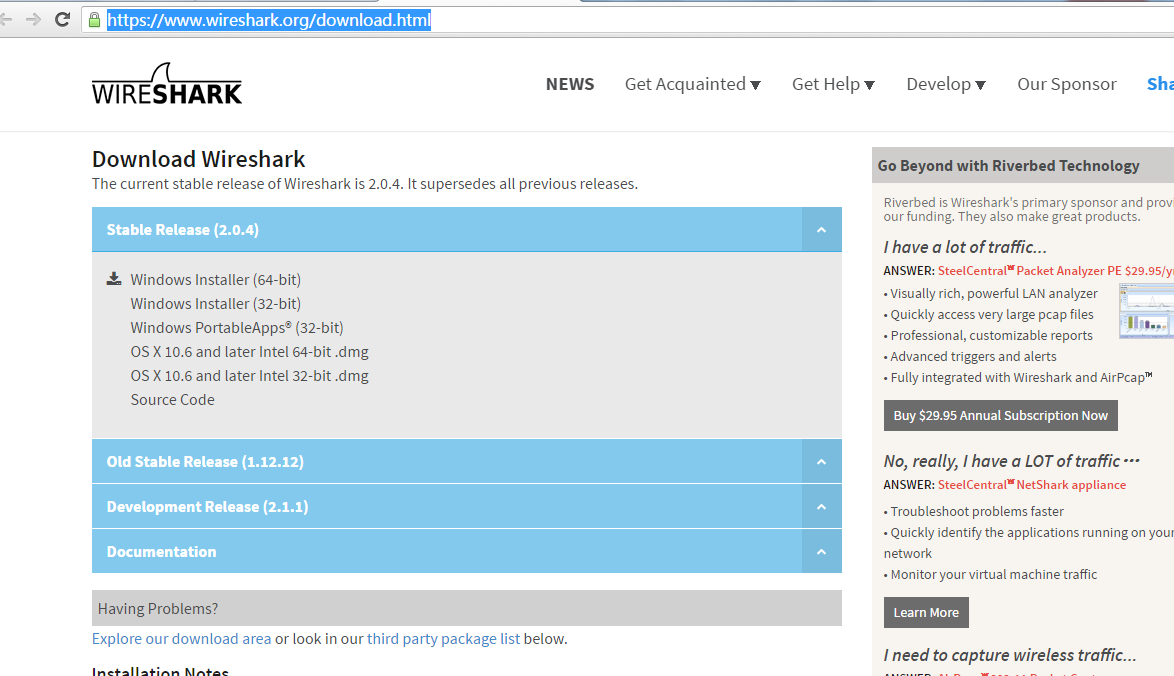
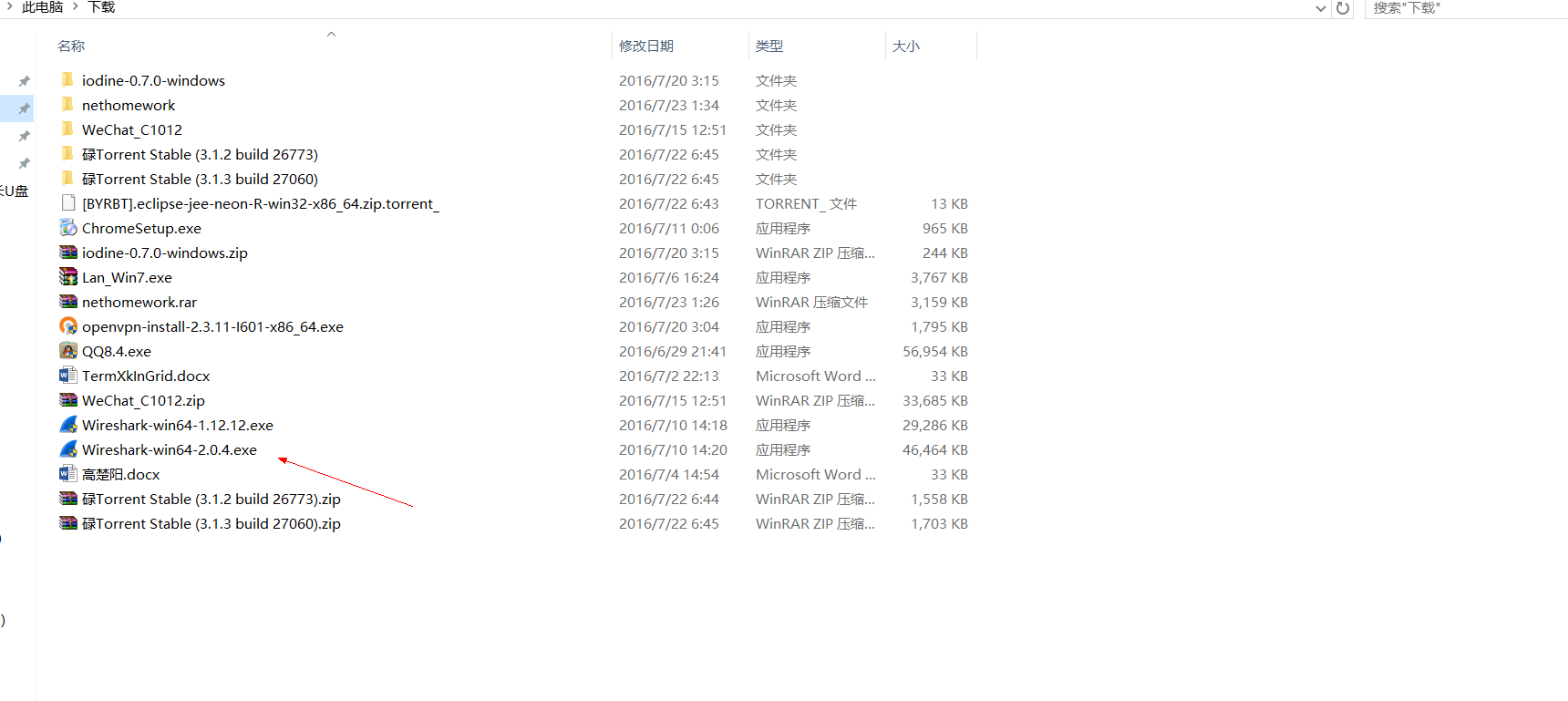
第一题:

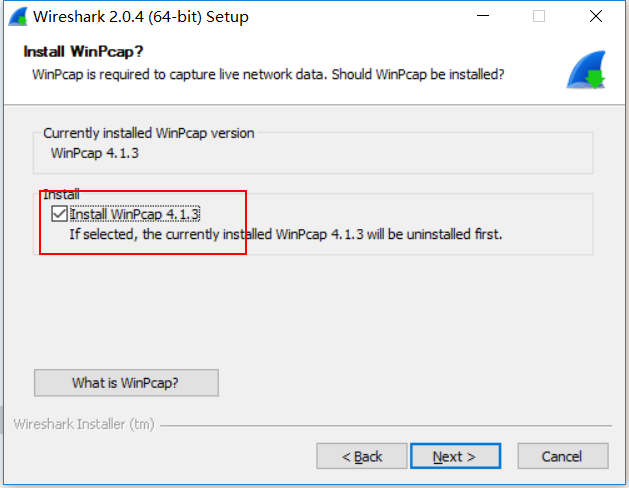
1. wireshark的安装

使用环境win7—64位,在官网https://www.wireshark.org/download.html下载wireshark2.0.4压缩包

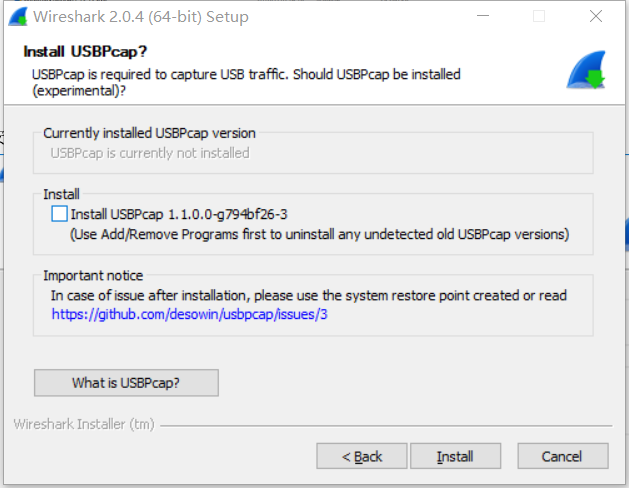




安装时要注意一定要安装上WinPcap，这样才能实时的抓取网络中的数据包



不需要安装USBPcap

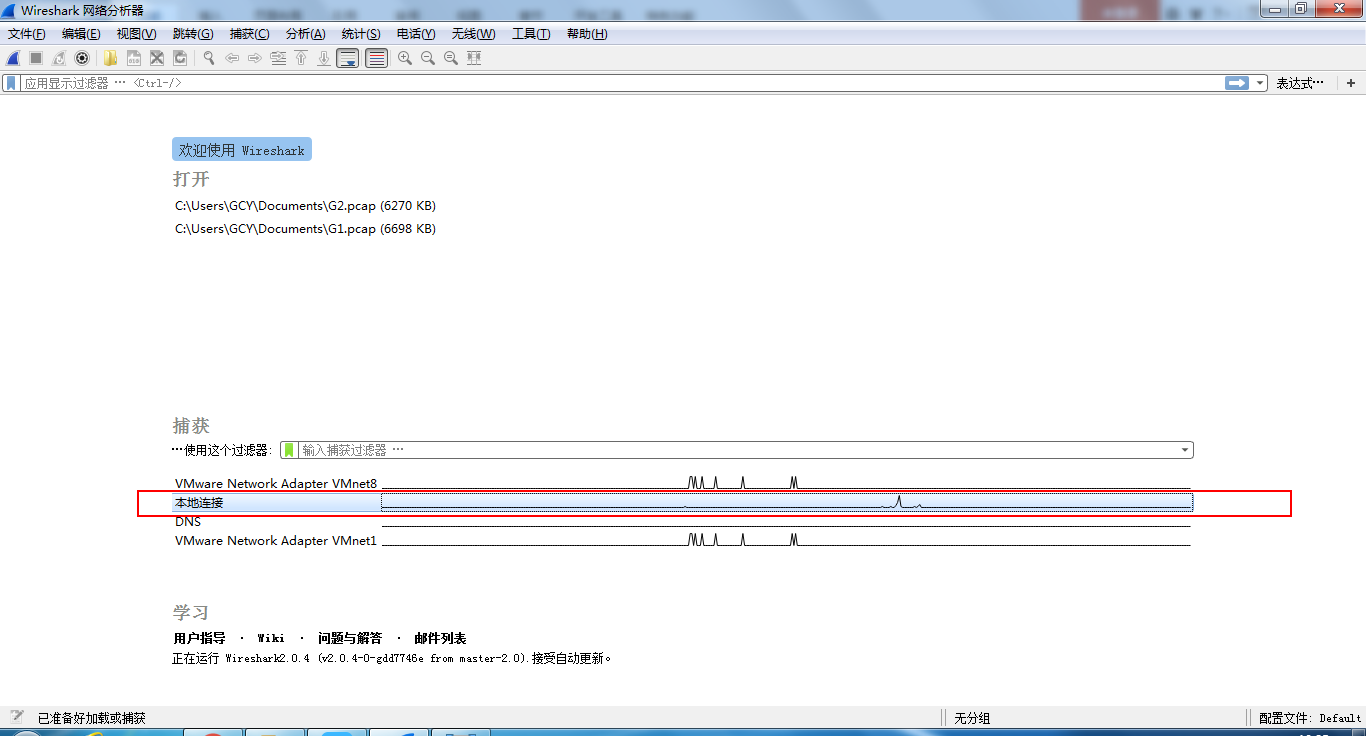


并且把wireshark的主要工具都安装上



1. Wireshark的使用方法

由于需要抓取本机在互联网上产生流量的数据包，在捕获部分应该选择本地连接网卡



为了查看首包更加方便，需要在wireshark的列中增加一条Delta，打开编辑->首选项->外观->列，在右边首选项增加一条记录Delta，类型选择delta time displayed



