Modélisation du slicing dans les réseaux 5G

L. Decreusefond & A. Vergne

2025

1 Introduction

La 5G permet de déployer des réseaux virtuels de bout en bout, avec des profils de qualité de service (QoS) spécifiques, au-dessus d'une infrastructure physique commune. Le slicing est le terme utilisé pour désigner la fonctionnalité qui rend cette coexistence possible. Chaque réseau virtuel déployé est appelé une tranche dans la terminologie de la 5G. Le découpage en tranches est particulièrement difficile à dimensionner pour respecter les contraintes de qualité de service des tranches. Par exemple, des tranches telles que le haut débit mobile amélioré (eMBB) et les communications ultra fiables à faible latence (URLLC) ont des exigences contradictoires en matière de qualité de service

On s'intéresse ici à deux types distincts : les flux URLLC et eMMB. Les premiers sont des flux qui doivent être ultra-fiables et avec une faible latence. Traduit en langage de files d'attente, cela signifie que la perte doit être infime et qu'on ne peut pas se permettre de les retarder par une mise en buffer.

Pour les flux, eMMB, comme d'habitude, ils passent quand ils peuvent même si normalement, ils ne devraient pas souffrir de trop de délai.

Toute la difficulté est de rtouver un moyen physique qui permette de réaliser cette priorisation tout en étant capable d'en étudier les performances pour dimensionner les ressources.

Dans un premier temps, nous regardons un modèle théorique qui s'étudie relativement bien. Dans un deuxième temps, nous envisagerons un modèle plus réaliste à mettre en œuvre mais qui s'étudie plus difficilement.

2 Préliminaires

On rappelle que la formule d'Erlang-B donne la probabilité que S serveurs soient occupés à l'état stationnaire :

$$\mathrm{Erl}_B[\rho, S] = \frac{\frac{\rho^S}{S!}}{\sum_{j=0}^{S} \frac{\rho^j}{j}}.$$

On a la relation de récurrence :

$$\frac{1}{\mathrm{Erl}_B[\rho, 0]} = 1$$
$$\frac{1}{\mathrm{Erl}_B[\rho, S]} = 1 + \frac{S}{\rho\,\mathrm{Erl}_B[\rho, S-1]}.$$

Partie 1. *1) Écrire une fonction Python qui renvoie le nombre moyen de clients dans une file M/M/S/S à l'état stationnaire sans calculer de factorielle.*

```
def mean_number_waiting_customers(arrival_rate,
service_rate,nb_of_servers):

    return ...
```

2) Pour un choix de paramètre ρ et S tels que $\mathrm{Erl}_B[\rho, S]$ soit petit (de l'ordre de 10^{-3}), qu'est-ce que l'on remarque à propos du nombre moyen de clients ? Expliquer ce phénomène en vous aidant des résultats connus sur la M/M/∞.

■

3 Modélisation

On suit le modèle proposé dans [1] qui n'est pas implémentable dans un système réel mais qui s'analyse mathématiquement très bien.

On considère une file d'attente avec un buffer infini et S serveurs.

Il y a deux classes de clients de type 1 et de type 2. Les clients de type 1 ont une priorité plus élevée que les clients de type 2. Les clients de type 1 arrivent selon un processus de Poisson de paramètre λ_1 et les clients de type 2 arrivent selon un processus de Poisson de paramètre λ_2. Les clients de type 1 ont une durée de service exponentielle de paramètre μ_1 et les clients de type 2 ont une durée de service exponentielle de paramètre μ_2.

Les clients de type 1 ne peuvent pas être bufferisés et doivent être servis immédiatement. Les clients de type 2 peuvent être bufferisés. On suppose que la capacité de la file d'attente est infinie.

Les clients de classe 1 préemptent les serveurs : s'il reste des serveurs libres, ils s'y mettent normalement mais si tous les serveurs sont pris, ils prennent la place d'un client

de classe 2. Celui-ci se retrouve dans le buffer et reprendra son service, là où il en était, dès qu'un serveur se libérera. S'il n'y a que des clients de classe 1 en service, le client de type 1 qui arrive est perdu.

