



République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université Mouloud MAMMARI de Tizi-Ouzou

Faculté des sciences

Département de Mathématiques

Mémoire de Master

Filière : Mathématiques.

Spécialité : Recherche Opérationnelle.

Thème

**Modélisation et Optimisation du télétrafic dans
un centre d'appels téléphoniques**

Réalisé par :

Mme HASSANI Karima

Mme OUHACHI Dyhia

Encadré par :

Pr HAMADOUCHE Djamel

Année universitaire 2018/2019

Remerciements

À terme de notre modeste travail, nous tenons tout d'abord à remercier Dieu le tout puissant de nous avoir accordé la patience et le courage afin d'accomplir notre mémoire.

Un grand merci aux membres de nos familles pour leurs présences, leurs préoccupations, leurs soucis qu'ils se fassent pour nous, leurs encouragements et leurs suivis avec patience lors du déroulement de ce travail.

Nous tenons également à exprimer toute notre gratitude à notre promoteur Mr HAMADOUCHE Djamel. On le remercie de nous avoir encadré, orienté, aidé et conseillé.

Il nous semble tout aussi important de remercier nos enseignants qui ont permis d'élargir nos connaissances tout au long de notre cursus.

Nous voudrions exprimer nos reconnaissances envers nos camarades qui nous ont apporté leurs soutiens moral et intellectuel tout au long de notre démarche.

Enfin, merci à tous ceux qui ont contribué de près ou de loin à la réalisation de ce mémoire.

Table des matières

Remerciements	1
Introduction générale	5
1 Systèmes de gestion de masses	7
1.1 Processus markoviens	7
1.1.1 Introduction	7
1.1.2 Loi t-instantanée du système	8
1.1.3 Générateur infinitésimal d'un système markovien	9
1.1.4 Modélisation du système par les équations différentielles de Chapman-Kolmogorov	9
1.1.5 Solution stationnaire d'un processus de Markov	10
1.1.6 Caractérisation d'une loi stationnaire	10
1.1.7 Ergodicité d'un processus de Markov	10
1.2 Systèmes de files d'attente	12
1.2.1 Introduction	12
1.2.2 Description	12
1.2.3 Caractéristiques d'un système d'attente	12
1.2.4 Notation de Kendall	13
1.2.5 Etude du système M/M/1	13
1.2.5.1 Etude du processus $(N_t)_{t \geq 0}$	14
1.2.5.2 Loi du système en régime permanent (stable)	14
1.2.5.3 Performances de la file M/M/1	15
1.2.6 Etude de système M/M/1/K	19
1.2.6.1 Etude du processus $(N_t)_{t \geq 0}$	19
1.2.6.2 Loi du système en régime permanent (stable)	20
1.2.6.3 Performances de la file M/M/1/K	21
1.2.7 Etude du système (M/M/1/K/[F])	23

1.2.7.1	Etude du processus $(N_t)_{t \geq 0}$	23
1.2.7.2	Loi du système en régime permanent (stable)	24
1.2.7.3	Performances de la file M/M/1/K/[F]	25
1.2.8	Etude du système M/M/s ($s > 1$)	26
1.2.8.1	Etude du processus $(N_t)_{t \geq 0}$	27
1.2.8.2	Loi du système en régime permanent (stable)	28
1.2.8.3	Performances de la file M/M/s, liées aux clients	29
1.3	Conclusion	30
2	Modélisation du télétrafic d'un centre d'appels téléphoniques	31
2.1	Introduction	31
2.2	Description et étude du problème	31
2.3	Position du problème	32
2.4	Détermination des lois	32
2.4.1	Loi des arrivées	32
2.4.2	Loi de la durée de service	33
2.5	Etude de la stabilité du système	33
2.6	Etude du système M/M/s/K	33
2.6.1	Etude du processus $(N_t)_{t \geq 0}$	34
2.6.2	Loi du système en régime permanent (stable)	35
2.6.3	Performances de la file M/M/s/K	36
2.6.3.1	Nombre moyen d'appels en cours et en attente de connexion	37
2.6.3.2	Nombre moyen d'appels en attente de connexion	37
2.6.3.3	Durée moyenne d'un appel dans le système	37
2.6.3.4	Durée moyenne d'un appel avant connexion	38
2.6.4	Probabilité de saturation	38
2.7	Etude de la file M/M/s/s	38
2.7.1	Etude du processus $(N_t)_{t \geq 0}$	39
2.7.2	Loi du système en régime permanent (stable)	40
2.7.3	Performances du système (M/M/s/s)	41
2.7.3.1	Nombre moyen d'appels en cours et en attente de connexion	41
2.7.3.2	Durée moyenne d'un appel dans le système	41
2.7.4	Probabilité de saturation	41
2.8	Conclusion	42

3	Simulation des mesures de performances d'un centre d'appels téléphoniques	43
3.1	Introduction	43
3.2	Simulation du processus des arrivées	43
3.2.1	Arrivées poissonniennes	43
3.2.2	Graphe des temps des arrivées	44
3.3	Simulation des lois et des performances des files d'attente considérées en fonction des paramètres d'entrée	45
3.3.1	File d'attente $M/M/s/K$	45
3.3.1.1	Simulation de la loi de $N(t)$	45
3.3.1.2	Simulation des caractéristiques de la file	47
3.3.1.3	Risque de saturation	54
3.3.2	File d'attente $M/M/s/s$	57
3.3.2.1	Simulation de la loi de N_t	57
3.3.2.2	Simulation des caractéristiques de la file	59
3.3.2.3	Risque de saturation	62
3.4	Conclusion	65
4	Optimisation du télétrafic dans un centre d'appels téléphoniques	66
4.1	Introduction	66
4.2	File d'attente $M/M/s/K$	66
4.2.1	Position du problème	66
4.2.2	Optimisation du risque de saturation	67
4.2.3	Illustrations numériques et graphiques	69
4.3	File d'attente $M/M/s/s$	70
4.3.1	Position du problème	70
4.3.2	Optimisation du risque de saturation	70
4.3.3	Illustrations numériques et graphiques	73
4.4	Conclusion	74
	Conclusion générale	75
	Bibliographie	76

Introduction générale

Le monde contemporain est de plus en plus soucieux de satisfaire les attentes des clients, tellement il y a plusieurs choses dans la vie pour lesquelles il faut attendre. Un spécialiste de l'analyse d'attente a déterminé qu'une personne adulte passait au minimum le dixième de son temps à attendre. On attend les auto-bus, les ascenseurs, dans les magasins, les cinémas, les stations d'essence, chez le dentiste, etc. Toutefois, lorsqu'on parle d'attente, on pense souvent à des personnes, or, les clients en attente sont aussi des commandes en attente de traitement, des machines en attente de réparation, des programmes d'ordinateurs qui attendent d'être exécutés. Tout cela se passe en posant la fameuse question, pourquoi y a-t-il attente? Sur cela, introduisons la notion des files d'attente.

La théorie des files d'attente est l'un des outils analytiques les plus puissants pour la modélisation de systèmes de logistiques et de communications. Cette théorie a pour objet l'étude de systèmes et réseaux où des entités, appelées clients, cherchent à accéder à des ressources, généralement limitées, afin d'en obtenir un service. Ce domaine de recherche remonte aux années 1909-1920 avec les travaux de l'ingénieur Danois Erlang concernant les réseaux téléphoniques. La théorie mathématique s'est ensuite développée notamment grâce aux contributions de Palm, Kolmogorov, Khintchine, Pollaczek,... et fait actuellement toujours l'objet de nombreuses publications scientifiques. Le modèle de files d'attente classique consiste en un système dans lequel les serveurs traitent un flux de requêtes en suivant des stratégies bien précises. De nombreuses applications illustrent ce modèle, en particulier dans les domaines des télécommunications, du transport routier et des réseaux informatiques.

Prenons par exemple le cas d'un centre d'appels téléphoniques qui veut mettre en œuvre un service d'assistance technique pour sa clientèle. Bien évidemment, ce centre cherche à acquérir et garder une bonne réputation, en résolvant les problèmes rencontrés par ses clients. Remarquons qu'il s'agit d'un phénomène stochastique. En effet, les appels arrivent d'une manière aléatoire, suivant par exemple un processus de poisson, chaque nouveau appel passe directement au serveur, s'il est libre, sinon il s'oriente vers la file dans le cas d'occupation de tout les serveurs. De même, la durée de chaque appel est aussi aléatoire par exemple de loi exponentielle, obtenant ainsi un système d'attente markovien qui est un modèle de base, qu'on doit analyser en fonction du temps.

Pour cela, nous allons développer des modèles mathématiques nécessaires à l'étude et l'analyse des performances pertinentes aux phénomènes d'attente, ensuite on passera à la simulation avec le logiciel de calcul scientifique MATLAB afin de structurer et décrire cette modélisation et ces performances.

On va s'intéresser aux appels rentrants dans un centre d'appels qui engendrent parfois la saturation du système, c'est-à-dire lors d'un nouvel appel, celui-ci peut être mis en attente ou rejeté. Ainsi notre problème est d'optimiser les performances par rapport aux paramètres du système pour éviter la congestion et assurer un meilleur service.

L'étude consiste à minimiser le risque de saturation dans le centre d'appels, tout en déterminant la capacité du système, le nombre de serveurs nécessaires, afin de garantir le bon fonctionnement du système.

Ce mémoire est composé d'une introduction générale, de quatre chapitres et d'une conclusion générale.

Le premier chapitre donne une revue générale sur les modèles markoviens et les systèmes d'attente markoviens qui nous permettront de modéliser et d'étudier le problème considéré.

Le deuxième chapitre traite de la modélisation et l'évaluation des performances des deux systèmes de files d'attente.

Le troisième chapitre présente la simulation des caractéristiques de chaque système en fonction des différents paramètres d'entrée.

Au dernier chapitre, on s'intéresse au problème d'optimisation qui repose sur la minimisation du risque de saturation en fonction des différents paramètres des systèmes d'attente étudiés.

On termine par une conclusion qui résume notre travail.

Chapitre 1

Systèmes de gestion de masses

1.1 Processus markoviens

1.1.1 Introduction

Les systèmes de gestion de masses sont des systèmes où des services sont offerts aux clients (personnes ou objets) et qu'on essaye de modéliser et d'optimiser par des processus.

Les processus aléatoires décrivent l'évolution d'une grandeur aléatoire en fonction du temps. Mathématiquement, un processus stochastique $\{X_t, t \in T\}$ permet de modéliser l'évolution dynamique d'un système aléatoire indexé par le temps t et défini sur un même espace de probabilité $(\Omega, \mathfrak{F}, P)$.

On va se limiter aux processus de Markov, c'est-à-dire aux processus dont leurs évolutions futures ne dépendent du passé qu'à travers leurs valeurs actuelles.

Définition 1.1

On dit que $(X_t)_{t \in T}$ est un processus de Markov si $\forall u \leq s \leq t \in T$ et $\forall x, i, j \in E$, $P(X_t = j / X_s = i, X_u = x) = P(X_t = j / X_s = i) = P_{ij}(t, s)$, propriété de Markov ou propriété d'absence de mémoire.

Si $P_{ij}(t, s) = P_{ij}(t - s)$, alors le processus $(X_t)_{t \in T}$ est dit processus markovien homogène. Dans ce qui suit, on suppose que $\{X_t, t \in T\}$ est markovien et homogène et donc

$$P(X_{(t+s)} = j / X_s = i) = P_{ij}(t + s - s) = P_{ij}(t).$$

Elle est appelée probabilité de transition de l'état i à l'état j pendant le laps de temps t (une durée t).

Soit $P(t) = (P_{ij}(t))_{(i,j) \in E * E}$, elle est appelée matrice de transition du système à l'instant t , (matrice de fonctions).

Proposition 1.1

1. $\sum_{j \in E} P_{ij}(t) = 1, \forall i \in E$.
2. $P_{ik}(t+s) = \sum_{j \in E} P_{ij}(t) P_{jk}(s)$, (équations de Chapman).

Remarque

La matrice stochastique $P(t)$ caractérise le processus markovien $(X_t)_{t \in T}$, c'est-à-dire on peut connaître n'importe quelle caractéristique (performances, position, ...) du système (processus) à partir de $P(t)$.

En effet, à partir de $P(t)$, on peut connaître la loi de n'importe quel vecteur $(X_{t_0}, X_{t_1}, \dots, X_{t_n})$, en particulier celle de n'importe quelle variable X_t . La loi conjointe (fini-dimensionnelle) est donnée en fonction de $P(t)$ et la loi initiale.

$$\begin{aligned} P(X_{t_n} = a_n, X_{t_{n-1}} = a_{n-1}, \dots, X_{t_1} = a_1, X_0 = a_0) \\ &= P(X_{t_n} = a_n / X_{t_{n-1}} = a_{n-1}) \dots P(X_{t_1} = a_1 / X_0 = a_0) P(X_0 = a_0) \\ &= P_{a_{n-1}a_n}(t_n - t_{n-1}) P_{a_{n-2}a_{n-1}}(t_{n-1} - t_{n-2}) \dots P_{a_0a_1}(t_1) P(X_0 = a_0). \end{aligned}$$

1.1.2 Loi t-instantanée du système

Soit $\pi(t) = \{P(X_t = j), j \in E\}$.

On a $\pi(t) = \pi(0) * P(t)$ avec $\pi(0) = \{P(X_0 = i), i \in E\}$.

En effet, $\forall j \in E$

$$\begin{aligned} P(X_t = j) &= \sum_{k \in E} P(X_t = j, X_0 = k) = \sum_{k \in E} P(X_t = j / X_0 = k) P[X_0 = k] \\ &= \sum_{k \in E} P_{kj}(t) P(X_0 = k). \end{aligned}$$

C'est-à-dire

$$P(X_t = j) = \sum_{k \in E} P(X_0 = k) P_{kj}(t), \forall j \in E.$$

D'où $\pi(t) = \pi(0) * P(t)$, (écriture matricielle).

1.1.3 Générateur infinitésimal d'un système markovien

On suppose que $\forall (i, j) \in E * E$, la fonction $P_{ij}(\cdot)$ est continue en 0, c'est-à-dire

$$\lim_{t>0} P_{ij}(t) = P_{ij}(0) = \begin{cases} 1 & i = j, \\ 0 & i \neq j. \end{cases}$$

On pose

pour $i \neq j$, $q_{ij} = \lim_{t>0} \frac{P_{ij}(t)}{t} = P'_{ij}(0)$, quand elle existe.

pour $i = j$, $q_{ii} = \lim_{t>0} \frac{P_{ij}(t)-1}{t} = P'_{ii}(0)$, quand elle existe.

On appelle générateur infinitésimal du processus $(X_t)_{t \in T}$, la matrice $Q = (q_{ij})_{(i,j) \in E \times E}$.

Proposition 1.2

- $\sum_{j \in E} q_{ij} = 0, \forall i \in E$.
- $P'(t) = QP(t) = P(t)Q$.

Preuve

Comme $\sum_{j \in E} P_{ij}(t) = 1, \forall i \in E$ alors $\left(\sum_{j \in E} P_{ij}(t)\right)' = \left(\sum_{j \in E} (P_{ij}(t))'\right) = 0$.

Ainsi $\sum_{j \in E} q_{ij} = 0, \forall i \in E$.

On a l'équation de Chapman-Kolmogorov $P_{ik}(t+s) = \sum_{j \in E} P_{ij}(t) P_{jk}(s), \forall i, k \in E$.

On dérive cette équation par rapport à $t = 0$:

$$\left. \frac{\partial P_{ik}(t+s)}{\partial t} \right|_{t=0} = \sum_{j \in E} \left. \frac{\partial P_{ij}(t)}{\partial t} \right|_{t=0} P_{jk}(s) = \sum_{j \in E} P'_{ij}(0) p_{jk}(s) = \sum_{j \in E} q_{ij} P_{jk}(s).$$

C'est à dire

$$P'(s) = Q * P(s), \forall i, k \in E.$$

On fait la même chose par rapport à s , c'est-à-dire on dérive l'équation de Chapman-Kolmogorov par rapport à s en 0 et on obtient $P'(t) = P(t) * Q$.

1.1.4 Modélisation du système par les équations différentielles de Chapman- Kolmogorov

On a

$$\begin{cases} P'(t) = Q * P(t), \\ P(0) = I_E. \end{cases}$$

La solution est donnée par

$$P(t) = \sum_{n \geq 0} \frac{(Qt)^n}{n!} = \lim_{n \rightarrow \infty} \sum_{k=0}^n \frac{(Qt)^k}{k!} \text{ avec } Q^0 = I_{E \times E}, \text{ et on note } P(t) = e^{Qt}.$$

Si Q est de dimension finie, c'est-à-dire E est fini et dans le cas où Q est diagonalisable, c'est-à-dire il existe une matrice P inversible telle que $Q = PDP^{-1}$ avec

$$D = \begin{bmatrix} \lambda_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_n \end{bmatrix} \text{ et } P = [V_1, V_2, \dots, V_n], \text{ avec } \begin{array}{l} \lambda_i : \text{ Valeurs propres.} \\ V_i : \text{ Vecteurs propres.} \end{array}$$

Alors la solution est $P(t) = P\Delta(t)P^{-1}$ où

$$\Delta(t) = \begin{bmatrix} e^{\lambda_1 t} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & e^{\lambda_n t} \end{bmatrix}$$

1.1.5 Solution stationnaire d'un processus de Markov

Définition 1.2

Une loi $\pi(t)$ d'un processus de Markov $(X_t)_{t \in T}$ est dite solution (loi) stationnaire si cette loi $\pi(t)$ ne dépend pas de t , c'est-à-dire

$$\pi(t) = \pi^* = \{P(X_t = i), \forall i \in E\}.$$

1.1.6 Caractérisation d'une loi stationnaire

Une loi $\pi^* = (\pi_i, i \in E)$ est une loi stationnaire si et seulement si elle est solution du système

$$\begin{cases} \pi^* * Q = 0, \\ \sum_{i \in E} \pi_i = 1. \end{cases}$$

1.1.7 Ergodicité d'un processus de Markov

Un processus de Markov $(X_t)_{t \in T}$ est dit ergodique si la loi $\pi^* = (\pi_j, j \in E)$ définie par $\pi_j = \lim_{t \rightarrow +\infty} P_{ij}(t)$, $\forall i \in E, \forall j \in E$ est l'unique solution stationnaire du processus de Markov, c'est-à-dire le système

$$\begin{cases} \pi^* * Q = 0, \\ \sum_{j \in E} \pi_j = 1. \end{cases} \text{ admet une seule solution } \pi^* = (\pi_j, j \in E),$$

avec

$$\pi_j = \lim_{t \rightarrow +\infty} P_{ij}(t), \forall i, j \in E.$$

Exemple 1.1

Parmi les processus qui vérifient la propriété de Markov (absence de mémoire), on cite ces deux processus :

1. Processus de poisson

Un processus de comptage $N(t)_{t \in T = \mathbb{R}_+}$ est dit processus de Poisson si :

- $N(0) = 0$.
- $N(t)_{t \in T = \mathbb{R}_+}$ est un PAIS (processus à accroissements indépendants et stationnaires).
- $P(N_{dt} = k) = \begin{cases} \lambda dt + o(dt) & k = 1, \\ 1 - \lambda dt + o(dt) & k = 0, \\ o(dt) & k \geq 2. \end{cases}$

Ainsi pour le processus de Poisson de taux λ , sa loi sera donnée par la formule suivante :

$$P_n(t) = P(N_t = n) = \frac{(\lambda t)^n}{n!} e^{-\lambda t}, \forall n \in \mathbb{N}.$$

2. Processus de naissance et de mort

Il s'agit d'un processus stochastique $(N_t)_{t \in T}$ à temps continu qui consiste à faire évoluer un système entre une infinité dénombrable d'états, vérifiant les hypothèses suivantes :

H_1 : À partir d'un état n à l'instant t , le processus ne pourra passer à l'instant $(t + dt)$ que dans l'un des états $n - 1, n, n + 1$ (dans un laps de temps très court dt , au plus il y a une occurrence).

H_2 : Ce processus est un processus à accroissements stationnaires et à accroissements indépendants.

H_3 : Au plus un évènement peut survenir à l'instant t .

On appelle alors naissance à la date t , le passage du système de l'état n à l'état $n + 1$ avec un taux de naissance λ_n , et mort à la date t , le passage de l'état n à l'état $n - 1$ avec un taux de mort μ_n .

On obtient ainsi la loi du système en régime stationnaire qui est notée P_n .

$$P_n = \begin{cases} P_0 = \frac{1}{1 + \sum_{n \geq 1} C_n} & n = 0, \\ P_n = C_n P_0 & n \geq 1, \end{cases} \text{ avec } C_n = \prod_{i=1}^n \frac{\lambda_{i-1}}{\mu_i}, \text{ à condition que } \sum_{n \geq 1} C_n < +\infty.$$

Cas particulier

$\lambda_n = \lambda, \forall n \geq 0$ et $\mu_n = \mu, \forall n \geq 1$, $\sum_{n \geq 1} C_n = \sum_{n \geq 1} \left(\frac{\lambda}{\mu}\right)^n = \sum_{n \geq 1} \rho^n$, avec $\rho = \frac{\lambda}{\mu}$ et $\sum_{n \geq 1} C_n < +\infty$ si $\rho < 1 \Leftrightarrow \lambda < \mu$.

1.2 Systèmes de files d'attente

1.2.1 Introduction

Dans cette partie, nous présentons les principaux éléments de la théorie des files d'attente issus du calcul de probabilités, comme les lois des différents types de files (après stationnarité), les performances liées au système d'attente,...

L'objectif de cette théorie est de rendre compte sur les phénomènes d'attente liés aux caractères aléatoires et imprévisibles des événements rencontrés. Nous nous intéressons aux modèles markoviens, en particulier aux arrivées poissonniennes et aux durées de service indépendantes et de même loi exponentielle.

1.2.2 Description

On considère un système destiné à offrir un certain service, celui (personne, objet) qui veut bénéficier de ce service est dit client et les postes de service sont appelés serveurs (conseillers) ou guichets.

File d'attente : lieu où attendent les clients avant d'être servis.

Systèmes d'attente : file d'attente + service en cours.

L'étude de ces systèmes porte sur la qualité et le rendement du service fourni.

Ce dernier sera caractérisé par la description de :

- La file d'attente (longueur de la file, durée d'attente d'un client,...).
- Les serveurs (nombre de clients servis par période d'activité, durée de répit (oisiveté),...).

Cette étude peut avoir deux buts qui se résument dans :

1. L'amélioration du fonctionnement dans le cadre des structures actives du système.
2. L'étude de l'investissement à consentir pour améliorer la qualité et le rendement du service.

