Cours de Modélisation et d'Evaluation de Performance

Auteur: PHAM Cong-Duc

L'outil file d'attente

- **On va essayer de lever les contraintes de l'analyse opérationnelle avec les files d'attente.**
- **On peut représenter un système par un ensemble de files d'attente, chaque file modélisant une ressource par exemple.**
- **Une file d'attente est définit par :**
  - La suite des instants d'arrivées des clients
  - La suite des temps de service des clients
  - La disciple de service qui donne l'ordre dans lequel seront servi les clients.
  - La capacité de la file
  - Le nombre de serveurs
  - Population totale de clients (rare)

Auteur: PHAM Cong-Duc

Notation de Kendall

- **Une file d'attente se note :**

  A/S/C (DS/K/L)
  A/S/C/K/L/ (DS)
  A/S/C/K/L/DS

- **Avec :**

A : processus d'arrivée
S : processus de sortie
C : nombre de serveurs
K : capacité maximale de la file
L : population de clients
DS : discipline de service

- **Symbole pour les arrivées et les services**
  - M : loi exponentielle (Markovienne)
  - D : loi constante
  - E_k : loi Erlang-k
  - H_k : loi hyper-exponentielle ordre k
  - GI : loi générale indépendante
  - G : loi générale

- **Symbole pour les discipline de service**
  - FCFS : First Come First Serve (Preempt)
  - LCFS : Last Come First Serve (Preempt)
  - QUANTUM : Round Robin
  - PS : Processor Sharing
  - RANDOM
  - PRIORITY

Auteur: PHAM Cong-Duc

Quelques quantités intéressantes

- **On définit l'intensité du traffic :**

$$\rho = \frac{Temps\ \ moyen\ \ de\ \ service}{Temps\ \ moyen\ \ entre\ \ 2\ \ arrivées} = \lambda S$$

- **Le taux d'occupation du serveur :**

$$U = Taux\ \ d'\ arrivée\ \ effective.Temps\ \ de\ \ service = \lambda'.S$$

- **Rappelons que la loi de Little reste valable quelque soit la loi d'arrivée, loi de priorité, temps de service, à la condition qu'il existe un régime stationnaire limite.**

$$L=\Lambda.R$$

Auteur: PHAM Cong-Duc

Chaîne de Markov

- **Une chaîne de Markov est un système qui peut prendre différents états parmi un ensemble d'états.**
- **Un processus de Markov suppose que la probabilité de passer d'un état à un autre ne dépend que de l'état courant. Il n'y a pas mémoire du passé.**
- **Deux types de chaînes sont à considérer :**
  - Chaîne à temps discret où les transitions ne peuvent se produire qu'à des instants précis.
  - Chaîne à temps continu où les transitions peuvent se produire à tout instant. Cependant, du fait de la propriété sans mémoire requise, le temps de séjour dans un état est distribué exponentiellement.
- **Une chaîne de Markov est dite *homogène* si les probabilités de transition ne dépendent pas du temps, *irréductible* si tout état est accessible à partir de n'importe quel autre état.**

Chaîne de Markov à temps discret

- **Propriété sans mémoire**

$$P(X_n = i_n \mid X_1 = i_1, X_2 = i_2, \ldots, X_{n-1} = i_{n-1}) = P(X_n = i_n \mid X_{n-1} = i_{n-1})$$

- **Equation de Chapman-Kolmogorov**

$$P_{ij}(m,n) = \sum_k P_{ik}(m,q) P_{kj}(q,n)$$

$$avec \quad P_{ij}(m,n) = P(X_n = j \mid X_m = i)$$

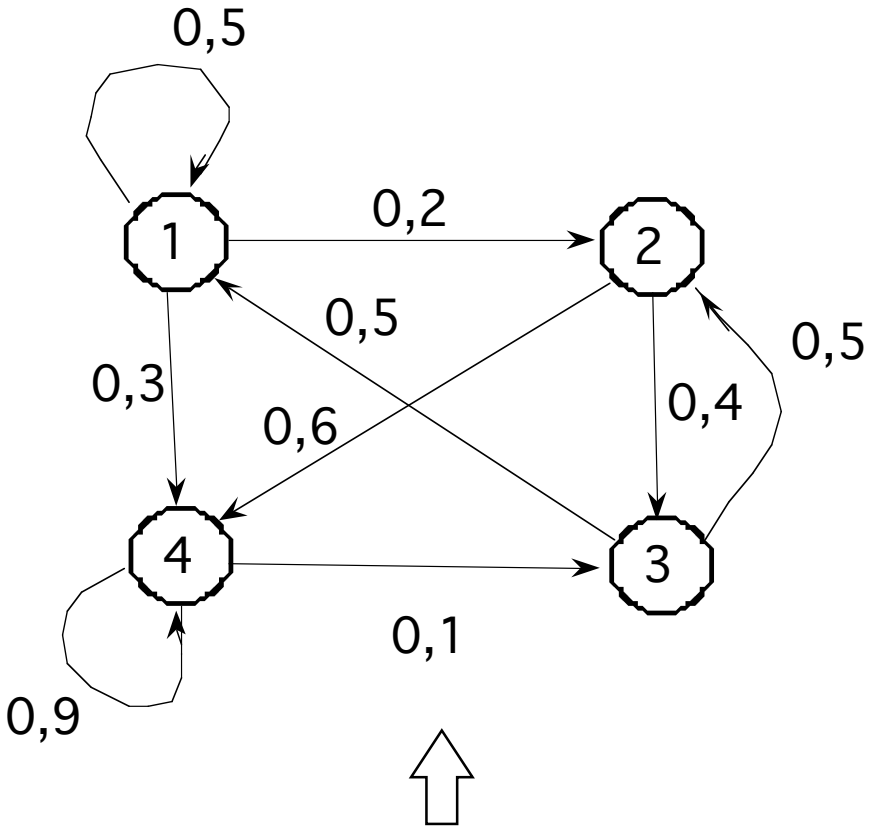
- **A l'état d'équilibre, on peut trouver le vecteur**

