FTP协议早在1971年就出现了，而我们学习过互联网的历史，互联网前身Arpanet网在1969年才出现，仅仅两年，FTP协议就横空出世，到现在还在普遍使用，可想而知这个协议的厉害之处，它的过人之处在于以简单的方式实现了文件的传输，这里的简单指的是使用简单，我们只要输入ip、用户名和密码即可。他是我们平时最容易碰到的协议，也是我们作为开发、测试、运维人员必须了解的协议。

一、FTP定义和概述

FTP：File Transfer Protocol，即文件传输协议，属于应用层协议，是用于在网络上进行文件传输的一套标准协议。使用C/S架构，即客户/服务器模式。

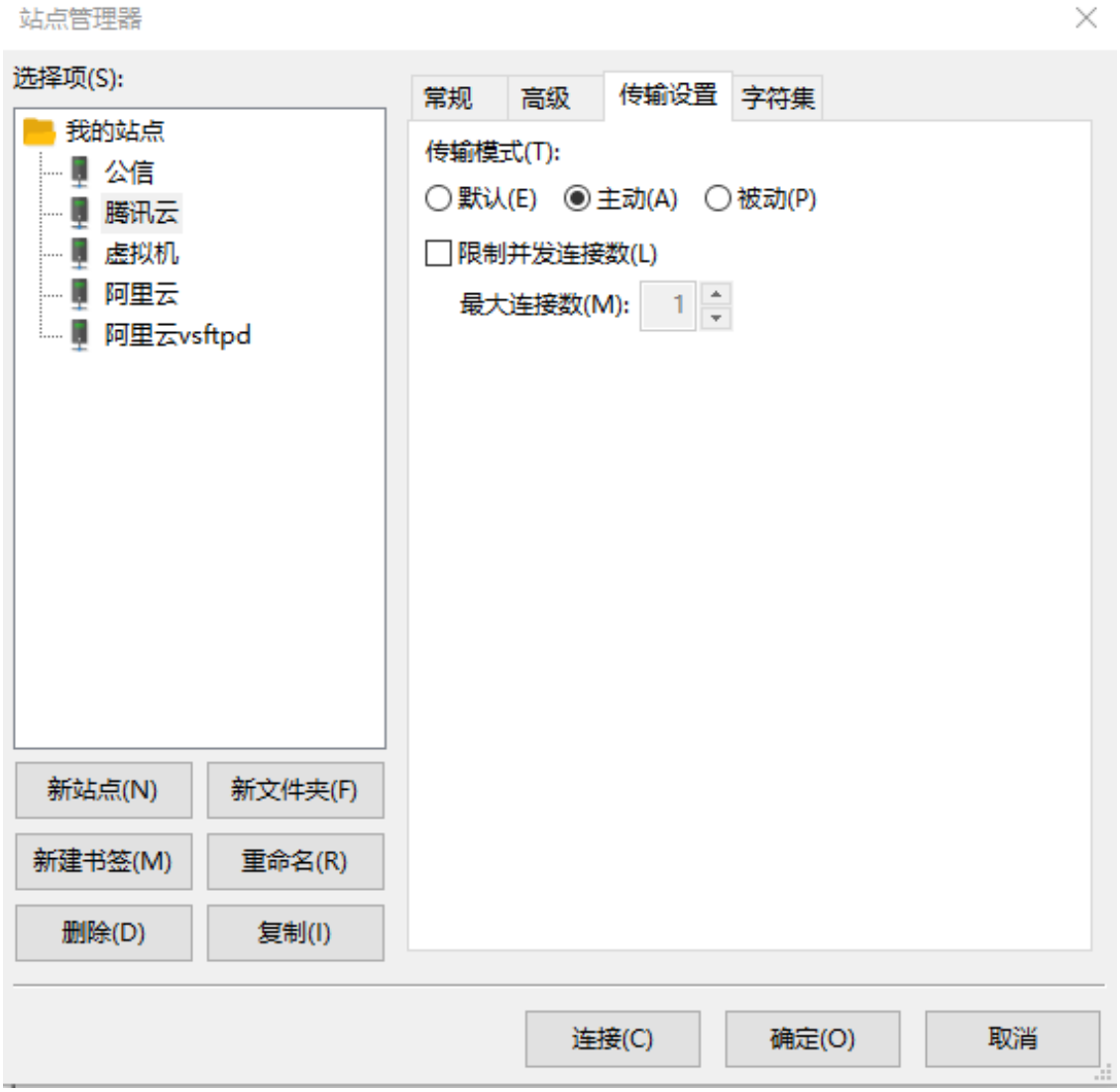
我们都知道FTP工作在21端口，但是实际上21端口只是一个传输控制端口，用于连接的建立、登录。而数据的传输是通过其他的端口来实现的，这个时候还会再发起新的握手，继而进行数据的传输。

