## Title: Additive increase/multiplicative decrease (AIMD) programming project

**Group members:** Gautham Pothana, Jithesh Kumar Kasi

**Description:**

In this programming project we are going to implement the AIMD algorithm for finding optimal congestion window in a fair and efficient manner. In the class, we learned about the AIMD algorithm for 2 users and we are going to implement AIMD for 2 users and try to extend it to N users. We have so far looked into the algorithm and started implementation. AIMD is a feedback control algorithm that is used in the transport layer of the TCP/IP stack by TCP protocols. In the absence of congestion, the AI phase is in charge of the linear expansion of the sender's congestion window. When congestion is detected, the MD phase is responsible for the exponential reduction of the sender's congestion window. There are numerous additive and multiplicative combinations of these two phases. Three additional combinations are AIAD, MIAD, and MIMD. However, they do not ensure fairness. AIMD, on the other hand, ensures fairness when competing for network bandwidth with other TCP flows. When there is no congestion in the network, MIMD increases the congestion window exponentially; as a result, the chances of congestion increase, and a large portion of the bandwidth is consumed by this particular flow, affecting the performance of other flows. Congestion occurs much more frequently as a result of its exponential growth. MIMD and AIMD fail to achieve stability, and all TCP flows fail to converge on using network bandwidth.