Formalismes Mathématiques des Systèmes

Niveau Master 1
Option : RN- RS

Prof. BOUALLOUCHE Louiza

Département Informatique / Université de Béjaïa

Programme

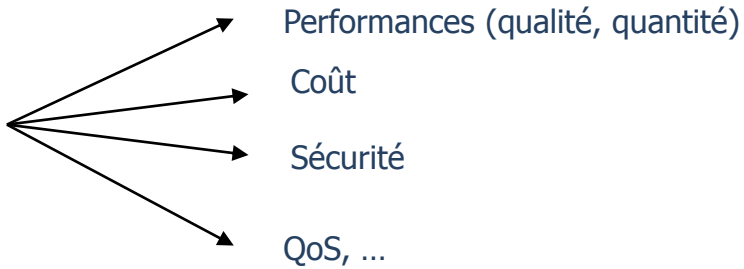
1. Introduction à l'évaluation de performances des systèmes et réseaux

2. Les Chaînes de Markov à Temps Discrets

3. Modélisation des systèmes par files d'attente

1. Introduction à l'évaluation de performances des systèmes et réseaux

Pourquoi l'Evaluation de Performances?



Exemples.

1. Système à Mémoire Virtuelle (performance en termes de qualité et quantité). On souhaite connaître le Taux d'Utilisation.

2. Politique de remplacement de caches dans les serveurs web. On souhaite connaître le Taux de Succès des requêtes de documents (Hit Rate).

3. Mécanisme d'équilibrage de Charge dans les systèmes de serveurs web. On souhaite connaître le temps de réponse ou de slowdown.

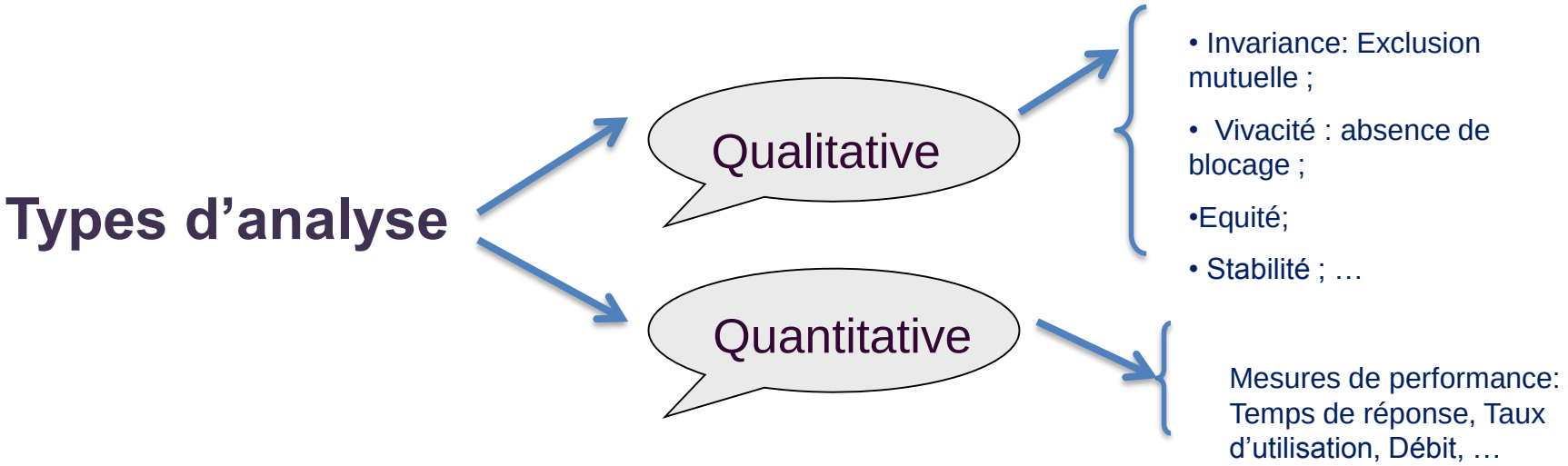
L'Evaluation de Performances intervient à deux niveaux

, …

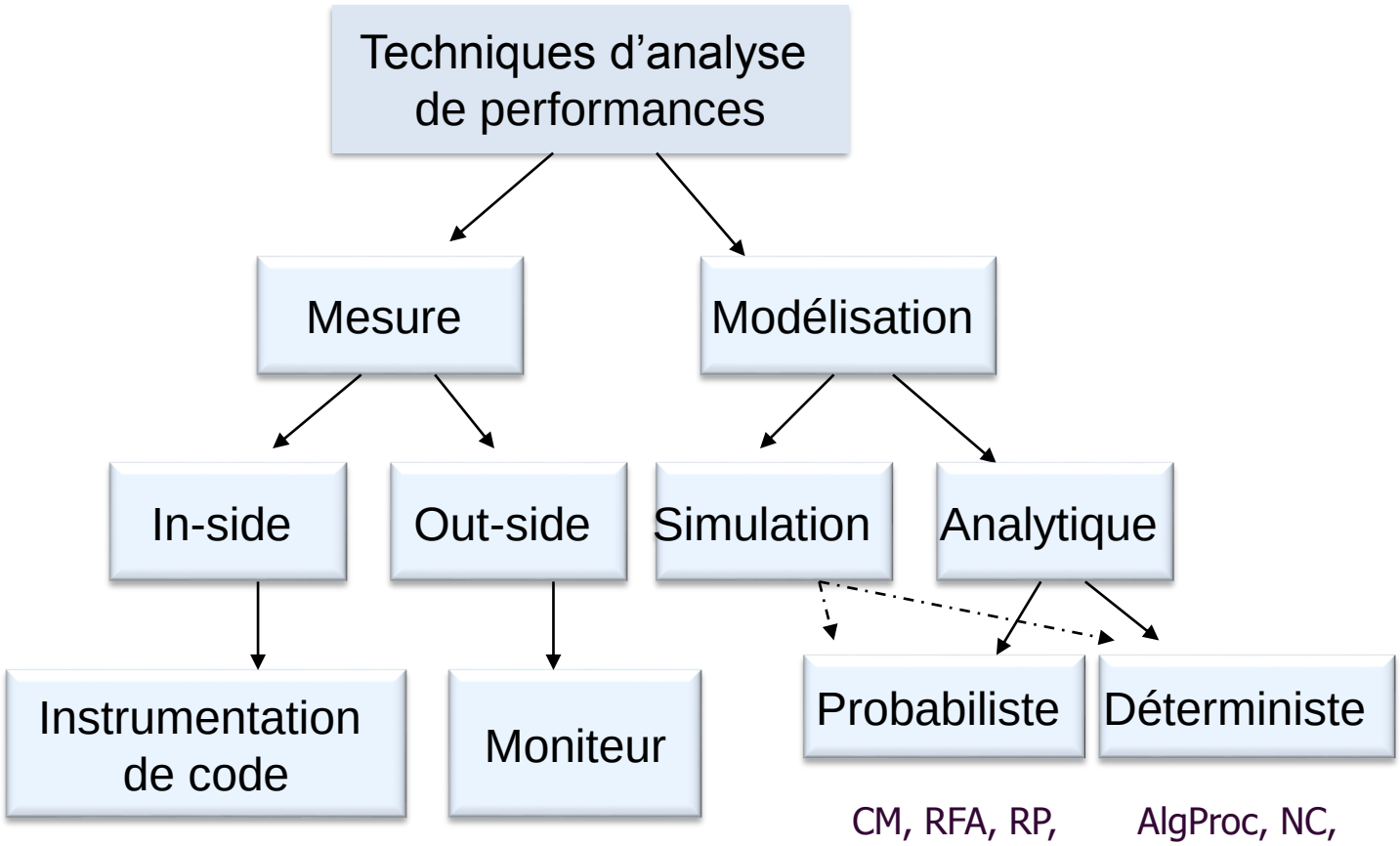
Conception: Le système n'existe pas.
Objectif: Concevoir un système en respectant un cahier des charges.
Ex. pour un Dimensionnement d'un RC: quel est débit à satisfaire?

Exploitation: Le système existe. Il s'agit de lui construire un nouveau modèle répondant aux nouvelles exigences.
Objectif: amélioration des ses performances.