提到数据传输，就不得不提FTP中使用的两种模式：主动模式和被动模式。

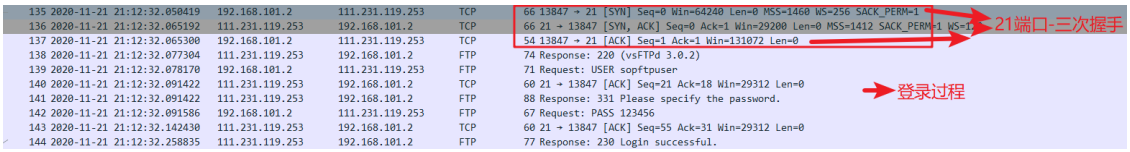
> 主动模式：服务器数据端口，会使用 tcp20 端口主动去连接实时协商客户端的随机端口（随机端口每次传输都不一样）
> 被动模式：客户端主动去连接服务器端，客户端数据端口与服务端数据端口都是随机的，是在传输数据时实时协商的；每次传输数据，端口都是会变的。

从定义来看，我们大概知道了，主动模式就是服务端固定用20端口，而客户端随机。
看起来要简单点，让我们先来分析下这个模式是如何工作的。首先FTP客户端软件FileZilla要确认连接的模式是主动模式。

二、主动模式



OK，下面我进行连接，并且用 `wireshark` 进行抓包。先看下登录的流程。



138号包：(服务端说)我已经准备好了，顺便告诉你我的使用的是vsftpd 3.0.2的服务。

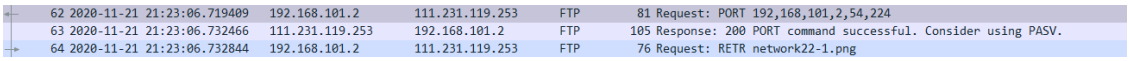
139号包：（客户端说）用 `sopftpuser` 这个用户名登录。

141号包：（服务端说）可以，你告诉我密码。

142号包：（客户端说）我的密码是著名的123456。

144号包：（服务端说）密码正确，恭喜 `sopftpuser` 登录成功。

第一，我们需要注意，客户端是与服务端的21端口进行的交互，且是明文传输，密码123456展示得清清楚楚。如果是安全要求比较高的情况下，这种方式显然是不行的。我们先往下看，下载一个文件试试。



62	2020-11-21 21:23:06.719409	192.168.101.2	111.231.119.253	FTP	81 Request: PORT 192,168,101,2,54,224
63	2020-11-21 21:23:06.732466	111.231.119.253	192.168.101.2	FTP	105 Response: 200 PORT command successful. Consider using PASV.
64	2020-11-21 21:23:06.732844	192.168.101.2	111.231.119.253	FTP	76 Request: RETR network22-1.png

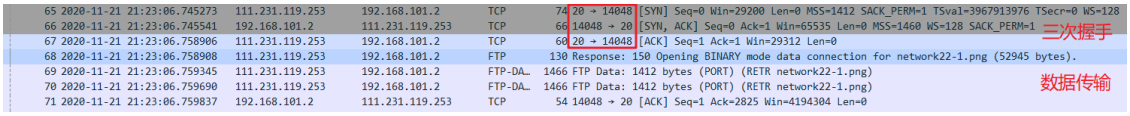
62号包：（客户端说）我想从IP=192.168.101.2上端口为54*256+224=14048连接你的数据端口。

> 这里说下：报文中有个PROT命令，比如是(h1,h2,h3,h4,p1,p2)，前面四个数是IP地址，后面两个是用于表示端口，固定的是p1*256+p2来计算，这里的256是个固定常数。

