

Evaluation de performances

YeQiong SONG, LORIA – Université de Lorraine / (song@loria.fr)

Master Informatique, Spécialité SSR,
UE Réseaux avancés,
Support de cours pour la partie évaluation de performances

Evaluation de performances

YeQiong SONG, ENSEM-INPL / LORIA (song@loria.fr)

Table des matières :

1	Introduction	2
2	Rappel des Chaînes de Markov et des files d'attente.....	8
3	File M/G/1	28
4	Evaluation des protocoles/réseaux	34
5	Evaluation des protocoles d'accès aléatoire.....	44
6	Annexe : Modélisation et simulation	51
	Bibliographie.....	75
	Annexe: TP Evaluation de performances par simulation en JMT	76

1 Introduction

L'évolution permanente des systèmes et réseaux informatiques et des télécommunications souligne le besoin croissant d'outils facilitant l'étude de leur comportement. Il est nécessaire d'avoir une assistance dans les phases de conception (comparaison des solutions), de développement (dimensionnement) et d'utilisation (gestion des flux et de la QoS : Quality of Service) des systèmes et réseaux. En effet, le développement d'un système complexe demande non seulement une modélisation qualitative pour vérifier sa correction logique mais aussi une validation a priori des performances du système lors de la phase de conception. En plus, lorsque ces systèmes possèdent des contraintes temporelles (applications temps réel, par exemple), nous devons inclure parmi les éléments à retenir des considérations de performances qui ne peuvent être abordées avec rigueur que grâce à l'utilisation de techniques quantitatives pour l'évaluation de performances.

Parmi de nombreux paramètres de performances d'un réseau nous nous intéressons dans ce document essentiellement aux deux suivants :

- le **temps de réponse** d'un message qui exprime le délai nécessaire pour qu'un message traverse un équipement réseau (contrôleur de communication, commutateur, ...) ou un réseau entier (temps de réponse de bout en bout). Ce paramètre intéresse surtout les utilisateurs (applications temps réel par exemple) et constitue le paramètre principal de qualité de service (QoS) d'un réseau.
- le **rendement** (throughput ou encore débit) d'un réseau qui exprime le rapport entre la quantité d'information (par seconde) et la quantité totale d'information (données utilisateurs + données de contrôle + données retransmises en cas d'erreurs) véhiculée par le réseau. Ce paramètre est généralement utilisé pour estimer l'efficacité d'un réseau. A ne pas confondre avec le *débit nominal* d'un réseau (e.g. 10 Mbit/s, 100 Mbit/s, ...).

Les réseaux informatiques et de télécommunications numériques sont conçus comme des ressources à partager par plusieurs utilisateurs. Afin de gérer le problème d'accès multiples à

une ressource, on utilise une mémoire tampon pour stocker momentanément des demandes que l'on n'arrive pas à traiter instantanément. Dans un équipement réseau, chaque fois il y a une mémoire tampon, il y a des attentes de traitement. Il est donc naturel de modéliser un réseau sous le formalisme de files d'attente. Ces délais d'attente dépendront de l'ensemble des trafics entrants. Selon la connaissance complète ou partielle dont on dispose sur l'application, leur évaluation peut donner des résultats du type exact, probabiliste ou de bornes. La figure suivante (Figure 1) met en évidence les attentes dans un réseau.

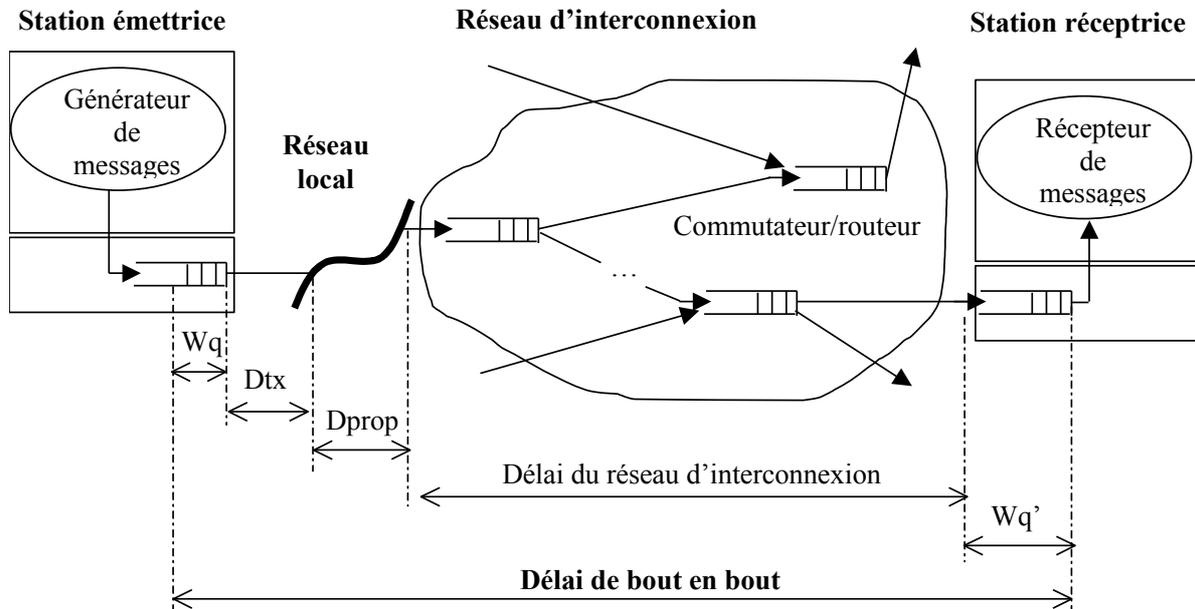
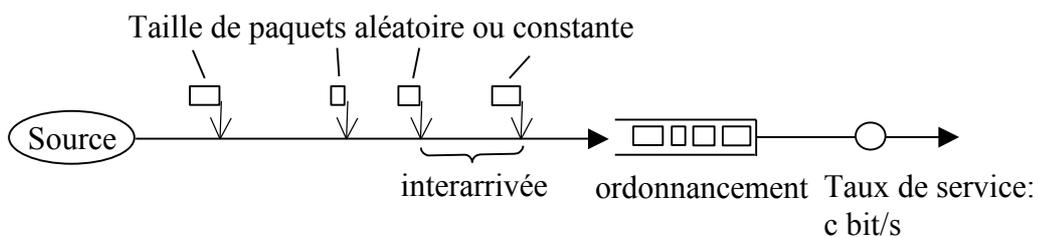


Figure 1- Les attentes dans un réseau de transmission de données

Examinons l'exemple suivant d'un équipement réseau de capacité de traitement c bit/s :



L'objectif de l'évaluation de performances est de fournir une garantie sur le temps de traversée du système d'un paquet.

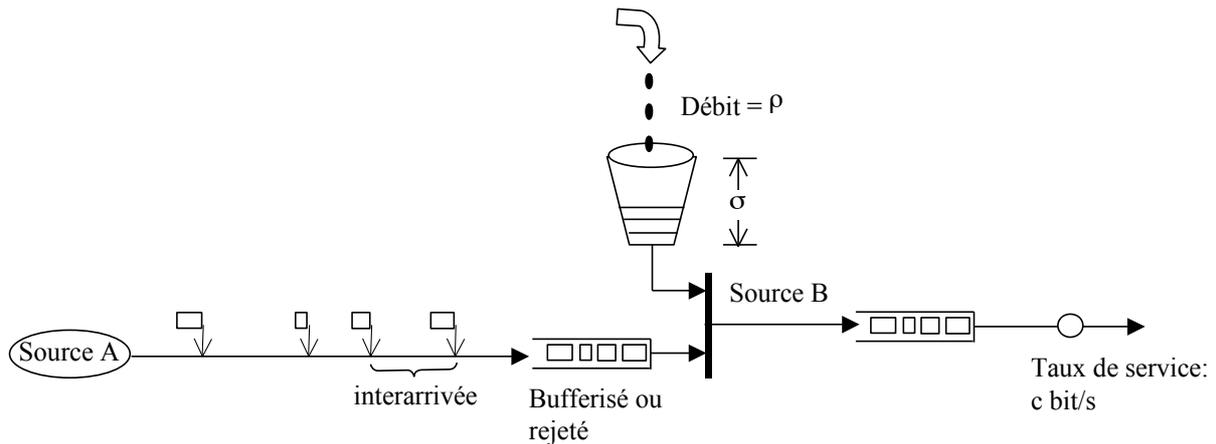
Si on ne dispose pas de connaissance sur le trafic d'entrée dans la file d'attente, aucune garantie ne sera possible.

Si on dispose d'une connaissance stochastique sur le trafic, une analyse à la base de la théorie des files d'attente fournira des résultats utilisables pour fournir une garantie probabiliste ou en termes de temps moyen de traversée. Deux cas particuliers sont couramment rencontrés :

- Si l'interarrivée des paquets est distribuée exponentiellement et les tailles de paquets suivent une distribution exponentielle, les résultats sur une file d'attente M/M/1 sont directement exploitables.

- Si l'interarrivée des paquets est distribuée exponentiellement et la taille des paquets est constante, ce système pourra être modélisé par une file M/D/1.

Maintenant si on veut une garantie déterminite, il va falloir disposer plus de connaissance sur le trafic d'entrée. Par exemple, trafic périodique ou périodique avec des giques (cas étudiés dans l'ordonnancement). Mais dans la pratique, ce type de trafic est souvent difficile à obtenir dans des réseaux de taille importante à cause des comportements en général non déterministes des équipements réseaux (asynchrones). C'est pour cela que l'on peut implémenter un limiteur de trafic (Leaky bucket) afin de « lisser » un flux d'entrée.



Notons que la garantie est donnée vis à vis de la source B mais non plus la source A.

La source B est dite (σ, ρ) -borné avec σ la taille de rafale et ρ le débit moyen. Le trafic entrant durant $[0, t]$ est alors borné par :

$$B(t) = \sigma + \rho t$$

Dans ce document, nous nous intéressons à étudier des approches d'évaluation de performances probabilistes. Le lecteur intéressé par l'évaluation de performances déterministes avec un trafic d'entrée (σ, ρ) -borné peut consulter des références ayant trait avec le « network calculus » [LeBoudec02].

Que ce soit dans un réseau à un seul saut (un réseau local à médium partagé par exemple) ou dans un grand réseau à sauts multiples (Internet par exemple), le point clé pour le calcul du temps de réponse de bout en bout est l'évaluation du temps de traversé de chaque nœud. Ce nœud peut être le médium de transmission partagé par des stations multiples dont l'accès est régulé par le protocole MAC (Medium Access Control) utilisé. Ce nœud peut aussi être un multiplexeur, un commutateur ou un routeur dont un même port de sortie est partagé par toutes les lignes en entrée. Un modèle appelé MIQSS (Multiple Input Queues Single Server) peut être utilisé pour décrire un tel système (Figure 2).

Le modèle MIQSS est constitué d'un ensemble de N sources $\{S_i\}_{1 \leq i \leq N}$ de **clients** (qui peuvent représenter des tâches dans un CPU ou des messages dans un réseau) partageant le serveur unique de capacité de traitement c (en bit/s pour la transmission de messages par exemple). Une source S_i est caractérisée par le flux de clients F_i qu'elle génère et les contraintes de performances (ici on ne considère que celle du temps de réponse) CTR_i qu'elle doit respecter.

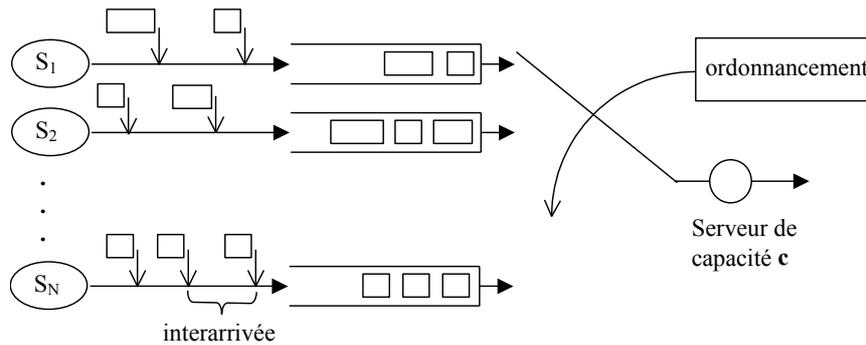


Figure 2. Modèle général pour la description des systèmes : MIQSS

La fonction F_i peut être :

- **Périodique ou sporadique**: elle est alors définie par (C_i, T_i) dans le cas d'une date de génération du premier client quelconque et (r_i, C_i, T_i) dans le cas où la date de génération du premier client notée par r_i est donnée ; C_i est le temps d'exécution d'un client par le serveur et T_i la période d'interarrivée (ou d'interarrivée minimale dans le cas sporadique). Dans la plupart du temps on considère le cas où C_i est constante. Le cas de C_i variable est souvent majoré par le temps d'exécution maximale (WCET : Worst Case Execution Time).
- **Périodique avec giges** : elle est caractérisée par (C_i, T_i, J_i) ou (r_i, C_i, T_i, J_i) où J_i représente la gigue. La gigue est considérée comme étant le déphasage maximal d'une arrivée par rapport à la période. Dans ce document la gigue a toujours une valeur positive (i.e., un retard par rapport à la période)
- **(σ_i, ρ_i) -borné** : elle est alors donnée sous forme d'une courbe linéaire caractérisée par la taille de la rafale d'arrivées des clients σ_i et le débit moyen ρ_i qui majore la vraie fonction cumulative d'arrivée du travail [LeBoudec02], [Chang00]. La quantité du travail apportée par un client est définie par W_i avec notamment $C_i = W_i / c$.
- **Aléatoire** en suivant une loi d'arrivée probabiliste avec un temps moyen d'inter-arrivée T_i et un temps moyen d'exécution de clients C_i (par exemple : une loi de Poisson pour décrire une loi d'inter-arrivée exponentielle avec la taille constante des clients, une loi de Poisson composée pour une loi d'inter-arrivées exponentielle et des tailles variables de clients).

Le flux d'arrivée périodique ou sporadique, avec ou sans giges correspond aux modèles de tâches classiques utilisés par la communauté de l'ordonnancement temps réel [Liu73]. Le flux (σ, ρ) -borné apparaît, quant à lui, plutôt dans la communauté de la QoS des réseaux (ATM et Internet) [Cruz91a], [Cruz91b], [LeBoudec02], [Chang00]. Un tel flux est obtenu par un lisseur de trafic (Leaky bucket par exemple). Un flux aléatoire est un modèle classique du domaine de l'évaluation de performances, notamment lorsqu'il s'agit de l'évaluation de performances des réseaux par le formalisme de files d'attente [Kleinrock75], [Kleinrock76]. Notons que les deux premiers cas peuvent être transposés en un modèle de flux (σ_i, ρ_i) -borné [Koubâa04]. Ceci ouvre la possibilité de choisir entre la théorie de l'ordonnancement et celle du « network calculus » [LeBoudec02] lors de l'évaluation de la borne supérieure des temps de réponse.

Les contraintes de temps de réponse CTR_i sont classiquement données par D_i qui est l'échéance relative à l'instant d'arrivée d'un client, et la garantie exigée (déterministe, probabiliste, en moyenne, ...).

Pour analyser ce modèle de file d'attente, nous devons nous intéresser à la caractérisation des demandes d'accès (exacte, probabiliste, fonction majorante de flux d'arrivée: pire trajectoire ou (σ, ρ) -borné), à l'étude des algorithmes d'ordonnancement

(qui ordonnent l'accès des messages aux ressources de transmission partagées), et aux méthodes d'analyse de temps de réponse. Les principaux formalismes/outils que nous pouvons utiliser sont :

- les **processus stochastiques** et la **théorie des files d'attente** pour une caractérisation probabiliste de flux d'arrivées et de temps de réponse
- les méthodes d'analyse à base de trajectoires (une caractérisation déterministe de flux d'arrivées) :
 - trajectoire du pire cas pour l'obtention du temps de réponse du pire cas (par WCRTA, **Worst Case Response Time Analysis**, la technique issue de la communauté temps réel)
 - flux d'arrivée majoré par (σ, ρ) pour l'obtention de la borne supérieure du temps de réponse (par **Network Calculus**, technique issue de la communauté de QoS de l'Internet)
- la **simulation à événements discrets** pour des systèmes complexes ne pouvant pas être étudiés facilement par des méthodes analytiques (trafic non Markovien ou qui ne peuvent être caractérisés de façon stochastique ou par des bornes du type pire cas ou (σ, ρ))

Sans perdre de la généralité, dans ce document, nous allons nous concentrer dans la plupart du temps sur l'évaluation du temps d'attente d'un message dans une file d'attente à un seul serveur, que ce dernier soit intermittent (cas du TDMA) ou non (Figure 2).

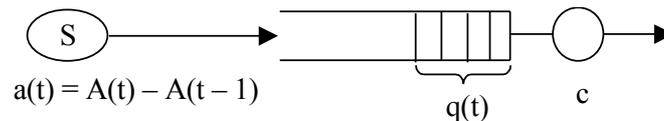


Figure 2 Modèle de base pour évaluer le temps de réponse de messages d'une source S

Notons que la source S peut représenter une des sources si les sources sont de priorités différentes ou une agrégation d'un ensemble de sources si les sources sont de même priorité. Dans le premier cas le serveur est intermittent dont la disponibilité est décrite par l'ordonnement utilisé (Round-Robin, Fixed Priority, WFQ, ...). Le temps de réponse obtenu est vis à vis d'une source particulière. Dans le second cas (par exemple : commutateur/routeur dans un réseau IP ne supportant pas de classification de trafics), le serveur est non-vacant et le temps de réponse (d'un message quelconque) est le même pour toutes les sources.

Dans un premier temps, pour simplifier la description mathématique de ce modèle de base nous supposons que les messages sont de même taille dont le temps de traitement dans le serveur est une unité de temps. Le temps d'attente d'un message dans une telle file (appelé aussi temps de réponse) dépend essentiellement du flux d'entrée en terme du nombre de messages par unité de temps $a(t) = A(t) - A(t-1)$, de la politique d'ordonnement (FIFO, avec priorité, ...) et de la capacité du serveur (c messages par unité de temps). L'évolution de notre modèle de base en termes de nombre de messages (un système à événements discrets) $q(t)$ se décrit avec l'équation de Lindley [LIN 52]:

$$q(t+1) = \max(0, q(t) + a(t+1) - c) \quad (1)$$

Cette équation est parfois écrite sous forme de : $q(t+1) = (q(t) + a(t+1) - c)^+$ avec $x^+ = \max(0, x)$.

Schématiquement l'évolution de $q(t)$ peut être décrite par la Figure 3.

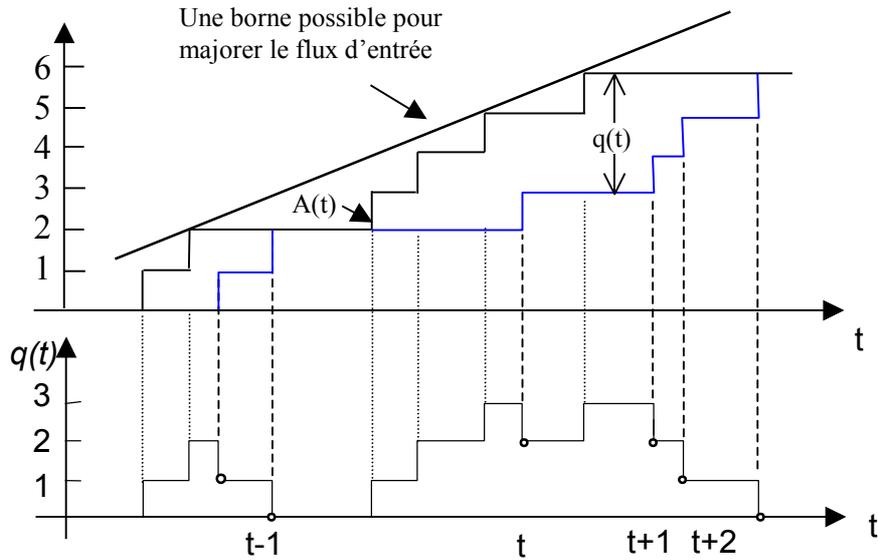


Figure 3 Une trajectoire de l'évolution du nombre de messages dans la file

Notons que l'hypothèse de taille constante de messages ne réduit pas la généralité de l'équation de Lindley. Au pire, on peut définir la quantité de travail $q(t)$ comme le nombre de bits dans la file d'attente, c le nombre de bits que le serveur peut traiter par unité de temps et $A(t)$ le nombre de bits générés par la source S pendant l'intervalle $[0, t[$.

Dans la suite de ce document nous allons nous intéresser à l'approche de résolution analytique des modèles en files d'attente. Une introduction à l'approche simulation est également donnée à la fin sous forme d'une annexe.

2 Rappel des Chaînes de Markov et des files d'attente

2.1 Notion de files d'attente

Basé sur deux entités: *client* et *serveur* (un client arrive dans une file, attend un service, reçoit le service, puis se dirige vers une autre file ou quitte le système), ce formalisme est largement utilisé pour modéliser des systèmes dans la vie quotidienne (acheter un ticket de cinéma au guichet, passer la caisse de supermarché, etc.), dans l'informatique (une tâche attend dans la RAM pour être traitée par la CPU), dans les télécommunications (un message dans la mémoire tampon de la carte réseau en attendant que le médium de transmission soit libre, dans un commutateur en attendant d'être acheminé vers une bonne ligne de sortie), dans le transport (voiture devant un péage, un feu tricolore), dans la gestion de production dans une usine, etc. La Figure 4 représente schématiquement les éléments d'un système d'attente.

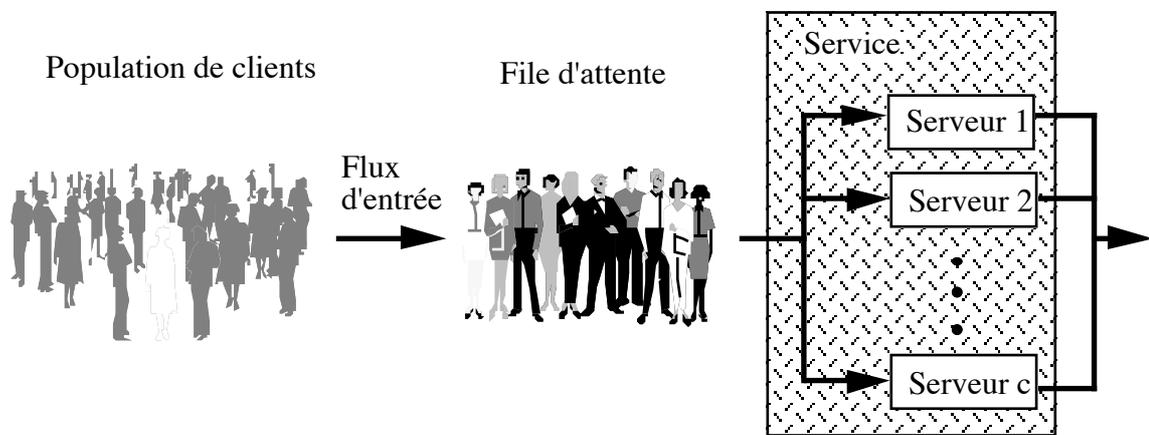


Figure 4 - Éléments d'un système d'attente

Ce système peut être modélisé en termes de files d'attente comme montré par la Figure 5 avec les variables qui peuvent nous intéresser.

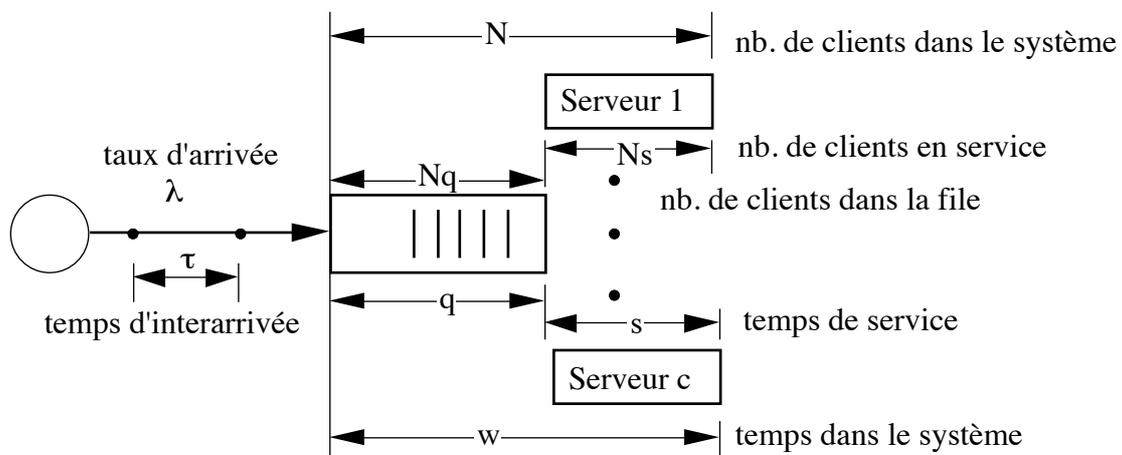


Figure 5 - Quelques variables aléatoires utilisées dans un modèle en files d'attente

2.2 Caractérisation d'une file d'attente

2.2.1 Notation de Kendall

Une file d'attente est notée $A/B/m/K/N/Z$ selon Kendall avec:

- A : Processus d'arrivée des clients (distribution d'interarrivée).
- B : Schéma de service (distribution de durée de service de clients).
- m : nombre de serveurs.
- K : capacité maximale de la file d'attente.
- N : nombre de clients utilisant le système.
- Z : discipline du système qui décrit la façon dont les clients sont ordonnancés.

Pour les arrivées et les services on utilise les symboles suivants:

- M : loi exponentielle.
- D : loi déterministe.
- G : loi générale.
- H_k : loi hyperexponentielle d'ordre k .
- E_k : loi d'Erlang d'ordre k .

Le nombre de serveurs peut varier de 1 à l'infini (noté ∞), de même pour K et N .

En ce qui concerne Z , les ordonnancements les plus utilisés sont:

- FIFO (First In, First Out).
- LIFO (Last In, First Out).
- RANDOM: le prochain client à servir est choisi aléatoirement.
- Round-Robin (cyclique avec un quantum Q).
- PS (Processor Sharing): Round-Robin avec Q tend vers 0. Tous les n clients servis en même temps avec un taux μ/n .
- PRIOR (Avec priorité, avec préemption ou sans préemption).

2.2.2 Résultats généraux

Nous donnons ici quelques résultats généraux valables aussi bien sur une file d'attente simple que sur un réseau de files d'attente.

Considérons la file simple suivante avec la capacité infinie de la file :

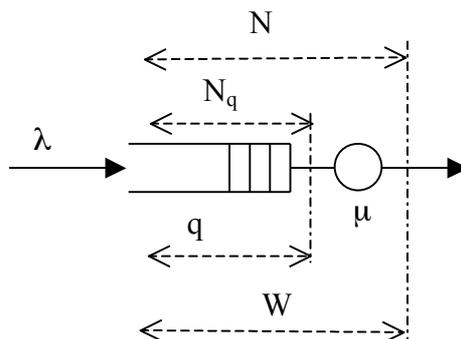


Figure 6 - File d'attente simple

Condition de stationnarité :

$$\lambda < \mu$$

qui s'exprime simplement que le système est stable si le taux d'arrivée des clients est inférieur au taux avec lequel le système évacue les clients.

Cette condition est aussi exprimée par la charge normalisée du système soit inférieure à 1 : $\rho < 1$.

Remarque 1 : dans le cas particulier d'une file D/D/1, la condition stationnaire est : $\lambda \leq \mu$

Remarque 2 : dans une file avec une capacité limitée, les clients arrivés voyant la file pleine sont rejetés. Le système est donc toujours stable.

Formules de LITTLE :

$$E(N) = \lambda E(W)$$

et

$$E(N_q) = \lambda E(q)$$

qui s'exprime simplement que le nombre moyen de client dans un système est égal au taux d'arrivées des clients multiplié par le temps moyen d'attente d'un client.

La relation suivante existe :

$$E(W) = E(q) + 1/\mu$$

2.3 Rappel sur la probabilité

2.3.1 Probabilité élémentaire

- Ω : l'univers des événements
- A (ou ω) : un sous-ensemble de Ω
- $P[\Omega] = 1$, $P[\emptyset] = 0$
- Les opérations des ensembles classiques

$$P[A \cup B] = \begin{cases} P[A] + P[B], & \text{si } A \cap B = \emptyset \\ P[A] + P[B] - P[A \cap B], & \text{sinon} \end{cases}$$

$$P[A \cap B] = P[A] \cdot P[B], \quad \text{si } A \text{ et } B \text{ sont indépendantes}$$

Probabilité conditionnelle :

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

Probabilité totale :

$$\bigcup_{i \in E} A_i = \Omega; \quad A_i \cap A_j = \emptyset, \quad \forall i \neq j$$

$$P[B] = \sum_{i \in E} P[B|A_i] P[A_i]$$

2.3.2 Variables aléatoires

Une v.a. X prend la valeur à l'issue d'une expérience aléatoire (e.g. loto).