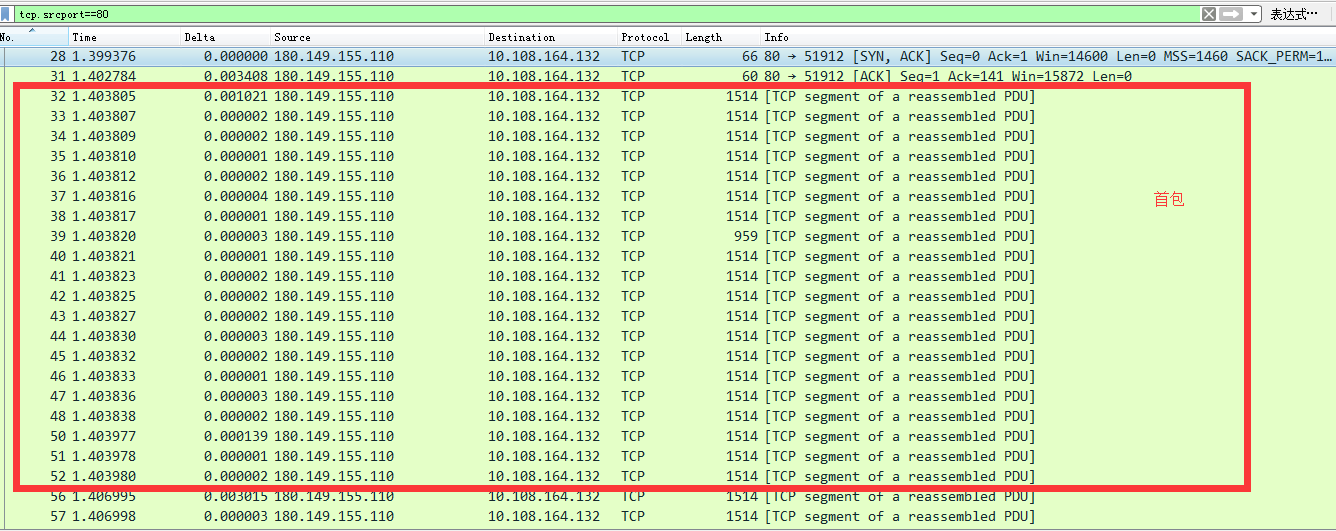
3下载wget进行抓包，wget是一个从网络上自动下载文件的自由工具，从http://soft.hao123.com/soft/appid/8311.html下载解压后在D:\UnxUtils\usr\local\wbin下面。

使用命令行cd D:\UnxUtils\usr\local\wbin 命令，进入目录然后依次下载URL1，URL2，URL3

1)下载url1，并对其进行抓包



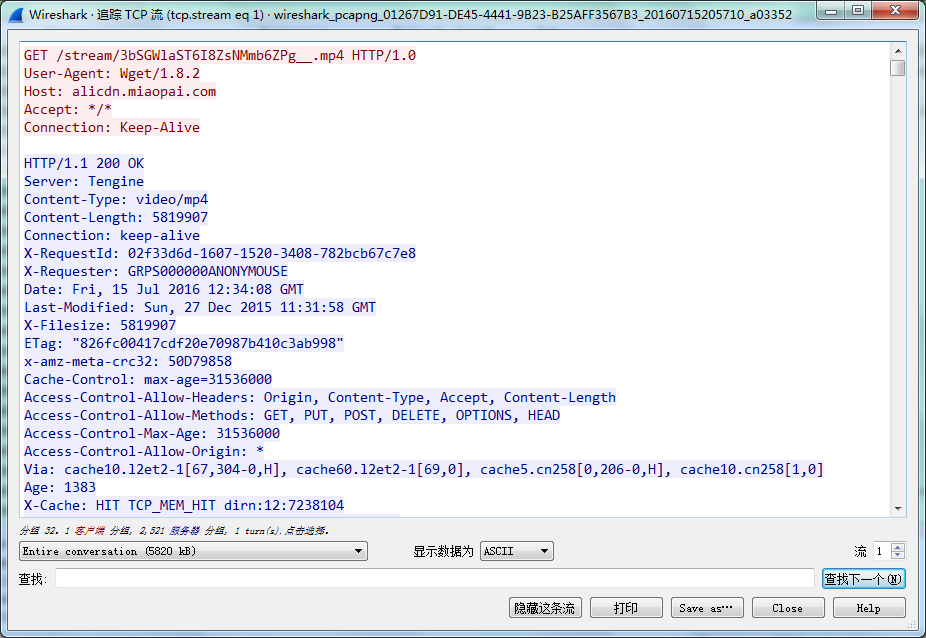
2）在过滤器中输入tcp.srcport==80,把80端口的TCP协议数据包过滤出来



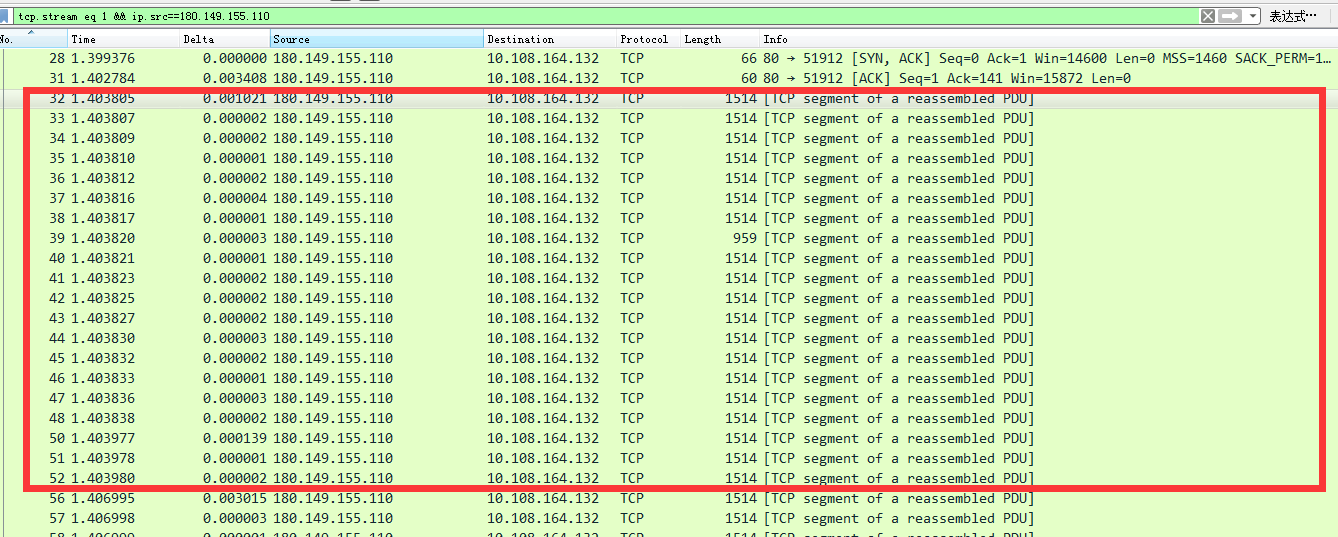
3）右键单击source=URL1并且是给本机IP发送的数据包，选择追踪流中的TCP流



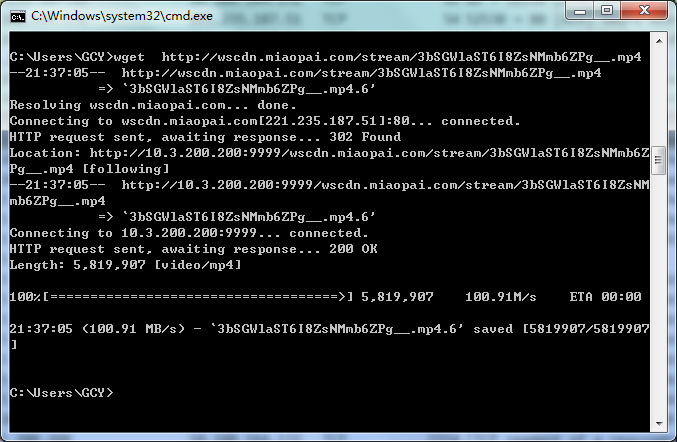
4）之后会显示如下对话框



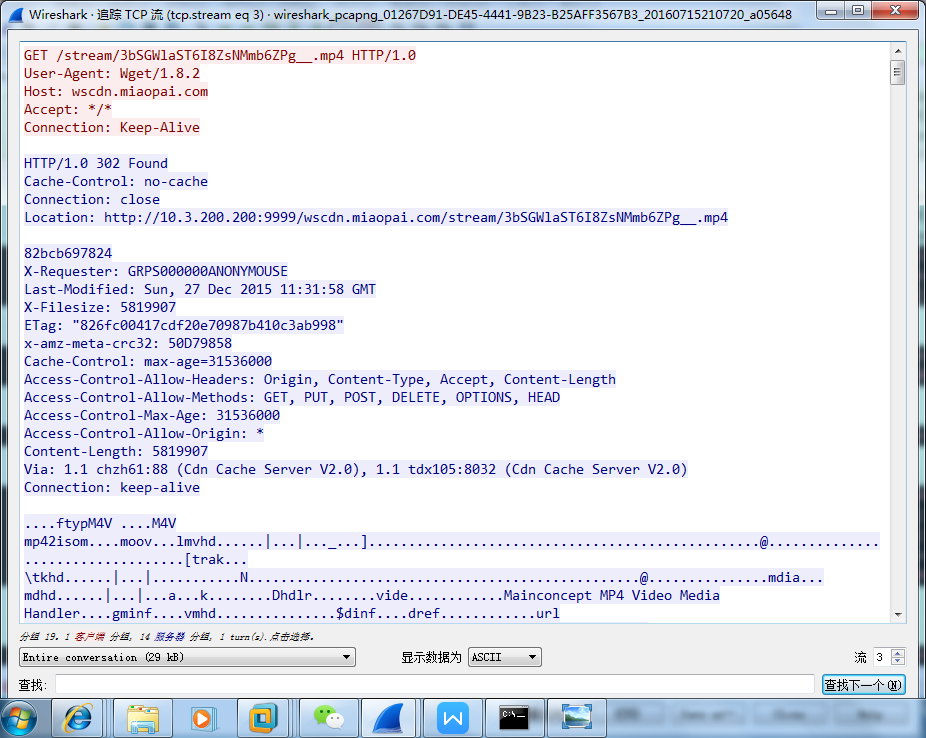
5）关闭对话框，在过滤器中的tcp.stream eq x后面输入&&ip.src==服务器IP地址，将服务器返回的数据包抓取出来，如下图，从图中可以看出RTT时间是3.4ms，首包数量是20，首包为握手成功后，即第一个RTT建立连接之后，下一个RTT时间内服务器给本发送的数据包的数量。



