

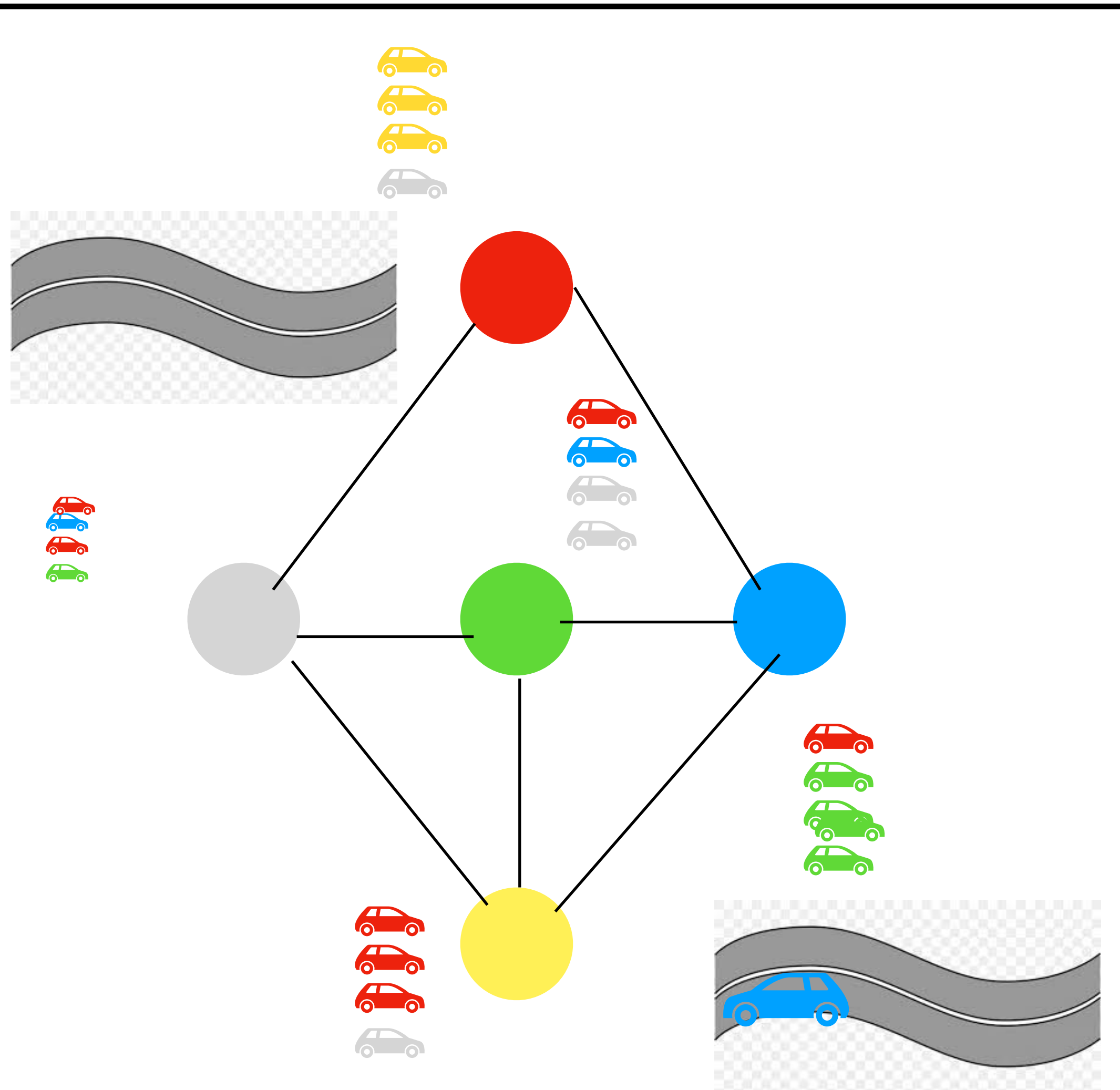
Traffic Generator

Components:

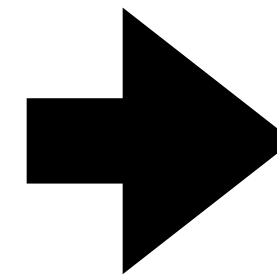
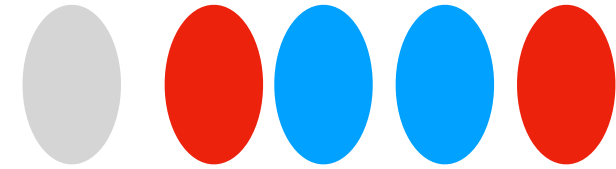
- 5 Waiting Queues for each node
- Waiting Queues have different rates to add Packets to Node Queues
- Added at the bottom of the queue
- Dropped packet when Node Queues are full

