Algorithmique du Network Calculus

Laurent Jouhet,
sous la direction d'Éric Thierry

LIP, équipe MC2
ENS de Lyon - Université de Lyon

7 novembre 2012

Le Network Calculus en deux minutes

Le Network Calculus expliqué à ma maman

Files d'attente

Files d'attente à la cantine...

Plusieurs magasins.

Données

Informations sur le nombre de personnes qui arrivent

Informations sur les temps de traitement

Questions : dans le pire des cas

- Est-ce que tout le monde va être servi ?
- Quelle doit être la taille des files d'attente ? (charge)
- Combien de temps est-ce que les gens attendent ? (délais)

Le Network Calculus en deux minutes

En pratique : comment ça marche et à quoi ça sert ?

Contexte

Mesure de performance dans les réseaux de communication :
Bornes sur les délais, et sur la charge des serveurs

Particularités

Analyse pire cas.
Outils mathématiques :
algèbre $(min, +)$, enveloppes.

Le Network Calculus en deux minutes

En pratique : comment ça marche et à quoi ça sert ?

Contexte

Mesure de performance dans les réseaux de communication :
Bornes sur les délais, et sur la charge des serveurs

Particularités

Analyse pire cas.
Outils mathématiques :
algèbre $(min, +)$, enveloppes.

Pas de probabilités aujourd'hui.

Applications pratiques

Qualité de service (internet)
Certification des réseaux de communication (avionique)

Bibliographie

et références dans cette présentation

Articles et ouvrages fondateurs


[Cruz, 1991] Cruz, R. L.
A calculus for network delay, Part I / Part II


[Chang, 2000] Chang, C.-S.
Performance Guarantees in communication networks


[Le Boudec, 2001] Le Boudec, J.-Y. and Thiran, P.
Network Calculus

Contributions

Blocs rouges

Autres sources

Références

Modéliser le trafic entrant et sortant

Fonctions croissantes cumulées

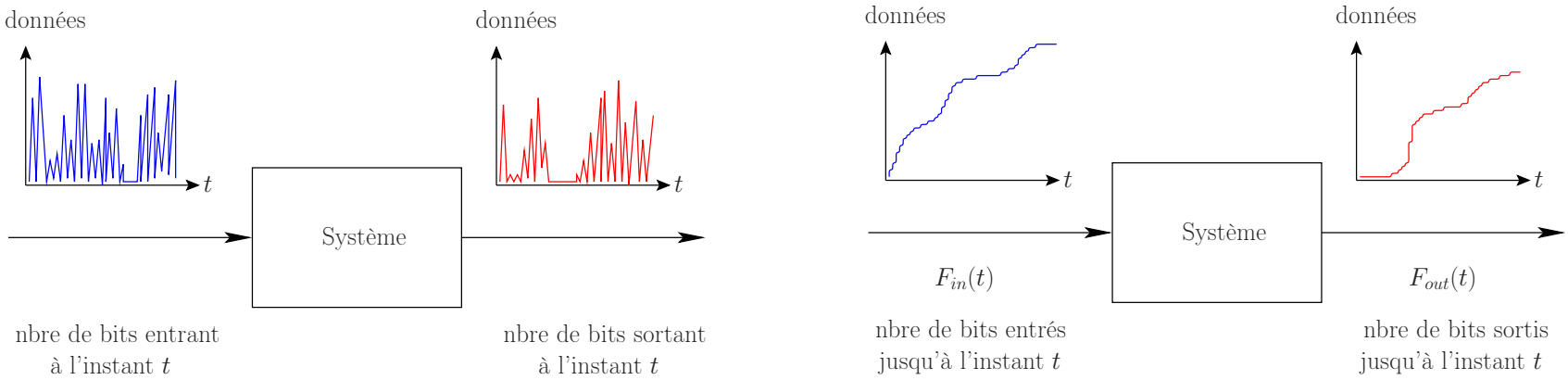


FIGURE : Entrée/sortie : instantanées, et cumulées.

Fonctions cumulées (modèle fluide)

Fonctions croissantes, continues à gauches et nulles en 0.

