Step 1. Capture a trace

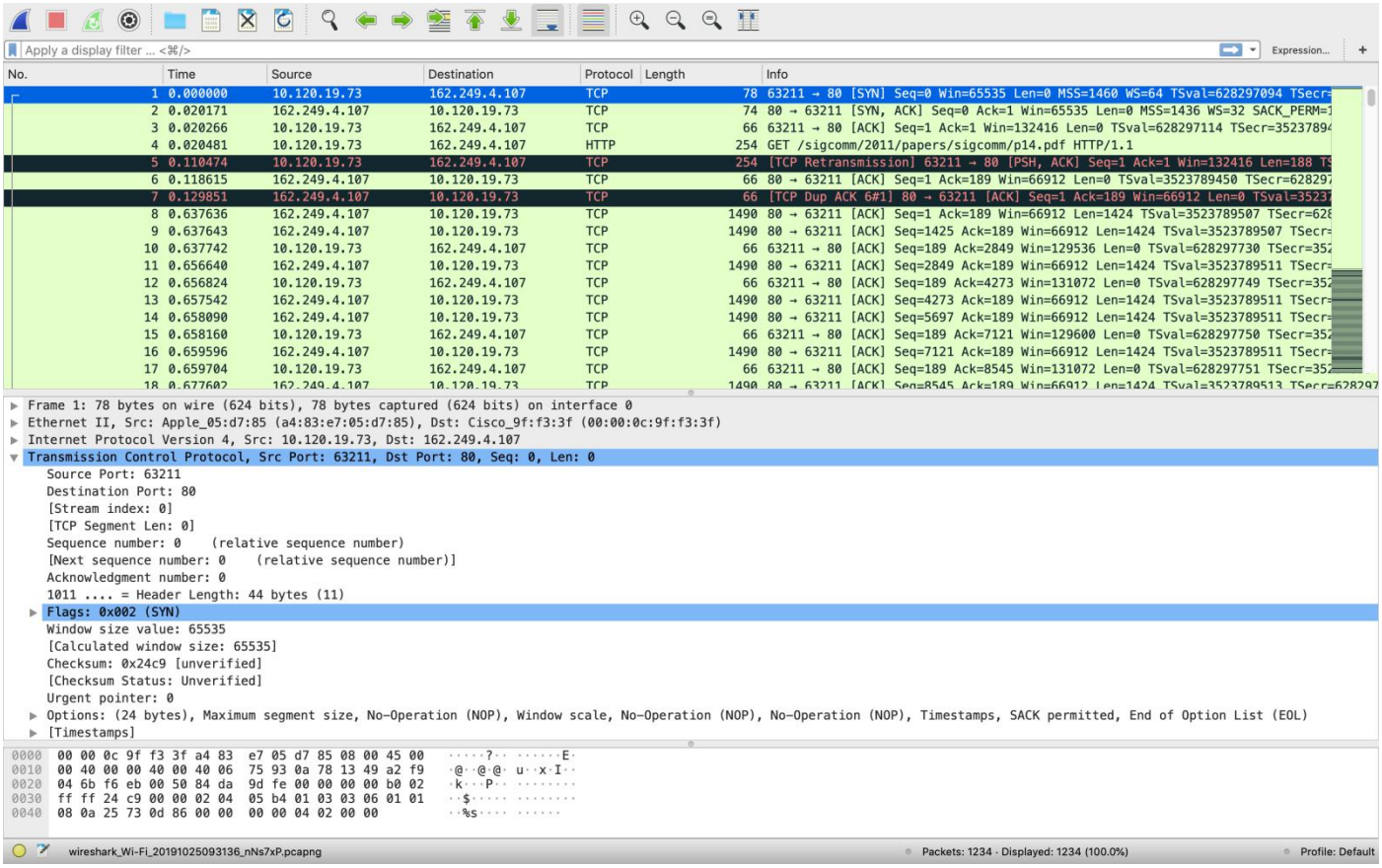
1) Find a URL, I am using this one, http://conferences.sigcomm.org/sigcomm/2011/papers/sigcomm/p14.pdf, Which size is about 900KB.
2) Fetch the URL with wget command.

```
➜  ~ wget http://conferences.sigcomm.org/sigcomm/2011/papers/sigcomm/p14.pdf
--2019-10-25 09:30:22--  http://conferences.sigcomm.org/sigcomm/2011/papers/sigcomm/p14.pdf
Resolving conferences.sigcomm.org (conferences.sigcomm.org)... 162.249.4.107
Connecting to conferences.sigcomm.org (conferences.sigcomm.org)|162.249.4.107|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 964684 (942K) [application/pdf]
Saving to: ‘p14.pdf’

p14.pdf             100%[===================>] 942.07K  1.36MB/s    in 0.7s

2019-10-25 09:30:24 (1.36 MB/s) - ‘p14.pdf’ saved [964684/964684]
```

