Homework 9

Problem 1:

Fill in the sequence and acknowledgment numbers for the beginning of a TCP session. Server A uses an initial sequence number of 3826305083 and Server B uses an initial sequence number of 3879144537. The port chosen by A is 55534 and the port chosen by B is 22 (SSH, SFTP). Only during the initial three way handshake will packets with SYN flag have one fictitious byte of data.

Packet 1:
Syn-packet: A:55534 -> B:22
seq 3826305083, ack 0, flags: SYN

Packet 2:
Syn-ack packet: B:22 -> A:55534
seq 3879144537, ack 3826305084 flags: SYN ACK

Packet 3:
Ack packet: A:55534 -> B:22
seq 3826305084, ack 3879144538 flags: ACK

Packet 4: from A to B has 21 data
A:55534 -> B:22
seq 3826305084, ack 3879144538 + 21 data, flags: ACK

Packet 5: from B to A has no data, but acknowledges Packet 4

Packet 6: from B to A has 21 data
Packet 6:
B:22 -> A:55534
seq 3879144538 ack 3826305105 + 21 data, flags: ACK

Packet 7 from A to B has no data, but acknowledges packet 6

Packet 8 from A to B has 1368 data and acknowledges Packet 7

Packet 9 from B to A has 784 data and acknowledges Packet 8

Packet 10 from A to B has no data and acknowledges Packet 9

Packet 11 from B to A has no data and acknowledges Packet 10

Packet 12 from A to B has 48 data and acknowledges Packet 11

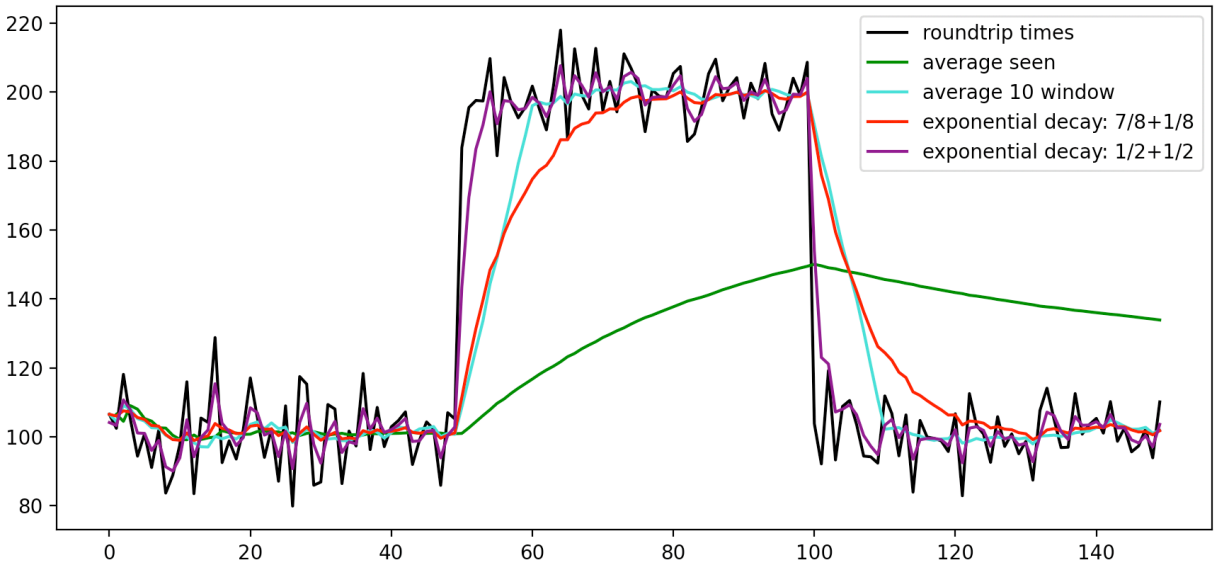
Problem 2:

The smoothed round-trip time time uses an exponential weighted average with parameters $(7/8,1/8)$ in order to estimate the next round-trip time. It has the advantage of storing only two

floating point values per TCP connection. Outdated data points still influence the smoothed round-trip time, but since they experience exponential decay, their influence becomees quickly negligible. Alternative ways to calculate smoothed round-trip times would be

1. to store all observed values and calculate the average as an estimate for the next round-trip-time. (In fact, you can calculate the new mean from a new value and the count of values, so you do **not** have to keep all values around.)
2. to store the last ten (e.g.) values and calculate the average.
3. To use exponential decay with a different convex combination of old and new values.

Below we give an example where the round-trip-time is for the first fifty packets normally distributed around 100, then for the next fifty packets around 200, and for the final fifty packets around 100 again with a standard deviation of 10. As you can see, the total average is close to useless in addition to be too storage intensive, whereas the average of a window of 10 packets is quite good at capturing changes. The linear coefficients used to calculate the new value for the exponential versions is difficult to optimize. Using $(1/2,1/2)$ does not lead to much smoothing, whereas $(7/8,1/8)$ is slower to adjust to new conditions. Its costs however are much superior than using an average of a window. Notice that other forms of moving average are used in time series analysis.



While it is dangerous to use synthetic data as we just did, it helps us to understand how smoothing works.

Your first task is to recreate the graph from the raw round-trip times provided in the associated data file. You can use any programming language and any reasonable tool to create the graphs. I obviously used the pyplot module of Python.

Your second task is to replace the 50 higher response times with a single response time of 200 and show how the different algorithms work.