**TCP实验报告**

学号：16020031075

姓名：王刚

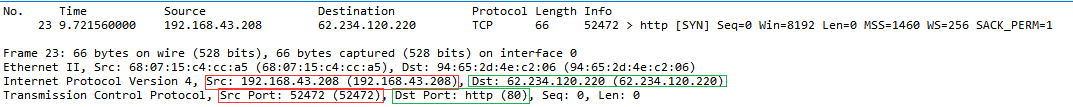
学科：计算机网络

任课教师：唐瑞春

问： 你的客户端电脑传送文件到 62.234.120.220 的 IP 地址和端口是什么？从62.234.120.220 接收文件的 IP 地址和端口是什么？

答：

由文件TCP.txt中的信息截图如下：



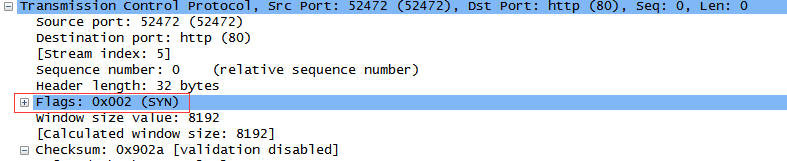
知：我的客户端电脑的IP地址为：192.168.43.208；端口号为52472；

接受文件的IP地址为：62.234.120.220；端口号为80。

1. 问：用来初始化客户端电脑和 62.234.120.220 的 TCP 连接的 TCP SYN 报文段的序号是什么？在报文段中，哪个地方表明这是一个 SYN 报文段？

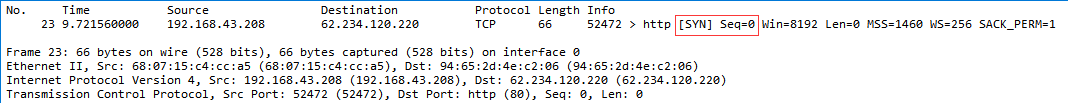
答：

在Wireshark里打开1中的信息，如下



知：Flags=0000 0000 0010  
由TCP首部信息：

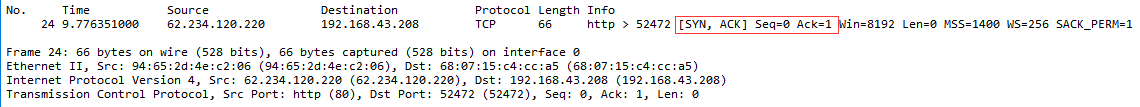
,知SYN=1，那么可以知道这是一个SYN报文段。

由

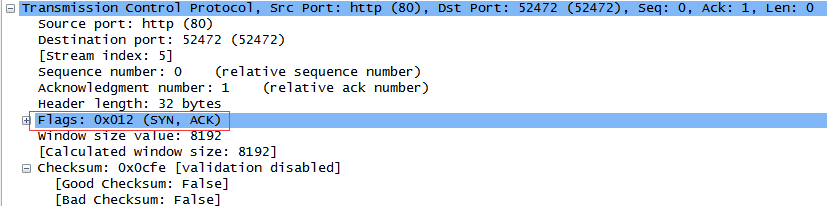
知：初始化SYN报文段的序号为：0。

1. 问：回复 SYN 报文段时，由62.234.120.220 发送的 SYNACK 报文段的序号是什么？在SYN，ACK 报文段中的 ACKnowledgement 域的值是什么？62.234.120.220是如何确定这个值的？表明这个段是 SYN，ACK 的标志位或者符号是什么？  
   答：

