EVALUATION DE PERFORMANCES
ET
APPLICATIONS AUX RESEAUX

PREMIERE PARTIE :

THEORIE DES PROBABILITES
CHAINES DE MARKOV
THOERIE DES FILES D'ATTENTE
SIMULATIONS A EVENEMENTS DISCRETS

Olivier Brun - Urtzi Ayesta

TABLE DES MATIÈRES

TABLE DES FIGURES

CHAPITRE 1

Introduction

1.1 Evaluation des performances des réseaux

Durant la dernière décennie, les réseaux de télécommunications ont connu un développement sans précédent qui s'est accompagné d'un accroissement important de leur complexité. La maîtrise de ces systèmes complexes nécessite des modèles et des outils permettant de prédire les performances d'un réseau et de les optimiser.

Comme illustré sur la figure 1.1, on considérera ici un réseau de télécommunication comme un système servant le trafic entrant. Ce trafic entrant est généré par les utilisateurs du réseau. On ne sait jamais qui communique avec qui, quand et pour combien de temps. Par conséquent, une caractéristique essentielle des modèles réseaux que nous allons étudier est que ce sont des modèles stochastiques, c'est à dire qu'ils intègrent l'incertitude liée d'une part aux arrivées des communications dans le réseau et d'autre part à leurs durées.

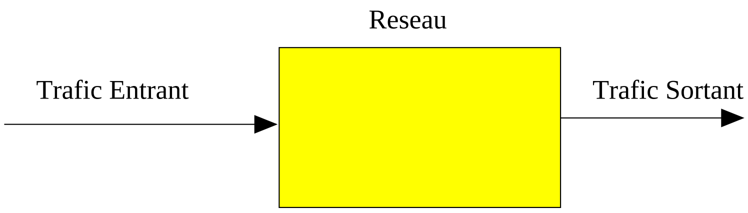

FIGURE 1.1 – Point de vue système.

Les modèles réseaux doivent nous permettre de répondre à trois grandes

questions :
- Etant donnés la demande (le trafic) et les ressources disponibles, quelle est la qualité de service offerte aux utilisateurs ?
- Etant donnés la demande et une qualité de service cible, comment dimensionner les ressources du système ?
- Etant donnés les ressources disponibles et une qualité de service cible, quelle est la demande maximale ?

Autrement dit, comme illustré sur la figure 1.2, ces modèles doivent nous permettre d'étudier les relations entre d'une part la demande (le trafic) et la qualité de service requise et d'autre part la capacité du système (quantité de ressources disponibles) et donc son coût.

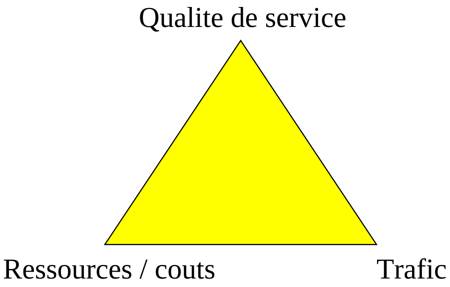

FIGURE 1.2 – Les trois facteurs clés.

De manière générale, les techniques d'évaluation des performances doivent nous permettre :
- de prédire les performances (QoS, utilisation des ressources) avant la construction ou l'extension d'un réseau.
- d'évaluer et comparer plusieurs scénarii : modification des équipements, configuration des protocoles de routage, etc.
- d'anticiper le comportement du réseau suite à une prévision d'augmentation du trafic ou suite à une panne d'équipement,
- etc.

1.2 Exemple introductif

Pour illustrer ceci, considérons tout de suite un exemple. Une société est raccordée au réseau téléphonique par une liaison supportant N appels simultanément (cf. figure 1.3). Un appel arrivant alors que les N circuits sont pris est bloqué et rejeté.

On sait que :
- La société compte 100 employés. En heure de pointe, chacun téléphone en moyenne 4 fois, soit 400 appels/heure (un appel toutes les 9 secondes).
- Chaque appel dure en moyenne 3 minutes (180 secondes).

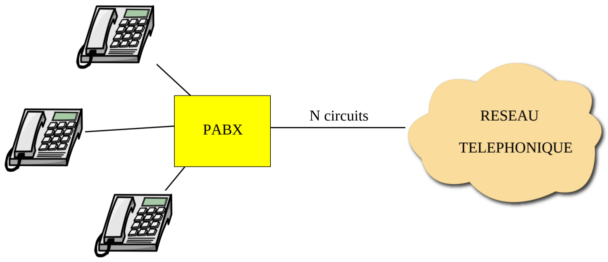

FIGURE 1.3 – Liaison téléphonique ayant N circuits.

Dans cet exemple, la capacité du système correspond aux N circuits. La demande correspond aux appels téléphoniques des employés de la société. Enfin, la qualité de service correspond au taux de blocage des appels. On voudrait un taux de blocage inférieur à 1%.

La première question que l'on peut se poser est la suivante. S'il n'y avait pas de blocage, combien y aurait t-il en moyenne d'appels simultanés sur le lien reliant l'entreprise au réseau téléphonique ? Nous verrons plus tard, que si la capacité $N \to \infty$, il y aurait en moyenne $Y = (1/9) * 180 = 20$ appels sur le lien.

Sachant que le coût de la liaison dépend directement de sa capacité, il est clair que prendre par exemple $N = 200$ si on ne doit avoir que 20 appels simultanés en moyenne n'est pas rentable.

Intuitivement, on peut penser que, vu que le trafic offert est de 20, une capacité $N = 20$ doit convenir. En fait, comme on le verra, ce choix conduit à ce que 16% des appels soient rejetés (64 appels sur les 400). Ceci s'explique en fait par la variabilité liée à l'arrivée des appels et à leurs durées.

Si on veut un taux de blocage de 1%, il faut en fait comme on le verra une capacité de $N = 30$ circuits.

1.3 Méthodes d'évaluation de performances

Les approches traditionnelles dans ce domaine sont le prototypage, la modélisation analytique et la simulation événementielle.

Le prototypage est en général hors de coût même pour des réseaux de taille modeste. La métrologie permet éventuellement de mesurer certains indices de performances, quand la quantité de données à collecter n'est pas monumentale. Mais il peut être difficile d'effectuer ces mesures sans perturber le réseau opérationnel. Surtout, cette approche ne permet pas de prédire les performances du réseau en cas de panne, d'évolution de la matrice de trafic ou de modification de la configuration du réseau.

La modélisation analytique, basée sur la théorie des files d'attente, est une approche puissante mais dont la capacité à représenter finement certains mécanismes complexes, et notamment le comportement réactif des protocoles, reste limitée. Par conséquent, une modélisation analytique exacte ayant toutes les propriétés précédentes semble aujourd'hui complètement inenvisageable compte tenu de nos connaissances théoriques. En particulier, il est bien connu qu'une modélisation analytique des phénomènes transitoires dans les réseaux de files d'attente est très complexe et peu de résultats généraux existent dans ce domaine.

Enfin, les modèles de simulation à événements discrets n'ont pas ces limitations, bien qu'ils requièrent souvent des temps de calcul prohibitifs pour des réseaux de taille réelle. De par sa grande flexibilité, la simulation constitue donc un outil souvent incontournable pour évaluer les performances des réseaux de communication.

1.4 Plan du cours

Le cours est organisé de la manière suivante :
- Rappels sur l'évaluation des performances :
  - Théorie des probabilités et chaînes de Markov (O. Brun)
  - Files d'attente élémentaires (U. Ayesta)
- Simulation événementielle (O. Brun)
- Réseaux à commutation de circuits (O. Brun)
- Réseaux à commutation de paquets (U. Ayesta)

CHAPITRE 2

Rappels sur la Théorie des Probabilités

L'intêret principal de la théorie des probabilités est de nous permettre de modéliser une expérience (ou un système) sans disposer de l'information necéssaire pour la décrire exactement. Supposons par exemple que l'on lance une pièce de monnaie en l'air et qu'on observe sur quelle face elle va tomber. Pour prédire si le résultat sera pile ou face, il faut disposer d'une modélisation parfaite de la pièce, de l'état de l'air, de la vitesse à laquelle elle est lancée, etc. De même, si on veut étudier le trafic dans un réseau téléphonique, il faudrait pouvoir prédire, pour chaque abonné au téléphone, quand-est qu'il va téléphoner, à qui, et pendant combien de temps. C'est parce que nous sommes incapables de faire cette modélisation, ou que nous y renonçons, qu'il est necéssaire de faire appel aux probabilités.

Ce chapitre présente un rappel (très) synthétique des principales notions de la théorie des probabilités.

2.1 Evénement

On considère un ensemble $\mathcal{S}$ qui représente l'ensemble des issues possibles d'une expérience. Par exemple, si je jette un dés, l'ensemble des issues possibles est $\mathcal{S} = \{1, 2, 3, 4, 5, 6\}$. C'est un ensemble discret. Au contraire, si mon expérience consiste à mesurer très précisément le temps jusqu'à ce que quelqu'un entre dans mon bureau, l'ensemble $\mathcal{S}$ est continu : $\mathcal{S} = \{x \in \mathbb{R} | x > 0\}$.

Un événement A est un sous-ensemble de l'ensemble $\mathcal{S}$ des issues possibles pour l'expérience. Si le résultat de l'expérience appartient à A, on dit que l'événement A s'est réalisé. Par exemple, je peux m'intéresser à l'événement $A = \{2, 4, 6\} \subset \mathcal{S}$ que le jet d'un dés donne un nombre pair. Ou bien m'intéresser à l'événement $B = \{x \in \mathbb{R} | x > 10.4\} \subset \mathcal{S}$ que personne ne rentre dans mon bureau avant 10.4 secondes.

L'ensemble $\mathcal{S}$ tout entier est appelé l'événement certain, alors que le sous-ensemble vide $\emptyset$ de $\mathcal{S}$ est appelé l'événement impossible.

Etant donnés deux événements A et B, si le résultat de l'expérience appartient à $A \cup B$, on dira que l'événement A **ou** l'événement B s'est produit. Si le résultat de l'expérience appartient à $A \cap B$, on dira que A **et** B se sont produits. Si $A \cap B = \emptyset$, les deux événements ne peuvent se produire simultanément : ils sont dits mutuellement exclusifs. Enfin, si le résultat de l'expérience n'appartient pas à A, c'est qu'il appartient à son complémentaire $\overline{A} = \{e \in \mathcal{S} | e \notin A\}$ dans $\mathcal{S}$ (non A).

Un ensemble d'événements $A_1, A_2, \ldots$ forment une partition de $\mathcal{S}$ s'ils sont mutuellement exclusifs ($A_i \cap A_j = \emptyset$ si $i \neq j$) et couvrent l'ensemble des issues possibles de l'expérience ($\cup_i A_i = \mathcal{S}$).

2.2 Probabilité d'un événement

Jusqu'à maintenant, on a considéré une expérience réalisée une seule fois. Supposons qu'elle soit répétée N fois. On peut alors, pour chaque événement A définir le nombre $N(A)$ de fois où il se produit (i.e le résultat appartient à A).

Empiriquement, on va définir la probabilité $P[A]$ de l'événement A comme la fréquence relative à laquelle se produit cet événement au cours d'une expérience répétée un nombre infini de fois :

$$P[A] = \lim_{N \to \infty} N(A)/N$$

On en déduit un certain nombre de propriétés :

1. $0 \leq P[A] \leq 1$
2. $P[\mathcal{S}] = 1$ et $P[\emptyset] = 0$
3. $P[A \cup B] = P[A] + P[B] - P[A \cap B]$
4. $P[A \cup B] = P[A] + P[B]$ si $A \cap B = \emptyset$
5. $P[\overline{A}] = 1 - P[A]$
6. $P[A] \leq P[B]$ si $A \subset B$

2.2.1 Probabilité conditionnelle

L'occurrence d'un événement A peut dépendre du fait que l'événement B s'est produit. La probabilité de l'événement A sachant que B s'est produit est donnée par,

$$P[A|B] = \frac{P[A \cap B]}{P[B]}$$

On a donc : $P[A \cap B] = P[A|B] \times P[B]$.

2.2.2 Loi des probabilités totales

Soit $B_1, B_2, \ldots, B_n$ des événements formant une partition de $\mathcal{S}$. Tout événement A peut alors s'écrire,

$$A = A \cap \mathcal{S} = A \cap (\cup_i B_i) = \cup_i (A \cap B_i)$$

d'où l'on déduit que,

$$P[A] = \sum_{i=1}^{n} P[A \cap B_i] = \sum_{i=1}^{n} P[A|B_i] \, P[B_i]$$

Cela permet de calculer la probabilité de A en conditionnant par les événements B_i. Autrement dit, on décompose le problème en sous-problèmes.

2.2.3 Formule de Bayes

De nouveau, $B_1, B_2, \ldots, B_n$ sont des événements formant une partition de $\mathcal{S}$. On veut calculer la probabilité de l'événement B_i sachant que l'événement A s'est produit,

$$P[B_i|A] = \frac{P[A \cap B_i]}{P[A]} = \frac{P[A|B_i] \, P[B_i]}{\sum_{j=1}^{n} P[A|B_j] \, P[B_j]}$$

La formule de Bayes permet de calculer une probabilité conditionnelle quand on connaît les probabilités conditionnelles inverses. Prenons un exemple. Soit 3 cartes. La première a ses deux faces rouges (rr). La deuxième a ses deux faces bleues (bb). Enfin la troisième a une face rouge et une face bleu (rb). On tire une carte au hasard et on ne regarde que la face de dessus. Elle est rouge. Quelle est la probabilité que l'autre face soit bleue ?

$$\begin{aligned} P[rb|rouge] &= \frac{P[rouge|rb] \, P[rb]}{P[rouge|rr] \, P[rr] + P[rouge|bb] \, P[bb] + P[rouge|rb] \, P[rb]} \\ &= \frac{1/2 \times 1/3}{1 \times 1/3 + 0 \times 1/3 + 1/2 \times 1/3} \\ &= 1/3 \end{aligned}$$

2.2.4 Indépendance

Deux événements A et B sont dits indépendants si et seulement si $P[A \cap B] = P[A]\,P[B]$. Dans ce cas, on a :

$$P[A|B] = \frac{P[A \cap B]}{P[B]} = P[A]$$

Autrement, savoir que B s'est produit n'apporte aucune information sur la probabilité que A se produise.

Exercice : on jette deux dés. Soit A l'événement "on a un 6 avec le premier dés" et B l'événement "on a un 1 avec le second dés". Montrer que les deux événements sont indépendants.

Exercice : on jette deux dés. Soit A l'événement "on a un 6 avec le premier dés" et B l'événement "la somme des deux dés fait 9". Montrer que les deux événements ne sont pas indépendants.

2.3 Variables aléatoires

Si on tire 10 fois à pile ou face, ce qui va nous intéresser c'est plus le nombre de fois où la pièce est tombée sur face que la séquence de piles et de faces obtenue. Ainsi, si on obtient le résultat pppfppffpf, ce qui nous intéresse c'est que la pièce est tombé 4 fois sur face.

De manière générale, on est souvent plus intéressé par une mesure (un nombre) associée à l'expérience que par le résultat proprement dit. Cette mesure est appelée variable aléatoire (v.a.). Ainsi, une variable aléatoire à valeurs réelles est une fonction $X : \mathcal{S} \rightarrow \mathbb{R}$ qui associe à chaque issue e possible de l'expérience un nombre réel $X(e)$.

L'image d'une variable aléatoire X est l'ensemble,

$$\mathcal{S}_X = \{x \in \mathbb{R} | \exists e \in \mathcal{S}, X(e) = x\}$$

L'image de X est donc l'ensemble des valeurs que cette v.a. peut prendre. Cet ensemble peut être fini ou dénombrable : on parlera alors de v.a. discrète. Sinon, on parlera de v.a. continue.

Etant donné un réel x, on peut définir les événements suivants,

$$\begin{aligned}[X = x] &= \{e \in \mathcal{S} | X(e) = x\} \\ [X \leq x] &= \{e \in \mathcal{S} | X(e) \leq x\}\end{aligned}$$

Pour des facilités d'écriture, ces événements sont notés de manière spéciale, mais ce sont des événements bien définis. Le premier correspond à l'ensemble des résultats possibles de l'expérience dont la valeur (par X) est égale à x. Le second aux issues possibles de l'expérience dont la valeur est inférieure à x.

2.3.1 Variables aléatoires continues

La fonction de répartition (CDF) de la v.a. X est définie par,

$$F_X(x) = P[X \leq x]$$

$F_X(x)$ représente ainsi la probabilité que l'événement $[X \leq x]$ se produise. Evidemment on a $\lim_{x\to\infty} F_X(x) = 1$. De plus, si X est à valeurs positives, $F_X(0) = 0$. La fonction de répartition d'une v.a. positive est donc croissante entre 0 et 1.

La probabilité d'un intervalle (i.e. $x_1 \leq X(e) \leq x_2$) est :

$$P[x_1 \leq X \leq x_2] = F_X(x_2) - F_X(x_1)$$

La densité de probabilité (pdf) d'une v.a. continue est définie par,

$$f_X(x) = \frac{dF_X(x)}{dx} = \lim_{dx\to 0} \frac{F_X(x+dx) - F_X(x)}{dx}$$

On a évidemment $F_X(x) = \int_0^x f_X(x)\,dx$ et $\int_0^\infty f_X(x)\,dx = 1$. Pour dx "petit", $f_X(x)\,dx$ correspond à $P[x \leq X \leq x + dx]$.

2.3.2 Variables aléatoires discrètes

Si X est une v.a. discrète prenant ses valeurs dans $x_1, x_2, \ldots$, alors on peut définir sa fonction de répartition par,

$$F_X(x) = \sum_{x_i \leq x} p_i$$

où $p_i = P[X = x_i]$. $F_X(x)$ est donc une fonction en escalier ayant des sauts de hauteurs p_i aux points x_i et telle que $F_X(x) = 0$ pour $x < x_1$ et $F_X(x) = 1$ pour $x > x_n$.

L'équivalent de la densité de probabilité est la fonction de masse de probabilité définie par,

$$p(x) = P[X = x] = \begin{cases} p_i & \text{si } \ x = x_i \\ 0 & \text{sinon} \end{cases}$$

2.3.3 Cas de plusieurs variables aléatoires

Si X et Y sont deux v.a., on peut définir leur CDF jointe et leur pdf jointe par,

$$F_{X,Y}(x,y) = P[X \leq x,\, Y \leq y] \quad \text{et} \quad f_{X,Y}(x,y) = \frac{\partial}{\partial x}\frac{\partial}{\partial y} F_{X,Y}(x,y)$$

Ainsi si dxet dy sont "petits", $f(x,y)\,dx\,dy$ correspond à $P[x \leq X \leq x + dx, y \leq Y \leq y + dy]$.

La densité marginale de Y est donnée par,

$$f_Y(y) = \int_0^\infty f_{X,Y}(x,y)\,dx$$

Les v.a. X et Y sont dites indépendantes si les événements $[X \leq x]$ et $[Y \leq y]$ sont indépendants quels que soient x et y. Cela se traduit par,

$$F_{X,Y}(x,y) = F_X(x)\,F_Y(y) \quad \text{et} \quad f_{X,Y}(x,y) = f_X(x)\,f_Y(y)$$

De la même façon, on peut définir la pdf conditionnelle de X sachant Y,

$$f_{X|Y}(x,y) = \frac{f_{X,Y}(x,y)}{f_Y(y)}$$

Si les deux v.a. sont indépendantes, on a $f_{X|Y}(x,y) = f_X(x)$.

2.3.4 Moments d'une variable aléatoire

Soit un entier $k \geq 1$. Le moment d'ordre k d'une v.a. continue est donné par,

$$E[X^k] = \int_0^\infty x^k\, f_X(x)\,dx$$

et pour une v.a. discrète,

$$E[X^k] = \sum_{i=1}^{\infty} x_i^k\, p_i$$

Les deux premiers moments ($k = 1, 2$) jouent un rôle particulier. Le premier est appelé l'espérance mathématique (ou moyenne) :

$$E[X] = \int_0^\infty x\, f_X(x)\,dx \text{ (v.a. continue)} \quad \text{ou} \quad E[X] = \sum_{i=1}^{\infty} x_i\, p_i \text{ (v.a. discrète)}$$

L'espérance est linéaire : $E[aX + bY] = a\,E[x] + b\,E[Y]$ pour a et b réels. Par contre, $E[X\,Y] = E[X]\,E[Y]$ n'est vérifiée que si X et Y sont des v.a.

indépendantes.

Le second moment intervient dans le calcul de la variance :

$$V[X] = E[(X - E[X])^2] = E[X^2] - E[X]^2$$

La variance vérifie $V[c\,X] = c^2\,V[X]$. Attention : $V[X+Y] = V[X] + V[Y]$ n'est vrai que si les v.a. X et Y sont indépendantes ! Enfin, signalons qu'on utilise souvent l'écart type d'une v.a. X, qui correspond à la racine carrée de la variance.

2.4 Quelques distributions de probabilités courantes

Jusqu'ici, nous avons considéré une expérience, ses résultats possibles et les événements correspondants. Nous avons défini la probabilité d'un événement comme la fréquence à laquelle il se produirait si l'expérience était renouvelée un nombre infini de fois. La notion de v.a. nous a permis d'associer une valeur à chaque résultat possible. La fonction de répartition, la densité de probabilité, l'espérance mathématique et la variance que nous avons vues ensuite, sont simplement des outils commodes pour caractériser la probabilité d'obtenir telle ou telle valeur.

Dans la pratique, quand on doit modéliser un problème, la démarche suivie est souvent inverse. On commence par identifier les v.a. intervenants dans le problème. Puis on suppose qu'elles suivent une certaine distribution de probabilités. Cette hypothèse devra ensuite être validée par des expérimentations, une fois l'analyse effectuée.

Dans cette optique de modélisation, certaines distributions de probabilités jouent un rôle important. Tout simplement parce qu'on les rencontre souvent dans les systèmes réels. Nous en présentons ici quelques unes.