Système entrée/sortie

On suppose qu'il n'y a ni perte ni création de données

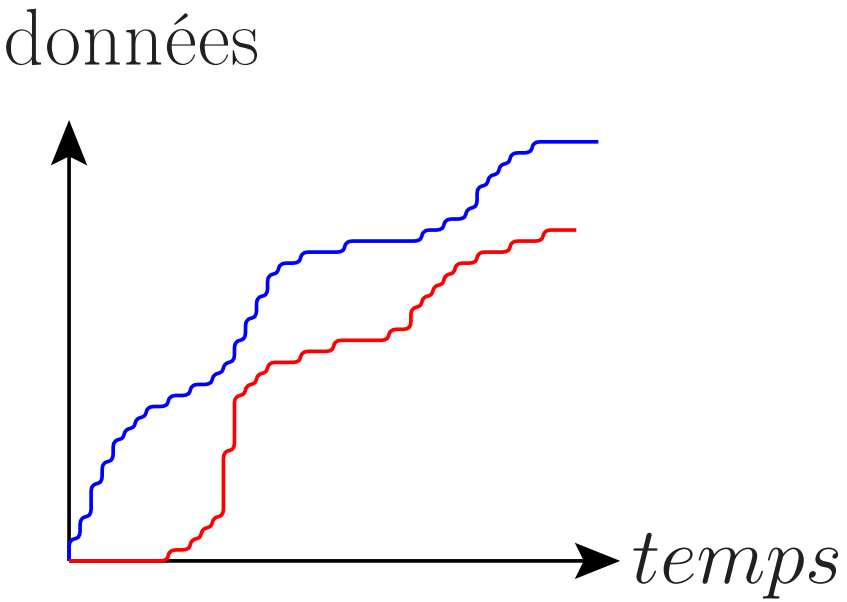
Trajectoire entrée/sortie

Flux en Entrée : F_{in}

Flux en Sortie : F_{out}

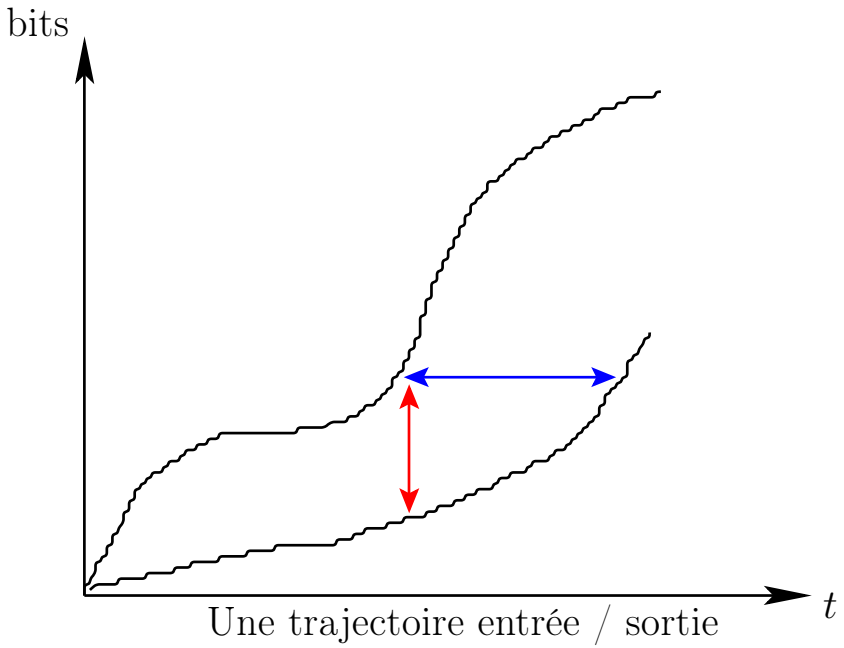
On a toujours

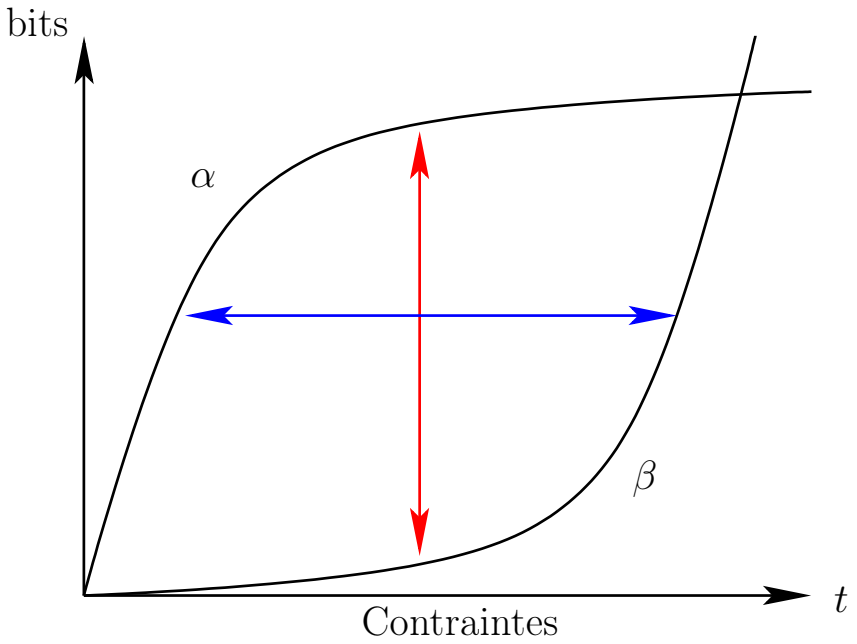
$$F_{in} \geq F_{out}$$



Modéliser les contraintes

Le principe





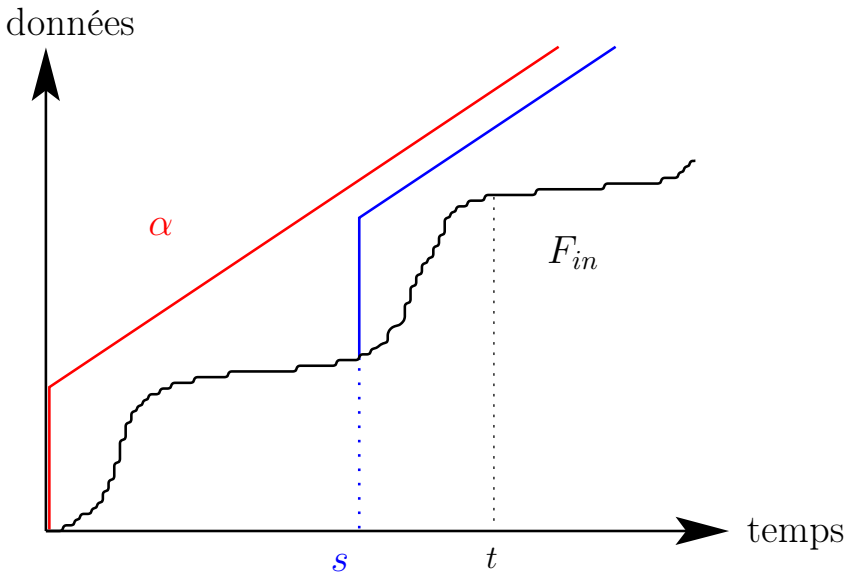
Enveloppes

Courbe d'arrivée, courbe de service :
valables pour toutes les trajectoires.
Objectif : travailler uniquement sur ces courbes.

Courbe d'arrivée

Borne supérieure sur la quantité de données qui peut entrer

L'arrivée est contrainte par α si



Courbe d'arrivée

Pour tous les instants :

$0 \leq s \leq t$

$F_{in}(t) - F_{in}(s) \leq \alpha(t-s)$

Autre écriture (même notion)

$F_{in} \leq F_{in} * \alpha$

Opérateur de Convolution

$$(f*g)(t) = \inf_{0 \leq s \leq t} (f(s) + g(t-s))$$

Courbes de service : définition mathématique

Borne inférieure sur la capacité de traitement d'un serveur

Plusieurs types de services. Un serveur fournit un service β (à tout flux en entrée) si

Service simple

$F_{out} \geq F_{in} * \beta$

Service strict

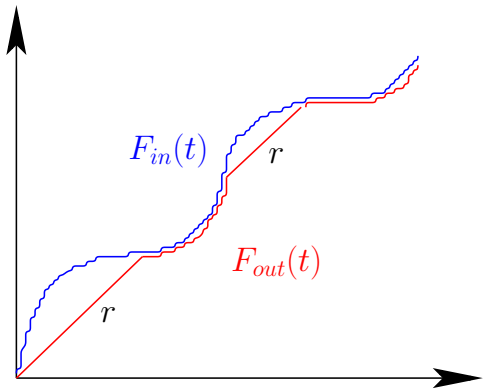
Pour toute période chargée $]s;t]$
$F_{out}(t) - F_{out}(s) \geq \beta(t-s)$.

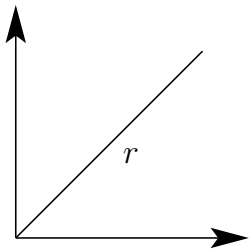
Période chargée : intervalle de temps pendant lequel la file d'attente est non vide.

Courbes de service

Des exemples pour mieux comprendre

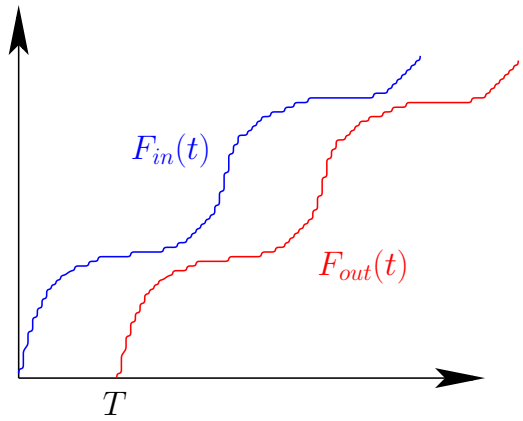
Débit $\geq r$

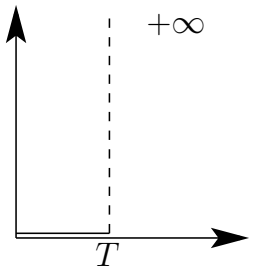




$\beta = r \times t$
Serveur simple (ou)
Serveur strict

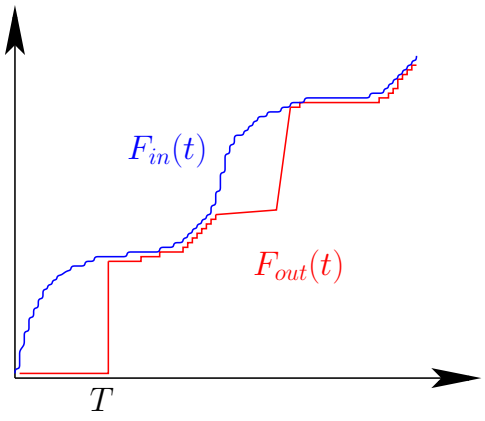
Attente $\leq T$

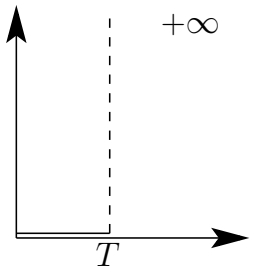




$\beta = \delta_T$
Serveur simple

Période chargée $\leq T$





$\beta = \delta_T$
Serveur strict

Serveur : une notion importante

Notations

Formalisation de la notion de serveur (non déterminisme)

$\mathcal{S}_{simple}(\beta) = \{(F_{in}, F_{out}) \mid F_{in} \geq F_{out} \geq F_{in} * \beta\}$
$\mathcal{S}_{strict}(\beta) = \{(F_{in}, F_{out}) \mid F_{in} \geq F_{out}$ et
$\forall$ période chargée $]s,t]$, $s \leq t$, $F_{out}(t) - F_{out}(s) \geq \beta(t-s)\}$.

Ce sont les serveurs qui sont simples ou stricts.