Putting in all together

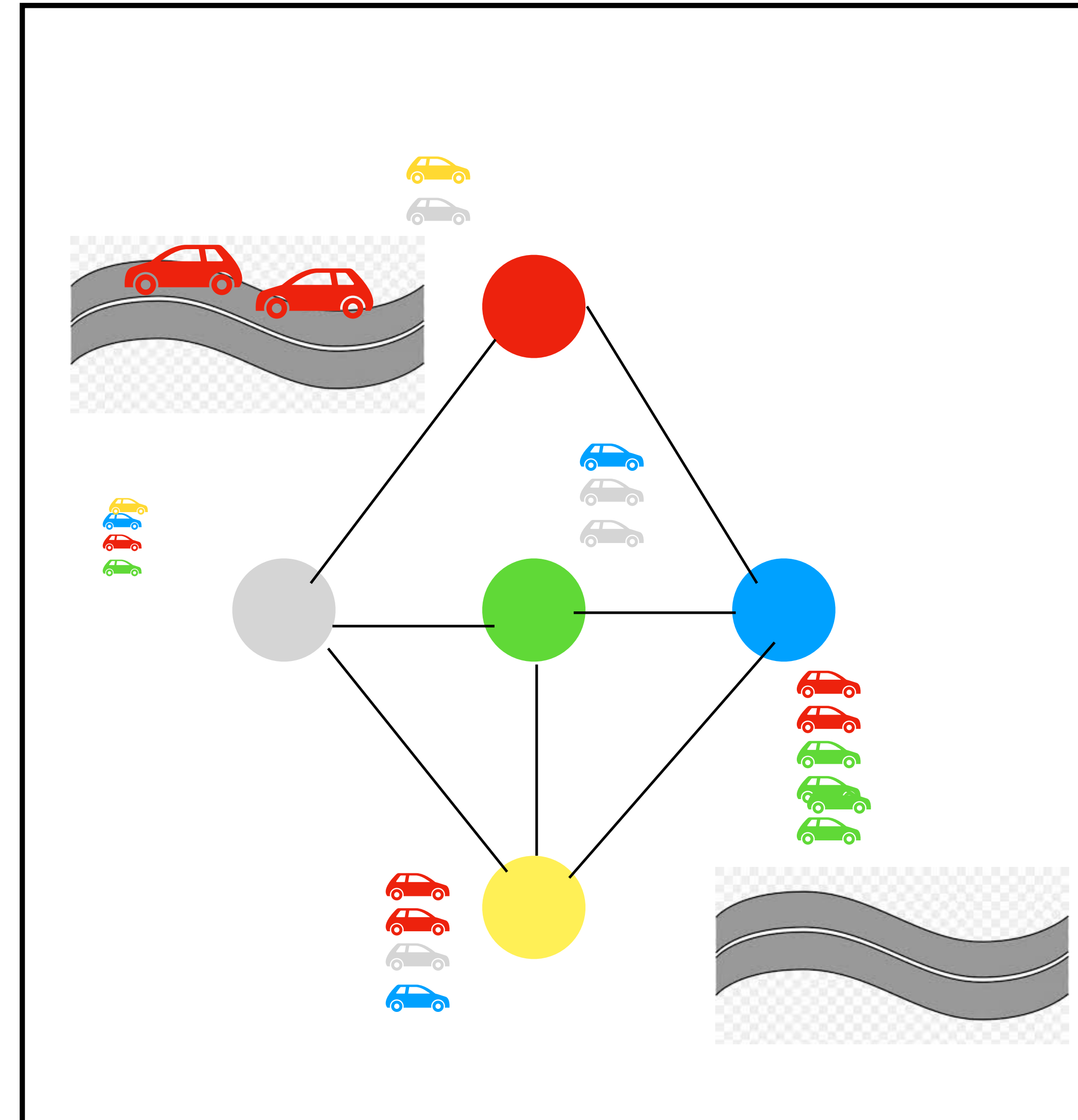
Old State:



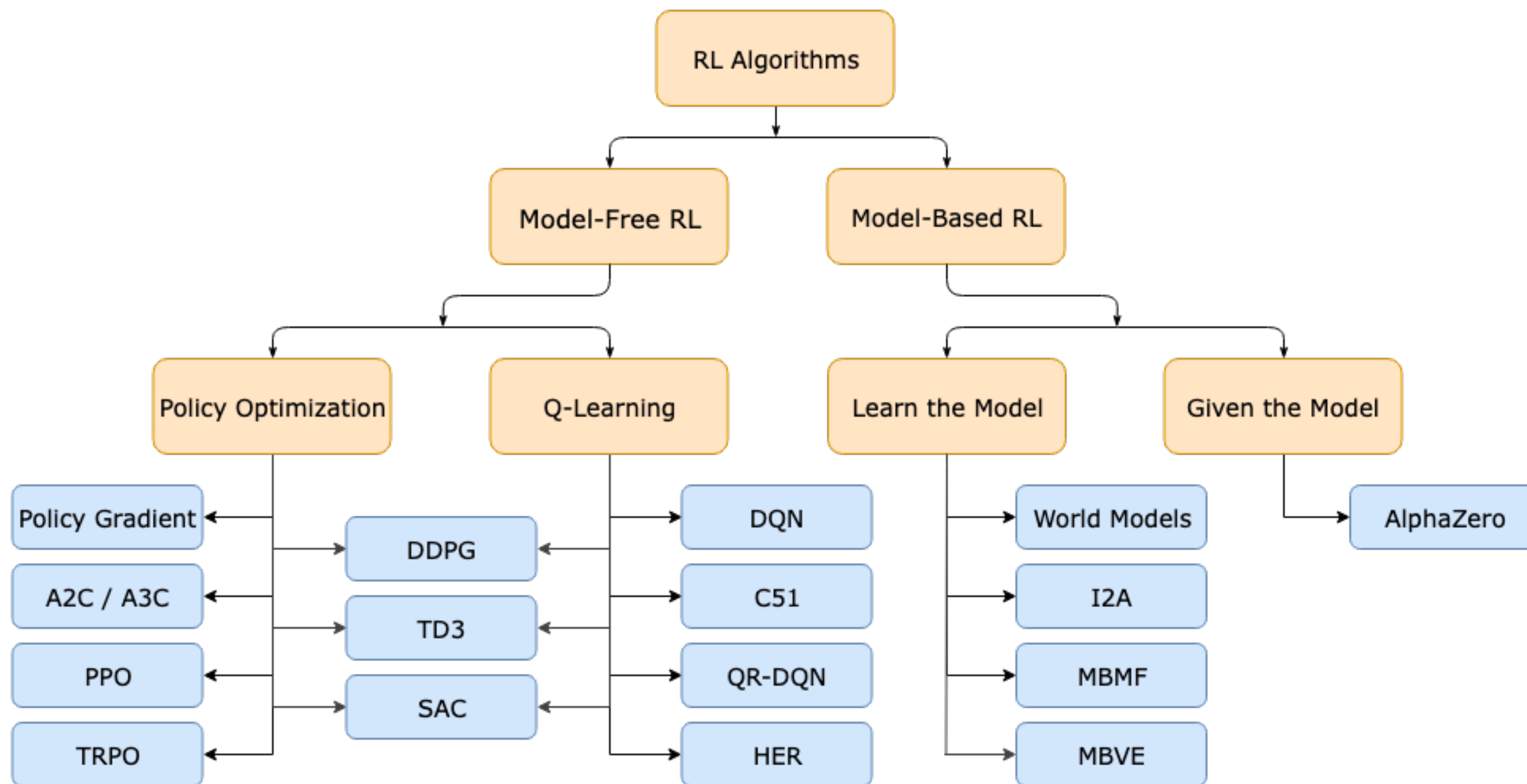
ACTION:



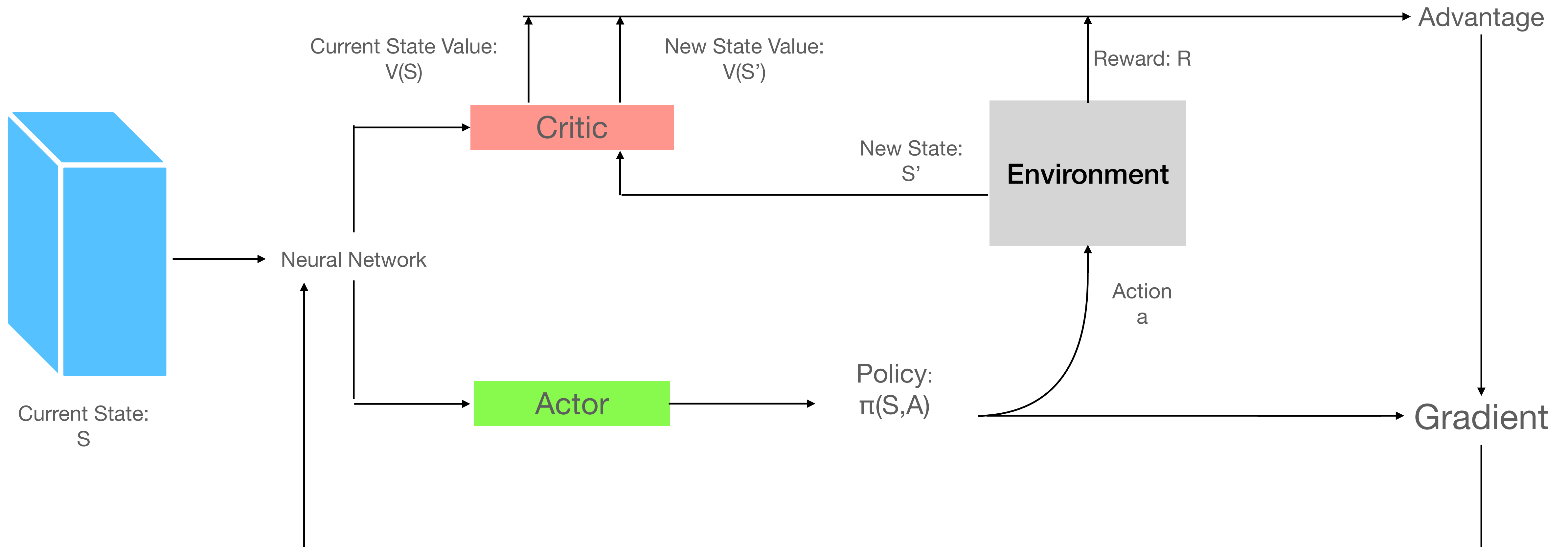
New State:



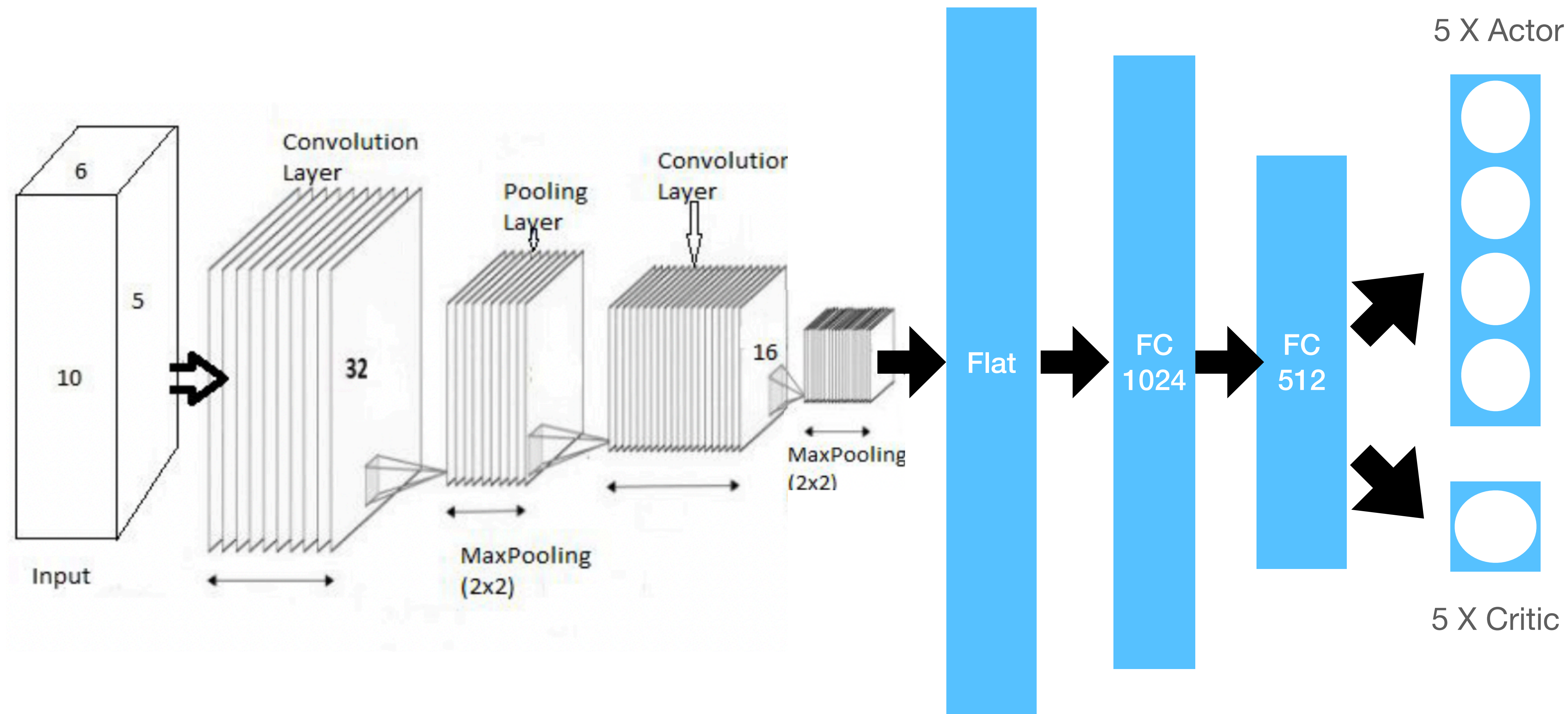
The Agent



The Actor-Critic algorithm

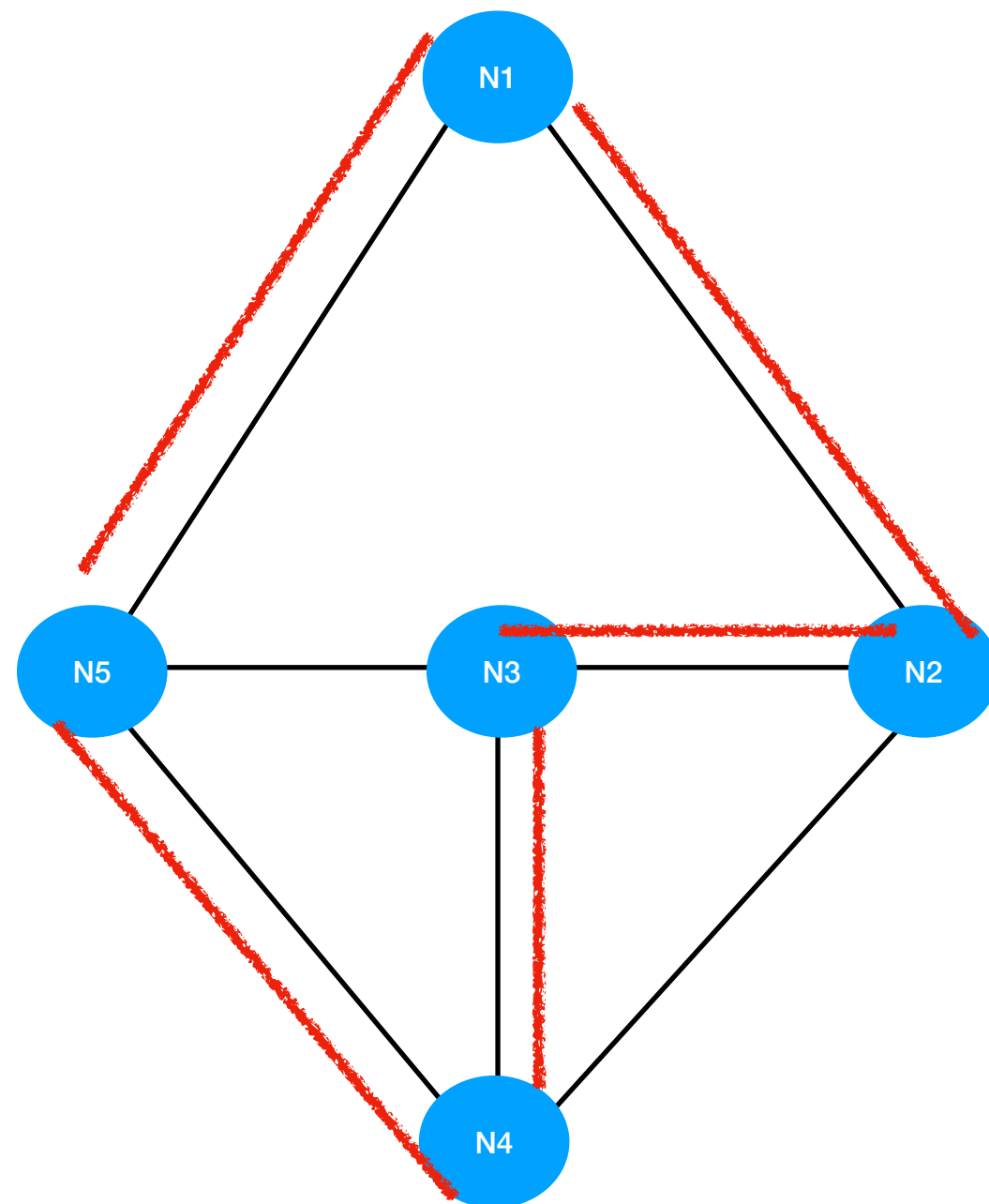


The Neural Network

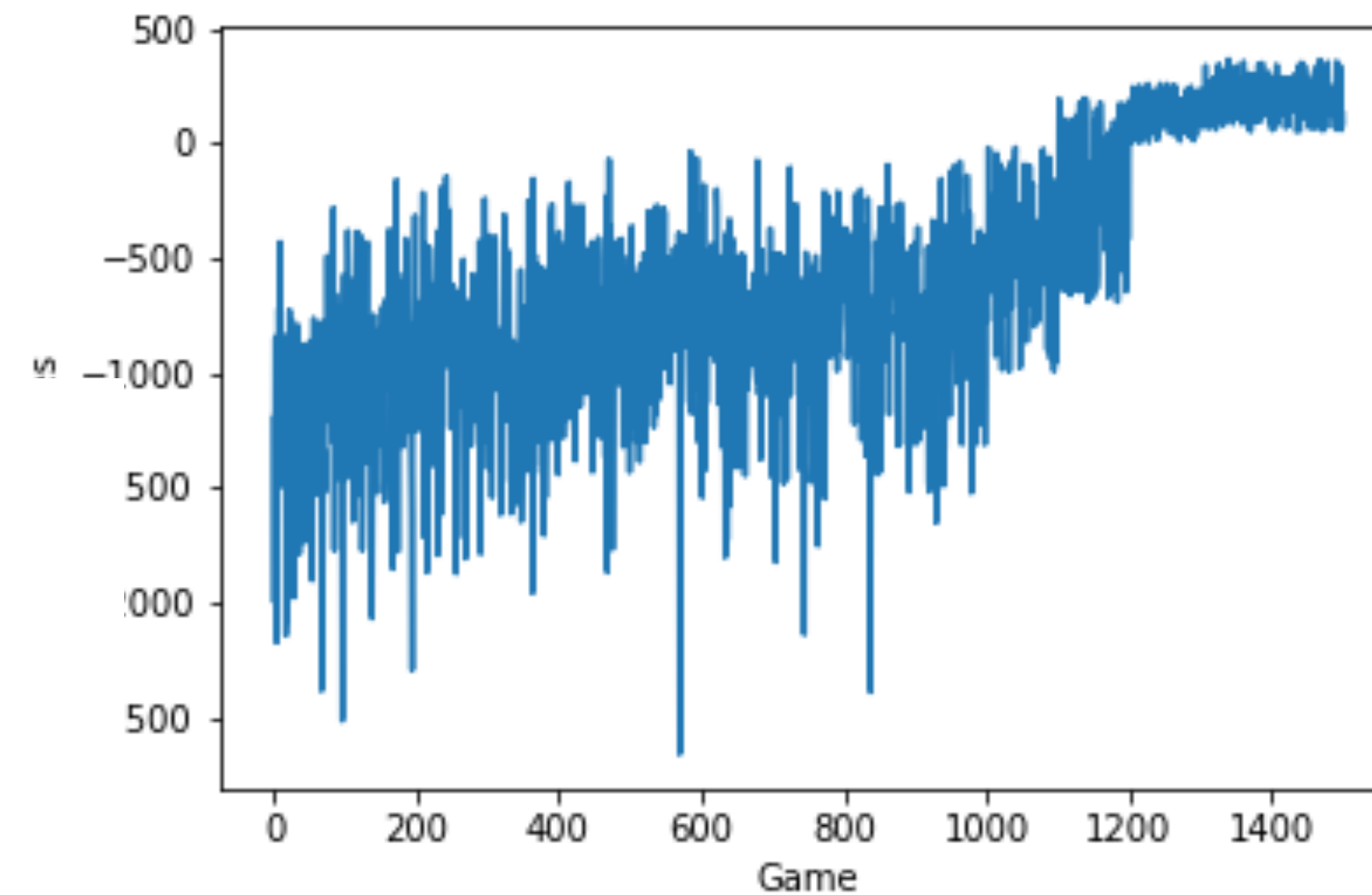


Training Results

