

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université Mouloud Mammeri de TiziOuzou



Faculté des sciences
Département des
mathématiques



Mémoire de Master

Filière : Mathématiques.

Spécialité : Recherche Opérationnelle.

Thème

Optimisation dans les réseaux de télécommunication :

Cas d'un central téléphonique.

Réalisé par :

Mr DJEBID Ali

Mr OUMOHAND Amer

Encadré par :

Mr HAMADOUCHE Djamel

Année universitaire 2017/2018

Remerciements

Nous remercions d'abord le bon dieu qui nous a donné le courage et la volonté de rédiger ce modeste travail.

Nous tenons à remercier M^r HAMADOUCHE Djamel, notre encadreur pour son aide, sa disponibilité et sa grande générosité.

Nous exprimons notre sincère gratitude à nos enseignants qui nous ont suivis durant notre formation.

Une reconnaissance particulière à nos camarades qui nous ont suivis durant l'élaboration de ce travail.

Nos remerciements à nos parents qui ont été toujours présents à nos cotés.

Nos remerciements vont également à toutes les personnes, qui à un moment ou à un autre, de près ou de loin ont apporté leurs aides.

Dédicaces

Nous dédions ce travail :

A nos familles (parents, frères, sœurs, proches),

A nos amis,

Et à nos professeurs.

Table des matières

Remerciements	1
Dédicaces	2
Introduction général	5
Chapitre 1 : Modèles markoviens et système d'attente	7
1.1 modèles 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	8
1.1.4 Modélisation du système par les équations différentielles de Chapman-Kolmogorov.....	9
1.1.5 Solution stationnaire d'un processus de Markov	9
1.1.6 Caractéristique d'une loi stationnaire	9
1.1.7 Ergodicité d'un processus de Markov.....	10
1.2 Systèmes de files d'attente	11
1.2.1 Introduction	11
1.2.2 Description	11
1.2.3 Caractéristique d'un système d'attente.....	11
1.2.4 Notation de Kendall	12
1.2.5 Etude de système M/M/1	12
1.2.6 Etude de système M/M/1/K	19
1.2.7 Etude de système M/M/1/K/[F]	21
1.2.8 Etude de système M/M/S	24
1.3 Conclusion.....	27
Chapitre 2 : Modélisation d'un système d'attente d'un central téléphonique	28
2.1 Introduction	28
2.2 Description du problème.....	28
2.3 Etude de cas	29
2.4 Position du problème	29
2.5 Détermination des lois	29
2.5.1 loi des arrivées	29
2.5.2 loi de la durée de service	30
2.6 Etude de la stabilité du système	30
2.7 Etude de système M/M/s/K/[F]	31
2.7.1 Etude du processus $(N_t)_{t \geq 0}$	31
2.7.2 Loi du système en régime permanent	32
2.7.3 Performances de la file M/M/s/K/[F]	33
2.7.4 Probabilité de saturation	34

2.8 Etude de la file M/M/s/s/[F]	35
2.8.1 Etude du processus $(N_t)_{t \geq 0}$	35
2.8.2 Loi du système en régie permanent	36
2.8.3 Performances de la file M/M/s/s/[F]	37
2.8.4 Probabilité de saturation	38
2.9 Conclusion.....	39
Chapitre 3 : Simulation du système d'attente du central téléphonique et de ses performances	40
3.1 Introduction	40
3.2 Simulation de la loi des arrivées	40
3.2.1 Générateur de la loi exponentielle	40
3.2.2 Loi des arrivées	41
3.2.3 Graphe de la loi des arrivées.....	41
3.3 Simulation des lois et des performances des files d'attente considérées en fonction des paramètres	42
3.3.1 File d'attente M/M/s/K/[F]	42
3.3.2 File d'attente M/M/s/s/[F]	56
3.4 Conclusion.....	65
Chapitre 4 : Optimisation dans le central téléphonique.....	66
4.1 Introduction.....	66
4.2 File d'attente M/M/s/K/[F]	66
4.2.1 Formulation du problème	66
4.2.2 Optimisation de la probabilité de saturation	66
4.2.3 Illustrations numérique et graphique	69
4.3 File d'attente M/M/s/s/[F]	72
4.3.1 Formulation du problème	72
4.3.2 Optimisation de la probabilité de saturation	72
4.3.3 Illustration numérique et graphique	75
4.4 Conclusion.....	77
Conclusion générale	78
Bibliographie	79

INTRODUCTION GÉNÉRALE

Faire la queue! Voilà une expression malheureusement présente à tout moment de notre existence. Un spécialiste de l'analyse d'attente a déterminé qu'une personne adulte passait au minimum un dixième de son temps à attendre, que ce soit chez un médecin, au guichet de la poste ou d'une banque, en voiture pour faire un plein d'essence ou encore à l'aéroport pour descendre d'un avion,... Mais ces attentes ne se manifestent pas toujours d'une manière physique, par exemple le traitement de données dans un système informatique ou dans les réseaux de télécommunications. Nous attendons en posant de sérieuses questions. Pourquoi cela dure-t-il autant ? Comment faire pour accélérer les choses ?...

Donc, on peut considérer un phénomène d'attente comme un certain nombre d'unités qui se présentent pour se faire servir. Nous allons nous intéresser au cas des réseaux de télécommunications, en particulier les fluctuations de la charge des lignes et des centraux téléphoniques.

Remarquons qu'il s'agit de phénomène stochastique. En effet, les appels "clients" arrivent dans le centre téléphonique de façon aléatoire, suivant par exemple un processus de poisson, chaque appel se dirige aussitôt vers la queue "file" et y prend place ou passe immédiatement au canal "serveur", s'il est libre, pour être transféré à son destinataire. La durée de chaque appel est aussi aléatoire par exemple de loi exponentielle, formant ainsi un système d'attente markovien (c'est-à-dire la loi des arrivées et la loi de service sont markoviennes M/M/.). Ce ne sont évidemment pas les seules formes qui peuvent se présenter, mais ce sont les plus fréquentes et les plus simples à étudier.

A fin de bien analyser l'évolution au cours du temps du système, on construit un modèle mathématique adéquat au phénomène d'attente. Sa simulation permet de structurer et de décrire avec rigueur cette modélisation et on travaillera sous le logiciel MATLAB.

La théorie des files d'attente a une très grande importance en recherche opérationnelle, elle sert à modéliser les systèmes d'attente et à évaluer leurs performances afin de mieux comprendre leurs comportements. L'origine de cette théorie remonte aux premiers travaux du chercheur danois K.Erlang en 1917 sur les réseaux téléphoniques, développée aussi par Engset en 1918 qui a travaillé sur le même problème mais à population finie. Celle-ci a ensuite fait un saut grâce à l'impulsion de nombreux chercheurs comme E.Borel, D.Kendall et surtout A.Kolmogorov qui a introduit la notion "sans mémoire" (chaîne markovienne). Son application s'est ensuite généralisée à divers problèmes après la seconde guerre mondiale comme les problèmes de gestion de stocks et à la télécommunication en général.

On s'intéresse aux appels qui arrivent dans un central téléphonique et qui engendrent parfois la saturation des canaux, c'est-à-dire un client qui passe un appel est rejeté ou bien

mis en attente. Notre problème consiste à optimiser certains paramètres et performances liés aux serveurs pour éviter la congestion des appels entrants vers le central et de garantir ainsi un meilleur service aux clients "abonnés". Il s'agit de déterminer le nombre de serveurs, capacité du système, etc, afin d'avoir un bon rendement. Cela revient à minimiser la probabilité de saturation du système ou le risque de saturation dans ce central téléphonique.

Le mémoire est organisé en quatre chapitres :

- Le premier chapitre décrit les modèles markoviens et les systèmes d'attente markoviens qui seront la base de modélisation de notre problème considéré.
- Le deuxième chapitre est composé de deux parties essentielles modélisation et évaluation des performances du système considéré.
- Le troisième chapitre est dédié à la simulation des ces différents caractéristiques en fonction des différentes données au préalable.
- Le dernier est consacré au problème d'optimisation qui consiste à minimiser le risque de saturation en fonction des différents taux du système d'attente.

Chapitre 1

Modèles markoviens et systèmes d'attente

1.1 Modèles Markoviens

1.1.1 Introduction

Les processus aléatoires décrivent l'évolution d'une grandeur aléatoire en fonction du temps. Mathématiquement, un processus stochastique $\{X_t, t \in T\}$ est une suite de variables aléatoires indexées par le temps t et définies sur un même espace de probabilité (Ω, E, P) , leurs étude s'inspire de la théorie des probabilités.

On va se limiter aux processus de Markov, c'est-à-dire leurs évolution futur ne dépend du passé qu'à travers de sa valeur actuelle.

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 $(X_t)_{t \in T}$ sera dit processus markovien homogène.

Dans ce qui suit, on suppose que $(X_t)_{t \in T}$ est markovien et homogène alors

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

Elle est appelé 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

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

Remarque 1.1

La matrice stochastique $P(t)$ caractérise le processus markovien $(X_t)_{t \in T}$ c'est-à-dire on peut connaître n'importe qu'elle 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 par :

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

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

Pour cela, $\forall j \in E$

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

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

1.1.3 Générateur infinitésimal d'un système markovien

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

$$\lim_{t \rightarrow 0} p_{ij}(t) = p_{ij}(0) = \begin{cases} 1 & \text{si } i = j, \\ 0 & \text{si } i \neq j. \end{cases} \text{ alors}$$

Pour $i \neq j, q_{ij} = \lim_{t \rightarrow 0} \frac{p_{ij}(t)}{t} = p'_{ij}(0)$, quand elle existe.

Pour $i = j, q_{ii} = \lim_{t \rightarrow 0} \frac{p_{ij}(t) - 1}{t} = p'_{ii}(0)$, 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 $(\sum_{j \in E} p_{ij}(t))' = (\sum_{j \in E} (p_{ij}(t))') = 0$.

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 $t=0$:

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

$$\Rightarrow P'(s) = QP(s) \forall i, k \in E.$$

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

$$\begin{cases} P'(t) = QP(t) \\ P(0) = I_E \end{cases}$$

la solution est donnée par

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

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

$$D = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix} \text{ et } P = [V_1, V_2, \dots, V_n]. \text{ Alors la solution est } P(t) = P\Delta(t)P^{-1} \text{ où}$$

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

1.1.5 Solution stationnaire d'un processus de Markov

Définition 1.2

Une loi $\pi(t)$ du 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 $\pi(t) = \pi_0^* = \{P(X_t = i), \forall i \in E\}$.

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 de processus de Markov, c'est-à-dire le système

$\begin{cases} \pi^* Q = 0 \\ \sum_{j \in E} \pi_j = 1 \end{cases}$ 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 cas :

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).
- $P(N_{dt} = k) = \begin{cases} \lambda dt + o(dt) & \text{si } k = 1, \\ 1 - \lambda dt + o(dt) & \text{si } k = 0, \\ o(dt) & \text{si } k \geq 2. \end{cases}$

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.

Sous les hypothèses suivantes :

H₁ : A 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).

H₂ : Ce processus est un processus à accroissements stationnaires et à accroissements indépendants.

H₃ : Au plus un événement 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 $n+1$, mort à la date t , le passage de l'état n à l'état $n-1$.

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

$$p_n = \begin{cases} p_0 = \frac{1}{1 + \sum_{n \geq 1} C_n} & \text{si } n = 0, \\ p_n = C_n p_0 & \text{si } n \geq 1, \end{cases} \text{ avec } C_n = \prod_{i=1}^n \frac{\lambda_{i-1}}{\mu_i}, \text{ à condition que } \sum_{n \geq 1} C_n < +\infty.$$

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 probabilité, 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 et d'engorgement liés aux caractères aléatoires et imprévisibles des événements rencontrés. Nous nous intéressons aux modèles markoviens c'est-à-dire, les arrivées poissonniennes et les durées de services indépendants et de même loi exponentielle.

1.2.2 Description

On considère un système destiné à offrir un service, la personne ou l'objet venant bénéficier de ce service est appelé client et les postes de service sont appelés serveurs ou guichets.

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

Système d'attente : file d'attente + service en cour.

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

Ce dernier sera caractérisé pas la description de :

- La file d'attente (longueur de la file, temps d'attente, ...).
- Serveurs (nombre de clients servis par période d'activité, durée de répit (oisiveté), ...).

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, Gama, ...), elle sera notée L_1 .

2/ Loi des durées de services de chaque client dans chaque guichet D_i (indépendantes de même loi exponentielle), cette loi sera notée L_2 .

3/ Le nombre de serveur dans le système : il sera noté s ($s \geq 1$).

4/ La capacité du système (c'est le nombre maximum de client 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

Elle permet de ramener la description textuelle des caractéristiques d'une file d'attente à une formule symbolique, elle est la définition mathématique d'une file d'attente qui est :

$(L_1, L_2, s, K, \text{FIFO ou autre discipline}, [O] \text{ ou } [F])$.

Remarque 2.1

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

1.2.5 Etude du système M/M/1

Nous abordons dans ce cas un système d'attente à 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 .

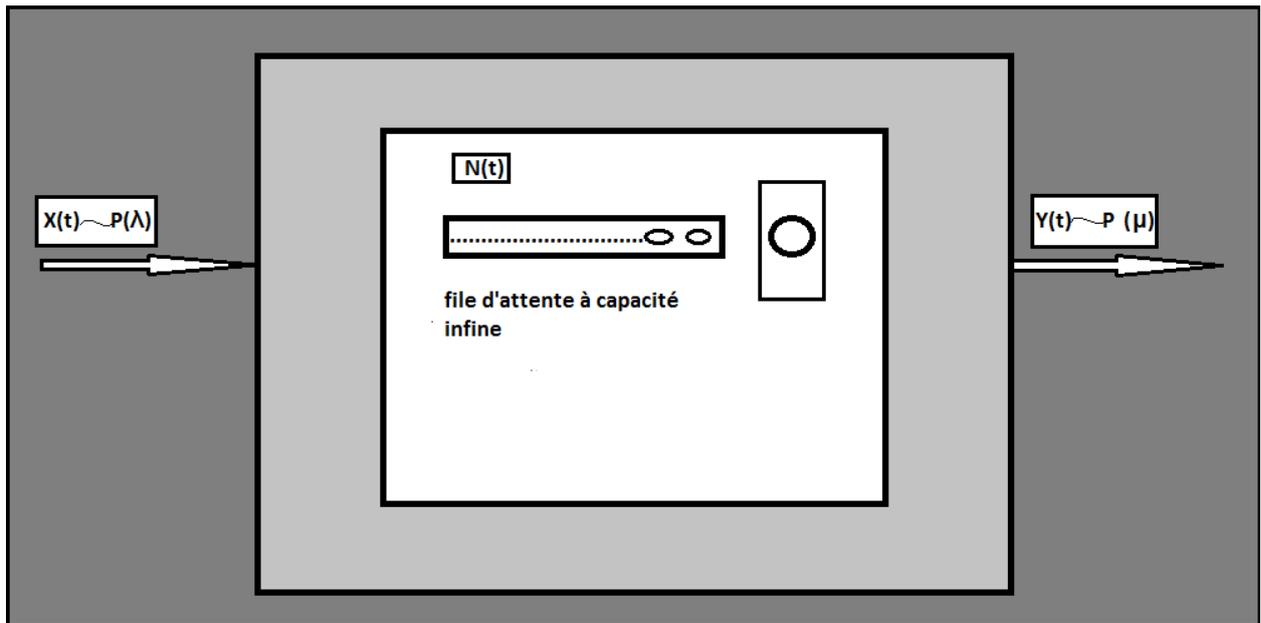


Figure 1.1 Système d'attente (M/M/1)

- (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 sortis 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) défini sur un ensemble infini $E = \{0, 1, \dots\} = \mathbb{N}$, de taux λ_n et μ_n à déterminer :

a. Taux de naissance λ_n

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

b. Taux de mort μ_n

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

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

$$p_n = \begin{cases} \frac{a_n}{1 + \sum_{n \geq 1} a_n} & \text{si } n \geq 1, \\ \frac{1}{1 + \sum_{n \geq 1} a_n} & \text{si } n = 0. \end{cases}$$

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 $\Rightarrow a_n = \rho^n$.

Alors

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

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

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

Donc $p_n = \rho^n(1 - \rho), n \geq 0$ avec $\rho < 1$.

1.2.5.3 Performance 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 client 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

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

$$= \frac{\rho}{1-\rho} = \frac{\frac{\lambda}{\mu}}{1-\frac{\lambda}{\mu}}$$

$$\Rightarrow \eta = \frac{\lambda}{\mu - \lambda}.$$

b. Ecart type σ_t

Afin de préciser le nombre de clients exact présents dans le système à un instant donné, la moyenne à elle seule ne suffit pas. D'où le calcul de l'écart type qui est donné par la formule suivante :

$$\sigma_t = \sqrt{V(N_t)} \text{ avec } V(N_t) = E(N_t^2) - E^2(N_t).$$

On a

$$E(N_t^2) = \sum_{n \geq 0} n^2 p_n = \sum_{n \geq 0} (n(n-1) + n) p_n$$

$$= \sum_{n \geq 0} n(n-1) p_n + \sum_{n \geq 0} n p_n.$$

On a

$$\sum_{n \geq 0} n(n-1) p_n = \sum_{n \geq 0} n(n-1) \rho^n (1-\rho)$$

$$= \rho^2 (1-\rho) \sum_{n \geq 0} n(n-1) \rho^{n-2} = \rho^2 (1-\rho) \sum_{n \geq 0} (\rho^n)'' = \rho^2 (1-\rho) \left(\sum_{n \geq 0} \rho^n \right)''$$

$$= \rho^2 (1-\rho) \left(\frac{1}{1-\rho} \right)'' = \rho^2 (1-\rho) \left(\frac{1}{(1-\rho)^2} \right)'$$

$$= \frac{2\rho^2}{(1-\rho)^2}.$$

Et donc

$$E(N_t^2) = \frac{2\rho^2}{(1-\rho)^2} + \eta = \frac{2\rho^2}{(1-\rho)^2} + \frac{\rho}{1-\rho}.$$

Alors

$$V(N_t) = E(N_t^2) - E^2(N_t) = \frac{2\rho^2}{(1-\rho)^2} + \frac{\rho}{1-\rho} - \left(\frac{\rho}{1-\rho} \right)^2$$

$$= \frac{\rho}{(1 - \rho)^2}.$$

Ainsi

$$\sigma_t = \sqrt{\frac{\rho}{(1 - \rho)^2}} = \frac{\sqrt{\rho}}{1 - \rho}.$$

$$\Rightarrow E(N_t) \in \left[\frac{\rho - \sqrt{\rho}}{1 - \rho}, \frac{\rho + \sqrt{\rho}}{1 - \rho} \right].$$

c. Nombre moyen de clients dans la file (queue) 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

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

Remarque 2.2

On peut calculer le nombre moyen de clients dans la file à la date t avec une autre méthode qui est la suivante.

Soient M_t une variable aléatoire qui compte le nombre de clients dans la file à la t et Z une autre variable qui désigne l'occupation ou non du serveur, alors le nombre de clients dans le système se traduit par:

$$N_t = M_t + Z, \text{ avec } Z \text{ une Bernoulli de paramètre } p.$$

On a

$$Z = \begin{cases} 1 & \text{si le serveur est occupé} \\ 0 & \text{sinon} \end{cases}$$

$$Z \sim \text{Bernoulli}(p), \text{ avec } p = P[Z = 1] = P[N_t > 0] = 1 - P[N_t = 0]$$

$$= 1 - p_0 = \rho.$$

Alors

$$E(M_t) = E(N_t - Z) = E(N_t) - E(Z) = \eta - \rho = \frac{\rho}{1 - \rho} - \rho$$

$$\Rightarrow \eta_q = \frac{\rho^2}{1 - \rho}.$$

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 (s'il y en a), R est une variable aléatoire et \bar{R} sa moyenne.

Une période de répit commence à la fin d'un service d'un client seul dans le système et s'achève lorsqu'un nouveau client arrive ou continue tant qu'aucun client n'arrive.

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 $(X_t)_{t \geq 0} \sim P(\lambda)$, alors il est à accroissements stationnaires, donc

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

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

Alors

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

D'où R suit une loi exponentielle de paramètre λ .

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

Remarque 2.3

Ce résultat est valable en régime permanent et transitoire.

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

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

Soit T la durée globale d'activité théorique. Sur cette période T , le serveur est libre sur une proportion (p) de ce temps avec :

$$p = P[\text{le système est vide}] = P[N_t = 0] = p_0 = 1 - \rho.$$

Car $p_n = \rho^n(1 - \rho)$.

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

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

Remarque 2.4

Ce résultat est valable juste en régime permanent.

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

A cause de l'alternance activité-répétition, on a $A = B$ à une unité près, 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-répétition, on a en moyenne \bar{B} périodes d'activité.

Donc

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

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

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'un client D est $exp(\mu)$, alors $\bar{D} = \frac{1}{\mu}$.

On a alors

$$\bar{N}_c \times \bar{D} = \bar{C} \Rightarrow \bar{N}_c = \frac{\bar{C}}{\bar{D}} = \frac{1}{\frac{1}{\mu} - \frac{1}{\lambda}}$$

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

1.2.6 Etude du système d'attente M/M/1/K

Il s'agit d'un système d'attente qui sert les clients avec un seul serveur suivant leur ordre d'arrivées. Sa capacité est limité 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 que 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 .

