**实验四：分析典型P2P软件的工作过程**

1. **实验名称**

分析典型P2P软件的工作过程

1. **实验目的**

利用逆向工程的方法，分析典型的P2P软件的基本工作过程

1. **实验完成人及完成时间**

史文翰

No.2014211218

Cla.2014211304

2016.11.27

1. **实验环境**

直接对BT分析例子文件Comcast\_Bittorrent\_no\_RST.pcap进行分析，不进行额外的抓包处理。

本地BT客户机（下载者）：192.168.0.2

1. **对抓包文件的信息分析**
2. **BT客户机向tracker服务器发送请求，tracker返回一个对等方列表**

CLIENT

192.168.0.2

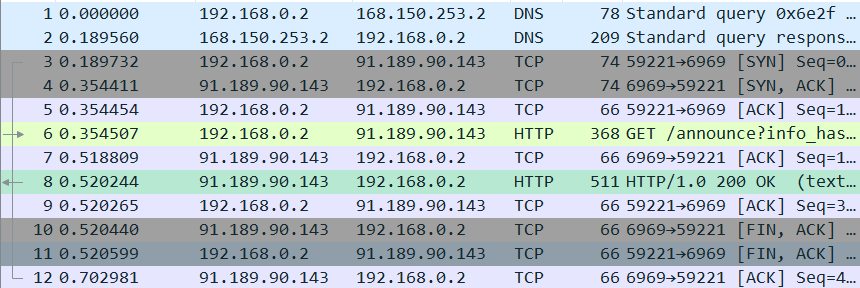
tacker

91.189.90.143

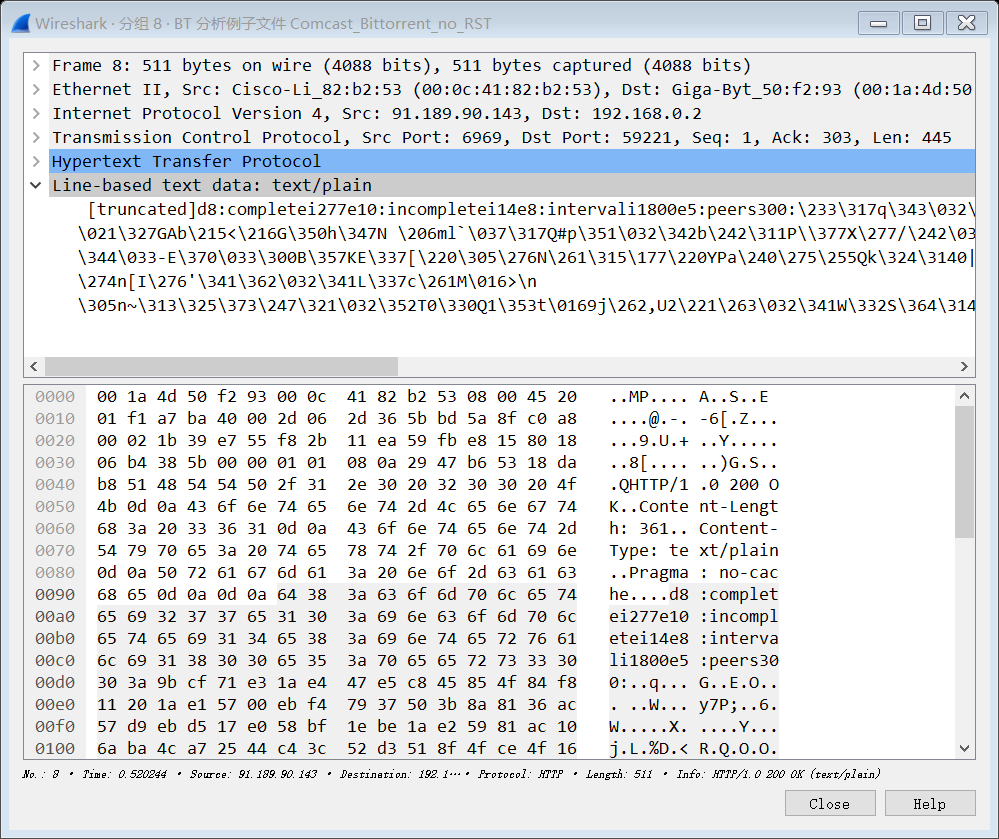
GET

**tracker**

这个过程涉及到DNS/HTTP和TCP协议，对应于pcap文件中的如下部分。



可知tacker的IP为91.189.90.143，BT客户端首先通过GET的方式索取对等方列表，分析tacker返回的内容如下：



可以看出tracker返回的信息中最重要的是peers300后面的十六进制标志串。查阅资料可知，分组内容interval显示1800，表示BT客户机每隔1800个时间单位就与tracker服务器重新联系一次，且peer的部分共有300bytes。“300：”串之后的每6个bytes表示了一个对等方，且以socket形式存储（IP：port模式），而300表示了共有300bytes的对等数据作为返回，因此对等方列表中共有50个对等方。

