**Hypothesis**

When handling a sudden influx of incoming requests, the challenge of handling a DDoS attack depends on the scale and resource power from the hardware of the server and the scale of the attack initiated by an attacker. The server faces a challenge of figuring out how to serve all possible users which becomes increasingly difficult the more complex these requests can become. If a server is unable to mitigate or identify a DDoS attack, my proposed solution would be a divide-and-conquer approach using a round-robin queueing system to stress handle a DDoS attack.

What makes this attack different from popular method of load-balancing is that this method would be using techniques from how Operating Systems handle large amounts of tasks for the CPU. Operating Systems use a preemptive scheduling solution to ensure that every tasks gets their share in the CPU on top of a priority system to make sure the most important tasks gets priority. Using this solution, we can use a scheduling algorithm for our servers so that if a server or router becomes unable to handle the large amounts of traffic, it will divide the work into groups and assign priority based on the types of requests being made. Slow, resource-intensive solutions could be given lower priority so that, while it would be inconvenient, these requests wouldn’t be troublesome for the server. Meanwhile, important tasks, such as security checks and network interfacing from an admin, would be given higher priority, allowing users with special privileges to remain in control of the situation. The system would work under a round robin system so that all users, regardless of priority, could still get a chance for server access rather than be blocked out due to a server experiencing overbearing capacity. While it may not stop a DDoS attack, it would allow the server’s security policy to remain in effect even under heavy stress.