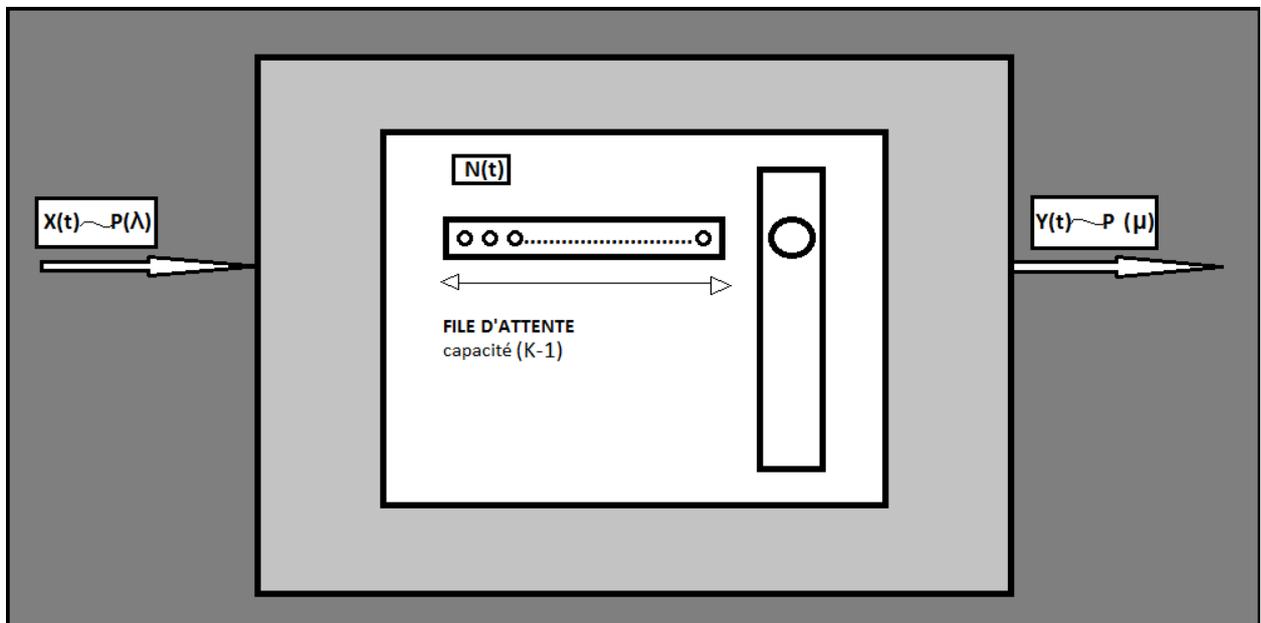


Figure 1.2 Système d'attente (M/M/1/K)

- (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 sortis 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 défini sur un ensemble fini $E = \{0, 1, \dots, k\}$, de taux λ_n et μ_n à déterminer.

a. Taux de naissance λ_n

On a

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

b. Taux de mort μ_n

On a

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

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

$$p_n = a_n \times p_0, \text{ avec, } a_n = \prod_{i=1}^n \frac{\lambda_{i-1}}{\mu_i}, 1 \leq n \leq K.$$

On a aussi

$$\sum_{n=0}^K p_n = p_0 + \sum_{n=1}^K p_n = p_0 \left(1 + \sum_{n=1}^K \prod_{i=1}^n \frac{\lambda_{i-1}}{\mu_i} \right).$$

On a

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

On pose $\frac{\lambda}{\mu} = \rho$.

Alors

$$a_n = \rho^n.$$

Ainsi

$$\sum_{n=0}^K p_n = p_0 \left(1 + \sum_{n=1}^K \rho^n \right) = 1$$
$$\Rightarrow p_0 = \frac{1}{\sum_{n=0}^K \rho^n}.$$

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 :

$$\lambda \neq \mu \Rightarrow \rho \neq 1 \Rightarrow p_0 = \frac{1 - \rho}{1 - \rho^{K+1}}.$$

Ainsi

$$p_n = \rho^n \times p_0.$$
$$\Rightarrow p_n = \frac{\rho^n (1 - \rho)}{1 - \rho^{K+1}}, 0 \leq n \leq K.$$

2^{ème} cas :

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

On a

$$p_n = \rho^n \times p_0 \text{ et } p_0 = \frac{1}{\sum_{n=0}^K \rho^n} = \frac{1}{K+1}.$$
$$\Rightarrow p_n = \frac{1}{K+1}, 0 \leq n \leq K.$$

Remarque 2.5

Cette probabilité est une probabilité uniforme sur $\{0, 1, \dots, K\}$.

1.2.7 Etude du système d'attente (M/M/1/K/[F])

On considère un système d'attente à un seul serveur, qui sert les clients dans leur ordre d'arrivées, on suppose que les clients à servir sont des particuliers, c'est-à-dire qu'ils sont limités à un nombre K et la capacité du système est aussi limitée à 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 .

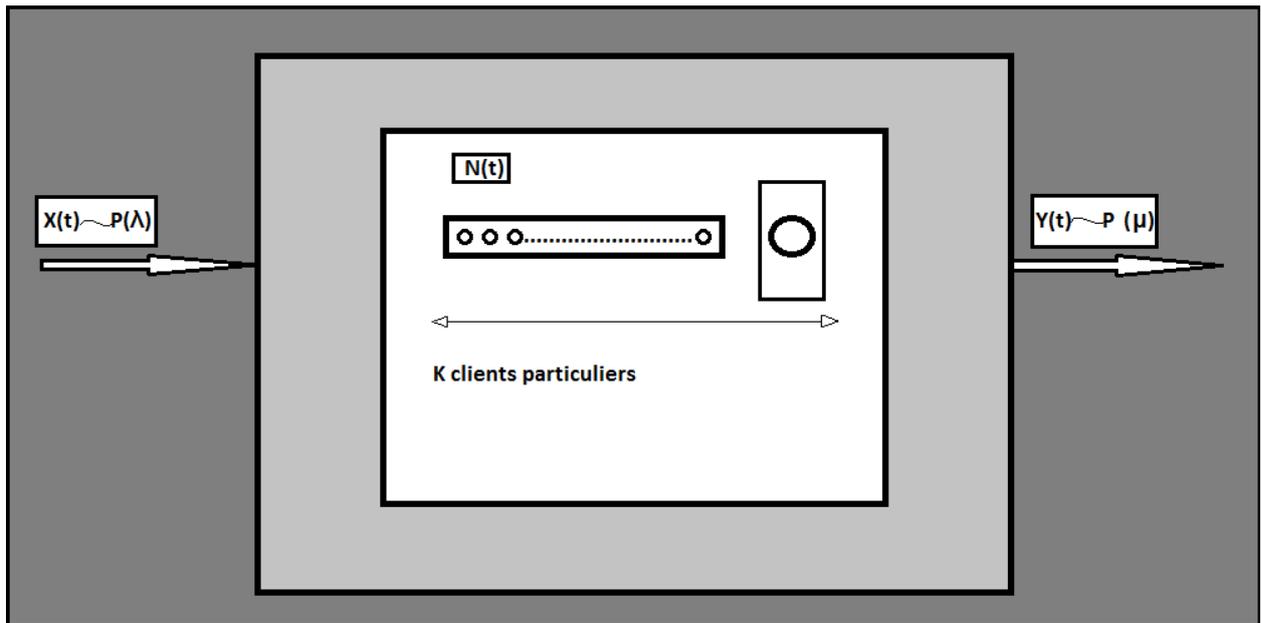


Figure 1.3 Système d'attente (M/M/1/K/[F])

- (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 sortis 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

$$\begin{aligned} \lambda_n dt + o(dt) &= P[N_{t+dt} = n + 1 / N_t = n] \\ &= P[\text{avoir une entrée parmi les } (K - n) \text{ restants}] \\ &= C_{K-n}^1 P[X_{t+dt} = X_t + 1] = \frac{(K - n)!}{1! (K - n - 1)!} (\lambda dt + o(dt)) = (K - n) \lambda dt + o(dt) \\ \Rightarrow \lambda_n &= \begin{cases} (K - n) \lambda & \text{si } n = \overline{0, K - 1}, \\ 0 & \text{si } n \geq K. \end{cases} \end{aligned}$$

b. Taux de mort μ_n

On a

$$\begin{aligned}
\mu_n dt + o(dt) &= P[N_{t+dt} = n/N_t = n + 1] \\
&= P[\text{avoir une sortie (client qui a terminé son service)}] \\
&= P[Y_{t+dt} = Y_t + 1] = \mu dt + o(dt) \\
\Rightarrow \mu_n &= \begin{cases} 0 & \text{si } n = 0, \\ \mu & \text{si } n = 1..K, \\ 0 & \text{si } n > K. \end{cases}
\end{aligned}$$

1.2.7.2 Loi du système en régime permanent (stable)

Comme c'est une somme finie, car la capacité du système est 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

$$p_n = \begin{cases} a_n \times p_0 & \text{si } 1 \leq n \leq K, \\ p_0 & \text{si } n = 0. \end{cases}$$

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

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

On pose $\frac{\lambda}{\mu} = \rho$.

Ainsi

$$p_n = \begin{cases} \frac{K!}{(K-n)!} \rho^n p_0 & \text{si } n = 1, 2, \dots, K, \\ \frac{1}{1 + \sum_{n=1}^K \frac{K!}{(K-n)!} \rho^n} & \text{si } n = 0. \end{cases}$$

Alors

$$p_n = \begin{cases} n! C_K^n \rho^n p_0 & \text{si } n = 1, 2, \dots, K, \\ \frac{1}{\sum_{n=0}^K n! C_K^n \rho^n} & \text{si } n = 0. \end{cases}$$

D'où

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

1.2.8 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 de toutes les autres et ont une même loi exponentielle de paramètre μ .

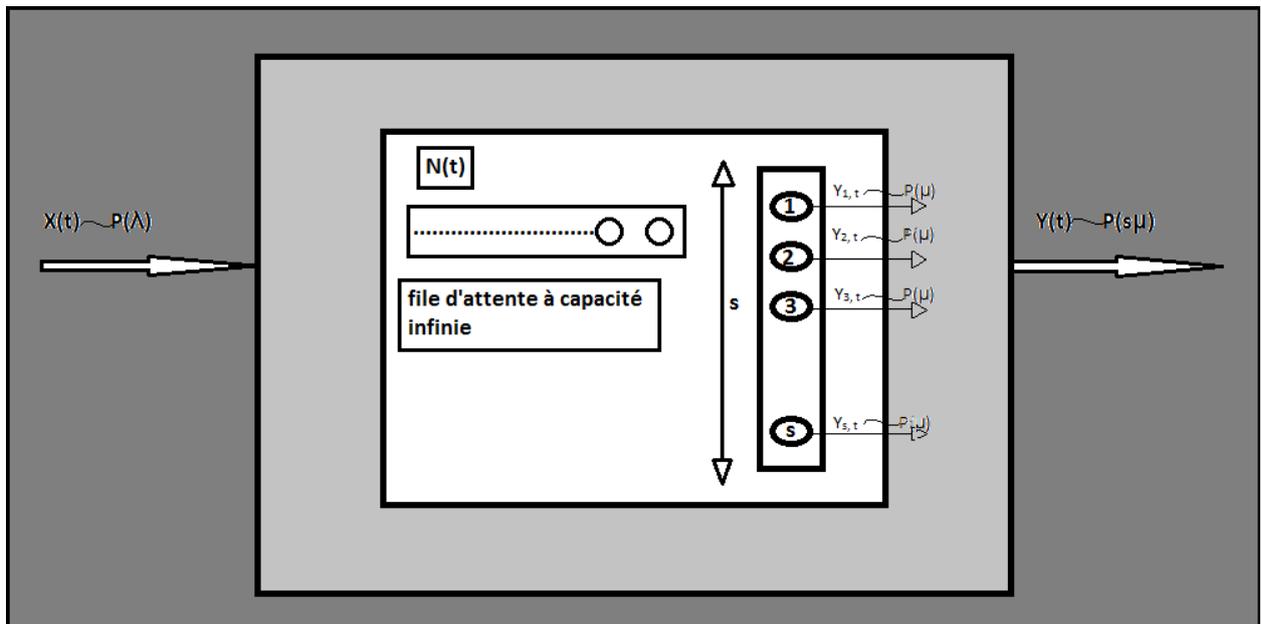


Figure 1.5 Système d'attente (M/M/s)

- (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 sortis 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

➤ Si $1 \leq n \leq s$, alors

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

➤ Si $n > s$, alors

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

D'où

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

1.2.8.2 Loi du système en régime permanent (stable)

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

On a

$$p_n = \begin{cases} a_n \times p_0 \text{ où } a_n = \prod_{i=1}^n \frac{\lambda_{i-1}}{\mu_i}, & \text{si } n \geq 1, \\ \frac{1}{1 + \sum_{n \geq 1} a_n} & \text{si } n = 0. \end{cases}$$

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

$$p_n = a_n \times p_0 = \left(\prod_{i=1}^n \frac{\lambda_{i-1}}{\mu_i} \right) \times p_0 = \frac{\lambda_0 \lambda_1 \dots \lambda_{n-1}}{\mu_1 \mu_2 \dots \mu_n} p_0 = \frac{\lambda \cdot \lambda \dots \lambda}{1 \mu 2 \mu \dots n \mu} p_0 = \frac{\lambda^n}{n! \mu^n} p_0$$

$$\Rightarrow p_n = \frac{\rho^n}{n!} p_0, \text{ avec } \rho = \frac{\lambda}{\mu}.$$

➤ Si $n > s$

On a

$$p_n = a_n \times p_0 = \left(\prod_{i=1}^n \frac{\lambda_{i-1}}{\mu_i} \right) \times p_0 = \left(\prod_{i=1}^s \frac{\lambda_{i-1}}{\mu_i} \prod_{i=s+1}^n \frac{\lambda_{i-1}}{\mu_i} \right) \times p_0$$

$$= \left(\frac{\lambda_0 \lambda_1 \dots \lambda_{s-1}}{\mu_1 \mu_2 \dots \mu_s} \times \frac{\lambda_s \lambda_{s+1} \dots \lambda_{n-1}}{\mu_{s+1} \mu_{s+2} \dots \mu_n} \right) p_0 = \left(\frac{\lambda \cdot \lambda \dots \lambda}{1 \mu 2 \mu \dots s \mu} \times \frac{\lambda \cdot \lambda \dots \lambda}{s \mu s \mu \dots s \mu} \right) p_0 = \frac{\lambda^n}{s! s^{n-s} \mu^n} p_0$$

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

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

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

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

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

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

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

$$\Rightarrow p_0 = \frac{1}{1 + \sum_{n=1}^s \frac{\rho^n}{n!} + \frac{\rho^s}{s!} \frac{\rho}{s - \rho}}.$$

Finalement si $\lambda < s\mu$ alors

$$p_n = \begin{cases} \frac{1}{1 + \sum_{n=1}^s \frac{\rho^n}{n!} + \frac{\rho^s}{s!} \frac{\rho}{s - \rho}} & \text{si } n = 0, \\ \frac{\rho^n}{n!} p_0 & \text{si } 1 \leq n \leq s, \\ \frac{\rho^n}{s! s^{n-s}} p_0 & \text{si } n > s. \end{cases}$$

1.3 Conclusion

Nous avons présenté dans ce chapitre deux notions qui sont les modèles markoviens et les files d'attente markoviennes, ces deux théories nous permettront de modéliser et d'étudier le problème considéré dans le prochain chapitre.

Chapitre 2

Modélisation du système d'attente d'un central téléphonique

2.1 Introduction

Dans ce chapitre, nous développons des modèles mathématiques nécessaires à l'étude et l'analyse des performances du système d'attente au niveau d'un central téléphonique. Ce dernier joue un rôle très important dans la gestion des appels chez un opérateur. La modélisation est une représentation logique et mathématique du comportement du système réel dans un contexte donné. Le modèle prend dans la majorité des cas la forme d'une série d'hypothèses concernant le fonctionnement du système et qui sont exprimées à l'aide de relations mathématiques.

2.2 Description du problème

Considérons un central téléphonique qui regroupe les lignes d'un ensemble d'abonnés fini dans une ville. Il est clair que ce central ne possède pas autant de canaux allant vers le réseau que de lignes allant vers les clients, soit alors s le nombre de ces canaux "serveurs" et K le nombre de ces abonnés "clients" avec ($K > s$).

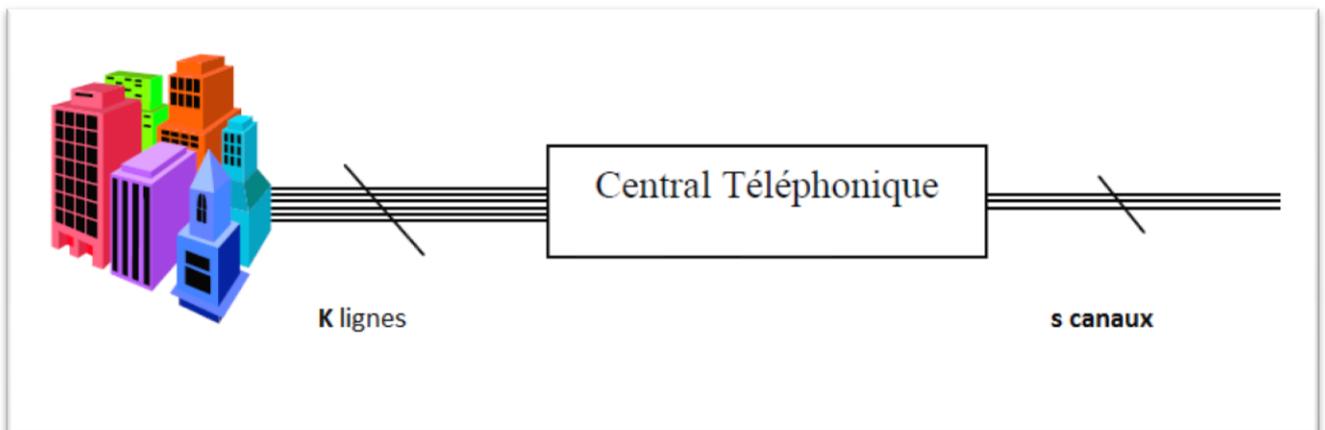


Figure 2.1 Description d'un central téléphonique

En pratique, un client qui passe un appel peut être directement servi s'il existe un canal libre, sinon il est mis en attente, ainsi le système est modélisé par une file d'attente $(L_1, L_2, s, K, \text{FIFO}, [F])$ ou bien l'appel est rejeté et dans ce cas, il n'existe pas de file ou de queue alors le système est modélisé par une file d'attente de type $(L_1, L_2, s, s, \text{FIFO}, [F])$.

2.3 Etude de cas

Après une visite à la direction d'Algérie télécom de la wilaya de Tizi ouzou et un entretien avec les responsables du service de maintenance sur les appels téléphoniques, on a constaté que les appels arrivent de manière aléatoire. On suppose qu'ils suivent un processus de Poisson de taux λ et que les durées des appels sont elles aussi aléatoires qu'on suppose aussi indépendantes, de même loi exponentielle de paramètre μ .

Les services de maintenance sont confrontés au problème de saturation des canaux, ce qui engendre le désabonnement des clients et la baisse des profits de l'entreprise.

Notre objectif est d'optimiser les performances du système et de minimiser le risque de saturation de ce dernier.

2.4 Position du problème

Il s'agit d'un problème de modélisation dans un central téléphonique, où des appels arrivent d'une manière aléatoire afin d'être transmis à leurs destinataires.

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

A : représente le flux d'arrivées des appels au central téléphonique, 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 nième appel. On s'intéresse aux durées entre deux arrivées successives (inter-arrivées), $(T_n = t_n - t_{n-1})_{n \geq 1}$ qui sont exponentielles de paramètre λ .

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

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

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

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

E : La nature du système, dans notre cas, seul les abonnés ont un accès au service de l'opérateur, autrement dit le système est fermé.

2.5 Déterminations des lois

2.5.1 Loi des arrivées

Pour délimiter la loi des arrivées des appels entrants dans un central téléphonique. 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 central téléphonique et K la capacité maximale du système (nombre d'abonnés utilisant le central).

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 entrant au cours d'un intervalle de temps $[0, t]$, $N(t) = \sup\{n, t_n < t\} = \sum_{i \geq 1} 1_{\{t_i \leq t\}}$.

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 le processus $\{N(t), t \geq 0\}$ est un processus de Poisson de taux λ .

Remarque 1

Si pendant la durée t , $n(t)$ arrivées se produisent, alors l'intensité du flux arrivant ou le taux d'arrivées est donné par: $\lambda = \lim_{t \rightarrow \infty} \frac{n(t)}{t}$.

2.5.2 Loi de la durée de service

Pour modéliser les durées des appels, on désigne par D_i la durée pendant laquelle le canal i ($i=1, 2, \dots, n$ et $n \leq s$) demeure occupé par un appel. Puisque on a supposé que les canaux fournissent le même service, il est assez naturel d'admettre que 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).

Remarque 2

Si D désigne la durée entre deux fins d'appels, elle est égale à $\text{Min}(D_1, D_2, \dots, D_n)$, il s'ensuit que la distribution de D est une distribution exponentielle de paramètre $n\mu$.

2.6 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 condition de stationnarité, étant donné la capacité finie du système, c'est-à-dire que le système finit par se stabiliser avec le temps.

2.7 Etude du système M/M/S/K/ [F], (K>S)

C'est un système d'attente fermé, c'est-à-dire que le nombre de clients ayant accès au service est limité à K clients particuliers, sa capacité est elle aussi égale à K et il comporte S serveurs.

Les arrivées sont supposées poissonniennes de taux λ .

Les durées de service qui sont les durées d'appels sont supposées indépendantes, de même loi exponentielle de paramètre μ .