至此，BT客户机已经拿到了对等实体的IP和端口信息，以便在之后的过程中以TCP的形式与其进行联络。

1. **尝试连接阶段**

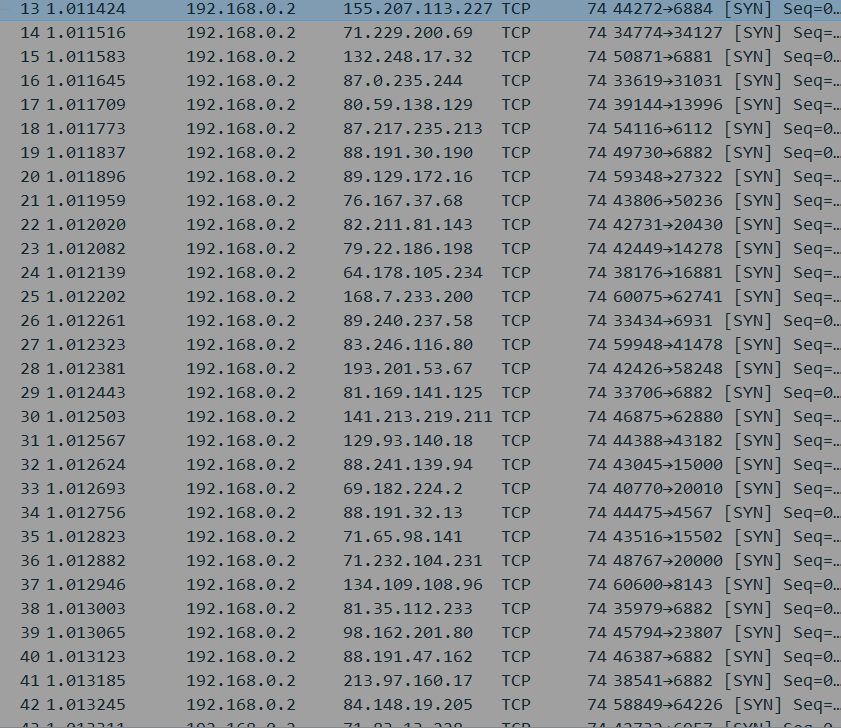
**TCP SYN**

**tracker**

CLIENT

192.168.0.2

在BT客户机拿到了tracker返回的一系列对等实体的信息后，BT客户机开始对列表中的一部分对等实体发出TCP的SYN信号请求建立连接。这个过程对应于pcap文件中的如下部分。



我们发现总共的SYN信号并没有达到50个，而是有选择地选取了tracker返回列表中的一部分对等体。

1. **连接确认阶段**

**REQUESET PIECE**

**handshake、unchoke、bitField**

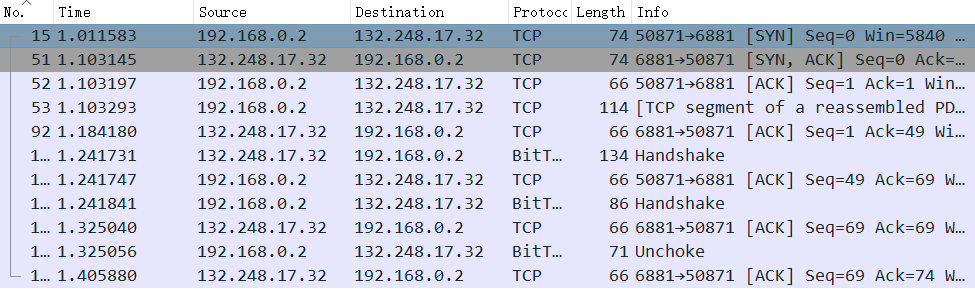
**handshake、unchoke、interested**

CLIENT

192.168.0.2

上述尝试建立TCP连接的过程是不具有确定性的，即对等体有概率拒绝掉这次请求，因此只有一少部分的对等体真正地与BT客户机建立TCP连接。一旦与某对等体建立起客户连接后，就开始与对方进行信息交互。包括握手信号、资源状态的互换、互通对资源的意愿情况等。