2.4.1 Distributions discrètes

Distribution de Bernoulli

Elle modélise une expérience avec deux résultats possibles : succès (avec probabilité p) ou échec (avec probabilité $q = 1 - p$). Une v.a. X suivant la distribution de Bernoulli est définie par,

$$X = \begin{cases} 1 & \text{avec probabilité } p \\ 0 & \text{avec probabilité } q = 1 - p \end{cases}$$

On obtient immédiatement $E[X] = p$ et $V[X] = pq$. La v.a. X peut par exemple modéliser un système avec deux états : ON et OFF.

Distribution Binomiale

Soit X une v.a. représentant le nombre de succès après n expériences de Bernoulli indépendantes,

$$X = \sum_{i=1}^{n} Y_i \quad \text{où} \quad Y_i \sim Bernouilli(p)$$

On a immédiatement $E[X] = n\,E[Y_i] = np$ et $V[X] = n\,V[Y_i] = np\,(1-p)$. La probabilité d'avoir i succès est de plus donnée par :

$$p_i = C_i^n\, p_i\,(1-p)^{n-i}$$

Distribution Géométrique

Soit X une v.a. représentant le nombre de tentatives avant le premiers succès dans une série d'expériences de Bernoulli indépendantes :

$$p_i = P[X = i] = (1-p)^{i-1}\,p$$

On en déduit le nombre moyen de tentatives $E[X] = 1/p$ et sa variance $V[X] = (1-p)/p^2$. La probabilité qu'il faille plus de n tentatives avant le premier succès est donnée par,

$$P[X > n] = \sum_{i=n+1}^{\infty} p_i = (1-p)^n$$

La distribution géométrique est sans mémoire :

$$\begin{aligned}
P[X > i+j \,|\, X > i] &= \frac{P[X > i+j\,,\, X > i]}{P[X > i]} \\
&= \frac{P[X > i+j]}{P[X > i]} \\
&= \frac{(1-p)^{i+j}}{(1-p)^i} \\
&= (1-p)^j = P[X > j]
\end{aligned}$$

Autrement dit, le fait de savoir qu'il y a déjà eu i tentatives infructueuses, ne change pas la probabilité qu'il en faille encore plus de j pour avoir un succès. Les tentatives passées n'ont pas d'influence sur les tentatives futures.

Distribution de Poisson

Une v.a. discrète X à valeurs entières positives suit la loi de Poisson de paramètre $\lambda\,t$ (deux réels positifs fixés) si sa distribution est donnée par,

$$p_i = P[X = i] = \frac{(\lambda\, t)^i}{i!}\, exp(-\lambda\, t)$$

On peut montrer que la loi de Poisson correspond à la limite de la loi binomiale $B(n, p)$ quand on fixe le nombre moyen de succès $np = \lambda\, t$ et qu'on fait tendre n vers l'infini. En conséquence, on peut voir le paramètre t comme une durée et λ comme un taux de succès. Ainsi, une v.a. $X \sim Poisson(\lambda t)$ permet de représenter le nombre d'événements sur un intervalle de longueur t pour un processus de Poisson d'intensité λ. Le nombre moyen d'événements est donné par $E[X] = \lambda\, t$ et sa variance est $V[X] = \lambda\, t$. De plus,

- La probabilité d'un événement (un succès) durant dt est $\lambda\, dt$,
- La probabilité de plus d'un événement durant dt est négligeable,
- Le nombre d'événements se produisant sur deux intervalles de temps disjoints sont indépendants.

Enfin, on peut signaler que le processus de Poisson est stable par agrégation, c'est à dire si $X_1 \sim Poisson(\lambda_1\, t)$ et $X_2 \sim Poisson(\lambda_2\, t)$, alors $X_1 + X_2 \sim Poisson((\lambda_1 + \lambda_2)\, t)$.

La loi de Poisson va nous servir par exemple à représenter le nombre d'appels téléphoniques sur un intervalle de temps.

2.4.2 Distributions continues

Distribution uniforme

Une v.a. $X \sim U(a, b)$ a une pdf constante sur l'intervalle $[a, b]$,

$$f_X(x) = \begin{cases} 1/(b-a) & a < x < b \\ 0 & \text{sinon} \end{cases}$$

Sa moyenne est $E[X] = (a + b)/2$ et sa variance est $V[X] = (b - a)^2/12$.

Distribution exponentielle

Une v.a. continue X suit une loi exponentielle de paramètre μ si sa CDF est donnée par (cf. figure 2.1),

$$F_X(x) = \begin{cases} 1 - exp(-\mu x) & x \geq 0 \\ 0 & x < 0 \end{cases}$$

La pdf est alors,

$$f_X(x) = \begin{cases} \mu\, exp(-\mu x) & x \geq 0 \\ 0 & x < 0 \end{cases}$$

La loi exponentielle va par exemple nous servir à modéliser la durée d'un appel téléphonique, ou encore la durée entre l'arrivée de deux appels. Sa moyenne et sa variance sont données par,

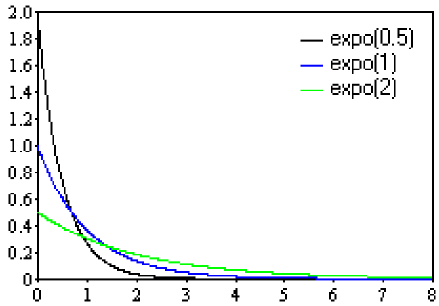

FIGURE 2.1 – Fonction de répartition de la distribution exponentielle.

$$E[X] = 1/\mu \quad \text{et} \quad V[X] = 1/\mu^2$$

Comme la loi géométrique dans le cas discret, la loi exponentielle a la propriété d'être sans mémoire. Supposons que X représente la durée aléatoire d'un appel téléphonique. Quelle est la probabilité que l'appel dure au moins x secondes de plus, sachant qu'il a déjà duré t secondes.

$$\begin{aligned} P[X > t + x \,|\, X > t] &= \frac{P[X > t + x\,,\, X > t]}{P[X > t]} \\ &= \frac{P[X > t + x]}{P[X > t]} \\ &= \frac{exp(-\mu(t + x))}{exp(-\mu t)} \\ &= exp(-\mu x) = P[X > x] \end{aligned}$$

Autrement dit, le fait de savoir que l'appel a déjà duré t secondes, ne change pas la probabilité qu'il en faille encore plus de x.

Remarquons enfin que le paramètre μ s'interprète comme un taux de terminaison. En effet, si on sait qu'un appel a déjà duré t secondes, quelle est la probabilité qu'il se termine en dt secondes supplémentaires :

$$P[X < t{+}dt \,|X > t] = P[X < dt] = 1{-}exp(\mu dt) = 1{-}(1{-}\mu dt{+}(\mu dt)^2/2\ldots) \approx \mu\, dt$$

Ainsi μ est le taux de terminaison de l'appel par unité de temps, $1/\mu$ étant la durée moyenne de l'appel.

Distribution normale

La pdf d'une v.a. gaussienne de paramètres μ et σ^2 est (cf. figure 2.2),

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} exp(-\frac{1}{2}(x-\mu)^2/\sigma^2)$$

Les paramètres μ et σ^2 correspondent à la moyenne et à la variance : $E[X] = \mu$ et $V[X] = \sigma^2$.

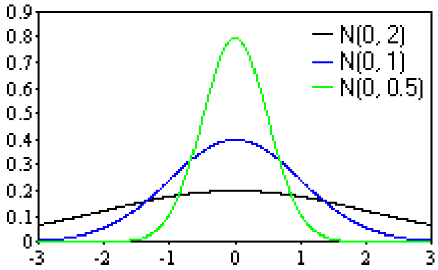

FIGURE 2.2 – Distribution normale.

Outre ses nombreuses applications dans le domaine de la physique, cette distribution est surtout importante car elle intervient dans le théorème central limite. Illustrons ceci par un exemple. Supposons que la durée d'un appel soit une v.a. de moyenne μ et de variance σ^2. On mesure n fois la durée d'un appel pour obtenir n v.a. $X_1, \ldots, X_n$ indépendantes et identiquement distribuées (i.i.d.). On s'intéresse à la moyenne empirique de ces mesures,

$$W_n = \frac{1}{n}\sum_{i=1}^{n} X_i$$

W_n est une v.a. dont l'espérance et la variance sont données par $E[W_n] = \mu$ et $V[W_n] = \sigma^2/n$. Intuitivement, on voit donc que plus on va avoir de mesures, plus la moyenne arithmétique sera "proche" de l'espérance de la durée d'un appel, avec une variance de mesure qui décroît linéairement. La loi faible des grands nombres précise cette intuition en affirmant que, pour une précision ϵ donnée, on a,

$$\lim_{n\to\infty} P[W_n - \mu \geq \epsilon] = 0$$

et la loi forte des grands nombres nous dit que W_n converge vers μ avec probabilité 1. Le théorème central limite nous permet de mesurer cette convergence. Il affirme que pour n suffisamment grand,

$$\frac{\sqrt{n}}{\sigma}(W_n - \mu) \sim N(0,1)$$

Si on connaît la moyenne μ et la variance σ de la durée d'un appel, on peut ainsi évaluer l'erreur commise en utilisant la moyenne empirique pour estimer la moyenne. On verra que ce résultat nous sera utile pour établir un critère d'arrêt pour des simulations événementielles.

CHAPITRE 3

Chaînes de Markov

3.1 Processus Stochastiques

En général les systèmes que nous considérons sont des systèmes dynamiques (dont l'état évolue au cours du temps) faisant intervenir de l'aléatoire. Par exemple, la longueur d'une file d'attente évolue au cours du temps en fonction de l'arrivée aléatoire des clients et du temps aléatoire qu'il faut pour les servir.

Un processus stochastique $(Y_t)_{t\geq 0}$ est une séquence de variables aléatoires indexées par le temps t. Le temps t peut être continu (resp. discret, i.e. $t = 1, 2, \ldots$) et l'on parle alors de processus à temps continu (resp. à temps discret). Pour t fixé, Y_t est simplement une v.a. prenant ses valeurs dans un certain espace d'états suivant une distribution qui peut dépendre de t. L'ensemble des états possibles pour le processus peut lui-même être discret ($Y_t = 0, 1, \ldots$) ou continu.

Pour caractériser un processus stochastique, nous ne considérerons ici que les statistiques de premier ordre, à savoir la distribution $F_{Y_t}(y) = P[Y_t \leq y]$ de l'état Y_t au cours du temps, et sa moyenne $E[Y_t]$. Une caractérisation au second ordre impliquerait d'établir la covariance du processus entre deux instants s et t. Cette covariance, donnée par,

$$R_{s,t} = E[\ (X_t - E[X_t])\,(X_s - E[X_s])\ ]$$

permet d'étudier la relation entre les états du système à deux instants différents.

Si on tire aléatoirement une valeur y_t de Y_t pour chaque valeur de t, la séquence de valeurs obtenues est appelée une trajectoire du système. Cela représente une succession au cours du temps d'états que le système peut prendre.

On peut s'intéresser au comportement transitoire du processus, c'est à dire à l'évolution des fonctions de répartition $F_{Y_t}(x)$ au cours du temps t. En général, on s'intéresse plutôt au comportement stationnaire (à l'équilibre) qui est obtenu si $F_{Y_t}(x)$ converge vers une distribution unique $F_Y(x)$ indépendante du temps quand $t \to \infty$. La figure 3.1 illustre la convergence des densités de probabilités d'un processus stochastique $(Y_i)_{i=1,2,\ldots}$ vers une densité stationnaire. Attention : le régime permanent peut n'exister que sous certaines conditions et il faut soigneusement établir ces conditions quand on étudie un processus stochastique.

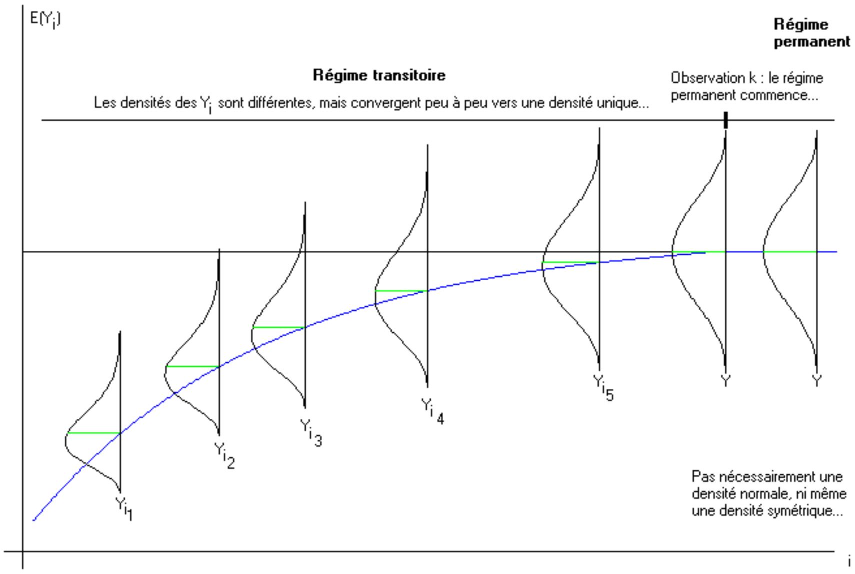

FIGURE 3.1 – Convergence de l'espérance et des densités de probabilités vers leurs valeurs stationnaires.

3.2 Processus Markovien

Soit un processus stochastique $(X_t)_{t\geq 0}$. Considérons des instants $t_1 < \ldots < t_n$ situés dans le passé. On suppose connaître l'état x_i du processus à l'instant t_i, $i = 1, \ldots, n$. L'état x_n est l'état courant et on s'intéresse à son état futur x_{n+1} à l'instant t_{n+1}. Le processus (X_t) a la propriété Markovienne si,

$$P[X_{t_{n+1}} \leq x_{n+1} \mid X_{t_n} = x_n, \ldots, X_{t_1} = x_1] = P[X_{t_{n+1}} \leq x_{n+1} \mid X_{t_n} = x_n]$$

et ce quelque soit n et les instants considérés. Concrètement, cette équation signifie que le processus est Markovien si son état dans le futur ne dépend que de l'état présent, et pas de ses états passés. Autrement dit, l'état courant contient toute l'information permettant de caractériser l'évolution future du processus.

Un processus Markovien à espace d'états discret est appelée une chaîne de Markov.

3.3 Chaîne de Markov à temps discret

Une chaîne de Markov à temps discret est un processus Markovien $(X_n)_{n=1,2,...}$ à temps discret dont l'espace d'états est discret, i.e. $X_n \in \{0, 1, 2, \ldots\}$ (le nombre d'états peut être fini).

3.3.1 Probabilités de transition

Une chaîne de Markov peut être caractérisée par ses probabilités de transition en une étape,

$$p_{i,j} = P[X_{n+1} = j \mid X_n = i]$$

Ainsi, $p_{i,j}$ est la probabilité de passer de i à j en un pas de temps. On notera que ces probabilités ne dépendent pas de l'instant n considéré : on dit dans ce cas que la chaîne de Markov est homogène.

Une chaîne de Markov à temps discret peut être représenté par un graphe tel que celui de la figure 3.2 (où p est un paramètre, e.g. $p = 1/3$). Les noeuds du graphe représentent les états que peut prendre le processus et les arcs représentent les transitions entre états. Le poids de l'arc (i, j) correspond à $p_{i,j}$.

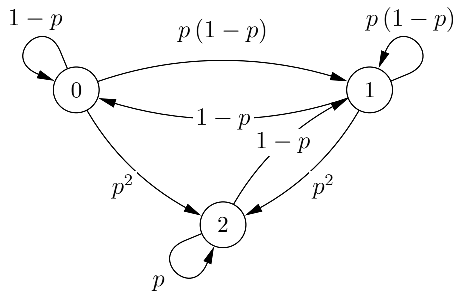

FIGURE 3.2 – Graphe de transitions d'une chaîne de Markov.

On appelle matrice des probabilités de transition la matrice $P = [p_{i,j}]$,

$$P = \begin{pmatrix} p_{0,0} & p_{0,1} & p_{0,2} & \cdots \\ p_{1,0} & p_{1,1} & p_{1,2} & \cdots \\ p_{2,0} & p_{2,1} & p_{2,2} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

La ligne i de cette matrice donne donc la probabilité de passer de l'état i vers tout autre état j (y compris l'état i lui même) en un pas de temps. Evidemment, comme une transition est effectuée à chaque pas de temps, la somme des éléments d'une ligne est égal à 1. On parle de matrice stochastique.

Si le système passe de l'état i à l'état j en deux étapes, c'est qu'il est passé par un état k (qui peut éventuellement être i ou j). On a donc :

$$p_{i,j}^{(2)} = \sum_k p_{i,k}\, p_{k,j}$$

Clairement $p_{i,j}^{(2)}$ est l'élément (i, j) de la matrice $P^2 = P\,P$. On peut ainsi établir que les probabilités de transition $p_{i,j}^{(n)}$ en n étapes s'obtiennent avec P^n, i.e. en mettant la matrice P à la puissance n.

Puisque on a toujours $P^n = P^m\, P^{n-m}$, il vient :

$$p_{i,j}^{(n)} = \sum_k p_{i,k}^m\, p_{k,j}^{n-m} \quad 0 \leq m \leq n$$

Cette équation est connue sous le nom d'équation de Chapman-Kolmogorov. Elle exprime que pour passer de i à j en n étapes, il faut passer par un état k à l'étape m.

3.3.2 Probabilités d'état

Notons $\pi_i^{(n)} = P[X_n = i]$ la probabilité que la chaîne de Markov soit dans l'état i à l'instant n. Pour manipuler ces probabilités d'états, on utilise en général le vecteur $\pi^{(n)} = \left[\pi_0^{(n)}, \pi_1^{(n)}, \ldots\right]$. Puisque la matrice P donne les probabilités de transition en une étape, on a,

$$\pi^{(n)} = \pi^{(n-1)}\, P$$

d'où l'on déduit récursivement que,

$$\pi^{(n)} = \pi^{(0)}\, P^n$$

Reprenons l'exemple de la figure 3.2 avec $p = 1/3$. Dans ce cas, la matrice P est donnée par (aux arrondis près),

$$P = \begin{pmatrix} 0.6666 & 0.2222 & 0.1111 \\ 0.6666 & 0.2222 & 0.1111 \\ 0 & 0.6666 & 0.3333 \end{pmatrix}$$

la matrice de transitions à 2 pas est donnée par,

$$P^2 = \begin{pmatrix} 0.5926 & 0.2716 & 0.1358 \\ 0.5926 & 0.2716 & 0.1358 \\ 0.4444 & 0.3704 & 0.1852 \end{pmatrix}$$

Partant de l'état i à l'instant initial, la distribution de l'état au bout de 2 transitions est donnée par la ligne i de P^2. Ainsi, si à l'instant initial la chaîne de Markov est dans l'état 0 de façon certaine, i.e. $\pi^{(0)} = [1, 0, 0]$, alors on sait que la distribution de probabilités de l'état à l'instant 2 est donnée par $\pi^{(2)} = [0.5926, 0.2716, 0.1358]$.

Si on continue le calcul jusqu'à l'ordre 8 (8 transitions), on a :

$$P^8 = \begin{pmatrix} 0.5714 & 0.2857 & 0.1429 \\ 0.5714 & 0.2857 & 0.1429 \\ 0.5714 & 0.2857 & 0.1429 \end{pmatrix}$$

Puisque toutes les lignes sont identiques, on voit que quel que soit l'état de départ, la distribution de probabilités atteinte est la même. On peut le confirmer par le calcul suivant,

$$\begin{aligned} \pi^{(8)} &= \left[\pi_0^{(0)}, \pi_1^{(0)}, \pi_2^{(0)}\right] P^8 \\ &= \Big[0.5714 \times (\pi_0^{(0)} + \pi_1^{(0)} + \pi_2^{(0)}), \\ &\quad 0.2857 \times (\pi_0^{(0)} + \pi_1^{(0)} + \pi_2^{(0)}), \\ &\quad 0.1429 \times (\pi_0^{(0)} + \pi_1^{(0)} + \pi_2^{(0)})\Big] \\ &= [0.5714, 0.2857, 0.1429] \end{aligned}$$

La chaîne de Markov converge donc vers un état d'équilibre indépendant de l'état initial. Le processus a oublié son état initial. Dans la suite, nous allons nous intéresser aux conditions sous lesquelles cet état d'équilibre existe et à son calcul.

3.3.3 Classification des états

On dit que l'état i conduit à l'état j si la chaîne de Markov peut passer de i à j avec une probabilité non nulle, i.e. $\exists n, p_{i,j}^{(n)} > 0$. Si la relation est réciproque, on dit que les états i et j communiquent. Une chaîne de Markov dont tous les états communiquent entre eux est dite irréductible.

Un état est absorbant si le processus reste bloqué dans cet état, i.e. $p_{i,i} = 1$. On distingue de plus les états transitoires des états récurrents. Un état est transitoire si, partant de cet état, la probabilité d'y revenir est strictement inférieure à 1. Si cette probabilité est égale à 1, l'état est récurent.

Soit i un état récurent. On note $T_{i,i}$ l'espérance du temps nécessaire pour revenir pour la première fois dans cet état quand $X_0 = i$. L'état i est dit positif récurent si $T_{i,i} < \infty$, et nul récurent si $T_{i,i} = \infty$.

Si, partant d'un état i, le temps nécessaire pour revenir dans cet état pour la première fois est necéssairement un multiple d'un entier d, cet état est dit périodique. Sinon, l'état est dit apériodique.

Un état positif récurent et apériodique est dit ergodique. Si tous les états sont ergodiques, la chaîne de Markov est ergodique.

Toutes ces définitions sont illustrées sur la figure 3.3.

