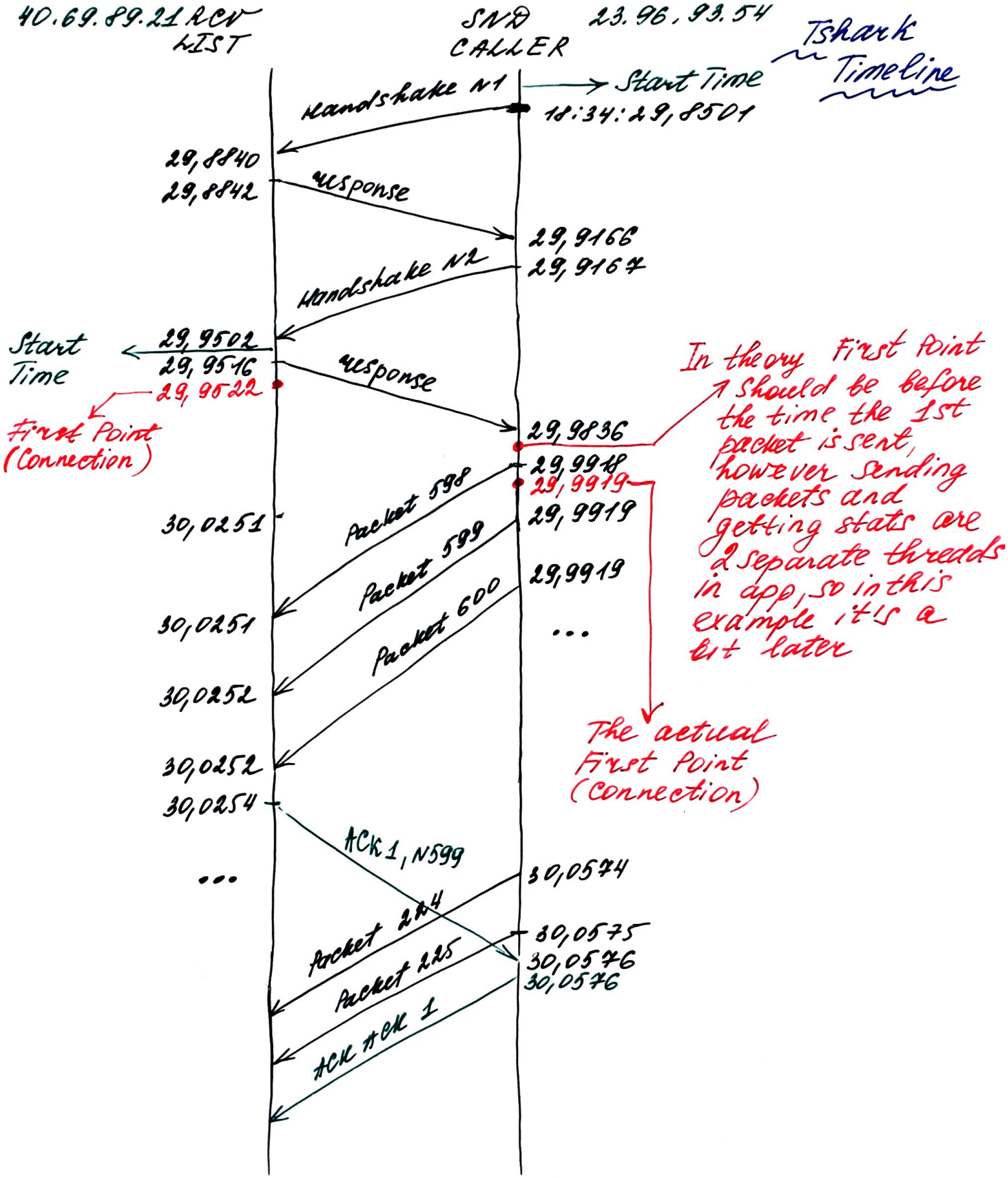
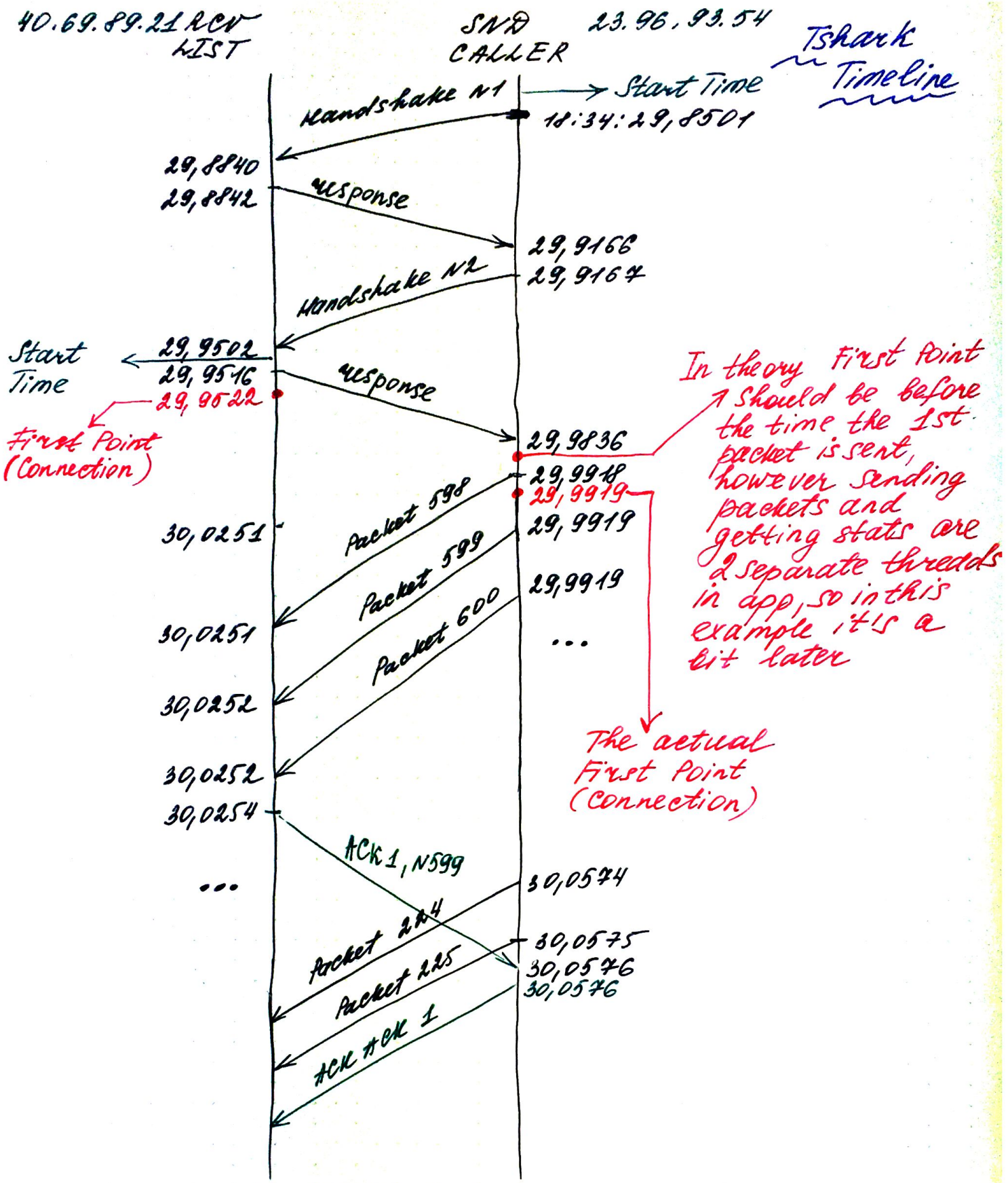