1.2.3 Caractéristiques d'un système d'attente

Les six caractères essentiels d'un système d'attente sont :

1. Loi des arrivées des clients dans le système (poisson, λ ,...), elle sera noté L_1 .
2. Loi des durées de service de chaque client dans chaque guichet D_i (indépendantes de même loi), cette loi sera notée L_2 .
3. Le nombre de serveurs dans le système, il sera noté s ($s \geq 1$).
4. La capacité du système (c'est le nombre maximum de clients qui peuvent être présents simultanément dans le système, c'est-à-dire les clients en attente et les clients en service), elle sera noté K .
5. La discipline de service : il s'agit de la règle déterminant l'ordre dans lequel les clients vont accéder aux serveurs lorsqu'ils sont libres, les disciplines les plus courantes sont :

- FIFO (PAPS) : premier arrivé, premier servi.
 - LIFO (DAPS) : dernier arrivé, premier servi.
 - SP (service avec priorité).
 - RS (service aléatoire).
6. Nature du système, nous distinguons deux cas :
- Système ouvert à tous les clients, il sera noté [O].
 - Système ouvert à des clients particuliers, il sera noté [F].
- Autrement dit, le système est ouvert à tous les clients ou bien à des clients particuliers.

1.2.4 Notation de Kendall

C'est une notation qui permet de décrire un système d'une file d'attente en ramenant sa description textuelle à une formule symbolique qui est :

$(L, L_2, s, K, FIFO$ ou autre discipline, $[O]$ ou $[F]$).

Remarque

La file d'attente notée (L_1, L_2, s) représente par défaut la file $(L_1, L_2, s, \infty, FIFO, [O])$.

1.2.5 Etude du système M/M/1

C'est un système d'attente avec un seul serveur qui sert les clients par ordre de leurs arrivées avec une capacité infinie et ouvert à tous les clients.

Le processus des arrivées est poissonien de taux λ et les durées de service des clients sont indépendantes de loi commune exponentielle de paramètre μ .

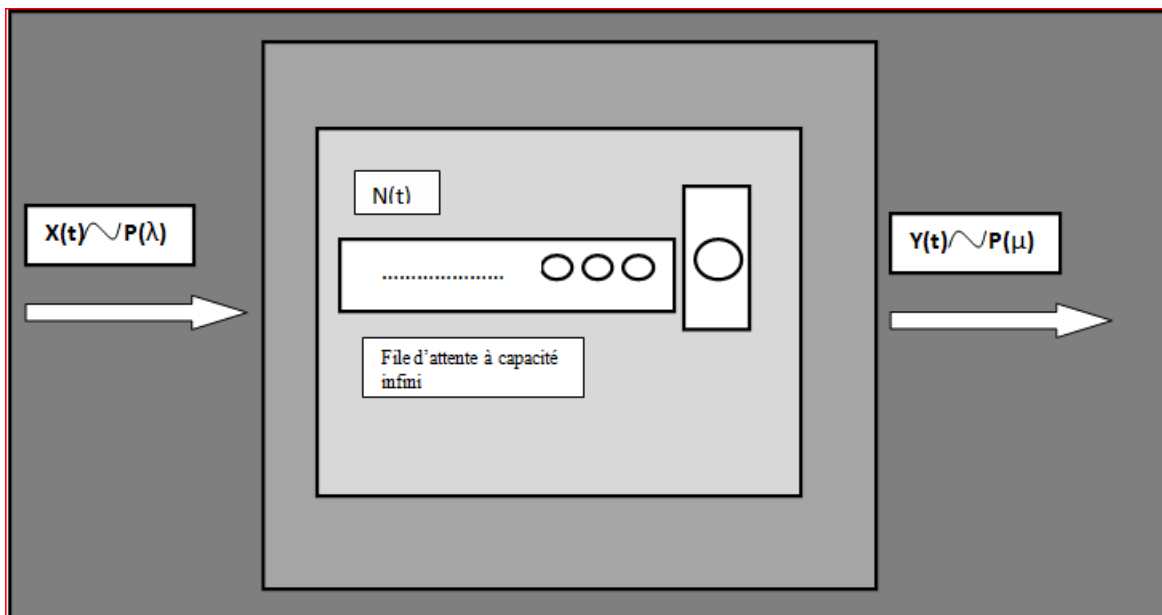


FIGURE 1.1 – Système d'attente (M/M/1)

- (X_t) est un processus de comptage, il est poissonien de taux λ , il compte le nombre de clients qui arrivent dans le système à la date t .
- (Y_t) est un processus de comptage, il est poissonien de taux μ , il compte le nombre de clients qui sortent à la date t .
- (N_t) est un processus de comptage, il compte le nombre de clients en cours et en attente de service à la date t .

1.2.5.1 Etude du processus $(N_t)_{t \geq 0}$

Il s'agit d'un processus de naissance et de mort (PNM) à espace d'états infini $E = \{0, 1, \dots\} = \mathbb{N}$, de taux λ_n et μ_n à déterminer :

a. Taux de naissance λ_n

On a

$$\begin{aligned} \lambda_n dt + o(dt) &= P[N_{t+dt} = n + 1 / N_t = n] = P[X_{t+dt} = X_t + 1] \\ &= P[X_{t+dt} - X_t = 1] = P[X_{dt} = 1] = \lambda dt + o(dt) \\ &\Rightarrow \lambda_n = \lambda, \forall n \geq 0. \end{aligned}$$

b. Taux de mort μ_n

On a

$$\begin{aligned} \mu_n dt + o(dt) &= P[N_{t+dt} = n - 1 / N_t = n] = P[Y_{t+dt} = Y_t + 1] \\ &= P[Y_{t+dt} - Y_t = 1] = P[Y_{dt} = 1] = \mu dt + o(dt) \\ &\Rightarrow \mu_n = \begin{cases} \mu & n \geq 1, \\ 0 & n = 0. \end{cases} \end{aligned}$$

1.2.5.2 Loi du système en régime permanent (stable)

On note $P_n = P[N_t = n]$, $n \geq 0$, la loi du système lorsqu'il se stabilise, on a

$$P_n = \begin{cases} \frac{a_n}{1 + \sum_{n \geq 1} a_n} & n \geq 1, \\ \frac{1}{1 + \sum_{n \geq 1} a_n} & n = 0, \end{cases}$$

avec

$$a_n = \prod_{i=1}^n \frac{\lambda_{i-1}}{\mu_i} = \frac{\lambda_0 \lambda_1 \dots \lambda_{n-1}}{\mu_1 \mu_2 \dots \mu_n} = \frac{\lambda^n}{\mu^n} = \left(\frac{\lambda}{\mu} \right)^n.$$

On pose $\frac{\lambda}{\mu} = \rho$, donc $a_n = \rho^n$ et

$$P_n = \begin{cases} \frac{\rho^n}{1 + \sum_{n \geq 1} \rho^n} & n \geq 1, \\ \frac{1}{1 + \sum_{n \geq 1} \rho^n} & n = 0. \end{cases}$$

Ainsi le système se stabilise si $\sum_{n \geq 1} \rho^n < +\infty$, c'est-à-dire $\rho < 1 \iff \lambda < \mu$.

$$\text{On aura alors } P_n = \begin{cases} \rho^n (1 - \rho) & n \geq 1, \\ 1 - \rho & n = 0. \end{cases}$$

C'est-à-dire $P_n = \rho^n (1 - \rho)$, $n \geq 0$ avec $\rho < 1$.

1.2.5.3 Performances de la file M/M/1

A. Caractéristiques du système liées aux clients

a. Nombre moyen de clients dans le système à la date t

Le nombre moyen de clients dans le système à la date t se traduit par l'espérance mathématique notée η et qui est donnée par la formule suivante :

On a

$$\begin{aligned} \eta &= E(N_t) = \sum_{n \geq 0} n P[N_t = n] = \sum_{n \geq 0} n p_n = \sum_{n \geq 0} n \rho^n (1 - \rho) \\ &= \rho (1 - \rho) \sum_{n \geq 1} n \rho^{n-1} = \rho (1 - \rho) \sum_{n \geq 1} (\rho^n)' \\ &= \rho (1 - \rho) \left(\sum_{n \geq 0} \rho^n \right)' = \rho (1 - \rho) \left(\frac{1}{1 - \rho} \right)' = \rho (1 - \rho) \frac{1}{(1 - \rho)^2} \\ &= \frac{\rho}{1 - \rho} = \frac{\frac{\lambda}{\mu}}{1 - \frac{\lambda}{\mu}} \\ &\Rightarrow \eta = \frac{\lambda}{\mu - \lambda}. \end{aligned}$$

b. Nombre moyen de clients dans la file à la date t

Le nombre moyen de client en attente dans la file à la date t noté η_q est la donnée de l'espérance mathématique qui est $E(M_t)$, avec M_t le nombre de clients dans la file à la date t .

On a

$$\begin{aligned}\eta_q = E(M_t) &= \sum_{n \geq 0} nP[M_t = n] = \sum_{n \geq 0} nP[N_t = n + 1] = \sum_{n \geq 0} nP_{n+1} \\ &= \sum_{n \geq 0} n\rho^{n+1}(1 - \rho) = \rho^2(1 - \rho) \sum_{n \geq 0} n\rho^{n-1} \\ &= \rho^2(1 - \rho) \sum_{n \geq 0} (\rho^n)' = \rho^2(1 - \rho) \left(\sum_{n \geq 0} \rho^n \right)' = \frac{\rho^2}{(1 - \rho)} \\ &\Rightarrow \eta_q = \eta - \rho.\end{aligned}$$

c. Durée moyenne de séjour d'un client dans le système

On utilise la formule de Little : $\bar{W} = \frac{\eta}{\Lambda}$ avec $\Lambda = \sum_{i=0}^n \lambda_i P_i$

$$\Lambda = \sum_{i=0}^n \lambda_i P_i = \lambda_0 P_0 + \lambda_1 P_1 + \dots + \lambda_n P_n + \dots$$

c'est à dire

$$\Lambda = \sum_{i=0}^n \lambda P_i = \lambda \sum_{i=0}^n P_i = \lambda \text{ car } \left(\sum_{i=0}^n P_i = 1 \right).$$

D'où $\Lambda = \lambda$.

Alors

$$\bar{W} = \frac{\eta}{\lambda} = \frac{\lambda}{(\mu - \lambda)\lambda} = \frac{1}{\mu - \lambda}.$$

d. Durée moyenne d'attente d'un client dans la file

On utilise la formule de Little : $\bar{W}_q = \frac{\eta_q}{\Lambda}$,

$$\text{avec } \Lambda = \sum_{i=0}^k \lambda_i P_i = \lambda \text{ et } \eta_q = \frac{\lambda^2}{\mu(\mu - \lambda)}$$

c'est à dire

$$\bar{W}_q = \frac{\eta_q}{\Lambda} = \frac{\lambda^2}{\mu(\mu - \lambda)\lambda} = \frac{\lambda}{\mu(\mu - \lambda)}.$$

B. Caractéristiques du système liées au serveur

a. Durée moyenne d'une période de répit (oisiveté), notée \bar{R}

Soit R la durée d'une période de répit, R est une variable aléatoire et \bar{R} sa moyenne. Une période de répit commence à la fin de service du dernier client dans le système et s'achève lorsqu'un client rentre, ou continue tant qu'aucun client ne rentre.

Pour $s \geq 0$,

$$F_R(s) = P[R \leq s] = 1 - P[R > s].$$

On a

$$P[R > s] = P[X_{t+s} = X_t] = P[X_{t+s} - X_t = 0].$$

On sait bien que $(X_t)_{t \geq 0} \rightsquigarrow P(\lambda)$, alors il est à accroissements stationnaires, donc

$$P[R > s] = P[X_s = 0] = \exp^{-\lambda s}.$$

$$\Rightarrow F_R(s) = \begin{cases} 1 - e^{-\lambda s} & s \geq 0, \\ 0 & \text{sinon.} \end{cases}$$

Ainsi

$$\Rightarrow f_R(s) = \begin{cases} \lambda e^{-\lambda s} & s \geq 0, \\ 0 & \text{sinon.} \end{cases}$$

D'où R suit une loi exponentielle de paramètre λ , c'est-à-dire

$$\bar{R} = \frac{1}{\lambda}.$$

b. Nombre moyen de périodes de répit, noté \bar{A}

Soit A le nombre de périodes de répit, A est une variable aléatoire.

Soit T le temps d'activité global. Le serveur est oisif pendant une proportion de temps p qui est égale à

$$p = P[\text{le système est vide}] = P[N_t = 0] = P_0 = 1 - \rho,$$

car $P_n = \rho^n (1 - \rho)$.

On a

$$P_0 \times T = \bar{A} \times \bar{R} \Rightarrow \bar{A} = \frac{P_0 \times T}{\bar{R}}$$

$$\Rightarrow \bar{A} = \lambda(1 - \rho)T.$$

c. Nombre moyen de périodes d'activité (en régime permanent), noté \bar{B}

À cause de l'alternance activité-répit, il y a un nombre moyen de périodes d'activité égal au nombre moyen de périodes de répit.

On a $A = B$, ce qui veut dire qu'en moyenne $\bar{A} = \bar{B}$.

$$\Rightarrow \bar{B} = \lambda(1 - \rho)T.$$

d. Durée moyenne d'une période d'activité (en régime permanent), notée \bar{C}

On note C la durée d'une période d'activité, C est une variable aléatoire.

Si le serveur est oisif pendant une proportion $p = p_0$ du temps global T , alors il sera en activité pendant une proportion $1 - p_0 = 1 - (1 - \rho) = \rho$ du temps T .

À cause aussi de l'alternance activité-répit, on a en moyenne \bar{B} périodes d'activité.

Donc

$$\Rightarrow \bar{B} \times \bar{C} = \rho \times T \Rightarrow \bar{C} = \frac{\rho \times T}{\lambda(1-\rho)T},$$

c'est-à-dire

$$\bar{C} = \frac{1}{\mu - \lambda}.$$

e. Nombre moyen de clients servis par période d'activité (en régime permanent), noté \bar{N}_c

Soit N_c le nombre de clients servis par période d'activité.

On sait que la durée de service D d'un client est $\exp(\mu)$, alors $\bar{D} = \frac{1}{\mu}$.

On a alors

$$\bar{N}_c \times \bar{D} = \bar{C},$$

$$\Rightarrow \bar{N}_c = \frac{\bar{C}}{\bar{D}} = \frac{1}{\frac{1}{\mu}},$$

c'est-à-dire

$$\bar{N}_c = \frac{\mu}{\mu - \lambda}.$$

1.2.6 Etude de système M/M/1/K

Il s'agit d'un système d'attente qui sert les clients avec un seul serveur suivant leurs ordres d'arrivée. Sa capacité est limitée à K clients, c'est-à-dire que $N_t \in \{0, 1, \dots, K\}$ et que le service est accessible par tous les clients potentiels, ce qui veut dire le système est ouvert.

Les clients arrivent suivant un processus de Poisson de taux λ et les durées de service sont exponentielles de paramètre μ .

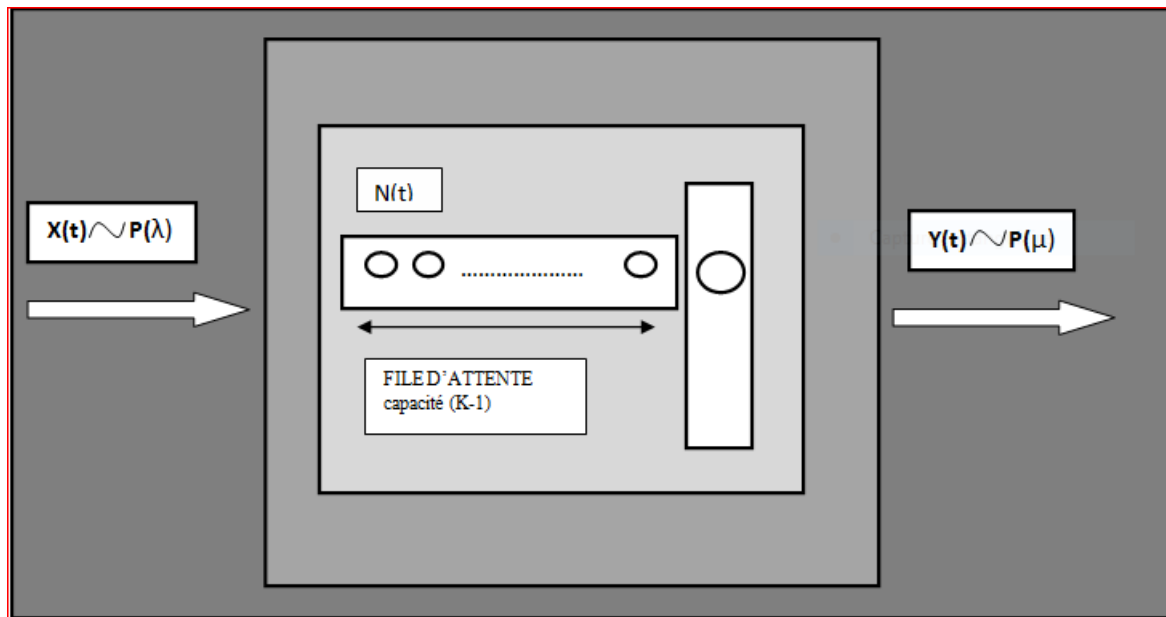


FIGURE 1.2 – Système d'attente (M/M/1/K)

- (X_t) est un processus de comptage, il est poissonien de taux λ , il compte le nombre de clients qui arrivent dans le système à la date t .
- (Y_t) est un processus de comptage, il est poissonien de taux μ , il compte le nombre de clients qui sortent à la date t .
- (N_t) est un processus de comptage, il compte le nombre de clients en cours et en attente de service à la date t .

1.2.6.1 Etude du processus $(N_t)_{t \geq 0}$

$(N_t)_{t \geq 0}$ est un processus de naissance et de mort (PNM) de taux λ_n et μ_n à déterminer :

a. Taux de naissance λ_n

On a

$$\lambda_n dt + o(dt) = P[N_{t+dt} = n + 1 / N_t = n] = P[X_{t+dt} = X_t + 1] = P[X_{t+dt} - X_t = 1]$$

$$= P[X_{dt} = 1] = \lambda dt + o(dt).$$

$$\Rightarrow \lambda_n = \begin{cases} \lambda & 0 \leq n \leq K-1, \\ 0 & \text{sinon.} \end{cases}$$

b. Taux de mort μ_n

On a

$$\mu_n dt + o(dt) = P[N_{t+dt} = n-1 / N_t = n] = P[Y_{t+dt} = Y_t + 1] = P[Y_{t+dt} - Y_t = 1]$$

$$= P[Y_{dt} = 1] = \mu dt + o(dt).$$

$$\Rightarrow \mu_n = \begin{cases} 0 & n = 0, \\ \mu & 0 < n \leq K, \\ 0 & n > K. \end{cases}$$

1.2.6.2 Loi du système en régime permanent (stable)

On note $P_n = P[N_t = n]$, $0 \leq n \leq K$, la loi du système en régime permanent, lorsqu'il existe.

On a

$$P_n = a_n \times P_0,$$

avec $a_n = \prod_{i=1}^n \frac{\lambda_{i-1}}{\mu_i}$, $1 \leq n \leq K$, et

$$a_n = \prod_{i=1}^n \frac{\lambda_{i-1}}{\mu_i} = \frac{\lambda_0 \lambda_1 \dots \lambda_{n-1}}{\mu_1 \mu_2 \dots \mu_n} = \frac{\lambda^n}{\mu^n} = \left(\frac{\lambda}{\mu}\right)^n = \rho^n, \quad (\rho = \frac{\lambda}{\mu}).$$

Donc

$$\sum_{n=0}^K P_n = P_0 + \sum_{n=1}^K P_n = P_0 \left(1 + \sum_{n=1}^K \prod_{i=1}^n \frac{\lambda_{i-1}}{\mu_i}\right).$$

Ainsi

$$\sum_{n=0}^K P_n = P_0 \left(1 + \sum_{n=1}^K \rho^n\right) = 1,$$

c'est-à-dire

$$P_0 = \frac{1}{\sum_{n=0}^K \rho^n}.$$

Remarque

P_0 existe toujours car le dénominateur est une somme finie, c'est-à-dire le régime stationnaire s'établit $\forall \lambda, \mu > 0$.

Finalement, on distingue deux cas.

1^{er} cas :

$$\lambda = \mu \Rightarrow \rho = 1.$$

Ainsi

$$P_n = \rho^n \times P_0 \text{ et } P_0 = \frac{1}{\sum_{n=0}^K \rho^n} = \frac{1}{K+1},$$

c'est-à-dire

$$P_n = \frac{1}{K+1}, 0 \leq n \leq K.$$

2^{ème} cas :

$$\lambda \neq \mu \Rightarrow \rho \neq 1 \implies \frac{1-\rho}{1-\rho^{K+1}}.$$

Ainsi

$$P_n = \rho^n \times P_0 = \frac{\rho^n (1-\rho)}{1-\rho^{K+1}}, 0 \leq n \leq K.$$

1.2.6.3 Performances de la file M/M/1/K

On note par

- η : nombre moyen de clients dans le système à l'instant t .
- W : durée moyenne de séjour d'un client dans le système, et \bar{W} sa moyenne.

a. Nombre moyen de clients dans le système

On a

$$\eta = E(N_t) = \sum_{n=0}^K n P[N_t = n] = \sum_{n=0}^K n P_n$$

- Si $\lambda = \mu$

$$\eta = \sum_{n=0}^K n \frac{1}{K+1} = \frac{1}{K+1} \sum_{n=1}^K n = \frac{1}{K+1} \left[(K) \frac{K+1}{2} \right]$$

$$\Rightarrow \eta = \frac{K}{2}.$$

• Si $\lambda \neq \mu$

$$\begin{aligned} \eta &= \sum_{n=0}^K n \rho^n \frac{(1-\rho)}{1-\rho^{K+1}} = \frac{\rho(1-\rho)}{1-\rho^{K+1}} \sum_{n=0}^K n \rho^{n-1} = \frac{\rho(1-\rho)}{1-\rho^{K+1}} \sum_{n=0}^K (\rho^n)' \\ &= \frac{\rho(1-\rho)}{1-\rho^{K+1}} \left(\sum_{n=0}^K \rho^n \right)' = \frac{\rho(1-\rho)}{1-\rho^{K+1}} \left(\frac{1-\rho^{K+1}}{1-\rho} \right)' = \frac{\rho(1-\rho)[1-(K+1)\rho^K + K\rho^{K+1}]}{(1-\rho^{K+1})(1-\rho)^2} \\ &\Rightarrow \eta = \frac{\rho[1-(K+1)\rho^K + K\rho^{K+1}]}{(1-\rho)(1-\rho^{K+1})}. \end{aligned}$$

b. Durée moyenne de séjour d'un client dans le système

On utilise la formule de Little : $\bar{W} = \frac{\eta}{\Lambda}$,

avec

$$\begin{aligned} \Lambda &= \sum_{i=0}^K \lambda_i P_i = \sum_{i=0}^{K-1} \lambda P_i \\ &= \lambda(1 - P_K). \end{aligned}$$

Pour $\lambda = \mu$

$$\begin{aligned} \bar{W} &= \frac{\eta}{\lambda(1 - P_K)} = \frac{K}{2\lambda(1 - P_K)} = \frac{K}{2\lambda\left(1 - \frac{1}{K+1}\right)}, \\ \bar{W} &= \frac{K+1}{2\lambda}. \end{aligned}$$

Pour $\lambda \neq \mu$

$$\begin{aligned} \bar{W} &= \frac{\eta}{\lambda(1 - P_K)} = \frac{1}{\lambda\left(1 - \frac{\rho^K(1-\rho)}{1-\rho^{K+1}}\right)} * \frac{\rho[1-(K+1)\rho^K + K\rho^{K+1}]}{(1-\rho)(1-\rho^{K+1})}, \\ \bar{W} &= \frac{\rho[1-(K+1)\rho^K + K\rho^{K+1}]}{[\lambda(1-\rho^K)] * (1-\rho)}. \end{aligned}$$

1.2.7 Etude du système (M/M/1/K/[F])

On considère un système d'attente à un seul serveur, qui sert les clients dans leurs ordres d'arrivée, on suppose que les clients à servir sont des particuliers et qu'ils sont limités à un nombre K .