找到回复SYN的报文段：



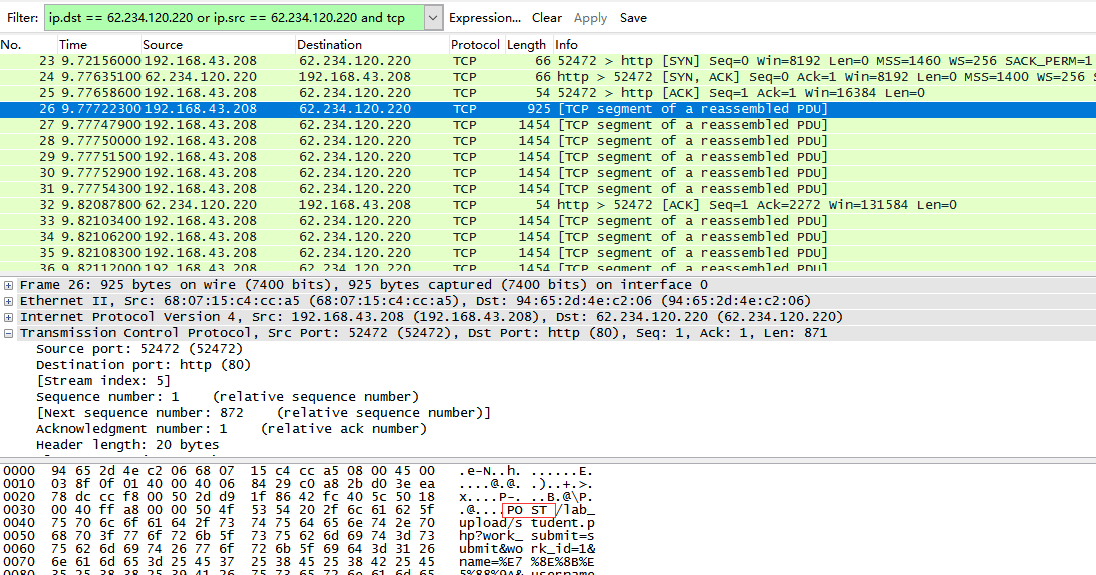
知：序号是0；ACK=1，ACK=SYN+1.  
查看Flags：



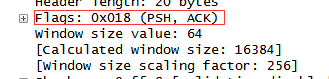
Flags=0000 0001 0010；也就是说Flags表明了ACK和SYN。

1. 问：包含 HTTP POST 命令的 TCP 段的序号是多少？注意，要找到 POST 命令，你需要在 Wireshark 窗口的底部发掘这个数据包内容域，找到一个在其 DATA 域中有标识POST 的段。  
   答：

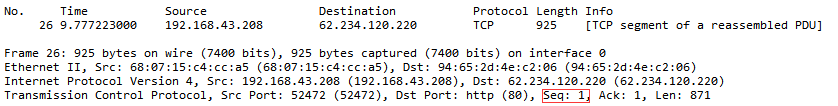
先找到POST字段：



发现其在NO.26里，打开其并找到Flags：



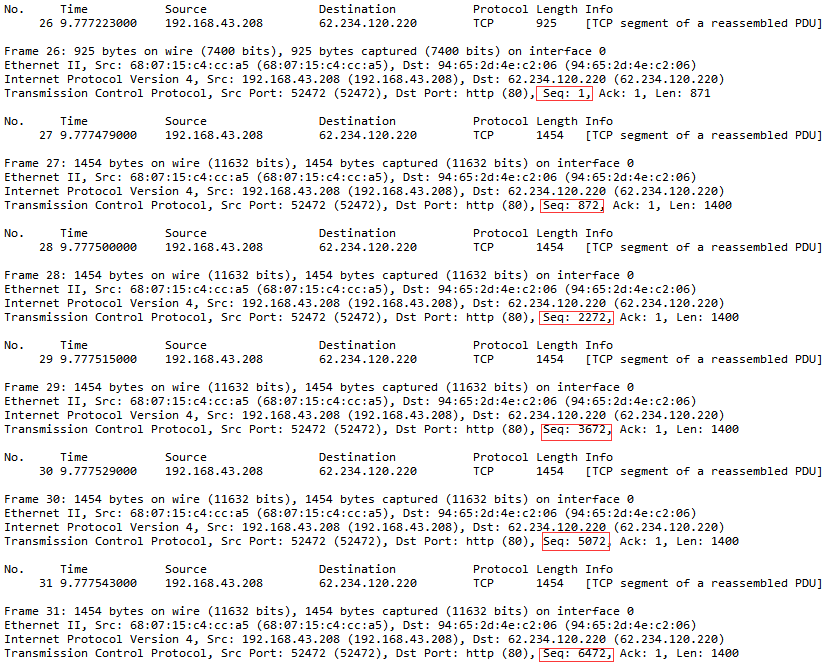
由：



知：序号为1.