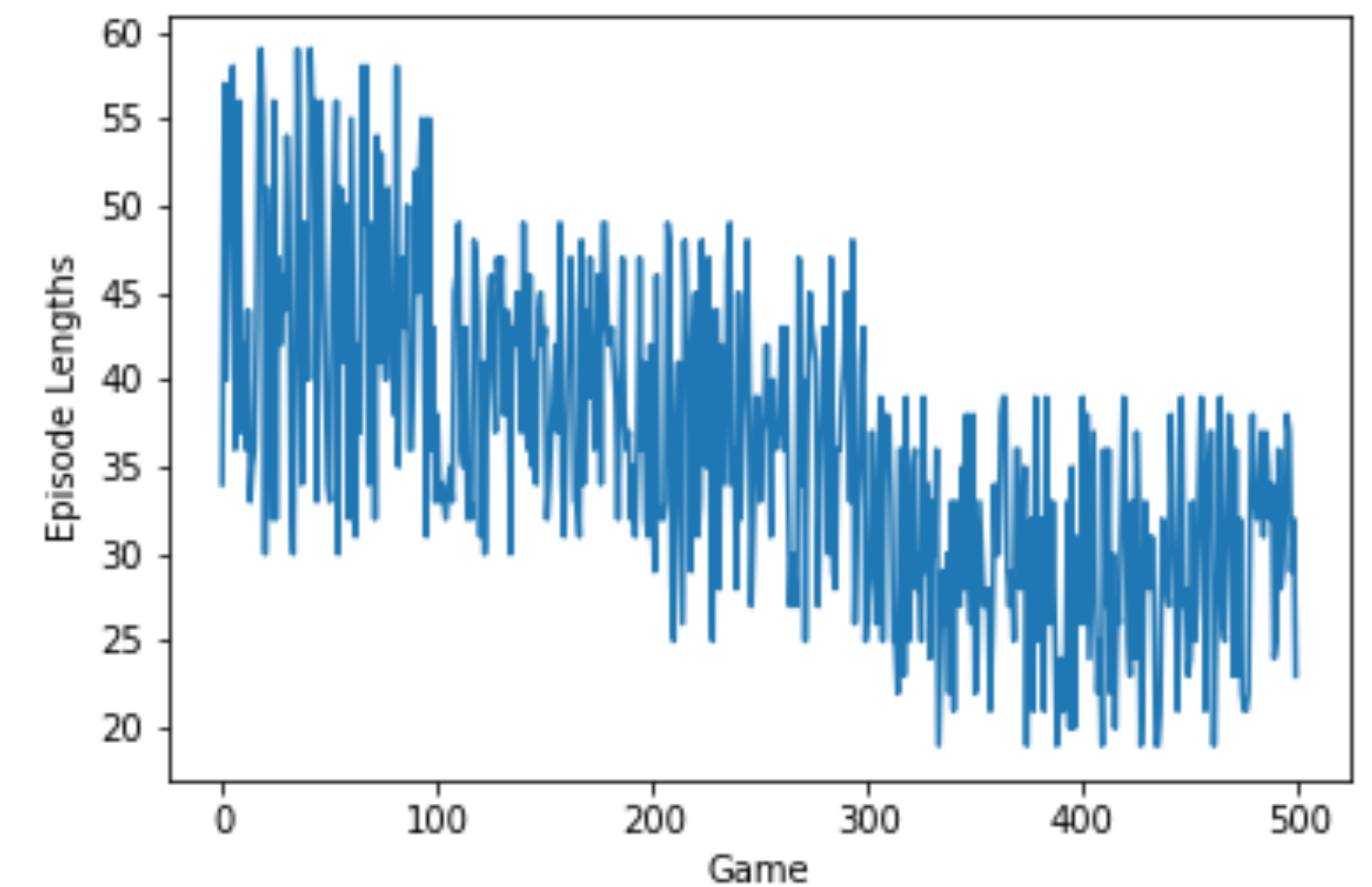
First Model: Constant routing penalty



Score Per Episode



Runtime Per Episode

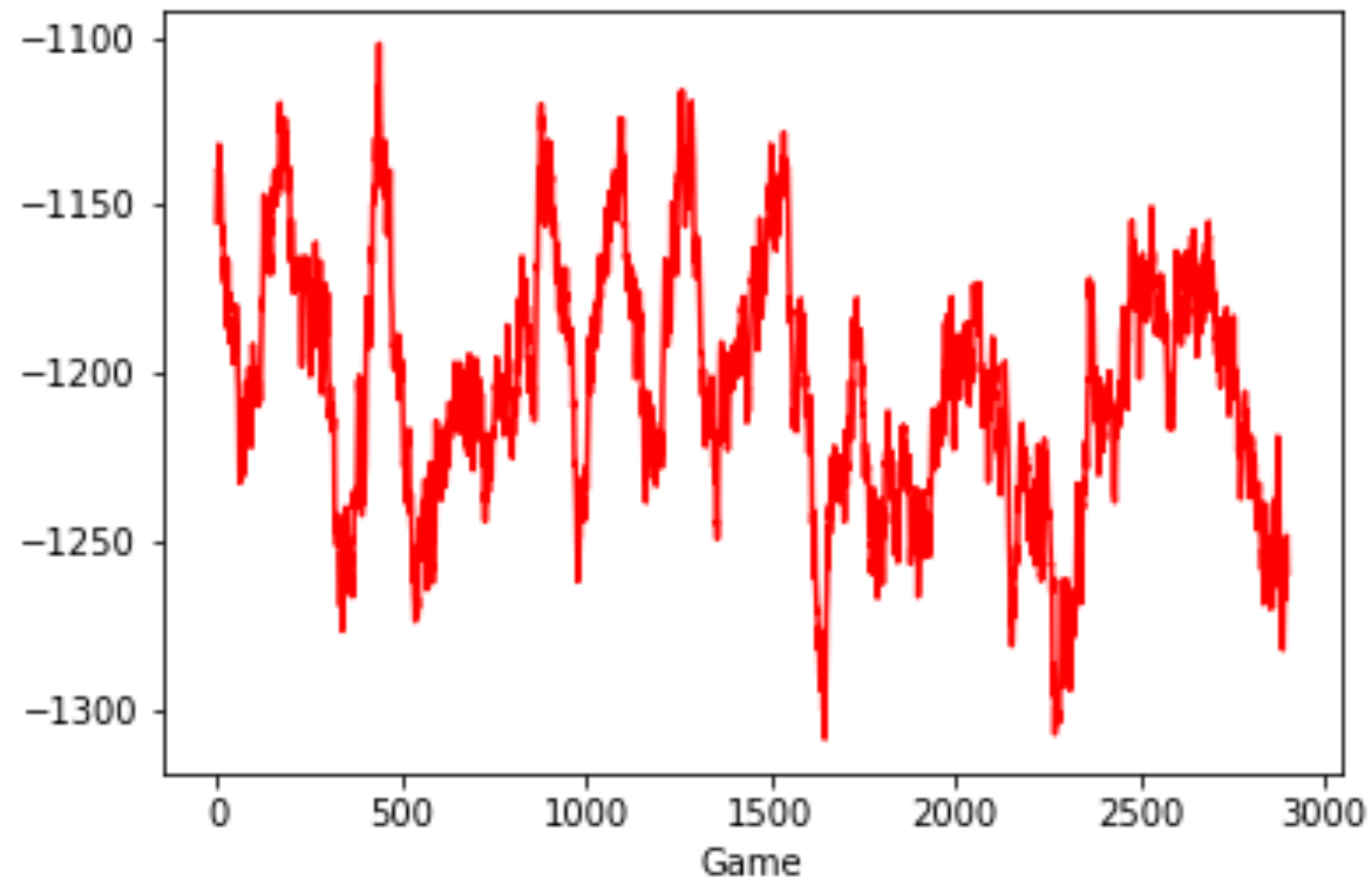


After Running 1200 episodes,
Agent policy stuck to one move
that loops over network and successfully
but not optimally delivers each packet.
During at least 2 training sessions.
Loop from action **[1,2,2,1,1]**, for example
is depicted in red on network