Evaluation de performances des systèmes informatiques



Classification des techniques d'analyse des systèmes



Approches probabilistes d'EP des systèmes

***Principe de ces techniques** :*

Les techniques de modélisation sont basées sur le principe consistant à définir successivement

- les états du système
- les transitions entre chaque état (dynamique du système)
- la temporisation des transitions

***Difficulté dans la modélisation**:*

Lors de la modélisation d'un système, la principale difficulté consiste à modéliser les délais, à savoir le temps que l'on passe dans chaque état en fonction des statistiques sur l'environnement.

***Système Markovien (sans mémoire)** :*

- Dans la plupart des cas le système est Markovien;
- s'il ne l'est pas, on modifie le modèle de telle manière à introduire une mémoire.

Modèles à Temps Discrets

C’est un niveau d’abstraction supérieur des Modèles à Temps Continu(MTC): le temps est regroupé en intervalles.

Ceci réduit la complexité du système car on ne s’intéresse plus aux détails de tout ce qui se passe à chaque instant mais à l’ensemble de tout ce qui s’est passé dans un intervalle de temps donné.

Inconvénient: Les MTD sont plus difficiles à représenter du fait que plusieurs événements puissent avoir lieu pendant une même unité de temps (événements concurrents).

Formalismes (haut niveau) permettant la modélisation de chaînes de Markov

- *Les Réseaux de Files d'Attente (RFA) :* approche orientée «ressources consommées par des clients»;

- *Les Réseaux de Petri Stochastiques* (RPS): analyse fine des synchronisations ;

- *Les Algèbres de Processus*: langages formels permettant de modéliser les systèmes informatiques concurrents ou distribués (composition concurrente, exécution parallèle);

- *Les Réseaux d'Automates (RA)* : intégration des synchronisations au modèle état-transition. Ils permettent à la fois de structurer l'approche par automates et de modéliser les délais. Il y a des formalismes à temps continu et à temps discret;

Comparaison de ces approches

Les approches «Algèbres de Processus » et «Réseaux d'Automates » n'ont pas de notion d'entité et de flot comme c'est le cas dans les RFA.

Elles offrent une vision structurée de sous-systèmes indépendants, qui interagissent entre eux (modèles modulaires).

Il est possible de générer des Chaînes de Markov régissant le système à partir de sous-systèmes interconnectés.

Techniques et outils de simulation

Simulateurs de bas et de haut niveau (orientés événements discrets)

1. Simulateurs (traduits en langages classiques : C, C++, Java,…) de bas niveau et/ou utilisant des formalismes analytiques RFA, RP, CM, …

2. Outils de simulation de haut niveau : QNAP2, Glomosim, NS2, OPNET, OMNET, JSIM, …

Principe et Etapes de Modélisation

La modélisation nécessite la maîtrise du système et la connaissance de sa charge de travail

1. Choix du modèle ==> Niveau de détail
   - Des entités
   - Des composants
2. Métriques de performances
   - Métriques Orientées Utilisateur
     Ex. Délai d'attente.
   - Métriques Orientées Système
     Ex. Temps de réponse.
3. Goulot et Instabilité (boucles de feedback positive et négative)

Caractérisation de la charge de travail (Workload)

Charge de travail = nombre de requêtes imposées par un ensemble de tâches dans une application donnée.

Exemple. ***Caractérisation du trafic web***

- La distribution de la taille des documents *est* de type heavy tailed.
- La popularité des documents *est* ZipF
- La corrélation : entre la taille et la popularité *est* faible; entre la taille et la fréquence d'accès *est* significative pour les fichiers html et images; …
- La localité temporelle : indique que les documents web référencés dans le passé récent *sont* les plus probables d'être référencés dans le futur proche.

2. Les Chaînes de Markov à Temps Discrets

Définition d'un Processus Aléatoire à Temps Discret.

Un processus aléatoire (PA) décrit l'évolution d'une grandeur aléatoire en fonction du temps. Si le temps est continu (Intervalle) le processus est dit continu, sinon il est dit discret.

Un processus Aléatoire à Temps Discret (PATD) $\{X_n\}_{n\in T}$ est une suite de variables aléatoires X_n prenant des valeurs dans un ensemble d'états $E = \{0, 1, 2, ..., m\}$ à des instants $n \in T$. $T=\{0, 1, 2, ...\}$.

$X_n = i$ signifie que le processus prend la valeur d'état $i \in E$ à l'instant $n \in T$.
c-à-d le système régi par ce processus se trouve dans l'état i à l'instant n.

Exemple:
Soit un PATD $\{X_n\}_{n\in T}$ représentant le nombre de tâches dans un serveur web sollicitant un traitement. La taille du buffer est limitée à 8 tâches.
$E = \{0, 1, 2, ..., 8\}$ $T=\{0, 1, 2, ...\}$
Un exemple de trace de $\{X_n\}$:
$X_0 = 0$, $X_1 = 1$, $X_2 = 0$; $X_3 = 4$, $X_4 = 3$, $X_5 = 2$, $X_6 = 3$, $X_7 = 5$, $X_8 = 6$, $X_9 = 7$
$X_4 = 3$ signifie qu'à l'instant 4 le système contient 3 tâches

Définition d'une Chaîne de Markov à Temps Discret.

Un PATD est une Chaîne de Markov à Temps Discret (CMTD) ssi pout tout $n \geq 0$, pour $i_1, i_2, \ldots, i_{n-1}, i, j \in E$;

$$P(X_{n+1} = j / X_n = i; X_{n-1} = i_{n-1}; \ldots; X_0 = i_0) = P(X_{n+1} = j / X_n = i).$$

Càd la probabilité que le processus se trouve dans l'état j à l'instant $n+1$ ($X_{n+1} = j$) ne dépend que de son état précédnt ($X_n = i$) *et non des états antérieurs. On dit que le PATD est sans mémoire.*

Une CMTD est un PATD vérifiant la propriété d'absence de mémoire.

Définition d'une CMTD Homogène

Une CMTD est dite homogène lorsque $P(X_{n+1} = j / X_n = i)$ ne dépend pas de n. On l'appelle probabilité de transition.

$$P_{ij}(n) = P(X_{n+1} = j / X_n = i) = P_{ij} \quad i, j \in E$$

P_{ij} est alors la probabilité de transition de l'état i à l'état j.

$P = [P_{ij}]_{i,j \in E}$ est la matrice de transition de la CMTD.

$$\sum_{j \in E}^{n} P_{ij} = 1, \qquad pour\ tout\ i \in E$$

Représentation des Chaînes de Markov à Temps Discrets

Une CMTD est représentée par sa matrice ou son graphe de transition.

Exemple 1.

Soit une CMTD représentant l'état d'une machine (qui évolue dans cet ensemble d'états $E = \{0, 1, 2\}$) à des instants donnés de l'ensemble T ={0, 1, 2, ...}

Les Probabilités de transition $P(X_{n+1} = j / X_n = i)$ données sont

$P_{00} = P_{01} = 0.4 \quad P_{11} = 0.4 \quad P_{20} = P_{21} = 0.5.$ On suppose qu'il n'y a pas de transition de l'état 1 vers 0.