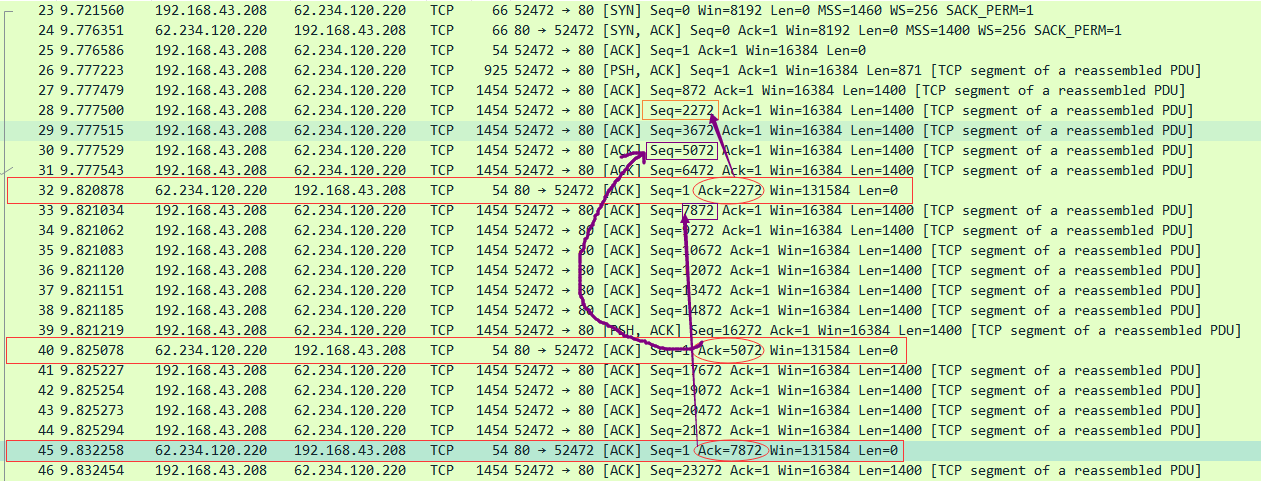
问：仔细思考一下作为 TCP 连接的第一个报文段的包含 HTTP POST 的 TCP 报文段。TCP 连接（包括包含 HTTP POST 的报文段）的开头 6 个报文段的序号是多少？每个报文段都在什么时间发送？每个报文段接收到 ACK 的时间是多少？

答：

由：  


知：开头的六个序列号为：1、872、2272、3672、5072、6472.