$$\pi = (P_0, P_1, \ldots, P_n)$$

$$par \ \ la \ \ résolution \ \ de \quad \boxed{\pi = \pi P}$$

Auteur: PHAM Cong-Duc

Exemple de chaîne à temps discret



Chaîne de Markov

Matrice des probabilités de transitions

$$P = \begin{pmatrix} 0,5 & 0,2 & 0 & 0,3 \\ 0 & 0 & 0,4 & 0,6 \\ 0,5 & 0,5 & 0 & 0 \\ 0 & 0 & 0,1 & 0,9 \end{pmatrix}$$

$$avec \quad \sum_i P_{ij} = 1$$

Auteur: PHAM Cong-Duc

Chaîne de Markov à temps continu

- **Propriété sans mémoire**

$$P(X_{t_n} = i_n \,|\, X_{t_1} = i_1, X_{t2} = i_2, \ldots, X_{t_{n-1}} = i_{n-1}) = P(X_{t_n} = i_n \,|\, X_{t_{n-1}} = i_{n-1})$$

- **Equation de Chapman-Kolmogorov**

$$P_{ij}(s,t) = \sum_k P_{ik}(s,u) P_{kj}(u,t)$$

$$avec \quad P_{ij}(s,t) = P(X_t = j \,|\, X_s = i)$$

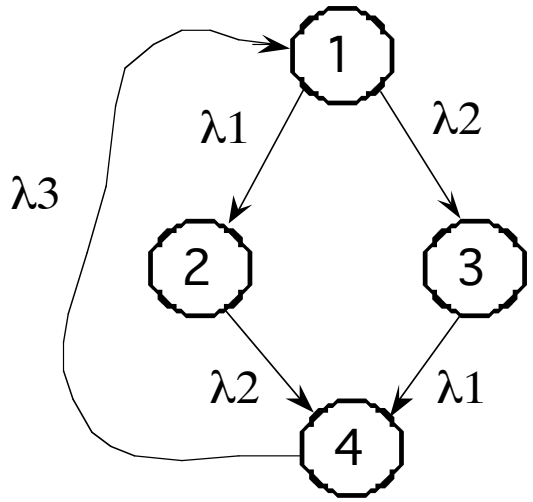
- **A l'état d'équilibre, on peut trouver le vecteur**

$$\pi = (P_0, P_1, \ldots, P_n)$$

par la résolution de $0 = \pi Q$

Auteur: PHAM Cong-Duc

Exemple de chaîne à temps continu



⇐ **Chaîne de Markov**

Matrice des taux de transitions ⇒

$$Q = \begin{pmatrix} -(\lambda_1+\lambda_2) & \lambda_1 & \lambda_2 & 0 \\ 0 & -\lambda_2 & 0 & \lambda_2 \\ 0 & 0 & -\lambda_1 & \lambda_1 \\ \lambda_3 & 0 & 0 & -\lambda_3 \end{pmatrix}$$

$$avec \quad \sum_i Q_{ij} = 0$$

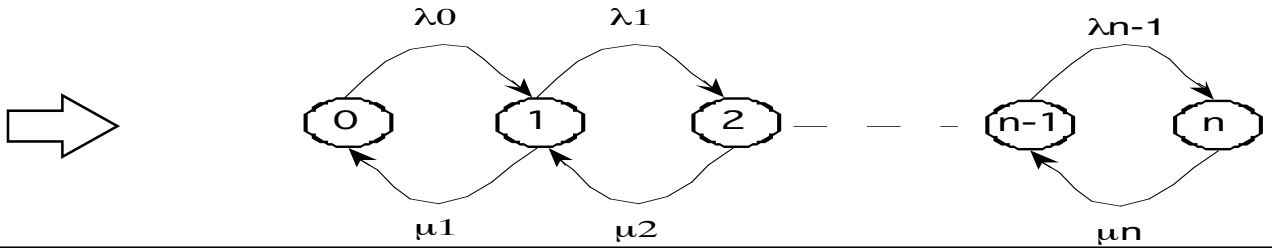
Processus de naissance et de mort

- **C'est un cas particulier de chaîne de Markov où seules les transitions d'un état à un état voisin sont permises.**
- **On s'intéresse au cas continu avec des taux de transitions.**
- **C'est le point de départ de la théorie des files d'attente.**

