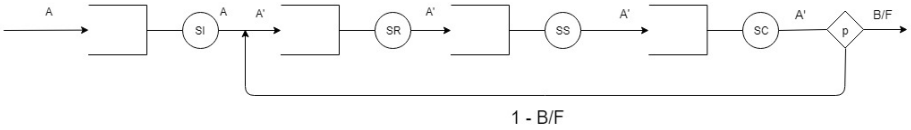
Contents

1. Exercice 1



C'est le schéma du réseau de filé d'attente.

1.1 Initialiser les valeurs initiales

```
self.A           = a
self.F           = 0.0422
self.B           = 0.0152
self.I           = 1
self.Y           = 0
self.R           = 10 * AMELIORE_VITESSE_SERVEUR
self.S           = 1.5 * AMELIORE_BANDE_PASSANTE
self.C           = 0.707
self.cooldown    = cooldown_time
self.warmup      = warmup_time
```

Fait attention: Tous des valeurs doivent être même unités. Moi, j'utilise le *Mbits* avec des tailles du tampons et le *Mbis/s* avec les vitesses du serveur.

1.2 Modéliser le réseau

```
self.N = ciw.create_network(
Arrival_distributions = [['Exponential', 1/self.A],
                         'NoArrivals',
                         'NoArrivals',
                         'NoArrivals'],
Service_distributions=
    [['Exponential', 1/self.I],
```

```
        ['Exponential', ( 1 / ( self.Y + (self.B/self.R)))],
        ['Deterministic', self.S/ self.B],
        ['Deterministic', self.C/ self.B]],
    Transition_matrices = [
                        [0.0, 1.0, 0.0, 0.0],
                        [0.0, 0.0, 1.0, 0.0],
                        [0.0, 0.0, 0.0, 1.0],
                        [0.0, 1 - self.B/self.F, 0.0, 0.0]
                        ],
    Number_of_servers = [1, 1 * AMELIORE_NOMBRE_SERVEUR , 1, 1]
            )
```

J'analyse chaque partie pourquoi j'ai déjà faire comme au-dessus:

1. ***Arrival_distributions***

```
Arrival_distributions = [['Exponential', 1/self.A],
            'NoArrivals',
            'NoArrivals',
            'NoArrivals']
```

Grâce au schéma au-dessus, on a tout seul arrivé. En plus, si le délai entre les arrivées de clients (défini comme inter-arrival temps) *(1 / A)* est aléatoire et imprévisible, les arrivées présentent une distribution exponentielle.

2. ***Service_distributions***

```
Service_distributions=
       [['Exponential', 1/self.I],
       ['Exponential', ( 1 / ( self.Y + (self.B/self.R)))],
       ['Deterministic', self.S/ self.B],
       ['Deterministic', self.C/ self.B]]
```

Selon l'énoncé, dans ce modèle, nous ignorons les détails de bas niveau des protocoles HTTP GET. Donc, on suppose que les temps de services sont distribués de manière exponentielle. Bien que cela puisse ne pas être vrai pour certains sites Web sites, cette hypothèse n'est pas vrai avec tous des cas, ces valeurs basées sur approximations conservatrices représentent une limite supérieure sur la vraies valeurs (il faut tester). De plus, étant donné la taille fixe des tampons, les tarifs du service à les nœuds SS et SC ne sont probablement pas exponentiels. Dans l'énoncé, on a testé cela après avoir visité 1000 page web.

- On a représenté la moyenne temps nécessaire pour effectuer une simulation pour chaque client arrivé. Donc le taux de service du SI est *1 / I*.
- Le taux de service du noeud SR est *1 / [Y +(B / R)]*.
- Le taux de service du noeud SS est *S/B*
- Le taux de service du noeud SC est *C/B*

3. ***Transition_matrices***

```
Transition_matrices = [
                      [0.0, 1.0, 0.0, 0.0],
                      [0.0, 0.0, 1.0, 0.0],
                      [0.0, 0.0, 0.0, 1.0],
                      [0.0, 1 - self.B/self.F, 0.0, 0.0]
                      ],
```