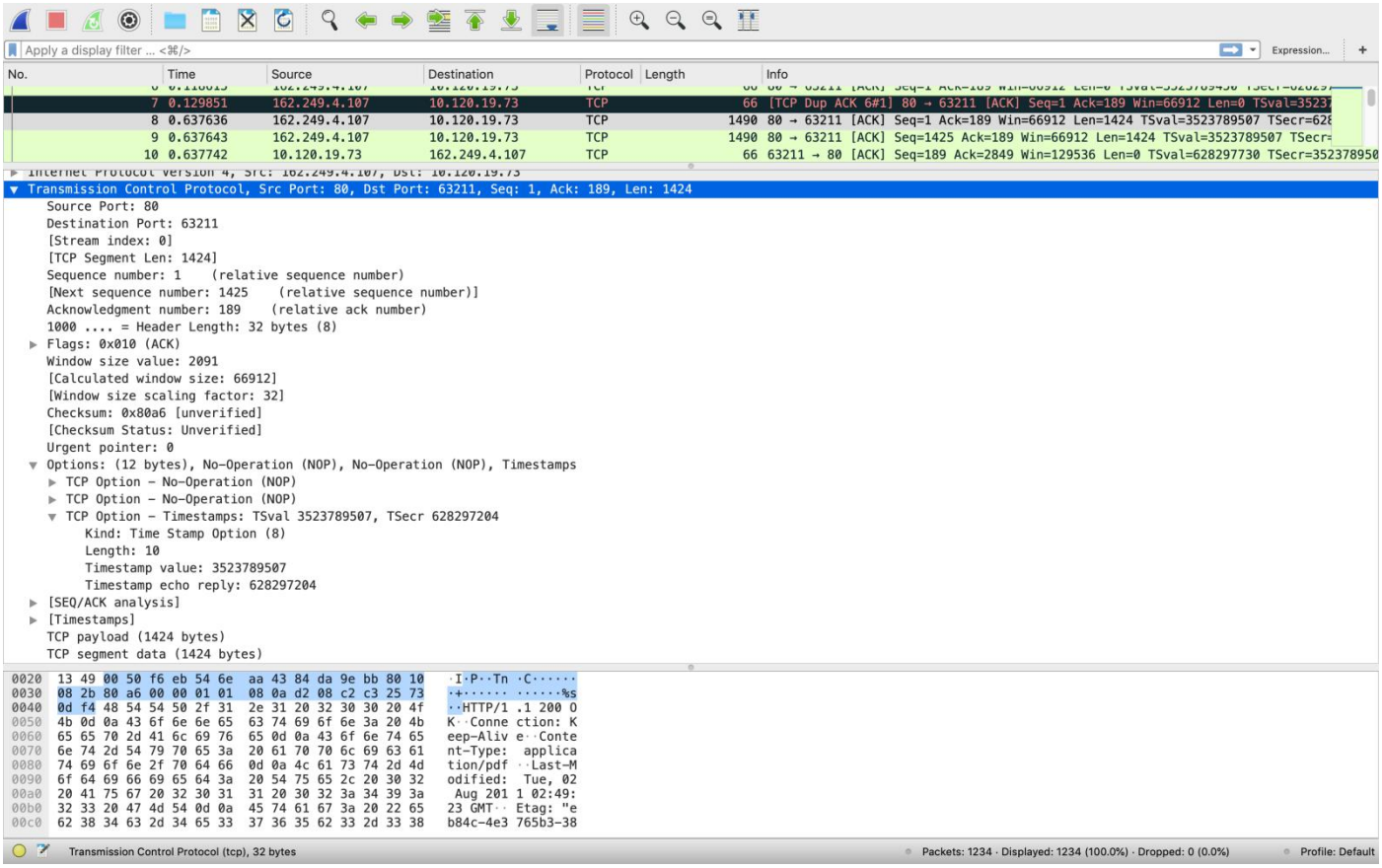
3) Set up the filter to be ‘tcp and host conferences.sigcomm.org’, then repeat the wget command in 2) step.