Les clients arrivent suivant un processus de Poisson de taux λ et les durées de service sont indépendantes, de même loi exponentielle de paramètre μ .

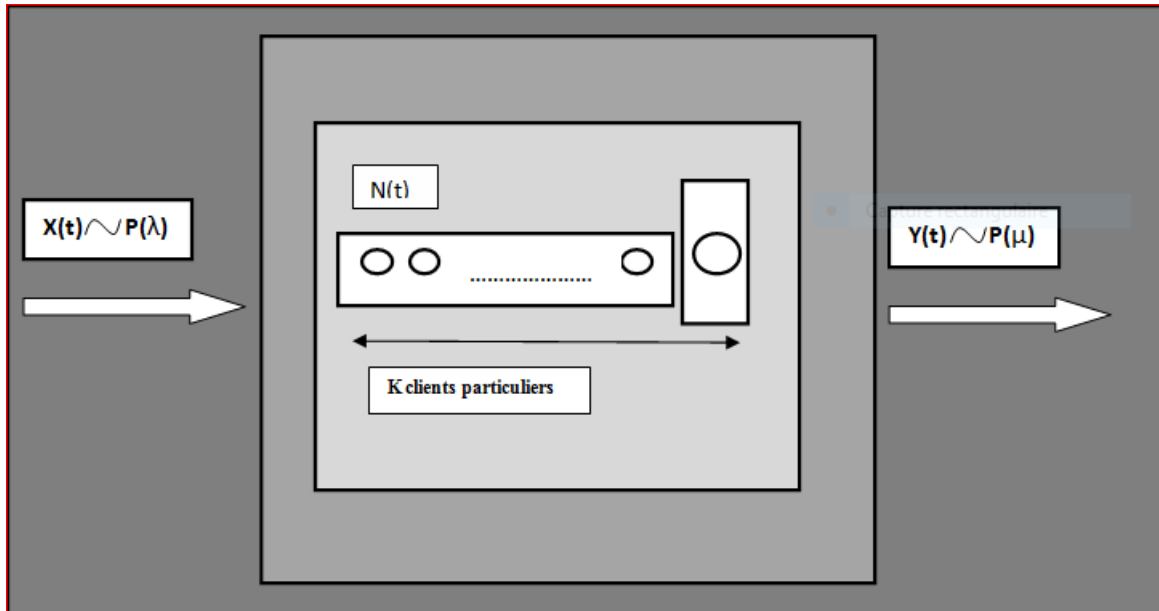


FIGURE 1.3 – Système d'attente (M/M/1/K/[F])

- (X_t) est un processus de comptage, il est poissonien de taux λ , il compte le nombre de clients qui arrivent dans le système à la date t
- (Y_t) est un processus de comptage, il est poissonien de taux μ , il compte le nombre de clients qui sortent à la date t .
- (N_t) est un processus de comptage, il compte le nombre de clients en cours et en attente de service à la date t .

On s'intéresse au processus $(N_t)_{t \geq 0}$.

1.2.7.1 Etude du processus $(N_t)_{t \geq 0}$

$(N_t)_{t \geq 0}$ est un processus de naissance et de mort (PNM) de taux λ_n et μ_n à déterminer :

a. Taux de naissance λ_n

On a

$$\lambda_n dt + o(dt) = P [N_{t+dt} = n + 1 / N_t = n]$$

$$P = [\text{avoir une entrée parmi les } (K - n) \text{ restants}]$$

$$= C_{K-n}^1 P[X_{t+dt} = X_t + 1] = \frac{(K-n)!}{1!(K-n-1)!} (\lambda dt + o(dt)) = (K-n) \lambda dt + o(dt)$$

$$\Rightarrow \lambda_n = \begin{cases} (K-n) \lambda & 0 \leq n \leq K-1, \\ 0 & n \geq K. \end{cases}$$

b. Taux de mort μ_n

On a

$$\mu_n dt + o(dt) = P[N_{t+dt} = n / N_t = n + 1]$$

P = [avoir une sortie (client qui a terminé son service)]

$$= P[Y_{t+dt} = Y_t + 1] = \mu dt + o(dt)$$

$$\Rightarrow \mu_n = \begin{cases} 0 & n = 0, \\ \mu & 1 \leq n \leq K, \\ 0 & n > K. \end{cases}$$

1.2.7.2 Loi du système en régime permanent (stable)

Comme c'est une somme finie, la capacité du système étant finie, alors le régime stationnaire s'établit.

On note $P_n = P[N_t = n]$, $0 \leq n \leq K$, la loi du système en régime permanent.

On a

$$P_n = \begin{cases} a_n \times P_0 & 1 \leq n \leq K, \\ P_0 & n = 0, \end{cases}$$

avec

$$a_n = \prod_{i=1}^n \frac{\lambda_{i-1}}{\mu_i} \text{ et } P_0 = \frac{1}{1 + \sum_{n=1}^K \prod_{i=1}^n \frac{\lambda_{i-1}}{\mu_i}}.$$

On a

$$a_n = \prod_{i=1}^n \frac{\lambda_{i-1}}{\mu_i} = \frac{\lambda_0 \lambda_1 \dots \lambda_{n-1}}{\mu_1 \mu_2 \dots \mu_n} = \frac{K \lambda \times (K-1) \lambda \times \dots \times (K-n+1) \lambda}{\mu \times \mu \times \dots \times \mu} = \frac{K!}{(K-n)!} \frac{\lambda^n}{\mu^n}$$

$$= \frac{K!}{(K-n)!} \left(\frac{\lambda}{\mu}\right)^n,$$

on pose $\frac{\lambda}{\mu} = \rho$, ainsi

$$P_n = \begin{cases} \frac{K!}{(K-n)!} \rho^n P_0 & 1 \leq n \leq K, \\ \frac{1}{1 + \sum_{n=1}^K \frac{K!}{(K-n)!} \rho^n} & n = 0, \end{cases}$$

$$= \begin{cases} n! C_K^n \rho^n P_0 & 1 \leq n \leq K, \\ \frac{1}{\sum_{n=0}^K n! C_K^n \rho^n} & n = 0. \end{cases}$$

D'où

$$P_n = \frac{n! C_K^n \rho^n}{\sum_{n=0}^K n! C_K^n \rho^n}, \forall 0 \leq n \leq K.$$

1.2.7.3 Performances de la file M/M/1/K/[F]

On note par

η : nombre moyen de clients dans le système à l'instant t .

\bar{W} : durée moyenne de séjour d'un client dans le système, et \bar{W} sa moyenne.

a. Nombre moyen de clients dans le système

On a

$$\eta = E(N_t) = \sum_{i=0}^K iP[N_t = i] = \sum_{i=0}^K iP_i.$$

On a le taux d'entrée global égal au taux de sortie global tel que

$$\Lambda = \sum_{i=0}^K \lambda_i P_i = \sum_{i=0}^K \mu_i P_i \Leftrightarrow \sum_{i=0}^K (K-i) \lambda P_i = \sum_{i=1}^K \mu P_i \Leftrightarrow$$

$$K\lambda \sum_{i=0}^K P_i - \lambda \sum_{i=0}^K iP_i = \mu \sum_{i=1}^K P_i = \mu(1 - P_0) \Leftrightarrow$$

$$\Lambda = (K - \eta) \lambda = \mu(1 - P_0) \Leftrightarrow$$

$$\eta = K - \left(\frac{\mu(1 - P_0)}{\lambda}\right).$$

b. Durée moyenne de séjour d'un client dans le système

On utilise la formule de Little :

$$\bar{W} = \frac{\eta}{\Lambda} = \frac{K - \mu(1 - P_0) \lambda^{-1}}{\mu(1 - P_0)} = \frac{K}{\mu(1 - P_0)} - \lambda^{-1},$$

Alors

$$\Rightarrow \bar{W} = \frac{K}{\mu(1 - P_0)} - \frac{1}{\lambda}.$$

1.2.8 Etude du système M/M/s ($s > 1$)

Il s'agit d'un système d'attente qui comporte une seule file et s serveurs, les clients arrivent dans le système selon un processus poissonien de taux λ .

Les durées de service pour chaque serveur sont indépendantes et ont une même loi exponentielle de paramètre μ .

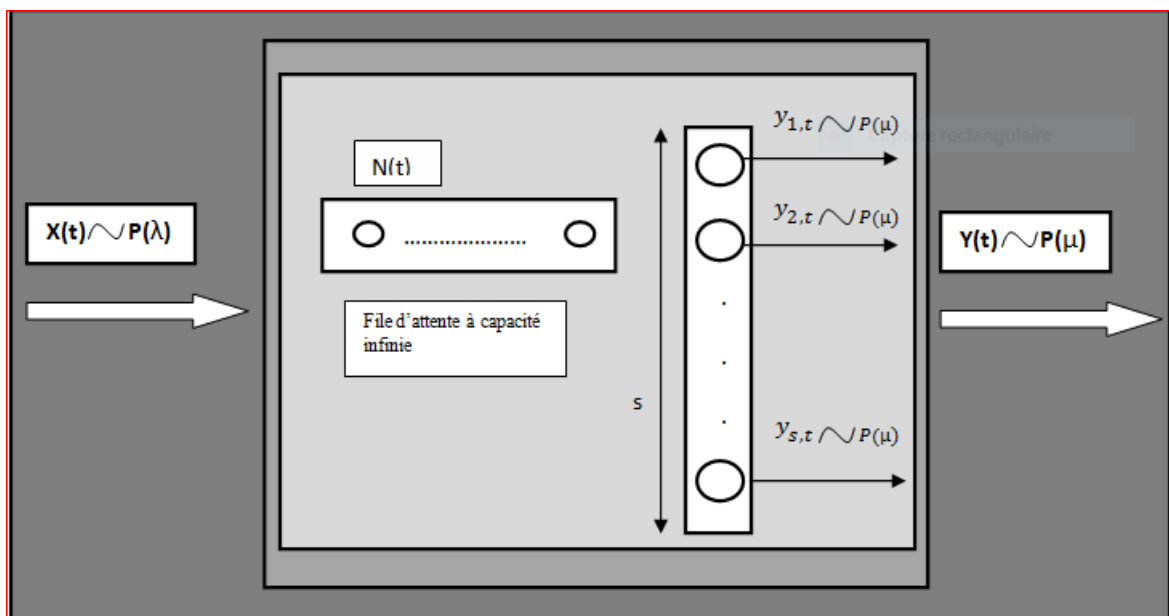


FIGURE 1.4 – Système d'attente (M/M/s)

- (X_t) est un processus de comptage, il est poissonien de taux λ , il compte le nombre de clients qui arrivent dans le système à la date t .
- (Y_t) est un processus de comptage, il est poissonien de taux $s\mu$, il compte le nombre de clients qui sortent du système à la date t .
- $(Y_{i,t})$ est un processus de comptage, il est poissonien de taux μ , il compte le nombre de clients servis par le serveur i , ($1 \leq i \leq s$), à la date t .
- (N_t) est un processus de comptage, il compte le nombre de clients en cours et en attente de service à la date t .

On s'intéresse dans notre étude au processus $(N_t)_{t \geq 0}$.

1.2.8.1 Etude du processus $(N_t)_{t \geq 0}$

Sous les hypothèses précédentes, $(N_t)_{t \geq 0}$ est un processus de naissance et de mort (PNM) de taux λ_n et μ_n à déterminer :

a. Taux de naissance λ_n

On a

$$\begin{aligned} \lambda_n dt + o(dt) &= P[N_{t+dt} = n+1 / N_t = n] = P[X_{t+dt} = X_t + 1] \\ &= P[X_{t+dt} - X_t = 1] = P[X_{dt} = 1] = \lambda dt + o(dt) \\ &\Rightarrow \lambda_n = \lambda, \forall n \geq 0. \end{aligned}$$

b. Taux de mort μ_n

On a

$$\mu_n dt + o(dt) = P[N_{t+dt} = n-1 / N_t = n] = A$$

On distingue deux cas

• $1 \leq n \leq s$, alors

$$\begin{aligned} A &= C_n^1 P[Y_{i_0, t+dt} = Y_{i_0, t} + 1] = \frac{n!}{1!(n-1)!} P[Y_{i_0, t+dt} - Y_{i_0, t} = 1] = nP[Y_{i_0, dt} = 1] \\ &= n(\mu dt + o(dt)) \\ &\Rightarrow \mu_n = n\mu, 1 \leq n \leq s. \end{aligned}$$

• $n > s$, alors

$$\begin{aligned} A &= C_s^1 P[Y_{i_0, t+dt} = Y_{i_0, t} + 1] = \frac{s!}{1!(s-1)!} P[Y_{i_0, t+dt} - Y_{i_0, t} = 1] = sP[Y_{i_0, dt} = 1] \\ &= s(\mu dt + o(dt)) \\ &\Rightarrow \mu_n = s\mu, n > s. \end{aligned}$$

D'où

$$\mu_n = (n \wedge s) \mu, n > 0.$$

1.2.8.2 Loi du système en régime permanent (stable)

On note $P_n = P[N_t = n]$, la loi du système en régime permanent quand il existe.

On a

$$P_n = \begin{cases} a_n \times P_0, a_n = \prod_{i=1}^n \frac{\lambda_{i-1}}{\mu_i} & n \geq 1, \\ \frac{1}{1 + \sum_{n \geq 1} a_n} & n = 0. \end{cases}$$

De là, on déduit que le régime stationnaire s'établit si $\sum_{n \geq 1} a_n < +\infty$ (condition de stabilité).

On distingue les cas suivants :

- Si $1 \leq n \leq s$

On a

$$\begin{aligned} P_n &= a_n \times P_0 = \left(\prod_{i=1}^n \frac{\lambda_{i-1}}{\mu_i} \right) \times P_0 = \frac{\lambda_0 \lambda_1 \dots \lambda_{n-1}}{\mu_1 \mu_2 \dots \mu_n} P_0 = \frac{\lambda^n}{n! \mu^n} P_0 \\ &\Rightarrow P_n = \frac{\rho^n}{n!} P_0, \rho = \frac{\lambda}{\mu}. \end{aligned}$$

- Si $n > s$

On a

$$\begin{aligned} P_n &= a_n \times P_0 = \left(\prod_{i=1}^n \frac{\lambda_{i-1}}{\mu_i} \right) \times P_0 = \left(\prod_{i=1}^s \frac{\lambda_{i-1}}{\mu_i} \prod_{i=s+1}^n \frac{\lambda_{i-1}}{\mu_i} \right) \times P_0 \\ &= \left(\frac{\lambda_0 \lambda_1 \dots \lambda_{s-1}}{\mu_1 \mu_2 \dots \mu_s} \times \frac{\lambda_s \lambda_{s+1} \dots \lambda_{n-1}}{\mu_{s+1} \mu_{s+2} \dots \mu_n} \right) P_0 = \left(\frac{\lambda \cdot \lambda \dots \lambda}{1 \mu 2 \mu \dots s \mu} \times \frac{\lambda \cdot \lambda \dots \lambda}{s \mu s \mu \dots s \mu} \right) P_0 = \frac{\lambda^n}{s! s^{n-s} \mu^n} P_0 \end{aligned}$$

On pose $\frac{\lambda}{\mu} = \rho$, alors

$$P_n = \frac{\rho^n}{s! s^{n-s}} P_0, n > s.$$

- Si $n = 0$

On a

$$P_0 = \frac{1}{1 + \sum_{n \geq 1} a_n}.$$

De plus

$$\sum_{n \geq 1} a_n = \sum_{n=1}^s a_n + \sum_{n \geq s+1} a_n = \sum_{n=1}^s \frac{\rho^n}{n!} + \sum_{n \geq s+1} \frac{\rho^n}{s! s^{n-s}} = \sum_{n=1}^s \frac{\rho^n}{n!} + \frac{\rho^s}{s!} \sum_{n-s \geq 1} \frac{\rho^{n-s}}{s^{n-s}}$$

$$= \sum_{n=1}^s \frac{\rho^n}{n!} + \frac{\rho^s}{s!} \sum_{k \geq 1} \left(\frac{\rho}{s}\right)^k, \text{ avec } k = n - s.$$

On pose $\frac{\rho}{s} = \tilde{\rho}$, alors

$$\begin{aligned} \sum_{n \geq 1} a_n &= \sum_{n=1}^s \frac{\rho^n}{n!} + \frac{\rho^s}{s!} \sum_{k \geq 1} (\tilde{\rho})^k = \sum_{n=1}^s \frac{\rho^n}{n!} + \frac{\rho^s}{s!} \left(\sum_{k \geq 0} (\tilde{\rho})^k - 1 \right) = \sum_{n=1}^s \frac{\rho^n}{n!} + \frac{\rho^s}{s!} \frac{\tilde{\rho}}{1 - \tilde{\rho}} \\ &= \sum_{n=1}^s \frac{\rho^n}{n!} + \frac{\rho^s}{s!} \frac{\rho}{s - \rho}. \end{aligned}$$

Si $\frac{\rho}{s} < 1$ autrement dit si $\lambda < s\mu$,

$$\begin{aligned} \Rightarrow \sum_{n \geq 1} a_n &= \sum_{n=1}^s \frac{\rho^n}{n!} + \frac{\rho^s}{s!} \frac{\rho}{s - \rho}. \\ \Rightarrow P_0 &= \frac{1}{\sum_{n=0}^s \frac{\rho^n}{n!} + \frac{\rho^s}{s!} \frac{\rho}{s - \rho}}. \end{aligned}$$

Finalement si $\lambda < s\mu$, on obtient

Pour $0 \leq n \leq s$

$$P_n = \frac{\frac{\rho^n}{n!}}{\sum_{n=0}^s \frac{\rho^n}{n!} + \frac{\rho^s}{s!} \frac{\rho}{s - \rho}}.$$

Pour $n > s$

$$P_n = \frac{\frac{\rho^n}{s! s^{n-s}}}{\sum_{n=0}^s \frac{\rho^n}{n!} + \frac{\rho^s}{s!} \frac{\rho}{s - \rho}}.$$

1.2.8.3 Performances de la file M/M/s, liées aux clients

a. Probabilité de saturation

On a

$$\pi(\rho, s) = P(N_t \geq s) = \sum_{n \geq s} P_n = \sum_{n \geq s} \frac{\rho^n}{s! s^{n-s}} P_0,$$

$$\text{avec } P_0 = \frac{1}{\sum_{n=0}^s \frac{\rho^n}{n!} + \frac{\rho^s}{s!} \frac{\rho}{s - \rho}}.$$

$$\pi(\rho, s) = \frac{\rho^s}{s!} P_0 \sum_{n \geq s} \left(\frac{\rho}{s}\right)^{n-s}.$$

On pose $k = n - s$, $\bar{\rho} = \frac{\rho}{s}$ et comme $\lambda < s\mu$ donc $\bar{\rho} < 1$ et on obtient

$$\begin{aligned} \pi(\rho, s) &= \frac{\rho^s}{s!} P_0 \sum_{k \geq 0} (\bar{\rho})^k = \frac{\rho^s}{s!} \frac{1}{1 - \bar{\rho}} P_0 \\ &= \frac{\rho^s}{s!} \frac{s}{s - \rho} P_0 = \frac{\rho^s}{(s-1)!(s-\rho)} P_0. \end{aligned}$$

b. Nombre moyen de clients dans la file

Soit M_t le nombre de clients dans la file,

$$\begin{aligned}\eta &= \mathbb{E}(M_t) = \sum_{k \geq 0} k P[M_t = k] = \sum_{k \geq 0} k P[N_t = k + s] \\ &= \sum_{k \geq 0} k P_{s+k} = \sum_{k \geq 0} k \frac{\rho^{s+k}}{s! s^k} P_0 = \sum_{k \geq 0} k \frac{\rho^s}{s!} \left(\frac{\rho}{s}\right)^k P_0.\end{aligned}$$

On pose $\bar{\rho} = \frac{\rho}{s}$,

$$\begin{aligned}\eta_q &= \frac{\rho \rho^s}{s s!} P_0 \sum_{k \geq 0} k (\bar{\rho})^{k-1} = \frac{\rho \rho^s}{s s!} P_0 \sum_{k \geq 0} (\bar{\rho}^k)' \\ &= \frac{\rho \rho^s}{s s!} P_0 \left(\frac{1}{1 - \bar{\rho}}\right)' = \frac{\rho \rho^s}{s s!} P_0 \frac{1}{(1 - \bar{\rho})^2} \\ &= \frac{\rho \rho^s}{s s!} P_0 \frac{s^2}{(s - \rho)^2} = \frac{\rho^s}{s!} \frac{s}{(s - \rho)} P_0 \frac{\rho}{(s - \rho)} \\ \Rightarrow \eta_q &= \pi(\rho, s) \frac{\rho}{(s - \rho)} = \pi(\rho, s) \frac{\lambda}{(s\mu - \lambda)}.\end{aligned}$$

c. Durée moyenne d'attente d'un client dans la file

On utilise la formule de Little : $\bar{W}_q = \frac{\eta_q}{\Lambda}$ avec $\Lambda = \sum_{i \geq 0} \lambda_i P_i = \lambda$.

$$\begin{aligned}\bar{W}_q &= \frac{\eta_q}{\Lambda} = \frac{\eta_q}{\lambda} = \pi(\rho, s) \frac{\rho}{\lambda(s - \rho)}, \\ \Rightarrow \bar{W}_q &= \pi(\rho, s) \frac{1}{\mu(s - \rho)} = \pi(\rho, s) \frac{1}{(s\mu - \lambda)}.\end{aligned}$$

Remarque

S'il y a vraiment attente, c'est-à-dire le système est saturé, donc $\pi(\rho, s) = 1$ et ainsi $\bar{W}_q = \frac{1}{(s\mu - \lambda)}$. En fait, on a W_q qui est une variable aléatoire suivant une loi exponentielle $\mathcal{E}(\theta)$ avec $\theta = s\mu - \lambda$ qui est positif car $\lambda < s\mu$.

