



HAL
open science

$0 = 0$, c'est le truc du noyau! Application aux files d'attente

Anne Bouillard, Céline Comte, Élie de Panafieu, Fabien Mathieu

► **To cite this version:**

Anne Bouillard, Céline Comte, Élie de Panafieu, Fabien Mathieu. $0 = 0$, c'est le truc du noyau! Application aux files d'attente. ALGOTEL 2019 - 21èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, Jun 2019, Saint Laurent de la Cabrerisse, France. hal-02118156

HAL Id: hal-02118156

<https://hal.science/hal-02118156>

Submitted on 2 May 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

0 = 0, c'est le truc du noyau !

Application aux files d'attente

Anne Bouillard¹ et Céline Comte^{1,2} et Élie de Panafieu¹ et Fabien Mathieu¹ †

¹Nokia Bell Labs France, Nozay, France

²Télécom ParisTech, Paris, France

Le *calcul réseau stochastique* est un outil de calcul de bornes d'erreur sur la performance des réseaux de files d'attente. Obtenir des bornes précises pour des réseaux constitués de plusieurs files ou soumis à des arrivées non-indépendantes est un exercice délicat. Dans ce papier, nous évaluons la pertinence des outils de combinatoire analytique pour aborder ce problème. Nous appliquons en particulier la *méthode du noyau* pour exprimer les fonctions génératrices des distributions des états des files, ce qui nous permet de calculer des bornes d'erreur d'une précision arbitraire sur le comportement à l'état stationnaire. Dans ce travail préliminaire, nous nous focalisons sur des exemples simples mais représentatifs des difficultés que la méthode du noyau permet de surmonter. Ces résultats sont validés par des simulations.

Cet article est une version condensée de l'article [BCdPM18].

Mots-clefs : files d'attente, combinatoire analytique, méthode du noyau, calcul réseau stochastique

1 Introduction

Le développement des nouvelles technologies de communication sans fil (5G) a jeté un nouvel éclairage sur la théorie des files d'attente, avec de fortes exigences en termes d'occupation du tampon, de latence et de fiabilité. Dans beaucoup de scénarios, les paquets arrivent groupés et sont traités par un serveur à taux constant. La file $G/D/1$ apparaît donc comme un modèle naturel. Le *calcul réseau stochastique* [FR15] a été développé pour analyser de telles files, et en particulier pour calculer des bornes d'erreur sur leur performance en intégrant des outils probabilistes dans le formalisme du *calcul réseau*. Malheureusement, si les bornes sont très précises pour une file isolée [PC14, CP15], elles deviennent très grossières lorsqu'elles sont appliquées à un réseau de plusieurs files [CBL06, NS17]. Les fonctions génératrices, elles, sont au cœur de la combinatoire analytique [FS09], un sous-domaine de la combinatoire. Cette communauté a développé des outils mathématiques tels que la *méthode du noyau* [BF02, BMM10] pour étudier les marches aléatoires [FG00], dont les liens avec la théorie des files d'attente ont déjà été identifiés et exploités [FIM17]. Dans cet article, nous montrons comment utiliser les séries génératrices et la méthode du noyau pour obtenir des bornes d'erreur précises sur les systèmes de files d'attente.

2 Présentation de la méthode sur une file simple

Dans cette partie, nous présentons l'exemple d'une file à un serveur et un flot de paquets, comme représentée en FIGURE 1. Les résultats présentés ici ne sont pas nouveaux (nous finissons par redécouvrir la formule de Pollaczek-Khinchine et appliquons les résultats développés par [BF02]) mais notre objectif est de présenter la méthode. Des résultats nouveaux seront présentés dans la partie 3.

Modèle de file d'attente La file est initialement vide. À chaque unité de temps $t \geq 1$, un paquet est servi (si la file est non-vide) puis \mathbf{A}_t paquets arrivent. La suite $(\mathbf{A}_t)_{t \geq 0}$ est i.i.d. de fonction génératrice A et de moyenne $\lambda < 1$. On note \mathbf{X}_t le nombre de paquets dans la file à la fin du temps t . L'évolution du système est décrite par les équations suivantes :

$$\mathbf{X}_0 = 0 \quad \text{et} \quad \mathbf{X}_{t+1} = (\mathbf{X}_t - 1)_+ + \mathbf{A}_{t+1}, \quad \forall t \geq 0, \quad \text{où } (\cdot)_+ = \max(\cdot, 0). \quad (1)$$

†Les auteurs sont membres du LINCS, voir <https://www.lincs.fr>.

