CSC360
Assignment 2 Design Document

1 Threads

The program will use 1 thread per simulated flow. Each of these threads will wait their arrival times then register themselves and wait to be told they are next. Then they will wait the transition time and return.

These threads will be managed by the main thread after it creates all of them.

2 Mutexes

Four mutexes will be used:

1. A mutex for the flow data.
2. A mutex for starting the transmission
3. A mutex for the transmission itself
4. A mutex for a counter of how many threads are running

3 Data

Each flow will be represented by a strict with priority, transition time, arrival time, and ID. These structs will be passed on to the thread for that flow.

They will also be pushed and popped from a priority queue based on the criteria given.

This queue will be protected by a mutex so it won't be modified concurrently.

4 Condition variables

The first three mutexes mentions all have convars associated with them.

1. This variable represents the condition that a flow has been added to the queue. Once it's unblocked, the main thread will pop from the queue and continue.

2. This variable represents the condition that a flow has been selected for transmission. It is broadcast to all the threads, which will then check if the selected id matches it's own. If so, it will continue to transmission, otherwise it will keep waiting.

3. This variable represents the condition that a thread has stopped transmission and that the main thread can trigger the next transmission if there is one in the queue.

5 Algorithm

```
read file
init mutexes
init thread for each flow

in the flow thread:
  wait transmission time
  insert flow data into priority queue protected by mutex lock
  signal the queue convar
  wait on the start convar until the start id matches own id
  lock the transmission convar
  wait transmission time and lock
  decrement mutex-protected thread counter


in the main thread:
  while thread count is > 0:
    wait on the queue convar
    pop top of stack and set start id to it's id
    broadcast start convar
    wait on tranmission convar
```