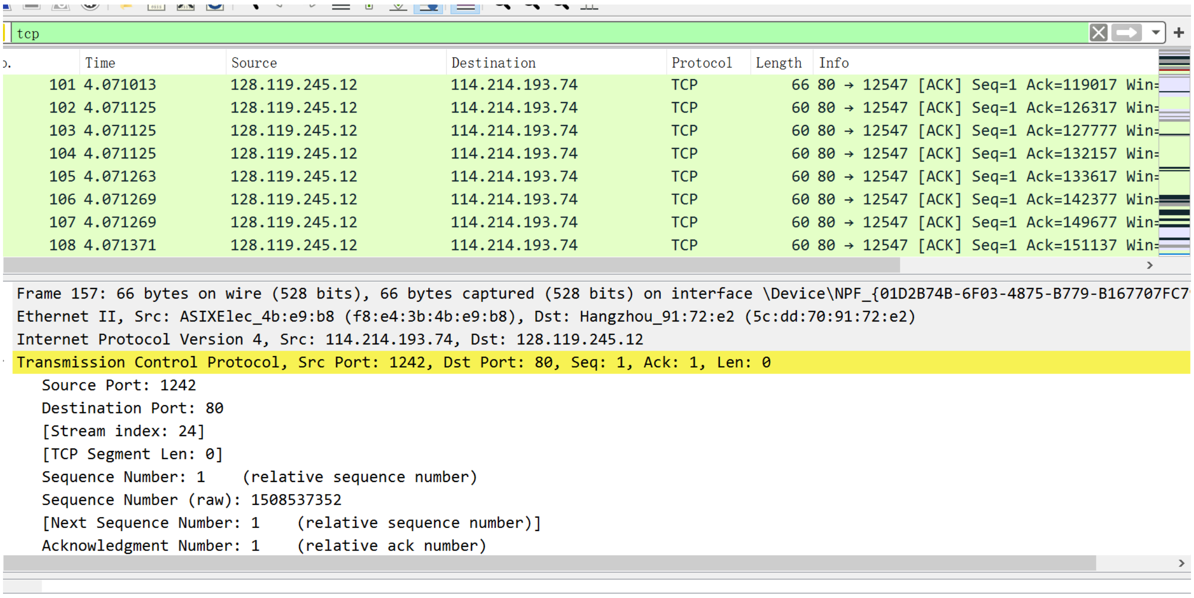
实验五

Wireshark Lab: TCP

阅读实验文档，按照要求进行实验。

1.首先要使用wireshark获取从计算机到服务器的TCP包。按照操作进行。先从网站下载一个 ASCII文件，然后登陆http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html上传这个文件，并使用wireshark进行抓包。

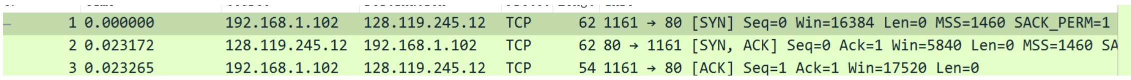


```
tcp
o.   Time        Source          Destination     Protocol  Length  Info
 101 4.071013    128.119.245.12  114.214.193.74  TCP       66 80 → 12547 [ACK] Seq=1 Ack=119017 Win=
 102 4.071125    128.119.245.12  114.214.193.74  TCP       60 80 → 12547 [ACK] Seq=1 Ack=126317 Win=
 103 4.071125    128.119.245.12  114.214.193.74  TCP       60 80 → 12547 [ACK] Seq=1 Ack=127777 Win=
 104 4.071125    128.119.245.12  114.214.193.74  TCP       60 80 → 12547 [ACK] Seq=1 Ack=132157 Win=
 105 4.071263    128.119.245.12  114.214.193.74  TCP       60 80 → 12547 [ACK] Seq=1 Ack=133617 Win=
 106 4.071269    128.119.245.12  114.214.193.74  TCP       60 80 → 12547 [ACK] Seq=1 Ack=142377 Win=
 107 4.071269    128.119.245.12  114.214.193.74  TCP       60 80 → 12547 [ACK] Seq=1 Ack=149677 Win=
 108 4.071371    128.119.245.12  114.214.193.74  TCP       60 80 → 12547 [ACK] Seq=1 Ack=151137 Win=

Frame 157: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{01D2B74B-6F03-4875-B779-B167707FC7
Ethernet II, Src: ASIXElec_4b:e9:b8 (f8:e4:3b:4b:e9:b8), Dst: Hangzhou_91:72:e2 (5c:dd:70:91:72:e2)
Internet Protocol Version 4, Src: 114.214.193.74, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 1242, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
   Source Port: 1242
   Destination Port: 80
   [Stream index: 24]
   [TCP Segment Len: 0]
   Sequence Number: 1    (relative sequence number)
   Sequence Number (raw): 1508537352
   [Next Sequence Number: 1    (relative sequence number)]
   Acknowledgment Number: 1    (relative ack number)
```