Step 2 & 3. Inspect the Trace and TCP Segment Structure

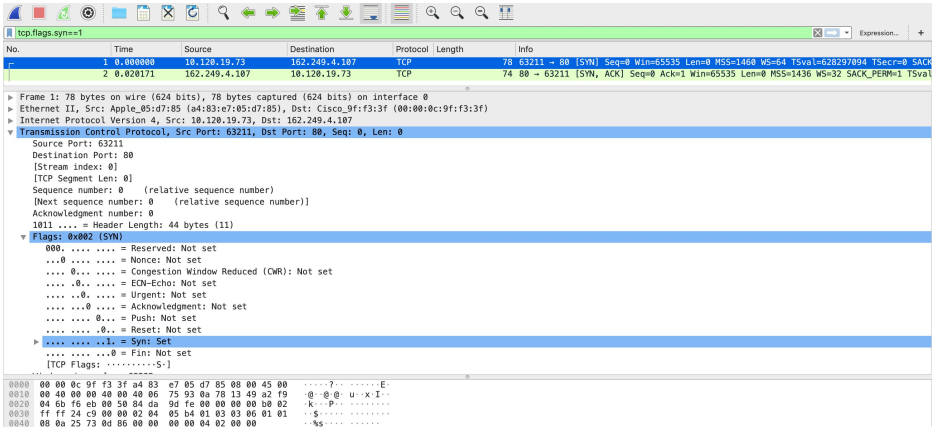
Expand a segment of trace, I find out the detailed structure of a tcp header, it looks like this:

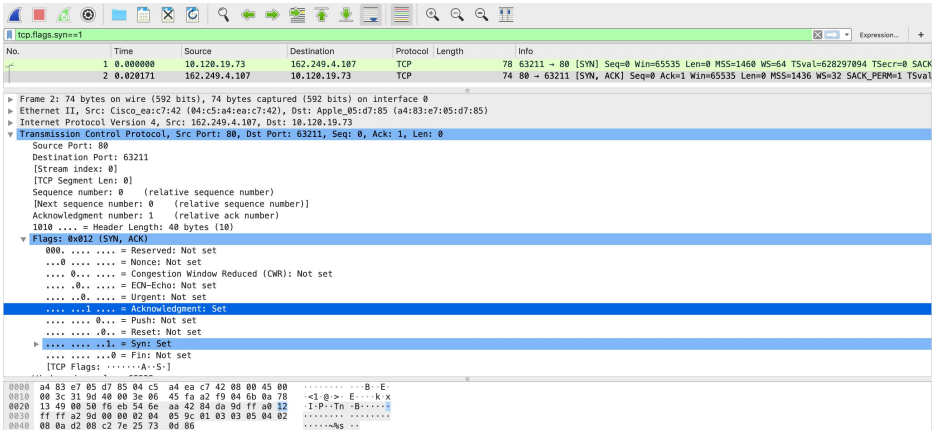
Source port	Dest Port	Sequence Number	Ack Number	Heade Length + Flags	Window size	Check sum	Urgent pointer	options	payloads
2 bytes	2 bytes	4 bytes	4 bytes	2 bytes	2 bytes	2 bytes	2 bytes	12 bytes	1424 bytes



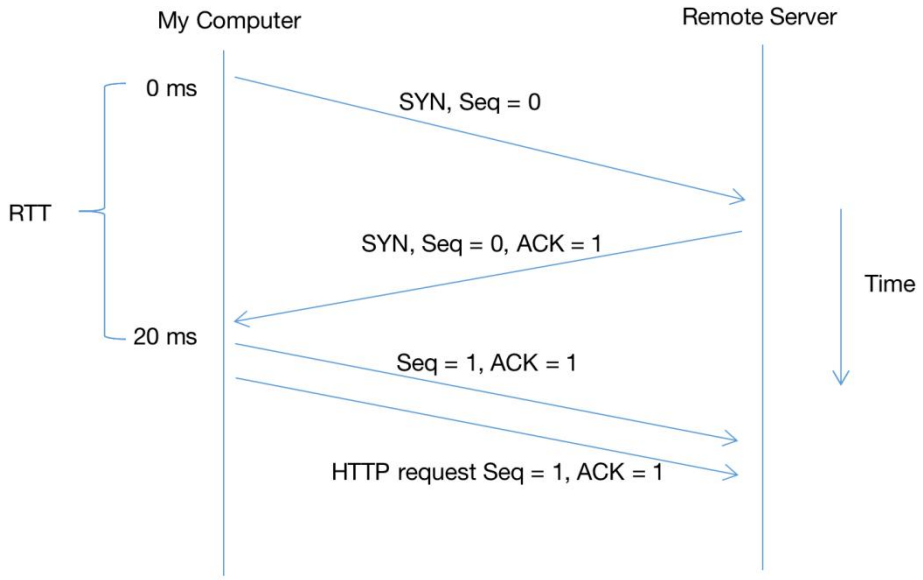
Step 4. TCP Connection Setup and Teardown

