Universidad San Francisco de Quito

CMP-4005 - REDES +LAB NRC: 4797

Deber 3

Gabriel Lara (215784)

Profesor: Alejandro Proaño Lozada

Fecha de Entrega: 16 de mayo de 2023

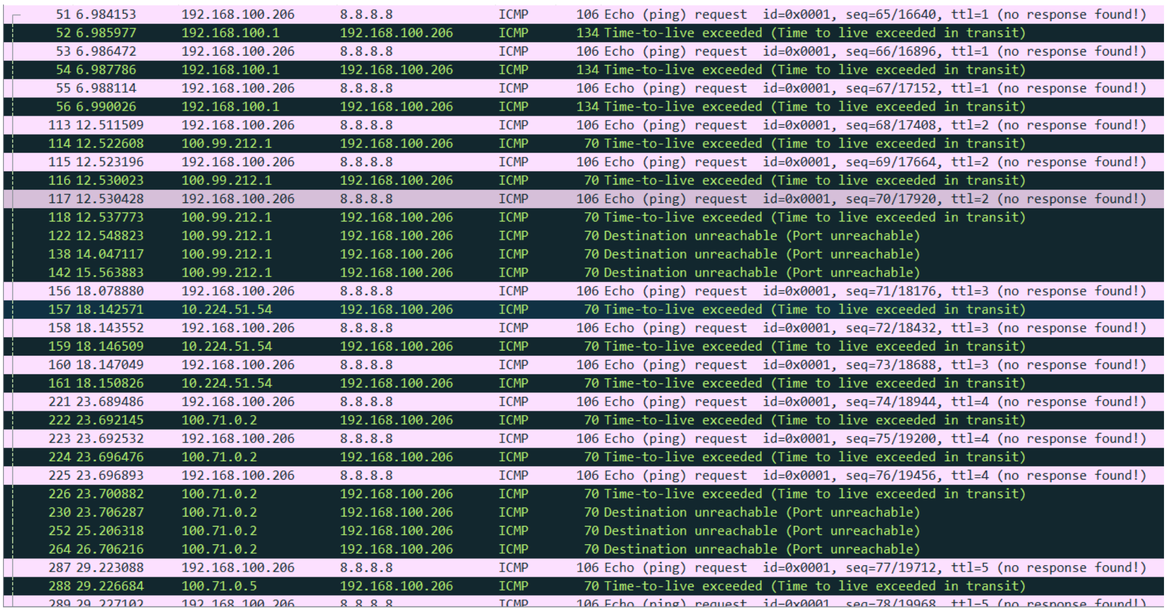
Pregunta 1.

1) Read the following Wireshark tutorial and use it to capture traffic from the following scenarios. Use screenshots to show your results.

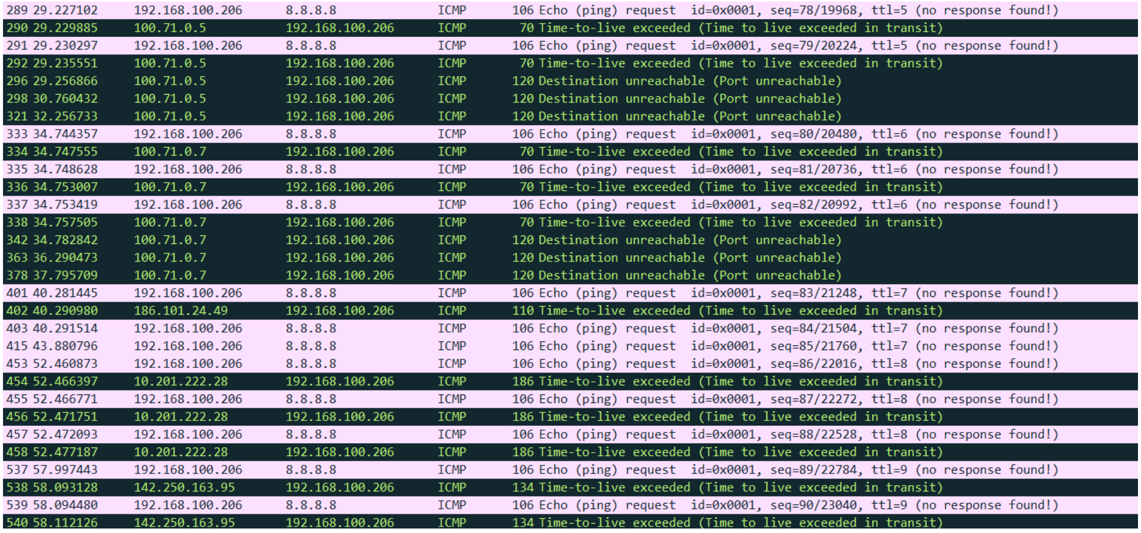
a) Run 10 traceroute commands against google.com

En este caso, utilizaremos un filtro en WireShark para la captura de los paquetes ICMP cuando se captura la red. Filtro=”ICMP”

Estos son los datos que se obtuvieron:

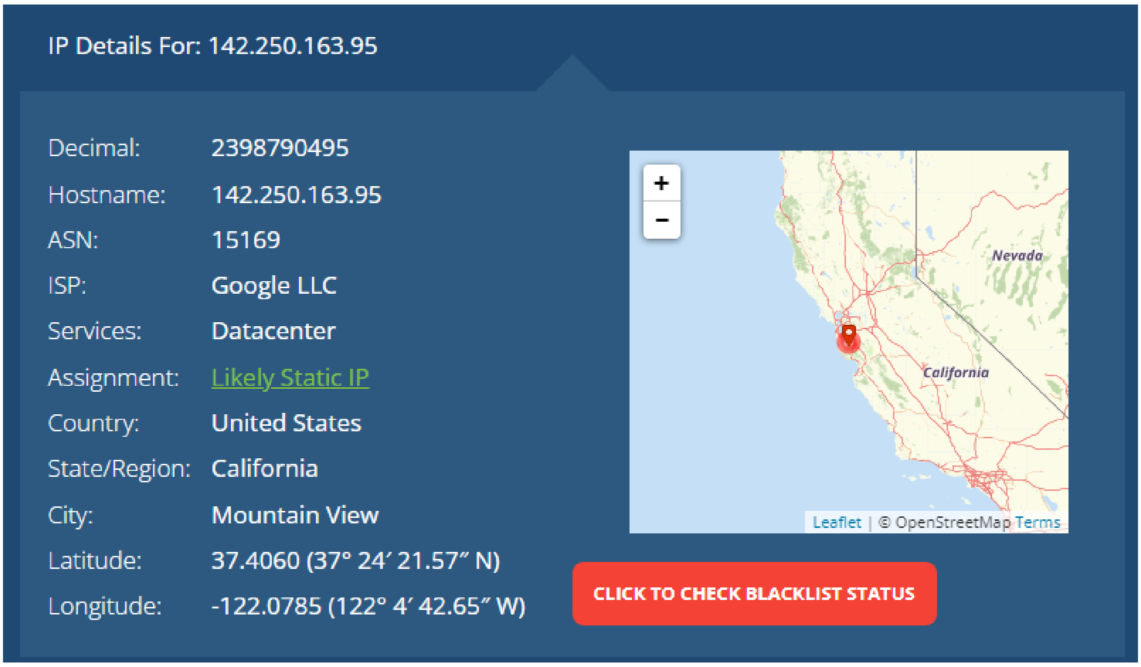


```
 51 6.984153    192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=65/16640, ttl=1 (no response found!)
 52 6.985977    192.168.100.1     192.168.100.206   ICMP   134 Time-to-live exceeded (Time to live exceeded in transit)
 53 6.986472    192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=66/16896, ttl=1 (no response found!)
 54 6.987786    192.168.100.1     192.168.100.206   ICMP   134 Time-to-live exceeded (Time to live exceeded in transit)
 55 6.988114    192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=67/17152, ttl=1 (no response found!)
 56 6.990026    192.168.100.1     192.168.100.206   ICMP   134 Time-to-live exceeded (Time to live exceeded in transit)
113 12.511509   192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=68/17408, ttl=2 (no response found!)
114 12.522608   100.99.212.1      192.168.100.206   ICMP    70 Time-to-live exceeded (Time to live exceeded in transit)
115 12.523196   192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=69/17664, ttl=2 (no response found!)
116 12.530023   100.99.212.1      192.168.100.206   ICMP    70 Time-to-live exceeded (Time to live exceeded in transit)
117 12.530428   192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=70/17920, ttl=2 (no response found!)
118 12.537773   100.99.212.1      192.168.100.206   ICMP    70 Time-to-live exceeded (Time to live exceeded in transit)
122 12.548823   100.99.212.1      192.168.100.206   ICMP    70 Destination unreachable (Port unreachable)
138 14.047117   100.99.212.1      192.168.100.206   ICMP    70 Destination unreachable (Port unreachable)
142 15.563883   100.99.212.1      192.168.100.206   ICMP    70 Destination unreachable (Port unreachable)
156 18.078880   192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=71/18176, ttl=3 (no response found!)
157 18.142571   10.224.51.54      192.168.100.206   ICMP    70 Time-to-live exceeded (Time to live exceeded in transit)
158 18.143552   192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=72/18432, ttl=3 (no response found!)
159 18.146509   10.224.51.54      192.168.100.206   ICMP    70 Time-to-live exceeded (Time to live exceeded in transit)
160 18.147049   192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=73/18688, ttl=3 (no response found!)
161 18.150826   10.224.51.54      192.168.100.206   ICMP    70 Time-to-live exceeded (Time to live exceeded in transit)
221 23.689486   192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=74/18944, ttl=4 (no response found!)
222 23.692145   100.71.0.2        192.168.100.206   ICMP    70 Time-to-live exceeded (Time to live exceeded in transit)
223 23.692532   192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=75/19200, ttl=4 (no response found!)
224 23.696476   100.71.0.2        192.168.100.206   ICMP    70 Time-to-live exceeded (Time to live exceeded in transit)
225 23.696893   192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=76/19456, ttl=4 (no response found!)
226 23.700882   100.71.0.2        192.168.100.206   ICMP    70 Time-to-live exceeded (Time to live exceeded in transit)
230 23.706287   100.71.0.2        192.168.100.206   ICMP    70 Destination unreachable (Port unreachable)
252 25.206318   100.71.0.2        192.168.100.206   ICMP    70 Destination unreachable (Port unreachable)
264 26.706216   100.71.0.2        192.168.100.206   ICMP    70 Destination unreachable (Port unreachable)
287 29.223088   192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=77/19712, ttl=5 (no response found!)
288 29.226684   100.71.0.5        192.168.100.206   ICMP    70 Time-to-live exceeded (Time to live exceeded in transit)
289 29.227102   192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=78/19968, ttl=5 (no response found!)
```



```
289 29.227102   192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=78/19968, ttl=5 (no response found!)
290 29.229885   100.71.0.5        192.168.100.206   ICMP    70 Time-to-live exceeded (Time to live exceeded in transit)
291 29.230297   192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=79/20224, ttl=5 (no response found!)
292 29.235551   100.71.0.5        192.168.100.206   ICMP    70 Time-to-live exceeded (Time to live exceeded in transit)
296 29.256866   100.71.0.5        192.168.100.206   ICMP   120 Destination unreachable (Port unreachable)
298 30.760432   100.71.0.5        192.168.100.206   ICMP   120 Destination unreachable (Port unreachable)
321 32.256733   100.71.0.5        192.168.100.206   ICMP   120 Destination unreachable (Port unreachable)
333 34.744357   192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=80/20480, ttl=6 (no response found!)
334 34.747555   100.71.0.7        192.168.100.206   ICMP    70 Time-to-live exceeded (Time to live exceeded in transit)
335 34.748628   192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=81/20736, ttl=6 (no response found!)
336 34.753007   100.71.0.7        192.168.100.206   ICMP    70 Time-to-live exceeded (Time to live exceeded in transit)
337 34.753419   192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=82/20992, ttl=6 (no response found!)
338 34.757505   100.71.0.7        192.168.100.206   ICMP    70 Time-to-live exceeded (Time to live exceeded in transit)
342 34.782842   100.71.0.7        192.168.100.206   ICMP   120 Destination unreachable (Port unreachable)
363 36.290473   100.71.0.7        192.168.100.206   ICMP   120 Destination unreachable (Port unreachable)
378 37.795709   100.71.0.7        192.168.100.206   ICMP   120 Destination unreachable (Port unreachable)
401 40.281445   192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=83/21248, ttl=7 (no response found!)
402 40.290980   186.101.24.49     192.168.100.206   ICMP   110 Time-to-live exceeded (Time to live exceeded in transit)
403 40.291514   192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=84/21504, ttl=7 (no response found!)
415 43.880796   192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=85/21760, ttl=7 (no response found!)
453 52.460873   192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=86/22016, ttl=8 (no response found!)
454 52.466397   10.201.222.28     192.168.100.206   ICMP   186 Time-to-live exceeded (Time to live exceeded in transit)
455 52.466771   192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=87/22272, ttl=8 (no response found!)
456 52.471751   10.201.222.28     192.168.100.206   ICMP   186 Time-to-live exceeded (Time to live exceeded in transit)
457 52.472093   192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=88/22528, ttl=8 (no response found!)
458 52.477187   10.201.222.28     192.168.100.206   ICMP   186 Time-to-live exceeded (Time to live exceeded in transit)
537 57.997443   192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=89/22784, ttl=9 (no response found!)
538 58.093128   142.250.163.95    192.168.100.206   ICMP   134 Time-to-live exceeded (Time to live exceeded in transit)
539 58.094480   192.168.100.206   8.8.8.8           ICMP   106 Echo (ping) request  id=0x0001, seq=90/23040, ttl=9 (no response found!)
540 58.112126   142.250.163.95    192.168.100.206   ICMP   134 Time-to-live exceeded (Time to live exceeded in transit)
```

Este es el resultado de la captura de paquetes que se obtuvo en WireShark, consinderando la IP (142.250.163.95)del ultimo paquete verificamos que efectivamente sea de google:



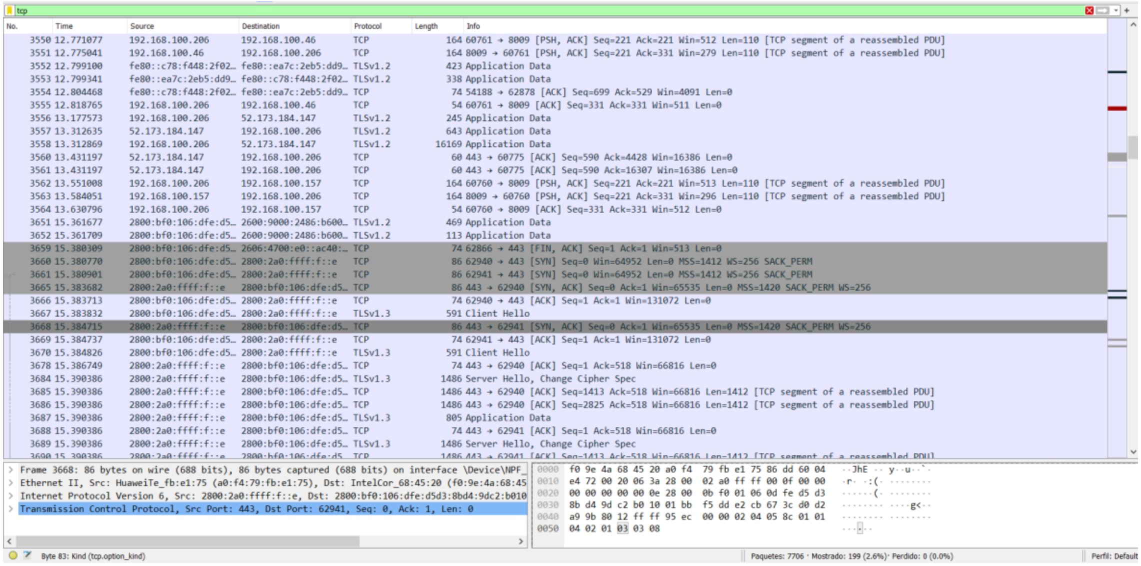
IP Details For: 142.250.163.95

Decimal:	2398790495
Hostname:	142.250.163.95
ASN:	15169
ISP:	Google LLC
Services:	Datacenter
Assignment:	Likely Static IP
Country:	United States
State/Region:	California
City:	Mountain View
Latitude:	37.4060 (37° 24′ 21.57″ N)
Longitude:	-122.0785 (122° 4′ 42.65″ W)

+
−
Nevada
California
Leaflet | © OpenStreetMap Terms
CLICK TO CHECK BLACKLIST STATUS