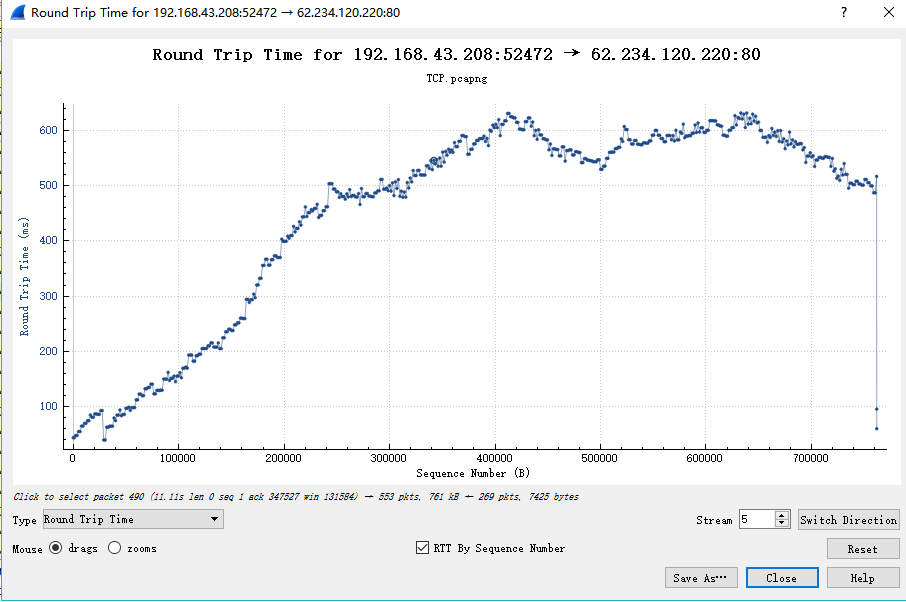
由：



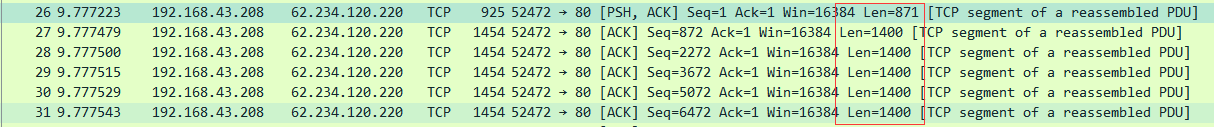
得到下表：

|  |  |  |  |
| --- | --- | --- | --- |
| Seq | 发送时间 | ACK接收时间 | RTT |
| 1 | 9.777223000 | 9.820878000 | 0.043655 |
| 872 | 9.777479000 | 9.820878000 | 0.043399 |
| 2272 | 9.777500000 | 9.820878000 | 0.043378 |
| 3672 | 9.777515000 | 9.825078000 | 0.047563 |
| 5072 | 9.777529000 | 9.825078000 | 0.047549 |
| 6472 | 9.777543000 | 9.832258000 | 0.054715 |

下附Seq-RTT图像：



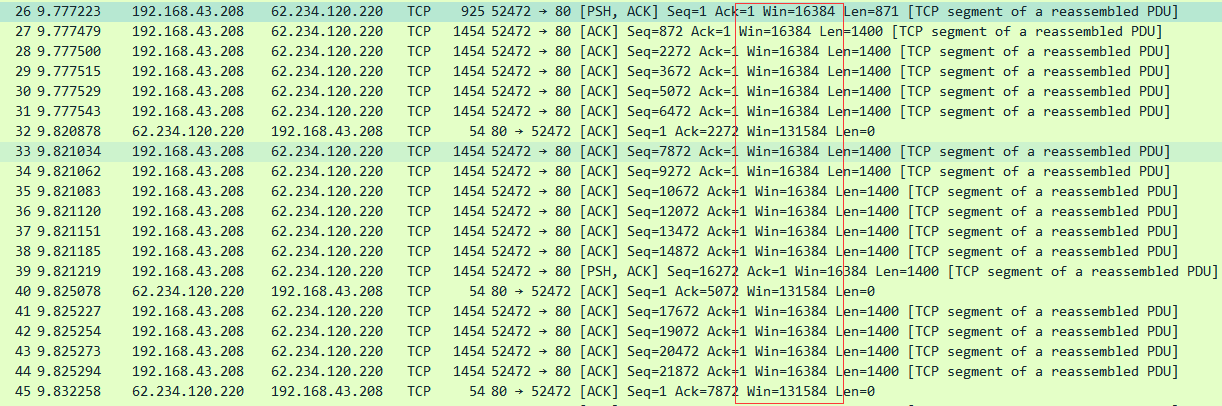
1. 问：开始的 6 个 TCP 报文段的长度各自是多少？  
   答：  
   由：