"β stricte $\Rightarrow$ β simple"

$\mathcal{S}_{strict}(\beta) \subseteq \mathcal{S}_{simple}(\beta)$

Obtenir des bornes

à partir des courbes d'arrivée et de service

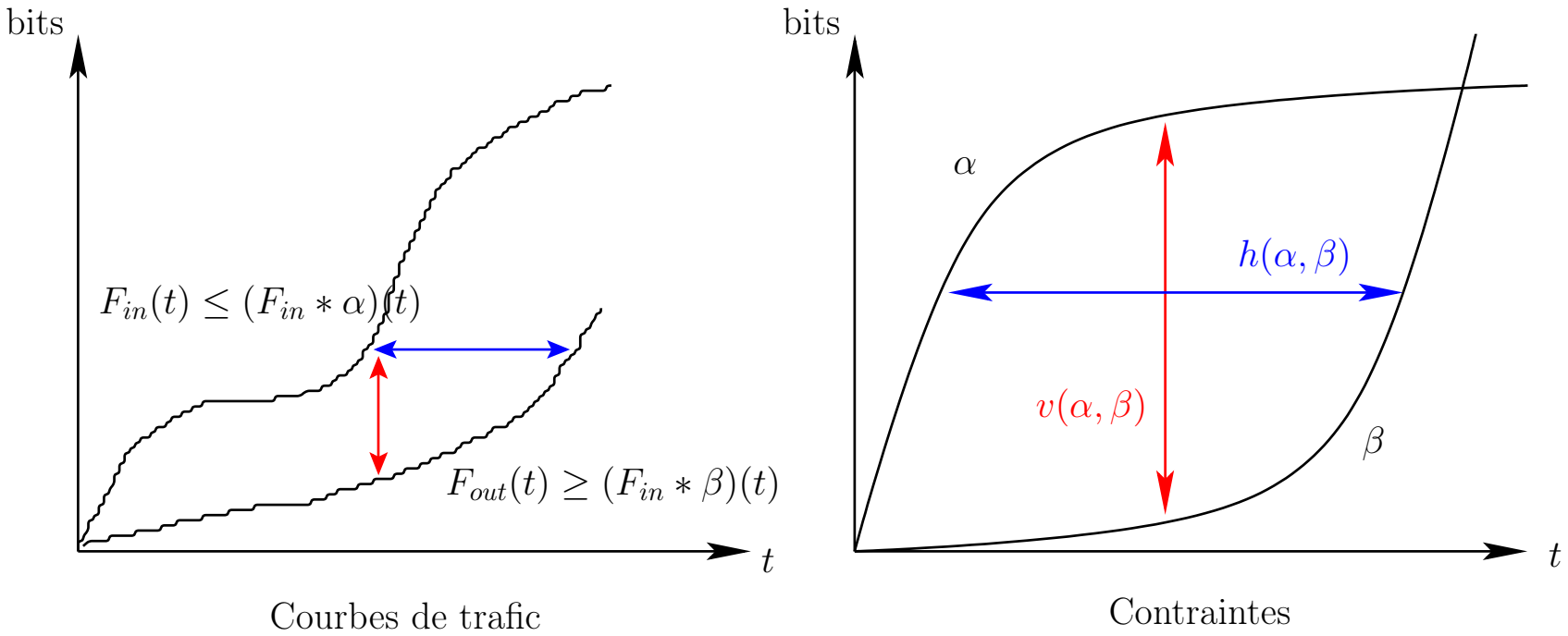


FIGURE : α et β : pour toutes les trajectoires possibles.

Politique de services : FIFO par flux.

Pour obtenir des bornes, on utilise des services simples.

Opérateurs (min,+)

Opérateurs fréquemment utilisés en Network Calculus

Opération point à point sur les fonctions : calculs

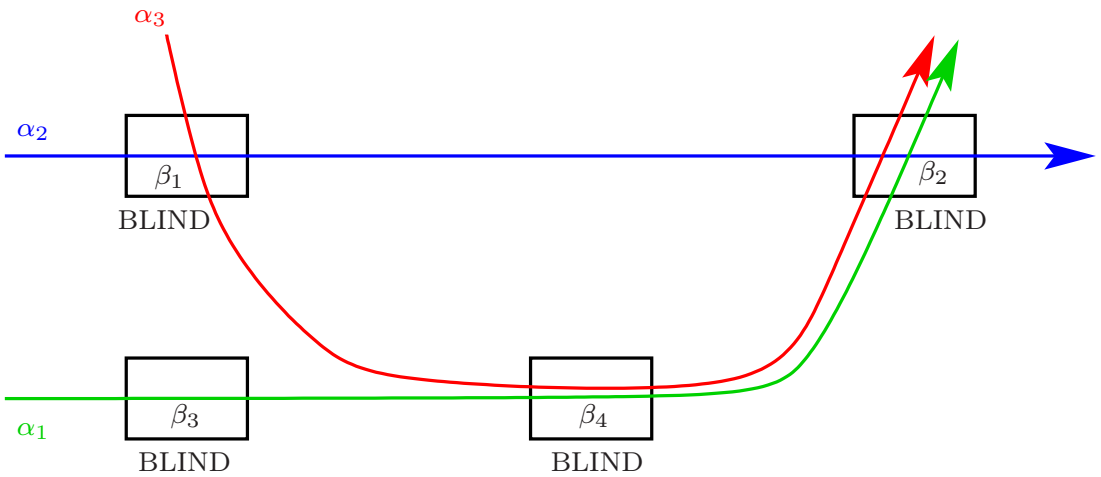
- Minimum : $\min(f,g)(t) = \min(f(t),g(t))$
- Addition : $(f+g)(t) = f(t)+g(t)$
- Partie positive : $(f_+(t)) = max(f(t),0)$

Opérateurs spécifiques

- Convolution : $(f*g)(t) = \inf_{0\le s\le t}(f(s)+g(t-s))$
- Déconvolution : $(f \oslash g)(t) = \sup_{u\ge 0}(f(t+u)-g(u))$
- Clôture sous-additive : $f^*(t) = \inf_{n\ge 0} f^{(n)}(t)$
- Clôture croissante positive : $f_\uparrow(t) = max(\sup_{0\le s\le t} f(s),0)$

Composition de plusieurs serveurs

Comment obtenir des courbes d'arrivée et de service ?



Plusieurs serveurs : cas simples

- Composer les serveurs (tandem)
- Propager les contraintes (courbe d'arrivée)
- Service résiduel.

Politique de service arbitraire (mais FIFO par flux)

Résultat classique : tandem

Concaténation de deux serveurs

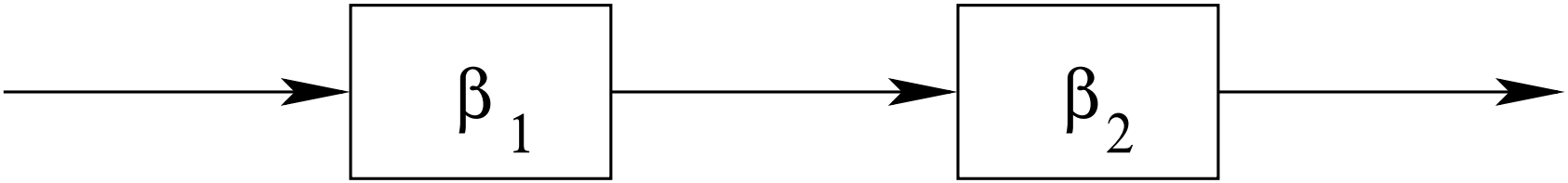


FIGURE : Concaténation $S_1 \circ S_2$ de deux serveurs

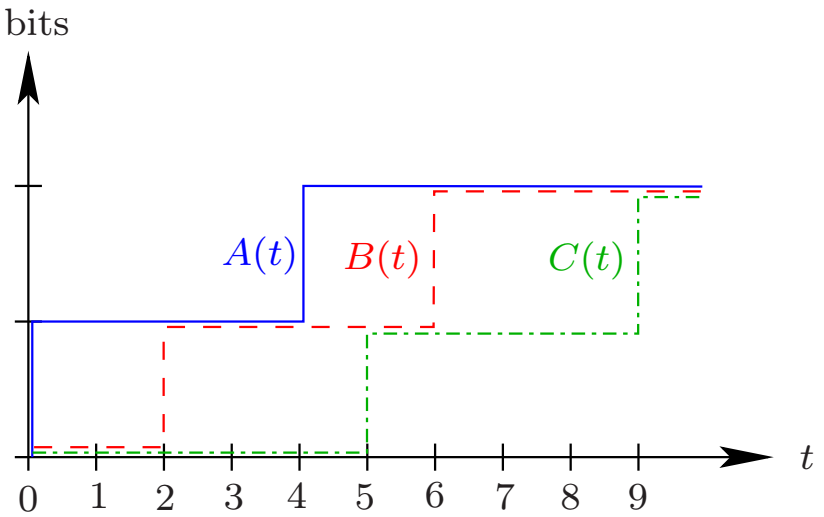
Tandem (service simple)

Un flux traversant deux serveurs en tandem. Alors la concaténation des deux serveurs offre un service simple minimal de courbe $\beta_1 * \beta_2$ à ce flux.

$$S_1 \subseteq \mathcal{S}_{simple}(\beta_1), S_2 \subseteq \mathcal{S}_{simple}(\beta_2) \Rightarrow S_1 \circ S_2 \subseteq \mathcal{S}_{simple}(\beta_1 * \beta_2)$$

