就行了，对于5的udp也是，如果发现udp端口5分钟都没有客户的udp数据包发过来，就把这个udp端口关闭，把那个和客户维持的tcp断开。而不论4 4a 还是5，只要客户主动断开，那么就把为这个客户开的一切资源全都关闭掉。不用担心，一般程序都会定时发送维持性连接信息的，不会在那连着不收不发就那么耗着不管了的，所以5分钟都没数据，就可以认为已经断开了，就关闭就行。

这个问题很严重，如果不做好不一会就会把服务器的端口全部占满，亲身体会，其实早断开了，但就是不释放，好几天都显示ESTABLISHED，等到4294967295秒后才会自动关闭，所以这个问题要慎重，保守一点好。

具体代码可参照些此文

socks4,socks5代理服务器实现

http://gc063tzf.blog.163.com/blog/static/11216292201211163647284/

原文地址：[http://gc063tzf.blog.163.com/blog/static/11216292201210246448371/](http://gc063tzf.blog.163.com/blog/static/11216292201210246448371/" \t "_blank)

* 分类：[转贴的文章](https://my.oschina.net/u/660063/blog?catalog=217040)

* 字数：5309

标签： [socks](https://my.oschina.net/u/660063/blog?search=socks) [socks4](https://my.oschina.net/u/660063/blog?search=socks4) [socks5](https://my.oschina.net/u/660063/blog?search=socks5) [网络协议](https://my.oschina.net/u/660063/blog?search=%E7%BD%91%E7%BB%9C%E5%8D%8F%E8%AE%AE)

* [打赏](javascript:void(0);)

* [点赞](javascript:void(0);)

* [收藏](javascript:void(0);)

* [分享](javascript:void(0);)

[](https://my.oschina.net/u/660063/home)

+ 关注

[李秀杰](https://my.oschina.net/u/660063/home)

 高级程序员  海淀

粉丝 2

博文 1

码字总数 0

相关博客

[Socks5代理Socks5 Proxy](https://my.oschina.net/u/1585857/blog/857530)

https://static.oschina.net/uploads/user/792/1585857_50.png?t=1397024850000 大学霸

13 [0](http://my.oschina.com:8888/J25/blog/190490#comment-list)

[pcDuino上配置Dante SOCKS5代理服务器](https://my.oschina.net/u/1156611/blog/147343)

https://static.oschina.net/uploads/user/578/1156611_50.jpg?t=1373540388000 realZ

394 [0](http://my.oschina.com:8888/J25/blog/190490#comment-list)

[http,socks4,socks5代理的区别](https://my.oschina.net/aiguozhe/blog/127279)

https://static.oschina.net/uploads/user/51/103999_50.png?t=1470401830000 爱国者

1349 [0](http://my.oschina.com:8888/J25/blog/190490#comment-list)

评论 *(0)*

**社区**

* [开源项目](https://my.oschina.net/project)
* [技术问答](https://my.oschina.net/question)
* [动弹](https://my.oschina.net/tweets)
* [博客](https://my.oschina.net/blog)
* [开源资讯](https://my.oschina.net/news)
* [技术翻译](https://my.oschina.net/translate)
* [专题](https://my.oschina.net/android)
* [招聘](https://job.oschina.net/)

**众包**

* [项目大厅](https://zb.oschina.net/projects)
* [软件与服务](https://zb.oschina.net/services)
* [接活赚钱](https://zb.oschina.net/account/choose-role)

**码云**

* [Git代码托管](https://git.oschina.net/)
* [Team](https://team.oschina.net/)
* [PaaS](https://git.oschina.net/paas_osc)
* [在线工具](http://tool.oschina.net/)

**活动**

* [线下活动](https://my.oschina.net/event?tab=latest&city=%E5%85%A8%E5%9B%BD&time=all)
* [发起活动](https://my.oschina.net/event/create)
* [源创会](https://my.oschina.net/event/ych)

关注微信公众号



下载手机客户端

[](https://my.oschina.net/app)

* ©开源中国(OSChina.NET)
* [关于我们](https://www.oschina.net/home/aboutosc)
* [广告联系](mailto:market@oschina.cn)
* [@新浪微博](https://weibo.com/oschina2010)
* [合作单位](https://www.oschina.net/home/aboutosc#partners)

开源中国社区是工信部[开源软件推进联盟](http://www.copu.org.cn/)指定的官方社区粤ICP备12009483号-3