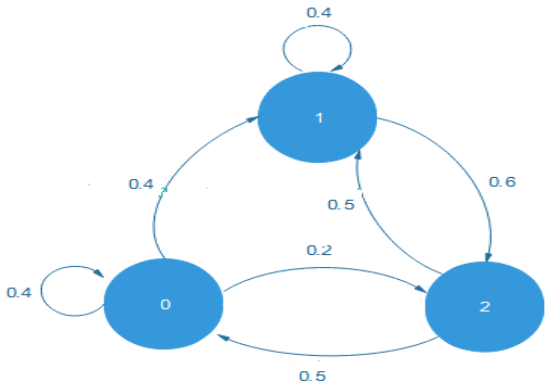
La matrice de transition correspondante :

$$P = \begin{bmatrix} 0.4 & 0.4 & 0.2 \\ 0 & 0.4 & 0.6 \\ 0.5 & 0.5 & 0 \end{bmatrix}$$

$P_{00} + P_{01} + P_{02} = 1 ; \quad P_{10} + P_{11} + P_{12} = 1; \quad P_{20} + P_{21} + P_{22} = 1$

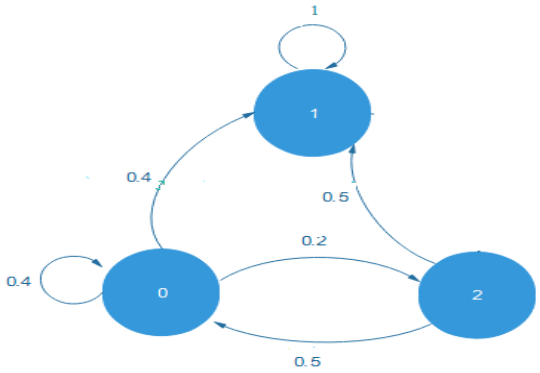
Représentation des Chaînes de Markov à Temps Discrets (Suite)

Le graphe de transition correspondant est:



Rem. La somme des valeurs des arcs sortant d'un état = 1

Exemple 2. On considère la CMTD représentée par ce graphe



Lorsque l'état 1 est atteint la CMTD y restera définitivement.

L'état 1 est appelé Etat Absorbant

Régime transitoire des CMTDs

Définition. Le régime transitoire est le régime d'évolution du système régi par cette CMTD qui n'a pas encore atteint un état stable (régime stationnaire).

Il est représenté par le vecteur des probabilités d'états $\pi(n)$
$\pi^{(n)} = (\pi_i^{(n)})_{i \in E}$ où $\pi_i^{(n)}$: proba. d'être à l'état i à la $n^{ème}$ transition (étape).

$$\sum_{i \in E} \pi_i^{(n)} = 1$$

$\pi^{(n)}$ consiste en une suite récurrente qui peut être obtenue uniquement si l'on connait le vecteur des probabilités d'état initial $\pi^{(0)}$

$$\pi^{(n)} = \pi^{(n-1)} * P \quad \text{ou} \quad \pi^{(n)} = \pi^{(0)} * P^{(n)}$$

où $P^{(n)}$ est la matrice de transition à n étapes $P^{(n)} = [P_{ij}^{(n)}]_{i,j \in E}$

$P_{ij}^{(n)} = P(X_{m+n} = j / X_m = i)$: *Proba. de passer de l'état i à l'état j en n étapes*

Il est aisé de montrer que $P^{(n)} = P^n$

Régime transitoire des CMTDs (Suite)

Exemple 3. On considère la CMTD de l'exemple 1 et on suppose qu'elle se trouve initialement dans l'état 1. Calculer les vecteurs des probabilités d'états $\pi^{(1)}$ $\pi^{(2)}$ $\pi^{(3)}$

$$P = \begin{bmatrix} 0.4 & 0.4 & 0.2 \\ 0 & 0.4 & 0.6 \\ 0.5 & 0.5 & 0 \end{bmatrix} \qquad \pi^{(0)} = (0,1,0)$$

$$\pi^{(1)} = (0,1,0) * \begin{bmatrix} 0.4 & 0.4 & 0.2 \\ 0 & 0.4 & 0.6 \\ 0.5 & 0.5 & 0 \end{bmatrix} = (0,\ 0.4,\ 0.6) \qquad (0+0.4+0.6=1 \text{ vérifié})$$

Régime transitoire des CMTDs (Suite)

$\pi^{(2)} = (0, 0.4, 0.6) * \begin{bmatrix} 0.4 & 0.4 & 0.2 \\ 0 & 0.4 & 0.6 \\ 0.5 & 0.5 & 0 \end{bmatrix} = (0.3, 0.46, 0.24)$ (*0.3+0.46+0.24=1 vérifié*)

$\pi^{(3)} = (0.3, 0.46, 0.24) * \begin{bmatrix} 0.4 & 0.4 & 0.2 \\ 0 & 0.4 & 0.6 \\ 0.5 & 0.5 & 0 \end{bmatrix} = (0.24, 0.424, 0.336)$

(0.24+0.424+0.336=*1 vérifié*)

Ex. La probabilité que le système soit à l'état 1 après 2 transitions est $\pi_1^{(2)} = 0.46$.

Régime stationnaire des CMTDs

Définition.

Le régime stationnaire est atteint lorsque le système régi par cette CMTD aurait fonctionné suffisamment longtemps (pendant un temps *n*, en théorie $n \rightarrow \infty$ *et en pratique n est fini).*

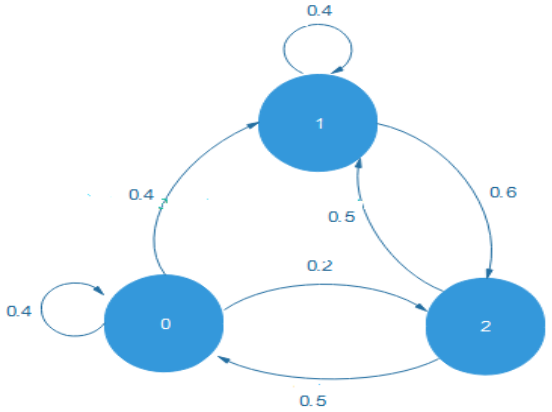
Définition.

Une CMTD est irréductible ssi de tout état *i* on peut atteindre tout état *j* en un nombre fini d'étapes *m*.

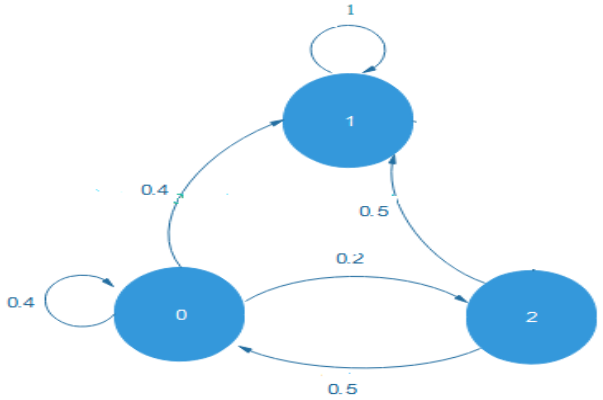
$\forall i, j \in E \ \exists m>0 \ / \ P_{ij}^{(m)} >0$

Régime stationnaire des CMTDs (Suite)

Exemple 4. Dans l'exemple 1, la CMTD est irréductible
Car de tout état (0, 1, 2) on peut atteindre
tout autre état (0, 1, 2).



Exemple 5. Dans l'exemple 2, la CMTD n'est pas
irréductible car de l'état 1 on ne peut
atteindre l'état 2 ni l'état 0 d'ailleurs.



Il est clair qu'une CMTD qui contient un état absorbant n'est pas irréductible

Régime stationnaire des CMTDs (Suite)

Définition.

Un état j est périodique si l'on ne peut y revenir qu'après un nombre d'étapes multiple de $k > 1$.

La période $D(j)$ de l'état j = PGCD des longueurs des circuits allant de j à j

$$D(j) = PGCD\{ k \geq 1 / P_{jj}^{(k)} > 0\}$$

Si D(j)>1 l'état j est dit périodique sinon (D(j)=1) il est apériodique.

Définition. La période d'une CMTD D(CMTD) est égale au PGCD des périodes de tous les états.

$$D(CMTD) = PGCD\{ D(i),\ i \in E \}$$

La CMTD est périodique si D(CMTD) >1, elle est apériodique si D(CMTD) =1

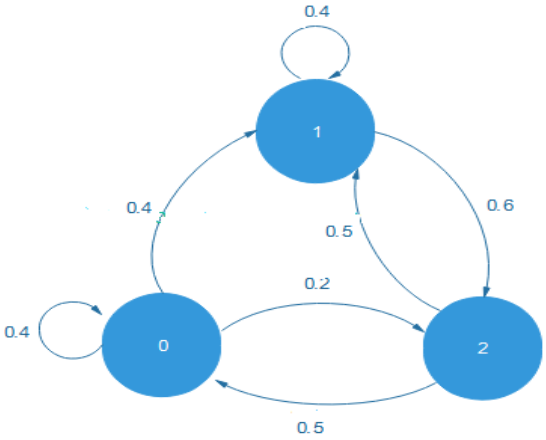
Régime stationnaire des CMTDs (Suite)

Exemple 6. ***Dans l'exemple 1, la CMTD est apériodique car:***

- *D(0) = PGCD{1,2,3,4,…} =1*
- *D(1) = PGCD{1,2,3,4,…} =1*
- *D(2) = PGCD{2,3,4,5…} =1*

Tous les états sont apériodiques

D(CMTD) = PGCD{D(0), D(1), D(2)} = 1



Exemple 7. ***La CMTD suivante est périodique car:***

- *D(1) = D(2) = PGCD{2,4,6…} = 2*

Les deux états sont périodiques

•D(CMTD) = PGCD{D(1), D(2)} = 2

La CMTD est périodique de période 2

Régime stationnaire des CMTDs (exemples)

Définition.