Résultat classique : tandem

La notion de service strict n'est pas stable par concaténation



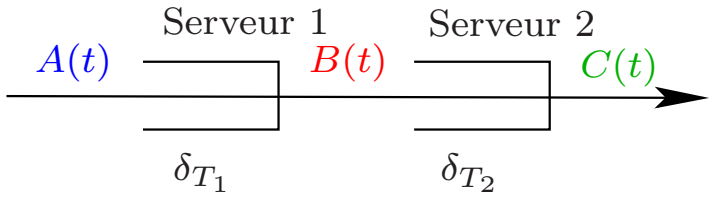


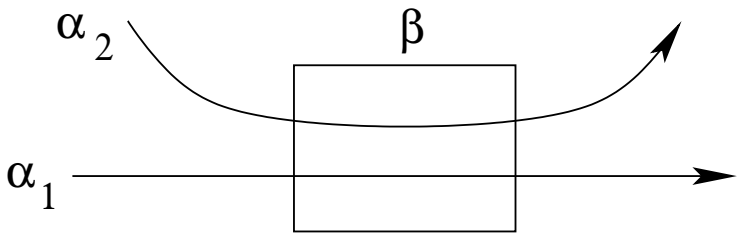
FIGURE : La concaténation n'est pas un serveur strict de courbe $\beta_1 * \beta_2$

Stabilité

Tandem : compatible avec un service simple, mais pas strict.

Résultat classique : service résiduel

En cas de trafic transverse



Service résiduel : serveur strict, multiplexage aveugle

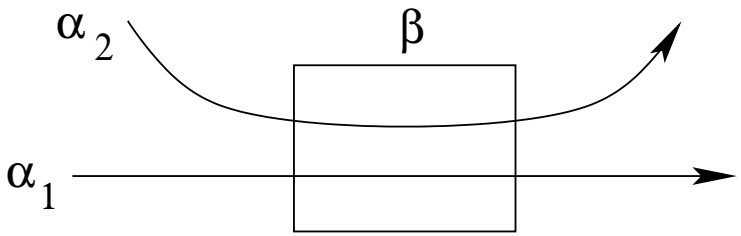
Service strict de courbe β pour les flux agrégés $A_1 + A_2$.
A_2 est borné supérieurement par α_2. On a :

1. service simple de courbe $(\beta - \alpha_2)_+$, au flux A_1 ;
   On perd le caractère strict.
2. si de plus A_2 a une priorité fixe supérieure à A_1, alors le flux A_1 reçoit un service simple de courbe $(\beta - \alpha_2)_+$.

Valable seulement si $(\beta - \alpha_2)_+$ est croissante.

Résultat classique : service résiduel

En cas de trafic transverse



Service résiduel : serveur strict, multiplexage aveugle

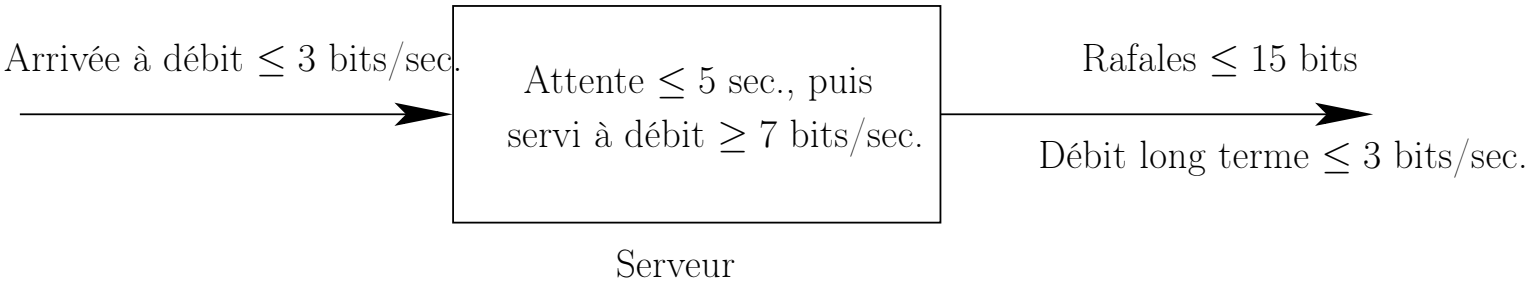
Service strict de courbe β pour les flux agrégés $A_1 + A_2$.
A_2 est borné supérieurement par α_2. On a :

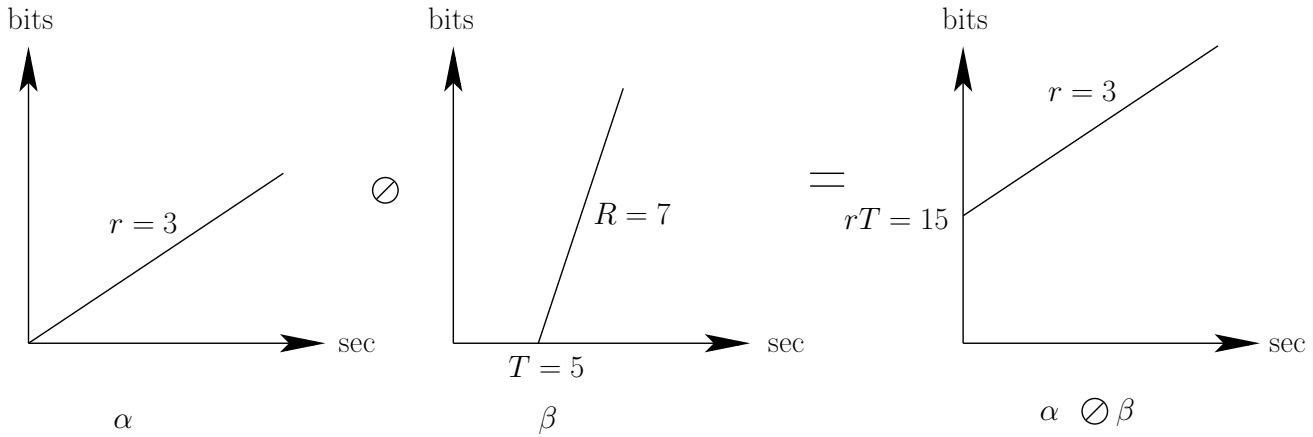
1. service simple de courbe $(\beta - \alpha_2)_\uparrow$, au flux A_1 ;
   On perd le caractère strict.
2. si de plus A_2 a une priorité fixe supérieure à A_1, alors le flux A_1 reçoit un service strict de courbe $(\beta - \alpha_2)_\uparrow$.

(Nouveau en NC). Remarque : $(\beta - \alpha_2)_\uparrow$ est croissante.

Résultat classique : propagation de contraintes

Principe : propager les contraintes d'arrivée





Courbes d'arrivée : indépendant du type de service.

Plusieurs notions de courbe de service

Pourquoi ?

Pas les mêmes utilisations :

Service simple

Notion fondamentale. Calcul de bornes.
Mise en tandem. Mais pas de service résiduel.

Service strict

Pratique pour les calculs : service résiduel.
Mais pas compatible avec la mise en tandem.

Raison historique :

Littérature

Différents modèles : Simple, Strict.
Mais aussi : faiblement strict, Variable Capacity Node, adaptative, S3C.

Plusieurs notions de courbe de service

Peut-on les comparer ?

On sait déjà qu'un serveur strict est aussi un serveur simple.

Plusieurs notions de courbe de service

Peut-on les comparer ?

On sait déjà qu'un serveur strict est aussi un serveur simple.

Théorème

Pour tout β (fixé), nous avons les inclusions suivantes :

$$\mathcal{S}_{vcn}(\beta) \subseteq \mathcal{S}_{strict}(\beta) \subseteq \mathcal{S}_{wstrict}(\beta) \subseteq \mathcal{S}_{simple}(\beta).$$

Notations : Variable Capacity Node, Faiblement strict.
Hiérarchie, et cas d'égalité.

Dans une même classe de fonction

Existe-t-il une "meilleure" courbe ?

Pour représenter un ensemble de trajectoires : serveur.

Autres propriétés étudiées (exemples)

On fixe le type de service :

- Monotonie ? oui
  si $\beta \leq \beta'$ alors $\mathcal{S}_T(\beta) \subseteq \mathcal{S}_T(\beta')$
- Clôture croissante positive ? (strict et wstrict)
  $\mathcal{S}_{strict}(\beta) = \mathcal{S}_{strict}(\beta\uparrow)$,
- Une courbe canonique (unique représentant) ? non
  Une seule courbe ne suffit pas pour modéliser finement.

