Task 1

a)

1072	80.783782	192.168.1.96	128.119.245.12	HTTP	565 GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
1077	80.891862	128.119.245.12	192.168.1.96	HTTP	535 HTTP/1.1 200 OK (text/html)
1079	80.975142	192.168.1.96	128.119.245.12	HTTP	511 GET /favicon.ico HTTP/1.1
1080	81.083011	128.119.245.12	192.168.1.96	HTTP	538 HTTP/1.1 404 Not Found (text/html)

The browser has sent one GET request excluding the favicon request.

b)

The second packet contains the response. The status code and phrase can be found in the top of the HTTP header.

```
Hypertext Transfer Protocol
 HTTP/1.1 200 OK\r\n
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
```

c)

4500 bytes

```
 Content-Length: 4500\r\n
      [Content length: 4500]
```

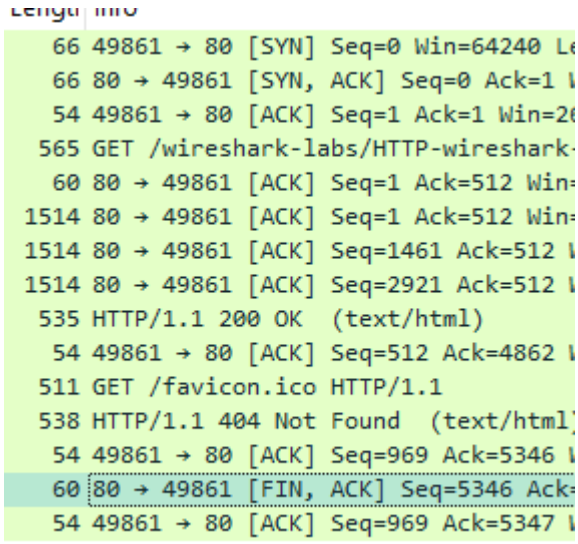
d)

4 segments. HTTP_Header_Length = 4861-4500 = 361 bytes

```
 [4 Reassembled TCP Segments (4861 bytes): #1074(1460), #1075(1460), #1076(1460), #1077(481)]
      [Frame: 1074, payload: 0-1459 (1460 bytes)]
      [Frame: 1075, payload: 1460-2919 (1460 bytes)]
      [Frame: 1076, payload: 2920-4379 (1460 bytes)]
      [Frame: 1077, payload: 4380-4860 (481 bytes)]
      [Segment count: 4]
      [Reassembled TCP length: 4861]
```

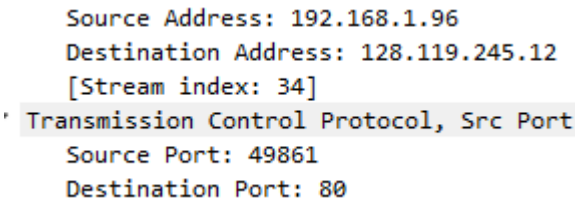
Task 2

a)



The same connection was used since the finish flag terminating the connection was used once and that was after the favicon request putting it inside the persistent connection.

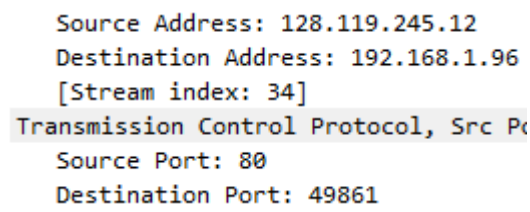
b)



Source IP is 192.168.1.96

Source Port is 49861

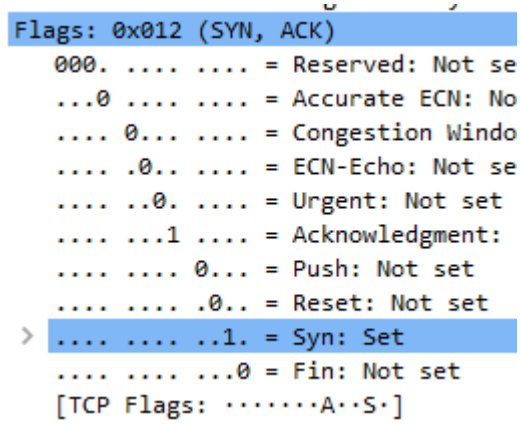
c)

