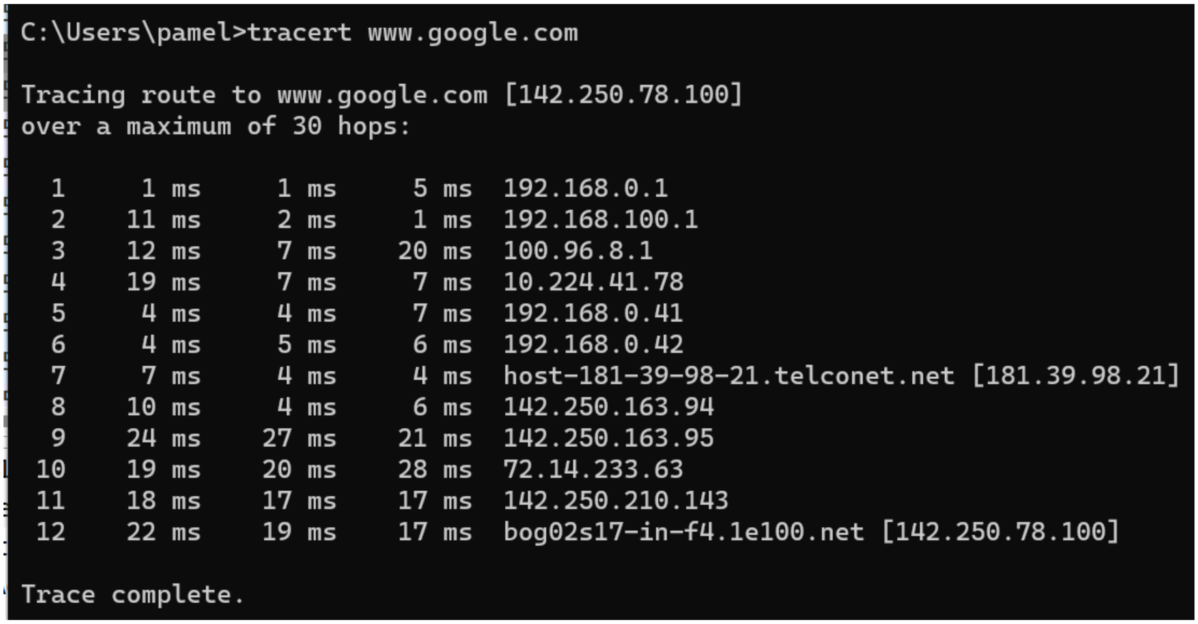
**Pamela Mena – 00211564**

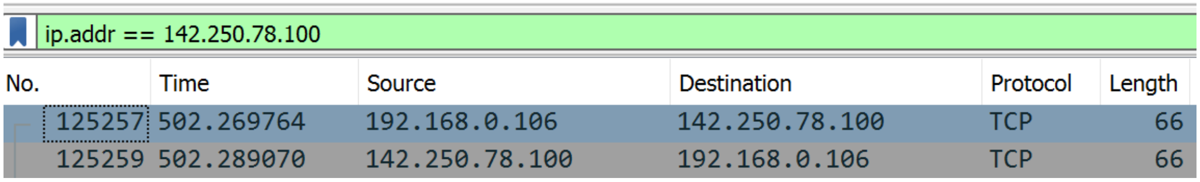
## Universidad San Francisco de Quito

## Tarea 03

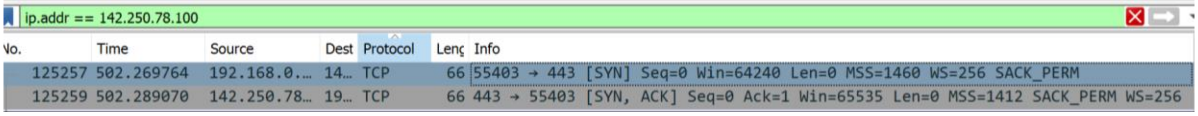
1. **Read the following Wireshark tutorial and use it to capture traffic from the following scenarios. Use screenshots to show your results.**
    a. **Run 10 traceroute commands against google.com.**