- **On introduit les données suivantes :**

  λ_k = taux de naissances quand la population est k

  μ_k = taux de morts quand la population est k



Processus de naissance et de mort

- **Résultats**

$$Q = \begin{pmatrix} -\lambda_0 & \lambda_0 & 0 & 0 & 0 & . \\ \mu_1 & -(\lambda_1+\mu_1) & \lambda_1 & 0 & 0 & . \\ 0 & \mu_2 & -(\lambda_2+\mu_2) & \lambda_2 & 0 & . \\ 0 & 0 & \mu_3 & -(\lambda_3+\mu_3) & \lambda_3 & . \\ 0 & 0 & 0 & 0 & 0 & . \\ . & . & . & . & . & . \end{pmatrix}$$

$$\begin{cases} P_n = \prod_{k=0}^{n-1} \dfrac{\lambda_k}{\mu_{k+1}} P_0 \\ P_0 = \left[1 + \sum_{k=1}^{\infty} \left(\prod_{i=0}^{k-1} \dfrac{\lambda_i}{\mu_{i+1}} \right) \right]^{-1} \end{cases}$$

Auteur: PHAM Cong-Duc

Processus de naissance pure

- **Cas où** $\begin{cases} \lambda_k = \lambda \\ \mu_k = 0 \end{cases}$

⇨ **On obtient la distribution de Poisson**

$$P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}, \qquad k \geq 0, t \geq 0$$

$$Avec \;\; P_k(t) = P[k \;\; naissances \;\; pendant \;\; [0,t]]$$

✎ **Les processus de Poisson se rencontrent souvent dans les phénomènes physiques et naturels. Généralement, c'est le cas lorque les arrivées sont produites par un grand nombre de sources indépendantes (entrée d'un calculateur, central téléphonique...)**

Processus de Poisson et loi exponentielle

- **A partir de la distribution de Poisson, on peut en déduire la fonction de répartition (PDF) et la fonction de densité (pdf).**

$$PDF = A(t) = 1 - e^{-\lambda t}$$

$$pdf = a(t) = \lambda e^{-\lambda t}$$

- **C'est la loi exponentielle. Si le nombre d'arrivées est un processus de Poisson, le temps inter-arrivée suit la loi exponentielle.**

- **La loi exponentielle est dite sans mémoire, et c'est la seule. Par exemple le temps de séjour ne dépend pas du tout du temps de séjour passé.**

Rappel des principales lois

- **Processus de Poisson**

$$P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}, \qquad k \geq 0, t \geq 0$$

$$Avec \;\; P_k(t) = P[k \;\; naissances \;\; pendant \;\; [0,t]]$$

$$E[X] = Var[X] = \frac{1}{\lambda}$$

- **Loi exponentielle**

$$PDF = A(t) = 1 - e^{-\lambda t}$$

$$pdf = a(t) = \lambda e^{-\lambda t}$$

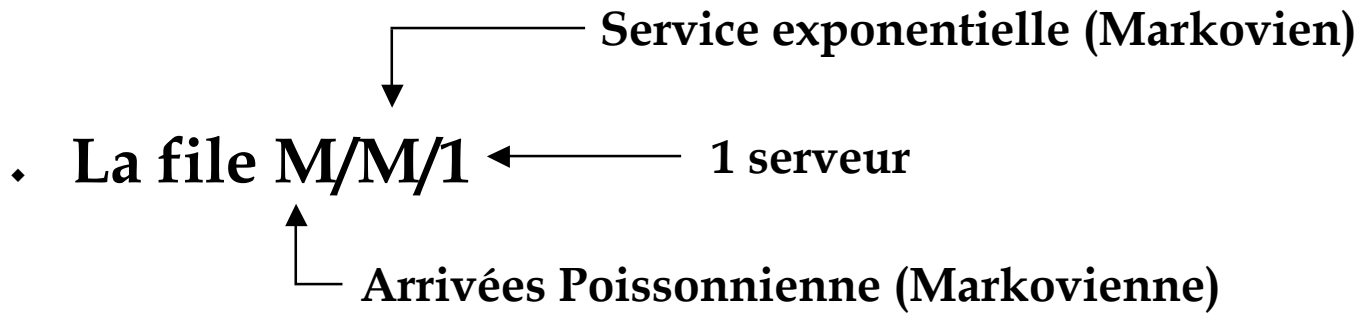
$$E[X] = \frac{1}{\lambda}$$

$$Var[X] = \frac{1}{\lambda^2}$$

$$CCV = 1$$

Auteur: PHAM Cong-Duc

File élémentaire M/M/1



- C'est un processus de naissance et de mort avec

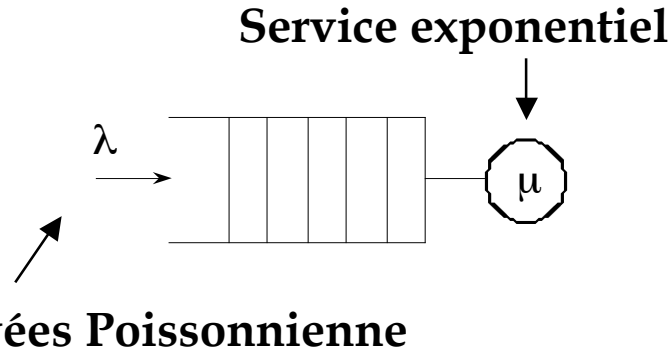
$$\begin{cases} \lambda_k = \lambda \\ \mu_k = \mu \end{cases}$$

- On peut assimiler λ et μ à des débits d'entrée et de sortie.