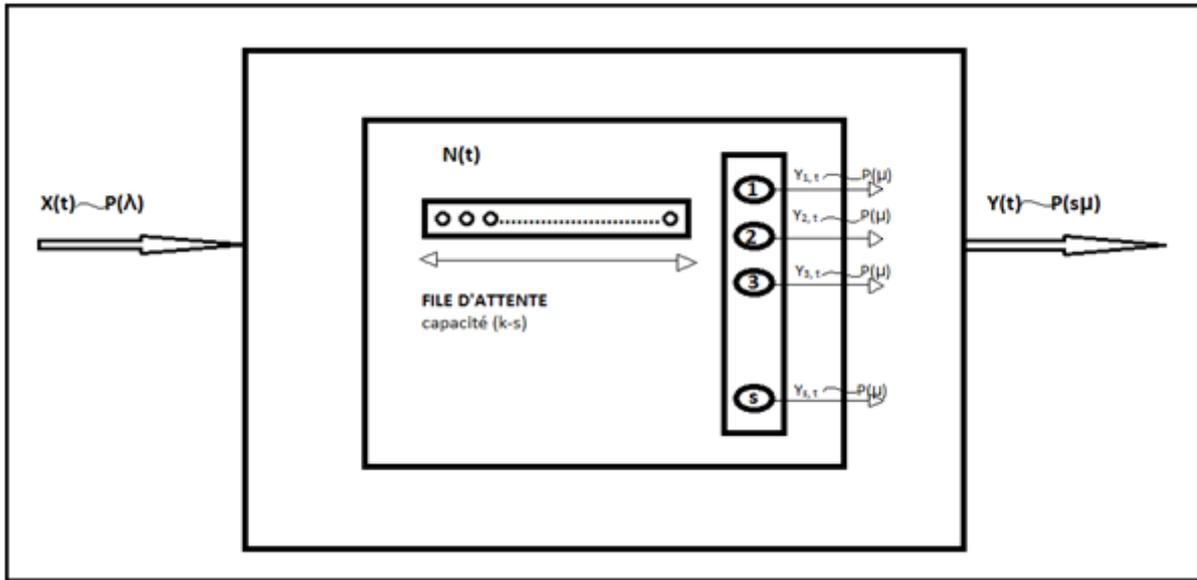


Figure 2.2 Système d'attente (M/M/s/K/[FIFO] / [F]).

- (X_t) est un processus de comptage, il est poissonien de taux λ , il compte le nombre d'appels qui arrivent dans le central téléphonique 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 canal 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 $(N_t)_{t \geq 0}$.

2.7.1 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 $\{0, 1, 2, \dots, K\}$, de taux λ_n et μ_n à déterminer :

a) Taux de naissance λ_n :

On a

$$\begin{aligned} \lambda_n dt + o(dt) &= P(N_{t+dt} = n + 1 / N_t = n) = C_{K-n}^1 P(X_{t+dt} = X_t + 1) \\ &= \frac{(K-n)!}{1! (K-n-1)!} \lambda dt + o(dt) = (K-n) \lambda dt + o(dt) \end{aligned}$$

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

b) Taux de mort μ_n :

On a

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

➤ Si $1 \leq n \leq s$

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

➤ Si $s < n \leq K$

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

D'où

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

2.7.2 Loi du système en régime permanent (stable)

On note $p_n = P[N_t = n], 0 \leq n \leq K$.

$$\text{On a : } p_n = \begin{cases} a_n p_0 & \text{si } 1 \leq n \leq K, \\ p_0 & \text{si } n = 0. \end{cases}$$

$$\text{Avec : } a_n = \prod_{i=1}^n \left(\frac{\lambda_{i-1}}{\mu_i} \right) \text{ et } p_0 = \frac{1}{1 + \sum_{n=1}^K a_n}.$$

Déterminons (a_n) :

➤ Si $1 \leq n \leq s$

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

$$= C_K^n \rho^n, \text{ avec } \rho = \frac{\lambda}{\mu}.$$

➤ Si $s < n \leq K$

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

Déterminons p_0 :

$$p_0 = \frac{1}{1 + \sum_{n=1}^K a_n} = \frac{1}{1 + \sum_{n=1}^s C_K^n \rho^n + \sum_{n=s+1}^K \frac{C_K^n n! \rho^n}{s! s^{n-s}}}$$

$$\Rightarrow p_0 = \frac{1}{\sum_{n=0}^s C_K^n \rho^n + \sum_{n=s+1}^K \frac{C_K^n n! \rho^n}{s! s^{n-s}}}$$

Donc

$$p_n = \begin{cases} C_K^n \rho^n p_0 & \text{si } 1 \leq n \leq s, \\ \frac{C_K^n n! \rho^n}{s! s^{n-s}} p_0 & \text{si } s < n \leq K, \\ \frac{1}{\sum_{n=0}^s C_K^n \rho^n + \sum_{n=s+1}^K \frac{C_K^n n! \rho^n}{s! s^{n-s}}} & \text{si } n = 0. \end{cases}$$

2.7.3 Performances de la file M/M/s/K/ [F]

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 d'un appel dans le système, et \bar{W} sa moyenne.
- W_q : temps d'attente d'un appel avant connexion, et \bar{W}_q sa moyenne.

2.7.3.1 Nombre moyen d'appels en cours et en attente de connexion à l'instant t

$$\begin{aligned} \eta = E(N_t) &= \sum_{n=0}^K n p_n = \sum_{n=1}^s n p_n + \sum_{n=s+1}^K n p_n \\ &= \sum_{n=1}^s n C_K^n \rho^n p_0 + \sum_{n=s+1}^K n \frac{K!}{(K-n)! s! s^{n-s}} \rho^n p_0 \\ &= \sum_{n=1}^s n \frac{K!}{n! (K-n)!} \rho^n p_0 + \sum_{n=s+1}^K n \frac{K!}{(K-n)! s! s^{n-s}} \rho^n p_0 \\ &= \sum_{n=1}^s \frac{K!}{(n-1)! (K-n)!} \rho^n p_0 + \sum_{n=s+1}^K s^{s-1} \frac{K!}{(K-n)! (s-1)! s^n} n \rho^n p_0 \\ &= K! p_0 \left(\sum_{n=1}^s \frac{1}{(n-1)! (K-n)!} \rho^n + \frac{s^{s-1}}{(s-1)!} \sum_{n=s+1}^K \frac{1}{(K-n)! s^n} n \rho^n \right). \end{aligned}$$

2.7.3.2 Nombre moyen d'appels en attente de connexion à l'instant t

$$\begin{aligned}\eta_q &= \sum_{h=0}^{K-s} h P(N_t = s+h) = \sum_{h=1}^{K-s} h p_{s+h} = \sum_{h=1}^{K-s} h \frac{K!}{s! (K-s-h)!} \frac{\rho^{s+h}}{s^h} p_0 \\ &= \frac{K!}{s!} \rho^s p_0 \sum_{h=1}^{K-s} \frac{h}{(K-s-h)!} \left(\frac{\rho}{s}\right)^h = \frac{K!}{s!} \rho^s p_0 \sum_{h=1}^{K-s} \frac{h}{(K-s-h)!} (\bar{\rho})^h, \\ &\text{avec } \bar{\rho} = \frac{\rho}{s}.\end{aligned}$$

2.7.3.3 Durée moyenne d'un appel dans le système

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

$$\begin{aligned}\Lambda &= \sum_{i=0}^K \lambda_i p_i = \sum_{i=0}^K (K-i) \lambda p_i = \lambda \sum_{i=0}^K (K-i) p_i = \lambda \left(\sum_{i=0}^K K p_i - \sum_{i=0}^K i p_i \right) \\ &= \lambda \left(K \sum_{i=0}^K p_i - \sum_{i=0}^K i p_i \right).\end{aligned}$$

On a déjà $\eta = \sum_{i=0}^K i p_i$ et $\sum_{i=0}^K p_i = 1$ alors $\Lambda = \lambda(K - \eta)$.

Donc $\bar{W} = \frac{\eta}{\Lambda} = \frac{\eta}{\lambda(K-\eta)}$.

2.7.3.4 Temps d'attente moyen d'un appel avant connexion

On utilise la formule de Little : $\bar{W}_q = \frac{\eta_q}{\Lambda}$ avec $\Lambda = \sum_{i=0}^K \lambda_i p_i = \lambda(K - \eta)$.

Ainsi

$$\bar{W}_q = \frac{\eta_q}{\Lambda} = \frac{\eta_q}{\lambda(K - \eta)}.$$

2.7.4 Probabilité de saturation

$$\begin{aligned}\pi(\rho, s, K) &= P(N_t \geq s) = \sum_{n=s}^K P(N_t = n) = \sum_{n=s}^K p_n = \sum_{n=s}^K \frac{K!}{s! (K-n)!} \frac{\rho^n}{s^{n-s}} \\ &= \frac{K!}{s!} \rho^s p_0 \sum_{n=s}^K \frac{1}{(K-n)!} \frac{\rho^{n-s}}{s^{n-s}} = \frac{K!}{s!} \rho^s p_0 \sum_{n=s}^K \frac{\left(\frac{\rho}{s}\right)^{n-s}}{(K-n)!}.\end{aligned}$$

On pose $h=n-s$ et $\tilde{\rho} = \left(\frac{\rho}{s}\right)$ on obtient

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

2.8 Etude de la file M/M/s/s/ [F]

Il s'agit d'un système fermé, on suppose que K clients ont accès au service, et ils 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, ($K > s$).

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

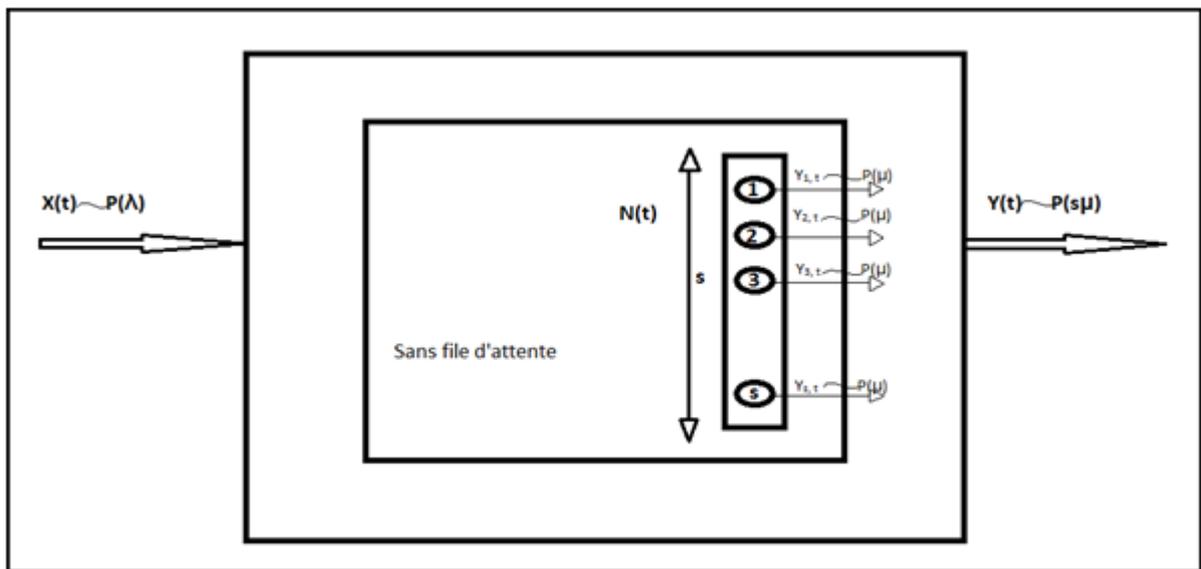


Figure 2.3 Système d'attente (M/M/s/s/[FIFO]/[F]).

- (X_t) est un processus de comptage, il est poissonien de taux λ , il compte le nombre d'appels qui arrivent dans le central téléphonique 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 canal 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 $(N_t)_{t \geq 0}$.

2.8.1 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

$$\begin{aligned} \lambda_n dt + o(dt) &= P(N_{t+dt} = n + 1 / N_t = n) = C_{K-n}^1 P(X_{t+dt} = X_t + 1) \\ &= \frac{(K-n)!}{1!(K-n-1)!} \lambda dt + o(dt) = (K-n)\lambda dt + o(dt) \end{aligned}$$

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

b) Taux de mort μ_n :

On a

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

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

2.8.2 Loi du système en régime permanent (stable)

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

$$\text{On a } p_n = \begin{cases} a_n p_0 & \text{si } 1 \leq n \leq s, \\ p_0 & \text{si } n = 0. \end{cases}$$

$$\text{Avec } a_n = \prod_{i=1}^n \left(\frac{\lambda_{i-1}}{\mu_i} \right) \text{ et } p_0 = \frac{1}{1 + \sum_{n=1}^s a_n}.$$

Déterminons (a_n) :

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

Déterminons p_0 :

On a

$$p_0 = \frac{1}{1 + \sum_{n=1}^s a_n} = \frac{1}{1 + \sum_{n=1}^s C_K^n \rho^n} \Rightarrow p_0 = \frac{1}{\sum_{n=0}^s C_K^n \rho^n}.$$

D'où

$$p_n = \begin{cases} C_K^n \rho^n p_0 & \text{si } 1 \leq n \leq s, \\ \frac{1}{\sum_{n=0}^s C_K^n \rho^n} & \text{si } n = 0. \end{cases}$$

Donc la loi du système est $p_n = C_K^n \rho^n p_0$ si $0 \leq n \leq s$ et $p_0 = \frac{1}{\sum_{n=0}^s C_K^n \rho^n}$.

$$= \begin{cases} \frac{C_K^n \rho^n}{\sum_{n=0}^s C_K^n \rho^n} & \text{si } 0 \leq n \leq s, \\ 0 & \text{sinon.} \end{cases}$$

Remarque 3

Si $K = s$, alors

$$p_0 = \frac{1}{(1 + \rho)^s}.$$

D'ou

$$p_n = \begin{cases} \frac{C_K^n \rho^n}{(1 + \rho)^s} & \text{si } 1 \leq n \leq s, \\ \frac{1}{(1 + \rho)^s} & \text{si } n = 0. \end{cases}$$

2.8.3 Performances du système M/M/s/s/ [F]

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 d'un appel dans le système, et \bar{W} sa moyenne.
- W_q : temps d'attente d'un appel avant connexion, et \bar{W}_q sa moyenne.

2.8.3.1 Nombre moyen d'appels en cours et en attente de connexion à l'instant t

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

$$\begin{aligned}\Lambda &= \sum_{i=0}^s \lambda_i p_i = \sum_{i=0}^s \mu_i p_i \Leftrightarrow \sum_{i=0}^s (K-i)\lambda p_i = \sum_{i=0}^s i\mu p_i \Leftrightarrow \lambda \sum_{i=0}^s (K-i)p_i = \mu \sum_{i=0}^s i p_i \\ &\Leftrightarrow \lambda \left(K \sum_{i=0}^s p_i - \sum_{i=0}^s i p_i \right) = \mu \sum_{i=0}^s i p_i\end{aligned}$$

Comme $\eta = \sum_{i=0}^s i p_i$ et $\sum_{i=0}^s p_i = 1$ alors

$$\lambda(K - \eta) = \mu\eta \Leftrightarrow \lambda K = (\mu + \lambda)\eta \Leftrightarrow \eta = \frac{K\lambda}{\mu + \lambda}.$$

2.8.3.2 Durée moyenne d'un appel dans le système

Soit W le temps d'attente dans le système.

On utilise la formule de Little, $\bar{W} = E(W) = \frac{\eta}{\Lambda}$ avec $\Lambda = \sum_{i=0}^s \lambda_i p_i = \sum_{i=0}^s \mu_i p_i$,

$$\Lambda = \sum_{i=0}^s \mu_i p_i = \sum_{i=0}^s i\mu p_i = \mu \sum_{i=0}^s i p_i = \mu\eta$$

Donc

$$\bar{W} = \frac{\eta}{\Lambda} = \frac{\eta}{\mu\eta} \Rightarrow \bar{W} = \frac{1}{\mu}.$$

Remarque 4

Comme on a supposé que le système n'admet pas de file d'attente « queue », alors 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 sont nuls.

2.8.4 probabilité de saturation

$$\begin{aligned}\pi(\rho, s, K) &= P(N_t \geq s) = \sum_{n=s}^K P(N_t = n) = P(N_t = s) + \sum_{n=s+1}^K P(N_t = n) \\ &= P(N_t = s), \quad \text{car } P(N_t = n) = 0, s+1 \leq n \leq K\end{aligned}$$

D'où

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

Remarque 5

Dans ce cas, cette probabilité représente aussi la probabilité de rejet d'appels.

2.9 Conclusion

On s'est intéressé à la modélisation de deux types de files d'attente représentant le problème d'attente dans un central téléphonique. L'une représente le cas où le système est sans attente ($M/M/s/s/[F]$) et l'autre avec attente ($M/M/s/K/[F]$). Ces deux modèles vont être simulés dans le chapitre suivant afin de bien comprendre leurs évolutions en fonction des différents paramètres λ , μ et s .

Chapitre 3

Simulation du système d'attente du central téléphonique et de ses performances

3.1 Introduction

La modélisation analytique à l'aide de la théorie des files d'attente a été évoquée dans le chapitre précédent. Cependant, bien souvent, on est confronté à des réalisations ou à des projets de réalisations tellement spécifiques que les outils mathématiques ne suffisent pas à résoudre nos problèmes, au moins en première étape de modélisation. La simulation sera alors utilisée.

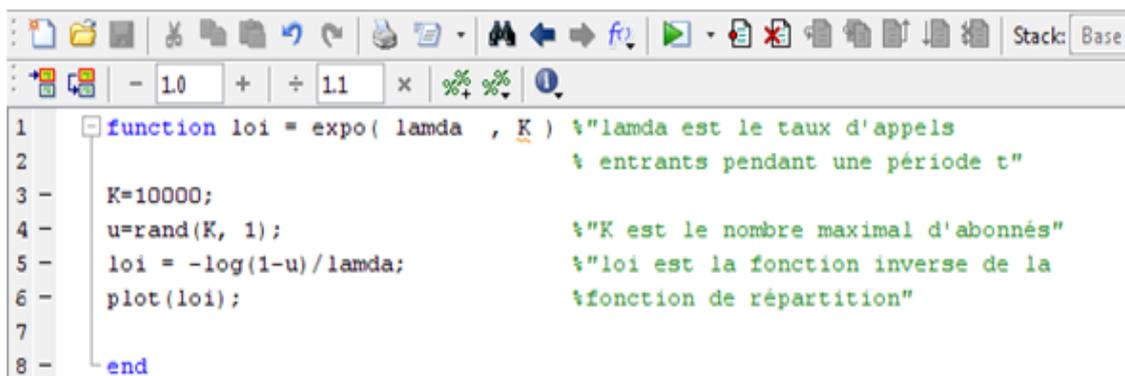
La simulation consiste à reproduire d'une manière synthétique le comportement dynamique des systèmes par un logiciel approprié, elle permettra aussi leurs expérimentations et évaluation. Dans notre cas, nous simulons une file d'attente en utilisant des distributions de probabilité pour générer au hasard les arrivées et les services. En observant comment varie la file et le temps d'attente, nous pouvons en déduire des estimés pour la longueur moyenne de la file et le temps moyen d'attente.

Ainsi les modèles de simulation peuvent être programmés dans une variété de langage, qui est en quelques sortes l'interface entre la machine et son utilisateur. On utilise le logiciel de calcul scientifique (solveur) MATLAB.

3.2 Simulation de la loi des arrivées

3.2.1 Générateur de la loi exponentielle

On va déterminer le nombre d'événement $N(t)=\max \{n ; t_n \leq t\}$, qui suit une loi de Poisson de paramètre λt . On utilise la fonction inverse de la fonction de répartition de la loi exponentielle $F(t) = 1 - e^{-\lambda t}$ afin de générer son générateur avec le code MATLAB suivant :



```
1 function loi = expo( lamda , K ) %"lamda est le taux d'appels
2                               % entrants pendant une période t"
3 - K=10000;
4 - u=rand(K, 1);                %"K est le nombre maximal d'abonnés"
5 - loi = -log(1-u)/lamda;        %"loi est la fonction inverse de la
6 - plot(lois);                  %fonction de répartition"
7
8 - end
```

Figure 3.1 Programme de la loi exponentielle

3.2.2 Loi des arrivées

```
1 function X = poisson( d, f, lamda )
2 -     K=10000;                               % d est l'instant de debut
3 -     y=expo(lamda,K);                       % f est l'instant de fin
4 -     i=1;                                   % y est la fonction exponentielle
5 -     X(i)=d+y(1);
6 -     while X(i)<f
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 -        title('loi des arrivées d"appels');
13 -    end
14 -    X(i)=f;
15 -
16 - end
```

Figure 3.2 Programme de la loi des arrivées

Pour bien comprendre l'évolution des arrivées d'appels au central téléphonique, on a choisi les paramètres suivants : $\lambda=10$ appels par minute, $d=0$ l'instant de début d'étude et $f=400$ minute, l'instant de fin.

3.2.3 Graphe de la loi des arrivées

Le résultat de la simulation est

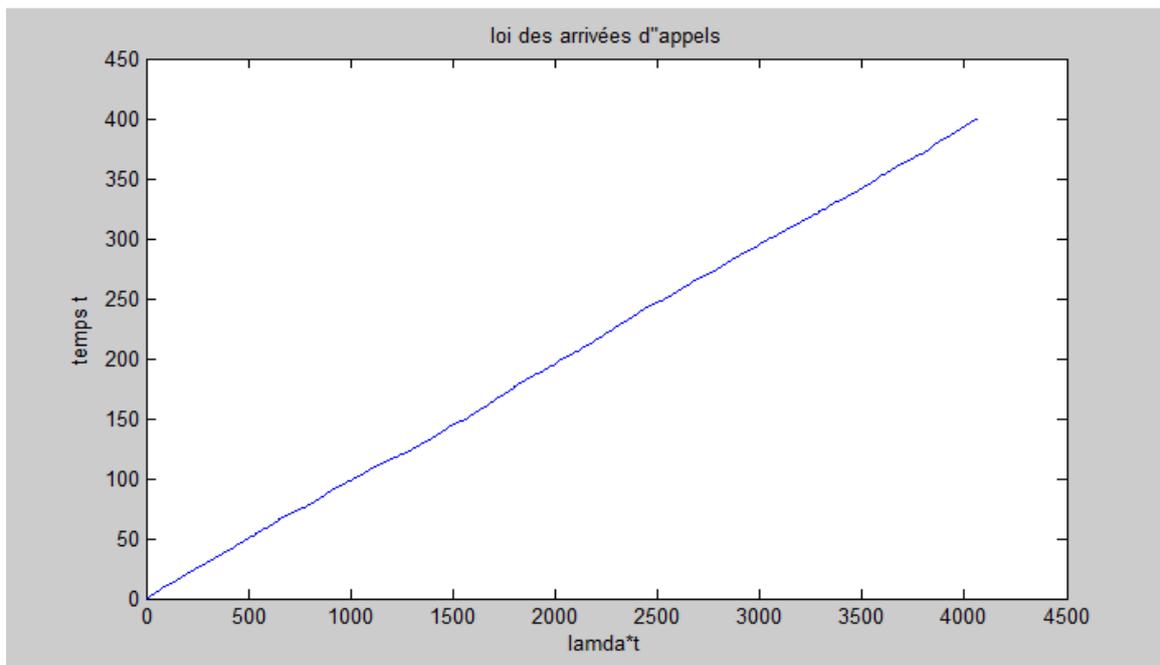


Figure 3.3 Evolution de la loi des arrivées

La figure ci-dessus décrit la variation de la loi des arrivées d'appels (arrivées poissonniennes) pendant un intervalle de temps [0, 400] en minutes.