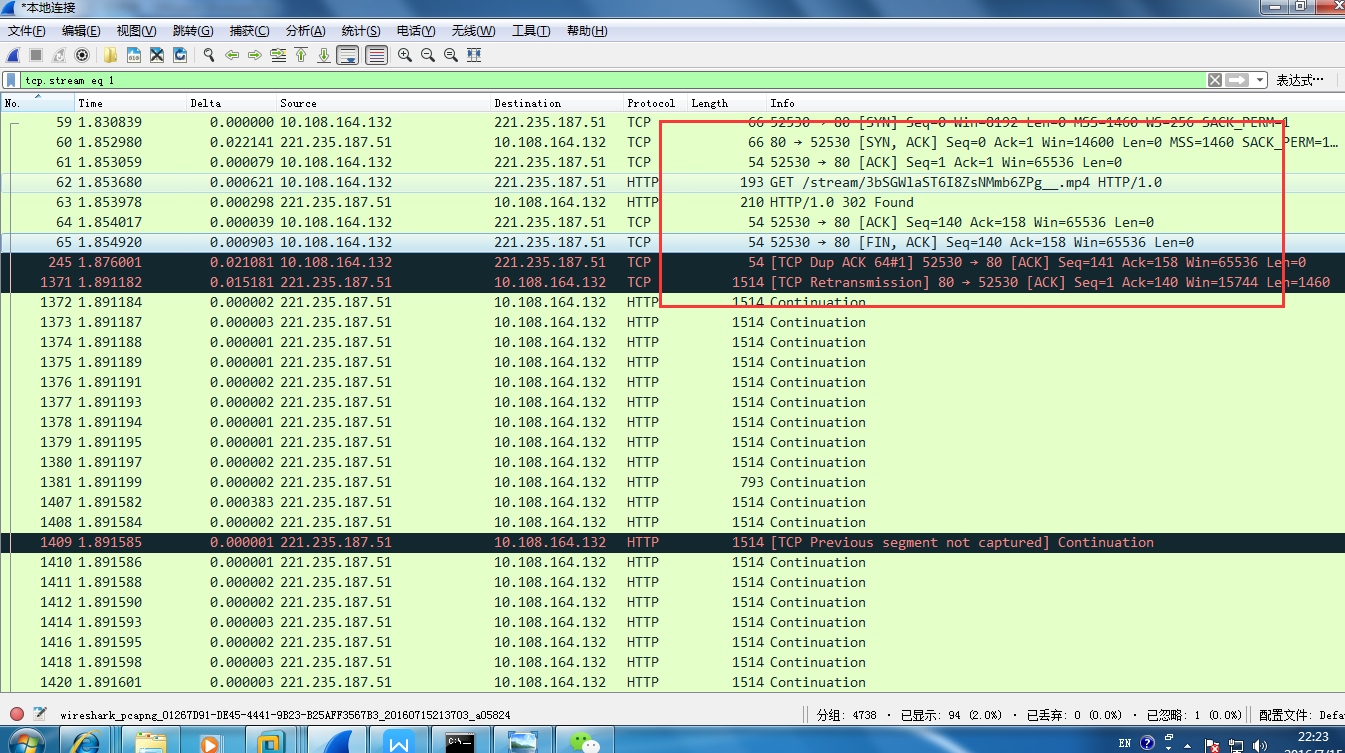
1. 依照上面的步骤对url2，使用wget下载



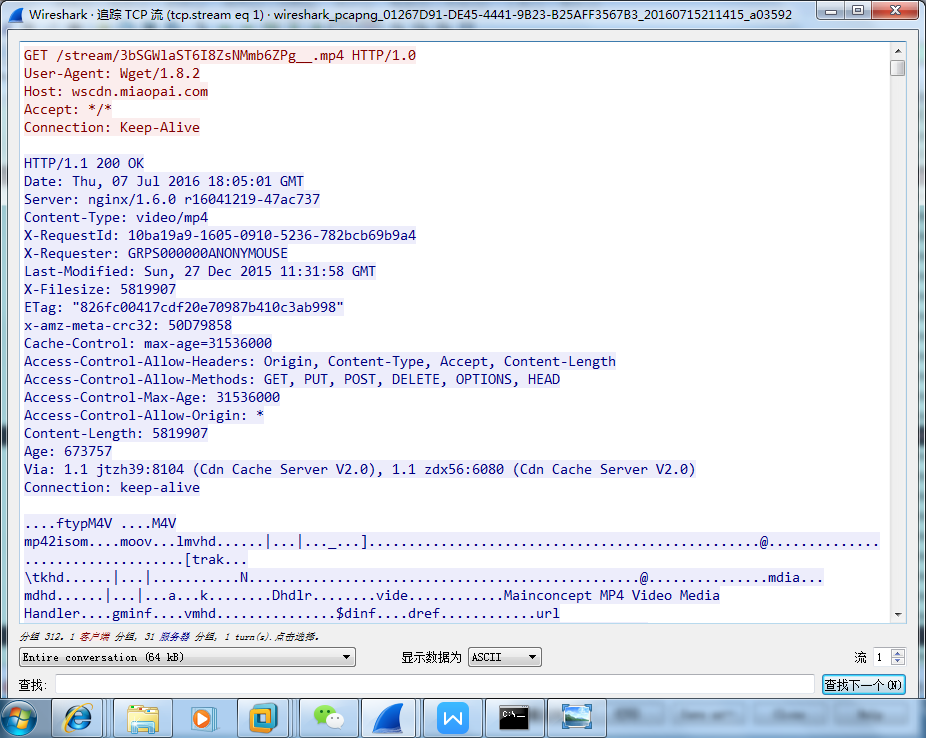
1. 使用tcp.srcport==80，并追踪流之后发现，发生了302劫持



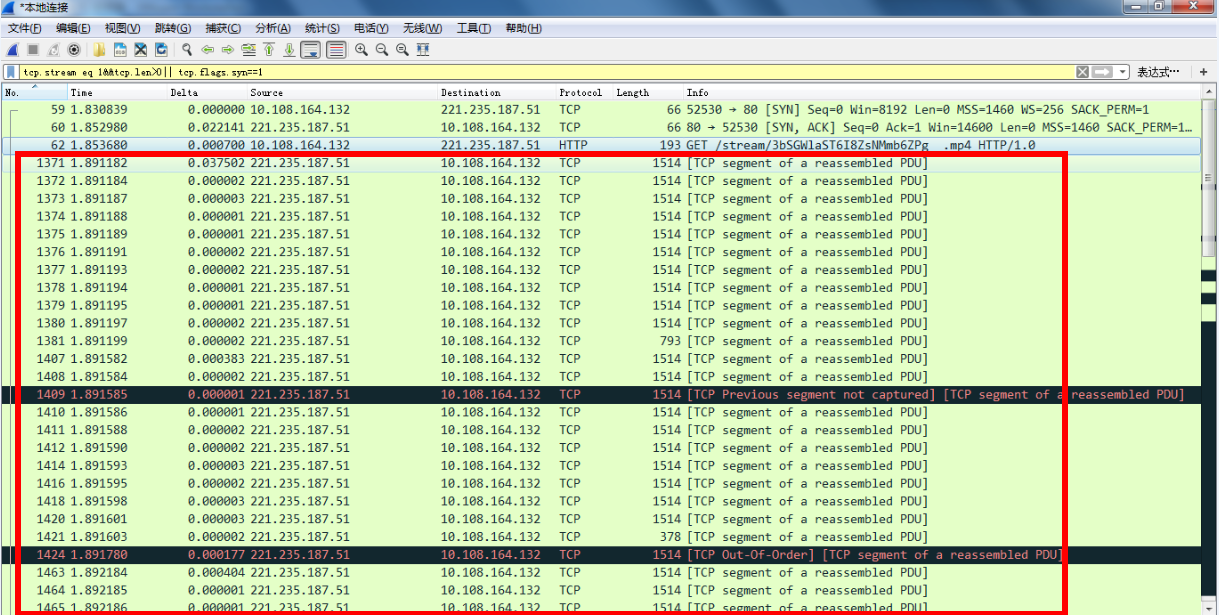
8）使用白金提出的ctrl+D删除下图红框内从HTTP GET请求包（包括它）向下数的4个10.3.200.200劫持的数据包的方法，屏蔽掉302劫持

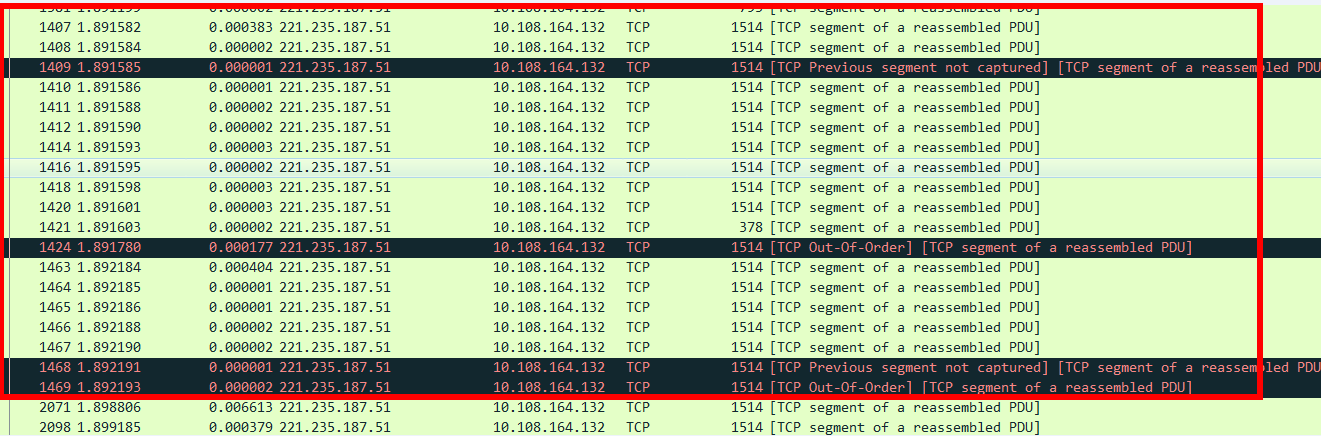


1. 删除完之后发现HTTP continuation包全部消失，并且HTTP 302消息也变成了HTTP 200

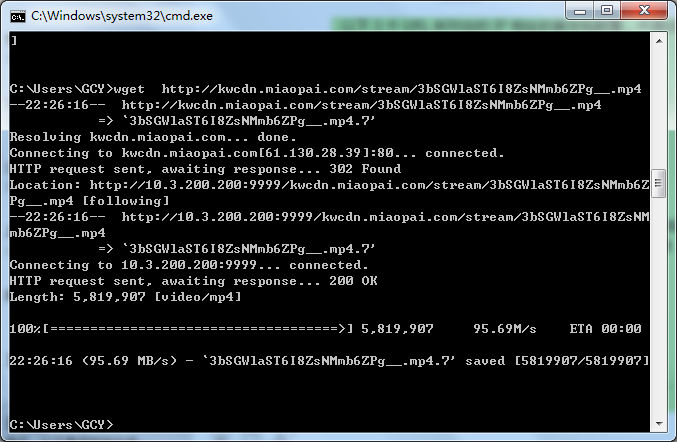


1. 在过滤器中的tcp.stream eq x后面输入&&tcp.len>0||tcp.flags.syn==1,过滤出本机与url2之间的数据包通信，并数出红框里的首包，首包数为30

