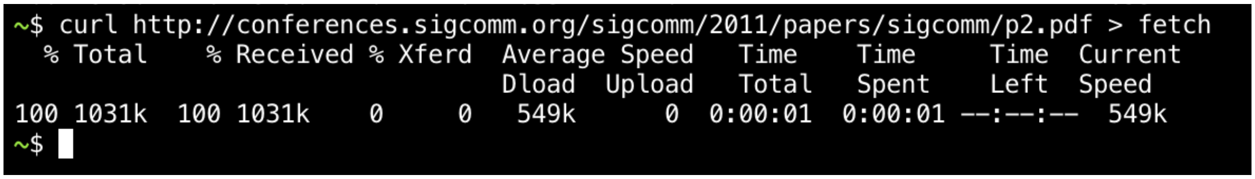
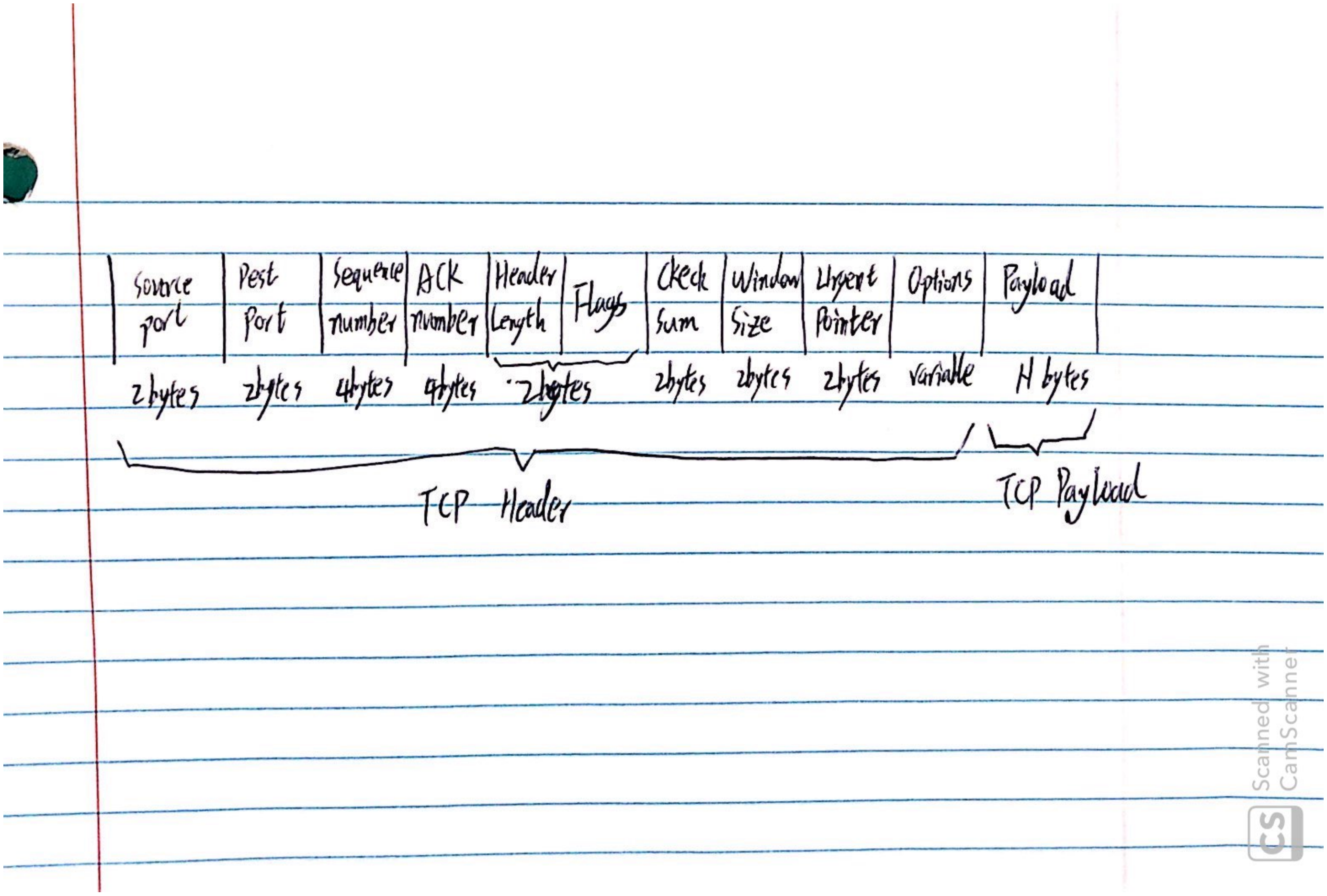


```
~$ curl http://conferences.sigcomm.org/sigcomm/2011/papers/sigcomm/p2.pdf > fetch
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 1031k  100 1031k    0     0   549k      0  0:00:01  0:00:01 --:--:--  549k
~$ 
```