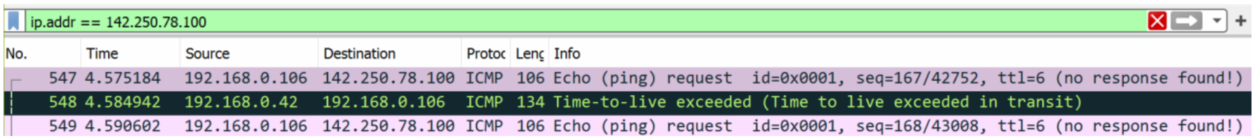
```
C:\Users\pamel>tracert www.google.com

Tracing route to www.google.com [142.250.78.100]
over a maximum of 30 hops:

  1     1 ms     1 ms     5 ms  192.168.0.1
  2    11 ms     2 ms     1 ms  192.168.100.1
  3    12 ms     7 ms    20 ms  100.96.8.1
  4    19 ms     7 ms     7 ms  10.224.41.78
  5     4 ms     4 ms     7 ms  192.168.0.41
  6     4 ms     5 ms     6 ms  192.168.0.42
  7     7 ms     4 ms     4 ms  host-181-39-98-21.telconet.net [181.39.98.21]
  8    10 ms     4 ms     6 ms  142.250.163.94
  9    24 ms    27 ms    21 ms  142.250.163.95
 10    19 ms    20 ms    28 ms  72.14.233.63
 11    18 ms    17 ms    17 ms  142.250.210.143
 12    22 ms    19 ms    17 ms  bog02s17-in-f4.1e100.net [142.250.78.100]

Trace complete.
```



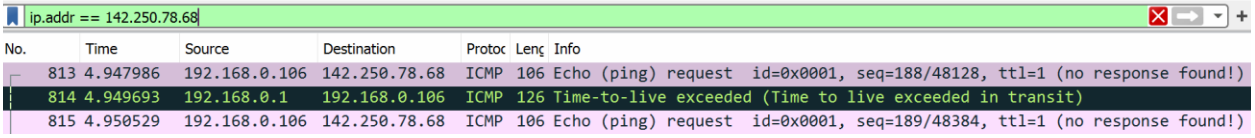
```
ip.addr == 142.250.78.100
No.     Time        Source          Destination     Protocol  Length
125257  502.269764  192.168.0.106   142.250.78.100  TCP       66
125259  502.289070  142.250.78.100  192.168.0.106   TCP       66
```



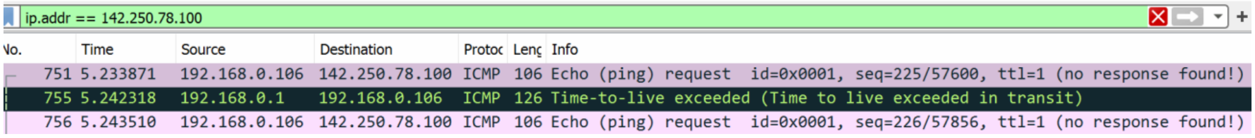
```
ip.addr == 142.250.78.100
No.     Time        Source        Dest  Protocol  Leng  Info
125257  502.269764  192.168.0...  14…   TCP       66    55403 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
125259  502.289070  142.250.78…   19…   TCP       66    443 → 55403 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1412 SACK_PERM WS=256
```



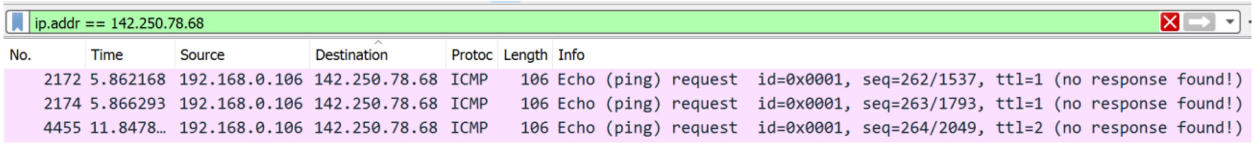
```
ip.addr == 142.250.78.100
No.  Time      Source         Destination     Protoc Leng Info
547  4.575184  192.168.0.106  142.250.78.100  ICMP   106  Echo (ping) request  id=0x0001, seq=167/42752, ttl=6 (no response found!)
548  4.584942  192.168.0.42   192.168.0.106   ICMP   134  Time-to-live exceeded (Time to live exceeded in transit)
549  4.590602  192.168.0.106  142.250.78.100  ICMP   106  Echo (ping) request  id=0x0001, seq=168/43008, ttl=6 (no response found!)
```