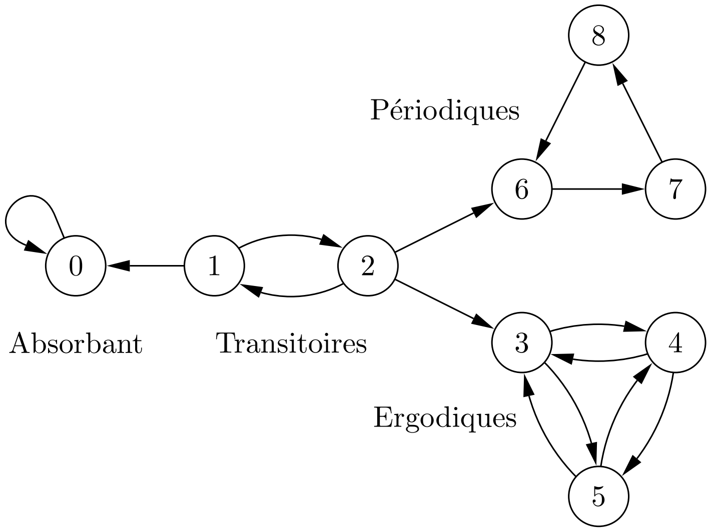

FIGURE 3.3 – Classification des états d'une chaîne de Markov.

3.3.4 Distribution stationnaire

Thèorème de Kolmogorov

Pour une chaîne de Markov irréductible et apériodique, la distribution stationnaire,

$$\pi_j = \lim_{n\to\infty} \pi_j^{(n)}$$

existe et est indépendante de l'état initial. π_j représente la proportion de temps que le processus passe dans l'état j. De plus, soit :

- Les états sont tous transitoires ou tous nuls récurrents, et dans les deux cas $\pi_j = 0, \forall j$,
- Les états sont tous positifs récurrents et la distribution stationnaire π (vecteur) est la solution des équations suivantes,

$$\pi = \pi\, P \quad \text{et} \quad \pi\, e^T = 1$$

où e est un vecteur ligne dont tous les composantes valent 1 et e^T est le vecteur colonne transposé. Ces équations peuvent également s'écrire :

$$\pi_j = \sum_i \pi_i\, p_{i,j} \quad \text{et} \quad \sum_j \pi_j = 1$$

Equations d'équilibre global

L'équation $\pi = \pi\, P$, ou $\pi_j = \sum_i \pi_i\, p_{i,j}$ pour $j = 0, 1, \ldots$, est souvent appelé la condition d'équilibre global. En effet, on a $\sum_i p_{j,i} = 1$ puisqu'il y a forcément une transition depuis l'état j, soit vers j lui même, soit vers un autre état. L'équation $\pi = \pi\, P$ implique donc,

$$\pi_j = \pi_j \sum_i p_{j,i} = \sum_i \pi_i\, p_{i,j} \quad j = 0, 1, \ldots$$

Le terme $p_{j,j}\, \pi_j$ apparaissant des deux côtés, on peut le supprimer pour obtenir :

$$\pi_j \sum_{i \neq j} p_{j,i} = \sum_{i \neq j} \pi_i\, p_{i,j} \quad j = 0, 1, \ldots$$

Cette équation exprime que la probabilité de sortir de l'état j vers un autre état est égale à la probabilité d'entrer dans l'état j en venant d'un autre état. C'est bien la notion d'équilibre (cf. figure 3.4).

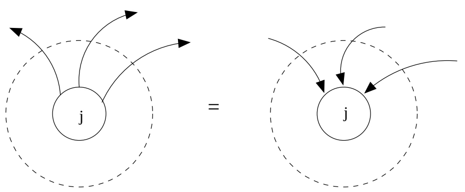

FIGURE 3.4 – Interprétation de l'équilibre global.

Exemple

Reprenons l'exemple de la figure 3.2 (avec p général). La condition d'équilibre de l'état 0 donne,

$$\pi_0 = (1-p)\,\pi_0 \; + \; (1-p)\,\pi_1 \quad \Rightarrow \quad \pi_0 = \frac{1-p}{p}\,\pi_1$$

La condition d'équilibre de l'état 1 donne,

$$\pi_1 = p(1-p)\,\pi_0 + p(1-p)\,\pi_1 + (1-p)\,\pi_2 \quad \Rightarrow \quad \pi_1 = \frac{1-p}{p}\,\pi_2\;,\; \pi_0 = \left(\frac{1-p}{p}\right)^2\,\pi_2$$

La condition de normalisation $\pi_0 + \pi_1 + \pi_2 = 1$ permet de déduire la valeur de π_2, et donc la solution stationnaire :

$$\pi = \left[\frac{(1-p)^2}{1-p(1-p)}, \frac{p(1-p)}{1-p(1-p)}, \frac{(p^2}{1-p(1-p)},\right]$$

qui, pour $p = 1/3$, donne bien $\pi = [0.5714, 0.2857, 0.1429]$.

3.4 Chaînes de Markov à temps continu

Par rapport au cas discret, les transitions peuvent survenir à tout instant. De plus, il n'y a plus de transition d'un état vers lui même.

Pour que le processus soit markovien, il faut que le temps de séjour dans un état soit sans mémoire : le temps restant ne dépend pas du temps déjà passé dans cet état. On a vu que la distribution exponentielle possède cette propriété, et c'est en fait la seule distribution continue ayant ce caractère "sans mémoire". Pour une chaîne de Markov continue, le temps de séjour dans un état est donc exponentiellement distribué.

Définissons le taux de transition de l'état i vers l'état j :

$$q_{i,j} = \lim_{\Delta t \to 0} \frac{P\left[X_{t+\Delta t} = j \mid X_t = i\right]}{\Delta t} \quad i \neq j$$

Quand le processus est dans l'état i, $q_{i,j}\,\Delta t$ représente ainsi la probabilité qu'une transition de i vers j survienne pendant la durée Δt. Le taux de sortie de l'état i est donc donnée par,

$$q_i = \sum_{j \neq i} q_{i,j}$$

La durée de séjour dans l'état i est donc une v.a. qui suit une distribution exponentielle de paramètre q_i.

La matrice des taux de transitions est définie par,

$$Q = \begin{pmatrix} -q_0 & q_{0,1} & q_{0,2} & \cdots \\ q_{1,0} & -q_1 & q_{1,2} & \cdots \\ q_{2,0} & q_{2,1} & -q_2 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Soit $\pi(t) = [\pi_0(t), \pi_1(t), \ldots]$ la distribution de probabilités de l'état à l'instant t. Examinons l'évolution de $\pi_i(t)$ sur Δt. Au 1èr ordre,

$$\pi_i(t + \Delta t) = (1 - q_i\, \Delta t)\, \pi_i(t) \ + \ \sum_{j \neq i} q_{j,i} \Delta t\, \pi_j(t) \quad \forall i$$

On en déduit que,

$$\frac{\pi_i(t + \Delta t) - \pi_i(t)}{\Delta t} = -q_i\, \pi_i(t) \ + \ \sum_{j \neq i} q_{j,i}\, \pi_j(t) \quad \forall i$$

Quand $\Delta t \to 0$, cela donne :

$$\frac{d\pi_i(t)}{dt} = -q_i\, \pi_i(t) \ + \ \sum_{j \neq i} q_{j,i}\, \pi_j(t) \quad \forall i$$

Soit sous forme matricielle :

$$\frac{d\pi(t)}{dt} = \pi(t)\, Q$$

La solution de cette équation différentielle matricielle est donnée par,

$$\pi(t) = \pi(0)\, e^{Q\,t} \quad \text{où} \quad e^{Q\,t} = I \ + \ Qt \ + \ \frac{1}{2!} Q^2 t^2 \ + \ \ldots$$

Cette équation nous donne (au moins numériquement) la solution transitoire de la chaîne de Markov. Si elle existe, la solution stationnaire $\pi = \lim_{t \to \infty}$ est indépendante du temps, i.e. $\frac{d\pi(t)}{dt} = 0$. La solution stationnaire vérifie donc :

$$\pi\, Q = 0$$

Cette équation nous donne la condition d'équilibre global. En effet, elle est équivalente à :

$$\pi_j \sum_{i \neq j} q_{j,i} = \sum_{i \neq j} \pi_i\, q_{i,j} \quad j = 0, 1, \ldots$$

Autrement dit, à l'équilibre, la somme des taux de transition sortant d'un état j est égale à la somme des taux de transition qui y entrent.

Pour déterminer la solution stationnaire :
– utiliser la condition $\pi Q = 0$ pour exprimer tous les π_j, $j \neq 0$, en fonction de π_0,

– puis utiliser la normalisation des probabilités $\sum_i \pi_i = 1$, pour déterminer π_0 et donc les π_j, $j \neq 0$.

3.5 Processus de naissances et de morts

Les processus de naissances et de morts sont des cas particuliers de chaînes de Markov à temps continu très utilisés dans la théorie des files d'attente. Dans un processus de naissances et de morts, les transitions se font entre états adjacents : les transitions $i \to i+1$ sont les naissances, et les transitions $i \to i-1$ sont les morts. Dans la théorie des files d'attente, les naissances correspondent à des arrivées de clients (des appels ou des paquets par exemple) et les morts à leurs départs.

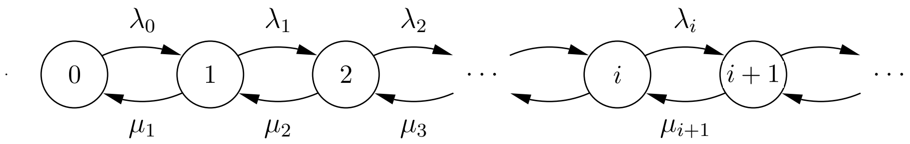

FIGURE 3.5 – Processus de naissances et de morts.

On notera λ_i le taux de naissance dans l'état i et μ_i le taux de mort, comme illustré sur la figure 3.5. Quand le processus est dans l'état i, la probabilité d'une naissance durant Δt est donc $\lambda_i \,\Delta t$ et celle d'une mort est $\mu_i \,\Delta t$ (temps de séjour exponentiellement distribué). Les taux de transitions $q_{i,j}$ sont donc donnés par :

$$q_{i,j} = \begin{cases} \lambda_i & \text{si} \quad j = i+1 \\ \mu_i & \text{si} \quad j = i-1 \\ 0 & \text{sinon} \end{cases}$$

Le générateur infinitésimal d'un processus de naissances et de morts est donc de la forme suivante :

$$Q = \begin{pmatrix} -\lambda_0 & \lambda_0 & 0 & \dots & \dots & \dots \\ \mu_1 & -(\lambda_1+\mu_1) & \lambda_1 & 0 & \dots & \dots \\ 0 & \mu_2 & -(\lambda_2+\mu_2) & \lambda_2 & 0 & \dots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \end{pmatrix}$$

3.5.1 Distribution stationnaire

Le vecteur de probabilités stationnaires π est solution de $\pi\, Q = 0$. On obtient ainsi,

$$\begin{array}{rcl} \mu_1 \pi_1 \; - \; \lambda_0 \pi_0 & = & 0 \\ \lambda_{i-1} \pi_{i-1} \; - \; (\lambda_i + \mu_i) \pi_i \; + \; \mu_{i+1} \pi_{i+1} & = & 0 \quad i \geq 1 \end{array}$$

Posons $v_i = \lambda_i \pi_i - \mu_{i+1}\pi_{i+1}$. Si on examine les deux équations ci-dessus, on voit que la première donne $v_0 = 0$ et la seconde $v_{i+1} - v_i = 0$. On en déduit que $v_i = 0$, $\forall i$, et donc que,

$$\pi_i = \frac{\lambda_{i-1}}{\mu_i} \pi_{i-1} \; = \; \frac{\lambda_{i-1} \lambda_{i-2}}{\mu_i \mu_{i-1}} \pi_{i-2} \; = \; \ldots \; = \; \prod_{k=0}^{i-1} \frac{\lambda_k}{\mu_{k+1}} \pi_0 \quad i \geq 1$$

On a donc tous les π_i en fonction de π_0. En utilisant la normalisation des probabilités, on obtient :

$$\sum_{i=0}^{\infty} \pi_i = 1 \quad \Rightarrow \pi_0 = \frac{1}{1 + \displaystyle\sum_{i=1}^{\infty} \prod_{k=0}^{i-1} \frac{\lambda_k}{\mu_{k+1}}}$$

3.5.2 Distribution transitoire

On suppose connaître la distribution $\pi(0)$ de l'état à l'instant 0, par exemple $\pi_k(0) = 1$ et $\pi_j(0) = 0$ pour $j \neq k$. On veut déterminer son évolution transitoire $\pi(t)$, et plus seulement sa valeur limite $\pi = \lim_{t \to \infty} \pi(t)$. On a vu que la distribution transitoire est solution de,

$$\frac{d\pi(t)}{dt} = \pi(t)\, Q$$

Pour un processus de naissances et de morts, cela donne :

$$\begin{array}{rcl} \dfrac{d\pi_0(t)}{dt} & = & \mu_1 \pi_1(t) \; - \; \lambda_0 \pi_0(t) \\ \dfrac{d\pi_i(t)}{dt} & = & \lambda_{i-1} \pi_{i-1}(t) \; - \; (\lambda_i + \mu_i) \pi_i(t) \; + \; \mu_{i+1} \pi_{i+1}(t) \quad i \geq 1 \end{array}$$

Ces équations permettent de déterminer $\pi(t)$ par intégration numérique. Soit Δt le pas d'intégration. Si on utilise la méthode d'Euler, on va ainsi déterminer les valeurs de $\pi(t)$ aux instants $t = \Delta t$, $t = 2\,\Delta t$, …, à partir de $\pi(0)$. Partant de $t = 0$, il suffit d'itérer sur les équations suivantes :

A/ $\pi_i(t + \Delta t) = \pi_i(t) + \Delta t \dfrac{d\pi_i(t)}{dt}$, $i \geq 0$

B/ $t = t + \Delta t$

où les valeurs des dérivées ont été données ci-dessus.

3.6 Processus de Poisson

Le processus de Poisson peut être défini comme un processus de naissances pur tel que $\lambda_i = \lambda$ et $\mu_i = 0$, $\forall i$ (cf. figure 3.6). Le paramètre λ, qui est indépendant de l'état du processus, correspond au taux de naissance.

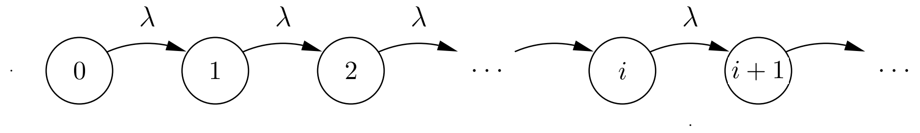

FIGURE 3.6 – Processus de Poisson.

Le processus de Poisson est un des modèles les plus utilisés dans la théorie des files d'attente. En effet, supposons que chaque naissance corresponde à l'arrivée d'un client (un appel ou un paquet par exemple) dans la file d'attente. Notons $N(t)$ l'état du processus de Poisson en supposant que $N(0) = 0$. On voit bien que $N(t) = i$ signifie qu'il y a eu i arrivées de clients entre 0 et t : le processus de Poisson permet de compter le nombre d'arrivées (cf. figure 3.7).

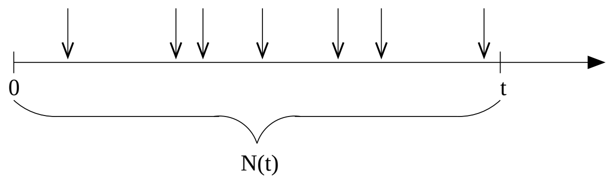

FIGURE 3.7 – Comptage du nombre d'arrivées.

Avec cette analogie, on voit bien que le temps de séjour dans un état correspond à la durée entre deux arrivées. Par conséquent :

- Le temps inter-arrivée est exponentiellement distribuée de paramètre λ, i.e,

$$P[\text{durée inter-arrivée} > t] = e^{-\lambda\, t}$$

- La durée moyenne entre deux arrivées est $1/\lambda$ et sa variance est $1/\lambda^2$.
- Pour dt "suffisamment petit", la probabilité d'avoir une arrivée durant dt est $\lambda\, dt$ et la probabilité de plus d'une arrivée est négligeable.

Toujours en supposant que $N(0) = 0$, intéressons nous à la probabilité d'avoir i arrivées entre 0 et t, c'est à dire à la probabilité $\pi_i(t)$ que $N(t) = i$. On peut pour cela utiliser les résultats sur la distribution transitoire d'un processus de naissances et de morts établis dans la section précédente. Avec $\lambda_i = \lambda$ et $\mu_i = 0$, $\forall i$, ces résultats deviennent :

$$\begin{aligned} \frac{d\pi_0(t)}{dt} &= -\lambda\,\pi_0(t) \\ \frac{d\pi_i(t)}{dt} &= \lambda\,\pi_{i-1}(t)\ -\ \lambda\,\pi_i(t) \quad i \geq 1 \end{aligned}$$

La première équation a pour solution évidente $\pi_0(t) = e^{-\lambda\,t}$. C'est la probabilité qu'il n'y ait aucune arrivée entre 0 et t. Pour $i = 1$, si on multiplie la seconde équation par $e^{\lambda t}$, on a :

$$\frac{d\pi_1(t)}{dt}\,e^{\lambda\,t} + \lambda\,e^{\lambda\,t}\,\pi_1(t) = \lambda\,e^{\lambda\,t}\,\pi_0(t) = \lambda\,e^{\lambda\,t}\,e^{-\lambda\,t} = \lambda$$

On reconnaît dans le membre de gauche la dérivée d'un produit :

$$\frac{d}{dt}\left[\pi_1(t)\,e^{\lambda\,t}\right] = \lambda$$

La solution de cette équation est évidemment :

$$\pi_1(t)\,e^{\lambda\,t} = \lambda t \quad \Rightarrow \pi_1(t) = \lambda t\,e^{-\lambda\,t}$$

C'est la probabilité d'avoir une seule arrivée entre 0 et t. En procédant par réccurence, on montre ainsi que,

$$\pi_i(t) = \frac{(\lambda\,t)^i}{i!}\,e^{-\lambda\,t}$$

On voit ainsi que le nombre d'arrivée entre 0 et t d'un processus de Poisson est déterminé par la distribution de Poisson.

L'hypothèse de durées inter-arrivée exponentiellement distribuées peut sembler a priori restrictives. Bien qu'elle ne soit pas toujours vérifiée, le processus de Poisson est souvent un très bon modèle pour beaucoup de phénomènes physiques. L'explication tient dans un résultat très fort démontré par Khintchine (théorème de Palm-Khintchine). En gros, ce résultat stipule que si on observe un processus d'arrivées qui, au niveau microscopique est constitué d'un très grand nombre de micro-processus générant un événement d'arrivée très rarement, alors le processus d'arrivée correspond à un processus de Poisson. C'est ce qui explique par exemple que l'on considère que les arrivées d'appels dans un réseau téléphoniques se font suivant un processus de Poisson (grand nombre d'abonnés).

Enfin, citons quelques propriétés importantes du processus de Poisson :

- **Stabilité par agrégation :** la superposition de deux flots d'arrivées Poissonniens de taux λ_1 et λ_2 donne un processus de Poisson de taux $\lambda_1 + \lambda_2$.

- **Stabilité par désagrégation :** Si un flot d'arrivées Poissonien de taux λ est divisé en n sous-flots, chaque arrivée étant affectée au sous-flot i avec la probabilité p_i, alors les n sous-flots sont Poissoniens de paramètres $p_i\lambda$, $i = 1 \dots, n$.

- **Proprité PASTA :** Poisson Arrival See Time Averages (les arrivées poissonniennes "voient" les moyennes temporelles). Concrétement, cela signifie que si un système à la probabilité π_s d'être dans l'état s à l'équilibre, des arrivées suivant un processus de Poisson vont voir en moyenne le système dans l'état s.

CHAPITRE 4

Théorie des files d'attente

4.1 Introduction

La théorie des files d'attente est l'un des formalismes les plus utilisés pour la modélisation des réseaux de communication. De manière générale, la théorie des files d'attente s'intéresse à des systèmes dans lesquels des "clients" partagent une ou plusieurs ressources. Les dates d'arrivée des clients et la quantité de service qu'ils demandent sont imprévisibles. Ils sont en concurrence pour l'accès aux ressources. L'objectif est de déterminer les performances du système et notamment la qualité de service offerte aux clients en fonction de la capacité des ressources et de la politique de partage mise en oeuvre.

Dans les cas les plus simples, une file d'attente peut être représentée comme sur la figure 4.1. Le système est constitué d'un serveur muni d'une file d'attente. Des clients arrivent et demandent une certaine durée de service. Quand un client ne peut être servi parce que le serveur est déjà occupé, ce client est mis en attente dans la file d'attente.

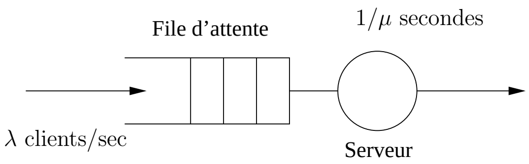

FIGURE 4.1 – File d'attente.

L'étude de cette file d'attente correspond à celle du processus stochastique gouvernant le nombre de clients dans le système. L'aléatoire réside ici dans la variabilité des durées services et dans l'imprévisibilité des arrivées de clients. Les instants d'arrivées constituent un processus stochastique ; plus précisément, on considère que le temps qui sépare les instants d'arrivée de deux clients consécutifs sont des variables aléatoires (v.a.) indépendantes et identiquement distribuées (i.i.d.). De même, les temps de service requis par les clients constituent des v.a. i.i.d. L'évolution du nombre de clients est donc aléatoire, et dépend du choix de la discipline de service, c'est à dire, de l'ordre dans lequel les client seront servis.

Dans ce chapitre, nous commençons par introduire les concepts de base de la théorie des files d'attente. Nous étudions ensuite le cas simple de quelques files d'attente markoviennes. Puis, nous présentons quelques résultats sur des systèmes non markoviens. Enfin, nous énoncons les principaux théorèmes de la théorie des réseaux de files d'attente.