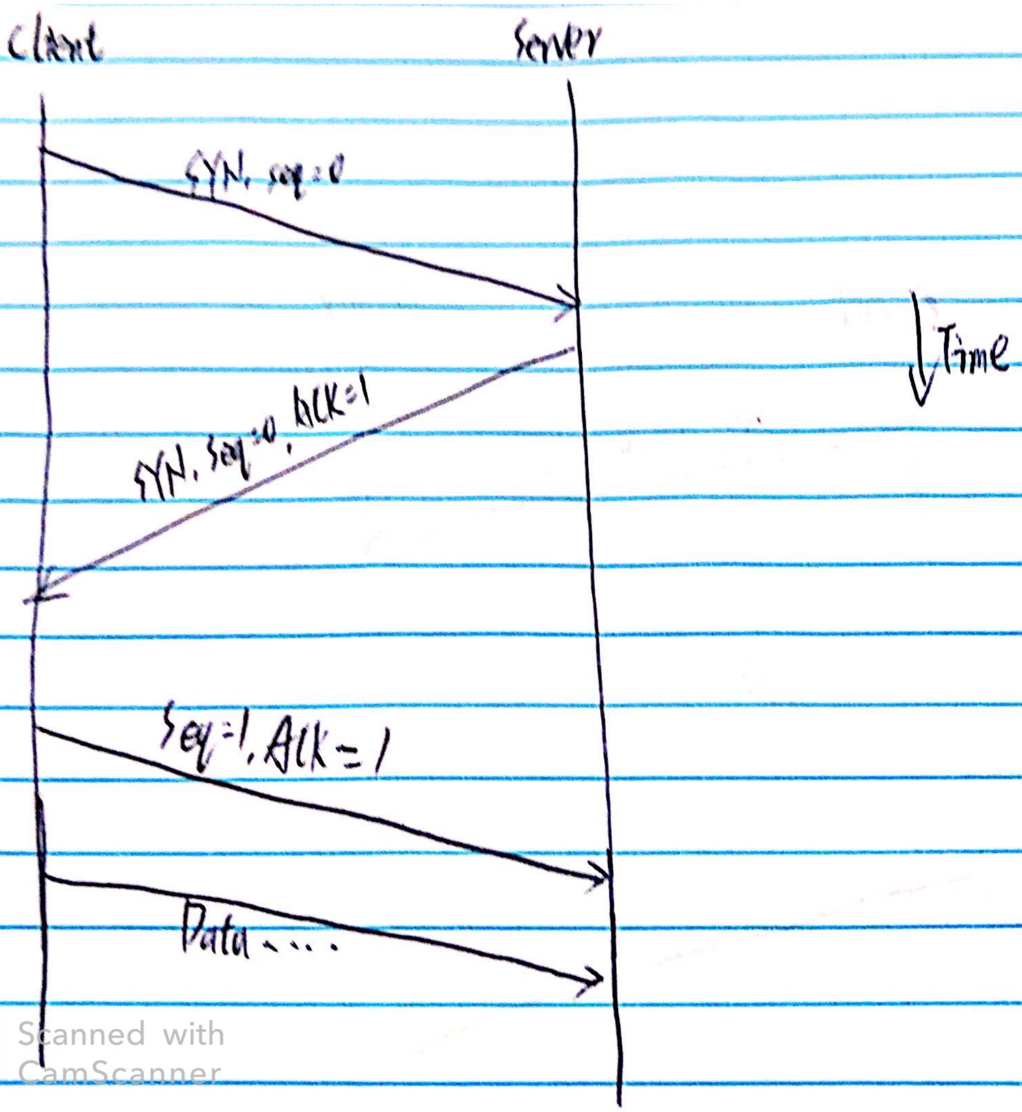
Step3:

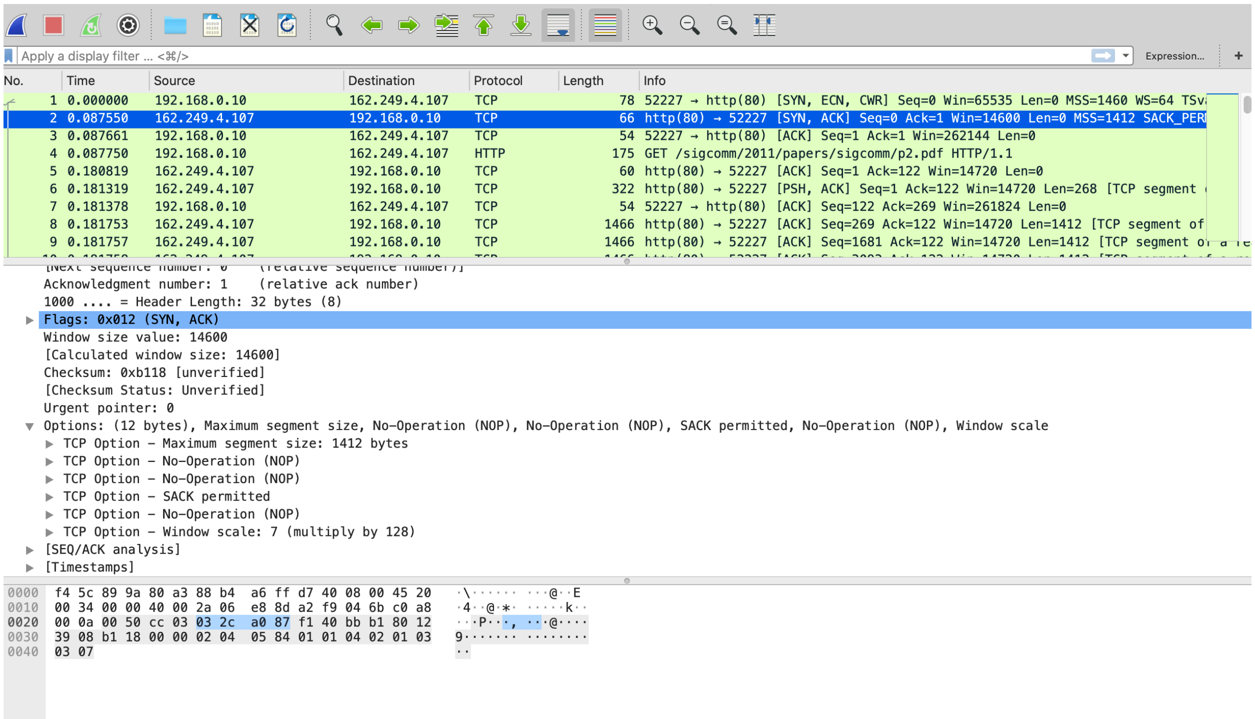
Turn-in: Hand in your drawing of a TCP segment.



Step 4: TCP Connection Setup/Teardown

Draw a time sequence diagram of the three-way handshake in your trace, up to and including the first data packet (the HTTP GET request) sent by your computer when the connection is established Put your computer on the left side and the remote server on the right side. As usual, time runs down the page, and lines across the page indicate segments


Client
Server
SYN, seq=0
Time
SYN, seq=0, ACK=1
Seq=1, ACK=1
Data ....