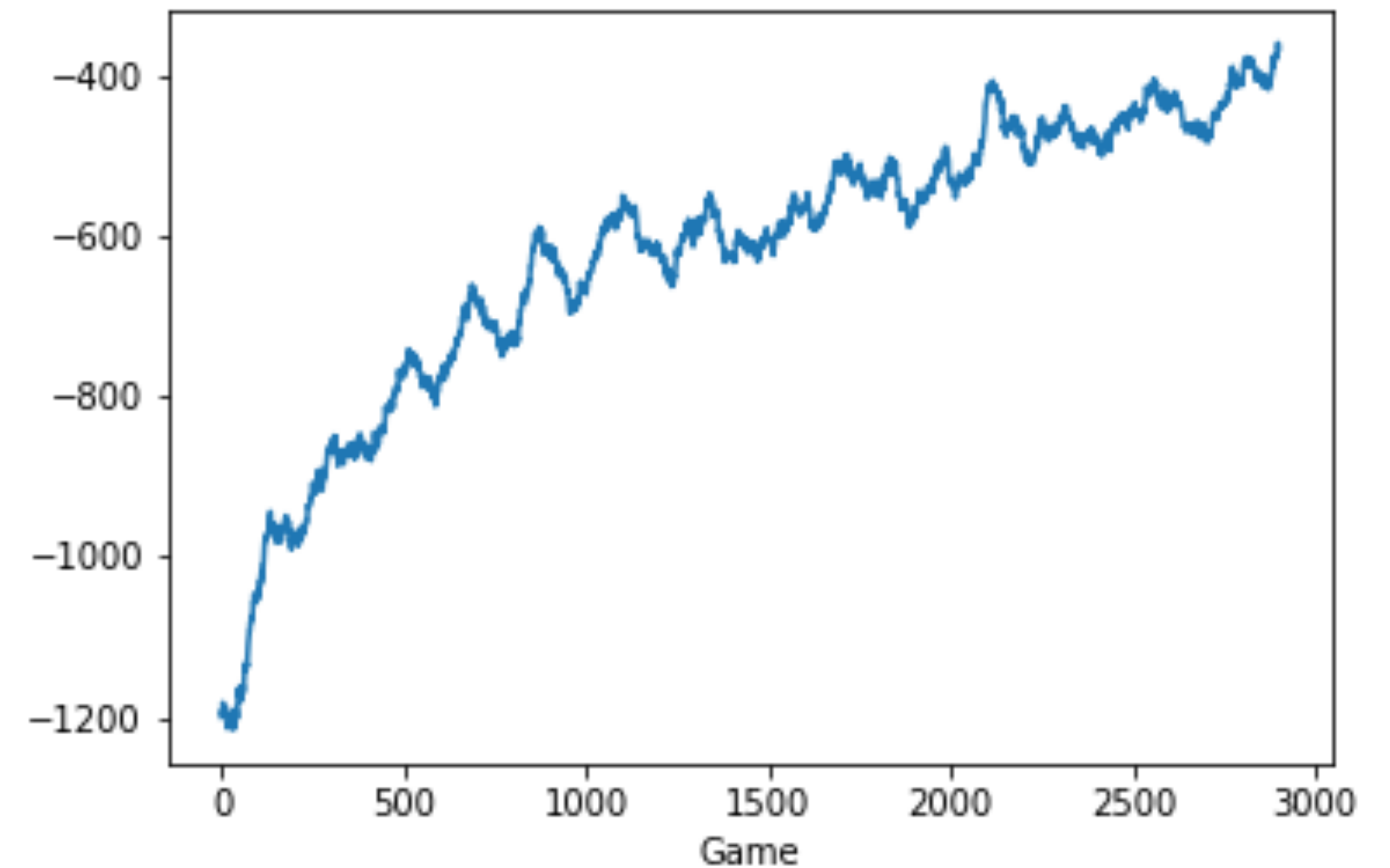
Second Model Scores

Second Model : Reward system now with routing penalties based on queue occupancy to favor busy/ congested nodes

Average score for 100 game for **Random baseline** VS our **Agent**, after training on 3000 Games.



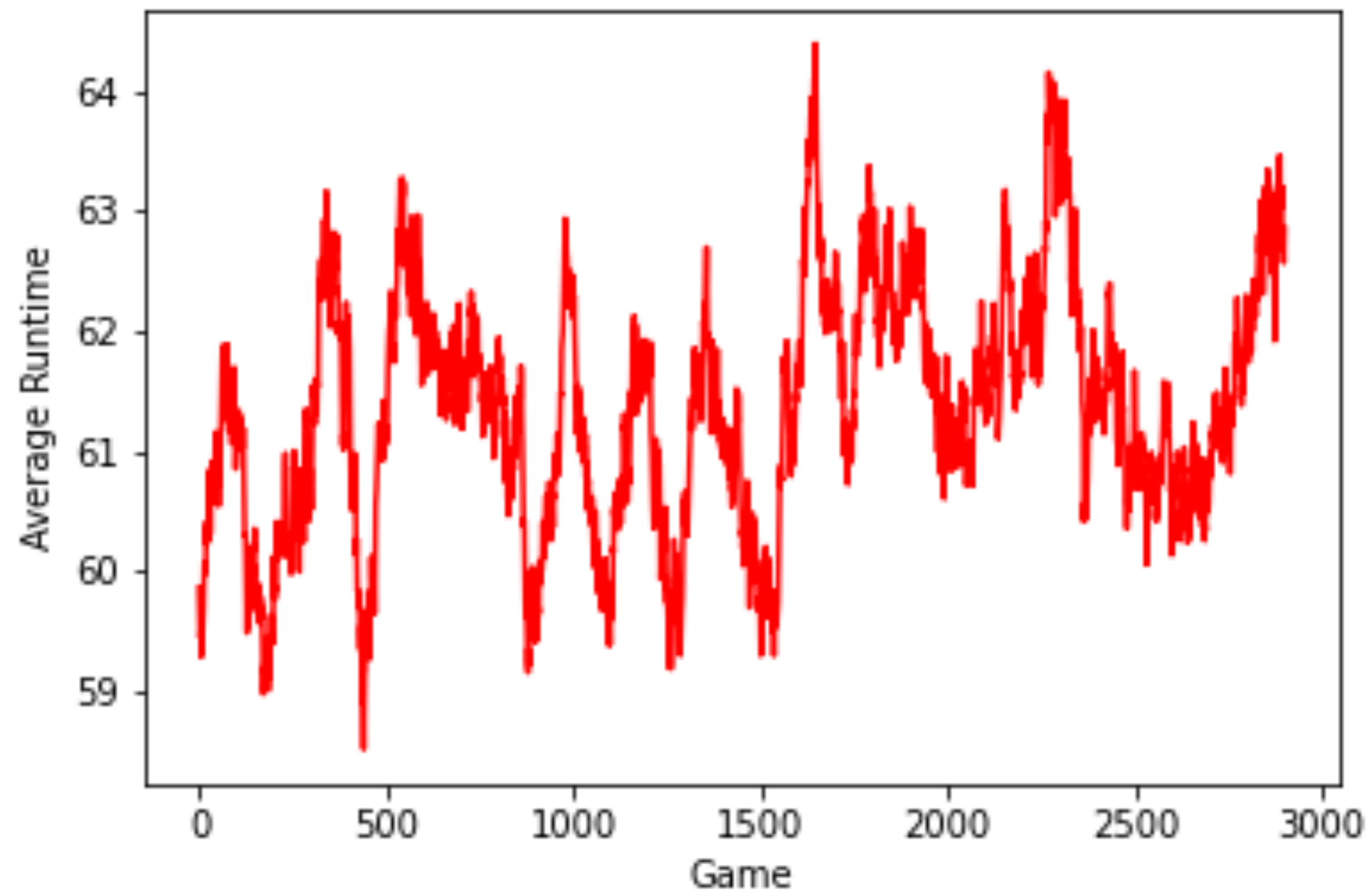
Average Score Per 100 Game Random baseline