Intérêt pratique : courbes faciles à manipuler ?

Autres modèles ?

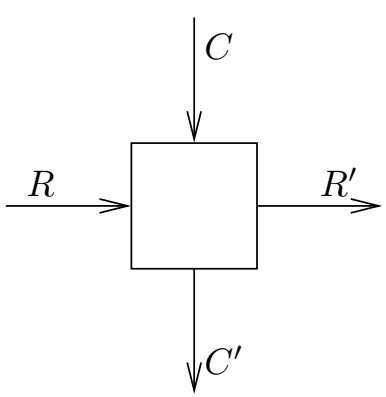
Comparaison avec le Real Time Calculus

Real Time Calculus

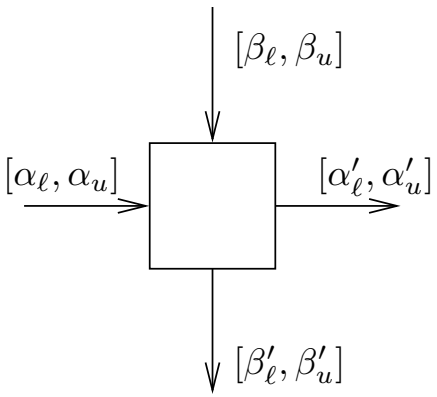
Des différences de modélisation :

Les équations qui relient les sorties aux entrées sont très différentes.

Au niveau d'un serveur, le service est vu comme une ressource.



(a)



(b)

RTC vs NC

Ces modèles sont-ils fondamentalement différents ?

Théorème

Pas de différence fondamentale entre NC et RTC :
On montre que RTC Greedy Processor et
NC Variable Capacity Node sont équivalents.

Méthode : établir une correspondance entre les équations de RTC [Wandeler] et celles de Variable Capacity Node [Thiele]

RTC en deux mots

Formalisme élégant. Outil complet (RTC Toolbox) ;
Inconvénient du service strict (pas de tandem).

Service intermédiaire

Existe-t-il une autre notion de service avec les propriétés souhaitées ?

Une notion de service intermédiaire ?

- $\mathcal{S}_{strict}(\beta) \subseteq \mathcal{S}_{inter}(\beta) \subseteq \mathcal{S}_{simple}(\beta)$,
- et qui soit stable par concaténation (tandem).

Service intermédiaire

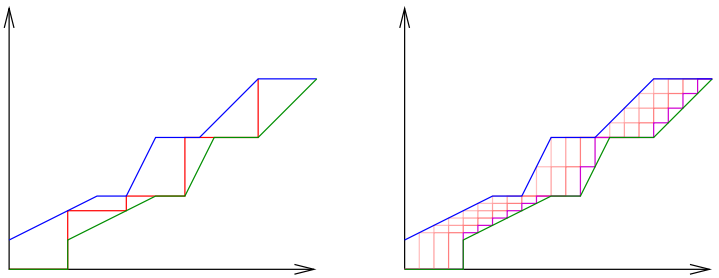
Existe-t-il une autre notion de service avec les propriétés souhaitées ?

Une notion de service intermédiaire ?

- $\mathcal{S}_{strict}(\beta) \subseteq \mathcal{S}_{inter}(\beta) \subseteq \mathcal{S}_{simple}(\beta)$,
- et qui soit stable par concaténation (tandem).

Résultat négatif

Une telle notion de service intermédiaire serait arbitrairement proche de la notion de service simple.



Approche classique

Utilisation classique du Network Calculus

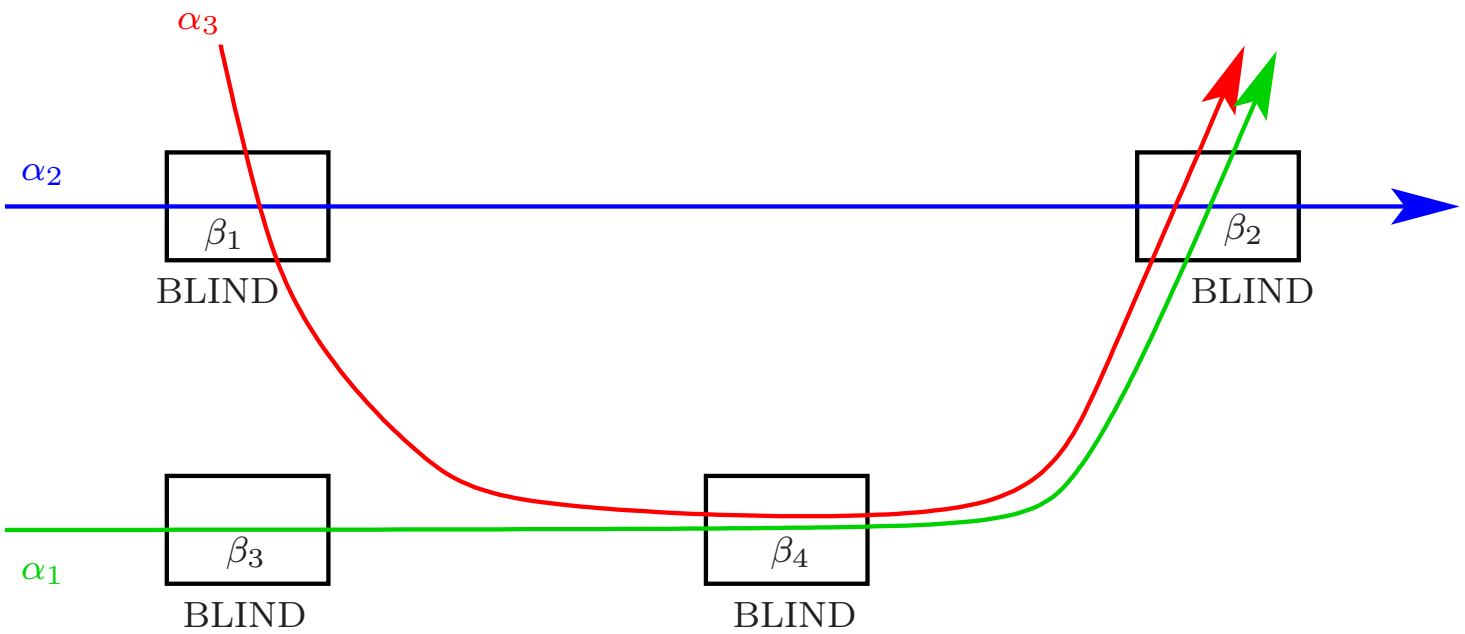


FIGURE : Courbes d'arrivée et de service résiduel, de proche en proche

"BLIND" : politique de service arbitraire.

