**République Algérienne Démocratique et Populaire**
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**

**Université A/MIRA de Béjaia**
**Faculté des Sciences exactes**
**Département de Mathématiques**

***MEMOIRE DE FIN DE CYCLE***

**en vue de l'obtention du Diplôme de Master**

**En Mathématiques**

**Option : *Statistique et Analyse Décisionnelle***

# Thème :

# Les Réseaux de Files d'Attente à Forme Produit

**Présenté par :**

– **M$^{lle}$ GUEDJALI Mounira.**
– **M$^{lle}$ SANA Wissem.**

**Devant le jury composé de :**

| | | | |
|---|---|---|---|
| **Président** | **M$^{me}$ M. Aoudia** | **MCA** | **U. A/Mira, Béjaia** |
| **Rapporteur** | **M$^{r}$ M. BOURAINE** | **MAA** | **U. A/Mira, Béjaia** |
| **Examinatrice** | **M$^{me}$ S. GUEBLI** | **MCB** | **U. A/Mira, Béjaia** |

Promotion,2015/2016.

# Table des matières

# Table des figures

# Liste des tableaux

# Introduction générale

Des phénomènes d'attente se manifestent sous des formes multiples comme par exemple : l'arrivée des voitures vers une station de service, la vente de billets auprès d'un guichet, l'exécution des tâches dans un centre de calcul, ...

L'étude des systèmes de files d'attente consiste à examiner le comportement au cours du temps de certaines grandeurs, comme la longueur des files d'attente, le temps de séjour et le temps d'attente des différents clients, qui en reflètent leurs performances. [12]

La théorie des files d'attente est principalement vue comme une branche de la théorie des probabilités appliquées. Les applications sont dans différents domaines, par exemple : les réseaux de transmission, les systèmes informatiques, les réseaux urbains, les banques, la gestion des avions au décollage ou à l'atterrissage, ....[12]

Cette théorie utilise des outils probabilistes pour étudier et modéliser le comportement d'un système donné. En quelques mots, cette théorie a pour objet l'étude des systèmes ou du comportement des " entités " appelées : clients, services, gestionnaire. Ces derniers cherchent à accéder à une ressource afin d'obtenir un service.

Dans certains cas, un client à besoin de recevoir plusieurs traitements avant de quitter le système. Par exemple, dans les systèmes de production, les banques, les systèmes informatiques d'où la notion des réseaux de files d'attente.

Les réseaux de files d'attente se composent de plusieurs files interconnectées. Ils sont

bien établis en tant qu'outils analytiques puissants d'analyses et de modélisation des systèmes. La raison principale de leur succès est dans la combinaison de la puissance expressive[3].
Les réseaux de files d'attente sont classés en deux catégories :

- Les réseaux de files d'attente mono-classe, dans lesquels circulent une classe de clients, c'est-à-dire un seul type d'entités, statistiquement indifférenciables ;
- Les réseaux de files d'attente multi-classes, dans lesquels circulent plusieures classes de clients, pouvant se distinguer par un schéma de routage spécifique et par des comportements différents au niveau de chaque station, tant au service que de l'ordonnancement dans le buffer d'attente.

Dans le cas de réseaux mono-classe, on fait également la distinction entre :

- Réseaux ouverts ;
- Réseaux fermés.

Dans le cas de réseaux multi-classes, il faut préciser pour chaque classe de clients s'il s'agit d'une classe ouverte où d'une classe fermée. Si toutes les classes de clients sont des classes ouvertes, on parlera de réseaux purement ouverts et si toutes les classes de clients sont des classes fermées, on parlera de réseaux purement fermés[3].

Dans les réseaux de files d'attente , il existe une classe particulière de réseaux , connue sous le nom de réseaux à *forme produit* qui ont la particularité de posséder une solution analytique très simple, soit ouvert où fermé tel que le réseau de *Jackson*, multi-classes comme le réseau $BCMP$.

L'objectif d'une étude de réseau de file d'attente est d'améliorer et mieux évaluer les performances des systèmes. Les critères de performance ne sont pas toujours les mêmes du point de vue d'un utilisateur, d'un administrateur et d'un concepteur, car leurs préoccupations ne sont pas les mêmes. Ces trois individus ont des demandes parfois communes et parfois contradictoires. Mais quel que soit le critère de performance considéré par un individu, ce critère prend la forme "rapport coût-performance". Les critères de performance les plus importants sont [14] :

- Le temps moyen de réponse : le temps séparant l'arrivée d'un client de la fin de son traitement ;
- Le débit : le nombre de clients traités par unité de temps ;

L'objectif de notre travail, dans ce mémoire, est se familiariser des réseaux de files d'attente, en général, et l'étude des réseaux de *Jackson* et leur généralisation avec les réseaux $BCMP$ en particulier. Ces derniers appartiennent à la classe des réseaux Markoviens à forme produit basés principalement sur les files d'attente $M/M/..$ Nous avons donné, par la suite, quelques relations existantes pour déterminer les paramétres de performance moyens, en l'occurrence le moyen temps, le nombre moyen de clients,..., de ce type de réseaux. Une application des réseaux de Jackson, dans le cas ouvert et fermé, a été réalisée dans le cas de systèmes informatiques.

Ce memoire s'articule autour de trois chapitres :

- Dans le premier chapitre, nous rappelons certaines notions et certaines caractéristiques relatives aux systèmes de files d'attente markoviens et non-markoviens.
- Le deuxième chapitre présente les réseaux de files d'attente mono-classe et multi-classes. Nous avons consacré la dernière partie de ce chapitre aux réseaux à forme produit (réseau de Jackson et BCMP).
- Une application des réseaux de Jackson et BCMP aux systèmes informatiques a été donnée au dernier chapitre.

Chapitre 1

# Systèmes de files d'attente

## 1.1 Introduction :

La théorie de files d'attente est une technique de la recherche opérationnelle qui permet de modéliser un système admettant un phénomène d'attente, de calculer ses performances et de déterminer ses caractéristiques pour aider les praticiens dans leurs prises de décisions. Des résultats et formulations théoriques sont bien établis pour les modèles de files d'attente avec arrivées poissonniennes et les durées de services exponentielles [28].

Dans ce chapitre, nous allons présenter les différents modèles de files d'attente à savoir : les modèles markoviens, les modèles non markoviens et leurs caractéristiques.

## 1.2 Description du phénomène d'attente :

Une file d'attente ou queue est un système stochastique composé d'un certain nombre (fini ou non) de places d'attente d'un ou plusieurs serveurs et bien sûr de clients qui arrivent, attendent, se font servir selon des règles de priorité données et quittent le système. La description précédente d'une file d'attente, dont une représentation schématique est donnée en figure (1.1), ne saurait capturer toutes les caractéristiques des différents modèles que comptent la littérature, mais elle identifie les éléments principaux permettant la classification de la grande majorité des files d'attente simples [11]. Tout système de file d'attente peut être représenté par le schéma suivant :

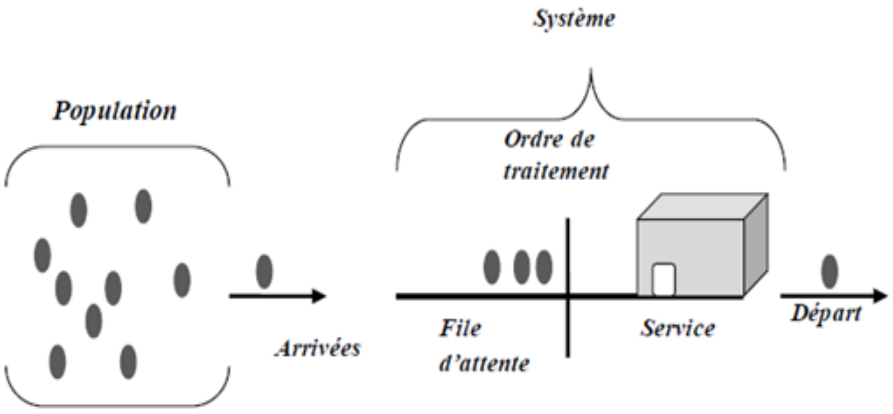


FIG. 1.1 – Structure générale d'un système de file d'attente

## 1.3 Classification des systèmes d'attente :

Pour identifier un système d'attente, on a besoin des spécifications suivantes [27] :

✓ La nature stochastique du processus des arrivées, qui est défini par la distribution des intervalles séparant deux arrivées consécutives ;

✓ La distribution du temps aléatoire de service ;

✓ Le nombre $m$ de serveurs (stations de service) qui sont montées en parallèle. On admet généralement que les temps de service correspondants suivent la même distribution et que les clients qui arrivent forment une seule file d'attente (dans le cas homogène) ;

✓ La capacité $N$ du système. Si $N < \infty$, la file d'attente ne peut dépasser une longueur de $N - m$ Unités. Dans ce cas, certains clients arrivant vers le système n'ont pas la possibilité d'y entrer ;

✓ la source des clients potentiels.

## 1.4 Mesures de performance d'une file d'attente :

L'étude d'une file d'attente ou d'un réseau de files d'attente a pour but de calculer ou d'estimer les performances du système dans des conditions de fonctionnement données. Ce calcul se fait le plus souvent pour le régime stationnaire uniquement, et les mesures les plus fréquemment utilisées sont :

- $\overline{N}$ : nombre moyen de clients dans le système ;

- $\overline{Q}$ : nombre moyen de clients dans la file d'attente ;
- $\overline{T}$ : temps moyen de séjour d'un client dans le système ;
- $\overline{W}$ : temps moyen d'attente d'un client dans la file ;
- $\overline{U}$ : taux d'utilisation de chaque serveur ;
- $\overline{S}$ : le temps moyen de service ;
- $\overline{A}$ : le temps moyen entre deux arrivées.

***Remarque* 1.4.1.** *De manière générale, une file est stable si et seulement si le nombre moyen d'arrivées de clients par unité de temps, noté $\lambda$ , est inférieur au nombre moyen des clients pouvant être servis par unité de temps. Si chaque serveur peut traiter $\mu$ clients par unité de temps et si le nombre de serveurs est $m$, une file est stable si et seulement si*

$$\lambda < m\mu \Leftrightarrow \rho = \frac{\lambda}{m\mu},$$

*où, $\rho$ est appelé l'intensité du trafic.*

### 1.4.1 Formule de Little

Soient $\lambda$ le taux des arrivées, $\lambda_e$ le taux réel des arrivées (taux d'entrée), $\overline{T}$ le temps moyen de séjour et $\overline{N}$ le nombre moyen de clients présents dans le système. De plus, si $\lambda$, $\overline{N}$ et $\overline{T}$ existent, ils sont reliés l'un à l'autre par l'équation suivante :

$$\overline{N} = \lambda_e \overline{T}. \tag{1.1}$$

Cette expression appelée formule de Little [22] est l'un des résultats les plus généraux et utile dans la théorie des files d'attente.
En utilisant cette formule, on obtient également :

- $\overline{Q}$=$\lambda\overline{W}$ ;
- $\overline{T}$=$\overline{W} + \frac{1}{\mu}$ ; où $\mu$ représente le taux de service ;
- $\overline{N}$=$\overline{Q} + \rho$.

## 1.5 Les différentes disciplines de service :

La discipline de service, est la règle de priorité déterminant l'ordre dans lequel les clients vont accéder à la ressource modélisé par le serveur. Les disciplines d'attente classiques, ainsi que leurs acronymes, sont :

- ✓ **FIFO** : *(first in, first out)* ou *FCFS (first come first served)* ou *PAPS (premier arrivé, premier servi)* : c'est la file standard dans laquelle les clients sont servis dans leur ordre d'arrivée. Notons que les disciplines *FIFO* et *FCFS* ne sont pas équivalentes lorsque la file contient plusieurs serveurs. Dans la première, le premier client arrivé sera le premier à quitter la file alors que dans la deuxième, il sera le premier à commencer son service. Rien n'empêche alors qu'un client qui commence son service après lui, dans un autre serveur, termine avant lui. En français, le terme PAPS comporte une ambiguïté, puisqu'il ne peut différencier une file "premier arrivé, premier servi" d'une file "premier arrivé, premier sorti".
- ✓ **LIFO** : *(last in, first out)* où *LCFS (last come, first served)* où *DAPS (dernier arrive, premier servi)*. Cela correspond à une pile, dans laquelle le dernier client arrivé *(donc posé sur la pile)* sera le premier traité *(retiré de la pile)*. A nouveau, les disciplines *LIFO* et *LCFS* ne sont pas équivalentes que pour une file mono-serveur.
- ✓ **RANDOM** *(aléatoire)* : Le prochain client qui sera servi est choisi aléatoirement dans la file d'attente.
- ✓ **Round-Robin** *(cyclique)* : Tous les clients de la file d'attente entrent en service à tour de rôle, effectuant un quantum $Q$ de leur temps de service et sont replacés dans la file, jusqu'à ce que leur service soit totalement accompli. Cette discipline de service a été introduite afin de modéliser des systèmes informatiques.
- ✓ **PS** *(Processor Sharing)* : C'est le cas limite de la distribution *Round-Robin* lorsque le quantum de temps $Q$ tend vers 0. Tous les clients sont servis en même temps, mais avec une vitesse inversement proportionnelle au nombre de clients simultanément présents. Si le taux du service est égal à $\mu$ et qu'à un instant donné il y a $n$ clients dans la station, tous les clients sont donc servis simultanément avec un taux $\frac{\mu}{n}$.
- ✓ **Avec priorité** : Chaque client a une priorité (statique ou dynamique, absolue ou relative), le serveur sélectionne le client de haute priorité.
  - ◦ *Priorité relative :* Un client accède au service selon sa priorité. La file est gérée par ordre de priorité de la plus forte à la plus faible.
  - ◦ *Priorité absolue :* Le service d'un client est interrompu lorsqu'un client de priorité supérieure se présente devant la file d'attente. Le client dont ce service est interrompu est remis en tête de la file.

## 1.6 Notation de Kendall :

Un modèle de file d'attente est totalement décrit selon la notation de Kendall. Dans sa version étendue, un modèle est spécifié par une suite de six symboles [27] :

$$A/B/s/N/K/D$$

La signification de chacun de ces symboles est :

- ✓ $A$ : Nature du processus des arrivées ;
- ✓ $B$ : Nature du processus de service ;
- ✓ $s$ : Nombre de serveurs en parallèle ;
- ✓ $N$ : Capacité du système (serveurs + file d'attente) ;
- ✓ $K$ : Taille de la population ;
- ✓ $D$ : Discipline de la file.

Dans la description des processus d'arrivée et de service, les symboles les plus courants sont :

- ✓ M : Distribution exponentielle *(qui vérifie donc la propriété de Markov)* ;
- ✓ E : Distribution d'Erlang ;
- ✓ G : Distribution générale *(on ne sait rien sur ses caractéristiques)* ;
- ✓ D : loi Déterministe *(temps d'inter-arrivées ou de service constant)* ; La forme abrégé : $A/B/s$ signifie que $N$ et $K$ sont infinies

## 1.7 Analyse mathématique d'un système de files d'attente :

L'étude mathématique d'un système de files d'attente se fait généralement par l'introduction d'un processus stochastique, défini de façon appropriée. On s'intéresse principalement au nombre de clients $X(t)$, se trouvant dans le système à l'instant $t$ $(t \geq 0)$.

En fonction des quantités qui définissent le système, on cherche à déterminer :

• Les probabilités d'état $P_n(t) = P(X(t) = n)$ , qui définissent le régime transitoire du processus stochastique $\{X(t), t \geq 0\}$. Il est évident que les fonctions $P_n(t)$ dépendent de l'état initial ou de la distribution initiale du processus.

• Le régime stationnaire du processus stochastique est défini par :

$$\pi_n = \lim_{n\to\infty} p_n(t) = P(X(+\infty) = n), n = 1, 2, 3, \ldots \tag{1.2}$$

où ,$\{\pi_n\}_{n\geq 0}$ est appelée distribution stationnaire du processus $\{X(t), t \geq 0\}$. A partir de cette distribution on pourra obtenir d'autres caractéristiques d'exploitation du système telle que :

- ⋆ Le nombre moyen de clients dans le système $L = E(X)$ ;
- ⋆ La durée d'attente d'un client ;
- ⋆ La durée moyen de séjour dans le système qui est composée de la durée moyen d'attente et la durée moyen de service ;
- ⋆ Le taux d'occupation des postes de service ;
- ⋆ Le pourcentage de clients n'ayant pu être servi ;
- ⋆ La durée moyen d'une période d'activité, c'est-à-dire l'intervalle de temps pendant lequel il y a toujours au moins un client dans le système.