File élémentaire M/M/1

- **Résultats**

$$\begin{cases} \lambda_k = \lambda \\ \mu_k = \mu \end{cases} \qquad \rho = \frac{\lambda}{\mu}$$



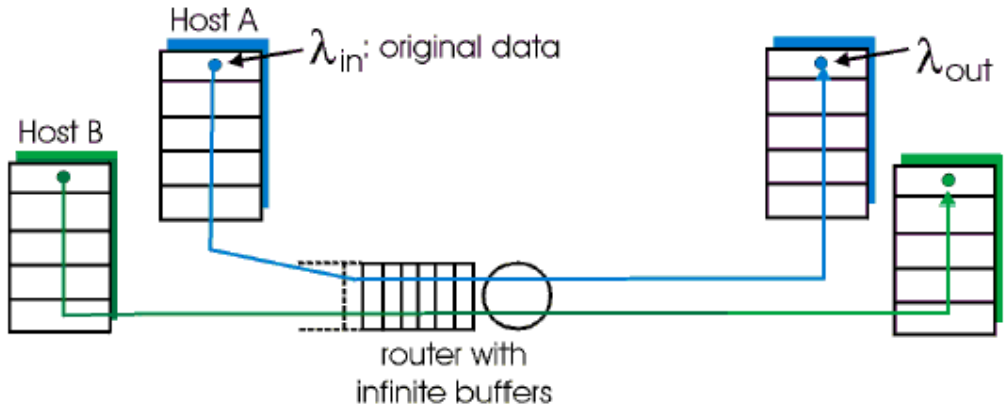
$$P_k = \rho^k P_0$$
$$P_0 = 1 - \rho$$
$$U = 1 - P_0 = \rho$$
$$P[n \geq N] = \rho^N$$

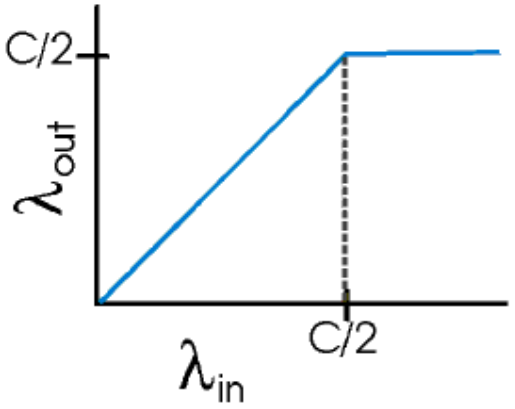
$$N = L = \sum_{k=0}^{\infty} k P_k = \frac{\rho}{1-\rho}$$
$$R = \frac{1}{\mu(1-\rho)}$$
$$S = \frac{1}{\mu}$$

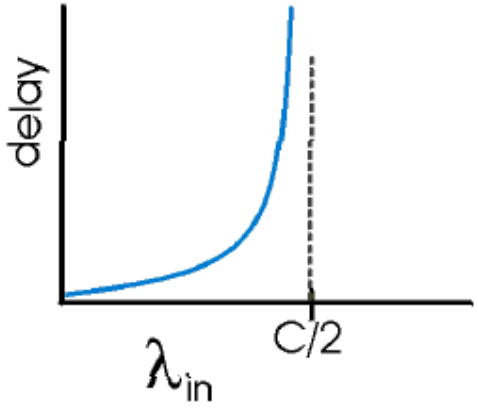
Exemple

Causes/coûts de la congestion: scenario 1

- Deux émetteurs, deux récepteurs
- un routeur, mémoire infinie
- Pas de retransmission