2.按照实验说明操作，先用TCP筛选，找到和gaia.cs.umass.edu的一系列信息，包括包含SYN的三次握手，HTTP POST消息，TCP segment of a reassembled PDU等。回答文档中的问题：

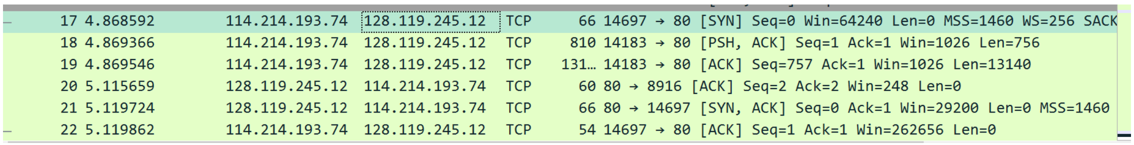
1，2是分析下载的抓包结果

1. 

```
   1 0.000000    192.168.1.102   128.119.245.12  TCP   62 1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM=1
   2 0.023172    128.119.245.12  192.168.1.102   TCP   62 80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SA
   3 0.023265    192.168.1.102   128.119.245.12  TCP   54 1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0
```

   源IP 192.168.1.102；端口号 1161

2. 从上图可以看出，网站IP 128.119.245.12，端口 80
3. 按操作更改设置，取消勾选http；



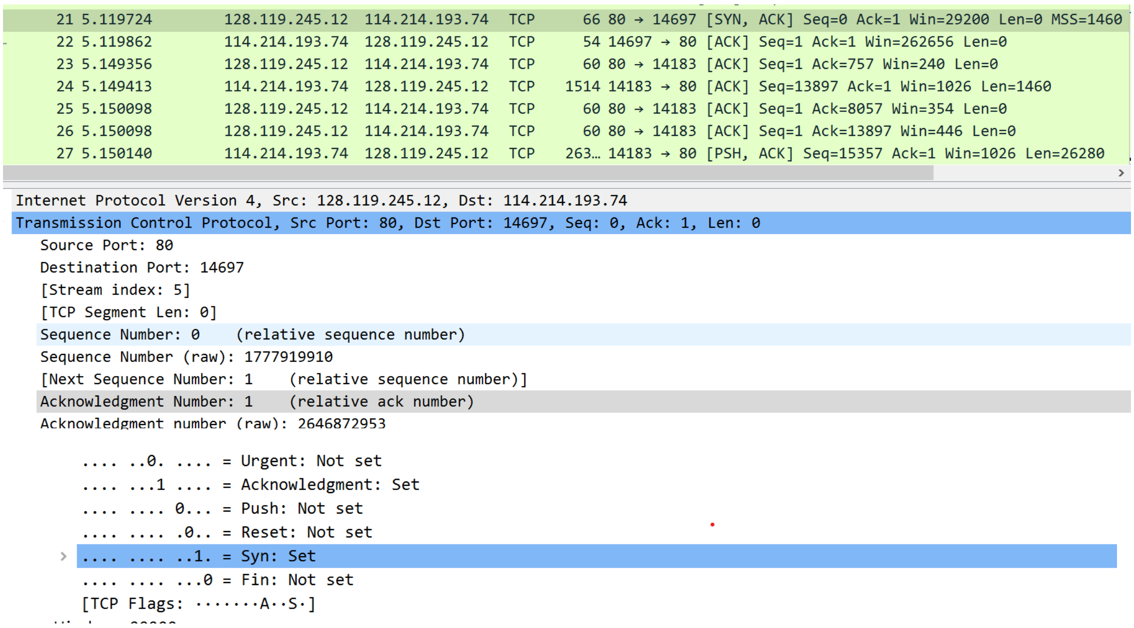
```
  17 4.868592    114.214.193.74  128.119.245.12  TCP     66 14697 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK
  18 4.869366    114.214.193.74  128.119.245.12  TCP    810 14183 → 80 [PSH, ACK] Seq=1 Ack=1 Win=1026 Len=756
  19 4.869546    114.214.193.74  128.119.245.12  TCP   131… 14183 → 80 [ACK] Seq=757 Ack=1 Win=1026 Len=13140
  20 5.115659    128.119.245.12  114.214.193.74  TCP     60 80 → 8916 [ACK] Seq=2 Ack=2 Win=248 Len=0
  21 5.119724    128.119.245.12  114.214.193.74  TCP     66 80 → 14697 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
  22 5.119862    114.214.193.74  128.119.245.12  TCP     54 14697 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
```

