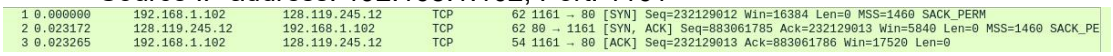
Exercise 1:

1. IP address of gaia.cs.umass.edu: 128.119.245.12 , Port: 80
   Source IP address: 192.168.1.102, Port: 1161



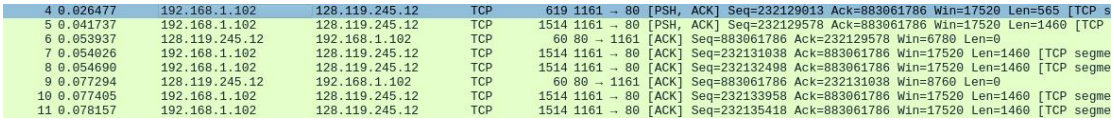

2. Seqno:232129013




3.

EstimatedRTT = (1-a)*EstimatedRTT + a*SampleRTT

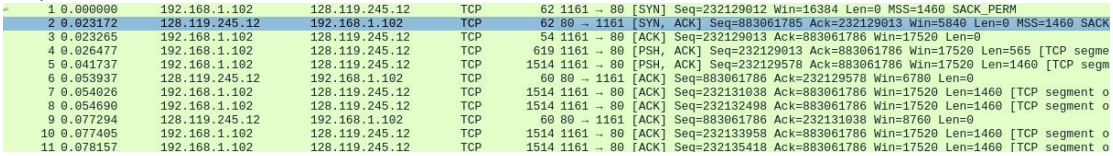
| seqno | Time sent | Ack received | RTT | estimatedRTT | length |
|---|---|---|---|---|---|
| 232129013 | 0.026477 | 0.053937 | 0.02746 | 0.02746 | 565 |
| 232129578 | 0.041737 | 0.077294 | 0.035557 | 0.028472 | 1460 |
| 232131038 | 0.054026 | 0.124085 | 0.070059 | 0.033670 | 1460 |
| 232132498 | 0.054690 | 0.169118 | 0.114428 | 0.043765 | 1460 |
| 232133958 | 0.077405 | 0.217299 | 0.139894 | 0.055781 | 1460 |
| 232135418 | 0.078157 | 0.267802 | 0.189645 | 0.072514 | 1460 |




4.