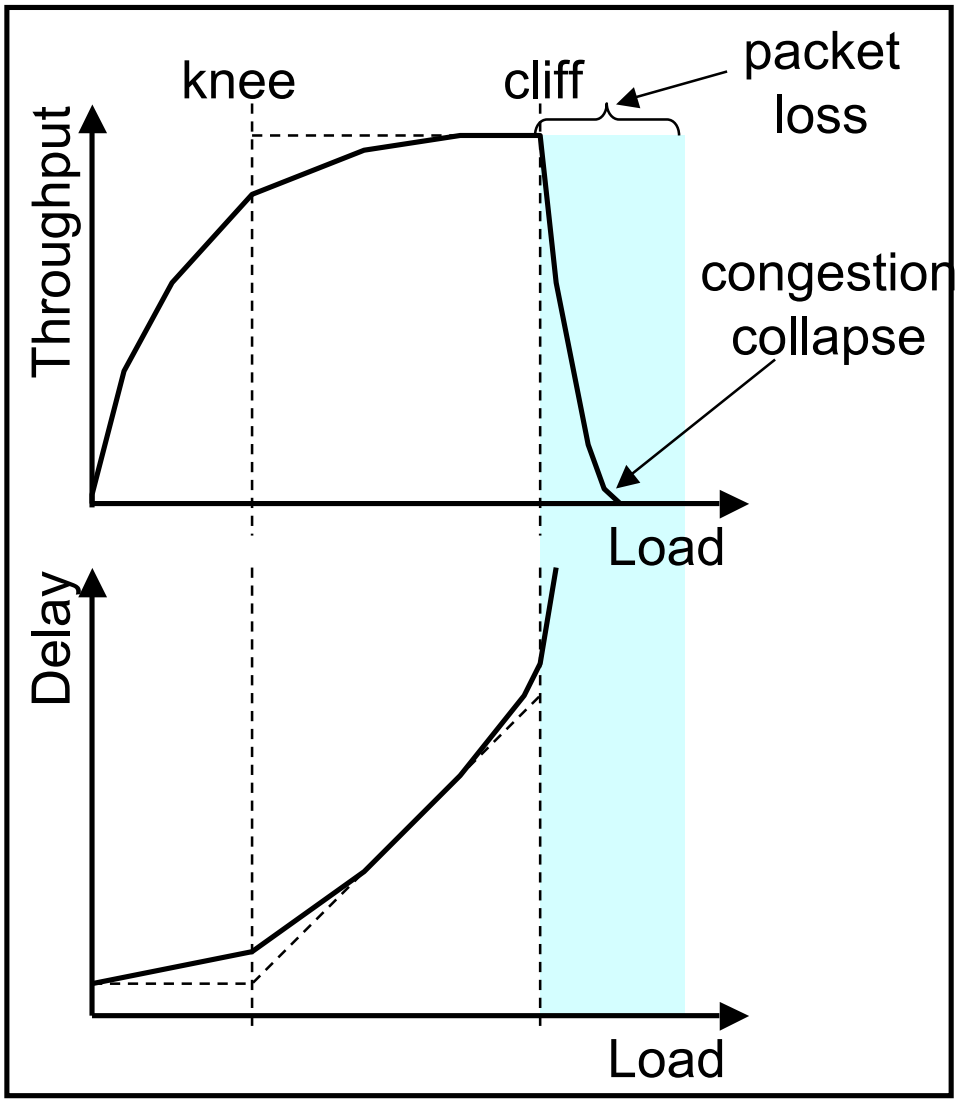






Exemple - 2

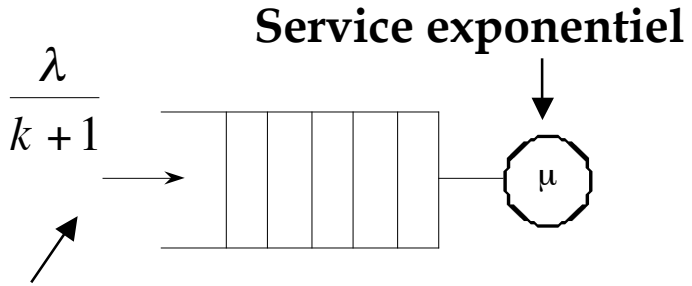
- **knee – point after which**
  - throughput increases very slowly
  - delay increases fast
- **cliff – point after which**
  - throughput starts to decrease very fast to zero (congestion collapse)
  - delay approaches infinity
  - delay = 1/(1 – U)



Arrivées découragées

- **Résultats**

$$\begin{cases} \lambda_k = \dfrac{\lambda}{k+1} \\ \mu_k = \mu \end{cases} \qquad \rho = \frac{\lambda}{\mu}$$



$$P_k = \frac{\rho^k}{k!} P_0$$

$$P_0 = \frac{1}{e^{\rho}}$$

$$U = 1 - P_0$$

$$P[n > m] = 1 - P_0 \sum_0^{N-1} \frac{\rho^k}{k!}$$

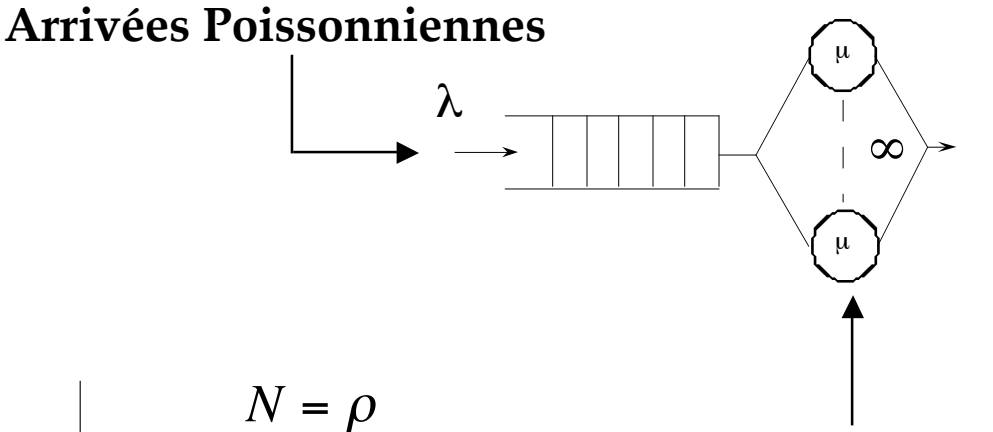
$$N = \rho$$

$$R = \frac{\rho}{\mu(1 - P_0)}$$

La file M/M/∞

- **Résultats**

$$\begin{cases}\lambda_k = \lambda \\ \mu_k = k\mu\end{cases} \qquad \rho = \frac{\lambda}{\mu}$$