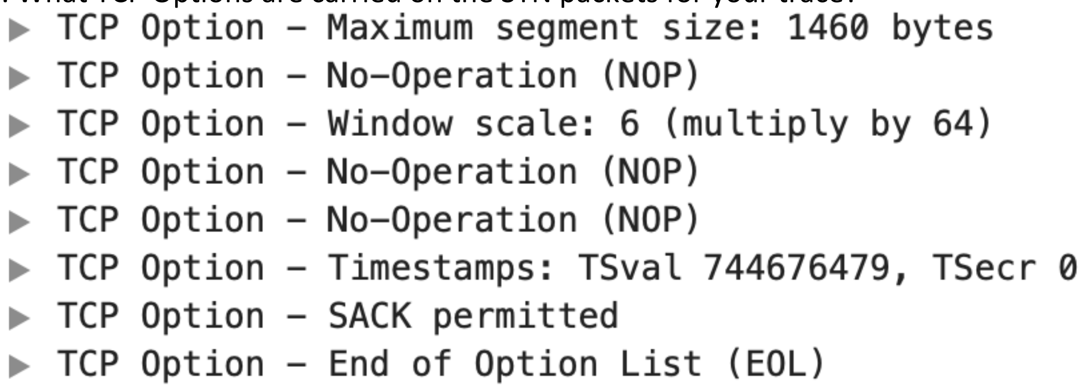


```
No.  Time      Source          Destination     Protocol  Length Info
   1 0.000000  192.168.0.10    162.249.4.107   TCP           78 52227 → http(80) [SYN, ECN, CWR] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSv
   2 0.087550  162.249.4.107   192.168.0.10    TCP           66 http(80) → 52227 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1412 SACK_PER
   3 0.087661  192.168.0.10    162.249.4.107   TCP           54 52227 → http(80) [ACK] Seq=1 Ack=1 Win=262144 Len=0
   4 0.087750  192.168.0.10    162.249.4.107   HTTP         175 GET /sigcomm/2011/papers/sigcomm/p2.pdf HTTP/1.1
   5 0.180819  162.249.4.107   192.168.0.10    TCP           60 http(80) → 52227 [ACK] Seq=1 Ack=122 Win=14720 Len=0
   6 0.181319  162.249.4.107   192.168.0.10    TCP          322 http(80) → 52227 [PSH, ACK] Seq=1 Ack=122 Win=14720 Len=268 [TCP segment
   7 0.181378  192.168.0.10    162.249.4.107   TCP           54 52227 → http(80) [ACK] Seq=122 Ack=269 Win=261824 Len=0
   8 0.181753  162.249.4.107   192.168.0.10    TCP         1466 http(80) → 52227 [ACK] Seq=269 Ack=122 Win=14720 Len=1412 [TCP segment of
   9 0.181757  162.249.4.107   192.168.0.10    TCP         1466 http(80) → 52227 [ACK] Seq=1681 Ack=122 Win=14720 Len=1412 [TCP segment of a re
```

```
    [Next sequence number: 0    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    1000 .... = Header Length: 32 bytes (8)
  ▶ Flags: 0x012 (SYN, ACK)
    Window size value: 14600
    [Calculated window size: 14600]
    Checksum: 0xb118 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▼ Options: (12 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted, No-Operation (NOP), Window scale
    ▶ TCP Option - Maximum segment size: 1412 bytes
    ▶ TCP Option - No-Operation (NOP)
    ▶ TCP Option - No-Operation (NOP)
    ▶ TCP Option - SACK permitted
    ▶ TCP Option - No-Operation (NOP)
    ▶ TCP Option - Window scale: 7 (multiply by 128)
  ▶ [SEQ/ACK analysis]
  ▶ [Timestamps]
```

```
0000  f4 5c 89 9a 80 a3 88 b4  a6 ff d7 40 08 00 45 20   ·\······ ···@··E
0010  00 34 00 00 40 00 2a 06  e8 8d a2 f9 04 6b c0 a8   ·4··@·*· ·····k··
0020  00 0a 00 50 cc 03 03 2c  a0 87 f1 40 bb b1 80 12   ···P··,  ···@····
0030  39 08 b1 18 00 00 02 04  05 84 01 01 04 02 01 03   9······· ········
0040  03 07                                              ··
```

