**Design Protocol**

Computer A will have a bucket. A will process a packet and place it in the bucket. If the bucket already has a packet holding in there, A will wait to process a packet. A will send its packet to B. B has a queue for incoming packets.

When A pushes a packet, it will keep mark of the time and the size of the queue. Then when A pushes another packet, it will take note of *that* time and the *current* queue size. It will calculate the speed of the packet before the current one and save it as a “timer”.

Here’s where the timer comes into play. When A pushes a packet into B’s queue and the queue is then full, A will wait the current “timer” amount before attempting to push the packet. Hopefully the current speed of the queue being processed is consistent with the current time calculated.

This should minimize the packet loss. A will also not continuously be checking B for if it’s ready. A will get its information when it pushes the packet, thus utilizing each push.