$$P_k = \frac{\rho^k}{k!} P_0$$

$$P_0 = \frac{1}{e^{\rho}}$$

$$U = 1 - P_0$$

$$P[n > m] = 1 - P_0 \sum_0^{N-1} \frac{\rho^k}{k!}$$

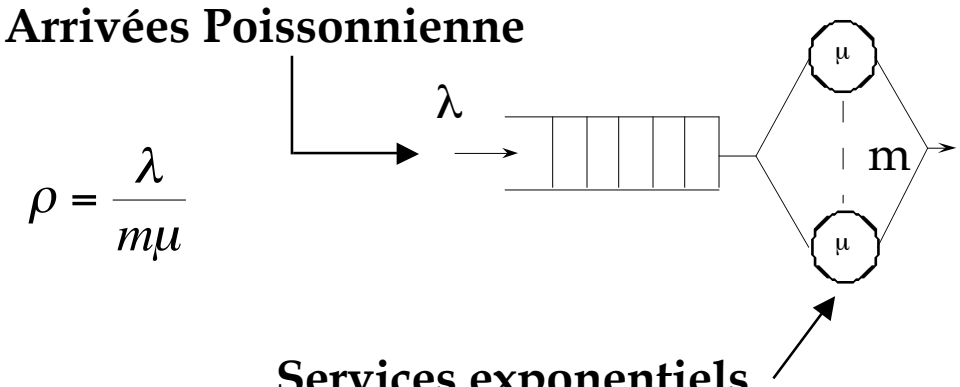
$$N = \rho$$

$$R = \frac{1}{\mu}$$

- **Même résultat que pour les arrivées découragées ! Sauf que l'expression de R est plus simple.**

La file M/M/m

- **Résultats**



$$\begin{cases} \lambda_k = \lambda \\ \mu_k = \begin{cases} k\mu & 1 \le k \le m \\ m\mu & k \ge m \end{cases} \end{cases} \qquad \rho = \frac{\lambda}{m\mu}$$

$$P_k = \begin{cases} \frac{1}{k!}(m\rho)^k P_0 & k \le m \\ \frac{1}{m!} m^m \rho^k P_0 & k \ge m \end{cases} \qquad P_0 = \frac{1}{\sum_0^{m-1} \frac{1}{k!}(m\rho)^k + \frac{(m\rho)^m}{m!(1-\rho)}}$$

$$U = 1 - P_0$$

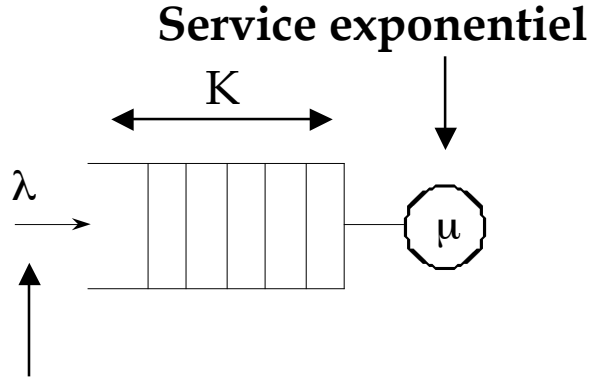
$$P[n > m] = \frac{1}{P_0} \frac{(m\rho)^m}{m!(1-\rho)}$$

La file M/M/1/K

- **Résultats**

$$\begin{cases} \lambda_k = \begin{cases} \lambda & k < K \\ 0 & k \geq K \end{cases} \\ \mu_k = \mu \qquad 1 \leq k \leq K \end{cases}$$

$$\rho = \frac{\lambda}{m\mu}$$



$$P_k = \rho^k P_0$$

$$P_0 = \frac{1-\rho}{(1-\rho^{K+1})}$$

$$U = 1 - P_0$$

P_K = Probabilité de rejet

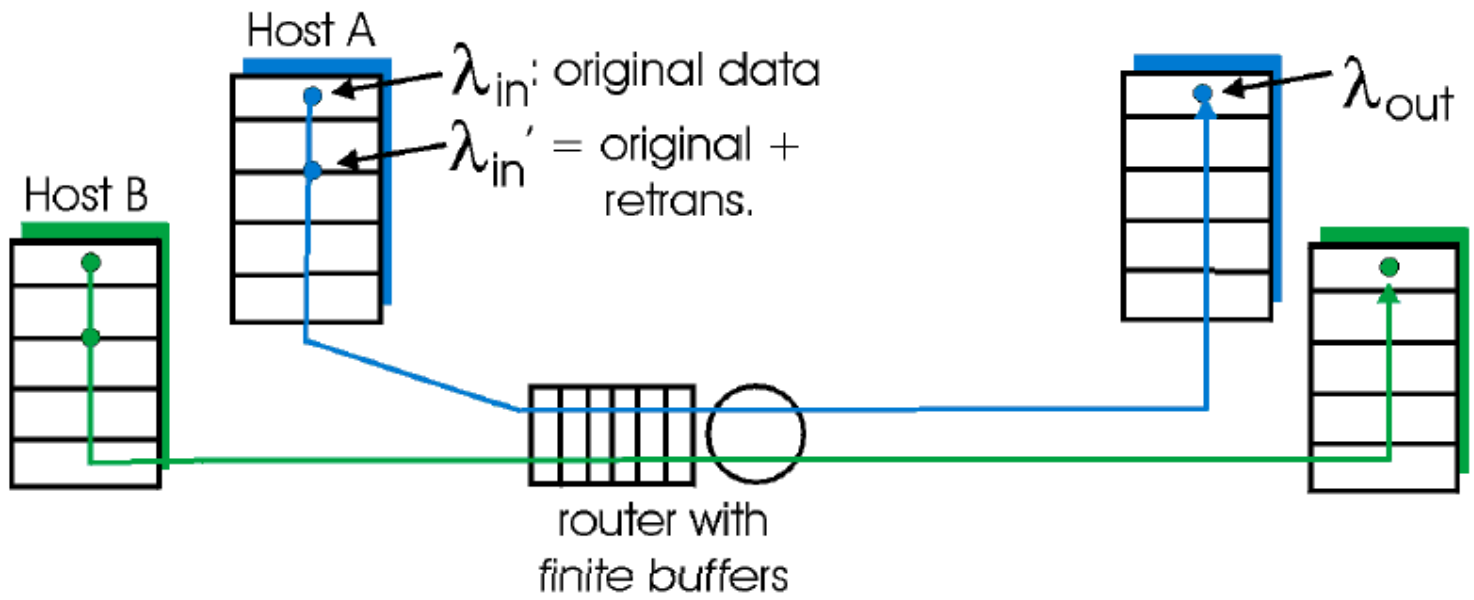
$$N = \frac{\rho}{1-\rho} - (K+1)\frac{\rho^{K+1}}{1-\rho^{K+1}}$$