1. Filter the syn packets with filter 'tcp.flags.syn==1'





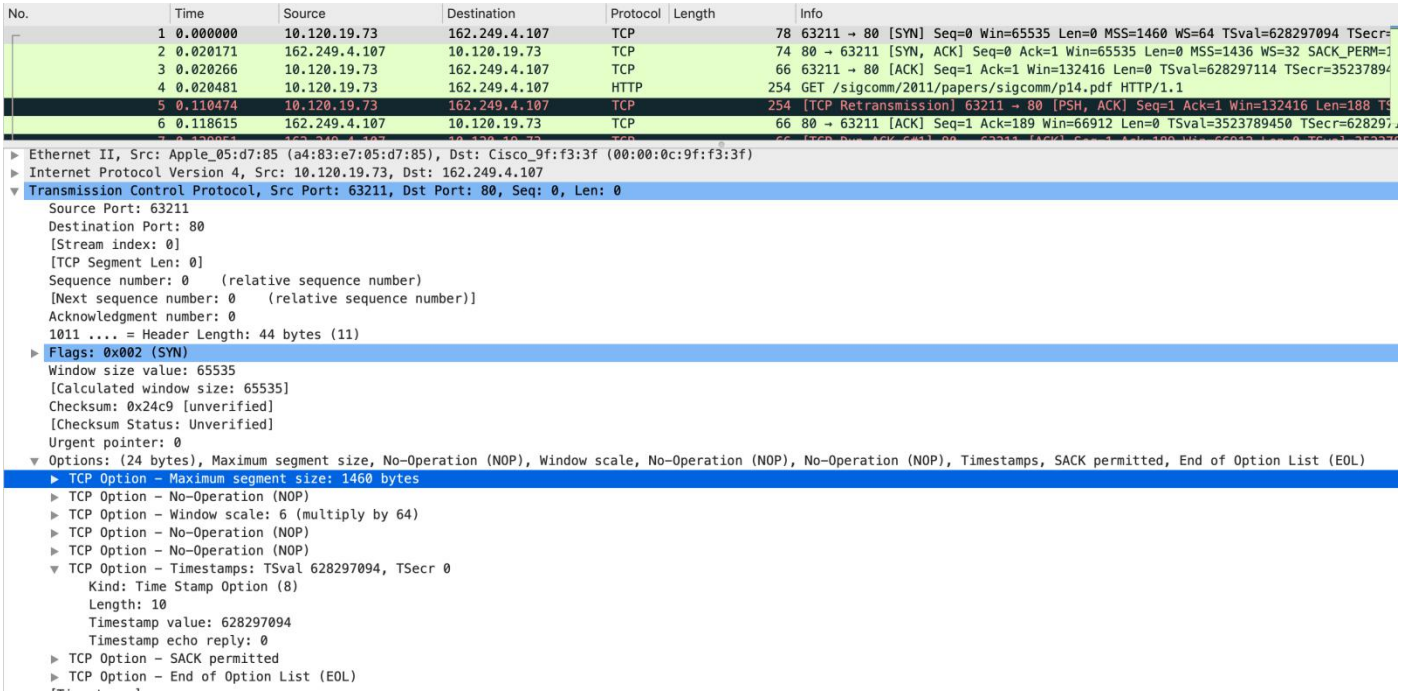
2. Draw a time sequence diagram for the three-way handshake.