   自己的抓包IP 114.214.193.74，端口 14697

3.**TCP Basics**

回答一些关于TCP段的问题

4. 可以从第3题的图中看出，Seq=0，识别是为了开始三次握手，这是第一次。
5.



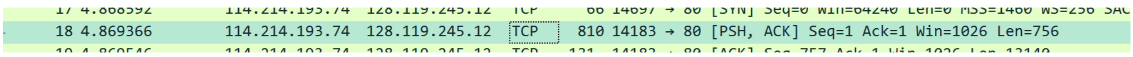
```
21 5.119724    128.119.245.12  114.214.193.74  TCP    66 80 → 14697 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
22 5.119862    114.214.193.74  128.119.245.12  TCP    54 14697 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
23 5.149356    128.119.245.12  114.214.193.74  TCP    60 80 → 14183 [ACK] Seq=1 Ack=757 Win=240 Len=0
24 5.149413    114.214.193.74  128.119.245.12  TCP  1514 14183 → 80 [ACK] Seq=13897 Ack=1 Win=1026 Len=1460
25 5.150098    128.119.245.12  114.214.193.74  TCP    60 80 → 14183 [ACK] Seq=1 Ack=8057 Win=354 Len=0
26 5.150098    128.119.245.12  114.214.193.74  TCP    60 80 → 14183 [ACK] Seq=1 Ack=13897 Win=446 Len=0
27 5.150140    114.214.193.74  128.119.245.12  TCP  263… 14183 → 80 [PSH, ACK] Seq=15357 Ack=1 Win=1026 Len=26280

Internet Protocol Version 4, Src: 128.119.245.12, Dst: 114.214.193.74
Transmission Control Protocol, Src Port: 80, Dst Port: 14697, Seq: 0, Ack: 1, Len: 0
   Source Port: 80
   Destination Port: 14697
   [Stream index: 5]
   [TCP Segment Len: 0]
   Sequence Number: 0    (relative sequence number)
   Sequence Number (raw): 1777919910
   [Next Sequence Number: 1    (relative sequence number)]
   Acknowledgment Number: 1    (relative ack number)
   Acknowledgment number (raw): 2646872953

      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
    > .... .... ..1. = Syn: Set
      .... .... ...0 = Fin: Not set
      [TCP Flags: ·······A··S·]
```

回应的Seq=0，Acknowledgment值为1，Ack可以表明确认字段的值是有效的，说明服务器收到了发送请求，回复SYN-ACK确认报文。

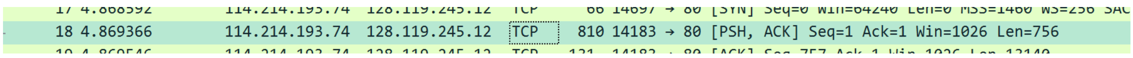
6.