63号包：（服务端说）200表示好的，我同意。

64号包：（客户端说）RETR表示下载文件，上传文件指令是STOR。表示我要下载network22-1.png这个图片。确实，我就是要下载这张图片。

这三个包是协商了客户端的数据端口。服务端的数据端口固定是20，因此两边的数据端口都已经确定，**下面就是再次握手和数据真正的传输了**。让我们继续往下看吧。



65	2020-11-21 21:23:06.745273	111.231.119.253	192.168.101.2	TCP	74	20 → 14048 [SYN] Seq=0 Win=29200 Len=0 MSS=1412 SACK_PERM=1 TSval=3967913976 TSecr=0 WS=128
66	2020-11-21 21:23:06.745541	192.168.101.2	111.231.119.253	TCP	66	14048 → 20 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=128 SACK_PERM=1
67	2020-11-21 21:23:06.758906	111.231.119.253	192.168.101.2	TCP	60	20 → 14048 [ACK] Seq=1 Ack=1 Win=29312 Len=0
68	2020-11-21 21:23:06.758908	111.231.119.253	192.168.101.2	FTP	130	Response: 150 Opening BINARY mode data connection for network22-1.png (52945 bytes).
69	2020-11-21 21:23:06.759345	111.231.119.253	192.168.101.2	FTP-DA…	1466	FTP Data: 1412 bytes (PORT) (RETR network22-1.png)
70	2020-11-21 21:23:06.759690	111.231.119.253	192.168.101.2	FTP-DA…	1466	FTP Data: 1412 bytes (PORT) (RETR network22-1.png)
71	2020-11-21 21:23:06.759837	192.168.101.2	111.231.119.253	TCP	54	14048 → 20 [ACK] Seq=1 Ack=2825 Win=4194304 Len=0

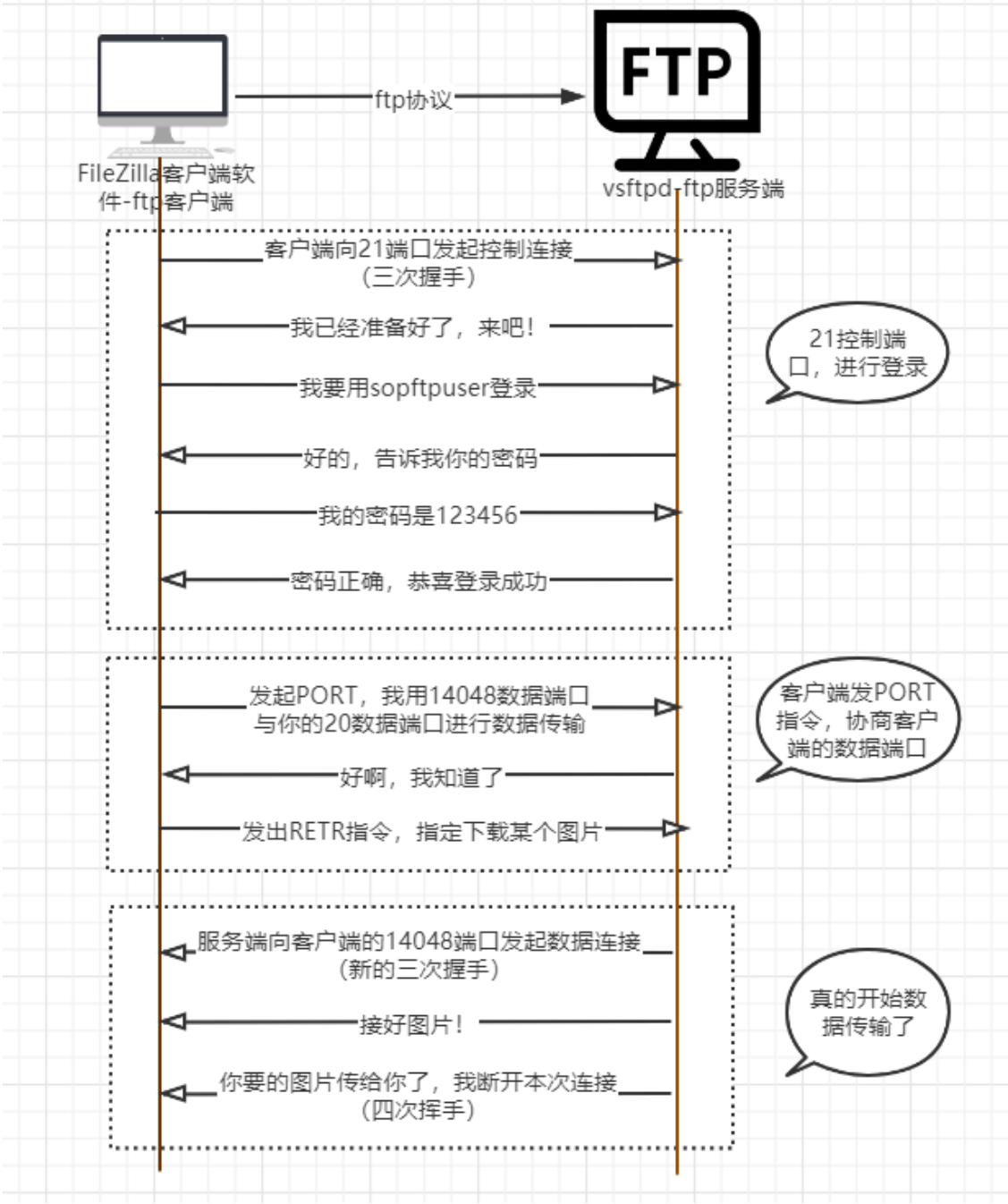
65、66、67号包：（服务端说）：三次握手，不同的是，**这次是服务端发起的。服务端用的是20端口，向客户端协商好的14048端接口发起连接**。