A chaque événement ω de Ω , on associe un nombre $X(\omega) \in E$ (espace fini ou infini).

X est une application probabilisable de l'espace (Ω, P) dans E .

Exemple : Expérience aléatoire du lancer d'un dé.

$$\Omega = \{1, 2, 3, 4, 5, 6\}$$

$$\text{v.a. : } X(\omega) = -1 \text{ si } \omega \leq 3$$

$$X(\omega) = 1 \text{ si } \omega \geq 4$$

$$E = \{-1, 1\}$$

Probabilité que X prenne une des valeurs de E :

$$P[X=x] = P[\{\omega \in \Omega \text{ t.q. } X(\omega)=x\}, x \in E]$$

Deux types de v.a. : Discrète et Continue :

- v.a. discrète X, $E \subset \mathbb{C}$

probabilité d'état : $p(n) = P[X=n]$, $n = -\infty, \dots, +\infty$, avec $\sum_{n=-\infty}^{+\infty} p(n) = 1$

- v.a. continue X, $E \subset \mathbb{R}$

fonction de densité de probabilité :

$$f_X(x) \text{ pour } x \in [-\infty, +\infty], \text{ avec } \int_{-\infty}^{+\infty} f_X(x) dx = 1$$

$$f_X(x) = \lim_{\Delta x \rightarrow 0} \frac{P[x < X \leq x + \Delta x]}{\Delta x}$$

- X est une v.a. à valeur positive :

Cas discret : $p(n) = 0$, $n = -\infty, \dots, -1$

Cas continu : $f_X(x) = 0$ pour $x \in [-\infty, 0[$

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

- Une v.a. est parfaitement caractérisée par sa fonction de répartition
- $F_X(x) \geq 0$ pour tout x
- $F_X(x) \geq F_X(y)$ pour tout $x \geq y$
- $F_X(-\infty) = 0$ et $F_X(+\infty) = 1$
- $P[a < X \leq b] = F_X(b) - F_X(a)$ pour tout $a \leq b$

Cas discret :

$$P[n < X \leq m] = \sum_{k=n+1}^m p(k) = F_X(m) - F_X(n)$$

Cas continu :

$$P[a < X \leq b] = \int_a^b f_X(y) dy = F_X(b) - F_X(a)$$

Les moments, Espérance et Variance

Cas discret :

$$E[X] = \sum_{n=-\infty}^{+\infty} np(n)$$

$$E[X^k] = \sum_{n=-\infty}^{+\infty} n^k p(n)$$

Cas continu :

$$E[X] = \int_{-\infty}^{+\infty} x f_X(x) dx$$

$$E[X^k] = \int_{-\infty}^{+\infty} x^k f_X(x) dx$$

- Une v.a. est parfaitement caractérisée par tous ses moments
- La variance $V[X]$ et l'écart-type $\sigma[X]$:

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

$$\sigma[X] = \sqrt{V[X]}$$

- Le coefficient de variation $cv[X]$: $cv[X] = \frac{\sigma[X]}{E[X]}$

2.3.3 Quelques lois de probabilité

Loi géométrique modifiée :

v.a. discrète de paramètre a ayant $P[X=n] = a^n(1-a)$; $n = 0, 1, \dots$

Interprétation :

X : nombre de tirages consécutifs de « pile »

$P[X=n]$: probabilité de tirer « pile » exactement n fois de suite

a : probabilité de tirer « pile »

$1-a$: probabilité de tirer « face »

$$E[X] = \sum_{n=1}^{\infty} n P[X = n] = \frac{a}{1-a}$$

$$V[X] = \frac{a}{(1-a)^2}$$

Exercice : démontrer ces deux propriétés ci-dessus.

Loi géométrique « de base » : $P[X=n] = a^{n-1}(1-a)$; $n = 1, 2, \dots$

Propriété sans mémoire de la loi géométrique :

$$P[X \leq n+n_0 \mid X \geq n_0] = P[X \leq n]$$

Explication : Ce n'est pas parce que l'on a déjà tiré 50 fois de suite « pile » que l'on a plus de chance, au coup suivant, de tomber sur « face ».

La pièce ne garde pas la mémoire des tirages précédents !

Il s'agit de la seule loi discrète sans mémoire

Loi de Poisson

v.a. discrète X de paramètre λ , ayant $P[X = n] = \frac{\lambda^n}{n!} e^{-\lambda}$

Propriété : $E[X] = V[X] = \lambda$

Exercice : démontrer cette propriété.

Les lois de Poisson s'additionnent : $Z = X + Y$, X de λ_1 et Y de λ_2

$$P[Z = n] = \frac{(\lambda_1 + \lambda_2)^n}{n!} e^{-(\lambda_1 + \lambda_2)}$$

Loi exponentielle

v.a. continue T de paramètre λ , ayant $f_T(t) = \lambda e^{-\lambda t}$; $t \geq 0$

$$(f_T(t) = 0; \quad t < 0)$$

Propriété 1 :

$$F_T(t) = P[T \leq t] = \int_{-\infty}^t f_T(x) dx = 1 - e^{-\lambda t}$$

Propriété 2 :

$$E[T] = \int_0^{\infty} t f_T(t) dt = \frac{1}{\lambda}$$

Exercice : démontrer la propriété 2

Propriété 3 (sans mémoire) :

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

Preuve :

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

Interprétation de la propriété 3 :

Paradoxe de l'inspection (exemple de passage de bus).

Deux passages consécutifs d'un bus suivent une distribution exponentielle de λ .

Si les bus passent en moyenne toutes les $1/\lambda = 15$ minutes, et qu'on attend depuis une heure.

Tout ce qu'on sait dire : le prochain bus n'arrivera en moyenne que dans 15 minutes !

La loi de Poisson est la seule loi continue sans mémoire.

2.3.4 Processus stochastiques

Un processus stochastique $\{X(t)\}_{t \in T}$ est une fonction du temps dont la valeur à chaque instant dépend de l'issue d'une expérience aléatoire.

A chaque $t \in T$, $X(t)$ est donc une v.a.

- T peut être discret ou continu
- L'ensemble de l'espace d'état E dans lequel $X(t)$ prend des valeurs peut être discret (fini ou infini) ou continu.

Processus de comptage :

Un processus stochastique $\{X\}$ à espace d'état discret est un processus de comptage ssi toute réalisation particulière (trajectoire) de X est une fonction croissante, nulle en 0 :

$$\begin{aligned} \text{En discret : } & X_0 = 0 ; \quad X_{n-1} \leq X_n, \quad \forall n \in \mathbf{Z} \\ \text{En continu : } & X(0) = 0 ; \quad X(t) \leq X(s), \quad \forall t < s \in \circ \end{aligned}$$

Un processus stochastique est dit à incréments indépendants ssi :

$$\text{En discret : } X_n - X_{n-1}, X_{n-1} - X_{n-2}, \dots, X_1 - X_0 \text{ sont indépendants}$$

$$\text{En continu : } X(t_n) - X(t_{n-1}), X(t_{n-1}) - X(t_{n-2}), \dots, X(t_1) - X(t_0) \text{ sont indépendants}$$

Un processus stochastique est dit stationnaire ssi :

$$\text{En discret : } X_n - X_{n-1} \text{ et } X_1 - X_0 \text{ sont distribuées selon la même loi } \forall n \in \mathbf{Z}$$

$$\text{En continu : } X(s+t) - X(s) \text{ et } X(t) - X(0) \text{ suivent la même loi } \forall t \text{ et } s \in [0, \infty[$$

Processus de Poisson :

Une chaîne à temps continu et à espace d'état discret $\{N(t)\}$ est un processus de Poisson de paramètre λ ssi :

- $\{N(t)\}$ est un processus de comptage
- $\{N(t)\}$ est un processus stationnaire et à incréments indépendants

$$- \quad P[N(s+t) - N(s) = k] = \frac{(\lambda t)^k}{k!} e^{-\lambda t}$$

Arrivées Poissonniennes \leftrightarrow inter-arrivées exponentielles

Décomposition et superposition

2.3.5 Transformé en Z et transformé de Laplace

Pour caractériser une v.a., au lieu de traiter directement sa fonction de distribution de probabilité ou sa fonction de densité, qui sont des fonctions dans le domaine temporel, parfois, il est souhaitable (ou plus pratique) de s'intéresser à son équivalent dans le domaine de Z (pour v.a. discrète) ou de Laplace (pour v.a. continue). Cette forme s'appelle la « fonction génératrice ».

Pour une v.a. discrète x , sa f.g. est donnée par sa transformé en z :

$$X(z) = E[z^x] = \sum_{i=0}^{\infty} z^i P[x = i]$$

où z est une variable complexe.

Nous avons l'équivalence suivante : f.g. \leftrightarrow p.d.f. (i.e. $P[X=i]$). Au fait,

$$P[x = i] = \frac{1}{i!} \left. \frac{d^i X(z)}{d^i z} \right|_{z=0}$$

L'intérêt de passer par la f.g. est la facilité d'obtention des moments d'une v.a.

$$E[x] = \left. \frac{dX(z)}{dz} \right|_{z=1} = \sum_{i=0}^{\infty} iz^{i-1} P[x = i] \Big|_{z=1}$$

$$E[x^2] = \left. \frac{d^2 X(z)}{d^2 z} \right|_{z=1} + \left. \frac{dX(z)}{dz} \right|_{z=1}$$

Pour une v.a. continue x , sa f.g. est donnée par la transformé de Laplace:

$$P(s) = E[e^{-sx}] = \int_0^{\infty} e^{-st} f_x(t) dt, \text{ quand } x \text{ est positive.}$$

$$\left. \frac{d^j P(s)}{ds^j} \right|_{s=0} = \int_0^{\infty} (-t)^j f_x(t) dt = (-1)^j E[x^j]$$

2.4 Chaînes de Markov

Les chaînes de Markov constituent un outil simple de modélisation et d'évaluation de performances d'une classe particulière de systèmes à événements discrets. Par ailleurs, l'analyse des chaînes de Markov est un préliminaire nécessaire à l'étude des files d'attente et des réseaux de Petri stochastiques.

NB : cette partie est inspirée essentiellement de [BAY 00].

2.4.1 Définition

2.4.1.1 Processus aléatoire

Un processus aléatoire à valeurs dans un ensemble E , est une famille de v.a (variable aléatoire) X_t à valeurs dans E indexées par un paramètre t décrivant l'ensemble T et on note $(X_t; t \in T)$.

E est appelé ensemble des états, E sera en général R, C, Z, N . T est l'ensemble des indices et ce sera en général R, N ou Z .

Si les v.a X_t sont définies sur l'ensemble Ω des événements élémentaires, on appellera *trajectoire* (ou *réalisation*) du processus $(X_t; t \in T)$ correspondant à l'événement élémentaire ω , la suite $(X_t(\omega); t \in T)$.

2.4.1.2 Loi de probabilité d'un processus aléatoire

On appelle loi de probabilité du processus $(X_t; t \in T)$ à valeurs dans R , la famille des lois conjointes des v.a $(X_{t1}, X_{t2}, \dots, X_{tk})$ pour tout entier $k \geq 1$ et tous $t1, t2, \dots, tk$ dans T .

2.4.1.3 Définition d'une chaîne de Markov

Définition 0 : On considère un processus $(X_t; t \in T)$ à valeurs dans un espace d'états dénombrable E (E sera N ou Z de dimension finie ou infinie mais dénombrable). La propriété de Markov traduit une forme particulière de dépendance des v.a X_t qui s'exprime par:

$$P[X_t = i \mid X_{t1} = i1, X_{t2} = i2, \dots, X_{tk} = ik] = P[X_t = i \mid X_{tk} = ik]$$

pour tout entier $k \geq 1$, pour tous $t1 < t2 < \dots < tk < t$, pour tous entiers $i1, i2, \dots, ik, i$.

Un processus Markovien avec un espace d'états discrets (par exemple $i1, i2, \dots, ik, i \in N$) est appelé une **chaîne de Markov** car son diagramme de transition entre les états forme souvent une chaîne (voir plus loin des exemples).

La définition est la même, que le paramètre t soit discret ou continu. Quand t est discret, on parle alors d'une chaîne de Markov à paramètre discret (notée CMD). Si t est continu, on parle d'une chaîne de Markov à paramètre continu (notée CMC).

Cette propriété signifie que l'état présent étant connu, l'avenir et le passé sont indépendants. C'est la propriété dite « sans mémoire ». Cette propriété simplifie l'analyse des systèmes Markovien car on n'a pas besoin de considérer tout le passé du système pour prédire leur évolution. Il suffit de connaître l'état présent de ces systèmes.

2.4.1.4 Modélisation

Une chaîne de Markov peut être utilisée pour décrire l'évolution d'un système à événements discrets. Ce système peut être observé à des instants discrets (équidistants ou quelconques) ou continuellement.

La Figure 7 illustre le cas où le système est observé à des instants $tn = nT$. Des systèmes de transmission comme TDM ou TDMA tombent dans ce cas.

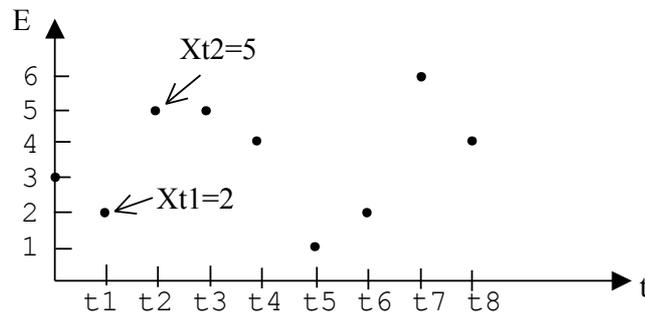


Figure 7 – CMD à instants de changement d'état équidistants

La Figure 8 montre le cas où on s'intéresse à l'état du système juste après un événement. tn est alors l'instant du $n^{ième}$ événement. La file M/G/1 correspond à cette situation.

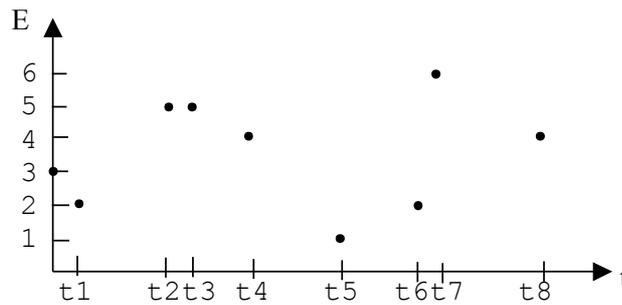


Figure 8 – CMD à instants de changement d'état quelconques

La Figure 9 explique la notion de CMC par rapport à un processus stochastique à espace d'état discret et à temps continu $\{X(t)\}_{t \geq 0}$. Notons que nous ne disposons jamais, dans le cas de CMC, d'un historique complet du processus car on ignore ce qui se passe entre deux observations (échantillons) consécutives, ceci quelque soit la distance entre les deux observations. Mais la propriété sans mémoire du processus Markovien n'exige pas une connaissance du passé pour l'évolution future du processus.

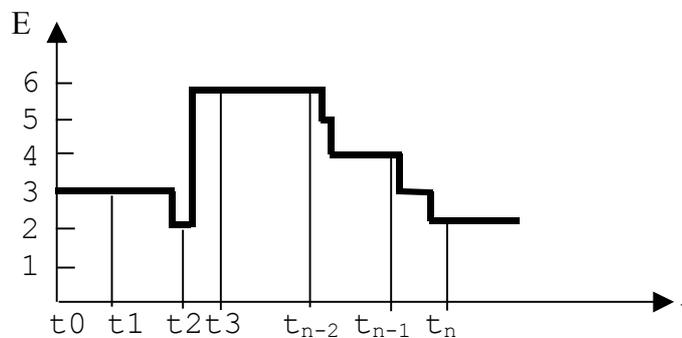


Figure 9 – Processus à espace d'état discret et à temps continu et CMC à des instants d'observation

Le tableau suivant donne des exemples des différents types de processus stochastiques.

	<i>En temps discret : $n \in \mathbf{Z}$</i>	<i>En temps continu : $t \in [0, +\infty]$</i>
Espace discret : $E \subset \mathbf{N}$	Nombre d'appels échangés suivant le jour de l'année : $\{A_n\}_{n=1, 2, \dots, 365}$	Nombre de messages arrivant dans l'intervalle $[0, t]$: $\{N(t)\}_{t \geq 0}$
Espace continu : $E \subset \mathbf{R}$	Temps moyen de traitement par rapport au jour de la semaine : $\{T_n\}_{n=1, 2, \dots, 7}$	Temps d'attente d'un message arrivant à l'instant t : $\{W(t)\}_{t \geq 0}$

2.4.2 Chaînes de Markov à paramètre Discret

2.4.2.1 Définition et représentation

Définition 1 : $(X_n ; n \in \mathbf{N})$ est une CMD ssi

$$P[X_n = j \mid X_{n-1} = i_{n-1}, X_{n-2} = i_{n-2}, \dots, X_0 = i_0] = P[X_n = j \mid X_{n-1} = i_{n-1}]$$

On note:

$$p_{ij}^{(m,n)} = P[X_n = j \mid X_m = i]$$

Une CMD est dite **homogène** si $P[X_n = j \mid X_m = i]$ ne dépend que de $n-m$ pour i et j fixés. On note alors :

$$p_{ij}^{(n)} = P[X_{n+m} = j \mid X_m = i]$$

Dans la suite de ce paragraphe, on s'intéressera uniquement aux CMD homogènes. On définit la **probabilité de transition** d'un état i vers un état j en une seule étape p_{ij} :

$$p_{ij} = p_{ij}^{(1)} = P[X_n = j \mid X_{n-1} = i] \quad \forall n \in \mathbb{N}$$

Il est clair que la somme des probabilité en sortant d'un état : $\sum_{j \in E} p_{ij} = 1$.

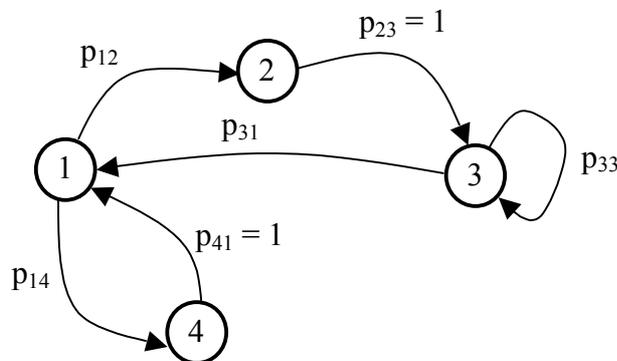
Notons aussi que $p_{ii} \geq 0$: il est possible de rester dans un état entre deux étapes consécutives.

Pour décrire l'évolution d'une CMD homogène, on peut utiliser soit sa **matrice de transition** ($P = [p_{ij}]_{i,j \in E}$) soit un graphe « **état-transition** ».

La matrice de transition est une matrice carrée d'ordre fini ou infini (selon que l'espace d'état E est fini ou infini) :

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdot & \cdot & \cdot & p_{1j} & \cdot & \cdot & \cdot \\ p_{21} & p_{22} & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ p_{i1} & \cdot & \cdot & \cdot & \cdot & p_{ij} & \cdot & \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{bmatrix}$$

Exemple 1 : Voici un exemple d'une CMD avec $E = \{1, 2, 3, 4\}$:



On a en plus $p_{12} + p_{14} = 1$ et $p_{31} + p_{33} = 1$
 Sous forme de la matrice de transition :

$$P = \begin{bmatrix} 0 & p_{12} & 0 & p_{14} \\ 0 & 0 & 1 & 0 \\ p_{31} & 0 & p_{33} & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

L'analyse d'une CMD se fait en deux régimes : transitoire et permanent. Nous donnons un exemple pour illustrer les paramètres de performances que nous pouvons calculer.

Exemple 2 : une souris dans un labyrinthe

Le labyrinthe est constitué de 5 pièces et d'une sortie (voir Figure 10). Les pièces sont reliées entre elles par des couloirs à sens unique. La souris ne peut pas faire le demi-tour dans un couloir. Une fois pénétré dans la pièce 5, la souris ne peut plus sortir.

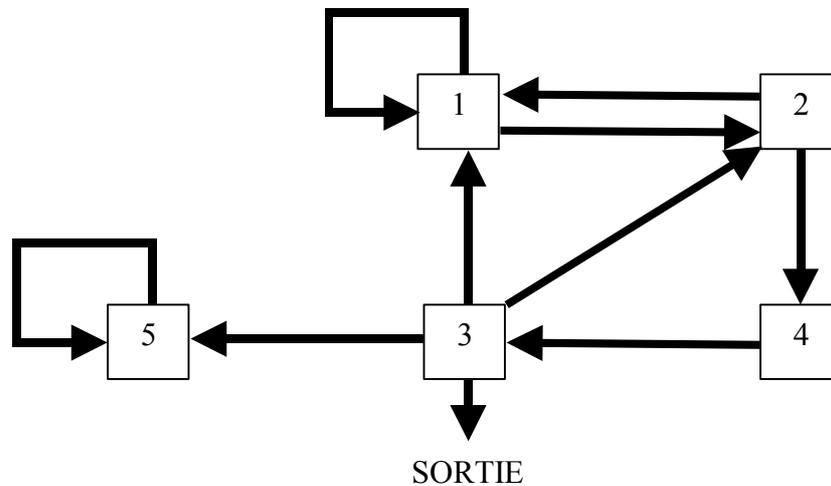
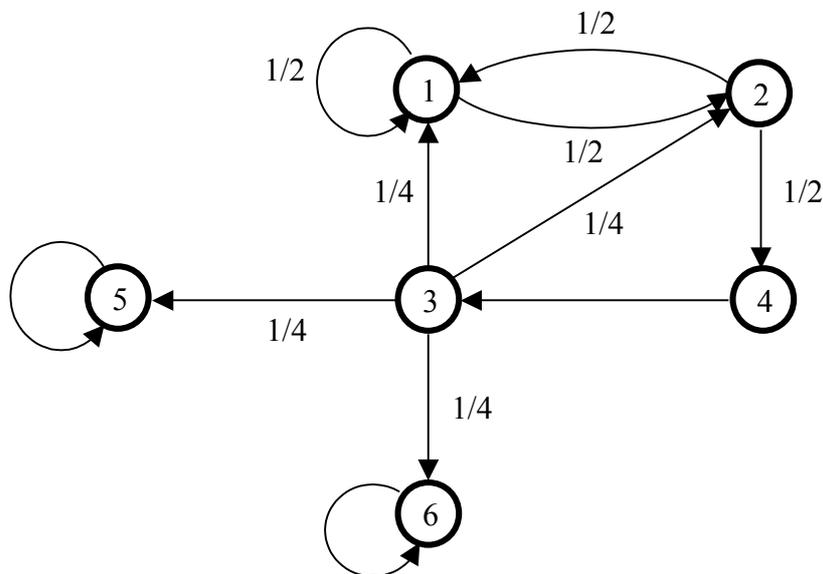


Figure 10 – Labyrinthe

On considère les déplacements de la souris dans le labyrinthe et on s'intéresse uniquement à la succession des pièces visitées et pas au temps passé dans chacune, ni au temps passé par la souris dans les couloirs. On fait les deux hypothèses suivante :

- La souris ne garde pas la mémoire des pièces précédemment visitées ;
- La souris choisit un des couloirs de sortie de façon équiprobable.

Ce système peut être modélisé par une CMD homogène suivante :



Sachant que la souris est placée initialement dans la pièce 2, ce modèle va nous permettre de répondre aux questions suivantes :

1. Quelle est la probabilité qu'elle soit à nouveau dans la pièce 2 après 4 déplacements ? ($\pi_2^{(4)} = 3/16$)
2. Combien de fois passera-t-elle en moyenne par la pièce 3 avant de sortir ou de tomber définitivement dans la pièce 5 ? ($R_{23} = 2$)

3. Quelle est la probabilité que la souris trouve un jour la sortie ? ($f_{26} = 1/2$)
4. Combien en moyenne fera-t-elle de déplacements pour revenir dans la pièce 2 ? ($M_2 = 5/2$)

2.4.2.2 Régime transitoire

L'analyse du régime transitoire consiste à déterminer le vecteur de la probabilité $\pi_j^{(n)} = P[X_n = j]$ pour que le processus $\{X_n\}_{n \in \mathbb{N}}$ se trouve dans l'état j ($j = 1, 2, \dots$) à la nième étape :

$$\pi^{(n)} = [\pi_j^{(n)}]_{j \in E} = [\pi_1^{(n)}, \pi_2^{(n)}, \dots]$$

Ce vecteur des **probabilités d'état** dépend:

- De la matrice de transition P ;
- Du vecteur initial $\pi^{(0)}$.

Il s'agit donc de décrire l'évolution du processus depuis l'état initial jusqu'à l'étape n , en passant par toutes les étapes intermédiaires.

D'après la formule des probabilités totales :

$$\pi_j^{(n)} = P[X_n = j] = \sum_{i \in E} P[X_n = j | X_{n-1} = i] P[X_{n-1} = i] = \sum_{i \in E} \pi_i^{(n-1)} p_{ij}$$

qui exprime que la probabilité de se trouver dans l'état j à la nième étape du processus n'est rien d'autre que la probabilité de passer d'un certain état i à l'état j pondérée par la probabilité d'être dans l'état i à l'étape précédente. Cette expression s'écrit sous forme matricielle :

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

Appliquer n fois cette relation, on obtient la relation qui relie les probabilités d'état à l'étape n à celles de l'état initial :

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

Sur l'**exemple 1**, pour se trouver dans l'état 1 à l'étape n , il fallait soit se trouver dans l'état 4 à l'étape précédente, soit à l'état 3 et faire une transition $3 \rightarrow 1$, c'est à dire : $\pi_1^{(n)} = \pi_4^{(n-1)} + \pi_3^{(n-1)} p_{31}$.

Une autre façon d'obtenir la même relation est de s'intéresser à l'évolution du processus depuis l'état initial jusqu'à l'étape n , sans considérer les étapes intermédiaires.

On définit alors $p_{ij}^{(m)}$, la **probabilité de transition en m étapes** de l'état i à l'état j :

$$p_{ij}^{(m)} = P[X_{n+m} = j | X_n = i] \quad \forall n \in \mathbb{N}$$

ainsi que la matrice de transition en m étapes: $P^{(m)} = [p_{ij}^{(m)}]_{i,j \in E}$

On peut appliquer encore une fois la formule des probabilités totales sur $\pi_j^{(n)}$:

$$\pi_j^{(n)} = P[X_n = j] = \sum_{i \in E} P[X_n = j | X_0 = i] P[X_0 = i] = \sum_{i \in E} \pi_i^{(0)} p_{ij}^{(n)}$$

Et sous forme matricielle : $\pi^{(n)} = \pi^{(0)}P^{(n)}$

Exercice: montrer que $P^{(n)} = P^n$

Pour la question 1 de l'exemple 2, on calcule $\pi_2^{(4)}$ sachant que $\pi_2^{(0)} = 1$. Pour cela il faut calculer P^4 . $\pi^{(0)} = [0, 1, 0, 0, 0, 0]$. Pour éviter de calculer P^4 on peut aussi utiliser la relation $\pi^{(n)} = \pi^{(0)}P^{(n)}$ qui nous demande de calculer $\pi_2^{(4)} = \pi_2^{(0)}p_{22}^{(4)} = p_{22}^{(4)}$. En observant le graphe d'état-transition de l'exemple 2, on peut constater qu'il n'y a que 3 chemins possibles :

- $2 \rightarrow 4 \rightarrow 3 \rightarrow 1 \rightarrow 2$, avec probabilité : $(1/2)(1)(1/4)(1/2) = 1/16$
- $2 \rightarrow 1 \rightarrow 2 \rightarrow 1 \rightarrow 2$, avec probabilité : $(1/2)(1/2)(1/2)(1/2) = 1/16$
- $2 \rightarrow 1 \rightarrow 1 \rightarrow 1 \rightarrow 2$, avec probabilité : $(1/2)(1/2)(1/2)(1/2) = 1/16$

la probabilité d'aller de 2 à 2 en 4 étapes est donc $3/16$.

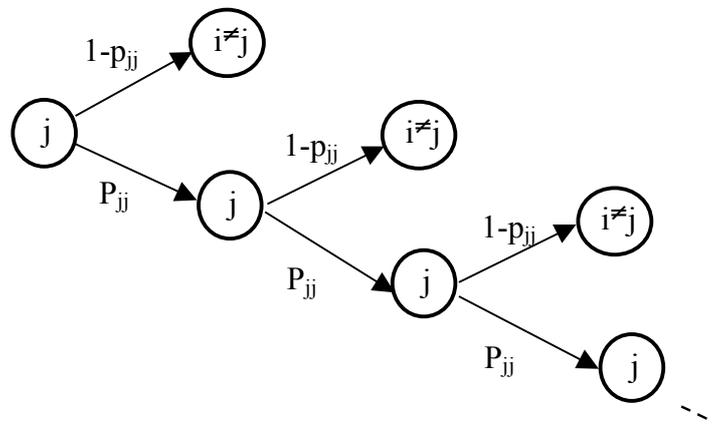
2.4.2.3 Distribution du temps de séjour dans un état.