Une CMTD est ergodique si elle est irréductible et apériodique. On dit que le système régi par cette CMTD est stable.

Définition. Une CMTD ergodique possède une distribution stationnaire unique

$\pi = (\pi_0, \pi_1, \ldots \pi_n)$. $\pi = \lim_{n \to \infty} \pi^{(n)}$

π est obtenue par résolution de ce système d'équation

$$\pi = \pi * P \quad \text{et} \quad \sum_{i \in E} \pi_i = 1$$

Rem. Ce système d'équation donne lieu à m+1 équations à m inconnues (m : nombre d'états). On y trouve 1 équation redondante.

Régime stationnaire des CMTDs (exemples)

Dans l'exemple 1, la CMTD est ergodique car irréductible et apériodique.

Dans l'exemple 2, la CMTD n'est pas ergodique car apériodique mais pas irréductible.

Dans l'exemple 7, la CMTD n'est pas ergodique car irréductible mais périodique .

Exemple 8. Etant ergodique, la CMDT de l'exemple 1 possède une distribution stationnaire $\pi = (\pi_0, \pi_1, \pi_2)$ **obtenue par**

$$\begin{cases} (\pi_0, \pi_1, \pi_2) = (\pi_0, \pi_1, \pi_2) * \begin{bmatrix} 0.4 & 0.4 & 0.2 \\ 0 & 0.4 & 0.6 \\ 0.5 & 0.5 & 0 \end{bmatrix} \\ \pi_0 + \pi_1 + \pi_2 = 1 \end{cases}$$

Régime stationnaire des CMTDs (Suite exemples)

$$\begin{cases} \pi_0 = 0.4 * \pi_0 + 0.5 * \pi_2 & (1) \\ \pi_1 = 0.4 * \pi_0 + 0.4 * \pi_1 + 0.5 * \pi_2 & (2) \\ \pi_2 = 0.2 * \pi_0 + 0.6 * \pi_1 & (3) \\ \pi_0 + \pi_1 + \pi_2 = 1 & (4) \end{cases}$$

L'une des 3 premières est redondante donc on la supprime. Les 3 équations (1), (3) et (4) peuvent être considérées pour obtenir les 3 inconnues.

$$\begin{cases} \pi_0 = 0.4 * \pi_0 + 0.5 * \pi_2 \\ \pi_2 = 0.2 * \pi_0 + 0.6 * \pi_1 \\ \pi_0 + \pi_1 + \pi_2 = 1 \end{cases}$$

La résolution de ce système donne lieu à cette solution:

$$(\pi_0, \pi_1, \pi_2) = (15/58, 25/58, 18/58)$$

Rem. Il serait intéressant de vérifier l'équation 2, que l'on a éliminée,

Quelques métriques de performance des CMTDs

- Soit $P_{ij}^{(m)}$ la probabilité d'aller de i vers j en m étapes (P_{ij}^{m})
- Soit $f_{jj}^{(n)}$ la probabilité que le premier retour en *j* ait lieu *n* étapes après l'avoir quitté.

  - Soit f_{jj} la probabilité de revenir en *j* après l'avoir quitté.

$$f_{jj} = \sum_{n \geq 1} f_{jj}^{(n)}$$

  - Soit M_j le temps moyen de retour en *j*

$$M_j = \sum_{n \geq 1} n * f_{jj}^{(n)}$$

Métriques de performance des CMTDs (suite)

❖ Soit $f_{ij}^{(m)}$ la probabilité d'aller de *i* à *j* en exactement *m* étapes (sans passer par *j* de manière intermédiaire). On peut établir le système d'équations

$$f_{ij}^{(1)} = P_{ij} \quad \text{et} \quad f_{ij}^{(m)} = \sum_{k \neq j} P_{ik} * f_{kj}^{(m-1)} \quad \textit{pour } m \geq 2$$

❖ f_{ij} : la probabilité d'aller de *i* à *j*

$$f_{ij} = \sum_{m \geq 1} f_{ij}^{(m)}$$

On obtient $f_{ij} = P_{ij} + \sum_{k \neq j} P_{ik} * f_{kj}$

❖ Soit R_{ij} : le nombre moyen de passages par l'état *j sachant que l'on vient de l'état de i*

$$R_{ij} = \frac{f_{ij}}{1 - f_{jj}}$$

Exemple de Modélisation par CMTD.

Soit un capteur sans fil équipé de deux batteries montées en parallèle et alimentent le capteur indépendamment l'une de l'autre. Chaque batterie a une fiabilité égale à *P* au cours d'une année et il n'y a pas possibilité de recharge.

Soit X_n le nombre de batteries déchargées au début de la $n^{ième}$ année.

1. Espace des états (E) et l'espace des temps (T) du processus X_n ?
2. Probabilités de transition d'un état *i* à un état *j* ?

On a 3 états possibles:

0: 0 batterie déchargée
1: 1 batterie déchargée
2: 2 batteries déchargées

$E = \{0, 1, 3\}$

Le système est analysé chaque année

$T = \{0, 1, 2, 3, ...\}$

Calcul des probabilités de transition.

$P_{ij} = P(X_n = j / X_{n-1} = i)$: Probabilité que le capteur ait *j* batteries déchargées à la $n^{ème}$ année sachant qu'il avait *i* batteries déchargées à la $(n-1)^{ème}$ année

$P_{00} = P(X_n = 0 / X_{n-1} = 0) = ?$
On a 1 possibilités: les 2 batteries restent chargées (fiables) avec la probabilité *P*P*

$$P_{00} = P^2$$

$P_{01} = P(X_n = 1 / X_{n-1} = 0) = ?$
On a 2 possibilités:

1. la batterie 1 reste chargée avec la probabilité *P* et la batterie 2 se décharge avec la probabilité *(1-p)* (avec probabilité *p*(1-p)*)
2. la batterie 2 reste chargée avec la probabilité *P* et la batterie 1 se décharge avec la probabilité *(1-p)* (avec probabilité *p*(1-p)*)

$$P_{01} = p(1-p) + p(1-p) = 2p(1-p)$$

$P_{02} = P(X_n = 2 / X_{n-1} = 0) = ?$
On a 1 possibilité: les 2 batteries se déchargent avec la probabilité *(1-P)*(1-P)*

$$P_{02} = (1-P)^2$$

Vérification : $P_{00} + P_{01} + P_{02} = P^2 + 2p(1-p) + (1-P)^2 = 1$

$P_{10} = P(X_n = 0 / X_{n-1} = 1) = ?$

Il n'y a pas de recharge donc

$$P_{10} = 0$$

$P_{11} = P(X_n = 1 / X_{n-1} = 1) = ?$

Il y a 1 possibilité: la batterie chargée au début de la $(n-1)^{ème}$ année doit rester chargée au début de la $(n)^{ème}$ année et ceci est avec proba. P.

$$P_{11} = P$$

$P_{12} = P(X_n = 2 / X_{n-1} = 1) = ?$

Il y a 1 possibilité: la batterie chargée au début de la $(n-1)^{ème}$ est décharge au début de la $(n)^{ème}$ année et ceci est avec proba. $1-P$.

$$P_{12} = 1-P$$

Vérification : $P_{10} + P_{11} + P_{12} = 0 + p + 1-P = 1$

De la même manière on obtient les 3 autres

$$P_{20} = 0 \quad P_{21} = 0 \text{ et } \quad P_{22} = 1$$

Vérification : $P_{20} + P_{21} + P_{22} = 1$

3. Modélisation des systèmes par files d'attente

Théorie des files d'attente (FA) créée en 1950.