***Remarque* 1.7.1.** *Il faut toutefois, constater que le calcul explicite du régime transitoire s'avère pénible, voire impossible, pour la plupart des modèles considérés, mis à part certains modèles particulièrement faciles à traiter. Nous nous contenterons donc dans la suite de déterminer le régime stationnaire d'un système d'attente. Notons qu'il existe des systèmes d'attente dont l'évolution temporelle n'est plus déterminée par le processus* $\{X(t), t \geq 0\}$,

## 1.8 Les files d'attente markoviennes :

Les modèles markoviens caractérisent les systèmes dans lesquels les deux quantités stochastiques principales, qui sont le temps inter-arrivées et la durée de service, sont des variables aléatoires indépendantes et exponentiellement distribuées. La propriété d'absence de mémoire de la loi exponentielle facilite l'étude de ces modèles. L'étude mathématique de tels systèmes se fait par l'introduction d'un processus stochastique approprié. Ce processus est souvent le processus $\{X(t), t \geq 0\}$ défini comme étant le nombre de clients dans le système à l'instant t. L'évolution temporelle du processus markovien est complètement définie grâce à la propriété d'absence de mémoire.

### 1.8.1 Système d'attente M/M/1 :

Le système de files d'attente M/M/1 est le système le plus élémentaire de la théorie des files d'attente. Le flot des arrivées est poissonnien de paramètre $\lambda$ et la durée de service est exponentielle de paramètre $\mu$.

#### Régime transitoire :

Soit $X(t)$ le nombre de clients présents dans le système à l'instant $t$ ($t > 0$). Grâce aux propriétés fondamentales du processus de Poisson et de la loi exponentielle, $X(t)$ est un processus markovien homogène. Les probabilités d'état $P_n(t) = P(X(t) = n)$ peuvent être calculées par les équations différentielles de Kolmogorov ci-dessous, connaissant les conditions initiales du processus.

$$p'_n(t) = -(\lambda + \mu)p_n(t) + \lambda p_{n-1}(t) + \mu p_{n+1}(t)$$

et

$$p_0(t) = -\lambda p_0(t) + \mu p_1(t)$$

#### Régime stationnaire :

Sous la condition d'ergodicité du système $\rho = \frac{\lambda}{\mu}$, pour laquelle le régime stationnaire existe, il est aisé d'obtenir les probabilités stationnaires

$$\pi_n(t) = \lim_{t\to\infty} p_n(t) = (1-p)p^n, \ \forall n \in \mathbb{N}.$$

$\pi=\{\pi_n\}_{n\geq 0}$ est appelé distribution stationnaire, elle suit une loi géométrique. Les caractéristiques de ce système sont données par les propositions suivantes :

**Proposition 1.1.** *[1] Le nombre moyen de clients dans le système est :*

$$\overline{N} = E(X) = \sum_{n=0}^{+\infty} n\pi_n = (1-\rho)\sum_{n=0}^{+\infty} n\rho^n. \tag{1.3}$$

*D'où :*

$$\overline{N} = \frac{\rho}{1-\rho}. \tag{1.4}$$

**Proposition 1.2.** *[1] Le nombre moyen de clients dans la file est :*

$$\overline{Q} = \sum_{n\geq 1}^{+\infty}(n-1)\pi_n = \frac{\rho^2}{1-\rho}. \tag{1.5}$$

***Remarque* 1.8.1.** *Le temps moyen de séjour dans le système $\overline{T}$ et le temps moyen d'attente dans la file $\overline{W}$ sont obtenus à partir des formules de Little.*

**Proposition 1.3.** *[1] Le temps moyen de séjour dans le système est :*

$$\overline{T} = \frac{\overline{N}}{\lambda} = \frac{1}{\mu(1-\rho)}. \tag{1.6}$$

**Proposition 1.4.** *[1] Le temps moyen d'attente dans la file est :*

$$\overline{W} = \frac{\overline{Q}}{\lambda} = \frac{\rho}{\mu-\lambda} = \frac{\rho}{\mu(1-\rho)}. \tag{1.7}$$

**Théorème 1.1.** *[8] Sous la condition de stabilité $\lambda < \mu$, à l'équilibre, le processus des départs d'une file $M/M/1$ a la même loi que le processus des arrivées. De plus, le nombre de clients dans la file à l'instant $t = t_0$ est indépendant du processus des départs avant $t = t_0$.*

### 1.8.2 La file d'attente M/M/m :

Dans ce modèle, $m$ serveurs identiques et indépendants partagent les mêmes places d'attente.

Les arrivées suivent un processus de Poisson de paramètre $\lambda$ et la durée de chaque service est une variable exponentielle de paramètre $\mu$. Les caractéristiques de ce système sont données par les relations suivantes :

**Proposition 1.5.** *[1] La probabilité qu'il y ait n clients dans le système à l'instant d'entrée est :*

$$p_n = \begin{cases} \frac{(\frac{\lambda}{\mu})^n}{n!}p_0, & si \;\; n \leq m; \\ \frac{(\frac{\lambda}{\mu})^n}{m!m^{n-m}}p_0, & si \;\; n \geq m. \end{cases} \tag{1.8}$$

*Où*

$$p_0 = [\sum_{n\geq 0}^{m}\frac{(\frac{\lambda}{\mu})^n}{n!} + \frac{(\frac{\lambda}{\mu})^m + 1}{m!(m-\frac{\lambda}{\mu})}]^{-1}. \tag{1.9}$$

$$\zeta = P(attente) = P(X \geq m) = \frac{p_m}{1-\rho}. \tag{1.10}$$

***Remarque* 1.8.2.** Le taux d'utilisation de chaque serveur est :

$$U = \rho = \frac{\lambda}{m\mu}. \tag{1.11}$$

**Proposition 1.6.** *[1] Le nombre moyen de clients présents dans le système et en attente sont respectivement :*

$$\overline{N} = m\rho + \frac{\rho\zeta}{1+\rho}, \tag{1.12}$$

$$\overline{Q} = \frac{\rho\zeta}{1-\rho}. \tag{1.13}$$

**Proposition 1.7.** *[1] Le temps moyen de réponse et d'attente sont respectivement :*

$$\overline{T} = \frac{1}{\mu}(1 + \frac{\zeta}{m(1-\rho)}); \tag{1.14}$$

$$\overline{W} = \frac{\zeta}{m\mu(1-\rho)}. \tag{1.15}$$

### 1.8.3 La file d'attente M/M/m/K :

La file *M/M/m/K* est une file markovienne composée de $m$ serveurs et disposant de $K$ places au total. Le nombre maximal de clients en attente est donc $K-m$. Si un client arrive alors que le système est plein, il ne peut yait entrer et doit repartir. Elle est donc toujours stable quel que soit l'intensité du trafic $\rho = \frac{\lambda}{\mu} < 1$.
Le taux de service de cette file est :

$$\mu_k = \begin{cases} k\mu, & k = 1, 2, \cdots, m-1; \\ m\mu, & k = m, m+1 \cdots K. \end{cases} \tag{1.16}$$

Comme tout client arrivant alors que le système est plein doit repartir, le taux effectif d'arrivées dans le système n'est pas $\lambda$ mais $\lambda_e = \sum_{k\geq 0}^{K-1} \lambda p_k = \lambda(1-p_k)$ où, $p_k$ est la probabilité qu'il y $k$ clients dans le système.
La distribution stationnaire est donné d'aprés la distribution du Processus de Naissance et de Mort par :

$$p_n = \prod_{i=1}^{n} \frac{\lambda_{i-1}}{\mu_i} p_0; \tag{1.17}$$

$$p_n = \left(\frac{\lambda}{\mu}\right)^n \frac{1}{n!}\ p_0 \quad \text{si} \quad 0 \leq n \leq m \tag{1.18}$$

si $m \leq n \leq K$

$$p_n = \left(\frac{\lambda}{m\mu}\right)^{n-m} \frac{\left(\frac{\lambda}{\mu}\right)^m}{m!}\ p_0 \tag{1.19}$$

D'ou :

$$p_n = \begin{cases} \frac{\rho^n}{n!} p_0, & \text{si } 0 \leq n \leq m; \\ \frac{\rho^n}{m! m^{n-m}} p_0, & \text{si } m \leq n \leq K. \end{cases}$$

Avec :

$$p_0 = \frac{1}{\sum\limits_{n=0}^{m} \frac{\rho^n}{n!} + \sum\limits_{n=m+1}^{K} \frac{\rho^n}{m! m^{n-m}}} \tag{1.20}$$

**Proposition 1.8.** *[1] On a :*

$$p(perte) = P(\ \textit{tous les serveurs sont occupés}\ ) = p_m \tag{1.21}$$

$$p_m = \frac{\rho^n}{m! m^{n-m}} p_0 \tag{1.22}$$

*avec $p_0$ est danée par (1.20)*

En ce qui concerne les caractéristiques du système, on a

$$\overline{N} = \sum_{k \geq 1}^{K} k p_k; \tag{1.23}$$

et

$$\overline{Q} = \sum_{k \geq m+1}^{K} (k - m) p_k. \tag{1.24}$$

C'est ce taux effectif $\lambda_e$ qu'il faut utiliser pour calculer $\overline{T}$ et $\overline{W}$ à l'aide des formules de Little.

### 1.8.4 La file d'attente M/M/$\infty$ :

Cette file est composée d'une infinité de stations de service identiques. Il est évident qu'aucune file d'attente ne se forme ; chaque client est servi dès son entrée. Ce système possède non seulement un intérêt théorique, mais il permet des études approximatives de phénomène d'attente de type *M/M/m* ou *M/M/m/m* comprenant un grand nombre de

stations en parallèle [27].
La distribution stationnaire de ce système d'attente est :

$$p_n = (\frac{\lambda}{\mu})^n \quad \frac{\exp -\frac{\lambda}{\mu}}{n!} \quad (n = 0, 1, 2, \cdots). \tag{1.25}$$

En ce qui concerne les caractéristiques du système, on a

$$\overline{N} = \frac{\lambda}{\mu} \quad et \quad \overline{T} = \frac{1}{\mu}. \tag{1.26}$$

Tandis que

$$\overline{Q} = \overline{W} = 0. \tag{1.27}$$

### 1.8.5 Système d'attente $M^{[X]}/M/1$ :

C'est un système $M/M/1$ pour lequel les arrivées se présentent en groupe. Le nombre de clients par groupe est une variable aléatoire $X$ strictement positive, et on note
$P(X = x) = C_n$.
En notant $P(n)$ la probabilité que le système soit à l'état $n$. Les équations de l'état d'équilibre sont données dans [24], sous la forme :

$$\begin{cases} (\lambda + \mu)p(n) = \mu p(n+1) + \lambda \sum_{k=1}^{n} p(n-k)C_k, \quad si \quad n \geq 1; \\ \lambda p(0) = \mu p(1). \end{cases} \tag{1.28}$$

Où $\lambda$ est le taux d'arrivée, et $\mu$ est le taux de service.
Soit les fonctions génératrices suivantes :

$$P(x) = \sum_{n=0}^{\infty} p(n)x^n \quad et \quad G(x) = \sum_{n=0}^{\infty} C_n x^n. \tag{1.29}$$

En multipliant les équations (1.22) par $x^n$ et en sommant sur $n$ on obtient :

$$\lambda P(x) + \mu[P(x) - p(0)] = \frac{\mu}{x}[P(x) - p(0)] + \lambda G(x)P(x). \tag{1.30}$$

D'où la fonction génératrice suivante :

$$P(x) = \frac{\mu p(0)(1-x)}{\mu(1-x) - \lambda x[1 - G(x)]} \quad si \quad |x| < 1. \tag{1.31}$$

Pour déterminer les valeurs de $P(0)$ on utilise la condition $p(1) = 1$. En faisant tendre $x$ vers 1 dans la relation (1.25), on obtient :

$$\frac{-\mu p(0)}{-\mu + \lambda E(X)} = 1,$$

donc

$$p(0) = 1 - \frac{\lambda E(X)}{\mu}.$$

En posant $\rho = \frac{\lambda E(X)}{\mu}$, où $E(X)$ est l'espérance de $X$, on aura $p(0) = 1 - \rho$. La condition de stabilité est donc $\rho < 1$ c'est-à-dire, $\frac{\lambda E(X)}{\mu} < 1$.

## 1.9 Les files d'attente non markoviennes :

En l'absence de l'exponentialité ou lorsque l'on s'écarte de l'hypothèse d'exponentialité de l'une des deux quantités stochastiques : le temps des inter-arrivées et la durée de service, ou en prenant en compte certaines spécificités des problèmes par introduction de paramètres supplémentaires, on aboutit à un modèle non markovien.
La combinaison de tous ces facteurs rend l'étude mathématique du modèle très délicate, voire impossible- on essaye alors de se ramener à un processus de Markov judicieusement choisi à l'aide de l'une des méthodes d'analyse suivantes :

1. **Méthode des étapes d'Erlang :**
   Son principe est d'approximer toute loi de probabilité ayant une transformation de Laplace rationnelle par une loi de *Cox* (mélange de lois exponentielles), cette dernière possède la propriété d'absence de mémoire par étape [1].

2. **Méthode de la chaîne de Markov induite :**
   Elaborée par Kendall, et souvent utilisée, elle consiste à choisir une suite d'instants $1, 2, 3, \cdots, n$ (déterministes ou aléatoires) tels que la chaîne induite $\{X_n, n \geq 0\}$, où $X_n = X(n)$ , soit markovienne et homogène.

3. **Méthode des variables supplémentaires :**
   Elle consiste à compléter l'information sur le processus $\{X_t, t \geq 0\}$ de telle manière à lui donner le caractère markovien. Ainsi, on se ramène à l'étude du processus $\{X(t), A(t_1), A(t_2), A(t_3), \cdots A(t_n), t \geq 0\}$. Les variables A $(t_k), k \in \{1, 2, \cdots n\}$ sont dites supplémentaires.

4. **Méthode des événements fictifs :**
   Le principe est d'introduire des événements fictifs qui permettent de donner une

interprétation probabiliste aux transformées de Laplace et aux variables aléatoires décrivant le système étudié.

5. **Simulation :**
   C'est un procédé d'imitation artificielle d'un processus réel effectué sur ordinateur. Elle nous permet d'étudier les systèmes les plus complexes, de prévoir leurs comportements et de calculer leurs caractéristiques. Les résultats obtenus ne sont qu'approximatifs, mais peuvent être utilisés avec une bonne précision. Cette technique se base sur la génération de variables aléatoires suivant les lois gouvernant le système.

### 1.9.1 Système d'attente $M/G/1$

Dans la file $M/G/1$, le temps de service ne suit plus une loi exponentielle mais une loi non négative quelconque d'espérance $\overline{S}$ et de variance $\sigma_S^2$ finies. Le processus stochastique décrivant l'évolution du nombre de client dans le système n'est plus une chaîne de Markov car le temps de service n'est plus sans mémoire, pour obtenir un processus markovien, il faudrait étendre la définition de l'état du système afin d'inclure également la durée de service déjà réçue par le client occupant le serveur.
Une autre approche consiste à n'observer le système qu'aux instants de fin de service, on obtient ainsi une chîne de Markov sous-jacente à temps discret [18].

- **La formule de Pollaczeck-Khinchin**

La formule de Pollaczeck-Khinchin est un résultat très élégant montrant que les différences de performances entre une file $M/G/1$ est une file $M/M/1$ se résume à un facteur multiplicatif.

**Théorème 1.2.** ***Formule de Pollaczeck-Khinchin*** *[18]*
*Le nombre moyen de clients en attente dans une file d'attente $M/G/1$ sous la condition de stabilité $\rho = \lambda\overline{S} < 1$, est donné par :*

$$\overline{Q} = (\frac{1 + C_2^S}{2})\frac{\rho^2}{1+\rho}, \tag{1.32}$$

*où $C_2^S$ est le coefficient de variation au carée du temps de service ($C_2^S = \frac{\sigma_S^2}{\overline{S}^2}$).*

En caractérisant les performances par le type de file en question, l'expression (1.26) devient :

$$\overline{Q}_{M/G/1} = (\frac{1+C_2^S}{2})\overline{Q}_{M/M/1}.$$

D'après la formule de Little, le temps moyen d'attente est donné par :

$$\overline{W}_{M/G/1} = (\frac{1+C_2^S}{2})\overline{W}_{M/M/1} = (\frac{1+C_2^S}{2})\frac{\rho\overline{S}}{1+\rho}. \tag{1.33}$$

Pour le calcul de $\overline{N}$ et $\overline{T}$ on utilise les relations suivantes :

$$\overline{N} = \overline{Q} + U = (\frac{1+C_2^S}{2})\frac{\rho^2}{1-\rho} + \rho, \tag{1.34}$$

et

$$\overline{T} = \overline{W} + \overline{S} = (\frac{1+C_2^S}{2})\frac{\rho^2\overline{S}}{1-\rho} + \overline{S}, \tag{1.35}$$

le taux d'utilisation du serveur étant toujours $U = \rho$.