Propriété 1 : Le temps (ou le nombre d'étapes) passé dans un état d'une CMD a une distribution géométrique.

Preuve : on s'intéresse au nombre d'étapes passées dans un certain état j.

Si $p_{jj} = 0$, on ne reste jamais dans l'état j.

Si $p_{jj} \neq 0$, on a, à chaque étape, une probabilité p_{jj} de rester dans l'état j et une probabilité de $(1 - p_{jj})$ d'en sortir.

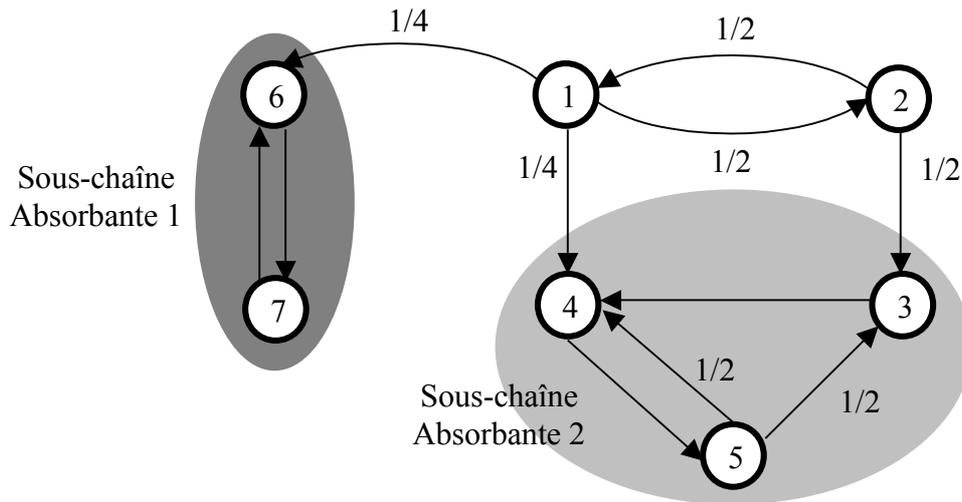


La distribution du nombre m d'étapes passées dans l'état j est donc donnée par : $(1 - p_{jj}) p_{jj}^m$.

2.4.2.4 Classification des états

Définition 2 : Une CMD est dite **irréductible** ssi de tout état i on peut atteindre tout état j (en un nombre fini d'étapes) : $\forall i, j \in E, \exists m > 1$ tel que $p_{ij}^{(m)} \neq 0$

Exemple 3 : La CMD suivante est non irréductible (réductible) car elle comporte deux sous-chaînes absorbantes $\{6, 7\}$ et $\{3, 4, 5\}$.



Définition 3 : Un état j est **périodique** si on ne peut y revenir qu’après un nombre d’étapes multiple de $k > 1 : \exists k > 1$ tel que $p_{ij}^{(m)} = 0$ pour m non multiple de k .

Dans l’exemple 3, état 6 est périodique de période = 2. Etat 4 est apériodique.

Définition 4 : La période d’une CMD est égale au PGCD de la période de chacun de ses états. Une CMD est dite périodique si sa période est supérieure à 1.

La CMD de l’exemple 3 est donc apériodique car son PGCD = 1.

Définition 5 : Soit $f_{jj}^{(n)}$ la probabilité que le **premier retour** en j ait lieu n étapes après l’avoir quitté (f comme « first return »). Soit f_{jj} la probabilité de revenir en j après l’avoir quitté : $f_{jj} = \sum_{n=1}^{\infty} f_{jj}^{(n)} \leq 1$ (car les événements du premier retour dans l’état j en 1, 2, ... étapes sont disjoints).

Soit M_j le « temps » moyen de retour en $j : M_j = \sum_{n=1}^{\infty} n f_{jj}^{(n)}$.

Un état j est dit :

- Transitoire si $f_{jj} < 1$
- Récurrent si $f_{jj} = 1$; de plus il est
 - o Récurrent nul si $M_j = \infty$
 - o Récurrent non nul si $M_j < \infty$

Notons qu’un état apériodique et récurrent non nul est communément appelé « ergodique ».

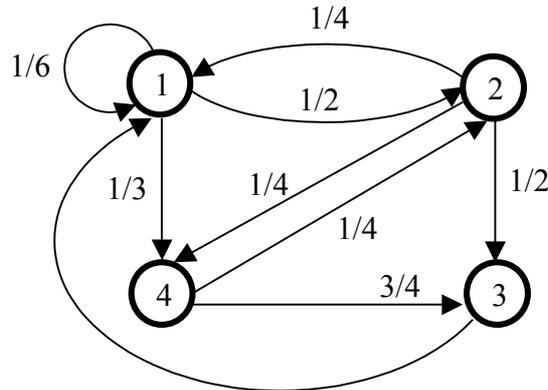
Attention à la différence entre $f_{jj}^{(n)}$ et $p_{jj}^{(n)}$! $f_{jj}^{(n)}$ est la probabilité du premier retour à l’état j qui a eu lieu à l’étape n (sans repasser par j aux étapes intermédiaires), tandis que $p_{jj}^{(n)}$ est la probabilité de revenir en j en n étapes (y compris les chemins qui repassent à l’état j par des étapes intermédiaires). Notons également la différence entre $p_{jj}^{(n)}$ et f_{jj} .

L’exemple 3 permet d’éclaircir ces différences :

- Etat 1 est transitoire car $f_{11} = \sum_{n=1}^{\infty} f_{11}^{(n)} = f_{11}^{(2)} = (1/2)(1/2) = 1/4 < 1$. Notons que $f_{11}^{(4)} = 0$ car on retourne à l’état 1 pour la deuxième fois en 4 étapes. Par contre $p_{11}^{(4)} = 1/16$.
- Etat 6 est récurrent non nul car $f_{66} = f_{66}^{(2)} = 1$ et $M_6 = 2 * f_{66}^{(2)} = 2$

- Etat 3 est récurrent non nul car $f_{33} = f_{33}^{(3)} + f_{33}^{(5)} + f_{33}^{(7)} + \dots = (\frac{1}{2})^1 + (\frac{1}{2})^2 + (\frac{1}{2})^3 + \dots = \sum_{n=0}^{\infty} (\frac{1}{2})^n - (\frac{1}{2})^0 = 1$ et $M_3 = 3 * (\frac{1}{2})^1 + 5 * (\frac{1}{2})^2 + 7 * (\frac{1}{2})^3 + \dots = \sum_{n=1}^{\infty} (2n+1) (\frac{1}{2})^n = 5$.
Notons que $p_{33}^{(3)} = f_{33}^{(3)}$, $p_{33}^{(5)} = f_{33}^{(5)}$, $p_{33}^{(6)} = f_{33}^{(3)} * f_{33}^{(3)} = \frac{1}{4}$, ... Mais $f_{33}^{(6)} = 0$.

Exercice : Dans la CMD suivante, vérifier que $f_{11}^{(1)} = 1/6$, $f_{11}^{(2)} = 1/8$, $f_{11}^{(3)} = 25/48$, ... Calculer $p_{11}^{(1)}$, $p_{11}^{(2)}$, $p_{11}^{(3)}$, ... ainsi que f_{11} .



Propriété 2 : Tous les états d'une CMD irréductible sont de même nature :

- Soit tous transitoires ;
- Soit tous récurrents nuls ;
- Soit tous récurrents non nuls.

Si de plus les états sont périodiques, ils sont tous de même période.

Propriété 3 : Tous les états d'une CMD irréductible finie sont récurrents non nuls.

2.4.2.5 Autres paramètres de performances d'une CMD

Définition 6 : Soit $f_{ij}^{(n)}$ la probabilité d'aller de i à j en exactement n étapes (donc sans passer par l'état j de façon intermédiaire).

Comme pour le calcul de $p_{ij}^{(n)}$, on peut facilement établir la relation suivante :

$$f_{ij}^{(1)} = p_{ij} \text{ et } f_{ij}^{(n)} = \sum_{k \neq j} p_{ik} f_{kj}^{(n-1)} \text{ pour } n > 1.$$

Qui exprime simplement le fait que pour aller de i à j en exactement n étapes, il suffit d'aller en une étape à un état k différent de j, puis d'aller de k à j en exactement n-1 étapes (sans repasser par j).

Attention à la différence entre $f_{ij}^{(n)}$ et $p_{ij}^{(n)}$!

Exercice : montrer que $f_{ij}^{(n)} = \sum_{k \neq j} p_{ik} f_{kj}^{(n-1)}$ pour $n > 1$.

Pour répondre à la question 4 de l'exemple 2, selon la relation précédente, on a :

$$f_{22}^{(1)} = p_{22} = 0$$

$$f_{22}^{(n)} = p_{21} f_{12}^{(n-1)} + p_{24} f_{42}^{(n-1)} = 0,5 f_{12}^{(n-1)} + 0,5 f_{42}^{(n-1)}$$

Il faut réappliquer cette relation à $f_{12}^{(n-1)}$ et à $f_{42}^{(n-1)}$.

$$M_2 = \sum_{n=1}^{\infty} n f_{22}^{(n)}.$$

Exercice : montrer que $M_2 = \sum_{n=1}^{\infty} n f_{22}^{(n)} = 2,5$. (calculer directement en examinant tous les chemins)

Définition 7 : Soit f_{ij} la probabilité d'aller de i en j (pour la première fois) en un nombre quelconque d'étapes (probabilité d'atteindre j en partant de i) : $f_{ij} = \sum_{n=1}^{\infty} f_{ij}^{(n)}$.

$$f_{ij} = \sum_{n=1}^{\infty} f_{ij}^{(n)} = f_{ij}^{(1)} + \sum_{n=2}^{\infty} f_{ij}^{(n)} = p_{ij} + \sum_{n=2}^{\infty} \sum_{k \neq j} p_{ik} f_{kj}^{(n-1)} = p_{ij} + \sum_{k \neq j} p_{ik} \sum_{n=2}^{\infty} f_{kj}^{(n-1)}$$

Pour $m = n - 1$:

$$f_{ij} = p_{ij} + \sum_{k \neq j} p_{ik} \sum_{m=1}^{\infty} f_{kj}^{(m)} = p_{ij} + \sum_{k \neq j} p_{ik} f_{kj}$$

Ce résultat exprime que pour aller de i en j (en un nombre quelconque d'étapes), soit on y va directement (en une étape), soit on va à un état k différent de j et il reste à aller de k à j (en un nombre quelconque d'étapes).

Notons que dans une CMD irréductible les probabilités f_{ij} sont toutes égales à 1.

Pour répondre à la question 3 de l'exemple 2, selon la relation précédente, on a :

$$\begin{aligned} f_{26} &= p_{21}f_{16} + p_{24}f_{46} = 0,5f_{16} + 0,5f_{46} \\ f_{16} &= p_{11}f_{16} + p_{12}f_{26} = 0,5f_{16} + 0,5f_{26} \Rightarrow f_{16} = f_{26} \\ f_{46} &= p_{43}f_{36} = f_{36} \\ f_{36} &= p_{36} + p_{31}f_{16} + p_{32}f_{26} + p_{35}f_{56} = 0,25 + 0,25f_{16} + 0,25f_{26} + 0,25f_{56} \end{aligned}$$

L'état 5 étant absorbant: $f_{56} = 0$.

$$\begin{aligned} \text{Finalement : } f_{26} &= 0,5f_{26} + 0,5(0,25 + 0,5f_{26}) \\ f_{26} &= 0,5 \end{aligned}$$

Définition 8 : Soit R_{ij} le nombre moyen de passage par l'état j sachant qu'on vient de l'état i :

$$R_{ij} = \sum_{n=1}^{\infty} n P[\text{exactement } n \text{ passage par } j \mid \text{état initial} = i] = \sum_{n=1}^{\infty} n P_{ij}(n)$$

$$\begin{aligned} P_{ij}(n) &= P[\text{aller de } i \text{ à } j](P[\text{aller de } j \text{ à } j])^{n-1}P[\text{ne pas revenir en } j] \\ &= f_{ij}(f_{jj})^{n-1}(1 - f_{jj}) \end{aligned}$$

$$R_{ij} = f_{ij}(1 - f_{jj}) \sum_{n=1}^{\infty} n (f_{jj})^{n-1} = f_{ij}(1 - f_{jj}) \left(\sum_{n=1}^{\infty} (f_{jj})^n \right)' = f_{ij}/(1 - f_{jj})$$

Notons que les R_{ij} n'ont pas de sens dans les chaînes irréductibles car $R_{ij} = \infty$.

On peut enfin répondre à la question 2 de l'exemple 2 : $R_{23} = f_{23}/(1 - f_{33})$

$$\begin{aligned} f_{23} &= p_{21}f_{13} + p_{24}f_{43} = 0,5f_{13} + 0,5f_{43} \\ f_{13} &= p_{11}f_{13} + p_{12}f_{23} = 0,5f_{13} + 0,5f_{23} \Rightarrow f_{13} = f_{23} \\ f_{43} &= p_{43} = 1 \Rightarrow f_{23} = 0,5f_{23} + 0,5 \Rightarrow f_{23} = 1 \\ f_{33} &= p_{31}f_{13} + p_{32}f_{23} = 0,5f_{13} + 0,25f_{23} = 0,5 \end{aligned}$$

On a donc $R_{23} = 2$.

2.4.2.6 Régime permanent

L'analyse du régime permanent d'une CMD consiste à s'intéresser à la limite lorsque n tend vers l'infini du vecteur des probabilités $\pi^{(n)}$.

Propriété 4 : Dans une CMD irréductible et apériodique le vecteur π des probabilités limites $\pi_j = \lim_{n \rightarrow \infty} \pi_j^{(n)}$ existe toujours et est indépendant de la distribution des probabilités initiales $\pi^{(0)}$.

Soit tous les états sont transitoires ou récurrents nuls (ce qui ne peut se produire que si la CMD est infinie) et $\pi_j = 0$ pour tout $j \in E$.

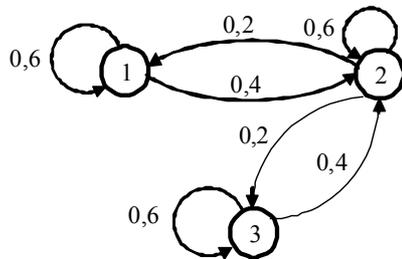
Soit tous les états sont récurrents non nuls (ce qui est le cas si la CMD est finie) et les π_j peuvent être obtenues par le système d'équations suivant :

$$\pi_j = \sum_{i \in E} \pi_i p_{ij} \quad \text{pour tout } j \in E$$

$$\sum_{i \in E} \pi_i = 1$$

La première ligne peut s'écrire sous forme matricielle : $\pi = \pi P$, ce qui correspond bien au passage à la limite sur le système d'équations liant les probabilités transitoires $\pi^{(n)} = \pi^{(n-1)} P$ (dès l'instant où cette limite existe), vue précédemment.

Exemple : Considérons la CMD suivante :



$$P = \begin{pmatrix} 0,6 & 0,4 & 0 \\ 0,2 & 0,6 & 0,2 \\ 0 & 0,4 & 0,6 \end{pmatrix}$$

Cette chaîne est clairement finie, irréductible et apériodique. On en déduit que tous ses états sont récurrents non nuls, que la limite, lorsque n tend vers l'infinie de $\pi^{(n)}$, existe, est indépendante de $\pi^{(0)}$ et est de solution du système d'équations suivant :

$$\pi = \pi P$$

$$\pi_1 + \pi_2 + \pi_3 = 1$$

Avec la solution : $\pi = [0,25 \quad 0,5 \quad 0,25]$.

Il est intéressant de vérifier ce résultat par le calcul itératif en partant de différents vecteurs de probabilités initiales en appliquant $\pi^{(n)} = \pi^{(n-1)} P$. Voici les résultats :

$\pi^{(0)}$	[1 0 0]	[0 1 0]	[0 0 1]
$\pi^{(1)}$	[0.60000 0.40000 0.00000]	[0.20000 0.60000 0.20000]	[0.00000 0.40000 0.60000]
$\pi^{(2)}$	[0.44000 0.48000 0.08000]	[0.24000 0.52000 0.24000]	[0.08000 0.48000 0.44000]
$\pi^{(3)}$	[0.36000 0.49600 0.14400]	[0.24800 0.50400 0.24800]	[0.14400 0.49600 0.36000]
$\pi^{(4)}$	[0.31520 0.49920 0.18560]	[0.24960 0.50080 0.24960]	[0.18560 0.49920 0.31520]
$\pi^{(5)}$	[0.28896 0.49984 0.21120]	[0.24992 0.50016 0.24992]	[0.21120 0.49984 0.28896]
$\pi^{(6)}$	[0.27334 0.49997 0.22669]	[0.24998 0.50003 0.24998]	[0.22669 0.49997 0.27334]
$\pi^{(7)}$	[0.26400 0.49999 0.23601]	[0.25000 0.50001 0.25000]	[0.23601 0.49999 0.26400]
$\pi^{(8)}$	[0.25840 0.50000 0.24160]	[0.25000 0.50000 0.25000]	[0.24160 0.50000 0.25840]
$\pi^{(9)}$	[0.25504 0.50000 0.24496]	[0.25000 0.50000 0.25000]	[0.24496 0.50000 0.25504]
$\pi^{(10)}$	[0.25302 0.50000 0.24698]	[0.25000 0.50000 0.25000]	[0.24698 0.50000 0.25302]
$\pi^{(11)}$	[0.25181 0.50000 0.24819]	[0.25000 0.50000 0.25000]	[0.24819 0.50000 0.25181]
$\pi^{(12)}$	[0.25109 0.50000 0.24891]	[0.25000 0.50000 0.25000]	[0.24891 0.50000 0.25109]
$\pi^{(13)}$	[0.25065 0.50000 0.24935]	[0.25000 0.50000 0.25000]	[0.24935 0.50000 0.25065]

$\pi^{(14)}$	[0.25039 0.50000 0.24961]	[0.25000 0.50000 0.25000]	[0.24961 0.50000 0.25039]
$\pi^{(15)}$	[0.25024 0.50000 0.24976]	[0.25000 0.50000 0.25000]	[0.24976 0.50000 0.25024]
$\pi^{(16)}$	[0.25014 0.50000 0.24986]	[0.25000 0.50000 0.25000]	[0.24986 0.50000 0.25014]
$\pi^{(17)}$	[0.25008 0.50000 0.24992]	[0.25000 0.50000 0.25000]	[0.24992 0.50000 0.25008]
$\pi^{(18)}$	[0.25005 0.50000 0.24995]	[0.25000 0.50000 0.25000]	[0.24995 0.50000 0.25005]
$\pi^{(19)}$	[0.25003 0.50000 0.24997]	[0.25000 0.50000 0.25000]	[0.24997 0.50000 0.25003]
$\pi^{(20)}$	[0.25002 0.50000 0.24998]	[0.25000 0.50000 0.25000]	[0.24998 0.50000 0.25002]

On constate que la convergence vers le vecteur limite est assez rapide dans la pratique, et bien indépendante du choix du vecteur des probabilités initiales.

Dans le cas où les probabilités π_j existent et sont non nulles (donc lorsque tous les états sont récurrents non nuls) on dit que la CMD admet un régime stationnaire. Les probabilités π_j sont alors souvent appelées les « probabilités stationnaires » de la CMD.

Remarque 1 : L'équation $\pi_j = \sum_{i \in E} \pi_i p_{ij}$ s'écrit de façon équivalente $\sum_{i \in E} \pi_j p_{ji} = \sum_{i \in E} \pi_i p_{ij}$ (car $\sum_{i \in E} p_{ji} = 1$). Cette équation s'interprète de la façon suivante : $\pi_j p_{ji}$ est le nombre moyen de transition de l'état j vers l'état i par unité de temps. $\sum_{i \in E} \pi_j p_{ji}$ est donc le flux moyen de sortie de l'état j . De la même façon, $\sum_{i \in E} \pi_i p_{ij}$ est le flux moyen d'entrée dans l'état j . L'équation reliant les probabilités en régime permanent (probabilité stationnaire) s'interprète donc comme :

Pour tout état j , flux sortant de l'état j = flux entrant dans l'état j

Remarque 2 : Dès l'instant où les probabilités stationnaires existent, on montre que $\pi_j = 1/M_j$ pour tout état j . Cela donne une façon simple de calculer M_j , le nombre moyen d'étapes pour revenir en j .

2.4.3 Chaînes de Markov à paramètre Continu

Définition 9 : $\{X(t)\}_{t \geq 0}$ est une CMC ssi

$$\begin{aligned}
 &P[X(t_n) = j \mid X(t_{n-1}) = i_{n-1}, X(t_{n-2}) = i_{n-2}, \dots, X(t_0) = i_0] \\
 &= P[X(t_n) = j \mid X(t_{n-1}) = i_{n-1}] \quad \forall n \text{ et } \forall t_0 < t_1 < \dots < t_n
 \end{aligned}$$

On s'intéresse uniquement aux CMC homogènes. C'est à dire celles dont la probabilité $P[X(t_n) = j \mid X(t_{n-1}) = i_{n-1}]$ ne dépendent pas des instants d'observation t_n et t_{n-1} mais uniquement de la durée $(t_n - t_{n-1})$ qui sépare les deux observations. On note alors :

$$p_{ij}(t) = P[X(s+t) = j \mid X(s) = i] \quad \forall s \geq 0$$

Propriété 5: Le temps passé dans un état d'une CMC homogène a une distribution exponentielle.

A tout état i de E , on associe donc une loi exponentielle T_i de taux μ_i . Autrement dit, la densité de probabilité associée au temps passé dans un état i est : $f_i(t) = \mu_i e^{-\mu_i t}$.

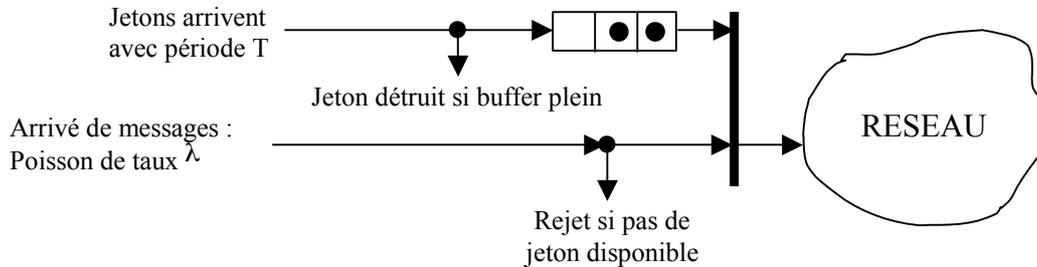
Suite... cf. [Baynat00]

2.5 Processus de Naissance et de Mort et file M/M/1

(cf. [Kleinrock75], [Baynat00] pour plus de détails)

Exercice : Contrôle d'admission par « Leaky bucket »

Pour fournir une certaine QoS (Quality of Service) un réseau implémente souvent des mécanismes de contrôle d'accès pour limiter à la source le flux entrant dans ce réseau (CAC : Call Admission Control d'ATM ou de RSVP en est un exemple). Le mécanisme le plus connu est le « leaky bucket » ou « token bucket » dont le fonctionnement peut être expliqué à l'aide de la figure ci-dessous.



Les messages arrivent selon un processus de Poisson de taux λ . L'accès au réseau est limité par la disponibilité des jetons dans le buffer dont la capacité est limitée à $B = 3$. Un message arrivé à l'entrée du réseau entre dans le réseau en prenant un jeton si au moins un est disponible. Si aucun jeton n'est dans le buffer, le message est rejeté. Un jeton est créé toutes les T secondes (ou un taux $r = 1/T$). Si le buffer est plein, le jeton est détruit. Autrement il est stocké dans le buffer. Avec un tel mécanisme, le flux d'entrée dans le réseau durant $[0, t]$ est limité à $rt + B$ (cf. protocole RSVP)

On s'intéresse au système à des instants particuliers $t_n = nT$ où un jeton vient d'être créé. Un slot de temps est défini comme un intervalle $[t_n, t_{n+1}[$. Au début d'un slot il peut donc y avoir entre 1 et 3 jetons dans le buffer. (Le buffer ne peut pas être vide à cet instant car un jeton vient d'être créé !).

On note $\{X_n\}_{n \in \mathbb{N}}$ le nombre de jetons dans le buffer à l'instant t_n .

Soit a_n la probabilité qu'il arrive n messages pendant un slot de temps T .

Soit b_n la probabilité qu'il arrive au moins n messages (donc n ou plus) pendant un slot T .

1. Donner les probabilités : a_0, a_1, a_2 et b_1, b_2, b_3
2. Si, à l'instant t_n , il y a 1 jeton disponible, combien peut-il y avoir de jetons à l'instant t_{n+1} ? Avec quelles probabilités ?
Si, à l'instant t_n , il y a 2 jetons disponibles, combien peut-il y avoir de jetons à l'instant t_{n+1} ? Avec quelles probabilités ?
Si, à l'instant t_n , il y a 3 jetons disponibles, combien peut-il y avoir de jetons à l'instant t_{n+1} ? Avec quelles probabilités ?
3. En déduire que le processus $\{X_n\}_{n \in \mathbb{N}}$ est une Chaîne de Markov Discrète. Donner le graphe et la matrice de transitions associées.
4. Cette chaîne est-elle irréductible ? Périodique ? Admet-elle un état stationnaire ? Justifier vos réponses.
5. Calculer les probabilités stationnaires π_k pour que le buffer contienne k jetons.
6. Les probabilités stationnaires π_k peuvent aussi être considérées comme les proportions de temps pendant lesquelles le buffer contient k jetons. Quel est le nombre moyen de jetons détruits par seconde ?
7. En déduire le débit moyen d'entrée λ_e des messages dans le réseau ainsi que la probabilité de rejet d'un message.

3 File M/G/1

3.1 Introduction

Jusqu'à présent, les files d'attente étudiées ont une distribution de temps de service exponentielle (M/M/1, ...). Cette distribution peut être considérée appropriée pour modéliser la durée de conversations téléphoniques dans un système de télécommunication classique mais s'avère insuffisant dans un système de communication numérique où la taille d'un paquet de données est souvent constante. La file M/D/1 est alors nécessaire. En général, la taille d'un paquet peut suivre une loi générale et dans ce cas, on parle de la file M/G/1 (G comme loi générale). Nous verrons que M/D/1 et M/M/1 peuvent être considérées comme des cas particuliers de M/G/1.

Considérons le modèle de file d'attente suivant :

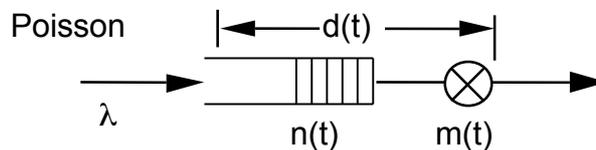


Fig. 1 Modèle de file d'attente M/G/1

Nous cherchons à caractériser le nombre de client dans le système $n(t)$ et le temps de séjour d'un client $d(t)$. Définissons d'abord les paramètres suivants :

- Le flux d'arrivée des clients suit le processus de Poisson avec un taux de λ clients par unité de temps. Le nombre de clients arrivés durant $[0, t]$ est donc donné par :

$$P_k(t) = e^{-\lambda t} \frac{(\lambda t)^k}{k!}$$

- Le temps de service est une variable aléatoire continue. Sa distribution est notée par :

$$M(t) = P[\text{Temps de service} \leq t]$$

- La fonction de densité (p.d.f) du temps de service est définie par :

$$m(t) = dM(t)/dt$$

- Le temps moyen de service est alors :

$$\bar{m} = \int_0^{\infty} t m(t) dt$$

3.2 Chaînes de Markov induite

La chaîne de Markov induite (Embedded Markov Chain) est largement utilisée dans l'analyse des systèmes probabilistes. Il consiste à choisir à la fois une suite des instants dans le temps pendant l'évolution d'un processus et un état de sorte que la chaîne de Markov se forme. A des instants $t_1 \leq t_2 \leq \dots \leq t_i, \dots$ on observe les variables d'état correspondantes $S_1, S_2, \dots, S_i, \dots$. Grâce à la chaîne induite, on a :

$$P[S_N = l_N \mid S_1 = l_1, S_2 = l_2, \dots, S_{N-1} = l_{N-1}] = P[S_N = l_N \mid S_{N-1} = l_{N-1}]$$

Pour notre modèle de la figure 1, nous choisissons le nombre de clients dans le système comme la variable d'état décrivant l'évolution du processus. Les instants où l'on observe le système correspondent aux instants de départ d'un client (cf. Fig.2).