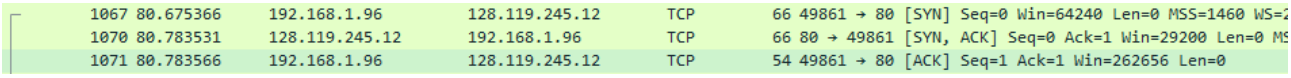


Source IP is 128.119.145.12

Source Port is 80

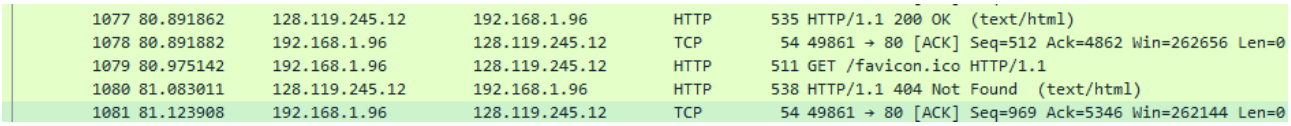
d)





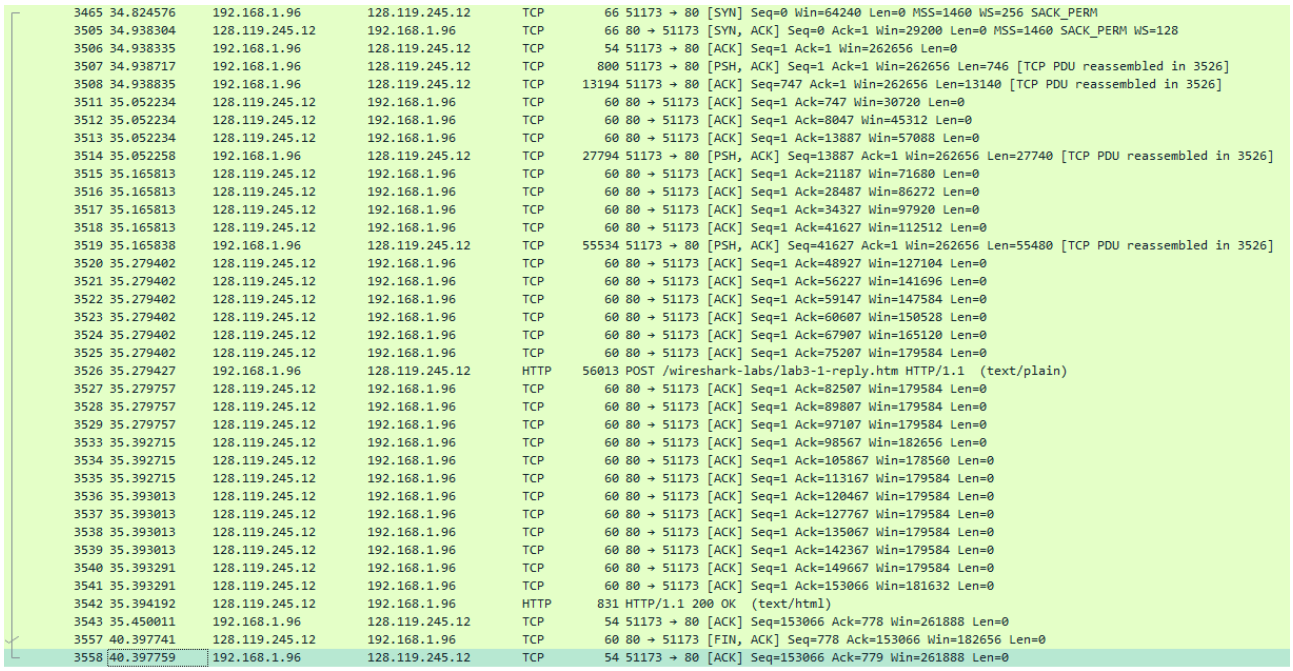
3 segments. Header flags for initiating connection SYN : SYN,ACK : ACK.

e)



The ACK to the server for receiveing the webpage and the ACK for recieving the response for the nonexistence of favicon.