Step 5. Connection Options

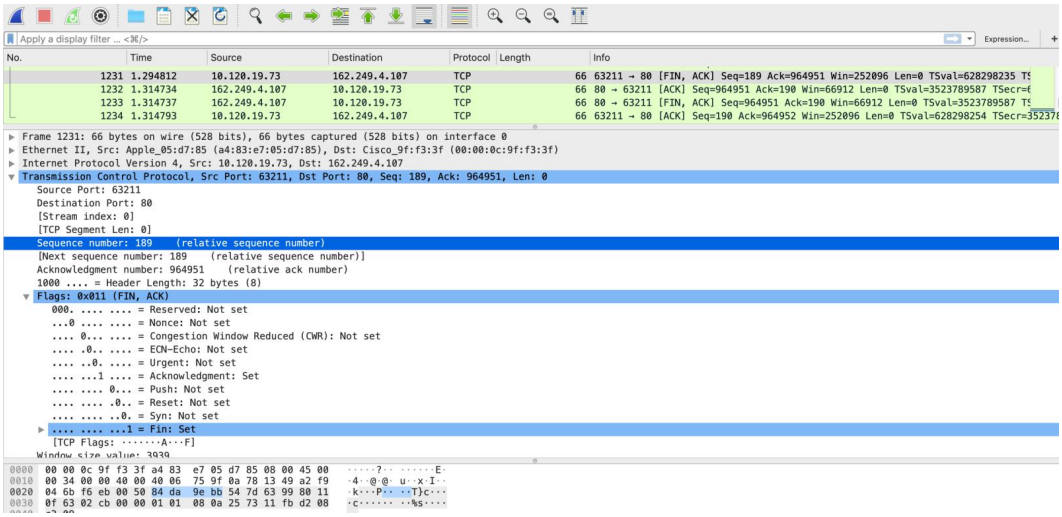
1. What TCP Options are carried on the SYN packets for your trace?

The tcp options are Maximum segment size, No-Operation(NOP), Window scale, No-Operation(NOP), Timestamps, SACK permitted, End of Option List(EOL).