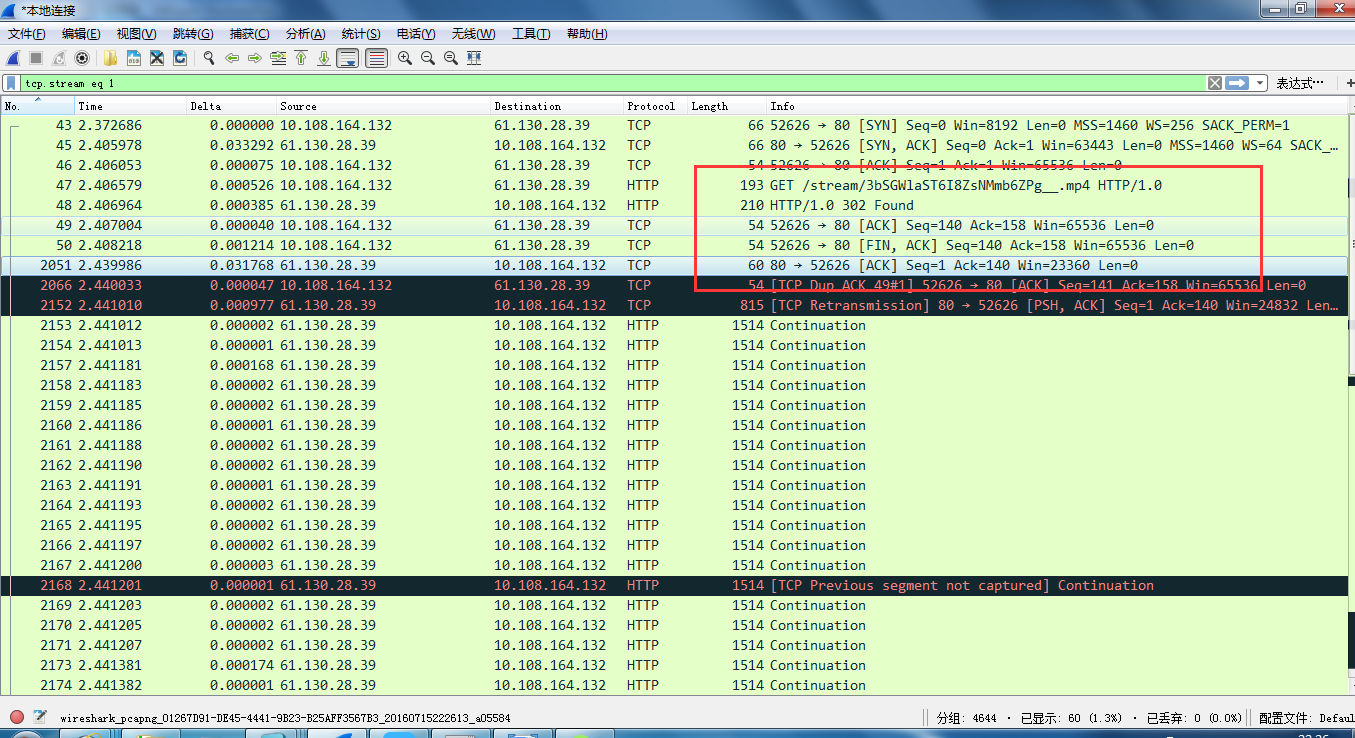


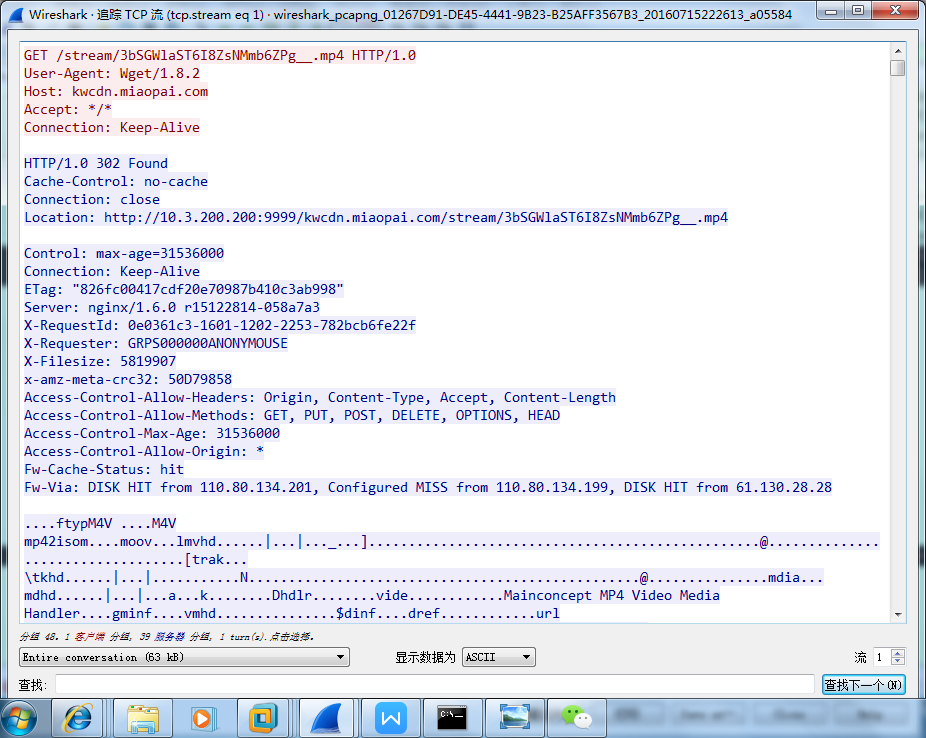


1. 使用wget对url3进行下载，和URL2一样也发生了劫持，方法与URL2一样，过滤掉劫持并得到首包，具体过程如下图

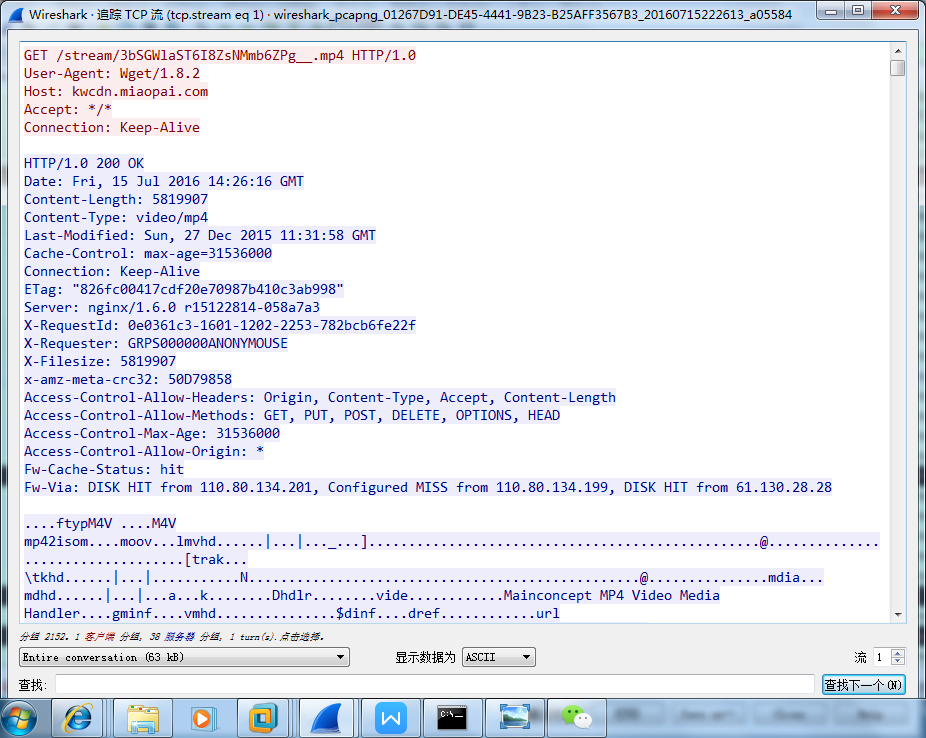


删除红框内的内容，实现过滤劫持

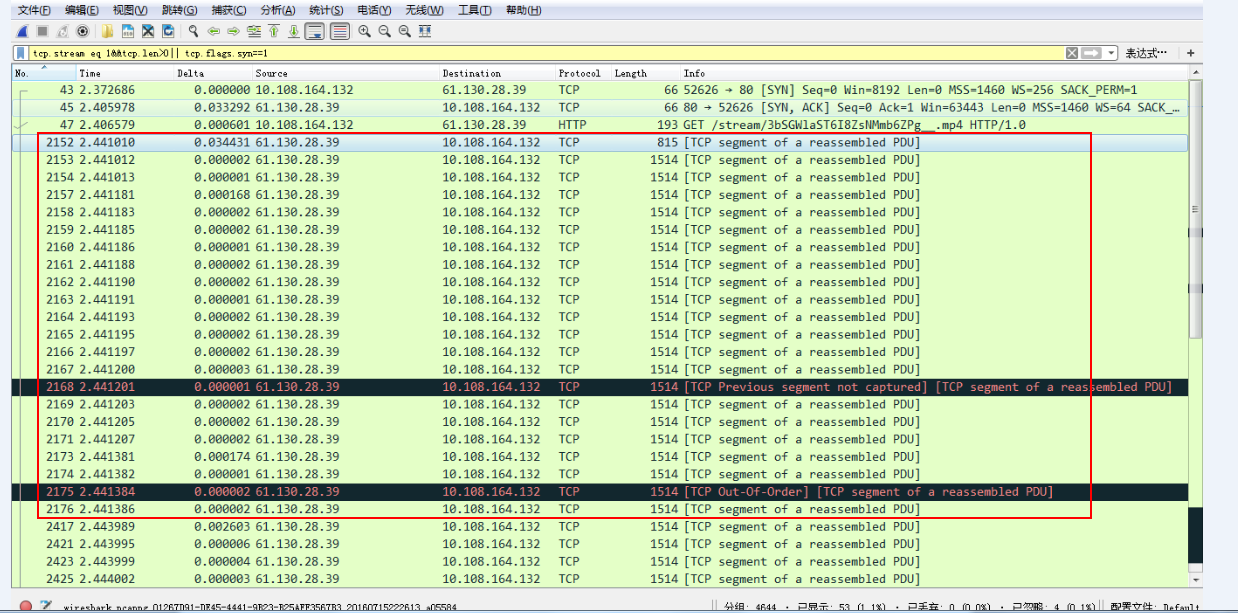




过滤掉302劫持后的200信息



输入过滤指令，然后数出首包数量为22



第二题：

Server端不支持WSale选项，那么服务端每次连接最大发送字节数是65535byte，而C/S双方协商MSS大小为1460，那么服务端一次给客户端最多发送65535/1460=44个数据包，

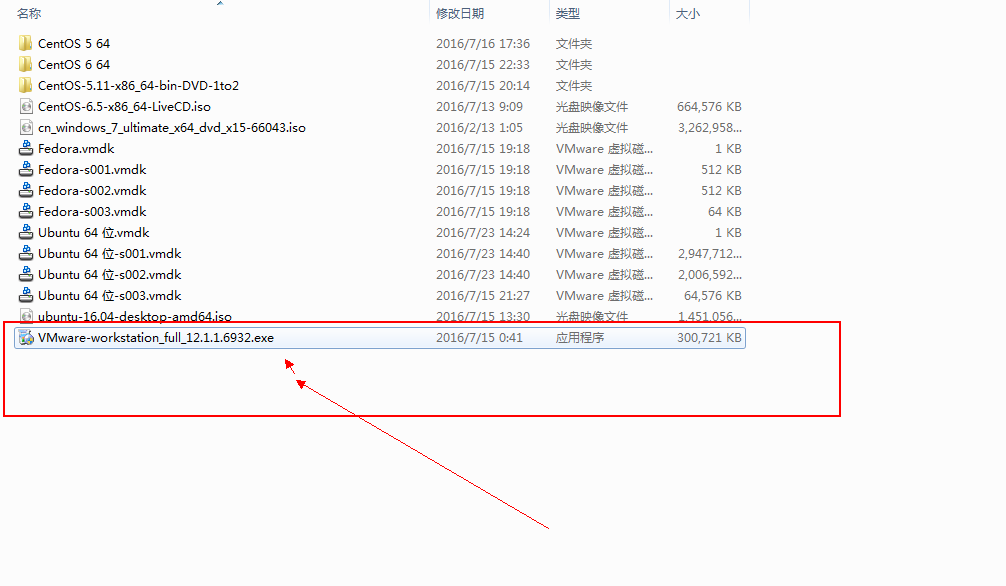
128KB的数据为128\*1024/1460=90,90=13+26+44+7，也就是服务端至少要给客户端发送四次数据，第一次发13个包，第二次发26个包，由于Server端不支持WSale选项，那么第三次只能发送44个包，第四次发送7个包，才能把数据发送完毕，加上第一次建立连接需要的时间，一共需要5次RTT时间才能将128KB数据传输完毕，由于RTT等于50ms，所以理论最快时间为250ms。