Task 3



3465	34.824576	192.168.1.96	128.119.245.12	TCP	66	51173 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
3505	34.938304	128.119.245.12	192.168.1.96	TCP	66	80 → 51173 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
3506	34.938335	192.168.1.96	128.119.245.12	TCP	54	51173 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
3507	34.938717	192.168.1.96	128.119.245.12	TCP	800	51173 → 80 [PSH, ACK] Seq=1 Ack=1 Win=262656 Len=746 [TCP PDU reassembled in 3526]
3508	34.938835	192.168.1.96	128.119.245.12	TCP	13194	51173 → 80 [ACK] Seq=747 Ack=1 Win=262656 Len=13140 [TCP PDU reassembled in 3526]
3511	35.052234	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=747 Win=30720 Len=0
3512	35.052234	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=8047 Win=45312 Len=0
3513	35.052234	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=13887 Win=57088 Len=0
3514	35.052258	192.168.1.96	128.119.245.12	TCP	27794	51173 → 80 [PSH, ACK] Seq=13887 Ack=1 Win=262656 Len=27740 [TCP PDU reassembled in 3526]
3515	35.165813	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=21187 Win=71680 Len=0
3516	35.165813	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=28487 Win=86272 Len=0
3517	35.165813	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=34327 Win=97920 Len=0
3518	35.165813	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=41627 Win=112512 Len=0
3519	35.165838	192.168.1.96	128.119.245.12	TCP	55534	51173 → 80 [PSH, ACK] Seq=41627 Ack=1 Win=262656 Len=55480 [TCP PDU reassembled in 3526]
3520	35.279402	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=48927 Win=127104 Len=0
3521	35.279402	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=56227 Win=141696 Len=0
3522	35.279402	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=59147 Win=147584 Len=0
3523	35.279402	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=60607 Win=150528 Len=0
3524	35.279402	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=67907 Win=165120 Len=0
3525	35.279402	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=75207 Win=179584 Len=0
3526	35.279427	192.168.1.96	128.119.245.12	HTTP	56013	POST /wireshark-labs/lab3-1-reply.htm HTTP/1.1 (text/plain)
3527	35.279757	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=82507 Win=179584 Len=0
3528	35.279757	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=89807 Win=179584 Len=0
3529	35.279757	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=97107 Win=179584 Len=0
3533	35.392715	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=98567 Win=182656 Len=0
3534	35.392715	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=105867 Win=178560 Len=0
3535	35.392715	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=113167 Win=179584 Len=0
3536	35.393013	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=120467 Win=179584 Len=0
3537	35.393013	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=127767 Win=179584 Len=0
3538	35.393013	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=135067 Win=179584 Len=0
3539	35.393013	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=142367 Win=179584 Len=0
3540	35.393291	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=149667 Win=179584 Len=0
3541	35.393291	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [ACK] Seq=1 Ack=153066 Win=181632 Len=0
3542	35.394192	128.119.245.12	192.168.1.96	HTTP	831	HTTP/1.1 200 OK (text/html)
3543	35.450011	192.168.1.96	128.119.245.12	TCP	54	51173 → 80 [ACK] Seq=153066 Ack=778 Win=261888 Len=0
3557	40.397741	128.119.245.12	192.168.1.96	TCP	60	80 → 51173 [FIN, ACK] Seq=778 Ack=153066 Win=182656 Len=0
3558	40.397759	192.168.1.96	128.119.245.12	TCP	54	51173 → 80 [ACK] Seq=153066 Ack=779 Win=261888 Len=0

a)

```
[Frame: 3507, payload: 0-745 (746 bytes)]
[Frame: 3508, payload: 746-13885 (13140 bytes)]
[Frame: 3514, payload: 13886-41625 (27740 bytes)]
[Frame: 3519, payload: 41626-97105 (55480 bytes)]
[Frame: 3526, payload: 97106-153064 (55959 bytes)]
```

It increases with each packet. But the last two packets indicated the link handled packets with a payload of little over 55 000 bytes.