### 1.9.2 Système d'attente $G/M/1$

C'est un système de file d'attente qui peut être considéré comme un système symétrique au précédent $M/G/1$. Il possède un processus d'arrivées général caractérisé par les intérvalles de temps entre chaque deux arrivées successive indépendants et identiquement distribués et une distribution de service exponentielle de taux $\mu = \frac{1}{\overline{S}}$ [18].

- Le temps moyen d'attente dans la file est donné par :

$$\overline{W} = \frac{\sigma\overline{S}}{1-\sigma} \tag{1.36}$$

- Le nombre moyen de clients dans la file d'attente est donné par :

$$\overline{N} = \frac{\rho\sigma}{1-\sigma} \tag{1.37}$$

## 1.10 Conclusion

Dans ce chapitre nous nous sommes intéressés à quelques notions de base sur la théorie de file d'attente. Ce chapitre est consacré à la présentation des systèmes d'attente classiques répartis en deux sections différentes à savoir ; les systèmes markoviens et les systèmes non markoviens ainsi que leurs caractéristiques.

Dans la pratique, les systèmes de files d'attente ne parmettent pas de modéliser une multitude de problèmes, car un client a besoin de plusieurs stations inter-connectées pour terminer son service. A cet effet, il faut considérer des réseaux de files d'attente.

Chapitre 2

# Réseaux de files d'attente

## 2.1 Introduction

Un réseau de file d'attente est un ensemble de files d'attente inter-connectées , dans lesquelles circulent une ou plusieurs classes de clients. Chaque classe se caractérise par un schéma de routage, par des comportements différents au niveau de chaque station de service et de l'ordonnancement dans la file d'attente. On peut distinguer différentes classes de clients :

- Les processus d'arrivés différent ;
- Les comportements des clients qui sont différents à chaque station ;
- Les différents chemins dans le réseau.

Dans ce chapitre, nous allons tout d'abord présenter les différents réseaux de files d'attente. Ensuite , nous nous intéresserons à une classe particulière de réseaux de files d'attente, connue sous le nom de *réseaux à forme produit*, qui ont la particularité de posséder une solution analytique très simple. Nous considérerons successivement les réseaux mono-classes ouverts, les réseaus mono-classes fermés, puis les réseaux multi-classes.

## 2.2 Les réseaux de files d'attente

### 2.2.1 Réseau mono-classe ouvert :

Dans un réseau de files d'attente ouvert, les clients arrivent de l'extérieur, circulent dans le réseau à travers les différentes stations, puis quittent le réseau. Le nombre de clients pouvant se trouver à un instant donné dans un réseau ouvert n'est donc pas limité.

Afin de spécifier complètement un réseau ouvert, il faut bien sûr caractériser chaque station, mais également le processus d'arrivée des clients et le routage (cheminement) des clients dans le réseau [14].

- **Le processus d'arrivée :** L'arrivée des clients dans le réseau sera décrite (comme pour une file simple) à l'aide d'un processus de renouvellement (et sera donc caractérisé par la distribution du temps d'inter-arrivée).
  Si l'arrivée des clients suit un processus de poisson, les inter-arrivées sont exponentielles et sont caractérisées par un unique paramètre : le taux d'arrivée $\lambda$. Il faut préciser, lorsqu'un client arrive dans le réseau, à quelle file il se rend. On caratérisera la plupart du temps le routage d'entrée de façon probabiliste : soit $p_{0i}$ la probabilité pour qu'un client qui arrive, se rende à la station $i$. Les probabilités $p_{0i}$ sont bien sûr telles que $\sum_{i=1}^{M} p_{0i} = 1$ ; où $M$ : le nombre de stations ;
- **Routage des clients :** Lorsqu'un client termine son service à une station, il faut préciser où ce client va se rendre : soit à une autre station, soit à l'extérieur (le client quitte alors le réseau). A nouveau, le routage des clients est très souvent caractérisé de façon probabiliste : soit $p_{ij}$ la probabilité pour qu'un client qui quitte la station $i$ se rende à la station $j$ et soit $p_{i0}$ la probabilité pour qu'un client qui quitte la station $i$ quitte le système. Les $p_{ij}$ sont tels que $\sum_{j=0}^{M} p_{ij} = 1$. La figure ci-dessous illustre un exemple de réseau mono-classe ouvert :

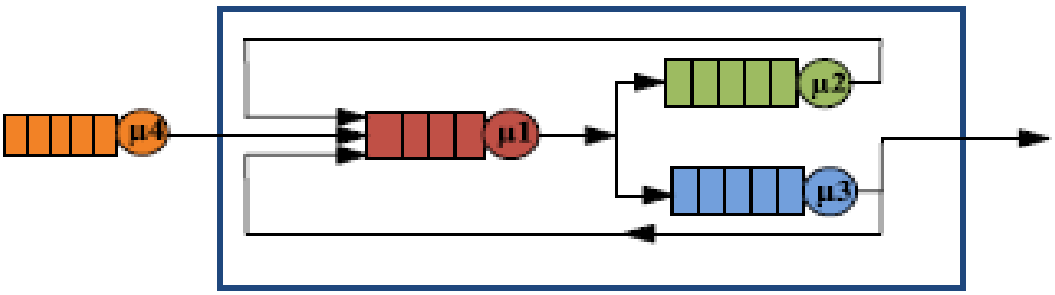


FIG. 2.1 – Réseau mono-classe ouvert

**Exemples.** *: Remontées mécaniques au ski, caisses de grandes surfaces.*

### 2.2.2 Réseau mono-classe fermés :

Dans un réseau de files d'attente fermé avec des inter-arrivées et services exponentielles, les clients sont en nombre constant. Soit $N$ le nombre total de clients du système. Il n'y a donc pas d'arrivée ni départ de clients. La spécification d'un réseau fermé se réduit donc à celle des différentes stations et à celle du routage des clients.
Pour un mécanisme de routage probabiliste, on définit $p_{ij}$ la probabilité qu'un client qui quitte la station $i$ se rende à la station $j$. Les $p_{ij}$ sont tels que :

$$\sum_{j=1}^{M} p_{ij} = 1.$$

La figure ci-dessous illustre un exemple de réseau mono-classe fermé :

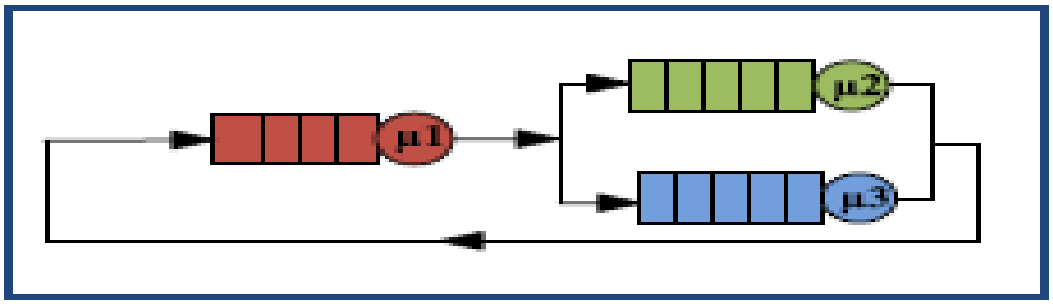


FIG. 2.2 – Réseau mono-classe fermé

**Exemples.** *: Palettes dans un atelier.*

### 2.2.3 Réseaux multi-classes :

Les réseaux de files d'attente peuvent être parcourus par différentes classes de clients. Soit $C$ le nombre de classes de clients. Ces différentes classes peuvent se distinguer par : des processus d'arrivée différents (si le réseau est ouvert), des comportements différents à chaque station (service et discipline de service) et des routages différents dans le réseau. On est alors amené à caractériser pour chaque classe $c$

- **Pour un réseau ouvert :** le processus d'arrivée (pour un processus d'arrivée poissonien, il suffit alors de donner le taux d'arrivée $\lambda_c$ des clients de classe $c$) ;
- **Pour un réseau fermé :** le nombre total $N_c$ de clients de classe $c$ ;

- **Le routage des clients :** Si la notion de réseau multi-classes nous permet d'introduire la notion de réseau mixte qui est un réseau ouvert vis-à-vis de certaines classes et fermé vis-à-vis des autres classes. Les clients peuvent changer de classes lors de leurs cheminement dans le réseau. On définit alors la probabilité $p_{c_i,s_j}$ pour qu'un client de classe $c$ qui quitte la station $i$ se rende à la station $j$ et se transforme en un client de classe $s$.
  La figure ci-dessous illustre un exemple de réseau multi-classes :

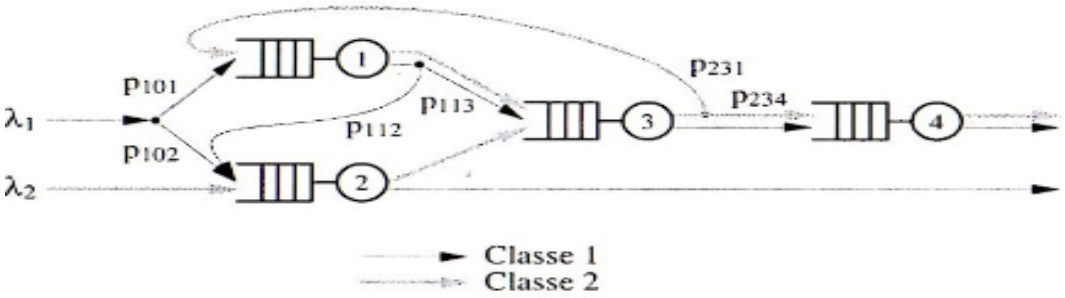


FIG. 2.3 – Réseaux multi-classes ouvert

### 2.2.4 Réseau ouvert à contrainte de population :

Certaines réseaux de files d'attente, bien qu'étant des modèles ouverts, peuvent être soumis à une limite supérieure sur le nombre total de clients pouvant s'y trouver simultanément. Cette "contrainte de population" implique que le réseau n'est ni un modèle ouvert, puisque le nombre de clients qui peuvent s'y trouver est limité, ni réellement un réseau fermé, puisque le nombre total de clients dans le système n'est pas constant. On parlera de "modèle ouvert à contrainte de population". Lorsqu'un client arrive dans le réseau alors que celui-ci est plein (la contrainte de population est atteinte), deux cas peuvent être envisagés. Soit le client est "rejeté", ce qui rejoint le modèle de la section précédente, soit le client est "mémorisé" et se place en attente dans une file externe.
La figure ci-dessous illustre un exemple de réseau ouvert à contrainte de population :

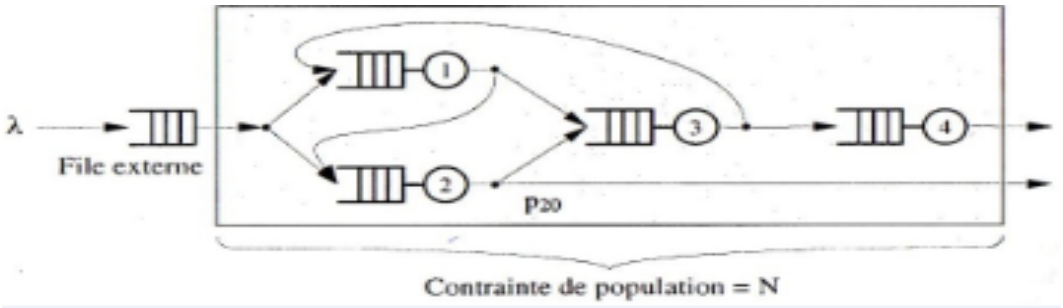


FIG. 2.4 – Réseau ouvert à contrainte de population

### 2.2.5 Réseaux à capacité limitées :

Les différentes stations du réseau peuvent avoir des capacités limitées. Lorsqu'une file est pleine, plus aucun client ne peut y entrer. Cela induit des blocages dans les stations en amonts et éventuellement des pertes de clients à l'entrée du système (si celui-ci est ouvert).

On distingue principalement deux types de blocage [21] :

- **Blocage avant service (BBS : Blocking Before Service) :** dans un blocage avant service (ou blocage de type réseau de communication), un client voulant commencer son service à une station donnée doit tout d'abord s'assurer qu'il y a une place libre dans la station de destination. Si c'est le cas, son service commence. Dans le cas contraire, le serveur de la station est bloqué et le client doit attendre la libération d'une place en aval avant de commencer son service ;
- **Blocage après service (BAS : Blocking After Service) :** dans un mécanisme de blocage après service (ou blocage de type système de production) un client commence son service sans attendre, dès l'instant où le serveur est disponible. Ce n'est qu'à la fin de son service qu'un blocage peut survenir.

La figure ci-dessous illustre un exemple de réseau à capacité limitée :

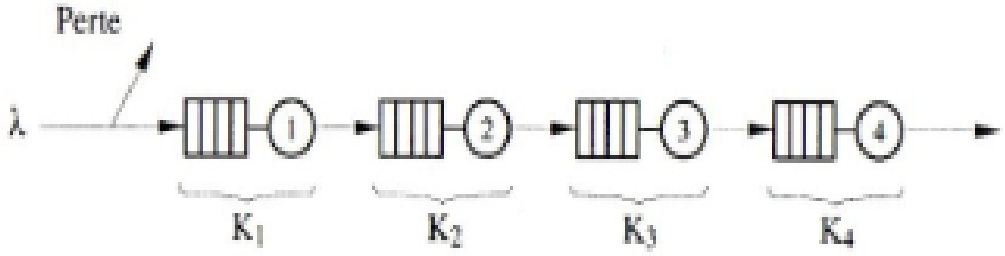


FIG. 2.5 – Réseaux à capacité limitée

## 2.3 Les réseaux de files d'attente à forme produit : Réseau Jackson

Un réseau de files d'attente est un ensemble de files d'attente inter-connectées. Nous allons présenter dans cette section à une classe particulière de réseaux de files d'attente, sous le nom de Réseaux à Forme Produit qui ont la particularité de posséder une solution analytique très simple.

### 2.3.1 Les réseaux mono-classe ouverts à forme produit

Dans un réseau ouvert, les clients arrivent dans le système depuis l'extérieur. Après avoir accompli un certain nombre d'opérations, ils quittent le système. De même que pour les files d'attente simples, ou la file $M/M/1$ est la plus simple à étudier, on s'intéressera dans un premier temps aux réseaux de files d'attente ouverts comportant :

- Une seule classe de clients ;
- Un processus d'arrivée des clients dans le système poissonien ;
- Un seul service à chaque station ;
- Un temps de service exponentiel à chaque station ;
- Une discipline de service FIFO pour toutes les files ;
- Des routages probabilistes : quand un client a plusieurs destinations possibles à la fin d'un service, il fait son choix en fonction d'un tirage aléatoire selon une certaine distribution de probabilité.