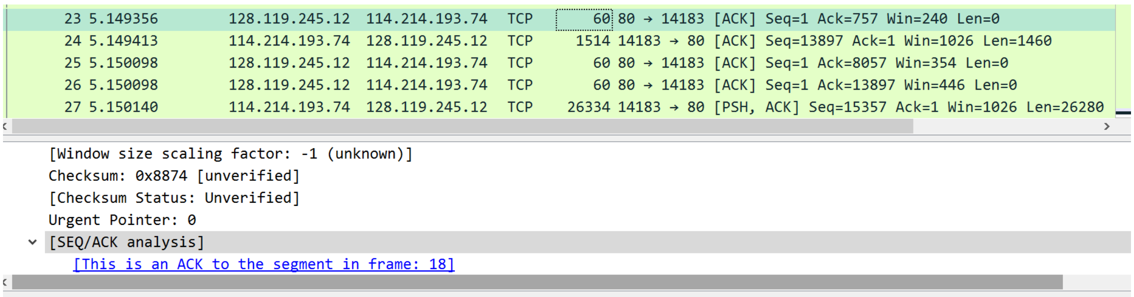
```
17 4.868592    114.214.193.74  128.119.245.12  TCP    66 14697 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SAC
18 4.869366    114.214.193.74  128.119.245.12  TCP   810 14183 → 80 [PSH, ACK] Seq=1 Ack=1 Win=1026 Len=756
19 4.869546    114.214.193.74  128.119.245.12  TCP  1314 14183 → 80 [ACK] Seq=757 Ack=1 Win=1026 Len=13140
```

seq=1

7.

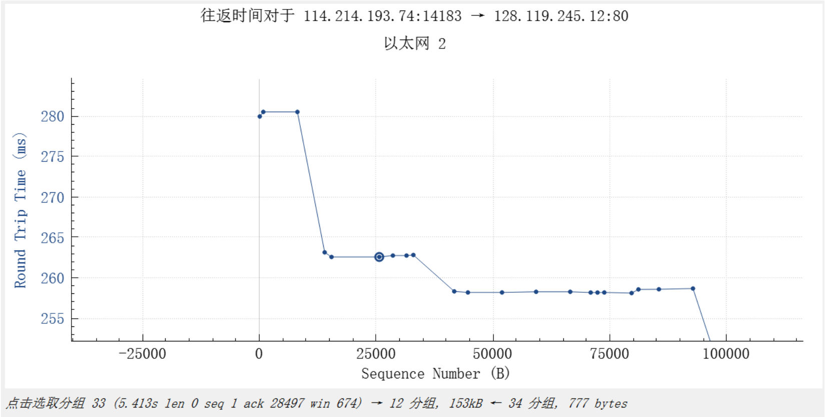


```
17 4.868592    114.214.193.74  128.119.245.12  TCP    66 14697 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SAC
18 4.869366    114.214.193.74  128.119.245.12  TCP   810 14183 → 80 [PSH, ACK] Seq=1 Ack=1 Win=1026 Len=756
19 4.869546    114.214.193.74  128.119.245.12  TCP  1314 14183 → 80 [ACK] Seq=757 Ack=1 Win=1026 Len=13140
```



```
23 5.149356    128.119.245.12  114.214.193.74  TCP    60 80 → 14183 [ACK] Seq=1 Ack=757 Win=240 Len=0
24 5.149413    114.214.193.74  128.119.245.12  TCP  1514 14183 → 80 [ACK] Seq=13897 Ack=1 Win=1026 Len=1460
25 5.150098    128.119.245.12  114.214.193.74  TCP    60 80 → 14183 [ACK] Seq=1 Ack=8057 Win=354 Len=0
26 5.150098    128.119.245.12  114.214.193.74  TCP    60 80 → 14183 [ACK] Seq=1 Ack=13897 Win=446 Len=0
27 5.150140    114.214.193.74  128.119.245.12  TCP  26334 14183 → 80 [PSH, ACK] Seq=15357 Ack=1 Win=1026 Len=26280

   [Window size scaling factor: -1 (unknown)]
   Checksum: 0x8874 [unverified]
   [Checksum Status: Unverified]
   Urgent Pointer: 0
 ˅ [SEQ/ACK analysis]
      [This is an ACK to the segment in frame: 18]
```