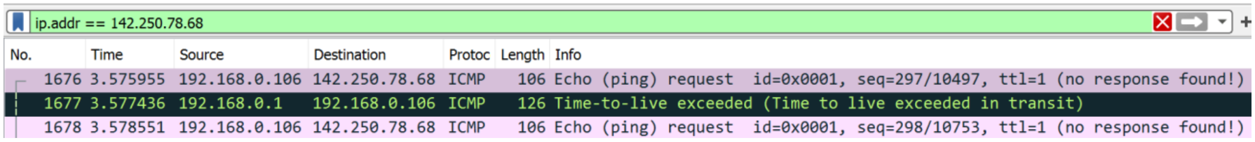
```
ip.addr == 142.250.78.68
No.  Time      Source         Destination    Protoc Leng Info
813  4.947986  192.168.0.106  142.250.78.68  ICMP   106  Echo (ping) request  id=0x0001, seq=188/48128, ttl=1 (no response found!)
814  4.949693  192.168.0.1    192.168.0.106  ICMP   126  Time-to-live exceeded (Time to live exceeded in transit)
815  4.950529  192.168.0.106  142.250.78.68  ICMP   106  Echo (ping) request  id=0x0001, seq=189/48384, ttl=1 (no response found!)
```



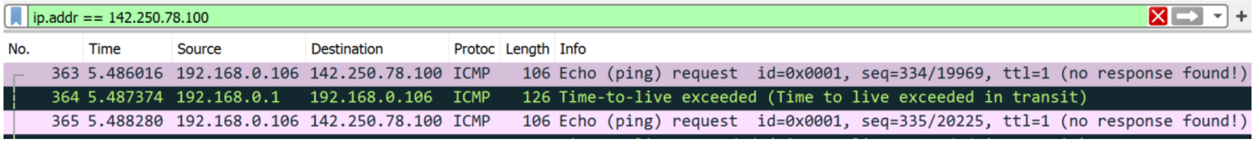
```
ip.addr == 142.250.78.100
No.  Time     Source        Destination    Protoc Leng Info
 751 5.233871 192.168.0.106 142.250.78.100 ICMP  106 Echo (ping) request  id=0x0001, seq=225/57600, ttl=1 (no response found!)
 755 5.242318 192.168.0.1   192.168.0.106  ICMP  126 Time-to-live exceeded (Time to live exceeded in transit)
 756 5.243510 192.168.0.106 142.250.78.100 ICMP  106 Echo (ping) request  id=0x0001, seq=226/57856, ttl=1 (no response found!)
```



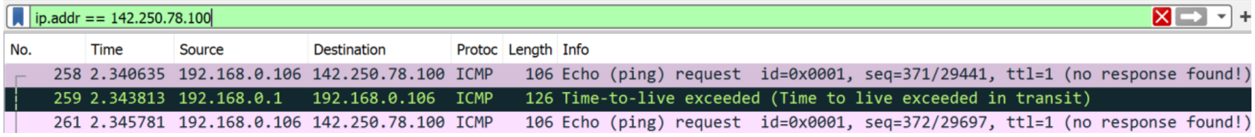
```
ip.addr == 142.250.78.68
No.   Time      Source        Destination   Protoc Length Info
 2172 5.862168  192.168.0.106 142.250.78.68 ICMP    106 Echo (ping) request  id=0x0001, seq=262/1537, ttl=1 (no response found!)
 2174 5.866293  192.168.0.106 142.250.78.68 ICMP    106 Echo (ping) request  id=0x0001, seq=263/1793, ttl=1 (no response found!)
 4455 11.8478…  192.168.0.106 142.250.78.68 ICMP    106 Echo (ping) request  id=0x0001, seq=264/2049, ttl=2 (no response found!)
```



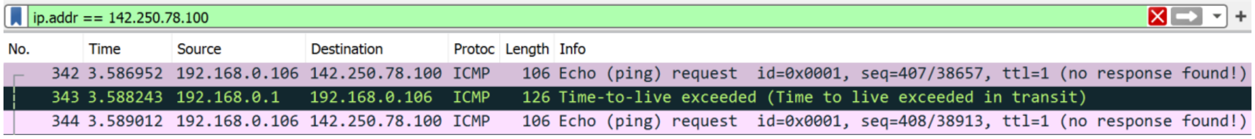
```
ip.addr == 142.250.78.68
No.   Time     Source        Destination   Protoc Length Info
 1676 3.575955 192.168.0.106 142.250.78.68 ICMP    106 Echo (ping) request  id=0x0001, seq=297/10497, ttl=1 (no response found!)
 1677 3.577436 192.168.0.1   192.168.0.106 ICMP    126 Time-to-live exceeded (Time to live exceeded in transit)
 1678 3.578551 192.168.0.106 142.250.78.68 ICMP    106 Echo (ping) request  id=0x0001, seq=298/10753, ttl=1 (no response found!)
```