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

3.3.1 File d'attente M/M/s/K/[F]

3.3.1.1 Simulation de la loi de N_t

On s'intéresse à l'évolution de la loi de $(N_t)_{t \geq 0}$ en fonction du taux des arrivées d'appels au central téléphonique λ , et supposons que le nombre d'abonnés K est égal à 100, le taux des durées d'appels μ est égal à 20 et le nombre de serveurs s égal à 25.

Le code sous MATLAB de la loi de N_t

```

1  function pntsk=pnsk(n,z)
2  mu=20;k=100;s=25;
3  if n<k
4      c1=(fact(k)*(s^s)/fact(s));
5  for lamda=0:z
6      sk1=0;
7      sk2=0;
8      com1=(fact(k)/(fact(n)*fact(k-n)))*(lamda/mu)^n;
9      com2=c1*(1/fact(k-n))*(lamda/(s*mu))^n;
10     for i=1:s
11         sk1=sk1+(fact(k)/(fact(i)*fact(k-i)))*((lamda/mu)^i);
12     end
13     for j=s+1:k
14         sk2=sk2+(c1*(1/fact(k-j))*(lamda/(s*mu))^j);
15     end
16     if n<=s
17         pn(1,lamda+1)=com1/(1+sk1+sk2);
18         pntsk=pn;
19     else if n>s && n<=k
20         pn(1,lamda+1)=com2/(1+sk1+sk2);
21         pntsk=pn;
22     end
23     end
24 end
25 lambda=0:z;
26 plot(lambda,pntsk)
27 xlabel('lamda')
28 ylabel('Pn')
29 else
30     disp('n n"appartient pas à [0 k] donnez une autre valeur')
31 end
32 end

```

Figure 3.4 Programme de la loi de N_t

Les graphes obtenus après simulation sont :

Pour $n=0$

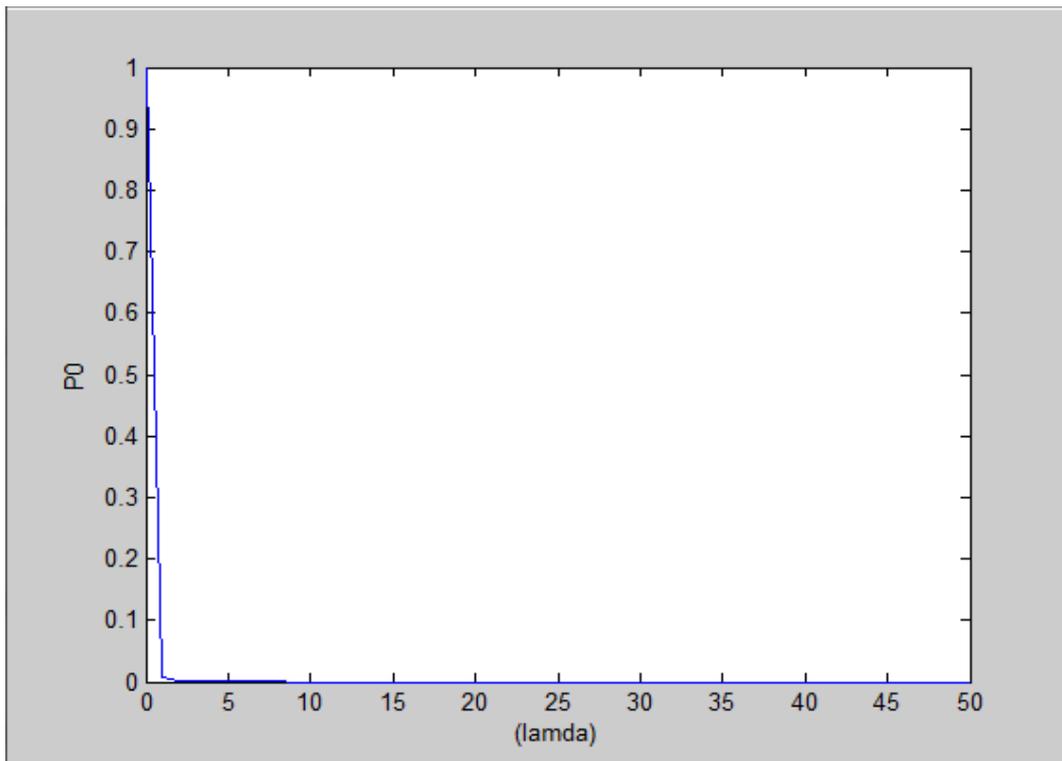


Figure 3.5 Loi de N_t pour $n=0$

Pour $n=10 \leq s$

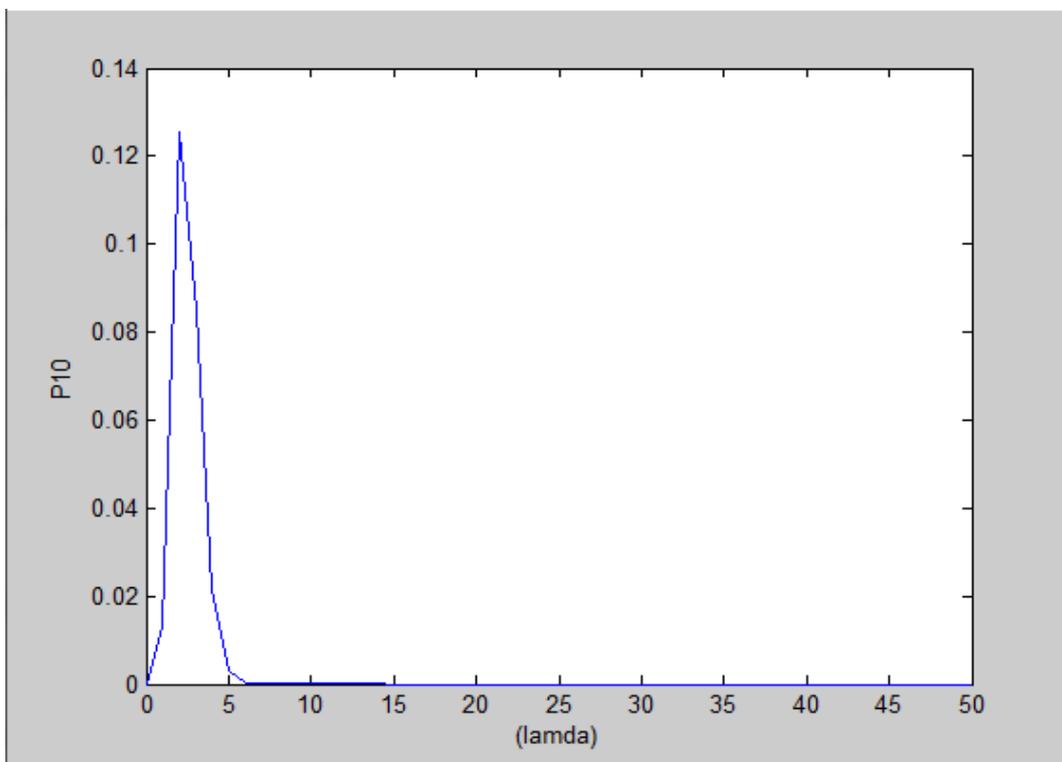


Figure 3.6 Loi de N_t pour $n=10$

Pour $n=30 > s$

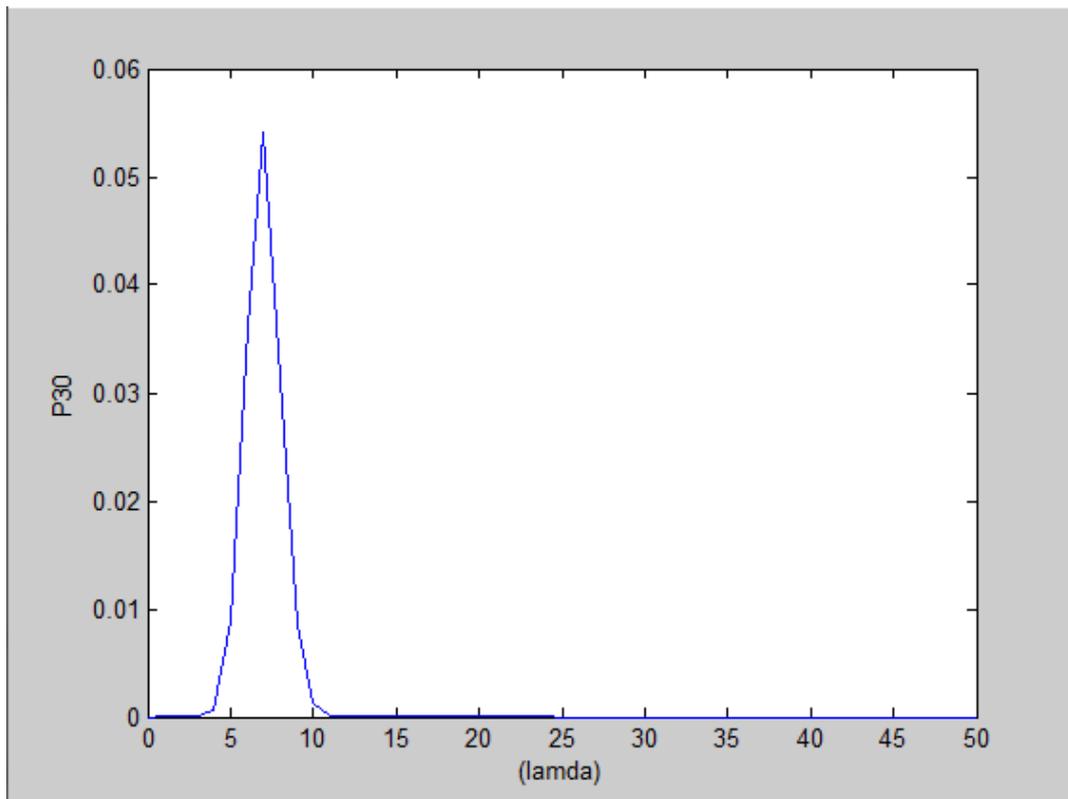


Figure 3.7 Loi de N_t pour $n=30$

3.3.1.2 Simulation des caractéristiques de la file

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

```

1  function pzerosk=p0sk(mu , lamda,s,k)
2  -      c1=( (fact(k) * (s^s) ) / fact(s) );
3  -      sk1=0;
4  -      sk2=0;
5  -      for i=1:s
6  -          sk1=sk1+(fact(k) / (fact(i) * fact(k-i))) * ((lamda/mu) ^ i);
7  -      end
8  -      for j=s+1:k
9  -          sk2=sk2+(c1 * (1 / fact(k-j)) * (lamda / (s * mu)) ^ j);
10 -      end
11 -      pzerosk=(1 / (1+sk1+sk2));
12 -  end

```

Figure 3.8 Programme de $P(N_t = 0)$

A- Nombre moyen d'appels en attente et en cours de connexion

i. Simulation en fonction de λ

Nous fixons $K=100$, $\mu=20$ et $s=25$, et regardons comment évolue cette caractéristique.

Le code de la formule sous MATLAB est

```

1  function nbrmappsk=appsk(z)
2  -
3  -   mu=20;
4  -   k=100;
5  -   s=25;
6  -   c1=((fact(k)*(s^s))/fact(s));
7  -   for lamda=0:z
8  -       sk1=0;
9  -       sk2=0;
10 -       for i=1:s
11 -           p0=p0sk(mu, lamda, s, k);
12 -           sk1=sk1+(i*(fact(k)/(fact(i)*fact(k-i)))*((lamda/mu)^i)*p0;
13 -       end
14 -       for j=s+1:k
15 -           p0=p0sk(mu, lamda, s, k);
16 -           sk2=sk2+(c1*(j/fact(k-j))*(lamda/(s*mu))^j)*p0;
17 -       end
18 -       moy(1, lamda+1)=sk1+sk2;
19 -       nbrmappsk=moy;
20 -   end
21 -   lambda=0:z;
22 -   plot(lambda, nbrmappsk)
23 -   xlabel(' (lamda) ')
24 -   ylabel('nombre moyen d"appels')
25 - end

```

Figure 3.9 Programme du nombre moyen d'appels dans le central téléphonique en fonction de λ

Le résultat obtenu est illustré par le graphe suivant

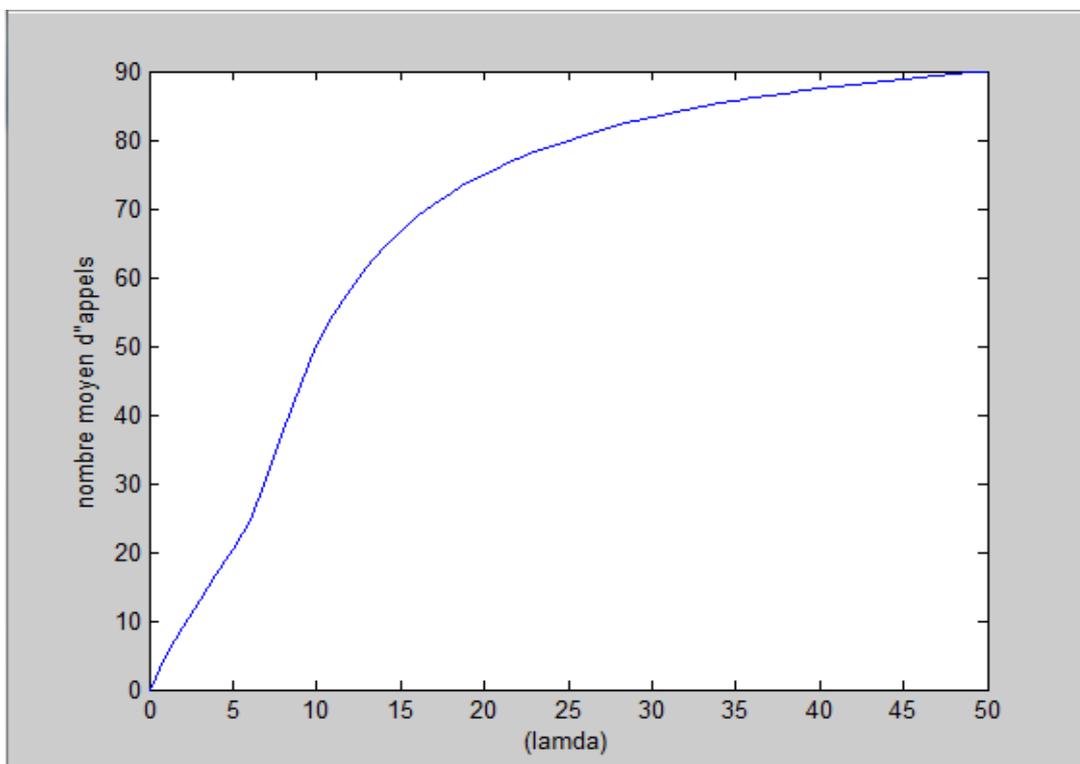


Figure 3.10 nombre moyen d'appels dans le central téléphonique en fonction de λ

On observe une croissance du nombre moyen d'appels à chaque fois que le taux d'arrivées λ augmente.

ii. Simulation en fonction de μ

Dans ce cas, on fixe $K=100$, $\lambda=20$ et $s=25$, et on évolue cette caractéristique en fonction du taux de mort.

Le code de la formule sous MATLAB est

```

1  function nbrmappskmu=appskmu(z)
2  |lamda=20;
3  k=100;
4  s=25;
5  c1= ( (fact(k) * (s^s) ) / fact(s) );
6  for mu=1:z
7      sk1=0;
8      sk2=0;
9      for i=1:s
10         p0=p0sk(mu, lamda, s, k) ;
11         sk1=sk1+(i*(fact(k) / (fact(i) * fact(k-i) ) ) * ( (lamda/mu)^i ) *p0;
12     end
13     for j=s+1:k
14         p0=p0sk(mu, lamda, s, k) ;
15         sk2=sk2+(c1*(j/fact(k-j) ) * (lamda/ (s*mu) ) ^j) *p0;
16     end
17     moy(1, mu) =sk1+sk2;
18     nbrmappskmu=moy;
19 end
20 plot(nbrmappskmu)
21 xlabel(' (mu) ')
22 ylabel('nombre moyen d'appels')
23 end

```

Figure 3.11 Programme du nombre moyen d'appels dans le central téléphonique en fonction de μ

Le résultat obtenu est représenté par le graphe suivant

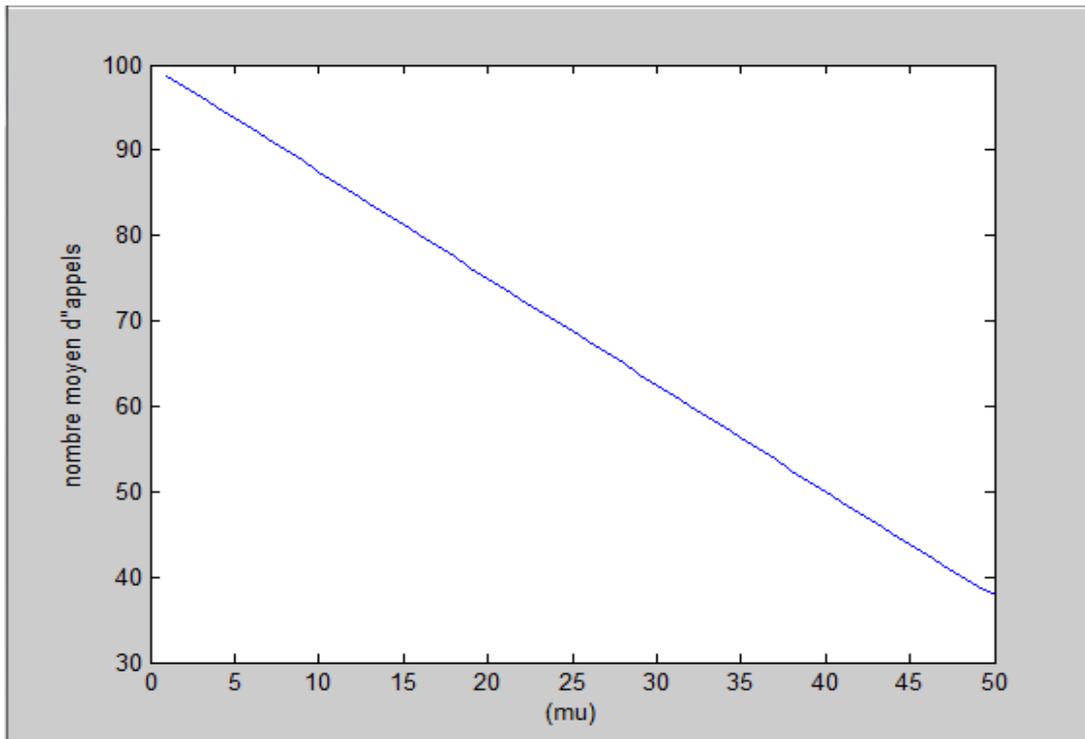


Figure 3.12 Nombre moyen d'appels dans le central téléphonique en fonction de μ

iii. Simulation en fonction de s

On évolue le nombre moyen d'appels dans le central téléphonique en fonction de nombre de canaux s tout en fixant $K=100$, $\lambda=20$ et $\mu=13$.

Le code de la formule sous MATLAB est

```

1  function nbrmappsks=appsks(z)
2  -   lamda=20;
3  -   k=100;
4  -   mu=15;
5  -   for s=1:z
6  -       c1= ((fact(k) *(s^s))/fact(s));
7  -       sk1=0;
8  -       sk2=0;
9  -       for i=1:s
10 -           p0=p0sk(mu, lamda, s, k);
11 -           sk1=sk1+(i*(fact(k)/(fact(i)*fact(k-i))))*((lamda/mu)^i)*p0;
12 -       end
13 -       for j=s+1:k
14 -           p0=p0sk(mu, lamda, s, k);
15 -           sk2=sk2+(c1*(j/fact(k-j))*(lamda/(s*mu))^j)*p0;
16 -       end
17 -       moy(1,s)=sk1+sk2;
18 -       nbrmappsks=moy;
19 -   end
20 -   plot(nbrmappsks)
21 -   xlabel(' (s) ')
22 -   ylabel('nombre moyen d"appels')
23 - end

```

Figure 3.13 Programme du nombre moyen d'appels dans le central téléphonique en fonction de s

Le graphe obtenu est

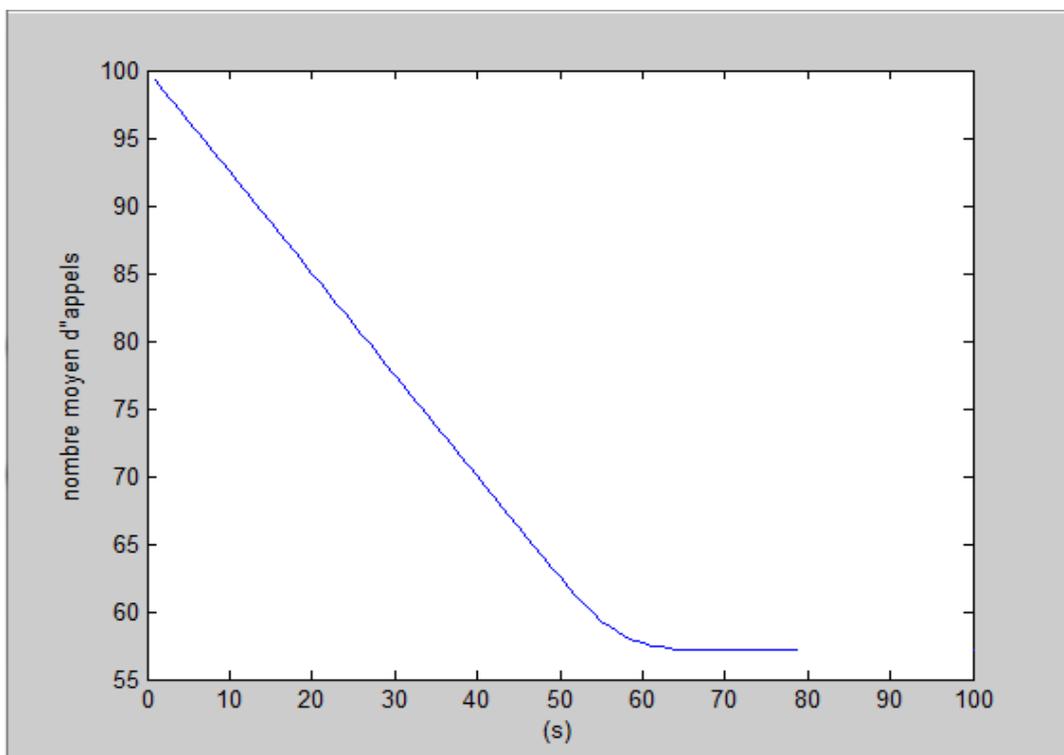


Figure 3.14 Nombre moyen d'appels dans le central téléphonique en fonction de s

On remarque dans les deux graphes précédents que le nombre moyen d'appels dans le central téléphonique décroît au fur et à mesure que le nombre de canaux s et le taux de sortie μ augmentent.