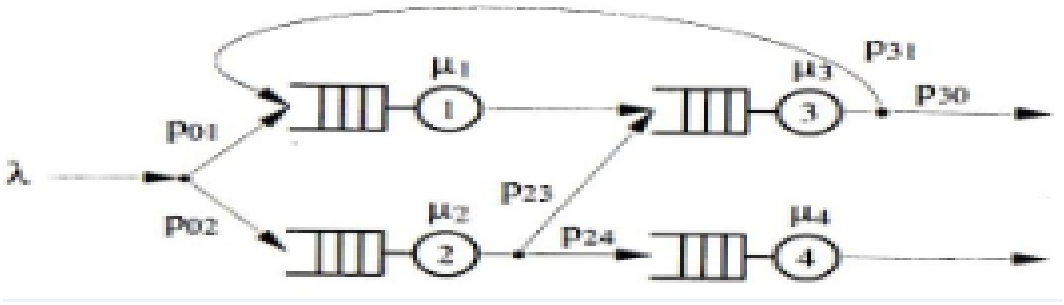


FIG. 2.6 – Réseau mono-classe ouvert

Ces réseaux sont connus sous le nom de réseaux de *Jackson ouverts*[15, 29]. On note $M$ le nombre de stations, $\lambda$ le taux d'arrivée des clients dans le réseau et $\mu_i$ le taux de service de la station $i$, $i = 1, \ldots, M$. Soit $p_{0i}$ la probabilité qu'un client qui arrive dans le système se rende à la station $i$, $p_{ij}$ la probabilité qu'un client qui termine son service à la station $i$ se rende à la station $j$ et $p_{i0}$ la probabilité qu'un client qui termine son service à la station $i$ quitte le système. On a :

$$\sum_{j=0}^{M} p_{ij} = 1; \quad i = 1, \ldots, M. \tag{2.1}$$

Cette équation est également vérifiée en $i = 0$, avec la convention $p_{00} = 0$. On a vu pour la file $M/M/1$ que, pour que le système reste stable, il faut que $\lambda < \mu$.

Dans le cas des réseaux de files d'attente, la condition de stabilité est logiquement liée, non seulement au taux d'arrivée des clients dans le réseau et aux taux de service $\mu_i$ des différentes stations, mais également au cheminement des clients.
Notons $e_i$ le taux de visite de la station $i$ ou le nombre moyen de passages à la station $i$. Pour $i = 1, \ldots, M$, en posant $\lambda_i$=$e_i\lambda$ le taux d'arrivée des clients à la station $i$, on a :

**La condition de stabilité du système :** $\lambda_i < \mu_i$ **;** $\quad i = 1, ..., M.$

**Calcul des taux de visite [3]**

Supposons que le réseau est stable et donc que pour chaque station $\lambda_i < \mu_i$. $\lambda_i$ Mesure le trafic à la station $i$. C'est donc à la fois le débit moyen d'entrée et le débit moyen de sortie de la station $i$. Ce trafic se décompose en plusieurs parties :

- le trafic venant de l'extérieur : $\lambda p_{0i}$ ;
- le trafic venant de la station j : $\lambda_j p_{ji}$ pour toutes les stations $j = 1, \ldots, M$.

On a donc :

$$\lambda_i = \lambda p_{0i} + \sum_{j=1}^{M} \lambda p_{ij}.$$

Comme $\lambda_i = e_i\lambda$, on en déduit le système d'équations que doivent satisfaire les taux de visite :

$$e_i = p_{0i} + \sum_{j=1}^{M} e_j p_{ji} \ , \quad i = 1, ..., M. \tag{2.2}$$

**Théorème 2.1.** *La probabilité stationnaire du réseau possède la forme produit suivante [9] :*

$$p(n) = \prod_{i=1}^{M} p_i(n_i),$$

*où* $p_i(n_i)$ *est la probabilité stationnaire d'une file* $M/M/1$ *ayant un taux d'arrivé* $\lambda_i$ *et un taux de service* $\mu_i$*, soit* $p_i(n_i) = (1-\rho_i)\rho_i^{n_i}$ *avec* $\rho_i = \frac{\lambda_i}{\mu_i}$.

***Remarque* 2.3.1.** Un réseau de *Jackson* est donc équivalent à un ensemble de files $M/M/1$ et la probabilité stationnaire $p(n)$ du réseau est égale au produit des probabilités marginales $p_i(n_i)$ de chacune des files étudiées en isolation.

### Calcul des paramètres de performances

Les paramètres de performances, débit moyen, nombre moyen de clients, temps moyen de réponse, doivent pouvoir être calculés par file ou pour l'ensemble du réseau :

| | Station i | Réseau |
|---|---|---|
| Débit moyen | $X_i$ | X |
| Nombre moyen de clients | $Q_i$ | Q |
| Temps myen de réponse | $R_i$ | R |

TAB. 2.1 – Paramètres de Perfermances

Les paramètres de performances de chaque station se déduisent de la décomposition en files $M/M/1$ :

$$X_i = \lambda_i = e_i\lambda, \tag{2.3}$$

$$Q_i = \frac{\rho_i}{1-\rho_i}, \text{ avec } \rho_i = \frac{\lambda_i}{\mu_i}, \tag{2.4}$$

$$R_i = \frac{Q_i}{X_i} = \frac{1}{\mu_i - \lambda_i}. \tag{2.5}$$

Les paramètres de performances du réseau s'en déduisent alors immédiatement :

$$X = \lambda, \tag{2.6}$$

$$Q = \sum_{i=1}^{M} Q_i, \tag{2.7}$$

$$R = \frac{Q}{X} = \frac{Q}{\lambda}. \tag{2.8}$$

***Remarque* 2.3.2.** En cas de stations multiserveurs on conserve les hypothèses précédentes, excepté que chaque station peut comporter plusieurs serveurs identiques (et indépendants les uns des autres). Soit $m_i$ le nombre de serveurs de la station i. Chacun de ces serveurs est exponentiel et de la même taux $\mu_i$. Les taux de visite $e_i$ et les taux moyens d'arrivée $\lambda_i$ se calculent alors de la même façon que dans le cas mono-serveur. La condition de stabilité du réseau est alors la suivante :

$$\lambda_i < m_i \mu_i$$

Elle exprime que le taux d'arrivée à chaque station doit être inférieur à la capacité de service maximale de la station. Cette dernière est obtenue en supposant que les $S_i$ serveurs travaillent en permanence et débitent donc globalement à taux $S_i\mu_i$. Sous cette condition, le réseau est équivalent à un ensemble de files $M/M/m$, et la probabilité stationnaire $p(n)$ du réseau est égale au produit des probabilités marginales $p_i(n_i)$ de chacune des files étudiées en isolation.

### 2.3.2 Les réseaux mono-classes fermés à forme produit

Dans un réseau fermé, les clients initialement dans le système $y$ circulent sans jamais en sortir et sans qu'aucun client de l'extérieur n'y entre. Ils sont donc en nombre constant. Comme dans le cas ouvert, on s'intéressera tout d'abord aux réseaux de files d'attente fermés comportant :

- une seule classe de clients ;
- un seul serveur à chaque station ;
- un temps de service exponentiel à chaque station ;

- une capacité de stockage illimitée à toutes les stations (ou au moins égale à $N$) ;
- des files $FIFO$ ;
- des routages probabilistes.

La figure ci-dessous illustre un exemple de réseau mono-classe fermé

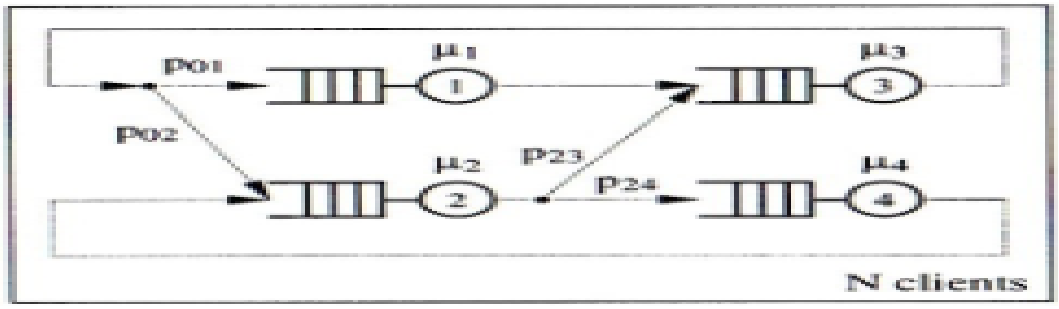


FIG. 2.7 – Réseau mono-classe fermé

Ces réseaux sont connus sous le nom de réseaux de *Jackson fermés*[16, 13]. On note $M$ le nombre de stations. $N$ le nombre total de clients, $\mu_i$ le taux de service de la station $i$, $i = 1, \dots M$ et $p_{ij}$ la probabilité qu'un client qui termine son service à la station $i$ se rende à la station $j$. Les probabilité $p_{ij}$ sont telles que :

$$\sum_{j=1}^{M} p_{ij} = 1, \quad i = 1, ..., M \tag{2.9}$$

Dans un réseau fermé, il n'y a bien entendu aucun problème de stabilité puisque le nombre de clients à chaque station est limité à la population du réseau et ne peut donc croitre à l'infini : pour toute station $i$, $n_i(t) < N$, ($n_i(t)$ est le nombre de clients présents à la station $i$) à tout instant $t$. La contrainte de population du réseau impose de plus que la condition $\sum_{i=1}^{M} n_i(t) = N$ est en permanence respectée.

**Théorème 2.2.** *La probabilité stationnaire du réseau possède la* $\{forme\ \ produit\}$ *suivante[9] :*

$$p(n) = \frac{1}{G(M,N)} \prod_{i=1}^{M} f_i(n_i),$$

*où* $f_i(n_i) = (\frac{e_i}{\mu_i})^{n_i}$. *et* $G(M,N)$ *est une constante de normalisation.*

**Calcul des taux de visite [3]**

Dans un réseau fermé, le nombre absolu de fois qu'un client passe par chaque station

est infini. On va donc s'intéresser ici au ($e_i$ ) taux de visite de la station $i$ ou nombre moyen de passage à la station $i$ entre deux passages par une station de référence (une station $j$ telle que, par convention, $e_j = 1$). De la même manière que dans le cas ouvert, on peut montrer que les $e_i$ sont solutions du système d'équations :

$$e_i = \sum_{j=1}^{M} e_j p_{ji} \ , \quad i = 1, \ldots, M. \tag{2.10}$$

Mais, contrairement au cas ouvert, comme les $e_i$ ne sont définis qu'à une constante près, ce système admet une infinité de solutions. Il suffit alors de choisir une station de référence.

### Calcul des paramètres de performances, algorithme de convolution

Il s'agit d'obtenir les paramètres de performances par station ou pour l'ensemble du réseau. Mais contrairement au cas ouvert, les paramètres de chaque station ne peuvent pas se déduire de l'analyse d'une file simple en isolation . Il faut donc manipuler l'expression des probabilités stationnaires. La première idée est de calculer les probabilités marginales de chaque station par sommation sur les probabilités stationnaires :

$$p_i(k) = \sum_{n \in E(M,N) | n_i = k} p(n) \ , \quad i = 1, \ldots, M \ \ et \ \ K = 0, \ldots, N. \tag{2.11}$$

Les paramètres de performances de chaque station s'en déduisent alors immédiatement :

$$U_i = 1 - p_i(0), \tag{2.12}$$

$$X_i = \sum_{k=1}^{N} p_i(k) \ \mu_i = (1 - p_i(0)) \, \mu_i, \tag{2.13}$$

$$Q_i = \sum_{k=1}^{N} k \ p_i(k), \tag{2.14}$$

$$R_i = \frac{Q_i}{X_i}. \tag{2.15}$$

Le problème est qu'un calcul des probabilités marginales par la relation (2.11) nécessite d'effectuer des sommations multiples très complexes. Heureusement, comme pour le calcul

de la constante de normalisation, on peut éviter ces sommations multiples. En remplaçant dans la relation (2.11) l'expression des probabilités stationnaires, on obtient en effet :

$$p_i(k) = \frac{1}{G(M,N)} \sum_{n \in E(M,N) | n_i = k} \prod_{j=1}^{M} \left(\frac{e_j}{\mu_j}\right)^{n_j}, \tag{2.16}$$

$$= \frac{1}{G(M,N)} \left(\frac{e_i}{\mu_i}\right)^{k} \sum_{n \in E(M,N) | n_i = k} \prod_{j=1,\ j \neq i}^{M} \left(\frac{e_j}{\mu_j}\right)^{n_j}. \tag{2.17}$$

On note alors $E_i(M,N)$ l'ensemble de tous les vecteurs $n$ de $E(M,N)$ qui sont tels que la somme des clients dans toutes les stations autres que la station $i$, $\sum_{j=1\ j\neq i}^{M} n_j = n$ ( et donc tels que le nombre $n_i$ de clients dans la station $i$, est égale à $N - n$) :

$$E_i(M,N) = \{n = (n_1 \dots n_M) \mid \sum_{j=1,\ j\neq i}^{M} n_j = n\}. \tag{2.18}$$

On note enfin $G_i(M-1,n)$, la constante de normalisation du réseau complémentaire, c' est-à-dire la constante de normalisation du réseau constitué des $M$ stations du réseaux initial privé de la station $i$, et dans lequel on place $n$ clients :

$$G_i(M-1,n) = \sum_{n \in E_i(M,N)} \prod_{j=1,\ j\neq i}^{M} \left(\frac{e_j}{\mu_j}\right)^{n_j}. \tag{2.19}$$

Les probabilités marginales $p_i(k)$ s'expriment alors simplement en fonction de ces constantes qu'il faut donc être capable de calculer :

$$p_i(k) = \left(\frac{e_i}{\mu_i}\right)^{k} \frac{G_i(M-1,N-k)}{G(M,N)}. \tag{2.20}$$

**Calcul des constants de normalisation du réseau complémentaire**

Dans un premier temps il est important de constater que la quantité $G(M-1,N)$ définie précédemment n'est rien d'autre que la constante du réseau complémentaire privée de la dernière station, $G_M(M-1,n)$. On peut alors écrire[3] :

$$G(M,N) = G_M(M-1,N) + \rho_M G(M,N-1). \tag{2.21}$$

Et comme il n'y a aucune raison de particulariser la station $M$, cette relation peut s'obtenir pour toute station $i$ de façon rigoureusement identique :

$$G(M,N) = G_i(M-1,N) + \rho_i G(M,N-1), \quad \text{où} \quad \rho_i = \frac{e_i}{\mu_i} \ , \quad i = 1,\ldots,M.$$

Cette relation nous permet, en l'inversant, d'obtenir par ( déconvolution) sur les constantes $G(M,N)$, les constantes de normalisations complémentaires :

$$G_i(M-1,n) = G(M,n) - \rho_i G(M,n-1) \ , \quad i = 1,\ldots,M. \tag{2.22}$$

**Calcul des paramètres de performances en fonction des constantes de normalisation**

On peut exprimer tous les paramètres de performances de la relation $i$ en fonction des constantes de normalisation qui, comme nous venons de le voir, sont extrêmement simples à calculer :

$$U_i = \frac{e_i}{\mu_i} \ \frac{G(M,N-1)}{G(M,N)} \ , \quad i = 1,\ldots,M. \tag{2.23}$$

$$X_i = e_i \ \frac{G(M,N-1)}{G(M,N)} \ , \quad i = 1,\ldots,M. \tag{2.24}$$

$$Q_i = \frac{1}{G(M,N)} \ \sum_{k=1}^{N} k \ \left(\frac{e_i}{\mu_i}\right)^k \ G_i(M-1,N-k) \ , \quad i = 1,\ldots,M. \tag{2.25}$$

$$R_i = \frac{Q_i}{X_i} = \frac{1}{e_i \ G(M,N-1)} \ \sum_{k=1}^{N} \left(\frac{e_i}{\mu_i}\right)^k \ G_i(M-1,N-k) \ , \quad i = 1,\ldots,M \tag{2.26}$$

Il est intéressant de noter à partir de la relation(2.24), que les débits des différentes stations sont contraints par la relation(2.27). Cette relation est connue sous le nom de loi des ***flots forcés*** :

$$\frac{X_i}{X_j} = \frac{e_i}{e_j}, \quad \text{pour tout } i \text{ et } j = 1,\ldots,M. \tag{2.27}$$

On aboutit finalement à l'algorithme de convolution suivant[14] :

**Algorithme de convolution**
Initialisation :
$G(1,n) = p_1^n \quad n =, 0 \ldots, N$
$G(m,0) = 1 \qquad m = 1, \ldots, M$
$G(M-1,0) = 1 \qquad i = 1, \ldots, M$
Pour m variant de 2 à M faire
pour n variant de 1 à N faire
$G(m,n) = G(m-1,n) + \rho_m G(m,n-1)$
Pour i variant de 1 à M faire
pour n variant de 1 à N faire
$G_i(M-1,n) = G(M,n) - \rho_i G(M,n-1)$
calculer les paramètres de performances moyens à l'aide des relations(2.23) à(2.26).