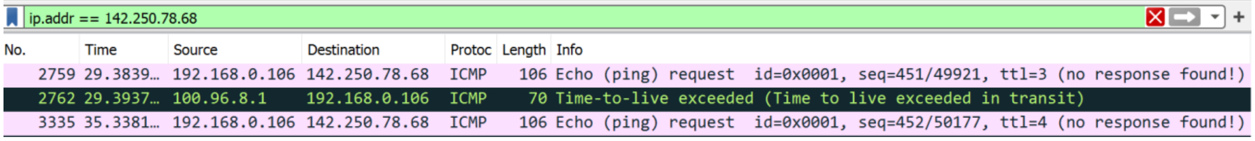
```
ip.addr == 142.250.78.100
No.   Time     Source        Destination    Protoc Length Info
  363 5.486016 192.168.0.106 142.250.78.100 ICMP    106 Echo (ping) request  id=0x0001, seq=334/19969, ttl=1 (no response found!)
  364 5.487374 192.168.0.1   192.168.0.106  ICMP    126 Time-to-live exceeded (Time to live exceeded in transit)
  365 5.488280 192.168.0.106 142.250.78.100 ICMP    106 Echo (ping) request  id=0x0001, seq=335/20225, ttl=1 (no response found!)
```



```
ip.addr == 142.250.78.100
No.   Time     Source        Destination    Protoc Length Info
  258 2.340635 192.168.0.106 142.250.78.100 ICMP    106 Echo (ping) request  id=0x0001, seq=371/29441, ttl=1 (no response found!)
  259 2.343813 192.168.0.1   192.168.0.106  ICMP    126 Time-to-live exceeded (Time to live exceeded in transit)
  261 2.345781 192.168.0.106 142.250.78.100 ICMP    106 Echo (ping) request  id=0x0001, seq=372/29697, ttl=1 (no response found!)
```

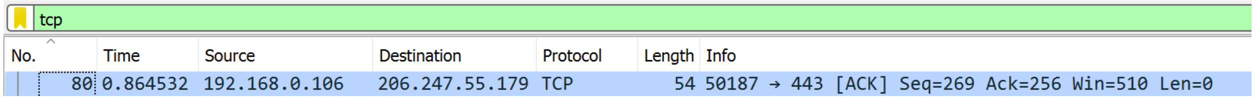


```
ip.addr == 142.250.78.100
No.   Time     Source        Destination    Protoc Length Info
  342 3.586952 192.168.0.106 142.250.78.100 ICMP    106 Echo (ping) request  id=0x0001, seq=407/38657, ttl=1 (no response found!)
  343 3.588243 192.168.0.1   192.168.0.106  ICMP    126 Time-to-live exceeded (Time to live exceeded in transit)
  344 3.589012 192.168.0.106 142.250.78.100 ICMP    106 Echo (ping) request  id=0x0001, seq=408/38913, ttl=1 (no response found!)
```