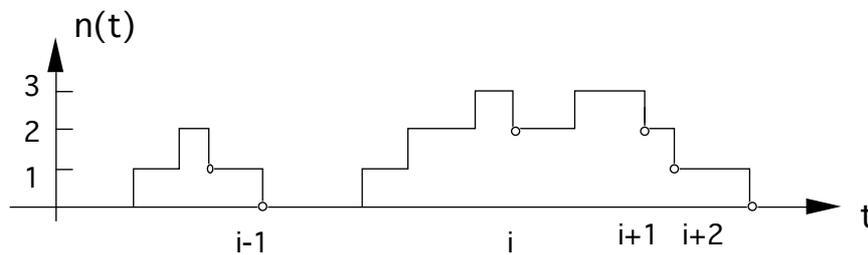


Fig. 2 Evolution du nombre de clients dans le système

Notons que :

- n_i : nombre de clients dans le système (ceux dans la file + celui dans le serveur) aux points d'observation (instants où un client quitte le système),
- a_i : nombre de clients arrivés durant le temps de service d'un client.

L'équation suivante décrit l'évolution du système :

$$n_{i+1} = n_i - U(n_i) + a_{i+1} \tag{1}$$

$$\text{avec } U(x) = \begin{cases} 0, & x \leq 0 \\ 1, & x > 0 \end{cases}$$

On voit que le nombre de clients à l'instant $i+1$ ne dépend que de celui de l'instant i et les nouvelles arrivées. Ce nombre de clients forme donc une chaîne de Markov à paramètre discret mais avec des instants de changement d'état quelconques. On peut montrer que n_i a une distribution stationnaire ssi $\rho = \lambda \bar{m} < 1$.

On définit :

$$\pi(k) = \lim_{i \rightarrow \infty} P[n_i = k]$$

Grâce à la stationnarité, on a :

$$\lim_{i \rightarrow \infty} E[n_{i+1}] = \lim_{i \rightarrow \infty} E[n_i] = \bar{n} \tag{2}$$

Prenons l'espérance de chaque côté de l'équation 1 et d'après l'équation 2, on a :

$$E[U(n_i)] = E[a_{i+1}] \quad (3)$$

$$\begin{aligned} E[a_{i+1}] &= \int_0^{\infty} E[a_{i+1} | \text{MessageLength} = t] m(t) dt \\ &= \int_0^{\infty} \lambda t m(t) dt = \lambda \int_0^{\infty} t m(t) dt = \lambda \bar{m} = \rho \end{aligned}$$

$$E[U(n_i)] = P[n_i > 0] = 1 - P_0$$

D'après (3), on a :

$$P_0 = 1 - \rho \quad (4)$$

3.2.1 Fonction génératrice de n_i

Pour obtenir les paramètres caractéristiques des variables aléatoires discrètes $\{n_i\}$, nous allons suivre l'approche par fonction génératrice (cf. fin de ce chapitre pour un rappel). La fonction génératrice de $\{n_i\}$ est la transformé en Z de sa fonction de densité qui est définie par :

$$P_i(z) = E[z^{n_i}] = \sum_{k=0}^{\infty} z^k P[n_i = k]$$

De l'équation 1, on a:

$$P_{i+1}(z) = E[z^{n_i - U(n_i) + a_{i+1}}]$$

Comme a_i et n_i sont indépendants, on a alors :

$$P_{i+1}(z) = E[z^{n_i - U(n_i)}] E[z^{a_{i+1}}] = E[z^{n_i - U(n_i)}] A(z) \quad (5)$$

où $A(z) = E[z^{a_i}]$ est indépendant de l'indice i grâce à la stationnarité.

Pour le terme qui reste, par la définition de la fonction génératrice (cf. §3.3), on a :

$$E[z^{n_i - U(n_i)}] = \sum_{k=0}^{\infty} z^{k - U(k)} P[n_i = k]$$

Comme on peut écrire :

$$E[z^{n_i - U(n_i)}] = P_0 + \sum_{k=1}^{\infty} z^{k-1} P[n_i = k] = P_0 + z^{-1} \left(\sum_{k=0}^{\infty} z^k P[n_i = k] - P_0 \right)$$

On a alors :

$$E[z^{n_i - U(n_i)}] = P_0 + z^{-1} (P_i(z) - P_0) \quad (6)$$

De (5) et (6) on a :

$$P_{i+1}(z) = [P_0 + z^{-1} (P_i(z) - P_0)] A(z)$$

Supposons que la solution stationnaire existe :

$$\lim_{i \rightarrow \infty} P_{i+1}(z) = \lim_{i \rightarrow \infty} P_i(z) = P(z)$$

on a la f.g. de n_i sous forme générale (En fait, il suffit que le flux d'arrivée soit un processus stationnaire et à incréments indépendants¹, mais sans nécessairement qu'il soit Poissonien) :

$$P(z) = \frac{P_0(1-z)A(z)}{A(z) - z}$$

Dans le cas du flux d'arrivée Poissonien, on a $P_0 = 1 - \rho$. Ce qui nous donne finalement :

$$P(z) = \frac{(1-\rho)(1-z)A(z)}{A(z) - z} \quad (7)$$

Théoriquement l'équation 7 nous donne toutes les informations sur la v.a. discrète n_j . Par exemple la transformé inverse de Z, i.e, par dérivés successives on peut obtenir sa distribution stationnaire $\pi(k)$ (cf. équation 13). La suite de ce chapitre donne le calcul des valeurs moyennes du nombre de clients et du temps de séjour d'un client.

3.2.2 Nombre moyen de clients dans la file M/G/1

$$\bar{n} = P'(z)|_{z=1}$$

La dérivé de l'équation 7 donne :

$$P'(z)[A(z) - z] + P(z)[A'(z) - 1] = (1 - \rho)(-1)A(z) + (1 - \rho)(1 - z)A'(z)$$

En dérivant encore une fois (règle de L'Hopital), on obtient :

$$P''(z)[A(z) - z] + 2P'(z)[A'(z) - 1] + P(z)A''(z) = 2(1 - \rho)(-1)A'(z) + (1 - \rho)(1 - z)A''(z)$$

Avec $z = 1$ et comme $A(1) = P(1) = 1$, on a :

$$\bar{n} = P'(1) = \frac{(1-\rho)A'(1)}{1 - A'(1)} + \frac{A''(1)}{2(1 - A'(1))}$$

$$A(z) = E[z^a] = \int_0^{\infty} E[z^a | \text{durée_transmission} = t] m(t) dt$$

$$A(z) = \int_0^{\infty} \sum_{n=0}^{\infty} z^n \frac{(\lambda t)^n e^{-\lambda t}}{n!} m(t) dt = \int_0^{\infty} e^{-\lambda t(1-z)} m(t) dt = M(\lambda(1-z)) \quad (8)$$

où $M(s)$ est la transformé de Laplace de $m(t)$.

¹ Un processus stochastique $X(t)$ est dit *stationnaire* ssi les v.a. $X(s+t) - X(s)$ et $X(t) - X(0)$ sont distribuées selon la même loi.

Un processus stochastique $X(t)$ est dit à *incréments indépendants* ssi les v.a. $X(t_n) - X(t_{n-1})$, $X(t_{n-1}) - X(t_{n-2})$, ..., $X(t_1) - X(t_0)$ sont indépendants.

$$A'(z)|_{z=1} = -\lambda M'(0) = \lambda \bar{m} = \rho$$

$$A''(z)|_{z=1} = \lambda^2 M''(0) = \lambda^2 \overline{m^2}$$

Finalement :

$$\bar{n} = \rho + \frac{\lambda^2 \overline{m^2}}{2(1-\rho)} \quad (9)$$

3.2.3 Temps de séjour de client d(t)

Grâce aux arrivées Poissonniennes, et en imaginant une nouvelle CMD aux instants distribués selon d(t), on a :

$$P(z) = E[z^n] = \int_0^{\infty} E[z^n | \text{délai} = t] d(t) dt$$

$$P(z) = \int_0^{\infty} \sum_{n=0}^{\infty} \frac{(\lambda t)^n e^{-\lambda t}}{n!} z^n d(t) dt = \int_0^{\infty} e^{-\lambda t(1-z)} d(t) dt = D(\lambda(1-z))$$

La transformé de Laplace de d(t) est donc :

$$D(s) = s(1-\rho)M(s)/[s-\lambda+\lambda M(s)] \quad (10)$$

La transformé inverse de Laplace nous donne la distribution de d(t).

Le temps moyen de séjour d'un client peut être obtenu soit en dérivant l'équation 10 soit en appliquant la formule de Little :

$$\bar{d} = \frac{\bar{n}}{\lambda} = \bar{m} + \frac{\lambda \overline{m^2}}{2(1-\rho)} \quad (11)$$

C'est la formule de Pollaczek-Khinchin. On remarque que cette équation est liée à celle de 9 par la formule de Little (peut être considérée comme une sorte de démonstration de la formule de Little).

3.2.4 Cas particuliers : M/M/1 et M/D/1

Pour un temps de service exponentiel de paramètre μ , $m(t) = \mu e^{-\mu t}$ et

$$M(s) = E[e^{-sM}] = \int_0^{\infty} e^{-st} [\mu e^{-\mu t}] dt = \frac{\mu}{\mu + s} \quad (12)$$

En remplaçant s par $\lambda(1-z)$ on a :

$$A(z) = \frac{\mu}{\mu - \lambda(z-1)}$$

et :

$$A'(z)|_{z=1} = \frac{\lambda \mu}{[\mu - \lambda(z-1)]^2}|_{z=1} = \lambda \bar{m} = \lambda \frac{1}{\mu} = \rho$$

$$A''(z)|_{z=1} = \frac{2\lambda\mu}{[\mu - \lambda(z-1)]^3}|_{z=1} = \lambda^2 \overline{m^2} = 2\rho^2$$

Selon l'équation 10 et 12, la transformé de Laplace de délai est :

$$D(s) = (\mu - \lambda)/(s + \mu - \lambda)$$

$$\bar{d} = \frac{1}{\mu - \lambda}$$

$$P(z) = (1-\rho)/(1-\rho z)$$

$$\bar{n} = \frac{\rho}{1-\rho}$$

Pour un temps de service constant (ou déterministe), $E[m(t)] = \bar{m}$, et

$$m(t) = \begin{cases} \delta(t), t = \bar{m} \\ 0, \text{ailleurs} \end{cases}$$

$$M(s) = \int_0^{\infty} m(t)e^{-st} dt = e^{-s\bar{m}}$$

$$P(z) = \frac{(1-\rho)(z-1)e^{\rho(z-1)}}{z - e^{\rho(z-1)}}$$

$$D(s) = \frac{s(1-\rho)e^{-s\bar{m}}}{s - \lambda + \lambda e^{-s\bar{m}}}$$

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

$$\bar{d} = \frac{\bar{m}(2-\rho)}{2(1-\rho)}$$

3.3 Rappel sur la fonction génératrice (f.g.)

Pour une variable aléatoire discrète x, sa f.g. est définie par :

$$X(z) = E[z^x] = \sum_{k=0}^{\infty} z^k P[x = k]$$

où z est une variable complexe.

Plus généralement, pour une v.a.d. $x \in \{0, 1, \dots\}$, on a l'espérance de f(x) :

$$E[f(x)] = \sum_{k=0}^{\infty} f(k)P[x = k]$$

Démonstration :

On démontre l'équation précédente en utilisant la probabilité conditionnelle (*très utile pour l'obtention des f.g. car permet de décomposer une somme en plusieurs termes plus faciles à traiter*)

$$E[f(x)] = E[f(x) | x=0]P[x=0] + E[f(x) | x=1]P[x=1] + \dots$$

$$\begin{aligned}
 &= E[f(0)]P[x=0] + E[f(1)]P[x=1] + \dots \\
 &= f(0)P[x=0] + f(1)P[x=1] + \dots \\
 &= \sum_{k=0}^{\infty} f(k)P[x = k]
 \end{aligned}$$

A partir de cette f.g., nous pouvons obtenir facilement les renseignements suivants :

- La densité de probabilité (p.d.f) :

$$P[x = i] = \frac{1}{i!} \left. \frac{d^i X(z)}{d^i z} \right|_{z=0} \tag{13}$$

- Le premier moment :

$$E[x] = \left. \frac{dX(z)}{dz} \right|_{z=1} = \sum_{i=0}^{\infty} iz^{i-1}P[x = i] \Big|_{z=1} = \sum_{i=0}^{\infty} iP[x = i]$$

- Le second moment :

$$E[x^2] = \left. \frac{d^2X(z)}{d^2z} \right|_{z=1} + \left. \frac{dX(z)}{dz} \right|_{z=1}$$

4 Evaluation des protocoles/réseaux

4.1 Evaluation de performances de TDMA

4.1.1 Modèle

TDMA (Time Division Multiple Access) est une des techniques de transmission utilisée dans de nombreux systèmes de transmission numérique (Multiplexeur, commutateur, ...). Le fonctionnement du TDMA peut être représenté par le schéma suivant :

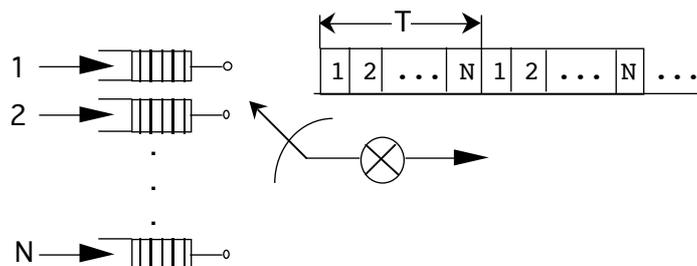


Fig. 1 Fonctionnement du TDMA

Chaque file est servi de façon cyclique par le serveur (qui représente le médium de transmission commun, souvent un câble de haut débit) avec une période de T. Un client ne peut quitter donc la file tant que l'interrupteur (voir Fig. 1) n'est pas fermé sur la file. Si la file est vide au moment de la fermeture de l'interrupteur, une tranche de temps (appelé un slot en anglais) sera quand même perdue.

Evaluer les performances d'un système TDMA revient à étudier une seule file de la Fig. 1. L'étude de TDMA utilise les résultats du M/G/1. Mais il convient de remarquer que TDMA est légèrement différent de M/D/1. En effet ans TDMA le serveur peut être oisif même s'il y a des clients en attente dans des files d'attente car le serveur est alloué en ce moment à une file

vide. On parle alors des files d'attente avec serveur intermittent (intermittently available server).

Pour que nos résultats soient les plus généraux possibles, on suppose dans un premier temps que le flux d'arrivée des clients est un processus à Accroissements Indépendants et Stationnaires (A.I.S). On verra ensuite le cas d'un flux d'arrivée Poissonnien lorsqu'il s'agit des calculs concrets.

On s'intéresse aux grandeurs suivantes :

- $X(t)$ = nombre de clients dans la file d'attente à l'instant t .
- $W_q(k)$ = temps d'attente du $k^{\text{ème}}$ client (ou Délai).

ainsi qu'à l'existence des lois limites de ces deux grandeurs lorsque $k \rightarrow +\infty$ et $t \rightarrow +\infty$.

Notons que nous ne comptons pas le client dans le serveur.

4.1.2 Etude analytique

4.1.2.1 Etude du nombre de clients dans la file

Comme il a été dit que le principe d'étude est comparable à celui utilisé dans M/G/1. Quand on considère le nombre de client dans une file comme la variable d'état et on l'observe à des instants $t_n = nT$, on obtient une CMD :

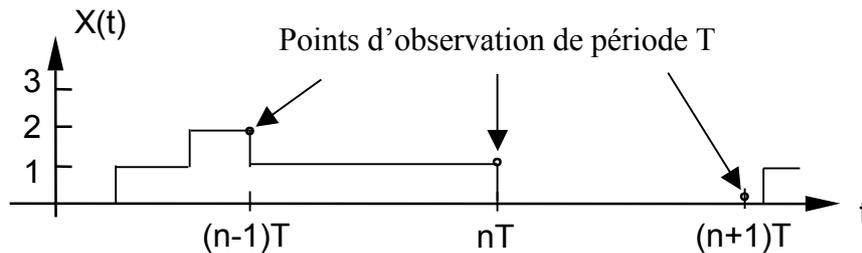


Fig. 2 : CMD d'une file TDMA

On note:

- $X_n = X(nT^-)$: nombre de clients dans la file juste avant la fermeture de l'interrupteur.
- $M_n = X(nT^+)$: nombre de clients dans la file juste après la fermeture de l'interrupteur.
- A_{n+1} : nombre de clients qui sont arrivés durant $[nT, (n+1)T]$. où T est la durée constante séparant deux fermetures consécutives de l'interrupteur.
- $(v(t) | t \geq 0)$ ($v(0)=0$): nombre de clients qui sont arrivés durant $]0, t]$, on a alors $v((n+1)T) - v(nT) = A_{n+1}$.

Le modèle dans Fig. 2 peut être décrit par l'équation de récurrence ci-dessous:

$$X_{n+1} = (X_n - 1)^+ + A_{n+1} \quad , \quad x^+ = \max(0, x) \quad (1)$$

Cette équation apparaît dans l'étude de M/G/1. Signalons que les variables aléatoires X_n et A_n n'ont pas la même signification que dans M/G/1. En effet, dans M/G/1:

- X_n est le nombre de clients dans le système à l'instant où le $n^{\text{ème}}$ client quitte le système.
- A_n est le nombre de clients qui sont arrivés durant le service du $n^{\text{ème}}$ client.

On peut démontrer que la distribution limite existe. La condition de stationnarité est $\rho = \lambda T < 1$, où λ est le taux d'arrivée de clients. En fait, si les variables aléatoires $A_1, A_2, \dots, A_n, \dots$ sont indépendantes et identiquement distribuées de même loi que A alors $(X_n)_{n \geq 0}$ est une chaîne de Markov homogène. En particulier: si $(v(t) | t \geq 0)$ est un processus à Accroissements Indépendants et Stationnaires, alors les variables aléatoires A_n sont indépendantes et identiquement distribuées.

Il est clair que:

$$M_n = \max(X_n - 1, 0) = (X_n - 1)^+ \text{ et } M_n \text{ vérifie :}$$

$$M_{n+1} = \max(M_n + A_{n+1} - 1, 0) \tag{2}$$

La connaissance de X_n fournit M_n ; par contre, si $M_n = 0$, on ne peut reconstituer X_n qui peut être 0 ou 1. Il est donc préférable d'étudier X_n et d'en déduire M_n .

On note que $X(t) = M_n + v(t) - v(nT); \forall t \in]nT, (n+1)T]$ (3)

L'étude de la chaîne (X_n) est faite dans la partie M/G/1. Nous prenons directement les résultats. La distribution limite existe pour cette chaîne de Markov quand $\rho = \lambda T < 1$. La fonction génératrice $P(z)$ s'écrit :

$$P(z) = E[z^X] = \{(1-\rho)(1-z)A(z)\} / \{A(z)-z\} \tag{4}$$

où

$A(z) = E(z^A)$ est la fonction génératrice de la variable aléatoire A ou encore de $v(T)$.

On sait que la distribution limite de X_n est définie par:

$$\pi(k) = \lim_{n \rightarrow \infty} P[X_n = k]$$

et on définit la distribution stationnaire de M_n :

$$\tilde{\pi}(k) = \lim_{n \rightarrow \infty} P[M_n = k] = \begin{cases} \pi(k+1) \text{ si } k > 0 \\ \pi(0) + \pi(1) \text{ si } k = 0 \end{cases}$$

M_∞ a donc pour distribution limite $(\tilde{\pi}(k))$.

En dérivant l'équation (4), nous obtenons le nombre moyen de clients dans la file à la fin de chaque slot :

$$N_q = E[X_\infty] = (1/2)[E(A) + \text{Var}(A)/(1-E(A))] \tag{5}$$

C'est la formule de Pollaczek-Khinchin.

On peut obtenir également le nombre moyen de clients juste après la fermeture de l'interrupteur:

$$E[M_\infty] = E[X_\infty] - (1 - \pi(0)) = N_q - \rho \tag{6}$$

Une observation intéressante est que l'équation (1) est exactement la même que celle de M/G/1. Est-ce que cela veut dire que M/G/1 (sans considérer le client qui est dans le service) et notre modèle ont les mêmes performances? La réponse est non. C'est évident qu'à cause de l'interrupteur, les clients dans notre modèle sont obligés d'attendre plus longtemps pour quitter le système.

Une étude en détail de M/G/1 montre que l'observation du nombre de clients aux instants où un client quitte le système sera toujours la même que les observations à n'importe quel instant aléatoire [GRO 85]. C'est une propriété très particulière de M/G/1.

Par contre, on peut démontrer qu'un système autonome comme notre modèle n'a pas cette propriété intéressante. Si l'on observe le modèle de la figure 2 à des instants quelconques, l'état stationnaire n'existe plus.

En effet, $X(t)$ pour un t quelconque n'admet pas de distribution limite quand $t \rightarrow +\infty$ (i.e. $X(t)$ ne converge pas en loi).

Le nombre de clients dans la file à n'importe quel instant t est donné par l'équation 3. En posant $t_n = nT + h$ et en prenant l'espérance de l'équation (3), on obtient le nombre moyen de clients dans la file observé à l'instant h .

$$E[X(t_n)] = E[M_n] + E[v(h)] \tag{7}$$

$E[M_n]$ est le nombre moyen de clients quand on observe le système juste après la fermeture de l'interrupteur, il est donné par l'équation 6. En particulier, quand $v(h)$ est un processus Poissonien, on a :

- $E[v(h)] = \lambda * h$.
- $E[X_\infty] = \rho + \rho^2 / [2 * (1 - \rho)]$
- $E[M_\infty] = E[X_\infty] - \rho = \rho^2 / [2(1 - \rho)]$

$E[X(t_n)]$ a la forme présentée sur la figure 3.

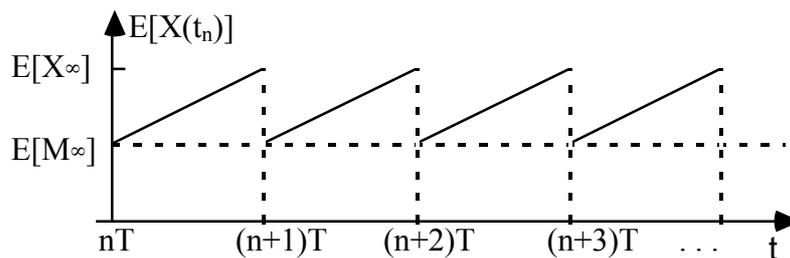


Fig. 3 Espérance du nombre de clients en fonction du temps

L'espérance est fonction du temps h . Si les points d'observation qu'on a choisis sont les débuts de chaque slot, $h = 0$ donc l'espérance obtenue est bien $E[M_\infty]$. Pour obtenir la moyenne sur h , il suffit de "moyenner" l'équation 3 sur l'intervalle $[0, T]$ et ce qui nous donne:

$$\bar{N}_q = \lim_{\tau \rightarrow \infty} \frac{1}{T} \int_{\tau}^{\tau+T} E[X(t)] dt = \frac{\rho}{2} + \frac{\rho^2}{2(1 - \rho)} \tag{8}$$

4.1.2.2 Etude du délai d'attente

Le délai d'attente d'un client dans la file est composé des éléments suivants:

- Un nouveau client arrivé trouve le système vide. S'il arrive h secondes après le début de l'intervalle considéré, un délai de (T - h) secondes aura lieu avant qu'il quitte le système. T secondes est la longueur d'un intervalle séparant deux fermetures consécutives de l'interrupteur.
- Soit L le nombre de clients qui sont arrivés pendant l'intervalle (0, h). Donc, le temps pour qu'ils quittent tous le système est L*T.
- Soit X_n le nombre de clients dans la file à la fin de l'intervalle précédent. Compte tenu qu'un client quitte la file au début de l'intervalle présent, le temps pour qu'ils quittent tous la file est alors $T*(X_n - 1)^+$.

Ainsi, en supposant la distribution limite atteinte (ce qui permet de supprimer l'indice n):

$$W_q = (X_{\infty} - 1)^+ * T + L * T + (T - h) \tag{9}$$

La transformée de Laplace du délai peut être obtenue à partir de cette équation comme suivant.

Le premier terme est indépendant des autres, selon l'équation 6 du chapitre 3 (M/G/1) on a la fonction génératrice de $(X_{\infty} - 1)^+$:

$$E[z^{(X-1)^+}] = E[z^{X-U(X)}] = P_0 + z^{-1}(P(z) - P_0) = 1 - \rho + z^{-1}[P(z) - (1-\rho)] \tag{10}$$

où P(z) est la f.g de X_{∞} .

Remplacer z par e^{-sT} la transformée de Laplace de $(X_{\infty} - 1)^+ * T$ devient:

$$\tilde{P}(s) = E[e^{-sT(X-1)^+}] = (1 - \rho) + e^{sT}(P[e^{-sT}] - (1 - \rho)) \tag{11}$$

Considérons maintenant la variable aléatoire: $f = LT + (T-h)$

On note que les variables aléatoires L et h ne sont pas indépendantes. Un calcul de la probabilité conditionnelle nous donne $P[t < f \leq t+dt]$.

Exercice : calculer cette probabilité $P[t < f \leq t+dt]$

La transformée de Laplace de f est alors:

$$F(s) = \frac{e^{-\lambda T}[1 - e^{-sT}] - e^{-sT}}{T\{s - \lambda[1 - e^{-sT}]\}} \tag{12}$$

La transformée de Laplace de W_q est donc:

$$W_q(s) = \tilde{P}(s)F(s) \tag{13}$$

En dérivant l'équation 13 au point $s = 0$, on obtient le délai moyen.

On peut aussi calculer le délai moyen directement. Puisque l'espérance de la somme des variables aléatoires est la somme des espérances de chaque variable même si les variables ne sont pas indépendantes. On peut prendre l'espérance de l'équation 9.

$$E[W_q] = E[(X_\infty - 1)^+] * T + E[L] * T + T - E[h] \tag{14}$$

Considérons le client en question qui arrive h secondes après le début de l'intervalle. La probabilité que L clients arrivent dans l'intervalle $(0, h)$ en suivant le processus Poissonien est donnée par :

$$P[L \text{ clients en } h \text{ secondes}] = (\lambda h)^L e^{-\lambda h} / L!$$

Par les caractéristiques du processus de Poisson on sait aussi qu'étant donné qu'il y a au moins un client qui arrive dans un intervalle, l'instant de cette arrivée suit une distribution uniforme dans l'intervalle. Ce qui est vrai pour tous les clients.

On sait que h est uniformément distribué dans $[0, T]$ et on a :

- $E[h] = T/2$;
- $E[L] = \lambda * T/2$;
- $E[(X_n - 1)^+] = E[X_\infty] - \rho = E[M_\infty]$.

On obtient le délai moyen d'un client dans la file:

$$E(W_q) = \frac{T}{2} + \frac{\lambda T^2}{2(1-\rho)} \tag{15}$$

On peut remarquer que les équations 8 et 15 sont reliées par la formule de Little. Notons également que le temps de réponse = $E(W_q) + 1$ si l'on suppose que la durée d'un slot = 1.

4.1.2.3 Résumé sur les résultats obtenus

1) En ce qui concerne le nombre de clients dans la file:

Le processus décrit par l'équation 1 est une chaîne de Markov admettant une distribution limite. La distribution de probabilité limite (et stationnaire) du nombre de clients dans la file d'attente aux instants nT^- (juste avant la fermeture de l'interrupteur) est donnée par sa fonction génératrice:

$$P(z) = E[z^X] = \frac{(1-\rho)(1-z)A(z)}{A(z)-z}$$

où:

$A(z) = E(z^{A_n})$ est la fonction génératrice de la variable aléatoire A_n et $\rho = \lambda T (< 1)$.

La fonction génératrice $P(z)$ permet de reconstituer la distribution limite de probabilité: $(\pi(k))_{k \geq 0}$.

On en déduit le nombre moyen de clients à l'instant nT^- :

$$E[X_{\infty}] = (1/2)[E(A) + V(A)/(1-E(A))]$$

Et le nombre moyen à l'instant nT^+ :

$$E[M_{\infty}] = E[X_{\infty}] - (1-\pi(0)) = E[X_{\infty}] - \rho$$

Le nombre moyen de clients dans la file \bar{N}_q est donné par:

$$\bar{N}_q = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_{\tau}^{\tau+T} E[X(t)] dt = E[X_{\infty}] - \frac{\rho}{2}$$

En particulier, lorsque le processus du flux d'arrivée est Poissonnien de paramètre λ :

$$\bar{N}_q = \frac{\rho}{2} + \frac{\rho^2}{2(1-\rho)}$$

2) En ce qui concerne le délai d'attente d'un client:

La distribution de probabilité limite du délai d'attente W_q est donnée par sa transformée de Laplace:

$$W_q(s) = \{[1-\rho] + e^{sT}[P(e^{-sT}) - (1-\rho)]\} \frac{e^{-\lambda T}[1-e^{-sT}] - e^{-sT}}{T\{s - \lambda[1-e^{-sT}]\}}$$

où:

$P(e^{-sT})$ est donné par l'équation (4) en remplaçant z par e^{-sT} .