1.3 Conclusion

Dans ce chapitre, nous avons présenté deux notions mathématiques très utilisées dans la modélisation des systèmes et phénomènes évoluant dans le temps. En premier lieu on a vu les processus markoviens qui sont un outil incontournable dans l'analyse des différents systèmes dynamiques. Ensuite les systèmes d'attente markoviens dont on a traité certains modèles en donnant leurs mesures de performances. Ces deux théories nous permettront de modéliser et d'étudier le problème considéré dans le prochain chapitre.

Chapitre 2

Modélisation du télétrafic d'un centre d'appels téléphoniques

2.1 Introduction

Lorsqu'une personne compose un numéro de téléphone d'un centre d'appels pour solliciter un service quelconque, la première question qui lui passe par l'esprit est : "est ce que je vais attendre beaucoup avant d'être servi si jamais j'arrive à joindre le centre d'appels?". Le meilleur scénario serait d'avoir directement une personne au bout de la ligne au lieu d'un répondeur automatique. En pratique, aucun centre d'appels n'est capable de satisfaire les clients de la sorte. C'est de là qu'un système de file d'attente s'est imposé pour créer un certain équilibre entre des gestionnaires de centres qui veulent fournir un bon service et des clients qui veulent être servis sans avoir à attendre beaucoup de l'autre côté.

Alors dans ce chapitre, nous allons développer des modèles mathématiques nécessaires à l'étude et l'analyse des performances du système d'attente au niveau d'un centre d'appels téléphoniques. La modélisation est une représentation mathématique et logique du comportement d'un système réel dans un contexte donné. L'étude de ce modèle va se prendre sur une série d'hypothèses concernant le fonctionnement du système et qui sont exprimées à l'aide des relations mathématiques.

2.2 Description et étude du problème

Considérons un centre d'appels téléphoniques disposant de s serveurs (conseillers, par exemple d'une banque) de clientèle formant une classe de clients. On suppose que l'arrivée des clients est poissonnienne et que les temps de service des serveurs suivent des lois exponentielles indépendantes. Ce centre d'appels peut donc être modélisé par une file d'attente $M/M/s$ de taux d'arrivée λ et de taux de sortie μ .

En pratique, les clients qui veulent accéder au centre d'appels peuvent abandonner en mettant fin à leurs appels par leurs propres initiatives après avoir attendu longtemps. Afin d'éviter de très longues attentes, certains centres d'appels limitent la taille de leur file d'attente à K clients.

Ainsi s'il y a déjà beaucoup de clients en attente lors de l'arrivée d'un nouveau client, alors celui-ci va être déconnecté immédiatement au lieu d'attendre le service de toute la file. Ainsi le système est modélisé par une file (L_1, L_2, s, K) . Comme il peut ne pas y avoir de file d'attente, c'est-à-dire au moment d'occupation de tous les serveurs, tout nouvel appel sera rejeté, alors le système est modélisé par une file d'attente (L_1, L_2, s, s) .

Notre but est d'optimiser les performances par rapport aux paramètres du système ainsi minimiser le risque de saturation dans le centre d'appels, ce qui engendre la perte de la clientèle et la baisse du profit.

2.3 Position du problème

Il s'agit d'un problème de modélisation dans un centre d'appels téléphoniques de la banque (par exemple BNA), où des appels arrivent d'une manière aléatoire afin de les traiter et régler les problèmes des clients.

Ce modèle peut être déterminé par les paramètres A, B, C, D et E suivants :

A : représente le flux d'arrivée des appels au centre d'appels téléphoniques, il est modélisé par une suite de variable aléatoire $(t_n)_{n \geq 0}$ et définie sur un espace de probabilité (Ω, F, P) vérifiant :

1. $t_0 = 0$,
2. $t_{n-1} \leq t_n, \forall n \geq 1$,

où $t_n, n \geq 1$ est l'instant d'arrivée du $n^{\text{ième}}$ appel. On s'intéresse aux durées entre deux arrivées successives (inter-arrivées) $(T_n = t_n - t_{n-1})_{n \geq 1}$, qu'on suppose exponentielles de paramètre λ .

$(T_n)_{n \geq 1}$ est une suite de variables aléatoires indépendantes et identiquement distribuées de loi exponentielle λ . Ce qui est équivalent à dire que le flux d'arrivées est poissonien de taux λ .

B : Représente la durée d'occupation d'un serveur, elles sont aléatoires indépendantes et identiquement distribuées de loi exponentielle de paramètre μ .

C : Le nombre de serveurs, qui sont parallèles et indépendants les uns des autres, chaque appel entrant est dirigé vers un serveur.

D : La capacité de mise en attente des appels entrants et des appels en cours.

E : La nature du système, dans notre cas, tous les clients ont un accès au service du centre, autrement dit le système est ouvert.

2.4 Détermination des lois

2.4.1 Loi des arrivées

Pour délimiter la loi des arrivées des appels entrants dans un centre d'appels téléphoniques, on utilise les temps d'inter-arrivées ou bien le nombre d'appels dans un intervalle donné.

Soit alors la suite $t_0 = 0 < t_1 < t_2 < \dots < t_n$, les instants d'arrivées des appels au centre d'appels téléphoniques et K la capacité maximale du système (nombre de clients accédants au centre).

En général, on observe les variables aléatoires (temps d'inter-arrivées) $T_n = t_n - t_{n-1}$, $n = 1, 2, \dots, K$ supposées indépendantes et identiquement distribuées et suivant une même loi exponentielle de densité $a(t) = \lambda e^{-\lambda t}$, avec λ étant l'inverse du temps moyen entre deux appels successifs, on dit alors que le système est markovien (poissonien).

Nous nous intéressons au nombre $N(t)$, ($t \geq 0$) d'appels entrants au cours d'un intervalle de temps $[0, t]$, $N(t) = \sup \{n, t_n \leq t\} = \sum_{i \geq 1} t_{i\{t_i \leq t\}}$. Puisque le processus $\{N(t), t \geq 0\}$ est un processus de comptage avec des temps d'inter-arrivées T_n de loi exponentielle de paramètre $\lambda > 0$, alors il est poissonien de taux λ .

2.4.2 Loi de la durée de service

Pour modéliser les durées des appels, on désigne par D_i la durée pendant laquelle le serveur i ($i = 1, 2, \dots, n$ et $n \leq s$) demeure occupé par un appel. Puisque on a supposé que les serveurs fournissent le même service, les variables aléatoires D_i sont indépendantes et identiquement distribuées (*i.i.d.*), ainsi nous modélisons les temps de service par la loi exponentielle de paramètre $\mu \geq 0$ (qui est markovienne en raison de sa propriété d'absence de mémoire).

2.5 Etude de la stabilité du système

La stabilité consiste à étudier les conditions d'existence d'un régime stationnaire (ou permanent) qui correspond au non-engorgement du système. Si le système n'est pas stable, il y'a congestion et il est inutile de procéder à une évaluation des performances dans ces conditions.

Dans notre modèle, il n'y'a pas de conditions de stationnarité, car la capacité du système est finie, c'est-à-dire que le système finit par se stabiliser avec le temps.

2.6 Etude du système M/M/s/K

C'est un système d'attente ouvert, où le nombre de clients ayant accès au service est limité à K clients, qui représente sa capacité et il comporte s serveurs.

Les arrivées sont supposées poissonniennes de taux λ et les durées de service qui sont les durées d'appels sont supposées indépendantes, de même loi exponentielle de paramètre μ .

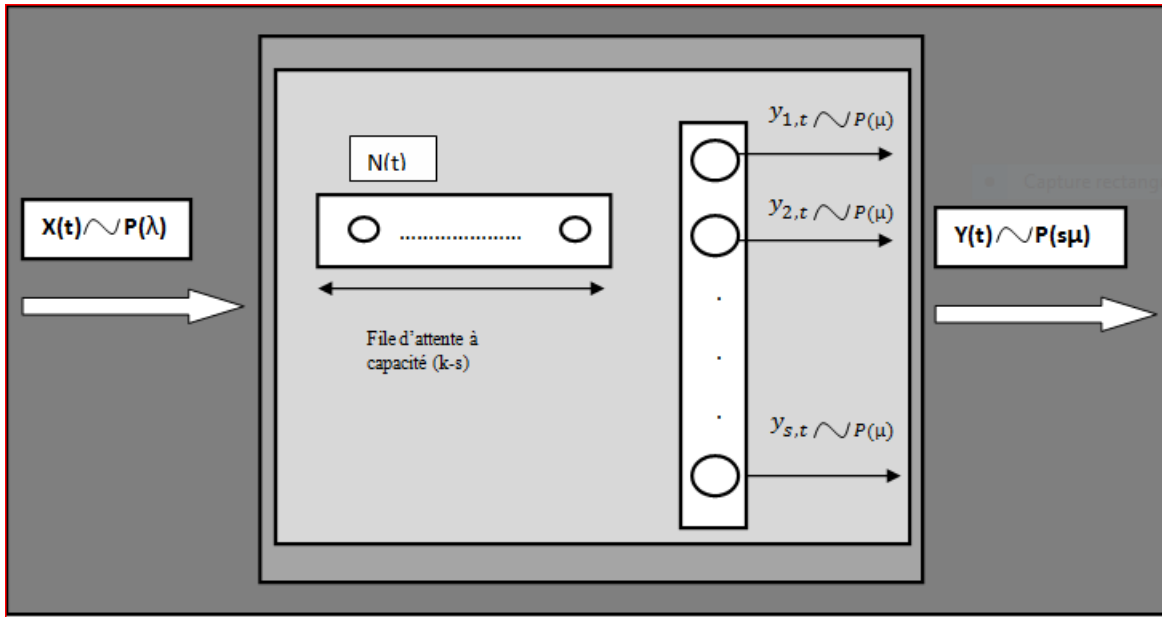


FIGURE 2.1 – Système d'attente M/M/s/K

- (X_t) est un processus de comptage, il est poissonien de taux λ , il compte le nombre d'appels qui arrivent dans le centre d'appels téléphoniques à la date t .
- (Y_t) est un processus de comptage, il est poissonien de taux $s\mu$, il compte le nombre d'appels terminés à la date t .
- $(Y_{i,t})$ est un processus de comptage, il est poissonien de taux μ , il compte le nombre d'appels passés par le serveur i , ($1 \leq i \leq s$), à la date t .
- (N_t) est un processus de comptage, il compte le nombre d'appels en cours et en attente de connexion à la date t .

On s'intéresse alors au processus $(N_t)_{t \geq 0}$.

2.6.1 Etude du processus $(N_t)_{t \geq 0}$

$(N_t)_{t \geq 0}$ est un processus de naissance et de mort (PNM) défini sur l'ensemble fini $\{0, 1, 2, \dots, K\}$, de taux λ_n et μ_n à déterminer :

a) Taux de naissance λ_n :

On a

$$\lambda_n dt + o(dt) = P(N_{t+dt} = n + 1 / N_t = n) = \begin{cases} P[X_{t+dt} = X_t + 1] & 0 \leq n \leq K - 1 \\ 0 & n \geq K \end{cases}$$

$$= \begin{cases} P[X_{dt} = 1] = \lambda dt + o(dt) & 0 \leq n \leq K - 1 \\ 0 & n \geq K \end{cases}$$

$$\Rightarrow \lambda_n = \begin{cases} \lambda & 0 \leq n \leq K-1, \\ 0 & \text{sinon.} \end{cases}$$

b) Taux de mort μ_n :

On a

$$\mu_n dt + o(dt) = P(N_{t+dt} = n-1 / N_t = n)$$

Si $1 \leq n \leq s$

$$\begin{aligned} P(N_{t+dt} = n-1 / N_t = n) &= C_n^1 P(Y_{i_0, t+dt} = Y_{i_0, t} + 1) = C_n^1 P(Y_{i_0, dt} = 1) \\ &= \frac{n!}{1!(n-1)!} (\mu dt + o(dt)) = n\mu dt + o(dt). \end{aligned}$$

Si $s \leq n \leq K$

$$\begin{aligned} P(N_{t+dt} = n-1 / N_t = n) &= C_s^1 P(Y_{i_0, t+dt} = Y_{i_0, t} + 1) = C_s^1 P[Y_{i_0, dt} = 1] \\ &= \frac{s!}{1!(s-1)!} (\mu dt + o(dt)) = s\mu dt + o(dt). \end{aligned}$$

D'où

$$\begin{aligned} \Rightarrow \mu_n &= \begin{cases} n\mu & 1 \leq n \leq s, \\ s\mu & s \leq n \leq K, \\ 0 & \text{sinon,} \end{cases} \\ &= \begin{cases} n \wedge s, & 1 \leq n \leq K, \\ 0 & \text{sinon.} \end{cases} \end{aligned}$$

2.6.2 Loi du système en régime permanent (stable)

On note

$$P_n = P[N_t = n], 0 \leq n \leq K.$$

$$\text{On a } P_n = \begin{cases} a_n P_0 & 1 \leq n \leq K, \\ P_0 & n = 0, \end{cases}$$

$$\text{avec } a_n = \prod_{i=1}^n \left(\frac{\lambda_{i-1}}{\mu_i} \right) \text{ et } P_0 = \frac{1}{1 + \sum_{n=1}^K a_n}.$$

Déterminons (a_n) :

Si $1 \leq n \leq s$

$$\begin{aligned} a_n &= \prod_{i=1}^n \left(\frac{\lambda_{i-1}}{\mu_i} \right) = \frac{\lambda_0 \lambda_1 \lambda_2 \dots \lambda_{n-1}}{\mu_1 \mu_2 \mu_3 \dots \mu_n} = \frac{\lambda \lambda \lambda \dots \lambda}{1 \mu 2 \mu 3 \mu \dots n \mu} \\ &= \frac{\lambda^n}{n! \mu^n} = \frac{1}{n!} \left(\frac{\lambda}{\mu} \right)^n \\ &= \frac{\rho^n}{n!}, \text{ avec } \rho = \frac{\lambda}{\mu}. \end{aligned}$$

Si $s < n \leq K$

$$\begin{aligned} a_n &= \prod_{i=1}^n \left(\frac{\lambda_{i-1}}{\mu_i} \right) = \frac{\lambda_0 \lambda_1 \lambda_2 \dots \lambda_{n-1}}{\mu_1 \mu_2 \mu_3 \dots \mu_s \mu_{s+1} \dots \mu_n} = \frac{\lambda \lambda \dots \lambda}{1 \mu 2 \mu \dots s \mu s \mu_{s+1} \dots s \mu_n} \\ &= \frac{\lambda^n}{s! \mu^s s^{n-s} \mu^{n-s}} = \frac{\lambda^n}{s! s^{n-s} \mu^n} = \frac{\rho^n}{s! s^{n-s}}. \end{aligned}$$

Déterminons P_0 :

$$\begin{aligned} P_0 &= \frac{1}{1 + \sum_{n=1}^K a_n} = \frac{1}{1 + \sum_{n=1}^s \frac{\rho^n}{n!} + \sum_{n=s+1}^K \frac{\rho^n}{s! s^{n-s}}} \\ &\Rightarrow P_0 = \frac{1}{\sum_{n=0}^s \frac{\rho^n}{n!} + \sum_{n=s+1}^K \frac{\rho^n}{s! s^{n-s}}}. \end{aligned}$$

Finalement, on obtient

$$P_n = \begin{cases} \frac{\rho^n}{n!} P_0 & 1 \leq n \leq s, \\ \frac{\rho^n}{s! s^{n-s}} P_0 & s < n \leq K, \\ \frac{1}{\sum_{n=0}^s \frac{\rho^n}{n!} + \sum_{n=s+1}^K \frac{\rho^n}{s! s^{n-s}}} & n = 0. \end{cases}$$

2.6.3 Performances de la file M/M/s/K

On note par

- η : nombre moyen d'appels en cours et en attente de connexion à l'instant t .
- η_q : nombre moyen d'appels en attente de connexion à l'instant t .
- W : durée moyenne d'un appel dans le système, et \overline{W} sa moyenne.
- W_q : durée moyenne d'un appel avant connexion, et \overline{W}_q sa moyenne.

2.6.3.1 Nombre moyen d'appels en cours et en attente de connexion

On a

$$\begin{aligned}
 \eta = E(N_t) &= \sum_{n=1}^K nP_n = \sum_{n=1}^s nP_n + \sum_{n=s+1}^K nP_n \\
 &= \sum_{n=1}^s n \frac{\rho^n}{n!} P_0 + \sum_{n=s+1}^K n \frac{\rho^n}{s!s^{n-s}} P_0 \\
 &= \sum_{n=1}^s \frac{\rho^n}{(n-1)!} P_0 + \sum_{n=s+1}^K n \frac{\rho^n}{(s-1)!s^{n-s}} P_0 \\
 &= \sum_{n=1}^s \frac{\rho^n}{(n-1)!} P_0 + \sum_{n=s+1}^K n \frac{s^{s-1}}{(s-1)!} \frac{\rho^n}{s^n} P_0 \\
 &= P_0 \left[\sum_{n=1}^s \frac{\rho^n}{(n-1)!} + \frac{s^{s-1}}{(s-1)!} \sum_{n=s+1}^K n (\bar{\rho})^n \right].
 \end{aligned}$$

avec $\bar{\rho} = \frac{\rho}{s}$.

2.6.3.2 Nombre moyen d'appels en attente de connexion

Soit

$$\begin{aligned}
 \eta_q &= \sum_{h=0}^{K-s} hP[N_t = s+h] = \sum_{h=1}^{K-s} hP_{s+h} \\
 &= \sum_{h=1}^{K-s} h \frac{\rho^{s+h}}{s!s^h} P_0 = \frac{P_0 \rho^s}{s!} \sum_{h=1}^{K-s} h \left(\frac{\rho}{s}\right)^h \\
 \Rightarrow \eta_q &= \frac{P_0 \rho^s}{s!} \sum_{h=1}^{K-s} h (\bar{\rho})^h, \text{ avec } \bar{\rho} = \frac{\rho}{s}.
 \end{aligned}$$

2.6.3.3 Durée moyenne d'un appel dans le système

On utilise la formule de Little : $\bar{W} = \frac{\eta}{\Lambda}$ avec $\Lambda = \sum_{i=0}^K \lambda_i P_i$
 $= \sum_{i=0}^{K-1} \lambda P_i = \lambda(1 - P_K)$, tel que $P_K = \frac{\rho^K}{s!s^{K-s}} P_0$, donc on aura

$$\bar{W} = \frac{\eta}{\Lambda} = \frac{\eta}{\lambda(1 - P_K)} = \frac{\eta}{\lambda \left(1 - \frac{\rho^K}{s!s^{K-s}} P_0\right)}.$$

2.6.3.4 Durée moyenne d'un appel avant connexion

On utilise la formule de Little : $\overline{W}_q = \frac{\eta_q}{\Lambda}$ avec $\Lambda = \sum_{i=0}^K \lambda_i P_i$
 $= \sum_{i=0}^{K-1} \lambda P_i = \lambda(1 - P_K)$ et $P_K = \frac{\rho^K}{s!s^{K-s}} P_0$.

Ainsi

$$\overline{W}_q = \frac{\eta_q}{\Lambda} = \frac{\eta_q}{\lambda(1 - P_K)} = \frac{\eta_q}{\lambda(1 - \frac{\rho^K}{s!s^{K-s}} P_0)}.$$

2.6.4 Probabilité de saturation

Soit

$$\pi(\rho, s, K) = P(N_t \geq s) = \sum_{n=s}^K P(N_t = n) = \sum_{n=s}^K P_n = \sum_{n=s}^K \frac{\rho^n}{s!s^{n-s}} P_0$$

$$= \frac{\rho^s}{s!} P_0 \sum_{n=s}^K \frac{\rho^{n-s}}{s^{n-s}} = \frac{\rho^s}{s!} P_0 \sum_{n=s=0}^K \left(\frac{\rho}{s}\right)^{n-s}.$$

On pose $h = n - s$, $\bar{\rho} = \frac{\rho}{s}$ et on obtient

$$\pi(\rho, s, K) = \frac{\rho^s}{s!} P_0 \sum_{h=0}^{K-s} (\bar{\rho})^h.$$

2.7 Etude de la file M/M/s/s

Il s'agit d'un système ouvert à des clients qui seront servis par s serveurs sans file d'attente, c'est-à-dire qu'un client est rejeté (n'accède pas au service) si tous les serveurs sont occupés. Les arrivées et les durées d'appels sont supposées respectivement poissonniennes de taux λ et indépendantes de même loi exponentielle de paramètre μ .

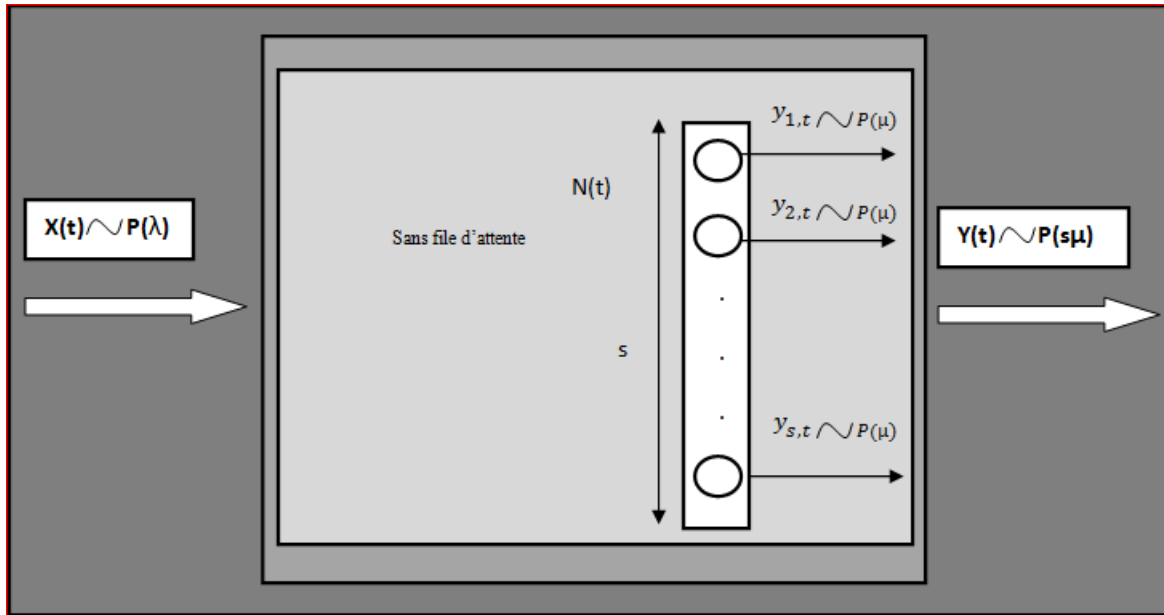


FIGURE 2.2 – Système d'attente $(M/M/s/s)$

- (X_t) est un processus de comptage, il est poissonien de taux λ , il compte le nombre d'appels qui arrivent dans le centre d'appels à la date t .
- (Y_t) est un processus de comptage, il est poissonien de taux $s\mu$, il compte le nombre d'appels terminés à la date t .
- $(Y_{i,t})$ est un processus de comptage, il est poissonien de taux μ , il compte le nombre d'appels passés par le serveur i , $1 \leq i \leq s$, à la date t .
- (N_t) est un processus de comptage, il compte le nombre d'appels en cours à la date t .

