ECE697AA Project

Question 1

$\begin{cases} e_c = 1 : \text{new slice request from class } c \text{ arrives} \\ e_c = -1 : \text{slice's response are being released to the system's resource} \\ e_c = 0 : \text{otherwise} \end{cases}$

following state s' is in S

$$r(s, a_s) = \begin{cases} r_c & \text{if } e_c = 1,\ a_s = 1 \text{ and } s' \in S \\ 0 & \text{otherwise} \end{cases}$$

$\begin{cases} a_s = 1 & \text{if arrival slice is accepted} \\ a_s = 0 & \text{otherwise} \end{cases}$

- Formulate network provider's resource allocation problem as a centralized optimization problem

→ Goal: reward maximization

$$\max_{n_c} \left[r(s, a_s) = \begin{cases} r_c & , \text{if } e_c = 1, a_c = 1, s' \in S \\ 0 & , \text{otherwise} \end{cases} \right] \quad \text{s.t.} \quad \Theta, \Omega, \Delta$$

with:

$$\Theta = \Theta_{cloud} + \Theta_{fog} \geq \sum_{c=1}^{C} r_c^{e} n_c \quad \text{and}$$

$$\Delta = \Delta_{cloud} + \Delta_{fog} \geq \ldots \quad \text{and}$$

$$\Omega = \Omega_{cloud} + \Omega_{fog} \geq \ldots$$

How to solve optimization problem 2

three classes of slices

class - 1
class - 2
class - 3

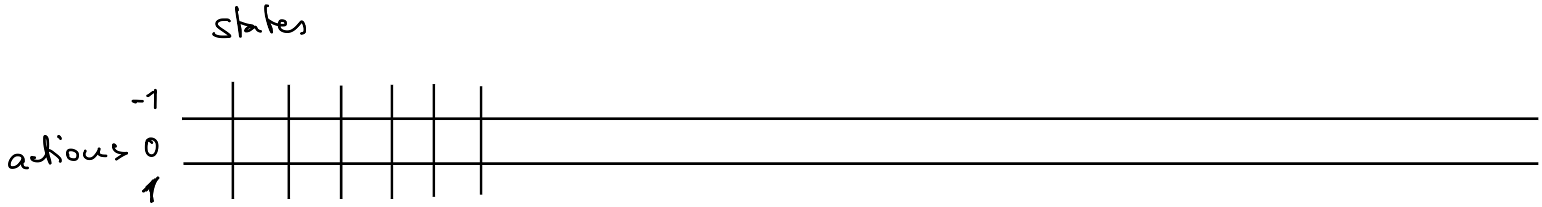
⇒ Each slice includes a number of InP and fog resources.
The number of fog resources for each slice of class c is given
⇒ Choose number that makes sense (not too high bc delay)

Q-learning

Q-table : actions $= \{-1, 0, 1\}$ $\Rightarrow$ num_actions $= 3$

states : 3-classes, each can be „at every position" of the state-vector of maximum length C $\Rightarrow$ num_c $= \sum_{i=1}^{C} 3i + 1$ ← for state with no class at all

$\Rightarrow$ Q-table of size $3 \times \left[\sum_{i=1}^{C} 3i + 1\right]$ with C dependent on resource requirements as well as the maximum resources



SMDP:

- Decision epoch t_i varies depending on how far requests are apart from each other which is drawn from Poisson distribution
- Since we have 3 classes, we always have
  $\mathcal{S} = \{n_0, n_1, n_2\}$
  depending on how many requests of class-1, -2 or -3 n_0, n_1 and n_2 are set, respectively.
  Let's assume every fog can have a max. # of allocated requests n_{max}
  $\Rightarrow$ # of possible states $= n_{max}^{(n_{max}^{n_{max}})}$
- Action space is pretty self-explainable
  $\mathcal{A} = \{0, 1\}$
  With $\dim(\mathcal{S}) = [n_{max}^{(n_{max}^{n_{max}})}, 1]$ and $\dim(\mathcal{A}) = [2, 1]$, we have $\dim(\text{Q-table}) = [\dim(\mathcal{S})[0], \dim(\mathcal{A})[0]] = [n_{max}^{(n_{max}^{n_{max}})}, 2]$
- State transition probability

I think this also means, that we have the prob.

$$\mathcal{L} = \begin{cases} u_c \frac{\mu_c}{z} & \text{for } e \text{, if } e_c = 1, a = 1 \\ 1 - \frac{z_s}{z} & \text{for } e \text{, if } e = 0, a = 1 \\ \lambda_c / z & \text{for } e \text{, if } e_c = -1, a = 1 \end{cases}$$

$\Rightarrow$ $1 - u_c \frac{\mu_c}{z}$ if $e_c = 1$ to have $a = 0$

$\Rightarrow$ see before: $\frac{z_s}{z}$ if $e_c = 0$ and $a = 0$

$\Rightarrow$ I'm not sure here, bc. in my opinion it would make sense to accept every completion process to make space for new slices which generate more reward, so I'm not sure if we just need to set $\lambda_c = z$

- Reward function like explained

$$r(s, a_s) = \begin{cases} r_c & \text{if } e_c = 1, a_c = 1, s' \in \mathcal{G} \\ 0 & \text{otherwise} \end{cases}$$