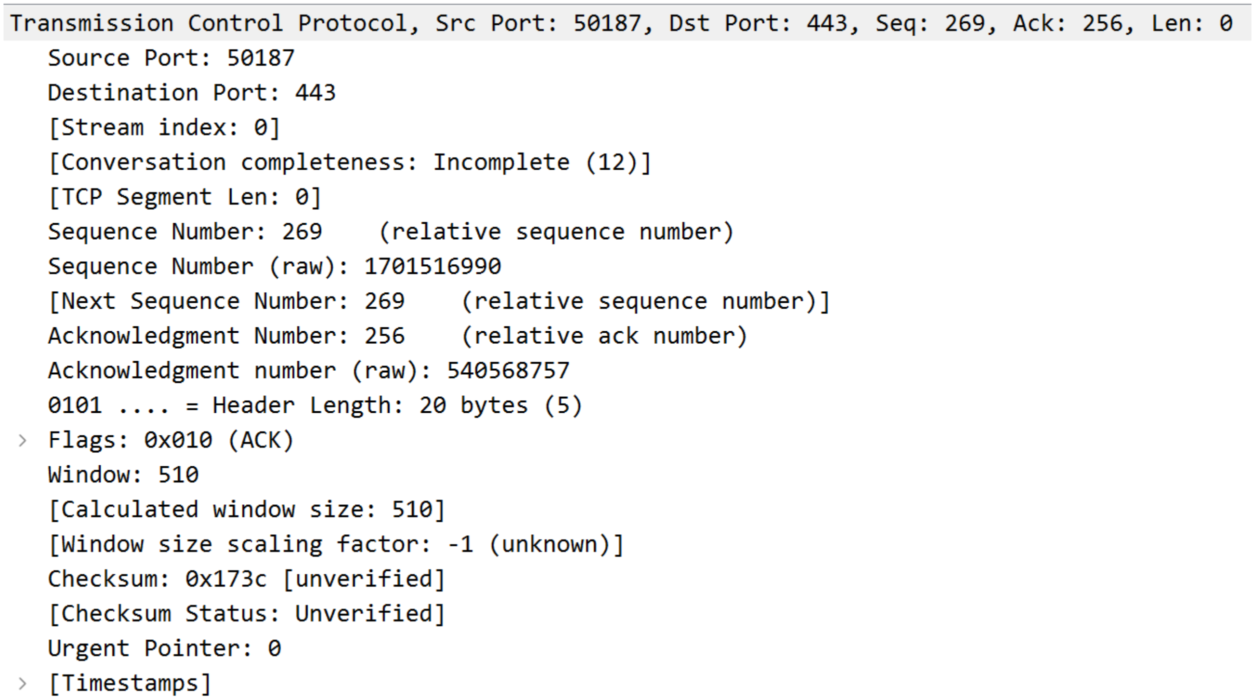


```
ip.addr == 142.250.78.68
No.   Time      Source        Destination   Protoc Length Info
 2759 29.3839…  192.168.0.106 142.250.78.68 ICMP    106 Echo (ping) request  id=0x0001, seq=451/49921, ttl=3 (no response found!)
 2762 29.3937…  100.96.8.1    192.168.0.106 ICMP     70 Time-to-live exceeded (Time to live exceeded in transit)
 3335 35.3381…  192.168.0.106 142.250.78.68 ICMP    106 Echo (ping) request  id=0x0001, seq=452/50177, ttl=4 (no response found!)
```

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

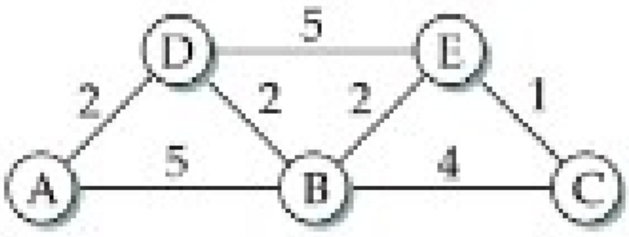


```
tcp
No.     Time      Source         Destination    Protocol  Length Info
     80 0.864532  192.168.0.106  206.247.55.179 TCP           54 50187 → 443 [ACK] Seq=269 Ack=256 Win=510 Len=0
```



```
Transmission Control Protocol, Src Port: 50187, Dst Port: 443, Seq: 269, Ack: 256, Len: 0
   Source Port: 50187
   Destination Port: 443
   [Stream index: 0]
   [Conversation completeness: Incomplete (12)]
   [TCP Segment Len: 0]
   Sequence Number: 269    (relative sequence number)
   Sequence Number (raw): 1701516990
   [Next Sequence Number: 269    (relative sequence number)]
   Acknowledgment Number: 256    (relative ack number)
   Acknowledgment number (raw): 540568757
   0101 .... = Header Length: 20 bytes (5)
 > Flags: 0x010 (ACK)
   Window: 510
   [Calculated window size: 510]
   [Window size scaling factor: -1 (unknown)]
   Checksum: 0x173c [unverified]
   [Checksum Status: Unverified]
   Urgent Pointer: 0
 > [Timestamps]
```

