P1.

AIMD. The throughput and delay both increase as we increase the size of the window. The system appears to have good throughput to delay at 19 window size.

P2.

Usually the only reason why timeouts happen is due to capacity drop of network in this setting. We can notice here in the animation that as the capacity drops as the throughput goes up so does the delay. How AIMD recovered cannot be solved reliably in networks that buffer all the packets and very high variance networks. Here the AIMD would keep growing the window despite filling up the network.

P3.

Here we look at the packets rtt’s if it is bigger than the threshold we decrease the window size otherwise we keep increasing. The higher thresholds allow for a higher window befor triggering a decrease so higher throughput. Instead of using Additive increase of +1 Additive increase of +2.5 seems to have highest performance in this setting.

P4.

For problem 4 I have implemented a version of MIMD looking at various difference value ranges. The difference is calculated by timestamp when the ack is received to when the ack is sent. Based on the ranges of the difference in packet time stamp values the either MI or MD is performed.