On en déduit le délai d'attente moyen d'un client:

$$E(W_q) = \frac{T}{2} + \frac{\lambda T^2}{2(1-\rho)}$$

Exercice : Etude de la chaîne $\{x_i\}$ induite à des instants où un client quitte le système.

Soit :

- x_i le nombre de clients à l'instant où le $i^{\text{ème}}$ client quitte la file
- a_i le nombre de clients arrivés pendant un intervalle T
- \hat{a}_i le nombre de clients arrivés pendant le temps aléatoire Y
- Y le temps d'attente quand un client arrive et trouve une file vide (un client qui est arrivé dans un système vide est obligé d'attendre le début de l'intervalle suivant)

On peut avoir la relation de récurrence suivante pour décrire l'évolution de $\{x_i\}$:

$$x_{i+1} = x_i - U(x_i) + a_{i+1}U(x_i) + \hat{a}_{i+1}[1 - U(x_i)]$$

Calculer $P(z) = E[z^x]$ et le nombre moyen de clients dans la file.

4.2 M/G/1 et évaluation de performances des commutateurs

Dans les réseaux d'interconnexion (Internet ou réseaux de télécommunication), un des problèmes essentiels est la conception et le dimensionnement de commutateurs (switch, routeur, autocommutateur, ...). Typiquement un commutateur assure la fonction d'interconnecter N lignes en entrée aux N lignes à la sortie (Fig. 4)

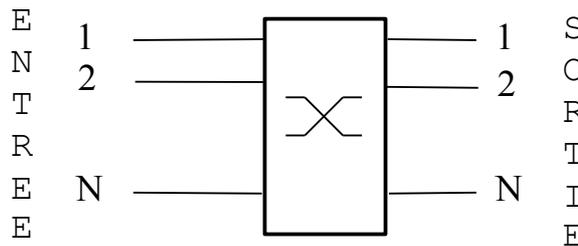


Fig. 4 Equipement d'interconnexion

Une façon simple d'interconnexion est l'utilisation de « cross-bar » (Fig. 5)

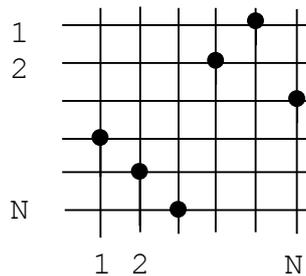


Fig. 5 Commutateur « Cross-bar » NxN

Dans le contexte de commutation par paquet, nous cherchons à éviter le rejet de paquets à cause des lignes de sortie occupées. Par exemple, si le débit de transmission est le même pour toutes les lignes (en entrée et en sortie), deux paquets en provenance des lignes 1 et 2 à la destination d'une même sortie ne pourront pas être transmis en même temps. Nous utilisons alors une mémoire tampon (buffer) pour stocker momentanément les paquets.

Deux implémentations de buffer peuvent être envisagées dans un commutateur du type « cross-bar » : buffer à la sortie et buffer en entrée (Fig. 6)

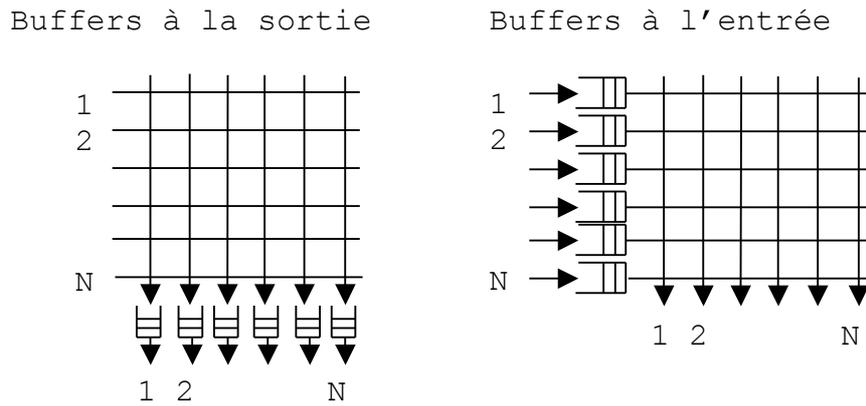
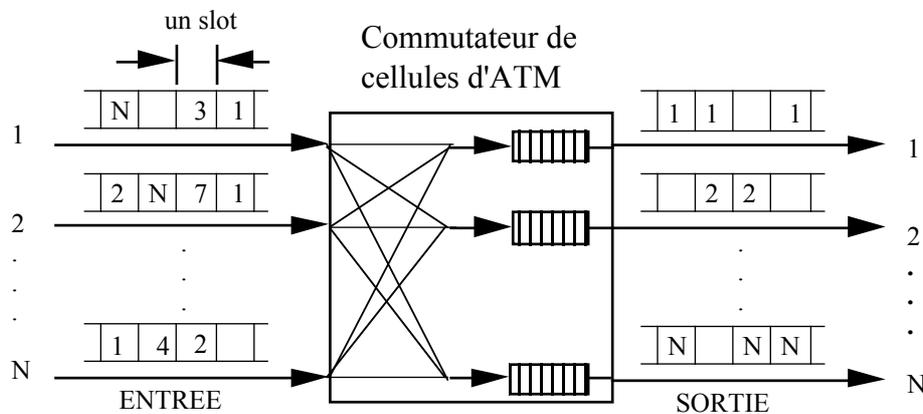


Fig. 6 Commutateur avec buffers

Par intuition on peut imaginer que le buffer en entrée conduit à une file d'attente plus longue (ou un temps moyen d'attente plus long) que le buffer à la sortie. Ceci car un paquet en attente dans un buffer en entrée pour une certaine sortie peut être bloqué par un autre paquet même si la sortie du premier est libre. Cette intuition peut être vérifiée en quantifiant les deux systèmes. Dans la suite nous ne traitons que le cas des buffers à la sortie.

Exercice :

On considère un commutateur ATM composé de N lignes en entrée et N en sortie de débit 155 Mb/s . Le commutateur fonctionne avec un débit N fois plus élevé que le débit d'une ligne ($155 \times N \text{ Mb/s}$). A la sortie d'une ligne on risque d'avoir jusqu'à N trames d'ATM (appelé cellule qui est de taille constante 53 octets) durant un "slot" si les N trames en entrée ont la même destination. Un buffer (ou une file d'attente) est donc placé devant chaque ligne de sortie afin de les stocker avant de les transmettre.



On s'intéresse au nombre de cellules $Q(t)$ et temps d'attente d'une cellule W dans une des files d'attente.

Hypothèse :

- Processus d'arrivée est Bernoulli avec p comme probabilité qu'une cellule arrive à une entrée durant un slot.
- Chaque cellule a une probabilité de $1/N$ pour une destination donnée.

Modèle :

Quand on s'intéresse au nombre de cellules, en régime stationnaire, dans la file à la fin de chaque slot Q_m , juste après le départ d'une cellule, on peut écrire :

$$Q_{m+1} = \max(0, Q_m + A_{m+1} - 1)$$

où A_m est le nombre de cellules arrivées dans la file d'attente durant le $m^{\text{ième}}$ slot, qui est le même que durant n'importe quel autre slot noté A. Sa distribution est Binomiale :

$$a_i = \text{Prob}[A=i] = \binom{N}{i} \left(\frac{p}{N}\right)^i \left(1 - \frac{p}{N}\right)^{N-i}$$

et sa f.g. :

$$A(z) = \sum_{i=0}^N z^i a_i = \left(1 - \frac{p}{N} + z \frac{p}{N}\right)^N$$

(On remarque que $Q_m = X_m - u(X_m)$, où X_m est le nombre de cellules dans la file d'attente à la fin du $m^{\text{ième}}$ slot juste avant le départ d'une cellule, avec $X_{m+1} = X_m + A_{m+1} - u(X_m)$.)

Travail à faire :

1 - Montrer que :

$$Q(z) = \frac{(1-p)(1-z)}{A(z) - z}$$

2 - Calculer le nombre moyen de cellules dans la file d'attente \bar{Q} ainsi que le temps moyen d'attente d'une cellule \bar{W} .

3 - Que devient une file de sortie quand $N \rightarrow \infty$?

5 Evaluation des protocoles d'accès aléatoire

Cette famille de protocoles est représentée par Aloha et CSMA/CD. Une description détaillée de ces protocoles peut être trouvée dans des ouvrages de réseaux. Ces protocoles, bien qu'ils soient anciens (années 1970-1980), sont largement utilisés dans les réseaux les plus répandus (Aloha dans les réseaux radiomobiles GSM, CSMA/CD dans les réseaux Ethernet, CSMA/CA dans le réseau sans fil IEEE802.11). Nous nous limitons à étudier dans ce chapitre le rendement de protocoles Aloha pur, Aloha segmenté et CSMA non-segmenté et non-persistant afin d'illustrer l'évolution en termes de performances de cette famille de protocoles.

5.1 Rendement de Aloha pur

Les hypothèses sont:

- Les messages sont de taille constante l dont la durée de transmission est m (avec $l = m \cdot \text{Débit}$).
- Le flux d'arrivée de messages suit une loi Poissonnienne avec un taux λ messages/seconde. $P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}$
- La file d'attente de chaque station est de capacité 1 et le nombre de stations est infini.

Le système peut être modélisé par la Fig. 1

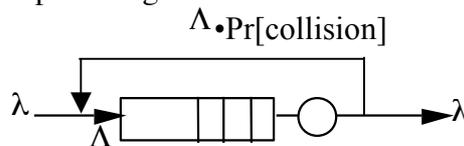


Fig. 1 Modèle de Aloha pur

où Λ est le taux de messages à transmettre par le médium de transmission (transmission réussite + retransmission à cause de collisions).

On suppose que la retransmission est non-persistante, i.e., une fois entré dans une collision, les stations impliquées retransmettent leur messages après un temps d'ajournement aléatoire qui suit une loi exponentielle. Comme la population de messages est infinie, chaque message peut donc être considéré comme s'il appartienne à une nouvelle station. C'est à dire que chaque message nouvel arrivée peut trouver une station vide. On peut considérer le système comme s'il y ait un nombre infini de stations. On peut montrer que Λ est aussi Poissonnien (un flux Poissonnien s'additionne et s'éclate « se split »).

Pour un message arrivé à l'instant t , la probabilité d'entrer en collision avec d'autres messages est donnée par (cf. Fig. 2):

$$\begin{aligned}
 P[\text{collision}] &= 1 - P[\text{pas de message} \mid \text{un message arrivé}] \\
 &= 1 - P_{2m}[i=0 \mid j=1] = 1 - \frac{P(i=0 \cap j=1)}{P(j=1)} \\
 &= 1 - \frac{P(i=0)P(j=1)}{P(j=1)} = 1 - e^{-2\Lambda m} \tag{1}
 \end{aligned}$$

(attention: il s'agit ici d'une probabilité conditionnelle!)

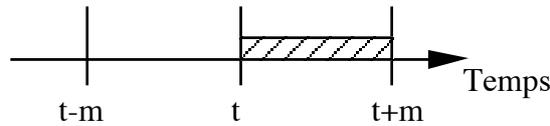


Fig. 2 Occupation du médium de Aloha pur

On a:

$$\Lambda = \lambda + \Lambda(1 - e^{-2\Lambda m}) \tag{2}$$

On définit:

- le rendement $\rho = \lambda m$
- la charge du médium $R = \Lambda m$

L'équation 2 devient alors:

$$\rho = R e^{-2R} \tag{3}$$

Le rendement maximal 18,4% est atteint lors que la charge du médium est de 50% (cf. Fig.4).

5.2 Rendement de Aloha segmenté

Le protocole Aloha segmenté propose une amélioration du rendement en n'autorisant la transmission qu'aux instants discrets séparés par des intervalles de temps fixe qui est égale à m (cf. Fig. 3).

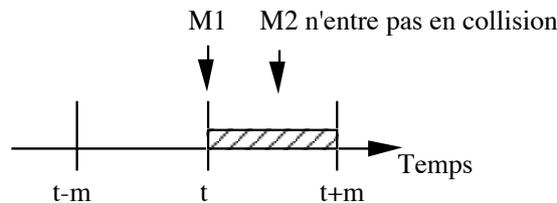


Fig. 3 Occupation du médium de Aloha segmenté

Pour un message qui arrive entre l'intervalle $t-m$ et t , il n'entre en collision que si un ou plusieurs autres messages arrivent durant cet intervalle. Pour un message qui commence à être transmis à l'instant t , la probabilité d'entrer en collision avec d'autres messages est donc:

$$P[\text{collision}] = 1 - e^{-\Lambda m} \tag{4}$$

et le rendement est donc:

$$\rho = R e^{-R} \tag{5}$$

Figure 4 montre le rendement des méthodes Aloha segmenté ainsi que Aloha pur en fonction de la charge. Pour la méthode Aloha segmenté, le rendement maximal 36,8% est atteint lors que la charge du médium est de 100%.

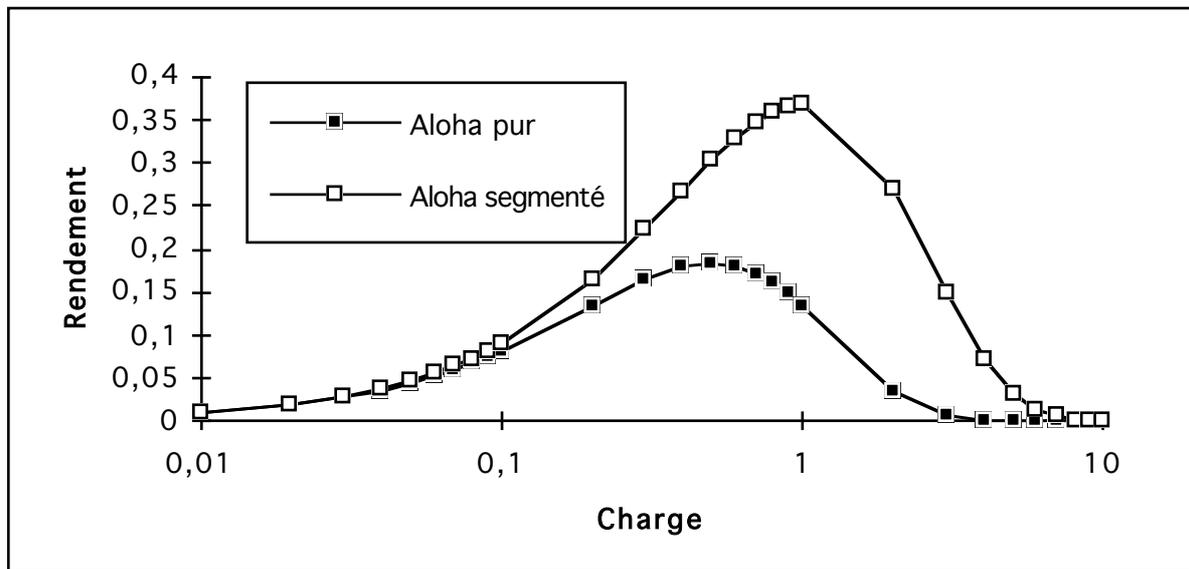


Fig. 4 Rendement des méthodes Aloha en fonction de la charge

5.3 Rendement de CSMA non-segmenté et non-persistant

CSMA non-segmenté propose une amélioration de rendement en ajoutant à chaque station la fonction d'écoute du médium. Pour diminuer les collision, une station ne peut émettre un message qu'en détectant un silence du médium. Les collisions ne sont pas exclut à cause des délais de propagation. Si une station entre en collision, elle retransmet le même message après avoir attendu un délai aléatoire $l/\alpha \gg m$ et qui a une distribution exponentielle (non-persistant).

Le rendement de ce protocole peut être défini par:

$$\rho = T/(B + I) \quad (6)$$

avec:

- T: durée moyenne de la période d'occupation du médium pour la transmission réussite d'un message,
- B: durée moyenne de la période d'occupation du médium pour les transmissions réussites ou non-réussites d'un message,
- I: durée moyenne d'une période oisive.

Fig. 5 montre les périodes occupées et oisives du médium.

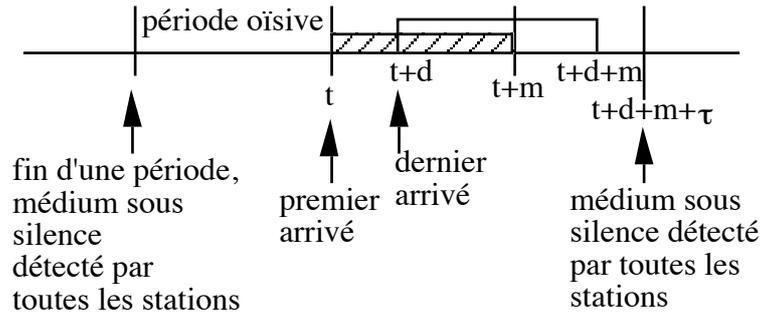


Fig. 5 Occupation du médium de CSMA non-segmenté

Grâce au processus de Poisson, on a :

$$I = 1/\Lambda$$

En plus, on sait que la durée d'une période occupée est au moins $m + \tau$. On suppose que $\tau \leq m$ et que le dernier message arrive à l'instant $t + d$ avec $d < \tau$. Selon Fig. 5 la période occupée va durer jusqu'à l'instant $t + d + m + \tau$, le moment où toutes les stations puissent détecter un silence du médium.

La distribution de cette variable aléatoire d est donc :

$$P[d \leq x] = P[\text{pas de messages arrivées durant } \tau - x] = \exp[-\Lambda(\tau - x)] \quad (7)$$

Avec

$$p(x) = P'[x] = \Lambda \exp[-\Lambda(\tau - x)]$$

La valeur moyenne de d est :

$$E[d] = \int_0^{\tau} x p(x) dx = \tau - (1 - e^{-\Lambda\tau})/\Lambda \quad (8)$$

La somme de la période oisive et la période occupée est donnée par :

$$I + B = m + 1/\Lambda + 2\tau - (1 - e^{-\Lambda\tau})/\Lambda \quad (9)$$

La durée moyenne pour une transmission réussite est :

$$T = mP[\text{pas de messages arrivés durant } \tau] = m \cdot e^{-\Lambda\tau} \quad (10)$$

En définissant: $R = m\Lambda$ and $a = \tau/m$, nous avons finalement le rendement du protocole CSMA non-persistant et non-segmenté :

$$\rho = R e^{-Ra} / [R(1 + 2a) + e^{-Ra}] \quad (11)$$

qui est fortement influencé par le paramètre a (délai de propagation par rapport à la taille de messages). La figure 6 montre le rendement de cette méthode d'accès en fonction de la charge avec les différentes valeurs de a .

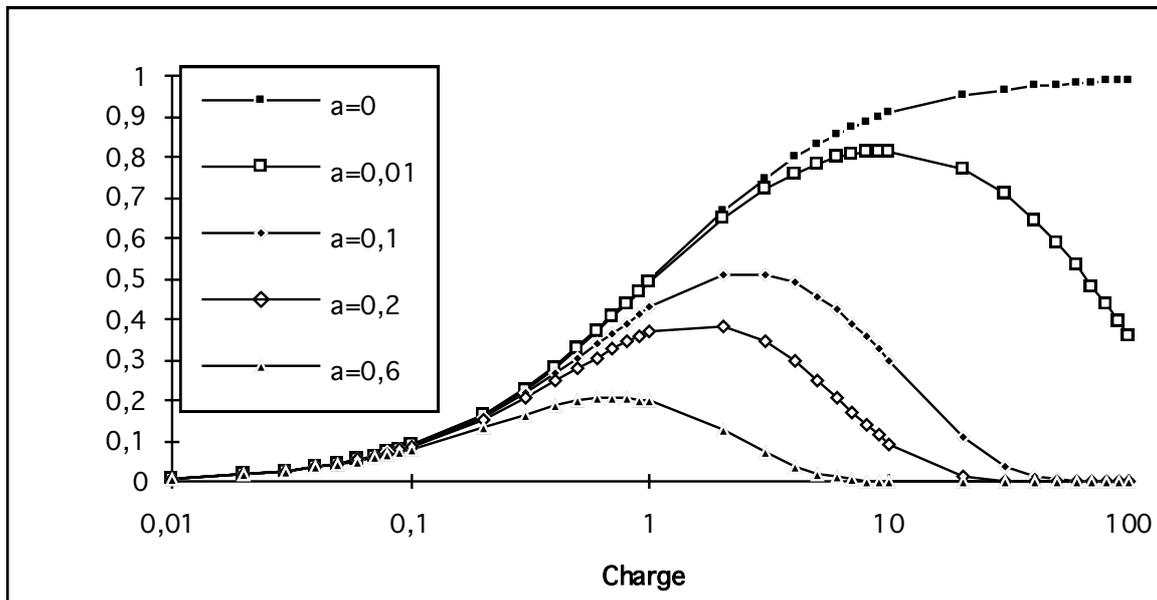


Fig. 6 Rendement de CSMA non-segmenté et non-persistant

5.4 Rendement de CSMA segmenté et non-persistant

Avec les techniques utilisées dans l'étude de CSMA non-segmenté et Aloha segmenté, le lecteur doit être capable de trouver le rendement du protocole CSMA segmenté.

Exercice : Montrer que le rendement du protocole CSMA segmenté avec retransmission non-persistante est donné par:

$$\rho = T/(B + I) = m\tau\Lambda e^{-\tau\Lambda}/[m(1 - e^{-\tau\Lambda}) + \tau]$$

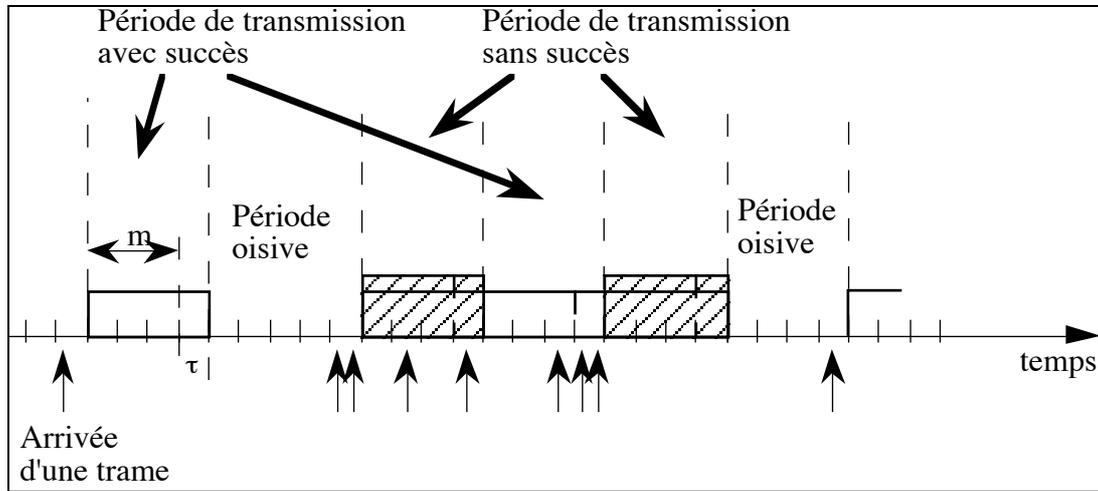
où:

- T : durée moyenne d'occupation du médium pour la transmission d'information utile, on note T' sa durée.
- B : durée moyenne de la période d'occupation du médium (pour les transmissions réussites ou non-réussites d'un message), on note B' sa durée.
- I : durée moyenne d'une période d'oisiveté, on note I' sa durée.

1 - Description du protocole

Le fonctionnement du protocole CSMA segmenté non-persistant est le même que celui du CSMA non-segmenté sauf que dans le premier l'axe du temps est segmenté en "slots" de durée τ (délai de propagation maximum entre deux stations quelconques). On suppose que la taille des trames est une constante m qui est un multiple de τ . Une trame arrivée dans une station, avant d'être émise, doit attendre le début du "slot" prochain et que la station ait détecté un silence du médium à l'instant du début de ce "slot". La durée de transmission d'une trame dure donc m/τ slots (un nombre entier). Si le médium est détecté occupé, l'émission sera

ajournée d'un délai aléatoire (non-persistant). Comme l'étude du CSMA non-segmenté, on suppose que le nombre de stations est infini et que l'ensemble de flux d'entrée (nouvelles trames + ré-émission des trames en collisions) forme un processus Poissonnien de taux Λ . La figure ci-dessous montre ce qu'on peut observer sur le médium.



On voit bien qu'une période oisive dure au minimum un "slot" et qu'une période de transmission avec succès ou sans succès dure $m + \tau$ secondes. Une collision a lieu si deux ou plus de trames arrivent à l'intérieur d'un même "slot" et sont émises au début du "slot" prochain.

2 - Période d'oisiveté $I' = k\tau, k \geq 1$

Pour qu'une période oisive dure un slot, il doit y avoir au moins une trame qui arrive durant le premier slot de cette période. Pour qu'une période oisive dure deux slots, il ne doit arriver aucune trame durant le premier slot et au moins une durant le second slot... La distribution de probabilité d'une période I' suit alors une loi géométrique:

$$P(I' = k\tau) = p^{k-1}(1-p)$$

avec:

$$p = P[\text{aucune trame durant 1 slot}].$$

Calculer $I = E[I']$.

(formule utile: $\sum_{n=0}^{\infty} p^n = \frac{1}{1-p}$).

3 - Période d'occupation $B' = k(m+\tau), k \geq 1$

Une période d'occupation contiendra k périodes de transmission s'il y a au moins une arrivée durant le dernier slot de chacune des premières $k - 1$ périodes de transmission et aucune arrivée durant le dernier slot de la k ième période de transmission. $P(B' = k(m + \tau))$ suit aussi une loi géométrique.

Montrer que: $B = E[B'] = (m + \tau)e^{\Lambda\tau}$.

4 - Durée de la transmission d'information utile T'

Une période de transmission avec succès consacre m secondes à la transmission d'information utile. Il n'y a pas de transmission d'information utile pour des périodes de transmission sans succès. Comme le nombre de périodes de transmission durant B' est $B'/(m + \tau)$, on a :

$$T = E[T'] = m(B/(m + \tau))P_{suc}(\text{période avec succès}).$$

où :

$$\begin{aligned} P_{suc} &= \text{Prob}(\text{une seule arrivée durant le dernier slot avant la période de transmission sachant} \\ &\text{qu'il y a au moins une arrivée durant la période de transmission précédente}) \\ &= \Lambda\tau e^{-\Lambda\tau}/(1 - e^{-\Lambda\tau}). \end{aligned}$$

6 Annexe : Modélisation et simulation

6.1 Introduction à la Modélisation et Évaluation de Performances

L'étude d'un système réel n'est que rarement réalisable dans un environnement opérationnel. La représentation du fonctionnement d'un système d'une manière plus ou moins précise est nécessaire pour nous permettre d'approcher le comportement du système. Cette représentation se nomme *modélisation*.

La démarche à suivre pour l'évaluation de performances peut être schématisée par Fig. 1.

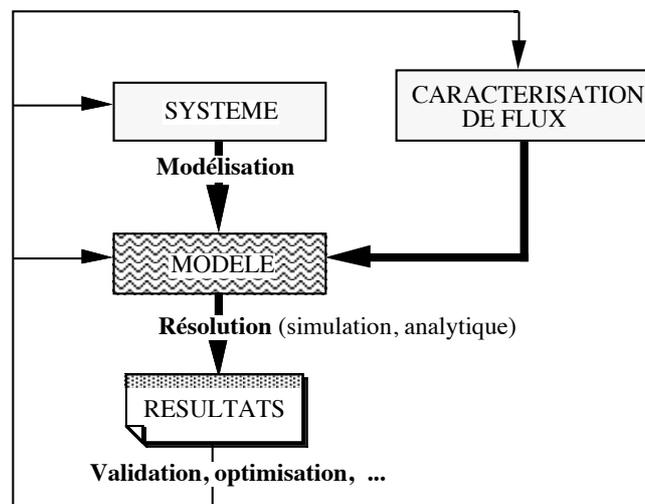


Fig. 1 Démarche de l'évaluation de performances.

Étant donné un système à étudier, nous devons d'abord créer un modèle de celui-ci. Il convient de souligner qu'un tel modèle est toujours un compromis entre la fidélité et la simplicité. En effet, il n'existe pas de modèle unique pour un système réel donné. Selon la connaissance du système de chacun, on peut obtenir soit un modèle très fidèle mais souvent peu exploitable, soit un modèle peu fidèle et très exploitable. Notre tâche ici est de trouver un modèle qui soit relativement fidèle au système réel et exploitable.

Bien que la méthode formelle de la modélisation n'existe pas, les quelques règles ci-dessous peuvent nous aider dans la démarche:

- 1) Définir clairement le but de l'étude et délimiter le système à étudier.
- 2) Lister tous les services offerts par le système et les issues possibles par service.
- 3) Sélectionner les critères de performances.
- 4) Lister les paramètres du système et de la charge.
- 5) Sélectionner les techniques d'évaluation.
- 6) Sélectionner les scénarios de charge.
- 7) Définir les plans d'expérience.
- 8) Analyser et interpréter les résultats.
- 9) Présenter les résultats.