On s'intéresse au processus $(N_t)_{t \geq 0}$.

2.7.1 Etude du processus $(N_t)_{t \geq 0}$

$(N_t)_{t \geq 0}$ est un processus de naissance et de mort (PNM) défini sur l'ensemble fini $\{1, 2, \dots, s\}$, de taux λ_n et μ_n à déterminer :

a) Taux de naissance λ_n :

On a

$$\lambda_n dt + o(dt) = P(N_{t+dt} = n + 1 / N_t = n) = P(X_{t+dt} = X_t + 1) = \lambda dt + o(dt)$$

$$\Rightarrow \lambda_n = \begin{cases} \lambda & 0 \leq n \leq s - 1, \\ 0 & n \geq s. \end{cases}$$

b) Taux de mort μ_n :

On a

$$\begin{aligned}\mu_n dt + o(dt) &= P(N_{t+dt} = n - 1 / N_t = n) = C_n^1 P(Y_{i_0, t+dt} = Y_{i_0, t} + 1) \\ &= \frac{n!}{1!(n-1)!} \mu dt + o(dt) = n\mu dt + o(dt)\end{aligned}$$

$$\Rightarrow \mu_n = \begin{cases} n\mu & 1 \leq n \leq s, \\ 0 & \text{sinon.} \end{cases}$$

2.7.2 Loi du système en régime permanent (stable)

On note $P_n = P[N_t = n]$, $0 \leq n \leq s$.

On a

$$P_n = \begin{cases} a_n P_0 & 1 \leq n \leq s, \\ P_0 & n = 0. \end{cases}$$

Avec $a_n = \prod_{i=1}^n \left(\frac{\lambda_{i-1}}{\mu_i} \right)$ et $P_0 = \frac{1}{1 + \sum_{n=1}^s a_n}$.

Déterminons (a_n) :

$$\begin{aligned}a_n &= \prod_{i=1}^n \left(\frac{\lambda_{i-1}}{\mu_i} \right) = \frac{\lambda_0 \lambda_1 \lambda_2 \dots \lambda_{n-1}}{\mu_1 \mu_2 \mu_3 \dots \mu_n} \\ &= \frac{\lambda \cdot \lambda \cdot \lambda \dots \lambda}{1\mu \cdot 2\mu \cdot 3\mu \dots n\mu} = \frac{\lambda^n}{n! \mu^n} \\ &= \frac{1}{n!} \rho^n, \text{ avec } \rho = \frac{\lambda}{\mu}.\end{aligned}$$

Déterminons P_0 :

On a

$$P_0 = \frac{1}{1 + \sum_{n=1}^s a_n} = \frac{1}{1 + \sum_{n=1}^s \frac{\rho^n}{n!}} \Rightarrow P_0 = \frac{1}{\sum_{n=0}^s \frac{\rho^n}{n!}}.$$

D'où

$$P_n = \begin{cases} \frac{\rho^n}{n!} P_0 & 1 \leq n \leq s, \\ \frac{1}{\sum_{i=0}^s \frac{\rho^i}{i!}} & n = 0. \end{cases}$$

Donc la loi du système est $P_n = \frac{\rho^n}{n!} P_0$ si $0 \leq n \leq s$ et $P_0 = \frac{1}{\sum_{n=0}^s \frac{\rho^n}{n!}}$.

$$P_n = \begin{cases} \frac{\rho^n}{n!} \frac{1}{\sum_{i=0}^s \frac{\rho^i}{i!}} & 0 \leq n \leq s, \\ 0 & \text{sinon.} \end{cases}$$

2.7.3 Performances du système ($M/M/s/s$)

On note par

- η : nombre moyen d'appels en cours et en attente de connexion à l'instant t .
- W : durée moyenne d'un appel dans le système, et \bar{W} sa moyenne.

2.7.3.1 Nombre moyen d'appels en cours et en attente de connexion

À l'équilibre (régime stationnaire), le taux d'entrée global est égal au taux de sortie global donc

$$\begin{aligned}\Lambda = \sum_{i=0}^s \lambda_i P_i &= \sum_{i=0}^s \mu_i P_i \Leftrightarrow \sum_{i=0}^{s-1} \lambda P_i = \sum_{i=0}^s i \mu P_i \\ \Leftrightarrow \lambda \sum_{i=0}^{s-1} P_i &= \mu \sum_{i=0}^s i P_i \Leftrightarrow \lambda (1 - P_s) = \lambda \left(1 - \frac{\rho^s}{s!} P_0\right) = \mu \eta \\ \Rightarrow \eta &= \frac{\lambda}{\mu} \left(1 - \frac{\rho^s}{s!} P_0\right).\end{aligned}$$

2.7.3.2 Durée moyenne d'un appel dans le système

Soit W le temps d'attente dans le système.

On utilise la formule de Little :

$$\begin{aligned}\bar{W} = E(W) &= \frac{\eta}{\Lambda} \text{ avec } \Lambda = \sum_{i=0}^s \lambda_i P_i = \lambda (1 - P_s) \\ \Rightarrow \bar{W} &= \frac{\eta}{\Lambda} = \frac{\lambda (1 - P_s)}{\mu \cdot \lambda (1 - P_s)} = \frac{1}{\mu}.\end{aligned}$$

Remarque

Ceci est normal puisqu'il n'y a pas d'attente dans le système, donc il n'y a que le temps de service $\frac{1}{\mu}$. Et pour le nombre moyen d'appels en attente de connexion η_q , ainsi que le temps moyen d'attente d'un appel avant connexion \bar{W}_q , ils sont nuls.

2.7.4 Probabilité de saturation

On a

$$\pi(\rho, s) = P(N_t \geq s) = P(N_t = s).$$

D'où

$$\pi(\rho, s) = P(N_t = s) = \frac{\frac{\rho^s}{s!}}{\sum_{n=0}^s \frac{\rho^n}{n!}}.$$

2.8 Conclusion

Dans ce chapitre, on s'est intéressé à la modélisation de deux types de systèmes de files d'attente représentant le problème d'attente dans un centre d'appels téléphoniques. L'un représente le cas où le système est sans attente ($M/M/s/s$) et l'autre avec attente ($M/M/s/K$). On a commencé en premier lieu par la modélisation de ces deux modèles en utilisant des distributions de probabilité, par la suite on s'est mis au calcul des différentes mesures de performance.

Ces deux modèles vont être simulés dans le chapitre suivant afin de bien comprendre leurs évolutions en fonction des différents paramètres d'entrée λ , μ et s .

Chapitre 3

Simulation des mesures de performances d'un centre d'appels téléphoniques

3.1 Introduction

La réalisation des projets dans la vie pratique est tellement compliquée que les outils mathématiques ne suffisent pas à résoudre facilement les problèmes.

Ainsi, après la modélisation analytique du système à l'aide de la théorie des files d'attente, on utilise les logiciels de simulation qui permettent de simuler le comportement dynamique d'un système qui est représenté par un modèle mathématique ainsi que l'évaluation de ses performances. On simule alors une file d'attente en utilisant des distributions de probabilité pour générer au hasard les arrivées et les services. En évaluant le comportement de la file et le temps d'attente, nous pouvons déduire la longueur moyenne de la file et le temps moyen d'attente.

Les modèles de simulation peuvent être programmés dans plusieurs langage, on s'intéresse dans notre cas d'étude à la simulation avec le logiciel de calcul scientifique MATLAB.

3.2 Simulation du processus des arrivées

3.2.1 Arrivées poissonniennes

On représente le programme des temps des arrivées :

```

1  function x= poisson(s,e,lamda)
2  -   K=15000;                               %s est l'instant de début
3  -   y=(expo(lamda,K));                     % e est l'instant de fin
4  -   i=1;                                    % y est la fonction exponentielle
5  -   x(i)=s+y(1);
6  -   while x(i)<e
7  -       x(i+1)=x(i)+y(i+1);
8  -       i=i+1;
9  -       plot(x);
10 -      xlabel('lamda*t');
11 -      ylabel('temps t');
12 -   end
13 -   x(i)=e;
14 -   end
15

```

FIGURE 3.1 – Programme des temps des arrivées

Pour observer l'évolution des arrivées d'appels au centre d'appels téléphoniques, on donne comme exemple les valeurs suivantes :

$\lambda = 5$ appels par minute.

$s = 0$ minute, l'instant de début.

$e = 420$ minutes, l'instant de fin.

3.2.2 Graphe des temps des arrivées

La simulation donne le résultat suivant :

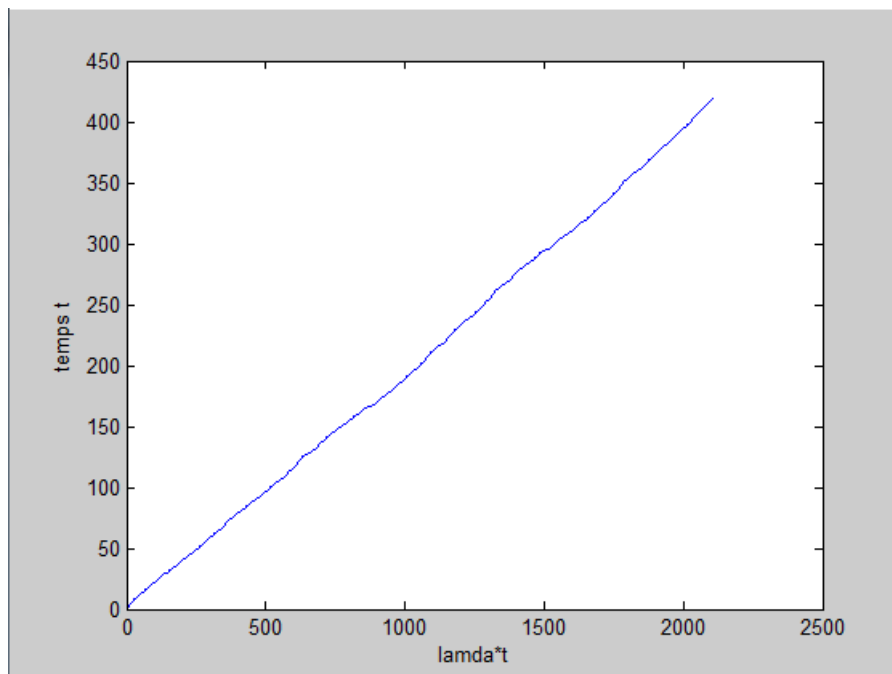


FIGURE 3.2 – Evolution des temps des arrivées

Le graphe ci-dessus représente la variation de la loi des arrivées d'appels (arrivées poissonniennes) pendant un intervalle de temps $[0, 420 \text{ min}]$.

Où on remarque l'évolution du graphe d'une façon croissante.

3.3 Simulation des lois et des performances des files d'attente considérées en fonction des paramètres d'entrée

3.3.1 File d'attente $M/M/s/K$

3.3.1.1 Simulation de la loi de $N(t)$

On considère la loi de $(N_t)_{t \geq 0}$ en fonction du taux des arrivées d'appels au centre d'appels téléphoniques λ .

On suppose que le nombre maximal de clients K est égal à 80, la moyenne des durées d'appels $d = \frac{1}{\mu}$ est égal à $\frac{1}{10}$ et le nombre de serveurs s égal à 15.

Le programme de la loi de $(N_t)_{t \geq 0}$ est le suivant :

```

1  function loi= pntsk(n, z)
2  mu=10; K=80; s=15;
3  if (n<=K)
4      r=((s^s)/fact(s));
5  for lamda=0:z;
6      som1=0;
7      som2=0;
8      loi1=((lamda/mu)^n)/fact(n);
9      loi2= r*((lamda/(s*mu))^n);
10 for i=0:s
11     som1=som1+((lamda/mu)^i)/fact(i);
12 end
13 for j=s+1:K
14     som2=som2+(r*((lamda/(s*mu))^j));
15 end
16 if (n<=s)
17     pn(1,lamda+1)=loi1/(som1+som2);
18     loi=pn;
19 else if (n>s && n<=K)
20     pn(1,lamda+1)=loi2/(som1+som2);
21     loi=pn;
22 end
23 end
24 end
25 lamda=0:z;
26 plot(lamda, loi)
27 xlabel('lamda')
28 ylabel('pn')
29 else
30 disp('donner une autre valeur à n')
31 end
32 end

```

FIGURE 3.3 – Programme de la loi de $(N_t)_{t \geq 0}$

Les graphes obtenus après simulation sont :

Pour $n=0$

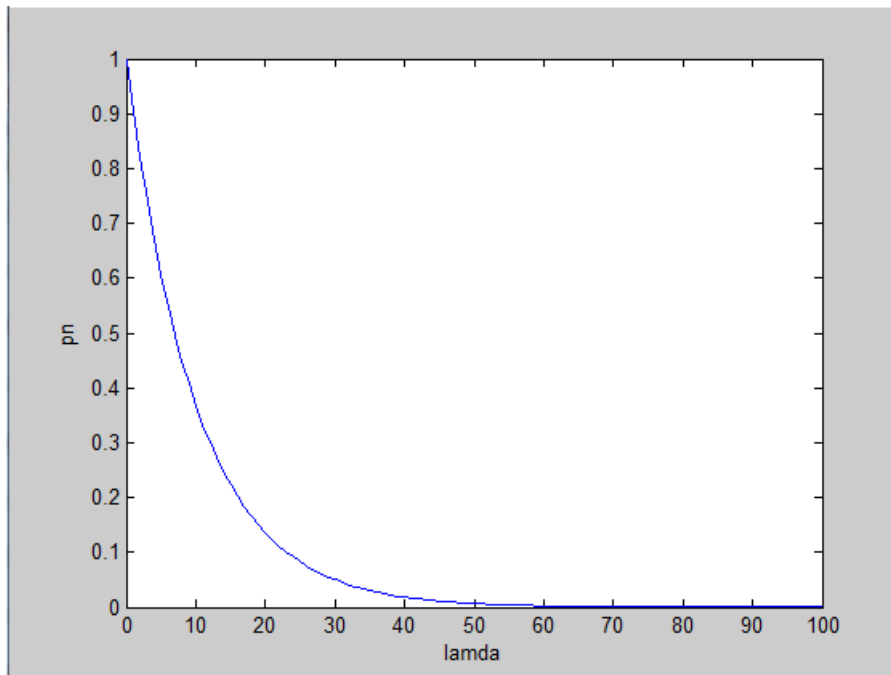


FIGURE 3.4 – Loi de N_t pour $n = 0$

Pour $n = 5 \leq s$

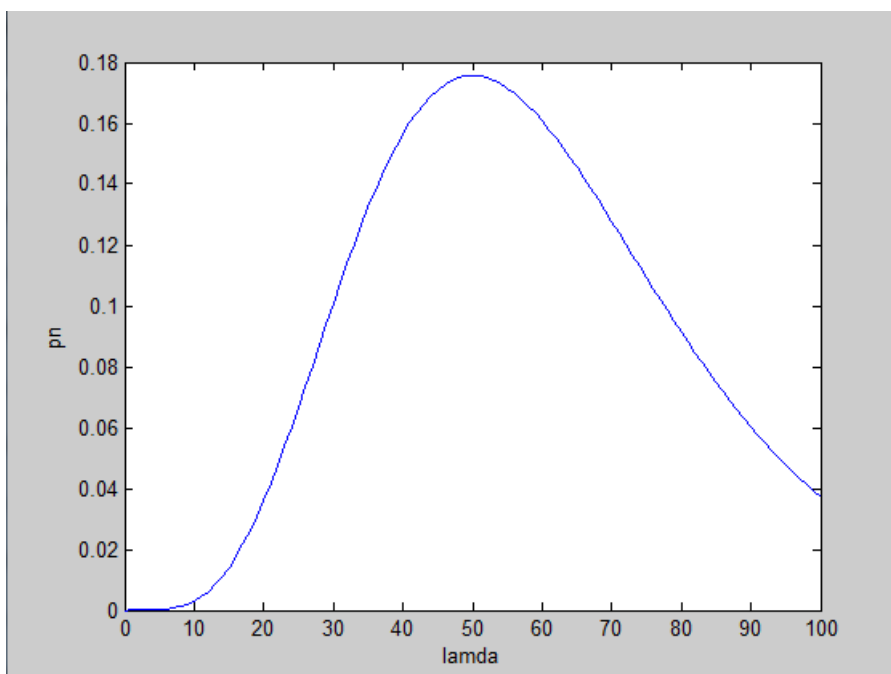


FIGURE 3.5 – Loi de N_t pour $n = 5$

Pour $n = 25 > s$

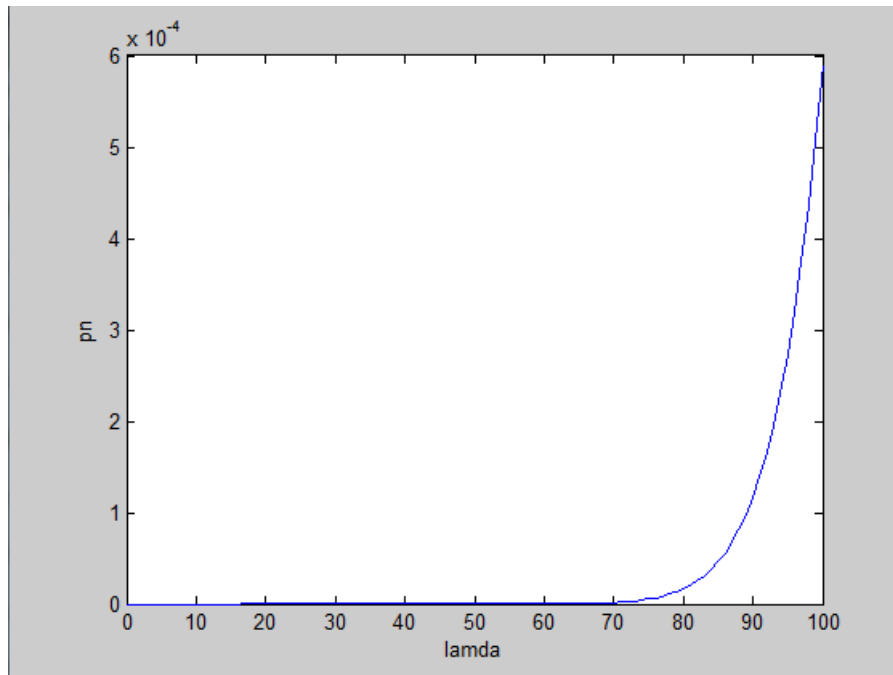


FIGURE 3.6 – Loi de N_t pour $n = 25$

3.3.1.2 Simulation des caractéristiques de la file

On donne d'abord le programme de $P(N_t = 0) = P_0$.

```
1  function pzero= p0( mu, lamda, s, K)
2  -      r=(s^s)/fact(s);
3  -      som1=0;
4  -      som2=0;
5  -      for i=0:s
6  -          som1=som1+(lamda/mu)^i/fact(i);
7  -      end
8  -      for j=s+1:K
9  -          som2=som2+(r*((lamda/(s*mu))^j));
10 -      end
11 -      pzero=(1/(som1+som2));
12 -
13 -      end
```

FIGURE 3.7 – Programme de $P(N_t = 0) = P_0$

A- Nombre moyen d'appels en attente de connexion

i. Simulation en fonction de λ

Nous fixons $K = 80$, $\mu = 10$, $s = 15$ et observons l'évolution de cette caractéristique.

```
1  function nbrmoyapp = appelsk(z)
2  -   mu=10; K=80; s=15;
3  -   for lamda =0:z;
4  -       w = p0( mu, lamda, s, K);
5  -       som=0;
6  -       for i=1:K-s
7  -           som=som+(i*((lamda/(s*mu))^i));
8  -       end
9  -       somme=som*(w*((lamda/mu)^s)/fact(s));
10 -      moy(1,lamda+1)=somme;
11 -      nbrmoyapp = moy;
12 -   end
13 -   lamda=0:z;
14 -   plot(lamda, nbrmoyapp)
15 -   xlabel('lamda')
16 -   ylabel('Nombre moyen d'appels')
17 - end
```

FIGURE 3.8 – Programme du nombre moyen d'appels en attente de connexion en fonction de λ

Le graphe obtenu après la simulation est :

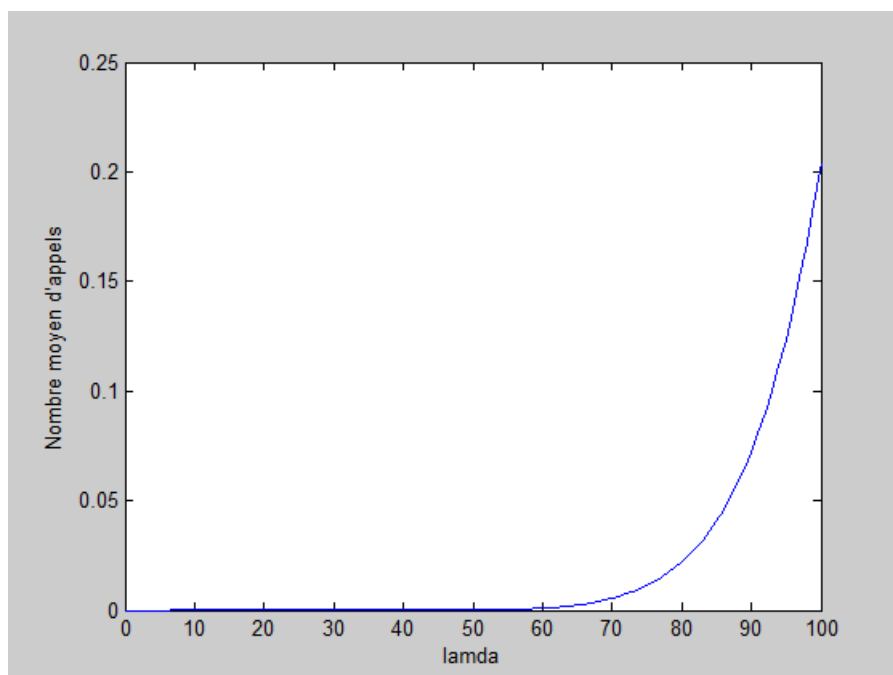


FIGURE 3.9 – Nombre moyen d'appels en attente de connexion en fonction de λ

ii. Simulation en fonction de μ

Nous fixons $K = 80$, $\lambda = 20$, $s = 15$ et observons l'évolution cette caractéristique en fonction du taux de mort.

```
1 function nbrmoyapp = appelskmu(z)
2   lamda=20; K=80; s=15;
3   for mu =1:z;
4       w= p0( mu, lamda, s, K);
5       som=0;
6       for i=1:K-s
7           som=som+(i*((lamda/(s*mu))^i));
8       end
9       somme=som*(w*((lamda/mu)^s)/fact(s));
10      moy(1,mu)=somme;
11      nbrmoyapp = moy;
12  end
13  plot(nbrmoyapp)
14  xlabel('mu')
15  ylabel('Nombre moyen d'appels')
16  end
```