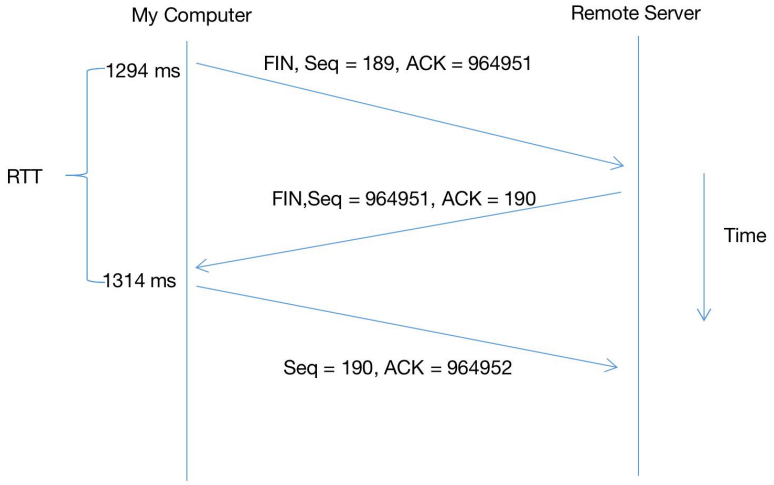


Step 6. FIN/RST Tear down

1. Check the FIN packets

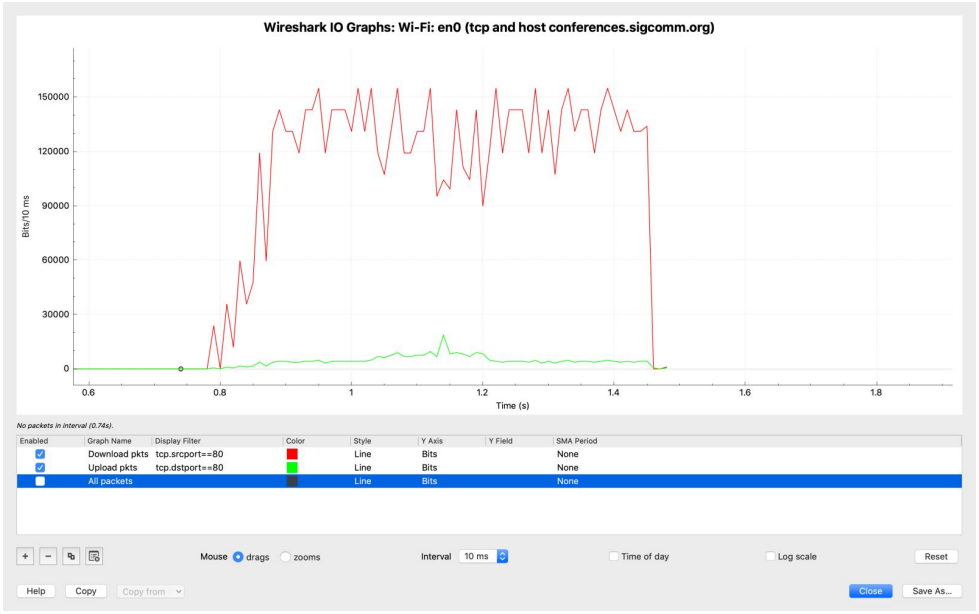


2. Draw a diagram for the tear down.



Step 7. TCP Data Transfer

1. TCP Stream Graph.

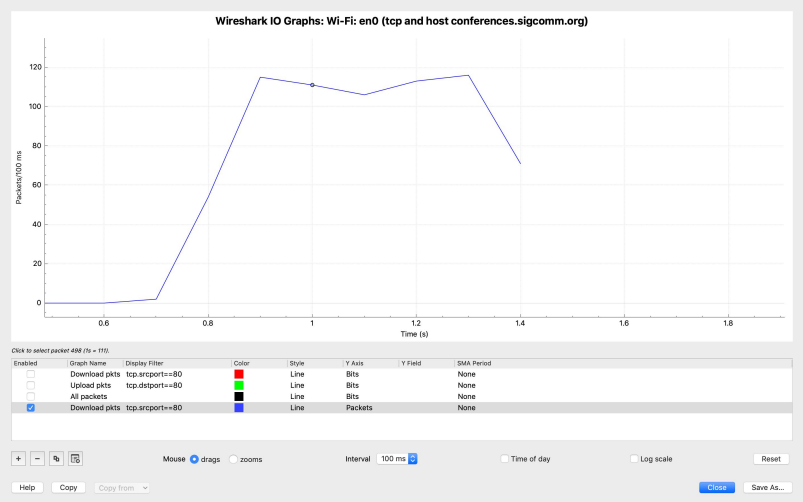


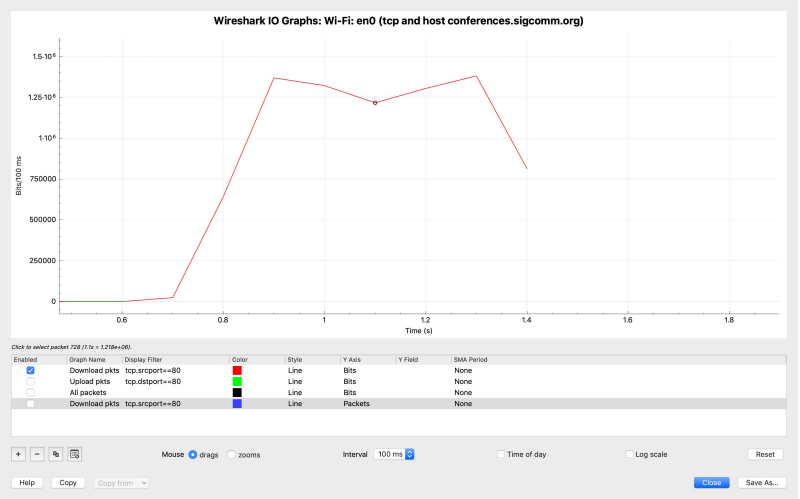
2. Answer the following questions to show your understanding of the data transfer:

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

Download rate in packets/s is around 100 (pkts/100ms) * 10 = 1000 pkts.