b)

The value of the Acknowledgement number of a packet is the sequence number + the length of the tcp packet it's acknowleding.

```
[TCP Segment Len: 746]
Sequence Number: 1    (relative sequence number)
Sequence Number (raw): 3027493083
[Next Sequence Number: 747    (relative sequence number)]
Acknowledgment Number: 1    (relative ack number)
Acknowledgment number (raw): 996409515
```

Sent PDU.

```
Sequence Number: 1    (relative sequence number)
Sequence Number (raw): 996409515
[Next Sequence Number: 1    (relative sequence number)]
Acknowledgment Number: 747    (relative ack number)
Acknowledgment number (raw): 3027493829
```

ACK. It uses the previous packets acknowledgement as sequence number (996409515). Sequence number of the PSH is 3 027 493 083. ACK number of the ack packet is 3 027 493 829. The

difference between the Ack number and the previous sequence number is 746 just like the length of the tcp PSH packet.

Last ack number 3027646148. Last Ack – First sequence number = 153 065 bytes.

c)

First packet 746

Second 5800

All in the middle 7300.

Last 3399.

d)

There is only 2 closing packets captured in this instance 1 initiation from the server and one ACK from my computer. This was sent 5 seconds after the last HTTP. (There was expected to be 4 packets to close but repeated capturing confirmed that we only captured 2).

Task 4

a)

Estimated RTT: 0.875*Estimated RTT + 0.125*RTT

No	Seq. Number	Payload Length	Time sent	Time ACKed	RTT	Estimated RTT
4	1	565	0,026477	0,053937	0,02746	0,02746
5	566	1460	0,041737	0,077294	0,035557	0,028472125
7	2026	1460	0,054026	0,124085	0,070059	0,03367048
8	3486	1460	0,05469	0,169118	0,114428	0,04376517
10	4946	1460	0,077405	0,217299	0,139894	0,05578128
11	6406	1460	0,078157	0,267802	0,189645	0,07251424

b)

No	Seq. Number	Payload Length	Time sent	Time ACKed	RTT	Estimated RTT
6	1	746	31,03588	31,14940	0,11352	0,11352
7	747	13140	31,03600	31,14940	0,11340	0,11351
11	13887	27740	31,14943	31,26299	0,11356	0,11351