B- Temps moyen d'attente dans le central téléphonique

i. Simulation en fonction de λ

Nous fixons $K=100$, $\mu=20$ et $s=25$, et regardons comment évolue cette caractéristique.

On donne le programme sous MATLAB qui évalue la caractéristique considérée.

```

1  function tempsmoyen1=temps1(z)
2  -   s=25;
3  -   k=100;
4  -   mu=20;
5  -   for lamda=1:z
6  -       c1=( (fact(k) *(s^s) )/fact(s) );
7  -       sk1=0;
8  -       sk2=0;
9  -       for i=1:s
10 -           p0=p0sk(mu, lamda, s, k);
11 -           sk1=sk1+(i*(fact(k) / (fact(i) *fact(k-i) ))) * ((lamda/mu) ^i) *p0;
12 -       end
13 -       for j=s+1:k
14 -           p0=p0sk(mu, lamda, s, k);
15 -           sk2=sk2+(c1*(j/fact(k-j)) * (lamda/ (s*mu) ) ^j) *p0;
16 -       end
17 -       moye=sk1+sk2;
18 -       moyen=moye/ (lamda*(k-moye) );
19 -       moy(1, lamda)=moyen;
20 -       tempsmoyen1=moy;
21 -   end
22 -   plot(tempsmoyen1)
23 -   xlabel(' (lamda) ')
24 -   ylabel(' temps d"attente')
25 - end

```

Figure 3.15 Programme du temps d'attente moyen dans le central téléphonique en fonction de λ

Le graphe obtenu après simulation est

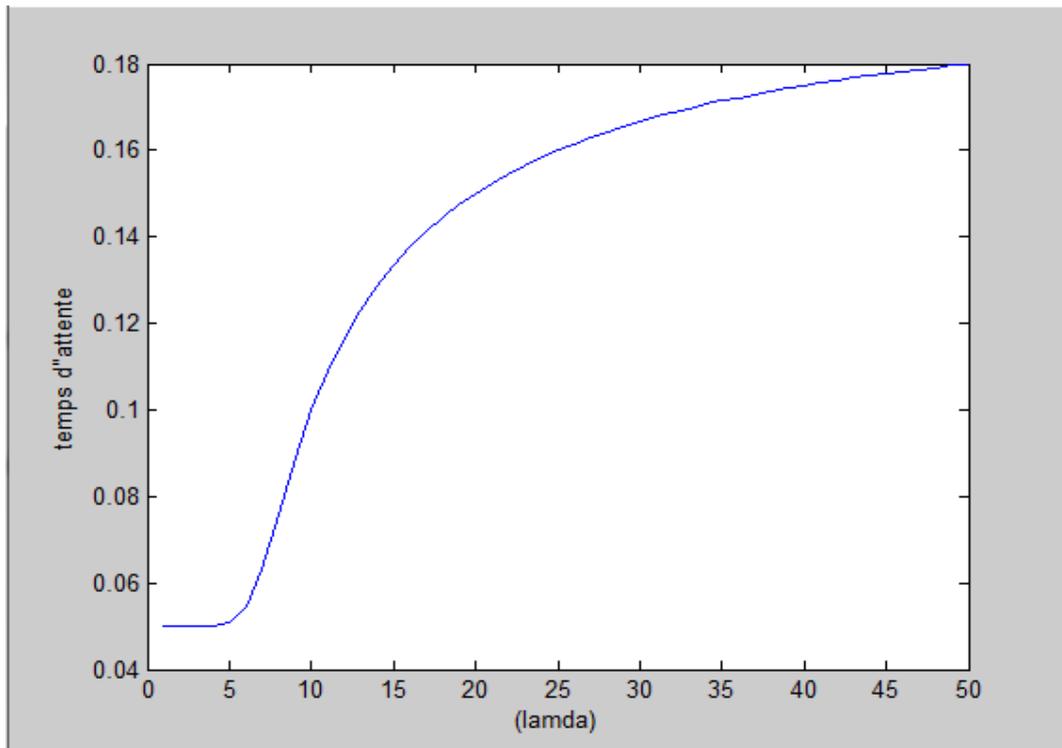


Figure 3.16 Temps d'attente dans le central téléphonique en fonction de λ

ii. Simulation en fonction de μ

On s'intéresse dans ce cas à l'évolution du temps d'attente dans le central téléphonique tout en variant μ .

Le programme permettant d'évoluer la caractéristique est donné sous MATLAB

```

1  function tempsmoyenmu=tempsmu(z)
2  -
3  -   s=25;
4  -   k=100;
5  -   lamda=20;
6  -   for mu=1:z
7  -       c1=((fact(k)*(s^s))/fact(s));
8  -       sk1=0;
9  -       sk2=0;
10 -       for i=1:s
11 -           p0=p0sk(mu,lamda,s,k);
12 -           sk1=sk1+(i*(fact(k)/(fact(i)*fact(k-i))))*(lamda/mu)^i*p0;
13 -       end
14 -       for j=s+1:k
15 -           p0=p0sk(mu,lamda,s,k);
16 -           sk2=sk2+(c1*(j/fact(k-j))*(lamda/(s*mu))^j)*p0;
17 -       end
18 -       moye=sk1+sk2;
19 -       moyen=moye/(lamda*(k-moye));
20 -       moy(1,mu)=moyen;
21 -       tempsmoyenmu=moy;
22 -   end
23 -   plot(tempsmoyenmu)
24 -   xlabel('(mu)')
25 -   ylabel('temps d"attente')
26 - end

```

Figure 1.17 Programme du temps d'attente moyen dans le central téléphonique en fonction de μ

Le graphe obtenu est

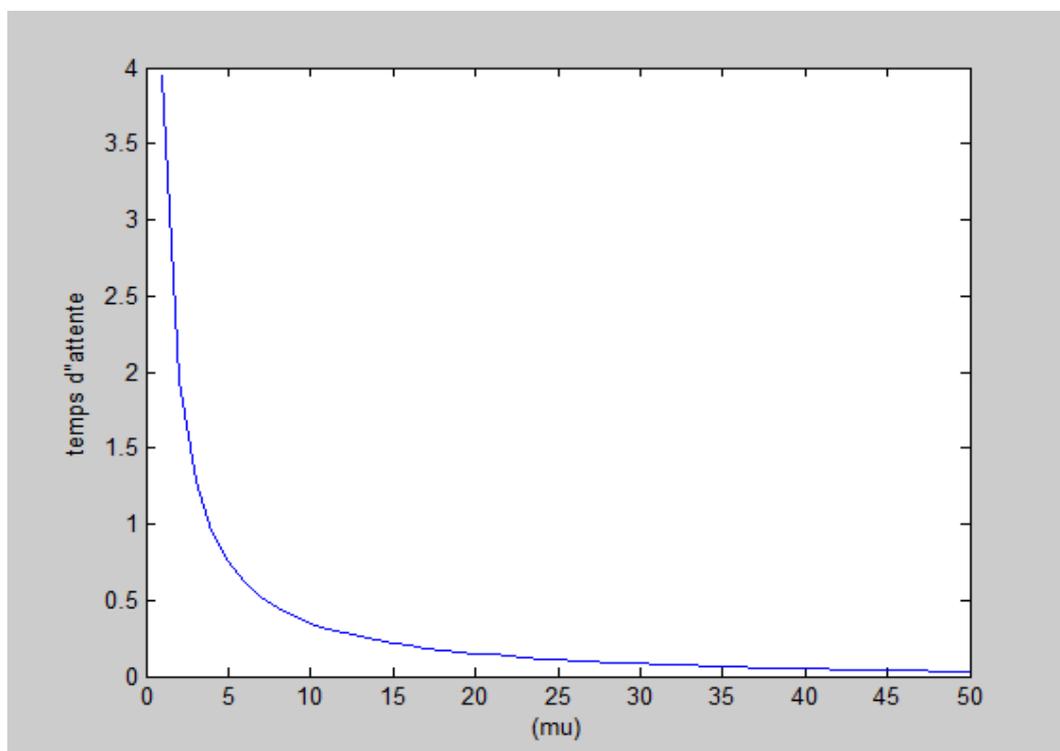


Figure 1.18 Temps d'attente dans le central téléphonique en fonction de μ

iii. Simulation en fonction de s

On évolue le temps moyen d'appels dans le central téléphonique en fonction de nombre de canaux s tout en fixant $K=100$, $\lambda=20$ et $\mu=13$.

Le code de la formule sous MATLAB est

```
1 function tempsmoyens=tempss(z)
2 mu=15;
3 k=100;
4 lamda=20;
5 for s=1:z
6     c1=((fact(k)*(s^s))/fact(s));
7     sk1=0;
8     sk2=0;
9     for i=1:s
10        p0=p0sk(mu,lamda,s,k);
11        sk1=sk1+(i*(fact(k)/(fact(i)*fact(k-i))))*((lamda/mu)^i)*p0;
12    end
13    for j=s+1:k
14        p0=p0sk(mu,lamda,s,k);
15        sk2=sk2+(c1*(j/fact(k-j))*(lamda/(s*mu))^j)*p0;
16    end
17    moye=sk1+sk2;
18    moyen=moye/(lamda*(k-moye));
19    moy(1,s)=moyen;
20    tempsmoyens=moy;
21 end
22 plot(tempsmoyens)
23 xlabel(' (s) ')
24 ylabel('temps d"attente')
25 end
```

Figure 1.19 Programme du temps d'attente moyen dans le central téléphonique en fonction de s

Le graphe obtenu est

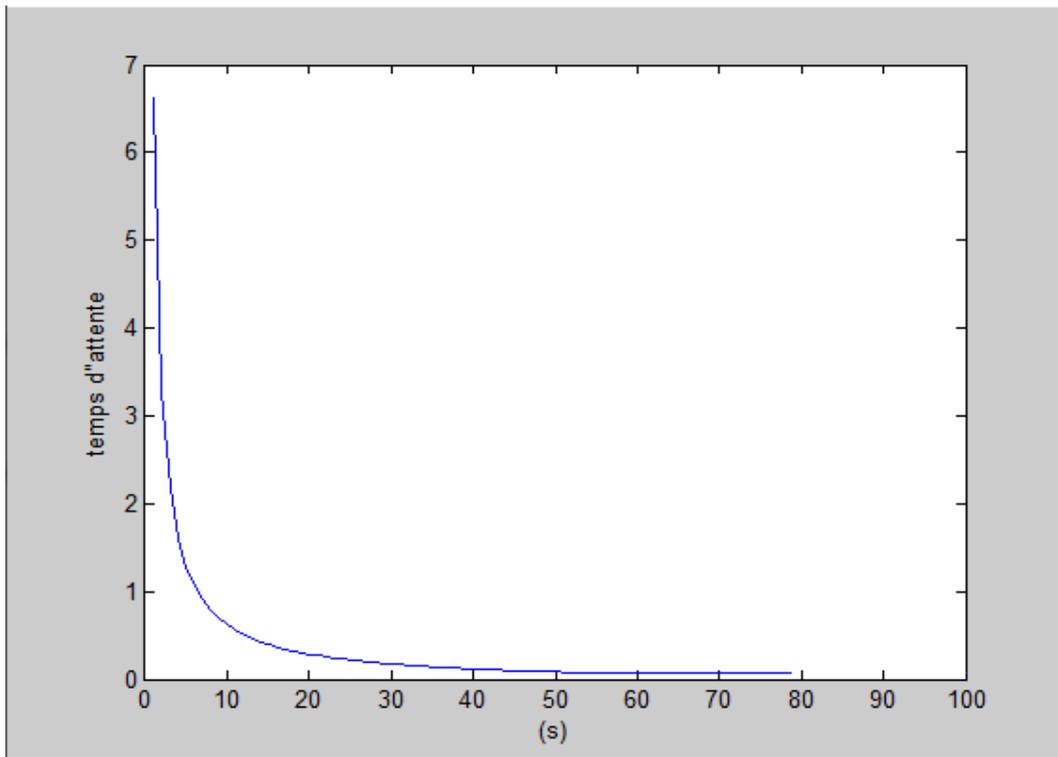


Figure 3.20 Temps d'attente dans le central téléphonique en fonction de s

3.3.1.3 Risque de saturation

Dans ce cas, on s'intéresse à l'évolution du risque de saturation du central téléphonique en fonction des différents paramètres λ , μ , et s.

a) Evolution en fonction de λ

On suppose que $k=100$, $\mu=20$, et $s=25$. Le code sous MATLAB est

```

1  function risquesaturationskl = risqueklamda(z)
2  -   mu=15;
3  -   k=100;
4  -   s=25;
5  -   for lamda=0:z
6  -       soml=0;
7  -       c1=(fact(k)/fact(s))*(s^s)*p0sk(mu, lamda, s, k);
8  -       for n=s:k
9  -           soml=soml+c1*(1/fact(k-n))*((lamda/(s*mu))^n);
10 -       end
11 -       risque(1, lamda+1)=soml;
12 -       risquesaturationskl=risque;
13 -   end
14 -   lambda=0:z;
15 -   plot(lambda, risquesaturationskl)
16 -   xlabel(' (lamda) ')
17 -   ylabel(' P (Nt >= s) ')
18 - end

```

Figure 3.21 Programme du risque de saturation en fonction de λ

Le graphe obtenu est

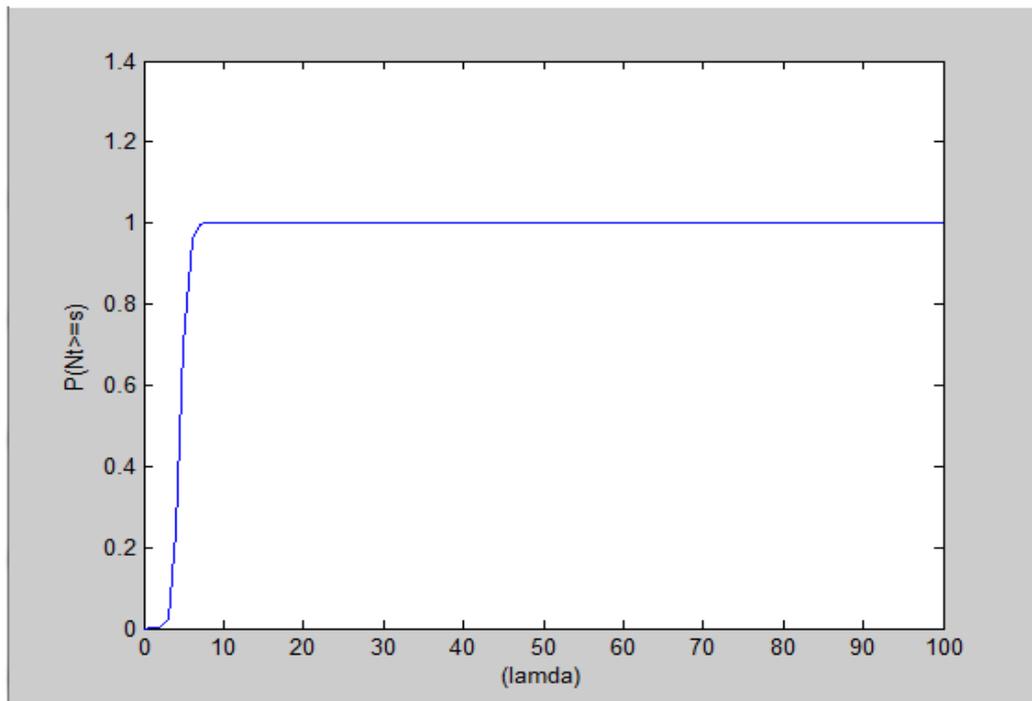


Figure 3.22 Risque de saturation en fonction de λ

b) Evolution en fonction de μ

On suppose que $k=100$, $\lambda=20$, et $s=25$. Le code sous MATLAB est

```

1  function risquesaturationskmu = risquekmu(z)
2  -   lamda=20;
3  -   k=100;
4  -   s=25;
5  -   for mu=1:z
6  -       som1=0;
7  -       c1=(fact(k)/fact(s))*(s^s)*p0sk(mu,lamda,s,k);
8  -       for n=s:k
9  -           som1=som1+c1*(1/fact(k-n))*((lamda/(s*mu))^n);
10 -       end
11 -       risque(1,mu)=som1;
12 -       risquesaturationskmu=risque;
13 -   end
14 -   plot(risquesaturationskmu)
15 -   xlabel(' (mu) ')
16 -   ylabel(' P(Nt >=s) ')
17 - end

```

Figure 3.23 Programme du risque de saturation en fonction de μ

Le graphe d'évolution est

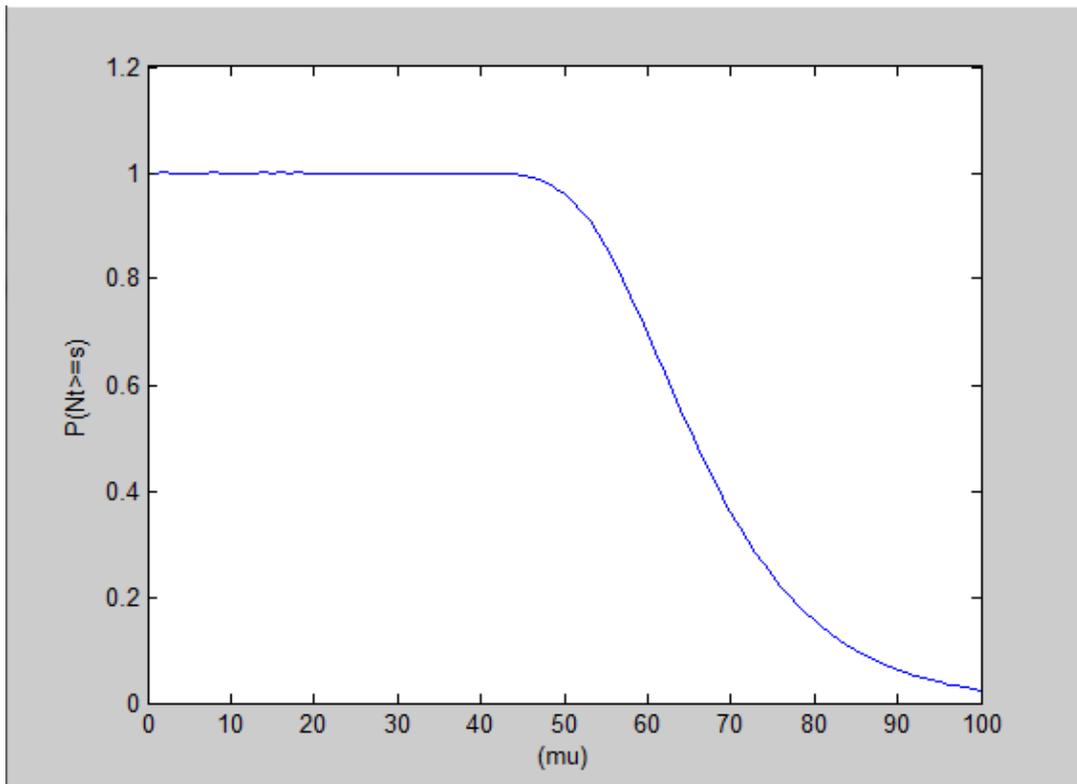


Figure 3.24 Risque de saturation en fonction de μ

c) Evolution en fonction de s

On suppose que $k=100$, $\lambda=20$, et $\mu=15$. Le code sous MATLAB est

```

1  function risquesaturationsk = risques(z)
2  -     lamda=20;
3  -     k=100;
4  -     mu=15;
5  -     for s=1:z
6  -         som1=0;
7  -         c1=(fact(k)/fact(s)) * (s^s) * p0sk(mu, lamda, s, k);
8  -         for n=s:k
9  -             som1=som1+c1*(1/fact(k-n)) * ((lamda/(s*mu))^n);
10 -        end
11 -        risque(1,s)=som1;
12 -        risquesaturationsk=risque;
13 -    end
14 -    plot(risquesaturationsk)
15 -    xlabel('nombre de canaux s')
16 -    ylabel('P(Nt >=s)')
17 - end

```

Figure 3.25 Programme du risque de saturation en fonction de μ

Le résultat obtenu est le suivant

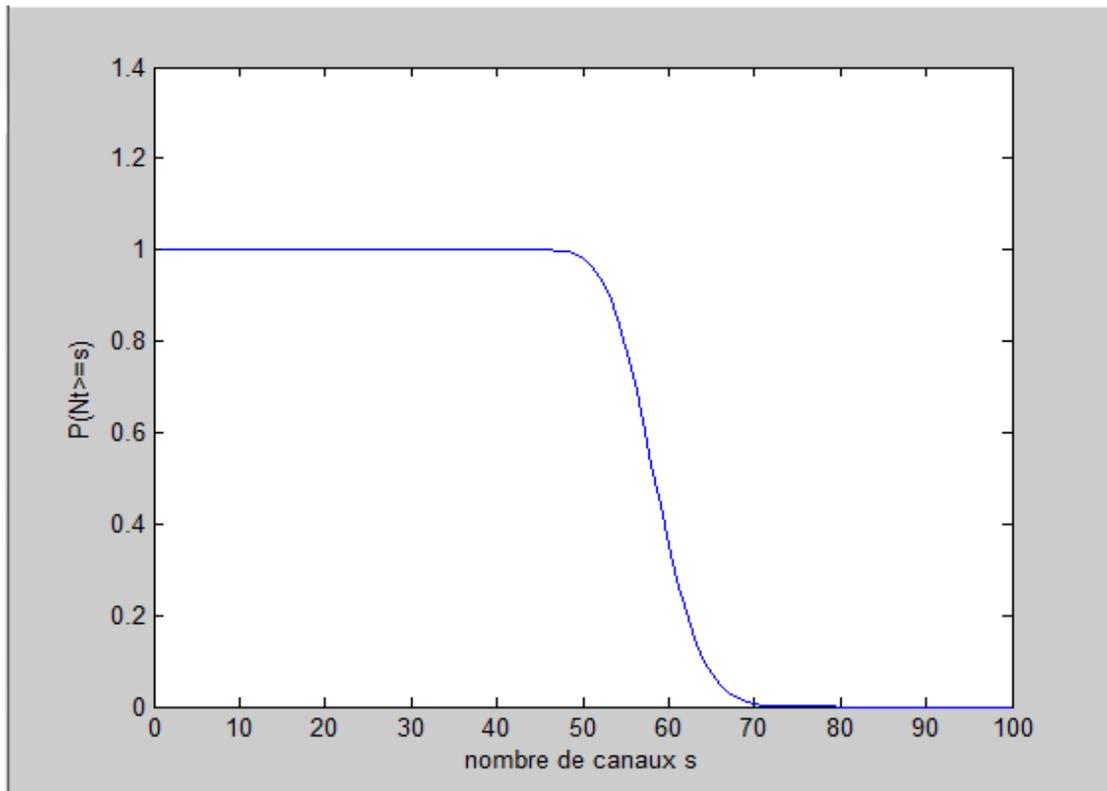


Figure 3.26 Risque de saturation en fonction de s

3.3.2 File d'attente M/M/s/s/[F]

3.3.2.1 Simulation de la loi de N_t

On s'intéresse à l'évolution de la loi de $(N_t)_{t \geq 0}$ en fonction du taux des arrivées d'appels au central téléphonique λ , supposons que le nombre d'abonnés K est égal à 100, le taux des durées d'appels μ est égal à 20 et le nombre de serveurs s égal à 25.