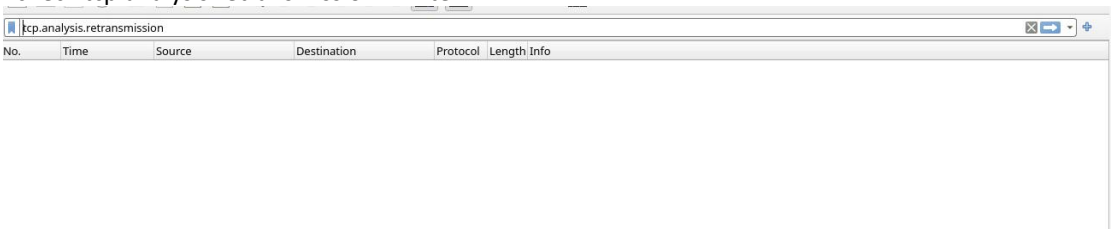
minimum amount of available buffer:5840

No, because it's increased after 5840



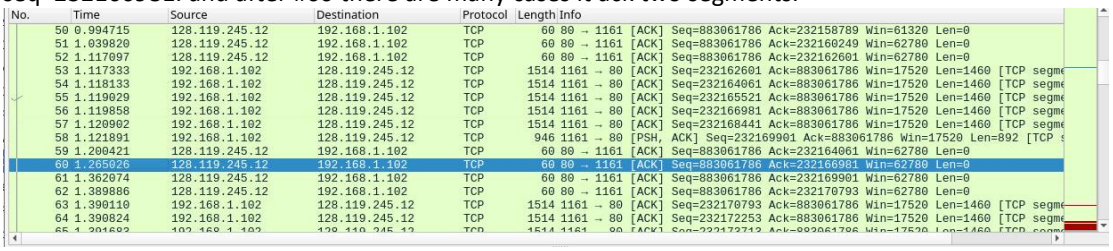

5. No retransmission segment

I check tcp.analysis.retransmission in filter

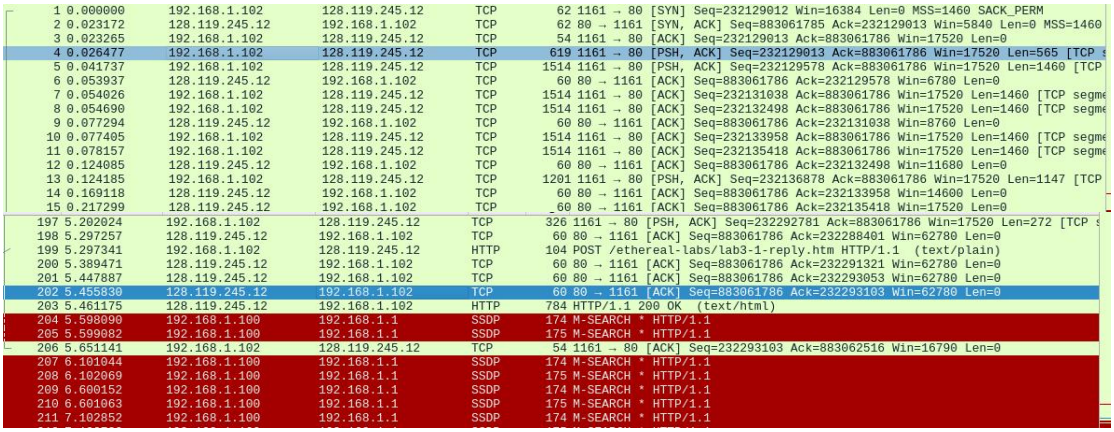



6. 1460

At first receiver will ack each packet and at #60 it ack two segments seq=232165521 and seq=232166981. and after #60 there are many cases it ack two segments.




7. Throughout = Total amount data / Total transmission time



| | | | | | |
|---|---|---|---|---|---|
| 1 | 0.000000 | 192.168.1.102 | 128.119.245.12 | TCP | 62 1161 → 80 [SYN] Seq=232129012 Win=16384 Len=0 MSS=1460 SACK_PERM |
| 2 | 0.023172 | 128.119.245.12 | 192.168.1.102 | TCP | 62 80 → 1161 [SYN, ACK] Seq=883061785 Ack=232129013 Win=5840 Len=0 MSS=1460 |
| 3 | 0.023265 | 192.168.1.102 | 128.119.245.12 | TCP | 54 1161 → 80 [ACK] Seq=232129013 Ack=883061786 Win=17520 Len=0 |
| 4 | 0.026477 | 192.168.1.102 | 128.119.245.12 | TCP | 619 1161 → 80 [PSH, ACK] Seq=232129013 Ack=883061786 Win=17520 Len=565 [TCP |
| 5 | 0.041737 | 192.168.1.102 | 128.119.245.12 | TCP | 1514 1161 → 80 [PSH, ACK] Seq=232129578 Ack=883061786 Win=17520 Len=1460 [TCP |
| 6 | 0.053937 | 128.119.245.12 | 192.168.1.102 | TCP | 60 80 → 1161 [ACK] Seq=883061786 Ack=232129578 Win=6780 Len=0 |
| 7 | 0.054026 | 192.168.1.102 | 128.119.245.12 | TCP | 1514 1161 → 80 [ACK] Seq=232131038 Ack=883061786 Win=17520 Len=1460 [TCP segme |
| 8 | 0.054690 | 192.168.1.102 | 128.119.245.12 | TCP | 1514 1161 → 80 [ACK] Seq=232132498 Ack=883061786 Win=17520 Len=1460 [TCP segme |
| 9 | 0.077294 | 128.119.245.12 | 192.168.1.102 | TCP | 60 80 → 1161 [ACK] Seq=883061786 Ack=232131038 Win=8760 Len=0 |
| 10 | 0.077405 | 192.168.1.102 | 128.119.245.12 | TCP | 1514 1161 → 80 [ACK] Seq=232133958 Ack=883061786 Win=17520 Len=1460 [TCP segme |
| 11 | 0.078157 | 192.168.1.102 | 128.119.245.12 | TCP | 1514 1161 → 80 [ACK] Seq=232135418 Ack=883061786 Win=17520 Len=1460 [TCP segme |
| 12 | 0.124085 | 128.119.245.12 | 192.168.1.102 | TCP | 60 80 → 1161 [ACK] Seq=883061786 Ack=232132498 Win=11680 Len=0 |
| 13 | 0.124185 | 192.168.1.102 | 128.119.245.12 | TCP | 1201 1161 → 80 [PSH, ACK] Seq=232136878 Ack=883061786 Win=17520 Len=1147 [TCP |
| 14 | 0.169118 | 128.119.245.12 | 192.168.1.102 | TCP | 60 80 → 1161 [ACK] Seq=883061786 Ack=232133958 Win=14600 Len=0 |
| 15 | 0.217299 | 128.119.245.12 | 192.168.1.102 | TCP | 60 80 → 1161 [ACK] Seq=883061786 Ack=232135418 Win=17520 Len=0 |
| 197 | 5.202024 | 192.168.1.102 | 128.119.245.12 | TCP | 326 1161 → 80 [PSH, ACK] Seq=232292781 Ack=883061786 Win=17520 Len=272 [TCP |
| 198 | 5.297257 | 128.119.245.12 | 192.168.1.102 | TCP | 60 80 → 1161 [ACK] Seq=883061786 Ack=232288401 Win=62780 Len=0 |
| 199 | 5.297341 | 192.168.1.102 | 128.119.245.12 | HTTP | 104 POST /ethereal-labs/lab3-1-reply.htm HTTP/1.1 (text/plain) |
| 200 | 5.389471 | 128.119.245.12 | 192.168.1.102 | TCP | 60 80 → 1161 [ACK] Seq=883061786 Ack=232291321 Win=62780 Len=0 |
| 201 | 5.447887 | 128.119.245.12 | 192.168.1.102 | TCP | 60 80 → 1161 [ACK] Seq=883061786 Ack=232293053 Win=62780 Len=0 |
| 202 | 5.455830 | 128.119.245.12 | 192.168.1.102 | TCP | 60 80 → 1161 [ACK] Seq=883061786 Ack=232293103 Win=62780 Len=0 |
| 203 | 5.461175 | 128.119.245.12 | 192.168.1.102 | HTTP | 784 HTTP/1.1 200 OK (text/html) |
| 204 | 5.598090 | 192.168.1.100 | 192.168.1.1 | SSDP | 174 M-SEARCH * HTTP/1.1 |
| 205 | 5.599082 | 192.168.1.100 | 192.168.1.1 | SSDP | 175 M-SEARCH * HTTP/1.1 |
| 206 | 5.651141 | 192.168.1.102 | 128.119.245.12 | TCP | 54 1161 → 80 [ACK] Seq=232293103 Ack=883062516 Win=16790 Len=0 |
| 207 | 6.101044 | 192.168.1.100 | 192.168.1.1 | SSDP | 174 M-SEARCH * HTTP/1.1 |
| 208 | 6.102069 | 192.168.1.100 | 192.168.1.1 | SSDP | 175 M-SEARCH * HTTP/1.1 |
| 209 | 6.600152 | 192.168.1.100 | 192.168.1.1 | SSDP | 174 M-SEARCH * HTTP/1.1 |
| 210 | 6.601063 | 192.168.1.100 | 192.168.1.1 | SSDP | 175 M-SEARCH * HTTP/1.1 |
| 211 | 7.102852 | 192.168.1.100 | 192.168.1.1 | SSDP | 174 M-SEARCH * HTTP/1.1 |