 知，开始的6个报文段的长度如下表所示(**注意要加上首部20个字节**)，单位字节：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Seq | 1 | 872 | 2272 | 5072 | 6472 |
| Len | 891 | 1420 | 1420 | 1420 | 1420 |

1. 问：在整个跟踪过程中，在接收端广告（advertise）的可用缓存空间的最小值是多少？接收端有没有因缓存空间不足而限制发送端的发送？

答：  
由：



知：

接收窗口的最小值是min{131584, 131584,131584},即131584，因而接收端广告的可用缓存空间的最小值是131584。

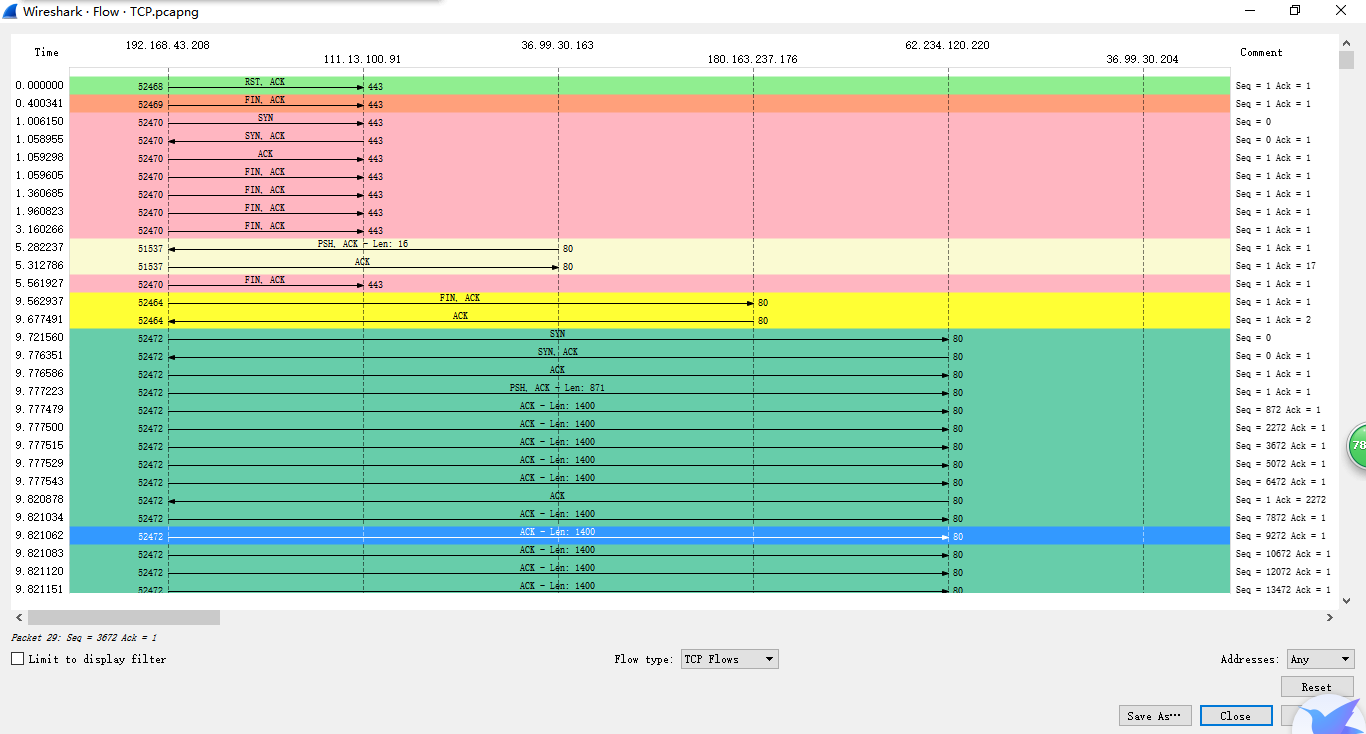
发送窗口大小一直是16384，小于接收窗口大小且保持恒定，因此没有这个限制。

1. 问：在跟踪文件中，有重传的报文段么？回答这个问题，你需要检查哪个地方？

答：  
经过查询，在跟踪文件中没有重传的报文段。检查我的电脑发出的Seq字段，若有重复的Seq字段，那么，证明发生了数据重传，若无重复的Seq，那么则未发生数据重传。