Approche classique

Utilisation classique du Network Calculus

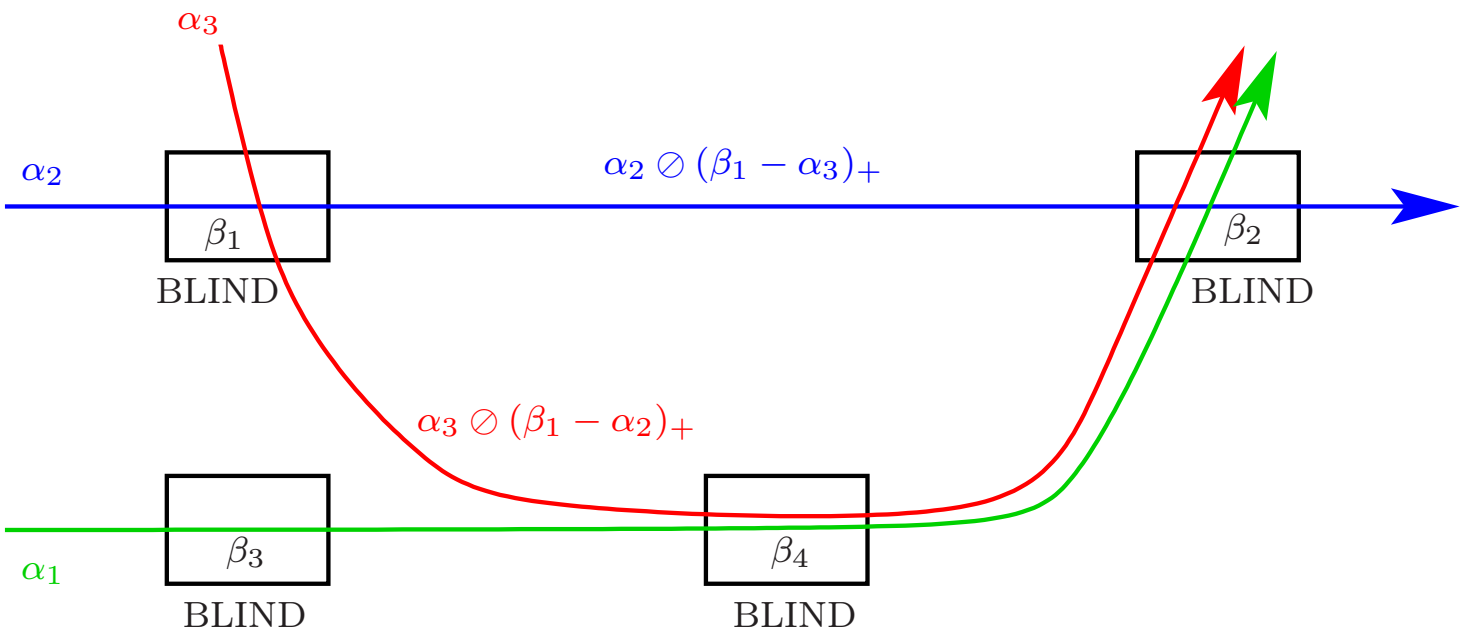


FIGURE : Courbes d'arrivée et de service résiduel, de proche en proche

"BLIND" : politique de service arbitraire.

Approche classique

Utilisation classique du Network Calculus

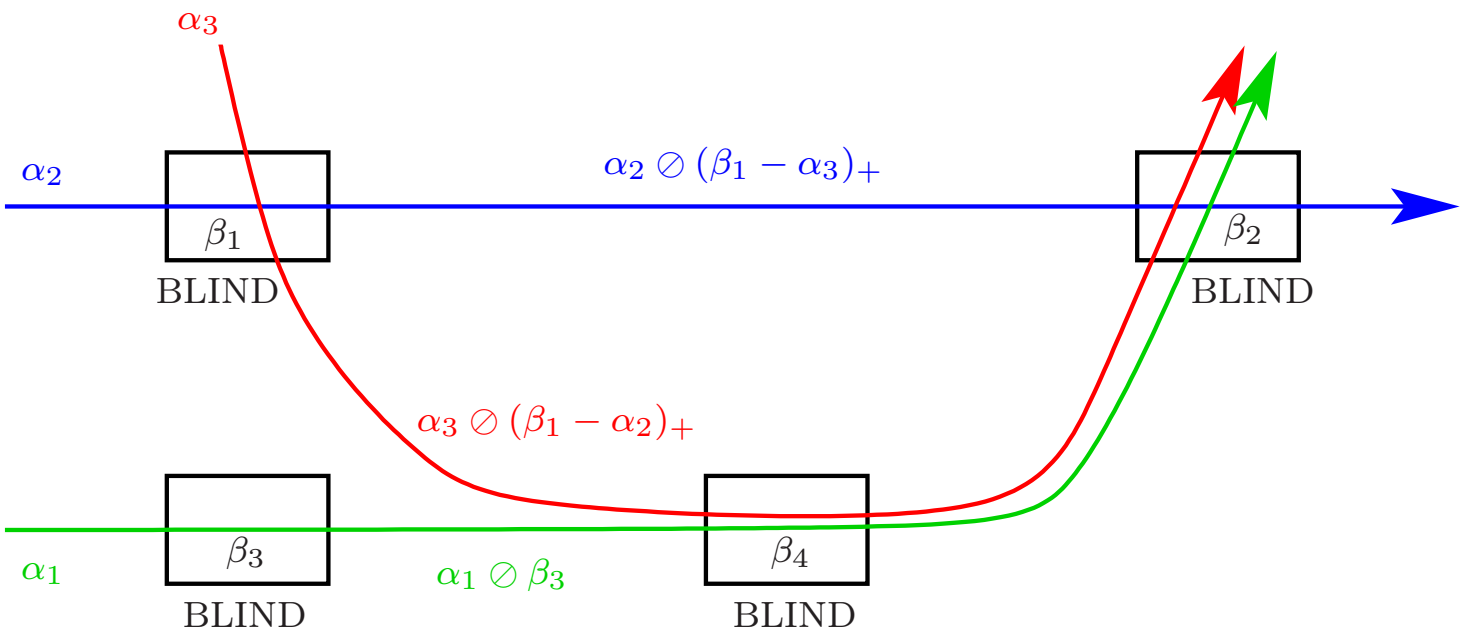


FIGURE : Courbes d'arrivée et de service résiduel, de proche en proche

"BLIND" : politique de service arbitraire.

Approche classique

Utilisation classique du Network Calculus

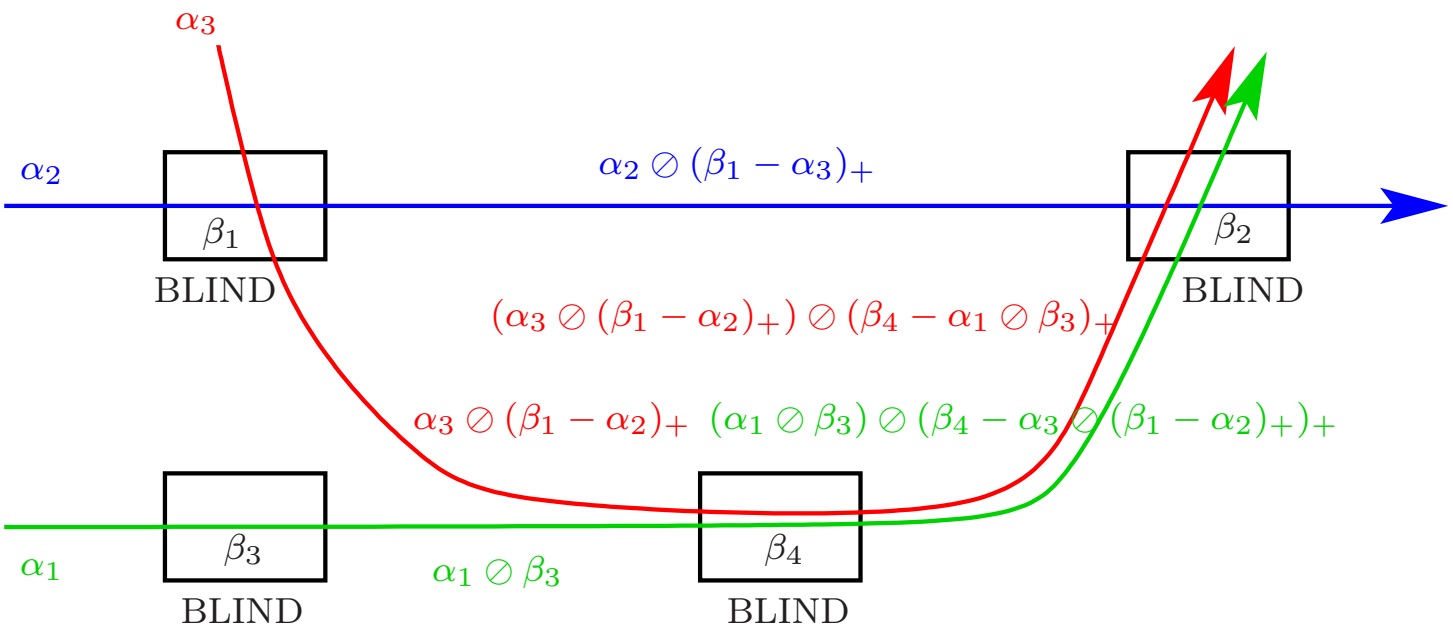


FIGURE : Courbes d'arrivée et de service résiduel, de proche en proche

"BLIND" : politique de service arbitraire.

Classes de fonctions stables

Quelle classe de fonctions utiliser en pratique ?

Notion centrale

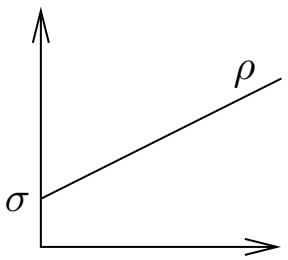
Composition des opérateurs :
déconvolution, soustraction, clôture positive...

Question importante [Bouillard et al 2007]

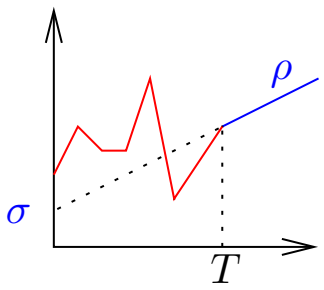
Classes de fonctions stables par ces opérateurs ?