Les clients de classe 2 ne peuvent accéder à un serveur que s'il y en a de libre. S'ils arrivent et qu'il n'y a pas de serveur libre, ils sont mis dans le buffer.

On note Q_1 le nombre de clients de type 1 dans le système et Q_2 le nombre de clients de type 2 dans le système. On note S_1 le nombre de serveurs occupés par des clients de type 1 et S_2 le nombre de serveurs occupés par des clients de type 2. On note B le nombre de clients de type 2 dans le buffer.

Partie 2. *1) Quelles sont les contraintes sur les variables d'état du système et comment les variables Q_2, S_2 et B sont-elles reliées ? Expliquer en particulier pourquoi $q_1 + s_2 < S$ ne peut se produire que si $b = 0$.*

2) Montrer que le processus Q_1 est un processus de Markov et reconnaître sa dynamique comme celle d'une file simple dont on précisera les caractéristiques.

3) Écrire les transitions possibles du processus de Markov (Q_1, Q_2). Montrer en particulier que le taux de transition de l'état (q_1, q_2) à l'état $(q_1, q_2 - 1)$ est donnée par

$$\min\{q_2, S - q_1\}\mu_2. \tag{1}$$

4) Simuler l'évolution de ce système en Python. On prendra comme valeurs

$$S = 10,\ \mu_1 = 2,\ \mu_2 = 1, \lambda_1 = 4, \lambda_2 = 3.$$

On note (x_1, x_2) le processus ainsi construit. On vérifiera notamment que

$$\frac{1}{T}\int_0^T \mathbf{1}_{\{S\}}(x_1(s))\ ds \xrightarrow{T\to\infty} \frac{\rho_1^S/S!}{\sum_{j=0}^S \rho_1^j/j!}.$$

Préalablement, on justifiera cette identité.

4 Stationnarité

Il est montré dans [1] que ce système admet un régime stationnaire si et seulement si ρ_2 plus le nombre moyen de clients dans une file M/M/S/S de charge ρ_1 est strictement inférieur à S :

$$\rho_2 + \frac{1}{\sum_{j=0}^S \rho_1^j/j!} \sum_{k=0}^{S} k\,\frac{\rho_1^k}{k!} < S, \tag{2}$$

où $\rho_i = \lambda_i/\mu_i$.

Partie 3. *1) Illustrer ce résultat par simulation. Serait-il possible de deviner* (2)

par simulation ?

■

5 Calcul de la probabilité stationnaire

En s'aidant de la section 6 de [1], on veut calculer la probabilité stationnaire π de notre système sous réserve que (2) soit satisfaite. Par définition π est un *vecteur* de taille infinie indexée par les valeurs possibles de q_1 et q_2. On numérote les états (q_1, q_2) en ordre lexicographique à droite :

$$(0,0) \prec (1,0) \prec \ldots \prec (S,0) \prec (0,1) \prec \ldots (S,1) \prec \ldots$$

et on forme les vecteurs ligne à $S+1$ coordonnées

$$x_i = (\pi_{(0,i)}, \pi_{(1,i)}, \cdots, \pi_{(S,i)}).$$

On note M la matrice $(S+1)\times(S+1)$ qui correspond au générateur d'une file M/M/S/S de paramètres λ_1 et μ_1. Le générateur de (Q_1, Q_2) s'écrit sous tri-diagonale par blocs de la forme

$$\begin{pmatrix}
M-\lambda_2\,\mathrm{Id} & \lambda_2\,\mathrm{Id} & & & & & & & \\
A_1 & B_1 & \lambda_2\,\mathrm{Id} & & & & & & \\
 & A_2 & B_2 & \lambda_2\,\mathrm{Id} & & & & & \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots \\
 & & & A_S & B_S & \lambda_2\,\mathrm{Id} & & & \\
 & & & & A_S & B_S & \lambda_2\,\mathrm{Id} & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots
\end{pmatrix}$$