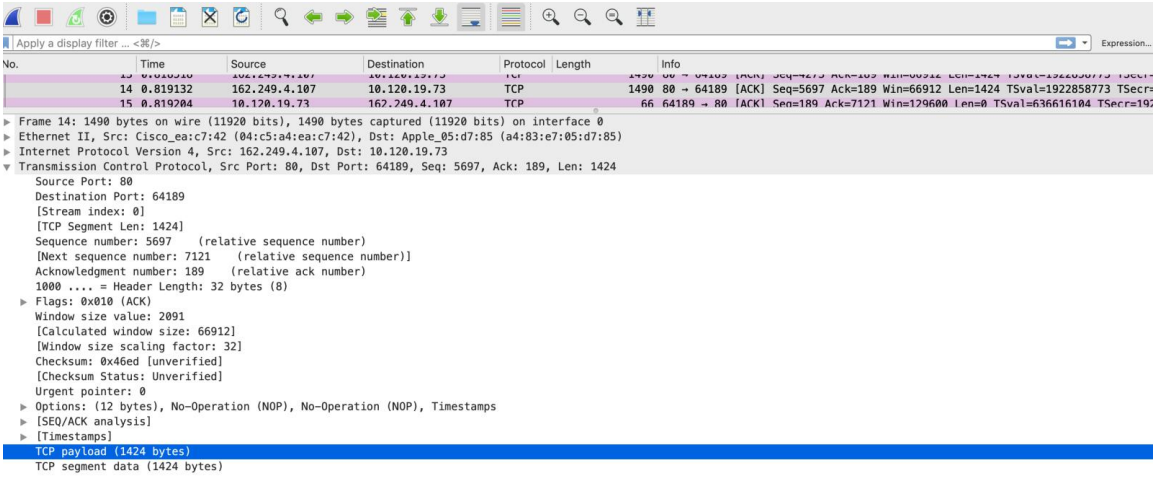
in bits/s is around 1.25 * 10^6 (bits/100ms) * 10 = 12.5 * 10^6 bits





2) What percentage of this download rate is content?

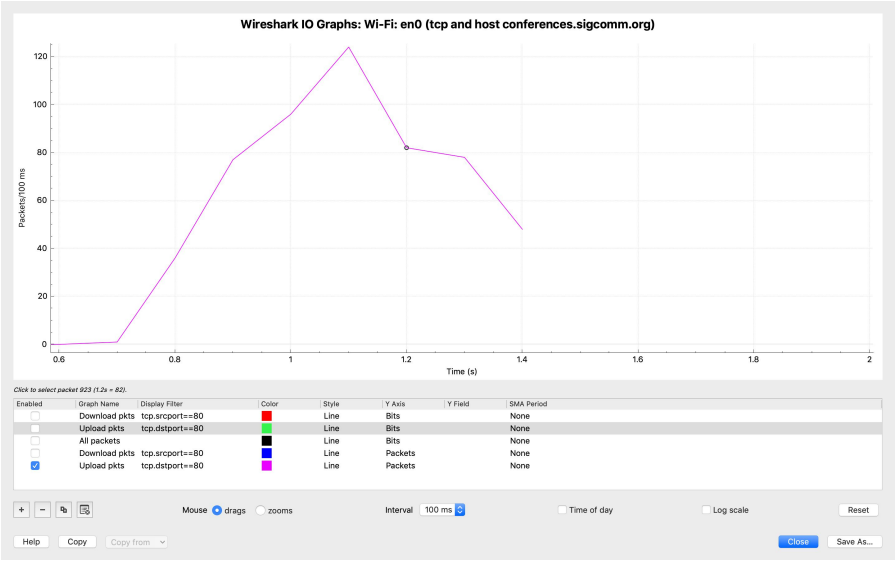
The packet is 1490 bytes and payload is 1424 bytes, thus the content percentage is 1424 / 1490 = 95%.

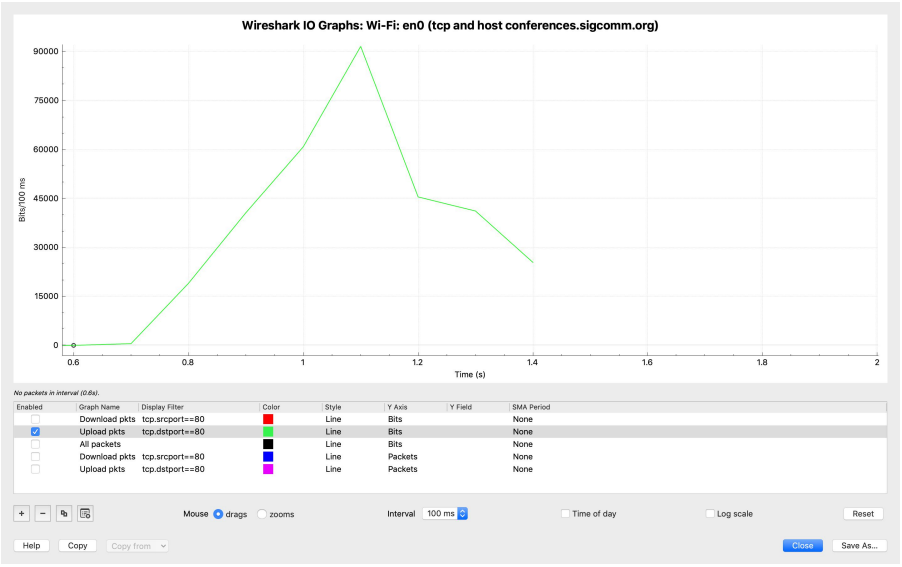


3) What is the rough data rate in the upload direction in packets/second and bits/second due to the ACK packets?