Objectif : calculer des courbes d'arrivée de proche en proche.

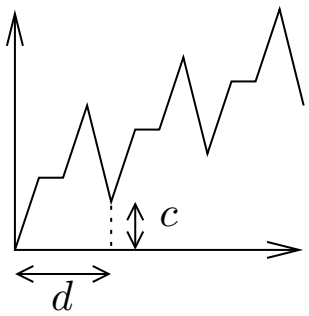
Classes de fonctions considérées



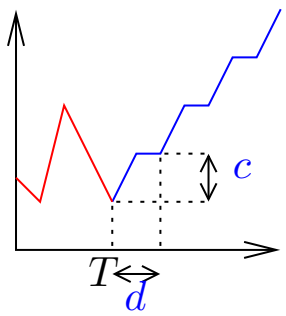
affine



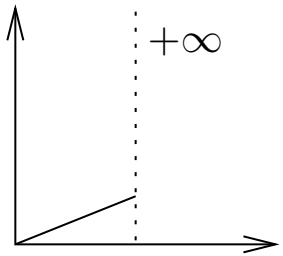
ultimement affine



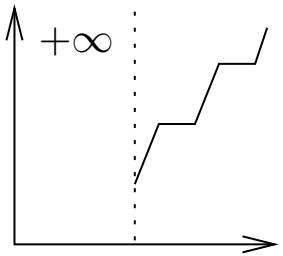
pseudo-périodique



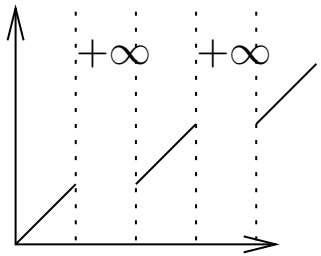
ultimement pseudo-périodique



franche



pas franche



pas franche

$\mathcal{F}[X, Y]$ affines par morceaux : sauts, changement de pentes en des points d'abscisse dans X et d'ordonnée dans Y

Stabilité des classes de fonctions

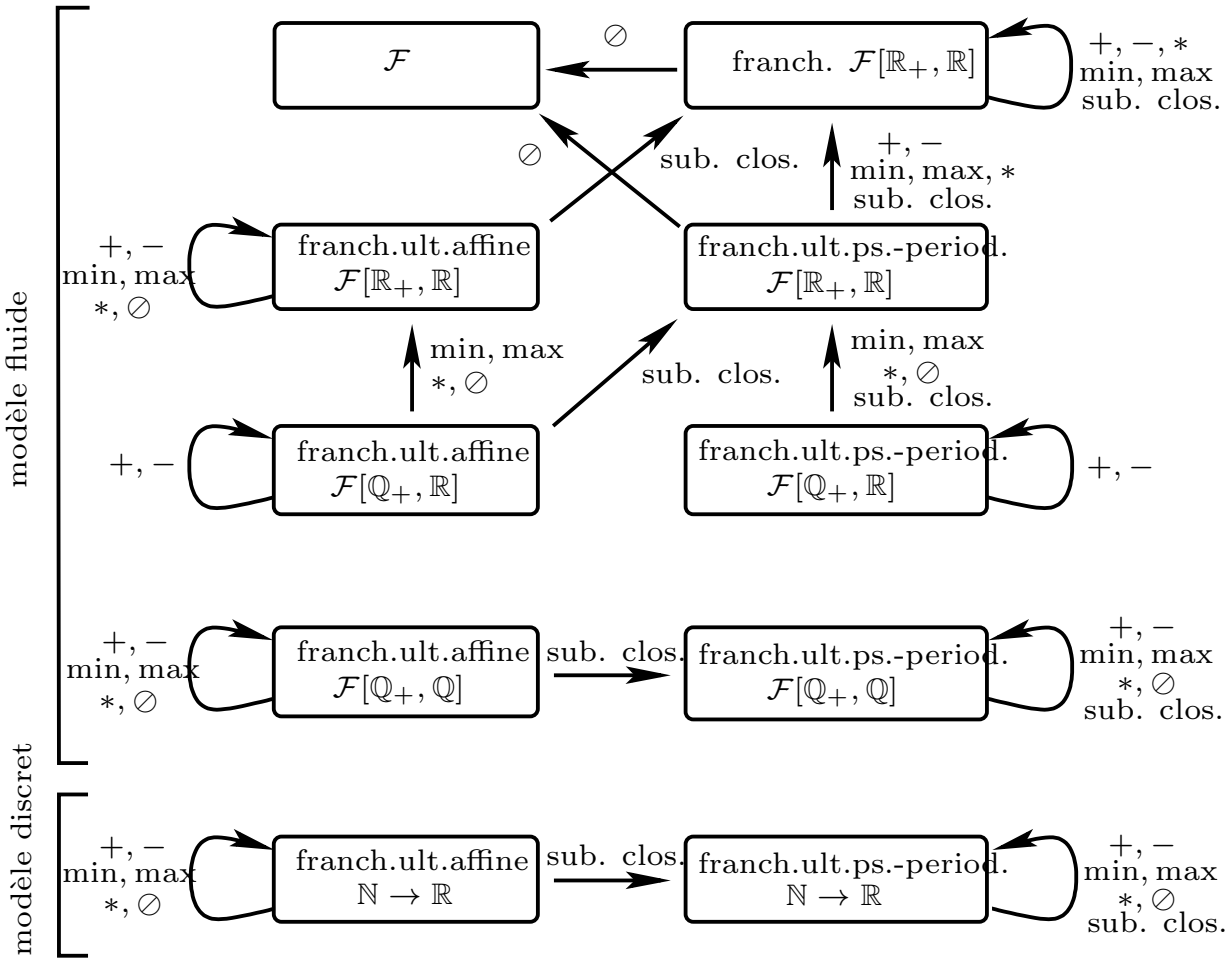
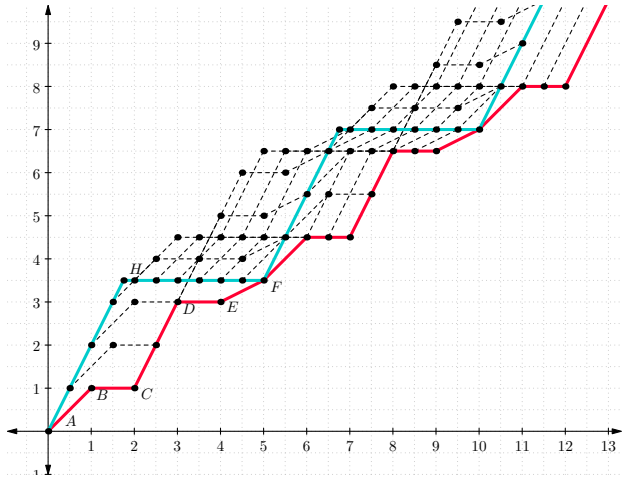


FIGURE : Stablilité des classes de fonctions [Bouillard et al 2007]

Reconstruction des fonctions

Peut-on travailler avec des fonctions plus simples ?



Fonctions continues, affines par morceaux, et ultimement pseudo périodiques

Théorème

Ces fonctions peuvent être obtenues à partir de fonctions affines par application des opérateurs du NC.

Utilisation de la clôture sous-additive. Résultat partiel.

Réseau

Topologie

Sans dépendances cycliques

Simple ? de nombreuses questions qui semblent compliquées.

Politique de service

Arbitraire (ou "aveugle")
FIFO par flux [Rizzo et al].

Hypothèse simple sur la politique de service.

Approche globale

Obtenir des courbes d'arrivée et de service de manière globale

Approches classiques :

- Propager les contraintes (courbe d'arrivée)
- Composer les serveurs (tandem)
- Service résiduel.

Approches globales :

- Calculer directement une courbe de service globale.
  Valable pour n'importe quelle courbe d'arrivée.
- Déterminer directement le pire cas
  pour une courbe d'arrivée donnée.