4.2 Les concepts de base des files d'attente

Nous introduisons ci-dessous les principaux concepts utilisés dans l'analyse des files d'attente (cf. Figure 4.2) :
- Arrivées, services et départs,
- Discipline de service,
- Notation de Kendall,
- Nombre de clients,
- Temps de réponse,
- Courbes de charge,.
- Périodes d'activité et d'inactivité,

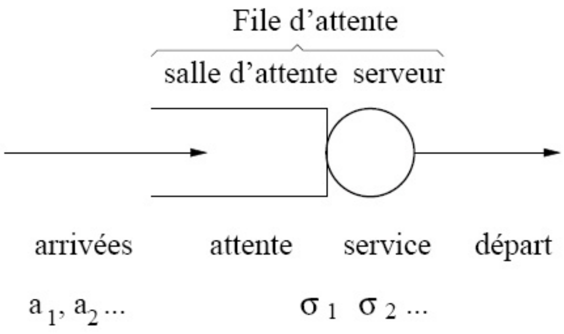

FIGURE 4.2 – Representation graphique d'une file d'attente

Arrivées, services et départs

Les processus d'arrivée est décrit par la suite des dates successives d'arrivée :

$$a_1, a_2, \ldots, a_n, a_{n+1}.$$

où a_n est la date d'arrivée du nème client. La différence entre deux dates d'arrivée consécutives est appelée inter-arrivée. On note par τ_n la n-ième inter-arrivée : $\tau_n = a_{n+1} - a_n$. De façon équivalente, on peut donc décrire le processus d'arrivée par la suite des inter-arrivées,

$$\tau_1, \tau_2, \ldots, \tau_n, \tau_{n+1}, \ldots$$

Definition 1 ***Taux d'arrivée*** *Le taux d'arrivée des clients (également appelé intensité des arrivées), est le nombre moyen de clients qui arrivent dans le système par unité de temps.*

On écrit :

$$\lambda = \lim_{t \to \infty} \frac{A(t)}{t},$$

où :

$$A(t) = \max_n \{n | a_n \leq t\}.$$

A chaque client entrant dans la file d'attente est associé une demande de service. Le service demandé par le n-ième client sera noté σ_n.

Après son arrivée dans le système à la date a_n, le n-ème client passe éventuellement un certain temps dans la file d'attente en attendant son tour. Puis ce client entre en service pour une durée σ_n. Quand il a reçu la quantité de service demandée, ce client quitte le système. On notera d_n sa date de départ. Le processus aléatoire des départs est alors défini par la suite,

$$d_1, d_2, \ldots, d_n, d_{n+1}, \ldots$$

Discipline de service

La discipline de service précise la politique de partage de la capacité du (des) serveur(s) entre les clients. Deux grandes politiques peuvent être utilisées : soit la capacité d'un serveur est complètement dédiée à un seul client à la fois, soit elle est partagée entre tous les clients présents.

Dans le premier cas, la discipline de service décrit l'ordre dans lequel les clients sont servis. Les disciplines les plus courantes sont alors :

- FIFO (First-In-First-Out) ou, en français, PAPS (Premier Arrivé Premier Servi) : service dans l'ordre défini par les arrivées. On rencontre également l'acronyme FCFS (First-Come- First-Served).

- LIFO (Last-In-First-Out) ou, en français, DAPS (Dernier Arrivé Premier Servi),
- Priorités : Il y a plusieurs classes de clients, chacune avec un niveau de priorité. C'est le client qui a la priorité la plus haute parmi ceux présents qui est servi en premier. S'il y en a plusieurs, c'est le premier arrivé (discipline FIFO par classe).

Concernant les politiques partageant la capacité d'un serveur entre les clients présents, nous n'étudierons ici que la discipline PS (Processor Sharing) ou Politique à temps partagé. Avec cette discipline, la capacité du serveur est partagée équitablement entre tous les clients présents.

La notation de Kendall

Il est clair qu'en fonction des hypothèses faites sur le processus des arrivées, sur la loi de service et sur la discpline de service, il existe une multitude de modèles de files d'attente. La notation de Kendall constitue un moyen commode de préciser le modèle étudié.Cette notation a la forme suivante :

$$A/S/P/K/D$$

où :

A et S désignent respectivement la loi des inter-arrivées et la loi de service. A et S peuvent prendre les valeurs suivantes (liste non exhaustive) :
- M pour la distribution exponentielle (M comme Markov) ; Si A=M, le processus d'arrivée est un processus de Poisson.
- D pour la distribution déterministe ; Si A=D, le processus d'arrivée est périodique, tandis que si S=D, les durées de service demandées correspondent à une constante.
- G (ou GI) pour une distribution générale (arbitraire). Un résultat établi dans le cas A=G sera donc valable quel que soit le processus des arrivées. De même, un résultat établi dans le cas S=G sera valable pour toute loi de service.

P désigne le nombre de serveurs. P peut prendre n'importe quelle valeur entière, y compris l'infini. Par défaut, P=1.

K désigne la capacité du système, c'est-à-dire le nombre maximum de clients qui peuvent être présents simultanément, en comptant le(s) client(s) en service. K peut prendre n'importe quelle valeur supérieure à P, y compris l'infini. Par défaut, $K = \infty$.

D désigne la discipline de service. Dans ce cours, nous étudierons surtout les cas où D=FIFO et D=PS. Par défaut, D=FIFO.

Pour donner un exemple, la file $M/M/1$ correspond à la file $M/M/1/\infty/FIFO$, c'est à dire une file d'attente avec un serveur, des arrivées suivant un processus de Poisson, des demandes de service exponentiellement distribuées, une file d'attente de capacité infinie et une discipline FIFO. Dans la suite de ce chapitre, nous allons étudier attentivement quelques exemples de tels systèmes.

Client	Date d'arrivée	durée de service
1	0	4
2	2	8
3	6	4
4	8	4
5	15	2

TABLE 4.1 – Arrivées et services

Nombre de clients

L'uns des principales fonctions que l'on utilise pour étudier le comportement d'une file d'attente est $N(t)$, le nombre de clients présents dans la file d'attente à l'instant $t \geq 0$. $N(t)$ est une fonction à valeurs dans l'ensemble $\{0, 1, \ldots, K\}$, où K est la capacité de la file d'attente, donc $N(t)$ est une courbe en escalier.

Illustrons ce concept sur un exemple. Considérons une file d'attente avec un serveur, de capacité d'attente illimitée et avec la discipline de service FIFO. Supposons que les dates d'arrivée et les durées de service des 5 premiers clients soient celles données dans le Tableau 4.1.

L'évolution du nombre de clients correspond alors à la courbe représentée sur la Figure 4.3. On a représenté en dessous les instants de début et de fin de service des cinq clients.

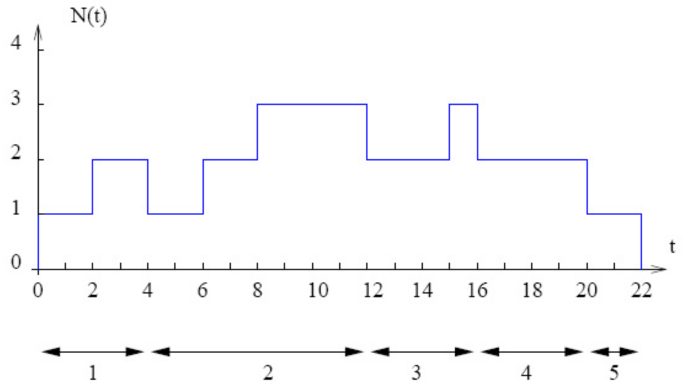

FIGURE 4.3 – Courbe du nombre de clients

Temps de réponse

Le temps de réponse (ou temps de séjour) d'un client dans une file d'attente est la différence entre sa date de départ et sa date d'arrivée. Avec les notations introduites précédemment, le temps de réponse du client n vaut

$$T_n = d_n - a_n.$$

Généralement, le temps de réponse se décompose en d'une part le temps d'attente (temps passé dans la file d'attente) et la durée de service (temps passé dans un des serveurs).

4.2.1 La formule de Little

La formule de Little est l'un des résultats les plus utilisés dans la théorie des files d'attente. Cette formule permet d'établir un lien entre le nombre moyen de clients et le temps de réponse moyen des clients.

Proposition 1 *Soit λ le taux d'arrivée dans une file d'attente à l'équilibre. Soit $\overline{N} = E[X]$ le nombre moyen de clients dans le système et soit $\overline{T}$ le temps de réponse moyen d'un client. Alors*

$$\overline{N} = \lambda \overline{T}.$$

Cette formule peut s'appliquer quand le système est stable. Elle est très utile car souvent on connaît $\overline{N}$ et λ. Elle dit que le nombre moyen de clients dans le système est égal au taux moyen d'arrivée multiplié par le temps moyen passé dans le système. Notons que ce résultat est extrêmement général puisqu'il ne fait aucune hypothèse sur le processus des arrivées, sur la loi de service, sur le nombre de serveurs ou encore sur la discipline de service.

Une explication intuitive est la suivante. Le rapport $\overline{N}/\overline{T}$ est égal au taux de départ, qui doit être égal au taux d'arrivée λ car le système se trouve en équilibre.

Charge et courbe de charge

La charge (workload) du système à l'instant t, $W(t)$, est la quantité de travail qui reste à effectuer à cet instant. Une courbe de charge est un graphe de la fonction $W(t)$. Quand on la construit, on a le choix de l'unité de travail. Celle-ci dépend en fait de l'application. Par exemple dans les réseaux de communication la quantité de travail pourra être des octets ou des bits. En général, on se ramène à des secondes en utilisant la vitesse du serveur, mesurée en nombre d'unités de travail par unité de temps. Par exemple, dans un réseau de communication :

$$\frac{N octets}{C octets/sec} = \frac{N}{C} sec.$$

La courbe de charge correspondant au Tableau 4.1 est représenté dans la Figure 4.4.

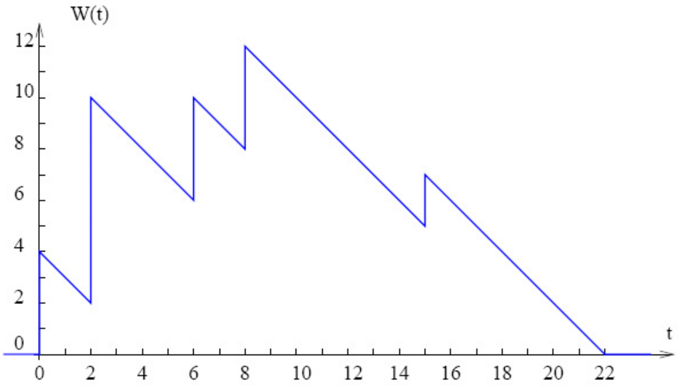

FIGURE 4.4 – Courbe de la charge $W(t)$

On voit sur cette figure les propriétés principales de cette fonction :

- C'est une courbe en dents de scie.
- Quand $W(t) > 0$, $W(t)$ décroît à vitesse constante sauf quand il y a une arrivée. La pente de la courbe est égale à la vitesse du serveur. Si on a choisi la même unité en abscisse et en ordonnée, la pente vaut exactement -1.
- Elle croît par sauts aux instants d'arrivée. L'amplitude du saut est égal au temps de service requis par le client qui vient d'arriver.
- Quand $W(t)$ atteint 0, elle reste constamment nulle jusqu'à la prochaine arrivée.

La courbe de charge $W(t)$ et le nombre de clients $N(t)$ sont deux manières complémentaires de décrire l'évolution d'une file d'attente. Bien évidemment, ces deux courbes ont des relations étroites. D'abord nous allons observer comment on peut relier ces quantités entre elles graphiquement. Sur la Figure 4.5, on a représenté les courbes $W(t)$ et $N(t)$ pour l'exemple considéré précedemment. On voit comment on peut "lire" les dates de départ d_n sur la courbe de $W(t)$ en prolongeant les parties décroissantes de la courbe jusqu'à l'axe horizontal. Ceci s'explique de la manière suivante. Quand un client arrive à la date a_n, il trouve dans la file d'attente une quantité de travail $W(a_n)$. Comme la politique de service est FIFO, ce client entrera en service quand cette charge sera servie, soit donc à la date $a_n + W(a_n)$. Il finira son propre service à la date

$$d_n = a_n + W(a_n) + \sigma_n.$$

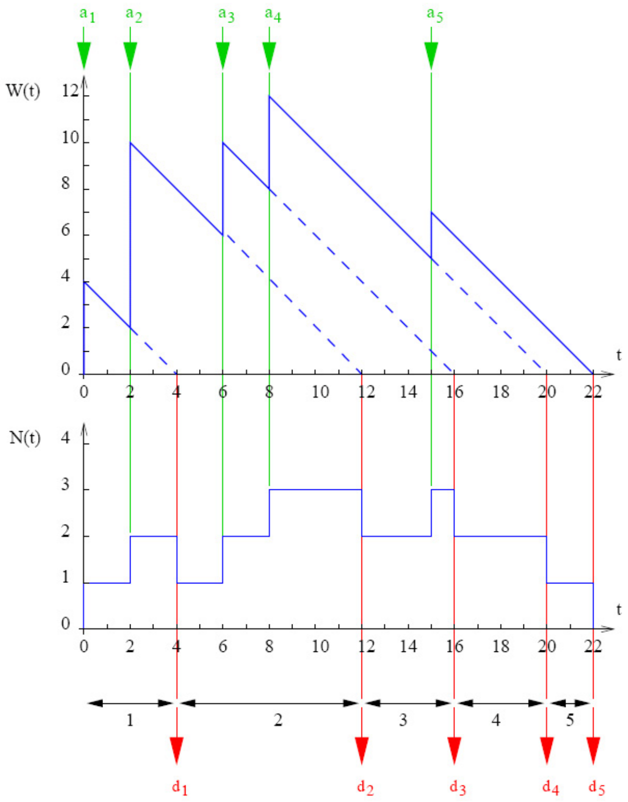

FIGURE 4.5 – Relation entre les courbes $W(t)$ et $N(t)$

Cette date se lit donc directement sur la courbe de $W(t)$. On note que le raisonnement qui précède ne vaut que si la discipline de service est FIFO.

Périodes d'activité et d'inactivité

Une période d'activité (busy period) du serveur (ou des serveurs) est un intervalle de temps pendant lequel des clients sont servis continûment. De façon similaire, une période d'inactivité (idle period) est un intervalle de temps pendant lequel aucun service n'est délivré. Il est clair qu'une file d'attente suit une succession de périodes d'activité et d'inactivité.

Dans le cas des disciplines que nous considérons le serveur travaille quand, et seulement quand il y a des clients. Donc les périodes d'activité sont les intervalles où $W(t) > 0$ et $N(t) > 0$. On note que la durée des périodes d'activité et inactivité ne dépend pas de la discipline. On dit que la discipline est "work conserving".

4.3 Files d'attente Markoviennes

Dans cette partie, nous utilisons les résultats obtenus dans le chapitre précédent sur les processus de naissances et de morts pour résoudre quelques modèles simples mais importants de files d'attente.

4.3.1 La file $M/M/1$

Dans cette file les clients arrivent selon un processus de Poisson de taux λ (les inter-arrivées sont donc exponentiellement distribuées de moyenne $1/\lambda$ secondes). Chaque client requiert une quantité de service aléatoire suivant une loi exponentielle de paramètre μ, c'est à dire que la durée moyenne d'un service est $1/\mu$ secondes. Il y a un seul serveur et les clients sont servis un par un dans l'ordre de leur arrivée (FIFO). La capacité de la file d'attente est infinie.

Soit $N(t)$ le nombre de clients à l'instant t. Le processus $\{N(t), t \geq 0\}$ prend ses valeurs dans $\{0, 1, 2, \ldots\}$. Les inter-arrivées et les durées de service étant exponentiellement distribuées, le temps de séjour dans un état $N(t) = i$ est lui aussi exponentiellement distribué. De plus, les transitions se font vers les états $N(t) = i - 1$ (départ d'un client) et $N(t) = i + 1$ (arrivée d'un client). Il s'en suit que le processus $\{N(t), t \geq 0\}$ est un processus de naissances et de morts dont le diagramme de transition est représenté sur la figure 4.6.

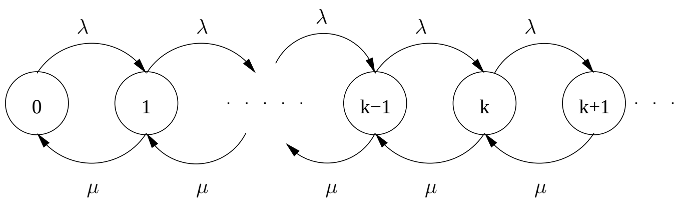

FIGURE 4.6 – Diagramme de transition de la file M/M/1.

Soit π_k, $k \geq 0$ la probabilité stationnaire qu'il y ait k clients dans le système, si elle existe. D'après les résultats obtenus sur les processus de naissances et de morts, on sait que si le régime d'équilibre existe, on a :

$$\pi_k = \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} \pi_0 = \left(\frac{\lambda}{\mu}\right)^k \pi_0 \quad k \geq 1$$

où,

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{\infty} (\lambda/\mu)^k} = 1 - \lambda/\mu$$

Définissons $\rho = \lambda/\mu$ l'intensité du traffic, c'est à dire la quantité moyenne de travail qui arrive au système par unité de temps. On a alors le résultat suivant.

Proposition 2 *Soit une file d'attente $M/M/1$ avec un taux d'arrivée λ et durée moyenne de service $1/\mu$. Si $\rho < 1$*

$$\pi_k = (1 - \rho)\,\rho^k \tag{4.1}$$

pour tout $k \geq 0$.

La condition $\rho < 1$ est dite condition de stabilité de la file $M/M/1$. Elle s'interprète par la nécessité qu'en moyenne, la quantité de clients qui arrive au système dans une unité de temps doit être inférieure à la quantité moyenne de clients que le serveur peut servir dans ce même temps.

Une conclusion importante que l'on peut déduire du résultat précédent est que la distribution stationnaire ne dépend de λ et μ qu'au travers de leur rapport, la charge ρ.

Sous la condition de stabilité, le nombre moyen de clients à l'équilibre est donné par,

$$E[N] = \sum_{k=0}^{\infty} k\pi_k = (1 - \rho) \sum_{k=0}^{\infty} k\rho^k = \frac{\rho}{1 - \rho}.$$

Observons que $E[N] \to \infty$ si $\rho \to 1$, donc dans la pratique, si le système n'est pas stable le nombre moyen de clients divergera.

On peut aussi s'intéresser à la probabilité qu'il y ait plus de K clients dans la file d'attente :

$$P(N \geq K) = \sum_{k=K}^{\infty} \pi_k == (1 - \rho) \sum_{k=K}^{\infty} \rho^k = \rho^K.$$

En utilisant la loi de Little, on obtient directement le temps de réponse moyen des clients,

$$T = E[N]/\lambda = \frac{1}{\mu - \lambda}.$$

Quel est le taux de sortie des clients (débit R) d'une file d'attente $M/M/1$ à l'équilibre ? Puisque on est à l'équilibre, tout ce qui entre devrait sortir, et par consequent la réponse devrait être $R = \lambda$. Vérifions-le :

$$R = (1 - \pi_0)\mu = \lambda.$$

4.3.2 La file $M/M/1/K$

Dans la pratique, les files d'attente sont toujours de capacité finie, et le dimensionnement des buffers est un problème important. Ainsi, un nouveau client sera perdu (rejeté) lorsqu'il trouvera le système plein en arrivant.

Une file $M/M/1/K$ est représentée sur la figure 4.7. Le fonctionnement de cette file d'attente est similaire à celui de la file $M/M/1$, sauf en ce qui concerne le rejet des clients quand le nombre de clients dans le système atteint K (y compris celui en service). S'il y a K clients dans le système et qu'un nouveau client se présente, ce dernier sera rejeté.

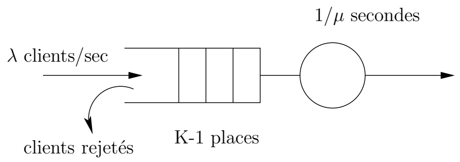

FIGURE 4.7 – File M/M/1/K.

Observons que pour cette file d'attente il n'y a pas de condition de stabilité. Le nombre de clients présents dans le système étant au maximum K, il ne peut pas diverger vers l'infini. Si le débit d'arrivée des clients λ est supérieur à la capacité de traitement μ du serveur ($\rho > 1$), alors la distribution stationnaire aura tendance à charger les états proches de K.

De façon évidente, le nombre de clients dans une file $M/M/1/K$ correspond à un processus de naissances et de morts, dont le diagramme de transitionn est représenté sur la figure 4.8.

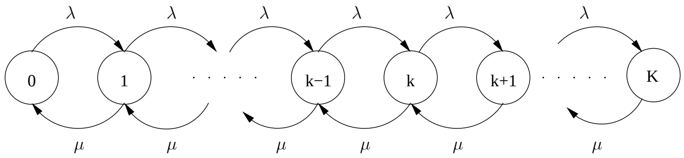

FIGURE 4.8 – Diagramme de transition de la file M/M/1/K.

En utilisant là encore les résultats obtenus sur les processus de naissances et de morts, on obtient,

$$\pi_k = \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} \pi_0 = \rho^k \pi_0 \quad k \geq 1$$

où,

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{K} \rho^k} = \begin{cases} \frac{1-\rho}{1-\rho^{K+1}} & \rho \neq 1 \\ \frac{1}{K+1} & \rho = 1 \end{cases}$$

Proposition 3 *Soit une file d'attente $M/M/1/K$ avec un taux d'arrivée λ et durée moyenne de service $1/\mu$. Si $\rho \neq 1$*

$$\pi_k = \frac{(1-\rho)\rho^k}{1-\rho^{K+1}} \quad k = 0, 1, \ldots, K \tag{4.2}$$

Si $\rho = 1$

$$\pi_k = \frac{1}{K-1} \quad k = 0, 1, \ldots, K \tag{4.3}$$

Enfin, notons que la probabilité de rejet est $\pi(K)$ (proprieté PASTA).

4.3.3 La file $M/M/C$

Dans cette file d'attente, il y a un nombre C de serveurs et la file d'attente à une capacité infinie (cf figure 4.9). Si lorsque un client arrive il y a au moins un serveur disponible, le client entre en service immédiatement. Dans le cas contraire, il est placé dans la file d'attente.

