Cloud applications have varying demands for network bandwidth. These variations can be driven by user demand, such as a sudden surge in interest in a particular service, or by system events, such as the emergence or destruction of a new network device, like a router or an end-host. Because the Internet is massive, and the demand grows everyday, we need our system to be scalable. Furthermore, there are malicious users who may attempt to subvert our system by using more bandwidth than what would be a fair allocation based on the existing demand on the networking system. Thus, there are many factors deciding how much bandwidth different nodes get in a network. However, maintaining Quality of Service (QoS) by guaranteeing a certain amount of bandwidth to end-hosts becomes an interesting problem and important business concern for many companies.

Our goal is to provide an efficient solution to finding optimal bandwidth guarantees in a dynamic network environment to provide QoS to network users. We want to allow cloud applications to specify their bandwidth requirements, and we want our system to compute, for each application, how much bandwidth we can guarantee and rate-limit the end-hosts. In addition, as the demand and the capacity of the network changes, we want our system to respond to those changes in real time and update the QoS conditions.

In a recent paper by Databricks, network architects proposed a system called CloudMirror\(^1\), an efficient solution to guarantee QoS in cloud computing environments. Unlike other virtual network abstractions, CloudMirror takes into account actual usage of network

\(^1\) [https://www.usenix.org/system/files/conference/hotcloud13/hotcloud13-lee_0.pdf](https://www.usenix.org/system/files/conference/hotcloud13/hotcloud13-lee_0.pdf)
applications to construct a Tenant Application Graph (TAG) that helps it compute bandwidth guarantees. The authors propose that the TAG abstraction is superior to assigning bandwidth guarantees purely by looking at the physical distribution of network nodes. For our senior project, we would like to use the learnings from the CloudMirror paper as a foundation to build a system that can efficiently provide QoS in the cloud.

After doing some initial research, our high-level architecture is as follows: we will implement virtualized end-hosts (using hypervisor or other virtualization software), virtualized switches/routers, and a centralized server that monitors the network usage and performs scheduling. This is a 3-component system. Many modern systems use the hybrid model of performing scheduling and computation in both end-hosts and switches, so we use that design pattern here. As a distributed system, the end-hosts, switches, and central server must communicate to compute bandwidth guarantees for the whole system. In addition, we will make an improvement to the system in the CloudMirror paper to add something new as part of our project.

This project will be done in collaboration with Eric Yu. To fulfill the requirements of CPSC 490, I will specify my own responsibilities as follows. I will be responsible for architecting and implementing the switches/routers and the end-hosts so that they will integrate with the overall system that Eric and I will be building. In addition, I will oversee the integration of CloudMirror into our existing environment so as to allow benchmarking.

My Deliverables

1. I will write a detailed document describing the design and architecture of the routers/switches and the end-hosts and how they fit into the architecture of our overall system.
2. I will implement my design of the routers/switches and end-hosts in a simulated environment as a proof of concept.

3. I will integrate CloudMirror into our simulated environment so that both systems are deployed in the same simulated environment.

4. I will write up a report detailing important design decisions, features, and how they relate to the benchmarking results.

5. I will write an overall report describing how my designs and implementations fit into the overall system.