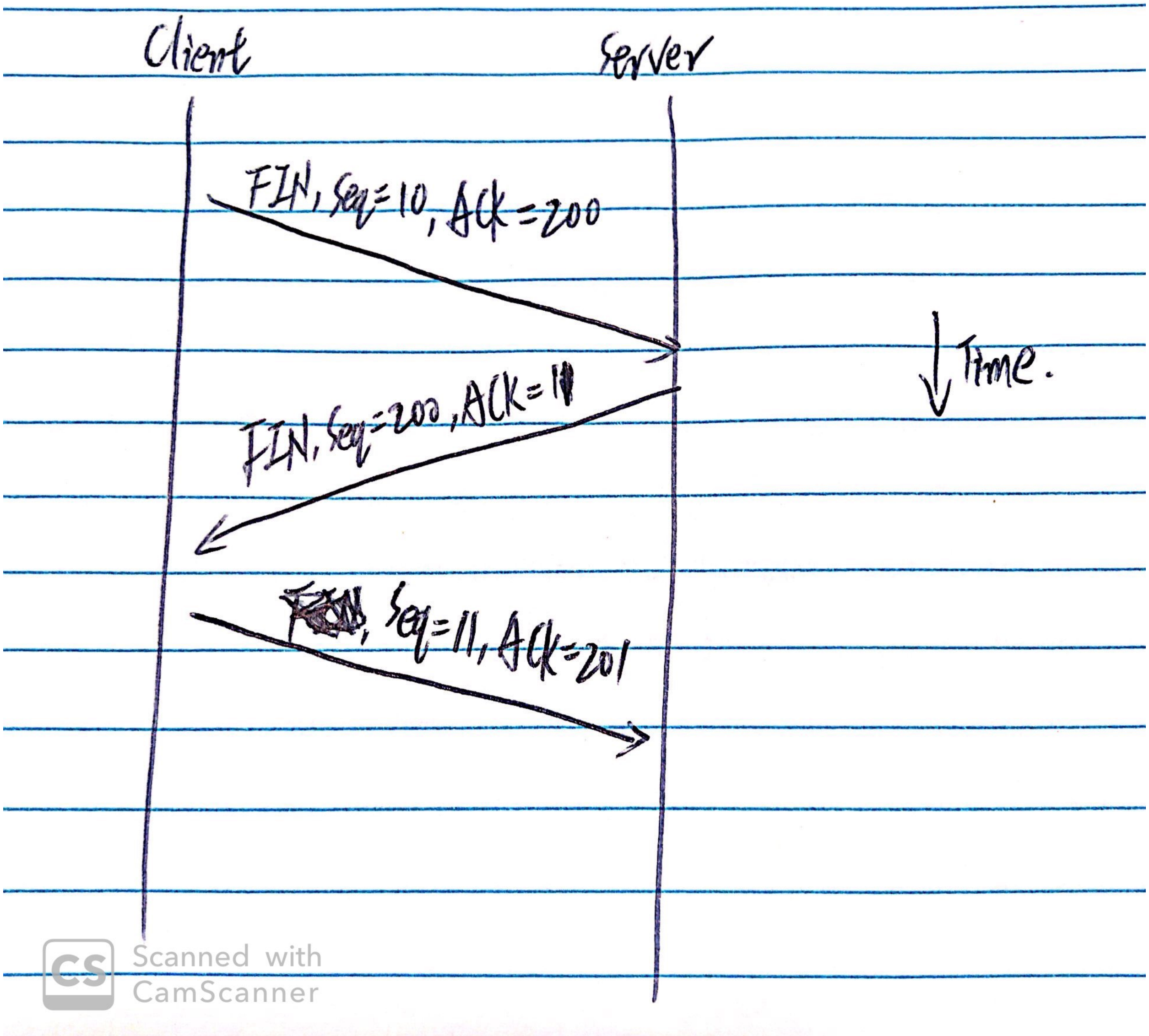
1. What TCP Options are carried on the SYN packets for your trace?