On donne le script de la loi de (N_t) sous MATLAB

```

1  function pntss=pnss(n,z)
2  -   mu=20;
3  -   k=100;
4  -   s=25;
5  -   if n>=0 && n<=s
6  -   -   for lamda=0:z
7  -   -   -   ss1=0;
8  -   -   -   com=(fact(k)/(fact(n)*fact(k-n)))*(lamda/mu)^n;
9  -   -   -   for i=1:s
10 -   -   -   -   ss1=ss1+(fact(k)/(fact(i)*fact(k-i)))*((lamda/mu)^i);
11 -   -   -   end
12 -   -   -   pn(1,lamda+1)=com/(1+ss1);
13 -   -   -   pntss=pn;
14 -   -   end
15 -   -   lambda=0:z;
16 -   -   tableau=[lambda;pntss]
17 -   -   plot(lambda,pntss,'r')
18 -   -   xlabel(' (lamda) ')
19 -   -   ylabel(' (Pn) ')
20 -   -   else
21 -   -   -   disp('Pn=0')
22 -   -   end
23 -   end

```

Figure 3.27 Programme de la loi de (N_t)

Les résultats de la simulation sont :

Pour n=0

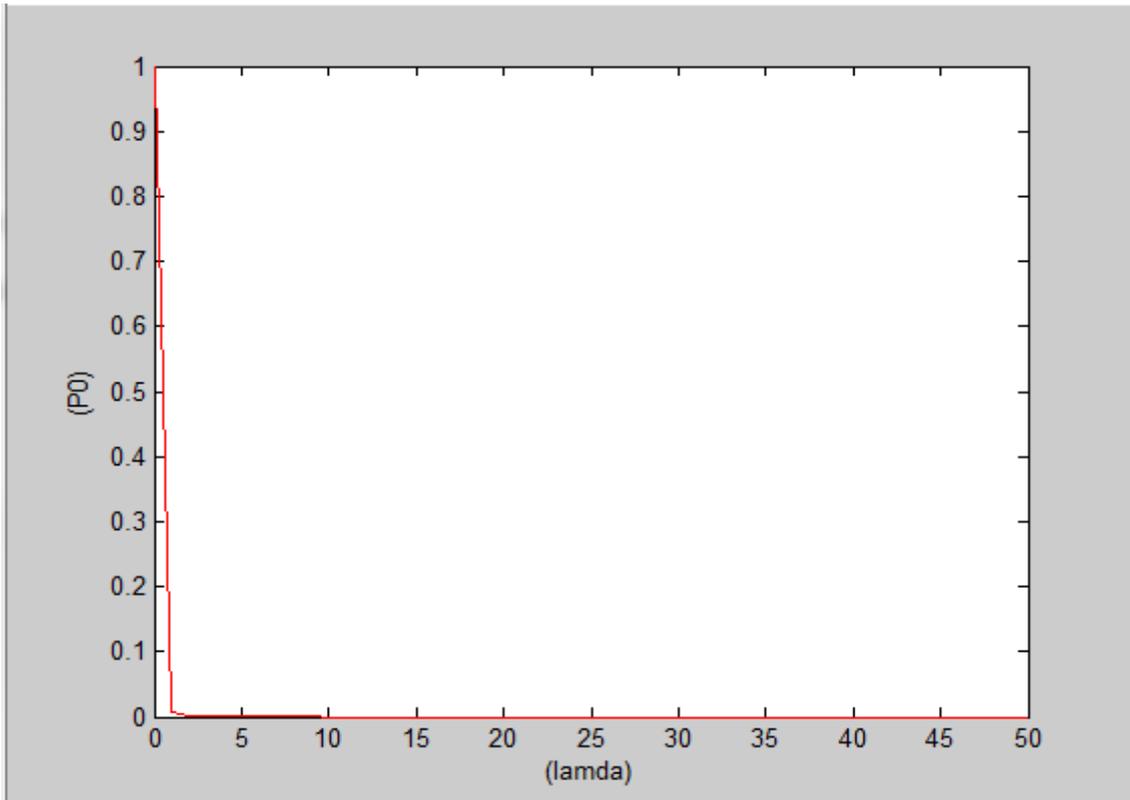


Figure 3.28 Loi de $N(t)$ pour $n=0$

Pour $n=10$

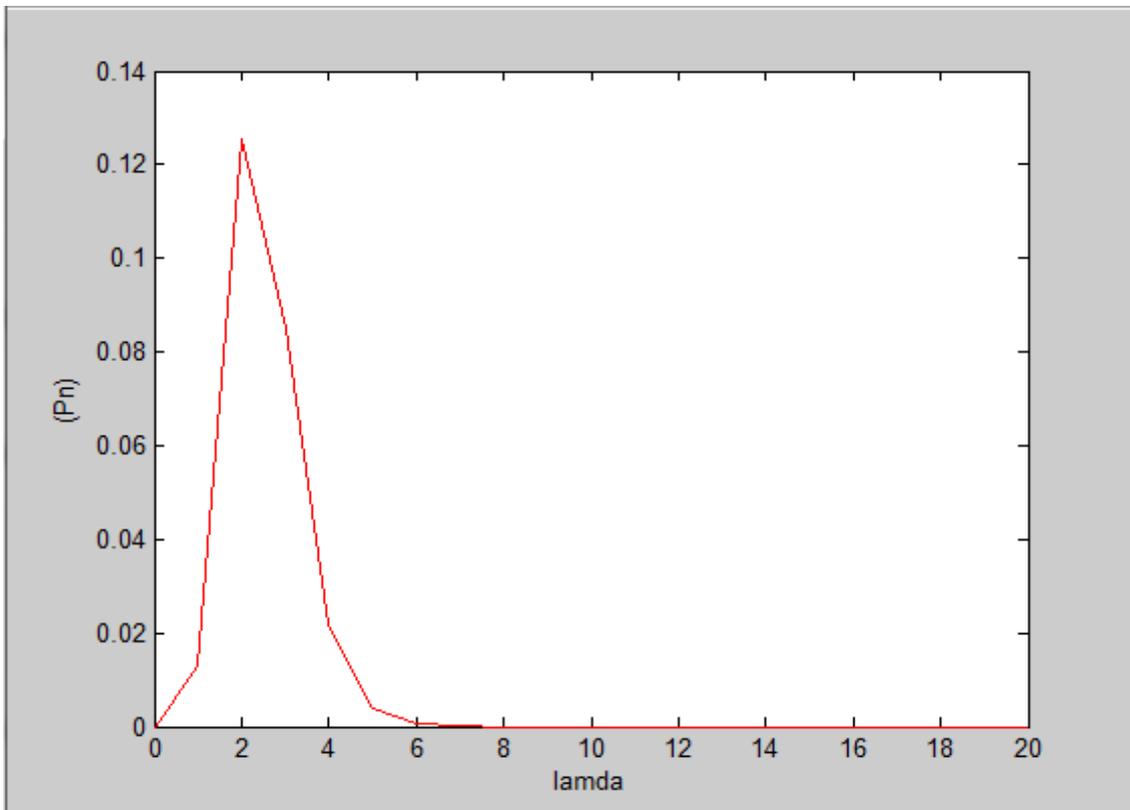


Figure 3.29 Loi de $N(t)$ pour $n=10$

Remarque 1

On s'aperçoit que ce cas ressemble à celui de la file M/M/s/K/ [F], on obtient presque les mêmes probabilités.

3.3.2.2 Simulation des caractéristiques de la file

On donne d'abord le code de $P(N_t = 0)$

```
1 - function pOfss=pOfs(k, s, mu, lamda)
2 -     som1=0;
3 -     for i=0:s
4 -         som1=som1+(fact(k)/(fact(i)*fact(k-i)))*((lamda/mu)^i);
5 -     end
6 -     pOfss=1/som1;
7 - end
```

Figure 3.30 Programme de $P(N_t = 0)$

A- Nombre moyen d'appels dans le central téléphonique

i. Simulation en fonction de λ

Nous fixons $K=100$ et $\mu=20$, et regardons comment évolue cette caractéristique.

On donne le programme sous MATLAB qui évalue la caractéristique considérée.

Le code sous MATLAB est

```
1 - function nobmoyapplss = appss(z)
2 -     mu=20;
3 -     K=100;
4 -     for lamda=0:z
5 -         moyen(1, lamda+1)=(K*lamda)/(mu+lamda);
6 -         nobmoyapplss=moyen;
7 -     end
8 -     lambda=0:z;
9 -     plot(lambda, nobmoyapplss, 'r')
10 -     xlabel(' (lamda) ')
11 -     ylabel('nombre moyen d"appels')
12 - end
```

Figure 3.31 Programme du nombre moyen d'appels dans le central téléphonique en fonction de λ

On obtient le graphe suivant

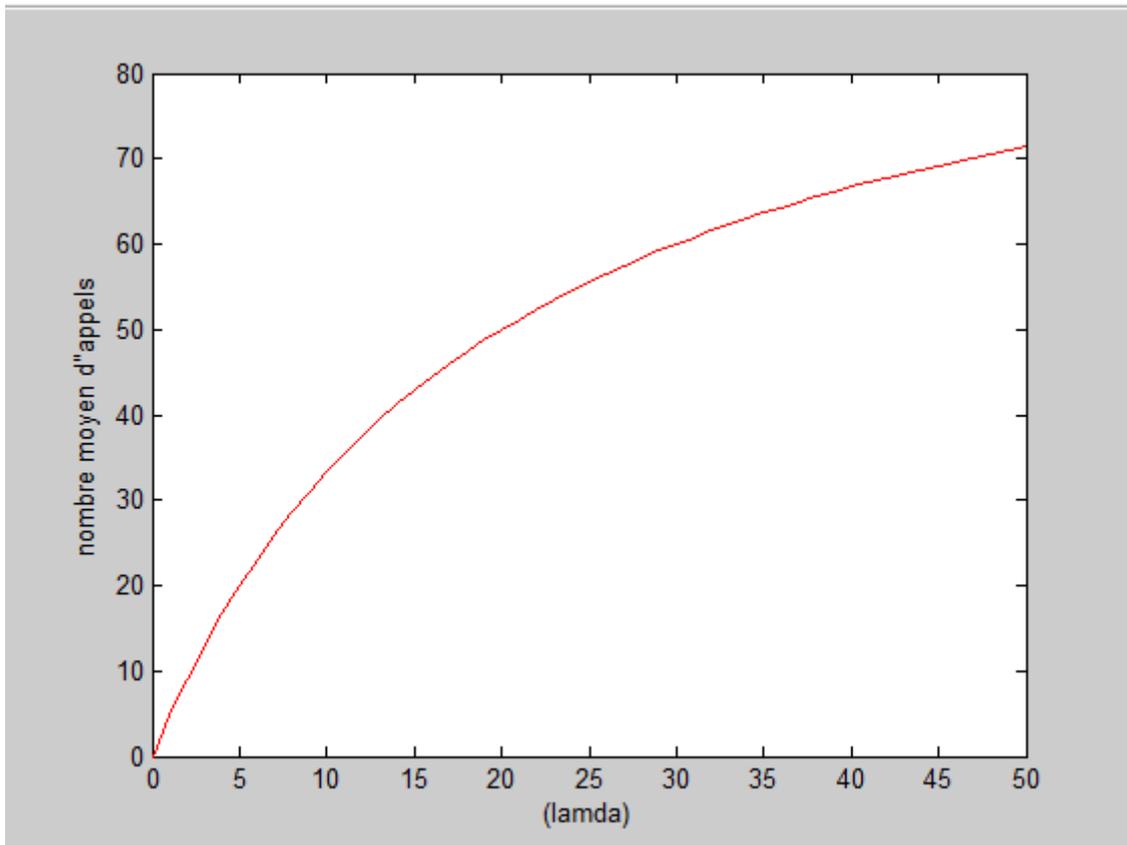


Figure 3.32 Nombre moyen d'appels dans le central téléphonique en fonction de λ

ii. Simulation en fonction de μ

On suppose que $k=100$ et $\lambda=20$, le code sous MATLAB est

```

1  function nobmoyapplsmu = appsmu(z)
2  -     lamda=20;
3  -     K=100;
4  -     for mu=0:z
5  -         moyen(1,mu+1) = (K*lamda) / (mu+lamda);
6  -         nobmoyapplsmu=moyen;
7  -     end
8  -     muu=0:z;
9  -     plot(muu,nobmoyapplsmu,'r')
10 -     xlabel('(mu)')
11 -     ylabel('nombre moyen d'appels')
12 - end

```

Figure 3.33 Programme du nombre moyen d'appels dans le central téléphonique en fonction de μ

Le résultat obtenu est

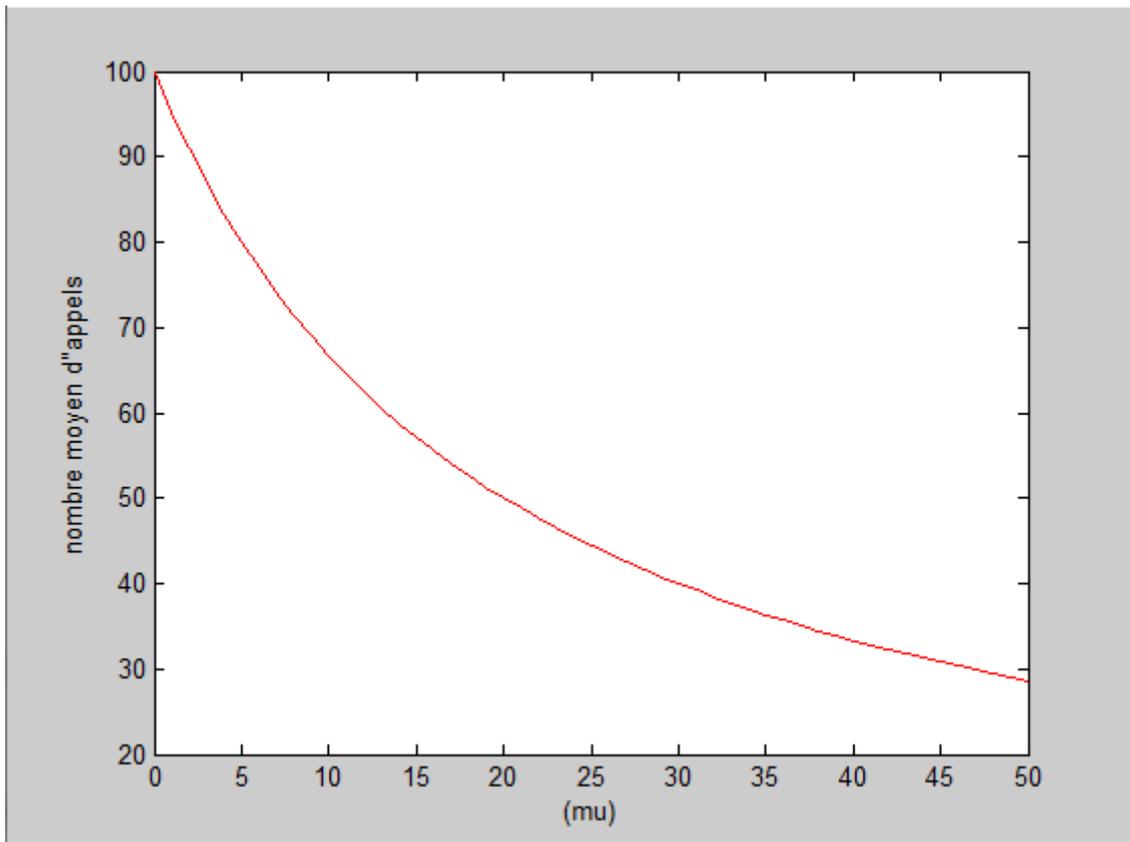


Figure 3.34 Nombre moyen d'appels dans le central téléphonique en fonction de μ

B- Temps moyen d'attente dans le central téléphonique

Dans ce cas, on s'intéresse au temps moyen d'attente en fonction de μ .