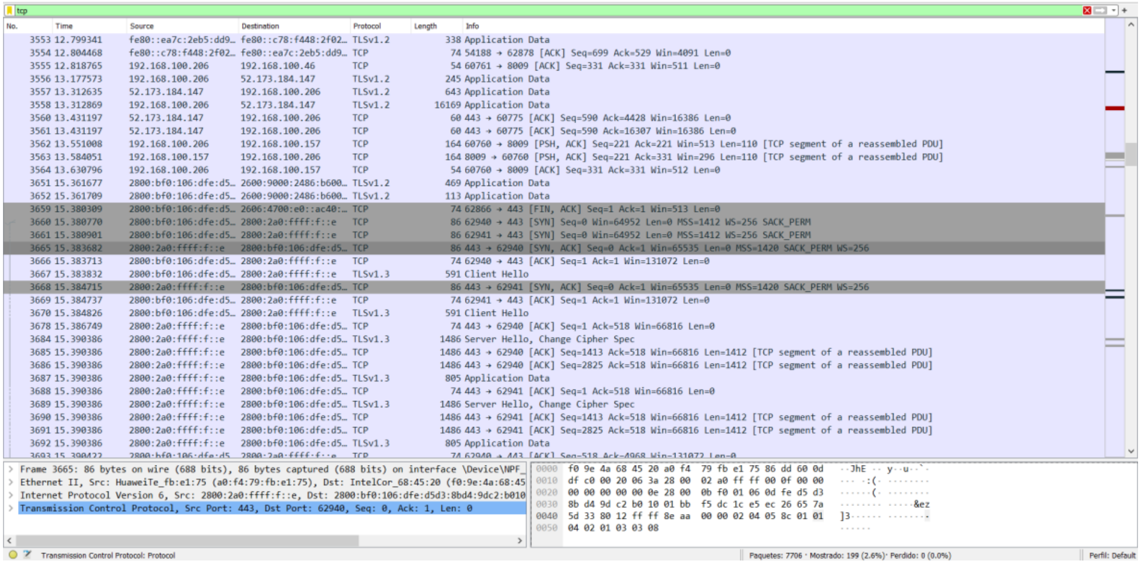
b) Watch a video from youtube.com. Capture the TCP handshake, and the congestion window.

Con los datos obtenidos se busca la captura del paquete [SYN] enviado desde la computadora hasta el servidor de Youtube. Este paquete es con el cual podemos identificar el inicio del handshake de TCP. Con la información que WireShark nos brinda concemos que en primera instancia se establece una ventana de congestión MSS, que es Máximum Sender Size, en teste caso de 1412ms.



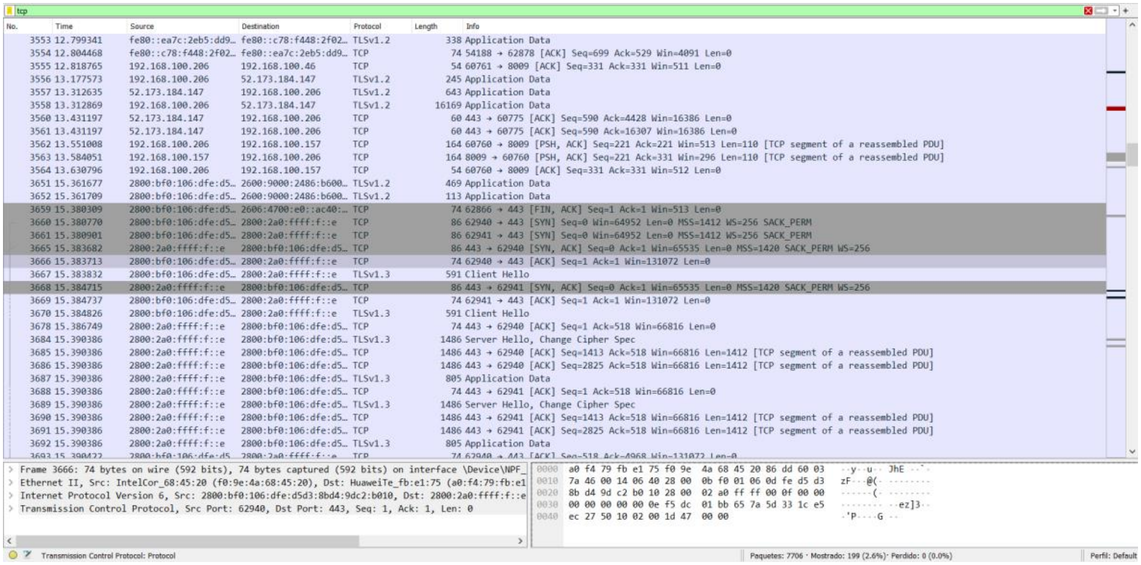
```
tcp
No.   Time      Source                Destination            Protocol  Length Info
 3550 12.771077 192.168.100.206       192.168.100.46         TCP        164 60761 → 8009 [PSH, ACK] Seq=221 Ack=221 Win=512 Len=110 [TCP segment of a reassembled PDU]
 3551 12.775041 192.168.100.46        192.168.100.206        TCP        164 8009 → 60761 [PSH, ACK] Seq=221 Ack=331 Win=279 Len=110 [TCP segment of a reassembled PDU]
 3552 12.799100 fe80::c78:f448:2f02…  fe80::ea7c:2eb5:dd9…   TLSv1.2    423 Application Data
 3553 12.799341 fe80::ea7c:2eb5:dd9…  fe80::c78:f448:2f02…   TLSv1.2    338 Application Data
 3554 12.804468 fe80::c78:f448:2f02…  fe80::ea7c:2eb5:dd9…   TCP         74 54188 → 62878 [ACK] Seq=699 Ack=529 Win=4091 Len=0
 3555 12.818765 192.168.100.206       192.168.100.46         TCP         54 60761 → 8009 [ACK] Seq=331 Ack=331 Win=511 Len=0
 3556 13.177573 192.168.100.206       52.173.184.147         TLSv1.2    245 Application Data
 3557 13.312635 52.173.184.147        192.168.100.206        TLSv1.2    643 Application Data
 3558 13.312869 192.168.100.206       52.173.184.147         TLSv1.2  16169 Application Data
 3560 13.431197 52.173.184.147        192.168.100.206        TCP         60 443 → 60775 [ACK] Seq=590 Ack=4428 Win=16386 Len=0
 3561 13.431197 52.173.184.147        192.168.100.206        TCP         60 443 → 60775 [ACK] Seq=590 Ack=16307 Win=16386 Len=0
 3562 13.551008 192.168.100.206       192.168.100.157        TCP        164 60760 → 8009 [PSH, ACK] Seq=221 Ack=221 Win=513 Len=110 [TCP segment of a reassembled PDU]
 3563 13.584051 192.168.100.157       192.168.100.206        TCP        164 8009 → 60760 [PSH, ACK] Seq=221 Ack=331 Win=296 Len=110 [TCP segment of a reassembled PDU]
 3564 13.630796 192.168.100.206       192.168.100.157        TCP         54 60760 → 8009 [ACK] Seq=331 Ack=331 Win=512 Len=0
 3651 15.361677 2800:bf0:106:dfe:d5…  2600:9000:2486:b600…   TLSv1.2    469 Application Data
 3652 15.361709 2800:bf0:106:dfe:d5…  2600:9000:2486:b600…   TLSv1.2    113 Application Data
 3659 15.380309 2800:bf0:106:dfe:d5…  2606:4700:e0::ac40:…   TCP         74 62866 → 443 [FIN, ACK] Seq=1 Ack=1 Win=513 Len=0
 3660 15.380770 2800:bf0:106:dfe:d5…  2800:2a0:ffff:f::e     TCP         86 62940 → 443 [SYN] Seq=0 Win=64952 Len=0 MSS=1412 WS=256 SACK_PERM
 3661 15.380901 2800:bf0:106:dfe:d5…  2800:2a0:ffff:f::e     TCP         86 62941 → 443 [SYN] Seq=0 Win=64952 Len=0 MSS=1412 WS=256 SACK_PERM
 3665 15.383682 2800:2a0:ffff:f::e    2800:bf0:106:dfe:d5…   TCP         86 443 → 62940 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1420 SACK_PERM WS=256
 3666 15.383713 2800:bf0:106:dfe:d5…  2800:2a0:ffff:f::e     TCP         74 62940 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
 3667 15.383832 2800:bf0:106:dfe:d5…  2800:2a0:ffff:f::e     TLSv1.3    591 Client Hello
 3668 15.384715 2800:2a0:ffff:f::e    2800:bf0:106:dfe:d5…   TCP         86 443 → 62941 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1420 SACK_PERM WS=256
 3669 15.384737 2800:bf0:106:dfe:d5…  2800:2a0:ffff:f::e     TCP         74 62941 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
 3670 15.384826 2800:bf0:106:dfe:d5…  2800:2a0:ffff:f::e     TLSv1.3    591 Client Hello
 3678 15.386749 2800:2a0:ffff:f::e    2800:bf0:106:dfe:d5…   TCP         74 443 → 62940 [ACK] Seq=1 Ack=518 Win=66816 Len=0
 3684 15.390386 2800:2a0:ffff:f::e    2800:bf0:106:dfe:d5…   TLSv1.3   1486 Server Hello, Change Cipher Spec
 3685 15.390386 2800:2a0:ffff:f::e    2800:bf0:106:dfe:d5…   TCP       1486 443 → 62940 [ACK] Seq=1413 Ack=518 Win=66816 Len=1412 [TCP segment of a reassembled PDU]
 3686 15.390386 2800:2a0:ffff:f::e    2800:bf0:106:dfe:d5…   TCP       1486 443 → 62940 [ACK] Seq=2825 Ack=518 Win=66816 Len=1412 [TCP segment of a reassembled PDU]
 3687 15.390386 2800:2a0:ffff:f::e    2800:bf0:106:dfe:d5…   TLSv1.3    805 Application Data
 3688 15.390386 2800:2a0:ffff:f::e    2800:bf0:106:dfe:d5…   TCP         74 443 → 62941 [ACK] Seq=1 Ack=518 Win=66816 Len=0
 3689 15.390386 2800:2a0:ffff:f::e    2800:bf0:106:dfe:d5…   TLSv1.3   1486 Server Hello, Change Cipher Spec
 3690 15.390386 2800:2a0:ffff:f::e    2800:bf0:106:dfe:d5…   TCP       1486 443 → 62941 [ACK] Seq=1413 Ack=518 Win=66816 Len=1412 [TCP segment of a reassembled PDU]

> Frame 3668: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface \Device\NPF_
> Ethernet II, Src: HuaweiTe_fb:e1:75 (a0:f4:79:fb:e1:75), Dst: IntelCor_68:45:20 (f0:9e:4a:68:45
> Internet Protocol Version 6, Src: 2800:2a0:ffff:f::e, Dst: 2800:bf0:106:dfe:d5d3:8bd4:9dc2:b010
> Transmission Control Protocol, Src Port: 443, Dst Port: 62941, Seq: 0, Ack: 1, Len: 0

0000  f0 9e 4a 68 45 20 a0 f4  79 fb e1 75 86 dd 60 04   ··JhE ·· y··u··`·
0010  e4 72 00 20 06 3a 28 00  02 a0 ff ff 00 0f 00 00   ·r· ·:(· ········
0020  00 00 00 00 00 0e 28 00  0b f0 01 06 0d fe d5 d3   ······(· ········
0030  8b d4 9d c2 b0 10 01 bb  f5 dd e2 cb 67 3c d0 d2   ········ ····g<··
0040  a9 9b 80 12 ff ff 95 ec  00 00 02 04 05 8c 01 01   ········ ········
0050  04 02 01 03 03 08                                  ······

Byte 83: Kind (tcp.option_kind)      Paquetes: 7706 · Mostrado: 199 (2.6%)· Perdido: 0 (0.0%)      Perfil: Default
```