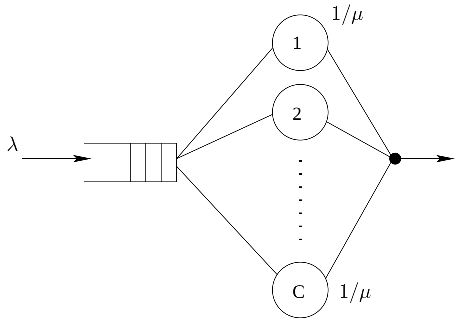

FIGURE 4.9 – File M/M/C.

Soit à nouveau $\{N(t), t \geq 0\}$ le nombre de clients dans le système. $N(t)$ est là encore un processus de naissances et de morts prenant ses valeurs dans $\{0, 1, 2, \ldots\}$. Soit i un état arbitraire. Le taux de naissance est $\lambda_i = \lambda$, $\forall i \geq 0$. Le taux de mort est $\mu_i = min(i\mu, C\mu)$. Avec ces valeurs de λ_i et μ_i, on obtient imédiatement le résultat suivant avec les mêmes techniques que précédemment.

Proposition 4 *Soit une file d'attente $M/M/C$ avec un taux d'arrivée λ et une durée moyenne de service $1/\mu$. Si $\rho = \lambda/\mu < C$*

$$\pi_i = \begin{cases} \pi_0 \frac{\rho^i}{i!} & \text{pour } i = 0, 1, 2, \ldots, C \\ \pi_0 \frac{\rho^i C^{C-i}}{C!} & \text{pour } i \geq C \end{cases}$$

où

$$\pi_0 = \left(\sum_{i=0}^{C-1} \frac{\rho^i}{i!} + \frac{\rho^C}{C!(1-\rho/C)} \right)^{-1}. \tag{4.4}$$

La probabilité qu'un client soit placé dans la file d'attente en arrivant est,

$$\begin{aligned} P(attente) &= \sum_{i=C}^{\infty} \pi(i) = \sum_{i=C}^{\infty} \pi(0) \frac{\rho^i C^{C-i}}{C!} \\ &= \frac{\left(\frac{\rho^C}{C!}\right)\left(\frac{1}{1-\rho/C}\right)}{\left(\sum_{i=0}^{C-1} \frac{\rho^i}{i!} + \frac{\rho^C}{C!(1-\rho/C)}\right)} \end{aligned}$$

Cette probabilité est largement utilisé dans le dimensionnement des centres d'appels (hotlines) par exemple. Elle donne la probabilité que il n'y ait aucun serveur disponible pour un appel qui arrive. Elle est connue sous le nom de formule d'Erlang-C.

Le cas $C = \infty$

Dans ce cas, il y a un nombre infini de serveurs. Un client entre donc toujours immédiatement en service à son arrivée. La résolution du processus de naissances et de morts associé donne :

$$\pi_i = \pi_0 \frac{\rho^i}{i!}.$$

Avec $1 = \sum_{i=0}^{\infty} \pi_i$, on obtient $\pi_0 = e^{-\rho}$. Par conséquent la distribution stationnaire de la file $M/M/\infty$ est

$$\pi_i = \frac{\rho^i}{i!} e^{-\rho}, \quad i \geq 0.$$

Cette distribution est identique à une distribution de Poisson de paramètre ρ. Cette distribution existe toujours, et donc il n'y a pas de condition de stabilité pour la file $M/M/\infty$. Il se trouve que cette distribution de Poisson de paramètre ρ est aussi la distribution stationnaire de la file $M/GI/1$. Donc le résultat ne depend pas de la distribution du temps de service, mais uniquement de sa moyenne. On parle d'insensibilité.

4.3.4 La file $M/M/C/C$

Dans ce modèle, il y a C serveurs mais pas de file d'attente (cf figure 4.10). Lorsqu'un client arrive, s'il y a au moins un serveur disponible, le client entre directement en service. Sinon, le client est rejeté.

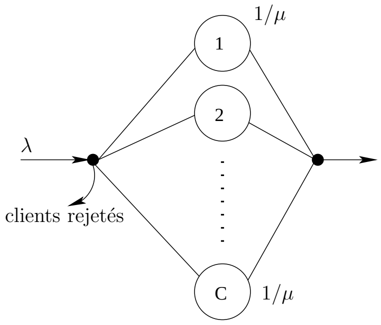

FIGURE 4.10 – File M/M/C/C.

Le nombre de clients dans le système peut être modelisé comme un processus de naissance et de morts. Le taux d'arrivée dans un état $i < C$ est $\lambda_i = \lambda$ tandis que $\lambda_C = 0$. Le taux de mort est $\mu_i = i\mu$ dans l'état $i = 0, 1, \ldots, C$.

Proposition 5 *Soit une file d'attente $M/M/C/C$ avec un taux d'arrivée λ et une durée moyenne de service $1/\mu$. La distribution stationnaire est :*

$$\pi_i = \pi_0 \frac{\rho^i}{i!} \quad i = 0, 1, \ldots, C \tag{4.5}$$

où,

$$\pi_0 = \frac{1}{\sum_{i=0}^{C} \rho^i / i!}.$$

Comme on le verra dans la seconde partie du cours, ce système est d'un grand intérêt en téléphonie. En particulier, π_C donne la probabilité que tous les serveurs soient occupés, c'est à dire la probabilité qu'une arrivée soit rejetée (propriété PASTA). Elle est donnée par :

$$\pi_C = \frac{\rho^C / C!}{\sum_{j=0}^{C} \rho^j / j!}$$

Cette formule est connue sous le nom de formule d'Erlang-B. Signalons de plus que de façon similaire à la file $M/M/\infty$, la Proposition 5 est valable pour n'importe quelle distribution du temps de service (insensibilité).

4.4 Exemples d'application

4.4.1 Comparaison de systèmes

Une entreprise fournissant des services web désire renouveler son parc de serveurs informatiques. Elle a le choix entre trois options illustrées sur la figure 4.11 :

- L'option A consiste à utiliser un seul serveur de capacité 2μ,
- L'option B consiste à utiliser deux serveurs de capacité μ en parallèle, chacun avec sa propre file d'attente, et à transmettre 50% des requêtes vers chaque serveur,
- L'option C consiste à utiliser deux serveurs de capacité μ en parallèle, mais partageant cette fois-ci la même file d'attente. Dans ce cas, s'il y a une requête en attente de traitement, elle est transmise vers le premier serveur qui se libère.

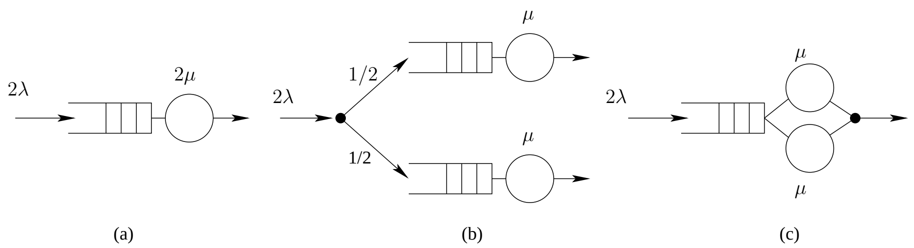

FIGURE 4.11 – Les trois options envisagées.

Dans chacune des options envisagées, les files d'attente sont de capacités infinies et sont gérées suivant la discipline FIFO. On suppose que les requêtes arrivent suivant un processus de Poisson de taux $2\,\lambda$. Une fois prise en charge par un serveur, la durée de traitement d'une requête est aléatoire et suit une distribution exponentielle de moyenne $\frac{1}{2\mu}$ dans l'option A, et de moyenne $\frac{1}{\mu}$ dans les options B et C.

On désire déterminer la configuration qui permet de minimiser le temps de réponse moyen aux requêtes. On remarquera que cette comparaison est équitable dans la mesure où le taux d'utilisation d'un serveur, dans les trois configurations envisagées, est toujours $\nu = \lambda/\mu$. On supposera dans la suite que $\nu < 1$.

Etude de l'option A

Cette option correspond à une file d'attente $M/M/1$ avec un taux d'arrivée 2λ, un taux de service 2μ, et donc un taux d'utilisation $\nu = 2\lambda/(2\mu)$. Le nombre moyen N_A de requêtes présentes dans le système (en attente ou en traitement) et le temps de séjour moyen T_A d'une requête sont donnés par,

$$N_A = \frac{\nu}{1-\nu} \quad \text{et} \quad T_A = N_A/(2\lambda) = \frac{1}{2\mu\,(1-\nu)}$$

Etude de l'option B

Le processus de Poisson étant stable par désagrégation, le processus d'arrivée des requêtes sur chacun des deux serveurs est un processus de Poisson de taux λ. Chacun des serveurs peut donc être modélisé par une file $M/M/1$, de taux d'arrivée λ, de taux de service μ et de taux d'utilisation $\nu = \lambda/\mu$. Le temps de séjour moyen T_B^i d'une requête sur le serveur i est donc,

$$T_B^i = \frac{1}{\mu - \lambda} = \frac{1}{\mu(1-\nu)} \quad i = 1, 2$$

Le temps de séjour moyen T_B d'une requête dans la configuration B est,

$$T_B = \frac{T_B^1 + T_B^2}{2} = \frac{1}{\mu(1-\nu)}$$

On voit ainsi que $T_A = T_B/2$. Autrement dit, le temps de réponse moyen de l'option A est deux fois plus court que celui de l'option B ! Intuitivement, ceci s'explique par le fait que dans l'option B, quand une des deux files est vide, la moitié de la capacité n'est pas utilisée. Au contraire, dans l'option A, la totalité de la capacité est utilisée dès lors qu'il y a au moins une requête à servir.

Etude de l'option C

Cette configuration peut être modélisée par une file d'attente $M/M/2/\infty$. En utilisant les résultats obtenus sur la file $M/M/C$ avec $C = 2$ et $\rho = 2\lambda/\mu = 2\nu$, on obtient,

$$\pi_i = \begin{cases} \pi_0 \frac{\rho^i}{i!} & i = 0, 1 \\ \pi_0 \, \rho^i \, \frac{2^{2-i}}{2!} & i \geq 2 \end{cases}$$

Pour $i \geq 2$, on a ainsi :

$$\pi_i = \pi_0 \, (2\nu)^i \, \frac{2^{2-i}}{2} = \pi_0 \, \nu^i \, 2^{i+2-i-1} = 2\pi_0 \nu^i$$

Vu que $\pi_1 = 2\pi_0\nu$, on en déduit que $\pi_i = 2\pi_0\nu^i$ pour $i \geq 1$. Avec la condition de normalisation des probabilités, on a alors,

$$\pi_0 = \left(1 + 2 \sum_{i=1}^{\infty} \nu^i\right)^{-1} = \left(1 + 2(\frac{1}{1-\nu} - 1)\right)^{-1} = \frac{1-\nu}{1+\nu}$$

Le nombre moyen de requêtes N_C présentes dans le système est alors,

$$N_C = \sum_{i=1}^{\infty} i\pi_i = 2\pi_0\nu \sum_{i=1}^{\infty} i\nu^{i-1} = 2\,\frac{1-\nu}{1+\nu}\nu\frac{1}{(1-\nu)^2} = \frac{2\nu}{1-\nu^2}$$

Avec la loi de Little, on a le temps de séjour moyen T_C d'une requête,

$$T_C = N_C/(2\lambda) = \frac{1}{\mu\,(1-\nu^2)}$$

Comparons alors avec l'option A :

$$\frac{T_A}{T_C} = \frac{\mu(1-\nu^2)}{2\mu(1-\nu)} = \frac{1+\nu}{2}$$

Sachant que $\nu < 1$, on voit ainsi que $T_A < T_C$. L'explication intuitive est liée au fait que quand il y a un seul client dans le système, il est servi avec le débit μ dans l'option C, alors qu'il est servi avec le débit 2μ dans l'option A.On voit d'ailleurs avec l'équation ci-dessus que quand le système est proche de la saturation ($\nu \approx 1$), les deux options deviennent équivalentes.

4.4.2 Etude d'un mécanisme de token bucket

Un "token bucket" est un mécanisme de policing utilisé par les opérateurs de réseaux IP pour réguler le trafic en entrée de leur réseau et garantir qu'un client n´émet pas plus de trafic en moyenne que ce que prévoit son contrat. Le principe est illustré sur la figure 4.12.

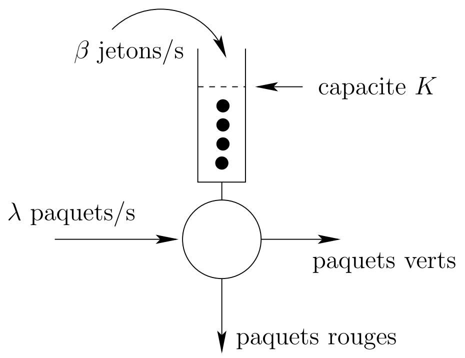

FIGURE 4.12 – Principe d'un token bucket.

Le fonctionnement est le suivant. Le token bucket correspond à une file d'attente FIFO dans laquelle des jetons sont placés régulièrement. On note β le taux d'arrivée des jetons. Le token bucket a d'autre part une capacité limitée : il ne peut contenir plus de K jetons. Lorsqu'un paquet IP arrive pour être transmis, il est marqué vert s'il reste des jetons dans le token bucket, et rouge sinon. Les paquets verts sont ensuite transmis de façon prioritaire, alors que les paquets rouges sont soit déclassés (priorité moins importante), soit directement jetés. Ainsi, le token bucket permet de garantir que le débit λ_{vert} du trafic vert ne pourra pas dépasser β paquets/seconde.

Pour simplifier, on suppose ici que les paquets IP arrivent suivant un processus de Poisson. La durée entre l'arrivée de deux paquets IP consécutifs est donc exponentiellement distribuée de moyenne $1/\lambda$. De la même façon, on supposera que les jetons arrivent à l'entrée du token bucket suivant un processus de Poisson de taux β. Enfin, on supposera que $\beta \neq \lambda$.

Supposons que le token bucket ne soit pas vide, et considérons le jeton le plus ancien dans cette file. Ce jeton va être consommé quand un paquet arrivera. Sachant que les inter-arrivées de paquets sont exponentiellement distribuées de moyenne $1/\lambda$, on en déduit que le jeton va rester en tête du token bucket une durée exponentiellement distribuée de moyenne $1/\lambda$.

On peut alors modéliser l'évolution du nombre de jetons du token bucket par une file d'attente $M/M/1/K$. Les arrivées de jeton se faisant suivant un processus de Poisson de taux β et les durées de service (attente du prochain paquet) étant exponentiellement distribuées de moyenne $1/\lambda$. le taux d'utilisation ρ dans ce modèle est $\rho = \beta/\lambda \neq 1$. On en déduit la probabilité π_0 que le token bucket soit vide,

$$\pi_0 = \frac{1-\rho}{1-\rho^{K+1}}$$

En utilisant la propriété PASTA du processus de Poisson, on voit ainsi que la probabilité qu'un paquet soit marqué vert est $1 - \pi_0$. On en déduit le débit λ_{vert} du trafic vert en paquets/seconde :

$$\lambda_{vert} = (1 - \pi_0)\lambda = \frac{1-\rho^K}{1-\rho^{K+1}}\,\beta$$

Observons que l'on a bien $\lambda_{vert} < \beta$ puisque pour $\rho \neq 1$ on a toujours $1 - \rho^K < 1 - \rho^{K+1}$.

4.5 Files d'attente à loi de service générale

4.5.1 La file M/G/1/FCFS

Pour cette file, le processus d'arrivée est toujours un processus de Poisson de taux λ, mais les durées de service suivent maintenant une loi générale :

$$P(S \leq x) = B(x),$$

dont la moyenne est notée $E[S] = 1/\mu$. Ici, le processus $N(t)$ n'est plus un processus de Markov, et les techniques précédentes ne s'appliquent plus. En fait, la probabilité d'avoir une transition de l'état $\{N = n\}$ vers l'état $\{N = n - 1\}$

dépend maintenant de la quantité de service que le client en service a déjà reçu. Cette information n'est pas incluse dans la variable $N(t)$, sauf dans le cas d'une loi de service exponentielle (grâce à sa propriété sans-mémoire) qui est donc le seul cas pour lequel le système est markovien.

Néanmoins, le nombre moyen de clients, le temps moyen d'attente et le temps moyen de réponse dans le système peuvent être calculés. En fait, même la distribution de ces variables peut être calculée en étudiant la chaîne de Markov incluse (en n'observant le système qu'aux instants de départs).

Dans cette partie, nous montrons en suivant une autre approche comment calculer le temps d'attente moyen dans une file $M/G/1/FCFS$. La formule obtenue est connue sous le nom de formule de Pollaczek-Khinchin.

Soit $N_q(t)$ le nombre de clients qui attendent dans la file à l'instant $t \geq 0$. On a la relation $N_q(t) = \max\{N(t) - 1, 0\}$. De plus, notons $R(t)$ le temps de service résiduel du client en service avant de quitter le système (voit Figure 4.13). On note que si $N(t) = 0$ alors $R(t) = 0$. Enfin, notons $\rho = \lambda/\mu$ le taux d'utilisation du système. On suppose ci-dessous que $\rho < 1$ et on admettra que c'est une condition nécessaire et suffisante pour que le régime stationnaire existe.

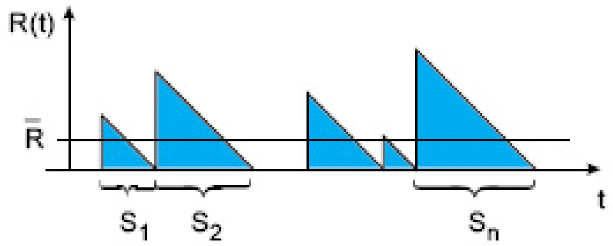

FIGURE 4.13 – Temps de service résiduel

Grâce à la propriété PASTA, nous avons la relation

$$E[W] = E[N_q]\frac{1}{\mu} + E[R]$$

L'idée clé est de remarquer que si l'on applique la loi de Little à la file d'attente (sans le serveur), on obtient :

$$E[W] = \lambda E[N_q],$$

et par conséquent,

$$E[W] = \frac{E[R]}{1-\rho},$$

Ainsi, il ne nous reste qu'à déterminer la valeur de $E[R]$ pour avoir une expression de $E[W]$. Pour cela, considérons la Figure 4.13 sur un intervalle de temps t très long. La valeur moyenne de $R(t)$ peut être calculée de la façon suivante :

$$\begin{aligned} E[R] &= \lim_{t\to\infty} \frac{1}{t}\int_0^t R(s)ds \approx \lim_{t\to\infty} \frac{1}{t}\sum_{i=1}^{n} \frac{1}{2}S_i^2 \\ &= \lim_{t\to\infty} \frac{n}{t}\frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}S_i^2 \\ &= \lambda\frac{1}{2}E[S^2]. \end{aligned}$$

On obtient ainsi le résultat suivant.

Proposition 6 *Si $\rho = \lambda/\mu < 1$, le temps d'attente moyen dans une file $M/G/1/FCFS$ est donné par l'expression suivante (formule P-K) :*

$$E[W] = \frac{\lambda E[S^2]}{2(1-\rho)}. \tag{4.6}$$

Comme conséquence directe, on obtient le temps de réponse moyen du système,

$$E[T] = \frac{1}{\mu} + E[W] = \frac{1}{\mu} + \frac{\lambda E[S^2]}{2(1-\rho)}$$

En utilisant la formule de Little, on en déduit le nombre moyen de clients en attente ainsi que le nombre moyen de clients dans le système :

$$E[N_q] = \frac{E[S^2]}{2(1-\rho)}, \quad \text{et} \quad E[N] = \rho + \frac{\lambda^2 E[S^2]}{2(1-\rho)}$$

On notera en particulier que le nombre moyen de clients, ainsi que le temps de séjour augmentent avec la "variabilité" de la distribution du temps de service.

A titre d'exemple, considérons tout d'abord le cas d'une file $M/M/1$ dont nous connaissons déjà la solution. La loi de service est exponentielle, ce qui implique que $E[S^2] = 2(\frac{1}{\mu})^2$. Par conséquent,

$$\begin{aligned} E[N] &= \rho + \frac{\lambda^2 E[S^2]}{2(1-\rho)} \\ &= \rho + \frac{\rho^2}{1-\rho} = \frac{\rho}{1-\rho}, \end{aligned}$$

et

$$\begin{aligned} E[T] &= \frac{1}{\mu} + \frac{\lambda E[S^2]}{2(1-\rho)} \\ &= \frac{1}{\mu}\left(1 + \frac{\rho}{1-\rho}\right) = \frac{1}{\mu}\frac{1}{1-\rho}. \end{aligned}$$

Considérons maintenant le cas d'une file $M/D/1$. La durée de service est une constante, et on a donc $E[S^2] = E[S]^2$. On en déduit,

$$E[N] = \rho + \frac{1}{2}\frac{\rho^2}{1-\rho},$$

et

$$E[T] = \frac{1}{\mu}\left(1 + \frac{1}{2}\frac{\rho}{1-\rho}\right).$$

4.5.2 La file M/G/1/PS

On considère un système avec un seul serveur de capacité μ dans lequel les arrivées se produisent suivant un processus de Poisson de taux λ. A chaque instant, le serveur partage équitablement sa capacité entre les clients présents. Ainsi, s'il y a n clients, chacun est servi avec le taux de service μ/n. On a le résultat suivant.

Proposition 7 *Si $\rho = \lambda/\mu < 1$, alors la distribution stationnaire de la file $M/G/1/PS$ existe et elle est donnée par,*

$$\pi_n = \pi_0\,\rho^n \ , \ n = 0, 1, \ldots \quad \textit{où} \quad \pi_0 = 1 - \rho$$

On peut faire deux remarques importantes. Tout d'abord, on remarque que cette distribution est identique à celle d'une file $M/M/1/FCFS$ de mêmes paramètres. Ensuite, cette distribution ne dépend pas réellement de la distribution des durées de service, mais seulement de sa moyenne.