Convolution multidimensionnelle

Nouvel opérateur pour les configurations "Pay Multiplexing Only Once"

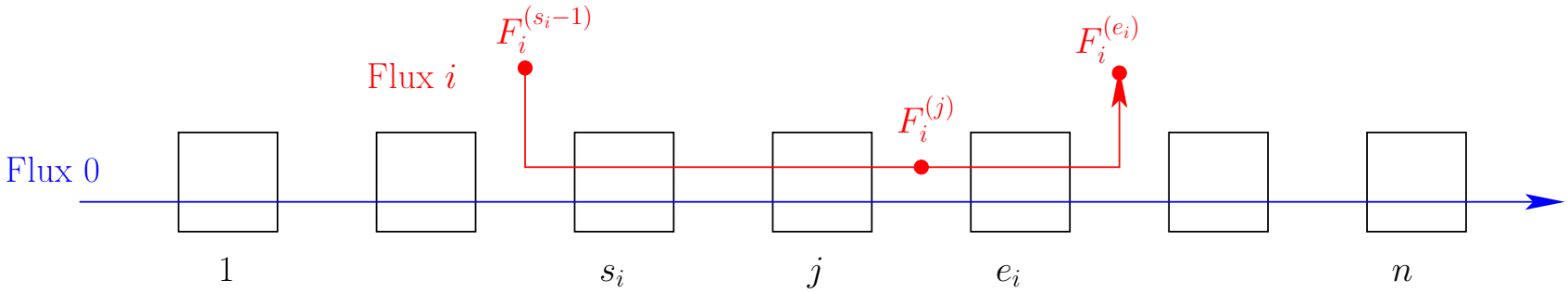


FIGURE : Chaque flux transverse interfère sur un sous-chemin

Historique

- Configurations étudiées dans [Schmitt et al, 2006]
- Opérateur introduit dans [Bouillard et al, 2007]
- Objectif : donner un algorithme dans le cas où les courbes d'arrivée et de service sont affines par morceaux et concaves/convexes.

Convolution multidimensionnelle

Opérateur qui généralise la convolution

Définition [Bouillard et al, 2007]

Serveurs numérotés dans $J = \{1, \ldots, n\}$, flux dans $I = \{1, \ldots, k\}$.
Famille de fonctions $\{f_i\}_{i \in I}$, et $\{J_i\}_{i \in I} \subseteq J$.
Convolution multidimensionnelle :

$$\psi(t) = \inf_{\substack{u_1, \ldots, u_n \geq 0 \\ u_1 + \cdots + u_n = t}} \sum_{i \in I} f_i\Big(\sum_{j \in J_i} u_j\Big)$$

Définition / existence d'une courbe de service
pour le service fourni au flux observé.
Expression mathématique : formule close.

Convolution multidimensionnelle

Aspect algorithmique

Cas particulier, où les courbes d'arrivée et de service sont affines par morceaux et concaves / convexes.
On note N le nombre total de segments des α_i et des β_i.

Algorithme pour obtenir la courbe de service

On peut donner la courbe de service : temps exponentiel en N.

Calcul des bornes sur les délais et la charge

En utilisant la programmation linéaire,
on peut calculer les bornes pire cas pour le flux principal en temps polynomial :
Programme linéaire à N contraintes linéaires sur N variables.

Convolution multidimensionnelle

Intérêt de cette méthode

On obtient une courbe de service :
valable pour toute courbe d'arrivée.

Une méthode effective ?

Méthode nouvelle pour trouver des bornes
sur la charge et le délai :

- dans certains cas, les bornes obtenues sont meilleures qu'avec les méthodes classiques
- dans d'autres, non.

Difficulté d'évaluer la qualité des bornes : quel est le pire cas ?

Obtenir des bornes exactes

Pas d'algorithme (même coûteux) qui donne des bornes exactes

Objectif

Donner un algorithme qui détermine des bornes exactes pour analyser un réseau sans dépendances cycliques.

Hypothèses supplémentaires

On étudie un flux en particulier (délai), ou un serveur (charge).
Politique de service arbitraire.
Les courbes d'arrivée et de service sont affines par morceaux.
On connaît la courbe d'arrivée du flux considéré.
Service strict.

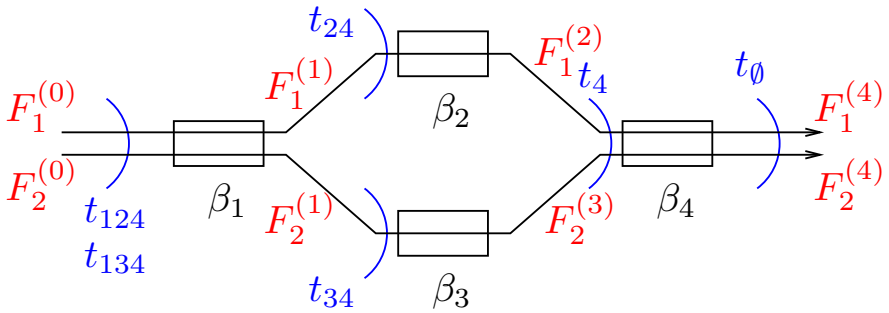
Outil

Construction de problèmes d'optimisation linéaire.
On ne cherche pas de courbe de service globale.

Analyses des réseaux sans dépendances cycliques

Constuction des programmes linéaires

Le flux principal a un départ et une *arrivée*.



Variables LP (pas des fonctions !)

- Des variables temporelles t_π : instants auxquels le pire cas se produit pour chaque chemin π menant à l'arrivée. Nombreuses !
- Des variables représentant le trafic cumulé en entrée/sortie de chaque serveur pour chaque flux transverse *aux temps* t_π. Idem pour le flux principal.

Analyses des réseaux sans dépendances cycliques

Contraintes (linéaires)

Etant donné un ordre total sur les t_π, on a :

Contraintes linéaires

- Courbe d'arrivée
- Service strict
- Périodes chargées
- Cohérence des flux
- Fonctions croissantes

Utilise fortement les hypothèses de service strict, courbes affines par morceaux concaves/convexes.

Restent les contraintes de temps

A priori, pas des contraintes linéaires.

Analyses des réseaux sans dépendances cycliques

Gestion des contraintes temporelles

Un nombre potentiellement exponentiel de variables t_π.

Contraintes pas linéaires

Il faudrait un ordre total sur les t_π.
On a un ordre partiel.

Plusieurs programmes linéaires

On génère un nouveau programme linéaire pour
chaque ordre total sur les t_π compatible avec l'ordre partiel.

Un nombre de tris topologiques
potentiellement exponentiel en le nombre de t_π !

Analyses des réseaux sans dépendances cycliques

Premier algorithme qui donne une borne exacte

Ces programmes correspondent à de vraies trajectoires.

Borne sur le délai (ou sur la charge)

On obtient le pire cas exact.

La complexité est énorme !

Oui, mais résultat de NP-complétude (X3C).

Cas particulier

Si l'ordre "partiel" est déjà total : algorithme polynomial.

Accélération

On peut supprimer des contraintes (temporelles !) pour limiter le nombres de programmes : les bornes ne sont plus exactes.

Publications
Conférences internationales


Anne Bouillard, Laurent Jouhet, and Éric Thierry.
Computation of a $(min, +)$ mutli-dimensional convolution for end-to-end performance analysis.
VALUETOOLS'2008.


Anne Bouillard, Laurent Jouhet, and Éric Thierry.
Comparison of different classes of service curves in network calculus.
WODES'2010.


Anne Bouillard, Laurent Jouhet, and Éric Thierry.
Tight performance bounds in the worst-case analysis of feed forward networks.
INFOCOM'2010.

Publications


Anne Bouillard, Marc Boyer, and Laurent Jouhet.
Notations pour le calcul réseau.
MSR'09.


Anne Bouillard, Laurent Jouhet, and Éric Thierry.
End-to-end performance guarantees for multipath flows.
Rapport HAL hal-00289106, juin 2008.


Anne Bouillard, Laurent Jouhet, and Éric Thierry.
Service curves in network calculus : dos and don'ts.
Rapport HAL inria-00431674, novembre 2009.

Perspectives

Cycles

Les résultats classiques s'appliquent mal...

Politiques de services

Dans le cadre du multiplexage aveugle.

Merci à tous !

Questions

Des questions scientifiques ?

Merci à tous !

Déjeuner

Salle passerelle : 4ème étage.

Ascenseur de l'autre côté.

Tout le monde est bienvenu !

NC vs RTC

Equations vérifiées pour RTC-GP et NC-VCN

RTC Greedy Processor (Wandeler)

$(A7)\ R'[s,t) = C[s,t) - C'[s,t)$

$(A8)\ C'[s,t) = \sup_{s \le u \le t} \Big[C[s,u) - R(s,u) - B(s), 0 \Big]$

$(A10)\ B(t) - B(s) = R[s,t) - R'[s,t)$

NC Variable Capacity Node (Thiele)

$(B1)\ R'(t) = \inf_{0 \le u \le t} \Big[C(t) - C(u) + R(u) \Big]$

$(B2)\ C'(t) = C(t) - R'(t)$

$(B3)\ B(t) = R(t) - R'(t)$