问：接收方在一个 ACK 中，通常确认多少数据？你能辨别出这样一种情形吗：即接收方对收到的报文段，每隔一个确认一次？

答：接收方通常在啊一个ACK中确认1~2个报文段数据。

利用Wireshark里面的Flow Graph功能，得到下图：  


可以看出，在这次跟踪中，每个ACK确认2个报文段。

10.

问：这个 TCP 连接的吞吐量（每单位时间传输的字节数）是多少？解释你是如何计算这个数值的？  
由：





知：第一个报文段的序号是1,时间是9.777223，最后一个报文段序号是761976，时间是18.410307.

故吞吐量为：(761976 - 1 + 1) / (18.410307 - 9.777223) = 88262 bit/s = 86.2 kb/s

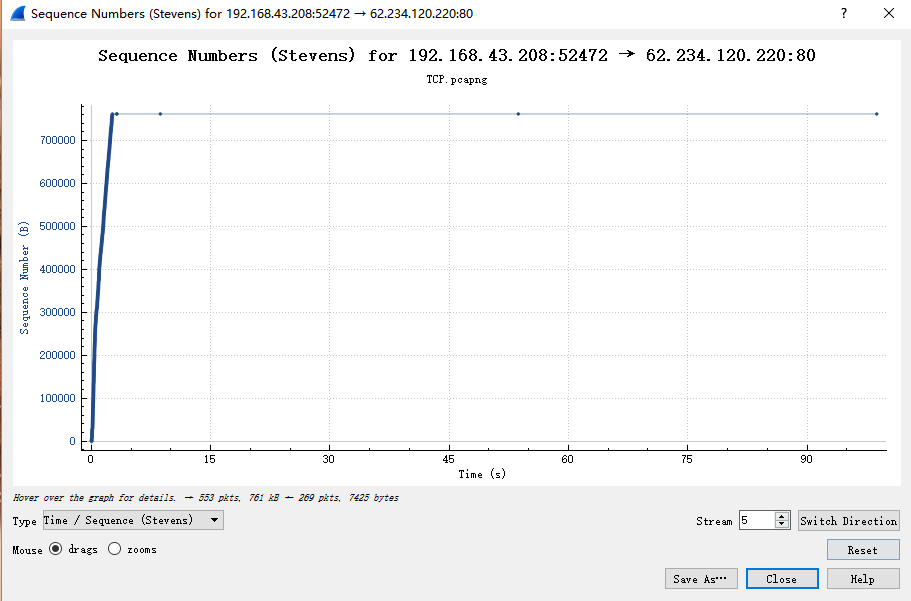
11.

问：用Time-Sequence-Graph(Stevens)中的画图工具观察从客户端发送到62.234.120.220服务器的TCP段的序列号－时间图。你怎样判断TCP的慢启动(slowstart)开始和结束？拥塞避免在什么地方开始起作用的？注意在实际的跟踪中，不是所有的都像教材那样简单漂亮的形式。同时还要注意在Time-Sequence-Graph(Stevens)中纵坐标所代表的变量与教材中是否不同。

总结这次实验中所得到的 TCP 数据与我们在教材中所学的理想情况有什么不同？

答：

使用WireShark的TSG画图工具得到：



对于慢启动的特点是发送端的Congestion window值呈指数增长。

1、2、5、7、4、4、4、4、4、6、4、4...[以上观测点是肉眼数出来的].

前四轮为慢启动，第五轮及第五轮以后发送的数量稳定在一个数值附近，因此可以大致判断拥塞避免在此时发生作用。

不同之处：  
 （1）未出现数据重传的现象。

（2）未出现明显的拥塞控制。