Elle a servi dans les années 70 à la modélisation des SI centralisés et des RTD.

Analyse par FA → mesures de performance (temps moyen de réponse, nombre moyen de demandes en attente, taux d'utilisation, débit en sortie, taux de perte, ….).

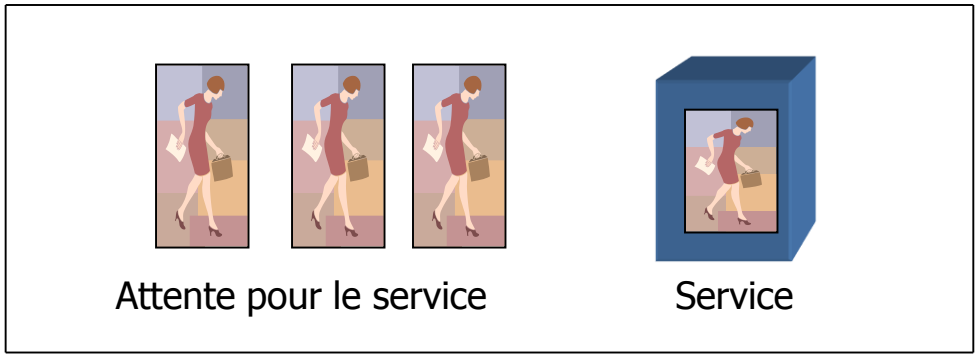
Il existe plusieurs modèles de FA : simples, avec priorités (priority queues), avec rappels (retrial queues), avec arrivées par groupe (batch arrival queues), avec vacances (vacation queues), …

Réseaux de files d'attente : Jackson, BCMP (années 70).

Récemment les G-Nets (*Generalized Networks* ou *Gelenbe Networks*).

Notion de partage de ressources

Ressource Hardware ou Software

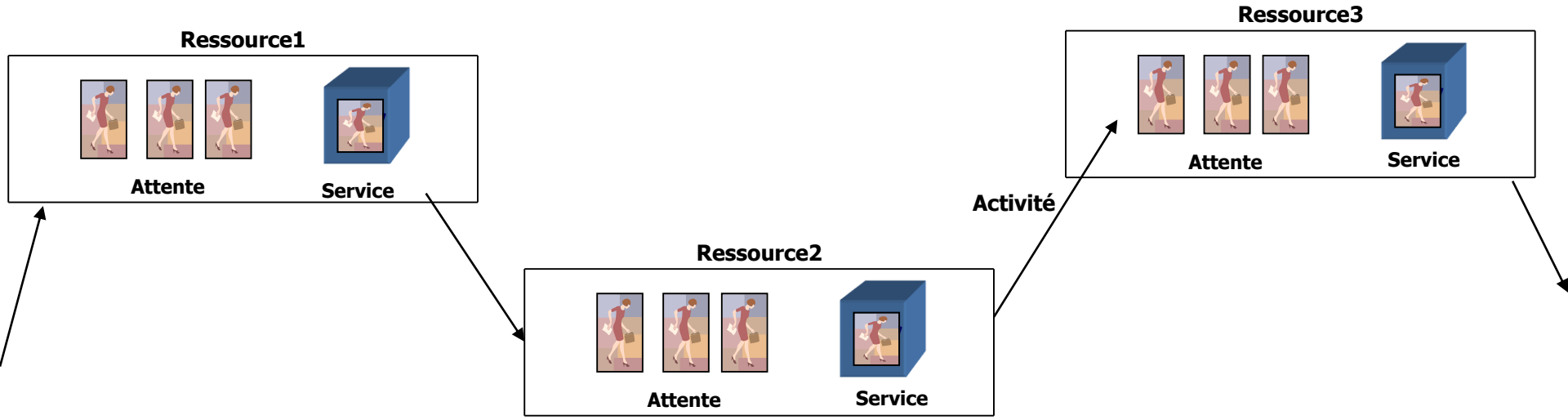


Ressource

Accès à une ressource

Activité = attente + réalisation du service →

Modèle = Système de file d'attente simple

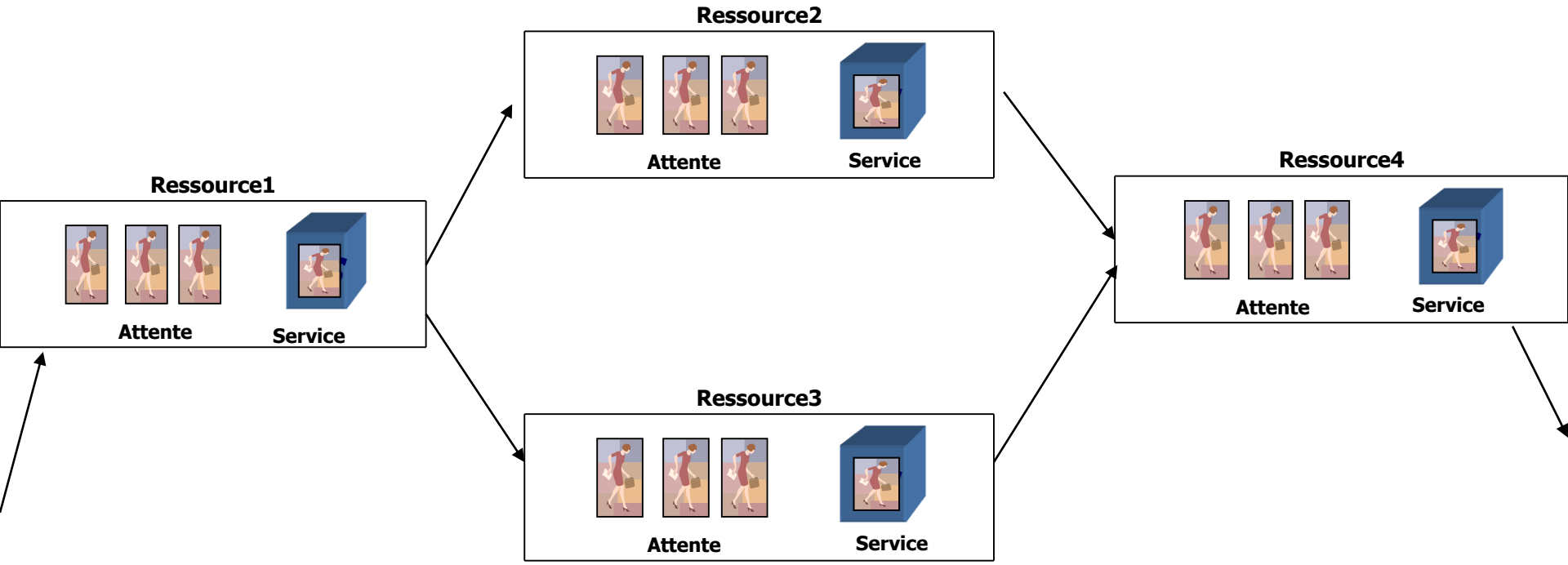


Accès à une succession de ressources

Activité = accès à une succession de ressources :{ (attente1 + service1)+ (attente2 + service2) + (attente3 + service3)}

----→

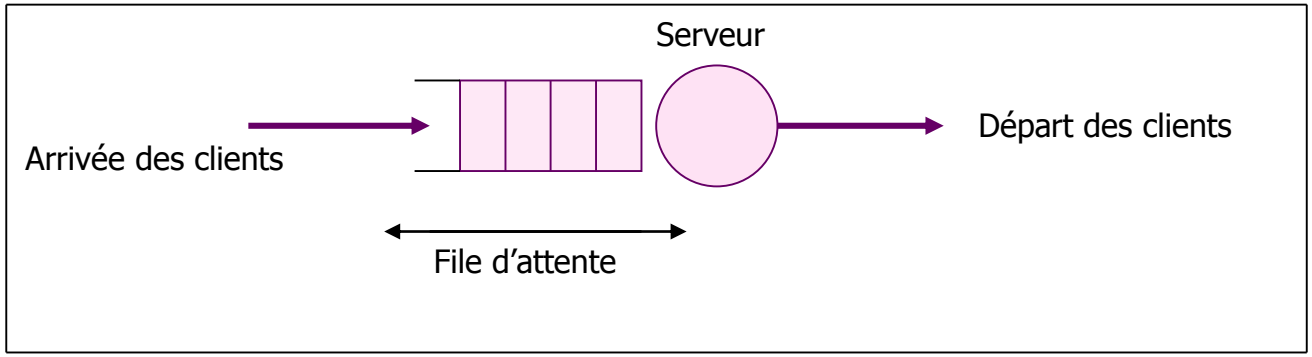
Modèle = Réseau de file d'attente



Accès à une succession de ressources

Accès aléatoire

Structure de base d'une file d'attente



Client : entité dynamique dans un système d'attente

Serveur : entité statique qui fournit le service demandé par le client

Hypothèses :

- Processus des arrivées
- Mécanisme de service
- Discipline de service (ordonnancement de la file)
- Taille de la file, etc.