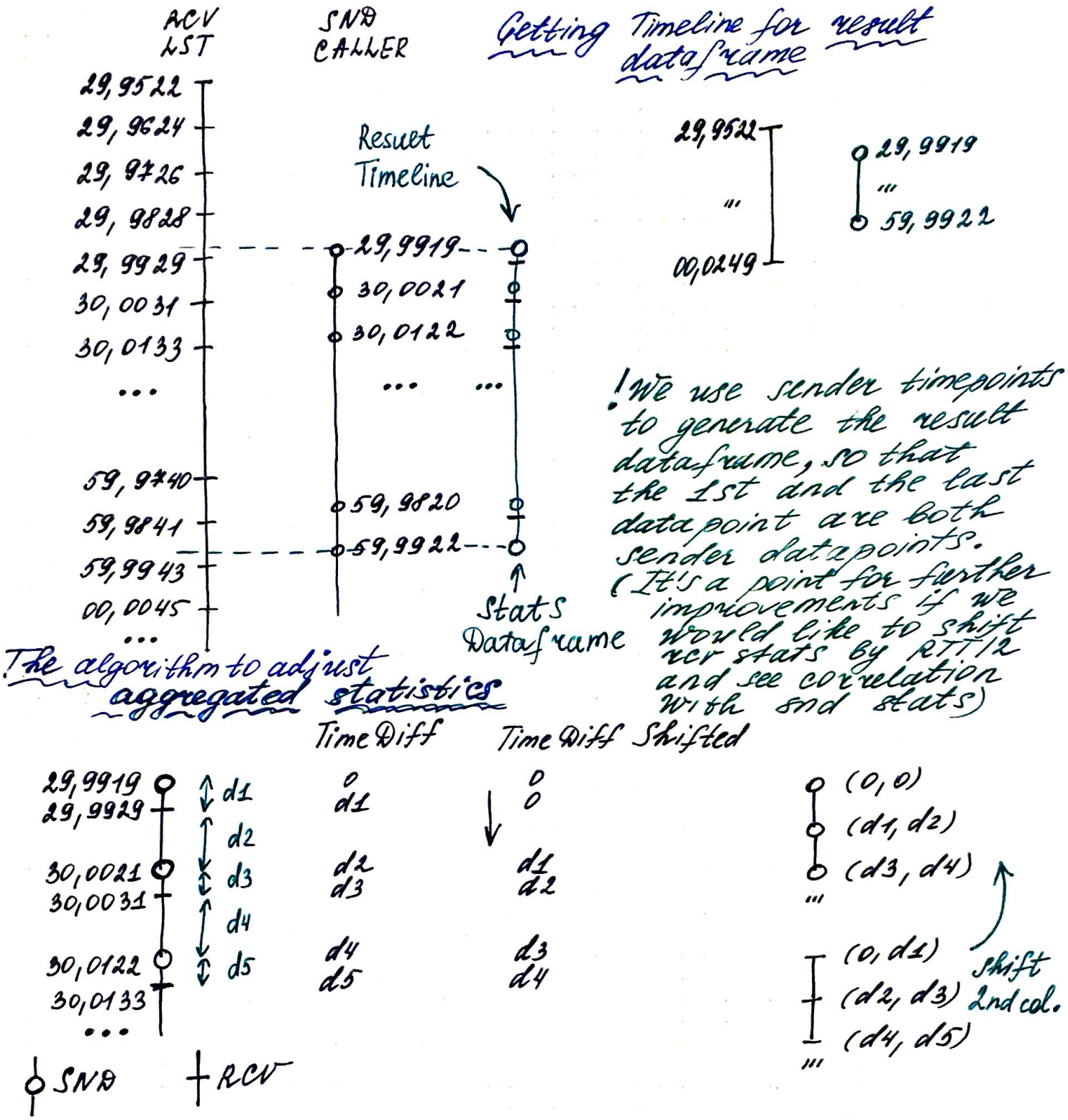
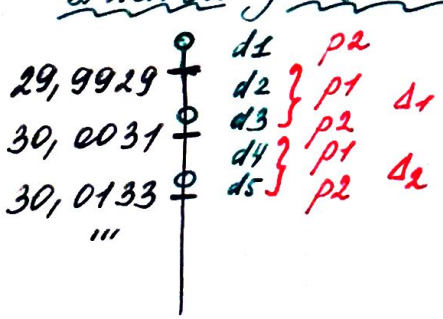