Une fois que le modèle est obtenu, l'étape suivante consiste à mener une étude (souvent statistique s'il s'agit d'un système à événements discrets) de caractérisation des applications supportées par le système. Cette caractérisation se résulte souvent une description aléatoire ou déterministe de flux (de matière, d'information, ...).

Muni d'un modèle du système et en connaissant les applications supportées par ce système, on peut commencer à étudier le comportement du modèle. Selon le *formalisme* qu'on utilise pour la description du modèle, on appliquera des *techniques de résolution* différentes. Afin d'obtenir des résultats pertinents, il est indispensable de fixer a priori des *critères de performances* qui nous intéressent.

Les résultats obtenus permettent non seulement de comprendre le comportement du système réel, mais aussi de valider la conception (s'il s'agit d'un nouveau système), voire de proposer d'autres alternatives (paramétrage, dimensionnement, configuration, ...) afin d'optimiser le système. A ce stade, les résultats peuvent aussi être utilisés pour améliorer le modèle et l'étude de caractérisation.

Dans la suite de ce chapitre nous décrivons d'abord deux formalismes usuels lorsque l'on modélise un système à événements discrets: *Réseaux de Petri* et *Files d'attente*, ensuite nous développons les techniques de résolution des modèles en files d'attente: *théorie des files d'attente* et *simulation*. Nous notons que l'accent est mis sur l'évaluation de performances par simulation plutôt que par l'étude analytique basée sur des résultats issues de la théorie des files d'attente.

6.2 Formalismes de Modélisation

Aujourd'hui, il existe beaucoup de formalismes (ou techniques) de modélisation. Le choix de formalismes est déterminé d'une part par la nature du système à modéliser et d'autre part par les résultats que nous voulons obtenir. Deux formalismes souvent utilisés dans le domaine de systèmes et réseaux informatiques sont:

- Les réseaux de Petri qui permettent une analyse plutôt qualitative. Cette technique est souvent utilisée pour la validation et la vérification d'un système.
- Les réseaux de files d'attente qui permettent une analyse quantitative. Cette technique est souvent utilisée pour l'évaluation des performances d'un système.

6.2.1 Réseaux de Petri

Un *réseau de Petri* (RdP) [BRA 83], [DAV 92] est un graphe orienté défini par un quadruplet $\{P, T, A, M_0\}$ où:

- $P = \{P_1, \dots, P_m\}$ est un ensemble fini de places représentées par des cercles.
- $T = \{T_1, \dots, T_n\}$ est un ensemble fini de transitions représentées par des rectangles.
- $A = \{A_1, \dots, A_p\}$ est un ensemble fini d'arcs reliant places et transitions et réciproquement.
- M_0 est le marquage initial qui représente l'état initial du réseau.

A chaque arc, on associe une fonction de valuation entière: $\text{Pré}(p,t)$ pour les arcs reliant une place à une transition et $\text{Post}(t,p)$ pour les arcs reliant une transition à une place. On peut définir le réseau sous la forme $\langle P, T, \text{Pré}, \text{Post} \rangle$

A un instant donné, le marquage M d'un RdP est défini par un vecteur d'entiers qui détermine le nombre de marques dans chaque place. Le marquage d'un réseau peut ainsi servir à la caractérisation de l'état du système modélisé. Les conditions d'évolution d'un marquage à un autre sont définies par les transitions.

Le tir d'une transition franchissable conduit à un nouveau marquage en retranchant de chaque place p en entrée de la transition $Pré(p, t)$ marques et en ajoutant à chaque place p en sortie de la transition $Post(t, p)$ marques.

La transition n'est donc franchissable que si chacune de ses places en entrée a un nombre de marques supérieur à la valuation de l'arc qui la relie à la transition.

A partir d'un marquage, plusieurs transitions peuvent être franchissables. Le choix est alors arbitraire.

Les fonctions Pré et Post sont représentées par des matrices à p lignes et t colonnes (p et t étant le nombre de places et de transitions).

Un réseau est dit **borné** si chacune de ses places a un nombre de marques qui est à tout instant fini.

Un réseau est dit vivant si toutes ses transitions peuvent être franchies à partir d'un marquage.

Un réseau est dit **sain ou sauf** pour un marquage initial M_0 si, pour tout marquage obtenu à partir de M_0 par une séquence de tirs finie, aucune place ne possède plus d'une seule marque.

Un réseau qui est à la fois vivant et sauf est dit **conforme**.

On appelle **graphe d'état** un réseau de Petri tel que, à chaque instant, une seule de ses places est marquée et ce, par une seule marque.

Un réseau de Petri autonome est dit **étiqueté** si on lui associe un vocabulaire et une application de l'ensemble des transitions vers le vocabulaire et qui associe à chaque transition une étiquette du vocabulaire.

Les *Réseaux de Petri temporisés* tiennent compte des durées d'exécution des actions associées aux places ou aux transitions du réseau.

- Si les actions sont associées aux *places*, alors la temporisation s'applique à la disponibilité de la marque: lorsque celle-ci entre dans une place, elle n'est disponible qu'après un délai après lequel on détermine si les transitions sont franchissables.
- Lorsque les actions sont associées aux *transitions*, la temporisation s'applique au franchissement des transitions: lorsqu'une marque entre dans une place, elle est immédiatement disponible pour déterminer les transitions franchissables. Toutefois, les marques ne seront déposées dans les places en sortie de la transition qu'après un certain délai.

Les *Réseaux de Petri stochastiques* sont les réseaux de Petri temporisés mais avec les *temps aléatoires* associés aux places ou aux transitions.

6.2.2 Files d'attente

Basé sur deux entités: *client* et *serveur* (un client arrive dans une file, attend un service, reçoit le service, puis se dirige vers une autre file ou quitte le système), ce formalisme est largement utilisé pour modéliser des systèmes dans la vie quotidienne (acheter un ticket de cinéma, sortir à la caisse de supermarché, etc.), dans l'informatique, dans les télécommunications, dans le transport, dans la gestion de production, etc. Fig. 2 représente schématiquement les éléments d'un système d'attente.

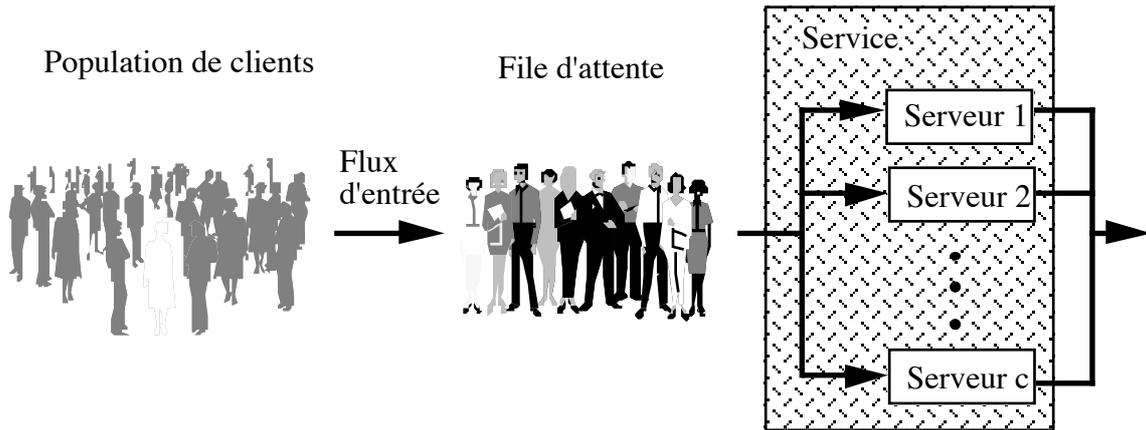


Fig. 2 Éléments d'un système d'attente

Ce système peut être modélisé en terme de files d'attente comme montré par la fig. 3 avec les variables qui peuvent nous intéresser.

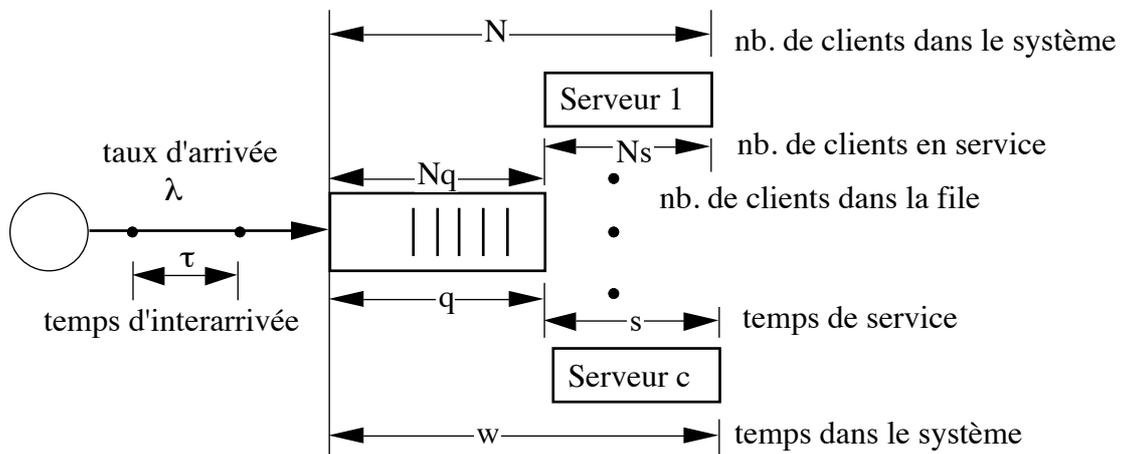


Fig. 3 Quelques variables aléatoires utilisées dans un modèle en files d'attente

6.2.3 Comparaison de formalismes

Fig. 4 et Fig. 5 donnent deux exemples simples pour montrer la modélisation d'une file d'attente en deux formalismes différents. Le lecteur peut faire une comparaison rapide pour constater que la modélisation en RdP peut conduire à des modèles de taille importante.

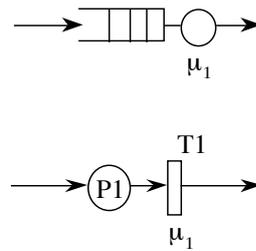


Fig. 4 une file d'attente

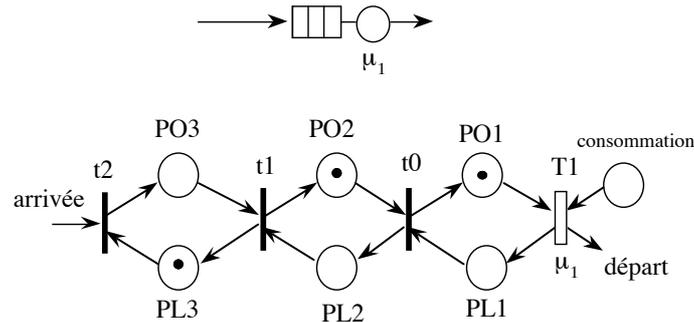


Fig. 5 FIFO à 3 places

Avant de terminer ce paragraphe, il convient de signaler que les réseaux de Petri étendus (temporisés et stochastiques) permettent également d'évaluer les performances des systèmes complexes (cf. [MAR 93] et les actes associées pour plus ample information). En effet, le formalisme en files d'attente est bon pour la description de clients dont le comportement est asynchrone. Quand il s'agit de dépendance entre les différents clients, ce formalisme montre un manque de puissance d'expression car on est obligé d'introduire d'autres mécanismes ce qui compliquera la résolution du modèle. Face au problème de modélisation des systèmes dans lesquels la synchronisation ou d'autres types de dépendance existent, les réseaux de Petri étendus commencent à occuper le terrain en vue de combler le manque de files d'attente. Quant à la résolution de ce type de modèle, il faut attendre encore un peu de temps pour que la recherche dans ce domaine progresse.

Au lieu d'utiliser les outils basés sur les réseaux de Petri (ou machines d'état-transition pour rester plus général) pour l'évaluation de performances, il est également possible de combiner d'une part, la technique formelle (avec outils/langages tels que SDL, ...), pour la spécification et la validation/vérification logique des systèmes (réseaux/protocoles), et d'autre part, les outils traditionnels basés sur des files d'attente pour l'évaluation des performances [DEM 92].

6.3 Techniques de Résolution de Modèles en Files d'attente

Trois techniques peuvent être utilisées:

- *mesure* d'un système existant (souvent une maquette),
- *étude analytique* (souvent un modèle mathématique en utilisant la théorie des files d'attente),
- *simulation* (souvent à événements discrets).

Pour la mesure d'un système réel, il s'agit de collecter des données statistiques sur un système en état de fonctionnement. Cela exige l'existence du système et le coût d'étude est souvent très élevé, et la durée d'étude longue. En fait, l'existence d'un système est déjà contradictoire avec cette étude, qui est souvent une phase de pré-implémentation.

La simulation d'un modèle demande:

- la connaissance parfaite du système étudié qui permet la définition d'un modèle simplifié adapté à l'étude, i.e. le modèle n'est pas forcément une description en langage de simulation du système exact. Il peut se limiter aux paramètres et aux comportements intéressants tout en conservant une certaine finesse.
- des données d'entrées. En considérant le modèle comme une boîte noire, ces données sont essentielles pour démarrer la simulation. La collecte de ces données est une étape primordiale. On peut en déduire soit des caractères statistiques (et une distribution de probabilité) s'il s'agit d'une entrée aléatoire, soit un tableau de données si l'entrée est déterministe.

6.3.1 Théorie des files d'attente

La théorie des files d'attente (queueing theory) a pour objectif étudier les phénomènes d'attente. Les bases de cette théorie est la *probabilité* et les *processus stochastiques*. Il n'est pas utile (et impossible) de reporter dans ce paragraphe de nombreux résultats de cette théorie. Les concepts de base sont introduits dans les 5 chapitres précédents de ce polycopié. Le lecteur désirant d'approfondir sa connaissance en la matière peut consulter un très bon livre de Kleinrock [KLE 75], [KLE 76] ou un bon livre en français [BAY 00].

6.3.2 Simulation à événements discrets

La *simulation* représente un ensemble de techniques permettant d'approcher le comportement d'un système quelconque de façon stochastique. Le *simulateur* est le programme contenant l'algorithme utilisé pour simuler le système étudié.

Il est bien sûr possible d'écrire un programme de simulation en utilisant un langage de programmation générique tel que Java ou C. C'est d'ailleurs très formateur d'écrire ainsi au moins une fois dans sa vie un simulateur. Mais il s'avère très vite que l'effort de programmation est trop important pour développer des simulateurs complexes de grande taille.

Afin de faciliter l'utilisation de la simulation, de nombreux efforts ont été faits sur la construction des outils de simulation dédiés. Ces outils sont souvent dotés d'une interface graphique facilitant ainsi la construction des modèles et l'exploitation des résultats. On peut trouver beaucoup d'outils de simulation, citons les quelques uns les plus connus: Opnet (www.opnet.com) qui dispose des bibliothèques de modèles pré-construits, ns (www.isi.edu/nsnam) qui est bien connu du monde académique et gratuit, OMNeT++ (www.omnetpp.org) qui est aussi gratuit et suit le même principe que Opnet (sans les bibliothèques de modèles), ...

En France, Modline/Qnap2 (www.simulog.fr) est un logiciel connu. Il est développé par l'INRIA et Bull dans les années 80. C'est un logiciel très facile à utiliser et bien adapté à modéliser les réseaux de files d'attente. QNAP2 est le langage de description de modèles de simulation et le moteur de simulation, tandis que Modline ajoute une interface graphique et un

environnement de développement intégré. Il n'est malheureusement plus maintenu et son interface graphique est parfois déroutante pour un débutant.

Un autre logiciel plus récent et qui suit le même principe et offre les fonctionnalités similaire à Modline/QNAP2 est JMT (Java Modelling Tools) développé par Polytechnique de Milan (<http://jmt.sourceforge.net/>). C'est un logiciel bien adapté pour la pédagogie et la recherche.

6.4 Simulation à événements discrets

Comme expliqué précédemment la simulation est une approche d'évaluation de performances très générale. Complémentaire à l'approche analytique, elle n'est pas contrainte pas des hypothèses bien formées, et dans ce sens, elle permet d'évaluer les performances des systèmes d'un très large spectre. Il faut faire tout de même attention quant à l'interprétation de résultats à l'issue des simulations. Il faut s'assurer en particulier que le système simulé a bien un régime stationnaire si les performances asymptotiques sont attendues. Il faut aussi connaître la précision des résultats afin de fournir des conclusions statistiquement parlant fiable. Dans cette section nous présentons rapidement les éléments de base à connaître quand on mène une étude de simulation.

6.4.1 Simulation par événements et gestion de l'échéancier

Une simulation est conduite par des événements. Un événement correspond à un changement d'état du système. Souvent, la simulation fonctionne avec un échéancier des événements et le simulateur saute d'un événement au suivant, sans que le temps entre les deux événements prenne du temps de calcul puisqu'il ne se passe rien entre les événements (le temps est compressé). Lorsque le simulateur passe d'un événement au suivant, la date saute de la valeur courante à celle du nouvel événement. L'ensemble des événements avec leur date constitue en l'échéancier de la simulation.

Pour mieux comprendre, prenons un exemple simple d'une file d'attente.

La file d'attente à simuler est une file FIFO de capacité illimitée, avec un flux d'arrivée de clients suivant un processus de Poisson du taux d'arrivée λ (autrement dit les intervalles d'inter-arrivées suivent une loi exponentielle du taux λ), et un temps de service suivant une distribution exponentielle du taux μ . (C'est ce qu'on appelle M/M/1 dans le jargon de la théorie des files d'attente)

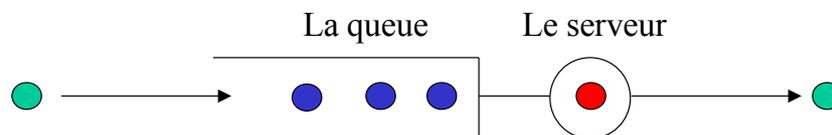


Figure 6. Une file d'attente

On simule cette file d'attente jusqu'à n clients soient sortis. On veut estimer :

- Le temps moyen d'attente d'un client dans la queue: $d(n)$
- Le nombre moyen de clients dans la queue: $Q(n)$

Ces deux paramètres peuvent être approchés par les estimations suivantes :

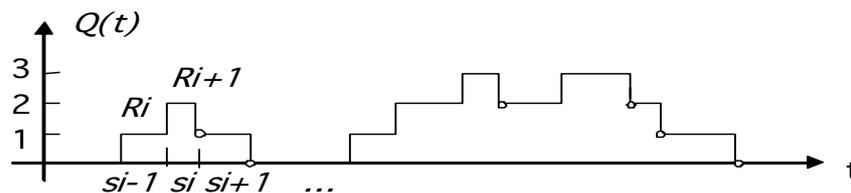
$$\hat{d}(n) = \frac{\sum_{i=1}^n D_i}{n} \quad \text{et} \quad \hat{Q}(n) = \frac{\int_0^T Q(t)dt}{T} = \frac{\sum_{i=1}^m R_i}{T}$$

où :

- D_i : temps d'attente du client i
- T : temps pour traiter n clients
- $Q(t)$: nombre de clients dans la queue à l'instant t
- R_i : surface du rectangle sous $Q(t)$ entre s_{i-1} et s_i (cf. Fig.7): nombre d'événements produits pendant $[0, T]$

L'estimation de $d(n)$ est très simple. Il s'agit simplement de cumuler les $D_i = \text{instant du départ} - \text{instant d'arrivée du client } i$, puis à la fin de simulation, le diviser par le nombre total de clients passés par le système.

Le principe du calcul du nombre moyen de clients dans le système peut être illustré par la figure suivante qui montre une trajectoire de l'évolution de $Q(t)$. $R_i = Q(s_i) * (s_i - s_{i-1})$



Par la théorie des files d'attente, nous avons la valeur théorique du temps moyen d'attente de clients et le nombre moyen de clients :

$$\bar{d} = \frac{1}{\mu - \lambda}$$

$$\bar{n} = \frac{\rho}{1 - \rho}$$

Notons que ces deux paramètres sont liés par la formule de Little :

$$\bar{n} = \lambda \bar{d}$$

Une question se pose alors. Quelle est la valeur de n (ou de T) convenable pour que nos résultats de simulation soit fiable ? C'est-à-dire que les valeurs estimées par la simulation sont suffisamment proches de celles données par la théorie ? Nous verrons ce problème dans un paragraphe plus loin ayant trait avec l'estimation de la précision des résultats de simulation.

Afin de simuler ce système, on fait recours à la méthode de simulation à événements discrets. Il s'agit de s'intéresser aux variables d'état et aux événements qui provoquent le changement

d'état. Dans notre cas, le paramètre nécessaire (variable d'état) est le nombre de clients en attente dans la queue.

Deux types d'événements permettent le changement d'état : *arrivée* et *départ* d'un client.

Pour faire avancer le temps de simulation (une horloge virtuelle), on avance le temps à l'événement prochain. La figure suivante donne une illustration de ce principe.

Soit :

- t_i = l'instant d'arrivée du $i^{\text{ème}}$ client ($t_0 = 0$)
- $A_i = t_i - t_{i-1}$ = temps d'inter-arrivées entre $(i-1)^{\text{ème}}$ et $i^{\text{ème}}$ arrivée de clients
- S_i = temps de service du serveur pour service le $i^{\text{ème}}$ client (temps d'attente exclus)
- D_i = temps d'attente dans la queue du $i^{\text{ème}}$ client
- $c_i = t_i + D_i + S_i$ = l'instant où le $i^{\text{ème}}$ client finit son service et quitte le système
- s_i = l'instant d'occurrence du $i^{\text{ème}}$ événement du type quelconque (le $i^{\text{ème}}$ valeur que l'horloge de simulation prend, excluant la valeur $s_0 = 0$)

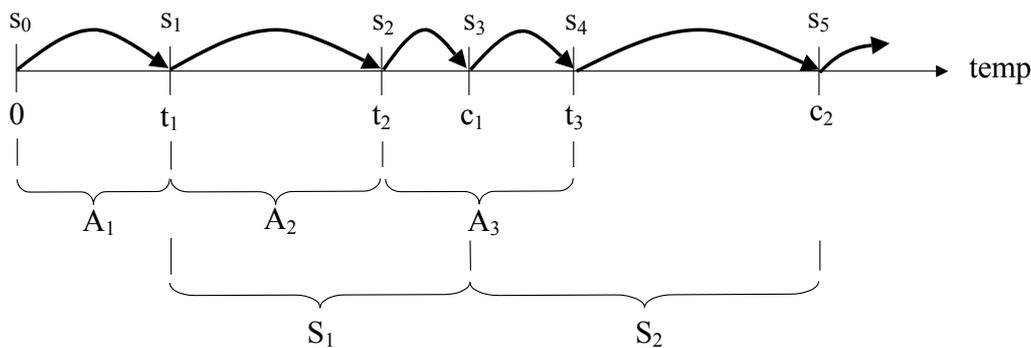


Figure 7 : une trajectoire des événements

On comprend maintenant que la tâche principale d'un simulateur est de :

- générer les différents événements de façon aléatoire en suivant une certaine loi de probabilité
- gérer la liste des événements (comme ce qu'illustre la Figure 7) : instant de prochaine arrivée t_i et instant du prochain départ c_i .
- Faire la statistique sur des paramètres d'intérêt (e.g. nombre de clients dans la file d'attente et le temps d'attente de chaque client).

6.4.2 Génération des événements suivant une loi de probabilité

Le modèle de simulation doit reproduire des événements aléatoires : arrivée un paquet dans le réseau, occurrence de pannes, etc. Le modèle de ces comportements consiste en variables aléatoires de loi connue. En informatique (programmation), la base pour générer une variable aléatoire qui suit une loi quelconque est l'utilisation de la fonction *Random()* qui permet de tirer des valeurs uniformément réparties dans l'intervalle $]0,1[$.

Pour illustrer le principe, prenons le cas d'une variable aléatoire continue: X

Plusieurs méthodes existent pour générer des variables aléatoires. Nous présentons la méthode par fonction inverse qui est générale est facile à programmer. Cette méthode consiste à :

- tirer une variable u uniformément distribuée en $]0,1[$
- prendre $X = F^{-1}(u)$ tel que $u = F(x)$

X suit alors la loi cherchée.

Preuve: il suffit de vérifier que $P[X \leq x] = F(x)$

$$P[X \leq x] = P[F^{-1}(u) \leq x] = P[u \leq F(x)] = F(x)$$

La dernière égalité s'obtient car u est uniformément distribuée dans $]0,1[$, ce qui donne :

$$P[u \leq a] = a; (0 \leq a \leq 1)$$

Maintenant, examinons le cas de la loi exponentielle $F(x) = 1 - e^{-\lambda x}$

Soit u une variable aléatoire uniformément distribuée en $]0, 1[$.

On note : $F(x) = 1 - e^{-\lambda x} = u$.

La fonction inverse de $F(x)$ s'obtient simplement par l'expression de x en fonction de u .

$$e^{-\lambda x} = 1 - u$$

$$x = -\ln(1-u)/\lambda$$

Comme u peut être généré par la fonction `Math.random()` en programmation (en java), on a :

$$x = -\text{Math.log}(1 - \text{Math.random}()) / \lambda = -\text{Math.log}(\text{Math.random}()) / \lambda$$

6.4.3 Résultats statistiques et estimation de la précision²

Pendant une simulation, nous pouvons bien sûr nous intéresser à la trace d'une simulation pour comprendre les comportements transitoires. Mais la plupart du temps, nous nous intéressons plutôt au comportement d'un système à long terme, autrement dit, à son régime stationnaire.

Comme les événements sont souvent aléatoires, les résultats qui représentent le comportement du système sont alors d'ordre probabiliste. On peut s'intéresser à la valeur moyenne, la variance, voire à la connaissance complète du système par la distribution de probabilité.

Tous ces résultats ne sont bien sûr qu'une estimation des valeurs réelles car la durée d'une simulation ne peut être allongée infiniment (tends vers l'infini). D'où vient le problème de la précision de l'estimation car la simulation est une expérience statistique.

Prenons un exemple: estimer, par simulation, $E(W)$ dans une file M/M/1.

Théoriquement:

$$E(W) = 1/\mu(1 - \rho)$$

Pour $\rho=0.9$ et $\mu=1$, simuler 10 000 clients, puis faire la moyenne, on obtient une valeur de temps moyen.

On répète 100 fois, on obtient des valeurs « proches » de 10 mais assez dispersées! (figure 8)

² Cette section est basée sur le polycopié de Hébuterne de l'INT.

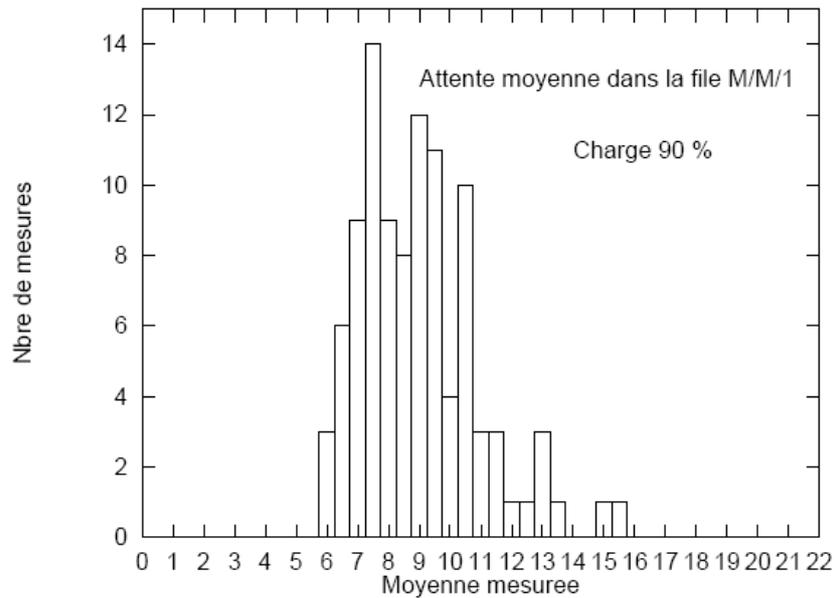


Figure 8. Dispersion des mesures (extrait du polycopié de Hébuterne)

Souvent on exige un niveau de confiance pour une expérience:

$$P[a \leq \text{la vraie valeur du paramètre} \leq b] = 0,95$$

L'intervalle de confiance (confidence interval) est alors : $[a,b]$

Il existe de nombreuses méthodes d'estimation de la précision. Dans ce document, nous présentons d'abord la méthode la plus simple (et peut-être moins précise dans certains cas). Il s'agit de la Méthode des blocs.