Paramètres d'un système de files d'attente

Processus d'arrivée: Processus statistique par lequel les clients sont générés sur le temps. Ex : Processus de Poisson (le nombre de clients qui arrivent par unité de temps suit une loi de Poisson).

* La population finie ou infinie * la taille de la file affecte le nombre de clients dans la source

Distribution et capacité de service. Caractérisation de la quantité de service requise par un client individuel appelée « *demande de travail ou de service* » en nombre d' « *unités de travail ou de service* ».

En général, les demandes de service sont identiquement distribuées avec une distribution commune

Ex. Notons par C (unités/sec), la capacité du serveur

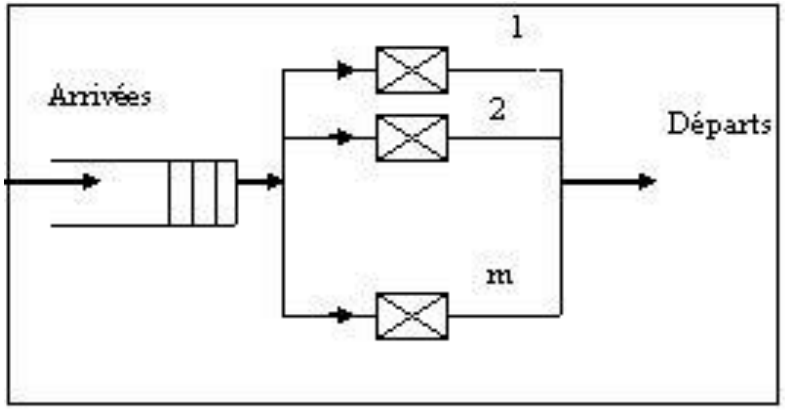
Si la demande de travail ou de service est S (unités de travail) →

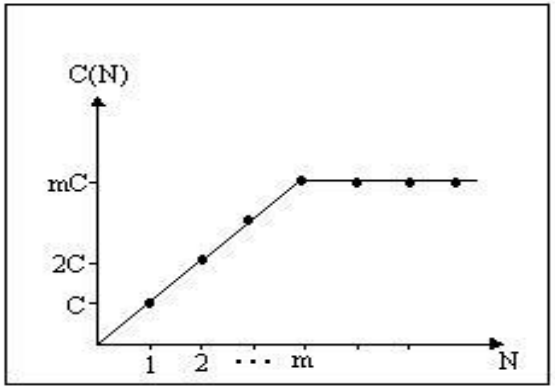
S/C est la « durée de service » → $\overline{S}$/C est la « durée moyenne de service »

C/$\overline{S}$ est le « taux ou intensité de service »

Capacité de service variable

La capacité peut varier en fonction du niveau de congestion à la station de Service.





File multi-serveurs

Si *C* : capacité de chaque serveur

et *N* : l'état du système (nombre de clients dans station (file+serveurs))

Capacité totale de la station est ***C(N) = C* min (m, N)***

Discipline d'ordonnancement ou de service.

L'ordre selon lequel les clients seront servis.

- ***FIFO*** (First In First Out),
- ***SIFO*** (Shortest In First Out)
- ***Round Robin***
- **Foreground – background**
  - **File Foreground** (prioritaire)
  - **File Background**
- ***PR*** (Preemption: Priorité préemptive)**,**
- ***HOL*** (Head-of-Line: Priorité non-préemptive), ….

Notations de Kendall pour les files d'attente

Une file d'attente est généralement représentée suivant la notation de Kendall suivante:
A/B/S/N/D/L

Où

- A indique le type (la loi) de la v.a X qui représente la durée entre 2 arrivées successives
- B indique la loi de la v.a Y qui représente la durée de service
- S est le nombre de serveurs parallèles
- N est la capacité maximale de la file d'attente
- D indique la discipline de service
- L indique la population des usagers

Lorsque les 3 derniers éléments de la notation de Kendall ne sont pas spécifiés, ils correspondent par défaut aux valeurs suivantes:

N=+∞, D= FIFO et L=+∞

Pour spécifier A et B, on utilise les symboles suivants:

M (Markov): distribution exponentielle, **A = M** signifie **que** la durée entre 2 arrivées successives est exponentielle → le processus d'arrivée est de Poisson.

E_k : distribution d'Erlang d'ordre k.

H_k : distribution Hyper-exponentielle d'ordre k.

C_k ou Cox_k : distribution de Cox d'ordre k.

G : distribution générale, …

Pour des systèmes plus complexes, pour spécifier **A**, on introduit souvent des processus tels que :

$MMPP_k$ (Markov Modulated Poisson Processus) : permet d'introduire de la corrélation

IPP (Interrupted Poisson Processus) : processus de Poisson avec interruption

IBP (Interrupted Bernoulli Processus) : l'équivalent du processus IPP, …

Définition. Loi exponentielle

Soit la variable aléatoire X suivant une loi exponentielle de paramètres λ (X → exp(λ))

Cette loi est définie par sa fonction de densité $f(x) = \lambda e^{-\lambda x}$, $x>0$

Et sa fonction de répartition $F(x) = 1 - e^{-\lambda x}$, $x>0$,

$E(x) = 1/\lambda$ $\quad V(x) = 1 / \lambda^2$

Si X représente la durée de traitement d'une tâche,

E(x) = 1/ λ est la durée moyenne de traitement d'une tâche et

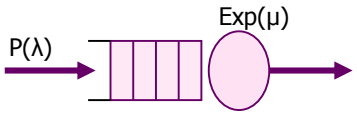
λ est le taux de traitement ou le taux de service : le nombre moyen de tâches traitées par UT,

Exemples de modèles de files d'attente

M/M/1

X : v.a. représentant la durée entre 2 arrivées
Y : v.a. représentant la durée de service
X → exp(λ) et Y → exp(μ)

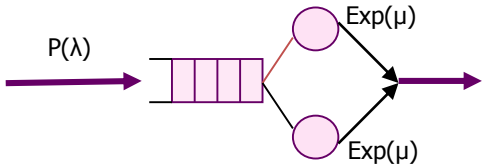
$F(x) = P(X \leq x) = 1 - e^{-\lambda x}$,
$G(y) = P(Y \leq y) = 1 - e^{-\mu x}$



λ = taux d'arrivée
1/ λ = durée moy. Inter-arrivée
μ = taux de service
1/μ = durée moy. service

Condition de stabilité: λ/μ <1

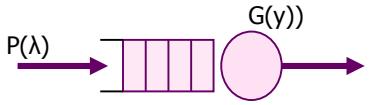
M/M/2



Condition de stabilité: λ/(2*μ) <1

M/G/1

X : durée entre 2 arrivées
Y : durée de service
$F(x) = P(X \leq x) = 1 - e^{-\lambda x}$
G(y) générale ou inconnue
μ: taux de service



Condition de stabilité: λ/μ <1

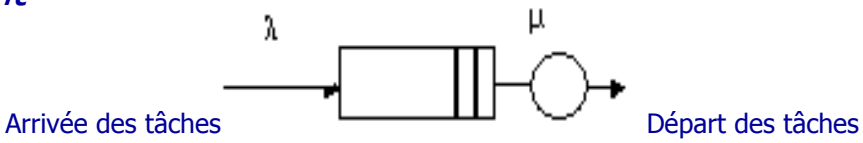
Exemples de modèles de systèmes réels

Exemple1. On considère un serveur web simple vers lequel arrivent des tâches web selon un processus de poisson i.e les durées entre 2 arrivées successives suivent une loi exponentielle. Le nombre de tâches qui arrivent par seconde est 440.