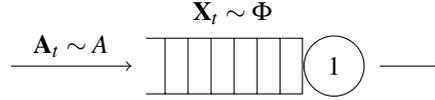


FIGURE 1: Une file à un serveur traversée par un unique flot de paquets. Le traitement d'un paquet par le serveur nécessite une unité de temps.

Méthode symbolique On définit la fonction génératrice [FS09] de l'état de la file par

$$\Phi(u, z) = \sum_{n \geq 0} \sum_{t \geq 0} \mathbb{P}(\mathbf{X}_t = n) u^n z^t. \quad (2)$$

Pour chaque $t \in \mathbb{N}$, extraire le coefficient de z^t donne la fonction génératrice de la variable aléatoire \mathbf{X}_t :

$$[z^t] \Phi(u, z) = \sum_{n \geq 0} \mathbb{P}(\mathbf{X}_t = n) u^n. \quad (3)$$

En utilisant la *méthode symbolique* [FS09], on peut traduire l'équation (1) en l'équation suivante vérifiée par la fonction génératrice Φ :

$$\Phi(u, z) = 1 + zA(u) [(\Phi(u, z) - \Phi(0, z))u^{-1} + \Phi(0, z)]. \quad (4)$$

Bien que cette équation caractérise complètement la fonction Φ , en tirer une formule explicite pour $\Phi(u, z)$ n'est pas immédiat puisque cela nécessiterait de connaître une expression pour $\Phi(0, z)$. Nous présentons maintenant une méthode systématique pour contourner ce problème.

Méthode du noyau L'équation (4) peut se ré-arranger en

$$\Phi(u, z) [1 - zA(u)u^{-1}] = 1 - \Phi(0, z)zA(u) [u^{-1} - 1]. \quad (5)$$

Lorsque le membre de gauche de (5) s'annule, le membre de droite s'annule aussi : $0 = 0$. La méthode du noyau [BF02, BMM10] exploite cette remarque simple en choisissant $u = U(z)$ tel que le second facteur du membre de gauche s'annule. Ici, $U(z)$ est implicitement défini par l'égalité $U(z) = zA(U(z))$, qui se trouve aussi être l'équation qui caractérise la fonction génératrice T_A du nombre de nœuds dans un arbre de Galton-Watson dont la distribution du nombre d'enfants par nœud admet A comme fonction génératrice. L'annulation du membre de droite de (5) permet de calculer $\Phi(0, z)$ en fonction de $U(z) = T_A(z)$, puis de trouver $\Phi(u, z)$. On obtient ainsi

$$\Phi(0, z) = \frac{1}{1 - T_A(z)} \quad \text{et} \quad \Phi(u, z) = \frac{1 + \frac{1}{1 - T_A(z)} zA(u) (1 - u^{-1})}{1 - zA(u)u^{-1}}. \quad (6)$$

Performances asymptotiques Nous souhaitons calculer des *bornes d'erreur*, c'est-à-dire la probabilité qu'une variable aléatoire \mathbf{X} , distribuée selon la distribution stationnaire de la chaîne de Markov $(\mathbf{X}_t)_{t \geq 1}$, dépasse une valeur R . On calcule d'abord la fonction génératrice Π de la distribution stationnaire, puis on obtient le comportement asymptotique de la borne d'erreur à partir de Π .

La fonction génératrice $\Pi(u)$ est la limite de $[z^t] \Phi(u, z)$ quand t tend vers $+\infty$, et cette limite se calcule en étudiant $\Phi(u, z)$ au voisinage de $z = 1$. En effet, on vérifie aisément que $z = 1$ est la singularité dominante de $\Phi(u, z)$. En utilisant le développement $T_A(z) = 1 + (z - 1)T'_A(1) + o(z - 1)$, l'égalité $T'_A(1)(1 - \mathbb{E}[\mathbf{A}_t]) = 1$ et le résultat de [FS09, p. 294, Th. V.1], on retrouve la formule de Pollaczek-Khinchine

$$\Pi(u) = (1 - \lambda) \frac{A(u)(u - 1)}{u - A(u)}.$$

La fonction génératrice de la borne d'erreur est

$$E(u) = \sum_{R \geq 0} \left(\sum_{n \geq R} \pi(n) \right) u^R = \frac{1 - u\Pi(u)}{1 - u}. \quad (7)$$