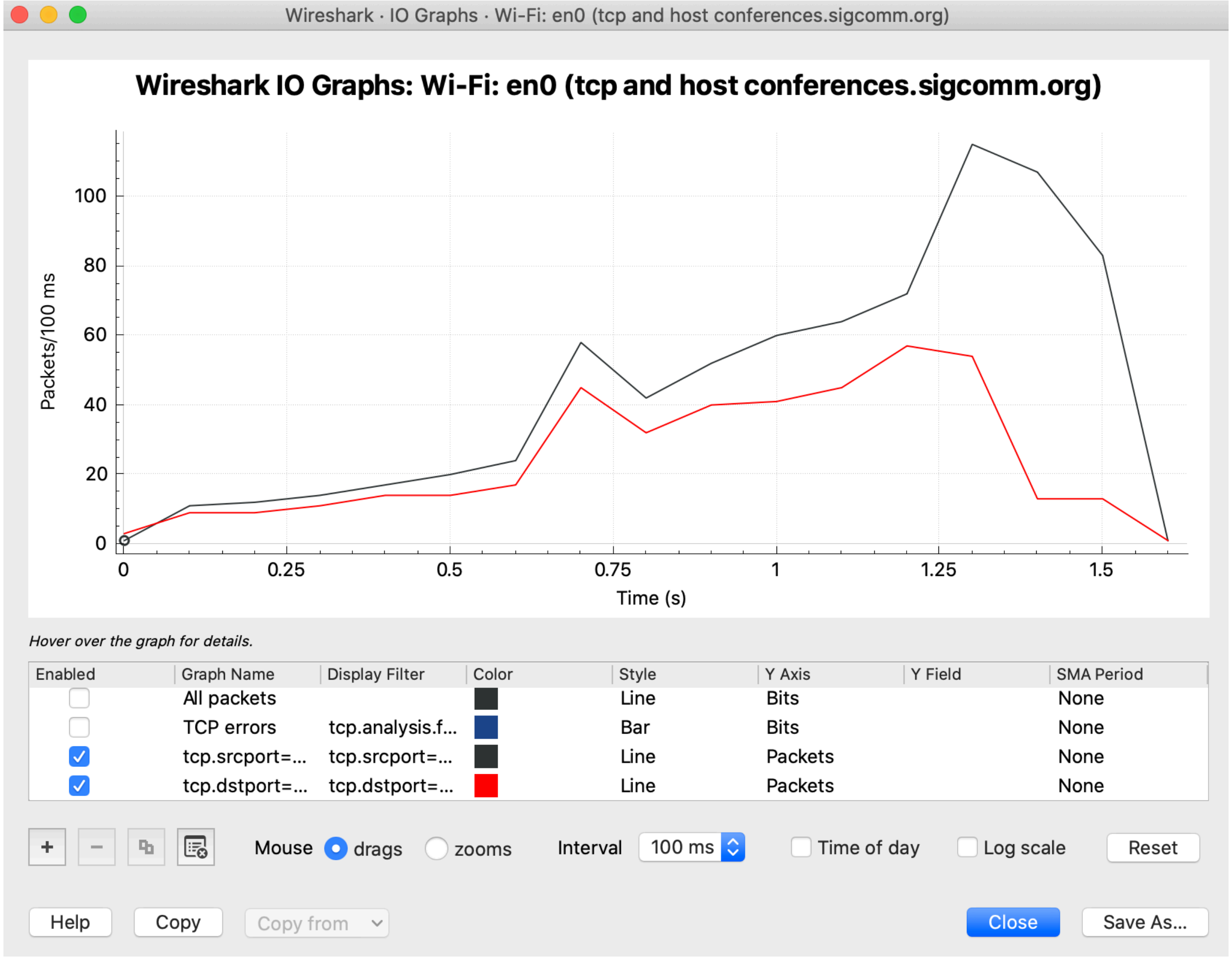
```
▶ TCP Option - Maximum segment size: 1460 bytes
▶ TCP Option - No-Operation (NOP)
▶ TCP Option - Window scale: 6 (multiply by 64)
▶ TCP Option - No-Operation (NOP)
▶ TCP Option - No-Operation (NOP)
▶ TCP Option - Timestamps: TSval 744676479, TSecr 0
▶ TCP Option - SACK permitted
▶ TCP Option - End of Option List (EOL)
```

Draw a picture of the teardown in your trace, starting from when the first FIN or RST is issued until the connection is complete. As before, show the sequence and ACK numbers on each segment. If you have

FINs then use the time difference to estimate the round-trip time.



