

Shreya Srinivas  
Dan Haugh  
Capstone Abstract  
CS176

## **Concurrent Packet Firewall**

For the CS176 capstone project, we built a prototype of a firewall which processed packets serially and concurrently. Packets were generated using an external class and they contained specific information, such as permissions for source and destination addresses. Our program modelled the operations of a real firewall which uses these payloads to filter packets through an access control pipeline. In addition, we collected statistics in a histogram that measured how packets were being delivered.

We had to consider how the distribution of packets would be handled through the serial and concurrent versions of the firewall. For the serial version, packets were dispatched through a single worker thread. We used several threads to dispatch the packets which were subsequently processed in a concurrent queue and sent to their respective destinations. In addition to this pipeline, we had to solve a number of issues regarding deadlock and ensure that packets were being sent and received in a linearizable manner. We created an array of read write locks for each source address and if a thread fails to acquire a lock for this particular packet, it would continue to process another packet. Several of the concurrent data structures we used were available through Java's concurrency package which uses wait free algorithms as described in "Simple, Fast, and Practical Non-Blocking and Blocking Concurrent Queue Algorithms" by Maged M. Michael and Michael L. Scott.