Task 5

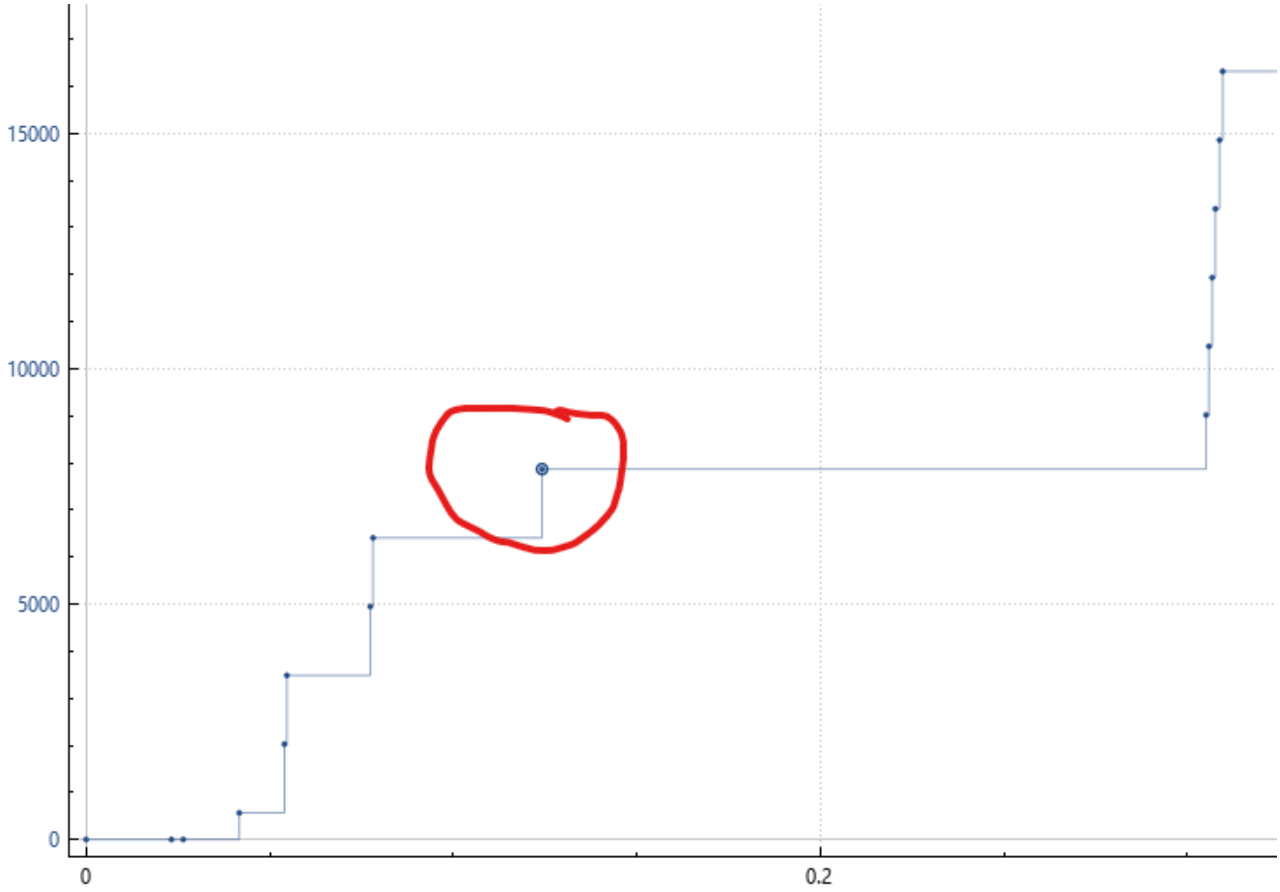
a)

no retransmission.

Retransmission would look like a drop