40.69.89.21 RCV LIST
SND CALLER
23.96.93.54
Tshark Timeline
Handshake N1
Start Time
18:34:29,8501
29,8840
29,8842
Response
29,9166
29,9167
Handshake N2
Start Time
29,9502
29,9516
29,9522
Response
First Point (Connection)
29,9836
In theory First Point should be before the time the 1st packet is sent, however sending packets and getting stats are 2 separate threads in app, so in this example it's a bit later
29,9918
29,9919
30,0251
Packet 598
29,9919
Packet 599
29,9919
Packet 600
30,0251
...
30,0252
30,0252
The actual First Point (Connection)
30,0254
ACK 1, N599
...
30,0574
Packet 224
30,0575
Packet 225
30,0576
30,0576
ACK #ck 1


40.69.89.21 RCV
LIST
SND
CALLER
23.96.93.54
Tshark
Timeline
Handshake N1
Start Time
18:34:29,8501
29,8840
29,8842
Response
29,9166
29,9167
Handshake N2
Start
Time
29,9502
29,9516
29,9522
Response
First Point
(Connection)
29,9836
In theory First Point
should be before
the time the 1st
packet is sent,
however sending
packets and
getting stats are
2 separate threads
in app, so in this
example, it's a
bit later
29,9918
29,9919
30,0251
Packet 598
29,9919
Packet 599
29,9919
Packet 600
30,0251
...
30,0252
30,0252
30,0254
The actual
First Point
(Connection)
ACK 1, N599
...
30,0574
Packet 224
30,0575
Packet 225
30,0576
30,0576
ACK #CK 1