下面就是开始传输这张图片。

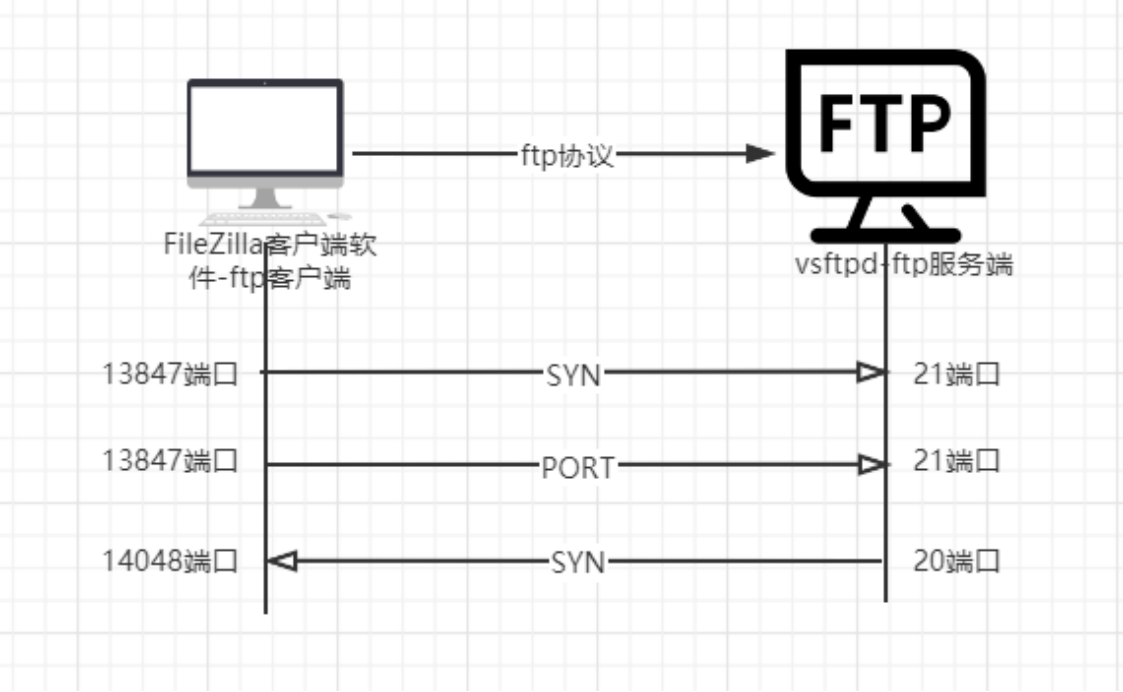
可以看到，在主动模式下，**客户端连接FTP服务器的21端口仅仅是为了传输控制信息，称为“控制连接”**。

当需要进行数据传输时，就会先协商好客户端一个数据端口，服务端用的是20端口，重新三次握手建立TCP连接，继而进行数据传输。文件下载结束，连接自动关闭。

此外需要注意的是，不光是操作文件如此，哪怕执行一个 `ls` 命令也是需要新建数据连接，不是一个高效的方式。以上的过程整理为：


ftp协议
FTP
FileZilla客户端软件-ftp客户端
vsftpd-ftp服务端
客户端向21端口发起控制连接
（三次握手）
我已经准备好了，来吧！
我要用sopftpuser登录
好的，告诉我你的密码
我的密码是123456
密码正确，恭喜登录成功
21控制端口，进行登录
发起PORT，我用14048数据端口与你的20数据端口进行数据传输
好啊，我知道了
发出RETR指令，指定下载某个图片
客户端发PORT指令，协商客户端的数据端口
服务端向客户端的14048端口发起数据连接
（新的三次握手）
接好图片！
你要的图片传给你了，我断开本次连接
（四次挥手）
真的开始数据传输了