back to a lower sequence number trying to retransmit what was lost.

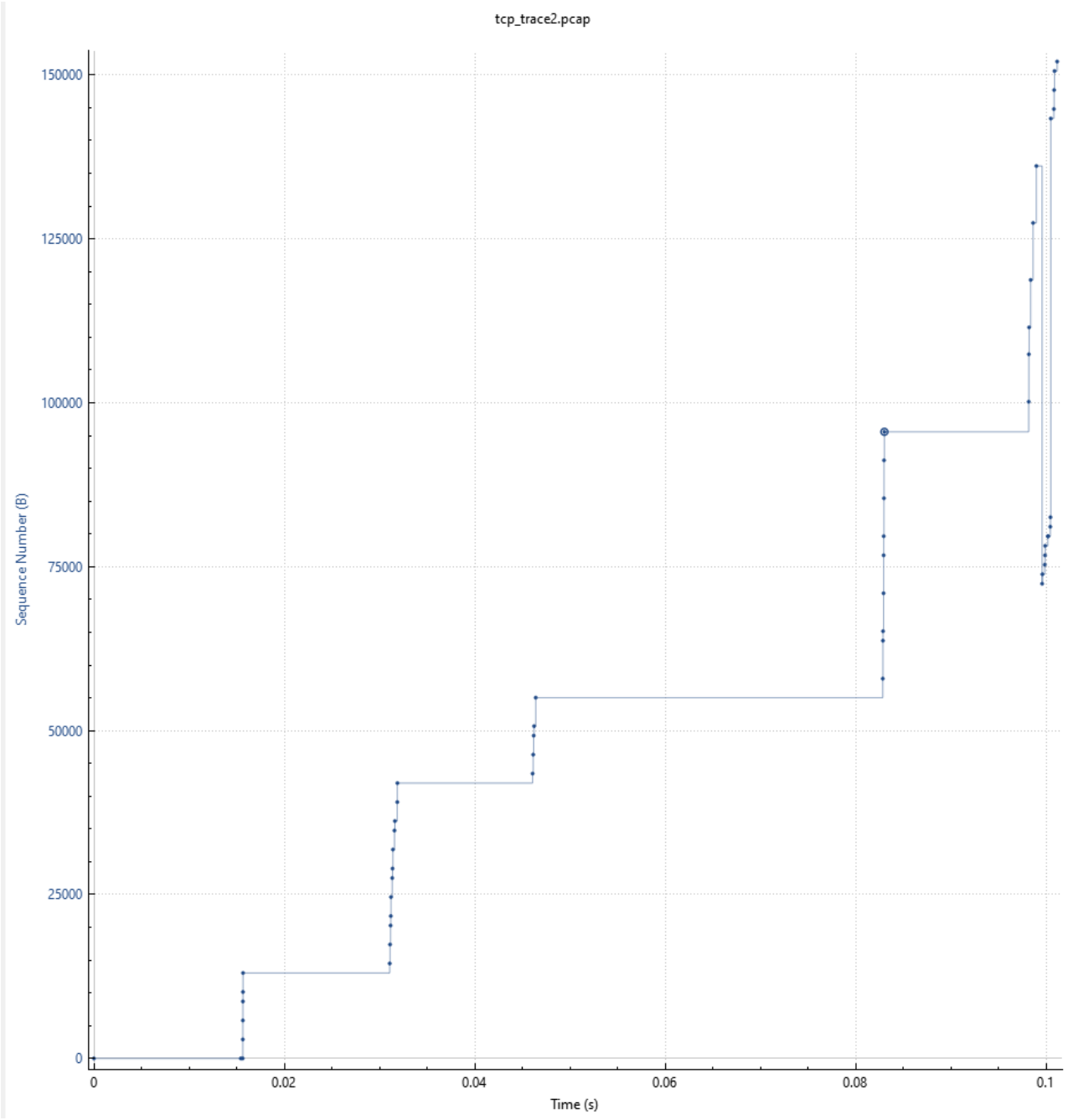
b)

