1. I will use one main thread and one thread for each flow. The main thread will time and create the flow threads, the flow threads will send the flows.
2. There is no overall controller thread, the threads interact via the pthread_cond_broadcast(&trans_cvar); command.
3. I will use one mutex. It will only allow one thread to send its flow at once.
4.The main thread will be keeping track of the time since the program started.
5. I will represent flows with a struct that includes the flows information.
6. Because of the mutex, only one thread should be modifying data at once.
7. I will be using one convar.
    a) The convar represents a flow finishing its transmission, which signals the transmission of the next flow.
    b) My mutex that allows one flow at a time is related to this convar. Once the current flow is finished the mutex should be unlocked to allow the next flow to start.
    c) The Mutex should be unlocked.

8.

```
import flows from file.
create thread for each flow.
When flow arrives, check if it can be sent:
lock mutex
add flow to queue
sort queue based on rules
wait for current transmission to finish with convar
send next flow from top of queue
when flow finishes remove from queue and unlock mutex
broadcast convar so next transmission knows it can go
wait for all threads to finish
exit program
```