Total amount data = 232293103 - 232129013 = 164090 bytes
Total transmission time = 5.455830 - 0.026477 = 5.429353 seconds
Throughout = 164090 / 5.429353 = 30.223 KByte/s

Exercise 2:

| No | Source IP | Destination IP | Protocol | Info |
|---|---|---|---|---|
| 295 | 10.9.16.201 | 10.99.6.175 | TCP | 50045 > 5000 [SYN] Seq=2818463618 win=8192 MSS=1460 |
| 296 | 10.99.6.175 | 10.9.16.201 | TCP | 5000 > 50045 [SYN, ACK] Seq=1247095790 Ack=2818463619 win=262144 MSS=1460 |
| 297 | 10.9.16.201 | 10.99.6.175 | TCP | 50045 > 5000 [ACK] Seq=2818463619 Ack=1247095791 win=65535 |
| 298 | 10.9.16.201 | 10.99.6.175 | TCP | 50045 > 5000 [PSH, ACK] Seq=2818463619 Ack=1247095791 win=65535 |
| 301 | 10.99.6.175 | 10.9.16.201 | TCP | 5000 > 50045 [ACK] Seq=1247095791 Ack=2818463652 win=262096 |
| 302 | 10.99.6.175 | 10.9.16.201 | TCP | 5000 > 50045 [PSH, ACK] Seq=1247095791 Ack=2818463652 win=262144 |
| 303 | 10.9.16.201 | 10.99.6.175 | TCP | 50045 > 5000 [ACK] Seq=2818463652 Ack=1247095831 win=65535 |
| 304 | 10.9.16.201 | 10.99.6.175 | TCP | 50045 > 5000 [FIN, ACK] Seq=2818463652 Ack=1247095831 win=65535 |
| 305 | 10.99.6.175 | 10.9.16.201 | TCP | 5000 > 50045 [FIN, ACK] Seq=1247095831 Ack=2818463652 win=262144 |
| 306 | 10.9.16.201 | 10.99.6.175 | TCP | 50045 > 5000 [ACK] Seq=2818463652 Ack=1247095832 win=65535 |
| 308 | 10.99.6.175 | 10.9.16.201 | TCP | 5000 > 50045 [ACK] Seq=1247095831 Ack=2818463653 win=262144 |

1.2818463618
2.1247095790 2818463619 ensure the seqno from client is seqno + 1
3.2818463619 1247095791 No data contained
4. Both client and server do the acitve close. Because they both send FIN to each other and receive the ack which ACK is previous seqno +1. then close the connection. simultaneous close.
5. client->server = 2818463652 - 2818463619 = 33 bytes
server->client = 1247095831 - 1247095791 = 40 bytes

Final ack - initial seqno - 2 = total transferred data bytes
Like 2818463653 - 2818463618 - 2 = 33 bytes = client->server
This is because the SYN, FIN segments will +1 for ack. And for data segments the ack num is depends on the previous data amount, the ack num is always equal to the previous number + previous data amount.