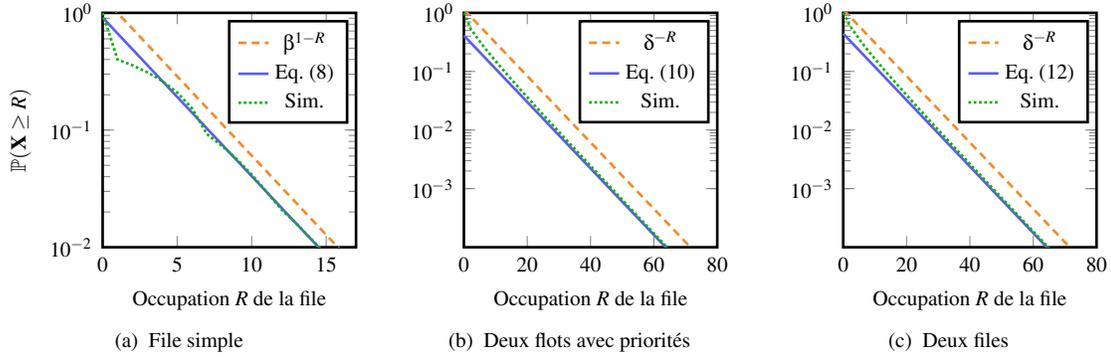


FIGURE 2: Evaluation numérique. Paramètres $A = D_{2/30,6}$ et $B = D_{2/5,1}$.

d'inégalité de Doob à laquelle nous comparer, mais nous traçons la courbe δ^{-R} pour nous rendre compte de ce qu'aurait donné une inégalité qui, comme Doob, ne tient pas compte du terme constant.

Enfin, la FIGURE 2c illustre le cas de deux files en tandem et montre l'occupation du tampon dans la seconde file. Les bornes d'erreur ne diffèrent que d'un facteur constant par rapport au cas précédent. Il apparaît que le comportement asymptotique calculé dans ces deux cas, bien qu'exact, soit une borne inférieure de la borne d'erreur. Pour encore plus de précision, les formules que nous obtenons permettent de calculer des bornes d'erreur exactes des premiers termes en utilisant les développements de Taylor.

Références

- [BCdPM18] Anne Bouillard, Céline Comte, Elie de Panafieu, and Fabien Mathieu. Of kernels and queues : When network calculus meets analytic combinatorics. In *30th International Teletraffic Congress, ITC 2018, Vienna - Volume 2*, pages 49–54, 2018.
- [BF02] Cyril Banderier and Philippe Flajolet. Basic analytic combinatorics of directed lattice paths. *Theor. Comput. Sci.*, 281(1–2) :37–80, 2002.
- [BMM10] Mireille Bousquet-Mélou and Marni Mishna. Walks with small steps in the quarter plane. *Contemp. Math.*, 520 :1–40, 2010.
- [CBL06] Florin Ciucu, Almut Burchard, and Jörg Liebeherr. Scaling properties of statistical end-to-end bounds in the network calculus. *IEEE Trans. Inform. Theory*, 52(6) :2300–2312, 2006.
- [CP15] Florin Ciucu and Felix Poloczek. On multiplexing flows : Does it hurt or not ? In *IEEE Conf. on Comput. Commun., INFOCOM*, pages 1122–1130, 2015.
- [FG00] Philippe Flajolet and Fabrice Guillemin. The formal theory of birth-and-death processes, lattice path combinatorics and continued fractions. *Advances in Applied Probability*, 32(03) :750–778, 2000.
- [FIM17] Guy Fayolle, Roudolf Iasnogorodski, and Vadim Malyshev. *Random Walks in the Quarter Plane : Algebraic Methods, Boundary Value Problems, Applications to Queueing Systems and Analytic Combinatorics*. Springer Publishing Company, Incorporated, 2nd edition, 2017.
- [FR15] Markus Fidler and Amr Rizk. A guide to the stochastic network calculus. *IEEE Commun. Surveys and Tutorials*, 17(1) :92–105, 2015.
- [FS09] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 1 edition, 2009.
- [NS17] Paul Nikolaus and Jens B. Schmitt. On per-flow delay bounds in tandem queues under (in)dependent arrivals. In *IFIP Networking Conference*, pages 1–9, 2017.
- [PC14] Felix Poloczek and Florin Ciucu. Scheduling analysis with martingales. *Perform. Eval.*, 79 :56–72, 2014.