Download rate in packets/s is around 85 (pkts/100ms) * 10 = 850 pkts.

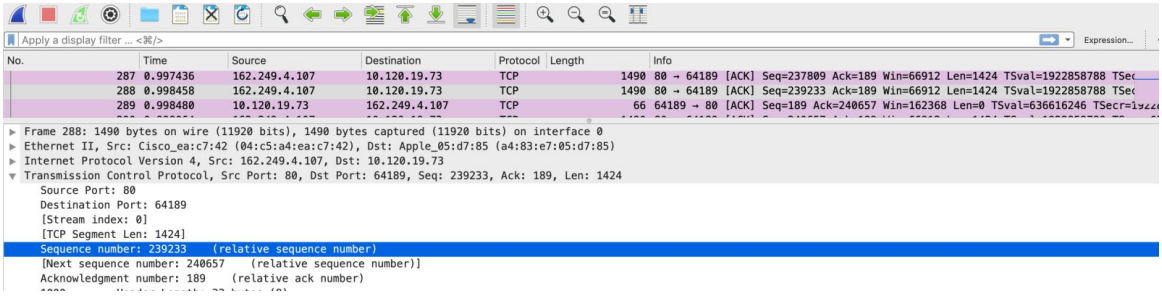
in bits/s is around 0.06 * 10^6 (bits/100ms) * 10 = 0.6 * 10^6 bits

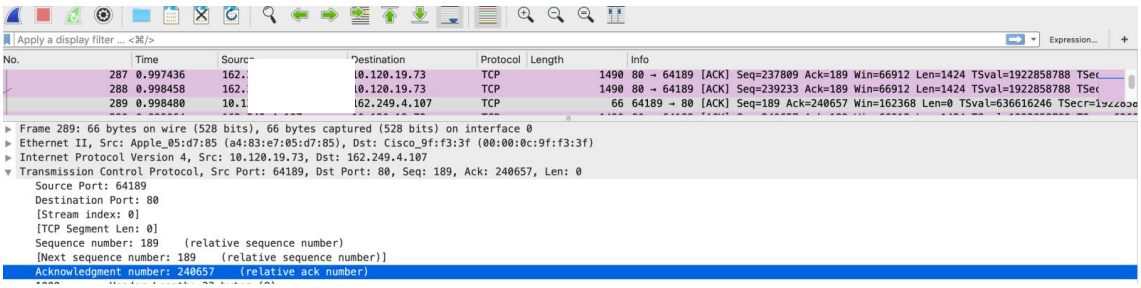




4) If the most recently received TCP segment from the server has a sequence number of X, then what ACK number does the next transmitted TCP segment carry?

As is shown below, the segment from the server has seq num 239233, and the next transmitted segment has an ACK of 240657, we could see that 240657 - 239233 = 1424, which is the payload bytes number. Thus if the seq num is X, then the next transmitted ACK will be X plus the payload bytes.




Apply a display filter ... <⌘/>
Expression...
No. Time Source Destination Protocol Length Info
287 0.997436 162. 10.120.19.73 TCP 1490 80 → 64189 [ACK] Seq=237809 Ack=189 Win=66912 Len=1424 TSval=1922858788 TSec
288 0.998458 162. 10.120.19.73 TCP 1490 80 → 64189 [ACK] Seq=239233 Ack=189 Win=66912 Len=1424 TSval=1922858788 TSec
289 0.998480 10.1 162.249.4.107 TCP 66 64189 → 80 [ACK] Seq=189 Ack=240657 Win=162368 Len=0 TSval=636616246 TSecr=1922
Frame 289: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: Apple_05:d7:85 (a4:83:e7:05:d7:85), Dst: Cisco_9f:f3:3f (00:00:0c:9f:f3:3f)
Internet Protocol Version 4, Src: 10.120.19.73, Dst: 162.249.4.107
Transmission Control Protocol, Src Port: 64189, Dst Port: 80, Seq: 189, Ack: 240657, Len: 0
Source Port: 64189
Destination Port: 80
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 189 (relative sequence number)
[Next sequence number: 189 (relative sequence number)]
Acknowledgment number: 240657 (relative ack number)