FIGURE 3.10 – Programme du nombre moyen d'appels en attente de connexion en fonction de μ

Le graphe obtenu après la simulation est :

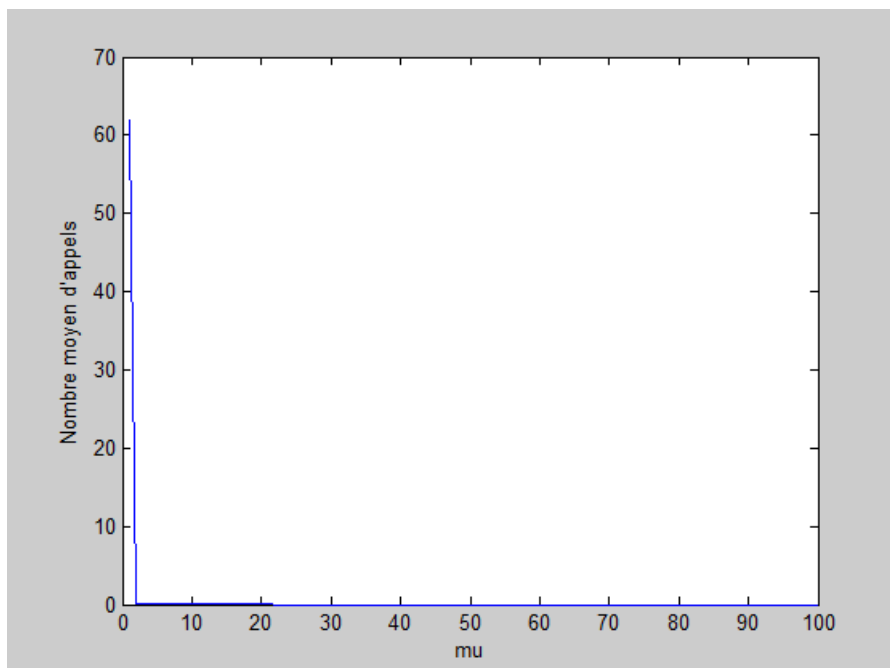


FIGURE 3.11 – Nombre moyen d'appels en attente de connexion en fonction de μ

iii. Simulation en fonction de s

On fait évoluer le nombre moyen d'appels en attente de connexion en fonction du nombre de serveurs s tout en fixant $K = 80$, $\mu = 10$ et $\lambda = 20$.

```
1 function nbrmoyapp = appelsks(z)
2   lamda=20; K=80; mu=10;
3   for s =1:z;
4     w = p0( mu, lamda, s, K);
5     som=0;
6     for i=1:K-s
7       som=som+(i*((lamda/(s*mu))^i));
8     end
9     somme=som*(w*((lamda/mu)^s)/fact(s));
10    moy(1,s)=somme;
11    nbrmoyapp=moy;
12  end
13  plot(nbrmoyapp)
14  xlabel('s')
15  ylabel('Nombre moyen d'appels')
16  end
```

FIGURE 3.12 – Programme du nombre moyen d'appels en attente de connexion en fonction de s

Le graphe après la simulation est :

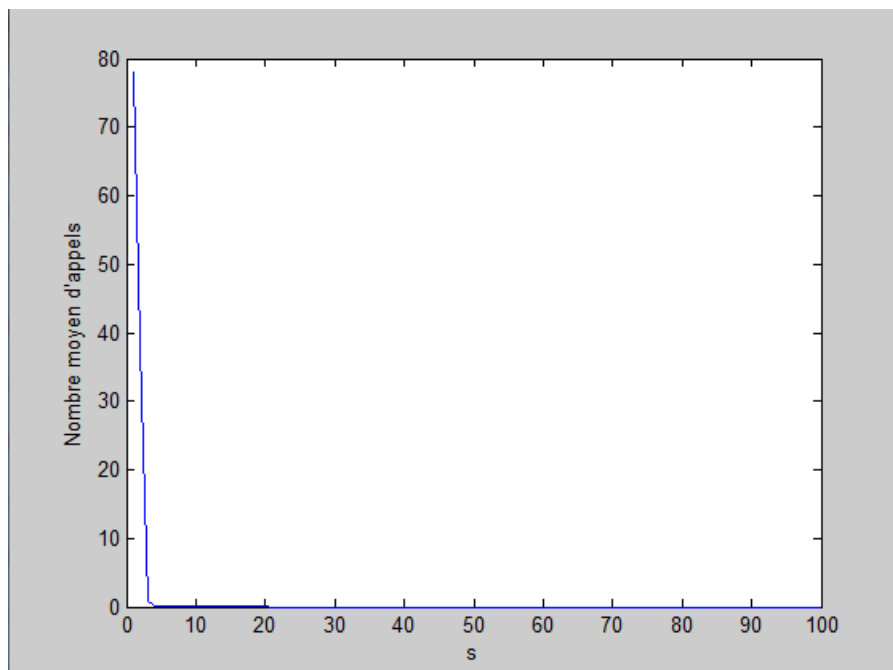


FIGURE 3.13 – Nombre moyen d'appels en attente de connexion en fonction de s

B- Durée moyenne d'un appel avant connexion

i. Simulation en fonction de λ

Nous fixons $K = 80$, $\mu = 10$, $s = 15$ et cette caractéristique évolue comme suit, on donne le programme suivant :

```
1 - function tempsmoyenne1 = temps1sk(z)
2 -     s=15; K=80; mu=10;
3 -     for lamda=1:z
4 -         w=p0(mu, lamda, s, K);
5 -         som=0;
6 -         for i=1:K-s
7 -             som=som+(i*((lamda/(s*mu))^i));
8 -         end
9 -
10 -         lambda=lamda*(1-(((lamda/mu)^K)/(fact(s)*(s^(K-s))*w)));
11 -         moyenn= ((w*((lamda/mu)^s))/fact(s))*som/lambda ;
12 -         temp(1, lamda+1)=moyenn;
13 -         tempsmoyenne1=temp;
14 -     end
15 -     plot(tempsmoyenne1)
16 -     xlabel('lamda')
17 -     ylabel('Durée moyenne d''attente')
18 - end
```

FIGURE 3.14 – Programme de la durée moyenne d'un appel avant connexion en fonction de λ

Le graphe après la simulation est :

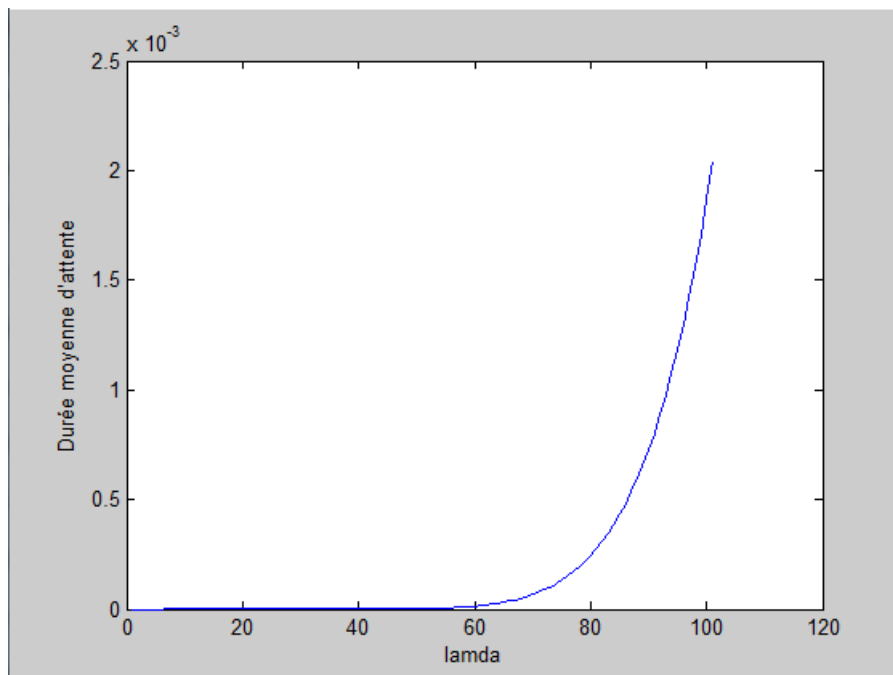


FIGURE 3.15 – Durée moyenne d'un appel avant connexion en fonction de λ

Où on remarque l'évolution du graphe d'une façon croissante.

ii. Simulation en fonction de μ

On s'intéresse à l'évolution du temps d'attente d'un appel avant connexion. Le programme permettant d'observer l'évolution de la caractéristique en fonction de μ .

```
1  function tempsmoyenne2 = temps2sk(z)
2  -   s=15; K=80; lamda=20;
3  -   for mu=1:z
4  -       w=p0(mu, lamda, s, K);
5  -       som=0;
6  -       for i=1:K-s
7  -           som=som+(i*((lamda/(s*mu))^i));
8  -       end
9  -
10 -       lambda=lamda*(1-(((lamda/mu)^K)/(fact(s)*s^(K-s))*w));
11 -       moyenn= (((w*((lamda/mu)^s))/fact(s))*som)/lambda ;
12 -       temp(1,mu)=moyenn;
13 -       tempsmoyenne2=temp;
14 -   end
15 -   plot(tempsmoyenne2)
16 -   xlabel('mu')
17 -   ylabel('Durée moyenne d''attente')
18 - end
```

FIGURE 3.16 – Programme de la durée moyenne d'un appel avant connexion en fonction de μ

Le graphe obtenu après la simulation est :

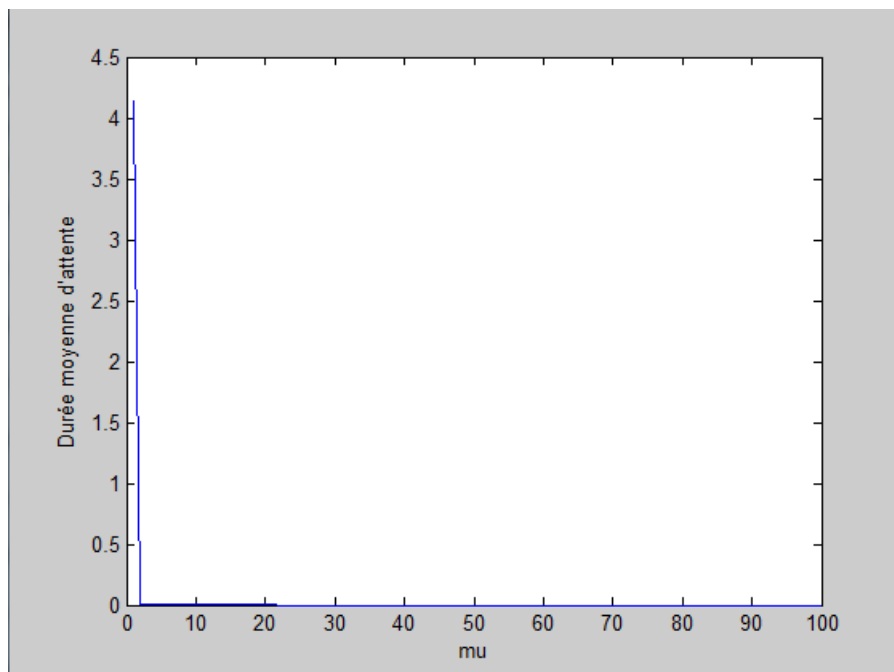


FIGURE 3.17 – Durée moyenne d'un appel avant connexion en fonction de μ

iii. Simulation en fonction de s

On évolue le temps moyen d'un appel avant connexion en fonction du nombre de canaux s tout en fixant $K = 80$, $\lambda = 20$ et $\mu = 10$.

On donne le programme suivant :

```
1 function tempsmoyenne3 = temps3sk(z)
2 mu=10; K=80; lamda=20;
3 for s=1:z
4     w=p0(mu, lamda, s, K);
5     som=0;
6     for i=1:K-s
7         som=som+(i*((lamda/(s*mu))^i));
8     end
9
10    lambda=lamda*(1-(((lamda/mu)^K)/(fact(s)*s^(K-s)*w)));
11    moyenn= ((w*((lamda/mu)^s))/fact(s))*som/lambda;
12    temp(1,s)=moyenn;
13    tempsmoyenne3=temp;
14 end
15 plot(tempsmoyenne3)
16 xlabel('s')
17 ylabel('Durée moyenne d\'attente')
18 end
```

FIGURE 3.18 – Programme de la durée moyenne d'un appel avant connexion en fonction de s

Le graphe obtenu après la simulation est :

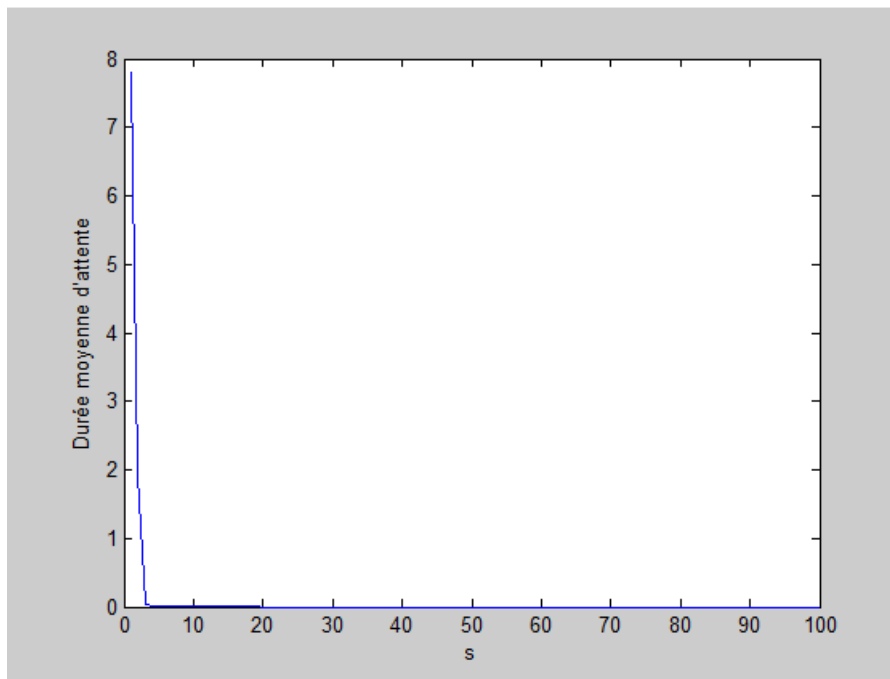


FIGURE 3.19 – Durée moyenne d'un appel avant connexion en fonction de s

3.3.1.3 Risque de saturation

On s'intéresse dans ce cas à l'évolution du risque de saturation dans le centre d'appels téléphoniques en fonction des paramètres λ , μ et s .

i. Evolution en fonction de λ

On suppose que $K = 80$, $\mu = 10$ et $s = 15$.

Le programme est le suivant :

```
1  function risquesaturation1 = risque1(z)
2  -   mu=10; K=80; s=15;
3  -   for lamda=0:z
4  -       loi=(lamda/mu)^s/fact(s);
5  -       w=p0(mu,lamda,s,K);
6  -       som=0;
7  -       for i=0:K-s
8  -           som=som+(lamda/(mu*s))^i;
9  -       end
10 -       risque=(loi*w)*som;
11 -       risquee(1,lamda+1)=risque;
12 -       risquesaturation1=risquee;
13 -   end
14 -   lambda=0:z;
15 -   plot(lambda,risquesaturation1)
16 -   xlabel('lamda')
17 -   ylabel('Risque de saturation')
18 - end
```

FIGURE 3.20 – Programme du risque de saturation en fonction de λ

Le graphe obtenu après la simulation est :

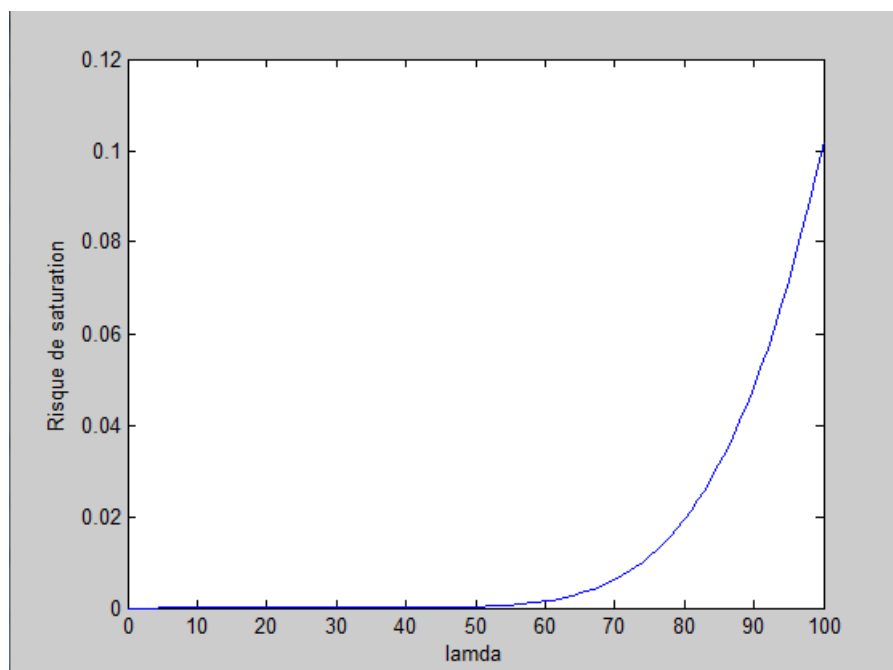


FIGURE 3.21 – Risque de saturation en fonction de λ

ii. Evolution en fonction de μ

On prend $K = 80$, $\lambda = 20$ et $s = 15$.

Le programme est le suivant :

```
1  function risquesaturation2 = risque2(z)
2  - lamda=20; K=80; s=15;
3  - for mu=1:z
4  -     loi=((lamda/mu)^s)/fact(s);
5  -     w=p0(mu,lamda,s,K);
6  -     som=0;
7  -     for i=0:K-s
8  -         som=som+((lamda/(mu*s))^i);
9  -     end
10 -     risque=(loi*w)*som;
11 -     risquee(1,mu)=risque;
12 -     risquesaturation2=risquee;
13 - end
14 - plot(risquesaturation2)
15 - xlabel('mu')
16 - ylabel('Risque de saturation')
17 - end
```

FIGURE 3.22 – Programme du risque de saturation en fonction de μ

Le graphe obtenu après la simulation est :

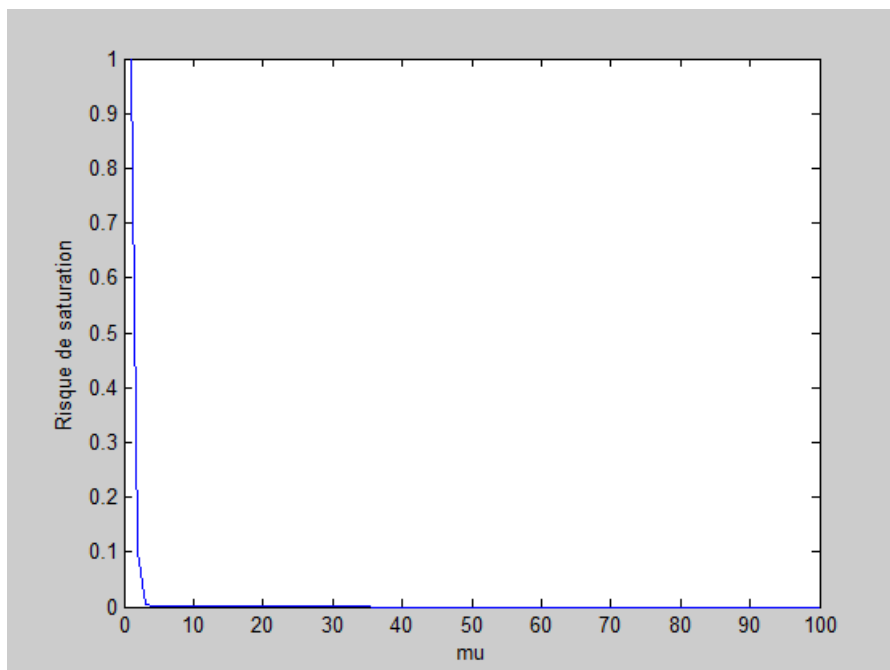


FIGURE 3.23 – Risque de saturation en fonction de μ

iii. Evolution en fonction de s

On suppose que $K = 80$, $\mu = 10$ et $\lambda = 20$.

Le programme est le suivant :

```
1  function risquesaturation3 = risque3(z)
2  lamda=20; K=80; mu=10;
3  for s=1:z
4      w=p0(mu, lamda, s, K);
5      loi=((lamda/mu)^s)/fact(s);
6      som=0;
7      for i=0:K-s
8          som=som+((lamda/(mu*s))^i);
9      end
10     risque=(loi*w)*som;
11     risquee(1,s)=risque;
12     risquesaturation3=risquee;
13 end
14 plot(risquesaturation3)
15 xlabel('s')
16 ylabel('Risque de saturation')
17 end
```

FIGURE 3.24 – Programme du risque de saturation en fonction de s

Le graphe obtenu après simulation est :

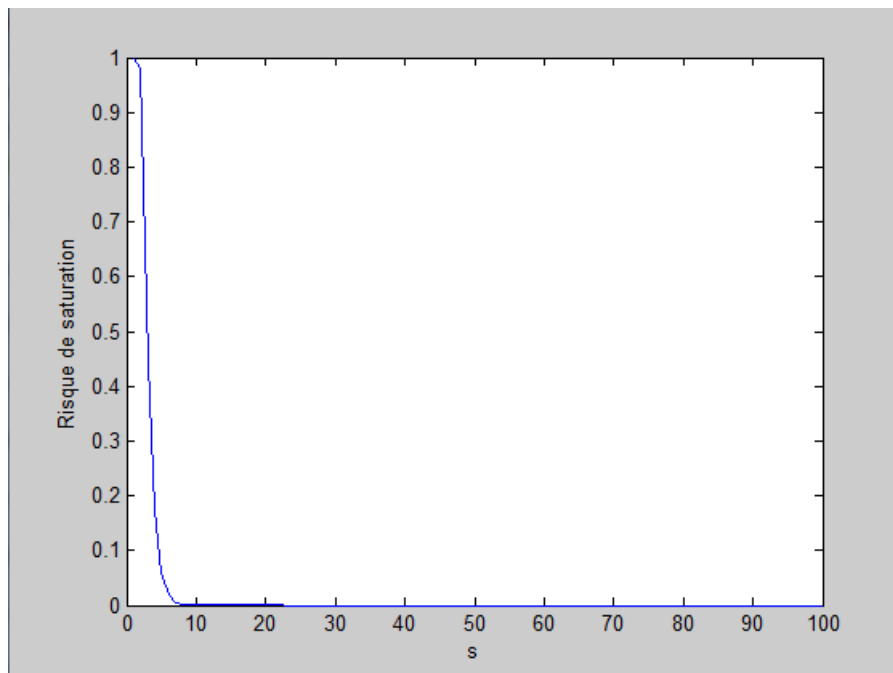


FIGURE 3.25 – Risque de saturation en fonction de s

3.3.2 File d'attente $M/M/s/s$

3.3.2.1 Simulation de la loi de N_t

On considère la loi $(N_t)_{t \geq 0}$ en fonction du taux des arrivées d'appels au centre d'appels téléphoniques λ . On suppose que le taux de sortie est égal à 10 et le nombre de serveurs s est égal à 15.