4.6 Réseaux de files d'attente

Dans cette section, on présente les principaux résultats connus sur les réseaux de files d'attente. Les modèles à file d'attente unique ne permettent pas de modéliser des systèmes complexes. Par exemple, le problème du passage aux guichets d'une poste ne se résout pas avec une seule file, car chaque serveur à sa propre file d'attente. L'idée est donc de représenter chaque attente par une file simple, et de connecter ces files entre elles dans un réseau. Les questions que l'on se pose sont les mêmes : étant donnés les paramètres du modèle, quelle est la distribution du nombre de clients dans un site donné (ou à défaut sa moyenne), que valent les temps d'attente, quelles sont les conditions de stabilité du système, etc...

Le problème est évidemment plus compliqué que dans le cas d'une file simple. On ne peut pas espérer obtenir de résultats sans faire d'hypothèses plus restrictives (mais le moins possible) sur les lois d'arrivée, de service, le type des files simples qui constituent le réseau, le routage (la façon dont les clients se déplacent dans le reseau) ou la topologie de ce réseau. Dans certains cas que nous allons voir, les résultats obtenus dans la Section 4.3 peuvent être généralisés.

4.6.1 Les réseaux de Jackson

Les réseaux de Jackson ont pour particularité la spécification du routage des clients de manière aléatoire. Les paramètres d'un tel réseau sont :

- Le nombre N de stations
- Le vecteur $\vec{\lambda}^0 = (\lambda_1^0, \dots, \lambda_N^0)$ des taux d'arrivées extérieures dans chaque file,
- Le vecteur $(\mu_1, \dots, \mu_N)$ des taux de service,
- La matrice carrée $N \times N$ de routage interne $\mathbf{R}$, dont chaque composante $r_{i,j}$ est la probabilité qu'un client sortant du noeud i aille vers le noeud j. La probabilité que le client sorte du système est alors $r_{i,0} = 1 - \sum_j r_{i,j}$.

La matrice $\mathbf{R}$ est une matrice sous-stochastique, ce qui revient à dire qu'un client a toujours un ou plusieurs chemins qui mènent vers l'extérieur, et finit donc par en prendre un. On suppose donc qu'il existe au moins un $r_{i,0} > 0$.

Le processus $N(t) = (N_1(t), \dots, N_N(t))$ qui décrit le nombre de clients dans chaque file d'attente du reseau au temps t, est une chaîne de Markov en temps continu sur $\mathbb{N}^N$. En utilisant une notation vectorielle : $\mathbf{n} = (n_1, \dots, n_N)$ et $\mathbf{e}_i = (0, \dots, 1, \dots, 0)$ pour le i-ième vecteur unitaire, on décrit le générateur $\mathbf{Q}$ par le tableau de transitions :

$$\begin{array}{lcll} \mathbf{n} & \to & \mathbf{n} + \mathbf{e}_i & \text{avec taux} \quad \lambda_i^0 \\ \mathbf{n} & \to & \mathbf{n} - \mathbf{e}_i & \text{avec taux} \quad \mu_i r_{i,0} \\ \mathbf{n} & \to & \mathbf{n} - \mathbf{e}_i + \mathbf{e}_j & \text{avec taux} \quad \mu_i r_{i,j} \end{array}$$

On a alors :

Theorème 4.1 deJackson *Soit le vecteur $\vec{\lambda} = (\lambda_1, \dots, \lambda_N)$ solution du système linéaire :*

$$\vec{\lambda} = \vec{\lambda}^0 + \vec{\lambda}\mathbf{R}.$$

Si la condition de stabilité $\lambda_i < \mu_i$, $i = 1, \dots, N$, est satisfaite, alors le processus $\{N(t), t \geq 0\}$ admet une distribution stationnaire unique donnée par :

$$P(N_1 = n_1, \dots, N_N = n_N) = \prod_{i=1}^{N} \left(1 - \frac{\lambda_i}{\mu_i}\right) \left(\frac{\lambda_i}{\mu_i}\right)^{n_i} \tag{4.7}$$

La preuve du Théorème de Jackson et des théorèmes similaires peut se faire par verification directe : il suffit juste de verifier que la distribution proposée satisfait les equations d'équilibre. Dans le cas présent, on doit alors avoir pour tout vecteur $\mathbf{n}$:

$$\begin{aligned}\pi(\mathbf{n})\left(\sum_{i=0}^{N}\mu_i\mathbf{1}_{\{n_i>0\}}+\lambda_i^0\right) &= \sum_{j=0}^{N}\sum_{i=0}^{N}\pi(\mathbf{n}+\mathbf{e}_j-\mathbf{e}_i)\mu_j r_{j,i}\mathbf{1}_{\{n_i>0\}} \\ &\quad +\sum_{j=1}^{N}\pi(\mathbf{n}+\mathbf{e}_j)\mu_j r_{j,0}+\sum_{j=1}^{N}\pi(\mathbf{n}-\mathbf{e}_j)\lambda_j^0\mathbf{1}_{\{n_j>0\}}\end{aligned}$$

Ce qu'on peut vérifier avec un peu de patience en utilisant (4.7). L'unicité se démontre à partir du fait que la chaîne est irréductible.

D'après le Théorème 4.1, la distribution stationnaire de la file i est donnée par

$$P(n_i)=\left(1-\frac{\lambda_i}{\mu_i}\right)\left(\frac{\lambda_i}{\mu_i}\right)^{n_i}.$$

Donc la distribution stationnaire de la file i est identique à la distribution dans une file $M/M/1$ avec un taux d'arrivée λ_i et un temps de service moyen μ_i. La probabilité d'un état du réseau s'obtient comme produit des probabilités marginales : on dit que cette distribution a une “forme produit”. Ainsi, dans l'etat stationnaire, l'état de chaque file d'attente est celui d'une file $M/M/1$, indépendante du reste du réseau.

Exemple

Considérons l'exécution de jobs sur un ordinateur (cf. figure 4.14). Les jobs sont soumis par des utilisateurs supposés en grand nombre suivant un processus de Poisson de taux λ. Chaque job alterne ensuite entre une phase d'exécution sur la CPU, de durée exponentiellement distribuée de moyenne $1/\mu_1$, et une phase d'accès disque, de durée exponentiellement distribuée de moyenne $1/\mu_2$. Le nombre de phases d'exécution sur la CPU est géométriquement distribué de paramètre q, et donc un job s'exécute en moyenne $1/q$ fois sur la CPU.

Les équations de trafic s'écrivent :

$$\begin{aligned}\lambda_1 &= \lambda+\lambda_2 \\ \lambda_2 &= q\,\lambda_1\end{aligned}$$

On en déduit $\lambda_1=\lambda/(1-q)$ et $\lambda_2=\lambda q/(1-q)$. En posant $\rho_1=\lambda_1/\mu_1<1$ et $\rho_2=\lambda_2/\mu_2<1$, on obtient :

$$P(N_1=n_1,N_2=n_2)=(1-\rho_1)(1-\rho_2)\rho_1^{n_1}\rho_2^{n_2}\quad n_1\geq 0, n_2\geq 0$$

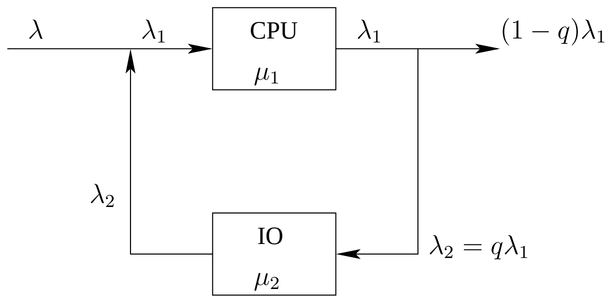

FIGURE 4.14 – Exécution de jobs sur un oridnateur.

On en déduit que $E[N_i] = \rho_i/(1-\rho_i)$, $i = 1, 2$ et le temps moyen d'exécution d'un job est alors donné par $(E[N_1] + E[N_2])/\lambda$.

4.6.2 Réseaux fermés

Il existe aussi des systèmes dans lesquels le nombre de clients est fixé. Un exemple classique est le problème machines/réparateurs, où les machines d'un site (nombre fini) sont susceptibles de tomber en panne et sont réparées par des personnes compétentes (en nombre fini aussi). Les réparateurs sont alors représentés par des serveurs et les machines par des clients, qui alternativement passent par une période de bon fonctionnement, puis vont se faire réparer par un réparateur libre.

Les paramètres d'un tel reseau sont :
- Le nombre N de stations
- Le vecteur $(\mu_1, \ldots, \mu_N)$ des taux de service,
- La matrice carrée $N \times N$ de routage interne $\mathbf{R}$.
- Le nombre K de clients présents.

Comme aucun client n'entre ni ne sort, le nombre de clients total dans le réseau est conservé. Comme précédemment, $\{N(t), t \geq 0\}$ est une CMTC, mais son espace d'états est (au plus) $\{\mathbf{n} \in \mathbb{N}^N | n_1 + \ldots + n_N = K\}$. On suppose que $\mathbf{R}$ est irréductible, et par consequent chaque client visite toutes les files d'attente du réseau.

On a le résultat suivant :

Theorème 4.2 Jackson pour les reseaux fermés. *Soit un reseau de Jackson fermé avec K clients et N serveurs. Soit $\theta = (\theta_1, \ldots, \theta_N)$ une solution positive du système linéaire :*

$$\theta = \theta\mathbf{R}.$$

Le processus $\{N(t), t \geq 0\}$ admet une distribution de probabilité stationnaire *donnée par :*

$$P(N_1 = n_1, \ldots, N_N = n_N) = G \prod_{i=1}^{N} \left(\frac{\theta_i}{\mu_i}\right)^{n_i} \mathbf{1}_{\{n_1+\cdots+n_N=K\}},$$

où

$$G = \frac{1}{\sum_{\forall \mathbf{n}: \sum n_i = K} P(N_1 = n_1, \ldots, N_N = n_N)}.$$

La solution pour θ n'est pas unique. Cependant, comme l'ensemble des solutions forme une droite vectorielle, on peut facilement verifier que la distribution obtenue ne depend pas du choix de la solution de cette equation.

Le calcul des probabilités et des autres mesures de performances (nombre moyen de clients, débits) passe par le calcul de la constante de normalisation G. Il s'agit d'une somme avec un nombre de termes $\begin{pmatrix} N+K-1 \\ N \end{pmatrix}$, ce qui n'est pas toujours faisable. Il existe toutefois des algorithmes récursifs rapides pour ce calcul.

Pour finir cette section, on note qu'il est possible de généraliser ces résultats avec des hypothèses plus faibles sur les disciplines et les lois de service. Ainsi, le résultat de l'existence d'une forme produit, reste valable pour certaines valeurs plus generales de la discipline de service et de la loi de service. Il est également permis d'attribuer des classes aux clients, et des lois de service dependant de la classe. Ces reseaux généralisant ceux de Jackson sont dits "BCMP" d'après les initiales de leurs découvreurs Basket, Chandy, Muntz et Palacios.

CHAPITRE 5

Simulation à Evénements Discrets

5.1 Introduction

Dans ce chapitre, nous allons présenter les principales techniques permettant de mettre en oeuvre une simulation à événements discrets. Dans la partie 5.2, nous décrivons les méthodes permettant de générer des nombres aléatoires suivant une distribution cible. Puis, dans la partie 5.3, nous présentons les techniques de simulation à événements discrets ainsi que les méthodes d'estimation des performances du système.

5.2 Génération de nombres aléatoires

Il peut sembler paradoxal de vouloir concevoir des algorithmes numériques pour produire des nombres aléatoires. En effet, un nombre obtenu d'une méthode numérique est le résultat d'une opération déterministe, non stochastique. C'est pourquoi on parle généralement de nombres pseudo-aléatoires. Il s'agit en fait du résultat d'une opération dont le résultat ne pouvait être induit en première analyse. Par exemple, si on énumère les nombres : 0.1, 0.2, 0.3, 0.4, 0.5 et 0.6, il est facile de se douter du nombre suivant de la suite... Par contre, la suite 0.4, 0.01, 0.89, 0.045 et 0.23 semble "plus aléatoire" dans le sens où il est difficile d'appréhender la relation entre ces nombres.

Dans le domaine de la simulation, il ne suffit pourtant pas de produire des suites étranges de quelques nombres suivant des algorithmes bizarres. En effet, il s'agit de reproduire les événements suivant les lois statistiques qui les gouvernent dans le système réél. Par exemple, si on suppose que les appels des abonnés arrivent dans un réseau téléphonique suivant un processus de Poisson de moyenne $1/\lambda$, il faudra générer des nombres aléatoires suivant la distribution exponentielle $F_X(t) = 1 - exp(-\lambda t)$ pour représenter la durée aléatoire entre deux appels consécutifs.

Si on veut simuler fidélement le système réél, sans biais dû aux générateurs aléatoires, il faut que les générateurs utilisés satisfassent aux critères suivants :

- La séquence de nombres obtenus doit apparaître comme adhérent à la loi de probabilité simulée et ces nombres ne doivent exprimer aucune corrélation statistique entre eux.
- La méthode doit générer une longue séquence d'échantillons avant de présenter un comportement cyclique.
- Le générateur doit calculer chaque nombre de la séquence rapidement, en utilisant un minimum possible de mémoire.
- Nous aimerions pouvoir reproduire aisément la séquence générée.
- Il devrait être aisé de séparer en "flux" la séquence de nombres aléatoires générés, par exemple pour utiliser le même générateur pour 2 sources de trafic dans une simulation de réseaux.

Des tests statistiques, présentés brièvement, permettront d'évaluer l'adéquation des séquences aléatoires produites par un algorithme à la loi de probabilité simulée.

5.2.1 Générateurs Uniformes

L'outil de base de la génération de nombres aléatoires en général consiste en une séquence de nombres aléatoires g_i indépendants et identiquement distribués selon la loi uniforme sur l'intervalle $[0, 1]$, notée $U(0, 1)$. La classe particulière de générateurs uniformes la plus utilisée actuellement est celle des générateurs congruentiels linéaires (linear congruential generators, en abrégé LCG).

Les générateurs congruentiels linéaires LCG

Les générateurs $LCG(a, c, m, g0)$ définissent la séquence de nombres aléatoires g_i par la relation :

$$g_i = (ag_{i-1} + c) \quad mod \quad m$$

où m (le module), a (le multiplicateur), c (l'incrément) et g_0 (la racine ou graine, seed en anglais) sont des entiers naturels. On définit les conditions suivantes entre les paramètres : $m > 0$, $a < m$, $c < m$ et $g_0 < m$. Posant ensuite :

$$v_i = g_i/m$$

on prend les v_i comme échantillons d'une variable aléatoire V distribuée suivant $U(0, 1)$.

Lorsqu'on choisit les quatre paramètres a, c, m et g_0 avec soin, la suite $\{v_i\}$ décrit des échantillons dont les propriétés statistiques sont suffisamment proches de ceux pouvant être obtenus théoriquement suivant $U(0,1)$. Remarquons toutefois que les échantillons v_i qu'ils donnent ne couvrent pas la totalité de l'intervalle réel $[0,1]$, mais seulement les valeurs rationnelles discrètes de la suite $0, 1/m, 2/m, \ldots (m-1)/m$. Toutefois, cette imprécision est faible si on prend m très grand.

Traçons la séquence de nombres générée par LCG(3,5,17,8)...

Terme	Calcul	Valeur
g_0	(par définition)	8
g_1	(3 * 8 + 5) mod 17	12
g_2	(3 * 12 + 5) mod 17	7
g_3	(3 * 7 + 5) mod 17	9
g_4	(3 * 9 + 5) mod 17	15
g_5	(3 * 15 + 5) mod 17	16
g_6	(3 * 16 + 5) mod 17	2
g_7	(3 * 2 + 5) mod 17	11
g_8	(3 * 11 + 5) mod 17	4
g_9	(3 * 4 + 5) mod 17	0
g_{10}	(3 * 0 + 5) mod 17	5
g_{11}	(3 * 5 + 5) mod 17	3
g_{12}	(3 * 3 + 5) mod 17	14
g_{13}	(3 * 14 + 5) mod 17	13
g_{14}	(3 * 13 + 5) mod 17	10
g_{15}	(3 * 10 + 5) mod 17	1
g_{16}	(3 * 1 + 5) mod 17	8

Puisque l'opération modulo par m ne peut retourner que m résultats différents, il est certain que la séquence générée par notre relation présente un comportement cyclique. Le nombre de nombres générés avant que le cycle ne se complète se nomme la période du générateur. Un LCG dont la période est m est dit de période maximale. Par exemple, LCG(3,5,17,8) que nous avons tracé est de période 16, mais pas de période maximale - la valeur 6 n'est pas générée. Lorsqu'un générateur présente cette propriété de période maximale, toute racine donnée en initialisation du procédé produira un cycle de m valeurs aléatoires distinctes. Toutefois, la période d'un générateur dont la période n'est pas maximale dépend du choix de la racine g_0 . Ainsi, LCG(8,0,10,3) génére la suite 3,4,2,6,8 de période 5 alors que LGC(8,0,10,5) génére la suite 5,0 de période 2.

Il est clair qu'un générateur de période maximale est optimal en regard de nos besoins. Un théorème démontré par Hull et Dobell en 1962 pose les conditions nécessaires et suffisantes suivantes à cette propriété :

Théorème - Un LCG a une période maximale si et seulement si ces trois conditions sont satisfaites :

1. Les nombres m et c sont premiers entre eux (le seul entier positif qui divise entièrement à la fois m et c est 1).

2. Si q est un nombre premier diviseur de m, il est aussi diviseur de $(a-1)$.
3. Si 4 est un diviseur de m, 4 est aussi diviseur de $(a-1)$.

Ce résultat permet de construire un générateur aléatoire congruentiel linéaire de période maximale mais ne garantit pas que les propriétés statistiques de distribution des échantillons sont suffisamment uniforme. La construction d'un tel générateur est extrêmement difficile.

Il reste que les générateurs basés sur cette méthode qui présente les propriétés qui nous plaisent satisfont aussi aisément les autres critères de sélection d'un générateur aléatoire. En effet, le temps de génération d'un nombre est très rapide et coûte une quantité négligeable de mémoire. De plus, la reproducibilité et la sécabilité en flux indépendants sont très faciles. En effet, pour avoir des flux de n nombres, il suffit de prendre un LCG avec pour racine g_0 comme flux 1, ce même LCG mais avec la racine g_n comme flux 2, encore ce même LCG avec la racine g_2n comme flux 3, etc.

Générateurs linéaires congruentiels mixtes $(c > 0)$ Ces générateurs peuvent atteindre la période maximale m. En général, on veut m aussi grand que possible. On prend souvent $m = 2^w$, où w représente la taille (en bits) des mots de l'ordinateur ($m = 2^32$ ou $m = 2^64$ suivant l'architecture). L'opération modulo est alors extrêmement rapide, car réalisée par troncature (integer overflow). Une autre optimisation possible est de prendre a de la forme $a = 2^k + 1$ ce qui permet de réaliser la multiplication par décalage de k positions de la représentation binaire vers la gauche. Malheureusement, les générateurs qui utilisent un multiplicateur de cette forme présentent des propriétés statistiques insatisfaisantes.

Parmi ces générateurs, on peut citer l"'ANSI-C system generator" utilisé par la fonction rand() qui correspond à LCG(2^{31}, 1103515245, 12345, 12345) et qui est de période maximale.

Générateurs linéaires congruentiels multiplicatifs Ces générateurs, les plus utilisés, ne peuvent avoir une période maximale. Il est toutefois possible d'obtenir une période de $(m-1)$ si m et a sont choisis avec soin.

Comme pour les générateurs mixtes, on peut vouloir optimiser l'opération modulo en prenant $m = 2^w$. Toutefois, on peut alors montrer que la période d'un tel générateur n'est qu'au maximum 2^{w-2} (le quart des m valeurs possibles).

Une autre méthode consiste à prendre m comme le plus grand nombre premier plus petit que 2^w . Il existe alors des méthodes permettant de choisir le multiplicateur a de manière à conserver la période la plus grande possible, soit $(m-1)$. Nous ne donnerons pas plus de détails sur ce sujet difficile.

Citons deux générateurs de ce type :
– *Minimal Standard Generator* : LCG($2^{31}-1$, 7^5, 0,1) de période $2^{31}-2$.
– *NAG's generator* : LCG(2^{59}, 13^{13},0,123456789 ($2^{32}+1$) de période 2^{57}.

5.2.2 Tests sur les générateurs uniformes

La "qualité" d'un générateur se détermine en évaluant d'une part l'indépendance statistique des nombres générés, et d'autre part leur uniformité. Ils existent de nombreux tests statistiques permettant de vérifier ces deux critères. Nous ne présentons ici que le test du khi-carré.

Ce test permet de vérifier si une série de nombres se comportent d'une manière telle qu'on puisse les dire distribués selon une certaine loi de probabilités.

Rappelons tout d'abord que la distribution du khi carré est un cas particulier de la distribution Gamma. La distribution du $\chi^2_k(x)$ avec k degrés de libertés correspond au cas particulier où $a = 2$ et $b = k/2$ (moyenne égale à k). Comme cette distribution est très utilisée en statistique, il existe des tables donnant ses valeurs pour différents k et différents niveaux de confiance $1-x$ (cf. tableau 5.1).

La méthode consiste d'abord à répartir les n valeurs aléatoires tirées (par exemple $n = 10000$) en un ensemble de k intervalles (typiquement $k = 100$) couvrant $[0,1]$: $[0,1/k],\ldots,[(k-1)/k,1]$. On calcule ensuite le nombre f_i d'échantillons observées dans l'intervalle $[(i-1)/k, i/k]$, qui théoriquement doit être proche de n/k. Puis on calcule la valeur Q suivante :

$$Q = \frac{k}{n}\sum_{i=1}^{k}(f_i - \frac{n}{k})^2$$

On compare alors Q avec $\chi^2_{k-1}(1-\alpha)$ pour un seuil de tolérance α petit (typiquement $\alpha = 1\%$). Si $Q > \chi^2_{k-1}(1-\alpha)$, il y a de forts doutes sur l'uniformité des échantillons générés.