Con el numero de secuencia obtenido buscamos el paquete [SYN,ACK] el cual es la respuesta por parte del servidor de Youtube



```
tcp
No.   Time       Source                  Destination             Protocol  Length Info
 3553 12.799341  fe80::ea7c:2eb5:dd9…   fe80::c78:f448:2f02…   TLSv1.2     338 Application Data
 3554 12.804468  fe80::c78:f448:2f02…   fe80::ea7c:2eb5:dd9…   TCP          74 54188 → 62878 [ACK] Seq=699 Ack=529 Win=4091 Len=0
 3555 12.818765  192.168.100.206        192.168.100.46         TCP          54 60761 → 8009 [ACK] Seq=331 Ack=331 Win=511 Len=0
 3556 13.177573  192.168.100.206        52.173.184.147         TLSv1.2     245 Application Data
 3557 13.312635  52.173.184.147         192.168.100.206        TLSv1.2     643 Application Data
 3558 13.312869  192.168.100.206        52.173.184.147         TLSv1.2   16169 Application Data
 3560 13.431197  52.173.184.147         192.168.100.206        TCP          60 443 → 60775 [ACK] Seq=590 Ack=4428 Win=16386 Len=0
 3561 13.431197  52.173.184.147         192.168.100.206        TCP          60 443 → 60775 [ACK] Seq=590 Ack=16307 Win=16386 Len=0
 3562 13.551008  192.168.100.206        192.168.100.157        TCP         164 60760 → 8009 [PSH, ACK] Seq=221 Ack=221 Win=513 Len=110 [TCP segment of a reassembled PDU]
 3563 13.584051  192.168.100.157        192.168.100.206        TCP         164 8009 → 60760 [PSH, ACK] Seq=221 Ack=331 Win=296 Len=110 [TCP segment of a reassembled PDU]
 3564 13.630796  192.168.100.206        192.168.100.157        TCP          54 60760 → 8009 [ACK] Seq=331 Ack=331 Win=512 Len=0
 3651 15.361677  2800:bf0:106:dfe:d5…   2600:9000:2486:b600…   TLSv1.2     469 Application Data
 3652 15.361709  2800:bf0:106:dfe:d5…   2600:9000:2486:b600…   TLSv1.2     113 Application Data
 3659 15.380309  2800:bf0:106:dfe:d5…   2606:4700:e0::ac40:…   TCP          74 62866 → 443 [FIN, ACK] Seq=1 Ack=1 Win=513 Len=0
 3660 15.380770  2800:bf0:106:dfe:d5…   2800:2a0:ffff:f::e     TCP          86 62940 → 443 [SYN] Seq=0 Win=64952 Len=0 MSS=1412 WS=256 SACK_PERM
 3661 15.380901  2800:bf0:106:dfe:d5…   2800:2a0:ffff:f::e     TCP          86 62941 → 443 [SYN] Seq=0 Win=64952 Len=0 MSS=1412 WS=256 SACK_PERM
 3665 15.383682  2800:2a0:ffff:f::e     2800:bf0:106:dfe:d5…   TCP          86 443 → 62940 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1420 SACK_PERM WS=256
 3666 15.383713  2800:bf0:106:dfe:d5…   2800:2a0:ffff:f::e     TCP          74 62940 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
 3667 15.383832  2800:bf0:106:dfe:d5…   2800:2a0:ffff:f::e     TLSv1.3     591 Client Hello
 3668 15.384715  2800:2a0:ffff:f::e     2800:bf0:106:dfe:d5…   TCP          86 443 → 62941 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1420 SACK_PERM WS=256
 3669 15.384737  2800:bf0:106:dfe:d5…   2800:2a0:ffff:f::e     TCP          74 62941 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
 3670 15.384826  2800:bf0:106:dfe:d5…   2800:2a0:ffff:f::e     TLSv1.3     591 Client Hello
 3678 15.386749  2800:2a0:ffff:f::e     2800:bf0:106:dfe:d5…   TCP          74 443 → 62940 [ACK] Seq=1 Ack=518 Win=66816 Len=0
 3684 15.390386  2800:2a0:ffff:f::e     2800:bf0:106:dfe:d5…   TLSv1.3    1486 Server Hello, Change Cipher Spec
 3685 15.390386  2800:2a0:ffff:f::e     2800:bf0:106:dfe:d5…   TCP        1486 443 → 62940 [ACK] Seq=1413 Ack=518 Win=66816 Len=1412 [TCP segment of a reassembled PDU]
 3686 15.390386  2800:2a0:ffff:f::e     2800:bf0:106:dfe:d5…   TCP        1486 443 → 62940 [ACK] Seq=2825 Ack=518 Win=66816 Len=1412 [TCP segment of a reassembled PDU]
 3687 15.390386  2800:2a0:ffff:f::e     2800:bf0:106:dfe:d5…   TLSv1.3     805 Application Data
 3688 15.390386  2800:2a0:ffff:f::e     2800:bf0:106:dfe:d5…   TCP          74 443 → 62941 [ACK] Seq=1 Ack=518 Win=66816 Len=0
 3689 15.390386  2800:2a0:ffff:f::e     2800:bf0:106:dfe:d5…   TLSv1.3    1486 Server Hello, Change Cipher Spec
 3690 15.390386  2800:2a0:ffff:f::e     2800:bf0:106:dfe:d5…   TCP        1486 443 → 62941 [ACK] Seq=1413 Ack=518 Win=66816 Len=1412 [TCP segment of a reassembled PDU]
 3691 15.390386  2800:2a0:ffff:f::e     2800:bf0:106:dfe:d5…   TCP        1486 443 → 62941 [ACK] Seq=2825 Ack=518 Win=66816 Len=1412 [TCP segment of a reassembled PDU]
 3692 15.390386  2800:2a0:ffff:f::e     2800:bf0:106:dfe:d5…   TLSv1.3     805 Application Data
 3693 15.390422  2800:bf0:106:dfe:d5    2800:2a0:ffff:f::e     TCP          74 62940 → 443 [ACK] Seq=518 Ack=4968 Win=131072 Len=0

> Frame 3665: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface \Device\NPF_
> Ethernet II, Src: HuaweiTe_fb:e1:75 (a0:f4:79:fb:e1:75), Dst: IntelCor_68:45:20 (f0:9e:4a:68:45
> Internet Protocol Version 6, Src: 2800:2a0:ffff:f::e, Dst: 2800:bf0:106:dfe:d5d3:8bd4:9dc2:b010
> Transmission Control Protocol, Src Port: 443, Dst Port: 62940, Seq: 0, Ack: 1, Len: 0

0000  f0 9e 4a 68 45 20 a0 f4  79 fb e1 75 86 dd 60 0d   ··JhE ·· y··u··`·
0010  df c0 00 20 06 3a 28 00  02 a0 ff ff 00 0f 00 00   ··· ·:(· ········
0020  00 00 00 00 00 0e 28 00  0b f0 01 06 0d fe d5 d3   ······(· ········
0030  8b d4 9d c2 b0 10 01 bb  f5 dc 1c e5 ec 26 65 7a   ········ ·····&ez
0040  5d 33 80 12 ff ff 8e aa  00 00 02 04 05 8c 01 01   ]3······ ········
0050  04 02 01 03 03 08                                  ······

Transmission Control Protocol: Protocol        Paquetes: 7706 · Mostrado: 199 (2.6%)· Perdido: 0 (0.0%)        Perfil: Default
```