Méthode des blocs (batch means):

Considérons l'exemple comme une expérience de 1 000 000 mesures, découpée en $N=100$ blocs de $m=10\ 000$ mesures. On a :

Moyenne sur chaque bloc: $Y_k = \frac{1}{m} \sum_{i=1}^m X_i$

Moyenne empirique: $\bar{Y} = \frac{1}{N} \sum_{k=1}^N Y_k$

Variance empirique: $\hat{\sigma}^2 = \frac{1}{N-1} \sum_{k=1}^N (Y_k - \bar{Y})^2$

Supposons l'indépendance entre blocs, selon le théorème central limite, \bar{Y} est asymptotiquement distribué selon une loi Normale de variance σ^2/N .

Loi Normale (Gaussienne ou Laplace-Gausse):

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Loi Normale centrée réduite (centrée et de variance unité) :

$$\varphi(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}$$

$$\Phi(x) = \int_{-\infty}^x \varphi(t) dt = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$$

$$t = x - \mu$$

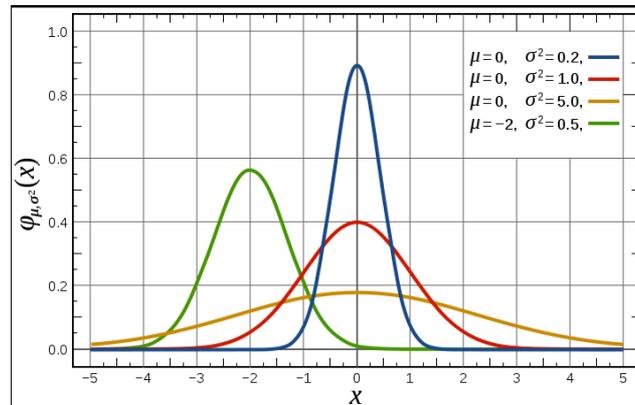
$(x - \mu) / \sigma$ est une v.a. avec variance = 1

$$P\left[Z = \frac{\bar{Y} - E(Y)}{\sigma / \sqrt{N}} \leq x\right] = \Phi(x)$$

Où Z est la valeur de l'écart entre $E(Y)$ et \bar{Y} .

Avec la symétrie de la loi Normale, on peut réécrire l'équation sous la forme suivante :

$$P[|Z| \leq x] = \Phi(x) - \Phi(-x) = 1 - 2\Phi(-x)$$



Maintenant, on veut un niveau de confiance de 95% : $P[|Z| \leq x] = 1 - 2\Phi(-x) = 0,95$

$$\Phi(-x) = 0,025$$

$$\Phi(x) = 1 - \Phi(-x) = 0,975$$

A l'aide de la table de valeurs de la loi Normale centrée réduite (cf. par exemple : http://fr.wikipedia.org/wiki/Loi_normale), on sait que : $\Phi(1,96) = 0,975$.

On utilise $\hat{\sigma}$ comme estimateur de σ , on a enfin :

$$P\left[\left|\frac{\bar{Y} - E(Y)}{\hat{\sigma} / \sqrt{N}}\right| \leq 1,96\right] = 0,95$$

L'inversion de cette relation permet de trouver l'intervalle de confiance :

$$\bar{Y} - 1,96 \frac{\hat{\sigma}}{\sqrt{N}} \leq \text{Moyenne} \leq \bar{Y} + 1,96 \frac{\hat{\sigma}}{\sqrt{N}}$$

Ce résultat peut être utilisé pour régler la longueur de simulation, c'est-à-dire le nombre de blocs nécessaires pour obtenir une précision acceptable par rapport au niveau de confiance exigé.

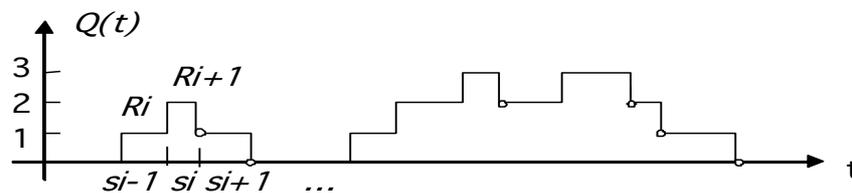
Dans la pratique, $N=30$ donne déjà une bonne approximation pour la plupart de configurations [FIC02]. Mais attention, il est vrai si et seulement si l'hypothèse de l'indépendance entre blocs est vérifiée. Bien sûr, si le découpage de blocs n'assure pas l'indépendance entre blocs successifs, la méthode des blocs ne s'applique plus car ni l'estimateur de variance, ni la loi de grands nombres n'est valable.

Il existe dans ce cas d'autres méthodes, notamment la méthode régénérative basée sur le processus de renouvellement.

Dans la pratique pour une file d'attente, nous pouvons découper les blocs à des points où le nombre de clients devient nul.

Méthode régénérative :

Prenons toujours notre exemple de la file M/M/1. Voici une trajectoire possible de l'évolution du nombre de clients dans la file d'attente sur l'axe du temps.



Vis-à-vis du serveur, on observe une alternation entre une période occupée et une période oisive. Intuitivement, lorsque la file se vide, le prochain client arrivant va trouver un serveur libre. Son traitement, ainsi que tous ses successeurs, ne dépendra pas de ses prédécesseurs. Ainsi, chaque période occupée conduit à une série de mesures indépendantes les unes des autres. Conclusion : les périodes occupées successives vont donner une suite de variables aléatoires indépendantes.

Soit une suite de points (S_i) t.q. $S_i = T_1 + T_2 + \dots + T_i$, où T_k sont des v.a. i.i.d. de loi F : $P[T_1 \leq t] = P[T_k \leq t] = F(t)$. La séquence $(S_i, i \geq 0)$ est un *processus de renouvellement*.

Plus généralement, le processus $\{Z_t, t \geq 0\}$ est un *processus régénératif* s'il existe une suite $(S_i, i \geq 0)$, qui est un processus de renouvellement, tel que $\{Z_t, S_i \leq t \leq S_{i+1}\}$ soit une réplique i.i.d. de $\{Z_t, S_1 \leq t \leq S_2\}$.

On peut bien sûr aussi définir une version discrète. Ainsi, pour une file M/M/1, la suite b_k des numéros de clients qui arrivent et trouvent le serveur libre est un processus de renouvellement. Le temps d'attente du n -ème client est un processus régénératif discret, dans lequel le processus de renouvellement est inclus (ou « imbedded »). On note $\{b_k\}$ le renouvellement discret. a_k désigne le nombre de clients constituant le k -ième période occupée. On a :

$$b_k = b_{k-1} + a_k$$

On note $\{Z_n, n \geq 0\}$ le processus régénératif, qui se reproduit « identiquement » après chaque b_k .

Soit une fonction f définie sur le processus, on cherche à calculer $E[f(Z)]$. Ainsi, si $f(x)=x$, on calcul la moyenne, etc. Le théorème clé affirme que :

$$E[f(Z)] = \frac{E[\sum_{i=1}^{a_1} f(Z_i)]}{E[a_1]}$$

C'est-à-dire qu'on peut restreindre l'étude du phénomène à un seul intervalle de régénération : on somme les mesures sur un intervalle de régénération. L'espérance mathématique divisée par le nombre moyen de mesures dans l'intervalle, donne la moyenne cherchée.

Maintenant, appliquons cette méthode à la simulation.

On note W le temps d'attente des clients (W joue le rôle de Z), et Y_i les sommes sur chaque intervalle de renouvellement :

$$Y_i = \sum_{j=b_i+1}^{b_i+a_i+1} f(W_j)$$

Remarquons que nous pouvons toujours écrire :

$$E[W] = \lim_{n \rightarrow \infty} \frac{W_1 + \dots + W_n}{n}$$

$$E[W] = \lim_{k \rightarrow \infty} \frac{Y_1 + \dots + Y_k}{a_1 + \dots + a_k}$$

$$E[W] = \lim_{k \rightarrow \infty} \frac{(Y_1 + \dots + Y_k) / k}{(a_1 + \dots + a_k) / k}$$

D'autre part, on a : $\lim_{k \rightarrow \infty} (Y_1 + \dots + Y_k) / k = E[Y_1]$ et $\lim_{k \rightarrow \infty} (a_1 + \dots + a_k) / k = E[a_1]$. Le théorème affirme simplement que la limite du quotient est égale au quotient des limites :

$$E[W] = \lim_{k \rightarrow \infty} \frac{(Y_1 + \dots + Y_k) / k}{(a_1 + \dots + a_k) / k} = \frac{E[Y_1]}{E[a_1]}$$

Dans la simulation, la détection des points de régénération peut se faire facilement (file d'attente vide), ce qui nous permet de collecter les Y_i successifs. L'expérience est conduite pour N intervalles de régénération successifs. A la différence de la méthode des blocs, les Y_i successifs sont maintenant indépendants les uns des autres par construction. Il est à noter que les intervalles successifs n'ont plus la même longueur (ne servent pas le même nombre de clients). Par conséquent, le calcul revient à estimer $E[W] = E[Y_1]/E[a_1]$, et donc à trouver l'intervalle de confiance sur un quotient.

Nous rappelons que a_k désigne le nombre de mesures effectuées dans le k -ième intervalle de renouvellement. La séquence $\{Y_i, a_i\}$ est une suite de vecteurs i.i.d. Appelons r la vraie valeur du quotient :

$$r = \frac{E[Y_1]}{E[a_1]}$$

La valeur estimée par la simulation est :

$$\hat{r} = \frac{\sum Y_k}{\sum a_k}$$

La suite $(V_j = Y_j - ra_j)$ est une suite i.i.d. de moyenne nulle. On montre que la variance de V est donnée par :

$$\begin{aligned} \sigma_V^2 &= \text{Var}(V) = \text{Var}(Y) - 2r\text{Cov}(Y, a) + r^2\text{Var}(a) \\ &= E[Y^2] - 2rE[Ya] + r^2E[a^2] \end{aligned}$$

On obtient un estimateur de cette quantité à partir des mesures collectées de la simulation.

Considérons la moyenne empirique $\bar{V} = (\sum V_k) / k = \bar{Y} - r\bar{a}$. Le théorème de la limite centrale nous apprend que \bar{V} tend vers une loi normale, dont la variance sera σ_V^2 / N (où N est le nombre total de mesures), ce qui se traduit en :

$$\lim_{n \rightarrow \infty} P\left\{ \frac{(\sum Y_k - r \sum a_k) / N}{\sigma_V / \sqrt{N}} \leq x \right\} = \Phi(x)$$

Il faut introduire l'estimateur \hat{r} , et $\bar{a} = (\sum a_k) / N$. Il vient alors :

$$P\left\{ (\hat{r} - r) \frac{\bar{a} \sqrt{N}}{\sigma_V} \leq x \right\} = \Phi(x)$$

D'où enfin l'intervalle de confiance pour un niveau de confiance de 95% :

$$P\left\{ \hat{r} - \frac{1,96\sigma_V}{\bar{a}\sqrt{N}} \leq E[W] \leq \hat{r} + \frac{1,96\sigma_V}{\bar{a}\sqrt{N}} \right\} \approx 0,95$$

Récapitulation :

On vient de démontrer comment calculer l'intervalle de confiance dans la méthode régénérative. Pour avoir une idée concrète de son utilisation, reprenons notre exemple de la file M/M/1.

Il s'agit de calculer une moyenne. Nous allons observer N cycles de régénérations (la détection des points de régénération dépend du problème posé. Et dans notre cas, les instants où la file devient vide). On note :

- a_k le nombre de mesures collectées au cours du cycle numéro k (i.e. le nombre de clients du cycle dans notre exemple),
- Y_k le cumul des mesures du cycle numéro k (i.e. la somme des temps d'attente des clients du cycle).

En même temps que se déroule la simulation, et à la fin de chaque cycle, la collecte des résultats consiste à tenir à jour les comptages suivants :

- $Y1 = \sum Y_k$
- $Y2 = \sum (Y_k)^2$

- $A1 = \sum a_k$
- $A2 = \sum (a_k)^2$
- $Ya = \sum Y_k a_k$

A la fin de la simulation, N cycles ayant été observés, on calculera :

$$\bar{W} = Y1 / A1$$

$$Var(V) = \frac{Y2 - 2 \times \bar{W} \times YA + A2 \times \bar{W}^2}{N - 1}$$

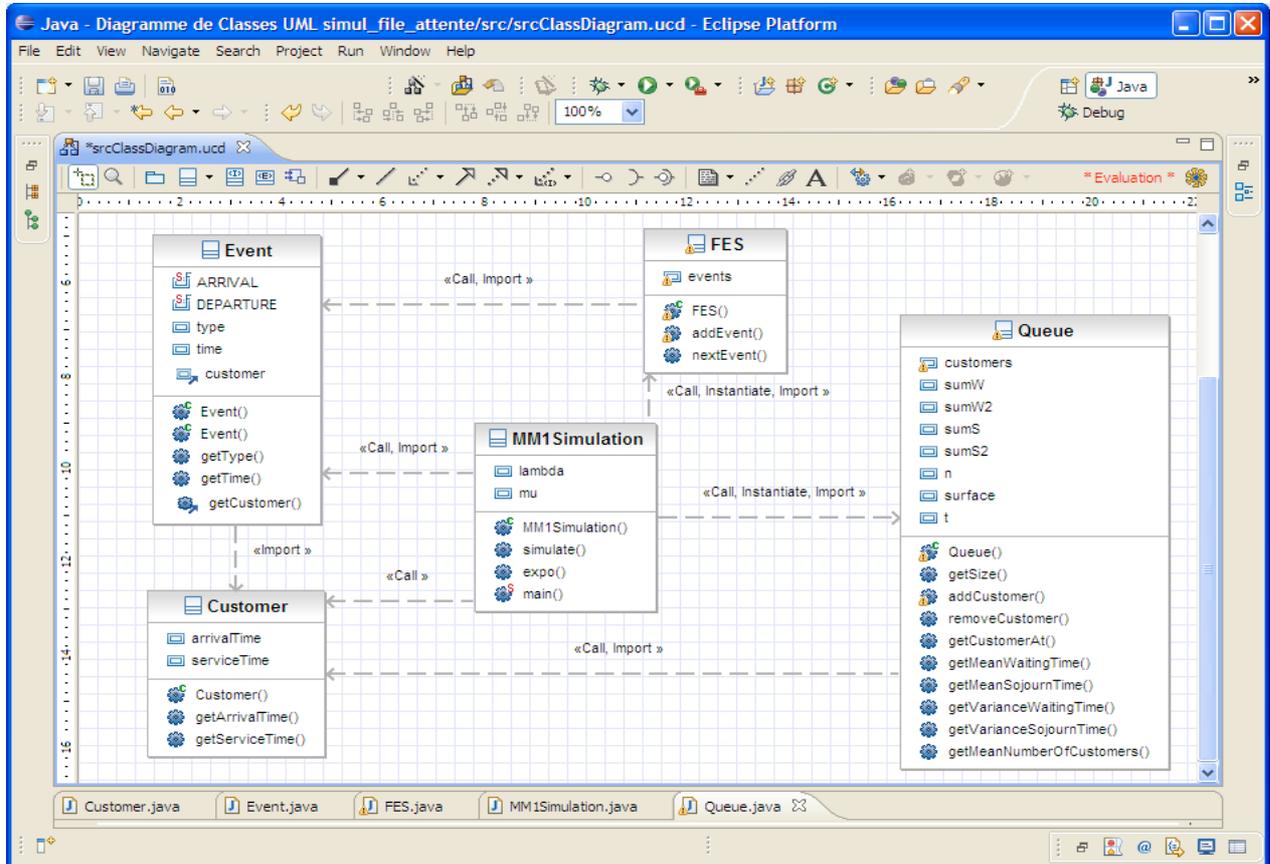
Quand la longueur de simulation augmente (nombre de cycles augmente), V doit fluctuer autour d'une valeur constante. L'imprécision ΔW sera donnée (au niveau de confiance 95%) :

$$\Delta W = 1,96 \frac{\sqrt{NV}}{A1}$$

et avec un risque d'erreur de 5%, $\bar{W} - \Delta W \leq E[W] \leq \bar{W} + \Delta W$.

6.4.4 Simulation (en java) d'une file M/M/1

Pour tout comprendre, la meilleure façon est de programmer un simulateur simple par soi-même. Voici le diagramme de classes, ainsi que quelques codes de définition de classes java pour vous aider à programmer un simulateur de M/M/1.



```

public class Client {
    protected double tempsArrivee; //c'est un instant
    protected double dureeService; //c'est une durée

    public Customer(double arrivalTime, double serviceTime) {
        this.tempsArrivee = arrivalTime;
        this.dureeService = serviceTime;
    }
    // si les deux attributs sont declares protected ou private,
    // il faut des méthodes publiques d'accès à la valeur de ces attributs
    public double getArrivalTime() {
        return arrivalTime;
    }

    public double getServiceTime() {
        return serviceTime;
    }
}

```

```

import java.util.Vector;
public class Queue {
    /* Liste des clients dans la queue */
    Vector lesClients;

    //Construire une queue vide
    public Queue() {
        lesClients = new Vector();
    }

```

```

public class Event {
    protected int type; //0 correspond à une arrive, 1 à un départ
    protected double instant; //instant où un événement s'est produit

    public Event(int type, double time) {
        this.type = type;
        this.instant = time;
    }
    public int getType() {
        return type;
    }
    public double getTime() {
        return instant;
    }
}

```

```

import java.util.Vector;
public class listeEvents { // cette classe modélise la liste des événements
futurs
    Vector events;

    public listeEvents () {
        events = new Vector();
    }

    public void addEvent(Event newEvent) {
        int insertIndex = 0; //sert à repérer l'endroit où l'événement newEvent
doit être insérer
        //on compare l'instant de newEvent avec l'instant des autres événements
déjà dans la liste
        while (insertIndex < events.size()) {
            Event e = (Event) events.elementAt(insertIndex); //on extrait le
insertIndex ème événement
            if (e.getTime() > newEvent.getTime()) break; //si l'instant de
newEvent est le plus près,

//on sort de la boucle
            insertIndex++; //sinon on incrémente insertIndex, et on reboucle
        }
        events.insertElementAt(newEvent, insertIndex); //on insert l'événement
newEvent au bon endroit
    }
}

```

La classe principale :

```

public class MM1Simulation {
    double lambda;
    double mu;
    ListeEvents liste;
    Queue q;
    double t;
}

public MM1Simulation (double lambda, double mu) {
    this.lambda = lambda;
    this.mu = mu;
    q = new Queue();
    liste= new listeEvents();
}

public double expo(double taux){
    return -Math.log(Math.random())/taux;
}

public void simulate (double simLength) {
//à remplir avec l'algorithme suivant
}

public static void main(String[] arg) {
    MM1Simulation s = new MM1Simulation(0.5, 1.0);
    s.simulate(1000000);
}

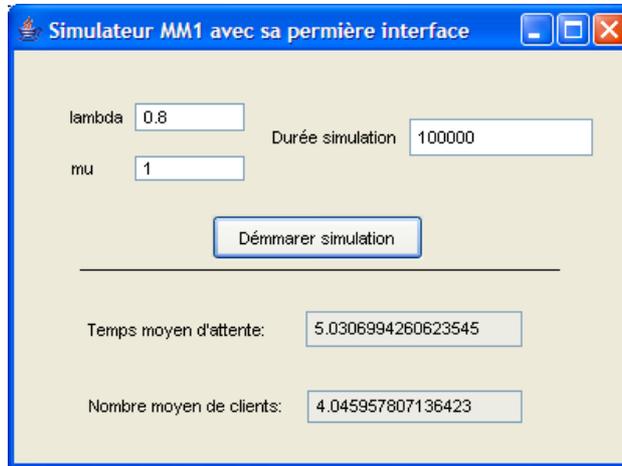
```

L'algorithme de la simulation est le suivant :

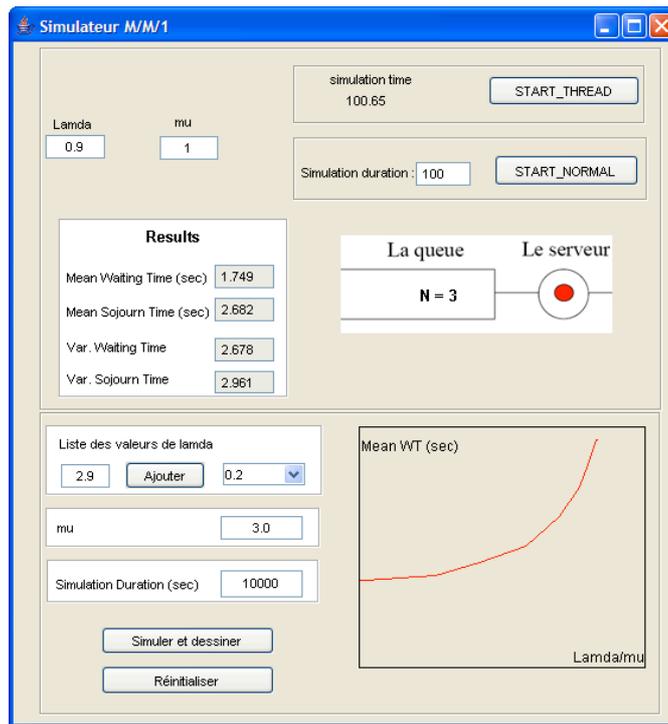
- La simulation commence à $t=0$. A cet instant, on génère le prochain événement s_1 qui correspond forcément à une arrivée de client.
- On ajoute cet événement dans la liste des événements
- On répète ensuite la procédure suivante jusqu'à la fin de simulation désirée (une durée de simulation ou bien un nombre maximum de clients traités)
 - o On lit dans l'ordre le prochain événement de la liste et on l'enlève de la liste (une fois l'événement est utilisé, il sera supprimé de la liste car n'est plus utile)
 - o On avance le temps à la date de cet événement (par exemple $t=t_1$, c'est-à-dire la date d'arrivée de l'événement s_1)
 - o Selon le type d'événement (une arrivée ou un départ), on fait le traitement correspondant :
 - S'il s'agit d'une arrivée,
 - on génère l'arrivée suivante (qui sera à l'instant $t+\text{expo}(\lambda)$) et on le met dans la liste d'événements mais à sa place appropriée (supposant qu'il y a une méthode dans la classe liste des événements qui saura l'insérer au bon endroit)
 - on calcule la durée de service du client (par exemple, pour le 1^{er} client de l'exemple de la figure ci-dessus, $S_1=\text{expo}(\mu)$)
 - on ajoute le client dans la file d'attente des clients avec son instant d'arrivée (e.g. t_1) et sa durée de service (e.g. S_1)
 - un cas particulier à traiter : si le client est le seul dans la file d'attente, il sera forcément celui qui va sortir prochainement. Il faut donc ajouter un événement correspondant au départ de ce client dans la liste des événements dont l'instant de départ est $t+$ durée de service. Sinon on ne l'insère pas dans la liste des événements mais laisse le traitement dans le cas où il y a un client qui quitte la file (cas suivant)
 - S'il s'agit d'un départ,
 - On supprime le client de la file d'attente. Si la file devient vide, on ne fait rien

- Si la file n'est pas vide, on génère l'événement correspondant au départ du client suivant (qui sera à l'instant $t + \text{expo}(\mu)$) et on le met dans la liste d'événements mais à sa place appropriée (supposant qu'il y a une méthode dans la classe liste des événements qui saura l'insérer au bon endroit)
- Fin de répétition

Vous pouvez aussi programmer une interface graphique simple :



Ou bien une interface incluant l'animation :



Enfin, on vous demande de traiter les résultats de simulations :

- Comparer les résultats de simulation avec les résultats théoriques (nombre moyen de clients et temps moyen d'attente)
- Comment obtenir les résultats de simulation avec un niveau de confiance de 95% ? Et avec quel intervalle de confiance pour chaque valeur simulée ?
- Etendre le simulateur pour simuler une file M/D/1

6.5 Critères de Performances des Réseaux

Les critères de performances ne sont pas toujours les mêmes du point de vue d'un utilisateur, d'un administrateur et d'un concepteur, car leurs préoccupations ne sont pas les mêmes. Ces trois individus ont des demandes parfois communes et parfois contradictoires. Mais quel que soit le critère de performance considéré par un individu, ce critère prend la forme "rapport coût-performance". Un utilisateur peut très bien demander un certain nombre de services avec un coût minimum. Un administrateur du réseau souhaite un gain maximum en investissant au minimum. Le point de vue du concepteur est alors plus équilibré. Il essaye de trouver la solution satisfaisant à la fois l'administrateur et l'utilisateur. On peut dire que les critères de performances de l'utilisateur et de l'administrateur sont inclus dans ceux du concepteur. Dans la suite, nous regardons d'une façon générale les critères de performances définis par un concepteur.

Les trois principaux critères utilisés pour évaluer les performances d'un réseau local sont:

- l'**efficacité** du protocole qui s'exprime par le rapport entre la quantité d'information utile, c.à.d sans tenir compte des trames de service (gestion ou "overhead"), et la quantité totale d'information véhiculée par le médium. Ce critère est généralement utilisé pour estimer l'efficacité de la structure des trames d'un protocole d'accès.
- le **taux d'utilisation** du médium (ou "throughput") qui s'exprime par le rapport entre la quantité d'information véhiculée par le médium (par seconde) et le débit du réseau. Ce critère est essentiellement utilisé pour évaluer globalement les performances. Un taux d'utilisation faible implique soit que le médium de transmission est peu sollicité par les stations, soit que la technique d'accès utilise "mal" la capacité du médium.
- le **délai** de transfert (ou temps de réponse) qui exprime le temps d'attente, pour une station donnée, entre l'instant d'arrivée d'une trame dans la station (souvent mis en file d'attente) et la réception de celle-ci par la station destinatrice. Ce critère est souvent exprimé en terme de valeur moyenne ou maximale.

Remarque 1: Un critère de puissance ("Power") peut être utilisé pour remplacer "taux d'utilisation" et "délai". Il est défini tout simplement comme le rapport entre "taux d'utilisation" et "délai".

Remarque 2: On distingue le *délai de transfert* et le *délai d'accès*. Le délai d'accès d'un protocole d'accès au médium correspond au délai d'attente de la trame qui se trouve à la première place de la file d'attente. Pour ce délai, on ne compte ni le temps d'attente pour qu'une trame se déplace de la dernière à la première place de la queue (file d'attente), ni le délai de transmission. Pour un utilisateur, c'est toujours le délai de transfert (temps de réponse de bout en bout) qui compte.

6.6 Évaluation de Performances des Réseaux

A présent, nous abordons la modélisation des réseaux de communication qui représente un problème de premier plan. En effet, la multiplication des réseaux de transmission associée aux nombreuses possibilités de connexion, conduisent à des systèmes d'une grande complexité qui mettent en jeu d'importantes quantités de ressources matérielles et logicielles. La conception et l'exploitation d'un réseau ainsi que la mise en place de

possibilités de communication apparaissent alors comme des tâches extrêmement délicates. L'utilisation d'outils de modélisation et d'évaluation de performance représente, de plus en plus, une alternative pour appréhender, dimensionner et optimiser les réseaux de communication et les systèmes informatiques en général.

Il n'existe malheureusement pas de méthode systématique pour établir des modèles des réseaux de communication; en effet, aux problèmes de modélisation que l'on peut rencontrer, il faut ajouter la multiplicité des fonctions qui interviennent dans la conception d'un réseau de transmission. De plus, les moyens de communication se diversifient en fonction des contraintes géographiques et des applications. Généralement, nous ne rencontrerons pas les mêmes problèmes en étudiant les réseaux grandes distances, les réseaux locaux, les réseaux sans fils ou les réseaux satellites, chacun de ces systèmes de communication possédant des propriétés particulières. De même, nous devons aborder l'étude de différents protocoles, chacun d'eux permettant la mise en oeuvre de services divers de qualités différentes.

Avec de nombreux travaux effectués dans ce domaine ainsi que l'expérience acquise, nous pouvons tout de même donner quelques règles qui sont utiles pour établir le modèle du protocole. Dans la plupart des cas, les modèles pour l'évaluation des protocoles d'accès multiple s'appuient sur le partage d'une seule voie de communication par plusieurs stations qui émettent et/ou reçoivent des messages. On distingue l'attente (pour accéder à une ressource) du service (possession de la ressource) afin d'utiliser les modèles de réseaux de files d'attente. Un réseau de files d'attente est composé d'un ensemble de stations de service (modélisant les ressources: par exemple canal de transmission, émetteur...) et d'un ensemble de clients (messages, paquets, trames...). Les files d'attente devant chaque station représentent les activités qui ont besoin d'accéder, ou qui ont déjà accédé à la ressource. Un réseau de files d'attente est caractérisé par un processus stochastique représentant l'arrivée des clients dans le réseau, les temps de service des clients aux stations, le cheminement des clients d'une station à l'autre, la discipline de service des clients à chaque station, etc.