Exemple. On a lancé un dé 90 fois et on a obtenu les issues 1 à 6 (k=6) avec les effectifs suivants : 12, 16, 20, 11, 13, 18. Si le dé n'est pas pipé, on attend comme effectifs moyens théoriques 15 pour toutes les issues.

$$Q = \frac{(12-15)^2}{15} + \frac{(16-15)^2}{15} + \frac{(20-15)^2}{15} + \frac{(11-15)^2}{15} + \frac{(13-15)^2}{15} + \frac{(18-15)^2}{15} = \frac{64}{15} = 4.266$$

Pour $k-1 = 5$ degrés de liberté et un seuil de tolérance de 5%, $\chi^2_5(0.05) = 11.1$. Cela signifie que la probabilité que Q soit supérieur à 11.1 est de 5% (voir figure 5.1). Comme $4.266 < 11.1$, on accepte l'hypothèse que le dé est régulier.

Degré de confiance k	90%	95%	97.5%	99%
1	2,7055435	3,8414588	5,0238862	6,6348966
2	4,6051702	5,9914646	7,3777589	9,2103404
3	6,2513886	7,8147279	9,3484036	11,344867
4	7,7794403	9,487729	11,143287	13,276704
5	9,2363569	11,070498	12,832502	15,086272
6	10,644641	12,591587	14,449375	16,811894
7	12,017037	14,06714	16,012764	18,475307
8	13,361566	15,507313	17,534546	20,090235
9	14,683657	16,918978	19,022768	21,665994
10	15,987179	18,307038	20,483177	23,209251
11	17,275009	19,675138	21,920049	24,72497
12	18,549348	21,02607	23,336664	26,216967
13	19,811929	22,362032	24,735605	27,68825
14	21,064144	23,684791	26,118948	29,141238
15	22,30713	24,99579	27,488393	30,577914
16	23,541829	26,296228	28,845351	31,999927
17	24,769035	27,587112	30,191009	33,408664
18	25,989423	28,869299	31,526378	34,805306
19	27,203571	30,143527	32,852327	36,190869
20	28,411981	31,410433	34,169607	37,566235
21	29,615089	32,670573	35,478876	38,932173
22	30,813282	33,924438	36,780712	40,28936
23	32,0069	35,172462	38,075627	41,638398
24	33,196244	36,415029	39,364077	42,97982
25	34,381587	37,652484	40,646469	44,314105
26	35,563171	38,885139	41,92317	45,641683
27	36,741217	40,113272	43,194511	46,962942
28	37,915923	41,337138	44,460792	48,278236
29	39,08747	42,556968	45,722286	49,587884
30	40,256024	43,772972	46,979242	50,892181
40	51,805057	55,758479	59,341707	63,69074
50	63,167121	67,504806	71,420195	76,153891
60	74,397006	79,081944	83,297675	88,379419
70	85,527043	90,531225	95,023184	100,42518
80	96,578204	101,87947	106,62857	112,32879
90	107,56501	113,14527	118,13589	124,11632
100	118,498	124,34211	129,5612	135,80672

TABLE 5.1 – Points percentiles de la loi du χ^2

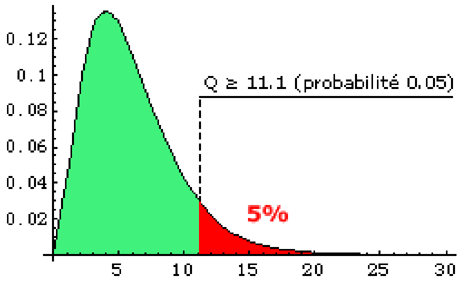

FIGURE 5.1 – Distribution du khi-carré.

5.2.3 Générateurs non uniformes

Méthode de la fonction percentile

On veut générer des échantillons $x_1, x_2, \ldots$ **indépendants et identiquement distribués** d'une certaine loi de probabilités dont la fonction de répartition est $F_X(x)$. On rappelle que $F_X(x) = P(X <= x)$. Si cette fonction est continue et croissante monotone, elle admet une inverse $F_X^{-1}(x)$. Cette inverse est dénommé fonction de distribution inverse ou, communément, **fonction percentile** de X. On a le résultat suivant :

Théorème Si U est une variable aléatoire uniformément distribuée sur $[0, 1]$, alors la variable aléatoire $Y = F_X^{-1}(U)$ a pour fonction de distribution $F_X(x)$.

En effet, la variable aléatoire $Y = F_X^{-1}(U)$ a une fonction de répartition définie par $F_Y(x) = P[F_X^{-1}(U) <= x] = P[U <= F_X(x)] = F_X(x)$. Le principe est illustré sur la figure 5.2.

Exemple de la loi exponentielle : Soit une va X exponentiellement distribuée de moyenne $\frac{1}{\lambda}$, i.e $F_X(x) = 1 - e^{-\lambda x}$. En posant $y = F_X(x)$,

$$1 - y = e^{-\lambda x} \quad \Rightarrow \quad x = \frac{-1}{\lambda}\, log(1 - y)$$

Ainsi on a $F_X^{-1}(y) = -log(1 - y)/\lambda$. Sachant que $1 - U$ est distribué uniformément, selon le théorème précédent, les variables $X_1 = \frac{-1}{\lambda} log(U_1)$, ..., $X_n = -\frac{-1}{\lambda} log(U_n)$, sont des variables aléatoires indépendantes distribuées selon la loi exponentielle de moyenne $\frac{1}{\lambda}$.

Méthode d'acceptation-rejet

Si la fonction de répartition $F_X(x)$ n'est pas facilement inversible, la méthode dite **d'acceptation-rejet** consiste à majorer la densité de probabilités $f(x) =$

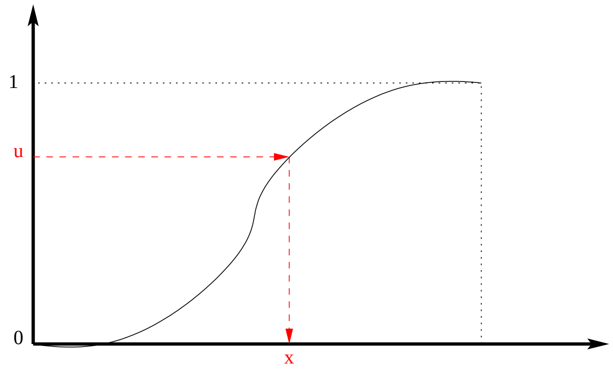

FIGURE 5.2 – Méthode de la fonction percentile.

$dF_X(x)/dx$ par une fonction $t(x)$ de notre choix : $t(x) >= f(x)$, $\forall x$. En général, $t(x)$ n'est pas une densité de probabilités, puisque

$$c = \int_{-\infty}^{\infty} t(x)\, dx >= \int_{-\infty}^{\infty} f(x)\, dx = 1$$

Mais $r(x) = t(x)/c$ est bien une densité de probabilités. Choisissons la fonction $t(x)$ de manière à ce que la loi associée à $r(x)$ soit facile à échantillonner (par la méthode de la fonction percentile). L'algorithme suivant permet alors de générer des valeurs x_i distribuées selon la loi associée à la densité $f(x)$:

1. Générer y ayant pour fonction de densité $r(x)$,
2. Générer u selon $U(0, 1)$,
3. Si $u <= f(y)/t(y)$, alors $x_i = y$. Sinon, reprendre à l'étape 1.

Ainsi, les valeurs x_i générées sont distribuées selon la loi décrite par $f(x)$. Le principe est illustré sur la figure 5.3.

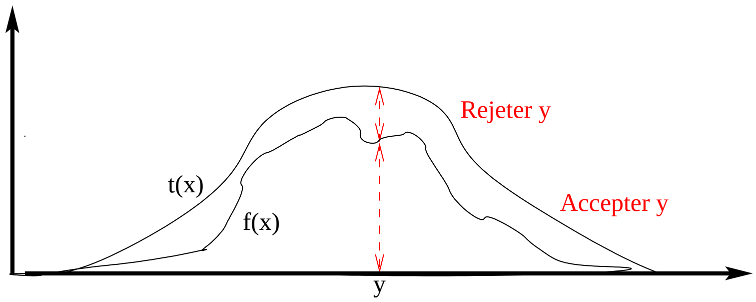

FIGURE 5.3 – Méthode d'acceptation-rejet.

5.3 Simulation événementielle

Nous nous intéressons ici à la simulation de systèmes dynamiques stochastiques à événements discrets. La première étape consiste à modéliser le système réel en précisant le quintuplet (E, X, Γ, f, x_0) où :

- E est un ensemble dénombrable d'événements qui font évoluer l'état du système,
- X est l'espace d'état discret du système,
- $\Gamma(x) \subseteq E$ est l'ensemble des événements admissibles dans l'état $x \in X$.
- $f: \ X \times E \to X$ est une fonction de transition donnant l'état $f(x, e)$ obtenu à partir de l'état courant x lorsque l'événement $e \in \Gamma(x)$ survient.
- x_0 est l'état initial du système.

Exemple. Prenons l'exemple d'une file $M/M/1/\infty$:

- $E = \{a, d\}$ où l'événement a représente l'arrivée d'un client tandis que d représente le départ d'un client,
- $X = \{0, 1, 2, \ldots\}$ représente des valeurs possibles du nombre de clients dans le système,
- $\Gamma(x) = \{a, d\} \quad \forall x > 0$ et $\Gamma(0) = \{a\}$,
- La fonction de transition $f: \ X \times E \to X$ est définie par

$$f(x, e) = \begin{cases} x + 1 & \text{si } e = a \\ \\ x - 1 & \text{si } e = d \text{ et } x > 0 \end{cases}$$

5.3.1 Algorithme de simulation

A chaque événement e est associé un instant t_e d'exécution (ou "d'activation") indiquant quand il doit se produire. L'état du système change dans le temps aux instants t_i où les événements ont lieu et reste constant entre deux événements.

L'exécution d'un événement e à la date t peut générer un ou plusieurs autres événements e' dont les dates d'exécution seront $t+v$, v étant une durée aléatoire. Pour reprendre l'exemple de la file $M/M/1/\infty$, l'événement a d'arrivée d'un paquet à la date t produira un autre événement d'arrivée à la date $t + v_a$ où v_a est une variable aléatoire exponentiellement distribuée de moyenne $1/\lambda$. De même, le départ d'un client produira, si la file n'est pas vide, l'entrée en service d'un autre client et donc un autre événement de départ à la date $t_0 + v_d$, où v_d est exponentiellement distribuée.

L'architecture élémentaire du noyau d'un simulateur à événements discrets est illustrée dans la figure 5.4. Normalement, le temps est représenté par une variable aléatoire t appelée l'**horloge de simulation**. Dans le noyau du simulateur, on trouve une liste $L = \{(e_k, t_k)\}$ où les événements en attente

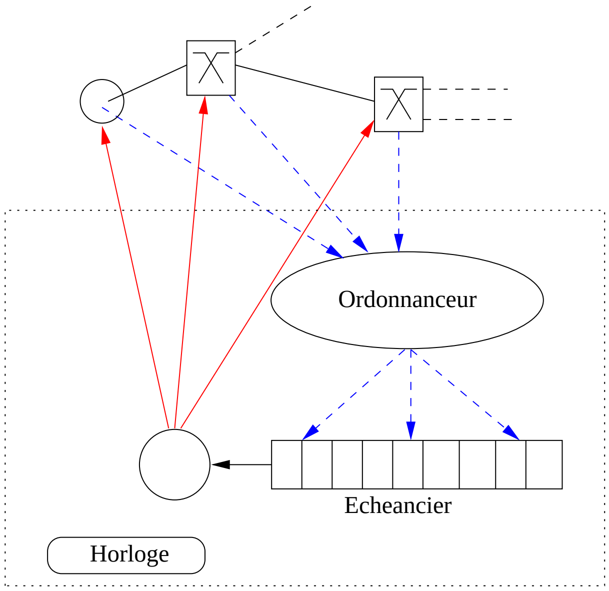

FIGURE 5.4 – Noyau d'un simulateur à événements discrets.

d'exécution sont ordonnés par ordre croissant des dates d'exécution. Cette liste est dénommée l'**échéancier**. Au cours de la simulation, le noyau exécute la boucle suivante :

- **Etape 1.** Supprimer le premier élément (e_i, t_i) de la liste $L = \{(e_k, t_k)\}$,
- **Etape 2.** Mettre à jour l'horloge de simulation : $t = t_i$,
- **Etape 3.** Mettre à jour l'état du système : $x_i = f(x_{i-1}, e_i)$,
- **Etape 4.** Ajouter à l'échéancier L les événements $(e'_i, t_i + v_i)$ "produits" par l'événement e_i.

Puisqu'une partie importante des activités du simulateur consiste à insérer et à retirer des événements de l'échéancier, la structure de données de celui-ci affecte grandement l'efficacité du simulateur. Trois structures de données sont souvent utilisées : la liste linéaire chaînée, les tas (heap) et les structures de listes linéaires indexées telles que les listes calendrier (calendar queues).

Progressions synchrones et asynchrones

L'horloge de simulation est avancée au fur et à mesure que les événements sont extraits de l'échéancier. On distingue deux techniques pour faire avancer cette horloge : progression à pas fixes ou synchrone et progression par événements ou asynchrone (cf figure 5.5).

- **Progression à pas fixes.** L'horloge de simulation avance à intervalles fixes de longueur Δt appelés cycles. A la fin de chaque cycle, on regarde

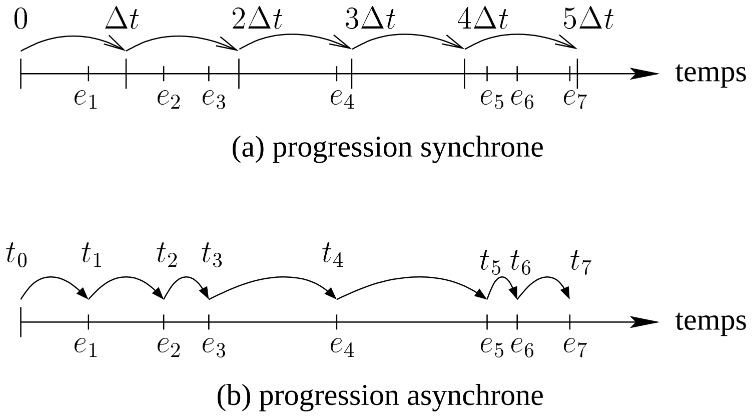

FIGURE 5.5 – Gestion du temps de simulation.

si des événements se sont produits durant $[i\,\Delta t, (i+1)\,\Delta t[$. S'il y en a, ils sont simulés comme s'ils avaient eu lieu à la date $(i+1)\,\Delta t$. Cette méthode a plusieurs inconvénients (choix de Δt, erreurs introduites, contraintes temporelles perdues...). De plus, cette méthode est parfois peu efficace car il peut y avoir beaucoup d'intervalles "vides".

- **Progression par événements.** Le noyau de simulation extrait de l'échéancier l'événement en tête de liste (qui a la date t_e la plus petite). L'horloge de simulation est actualisée à t_e et l'événement en question est exécuté. Ainsi, l'horloge de simulation progresse par sauts irréguliers et le système n'est évalué que quand il y a eu un changement d'état.

5.3.2 Exemple de simulation d'une file M/M/1

Nous allons illustrer les concepts introduits précédemment en reprenant l'exemple (élémentaire) d'une file $M/M/1/\infty$. Notre objectif est également de montrer comment cette simulation peut nous permettre de déterminer plusieurs mesures de performances.

On va simuler la file $M/M/1$ à partir du temps 0, avec une file initialement vide, et ce jusqu'à ce qu'un nombre $N = 5$ de clients aient été servis. On va estimer les mesures de performances suivantes :

- **Temps moyen dans le système.** Soit S_N le temps moyen passé dans le système par les N clients. Il s'agit d'une variable aléatoire puisque chaque séquence de tirages aléatoires différentes pour les temps inter-arrivée ou de service produira une valeur différente de S_N. Notons s_k le temps passé dans le système par le $k^{\text{ème}}$ client. Un estimé de S_N est,

$$\overline{S}_N = \frac{1}{N} \sum_{k=1}^{N} s_k = \frac{N-1}{N}\,\overline{S}_{N-1} \;+\; \frac{1}{N}\, s_N$$

– **Probabilité que le temps de service excède D.** Soit n_N le nombre de clients dont le temps de service excède la valeur D. Un estimé de la probabilité $P_N(D)$ est,

$$\overline{P}_N(D) = n_N/N$$

– **Taux d'utilisation.** Soit T_i le temps cumulé pendant lequel le système contient exactement i clients. Un estimé du taux d'utilisation ρ_N est alors,

$$\overline{\rho}_N = \frac{\sum_{i=1}^{\infty} T_i}{\sum_{i=0}^{\infty} T_i} = 1 - \frac{T_0}{\sum_{i=0}^{\infty} T_i}$$

Notez que le dénominateur correspond au temps total de simulation.

– **Nombre moyen de clients.** Un estimé du nombre moyen de clients X_N est,

$$\overline{X}_N = \frac{\sum_{i=1}^{\infty} i\, T_i}{\sum_{i=0}^{\infty} T_i}$$

Pour simuler ce système et obtnir les mesures précédentes, il faut réaliser des tirages aléatoires des durées inter-arrivée et de service (exponentiellement distribuées). On notera y_i la durée entre les arrivées a_{i-1} et a_i, et z_i le temps de service du $i^{\text{ème}}$ client.

L'algorithme de simulation se déroule de la façon suivante :

– **t=0** (initialisation) - Le système étant vide, on tire aléatoirement une première date d'arrivée et on obtient $y_1 = 0.4$. On initialise l'échéancier avec un premier événement d'arrivée $L = \{(a_1, 0.4)\}$. On rentre alors dans la boucle de simulation.
– **t=0.4** (arrivée a_1) - L'état évolue $x(0.4) = 1$ et le client entre en service. On tire la durée jusqu'à la prochaine arrivée $y_2 = 0.3$ et la durée de service $z_1 = 1.6$ et on ajoute 2 événements à l'échéancier $L = \{(a_2, 0.7), (d_1, 2.0)\}$.
– **t=0.7** (arrivée a_2) - L'état évolue $x(0.7) = 2$. On tire la durée jusqu'à la prochaine arrivée $y_3 = 0.4$ et on met à jour l'échéancier $L = \{(a_3, 1.1), (d_1, 2.0)\}$.
– **t=1.1** (arrivée a_3) - L'état évolue $x(1.1) = 3$. On tire la durée jusqu'à la prochaine arrivée $y_4 = 1.7$ et on met à jour l'échéancier $L = \{(d_1, 2.0), (a_4, 2.8)\}$.
– **t=2.0** (départ d_1) - Le nouvel état est $x(2.0) = 2$ et un nouveau client entre en service. On tire la durée de service $z_2 = 0.5$ et on insére d_2 dans l'échéancier $L = \{(d_2, 2.5), (a_4, 2.8)\}$.
– **t=2.5** (départ d_2) - Le nouvel état est $x(2.5) = 1$ et un nouveau client entre en service. On tire la durée de service $z_3 = 1.0$ et on insére d_3 dans l'échéancier $L = \{(a_4, 2.8), (d_3, 3.5)\}$.
– **t=2.8** (arrivée a_4) - Le nouvel état est $x(2.8) = 2$. On tire la durée jusqu'à la prochaine arrivée $y_5 = 1.7$ et on met à jour l'échéancier $L = \{(d_3, 3.5), (a_5, 4.5)\}$.

- **t=3.5** (départ d_3) - Le nouvel état est $x(3.5) = 1$ et un nouveau client entre en service. On tire la durée de service $z_4 = 0.3$ et on insére d_4 dans l'échéancier $L = \{(d_4, 3.8), (a_5, 4.5)\}$.
- **t=3.8** (départ d_4) - Le nouvel état est $x(3.8) = 0$. Le système étant vide, on n'insére pas de nouvel événement de départ. On a $L = \{(a_5, 4.5)\}$.
- **t=4.5** (arrivée a_5) - L'état évolue $x(4.5) = 1$ et le client entre en service. On tire la durée jusqu'à la prochaine arrivée $y_6 = 0.5$ et la durée de service $z_5 = 0.8$ et on ajoute 2 événements à l'échéancier $L = \{(a_6, 5.0), (d_5, 5.3)\}$.
- **t=5.0** (arrivée a_6) - L'état est $x(5.0) = 2$. On tire la durée jusqu'à la prochaine arrivée $y_7 = 1.2$ et on met à jour l'échéancier $L = \{(d_5, 5.3), (a_7, 6.2)\}$.
- **t=5.3** (départ d_5) - Le nouvel état est $x(5.3) = 1$ et un nouveau client entre en service. On tire la durée de service $z_6 = 0.7$ et on insére d_6 dans l'échéancier $L = \{(d_6, 6.0), (a_7, 6.2)\}$.

La simulation précédente génére une des trajectoires possibles du processus stochastique $\{X(t)\}_{t\geq 0}$ d'occupation d'une file $M/M/1$, que nous avons représentée sur la figure 5.6.

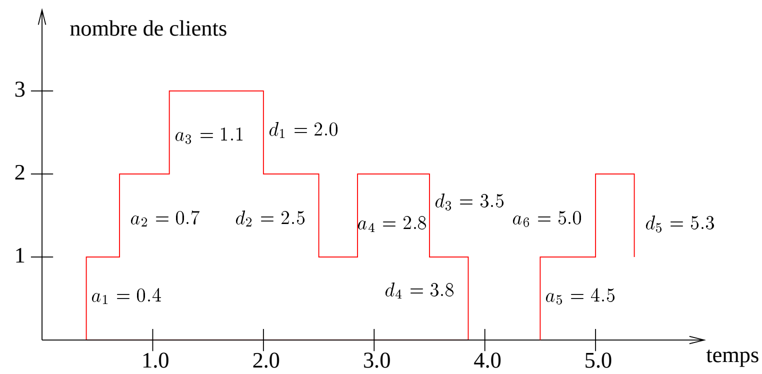

FIGURE 5.6 – Une des trajectoires possibles pour une file $M/M/1/\infty$.