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

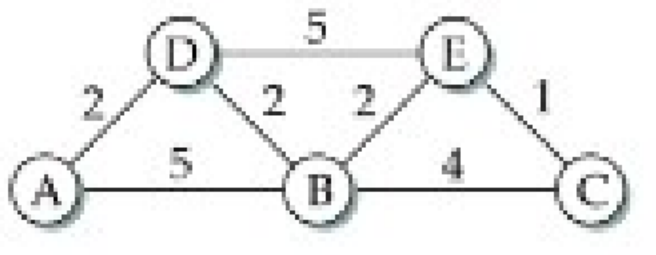


Routing para nodo A:

| Nodo | Costo | Salto |
|---|---|---|
| A | 0 | A |
| B | 5 | B – A |
| C | 9 | B – A |
| D | 2 | D – A |
| E | 7 | D – A |

Routing para nodo B:

| Nodo | Costo | Salto |
|---|---|---|
| A | 5 | A – B |
| B | 0 | B |
| C | 4 | C – B |
| D | 2 | D – B |
| E | 2 | E – B |



Routing para nodo E:

| Nodo | Costo | Salto |
|---|---|---|
| A | 7 | D – E |
| B | 2 | B – E |
| C | 1 | C – E |
| D | 5 | D – E |
| E | 0 | E |

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.**

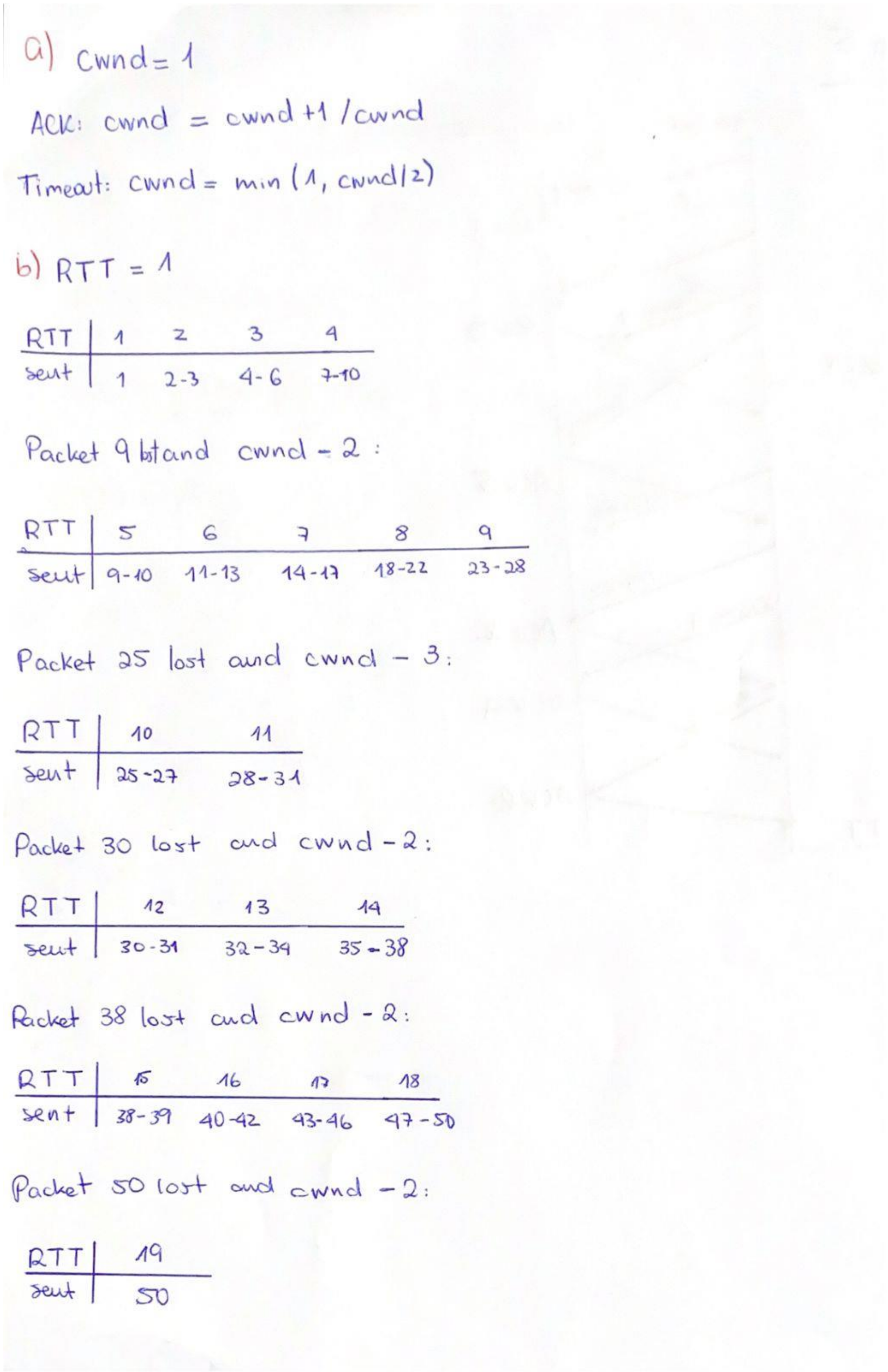
Cuando los paquetes no llegan a su destino a través de una conexión TCP, generalmente se debe a congestión en lugar de problemas de confiabilidad del enlace. Identificar la causa de un paquete perdido puede ser difícil. La asignación insuficiente de recursos puede resultar en dos problemas principales: posible injusticia y un estado de bloqueo conocido como "deadlock". Primero, exploremos el "deadlock". Si todos los buffers de un router están llenos de paquetes, el router no podrá aceptar nuevos marcos. Aún más problemático, podría ignorar marcos con ACKs que podrían liberar algunos de esos buffers. Por ejemplo, si dos routers vecinos, A y B, se envían paquetes entre sí y ambos están esperando que el otro reciba un paquete, ninguno puede avanzar. Esta situación se conoce como un "deadlock".