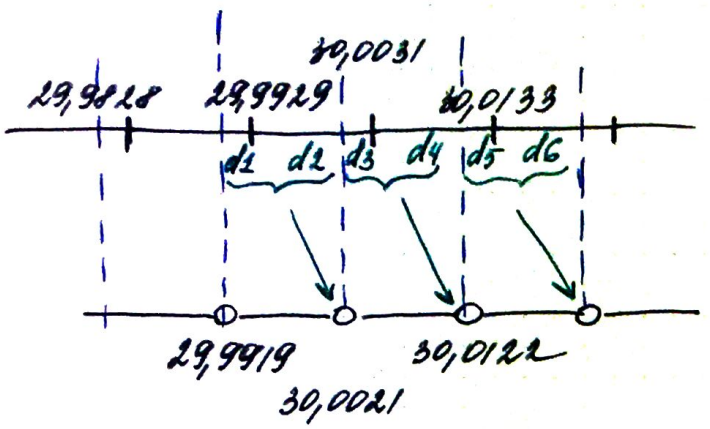


Auxilary df to work with RCV stats



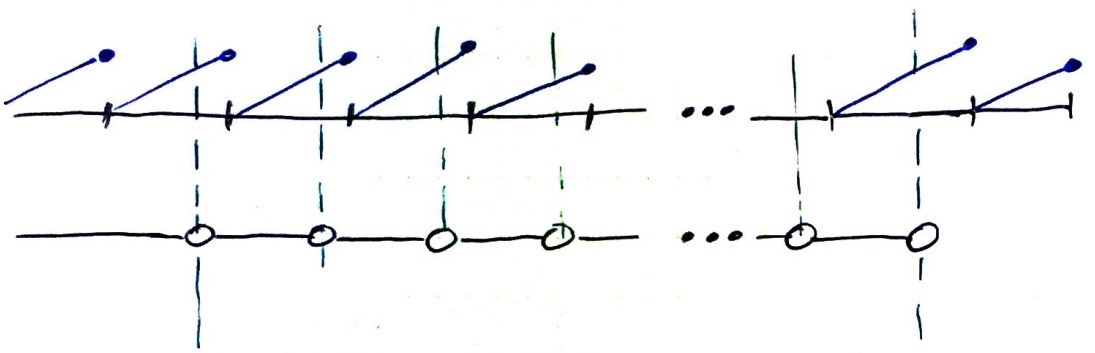
	TimeDiff_p1	TimeDiff_p2	RCV TimeDiff
	0	d1	mean(Δ1, ..., Δn)
	d2	d3	Δ1
	d4	d5	Δ2
			...