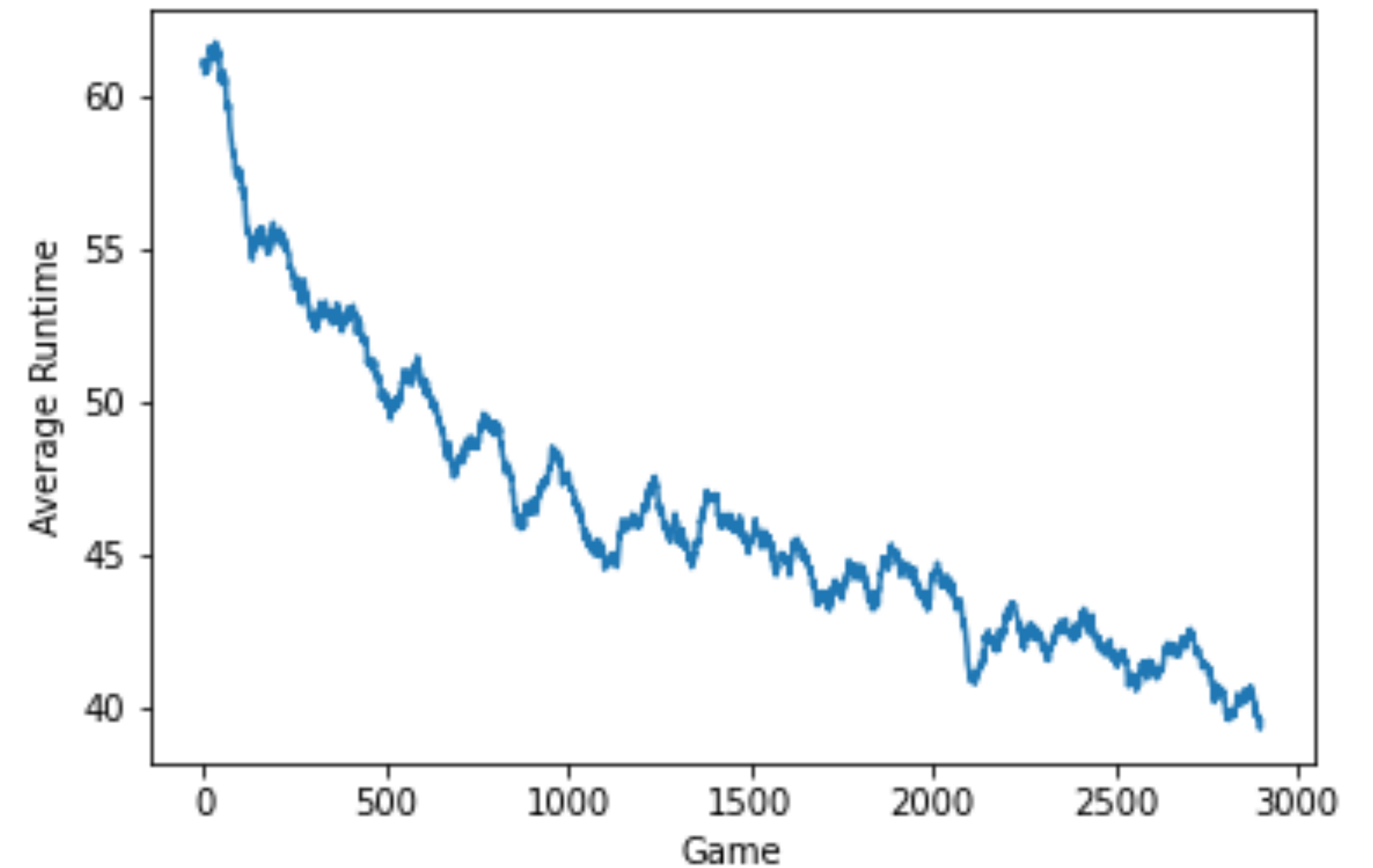
Average Score Per 100 Game Our Agent

Second Model Runtime

Average runtime for 100 game for **Random baseline** VS our **Agent**, after training on 3000 Games



Average Runtime Per 100 Game Random baseline



Average Runtime Per 100 Game our Agent

What is next?

- Training, training and training
- Plugging in traffic generation
- Test for latency, dropped packets and order
- Increase Queue and network size
- Using the CORE Emulator

Thank you! :)