我们选取几个对等体进行TCP流追踪进行分析，截图如下。

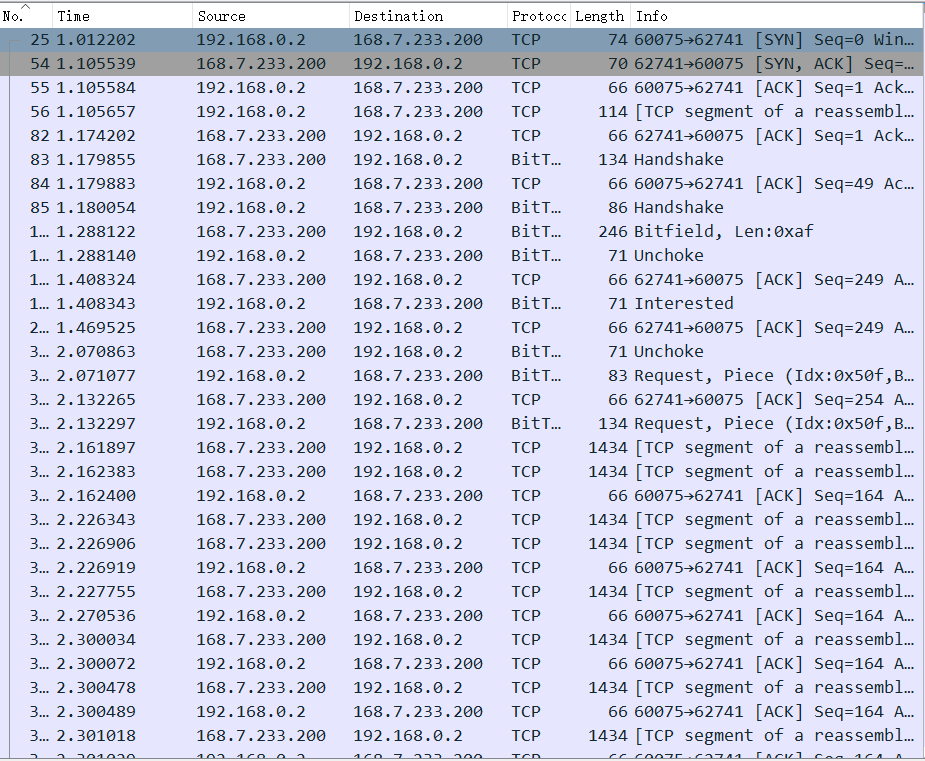


在于对等体132.248.17.32：6881进行TCP通讯的过程中，前三行是互相建立TCP连接的起始过程。此后，双方互相传递了handshake握手信号。其中还有client->peer的unchoke信号，这表示client在此阶段认为这个peer是符合传输条件的peer，但是在这个例子中，通讯没有继续进行，client端没有进一步收到bitField信息。