Le programme de la loi de $(N_t)_{t \geq 0}$ est le suivant :

```
1 - function pntss = pnss(n,z)
2 -     mu=10;
3 -     s=15;
4 -     if n>=0 && n<=s
5 -         for lamda=0:z
6 -             som=0;
7 -             loi=((lamda/mu)^n)/fact(n);
8 -             for i=0:s
9 -                 som=som+((lamda/mu)^i)/fact(i);
10 -            end
11 -            pn(1,lamda+1)=loi/som;
12 -            pntss=pn;
13 -        end
14 -        lamda=0:z;
15 -        tab=[lamda;pntss]
16 -        plot(lamda,pntss)
17 -        xlabel('lamda')
18 -        ylabel('loi Nt')
19 -     else
20 -         disp('pn=0')
21 -     end
22 - end
--
```

FIGURE 3.26 – Programme de la loi (N_t)

Les graphes obtenus après simulation sont :

Pour $n=0$

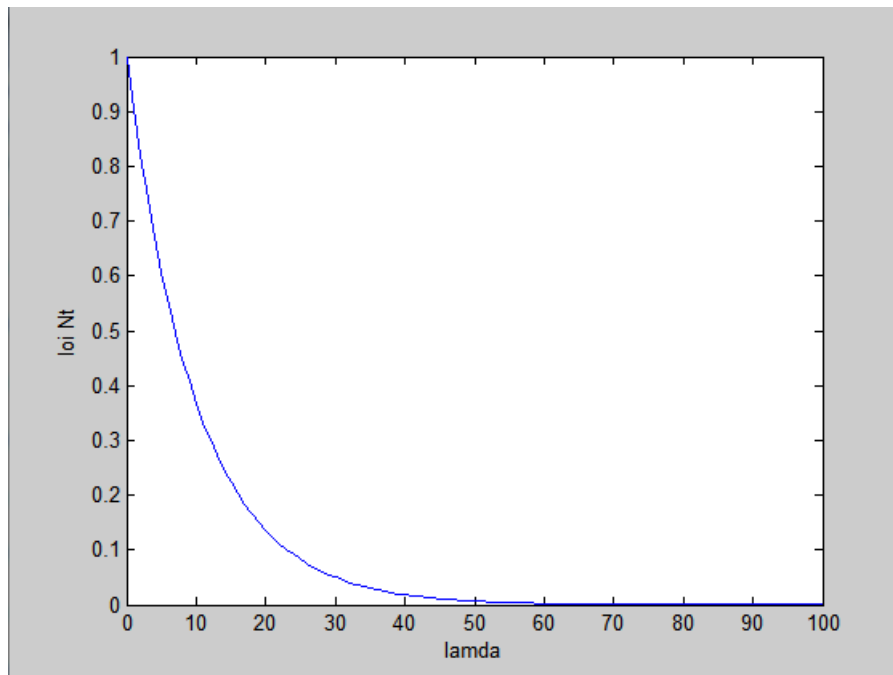


FIGURE 3.27 – Loi de (N_t) pour $n=0$

Pour $n=12$

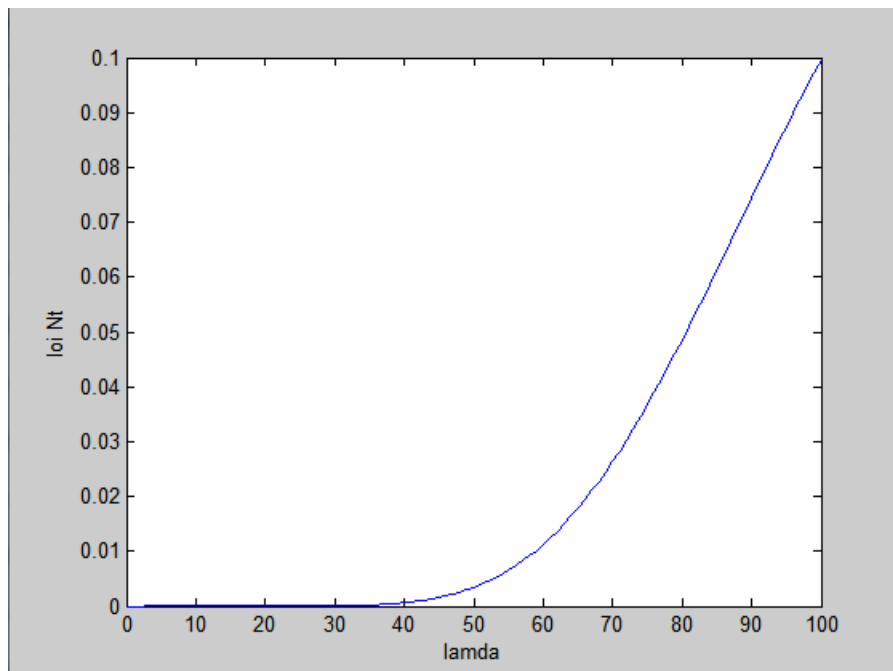


FIGURE 3.28 – Loi de (N_t) pour $n=12$

3.3.2.2 Simulation des caractéristiques de la file

On donne d'abord le programme de $P(N_t = 0) = P_0$ et celui de $P(N_t = s) = P_s$.

```
1 function pzeross = p00(s,mu,lamda)
2     som=0;
3     for i=0:s
4         som=som+((lamda/mu)^i)/fact(i);
5     end
6     pzeross=1/som;
7 end
```

FIGURE 3.29 – Programme de $P(N_t = 0) = P_0$

```
1 function ps = ps(s,mu,lamda)
2     som=0;
3     for i=0:s
4         som=som+((lamda/mu)^i)/fact(i);
5     end
6     ps=((lamda/mu)^s)/(fact(s)*som);
7 end
8
```

FIGURE 3.30 – Programme de $P(N_t = s) = P_s$

A- Nombre moyen d'appels dans le centre d'appels téléphoniques

i. Simulation en fonction de λ

Nous fixons $\mu = 10$, $s = 15$ et observons l'évolution de cette caractéristique

```
1 function nmbrmoy = appelss(z)
2     mu=10; s=15;
3     for lamda= 0:z
4         Ps= ps(s,mu,lamda);
5         moy(1,lamda+1)=(lamda*(1-Ps))/mu;
6         nmbrmoy=moy;
7     end
8     lambda=0:z;
9     plot(lambda,nmbrmoy)
10    xlabel('lambda')
11    ylabel('Nombre moyen d'appels')
12 end
```

FIGURE 3.31 – Programme du nombre moyen d'appels dans le centre d'appels en fonction de λ

Le graphe obtenu après la simulation est :

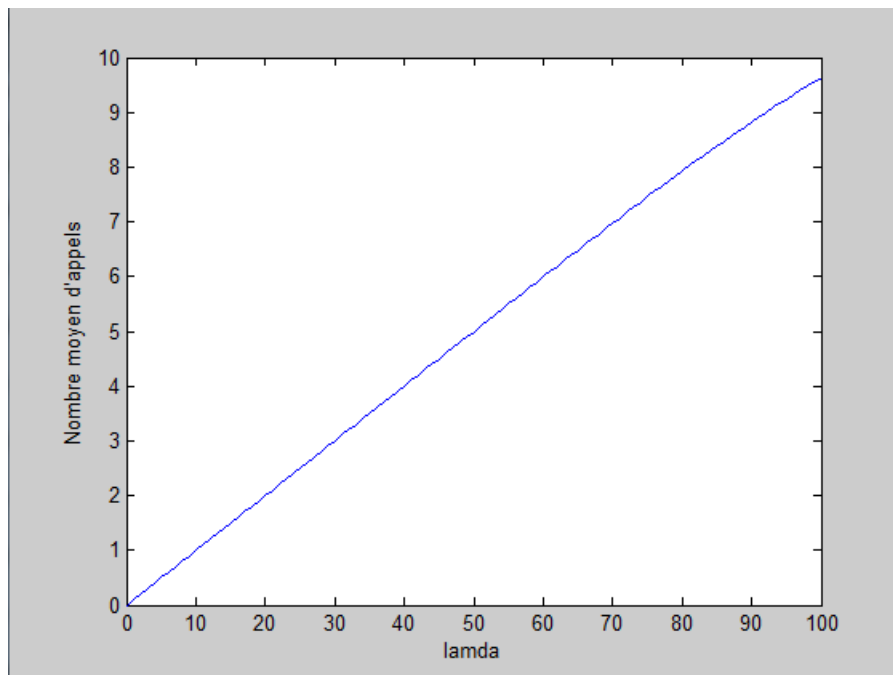


FIGURE 3.32 – Nombre moyen d'appels dans le centre d'appels en fonction de λ

Où on remarque l'évolution du graphe d'une façon croissante.

ii. Simulation en fonction de μ

Nous fixons $\lambda = 20$, $s = 15$ et observons l'évolution de cette caractéristique en fonction du taux de mort.

```
1  function nmbrmoy = appelssmu(z)
2  -   lamda=20; s=15;
3  -   for mu= 1:z
4  -       Ps= ps(s,mu,lamda);
5  -       moy(1,mu)=(lamda*(1-Ps))/mu;
6  -       nmbrmoy=moy;
7  -   end
8  -   mu=1:z;
9  -   plot(mu,nmbrmoy)
10 -   xlabel('mu')
11 -   ylabel('Nombre moyen d'appels')
12 -   end
..
```

FIGURE 3.33 – Programme du nombre moyen d'appels dans le centre d'appels en fonction de μ

Le graphe obtenu après la simulation est :

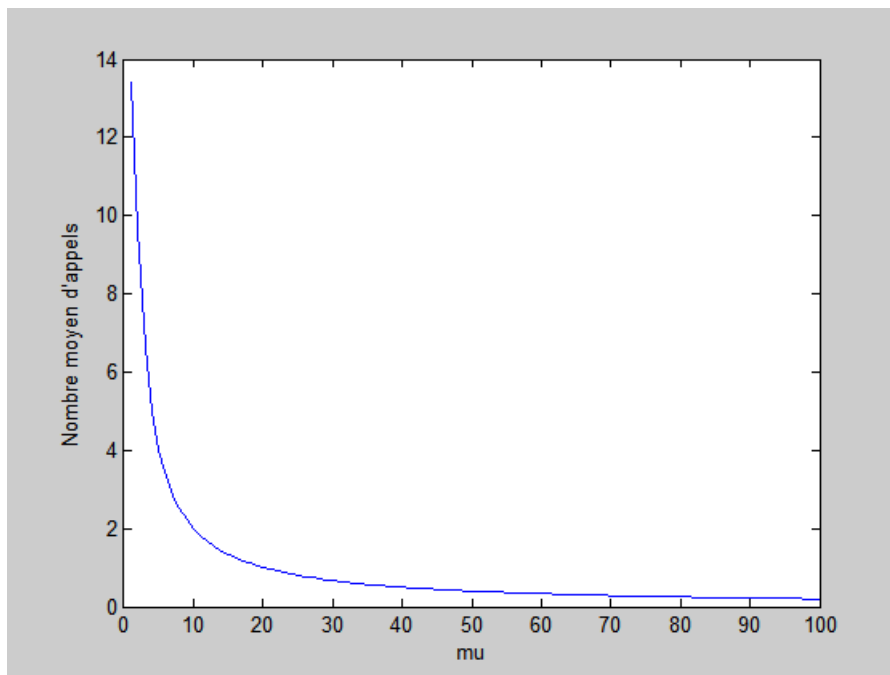


FIGURE 3.34 – Nombre moyen d'appels dans le centre d'appels en fonction de μ

Remarque

Le nombre moyen d'appels est constant par apport à s .

B- Durée moyenne d'un appel dans le centre d'appels téléphoniques

i. Simulation en fonction de μ

On s'intéresse à l'évolution du temps moyen d'attente dans le centre d'appels téléphoniques. Le programme permettant d'observer l'évolution de cette caractéristique en fonction de μ est :

```
1  function tempsmoy = tempsmoyss(z)
2  - for mu=1:z
3  -     tempsmu(1,mu)=(1/mu);
4  -     tempsmoy=tempsmu;
5  - end
6  plot(tempsmoy)
7  xlabel('mu')
8  ylabel('Durée moyenne d'un appel')
9  end
```

FIGURE 3.35 – Programme de la durée moyenne d'un appel dans le centre d'appels en fonction de μ

Le graphe obtenu après la simulation est :

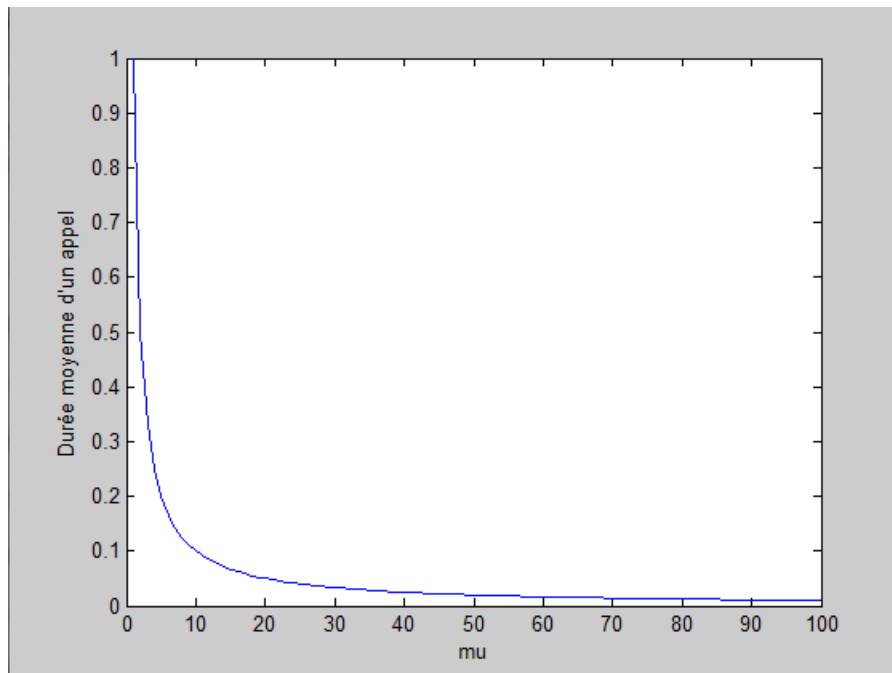


FIGURE 3.36 – Durée moyenne d'un appel dans le centre d'appels en fonction de μ

3.3.2.3 Risque de saturation

On s'intéresse dans ce cas à l'évolution du risque de saturation dans le centre d'appels téléphoniques en fonction des paramètres λ , μ et s .

i. Evolution en fonction de λ

On suppose que $\mu = 10$ et $s = 15$.

Le programme est le suivant :

```
1 - function risquesaturass = risquess(z)
2 -     mu=10; s=15;
3 -     for lamda=0:z
4 -         loi= ((lamda/mu)^s)/fact(s);
5 -         som=0;
6 -         for i=0:s
7 -             som=som+(((lamda/mu)^i)/fact(i));
8 -             risquee(1,lamda+1)=loi/som;
9 -             risquesaturass =risquee;
10 -        end
11 -    end
12 -    lambda=0:z;
13 -    plot(lambda,risquesaturass)
14 -    xlabel('lamda')
15 -    ylabel('Risque de saturation')
16 - end
```

FIGURE 3.37 – Programme du risque de saturation en fonction de λ

Le graphe obtenu après la simulation est :

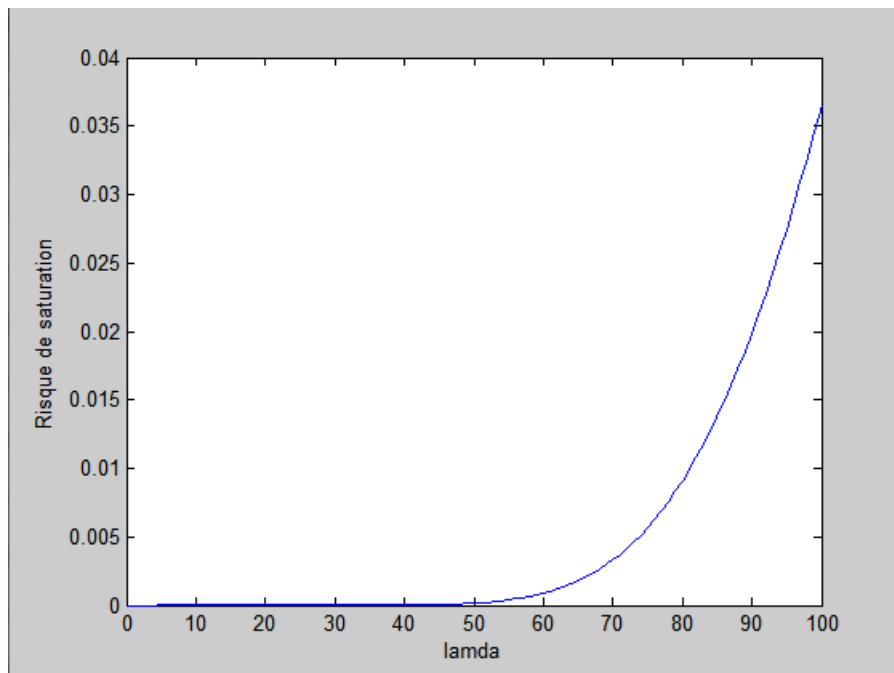


FIGURE 3.38 – Risque de saturation en fonction de λ

ii. Evolution en fonction de μ

On prend $\lambda = 20$, $s = 15$ et le programme est le suivant :

```
1  function risquesaturassmu = risquessmu(z)
2  lamda=20; s=15;
3  for mu=1:z
4      loi= ((lamda/mu)^s)/fact(s);
5      som=0;
6      for i=0:s
7          som=som+((lamda/mu)^i)/fact(i);
8          risquee(1,mu)=loi/som;
9          risquesaturassmu =risquee;
10     end
11 end
12 plot(risquesaturassmu)
13 xlabel('mu')
14 ylabel('Risque de saturation')
15 end
```

FIGURE 3.39 – Programme du risque de saturation en fonction de μ

Le graphe obtenu après la simulation est :

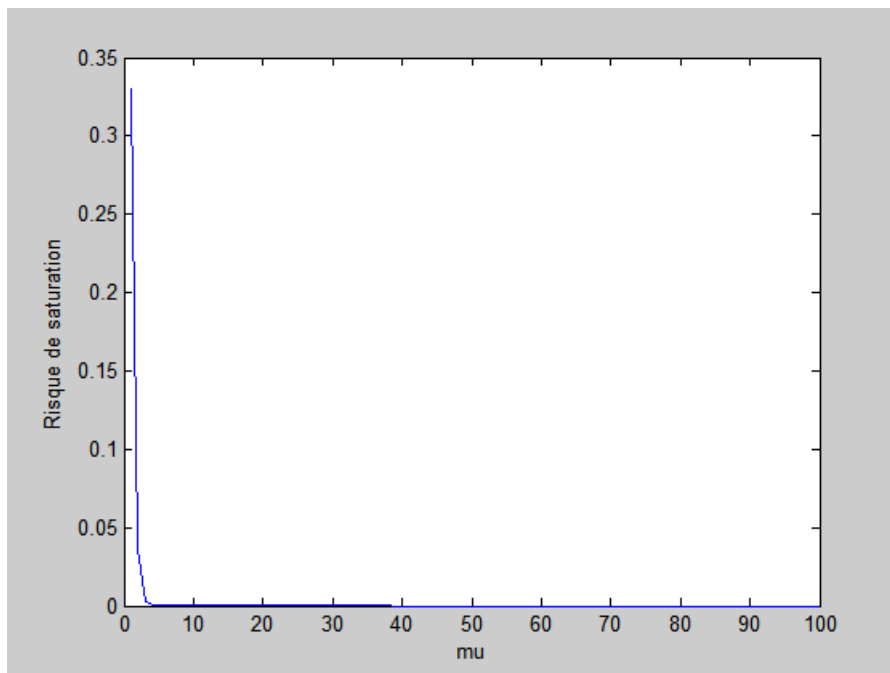


FIGURE 3.40 – Risque de saturation en fonction de μ

iii. Evolution en fonction de s

On suppose que $\mu = 10$ et $\lambda = 20$.