简化理解就是：



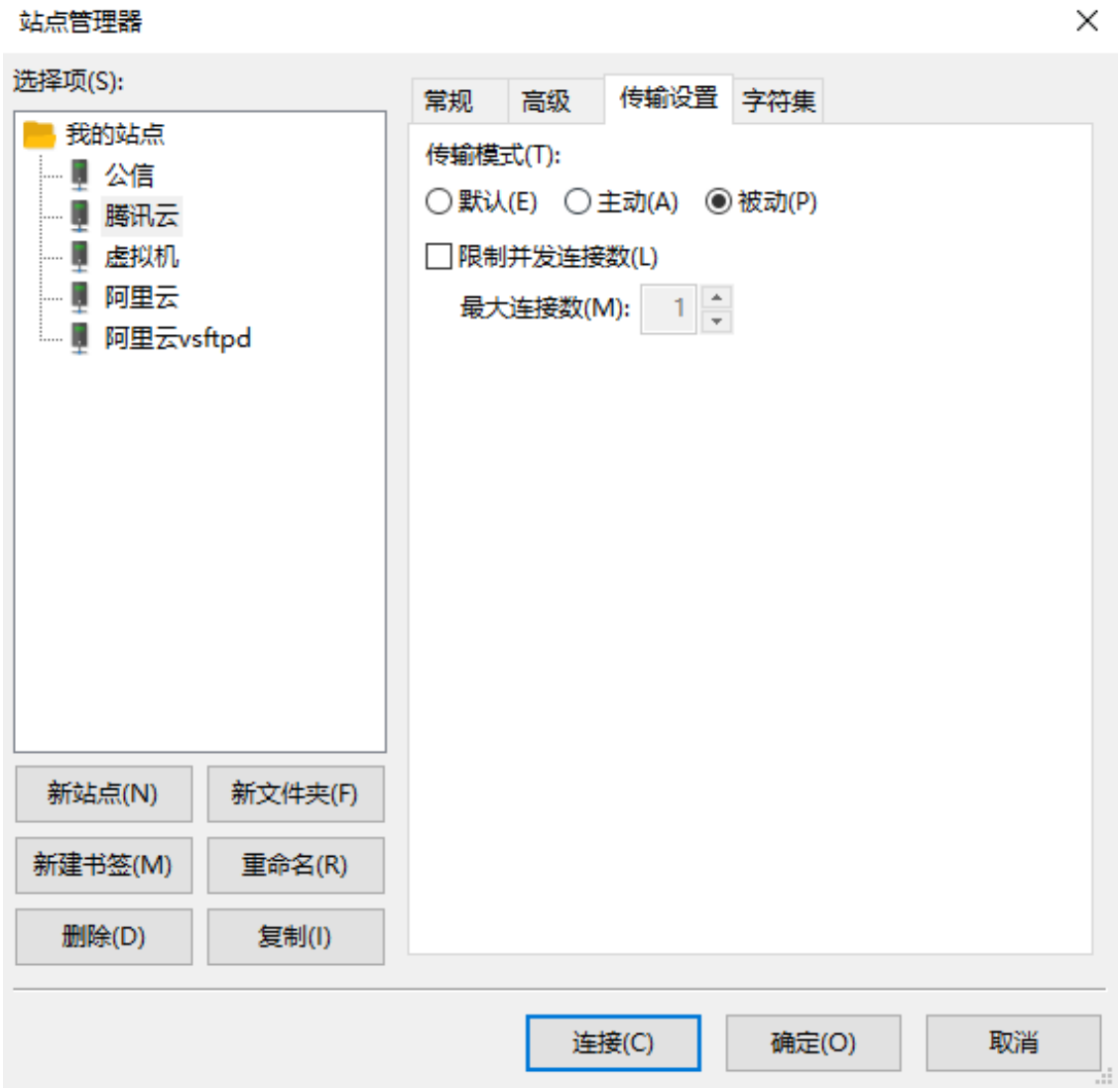
我们也可以看到，服务端固定用的是20端口进行数据传输，而客户端自己随机选择端口。这种模式实际上用的很少，因为数据连接的三次握手是服务端发起的，因此称为主动模式，不过客户端所在的主机如果防火墙阻挡了连接请求，传输就失败了。

这个时候可以试下被动模式。

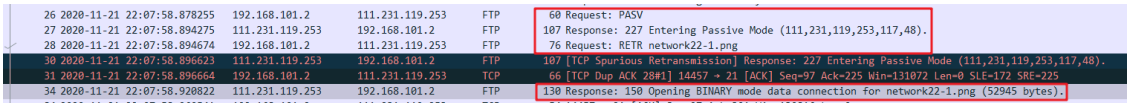
三、被动模式

为了解决服务器发起到客户端的连接问题，有了另一种 FTP 连接方式，即被动方式。**命令连接和数据连接都由客户端发起**，这样就解决了从服务器到客户端的数据端口的连接被防火墙过滤的问题。

