**Introduction**

As our world becomes more interconnected, small businesses, companies, and even certain individuals may opt to provide online services to anyone around the world through the internet. As computer architecture becomes more faster, more accessible, and more powerful, adversarial groups start to have more options to disrupt such activities through DDoS attacks, leading to a virtual arms race between server owners and adversaries to see whose network processing power can outmatch the opponents. While more sophisticated methods of identifying and bypassing DDoS attacks have been theorized and tested, many solutions struggle to the exponentially increasing scale of botnets networks that can overtake even the networks of industrial giants of the industry.

This issue is caused in part because many of these solutions question how to identify and stop an attack rather than how to adapt to such a large-scale attack. Rather than finding out how to defeat a DDoS attack, this research instead aims to find a method to adapt to such an attack such that the core functionality of the defending network can remain in control. The goal of this experiment would determine the possibility whether it could be possible to use the scheduling disciplines of Operating Systems to manage the scale of network queries in such a way that a DDoS attack would not disrupt the server’s control of its network. This research paper will describe the software being used to simulate a network under normal circumstances and a network undergoing a DDoS attack along with the layout of the network itself. Then, the experiment will be conducted showing three experiment trials: One where the network is experiencing normal traffic flow, one where the network is experiencing a DDoS attack, and one where the server is experiencing a DDoS attack but using the described DDoS prevention measures implemented. Data will be gathered and recorded along with an explanation regarding the effectiveness or ineffectiveness of the experiment trials and perceived flaws of the experiment along with improvements that could be utilized to improve the reliability of the experiment.

**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.