第三题：

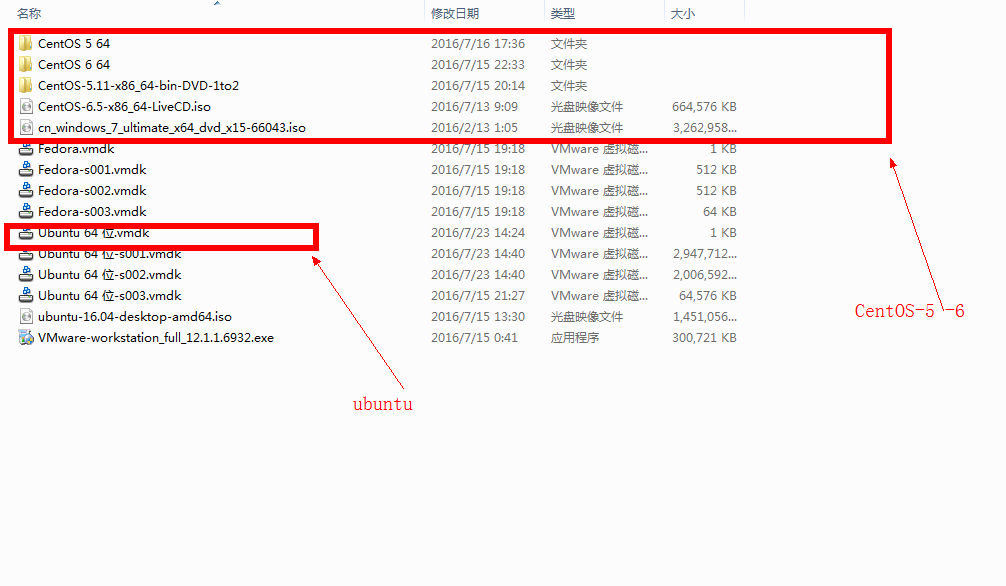
我使用的操作系统为WIN7，在虚拟机里安装了CentOS-5，CentOS-6作为被测试端，Ubuntu16-04为测试端，测试WIN7、CentOS-5，CentOS-6的重传次数，重传间隔。

1. 虚拟机和Linux操作系统的安装

1）下载VMware-workstationpro12安装包，并进行安装。



1. 下载CentOS-5，CentOS-6，Ubuntu16-04镜像或者克隆文件



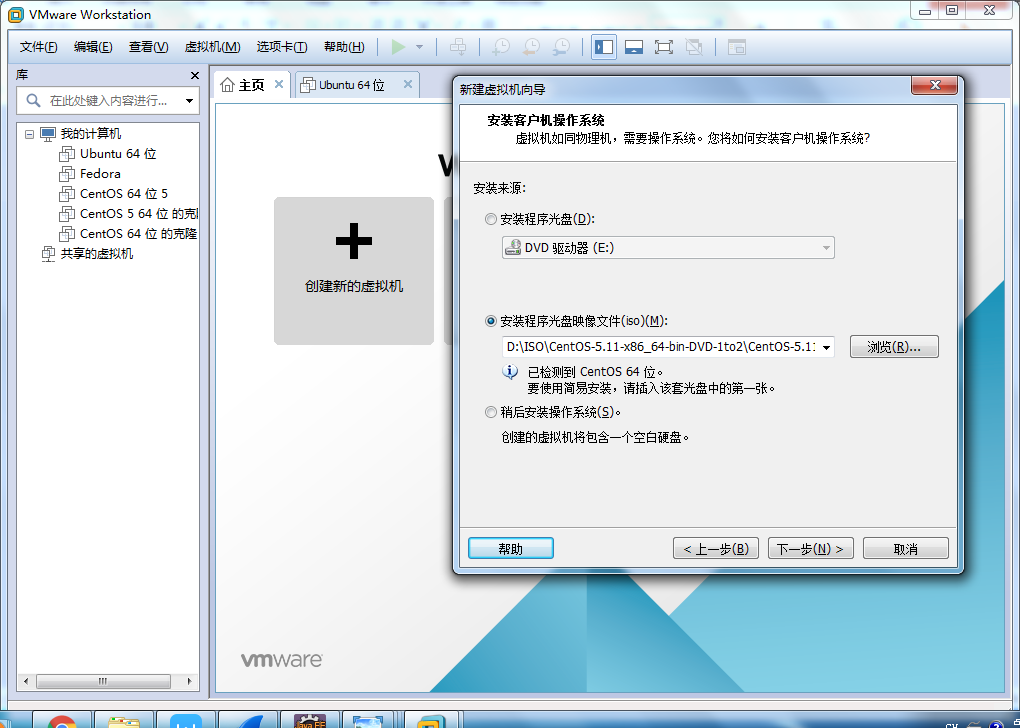
1. 使用VM-ware虚拟机加载镜像或者克隆

如果是加载克隆，就单击打开虚拟机

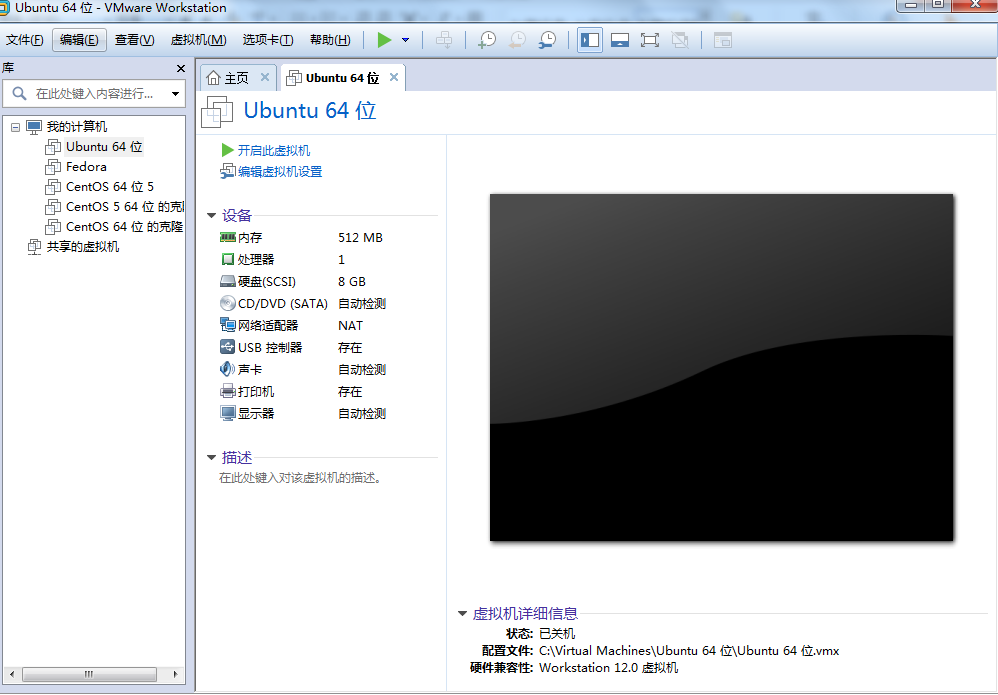


然后选择需要的VMDK格式的克隆文件，打开克隆系统

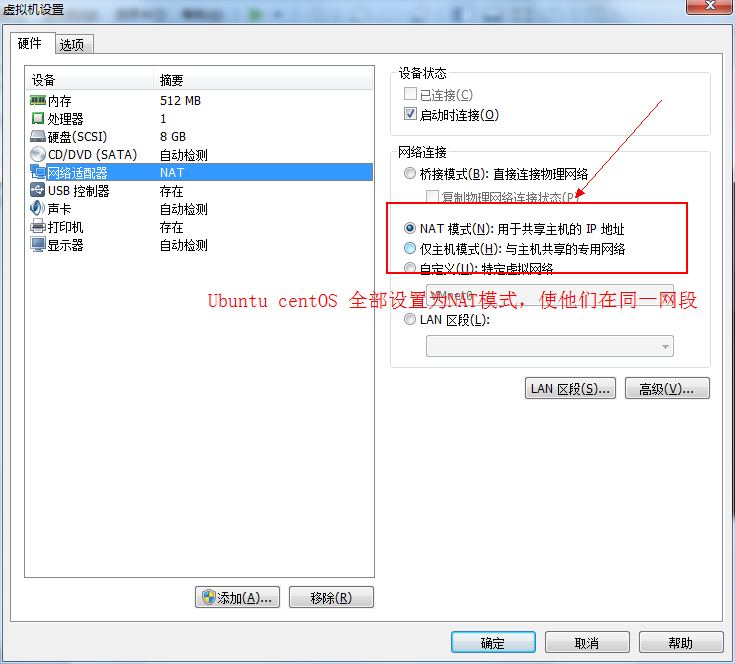
如果是加载镜像，就单击创建新的虚拟机，并选择镜像的位置，之后虚拟机会自动安装。



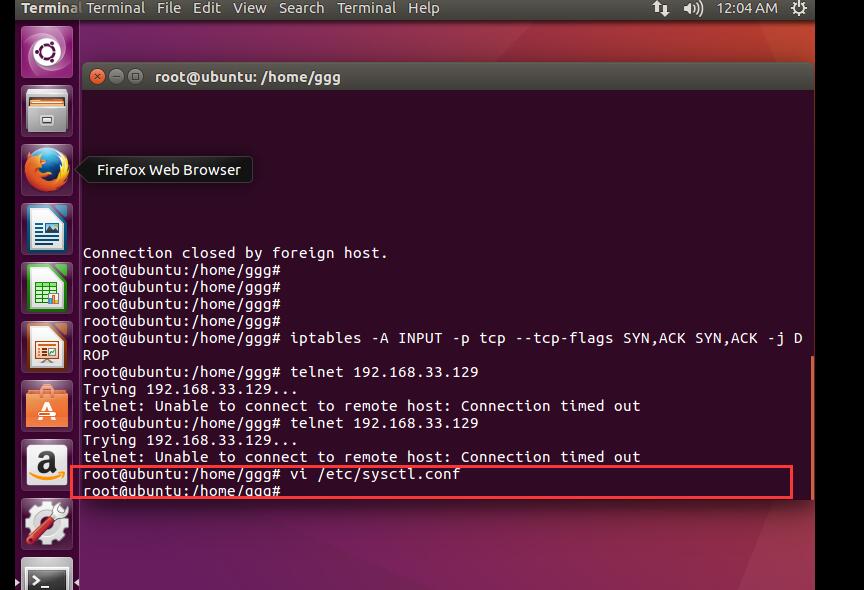
安装好的结果



1. 进入ubuntu系统终端，开始测试重传特性
2. 在终端使用sudo su 命令，输入密码，暂时获得ubuntu root权限，然后将ubuntu 、CentOS-5,CentOS-6虚拟机的网络适配器全部改成NAT模式，使他们在同一个局域网内。