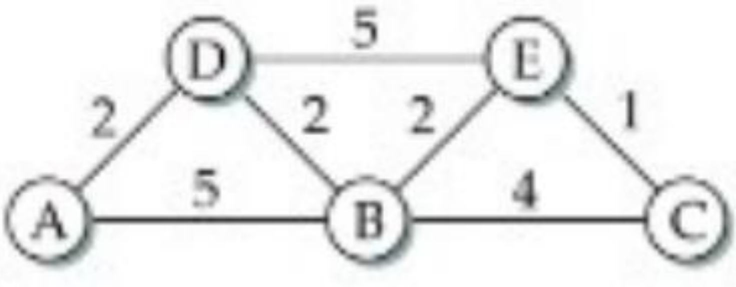
En este paquete seleccionado, terminamos el Sequence Number, y el valor de ACK.



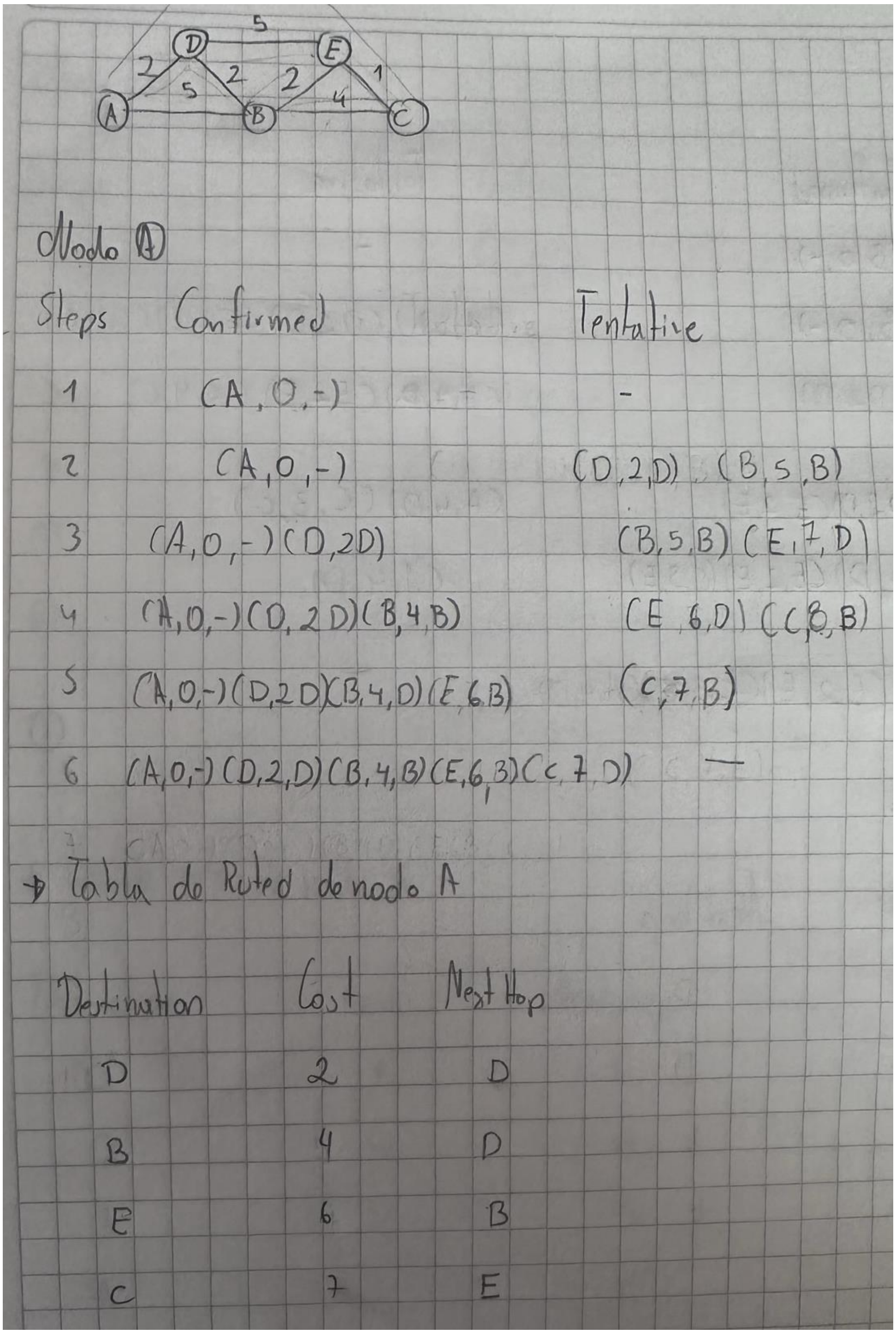
```
tcp
No.   Time       Source                  Destination             Protocol  Length Info
 3553 12.799341  fe80::ea7c:2eb5:dd9…   fe80::c78:f448:2f02…   TLSv1.2     338 Application Data
 3554 12.804468  fe80::c78:f448:2f02…   fe80::ea7c:2eb5:dd9…   TCP          74 54188 → 62878 [ACK] Seq=699 Ack=529 Win=4091 Len=0
 3555 12.818765  192.168.100.206        192.168.100.46         TCP          54 60761 → 8009 [ACK] Seq=331 Ack=331 Win=511 Len=0
 3556 13.177573  192.168.100.206        52.173.184.147         TLSv1.2     245 Application Data
 3557 13.312635  52.173.184.147         192.168.100.206        TLSv1.2     643 Application Data
 3558 13.312869  192.168.100.206        52.173.184.147         TLSv1.2   16169 Application Data
 3560 13.431197  52.173.184.147         192.168.100.206        TCP          60 443 → 60775 [ACK] Seq=590 Ack=4428 Win=16386 Len=0
 3561 13.431197  52.173.184.147         192.168.100.206        TCP          60 443 → 60775 [ACK] Seq=590 Ack=16307 Win=16386 Len=0
 3562 13.551008  192.168.100.206        192.168.100.157        TCP         164 60760 → 8009 [PSH, ACK] Seq=221 Ack=221 Win=513 Len=110 [TCP segment of a reassembled PDU]
 3563 13.584051  192.168.100.157        192.168.100.206        TCP         164 8009 → 60760 [PSH, ACK] Seq=221 Ack=331 Win=296 Len=110 [TCP segment of a reassembled PDU]
 3564 13.630796  192.168.100.206        192.168.100.157        TCP          54 60760 → 8009 [ACK] Seq=331 Ack=331 Win=512 Len=0
 3651 15.361677  2800:bf0:106:dfe:d5…   2600:9000:2486:b600…   TLSv1.2     469 Application Data
 3652 15.361709  2800:bf0:106:dfe:d5…   2600:9000:2486:b600…   TLSv1.2     113 Application Data
 3659 15.380309  2800:bf0:106:dfe:d5…   2606:4700:e0::ac40:…   TCP          74 62866 → 443 [FIN, ACK] Seq=1 Ack=1 Win=513 Len=0
 3660 15.380770  2800:bf0:106:dfe:d5…   2800:2a0:ffff:f::e     TCP          86 62940 → 443 [SYN] Seq=0 Win=64952 Len=0 MSS=1412 WS=256 SACK_PERM
 3661 15.380901  2800:bf0:106:dfe:d5…   2800:2a0:ffff:f::e     TCP          86 62941 → 443 [SYN] Seq=0 Win=64952 Len=0 MSS=1412 WS=256 SACK_PERM
 3665 15.383682  2800:2a0:ffff:f::e     2800:bf0:106:dfe:d5…   TCP          86 443 → 62940 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1420 SACK_PERM WS=256
 3666 15.383713  2800:bf0:106:dfe:d5…   2800:2a0:ffff:f::e     TCP          74 62940 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
 3667 15.383832  2800:bf0:106:dfe:d5…   2800:2a0:ffff:f::e     TLSv1.3     591 Client Hello
 3668 15.384715  2800:2a0:ffff:f::e     2800:bf0:106:dfe:d5…   TCP          86 443 → 62941 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1420 SACK_PERM WS=256
 3669 15.384737  2800:bf0:106:dfe:d5…   2800:2a0:ffff:f::e     TCP          74 62941 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
 3670 15.384826  2800:bf0:106:dfe:d5…   2800:2a0:ffff:f::e     TLSv1.3     591 Client Hello
 3678 15.386749  2800:2a0:ffff:f::e     2800:bf0:106:dfe:d5…   TCP          74 443 → 62940 [ACK] Seq=1 Ack=518 Win=66816 Len=0
 3684 15.390386  2800:2a0:ffff:f::e     2800:bf0:106:dfe:d5…   TLSv1.3    1486 Server Hello, Change Cipher Spec
 3685 15.390386  2800:2a0:ffff:f::e     2800:bf0:106:dfe:d5…   TCP        1486 443 → 62940 [ACK] Seq=1413 Ack=518 Win=66816 Len=1412 [TCP segment of a reassembled PDU]
 3686 15.390386  2800:2a0:ffff:f::e     2800:bf0:106:dfe:d5…   TCP        1486 443 → 62940 [ACK] Seq=2825 Ack=518 Win=66816 Len=1412 [TCP segment of a reassembled PDU]
 3687 15.390386  2800:2a0:ffff:f::e     2800:bf0:106:dfe:d5…   TLSv1.3     805 Application Data
 3688 15.390386  2800:2a0:ffff:f::e     2800:bf0:106:dfe:d5…   TCP          74 443 → 62941 [ACK] Seq=1 Ack=518 Win=66816 Len=0
 3689 15.390386  2800:2a0:ffff:f::e     2800:bf0:106:dfe:d5…   TLSv1.3    1486 Server Hello, Change Cipher Spec
 3690 15.390386  2800:2a0:ffff:f::e     2800:bf0:106:dfe:d5…   TCP        1486 443 → 62941 [ACK] Seq=1413 Ack=518 Win=66816 Len=1412 [TCP segment of a reassembled PDU]
 3691 15.390386  2800:2a0:ffff:f::e     2800:bf0:106:dfe:d5…   TCP        1486 443 → 62941 [ACK] Seq=2825 Ack=518 Win=66816 Len=1412 [TCP segment of a reassembled PDU]
 3692 15.390386  2800:2a0:ffff:f::e     2800:bf0:106:dfe:d5…   TLSv1.3     805 Application Data
 3693 15.390422  2800:bf0:106:dfe:d5    2800:2a0:ffff:f::e     TCP          74 62940 → 443 [ACK] Seq=518 Ack=4968 Win=131072 Len=0

> Frame 3666: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_
> Ethernet II, Src: IntelCor_68:45:20 (f0:9e:4a:68:45:20), Dst: HuaweiTe_fb:e1:75 (a0:f4:79:fb:e1
> Internet Protocol Version 6, Src: 2800:bf0:106:dfe:d5d3:8bd4:9dc2:b010, Dst: 2800:2a0:ffff:f::e
> Transmission Control Protocol, Src Port: 62940, Dst Port: 443, Seq: 1, Ack: 1, Len: 0

0000  a0 f4 79 fb e1 75 f0 9e  4a 68 45 20 86 dd 60 03   ··y··u·· JhE ··`·
0010  7a 46 00 14 06 40 28 00  0b f0 01 06 0d fe d5 d3   zF···@(· ········
0020  8b d4 9d c2 b0 10 28 00  02 a0 ff ff 00 0f 00 00   ······(· ········
0030  00 00 00 00 00 0e f5 dc  01 bb 65 7a 5d 33 1c e5   ········ ··ez]3··
0040  ec 27 50 10 02 00 1d 47  00 00                     ·'P····G ··

Transmission Control Protocol: Protocol        Paquetes: 7706 · Mostrado: 199 (2.6%)· Perdido: 0 (0.0%)        Perfil: Default
```