点击选取分组 33 (5.413s len 0 seq 1 ack 28497 win 674) → 12 分组, 153kB ← 34 分组, 777 bytes

第一段：len=756，序列号1，RTT=0.27999s

EstimatedRTT=RTT=0.27999s

剩下的同理，

第二段：len=13140，seq=757，RTT=0.280552s

$$EstimatedRTT = 0.875 * 0.27999 + 0.125 * 0.280552 = 0.28006025s$$

第三段：len=1460，seq=13897，RTT=0.263173s

$$EstimatedRTT = 0.875 * 0.28006025 + 0.125 * 0.263173 = 0.27794934s$$

第四段：len=26280，seq=15357，RTT=0.262849s

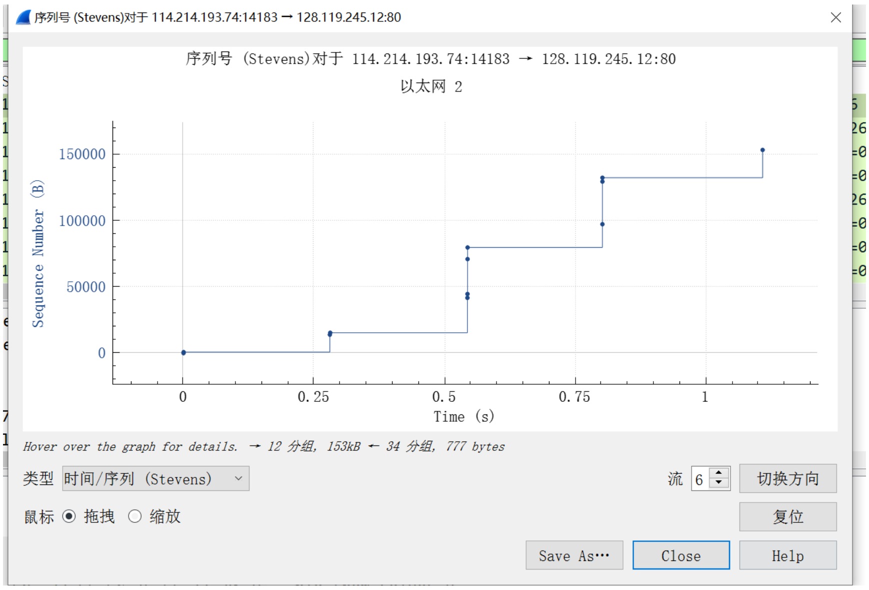
$$EstimatedRTT = 0.875 * 0.27794934 + 0.125 * 0.262849 = 0.27606180s$$

第五段：len=2920，seq=41637，RTT=0.258209s

$$EstimatedRTT = 0.875 * 0.27606180 + 0.125 * 0.258209 = 0.27383020s$$

第六段：len=8760，seq=44557，RTT=0.258292s

$$EstimatedRTT = 0.875 * 0.27383020 + 0.125 * 0.258292 = 0.271888s$$

8. 长度分别是810，13194，1514，26334，2974，8814
9. 最小的缓冲区Win=1026，是第一次ACK返回时的值。后面的值都比1026大，所以缓存空间的缺失不会限制发送。
10. 

序列号 (Stevens)对于 114.214.193.74:14183 → 128.119.245.12:80

序列号 (Stevens)对于 114.214.193.74:14183 → 128.119.245.12:80

以太网 2

Hover over the graph for details. → 12 分组, 153kB ← 34 分组, 777 bytes

类型 时间/序列 (Stevens)