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.**

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

c. **Plot the congestion window as a function of RTT for the 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.**



a) cwnd = 1

ACK: cwnd = cwnd +1 /cwnd

Timeout: cwnd = min (1, cwnd/2)

b) RTT = 1

| RTT | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| sent | 1 | 2-3 | 4-6 | 7-10 |

Packet 9 lost and cwnd - 2 :

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

Packet 25 lost and cwnd - 3:

| RTT | 10 | 11 |
|---|---|---|
| sent | 25-27 | 28-31 |

Packet 30 lost and cwnd - 2:

| RTT | 12 | 13 | 14 |
|---|---|---|---|
| sent | 30-31 | 32-34 | 35-38 |

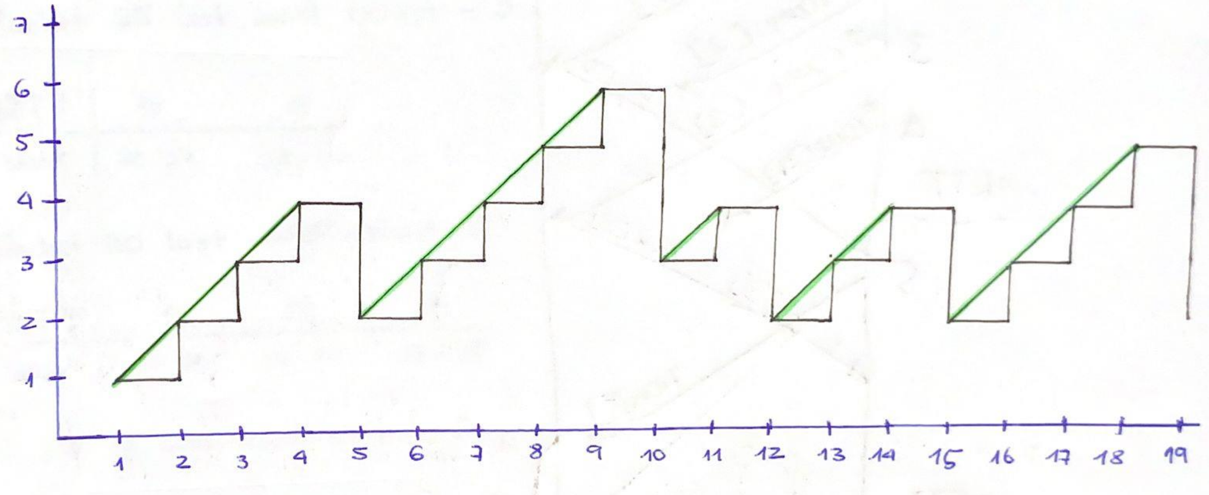
Packet 38 lost and cwnd - 2:

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

Packet 50 lost and cwnd - 2:

| RTT | 19 |
|---|---|
| sent | 50 |

d.


7
6
5
4
3
2
1
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19