4. Remarque

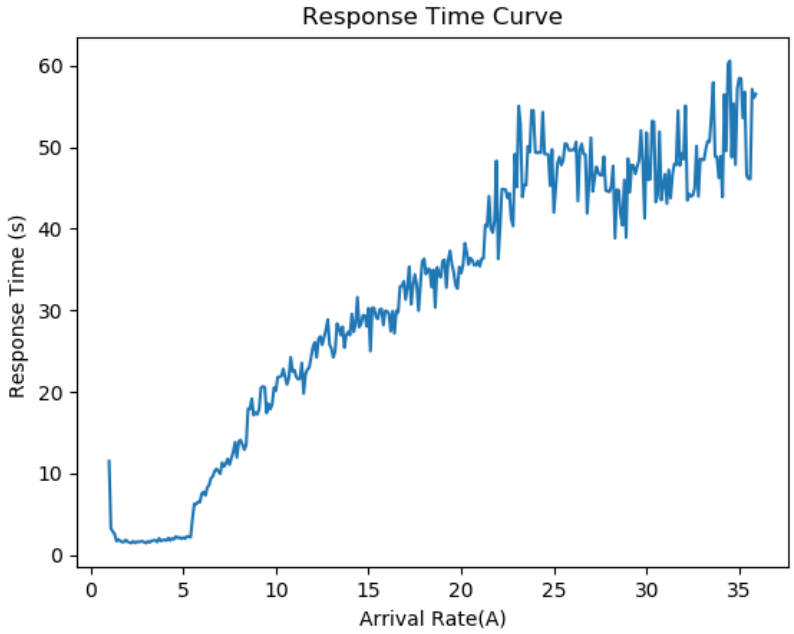
- On modélise un serveur WEB qui est une système 24/7. Quand ce serveur se démarche, il exist chaque records initiales qui provoquent un biais négatif. Donc, on récupère des information lorque le serveur est stable. Donc, il faut avoir le *WARM-UP*.
- De plus, quand on récupère, on a besoin de récupérer tous des 2 clients qui sont finit et qui sont entrain de traiter. Donc, il faut appliquer le *COOL-DOWN*.

2. Exercice 2

Avant calculer, il faut assumer des parametres comme ca:

```
AMELIORE_VITESSE_SERVEUR = 1
AMELIORE_BANDE_PASSANTE = 1
AMELIORE_NOMBRE_SERVEUR = 1
self.A           = a
self.F           = 0.0422
self.B           = 0.0152
self.I           = 1
self.Y           = 0
self.R           = 10 * AMELIORE_VITESSE_SERVEUR
self.S           = 1.5 * AMELIORE_BANDE_PASSANTE
self.C           = 0.707
```

Resultat:




```
============INIT================
==== MOYEN TEMPS DE REPONSE ====
moyen temps de reponse:
31.59798205865761
```

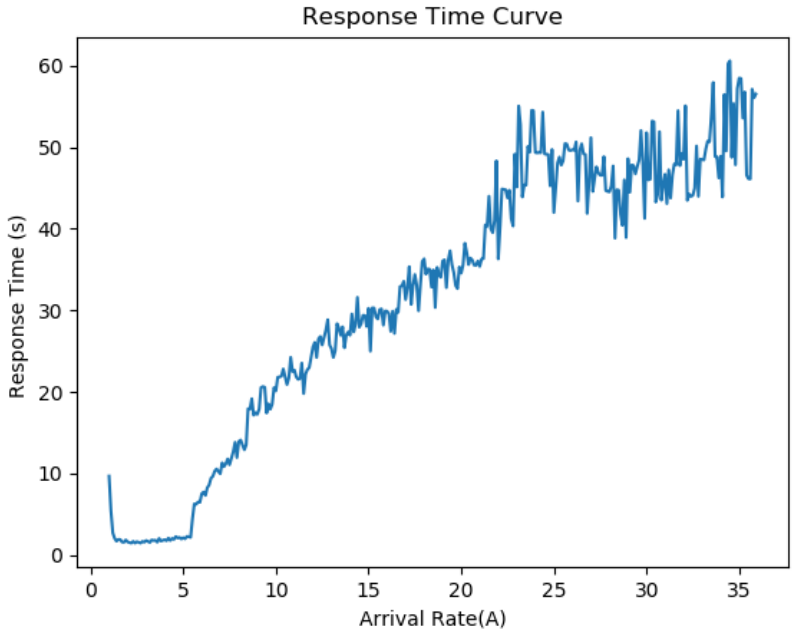
3. Exercice 3

3.1 Case 1: Double la vitesses du serveur(R)

Il faut assumer des parametres comme ca:

```
AMELIORE_VITESSE_SERVEUR = 2
AMELIORE_BANDE_PASSANTE = 1
AMELIORE_NOMBRE_SERVEUR = 1
```

Resultat:



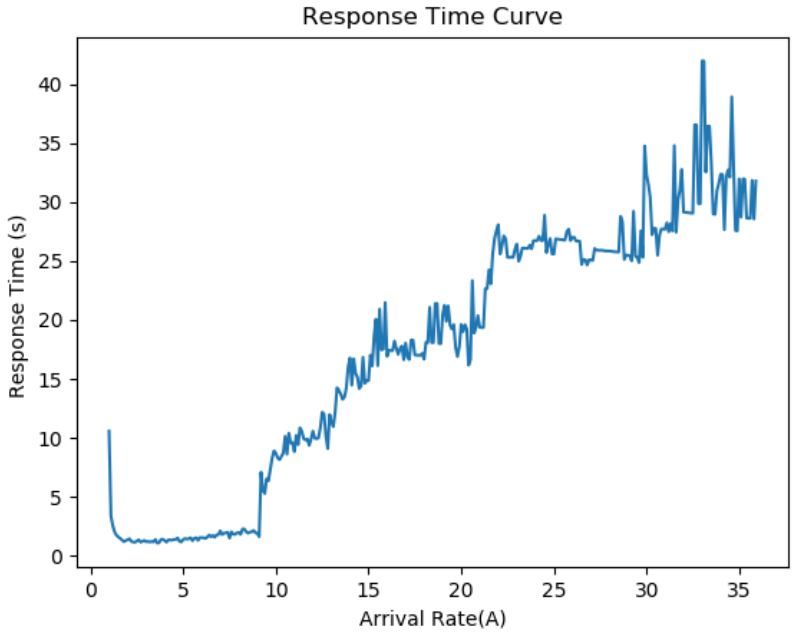

```
==== MOYEN TEMPS DE REPONSE ====
moyen temps de reponse:
31.599582012121115
```

3.2 Case 2: Double la bande passante réseau du serveur (S)

Il faut assumer des parametres comme ca:

```
AMELIORE_VITESSE_SERVEUR = 1
AMELIORE_BANDE_PASSANTE = 2
AMELIORE_NOMBRE_SERVEUR = 1
```

Resultat:

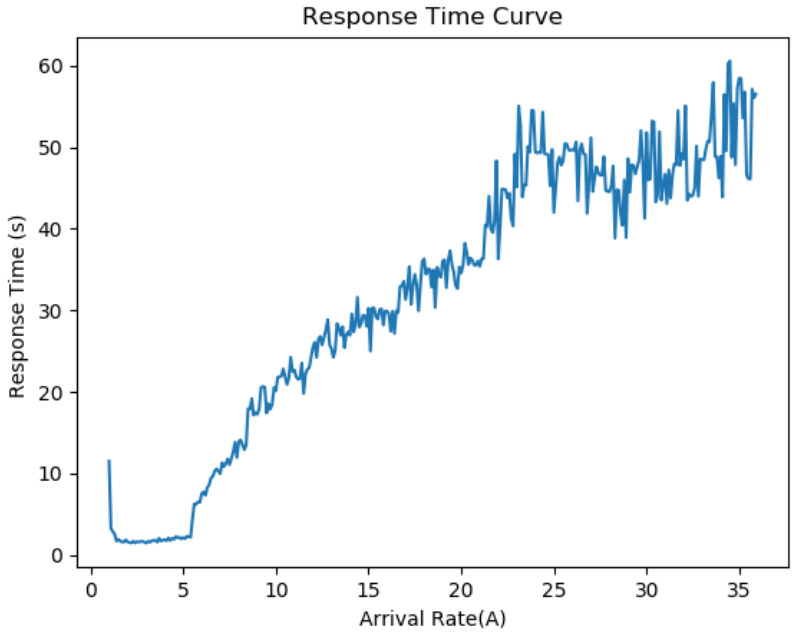



```
==== MOYEN TEMPS DE REPONSE ====
moyen temps de reponse:
17.32944030344586
```

3.3 Case 3: Double le nombre de serveur (dans la file d'attente SR)

```
AMELIORE_VITESSE_SERVEUR = 1
AMELIORE_BANDE_PASSANTE = 1
AMELIORE_NOMBRE_SERVEUR = 2
```

Resultat:




```
==== MOYEN TEMPS DE REPONSE ====
moyen temps de reponse:
31.59798199859626
```

3.4 : Evaluer et classer les trois propositions

Case	Le temps moyen de réponse
Initial	31.59798205865761
Case 1	31.599582012121115
Case 2	17.32944030344586
Case 3	31.59798199859626

En conclusion, doubler la bande passante réseau du serveur est le meilleur. Cela est logic. En fait, surtout des serveurs des grandes entreprise comme Google, Facebook... La vitesses envoie des données à Internet plus haut (Gb/s). Cela nous aide à réduire le temps de latence important lors de l'accès à leurs services.