at 0.1242 seconds, Sequence number 7866



The lack of receiver buffer space throttles the sender when it can’t send at full capable speed or risk losing packets.

Task 6



a)

A packet is lost when sequence number being transmitted becomes lower which can be seen at the end. Receiver begins sending the same ack number (dup ACK) which prompts the sender to start resending starting at the sequence number equal to the ACK number. The receiver keeps sending the same ACK number and the sender should keep retransmit all packets starting from the previously mentioned sequence number and going up. When the receiver stops sending dup ACKs it means the receiver has recieved all the packets that was lost and the sender can go back to transmitting the sequence after the one it was interupted at.

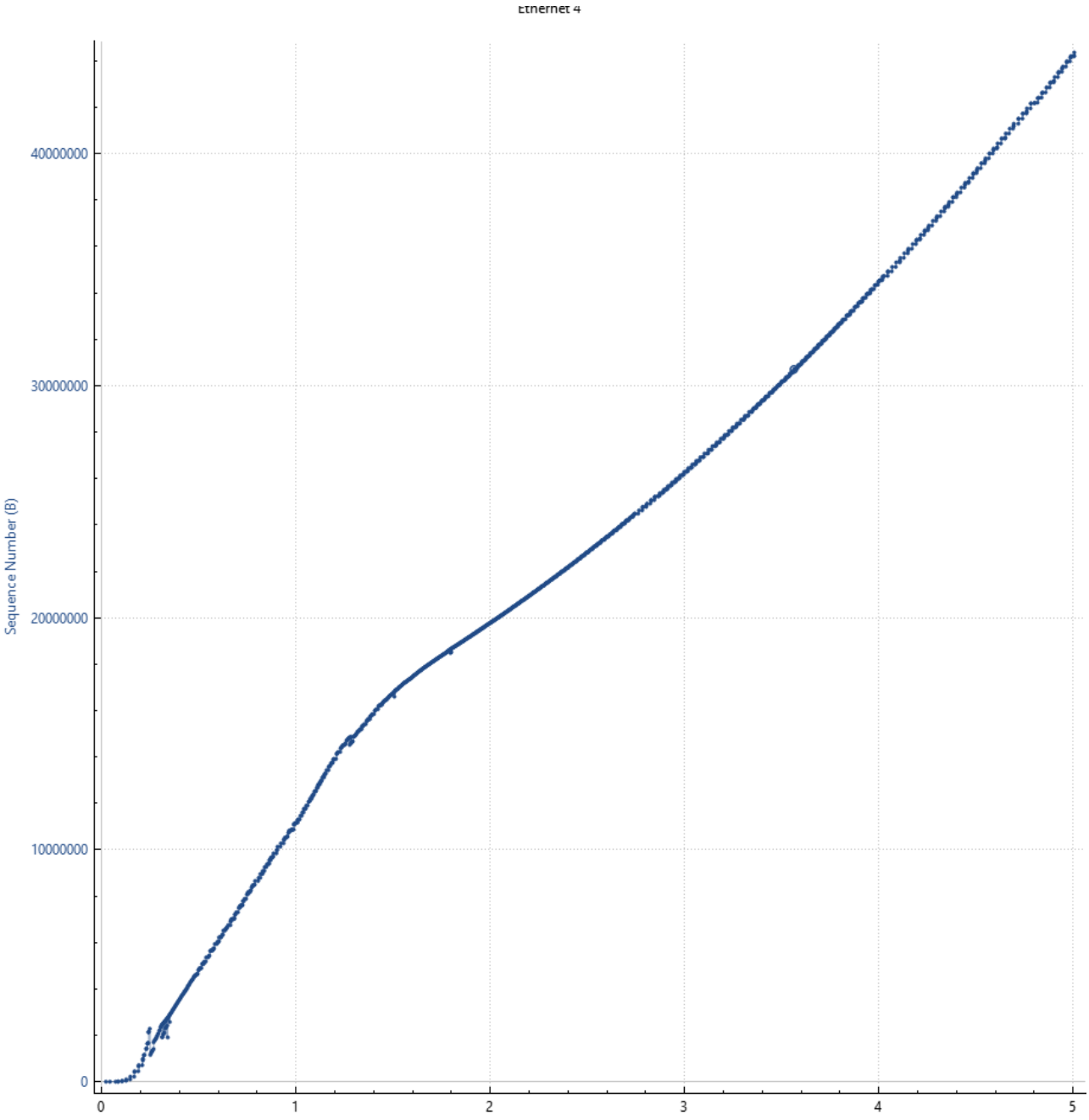
Packet with sequence number 82537 was lost. The sender noticed the loss at t = 0.0996 seconds.

b)