首先我调整 `FileZilla` 客户端传输模式为被动模式。并进行抓包。



由于登录的过程跟上面是一样的，因此不再赘述，直接看下下载图片的过程。



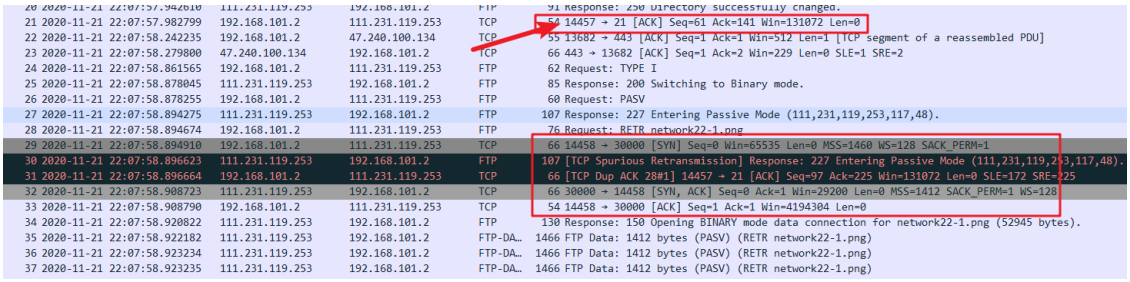
26号包：（客户端说）PASV指令表示让服务器在数据端口监听，进入被动模式。

27号包：（服务端说）发起PORT指令，你可以连接到111.231.119.253这个ip上来，端口我给你的是117*256+48=30000。

28号包：（客户端说）RETR指令，我想下载这个图片。

34号包：（服务端说）开始给你传输这个图片。

上面是协商好服务端的数据端口，下面就是建立新的数据连接开始真正的数据传输了，我们可以看下下面这个图：



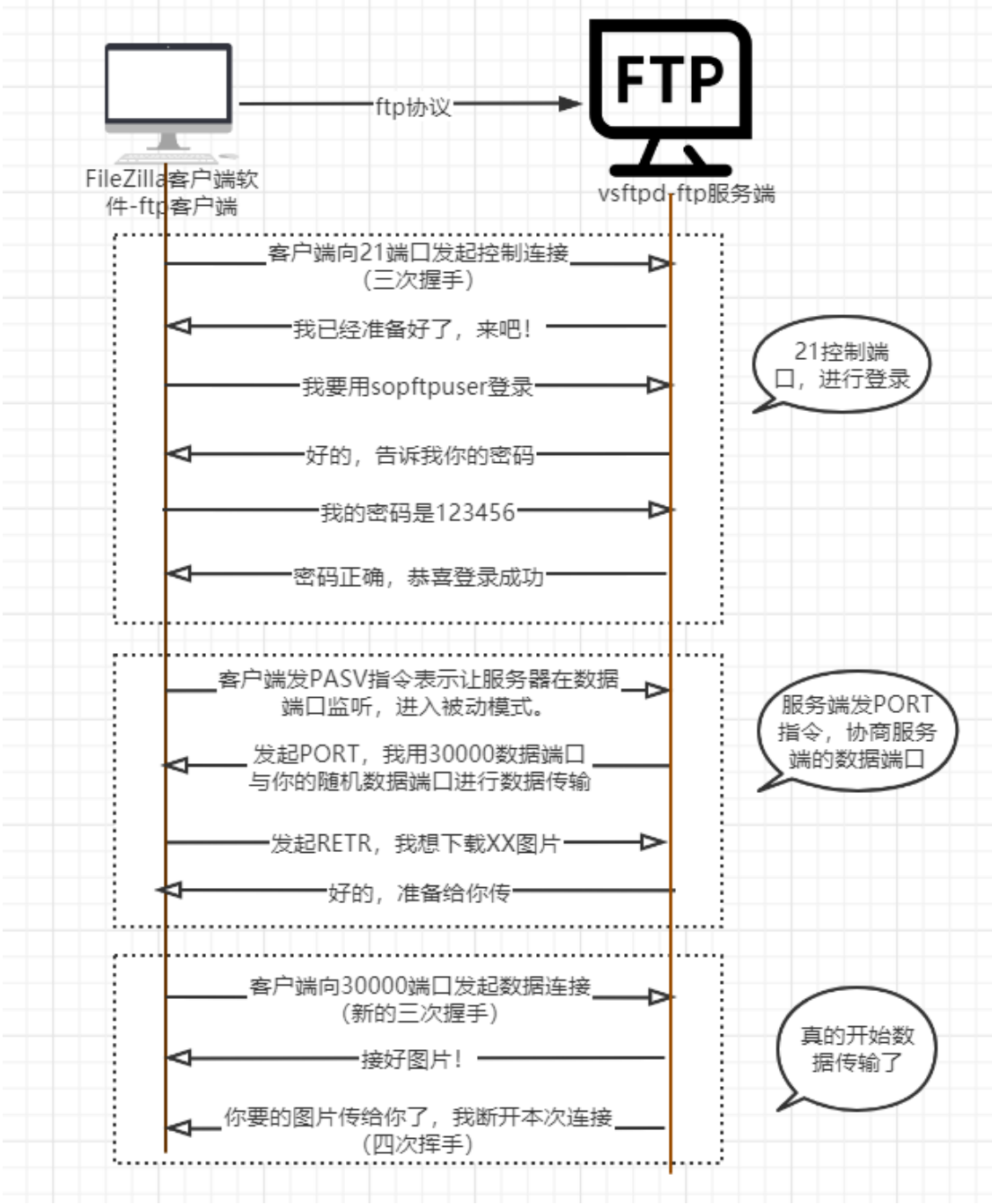
20	2020-11-21 22:07:57.942610	111.231.119.253	192.168.101.2	FTP	91 Response: 250 Directory successfully changed.
21	2020-11-21 22:07:57.982799	192.168.101.2	111.231.119.253	TCP	54 14457 → 21 [ACK] Seq=61 Ack=141 Win=131072 Len=0
22	2020-11-21 22:07:58.242235	192.168.101.2	47.240.100.134	TCP	55 13682 → 443 [ACK] Seq=1 Ack=1 Win=512 Len=1 [TCP segment of a reassembled PDU]
23	2020-11-21 22:07:58.279800	47.240.100.134	192.168.101.2	TCP	66 443 → 13682 [ACK] Seq=1 Ack=2 Win=229 Len=0 SLE=1 SRE=2
24	2020-11-21 22:07:58.861565	192.168.101.2	111.231.119.253	FTP	62 Request: TYPE I
25	2020-11-21 22:07:58.878045	111.231.119.253	192.168.101.2	FTP	85 Response: 200 Switching to Binary mode.
26	2020-11-21 22:07:58.878255	192.168.101.2	111.231.119.253	FTP	60 Request: PASV
27	2020-11-21 22:07:58.894275	111.231.119.253	192.168.101.2	FTP	107 Response: 227 Entering Passive Mode (111,231,119,253,117,48).