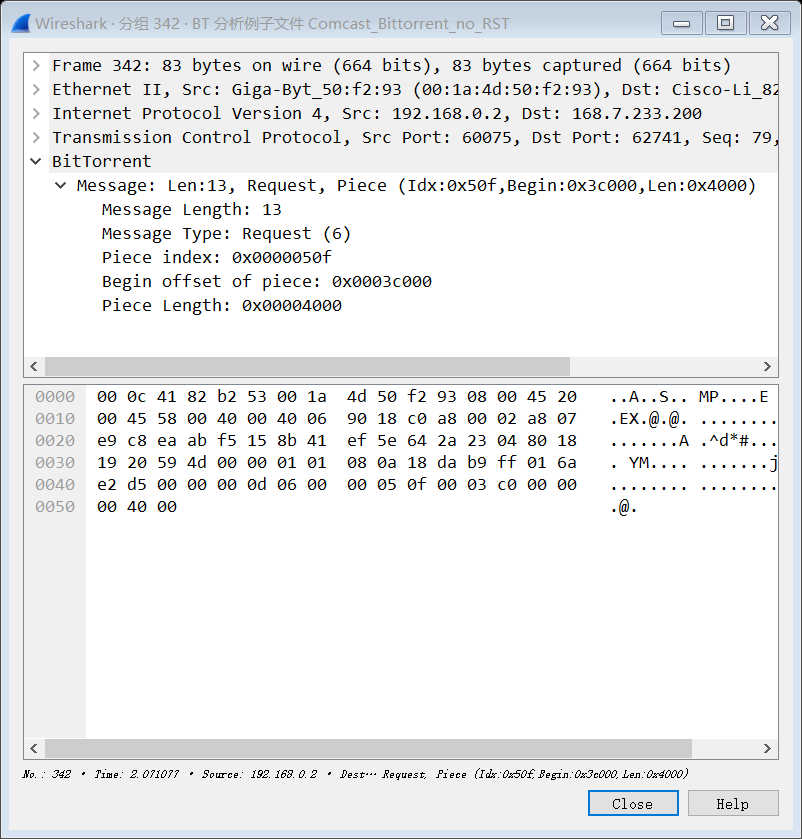
1. **资源请求与交互阶段**

如果进行handshake且unchoke并对资源interested的话，client与这个peer之间就可以进行资源的请求与交互了。

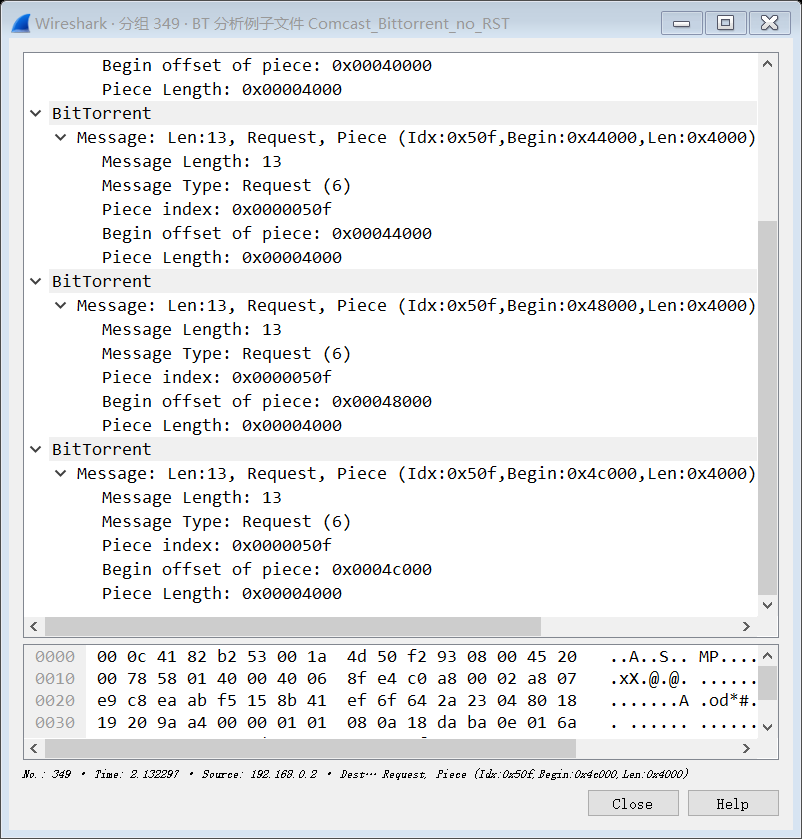
由于上例中不包含资源交互的过程，因此我们截取另一个peer的TCP流来说明这一点。



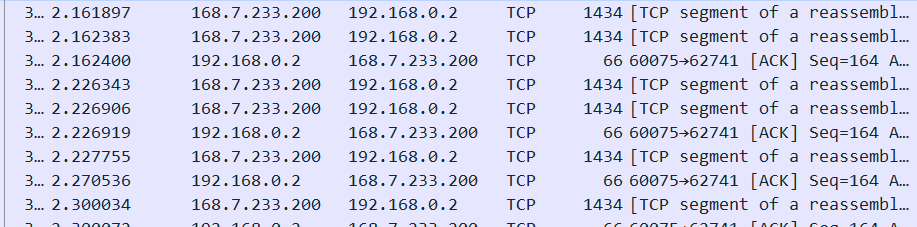
我们可以看出这个成功建立TCP连接的peer为168.7.233.200：62741。连接确认阶段中，双方互换了handshake信号，在peer收到了client的handshake信号后，向client发送了一个BitField信号，这个信号包含了peer拥有的资源情况。收到了peer的BitField信号后，client返回一个interested信号，表示对这个资源感兴趣。至此，连接信息的交互已经完毕，接下来进入资源请求阶段，通过request一个piece实现。Client向peer提出了一个request内容如下：



Request可以多次请求且一次请求多个piece，这些request信息被加载到peer的缓冲区中，同时peer开始着手传送这些piece。



部分传送过程如下。



也就是典型的TCP传输过程，包含了ACK信息和分段等技术。请注意，我们在这里只分析了一个peer的交互情况，由于tracker返回了一系列的peer，且这些peer中不止一个peer与client建立了TCP连接，因此类似这样的传输应该不知上述分析的这一个通路。

在此之后，通过典型TCP的结束过程可以结束一个peer与client的本次资源交互。