**Algorithme MVA**

Toute la difficulté du théorème de *Jackson* pour les réseaux fermés réside dans le calcul des constantes de normalisation et des constantes de normalisation complémentaires, nécessaires à l'obtention des probabilités marginales(2.20) et des paramètres de performances moyens(2.23) à(2.26). Si seuls les paramètres de performances moyens sont requis, il existe un algorithme récursif simple et performant, permettant d'éviter le calcul de ces constantes. Cet algorithme a été développé par Reiser [25, 26] est connu sous le nom : Algorithme de valeur moyenne (*MVA*). Le principe de l'algorithme *MVA* est d'exprimer les paramètres de performances moyens du réseau contenant $N$ clients en fonction des paramètres de performances du même réseau, mais contenant un client de moins, soit $N-1$ clients.
Tout d'abord réécrivons les relations(2.20), (2.23) à (2.26), en y faisant clairement apparaître la population du réseau :

$$p_i(k,N) = \left(\frac{e_i}{\mu_i}\right)^k \frac{G_i(M-1,N-k)}{G(M,N)}, \tag{2.28}$$

$$X_i(N) = e_i \frac{G(M,N-1)}{G(M,N)}, \tag{2.29}$$

$$Q_i(N) = \sum_{k=1}^{N} k \left(\frac{e_i}{\mu_i}\right)^k \frac{G_i(M-1,N-k)}{G(M,N)}, \tag{2.30}$$

$$R_i(k,N) = \frac{1}{\mu_i} \sum_{k=1}^{N} k \left(\frac{e_i}{\mu_i}\right)^{k-1} \frac{G_i(M-1,N-k)}{G(M,N-1)}. \tag{2.31}$$

L'algorithme repose sur la relation récursive suivante, exprimant le temps moyen de séjour d'un client à la station $i$ dans le réseau contenant $N$ clients, en fonction du nombre moyen de clients de la station $i$ dans le réseau contenant $N-1$ clients [14] :

$$R_i(N) = \frac{1}{\mu_i}\left(1 + Q_i(N-1)\right). \tag{2.32}$$

**Algorithme MVA**
Initialisation :
$Q_i((0\ldots0)) = 0 \qquad i = 1, \ldots, M.$
Pour n variant de 1 à N faire
$R_i(n) = \frac{1}{\mu_i}(1 + Q_i(n-1)) \quad i = 1, \ldots, M.$
$X(n) = \frac{n}{\sum_{i=1}^{M} e_i R_i(n)}$
$X_i(n) = e_i X(n) \quad i = 1, \ldots, M.$
$Q_i(n) = R_i(n) X_i(n) \quad i = 1, \ldots, M.$

***Remarque* 2.3.3.** Comme dans le cas ouvert, on peut étendre le théorème de Jackson (férmé) au cas de stations multi-serveurs (chaque station $i$ comporte $S_i$ serveurs identiques). Toutes les autres hypothèses sont conservées.

## 2.4 Les réseaux multi-classes à forme produit : les réseaux BCMP

### 2.4.1 Définition

On considère un réseau de files d'attente parcouru par différentes classes de clients [3]. On suppose dans un premier temps que les clients ne changent pas de classe lors de leur cheminement dans le réseau. Ce réseau posséde les caractéristiques suivantes :

- Un seul serveur à chaque station ;
- Une capacité de stockage illimitée à toutes les stations ;
- Des routages probabilistes pour chaque classe de clients.

On note $M$ le nombre de stations du réseau et $C$ le nombre de classes qui le parcourent. Les clients d'une classe donnée ne pouvant changer de classe, chaque classe est donc soit une classe ouverte, soit une classe fermée. On note $O$ l'ensemble des classes ouvertes du

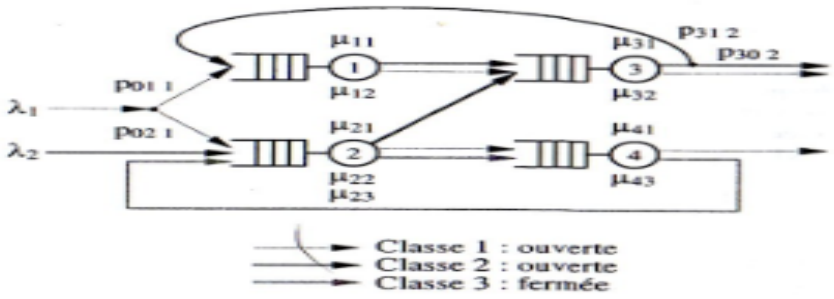


FIG. 2.8 – Réseau multi-classes à forme produit

réseau et $F$ l'ensemble des classes fermées : $O \cap F = \varnothing$ et $O \cup F = \{1 \dots C\}$.
Les clients d'une classe ouverte arrivent dans le système, accomplissent un certain nombre d'opérations, puis quittent le systéme. Les clients d'une classe fermée sont, quant à eux, en nombre constant, et ne peuvent ni arriver de l'extérieur, ni quitter le système.

Pour une classe ouverte donnée $c \in O$, on impose, de plus, un processus Poissonien d'arrivée des clients (de la classe $c$) dans le système de taux $\lambda_c$.

On note alors $p_{0i\ c}$ la probabilité qu'un client de classe $c$ qui arrive dans le système se rende à la station $i$, $p_{ij\ c}$ la probabilité qu'un client de classe $c$ qui termine son service à la station $i$ se rende à la station $j$ et $p_{i0\ c}$ la probabilité qu'un client de classe $c$ qui termine son service à la station $i$ quitte le systéme. Ces probabilités vérifient la relation :

$$\sum_{j=0}^{M} p_{ij\ c} = 1 \ , \quad i = 0, \dots, M. \tag{2.33}$$

avec la convention $p_{00} = 0$.

Pour une classe fermée donnée $c \in F$, soit $N_c$ le nombre de clients de classe $c$.
On note comme précédemment $p_{ij\ c}$ la probabilité qu'un client de classe $c$ qui termine son service à la station $i$ se rende à la station $j$. Ces probabilités vérifent la relation :

$$\sum_{j=1}^{M} p_{ij\ c} = 1 \ , \quad i = 1, \dots, M. \tag{2.34}$$

Finalement, chaque station peut être de quatre types différentes :*FIFO, PS, IS* et *LCFS-PR* (voir le tableau suivant) :

| Type | Discipline de service | Lois de service |
|---|---|---|
| 1 | FIFO (premmier arrivé, premier servi | Exponentielles indépendantes de la classe du client en service :$\mu_i$ taux de service |
| 2 | PS | Générales différentes pour chaque classe (à la transformée de laplace rationnelle) : $\frac{1}{\mu_{ic}}$ Le temps service moyen des clients de classe c. |
| 3 | IS (nombre de serveurs infinis) | |
| 4 | LCFS-PR dernier arrivé, (premier servi, avec préemption du service) | |

TAB. 2.2 – Types différents de chaque station

Ces réseaux sont connus sous le nom de réseau BCMP sans changement de classe (et à taux indépendant de l'état)[2].

### 2.4.2 Stabilité

Comme dans le cas des réseaux mono-classes, on définit tout d'abord la quantité ,
$e_{ic}$ : taux de visite de clients de classe $c$ à la station $i$ ou nombre moyen de passages d'un client de classe $c$ à la station $i$.

Ces quantités sont définies classe par classe et, pour chacune des classe, ont la même interprétation que dans le cas mono-classe.

Pour une classe $c$ ouverte, $e_{ic}$ s'interprète comme le nombre moyen de fois qu'un client de classe $c$ visite la station $i$ au cours de son séjour dans le système. Pour une classe $c$ fermée, $e_{ic}$ s'interprète comme le nombre relatif de fois qu'un client de classe $c$ passe à une station $i$ entre deux passages par une station de référence (une station $j$ telle que $e_{jc} = 1$).

La stabilité dans un réseau de files d'attente multi-classes est moins intuitive que celle d'un réseau mono-classe dès l'instant où le réseau comporte des classes ouvertes et des

classes fermées. On va donc dans un premier temps à des réseaux multi-classes purement ouverts, purement fermées ou mixtes.

**Réseaux purement ouverts**

Dans un réseau purement ouvert ($O = \{1, \ldots, C\}$), toutes les classes de clients sont des classes ouvertes. Tout d'abord, on connait le taux moyen d'arrivée des clients de classe $c$ à une station $i$ donnée :

$$\lambda_{ic} = \lambda_c e_{ic}. \tag{2.35}$$

Dans cette expression, le taux de visite $e_{ic}$ est bien entendu nul si la classe $c$ ne visite pas la station $i$. Donc :

$$\lambda_i = \sum_{c=1}^{C} \lambda_{ic}. \tag{2.36}$$

On en déduit $q_{ic}$, la probabilité pour qu'un client qui arrive à la station $i$ soit de classe $c$ :

$$q_{ic} = \frac{\lambda_{ic}}{\lambda_i}. \tag{2.37}$$

Un client de classe $c$ induira à chaque passages à la station $i$ une charge de travail moyenne $\frac{1}{\mu_{ic}}$. Un client d'une classe quelconque induira donc à chaque passage à la station $i$ une charge de travail moyenne $\sum_{c=1}^{C} \frac{q_{ic}}{\mu_{ic}}$. On définit alors le taux moyen de service de la station $i$ comme l'inverse de cette quantité :

$$\mu_i = \frac{1}{\sum_{c=1}^{C} \frac{q_{ic}}{\mu_{ic}}} = \frac{\lambda_i}{\sum_{c=1}^{C} \frac{\lambda_{ic}}{\mu_{ic}}}. \tag{2.38}$$

En effet, pour une station $FIFO$, la loi de service doit être exponentielle et indépendante de la classe du client en service. Donc $\mu_i = \mu_{ic}$ pour tout $c$.

La condidtion de stabilité exprime alors, comme dans le cas mono-classe, que le taux moyen d'arrivée des clients à la station $i$ (quelles que soient leurs classes) doit être inférieur au taux moyen de service : $\lambda_i < \mu_i$

A partir de la relation(2.38), on en déduit immédiatement que le réseau est stable si pour toute station $i$ :

$$\sum_{c=1}^{C} \frac{\lambda_{ic}}{\mu_{ic}} < 1, \ i = 1, \ldots, M.$$

.

**Réseaux purement fermés**

Toutes les classes de clients sont des classes fermées ($F = \{1, \ldots, C\}$). Comme dans le cas mono-classe, il n'y a bien sûr aucun problème de stabilité puisque le nombre de clients de chaque classe à chaque station est limité à $N_c$, le nombre total de clients de classe $c$ , dans le réseau.

**Réseaux mixtes**

Pour un réseau mixte, ce sont bien entendu les classes ouvertes qui vont pouvoir poser un problème de stabilité. Celles-ci se partagent, en général, les même stations que les classes fermées. Cependant, lorsque l'on atteint les limites de la stabilité du systéme, le nombre de clients présents à la station la plus limitative (en termes de stabilité), tend vers l'infini.

Ce sont bien entendu les clients des classes ouvertes qui s'accumulent à cette station. On comprend alors que la charge de travail induite par les clients des classes fermées, à cette même station, tend vers une quantité négligeable. La condition de stabilité du systéme est donc naturellement liée aux seule classe ouvertes.

Elle s'exprime de la même façon que dans le cas d'un réseau purement ouvert, les taux $\lambda_i$ et $\mu_i$ s'obtenant à partir des relations (2.36) et(2.38) en n'effectuant les sommations que sur l'ensemble des classes ouvertes ($c \in O$). Le réseau est donc stable si :

$$\sum_{c \in O} \frac{\lambda_{ic}}{\mu_{ic}} < 1, \text{ pour tout } i = 1, \ldots, M.$$

**Théorème 2.3.** *La probabilité stationnaire du réseau possède la "forme produit" suivante [3] :*

$$p(n) = \frac{\Lambda(n)}{G} \prod_{i=1}^{M} f_i(n_i) \tag{2.39}$$

$$\text{où : } f_i(n_i) = \begin{cases} n_i! \prod_{c=1}^{C} \frac{1}{n_{ic}!} \left(\frac{e_{ic}}{\mu_i}\right)^{n_{ic}}, & \text{si la station } i \text{ est de type 1 ;} \\ n_i! \prod_{c=1}^{C} \frac{1}{n_{ic}!} \left(\frac{e_{ic}}{\mu_{ic}}\right)^{n_{ic}}, & \text{si la station } i \text{ est de type 2 ou 4 ;} \\ \prod_{c=1}^{C} \frac{1}{n_{ic}!} \left(\frac{e_{ic}}{\mu_{ic}}\right)^{n_{ic}}, & \text{si la station } i \text{ est de type 3.} \end{cases}$$

*avec $n_i$ nombre total de clients à la station $i$ :* $n_i = \sum_{c=1}^{C} n_{ic}$

$$\Lambda(n) = \prod_{c \in O} (\lambda_c)^{K_c} = \begin{cases} \prod_{c=1}^{C} (\lambda_c)^{k_c}, & \text{si le réseau est purement ouvert ;} \\ 1, & \text{si le réseau est purement fermé.} \end{cases}$$

*$K_c$ nombre total de clients de classe c :* $K_c = \sum_{i=1}^{M} n_{ic}$
*et G est une constante de normalisation.*

### 2.4.3 Calcul des taux de visite

Les taux de visites se calculent classe par classe. Pour une classe ouverte $c \in O$, ils sont l'unique solution du système :

$$e_{ic} = p_{0i\ c} + \sum_{j=1}^{M} e_{jc} p_{ji\ c}\ , \quad i = 1, \ldots, M. \tag{2.40}$$

Pour une classe fermée $c \in F$, ils sont solutions du système :

$$e_{ic} = \sum_{j=1}^{M} e_{jc} p_{ji\ c}\ , \quad i = 1, \ldots, M. \tag{2.41}$$

Mais, à nouveau, ce système admet une infinité de solutions et les $e_{ic}$ ne sont définis qu'à une constante près. Il suffit alors de choisir une station de référence, une station $j$ visitée par les clients de classe $c$ et de poser $e_{jc} = 1$. Les autres taux de visite se déduisent alors sans ambiguïté .

### 2.4.4 Réseau purement ouvert

Si toutes les classes de clients sont des classes ouvertes ($O = \{1, \ldots, C\}$) , il existe une première simplification du théorème *BCMP* applicable dès l'instant où l'on s'intéresse uniquement au nombre total de clients présents aux différentes stations du réseau.

Les probabolités stationnaires ainsi que les paramètres moyens de perfermances seront calculés toutes classes confondues. On parlera de probabilités marginales (vis-à-vis des classes) et des paramètres de perfermances marginaux.
Notons :

- $n_{ic}(t)$ : Le nombre de clients de classe $c$ présents à la station $i$ à tout instant $t$ ;
- $n_i(t)$ : Le nombre total de clients présents à la station $i$ à tout instant $t$ ;
- $(n_1(t) \ldots n_M(t))$ : Le vecteur d'état marginal du réseau à tout instant $t$.

**Proposition 2.1.** *La probabilité marginale stationnaire du réseau possède la* $\ll$ *forme produit* $\gg$ *suivante :*

$$p(n_1, \ldots, n_M) = \prod_{i=1}^{M} p_i(n_i),$$

*où* $p_i(n_i) = \begin{cases} (1-\rho_i)\rho_i^{n_i}, & \textit{si la station } i \textit{ est de type 1, 2 ou 4 ;} \\ \frac{e^{-\rho_i}}{n_i!}\rho_i^{n_i}, & \textit{si la station } i \textit{ de type 3.} \end{cases}$

*avec* $n_i$ *nombre total de clients à la station* $i$ *:* $n_i = \sum_{c=1}^{C} n_{ic}$

$\rho_i = \frac{\lambda_i}{\mu_i} = \begin{cases} \sum_{c=1}^{C} \frac{\lambda_c e_{ic}}{\mu_i}, & \textit{si la station est de type 1 ;} \\ \sum_{c=1}^{C} \frac{\lambda_c e_{ic}}{\mu_{ic}}, & \textit{si la station } i \textit{ est de type 2,3 ou 4.} \end{cases}$

$\lambda_i$ *et* $\mu_i$ *donnés par les relations (2.36) et (2.38)*

Dans le cas d'un réseau contenant uniquement des stations de type $1, 2$ ou $4$, le comportement stationnaire marginal du réseau est donc strictement équivalent à celui d'un réseau de *Jackson* (mono-classe ouvert) dans lequel chaque station $i$ possède un taux de service $\mu_i$ donné par la relation(2.36), le taux d'arrivée des clients (toutes classes confondues) à la station $i$ étant donné par la relation(2.38).