Le code sous MATLAB est

```

1  function tempsmoyss=temps(z)
2  -  for mu=1:z
3  -      tempsmo(1,mu) = (1/mu);
4  -      tempsmoyss=tempsmo;
5  -  end
6  -  plot(tempsmoyss,'r')
7  -  xlabel('mu')
8  -  ylabel('temps d"attente')
9  -  end

```

Figure 3.35 Programme du temps d'attente moyen dans le central téléphonique en fonction de μ

Le résultat obtenu est

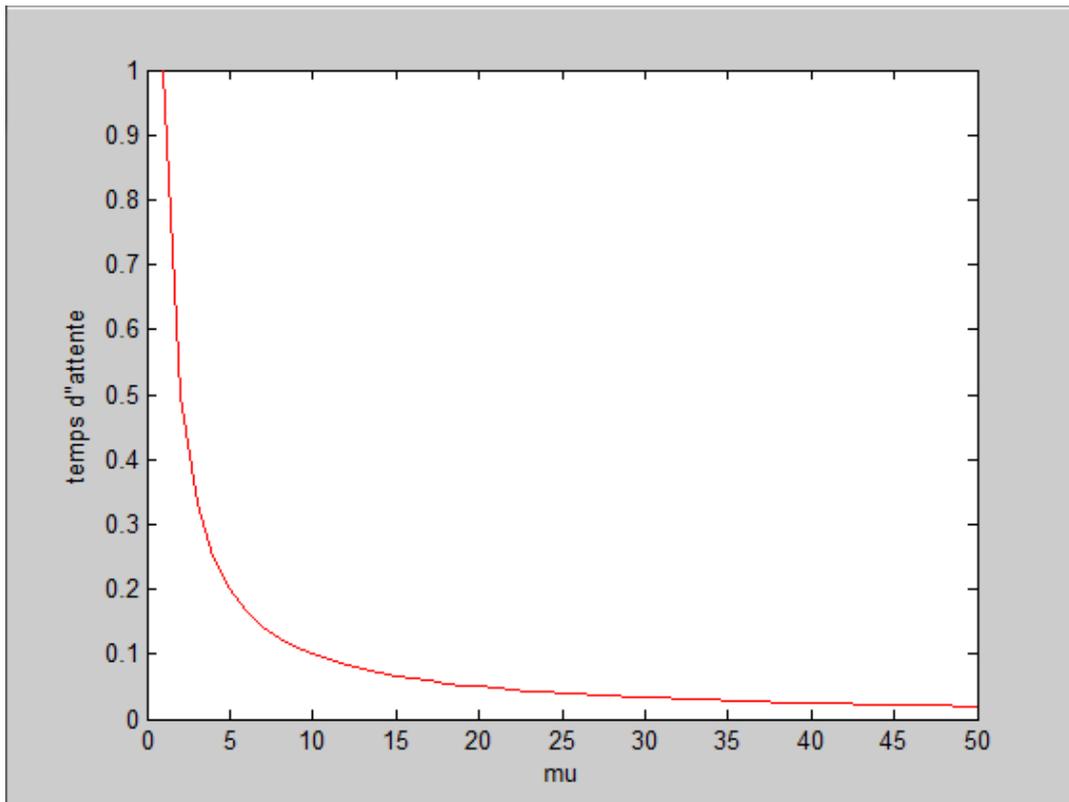


Figure 3.36 Temps moyen d'attente en fonction de μ

3.3.2.3 Risque de saturation

Dans ce cas, on s'intéresse à l'évolution du risque de saturation du central téléphonique en fonction des différents paramètres λ , μ , et s .

i. Evolution en fonction de λ

On suppose que $k=100$, $\mu=20$, et $s=25$. Le code sous MATLAB est

```

1  function risquesaturationssl=risquelamda(z)
2  -   k=100;
3  -   mu=20;
4  -   s=25;
5  -   for lamda=0:z
6  -       som2=0;
7  -       p0=p0fs(k,s,mu,lamda);
8  -       som2=som2+(fact(k)/(fact(s)*fact(k-s)))*((lamda/mu)^s)*p0;
9  -       risque(1,lamda+1)=som2;
10 -       risquesaturationssl=risque;
11 -   end
12 -   lambda=0:z;
13 -   plot(lambda,risquesaturationssl,'r')
14 -   xlabel(' (lamda) ')
15 -   ylabel(' P (Nt>=s) ')
16 - end

```

Figure 3.37 Programme du risque de saturation en fonction de λ

Le graphe obtenu est

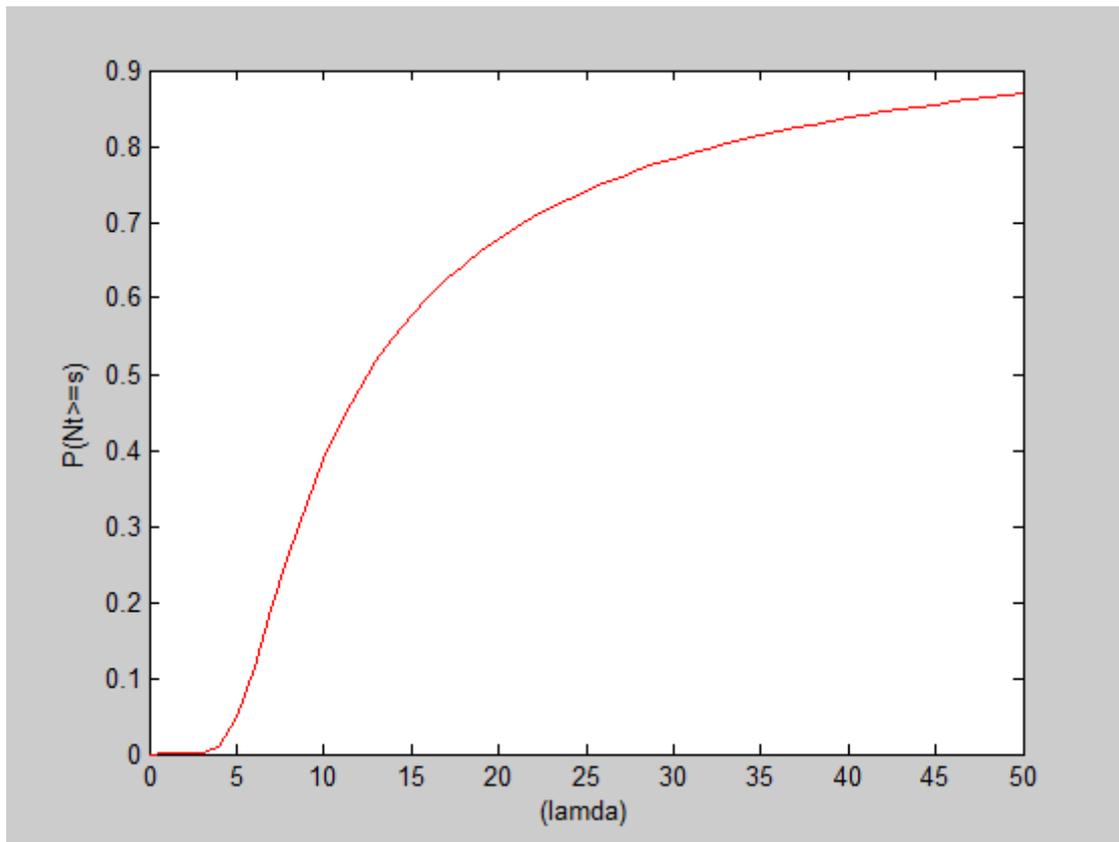


Figure 3.38 Risque de saturation en fonction de λ

ii. Evolution en fonction de μ

On suppose que $k=100$, $\lambda=20$, et $s=25$. Le code sous MATLAB est

```
1 function risquesaturationssm=risquemu(z)
2     k=100;
3     lamda=20;
4     s=25;
5     for mu=1:z
6         som2=0;
7         p0=pOfs(k,s,mu,lamda);
8         som2=som2+(fact(k)/(fact(s)*fact(k-s)))*((lamda/mu)^s)*p0;
9         risque(1,mu)=som2;
10        risquesaturationssm=risque;
11    end
12    plot(risquesaturationssm,'r')
13    xlabel('(mu)')
14    ylabel('P(Nt>=s)')
15 end
```

Figure 3.39 Programme du risque de saturation en fonction de μ

Le résultat obtenu est

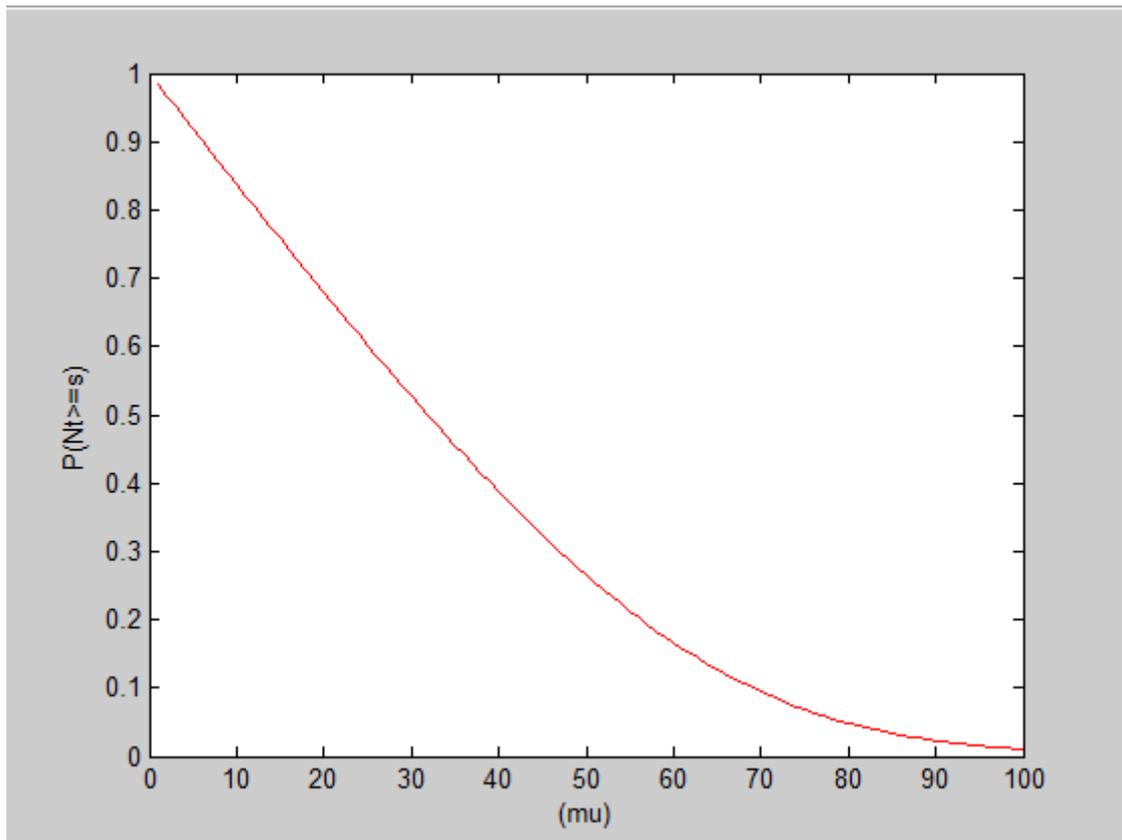


Figure 3.40 Risque de saturation en fonction de μ

iii. Evolution en fonction de s

On suppose que $k=100$, $\lambda=20$ et $\mu=15$. Le code sous MATLAB est

```

1  function risquesaturationss=risquess(z)
2  -   k=100;
3  -   mu=15;
4  -   lamda=20;
5  -   for s=1:z
6  -       som2=0;
7  -       p0=p0fs(k,s,mu,lamda);
8  -       som2=som2+(fact(k)/(fact(s)*fact(k-s)))*((lamda/mu)^s)*p0;
9  -       risque(1,s)=som2;
10 -       risquesaturationss=risque;
11 -   end
12 -   plot(risquesaturationss,'r')
13 -   xlabel('nombre de canaux s')
14 -   ylabel('P(Nt>=s)')
15 -
16 - end

```

Figure 3.41 Programme du risque de saturation en fonction de s

Le résultat obtenu est

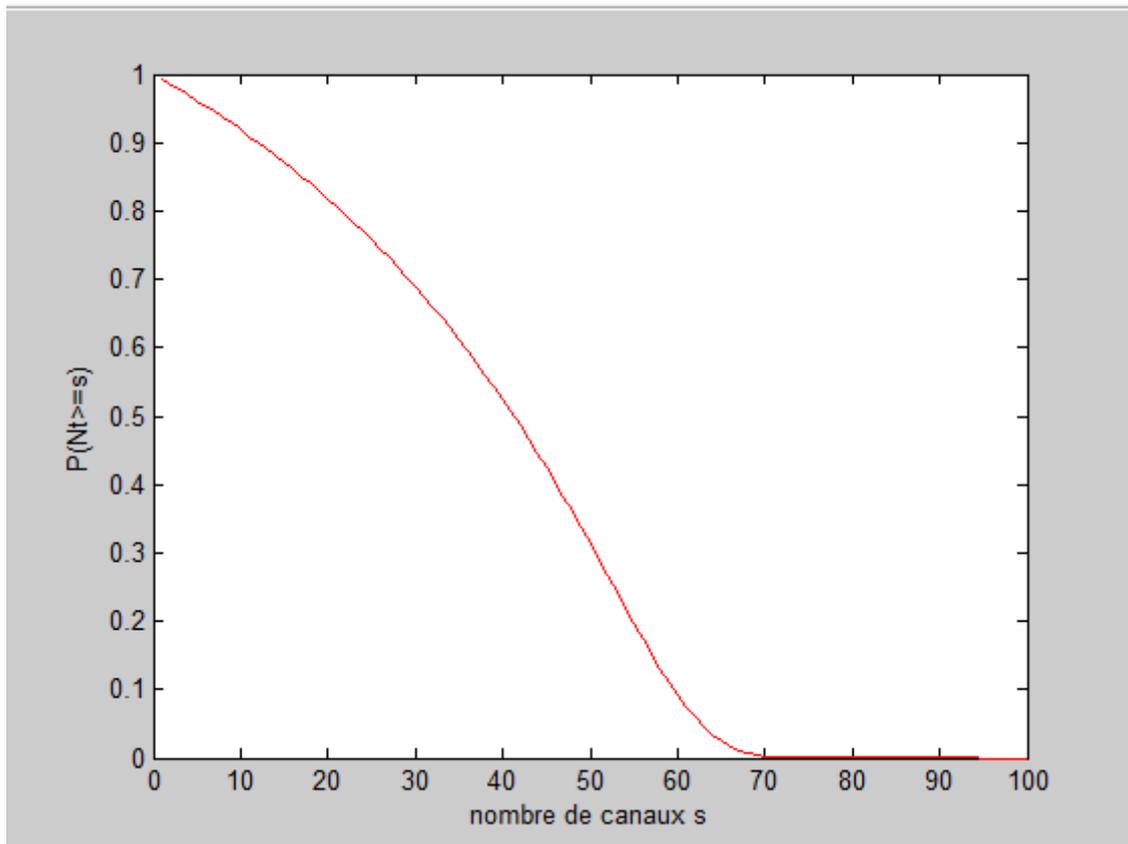


Figure 3.42 Risque de saturation en fonction de s

3.4 Conclusion

Ce chapitre nous a permis d'observer le comportement de nos systèmes et de leurs performances tout en variant les différents paramètres λ , μ et s . Ces résultats nous permettent d'évaluer le fonctionnement des phénomènes d'attente dans un central téléphonique.

Chapitre 4

Optimisation dans le central téléphonique

4.1 Introduction

L'optimisation est une branche des mathématiques et de l'informatique, en tant que discipline, cherchant à modéliser, analyser et à résoudre numériquement et analytiquement des problèmes qui sont souvent des systèmes physiques représentés par des modèles mathématiques. Elle joue un rôle très important dans la recherche opérationnelle, qui consiste à déterminer un minimum ou un maximum d'une fonction tout en respectant un ensemble de contraintes si elles existent afin d'améliorer et d'optimiser le système étudié.

Dans notre cas, on cherche à optimiser le nombre de serveurs nécessaires afin d'éviter la congestion du système, cela revient à déterminer le nombre « s » de canaux pour que la probabilité de saturation soit inférieure à un α donné.

4.2 File d'attente M/M/s/K/ [F]

4.2.1 Formulation du problème

On a

$$\pi(\rho, s, K) = \frac{K!}{s!} s^s p_0 \sum_{n=s}^K \frac{1}{(K-n)!} \left(\frac{\lambda}{s\mu}\right)^n.$$

Nous cherchons à optimiser la probabilité de saturation $\pi(\rho, s, K)$ par rapport au nombre de canaux, c'est-à-dire

$$\begin{aligned} \min_s \pi(\rho, s, K) &= \min_s \left(\frac{K!}{s!} s^s p_0 \sum_{n=s}^K \frac{1}{(K-n)!} \left(\frac{\lambda}{s\mu}\right)^n \right) \quad \text{avec } s \in \mathbb{N}^* \\ &\Rightarrow \left\{ \begin{array}{l} \min_s \left(\frac{K!}{s!} s^s p_0 \sum_{n=s}^K \frac{1}{(K-n)!} \left(\frac{\lambda}{s\mu}\right)^n \right) \\ s \in \mathbb{N}^* \end{array} \right. \end{aligned}$$

4.2.2 Optimisation de la probabilité de saturation

Nous procédons à la recherche d'une manière informatique, du nombre minimum de canaux nécessaires pour que la probabilité de saturation ne dépasse pas un seuil alpha donné avec $\alpha \in [0, 1]$, ainsi on suppose que $\lambda=13$, $\mu=7$ et $k=100$.

D'abord, on donne le script sous MATLAB du nombre moyen d'appels et le temps d'attente moyen dans le central téléphonique

a) Nombre moyen d'appels

```
1  function nombre=appl( mu, lamda, s, k)
2  -   c1= ((fact(k) * (s^s))/fact(s));
3  -   sk1=0;sk2=0;
4  -   for i=1:s
5  -       p0=p0sk(mu, lamda, s, k);
6  -       sk1=sk1+(i*(fact(k)/(fact(i)*fact(k-i)))*((lamda/mu)^i)*p0;
7  -   end
8  -   for j=s+1:k
9  -       p0=p0sk(mu, lamda, s, k);
10 -       sk2=sk2+(c1*(j/fact(k-j))*(lamda/(s*mu))^j)*p0;
11 -   end
12 -   moy=sk1+sk2;
13 -   nombre=moy;
14 - end
```

Figure 4.1 Script du nombre moyen d'appels dans le central téléphonique

b) Temps moyen d'attente

```
1  function temmoysk=tempmoyen(mu, lamda, s, k)
2  -   moyen=appl(mu, lamda, s, k)/(lamda*(k-appl(mu, lamda, s, k)));
3  -   temmoysk=moyen;
4  - end
```

Figure 4.2 Script du temps moyen d'attente dans le central téléphonique

Voici à présent le programme qui nous permet d'obtenir le nombre de canaux nécessaires pour que la probabilité de saturation soit inférieure à α donné, Et en plus, on obtient à chaque fois le nombre moyen d'appels et le temps d'attente moyen dans le central téléphonique en fonction de nombre minimum de canaux obtenus.

```

1 - function probadesatu=satu(alpha)
2 -     som1=0;
3 -     nb=1;
4 -     k=100; lamda=13; mu=7;
5 -     for n=nb:k
6 -         som1=som1+(1/fact(k-n))*((lamda/(nb*mu))^n);
7 -     end
8 -     probadesatu=(fact(k)/fact(nb))*(nb^nb)*p0sk(mu,lamda,nb,k)*som1;
9 -     if (probadesatu<=alpha)
10 -         s=nb;
11 -         nombre de canaux s=s
12 -         nombre moyen appels=appl(mu,lamda,s,k)
13 -         temps moyen attente=tempmoyen(mu,lamda,s,k)
14 -         probabilite de saturation=probadesatu
15 -     else while (probadesatu>=alpha)
16 -         nb=nb+1;
17 -         som1=0;
18 -         for n=nb:k
19 -             som1=som1+(1/fact(k-n))*((lamda/(nb*mu))^n);
20 -         end
21 -         p=(fact(k)/fact(nb))*(nb^nb)*p0sk(mu,lamda,nb,k)*som1;
22 -         probadesatu=p;
23 -         if (probadesatu<=alpha)
24 -             s=nb;
25 -             nombre de canaux s=s
26 -             nombre moyen appels=appl(mu,lamda,s,k)
27 -             temps moyen attente=tempmoyen(mu,lamda,s,k)
28 -             probabilite de saturation=probadesatu
29 -         end
30 -     end
31 - end
32 - end

```

Figure 4.3 Code d'optimisation de risque de saturation

On obtient le résultat suivant

```

>> satu(0.1)

nombre_de_canaux_s =

    72

nombre_moyen_appels =

    65.0557

temps_moyen_attente =

    0.1432

probabilite_de_saturation =

    0.0892

```

Figure 4.4 Résultat d'exécution du script d'optimisation en fonction des paramètres donné

4.2.3 Illustrations numérique et graphique

Ce tableau ci-dessous nous donne le nombre minimum de canaux pour différents alpha, avec la probabilité de saturation, le nombre moyen d'appels et le temps moyen d'attente dans le central téléphonique en fonction du nombre minimum de serveurs trouvé.

Alpha	Probabilité de saturation	Nombre de canaux	Nombre moyen d'appels	Temps moyen d'attente
0.001	0	80	NAN	NAN
0.01	0.0068	77	65.0025	0.1429
0.02	0.0125	76	65.0050	0.1429
0.07	0.0538	73	65.0325	0.1431
0.1	0.0892	72	65.0557	0.1432
0.2	0.1846	70	65.1459	0.1438
0.7	0.6862	64	66.1922	0.1506
0.9	0.8879	61	67.3439	0.1586

Tableau 1 Illustrations numériques des caractéristiques du système en fonction du « s » obtenu

On trace les courbes suivantes à partir du tableau ci-dessus

a) Courbe de la probabilité de saturation en fonction de nombre minimum de canaux

```
>> canaux=[61 64 70 72 73 76 77 80];  
>> prob_sat=[0.8879 0.6862 0.1846 0.0892 0.0538 0.0125 0.0068 0];  
>> plot(canaux,prob_sat)  
>> xlabel('nombre min de canaux')  
>> ylabel('risque de saturation')  
>> grid on
```

Figure 4.5 Code du risque de saturation en fonction du « s » obtenu

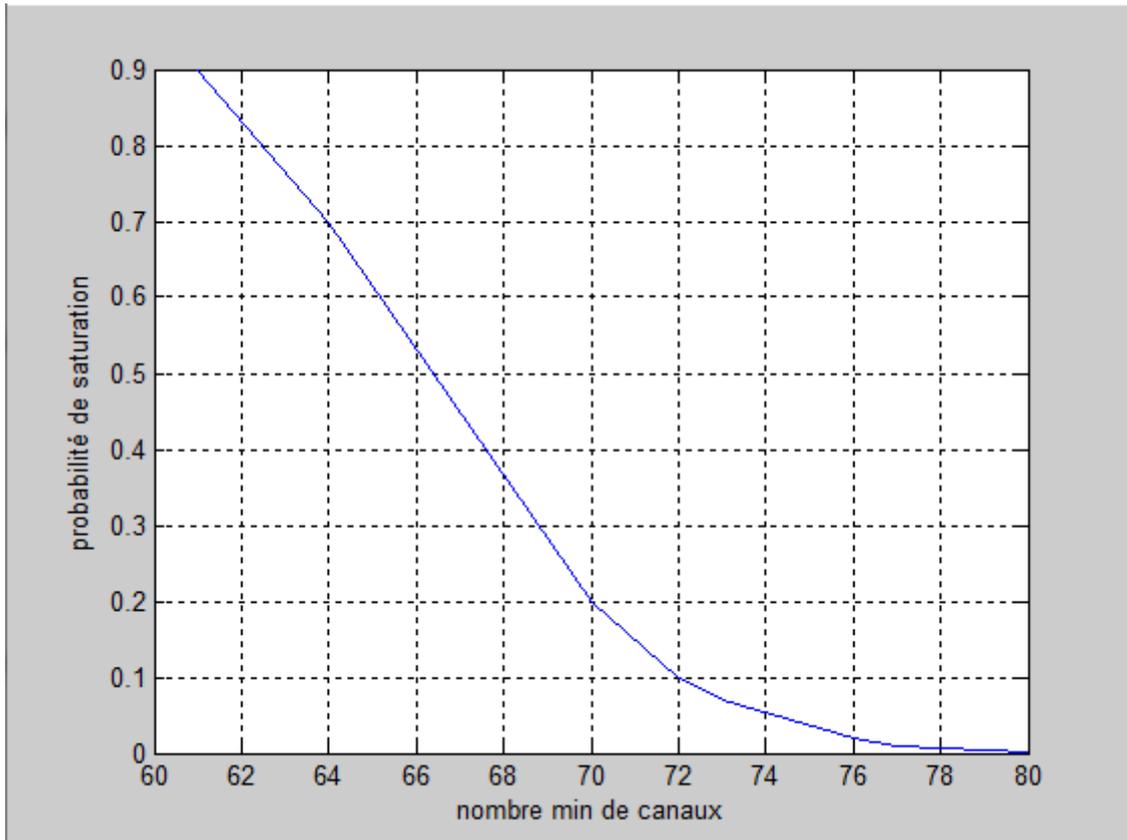


Figure 4.6 Probabilité de saturation en fonction du nombre minimum de canaux

On s'aperçoit que la probabilité de saturation diminue lorsque le nombre de canaux augmente jusqu'à son annulation à partir de 80 canaux.