28	2020-11-21 22:07:58.894674	192.168.101.2	111.231.119.253	FTP	76 Request: RETR network22-1.png
29	2020-11-21 22:07:58.894910	192.168.101.2	111.231.119.253	TCP	66 14458 → 30000 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=128 SACK_PERM=1
30	2020-11-21 22:07:58.896623	111.231.119.253	192.168.101.2	FTP	107 [TCP Spurious Retransmission] Response: 227 Entering Passive Mode (111,231,119,253,117,48).
31	2020-11-21 22:07:58.896664	192.168.101.2	111.231.119.253	TCP	66 [TCP Dup ACK 28#1] 14457 → 21 [ACK] Seq=97 Ack=225 Win=131072 Len=0 SLE=172 SRE=225
32	2020-11-21 22:07:58.908723	111.231.119.253	192.168.101.2	TCP	66 30000 → 14458 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1412 SACK_PERM=1 WS=128
33	2020-11-21 22:07:58.908790	192.168.101.2	111.231.119.253	TCP	54 14458 → 30000 [ACK] Seq=1 Ack=1 Win=4194304 Len=0
34	2020-11-21 22:07:58.920822	111.231.119.253	192.168.101.2	FTP	130 Response: 150 Opening BINARY mode data connection for network22-1.png (52945 bytes).
35	2020-11-21 22:07:58.922182	111.231.119.253	192.168.101.2	FTP-DA…	1466 FTP Data: 1412 bytes (PASV) (RETR network22-1.png)
36	2020-11-21 22:07:58.923234	111.231.119.253	192.168.101.2	FTP-DA…	1466 FTP Data: 1412 bytes (PASV) (RETR network22-1.png)
37	2020-11-21 22:07:58.923235	111.231.119.253	192.168.101.2	FTP-DA…	1466 FTP Data: 1412 bytes (PASV) (RETR network22-1.png)