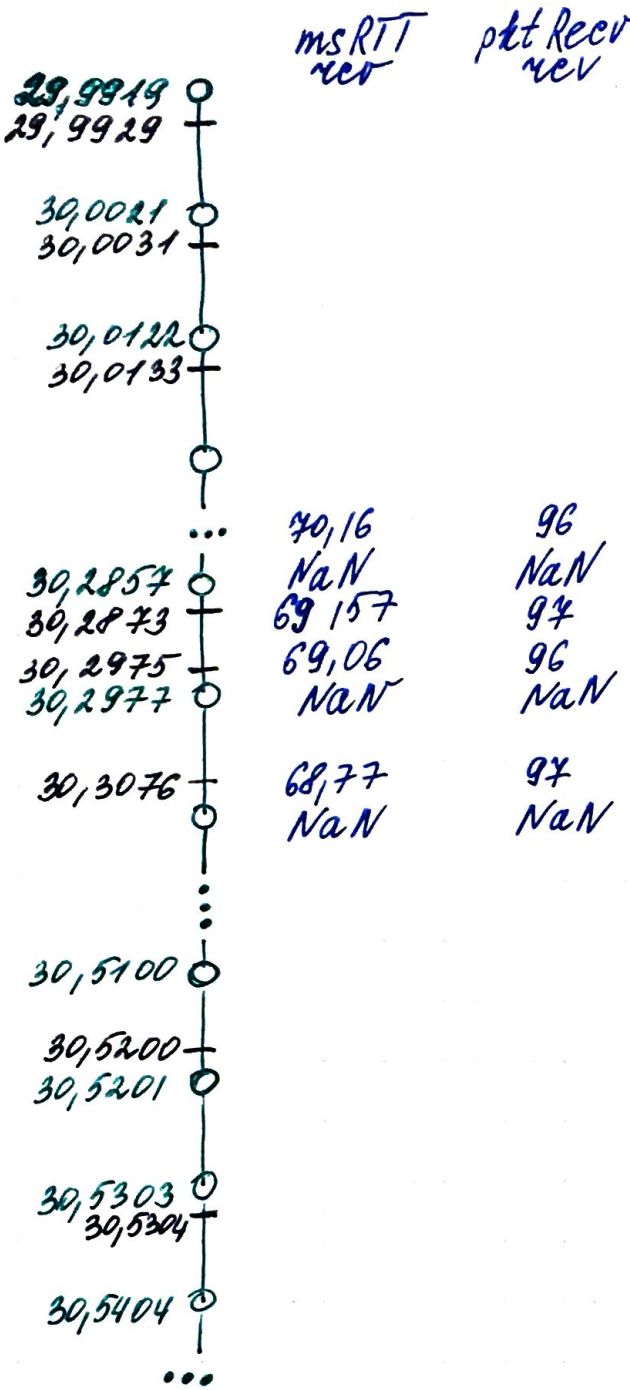
! Calculate which % d2, d3 is out of Δ1



! However, we can use linear interpolation for aggregated statistics as well

...




msRTT rcv
pktRecv rcv
29,9919
29,9929
30,0021
30,0031
30,0122
30,0133
...
70,16
96
30,2857
NaN
NaN
30,2873
69,157
97
30,2975
69,06
96
30,2977
NaN
NaN
30,3076
68,77
97
NaN
NaN
30,5100
30,5200
30,5201
30,5303
30,5304
30,5404
...