El siguiente paso es identificar el paquete ACK el cual contiene la respuesta al paquete SYN, ACK por parte de la computadora que lo envía al servidor de YouTube. Este proceso muestra que el handshake se ha realizado con éxito.

2. Use Dijkstra's to get the routing tables for nodes A, B and E



Nodo A:


5
D
E
2
5
2
2
4
1
A
B
C

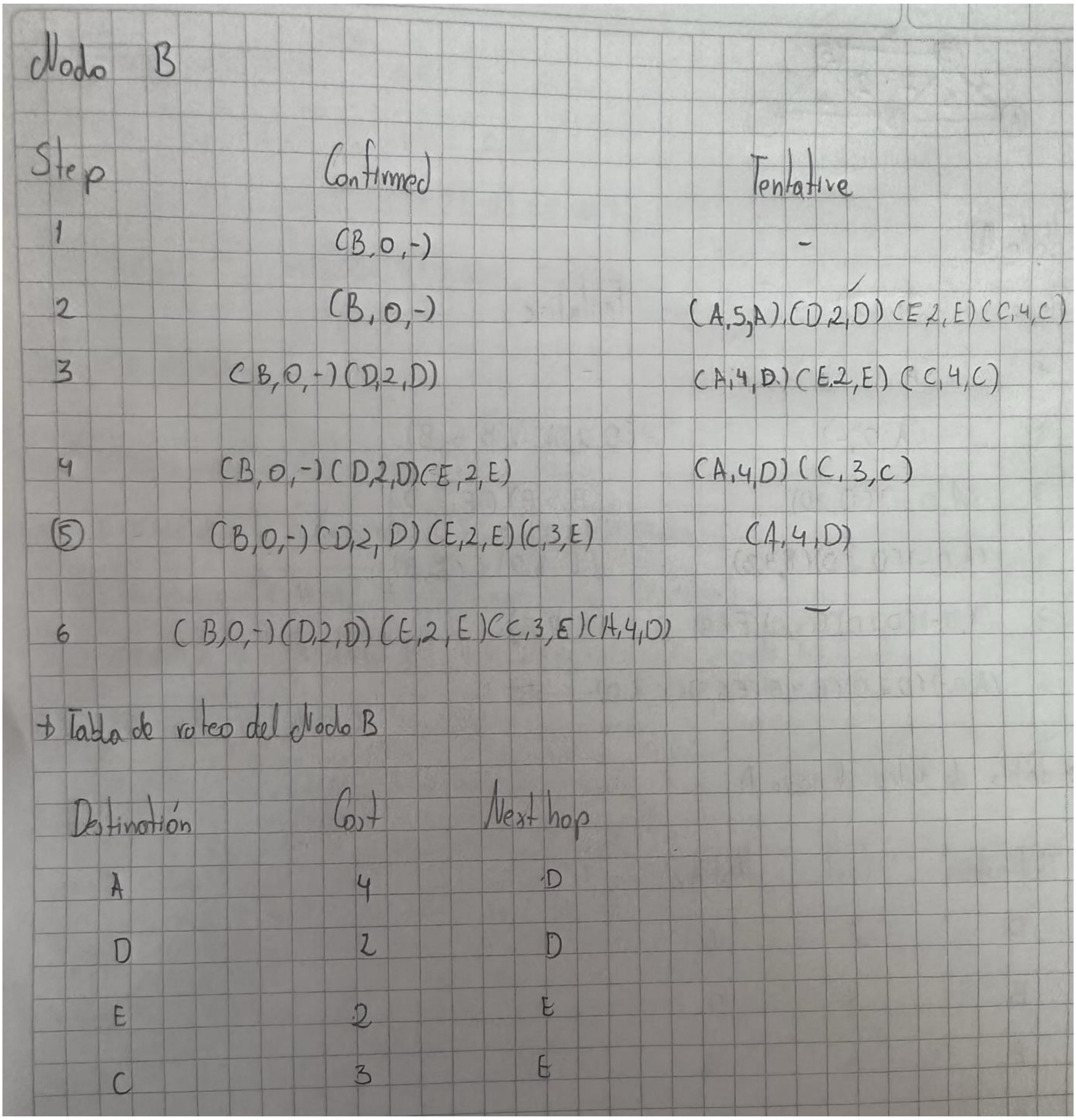
Nodo Ⓐ

Steps	Confirmed	Tentative
1	(A,0,-)	-
2	(A,0,-)	(D,2,D) (B,5,B)
3	(A,0,-)(D,2D)	(B,5,B) (E,7,D)
4	(A,0,-)(D,2D)(B,4,B)	(E,6,D) (C,8,B)
5	(A,0,-)(D,2D)(B,4,D)(E,6B)	(C,7,B)
6	(A,0,-)(D,2,D)(B,4,B)(E,6,B)(C,7,D)	—

→ Tabla de Ruted de nodo A

Destination	Cost	Next Hop
D	2	D
B	4	D
E	6	B
C	7	E

Nodo B:



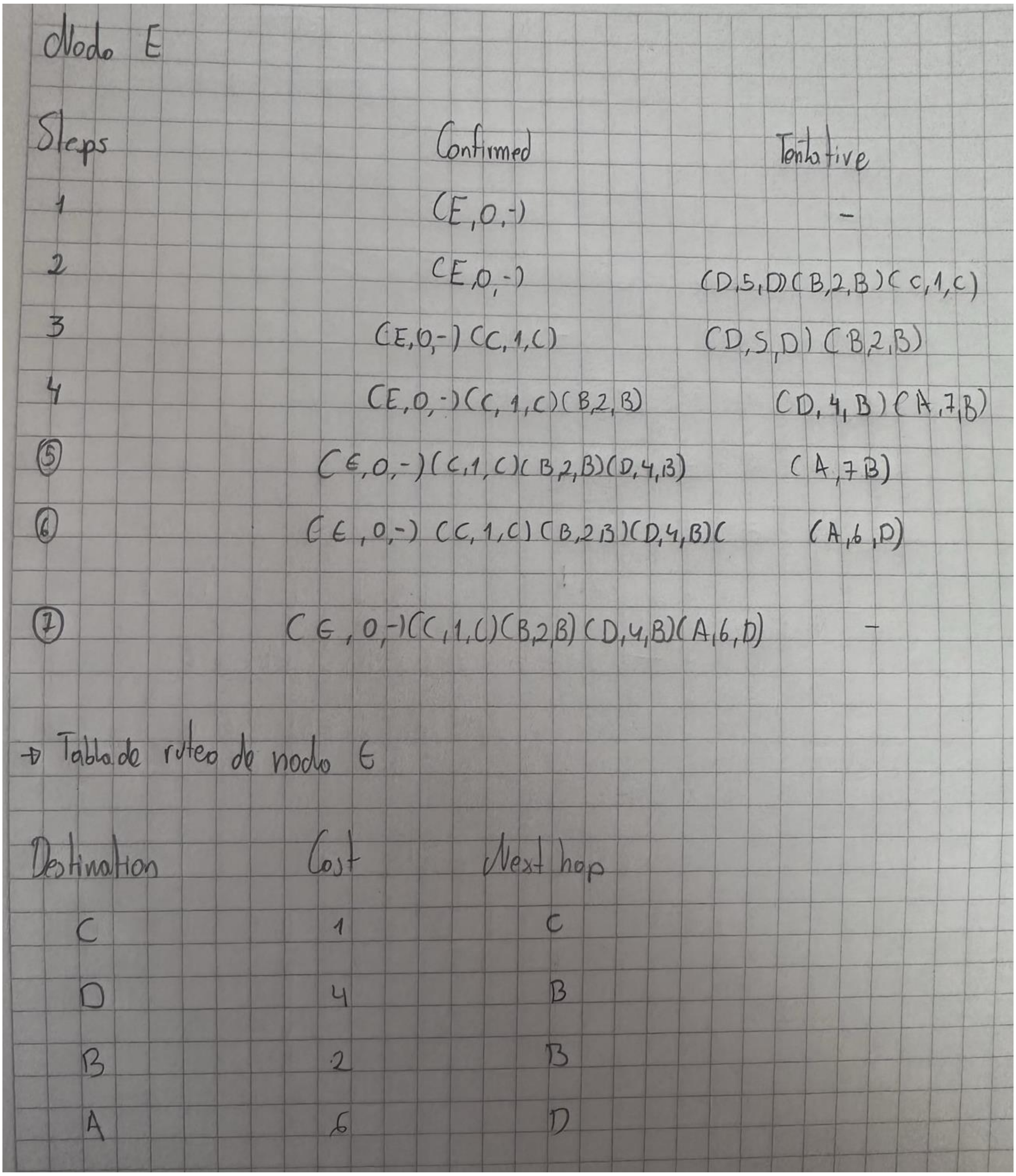
Nodo B

Step	Confirmed	Tentative
1	(B,0,-)	-
2	(B,0,-)	(A,5,A)(D,2,D)(E,2,E)(C,4,C)
3	(B,0,-)(D,2,D)	(A,4,D)(E,2,E)(C,4,C)
4	(B,0,-)(D,2,D)(E,2,E)	(A,4,D)(C,3,c)
5	(B,0,-)(D,2,D)(E,2,E)(C,3,E)	(A,4,D)
6	(B,0,-)(D,2,D)(E,2,E)(C,3,E)(A,4,D)	—

→ Tabla de roteo del Nodo B

Destinatión	Cost	Next hop
A	4	D
D	2	D
E	2	E
C	3	E

Nodo E:



Nodo E

Steps	Confirmed	Tentative
1	(E,0,-)	-
2	(E,0,-)	(D,5,D)(B,2,B)(C,1,C)
3	(E,0,-) (C,1,C)	(D,5,D) (B,2,B)
4	(E,0,-)(C,1,C)(B,2,B)	(D,4,B)(A,7,B)
5	(E,0,-)(C,1,C)(B,2,B)(D,4,B)	(A,7B)
6	(E,0,-) (C,1,C) (B,2B)(D,4,B)(	(A,6,D)
7	(E,0,-)(C,1,C)(B,2B) (D,4,B)(A,6,D)	-

→ Tabla de ruteo de nodo E

Destination	Cost	Next hop
C	1	C
D	4	B
B	2	B
A	6	D

3. Suppose a host wants to establish the reliability of a link by sending packets and measuring the percentage that are received; routers, for example, do this. Explain the difficulty of doing this over a TCP connection.

Respuesta:

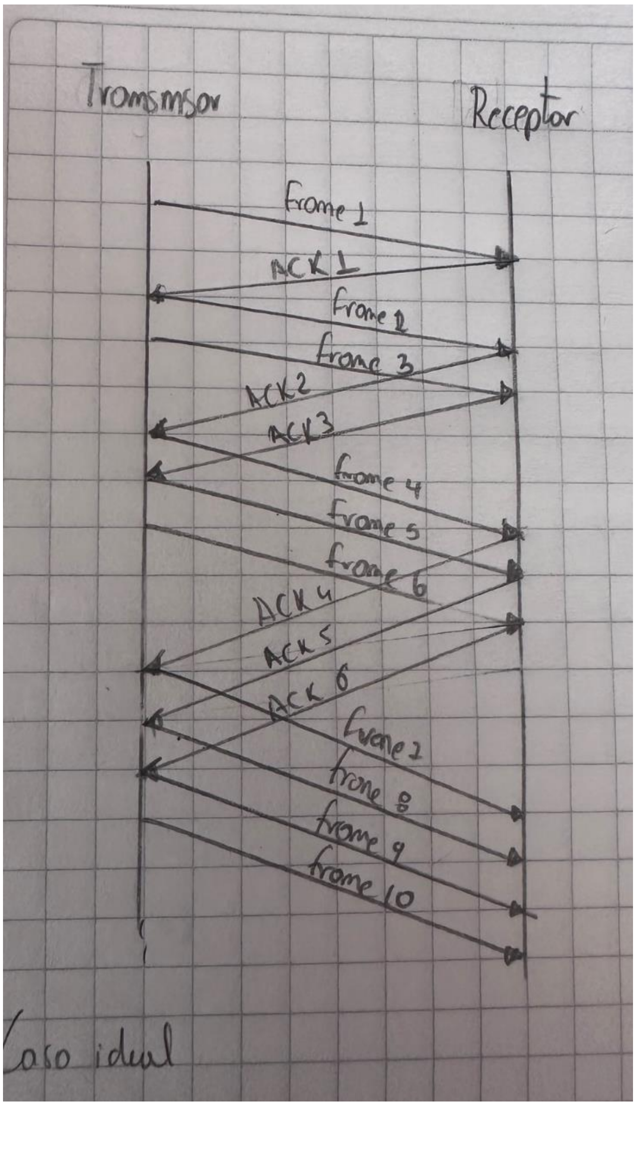
En este caso hay un problema si no tiene alguna forma de saber si el paquete llego al realizar el primer intento, o si fallo, no se hizo una retransmisión. Por lo que se necesita que el receptor responda inmediatamente y también debe tener la capacidad de medir el tiempo transcurrido, para que cuando se agote el tiempo de espera, vuelva a retransmitir. Esto para garantizar la confiabilidad de la conexión establecida. Utilizando una conexión TCP requiere que el enrutador transfiera datos de un lado a otro entre los hosts. Otra manera seria medir el porcentaje de paquetes recibidos, eso se puede lograr mediante implementaciones de Berkeley, que son protocolos de comunicación, a partir del protocolo TCP, por lo tanto, permitimos medir el tiempo de espero con una granularidad de 0.5 segundos y RTT's.