Step 5: TCP Data Transfer



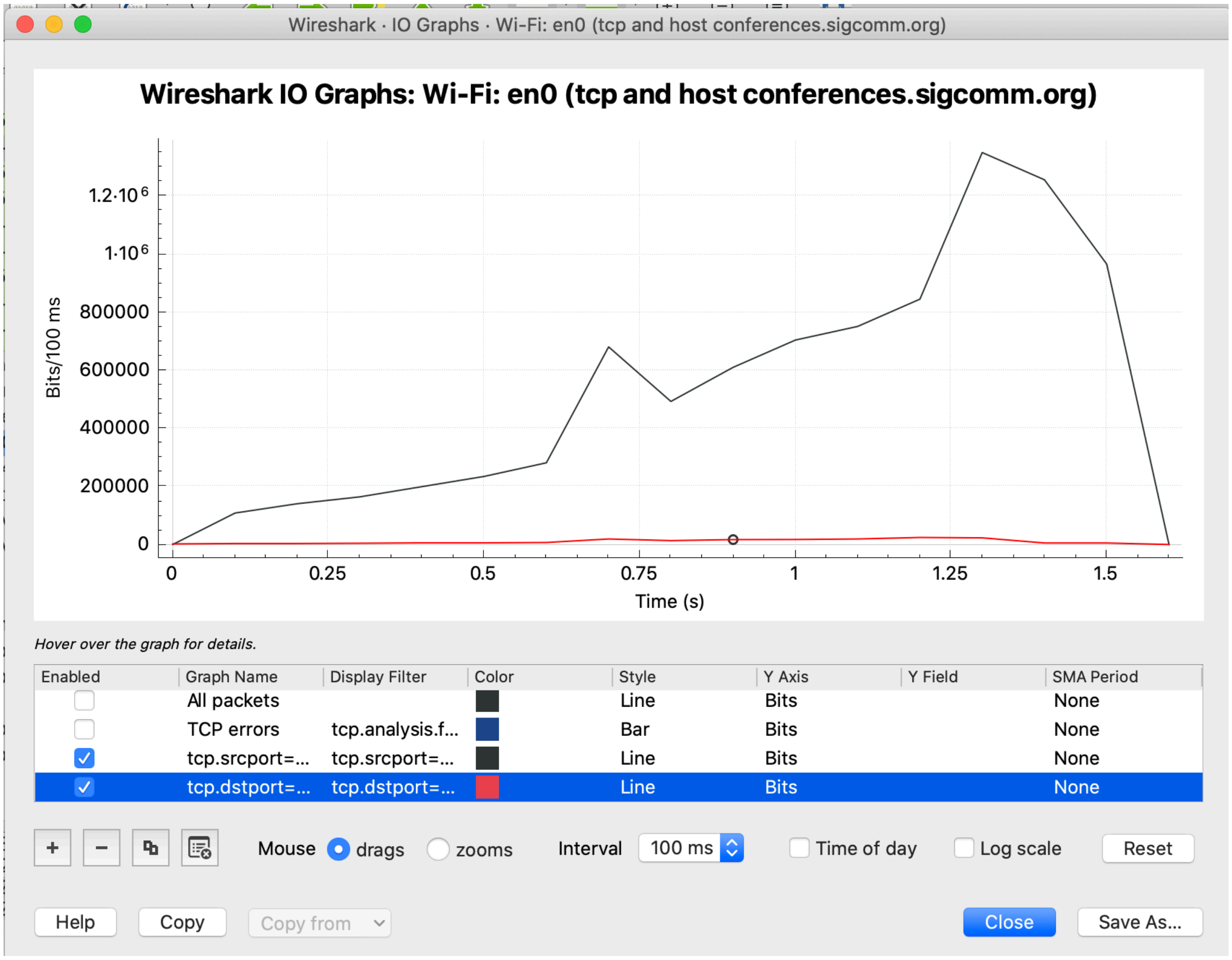
Wireshark · IO Graphs · Wi-Fi: en0 (tcp and host conferences.sigcomm.org)

Wireshark IO Graphs: Wi-Fi: en0 (tcp and host conferences.sigcomm.org)

Hover over the graph for details.

Enabled	Graph Name	Display Filter	Color	Style	Y Axis	Y Field	SMA Period
□	All packets		■	Line	Bits		None
□	TCP errors	tcp.analysis.f...	🟦	Bar	Bits		None
☑	tcp.srcport=...	tcp.srcport=...	■	Line	Packets		None
☑	tcp.dstport=...	tcp.dstport=...	🟥	Line	Packets		None

Mouse ● drags ○ zooms Interval 100 ms □ Time of day □ Log scale



Wireshark · IO Graphs · Wi-Fi: en0 (tcp and host conferences.sigcomm.org)

Wireshark IO Graphs: Wi-Fi: en0 (tcp and host conferences.sigcomm.org)

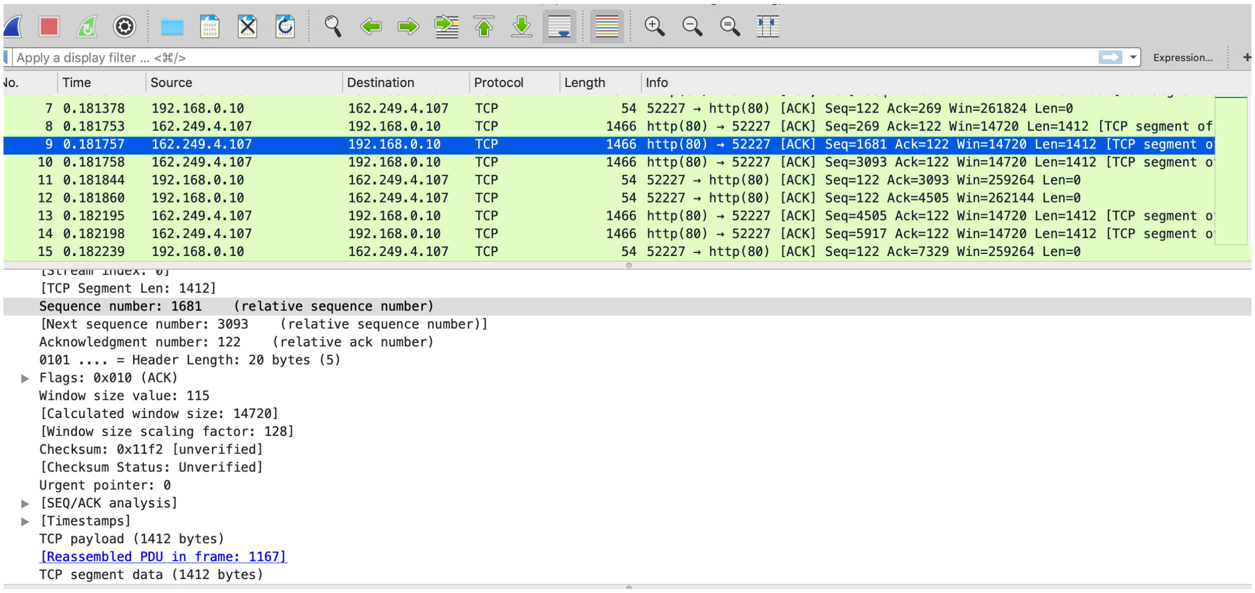
Hover over the graph for details.