Notons que la condition de stabilité du réseau multi-classes énoncée précédemment, $\sum_{c=1}^{C} \frac{\lambda_{ic}}{\mu_{ic}} < 1$, est cohérente avec la condition de stabilité du réseau de *Jackson* associé, $\rho_i < 1$. Cette équivalence était évidente dans le cas d'un réseau contenant uniquement des stations de type 1.

Dans ce cas, en effet, le service de chaque station est exponentiel et indépendant de la classe du client en service (et de taux $\mu_i$). Seul le routage distingue alors les différentes classes de clients dans le réseau. Dés l'instant où l'on ne s'intéresse plus aux classes de clients, il suffit de calculer le taux moyen d'arrivée à chaque station à l'aide de la relation(2.36), et de considérer ce réseau comme un réseau mono-classe.
On peut s'intéresser aux probabilités stationnaires et énoncer la proposition suivante. Celle-ci constitue la deuxième simplification du théorème $BCMP$ pour des réseaux purement ouverts et peut être considérée comme la généralisation du théorème de *Jackson* ouvert mono-classe.

**Proposition 2.2.** *La probabilité stationnaire du réseau possède* $\ll$ *la* $forme\ produit$ $\gg$ *suivante :*

$$p(n_1, \ldots, n_M) = \prod_{i=1}^{M} p_i(n_i)$$

$$\text{où } p_i(n_i) = \begin{cases} (1-\rho_i) n_i! \prod_{c=1}^{C} \frac{1}{n_{ic}!} \left(\frac{\lambda_c e_{ic}}{\mu_i}\right)^{n_{ic}}, & \textit{si la station } i \textit{ est de type 1 ;} \\ (1-\rho_i) n_i! \prod_{c=1}^{C} \frac{1}{n_{ic}!} \left(\frac{\lambda_c e_{ic}}{\mu_i}\right)^{n_{ic}}, & \textit{si la station } i \textit{ est de type 2 ou 4 ;} \\ e^{-\rho_i} \prod_{c=1}^{C} \frac{1}{n_{ic}!} \left(\frac{\lambda_c e_{ic}}{\mu_i}\right)^{n_{ic}}, & \textit{si la station } i \textit{ est de type 3.} \end{cases}$$

On peut finalement proposer une troisième simplification du théorème $BCMP$ en ne s'intéressant plus qu'à une classe particulière de clients.

**Proposition 2.3.** *Soit $p_{ic}(n_{ic})$ la probabilité marginale pour que la station $i$ contienne $n_{ic}$ clients d'une classe $c$ donnée, quel que soit le nombre de clients des autres classes présents à la station (et quel que soit l'état des autres stations du réseau). Ces probabilités s'expriment très facilement à l'aide de la relation suivante :*

$$p_{ic}(n_{ic}) = \begin{cases} (1-\rho'_{ic})\rho_{ic}^{n_{ic}}, & \textit{si la station } i \textit{ est de type 1,2,ou 4 ;} \\ \frac{e^{-\rho'_{ic}}}{n_{ic}!}\rho_{ic}'^{n_{ic}}, & \textit{si la station } i \textit{ est de type 3 .} \end{cases}$$

$$\text{où} \quad \rho'_{ic} = \frac{\lambda_c e_{ic}}{\mu'_{ic}}$$

$$\mu'_{ic} = \begin{cases} \mu_{ic}(1 - \sum_{s=1}^{C} \sum_{s \neq c} \frac{\lambda_s e_{is}}{\mu_{is}}), & \textit{si la station } i \textit{ est de type 2 ou 4 ;} \\ \mu_i - \sum_{s=1}^{C} \sum_{s \neq c} \lambda_s e_{is}, & \textit{si la station } i \textit{ est de type 1 ;} \\ \mu_{ic}, & \textit{si la station } i \textit{ est de type 3.} \end{cases}$$

Pour une station $i$ de type 1, 2 *ou* 4, $p_{ic}(n_{ic})$ possède donc l'expression des probabilités stationnaires d'une file $M/M/1$ mono-classe ayant un taux de service $\mu'_{ic}$ et soumise à un processus d'arrivée poissonien de taux $\lambda_{ic} = e_{ic}\lambda_c$. Ses paramétres de performances s'en déduisent alor immédiatement :

$$X_{ic} = \lambda_{ic} = e_{ic}\lambda_c, \tag{2.42}$$

$$R_{ic} = \frac{1}{\mu_{ic},} \tag{2.43}$$

$$Q_{ir} = R_{ic}X_{ic} = \rho'_{ic}. \tag{2.44}$$

### 2.4.5 Réseau purement fermé

Toutse les classes de clients sont maintenant des clases fermées ($F = \{1, \ldots, C\}$). Réécrivons, dans ce cas, l'expression de la constante de normalisation en $y$ faisant apparaître explicitement la dépendance avec la nombre $M$ de stations et le vecteur $N = (N_1, \ldots, N_C)$ de population du réseau :

$$G(M, N) = \sum_{n \in E(M,N)} \prod_{i=1}^{M} f_i(n_i), \tag{2.45}$$

Comme dans le cas mono-classe, on peut établir une relation de récurence permettant d'éviter un calcul direct de la constante de normalisation :

$$G(M, N) = \sum_{k=0,\ldots,0}^{N} f_M(k)G(M-1, N-k) = \sum_{k_1=0}^{N_1} \ldots \sum_{k_C=0}^{N_C} f_M(k)G(M-1, N-k). \tag{2.46}$$

Cette relation de récurrence permet, en théorie, de calculer toutes les constantes de normalisation $G(m, n)$ pour $m = 1, \ldots, M$ et $n = (0, \ldots, 0), \ldots, N$ en partant des conditions initiales :

$$G(1, n) = f_1(n) \quad n = (0, \ldots, 0), \ldots, N.$$

$$G(m, (0, \ldots, 0)) = 1 \quad m = 1, \ldots, M.$$

Initialisation :
$G(1,n) = f_1(n)$ $\quad n = (0,\ldots,0),\ldots,N$
$G(m,(0,\ldots,0)) = 1$ $\quad m = 1,\ldots,M$
Pour $m$ variant de 2 à $M$ faire
  Si la station $m$ et de type $1, 2$ ou $4$ Alors
    Pour $n$ variant de $(0,\ldots,0)$ à $N$ faire
      $$G(m,n) = G(m-1,n) + \sum_{c=1}^{C} \sum_{n_c \neq 0} \left(\frac{e_{mc}}{\mu_{mc}}\right) G(m, n-1_c)$$
  sinon (station de type 3)
    pour $n$ variant de $(0,\ldots,0)$ à $N$ faire
      $$G(m,n) = \sum_{k=(0,\ldots,0)}^{n} f_m(k) G(m-1, n-k)$$

TAB. 2.3 – Algorithme de convolution : calcul des constantes G(m, n)

**Calcul des paramètres de performances en fonction des constantes de normalisation**

Comme dans le cas mono-classe, on peut exprimer les paramètres de performances de chaque station en fonction des constantes de normalisation complémentaires. Les probabilités marginales de la station $i$, peuvent en effet s'exprimer de la façon suivante :

$$p_i(k) = \frac{1}{G(M,N)} \sum_{n \in E(M,N) | n_i = k} \prod_{j=1}^{M} f_j(n_j); \tag{2.47}$$

$$= \frac{1}{G(M,N)} f_i(k) \sum_{n \in E(M,N) | n_i = k} \prod_{j=1, j\neq i}^{M} f_j(n_j). \tag{2.48}$$

En définissant $E_i(M,k)$ l'ensemble de tous les vecteurs d'état $n$ de $E(M,N)$ qui sont tels que la somme des clients de classe $c$ dans toutes les stations autres que la stations $i$, $\sum_{j=1, j\neq i}^{M} n_{jc}$, est égale à $k_c$ pour toutes les classes $c = 1,\ldots,C$, (et donc tels que le nombre $n_{ic}$ de clients de classe $c$ dans la station $i$, est égal à $N_c - k_c$) :

$$E_i(M,k) = \{n = (n_1,\ldots,n_M) | \sum_{j=1}^{M} \sum_{j\neq i} n_{jc} = k_c, c = 1,\ldots,C\}$$

et $G_i(M-1,k)$, la constante de normalisation du réseau complémentaire, comme étant la constante de normalisation du réseau constitué des $M$ stations du réseau initial, privé de la station $i$ et dans lequel on place $k = (k_1,\ldots,k_C)$ clients :

$$G_i(M-1,k) = \sum_{n \in E(M,k)} \prod_{j=1, j\neq i}^{M} f_j(n_j).$$

On obtient immédiatement :

$$p_i(k) = f_i(k)\frac{G_i(M-1, N-k)}{G(M,N)}. \tag{2.49}$$

Afin de calculer les constantes de normalisation complémentaires on a :

$$G_i(M-1,n) = G(M,n) - \sum_{c=1 n_c \neq 0}^{C} \left(\frac{e_{ic}}{\mu_{ic}}\right) G(M, n-1_c). \tag{2.50}$$

Si la station $i$ est de types 3, une manipulation de la relation (2.46), nous donne de la même manière :

$$G_i(M-1,n) = G(M,n) - \sum_{k=(0,\ldots,0), k\neq(0,\ldots,0)}^{n} f_i(k) G_i(M-1, n-k) \tag{2.51}$$

Les paramètres de performance de chaque station se déduisent alors immédiatement des constantes de normalisation du réseau :

$$X_{ic} = e_{ic}\frac{G(M, N-1_c)}{G(M,N)}, \tag{2.52}$$

$$Q_{ic} = \sum_{k=(0,\ldots,0)}^{N} k_c \; p_i(k) = \frac{1}{G(M,N)} \sum_{k=(0,\ldots,0)}^{N} k_c f_i(k) G_i(M-1, N-k), \tag{2.53}$$

$$R_{ic} = \frac{Q_{ic}}{X_{ic}}. \tag{2.54}$$

**Algorithme de convolution : calcul des performances moyennes :**

Calcul des constantes $G(m,n)$, $m = 1,\ldots,M$ et n=(0, ..., 0), ..., N
Initialisation :
$G_i(M-1,(0,\ldots,0)) = 1 \qquad i = 1,\ldots,M$
Pour $i$ variant de 1 à $M$ faire
  Si la station $m$ et de type $1,2$ ou 4 Alors
    Pour $n$ variant de $(0,\ldots,0)$ à $N$ faire

$$G_i(M-1,n) = G(M-,n) - \sum_{c=1}^{C} \sum_{n_c,\neq 0} \left(\frac{e_{ic}}{\mu_{ic}}\right) G(M,n-1_c)$$

  sinon (station de type 3)
    pour $n$ variant de $(0,\ldots,0)$ à $N$ faire

$$G_i(M-1,n) = G(M,n) - \sum_{k=(0,\ldots,0),k\neq(0,\ldots,0)}^{n} f_i(k)G_i(M-1,n-k)$$

Calculer les paramètres de performance moyens à l'aide (2.52) à (2.54)

### 2.4.6 Algorithme $MVA$ pour les réseaux purement fermés

L'algorithme $MVA$ présenté en détail dans le cadre des réseaux mono-classe fermés peut être généralisé au cas d'un réseau multi-classe purement fermé. Le principe de base reste le même puisque l'algorithme consiste à exprimer les paramètres de performances du réseau ayant une population $N = (N_1,\ldots,N_C)$, en fonction de ceux du même réseau , mais contenant un client de classe $c$ en moins (et donc ayant une population $N - 1_c = (N_1,\ldots,N_{c-1},N_{c-1},N_{c+1},\ldots,N_C)$).
Les relations donnant les paramètres de performances doivent être réécrites, en y faisant clairement apparaître la population du réseau :

$$p_i(k,N) = f_i(k)\ \frac{G_i(M-1,N-k)}{G(M,N)}, \tag{2.55}$$

$$X_{ic}(N) = e_{ic}\frac{G(M,N-1_c)}{G(M,N)}, \tag{2.56}$$

$$Q_{ic}(N) = \sum_{k=(0,0,\ldots,0)}^{N} k_c p_i(k,N), \tag{2.57}$$

$$R_{ic}(N) = \frac{Q_{ic}(N)}{X_{ic}(N)} \tag{2.58}$$

La relation la plus importante de l'algorithme $MVA$ multi-classes est celle exprimant le temps moyen de séjour d'un client de classe $c$ à la station $i$ ( de type 1, 2 , *ou* 4 ), dans le réseau ayant une population $N$, en fonction du nombre moyen de clients présents à la station $i$, toutes classes confondues, dans le réseau ayant une population $N - 1_c$ :

$$R_{ic}(N) = \frac{1}{\mu_{ic}}(1 + Q_i(N - 1_c)). \tag{2.59}$$

Si la station $i$ est de type 3, le temps moyen de séjour d'un client de classe $c$ à cette station se réduit bien évidemment à son temps moyen de service :

$$R_{ic}(N) = \frac{1}{\mu_{ic}}. \tag{2.60}$$

**Algorithme MVA**
Initialisation :
$Q_i((0\ldots0)) = 0 \qquad i = 1, \ldots, M.$
Pour n variant de $(0\ldots0)$ à N $(n \neq (0, \ldots, 0))$ faire
Pour toutes les stations $i = 1 \ldots M$ de type 1 , 2 *ou* 4
$R_{ic}(n) = \frac{1}{\mu_{ic}}(1 + Q_i(n - 1_c)) \qquad c = 1, \ldots, C$
Pour toutes les stations $i = 1 \ldots M$ de type 3
$R_{ic}(n) = \frac{1}{\mu_{ic}} \qquad c = 1, \ldots, C$
$X_c(n) = \frac{n_c}{\sum_{i=1}^{M} e_{ic} R_{ic}(n)} \quad c = 1, \ldots, C.$
$X_{ic}(n) = e_{ic}\ X_c(n) \quad i = 1, \ldots, M,\ et\ c = 1, \ldots, C.$
$Q_{ic}(n) = R_{ic}(n)\ X_{ic}(n) \quad i = 1, \ldots, M,\ et\ c = 1, \ldots, C.$
$Q_i(n) = \sum_{c=1}^{C} Q_{ic}(n) \quad i = 1, \ldots, M.$

## 2.5 Conclusion

Chaque file d'attente est caractérisée par son processus d'arrivée, taux de service et la discipline de la file. Dans certains systèmes, un client accomplit son travail en passant par plusieurs serveurs d'où la notion des réseaux de files d'attente. Les réseaux de files d'attente ont une très grande importance car ils servent à modéliser des systèmes physiques ; ils permettent d'évaluer les performances et ils aident à mieux comprendre le comportement de ces systèmes.

Chapitre 3

# Application

## 3.1 Introduction

Les réseaux de files d'attente constituent un formalisme de modélisation largement utilisé pour l'évaluation des performances des systèmes à événements discrets tels que : les systèmes informatiques, les réseaux de communications et les systèmes de production,... Dans ce chapitre, nous proposons une application en domaine informatique dans le but d'analyser le trafic dans un serveur central en évaluant ses paramètres de performance. Pour analyser correctement le problème, nous proposons un modèle de rèseau Jackson ouvert et fermé. Tout d'abord, nous décrivons le modèle analytique, ensuite nous allons calculer ces paramètres de performance.

## 3.2 Application 01 : Modèle Serveur Central

Le système considéré est une unité centrale qui exécute des processus [1] et un ensemble d'unités d'entrée-sortie (disque A, disque B). Ces différente composante constituent les ressources du système [20].

On modélise ce système par un réseau de file d'attente ouvert dans la figure (3.1). Les clients du modèle sont les processus du système. Les différentes ressources sont modélisées par des stations (une pour le CPU, une pour chaque unité d'entrée- sortie). Les processus sont générés par les utilisateurs extérieurs, avec une fréquence moyenne $\lambda$.

Les clients présents dans la file d'attente se dirigent vers la station CPU, qui est le serveur qui distribue les tâches (jobs) aux autres appareils (ici disque A et B). Après service,

[1] Les clients de ce système sont appelés processus

les clients rejoignent la station disque A avec une probablité $p_A$, la station disque B avec une probablité $p_B$ ou quittent le système lorsque le service est terminé d'une interruption entrée-sortie(E/S).

