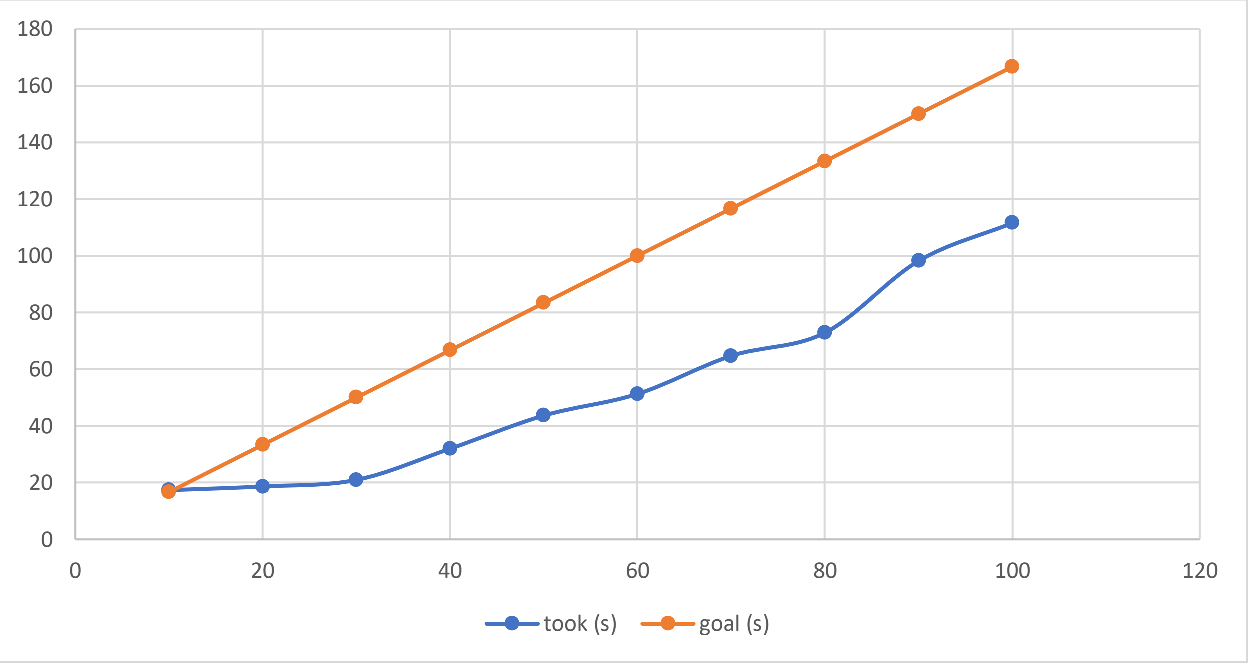
My Mac protocol uses a CSMA/CA protocol. Stations coordinate in an effort to *avoid* rather than resolve collisions by screening to see if their channel is active and using a randomized timer before screening again. The channel that each station uses for a particular packet is randomized when the packet is ready to be sent (in an attempt to evenly spread packets among channels). The TX power is set at less than half capacity when packets are first sent, then slowly incremented if the packets fail to be ACK'd. I randomly select one of 8 intervals (with size 20 ms), and check the channel again when that moment comes. The number of intervals doubles each time this process occurs. If a packet is delayed like this 10 times, it is not sent again.

I ran the project 10 times with the following inputs:

```
# ./project.py <# stations> 12 200 YourMac
```

I logged and graphed the resulting times. The orange line represents the expected output and the blue line represents my output.


took (s)
goal (s)