$$R = \frac{N}{\lambda(1-P_K)}$$

Exemple

Causes/coûts de la congestion: scenario 2

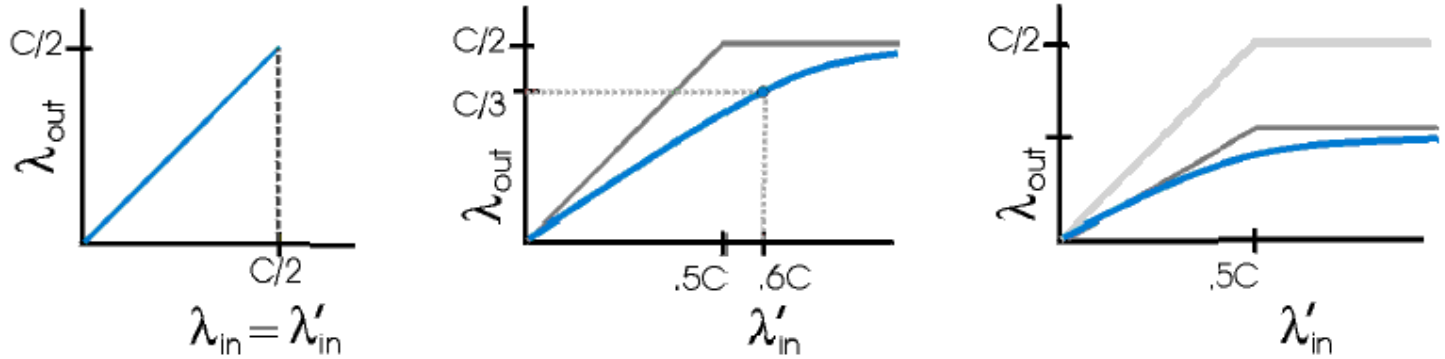
- Un routeur, *mémoire finie*
- L'émetteur retransmet les paquets perdus



Exemple - suite

Causes/coûts de la congestion: scenario 2

- $\lambda_{in} = \lambda_{out}$ (goodput)
- Si la retransmission est parfaite : $\lambda'_{in} > \lambda_{out}$
- La retransmission de paquet non perdu rend λ'_{in} ? que dans le cas parfait



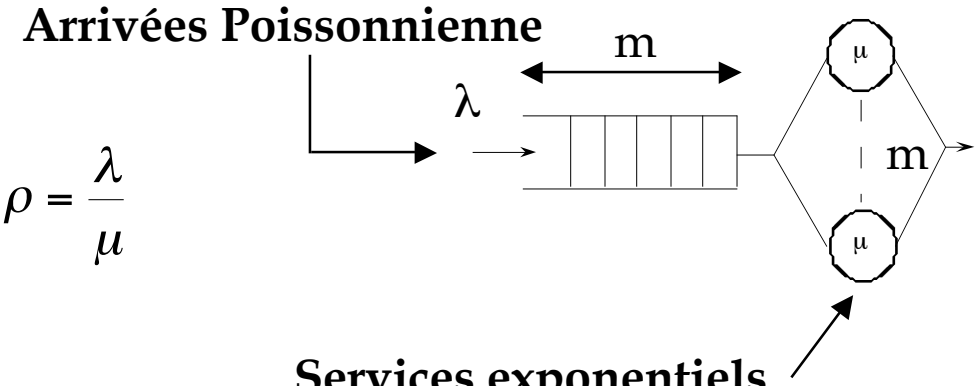
"coûts" de la congestion:

- Plus de travail (retrans) pour un même débit utile ("goodput")
- Retransmissions redondantes