où

$$B_j = M - A_j - \lambda_2\,\mathrm{Id}$$

et $A_j = \mathrm{diag}(a_{n,0}, \cdots, a_{n,S})$ avec

$$a_{n,j} = \min(S-j, n)\mu_2.$$

On admet qu'il existe une matrice R_S à coefficients positifs (et le plus petit possible) qui soit solution de l'équation matricielle :

$$\lambda_2\,\mathrm{Id} + RB_S + R^2 A_S = 0. \tag{3}$$

Pour trouver cette matrice R, on réécrit (3) sous la forme

$$R = -(\mathrm{Id} + R^2\tilde{A}_S)\tilde{B}_S^{-1}$$

où

$$\tilde{A}_S = \frac{1}{\lambda_2}A_S \text{ et } \tilde{B}_S = \frac{1}{\lambda_2}B_S.$$

On considère ensuite la suite de matrices

$$R_0 = 0$$
$$R_{n+1} = -(\mathrm{Id} + R_n^2 \tilde{A}_S)\tilde{B}_S^{-1}.$$

Assez rapidement cette suite converge vers une matrice solution de (3). Que cette matrice soit à coefficients positifs les plus petits possibles est une propriété qui est démontrée dans [2]. On admet aussi que le rayon spectral de cette matrice est strictement inférieur à 1 donc que $\mathrm{Id} - R$ est inversible et que l'on a

$$\sum_{j=0}^{\infty} R^j = (\mathrm{Id} - R)^{-1}. \tag{4}$$

Partie 4. *1) Montrer que la suite $(x_j, j \geq S)$ définie par*

$$x_j = x_S R^{j-S} \text{ pour } j \geq S.$$

est solution des équations d'équilibre au delà du rang S.

2) Établir que

$$x_{S-1} = -x_S(\tilde{B}_S + R\tilde{A}_S).$$

3) Expliciter par récurrence la suite de matrices $(T_j, j = S-1, \cdots, 0)$ telle que l'on ait

$$x_{j+1} = x_j T_j.$$

4) Montrer enfin que l'on a

$$x_0\left((M - \lambda_2 I) + T_0 A_1\right) = 0 \tag{5}$$

où $\tilde{M} = M/\lambda_2$.

5) Expliquer comment on calcule x_S.

6) Retrouver la Figure 2 de [1] pour $S = 5$.

■

6 Canaux de garde

Comme il est difficile d'implémenter la politique préemptive, on se contente souvent d'un système de canaux de garde. Les arrivées de classe 1 occupent un serveur tant qu'il y a un de libre et ne peuvent être mis dans la salle d'attente. En d'autres termes, si tous les serveurs sont pris, éventuellement en partie par des clients de classe 2, les clients de classe 1 qui arrivent sont perdus.

Les clients de classe 2 ne peuvent entrer dans les serveurs que s'il y a au moins G serveurs libres (avec G à déterminer mais généralement très petit devant S). Cette règle s'applique à leur arrivée ou au moment où un serveur se libère.

Partie 5. *1) Pourquoi est-ce que le processus (Q_1, Q_2) défini précédemment n'est plus un processus de Markov représentant ce système ?*
2) Représenter la dynamique de ce système par un processus de Markov dont on précisera le générateur infinitésimal.
3) Sans faire de calculs, est-ce que la condition de stabilité est plus ou moins contraignante sur ρ_2 que dans le premier modèle ?

■

Références

[1] Evsey Morozov et al. « Modified Erlang loss system for cognitive wireless networks ». In : *Mathematics* 10.12 (2022), p. 2101. URL : `https://partage.imt.fr/index.php/s/a4SSkqYgXR88f5J`.

[2] Marcel F Neuts. *Matrix-geometric solutions in stochastic models : an algorithmic approach*. Courier Corporation, 1994.