我们可以看到，客户端发起控制连接的时候随机端口用的是14457，下面进行数据连接的时候端口用的是14458，相差1。这里其实就是这个规律：

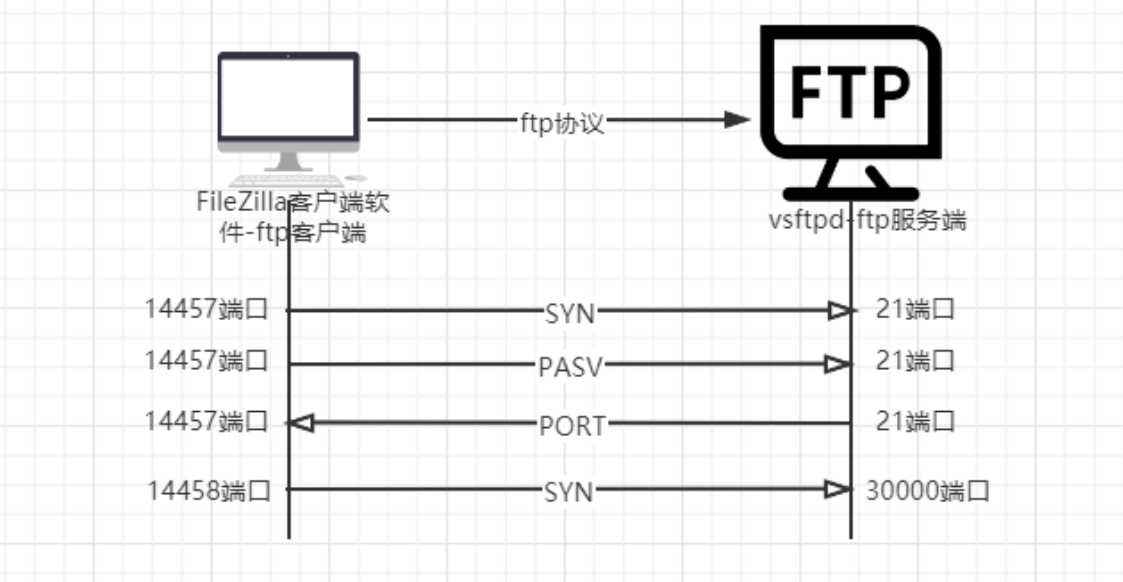
> 被动模式下，当开启一个 FTP 连接时，客户端打开两个任意的本地端口（N > 1024 和 N+1）。

29、32、33号包：三个包是数据连接的三次握手，由客户端发起。下面进行数据传输。

以上的过程归纳为：


ftp协议
FTP
FileZilla客户端软件-ftp客户端
vsftpd-ftp服务端
客户端向21端口发起控制连接（三次握手）
我已经准备好了，来吧！
我要用sopftpuser登录
好的，告诉我你的密码
我的密码是123456
密码正确，恭喜登录成功
21控制端口，进行登录
客户端发PASV指令表示让服务器在数据端口监听，进入被动模式。
发起PORT，我用30000数据端口与你的随机数据端口进行数据传输
发起RETR，我想下载XX图片
好的，准备给你传
服务端发PORT指令，协商服务端的数据端口
客户端向30000端口发起数据连接（新的三次握手）
接好图片！
你要的图片传给你了，我断开本次连接（四次挥手）
真的开始数据传输了

主动模式的简化模式为：



可以看到这个30000端口实际上是我在FTP配置文件中配置的30000-30010中的一个端口。

```
listen_ipv6=YES

pam_service_name=vsftpd
userlist_enable=YES
tcp_wrappers=YES

# newadd 20201121
userlist_file=/etc/vsftpd/user_list
userlist_deny=NO
user_sub_token=$USER
local_root=/home/sopftpuser/ftp

allow_writeable_chroot=YES

pasv_min_port=30000
pasv_max_port=30010
```

被动模式下服务端设置的端口范围

总结下本文核心要讲的主动模式和被动模式：

```
服务器主动连接客户端（主动模式）：

                客户端端口                              服务器端口
    命令连接 ： 随机 ----------------------------> tcp 21
    数据连接 ： 随机 <---------------------------- tcp 20

客户端主动连接服务器（被动模式）：

                客户端端口                              服务器端口
    命令连接 ： 随机 ----------------------------> tcp 21
    数据连接 ： 随机 ----------------------------> 随机

服务器被动模式数据端口：
    227 Entering Passive Mode (10,0,0,4,165,41).
    被动模式，服务端 随机端口号计算: 165 * 256 + 41 = 7661
```

所以在被动模式下（当然了，一般都会使用被动模式而不是主动模式），如果你是网络管理员，在申请端口白名单时，**可不能仅仅放开21端口哦**，被动端口也都要放开，否则无法正常连接FTP，比如我的云服务器上的安全组（可以理解为上层防火墙）的放开规则中，包含了21和这个被动端口范围：

::/0	TCP:80,18888	允许	放通Web服务HTTP（80），如Apache、Nginx	2021-04-21 09:22:36	编辑 插入 删除
0.0.0.0/0	TCP:30000,30001,30002,30003,30004,30005,30006,30007,30008,30009,30010	允许	ftp被动模式端口	2020-11-21 16:17:56	编辑 插入 删除
0.0.0.0/0	UDP:8301,8302,8600	允许		2020-11-08 22:02:49	编辑 插入 删除
0.0.0.0/0	TCP:8500,8300,8301,8302,8600,389	允许	consul-server访问端口	2021-02-02 21:48:15	编辑 插入 删除
0.0.0.0/0	TCP:22,23	允许	放通Linux SSH登录	2021-03-01 19:41:01	编辑 插入 删除
0.0.0.0/0	TCP:21	允许	21端口临时开放	-	编辑 插入 删除
0.0.0.0/0	TCP:3306,3307	允许	mysql	2021-04-21 09:22:49	编辑 插入 删除