La file M/M/m/m

- **Résultats**

$$\begin{cases} \lambda_k = \begin{cases} \lambda & k < m \\ 0 & k \geq m \end{cases} \\ \mu_k = k\mu \qquad 1 \leq k \leq m \end{cases} \qquad \rho = \frac{\lambda}{\mu}$$



$$P_k = \frac{\rho^k}{k!} P_0$$

$$P_0 = \frac{1}{\sum_{k=0}^{m} \frac{\rho^k}{k!}}$$

$$U = 1 - P_0$$

P_m = Probabilité de rejet

$$N = \rho\left(1 - \frac{\rho^m}{m!} P_0\right)$$

$$R = \frac{1}{\mu} \qquad \text{(pas d' attente)}$$

Auteur: PHAM Cong-Duc

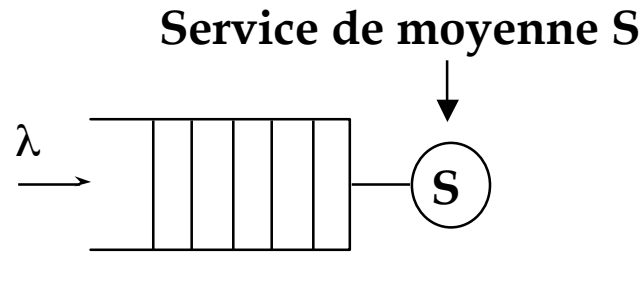
La file M/G/1

- **Des temps de service exponentiels sont facilement manipulables mais sont difficilement applicables dans des cas concrets.**
- **La file M/G/1 permet l'utilisation de temps de service distribués selon une loi générale.**
- **Ce n'est plus un processus de naissance et de mort.**
  - Par exemple si la durée du service est la constante S, et si le service du client précédent est commencé depuis une durée H, au moment de l'arrivée d'un client, on sait que ce dernier devra attendre pendant S-H. Ce qui se passera dans le futur dépend donc du passé par l'intermédiaire de H.

☞ **La file M/G/1 permet de lever les contraintes des temps de service exponentiels en généralisant la notion de service.**

La file M/G/1

- **Résultats**



Soit h(t) le pdf du temps de service,

$$b_n = \int_0^{\infty} t^n h(t) dt$$

On démontre alors que $\begin{cases} S = b_1 \\ Var[S] = b_2 - b_1^2 \\ \rho = \lambda S \end{cases}$

En utilisant $c^2 = CCV = \frac{Var[S]}{S^2} = \lambda^2 \frac{b_2}{\rho^2} - 1$ ⇨

$$N = \rho\left(1 + \frac{\rho}{1-\rho}\left(\frac{1+c^2}{2}\right)\right)$$

$$\frac{R}{S} = 1 + \frac{\rho}{1-\rho}\left(\frac{1+c^2}{2}\right)$$

$$\frac{W}{S} = \frac{\rho}{1-\rho}\left(\frac{1+c^2}{2}\right)$$

La file M/G/1/PS

- **La politique PS (Processor Sharing) est une allocation par quantum quand q->0 avec généralement q<<S.**
- **On démontre que :**

$$\rho = \lambda S$$

$$P_k = \rho^k P_0 \qquad N = \frac{\rho}{1-\rho}$$

$$P_0 = 1 - \rho$$

$$U = 1 - P_0 = \rho$$

$$P[n > N] = \rho^N \qquad R = \frac{S}{(1-\rho)}$$

- **On retrouve les mêmes résultats que pour la file M/M/1. La politique PS tend donc à effacer l'influence de la variance.**

La file M/G/1 Multi-classe

- **Dans le modèle multi-classes, plusieurs classes de clients avec des services différents peuvent coexister.**
- **Indexons les classes par i, on a :**

$$\begin{cases} U_i = \rho_i = \lambda_i S_i \\ U = \rho = \sum \rho_i \\ S = \sum_{i=1}^{C} \frac{\lambda_i}{\lambda} S_i \end{cases} \qquad c_s^2 = \frac{\sum_{i=1}^{C} \left(\rho_i S_i \left(\frac{1 + c_i^2}{2} \right) \right)}{\rho S} - 1$$

On définit par $\rho^{\{i\}} = \sum_{j=1}^{i} \rho_j$ la probabilité que le serveur

soit occupé par des clients de classe $1, 2, \ldots, i$

File M/G/1 Multi-classes (suite)

$$W_i = S_i \frac{\rho}{1-\rho}$$ si la priorité est PS (donc indépendant des c_i^2)

$$Wi = \frac{1}{1-\rho} \sum_{j=1}^{C} \rho_j S_j \left(\frac{1+c_j^2}{2} \right)$$ si la priorité est FCFS (indépendant des classes)

$$W_i = \frac{\sum_{j=1}^{C} \rho_j S_j \left(\frac{1+c_j^2}{2} \right)}{(1-\rho^{\{i-1\}})(1-\rho^{\{i\}})}$$ s' il y a des classes de priorité décroissante de 1 à C sans préemption

$$W_i = S_i \frac{\rho^{\{i-1\}}}{1-\rho^{\{i-1\}}} + \frac{\sum_{j=1}^{C} \rho_j S_j \left(\frac{1+c_j^2}{2} \right)}{(1-\rho^{\{i-1\}})(1-\rho^{\{i\}})}$$ s' il y a des classes de priorité décroissante de 1 à C avec préemption et reprise.

File M/G/1 Multi-classes (suite)

- **On a alors :**

$$W = \sum_{i=1}^{C} \frac{\lambda_i}{\lambda} W_i$$

$$\begin{cases} R_i = S_i + W_i \\ R = S + W = \sum_{i=1}^{C} \frac{\lambda_i}{\lambda} R_i \end{cases}$$

$$\begin{cases} N_i = \lambda_i R_i \\ N = \lambda R = \sum_{i=1}^{C} N_i \end{cases}$$