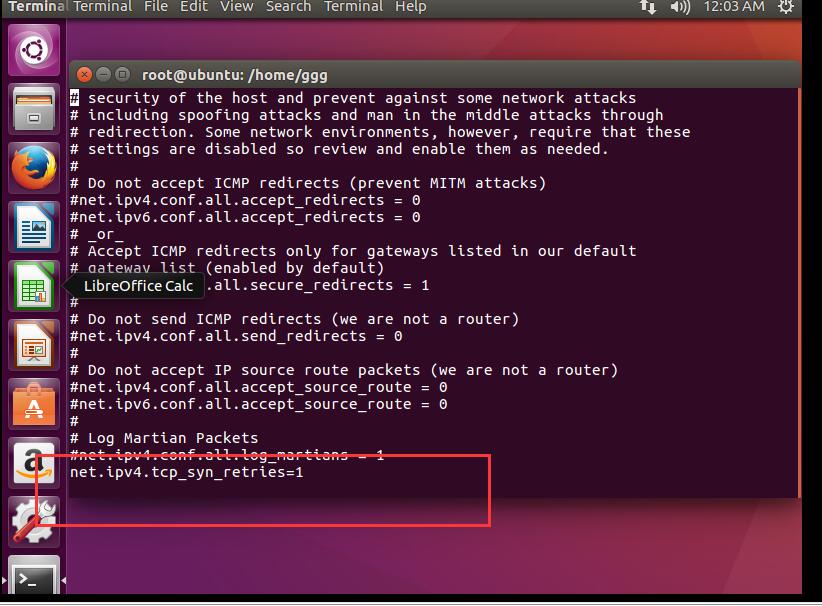


1. 使用vi /etc/sysctl.conf 命令打开Ubuntu系统内核配置文件

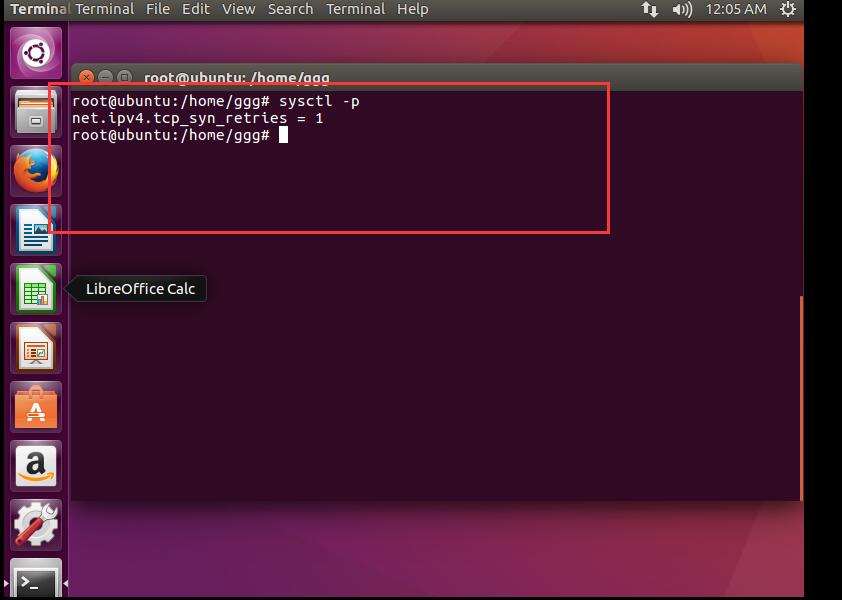


1. 在vi编辑器中，敲i键进入编辑模式，vi有两个模式：一个是编辑模式一个是命令模式。点击i可以从命令模式进入编辑模式，在点击esc键可以重新进入命令模式。我们一般进入编辑模式，来进行添加，修改，删除。但是当我们删除和修改的内容过多的时候，我们使用命令行模式，进行修改，这样方便，快捷，而命令行中，最常用到的是x，dd，u，p这四个命令：x:删除当前字符；dd：删除当前行；u:恢复前一步操作；p:复制之前删除的行。

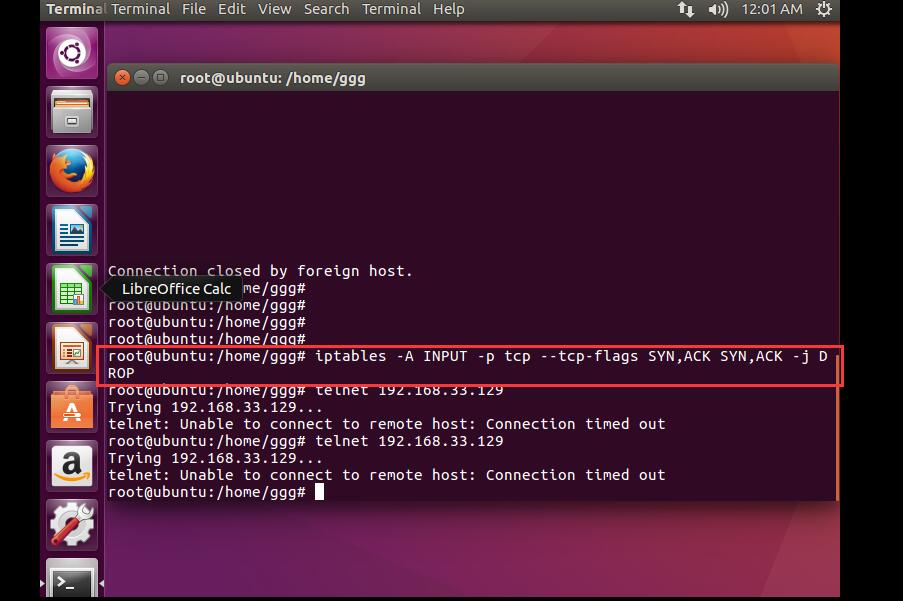
进入编辑模式后，输入net.ipv4.tcp\_syn\_retries = 1，然后按ESC回到命令行模式，敲：wq保存并退出。



1. 输入sysctl -p 命令，修改内核参数马上生效

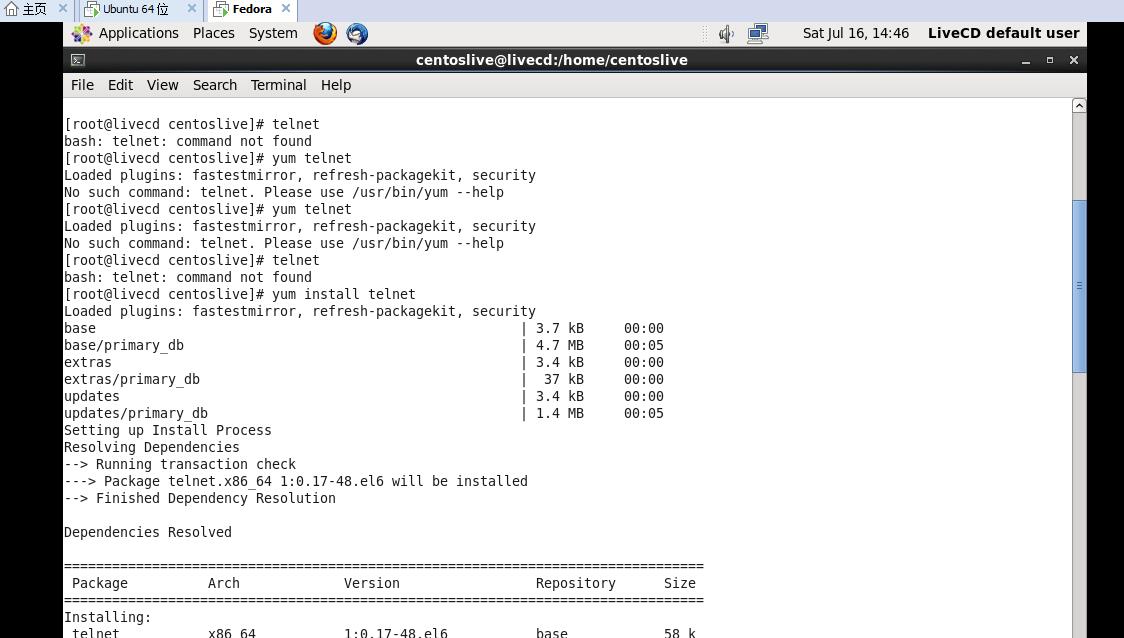


1. 配置iptables防火墙过滤syn，ack，使用命令iptables -A INPUT -p tcp --tcp-flags SYN,ACK SYN,ACK -j DROP。

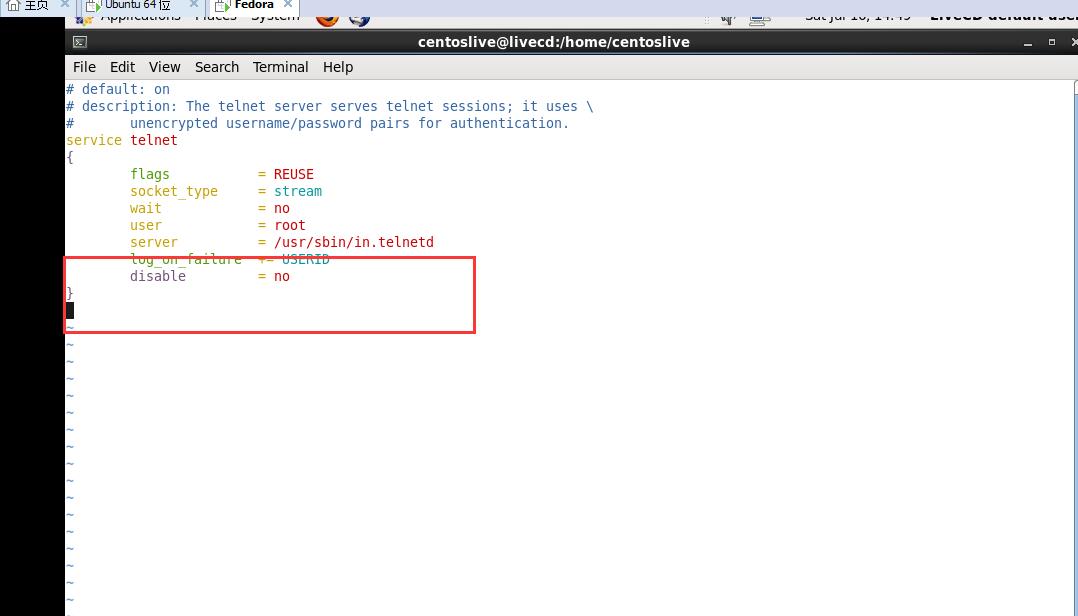


1. 打开CentOS-6虚拟机，使用su root，输入用户名，密码，获取centOS-6 root 权限，

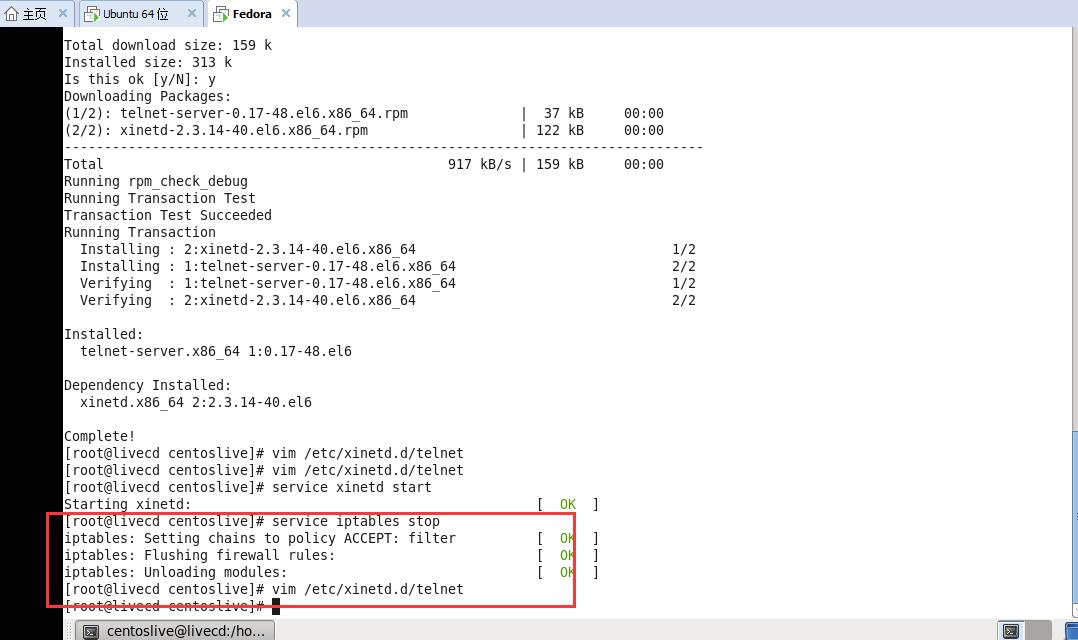
使用，使用yum install telnet-server分别安装telnet服务端



1. 使用vim /etc/xinetd.d/telnet命令，用VIM编辑器打开Telnet配置文件，改成disable = no，允许外面的电脑通过telnet连入CentOS，是用i进入INSERT模式进行修改，敲ESC，进入命令行模式，敲：wq保存并退出