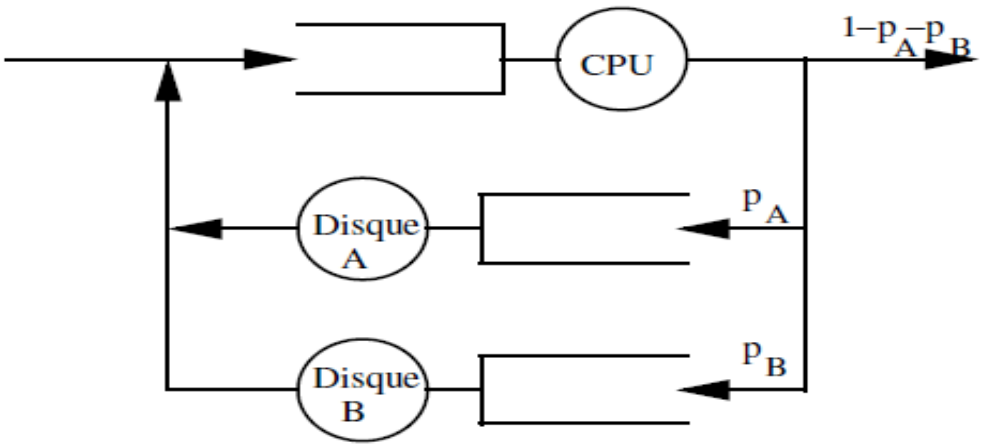


FIG. 3.1 – Modèle à serveur central

Afin de spécifier ce modèle, il faut caractériser :

- Une seule classe de clients ;
- le processus d'arrivée des clients dans le système (un processus de poisson de taux $\lambda = 3$ tâches par second) ;
- Les distributions de service de chacune des stations CPU et les deux disque sont exponentielles de temps de service moyens respectivement : $\frac{1}{\mu_{CPU}} = 0.005s$ , $\frac{1}{\mu_A} = 0.02s$, $\frac{1}{\mu_B} = 0.03s$ ;
- Une discipline de service FIFO pour toutes les files ;
- une capacité de stockage illimitée à toutes les stations ;
- Après avoir reçu son service dans la CPU, une tâche a une probabilité $P_A = 0.4$ d'être envoyée vers le disque A et $P_B = 0.5$ vers le disque B.

On veut analyser certaines performances de cet ordinateur, en déduire quels sont les appareils à améliorer, et notamment vérifier si un appareil en particulier est goulot d'étranglement (bottleneck) du système.

**Calcul des paramètres de performance :**

**Le 1*ér* cas :**

Pour calculer ces paramètres, on utilise les formules suivantes (déjà définies au chapitre 2) :

**le taux de visite :**

$$e_i = p_{0i} + \sum_{j=1}^{M} e_j p_{ji}, \quad i = 1, ..., M, \tag{3.1}$$

d'où

$$e_1 = 10;\ e_2 = 4;\ e_3 = 5.$$

**Le débit moyen :**

$$X_i = \lambda_i = e_i \lambda, \tag{3.2}$$

d'où

$$X_1 = 30;\ X_2 = 12;\ X_3 = 15.$$

**Le nombre moyen de clients :**

$$Q_i = \frac{\rho_i}{1 - \rho_i} \quad avec \quad \rho_i = \frac{\lambda_i}{\mu_i}, \tag{3.3}$$

d'où

$$Q_1 = 0.18 \text{ tâches ; } Q_2 = 0.32 \text{ tâches ; } Q_3 = 0.82 \text{ tâches .}$$

**Le temps moyen de réponse :**

$$R_i = \frac{Q_i}{X_i} = \frac{1}{\mu_i - \lambda_i}, \tag{3.4}$$

d'où

$$R_1 = 0.006\ s;\ R_2 = 0.027\ s;\ R_3 = 0.055\ s.$$

Les performances de chaque station sont résumées dans le tableaux suivant :

| | Station CPU | Station disque A | Station disque B |
|---|---|---|---|
| Les taux d'arrivées internes | 30 tâches/s | 12 tâches/s | 15 tâches/s |
| Débit moyen | 30s | 12s | 15s |
| Nombre moyen de clients | 0.18 tâches | 0.32 tâches | 0.82 tâches |
| Temps moyen de réponses | 0.006s | 0.027s | 0.055s |

TAB. 3.1 – Les paramètres de performances de chaque station avec deux disques

Les performances du réseau sont présentées dans le tableau ci-dessous :

| | |
|---|---|
| $\lambda$ | 3 tâches/s |
| Q | 1.32 tâches |
| R | 0.44 s |

***Remarque* 3.2.1.** Il est clair que l'appareil formant le goulot d'étranglement du système est le disque $B$. Si on le change par un disque deux fois plus rapide, $R_B = 0.019s$ et le temps de réponse moyens du système devient $R = 0.261s$, ce qui représente une amélioration d'environ 40% du temps de réponse. Si on avait divisé par deux le temps de service du disque $A$ au lieu de disque $B$, cette amélioration n'aurait été que de 14%, ce qui montre l'intérêt de localiser le goulot d'étranglement du système.
On peut se poser d'autres questions, comme par exemple de savoir si l'utilisation d'un seul disque, mettons le disque $A$, vers lequel toutes les opérations d'entrées-sorties sont dirigées, dégrade fortement les performances (la taille mémoire étant supposée suffisante).

**Le 2*éme* cas :** On utilise un seul disque (disque $A$) avec une probabilté de $p_A = 0.9$. On a refait tous les calculs en utilisant toujours les formules citées précédemment, on a obtenu les resultats suivants :

| | Station CPU | Station disque A |
|---|---|---|
| Les taux d'arrivées internes | 30 tâches/s | 27 tâches/s |
| Débit moyen | 30 s | 27 s |
| Nombre moyen de clients | 0.18 tâches | 1.17 tâches |
| Temps moyen de réponses | 0.006 s | 0.04 s |

TAB. 3.2 – Les paramètres de performances de chaque station avec un seul disque

Et pour le réseau on a :

| | |
|---|---|
| $\lambda$ | 3 tâches/s |
| Q | 1.35 tâches |
| R | 0.45 s |

D'après les résultats obtenu, les temps moyens de réponse pour le systéme avec un seul disque où avec deux disque sont presque égaux. Il est préférable donc d'opter pour un système avec un seul disque (cette solution est moins coûteuse).

## 3.3 Application 02 : Modèle à serveur central d'un système à temps partagé

Le modèle à serveur central de l'ordinateur étudié à l'application 01 est à présent inséré dans un réseau fermé comportant $N$ terminaux, comme représenté dans la figur ( 3.2). L'ordinateur fonctionne en mode interactif ; le nombre maximum de tâches qu'il doit traiter est $N$. le groupe des $N$ terminaux sont modélisés comme une file $M/M/S/N$, avec un temps de service moyen qui est le temps moyen de réflexion des utilisateurs $1/\mu_T = 1sec$ [20].

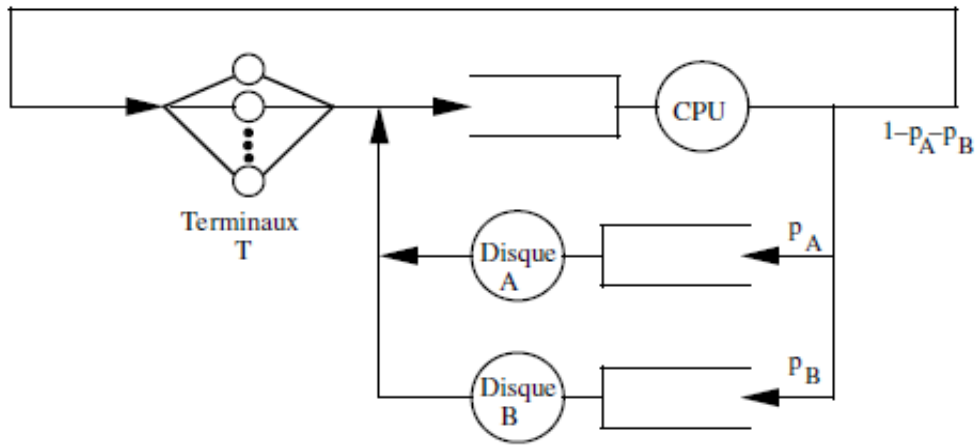


FIG. 3.2 – Modèle à serveur central d'un système à temps partagé

Les autres données restent inchangées par rapport au module précédent.
On a

$e_T = 1$ (référence),
$e_{CPU} = 10$;
$e_A = 4$;
$e_B = 5$.

On peut maintenant mettre en œuvre l'algorithme $MVA$ pour déterminer les performances moyennes de ce système en fonctions du nombre d'utilisateurs. On prendra comme temps de réponse moyen du système le temps moyen que passe une tâche dans l'ordinateur *(CPU+ disques A et B)*, entre son départ et son retour au terminal.
On applique l'algorithme $MVA$ en remplaçant le flux d'arrivées externes $\lambda_S$ par le flux d'arrivées provenant des terminaux, $\lambda_T$, qui n'est plus une donnée du problème.

**L'algorithme est initialisé aux valeurs** $Q_i(0) = 0$

| | |
|---|---|
| $R_T(1)$ | $\frac{1}{\mu_T} = 1\ s$ |
| $R_{CPU}(1)$ | $\frac{(1+E[Q_{CPU}(0)])}{\mu_{CPU}} = 0.005\ s$ |
| $R_A(1)$ | $\frac{(1+E[Q_A(0)])}{\mu_A} = 0.02\ s$ |
| $R_B(1)$ | $\frac{(1+E[Q_B(0)])}{\mu_B} = 0.03\ s$ |
| $\lambda_T(1)$ | $e_T \times \frac{1}{(e_T R_T + e_{CPU} R_{CPU} + e_A R_A + e_B R_B)} = 0.781$ tâche$/s$ |
| $\lambda_{CPU}(1)$ | $e_{CPU}\lambda_T(1) = 7.813$ tâche$/s$ |
| $\lambda_A(1)$ | $e_A\lambda_T(1) = 3.125$ tâche$/s$ |
| $\lambda_B(1)$ | $e_B\lambda_T(1) = 3.906$ tâche$/s$ |
| $Q_T(1)$ | $\lambda_T(1)R_T(1) = 0.781$ tâche |
| $Q_{CPU}(1)$ | $\lambda_{CPU}(1)R_{CPU}(1) = 0.039$ tâche |
| $Q_A(1)$ | $\lambda_A(1)R_T(A) = 0.063$ tâche |
| $Q_B(1)$ | $\lambda_B(1)R_T(B) = 0.117$ tâche |
| $R(1)$ | $\frac{(\lambda_{CPU}R_{CPU} + \lambda_A R_A + \lambda_B R_B)}{\lambda_T} = 0.28\ s$ |

On peut alors commencer la seconde itération, et ainsi de suite. On trouve les résultats suivants :

| N | 2 | 3 | 4 | 5 | 10 | 20 | 30 | 50 | 100 |
|---|---|---|---|---|---|---|---|---|---|
| $Q_T(N)$ | 1.533 | 2.251 | 2.928 | 3.560 | 5.821 | 6.661 | 6.667 | 6.667 | 6.667 |
| $Q_{CPU}(N)$ | 0.080 | 0.122 | 0.164 | 0.207 | 0.398 | 0.499 | 0.500 | 0.500 | 0.500 |
| $Q_A(N)$ | 0.130 | 0.204 | 0.282 | 0.365 | 0.800 | 1.135 | 1.143 | 1.143 | 1.143 |
| $Q_B(N)$ | 0.257 | 0.424 | 0.626 | 0.868 | 2.981 | 11.705 | 21.690 | 41.690 | 91.690 |
| $R(N)$ | 0.305 | 0.333 | 0.366 | 0.405 | 0.716 | 2.003 | 3.500 | 6.500 | 14.000 |

**Conclusion.** *D'après l'algorithme MVA, on remarque que lorsque N devient supérieur à*

20*, le nombre de tâches présentes en moyenne dans les terminaux, le CPU et le disque A a atteint une valeur constante quel que soit N, au contraire du disque B, qui est donc confirmé comme goulot d'étranglement du système. On constate alors une augmentation du temps moyen de réponse.*

## 3.4 Conclusion

Dans ce chapitre, nous avons modélisé un système informatique par un réseau de Jackson, dans le cas ouvert et fermé. A cet effet, plusieures variantes d'un modéle de serveur central ont été considérées. Nous avons calculé les mesures de performance de chaque variantes dans le but de choisir la meilleure solution.

# Conclusion générale

La théorie des files d'attente est une technique qui permet de modéliser un système admettant un phénomène d'attente, de calculer ses performances et déterminer ses caractéristiques pour aider les gestionnaires dans leurs prises de décisions. Aussi les files d'attente peuvent être considérées comme un phénomène caractéristique de la vie contemporaine.

Elles peuvent être enchainées pour former les réseaux de files d'attente où les départs d'une file d'attente entrent dans la prochaine file et cela dans le cas où un client a besoin de plusieurs services. Les réseaux de files d'attente peuvent être classifiées en : réseaux de files d'attente ouverts, fermés et mixtes.

Dans ce mémoire, nous nous sommes intéressés particulièrement aux réseaux de files d'attentes à forme produit Jackson et BCMP.

Dans un premier temps, nous avons montré l'intérêt et les applications des systèmes d'attente (classiques). Une attention particulière a été accordée aux modèles d'attente markoviens.

Dans un deuxième temps, nous avons étudié les réseaux de files d'attente à forme produit, où nous avons calculé les paramètres de performance par la file ou pour l'ensemble du réseau. Dans le cas d'un réseau multi-classes, ces paramètres doivent être calculés pour chaque classe de clients. Par la suite, nous avons examiné les mesures de performance des réseaux de Jackson et BCMP.

En guise d'application, nous avons modélisé un système informatique par un réseau

de Jackson, dans le cas ouvert et fermé. A cet effet, plusieures variantes d'un modéle de serveur central ont été considérées. Les mesures de performance de chaque variantes ont été calculées dans le but de choisir la meilleure configuration.

Pour aller loin dans le développement de ces modèles de réseaux ; plusieurs pistes restent à explorer :

La principale piste est de poursuivre les recherches sur une extension qui permetrait d'accroitre les possibilités d'application des modèles BCMP multi-classes : il s'agit de prendre en compte la capacité finie des secteurs internes dans le modèle analytique.

Il serait intéressant aussi d'examiner les performances d'un système en appliquant d'autres modèles de réseaux de files d'attente tels que les réseaux de files d'attente avec priorité pour tenir compte des particularités des clients.

# Annexe

Dans cette annexe, nous présentons quelques notions et lois de probabilités auxquelles nous avons fait appel dans ce mémoire.

## Processus stochastiques

Un processus stochastique $X(t)_{t\in T}$ est une fonction du temps dont la valeur à chaque instant dépend de l'issue d'une expérience aléatoire. A chaque instant $t \in T$, $X(t)$ est donc une variable aléatoire.
Un processus stochastique peut donc être considéré comme une famille de variable aléatoires (généralement non indépendantes). L'ensemble des temps $T$ peut être discret ou continu. $X(t)$ définit l'état du processus à un instant donné $t$. A nouveau, l'ensemble E des valeurs que peut prendre le processus à chaque instant est appelé espace d'état et peut, de même que $T$, être discret (finit ou infini) ou continu [27].

### Processus de comptage :

Soit $N(t)$ le nombre d'événements se produisant dans un intervalle de temps $[0, t]$. On cherche à déterminer la loi de probabilité de cette variable aléatoire.
$\{N(t), t \geq 0\}$ est appelé processus de comptage.
$N(t+s) - N(s)$ : nombre d'événements aléatoire se produisant dans l'intervalle $[s, s+t]$.

### Processus de Poisson

Le processus de poisson est le plus utilisé dans la théories des files d'attente. Il modélisera géneralement le processus d'arriveé des clients dans un système [10].