b) Courbe de probabilité de saturation en fonction d'alpha

```
>> alpha=[0.0001 0.01 0.02 0.07 0.1 0.2 0.7 0.9];  
>> proba_satu=[0 0.0068 0.0125 0.0538 0.0892 0.1846 0.6862 0.8879];  
>> plot(alpha,proba_satu)  
>> grid on  
>> xlabel('alpha')  
>> ylabel('probabilité de saturatuion')
```

Figure 4.7 Code de probabilité de saturation en fonction de α

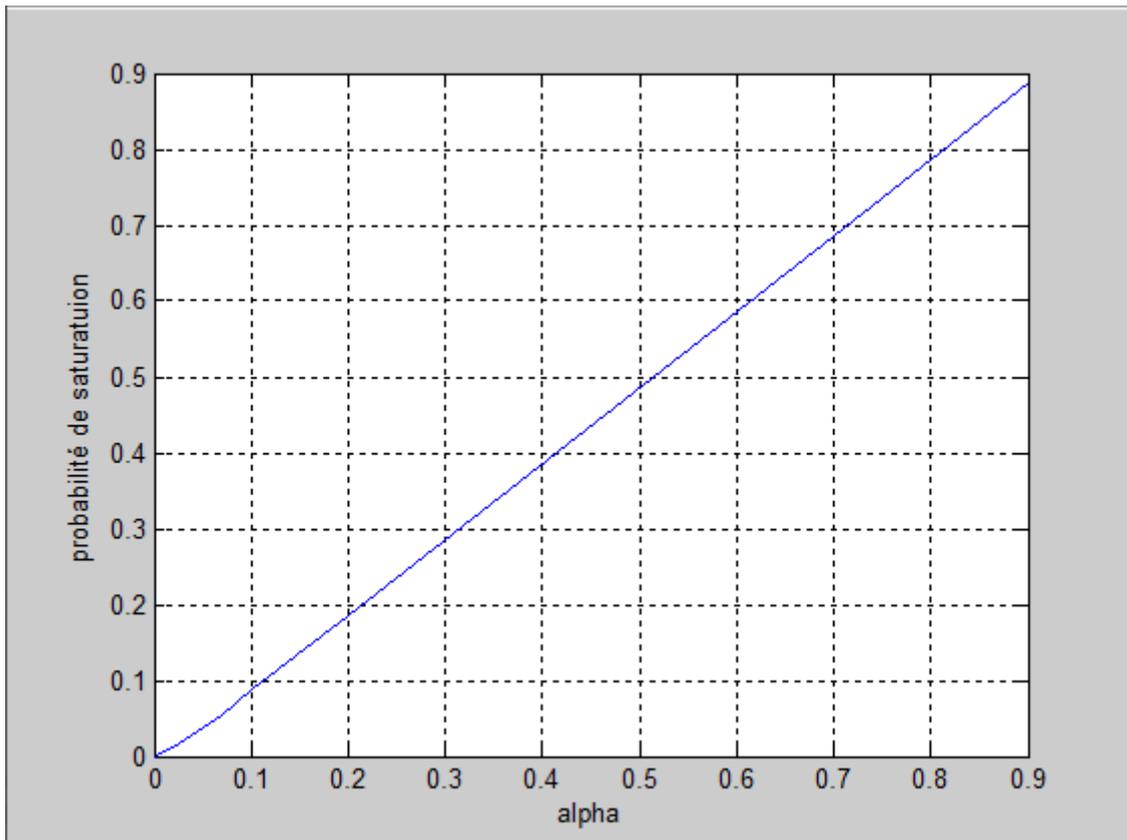


Figure 4.8 Probabilité de saturation en fonction α

c) Courbe du temps d'attente moyen en fonction de nombre minimum de canaux

```
>> canaux=[77 76 73 72 70 64 64];
>> temps_attente=[0.1429 0.1429 0.1431 0.1432 0.1438 0.1506 0.1586];
>> plot(canaux,temps_attente)
>> grid on
>> xlabel('nombre min de canaux')
>> ylabel('temps d"attente')
```

Figure 4.9 Code du temps d'attente en fonction de s

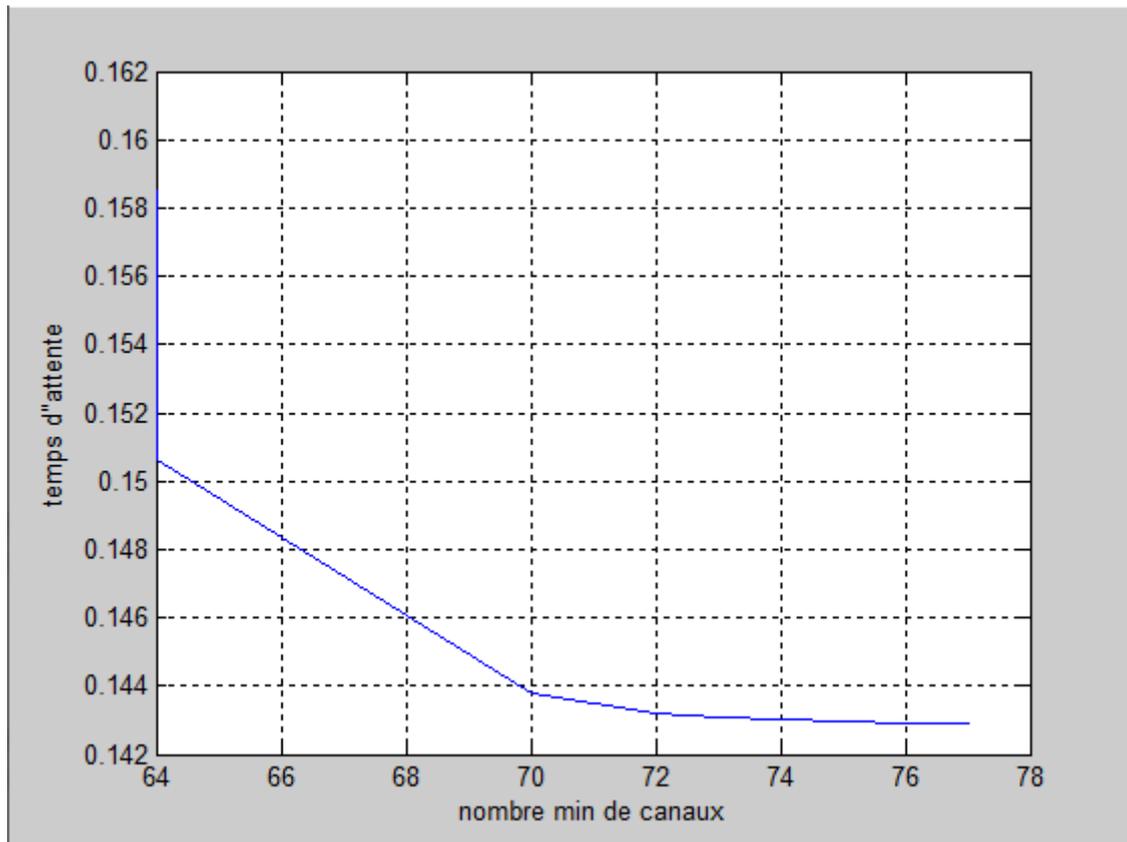


Figure 4.10 Temps moyen d'attente en fonction de s

On remarque que le temps moyen d'attente décroît d'une manière considérable lorsque le nombre de canaux $s \in \{64, 65, \dots, 70\}$.

4.3 File d'attente M/M/s/s/ [F]

4.3.1 Formulation du problème d'optimisation

On a

$$\pi(\rho, s, K) = C_K^s \rho^s p_0.$$

Nous cherchons à optimiser la probabilité de saturation $\pi(\rho, s, K)$ par rapport au nombre de canaux, c'est-à-dire

$$\min_s \pi(\rho, s, K) = \min_s (C_K^s \rho^s p_0) \text{ avec } s \in \mathbb{N}^*$$

$$\Rightarrow \begin{cases} \min(C_K^s \rho^s p_0) \\ s \in \mathbb{N}^* \end{cases}$$

4.3.2 Optimisation de la probabilité de saturation

On recherche d'une manière informatique, le nombre minimum de canaux nécessaires pour que la probabilité de saturation ne dépasse pas un seuil alpha donné, avec $\alpha \in [0, 1]$. On suppose que $\lambda=13$, $\mu=7$, et $k=100$.

D'abord, on donne le script sous MATLAB du nombre moyen d'appels et le temps d'attente moyen dans le central téléphonique

a) Nombre moyen d'appels

```
1  function nombremoy = ap(mu, lamda, k)
2  -      moyen=(k*lamda) / (mu+lamda);
3  -      nombremoy=moyen;
4  -  end
```

Figure 4.11 Script du nombre moyen d'appel dans le central téléphonique

b) Temps moyen d'attente

```
1  function tempsat=attente(mu)
2  -      tempsat=1/mu;
3  -  end
```

Figure 4.12 Code du temps moyen d'attente dans le central téléphonique

Voici à présent le programme qui nous permet d'obtenir le nombre de canaux nécessaires pour que la probabilité de saturation soit inférieure à alpha donné, de plus, on obtient à chaque fois le nombre moyen d'appels et le temps d'attente moyen dans le central téléphonique en fonction de nombre minimum de canaux obtenus.

```

1  function probadesatus=satus(alpha)
2  -     nb=1;
3  -     k=100;
4  -     mu=7;
5  -     lamda=13;
6  -     som=(fact(k)/(fact(nb)*fact(k-nb)))*((lamda/mu)^nb);
7  -     probadesatus=som* pOfs(k,nb,mu,lamda);
8  -     if (probadesatus<=alpha)
9  -         s=nb;
10 -         nombre de canaux s=s
11 -         nombre moyen appels=(k*lamda)/(mu+lamda)
12 -         temps moyen attente=1/mu
13 -         probabilite de saturation=probadesatus
14 -     else while (probadesatus>=alpha)
15 -         nb=nb+1;
16 -         som=(fact(k)/(fact(nb)*fact(k-nb)))*((lamda/mu)^nb);
17 -         probadesatus=som* pOfs(k,nb,mu,lamda);
18 -         if (probadesatus<=alpha)
19 -             s=nb;
20 -             nombre de canaux s=s
21 -             nombre moyen appels=(k*lamda)/(mu+lamda)
22 -             temps moyen attente=1/mu
23 -             probabilite de saturation=probadesatus
24 -         end
25 -     end
26 - end
27 - end

```

Figure 4.13 Code d'optimisation du risque de saturation

On obtient le résultat suivant

```

>> satus(0.1)

nombre_de_canaux_s =

    68

nombre_moyen_appels =

    65

temps_moyen_attente =

    0.1429

probabilite_de_saturation =

    0.0910

```

Figure 4.14 Résultat d'exécution du script d'optimisation en fonction des paramètres donnés

4.3.3 Illustrations numérique et graphique

Le tableau ci-dessous nous donne le nombre minimum de canaux pour différents alpha, avec la probabilité de saturation, le nombre moyen d'appels et le temps moyen d'attente dans le central téléphonique correspondant aux canaux.

Alpha	Probabilité de saturation	Nombre de canaux	Nombre moyen d'appels	Temps moyen d'attente
0.001	0	79	65	0.1429
0.01	0.0091	75	65	0.1429
0.02	0.0143	74	65	0.1429
0.07	0.0564	70	65	0.1429
0.1	0.0910	68	65	0.1429
0.2	0.1785	64	65	0.1429
0.7	0.6943	37	65	0.1429
0.9	0.8995	16	65	0.1429

Tableau 2 Illustration numérique des caractéristiques du système en fonction du « s » obtenu

On trace à partir du tableau ci-dessus les courbes suivantes

a) Courbe de probabilité de saturation en fonction du nombre minimum de canaux

```

>> canaux=[16 37 64 68 70 74 75 79];
>> proba_satu=[0.8995 0.6943 0.1785 0.0910 0.0564 0.0143 0.0091 0];
>> plot(canaux, proba_satu,'r')
>> grid on
>> xlabel('nombre min de canaux')
>> ylabel('probabilite de saturation')

```

Figure 4.15 Code de probabilité de saturation en fonction de s

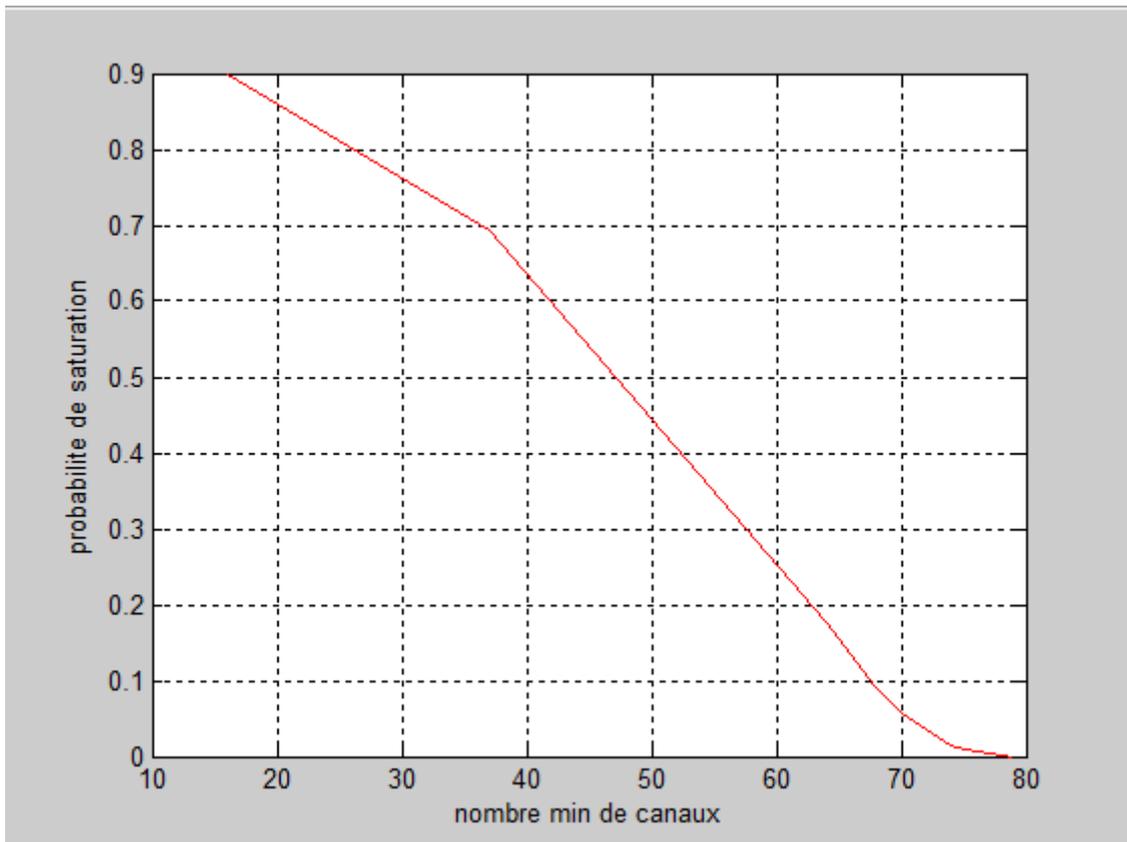


Figure 4.16 Graphe de probabilité de saturation en fonction du nombre minimum de canaux

On remarque que la probabilité de saturation s'annule à partir de 107 canaux après une diminution importante à partir de 106 canaux.

b) Courbe de probabilité de saturation en fonction α

```

>> alpha=[0.001 0.01 0.02 0.07 0.1 0.2 0.7 0.9];
>> proba_satu=[0 0.0091 0.0143 0.0564 0.0910 0.1785 0.6943 0.8995];
>> plot(alpha, proba_satu,'r')
>> grid on
>> xlabel('alpha')
>> ylabel('probabilite de saturation')

```

Figure 4.17 Script de la courbe de probabilité de saturation en fonction de α

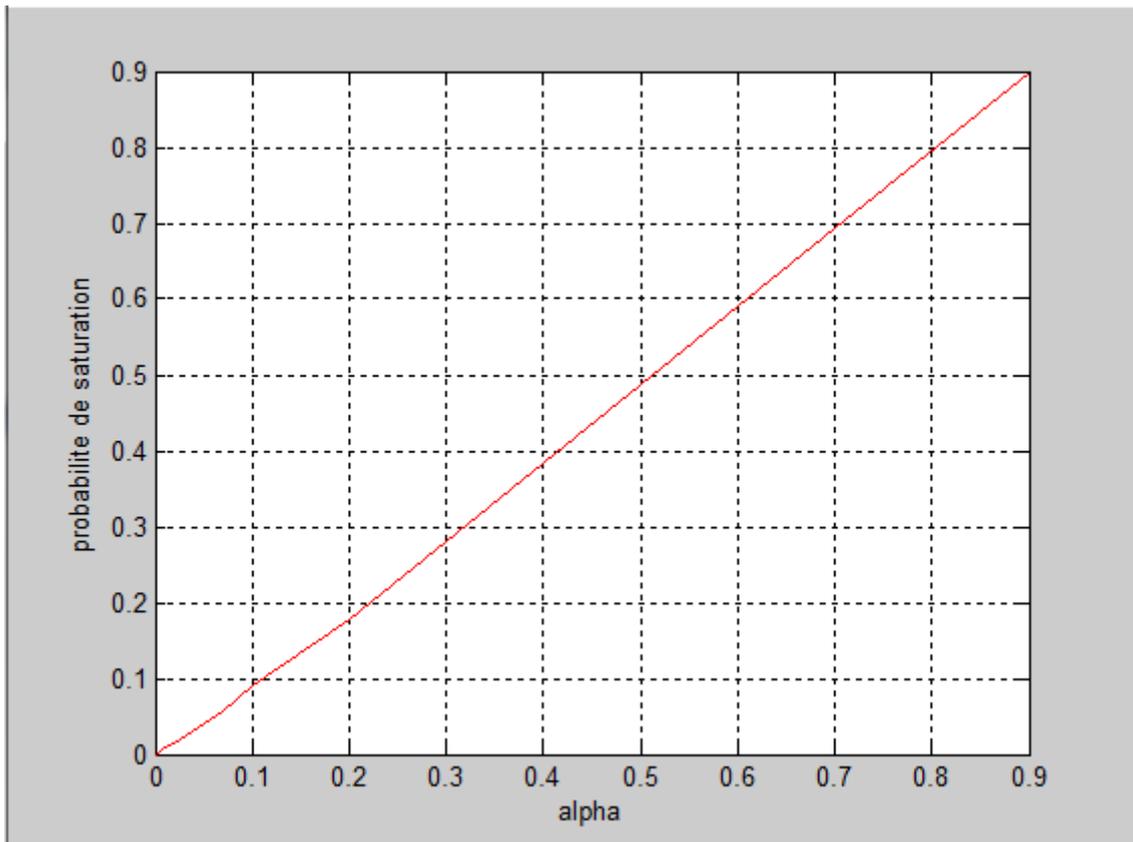


Figure 4.18 Graphe probabilité de saturation en fonction d'alpha

4.4 Conclusion

L'optimisation informatique des systèmes d'attente nous a donné le nombre de canaux nécessaires pour s'assurer du bon fonctionnement et un meilleur service aux clients, ainsi permet à l'entreprise de prendre une bonne décision.

Conclusion générale

Dans ce travail, nous avons introduit et illustré l'utilité de la théorie des files d'attente, cette dernière nous a permis de montrer l'importance de notre étude, qui consiste à prédire le comportement des systèmes d'attente dans les réseaux de télécommunication, en particulier les appels téléphoniques (appels entrants dans un central téléphonique). Ainsi, on a présenté une étude approfondie sur les systèmes $M/M/s/K/[F]$ et $M/M/s/s/[F]$ et leurs performances, ce qui a permis leurs simulation avec le logiciel solveur MATLAB.

Les files d'attente permettent de gérer la congestion du système tout en optimisant certaines performances (dans notre cas, minimiser le risque de saturation), afin de déterminer le nombre de canaux nécessaires pour s'assurer d'une bonne qualité de service, tel qu'il est donné dans le quatrième chapitre.

Ces modèles de files d'attente peuvent modéliser d'autres problèmes réels physiques similaires aux phénomènes d'attente dans le central téléphonique (non physique), on peut citer quelques exemples

- Atelier de réparation des machines dans une usine.
- Arrivées des bateaux dans un port.
- Gestion du service clients chez un opérateur téléphonique.

De nos jours, vu l'apparition des nouvelles technologies très puissantes comme la fibre optique reliant les équipements téléphoniques (centraux téléphoniques), cela nous permet de diminuer d'une manière importante la probabilité de saturation.

Bibliographie

- [1] Cours master 1 RO : Processus stochastiques, M^r. HAMADOUCHE.
- [2] Cours master 2 RO : Files d'attente, M^r. HAMADOUCHE.
- [3] Abdelhamid Zaidi (2012), probabilités et statistique à l'usage de l'ingénieur.
- [4] Babes Malika (1995), statistiques, files d'attente et simulation.
- [5] Sabin Lessard (2014), processus stochastiques (cours et exercices corrigés).
- [6] Jacques Teghem (2013), recherche opérationnelle tome 2 (gestion de production modèles aléatoires aide multicritère).
- [7] Adrion Biran et Moshe Breiner (2004 version 6 et 7), MATLAB pour l'ingénieur.
- [8] Bechar Karim et Baahmed Ahmed (2016/2017), Mémoire de master, optimisation dans les systèmes de files d'attente (cas de gestion des arrivées des avions dans un aéroport).