On suppose également que la durée de traitement des tâches par le serveur web suit une loi exponentielle de taux 800 tâches/s.

Une approche de modélisation de ce système est une file simple où :

Client == Tâche Service == traitement

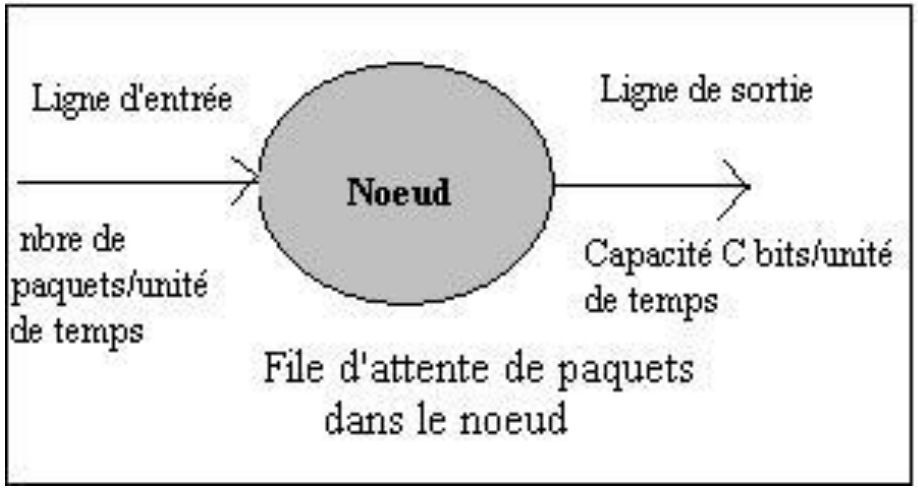


Condition de stabilité $\rho = \lambda/\mu = 440/800 = 0{,}55 < 1$

Temps moyen de réponse est donné par $E(T) = \frac{1}{\mu(1-\rho)}$

E(T)= 0,0027778 s

Exemple 2. On considère un commutateur de réseau de communication dont les fonctions sont le routage et l'acheminement des paquets.



- Les paquets arrivent suivant une moyenne de λ paquets/s.
- Ils attendent d'être transmis sur une ligne de sortie de capacité Cbits/s.

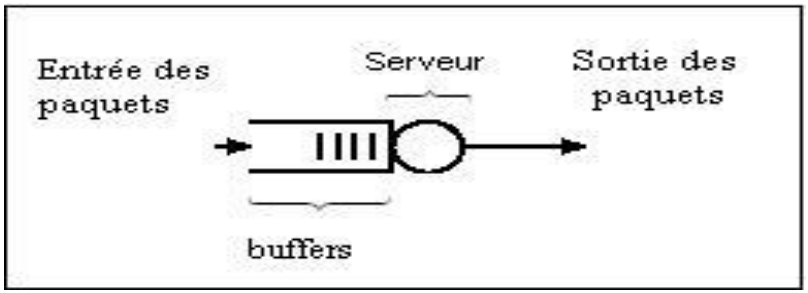
Rem. On suppose que les délais de traitement et de propagation sont négligeables et que l'on n'attend pas d'accusé de réception lors de la transmission d'un paquet.

Modélisation par FA

Modélisation: file d'attente à un serveur.

Client = ***Paquet*** *Service* = ***Transmission***.

Taux d'arrivée= **λ** Taux de service = **$\mu = C/T_p$** où Tp: taille d'un paquet (bits).

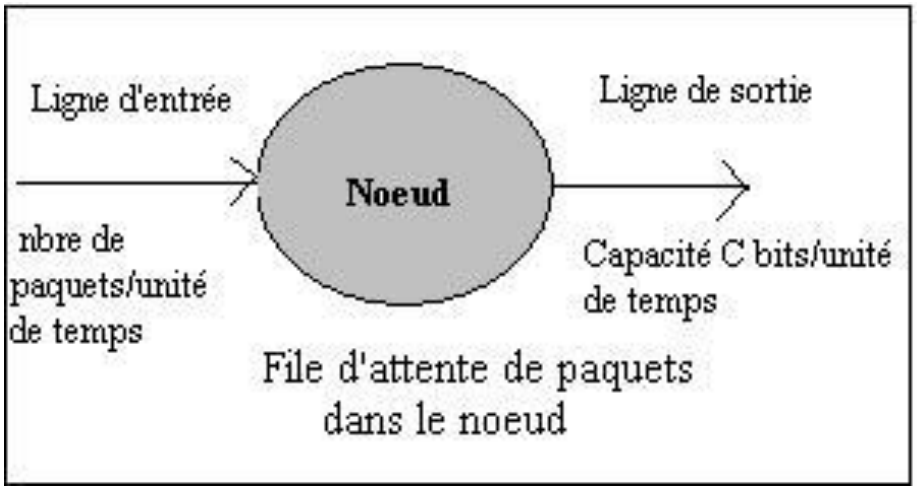


Condition de stabilité $\lambda/\mu<1$

Si processus d'arrivée ~~> P(λ) et processus de transmission ~~> P(μ)

→ Modèle ***M/M/1***

Modélisation d'un nœud de transmission en général



Cas où l'on néglige pas les délais de traitement et de propagation.

- Le modèle changera-il?
- Les paramètres changeront-ils?

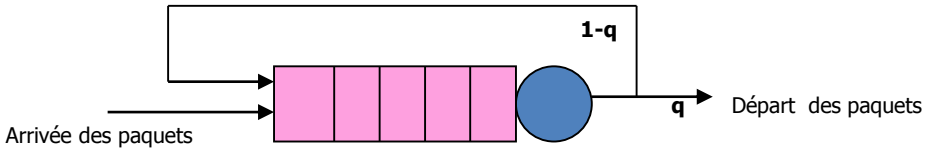
Cas où à chaque paquet transmis, il y a accusé de réception (ACK).

Même questions?

Autres modèles du nœud de transmission.

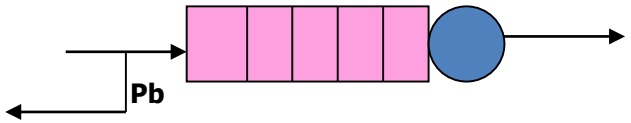
Modèle avec retour.

Si à chaque fin de transmission, le paquet est correctement transmis avec une probabilité q et erroné avec une probabilité (1-q) on peut avoir le modèle de file d'attente suivant:



Modèle avec perte.

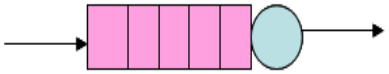
Capacité de la file limitée.



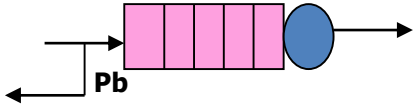
Il est évident qu'un modèle d'un nœud dépend fortement du protocole le régissant

Exemples de modèles simples d'un serveur web

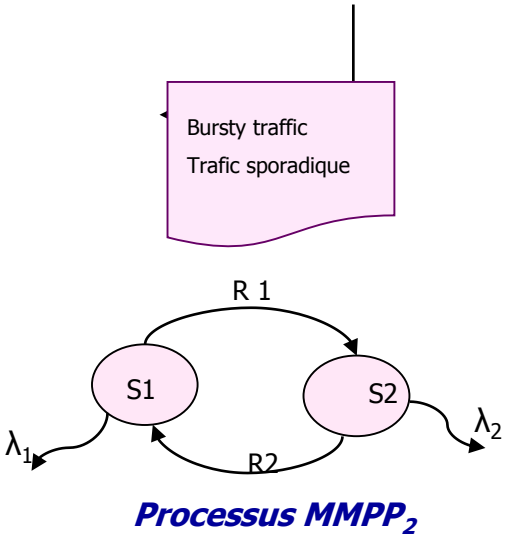
Client : tâche **Serveur** : traitement de la tâche



M/M/1, M/D/1, M/G/1 , M/G/1/PS



M/M/1/k, MMPP$_2$/G/1/k/PS.