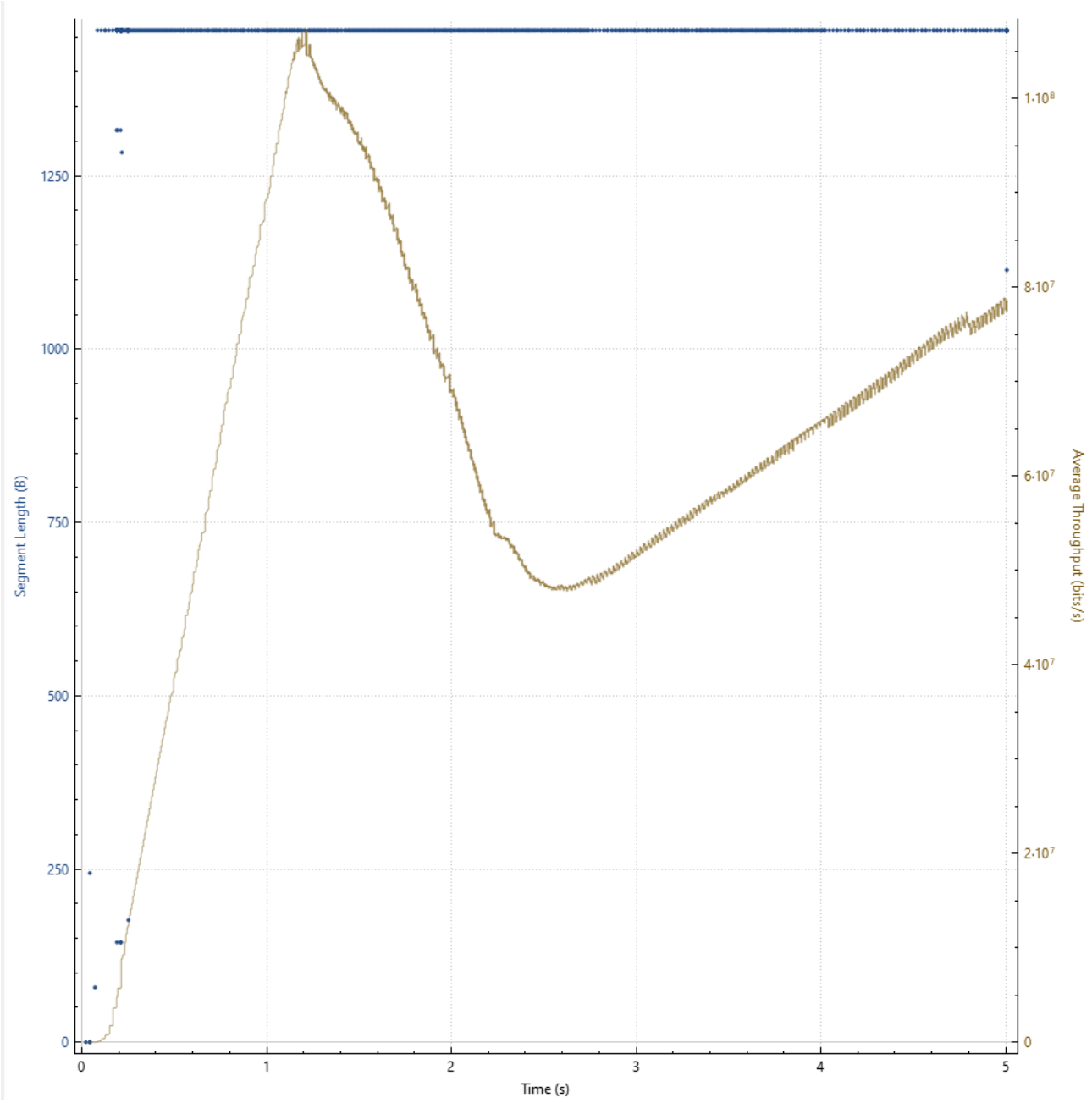
At 0.101194 seconds to 0.116712 seconds 28 packets are cummulatively acked and then it flags packets loss with 8 dup ACKs but then it probably changes it’s mind after recieving the packet it thought it lost and then ACKs 7 more packets and then sending an HTTP ”ok” message.

Task 7

Stevens



Throughput



a)

overall throughput = 78 600 000 bits/s

It’s the value on the leftside of the graph.

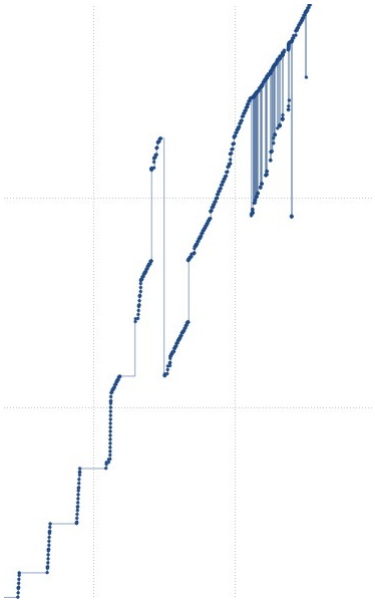
b)

It rose up to over 100 000 000 bits/s at 1.1 second and then it lowered down to under 50 000 000 bits/s at about 2.5 second mark and then rose until the end.

The graph resembles a system with an overswing step response closing in to the set value.

c)

*losses after the slow start



Yes it starts slow close to the start follwed by packet loss or packets out of order * prompting retransmission (the up and downs in the graph close to 0.1 seconds) and then the slope is linear (congestion avoidance) and at about 1.5 seconds the slope decreases for a bit and the transmission slows down then it slowly increses the speed again to fully utilize the link.

d)

It is not as aggressive at increasing the speed as in the books and more cautious with losing data. (more sophisticated).