**Définition 3.1.** [10]On dit qu'un processus de comptage $\{N(t), \ t \geq 0\}$ est un processus de Poisson s'il satisfait aux trois conditions suivantes :

**Condition 3.1.** $\{N(t), \ t \geq 0\}$ *est homogène dans le temps, donc les probabilités de transitions sont constante*

$$P[N(t+s) - N(s) = k] = P[N(t) = k] \ \ pour \ \ tout \ \ t > 0, \ \ s > 0, \ \ k = 0$$

**Condition 3.2.** $\{N(t), \ t \geq 0\}$ *est à accroissement indépendants ce qui signifie que pour tout système d'intervalles disjoints, le nombre d'événement s'y produisant sont des variables aléatoires indépendantes*

$$\begin{aligned} P[N(t+s) - N(s) = k, N(s) = j] &= P[N(t) = k, N(s) = j] \\ &= P[N(t) = k] \ \ P[N(s) = j] \\ &= P_k(t) \ \ P_j(s) \quad \forall s > 0, \quad \forall t > 0 \end{aligned}$$

**Condition 3.3.** *La probabilité que deux évènement an plus se produisent dans un petit intervalle $\Delta t$ est négligeable par rapport a la probabilité qu'il n'y ait qu'un seul événement :*

$$P_k(\Delta t) = \begin{cases} o(\Delta t), & si \quad k \geq 0 \\ \lambda(\Delta t) + o(\Delta t), & si \quad k = 1 \\ 1 - \lambda(\Delta t) + o(\Delta t), & si \quad k = 0 \end{cases}$$

*$\lambda$ est appelé : intensité du processus de Poisson.*

*Le processus ainsi décrit pourrait-être schématisé par le graphe suivant :*

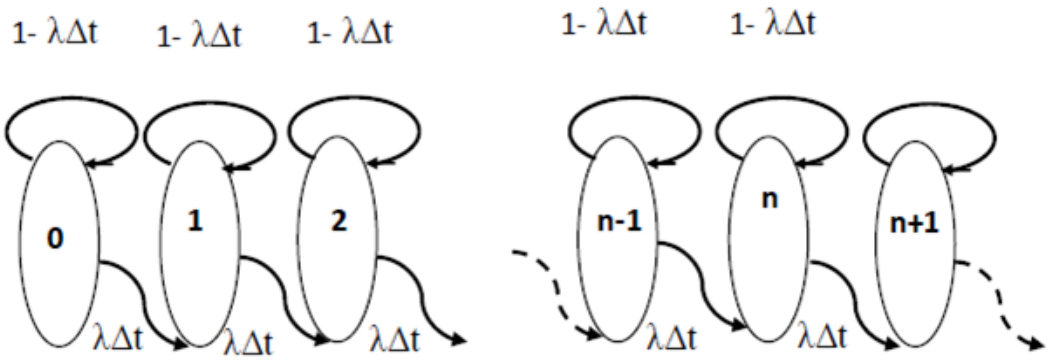


FIG. 3.3 – Processus de Poisson

# Chaînes de Markov

Un processus de Markov est un processus dans lequel le comportement futur ne dépend que du passé récent. Les suites markoviennes sont appelées "chaînes" de Markov. On distingue les chaînes de Markov à espace d'états discret et celles à espace d'états continu [19].

## Propriété de Markov

Un processus stochastique $\{X_t, t \geq 0\}$ définie sur un espace d'états S satisfait la propriété de Markov si, pour tout instant $t \geq 0$ et tous sous ensemble d'états $I \subseteq S$, il est alors vrai que :

$$P[X_{t+\Delta t} \in I | X_u, 0 \leq \mu \leq t] = P[X_{t+\Delta t} \in I | X_t], \Delta t \geq 0$$

Un processus stochastique vérifiant la propriété précédente est appelé "processus de Markov" ou "processus Markovien".

## Matrice stochastique

Considérons une classe importante de matrice carrées qui jouent un rôle décisif lors de l'étude des processus stochastiques à temps discret.
Une matrice $P$ est dite stochastique si :
• Tous les termes sont positifs ou nuls,

• La somme des termes de chaque ligne vaut 1.
Les lignes de P représentent donc des vecteurs de probabilité.

# Processus de Naissance et de Mort

Les processus de *Naissance et de Mort* interviennent dans la modélisation des systèmes de files d'attente, et permettent de façon générale de décrire l'évolution temporelle d'une population d'un type donné [9].

**Exemple 3.1.** *Dans un système de file d'attente, on considère les populations comprenant tous les chants dans le système à l'instant t.*
*Les processus de Naissance et de Mort sont des processus stochastiques à temps discret* $S = \{0, 1, \ldots\}$*, ils sont markoviens (sans mémoire)*
*A partir d'un état quelconque* $'n'$ *donné les transitions ne sont possibles que vers l'un ou l'autre des états voisins on parle alors de naissance et de mort.*

**Définition 3.2.** Soit $\{x(t), t \geq\}$ un processus stochastique d'espace d'états $s = \{0, 1, \ldots\}$ et homogène dans le temps.

$$P[X(t+s) = j | X(s) = i] = P_{ij}(t)$$

$\{x(t), t \geq\}$ est dit processus de naissance et de mort si les condition suivantes sont vérifies :

- $P_{i\ i+1}(\Delta t) = P[X(t+\Delta t) = i+1 | X(t) = i] = \lambda_i \Delta t + o(\Delta t)$,
- $P_{i\ i-1}(\Delta t) = \mu_i \Delta t + o(\Delta t)$,
- $P_{i\ i}(\Delta t) = 1 - (\lambda_i + \mu_i)\Delta t + o(\Delta t)$,
- $P_{i\ j}(\Delta t) = o(\Delta t)$, avec $\lambda_i > 0$ et $\mu_i > 0$

$\lambda_i$ : taux de naissance.

$\mu_i$ : taux de mort.

## Régime transitoire et Régime stationnaire

On s'intéresse au calcul des probabilités d'état : $P_n(t) = P[X(t) = n]$ D'après la formule des probabilités totales (F.P.T) [9] :

$$\begin{aligned} P_n(t+\Delta t) &= P[X(t+\Delta t) = n] \\ &= \sum_{i=0}^{\infty} P_i(t) P_{i\ n}(\Delta t) \\ &= P_{n-1}(t) P_{n-1\ n}(\Delta t) + P_n(t) P_{n\ n}(\Delta t) + P_{n+1}(t) P_{n+1\ n}(\Delta t) + o(\Delta t) \\ &= P_{n-1}(t)\lambda_{n-1}(\Delta t) + P_n(t)(1 - (\lambda_n - \mu_n)\Delta t) + P_{n+1}(t)\mu_{n+1}(\Delta t) + o(\Delta t) \\ &= P_{n-1}(t)\lambda_{n-1}(\Delta t) + P_n(t) - P_n(t)(\lambda_n - \mu_n)(\Delta t) + P_{n+1}(t)\mu_{n+1}\Delta t + o(\Delta t) \end{aligned}$$

$$\begin{aligned} P_n(t+\Delta t) - P_n(t) &= P_{n-1}(t)\lambda_{n-1}(\Delta t) - P_n(t)(\lambda_n - \mu_n)(\Delta t) + P_{n+1}(t)\mu_{n+1}\Delta t + o(\Delta t) \\ \frac{P_n(t+\Delta t) - P_n(t)}{\Delta t} &= P_{n-1}(t)\lambda_{n-1} - P_n(t)(\lambda_n - \mu_n) + P_{n+1}(t)\mu_{n+1} + \frac{o(\Delta t)}{\Delta t} \end{aligned}$$

Lorsque $\Delta t \longrightarrow 0$ alors :

$$P_n'(t) = \lambda_{n-1}P_{n-1}(t) - (\lambda_n - \mu_n)P_n(t) + \mu_{n+1}P_{n+1}(t), \forall n \geq 1.$$

De la même maiere, on obtient :

$$P_0'(t) = -\lambda_0 P_0(t) + \mu_1 P_1(t).$$

D'ou le système d'équation de *Chapmain-kolmogorov*

$$\begin{cases} P_0'(t) = -\lambda_0 P_0(t) + \mu_1 P_1(t). \\ P_n'(t) = \lambda_{n-1}P_{n-1}(t) - (\lambda_n + \mu_n)P_n(t) + \mu_{n+1}P_{n+1}(t), \forall n \geq 1. \end{cases}$$

Si de plus, on connait les conditions initiales ie : $q_i = P[X(0) = i]$ on déterminera ainsi, le régime transitoire du processus : $P_n(t) = \sum_{i=0}^{\infty} q_i P_{in}(t)$.
On s'intéresse au régime stationnaire :
ie :

$$P_n = \lim_{t\to\infty} P_n(t) = \lim_{t\to\infty} P[X(t) = n] = P[X = n].$$

Dans le cas des équations de *Chapmain-Kolmogorov*

$$\lim_{t\to\infty} P_n'(t) = 0, \forall n = 0, 1, \ldots$$

On obtient un système d'équation linéaire

$$\begin{cases} \mu_1 P_1 = \lambda_0 P_0 \\ \lambda_{n-1} P_{n-1} - (\lambda_n + \mu_n) P_n + \mu_{n+1} P_{n+1} = 0, \forall n \geq 1. \end{cases}$$

Pour résoudre ce système d'équation on additionne les $(n+1)$ $1^{ère}$ équation on trouve :

$$\lambda_n P_n = \mu_{n+1} P_{n+1}, \forall n \geq 0.$$

En admettant que $\lambda_0 > 0$

$$P_n = \frac{\lambda_0 \lambda_1 \lambda_2 \dots \lambda_{n-1}}{\mu_1 \mu_2 \dots \mu_n} P_0,$$

ou bien

$$P_n = \prod_{i=1}^{n} \frac{\lambda_{i-1}}{\mu_i} P_0.$$

Comme : $\sum_{n=0}^{\infty} P_n = 1$

donc

$$P_0 + \sum_{n=0}^{\infty} \prod_{i=1}^{n} \frac{\lambda_{i-1}}{\mu_i} P_0 = 1,$$

alors

$$P_0 = \frac{1}{1 + \sum_{n=0}^{\infty} \prod_{i=1}^{n} \frac{\lambda_{i-1}}{\mu_i}}.$$

Pour que le régime existe il faut que la somme ci-dessus converge.

***Remarque* 3.4.1.** Si l'espace des états est fini le régime stationnaire existe toujours.

# La loi exponentielle

Soit $X$ une variable aléatoire distribuée suivant une loi exponentielle de paramètre $\mu$ sa densité de probabilité est définie par [3] :

$$f(x) = \begin{cases} \mu \exp -(\mu x), & si \quad \text{x} \geq 0 \\ 0, & sinon \end{cases}$$

**Proposition 3.1.** *La fonction de repartition de cette loi est :*

$$F(x) = \begin{cases} 1 - \exp -(\mu x), & si \quad x \geq 0 \\ 0, & sinon \end{cases}$$

**Proposition 3.2.** *[3]la moyenne et la variance d'une loi exponentielle sont :*

$$E(x) = \frac{1}{\mu},$$

$$var(x) = \frac{1}{\mu^2}.$$

**Proposition 3.3.** *[3]Propriété "sans mémoire"*
*Une variable aléatoire $T$ est dite sans mémoire lorsque :*

$$P[T \leq t + t_0 | T > t_0] = P[T \leq t]$$

**Preuve.**

$$\begin{aligned} P[T \leq t + t_0 | T > t_0] &= \frac{P[t_0 < T \leq t + t_0]}{P[T > t_0]} \\ &= \frac{F_T(t + t_0) - F_T(t_0)}{1 - F_T(t_0)} \\ &= \frac{\exp^{-\lambda t_0}(1 - \exp^{-\lambda t})}{\exp^{-\lambda t_0}} \\ &= 1 - \exp^{-\lambda t} \\ &= P[T \leq t] \end{aligned}$$

*La propriété sans mémoire est la propriété essentielle de la loi exponentielle qui fait que cette dernière est trés largement utilisée.*

**Proposition 3.4.** *[3]La loi exponentielle est la seule variable aléatoire continue à posséder la propriété sans mémoire.*

## Loi de Poisson :

une variable aléatoire $X$ suit la loi de poisson de paramètre $\lambda$ si pour tout $k = 1, 2, \ldots$ sa densité de probabylité est de la forme [3] :

$$f(X) = P(X = n) = \frac{\lambda^n \exp(-\lambda)}{n!} \quad pour \quad n = 0, 1, 2, \ldots$$

Cette loi est aussi appelée loi des événements rares.

**Proposition 3.5.** *[3]La moyenne est la variance d'une loi de poisson de paramètre $\lambda$ sont :*

$$E[X] = V[X] = \lambda$$

**Proposition 3.6.** *[3]La somme de deux lois de Poisson indépendantes $X$ et $Y$ de paramètres $\lambda_1$ et $\lambda_2$ est une loi de Poisson de paramètre $\lambda_1 + \lambda_2$.*

**Proposition 3.7.** *[3]Les temps des arrivées à l'instant $T$ sont poisoniennes, les inter-arrivées sont indépendants et obéissent à une loi exponentielle de paramétre $\mu$.*

# Bibliographie


[1] Bouraine, L. *Support de cours sur les files d'attente.* Université A. Mira de Bejaia, 2015.

[2] Baskett F. Chandy K.M, Muntz R.R. et Palacios-Gomez F. *Open, Closed and Mixed networks of queues with different classes of customers*,J. ACM, vol. 22, p.248-260, 1975.

[3] Baynat, B. *Théorie des files d'attente-des chaînes de Markove aux réseaux à forme produit.* Paris, Hermèse Science Publications, 2000.

[4] Belarbi. F and Bouchentouf. A. A. Condition de stabilité d'un réseau de files d'attente à deux stations et N classes de clients. *General Mathematics,* Vol 18 :85-108, 2010.

[5] Benouaret. Z. *Stabilité forte dans les modèles de risque.* Thèse de Doctora, Université A. Mira de Bejaia, 2012.

[6] Borovkov. A. A. *Asymptotic methodes in queueing theory.* Wiley, New York, 1984.

[7] Boualem. M. *Sur la propriété de decomposition stochastique dans un système d'attente avec rappels et vacances.* Thèse de Doctorat, Université A. Mira de Bejaia, 2009.

[8] Burke. P. J. The output of a queuning system. *Operations Research,* Vol. 4 :699-704, 1956.

[9] Chabriac. C. *Processus stochastiques et modélisation.* Université de Toulouse le Mirail, Master 2, Année 2012-2013.

[10] Chretienne. P. And Faure. R. *Processus stochastique, leurs graphes, leurs usages.* Gauthier villars, Paris, 1957.

[11] Claudi. H. *Eléments de la théorie de file d'attente.* Technical report, Université de Toulouse 2, 2008.

[12] Djouhra. D. *Modélisation et simulation du flux dans un réseaux pour la regulation du trafic.* Ingénierie des données et connaissances.

[13] Gordon. W. J et Newell. G. F. *Closed Queueing Networks with Exponential Servers,* Operations Research, vol. 15, p. 252-267, 1967.

[14] Harbaoui. A *Vers une modélisation et un dimensionnement automatiques des applications réparties.* teèse de Doctorat, Université de Grenoble, 1998.

[15] Jackson. J. R *Networks of Waiting Lines*, Operations Research, vol.5, p.518-521, 1957

[16] Jackson. J. R *Jobshop-like Queueing Systems,* Management Science, vol. 10, p. 131-142, 1963.

[17] Lagnoux. A *Processus stochastique et modélisation.* Université de Toulouse, 2005.

[18] Liebling. T. M., De Werra. D and Heche. J-F. *Recherche Opérationnelle pour ingénieurs.* Presses Polytechniques et Universitaires Romandes, Tome 2, 2003.

[19] Lionel. B *Processus stochastique : Processus de poisson et chaîne de Markov.* 2004.

[20] Patrick Thiran. *Files d'attente.*

[21] Nitto Personé V. De. S. Balsamo and R. Onvural. *Analysis of queueing networks with blocking.* Kluwer Academic Publishers, 2001.

[22] Pnilippe. N. *Basic elements of queueing theory : application to medeling of computer systems. Tech report,* The University of Massachusetts, 2004.

[23] Prontère. A. *Insensibilité et Bornes Stochastiques dans les réseaux de files d'attente.* Phd Thesis, Ecole Polytechnique, Novembre 2003.

[24] Pujolle. G and Fdida. S. *Modèle de système et de réseaux,* Eyrolles, 1989.

[25] Reiser. M. "Mean-Value Analysis of Queueing Networks, a New Look at an Old Problem", in Arato. M, Butrimenko. A et Gelenbe. E (eds),*Performance of Computer System : Porc. of the 4th Int. Symp. on Modelling and Performance Evaluation of Computer Systems,* North-Holland, Amsterdam, p. 63-67, 1979.

[26] Reiser. M et La venberg S. S.,*Mean Value Analysis of Closed Multichain Queueing Networks,* J.ACM, vol.27, p. 313-323, 1980.

[27] Rugg. R. *Processus stochastique.* Presses Polytechniques Romandes, 1989.

[28] Willing. A. *A short introduction to queueing thehory.* Technical University Berlin, Telecommunication Networks Group, 1999.

[29] Whittle. P. *equilibrium distributions for an open migration process*, J. Applied Probabilities, p.567-571, 1968.