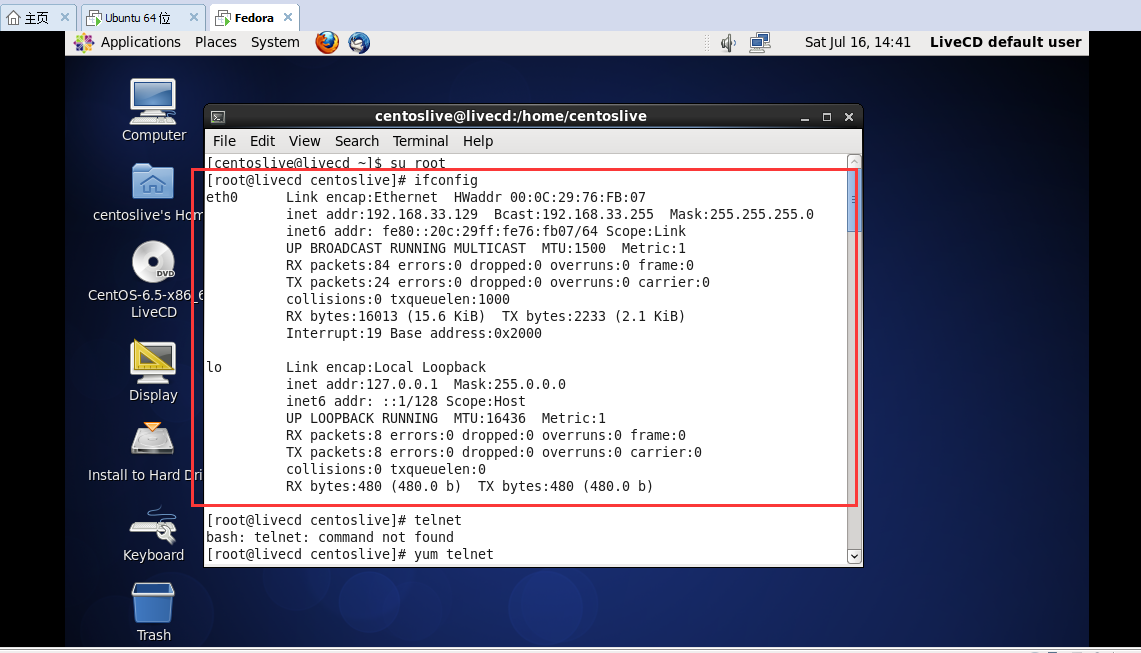


1. 使用service xinetd start，执行修改好的telnet配置文件，效果如下

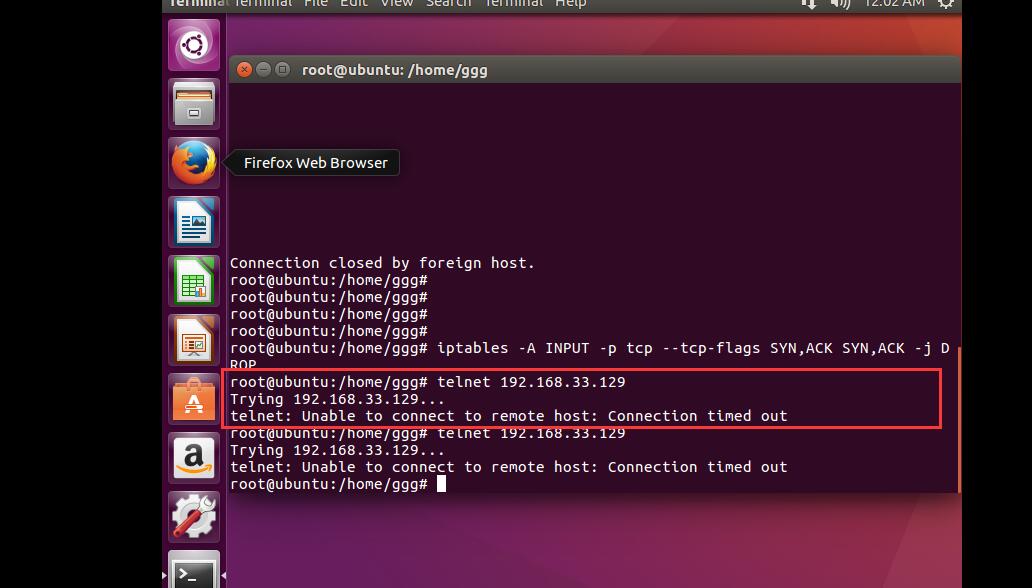


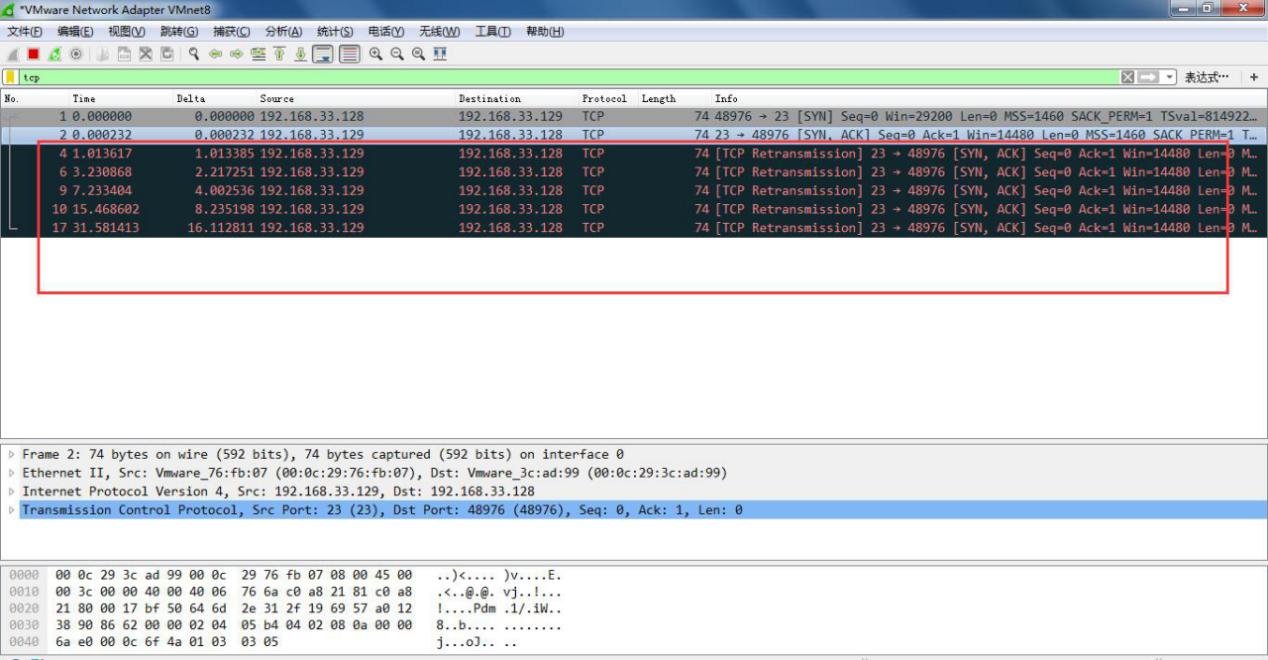
1. 关闭CentOS-6防火墙，使用命令service iptables stop

10）使用ifconfig命令得到CentOS-6的IP号



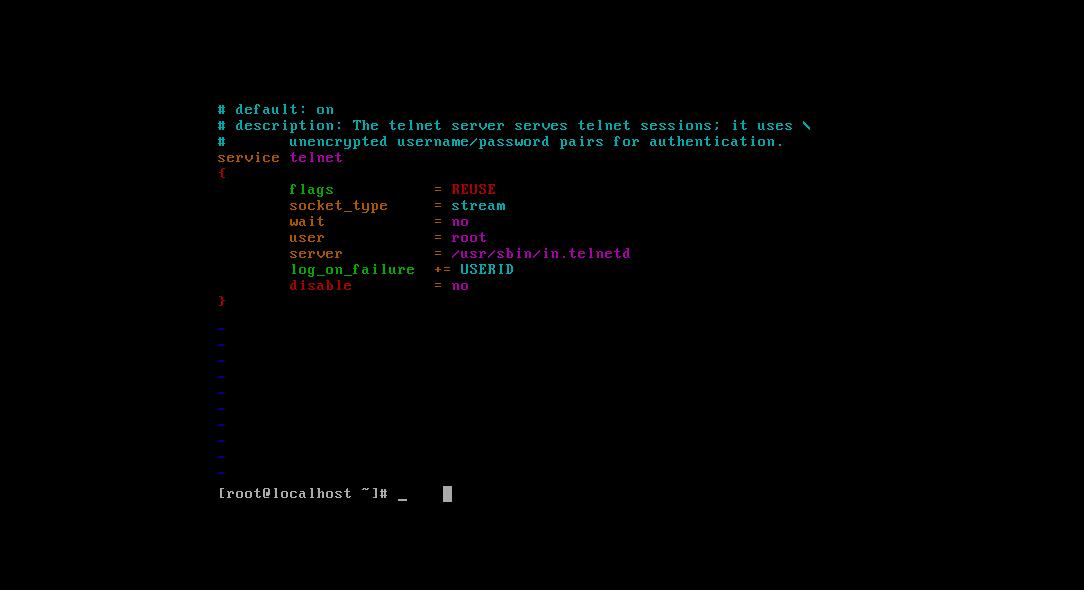
1. 使用ubuntu1604 telnet CentOS-6 并使用wireshark抓包