4. Consider a simple congestion control algorithm that uses linear increase and multiplicative decrease (no slow start). Assume the congestion window size is in units of packets rather than bytes, and it is one packet initially.

a) Give a detailed sketch of this algorithm.

Considerando un tamaño con valor 1 de ventana por parte del remitente, se envía la ventana llena en un solo lote, y por cada Acknowledgment que el remitente recibe, este aumentara la ventana efectiva en uno. En el caso que exista un tiempo de espera, la ventana se reduce a la mitad en términos de número de paquetes.

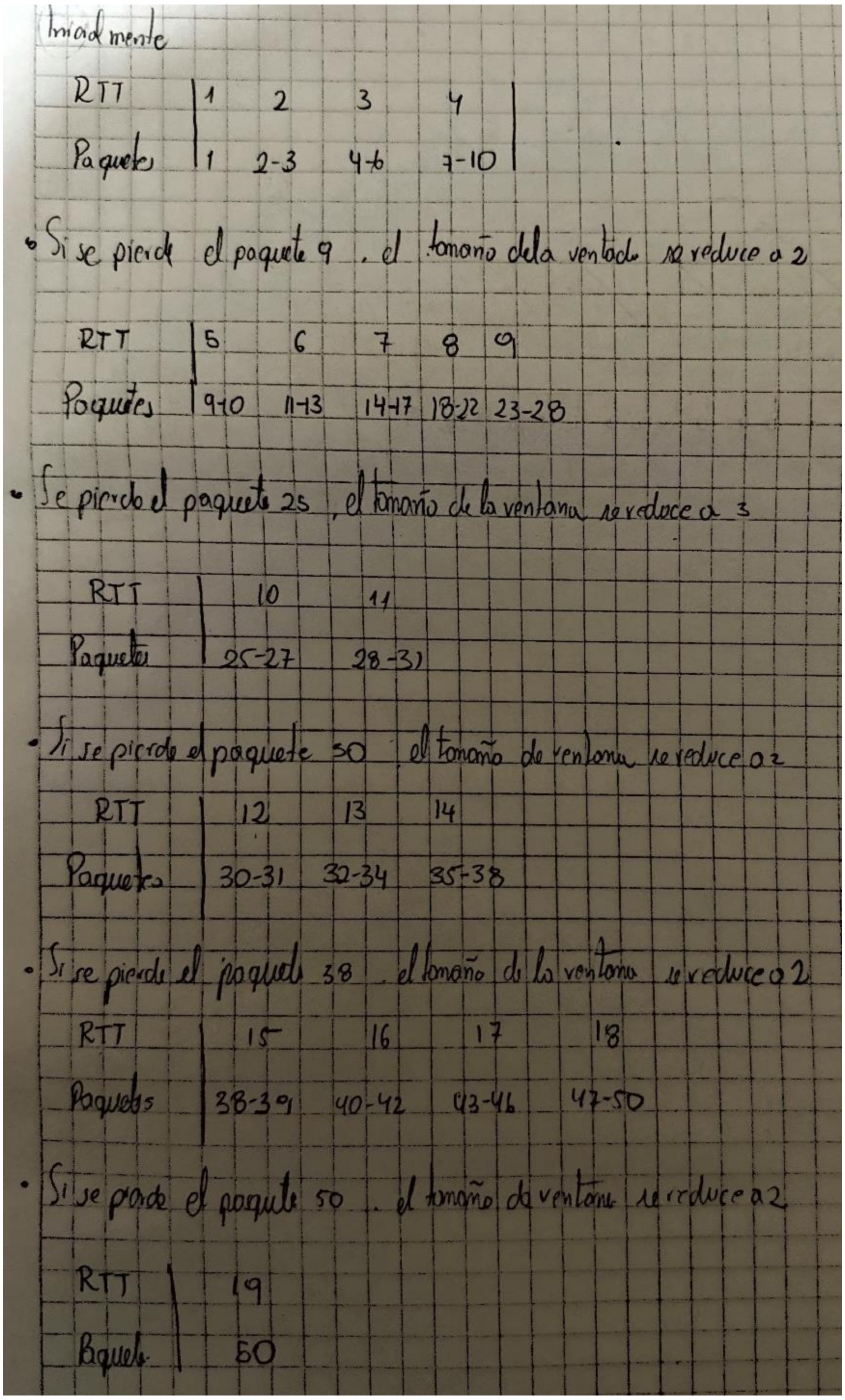
Sketch:



b) Assume the delay is latency only, and that when a group of packets is sent, only a single ACK is returned.

cthe situation in which the following packets are lost: 9, 25, 30, 38 and 50. For simplicity, assume a perfect timeout mechanism that detects a lost packet exactly 1 RTT after it is transmitted.

En este caso si se pierden estos paquetes el tamaño de la ventana tendrá un valor de 1. Cuando se recibe el primer ACK aumenta a 2, al inicio del segundo RTT se envían los paquetes 2 y 3. Al recibir los respectivos ACK, se aumenta el tamaño de ventana a 3, y posteriormente se envían los paquetes 4,5 y 6. Al recibir esos ACK´s, el tamaño de ventana aumenta a 4.



Inicialmente

RTT	1	2	3	4
Paquetes	1	2-3	4-6	7-10

- Si se pierde el paquete 9 , el tamaño dela ventada se reduce a 2

RTT	5	6	7	8	9
Paquetes	9-10	11-13	14-17	18-22	23-28

- Se pierde el paquete 25 , el tamaño de la ventana se reduce a 3

RTT	10	11
Paquetes	25-27	28-31

- Si se pierde el paquete 30 el tamaño de ventana se reduce a 2

RTT	12	13	14
Paquetes	30-31	32-34	35-38

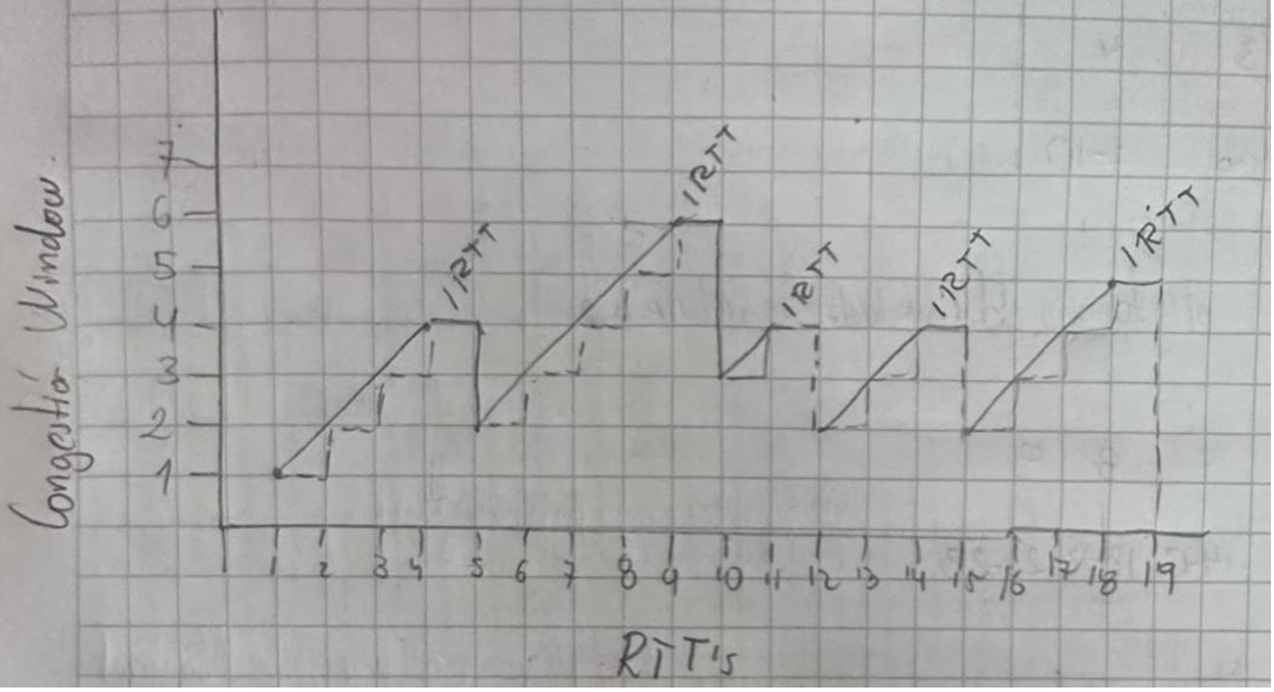
- Si se pierde el paquete 38 . el tamaño de la ventana se reduce a 2

RTT	15	16	17	18
Paquetes	38-39	40-42	43-46	47-50

- Si se pierde el paquete 50 . el tamaño de ventana se reduce a 2

RTT	19
Paquete	50

Plot


Congestió Window.
7
6
5
4
3
2
1
1RTT
1RTT
1RTT
1RTT
1RTT
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
RTT's