Le programme est le suivant :

```
1  function risquesaturass = risquesss(z)
2  -   lamda=20; mu=10;
3  -   for s=1:z
4  -       loi= ((lamda/mu)^s)/fact(s);
5  -       som=0;
6  -       for i=0:s
7  -           som=som+((lamda/mu)^i)/fact(i);
8  -           risquee(1,s)=loi/som;
9  -           risquesaturass =risquee;
10 -       end
11 -   end
12 -   plot(risquesaturass)
13 -   xlabel('s')
14 -   ylabel('Risque de saturation')
15 - end
```

FIGURE 3.41 – Programme du risque de saturation en fonction de s

Le graphe obtenu après la simulation est :

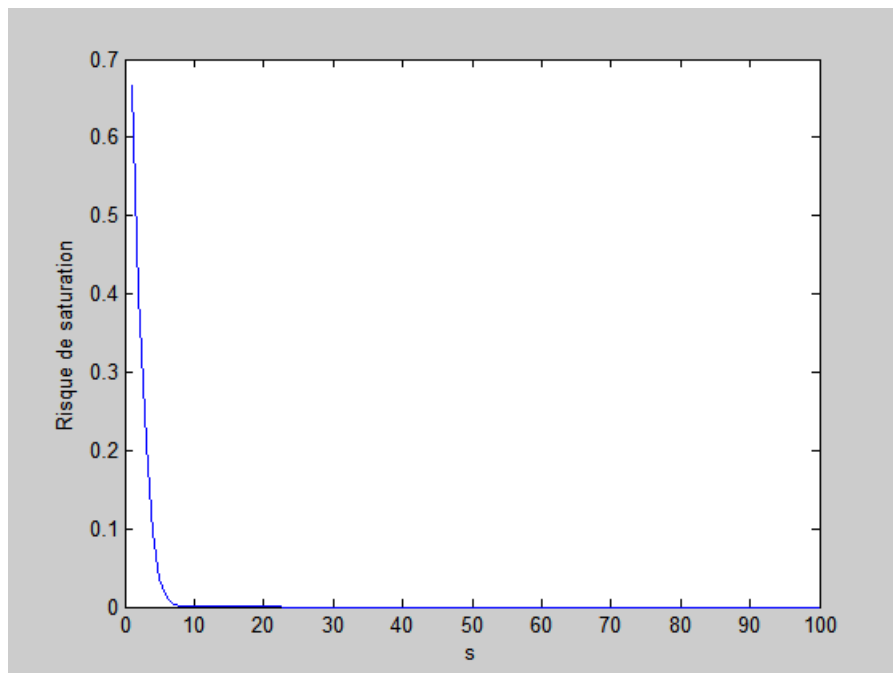


FIGURE 3.42 – Risque de saturation en fonction de s

3.4 Conclusion

On s'est intéressé dans ce chapitre à la simulation des caractéristiques des différentes files d'attente en fonction des différents paramètres d'entrée λ , μ et s , ce qui nous a permis de bien comprendre leurs évolutions, ainsi que l'évaluation du comportement dynamique des systèmes étudiés.

Chapitre 4

Optimisation du télétrafic dans un centre d'appels téléphoniques

4.1 Introduction

L'optimisation couvre un large éventail de techniques utilisées dans de nombreux domaines de la vie courante. Elle occupe une place très importante en recherche opérationnelle. En effet, elle assure la recherche d'une solution optimale pour un problème donné défini par une fonction objective qui consiste à déterminer un minimum ou un maximum en respectant un ensemble de contraintes sous forme d'égalités ou d'inégalités afin d'optimiser et améliorer le système étudié.

Notre étude consiste à chercher le nombre nécessaire de serveurs qui nous assure le bon déroulement du système, cela revient à optimiser le nombre s de serveurs pour que le risque de saturation soit inférieur à un α donné.

4.2 File d'attente $M/M/s/K$

4.2.1 Position du problème

On a le risque de saturation qui est donné par :

$$\pi(\rho, s, K) = \frac{\rho^s}{s!} p_0 \sum_{h=0}^{K-s} (\bar{\rho})^h.$$

Notre but est d'optimiser la probabilité de saturation $\pi(\rho, s, K)$ par rapport au nombre de serveurs (s), c'est-à-dire minimiser sa fonction objective.

$$\min_s \pi(\rho, s, K) = \min_s \left(\frac{\rho^s}{s!} p_0 \sum_{h=0}^K (\bar{\rho})^h \right) \text{ avec } s \in \mathbb{N}^*,$$

$$\iff \begin{cases} \min \left(\frac{\rho^s}{s!} p_0 \sum_{h=0}^K (\bar{\rho})^h \right) \\ s \in \mathbb{N}^*. \end{cases}$$

4.2.2 Optimisation du risque de saturation

Étant donné un seuil α , tel que $\alpha \in [0, 1]$, nous devons trouver le nombre minimum de serveurs nécessaires pour que la probabilité de saturation ne dépasse pas le seuil donné. On suppose que le taux d'entrée est égal à 2, le taux de sortie est égal à 3 et $K = 10$.

D'abord, donnons le programme sous Matlab du nombre moyen d'appels et de la durée moyenne d'un appel dans le centre d'appels.

i. Nombre moyen d'appels dans le centre d'appels

```
1  function nombremoy1=appell1(mu,lamda,s,K)
2  -   r=(s^(s-1))/fact(s-1);
3  -   w= p0( mu, lamda, s, K);
4  -   som1=0;
5  -   som2=0;
6  -   for i=1:s
7  -       som1=som1+(lamda/mu)^i/fact(i-1);
8  -   end
9  -   for j=s+1:K
10 -       som2=som2+(j*((lamda/(s*mu))^j));
11 -   end
12 -   somme=(som1+(r*som2))*w;
13 -   nombremoy1=somme;
14 - end
15
```

FIGURE 4.1 – Programme du nombre moyen d'appels dans le centre d'appels téléphoniques

ii. Durée moyenne d'un appel dans le centre d'appels

```
1  function tempsmoy1=temps1(mu,lamda,s,K)
2  -   loi=((lamda/mu)^K)/(fact(s)*(s^(K-s)));
3  -   pk=loi*p0( mu, lamda, s, K);
4  -   moyen=appell1(mu,lamda,s,K)/(lamda*(1-pk));
5  -   tempsmoy1=moyen;
6  - end
7
```

FIGURE 4.2 – Programme de la durée moyenne d'un appel dans le centre d'appels téléphoniques

On donne maintenant le programme permettant d'obtenir le nombre minimum de serveurs nécessaires pour que la probabilité de saturation soit inférieure à α donné, ainsi que le nombre moyen d'appels et la durée moyenne d'un appel dans le centre d'appels.

```

1 - function probasaturation = satural(alpha)
2 - som=0; K=10; lamda=2; mu=3; srv=1;
3 - for n=0:K-srv
4 -     som=som+((lamda/(mu*srv))^n);
5 - end
6 - probasaturation=(((lamda/mu)^srv)/fact(srv))*p0(mu, lamda, srv, K)*som;
7 - if (probasaturation <=alpha)
8 -     s=srv;
9 -     nombre_de_serveurs_s=s
10 -    nombre_moyen_dappels=appell(mu,lamda,s,K)
11 -    temps_moyen_dattente=temps1(mu,lamda,s,K)
12 -    probabilite_de_saturation=probasaturation
13 - else while (probasaturation>=alpha)
14 -     srv=srv+1;
15 -     for n=0:K-srv
16 -         som=som+((lamda/(srv*mu))^n);
17 -     end
18 -     risque=(((lamda/mu)^srv)/fact(srv))*p0(mu, lamda, srv, K)*som;
19 -     probasaturation=risque;
20 -     if (probasaturation<=alpha)
21 -         s=srv;
22 -         nombre_de_serveurs_s=s
23 -         nombre_moyen_dappels=appell(mu,lamda,s,K)
24 -         temps_moyen_dattente=temps1(mu,lamda,s,K)
25 -         probabilite_de_saturation=probasaturation
26 -     end
27 - end
28 - end
29 - end

```

FIGURE 4.3 – Programme d’optimisation du risque de saturation

```

>> satural(0.1)

nombre_de_serveurs_s =

     4

nombre_moyen_dappels =

     0.6677

temps_moyen_dattente =

     0.3338

probabilite_de_saturation =

     0.0293

ans =

     0.0293

```

FIGURE 4.4 – Resultats d’exécution du programme d’optimisation en fonction des paramètres donnés

4.2.3 Illustrations numériques et graphiques

Le tableau ci-dessous nous donne le nombre minimum de serveurs pour différentes valeurs de α , ainsi que la probabilité de saturation, le nombre moyen d'appels et la durée moyenne d'un appel dans le centre d'appels.

Alpha	Proba de saturation	Nbr de serveurs	Nbr moyen d'appels	Durée moyenne d'un appel
0.0001	$6.1515e^{-05}$	7	0.6667	0.3333
0.001	$5.7672e^{-04}$	6	0.6667	0.3333
0.01	0.0046	5	0.6668	0.3334
0.1	0.0293	4	0.6677	0.3338
0.3	0.1450	3	0.6760	0.3380
0.6	0.4942	2	0.7499	0.3750
0.9	0.6628	1	1.8713	0.9412

TABLE 4.1 – Illustrations numériques des caractéristiques du système en fonction du nombre de serveurs

À partir du tableau ci-dessus, on trace la courbe suivante :

*** Courbe de la probabilité de saturation en fonction du nombre minimum de serveurs**

```
>> serveurs=[1 2 3 4 5 6 7];
>> probabilite_satura=[0.6628 0.4942 0.1450 0.0293 0.0046 5.7672e-04 6.1515e-05];
>> plot(serveurs,probabilite_satura,'g')
>> grid on
>> xlabel('Nombre de serveurs')
>> ylabel('Probabilite de saturation')
```

FIGURE 4.5 – Programme de la probabilité de saturation en fonction de s

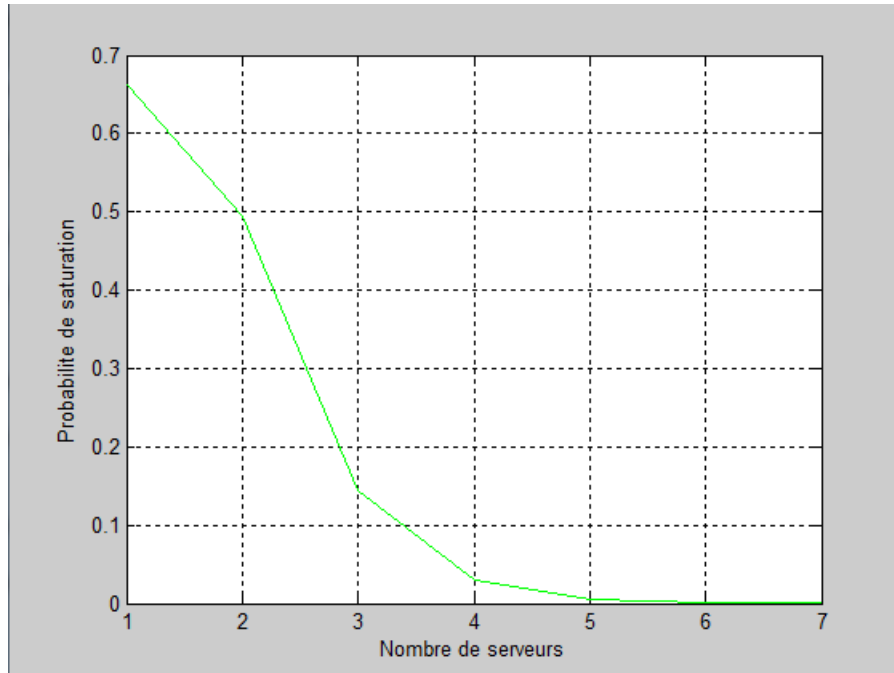


FIGURE 4.6 – Courbe de la probabilité de saturation en fonction du nombre minimum de serveurs

4.3 File d'attente $M/M/s/s$

4.3.1 Position du problème

On a le risque de saturation donné par :

$$\pi(\rho, s) = \frac{\frac{\rho^s}{s!}}{\sum_{n=0}^s \frac{\rho^n}{n!}}$$

Notre but est d'optimiser la probabilité de saturation $\pi(\rho, s)$ par rapport au nombre de serveurs (s), c'est-à-dire minimiser sa fonction objective.

$$\min_s \pi(\rho, s) = \min_s \left(\frac{\frac{\rho^s}{s!}}{\sum_{n=0}^s \frac{\rho^n}{n!}} \right) \text{ avec } s \in \mathbb{N}^*,$$

$$\iff \begin{cases} \min_s \left(\frac{\frac{\rho^s}{s!}}{\sum_{n=0}^s \frac{\rho^n}{n!}} \right) \\ s \in \mathbb{N}^* \end{cases}$$

4.3.2 Optimisation du risque de saturation

Étant donné un seuil α , tel que $\alpha \in [0, 1]$, nous devons trouver le nombre minimum de serveurs nécessaires pour que la probabilité de saturation ne dépasse pas le seuil donné. On suppose que le taux d'entrée est égal à 10 et le taux de sortie est égal à 5.

D'abord, donnons le code sous Matlab de $P(N_t = s) = P_s$, ensuite celui du nombre moyen d'appels et de la durée moyenne d'un appel dans le centre d'appels.

```

1  - function ps = ps(s,mu,lamda)
2  -     som=0;
3  -     for i=0:s
4  -         som=som+((lamda/mu)^i)/fact(i);
5  -     end
6  -     ps=((lamda/mu)^s)/(fact(s)*som);
7  - end
8

```

FIGURE 4.7 – Programme de $P(N_t = s) = P_s$

i. Nombre moyen d'appels dans le centre d'appels

```

1  - function nombremoy2=appel2(lamda,mu,s)
2  -     Ps=ps(s,mu,lamda);
3  -     nbr=(lamda*(1-Ps))/mu;
4  -     nombremoy2=nbr;
5  - end
6

```

FIGURE 4.8 – Programme du nombre moyen d'appels dans le centre d'appels

ii. Durée moyenne d'un appel dans le centre d'appels

```

1  - function tempsmoy2=temps2(mu)
2  -     tempsmoy2=1/mu;
3
4  - end
5

```

FIGURE 4.9 – Programme de la durée moyenne d'un appel dans le centre d'appels

On donne le programme permettant d'obtenir le nombre de serveurs nécessaires pour que la probabilité de saturation soit inférieure à α donné, ainsi que le nombre moyen d'appels et la durée moyenne d'un appel dans le centre d'appels.

```

1  function probasaturation = satura2(alpha)
2  - lamda=10; mu=5; srv=1;
3  - Ps=ps(srv,mu,lamda);
4  - loi=((lamda/mu)^srv)/fact(srv);
5  - probasaturation=loi*p00(srv,mu,lamda);
6  - if (probasaturation <=alpha)
7  -     s=srv;
8  -     nombre_de_serveurs_s=s
9  -     nombre_moyen_dappels=appel2(lamda,mu,s)
10 -     temps_moyen_dattente=temps2(mu)
11 -     probabilite_de_saturation=probasaturation
12 - else while (probasaturation>=alpha)
13 -     srv=srv+1;
14 -     loi=((lamda/mu)^srv)/fact(srv);
15 -     probasaturation=loi*p00(srv,mu,lamda);
16 -     if (probasaturation<=alpha)
17 -         s=srv;
18 -
19 -     nombre_de_serveurs_s=s
20 -     nombre_moyen_dappels=appel2(lamda,mu,s)
21 -     temps_moyen_dattente=temps2(mu)
22 -     probabilite_de_saturation=probasaturation
23 -     end
24 - end
25 - end
26 - end

```

FIGURE 4.10 – Programme d’optimisation du risque de saturation

```

>> satura2(0.1)

nombre_de_serveurs_s =

     4

nombre_moyen_dappels =

    0.6667

temps_moyen_dattente =

    0.2000

probabilite_de_saturation =

    0.0952

ans =

    0.0952

```

FIGURE 4.11 – Résultat d’exécution du programme d’optimisation en fonction des paramètres d’entrée

4.3.3 Illustrations numériques et graphiques

Le tableau ci-dessous nous donne le nombre minimum de serveurs pour différentes valeurs de α , ainsi que la probabilité de saturation, le nombre moyen d'appels et la durée moyenne d'un appel dans le centre d'appels.

Alpha	Proba de saturation	Nbr de serveurs	Nbr moyen d'appels	Durée moyenne d'un appel
0.001	$8.5948e^{-04}$	8	0.6667	0.2000
0.01	0.0034	7	0.6667	0.2000
0.03	0.0121	6	0.6667	0.2000
0.08	0.0367	5	0.6667	0.2000
0.1	0.0952	4	0.6667	0.2000
0.3	0.2105	3	0.6667	0.2000
0.6	0.4000	2	0.6667	0.2000
0.9	0.6667	1	0.6667	0.2000

TABLE 4.2 – Illustrations numériques des caractéristiques du système en fonction du nombre de serveurs

À partir du tableau ci-dessus, on trace la courbe suivante :

* Courbe de probabilité de saturation en fonction du nombre minimum de serveurs

```
>> serveurs=[1 2 3 4 5 6 7];
>> probabilite_satura=[0.6667 0.4000 0.2105 0.0952 0.0367 0.0121 0.0034];
>> plot(serveurs, probabilite_satura,'g')
>> grid on
>> xlabel('Nombre de serveurs')
>> ylabel('Probabilite de saturation')
```

FIGURE 4.12 – Programme de la probabilité de saturation en fonction de s

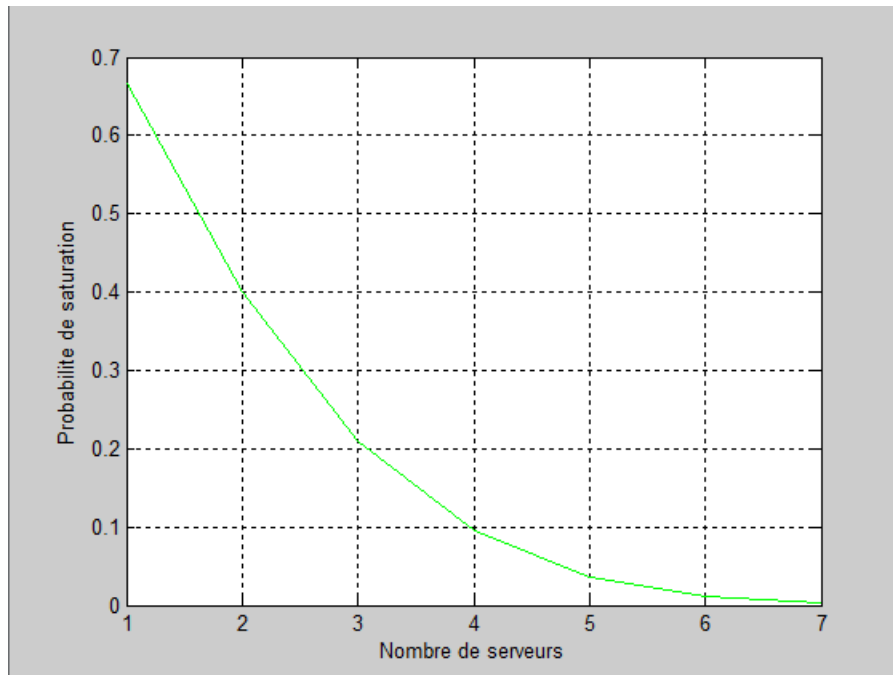


FIGURE 4.13 – Courbe de la probabilité de saturation en fonction du nombre minimum de serveurs

4.4 Conclusion

Dans ce chapitre, on a abordé un problème d'optimisation où on s'est intéressé à l'optimisation d'une manière informatique (avec le logiciel MATLAB) d'une des caractéristiques des deux systèmes de files d'attente qui nous a permis de déterminer le nombre minimum nécessaire de serveurs pour éviter tout risque de saturation et de congestion afin d'assurer un meilleur service ainsi qu'une bonne activité fluide dans les systèmes.

Conclusion générale

L'analyse et l'optimisation des systèmes dynamiques ainsi que l'évaluation de leurs performances dépendent essentiellement de leurs bonnes modélisations.

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

Dans ce mémoire, nous nous sommes intéressés particulièrement aux réseaux de files d'attentes ouverts, en particulier les appels téléphoniques entrants dans un centre d'appels téléphoniques. En premier lieu, nous avons accordé une attention particulière aux modèles et systèmes d'attente markoviens. En second lieu, nous avons pu présenter et étudier deux modèles de système de files d'attente $M/M/s/K$ et $M/M/s/s$ ainsi que leurs performances, tout en développant des programmes pour les simuler par la suite avec MATLAB. Enfin nous avons optimisé certaines performances (dans notre cas, minimiser le risque de saturation), afin de déterminer le nombre minimum nécessaire de serveurs pour garantir un bon déroulement et un meilleur service aux clients.

Une étude plus poussée de ce genre de systèmes est nécessaire pour améliorer et mieux évaluer les performances des systèmes informatiques, des réseaux de communications, systèmes industriels et systèmes complexes dans de nombreux domaines.

Bibliographie

- [1] Cours Master 1 RO : Processus stochastiques, Mr. HAMADOUCHE, (UMMTO).
- [2] Cours Master 2 RO : Processus stochastiques, Mr. HAMADOUCHE, (UMMTO).
- [3] Sabin Lessard (2014), processus stochastiques (cours et exercices corrigés).
- [4] Babes Malika (1995), statistiques, files d'attente et simulation.
- [5] Adrion Biran et Moshe Breiner (2004 version 6 et 7), MATLAB pour l'ingénieur.
- [6] Bechar Karim et Baahmed Ahmed, Mémoire de master, optimisation dans les systèmes de files d'attente (cas de gestion des arrivées des avions dans un aéroport), (UMMTO), (2016/2017).
- [7] Moulay Hachemi, Rahma Yasmina, Mémoire de Master, FILES D'ATTENTE ET APPLICATIONS, UNIVERSITÉ ABOUBAKR BELKAID - TLEMCEM, (2014/2015).

Résumé

Le monde contemporain est de plus en plus soucieux de satisfaire les attentes des clients, tellement il y a plusieurs choses dans la vie pour lesquelles il faut attendre. (les autobus, les ascenseurs, dans les magasins, chez le dentiste, etc.) Tout cela se passe en posant la fameuse question, pourquoi y a-t-il attente ? Sur cela, introduisons la notion des files d'attente.

La théorie des files d'attente est une théorie mathématique relevant du domaine des probabilités. Cette notion occupe une place très importante en recherche opérationnelle, elle sert à modéliser les systèmes d'attente et évaluer leurs performances afin de mieux comprendre leurs comportements.

Nous intéressons dans notre travail à l'étude et l'analyse des performances des systèmes de files d'attente dans un centre d'appels téléphoniques. Il s'agit d'un problème de modélisation dans un centre d'appels téléphoniques, où des appels arrivent d'une manière aléatoire, on suppose qu'ils suivent un processus de poisson de taux λ et que les durées des appels sont aussi aléatoire qu'on suppose indépendantes, de même loi exponentielle de paramètre μ .

Dans ce travail, nous nous sommes intéressés particulièrement aux réseaux de files d'attentes ouverts, en particulier les appels téléphoniques entrants dans un centre d'appels téléphoniques.

En premier lieu, nous avons accordé une attention particulière aux modèles et systèmes d'attente markoviens. En second lieu, nous avons pu présenter et étudier deux modèles de systèmes de files d'attente $M/M/s/K$ et $M/M/s/s$ ainsi que leurs performances, tout en développant des programmes pour les simuler par la suite avec MATLAB. Enfin nous avons optimisé certaines performances (dans notre cas, minimiser le risque de saturation), afin de déterminer le nombre minimum nécessaire de serveurs pour garantir un bon déroulement et un meilleur service aux clients.

Une étude plus poussée de ce genre de systèmes est nécessaire pour améliorer et mieux évaluer les performances des systèmes informatiques, des réseaux de communications, systèmes industriels et systèmes complexes dans de nombreux domaines.