Dans la suite, nous appellerons "station" ou "hôte" tout équipement terminal capable de traiter et de stocker de l'information. Une station possède des ressources matérielles et logicielles, généralement en quantité limitée; elle constitue l'interface avec l'utilisateur (humain ou application) qui représentera souvent le générateur de trafic. Nous distinguerons donc, dès à présent, la station et le réseau auquel elle est connectée; les utilisateurs s'adressent au réseau qui doit paraître transparent et fournir les services suivants: les travaux demandés sont correctement exécutés, et les résultats sont fournis avec un temps de réponse acceptable. De même, au sein d'un hôte, nous distinguerons les composantes suivantes:

- l'utilisateur (humain ou application représentant le générateur de trafic),
- les protocoles (procédures mises en oeuvre pour réaliser les différents services proposés),
- l'accès au réseau (interface de l'hôte avec le réseau de communication).

a) l'utilisateur

Il correspond généralement à la notion d'application (transfert de fichier, messagerie, partage de ressources, ...); il est responsable de la génération du trafic qui constituera la caractérisation de la charge pour l'application étudiée. Les performances d'un système dépendent étroitement de la demande en ressources matérielles et logicielles (système et applications) et de la charge générée. Ainsi la caractérisation quantitative des demandes en ressources de la charge constitue une part importante des études menées sur l'évaluation des performances (aussi bien pour les études concernant la mesure que la modélisation). La caractérisation de la charge fournit les bases pour construire une charge synthétique afin

d'étudier un système à mesurer ou pour obtenir des valeurs de paramètres représentatives pour des modèles analytiques ou des simulations. En première approximation, une hypothèse généralement admise et très fréquemment utilisée pour des commodités de calcul consiste à représenter un flux d'informations générées par un processus de Poisson. Le processus de Poisson est d'un grand intérêt pratique car il peut représenter correctement de nombreux phénomènes physiques; en effet, un processus de Poisson peut être observé dans un système réel quand les arrivées sont produites par un grand nombre de sources indépendantes. Cela ne veut pas dire qu'on peut exclure d'autres types de flux d'arrivée. Par exemple, le flux d'arrivée cyclique (déterministe) apparaît souvent dans des communications temps réel, et des tableaux pré-configurés au lieu d'une loi probabiliste seront nécessaires.

b) les protocoles

La conception d'un réseau téléinformatique est particulièrement complexe et nécessite la réalisation d'un nombre élevé de fonctions. Le modèle de référence OSI proposé par l'ISO (International Standardisation Organisation) a conduit à une structuration en couches permettant de décomposer le système jusqu'à des éléments directement réalisables. La philosophie de cette architecture s'appuie sur les notions de service et de protocole. Chaque couche prend à sa charge la réalisation d'un service qu'elle essayera de fournir avec une qualité optimum (souvent fonction de la complexité du protocole); les relations entre les couches sont toujours du même type: la couche (N+1) s'appuie sur le service fourni par la couche N. La réalisation du service de la couche N est obtenue par l'intermédiaire d'un protocole de la couche N constitué principalement de deux parties: la structuration des Unités de Données et les Règles de dialogue entre entités de même niveau. Nous appelons (N)-SDU les Unités de Données du Service (N) qui sont remises par l'utilisateur des services (N) afin qu'elles soient transférées aux utilisateurs des services (N) de l'autre extrémité. De même, le terme (N)-PDU désigne les Unités de Données du Protocole (N) qui sont acheminées par le protocole de la couche (N). Dans le cadre de la modélisation des protocoles de communication, ce seront ces entités PDU/SDU qui seront manipulées dans les stations et qui pourront représenter la charge à traiter.

c) l'accès au réseau

Le module d'accès au réseau constitue une interface intéressante entre l'hôte et le réseau, et permet de construire une station de traitement avec ses protocoles en faisant abstraction du réseau de transmission utilisé et du composant physique d'accès. D'autre part, l'interface d'accès au réseau joue un rôle important dans la mesure où elle peut introduire des problèmes de capacité de transmission, voire de congestion.

On peut ainsi proposer un modèle général (Fig. 6) qui permet de réaliser l'évaluation des performances des différents protocoles d'accès multiples.

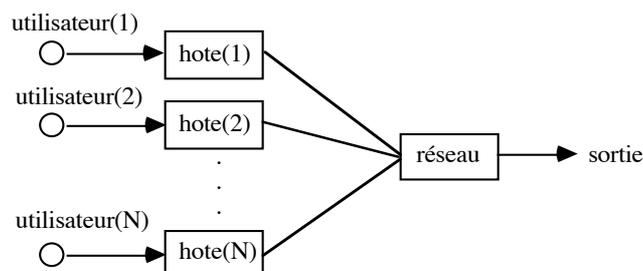


Fig. 9 Modèle général

Le modèle est composé de trois objets: "utilisateur(i)" qui représentent l'utilisateur en générant des flux d'information selon une certaine loi de probabilité, "hote(i)" qui représentent les station émettrices et réalisent les fonctions spécifiées par le protocole, et "réseau" qui représente le réseau, c.à.d. la voie de transmission. On note que les stations réceptrices n'interviennent pas dans le modèle car la réception n'influence pas l'analyse des performances du protocole à étudier.

Bibliographie

- [ALL 90] Arnold O. Allen, Probability, statistics and queueing theory with computer science applications, Second Edition, Academic press, Inc., 1990.
- [BAY 00] B. Baynat, Théorie des files d'attente: des chaîne de Markov aux réseaux à forme produit, Hermes, 2000.
- [BRA 83] G.W. Brams, Réseaux de Petri: théorie et pratique, Tome 1: Théorie et analyse; Tome 2: Modélisation et applications, Masson, Paris, 1983.
- [Chang00] Chang, C. S., Performance Guarantees in Communication Networks. New York: Springer-Verlag, 2000.
- [Cruz91a] Cruz, R. L., "A calculus for network delay, Part I", *IEEE Trans. on Information Theory*, 37(1):114-131, Jan. 1991.
- [Cruz91b] Cruz, R. L., "A Calculus for Network Delay, Part II: Network Analysis, *IEEE Transactions on Information Theory*, 37(1): 132-141, Jan. 1991.
- [DAV 92] R. David, H. Alla, Du Grafctet au réseaux de Petri, Hermes, Paris, 1992.
- [DEM 92] P. Dembinski, "Queueing network model for ESTELLE", proc. of 5th international Conf. on Formal Description Techniques (FORTE'92), Perros-Guirec, October 1992.
- [FIC 04] G. Fiche and G. Hébuterne, Communicating systems and networks: traffic and performance, Kogan Page Limited, 2004.
- [GRO 85] D. Gross and Carl M. Harris, Fundamentals of queueing theory, (2nd edition), John-Wiley, 1985.
- [HAY 84] J.F. Hayes, Modelling and analysis of computer communications networks, Plenum Press, 1984.
- [KLE 75] L. Kleinrock, Queueing systems, Vol. 1: Theory, John Wiley, 1975.
- [KLE 76] L. Kleinrock, Queueing systems, Vol. 2: Computer application, John Wiley, 1976.
- [Koubâa04] Koubâa, A., and Y.Q. Song, "Evaluation and improvement of response time bounds for real-time applications under non pre-emptive fixed priority scheduling", *International Journal of Production Research*, Vol.42, No.14, pp2899-2913, Taylor & Francis group, July 2004.
- [LeBoudec02] Le Boudec, J.Y. and P. Thiran, Network Calculus: A Theory of Deterministic Queueing Systems For The Internet, Online Version of the Book of Springer Verlag – LNCS 2050, July 2002. (http://lrcwww.epfl.ch/PS_files/NetCal.htm).
- [LIN 52] D.V. Lindley, "The theory of queues with a single server", *Proc. Camb. Phil. Soc.*, Vol. 48, pp.277-289, 1952.
- [Liu73] Liu,C.L. and J.W. Layland, "Scheduling Algorithms for Multiprogramming in Hard Real-Time Environment", *Journal of ACM*, Vol. 20 (1), pp46-61, 1973.
- [MAR 93] R. Marie, A. Jean-Marie, "Quantitative evaluation of discrete event systems: models, performances and techniques", *Proc. of PNPM'93*, 5th international workshop on Petri Nets and Performance Models, pp2-11, Toulouse, France, Oct. 19-22, 1993.
- [PUJ 99] G. Pujolle, Les Réseaux, Eyrolles, 1999.
- [ROB 90] T.G. Robertazzi, Computer networks and systems: queueing theory and performance evaluation, Springer-Verlag, 1990.
- [ROM 90] R. Rom, M. Sidi, Multiple access protocols, performance analysis, Springer-Verlag, 1990.
- [SER 00] C. Servin, Télécoms 1 : de la transmission à l'architecture des réseaux, 2^e édition, Dunod, 2000.
- [SER 99] C. Servin, Télécoms 2 : de l'ingénierie aux services, Dunod, 1999.
- [TAN 97] A. Tanenbaum, Réseaux, Prentice Hall et InterEditions, 3^e édition, 1997.

Annexe: TP Evaluation de performances par simulation en JMT

Exercice 1 :

On se propose d'étudier les performances d'un serveur supportant l'exécution des requêtes des n clients (terminaux). Soit un ordinateur (un serveur web par exemple) monoprocasseur ne mobilisant que la ressource unité centrale (cpu). Supposons que le temps de traitement d'une tâche (requête sur une page web) sur la cpu soit exponentiellement distribué et que sa moyenne soit de 500ms. L'intervalle de temps entre deux requêtes provenant d'un même terminal est supposé suivre également une loi exponentielle de moyenne égale à 20s (un clic de liens toutes les 20s dans un browser). Le modèle de file d'attente associé à ce système est une file M/M/1 de taux de service $\mu = 2$ par seconde et de taux d'arrivée $n \cdot (1/20)$ montré dans Fig.1.

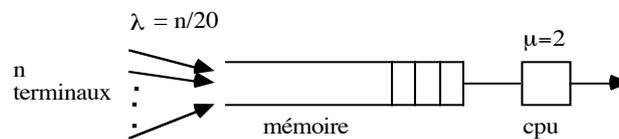


Fig.1 Modèle en file d'attente d'un ordinateur monoprocasseur

On s'intéresse au nombre de requêtes en attente N et le temps d'attente T .

D'après la théorie des files d'attente, on a:

$$p(k) = (1-\rho)\rho^k$$

$$E[N] = \rho/(1-\rho)$$

$$E[T] = E[N]/\lambda = 1/(\mu(1-\rho))$$

avec la condition de stationnarité: $\rho = \lambda/\mu = n/40 < 1$, i.e. $n < 40$.

Travail à faire :

- 1 Décrivez le modèle M/M/1 ci-dessus en JMT (JSIMgraph), puis simuler le modèle avec $\rho=0.9$. Commenter les résultats obtenus (temps moyen d'attente et nombre moyen de requête). Quels sont les autres indicateurs de performances qu'on peut observer ? (définir performance indices).
- 2 Comparer ces résultats avec ceux obtenus avec votre programme de simulateur M/M/1 réalisé en java.
- 3 Exporter le modèle vers JMVA (résolution analytique). Comparer les résultats analytiques avec ceux de la simulation.
- 4 Tracer la courbe de délais moyens en fonction de ρ (Utilisez « What-If Analysis » en examinant par exemple les cas $n=5$, $n=10$, ..., $n=35$).
- 5 Supposons que le temps de traitement d'une tâche en la cpu, au lieu d'être exponentiellement distribué, soit constant avec le même $\mu = 2$, on a une file M/D/1 avec : $E[T] = \mu/(1-\rho) - \mu\rho/(2(1-\rho))$. Proposez un modèle en JMT (JSIMgraph) en modifiant la station cpu, puis simuler le modèle pour tracer la courbe de délais moyens en fonction de ρ .
- 6 Supposons que l'ordinateur, au lieu d'être un monoprocasseur, soit un biprocasseur. Comparez M/M/1($\mu=2$) et M/M/2($\mu=1$) en modifiant le modèle de l'exercice 1.
- 7 Comparez les délais moyens pour les files M/M/1, M/M/2, et M/D/1.

Exercice 2 :

L'exemple présenté ici est un système informatique courant. En faisant une description et une analyse en détail d'un exemple simple, on essaie d'illustrer comment JMT peut être utilisé pour construire un simulateur et ensuite le résoudre.

A) Système à étudier

Il s'agit de modéliser le fonctionnement d'un ordinateur qui comporte:

- une unité centrale (cpu) qui traite les instructions des programmes
- deux périphériques nommés "disk_1" et "disk_2"
- un ensemble de terminaux connectés "terminal".

Le fonctionnement étudié est de type transactionnel; ceci signifie que l'utilisateur en émettant des commandes à partir de son terminal génère des tâches, chacune d'elles étant une suite de calculs en cpu et d'E/S disque. Une fois complètement exécutées, les tâches retournent vers le terminal (réponse à l'utilisateur).

On définit les tâches (les jobs) comme des clients, et cpu, disk_n, terminal comme des stations de service. Les clients circulent entre les stations selon une règle de routage. On a le schéma du système ci-dessous (Fig.2)

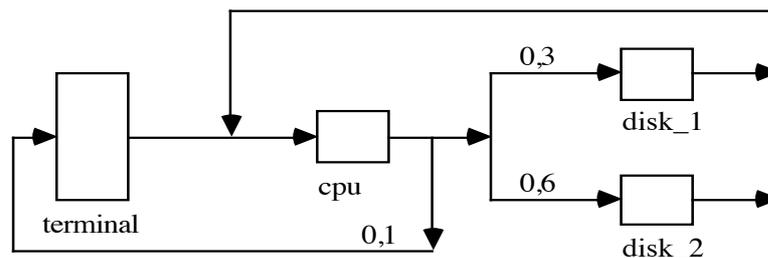


Fig.2 modèle d'un système d'ordinateur

On suppose que chaque station se compose d'un ou plusieurs serveurs et d'une file d'attente à capacité infinie. Un client entre dans une station, attend d'abord dans la file et puis est servi. Un client qui sort de la station "cpu" se dirige vers, soit disk_1 avec une probabilité de 0,3, soit disk_2 avec 0,6, soit terminal avec 0,1.

Pour la station "cpu", on suppose que le temps de service a une distribution hyper-exponentielle indépendante de l'état de la station avec $\lambda_1=10$, $\lambda_2=30$, et $p=0,5$. La discipline de la file d'attente est FIFO (First In First Out).

Pour une station "disk_n", le temps de service a une distribution exponentielle et la discipline est FIFO. On suppose que le temps moyen de traitement est 0,08s dans "disk_1", et 0,1s dans "disk_2".

Pour la station "terminal", on suppose qu'il y a autant de terminaux que d'utilisateurs. Donc, il n'y a pas d'attente pour un client entrant dans cette station. Un utilisateur devant un terminal envoie un job et attend la réponse. Entre la réception de cette réponse et l'envoi d'un nouveau job, il y a un temps de réflexion dont la

distribution est aussi exponentielle (e.g. temps entre deux clic des hyperliens). On suppose que ce temps est en moyenne 30s.

Le modèle en réseaux de files d'attente est montré ci-dessous (Fig.3):

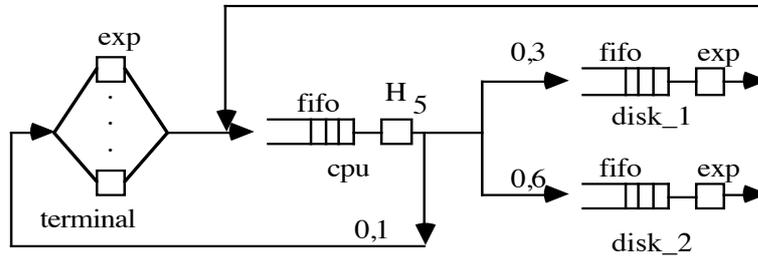


Fig.3 modèle en réseaux de files d'attente

B) Simuler en JMT

On s'intéresse aux indices de performances suivantes :

Define performance indices

Performance Indices
Define system performance indices to be collected and plotted by the simulation engine.

System Response Time
Add selected index

Performance Index	Class	Station/Region	Confidence Interval(0-1)	Max Relative Error(0-1)
Queue Length	--- All Classes ---	CPU	0.9	0.1
Queue Length	--- All Classes ---	Disk_1	0.9	0.1
Queue Length	--- All Classes ---	disk_2	0.9	0.1
Queue Length	--- All Classes ---	terminal	0.9	0.1
Throughput	--- All Classes ---	CPU	0.9	0.1
Throughput	--- All Classes ---	Disk_1	0.9	0.1
Throughput	--- All Classes ---	disk_2	0.9	0.1
Throughput	--- All Classes ---	terminal	0.9	0.1
Residence Time	--- All Classes ---	CPU	0.9	0.1
Residence Time	--- All Classes ---	Disk_1	0.9	0.1
Residence Time	--- All Classes ---	disk_2	0.9	0.1
Residence Time	--- All Classes ---	terminal	0.9	0.1
Utilization	--- All Classes ---	CPU	0.9	0.1
Utilization	--- All Classes ---	Disk_1	0.9	0.1
Utilization	--- All Classes ---	disk_2	0.9	0.1
Utilization	--- All Classes ---	terminal	0.9	0.1
Number of Customers	--- All Classes ---		0.9	0.1
System Throughput	--- All Classes ---		0.9	0.1
System Response Time	--- All Classes ---		0.9	0.1

Done

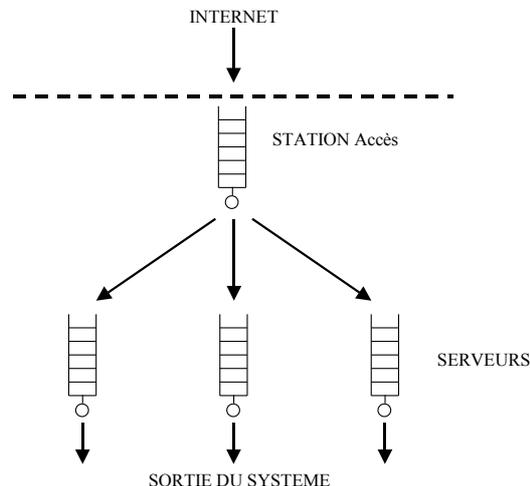
Travail à faire :

Simulez le modèle et commentez les résultats.

Exercice 3 : « conception d'une architecture web : étude des politiques de routage des requêtes sur les différents serveurs » (sujet d'après N. Navet)

Objectif : router les requêtes de façon à minimiser le temps de réponse des requêtes ainsi que la quantité de mémoire maximale (au niveau des serveurs) pour stocker les requêtes en attente de traitement.

Architecture du système :



Description du système :

La station Accès reçoit l'ensemble des requêtes en provenance de l'Internet (une station type source), vérifie leur validité (rôle de firewall) et effectue le routage vers les 3 serveurs qui seront chargés de traiter ces requêtes. Une fois le traitement effectué au niveau du serveur, les requêtes sortent du système (stations « sink »).

Le flux moyen d'arrivée des requêtes dans le système est de 50 requêtes par seconde, les temps inter-arrivées sont exponentiellement distribués. Le temps de traitement de chaque requête dans la station « Accès » est constant, il est de 2 millisecondes. Deux types de requêtes seront considérées, des requêtes « courtes » (ex : page web) et des requêtes « longues » (ex : interrogation d'une base de donnée). Il arrive 80% de requêtes courtes pour 20% de requêtes longues dans le système. Le temps moyen d'exécution au niveau des serveurs d'une requête courte est de 10 millisecondes alors que les requêtes longues nécessitent en moyenne 60 millisecondes. Les temps de service sont exponentiellement distribués.

Etape 1/3 :

1. Implémentez le modèle de ce système en JMT en considérant que la règle de routage est équiprobable entre les trois serveurs. Le flux d'arrivée des requêtes pourra être modélisé par une station de type SOURCE et c'est au niveau de cette source (station de référence) que les requêtes de deux classes « court » ou « long » seront associées.

1.1 Quel est le temps moyen de réponse d'un client dans le système ?

1.2 De façon empirique, dimensionner la taille des files d'attente au niveau des serveurs de telle façon qu'en moyenne un client ne soit pas rejeté plus d'une fois sur 100.

2. Implémentez le modèle de ce système en JMT en considérant que la règle de routage est Round-Robin (on alterne serveur1, serveur2, serveur3, serveur1...).

2.1 Quel est le temps moyen de réponse d'un client dans le système ?

2.2 De façon empirique, dimensionner la taille des files d'attente au niveau des serveurs de telle façon qu'en moyenne un client ne soit pas rejeté plus d'une fois sur 100.

2.3 Comparez les résultats des deux politiques de routage.

3. Imaginez une autre politique de routage, décrivez-la, implémentez la et comparez ses performances à celles de la politique Round-Robin. L'objectif est naturellement de trouver la politique la plus performante vis-à-vis du temps de réponse et des besoins en mémoire.

Etape 2/3 :

4. Nous nous plaçons maintenant dans une architecture informatique au sein de laquelle **les serveurs ont une puissance de traitement différente** : le premier serveur aura une puissance de 0.1, le second de 1 et le troisième de 10. Par rapport à l'étape précédente, les temps de service seront par conséquent divisés par la puissance de chaque serveur..

4.1 Supposons que les flux de requêtes dans le système sont inchangés. Implémenter une politique de routage probabiliste où la probabilité de routage est fonction de la puissance du serveur et comparer ses performances en termes de temps de réponse moyen avec la politique « Join the Shortest Load ». Comment expliquer ces résultats ?

4.2 Quel est le taux d'arrivées des requêtes si l'on veut maintenir inchangé la charge normalisée du système par rapport à l'étape 1 ?

4.3 Il existe plusieurs politiques de routage meilleures que « Join the Shortest Load » dans le cas de routage avec des serveurs de puissance différente. Proposer une ou plusieurs autres politiques et évaluer leurs performances en termes de temps de réponse moyen par rapport à « Join the Shortest Load » et au routage probabiliste.

Etape 3/3 :

Dans les deux premiers sujets, nous avons étudié les performances de diverses politiques de routage. La première classe de routage est celle des politiques « en boucle ouverte » c'est à dire des politiques qui n'utilisent pas d'informations sur l'état des serveurs. Dans cette classe de politiques nous trouvons le routage « probabiliste », le routage Round-Robin. Nous pouvons aussi imaginer d'autres politiques comme par exemple un routage dans lequel les files sont pré-affectées (par exemple : une file pour les requêtes longues et deux files pour les requêtes courtes avec un routage Round-Robin sur ses deux files).

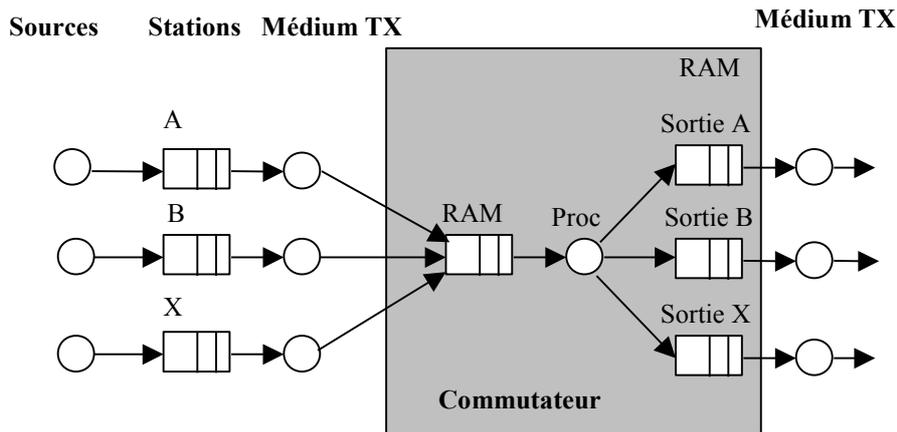
La seconde classe de politiques de routage est le routage « en boucle fermée » qui utilise des informations sur l'état des différents serveurs avant de décider. Parmi ces politiques, nous avons évoqué ou implanté « Join the Shortest Queue », « Join the Shortest Load » et une politique que nous pourrions appeler « Join the Shortest Response Time ».

5. **Nous vous demandons maintenant de considérer l'overhead** induit par la transmission d'information des serveurs vers la station chargée du routage dans le contexte du routage en boucle fermée. Une première politique est d'interroger les serveurs lorsqu'un client est prêt à être routé. Ceci nécessite un échange de messages (requêtes de la station qui effectue le routage – réponse du serveur). Quelle est la durée maximum en ms de cet échange de messages telle que la politique « Join the Shortest Response Time » soit toujours meilleure qu'un routage probabiliste où la probabilité est fonction de la puissance des serveurs ?

6. Jusqu'à maintenant nous avons étudié les politiques de routage, nous nous intéressons maintenant aux politiques d'ordonnancement sur les stations serveurs. Jusqu'à maintenant l'ordonnancement était FIFO, **nous nous proposons maintenant d'implanter un ordonnancement de type « Shortest Job First » (priorité à requêtes courtes)**. Quels sont les effets sur le temps de réponse moyen et sur la taille des files d'attente ? Quels problèmes voyez-vous à cette politique ?

Exercice 4 :

On souhaite étudier un réseau d'ordinateurs interconnectés par des nœuds à commutation de paquets (routeur IP dans l'Internet, commutateur de paquets dans un réseau X25, ATM ou Frame Relay, ...). Outre les débits des liaisons, les performances d'un commutateur jouent aussi un rôle très important dans le temps d'acheminement d'un paquet d'une station vers une autre. Dans cette étude on va d'abord s'intéresser aux performances d'un nœud de commutation, puis au temps d'acheminement d'une station vers une autre (de bout en bout). On peut représenter un tel nœud par le réseau de files d'attente suivant:



Le commutateur assure le trafic entre deux stations A et B et le trafic entre les autres stations ou nœuds représentés par la station X.

Le tableau ci-dessous (« traffic matrix » ou matrice de trafic) donne le nombre de paquets par seconde échangés entre les stations. Les arrivées sont supposées Poissonniennes:

	vers	A	B	X
de				
A			5	8
B		7		10
X		1	1	100

Pour faire l'exercice, prenons des valeurs suivantes des différents paramètres :

- Temps de commutation dans Proc = 5ms
- Débit de lignes A et B : 1200bit/s
- Débit de ligne X : 16000bit/s
- Longueur d'un paquet: 8 octets

1 - Construire le modèle (en JMT) représentant ce réseau de files d'attente et simuler pour savoir:

- Le délai moyen pour aller vers A et B
- Comment dimensionner les buffers de sortie de A et B (taille et probabilité de rejet)
- Modifier le modèle pour étudier l'influence du temps de commutation sur les performances (on prendra quelques valeurs comprises entre 1ms et 5ms)

2 - Si la taille des paquets n'est plus une constante mais une variable exponentiellement distribuée autour de la moyenne de 8 octets, ainsi que le temps de commutation (exponentielle avec la même moyenne que la question 1), modifier le modèle et refaire la simulation.

Comparer les résultats des questions 1 et 2. Lequel des deux est plus performant en terme de délai de traversé ?

3 - On considère maintenant que l'on a deux types de paquets P1 et P2. Le tableau ci-dessous donne le nombre de paquets par seconde échangés:

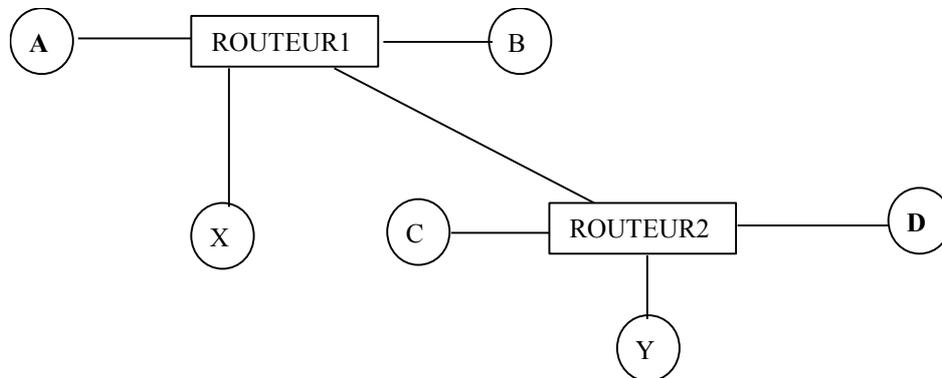
		A	B	X
A	P1		4	1
	P2		1	7
B	P1	5		2
	P2	2		8
X	P1	0,2	0,2	31,6
	P2	0,8	0,8	68,4

Modifier le modèle de la question 2 et le résoudre par simulation

4 - Mettre en place une politique permettant de privilégier le traitement des paquets P1. Comparer les résultats de 3 et 4.

5 - Supposons que les buffers de sortie sont de taille maximum de 10 paquets. Simuler 1, 2, 3 et 4 et comptabiliser le taux de rejets à cause de débordement des buffers.

6 - Evaluer par simulation le temps moyen pour qu'un paquet émis par A arrive dans la station D.



On suppose que C et D génèrent le même trafic que A et B. La ligne entre les deux routeurs ainsi que les lignes C et D sont de 1200bit/s et la ligne Y 16000 bit/s. Les trafics sont résumés dans le tableau suivant:

	A	B	X	Routeur2
A		5	8	1
B	7		10	1
X	1	1	90	5
Routeur2	1	1	5	

On suppose que les trafics sur le routeur2 sont symétriques aux ceux du routeur1. Son tableau ne sera pas répété ici.