Enabled	Graph Name	Display Filter	Color	Style	Y Axis	Y Field	SMA Period
☐	All packets		⬛	Line	Bits		None
☐	TCP errors	tcp.analysis.f...	🟦	Bar	Bits		None
☑	tcp.srcport=...	tcp.srcport=...	⬛	Line	Bits		None
☑	tcp.dstport=...	tcp.dstport=...	🟥	Line	Bits		None

Mouse ● drags ○ zooms   Interval 100 ms   ☐ Time of day   ☐ Log scale

1. What is the rough data rate in the download direction in packets/second and bits/second once the TCP connection is running well?
800 packets/sec
8000000 bits/sec

2. What percentage of this download rate is content? Show your calculation. To find out, look at a typical download packet; there should be many similar, large download packets. You can see how long it is, and how many bytes of TCP payload it contains.



```
No.  Time      Source          Destination     Protocol  Length  Info
  7 0.181378   192.168.0.10    162.249.4.107   TCP          54 52227 → http(80) [ACK] Seq=122 Ack=269 Win=261824 Len=0
  8 0.181753   162.249.4.107   192.168.0.10    TCP        1466 http(80) → 52227 [ACK] Seq=269 Ack=122 Win=14720 Len=1412 [TCP segment of
  9 0.181757   162.249.4.107   192.168.0.10    TCP        1466 http(80) → 52227 [ACK] Seq=1681 Ack=122 Win=14720 Len=1412 [TCP segment o
 10 0.181758   162.249.4.107   192.168.0.10    TCP        1466 http(80) → 52227 [ACK] Seq=3093 Ack=122 Win=14720 Len=1412 [TCP segment o
 11 0.181844   192.168.0.10    162.249.4.107   TCP          54 52227 → http(80) [ACK] Seq=122 Ack=3093 Win=259264 Len=0
 12 0.181860   192.168.0.10    162.249.4.107   TCP          54 52227 → http(80) [ACK] Seq=122 Ack=4505 Win=262144 Len=0
 13 0.182195   162.249.4.107   192.168.0.10    TCP        1466 http(80) → 52227 [ACK] Seq=4505 Ack=122 Win=14720 Len=1412 [TCP segment o
 14 0.182198   162.249.4.107   192.168.0.10    TCP        1466 http(80) → 52227 [ACK] Seq=5917 Ack=122 Win=14720 Len=1412 [TCP segment o
 15 0.182239   192.168.0.10    162.249.4.107   TCP          54 52227 → http(80) [ACK] Seq=122 Ack=7329 Win=259264 Len=0
```

```
    [Stream index: 0]
    [TCP Segment Len: 1412]
    Sequence number: 1681    (relative sequence number)
    [Next sequence number: 3093    (relative sequence number)]
    Acknowledgment number: 122    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
  ▶ Flags: 0x010 (ACK)
    Window size value: 115
    [Calculated window size: 14720]
    [Window size scaling factor: 128]
    Checksum: 0x11f2 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▶ [SEQ/ACK analysis]
  ▶ [Timestamps]
    TCP payload (1412 bytes)
    [Reassembled PDU in frame: 1167]
    TCP segment data (1412 bytes)
```

1412 bytes payload, total length is 1466, so rate is 96.3%

3. What is the rough data rate in the upload direction in packets/second and bits/second due to the ACK packets?
400 packets/second

4. If the most recently received TCP segment from the server has a sequence number of X, then what ACK number does the next transmitted TCP segment carry?
The Ack number tells the next expected sequence number so it will be X plus the number of TCP payload bytes in the data segment.