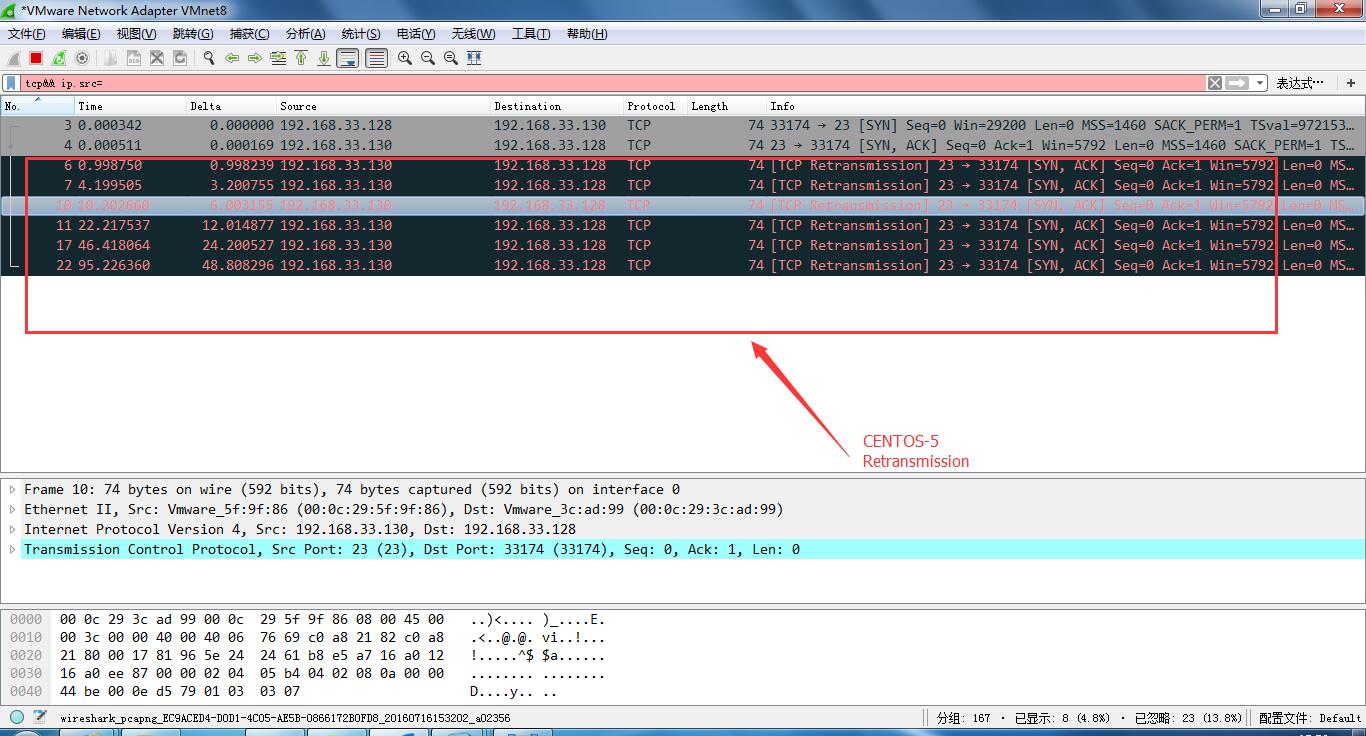




可以看到CentOS-6重传次数是5次，时延依照1,2,4,8,16的规律每次2倍递增

1. 依照上述方法对CentOS-5进行抓包分析，过程图如下

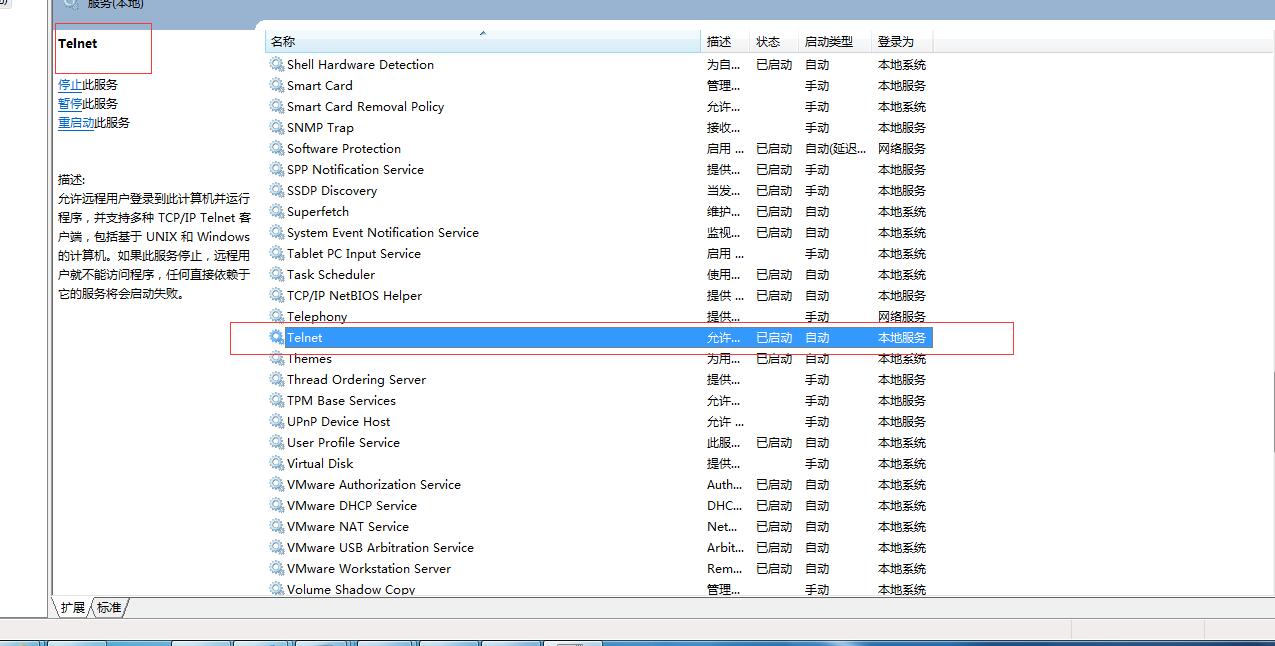




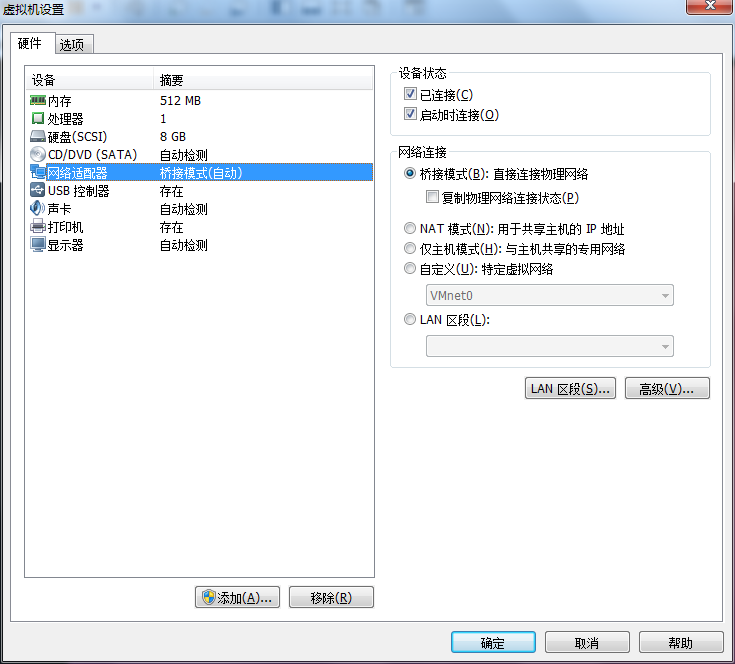
可以看到CentOS5的重传时延规律是3 6 12 24 48秒，重传了6次

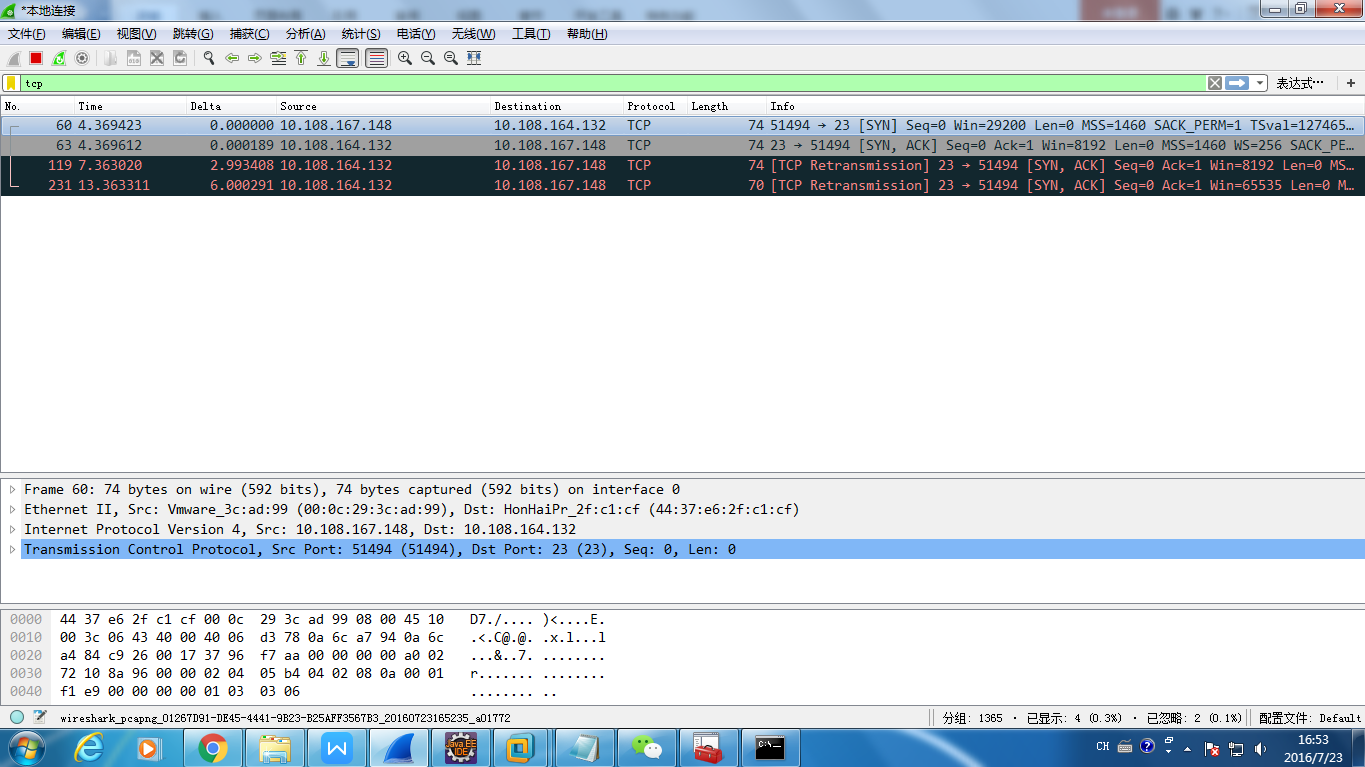
1. 测试win7重传时延特性

打开win7的Telnet服务



将ubuntu的网络适配器设置为桥接模式





Windows7超时重传仅有两次，时延分别为3,6秒

第四题

1.从群里下载冯同学的iodine-master.zip，解压，cd进去make

（<https://github.com/loganwhite/iodine>）

1. 然后执行make install 命令，安装iodine
2. 如果冯同学的代码无法进行安装，可以使用apt-get install iodine命令安装。

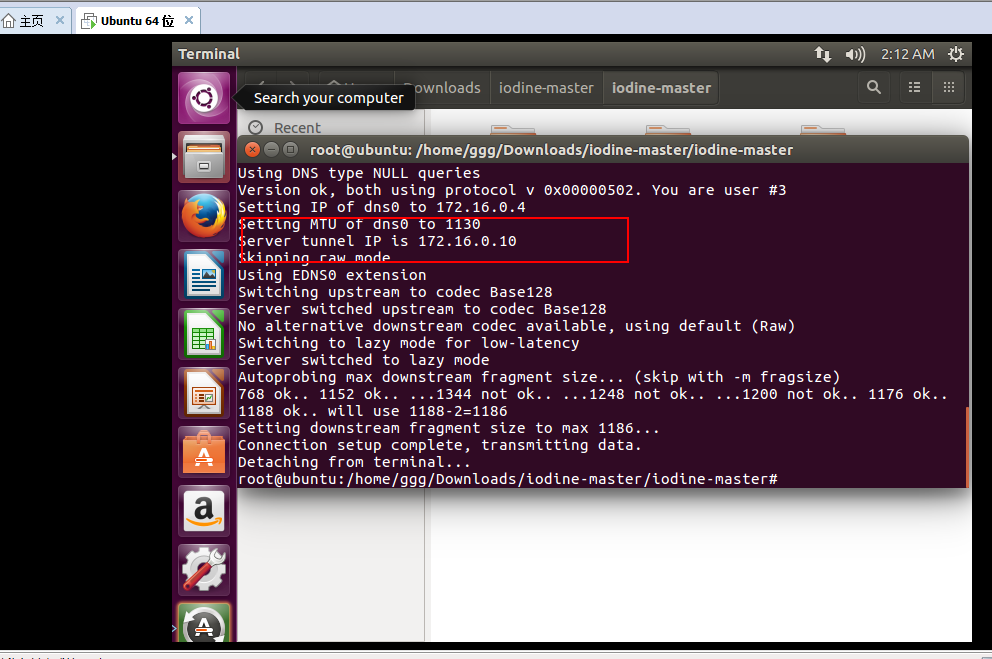
4.改浏览器（FireFox）的代理服务器设置编辑-首选项-高级-网络-连接-设置edit-preference—advanced-network-connection



5.输指令

iodine -r nsbroadband.strangled.net -P broad

出现下列内容，说明DNS隧道打通



Opened dns0

Opened UDP socket

Sending DNS queries for nsbroadband.strangled.net to 118.192.151.246

Autodetecting DNS query type (use -T to override).

Using DNS type NULL queries

Version ok, both using protocol v 0x00000502. You are user #2

Setting IP of dns0 to 172.16.0.3

Setting MTU of dns0 to 1130

Server tunnel IP is 172.16.0.10

Skipping raw mode

Using EDNS0 extension

Switching upstream to codec Base128

Server switched upstream to codec Base128

No alternative downstream codec available, using default (Raw)

Switching to lazy mode for low-latency

Server switched to lazy mode

Autoprobing max downstream fragment size... (skip with -m fragsize)

768 ok.. 1152 ok.. ...1344 not ok.. ...1248 not ok.. ...1200 not ok.. 1176 ok.. 1188 ok.. will use 1188-2=1186

Setting downstream fragment size to max 1186...

Connection setup complete, transmitting data.

Detaching from terminal...

1. 使用SSH进行远程登录，运行命令ssh -qTfnN -D 7070 [broad@172.16.0.10，](mailto:broad@172.16.0.10，)
2. 出现Are you sure you want to continue connecting (yes/no)?选择yes，输入用户名：broad，密码123456，iodine客户端配置完毕，可以绕过认证进行上网。