En ayant pris soin de retenir les variables s_k, n_N et T_i évoqués précédemment, on peut alors calculer les mesures de performance du système pour cette trajectoire :

- **Temps moyen dans le système.**

$$\overline{S}_5 = \frac{1}{5}\,(1.6 + 1.8 + 2.4 + 1.0 + 0.8) = 1.52$$

- **Probabilité que le temps de service excède D=2.**

$$\overline{P}_5(2.0) = 1/5 = 0.2$$

- **Taux d'utilisation.**

$$\overline{\rho}_5 = 1 - \frac{1.1}{5.3} \approx 0.79$$

– **Nombre moyen de clients.**

$$\overline{X}_5 = \frac{(1 \times 1.4) + (2 \times 1.9) + (3 \times 0.9) + 0 + \ldots}{5.3} \approx 1.49$$

Remarquons qu'en général il n'est pas vraiment nécessaire de retenir les valeurs s_k et T_i mais qu'il suffit de mettre à jour les moyennes au fur et à mesure que la simulation progresse.

L'exemple présenté ici soulève plusieurs questions, auxquelles nous allons essayer de répondre dans les paragraphes suivants :

- La simulation précédente sur un horizon de $N = 5$ clients constitue un exemple de ce que l'on appelle une simulation terminante. Un autre exemple serait une simulation jusqu'à un temps T donné. Ce type de simulation caractérise le fonctionnement du système sur un horizon de temps fini, fonctionnement qui dépend fortement des conditions initiales. On parle alors de simulation du régime transitoire. Le comportement du système sur le long terme est obtenu par ce qu'on appelle une simulation non terminante, c'est à dire en le simulant sur un temps considéré comme "infini". On parle de simulation du régime permanent. Ce point est développé au paragraphe 5.3.3.
- La simulation précédente a généré une trajectoire du processus stochastique $\{X(t)\}_{t \geq 0}$, sur des milliers possibles, et les mesures de performance que nous avons calculées ne sont valides que pour cette trajectoire. Elles ne nous apportent donc que peu d'informations sur le système. Par contre, en générant des centaines ou des milliers de trajectoires indépendantes du système, on va pouvoir estimer des mesures de performances relatives au comportement moyen du système. Ce point est développé au paragraphe 5.3.4.

5.3.3 Régime transitoire et régime permanent

Considérons un certain processus stochastique $Y_1, Y_2, \ldots$ en sortie d'une simulation. Pour reprendre une fois de plus l'exemple d'une file $M/M/1$, ce processus stochastique peut être formé du temps de séjour dans le système des clients qui s'y succèdent. Il s'agit bien d'un processus stochastique à temps discret et deux simulations événementielles distinctes (avec des graines différentes) nous fournirons deux réalisations de ce processus.

On définit alors $F_i(y \,|\, I) = P\,[Y_i <= y \,|\, I]$ la fonction de répartition de Y_i conditionnelle à l'ensemble des conditions initiales de la simulation. En général, les $F_i(y \,|\, I)$ sont différentes pour chaque Y_i et chaque ensemble de conditions initiales I. Cependant, dans certains cas, pour $i \geq k$, les répartitions $F_i(y \,|\, I)$ ressemblent de plus en plus à une certaine répartition commune $F(y)$ (cf. figure 5.7).

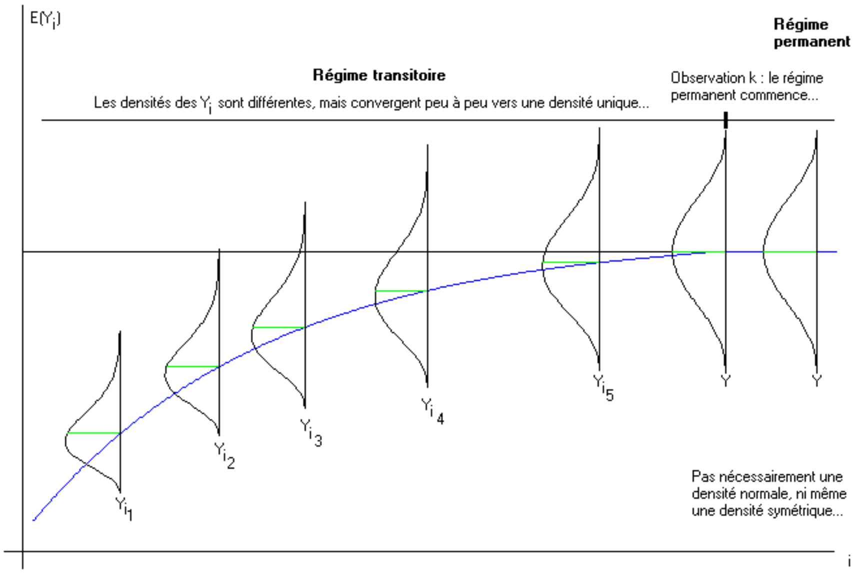

FIGURE 5.7 – Convergence de l'espérance et des densités de probabilités vers leurs valeurs stationnaires.

Informellement, on voit que la quantité qu'on observe passe d'un état très variant et dépendant des conditions initiales à un état stable et indépendant de ces conditions. Avant ce $i = k$, la simulation est dite en régime transitoire ; après, elle est dite en régime permanent. Au sens strict, la convergence de la simulation au régime permanent ne se réalise que lorsque i atteint l'infini ; en pratique, on accepte que le système est en régime stationnaire pour i suffisamment grand.

Notez que le régime permanent ne signifie pas que les variables aléatoires $Y_{k+1}, Y_{k+2}, \ldots$ prendront la même valeur : il signifie que ces variables seront distribuées selon la même loi de probabilités. Le régime permanent représente le comportement du système sur le long terme et c'est en général lui que l'on veut étudier.

5.3.4 Estimation des performances en régime permanent

Contrairement à la théorie des files d'attentes, où l'obtention du régime stationnaire peut facilement être obtenue pour certains modèles, la simulation du régime permanent d'un système est assez complexe (le contraire est vrai en ce qui concerne le régime transitoire). Il existe deux méthodes principales : la méthode des "moyennes par lot"(Batch Means) et la méthode des "simulations parallèles"(Independent Replications).

Méthode des moyennes par lot

Cette méthode, illustrée sur la figure 5.3.4, nécessite une seule trajectoire de simulation, mais très longue, que l'on subdivise en une période transitoire initiale et en n lots de m réalisations $x_{i,j}$ de l'état du système. Chaque lot est alors considéré comme une trajectoire de simulation indépendante tandis que la période transitoire, nécessaire à la montée en charge du système, est exclue des statistiques.

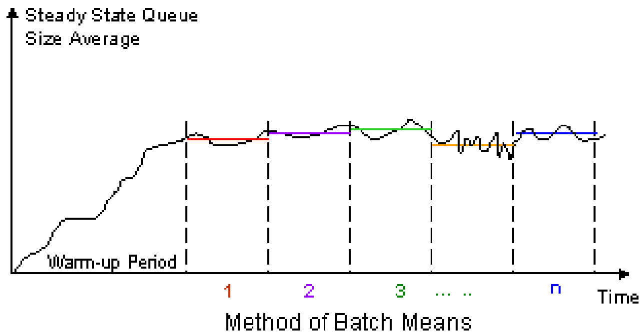

FIGURE 5.8 – Méthode des moyennes par lot.

Méthode des simulations parallèles

Cette méthode est la plus couramment utilisée lorsque la période transitoire du système est assez courte. Elle a de plus l'avantage de permettre l'analyse du comportement moyen du système en régime transitoire. L'idée est de générer n trajectoires indépendantes du système avec des graines différentes pour les générateurs aléatoires utilisés. Si on n'est intéressé que par le régime stationnaire, on supprime alors les échantillons $x_{i,j}$ de la période transitoire. Cette méthode est illustrée sur la figure 5.3.4.

Estimation de l'état moyen et de sa variance

Les deux méthodes précédentes fournissent n trajectoires indépendantes, chacune composée de n_i réalisations de l'état du système. La moyenne m_i de la trajectoire i est donnée par :

$$m_i = \sum_{j=1}^{n_i} x_{i,j}$$

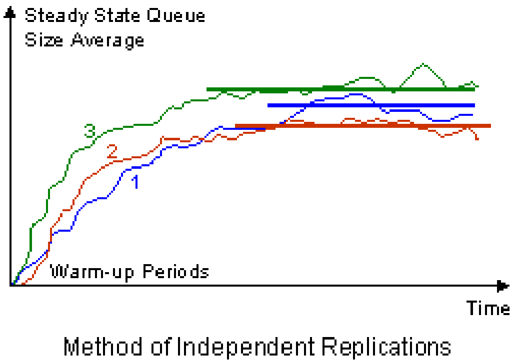

FIGURE 5.9 – Méthode des simulations parallèles.

On en déduit la moyenne empirique $\overline{X}_n$ de l'état du système :

$$\overline{X}_n = \frac{1}{n} \sum_{i=1}^{n} m_i$$

De même, on peut calculer la variance empirique associée aux n trajectoires générées :

$$S_n^2 = \frac{1}{n-1} \sum_{i=1}^{n} (m_i - \overline{X}_n)^2$$

Les variables $\overline{X}_n$ et S_n^2 sont des variables aléatoires qui constituent des estimateurs sans biais de la moyenne $\overline{X}$ et de la variance σ^2 du processus simulé, c'est à dire que $E\left[\overline{X}_n\right] = \overline{X}$ et $E\left[S_n^2\right] = \sigma^2$.

Intuitivement, il est clair que plus on fera de simulations, plus $\overline{X}_n$ fournira une bonne approximation de $\overline{X}$. Autrement dit, on s'attend à ce que la variance de $\overline{X}_n$ diminue quand n augmente, c'est à dire à ce que la probabilité que la moyenne empirique soit proche de $\overline{X}$ soit forte. On peut montrer que,

$$Var\left[\overline{X}_n\right] = \frac{\sigma^2}{n}$$

Sachant que S_n^2 est un estimateur de σ^2, on pourra estimer la variance de $\overline{X}_n$ de la façon suivante :

$$Var\left[\overline{X}_n\right] \approx \frac{S_n^2}{n}$$

Des exemples typiques d'évolution de la densité de probabilités de $\overline{X}_n$ et de la convergence de la simulation sont représentés sur les figures 5.10 et 5.11.

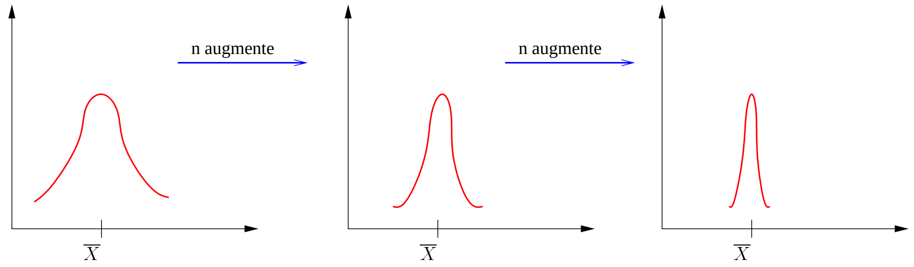

FIGURE 5.10 – La densité de probabilité de $\overline{X}_n$ se concentre autour de $\overline{X}$ au fur et à mesure que le nombre de trajectoires générées augmente.

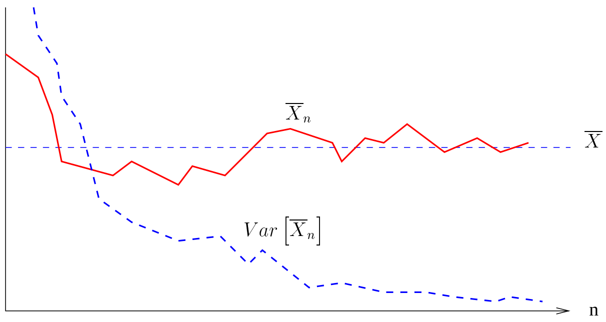

FIGURE 5.11 – Convergence de $\overline{X}_n$ vers $\overline{X}$ et de $Var\left[\overline{X}_n\right]$ vers 0.

Intervalle de confiance

La question qui se pose est celle de la confiance que l'on peut avoir dans l'estimateur empirique $\overline{X}_n$ en regard de la moyenne $\overline{X}$ de ce processus. Rappelons que le théorème central limite nous affirme que la variable aléatoire,

$$Z_n = \frac{\overline{X}_n \; - \; n\,\overline{X}}{\sigma_X\,\sqrt{n}}$$

a une fonction de répartition qui tend vers une loi normale quand n augmente, c'est à dire :

$$\lim_{n\to\infty} P\left[Z_n \leq x\right] = \Phi(x) \quad \text{où} \quad \Phi(x) = \int_{-\infty}^{x} \frac{1}{(2\pi)^{1/2}}\, e^{-y^2/2}\, dy$$

Ce résultat nous dit donc que pour n assez grand on peut affirmer que $\overline{X}_n$ est proche de $\overline{X}$ avec un degré de confiance donné. Ceci nous amène à définir l'intervalle de confiance I_C ainsi :

Degré de confiance n	50%	80%	90%	95%	98%	99%
1	1.000	3.078	6.314	12.706	31.821	63.657
2	0.816	1.886	2.920	4.303	6.965	9.925
3	0.765	1.638	2.353	3.182	4.541	5.841
4	0.741	1.533	2.132	2.776	3.747	4.604
5	0.727	1.476	2.015	2.571	3.365	4.032
6	0.718	1.440	1.943	2.447	3.143	3.707
7	0.711	1.415	1.895	2.365	2.998	3.499
8	0.706	1.397	1.860	2.306	2.896	3.355
9	0.703	1.383	1.833	2.262	2.821	3.250
10	0.700	1.372	1.812	2.228	2.764	3.169
11	0.697	1.363	1.796	2.201	2.718	3.106
12	0.695	1.356	1.782	2.179	2.681	3.055
13	0.694	1.350	1.771	2.160	2.650	3.012
14	0.692	1.345	1.761	2.145	2.624	2.977
15	0.691	1.341	1.753	2.131	2.602	2.947
16	0.690	1.337	1.746	2.120	2.583	2.921
17	0.689	1.333	1.740	2.110	2.567	2.898
18	0.688	1.330	1.734	2.101	2.552	2.878
19	0.688	1.328	1.729	2.093	2.539	2.861
20	0.687	1.325	1.725	2.086	2.528	2.845
21	0.686	1.323	1.721	2.080	2.518	2.831
22	0.686	1.321	1.717	2.074	2.508	2.819
23	0.685	1.319	1.714	2.069	2.500	2.807
24	0.685	1.318	1.711	2.064	2.492	2.797
25	0.684	1.316	1.708	2.060	2.485	2.787
26	0.684	1.315	1.706	2.056	2.479	2.779
27	0.684	1.314	1.703	2.052	2.473	2.771
28	0.683	1.313	1.701	2.048	2.467	2.763
29	0.683	1.311	1.699	2.045	2.462	2.756
30	0.683	1.310	1.697	2.042	2.457	2.750
40	0.681	1.303	1.684	2.021	2.423	2.704
50	0.679	1.299	1.676	2.009	2.403	2.678
60	0.679	1.296	1.671	2.000	2.390	2.660
70	0.678	1.294	1.667	1.994	2.381	2.648
80	0.678	1.292	1.664	1.990	2.374	2.639
90	0.677	1.291	1.662	1.987	2.368	2.632
100	0.677	1.290	1.660	1.984	2.364	2.626
$N(0,1)$	0.674 1.282	1.645	1.960	2.326	2.576	

Table 5.2 – Points percentiles des lois de Student t (n degrés de libertés) et normale $N(0,1)$

$$I_C = \left[\overline{X}_n - t_{n-1,1-\alpha/2}\sqrt{\frac{S_n^2}{n}}\ ,\ \overline{X}_n + t_{n-1,1-\alpha/2}\sqrt{\frac{S_n^2}{n}}\right] = \left[\overline{X}_n \pm t_{n-1,1-\alpha/2}\sqrt{\frac{S_n^2}{n}}\right]$$

On constate que la précision dépend de la variance des données observées, laquelle découle de la variance de la variable aléatoire dont on estime la valeur. Le paramètre $t_{n-1,1-\alpha/2}$ est une constante correspondant à un point percentile de la loi de Student avec $d = n-1$ degrés de liberté (cf. tableau 5.2). Notez qu'à mesure que d augmente, la loi de Student dégénère vers une loi normale $N(0,1)$. Si on a besoin d'une valeur de la percentile pour une probabilité positionnée entre deux valeurs tabulées, on utilise l'interpolation linéaire.

Il est cependant typique qu'on veuille plutôt choisir la précision β (erreur maximale) de l'intervalle de confiance que le nombre de réplications de la simulation. On doit alors fixer d'une part le niveau de confiance de l'intervalle (probabilité que $\overline{X}$ appartienne à I_C) et d'autre part, l'erreur absolue β de l'intervalle. Ceci étant fait, on génére des trajectoires du processus stochastique par simulation jusqu'à ce que la précision soit inférieure à β.

Il peut être utile de prédire le nombre de trajectoires nécessaires pour atteindre une bonne précision. En effet, si ce nombre est gigantesque, on peut éventuellement accepter une précision moindre. Supposons que n réplications de la simulation ont déjà été faites. On pourra prédire approximativement le nombre total $\nu_\alpha(\beta)$ de trajectoires à générer pour "rentrer" dans l'intervalle de confiance en utilisant la formule suivante :

$$\nu_\alpha(\beta) = min\left\{ i \geq n \mid t_{i-1,1-\alpha/2}\sqrt{\frac{S_n^2}{i}} \right\}$$

Comme ce nombre dépend de la variance S_n^2 des données utilisées pour calculer l'intervalle de confiance, il varie suite à chaque nouvelle trajectoire de simulation. Ainsi, ayant exécuté n réplications, si on exécute $\nu_\alpha(\beta) - n$ réplications de plus pour tâcher de compléter l'intervalle de confiance, il se peut que ce soit encore insuffisant (ou même excessif, ce qui n'est pas problématique du point de vue de la précision).

Détection du régime permanent

Comment peut-on n'utiliser que des échantillons du régime permanent ? Pour arriver à cela, il faut être en mesure de déterminer à quel moment le système passe du régime transitoire au régime permanent.

Une technique générale consiste à exécuter n trajectoires de simulation de durée m chacune, $\{x_{i,1}, \ldots, x_{i,m}\}_{i=1\ldots n}$. On prend alors la moyenne M_j des réalisations au temps j :

$$M_j = \frac{1}{n}\sum_{i=1}^{n} x_{i,j}$$

et on calcule une moyenne mobile $MM_j(w)$ de longueur $2w+1$ autour de M_j :

$$MM_j(w) = \frac{\sum_{s=-w}^{w} M_{j+s}}{2w+1} \quad j \geq w+1$$

L'indice l à partir duquel le système est en régime stationnaire correspond alors à l'index à partir duquel les $MM_j(w)$ ne semblent plus varier amplement. Evidemment, cette technique pose le problème du choix de w.

ρ	$100\,p$			
	95.0	99.0	99.9	99.99
0.10	3.61	6.33	10.23	14.12
0.20	5.01	8.93	14.53	20.14
0.30	7.00	12.64	20.71	28.79
0.40	10.06	18.39	30.31	42.23
0.50	15.18	28.05	46.47	64.89
0.60	24.70	46.13	76.79	107.45
0.70	45.51	85.87	143.61	201.36
0.80	105.78	201.53	338.52	475.51
0.90	435.74	838.10	1413.70	1989.40

TABLE 5.3 – Temps de convergence d'une file $M/M/1$ vers la valeur stationnaire à $100(1-p)$ près

Une autre méthode, applicable dans le cas de la simulation de files d'attente, consiste à utiliser un résultat obtenu par Abate et Whitt concernant la durée de la période transitoire d'une file $M/M/1/\infty$ dont le taux d'utilisation est ρ et dont l'état initial est vide. Ce résultat nous dit que le nombre moyen de clients s'approche à $100\,(1-p)$ % près de sa limite stationnaire au bout d'un temps $t_p(\rho)$ donné par :

$$t_p(\rho) = \frac{2\,C(\rho)}{(1-\rho)^2}\, ln \left(\frac{1}{p(1\ +\ 2C(\rho))} \right) \quad \text{où} \quad C(\rho) = [2+\rho+\sqrt{2+4\rho}]/4$$

Le tableau 5.3 fournit quelques valeurs courantes de cette fonction. Bien qu'établie pour une file $M/M/1$, cette formule fournie une approximation correcte dans le cas plus général des files $GI/GI/1$.

5.4 Conclusion

Dans ce chapitre, nous avons vu les principes fondamentaux des générateurs aléatoires ainsi que leur utilisation pour simuler un système stochastique dynamique. Nous nous sommes principalement focalisé sur la simulation de réseaux de files d'attente en régime permanent et avons vu comment estimer les principaux indices de performance du système.

De nombreux points n'ont pas été abordés dans cette présentation succinte (techniques de réduction de variance, analyse de sensibilité, etc.). Néanmoins, nous en avons vu assez pour bien comprendre les avantages et les limites de la simulation événementielle. Ses principaux avantages sont d'une part la grande variété des mesures de performances que l'on peut obtenir et d'autre part sa grande flexibilité, qui permet de simuler des mécanismes très complexes. L'inconvénient majeur de cette technique est qu'elle est extrêmement lourde en temps calcul pour des sytèmes de grande dimension, ce qui limite son utilisation dans le domaine des réseaux de télécommunications à quelques dizaines de noeuds.

Une voie prometteuse, permettant d'étendre les modèles analytiques au delà de leurs limites théoriques, consiste à combiner des modèles analytiques continus et des modèles discrets de simulation dans un framework global de modélisation hybride.

5.5 Bibliographie

Liste de références liées au domaine :

– **Cassandras C.** *Discrete Event Systems : Modeling and Performance Analysis*. Aksen Associates Inc. Publishers, 1993.
– **Phillips D, Ravindran A. and Solberg J.** *Operations Research : Principles and Practice*. John Wiley & Sons, 1976.
– **A.Law and W. Kelton**, *Simulation Modeling and Analysis*. 3rd ed., McGraw Hill, 2000.