Processus MMPP$_2$

Métriques de Performance de modèles de files d'attente

Il existe un bon nombre de formules fournissant les métriques de performances de certains systèmes d'attente (M/M/1 , M/M/1/k, M/M/s, ...) telles que

$P(n)$: Probabilité qu'il y ait n requêtes dans le système
N_m: Nombre moyen de requêtes dans le système
L_m : Longueur de la file ou Nombre moyen de requêtes en attente d'être servis
W_m : Délai moyen d'attente avant le service
T_m : Temps moyen de réponse ou temps moyen de séjour dans le système
U : Taux d'occupation d'un serveur, ...

Ces métriques de performance sont bien évidemment vérifiées lorsque le système est en régime stationnaire (stabilité et donc ne dépend pas du temps).

Métriques de Performance de modèles de files d'attente (Suite)

Métriques de performance du Système *M/M/1*

$P(n) = \rho^n (1-\rho)$

$N_m = \rho/(1-\rho)$

$L_m = \rho^2/(1-\rho)$

$W_m = \rho/ (\mu*(1-\rho))$

$T_m = 1/ (\mu*(1-\rho))$

La formule opérationnelle de Little est vérifiée pour tous les systèmes

$$N_m = \lambda * T_m$$

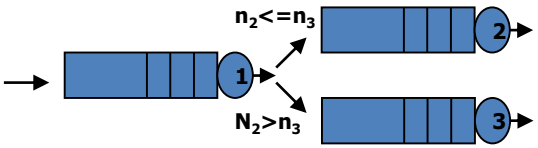
Réseaux de Files d'Attente (RFA)

Types de routage:

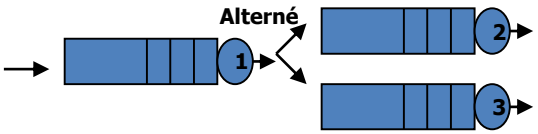
- **Routage vers la file la plus courte**
- **Routage cyclique**

Types de RFA :

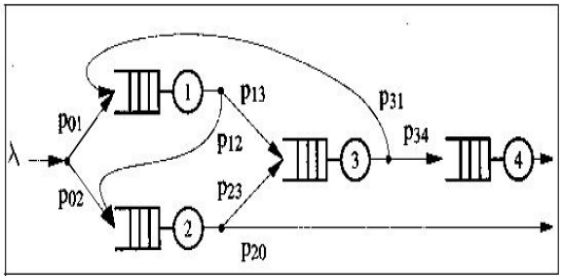
- **Réseaux ouverts**
- **Réseaux fermés**
- **Réseaux multi-classes**
- **réseaux à capacité limitée, …**



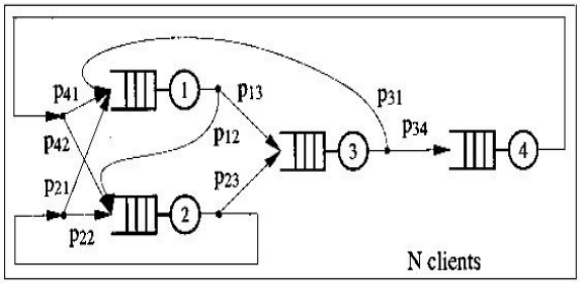
Routage vers la file la plus courte



Routage cyclique



Exemple de modèle de RFA. Routage probabiliste



Exemple de modèle RFA fermé.

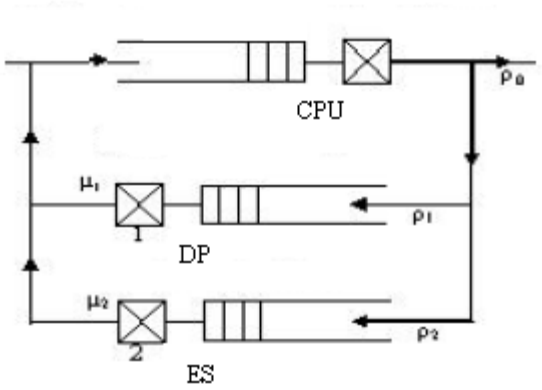
Réseaux de Files d'Attente à forme produit

Les RFA à forme produit = classe de RFA possédant une solution analytique très simple. On peut distinguer les Réseaux de **Jackson**, de **Kelly**, **BCMP** (Baskett, Chandy, Muntz and Palacios).

Les RFA à forme produit sont souvent utilisés dans la modélisation des SI complexes et des STD en particulier dans les réseaux à commutation de paquets.

RFA ⟹ Meilleure topologie, Routage avec un temps d'attente minimum, …

Exemple « un système multiprogrammé avec Mémoire Virtuelle ».

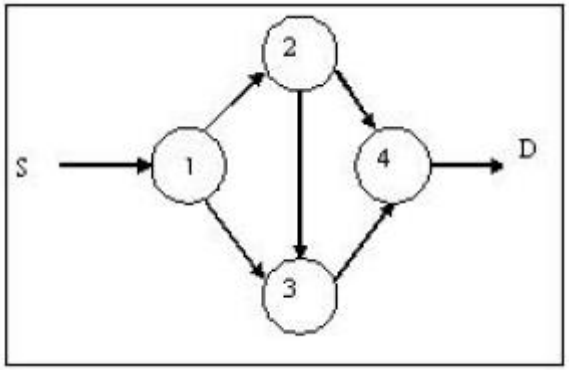


Les stations du réseau sont :
CPU, ES (Entré/Sortie), DP (Disque de pagination).

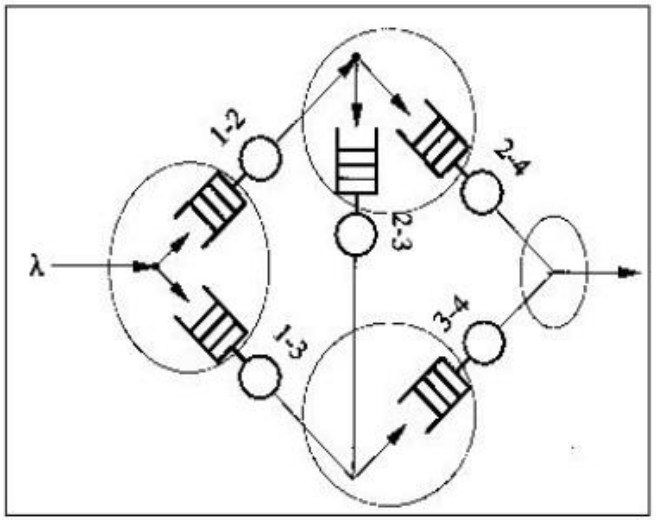
Les programmes arrivent vers la CPU et attendent une exécution.

- Lorsque un programme dans la CPU requière une E/S, il se dirige vers la station ES (proba. P2)
- Lorsqu'il fait un défaut de page il se dirige vers le DP (proba. P1)
- S'il termine son exécution totale, il quitte le réseau (proba. P0).

Exemple de modèles de réseau de communication



Les paquets générés par une source S doivent transiter dans le réseau afin d'atteindre leur destination D.



Métriques de Performances des RFA

Certaines classes de RFA vérifiant une forme produit (Jackson, BCMP,…) ont des solutions analytiques mesurant leurs performances.

NB. Deux métriques sont pratiquement possibles à obtenir, en l'occurrence:

- Tm : Temps de réponse du RFA
- Nm: Nombre moyen de requêtes dans le RFA

A condition de connaître chaque Nm_i ***des k sous-systèmes constituant le RFA***

$$Nm = \sum_{i=1}^{k} Nm_i$$

Et de la formule de Little ***Nm = λ* Tm***

On obtient ***Tm = Nm/ λ***

λ : taux de requêtes arrivant dans le RFA