流 6

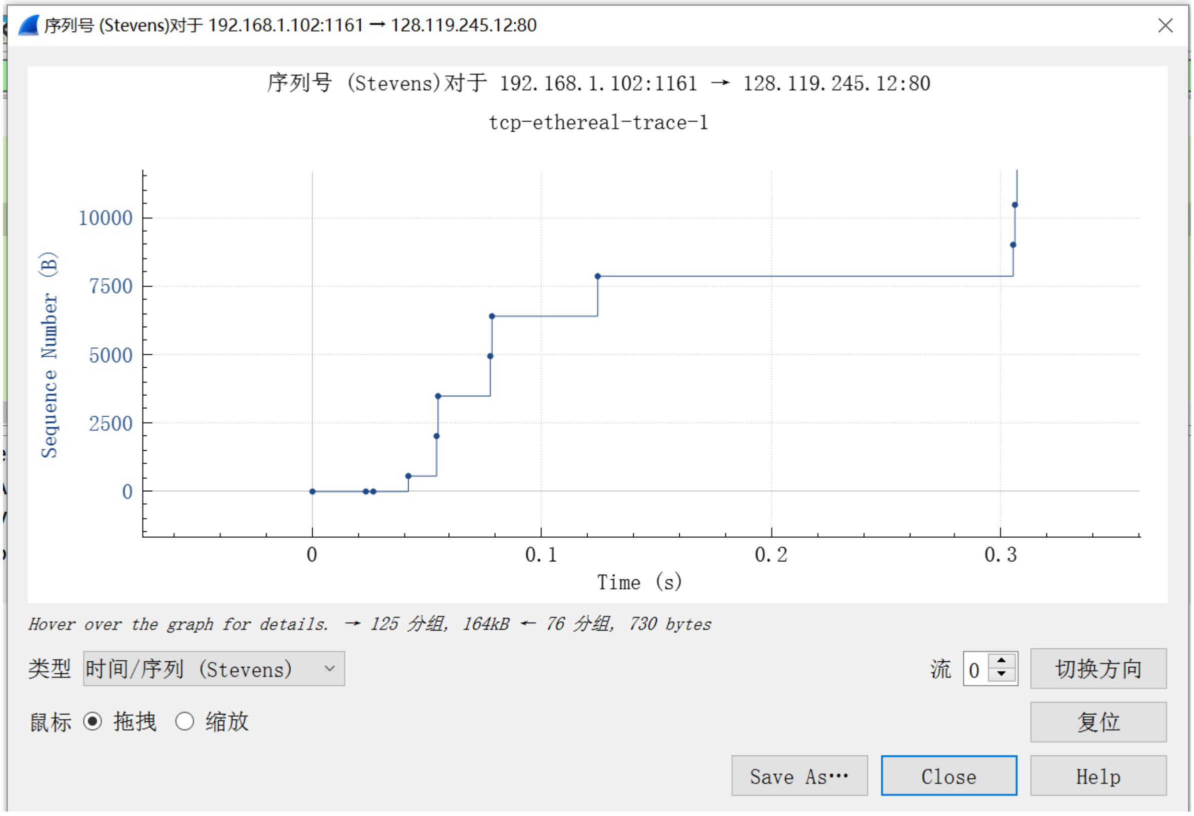
鼠标 ● 拖拽 ○ 缩放

由图可以看出，序列号时递增的，应该是没有重传。

11. 两次传输之间seq的差值，这个差值就是在前面的ACK中确认的数据
12. 可以由最后一个ACK的值减去最初的，除以时间差，就是吞吐量：$\frac{153078-1}{5.977683-4.869366} = 138116.62$bytes/s

4.TCP拥塞控制

由于我抓包获得的图像和理想的差距太大，不好分析，所以选择使用作者的抓包进行分析。



序列号 (Stevens)对于 192.168.1.102:1161 → 128.119.245.12:80

序列号 (Stevens)对于 192.168.1.102:1161 → 128.119.245.12:80

tcp-ethereal-trace-1

Hover over the graph for details. → 125 分组, 164kB ← 76 分组, 730 bytes

类型 时间/序列 (Stevens)

流 0

鼠标 ● 拖拽 ○ 缩放

13. 观察图像，慢启动从分组5到分组13，也即0.04174s到0.1242s；之后几乎停滞，进入了拥塞避免阶段。这个和课本上的区别：课本的比较理想化，是指数增长，但这个数据近似指数增长，但是数据拐角比较明显。