

Le calcul réseau dans un contexte embarqué

Marc Boyer

Equipe IRT
IRIT - UMR 5505
ENSEEIH

Séminaire IRCCyN – 6 décembre 2007

Première partie : Le calcul réseau

Introduction

Concepts de base de calcul réseau

NC et le dioid $(\wedge, +)$

Agrégation

Seconde partie : Faire du calcul réseau pour les systèmes embarqués

Introduction

Des résultats sans trop d'effort

Des résultats en réfléchissant un peu

Des résultats en contribuant

Conclusion

Première partie I

Le calcul réseau

Plan

Introduction

Concepts de base de calcul réseau

Flux

Courbe d'arrivée

Courbe de service

NC et le dioid $(\wedge, +)$

Le calcul réseau se plonge dans $(\wedge, +)$...

... et offre simplement des résultats importants

Agrégation

Qu'est-ce que le calcul réseau ?

- ▶ Une théorie pour obtenir des bornes garanties de délais et mémoire dans les réseaux.
Ce qui suppose
 - ▶ des garanties de débit/service minimal
 - ▶ des contraintes de trafic maximum
- ▶ Une théorie basée sur le semi-groupe $(\min, +)$.
- ▶ Historiquement développée pour réseaux Internet et ATM.

Plan

Introduction

Concepts de base de calcul réseau

Flux

Courbe d'arrivée

Courbe de service

NC et le dioid $(\wedge, +)$

Le calcul réseau se plonge dans $(\wedge, +)$...

... et offre simplement des résultats importants

Agrégation

Plan

Introduction

Concepts de base de calcul réseau

Flux

Courbe d'arrivée

Courbe de service

NC et le dioid $(\wedge, +)$

Le calcul réseau se plonge dans $(\wedge, +)$...

... et offre simplement des résultats importants

Agrégation

Flux

- ▶ On représente un flux par son *trafic cumulé* R .
 $R(t)$: nb de bits total envoyé par le flux sur $[0, t]$.
- ▶ Différent de la modélisation habituelle : débit instantané r .

$$R(t) = \int_0^t r(u) du$$

- ▶ Évidemment : $r(t) \geq 0 \Rightarrow R \nearrow$

Exemple de flux

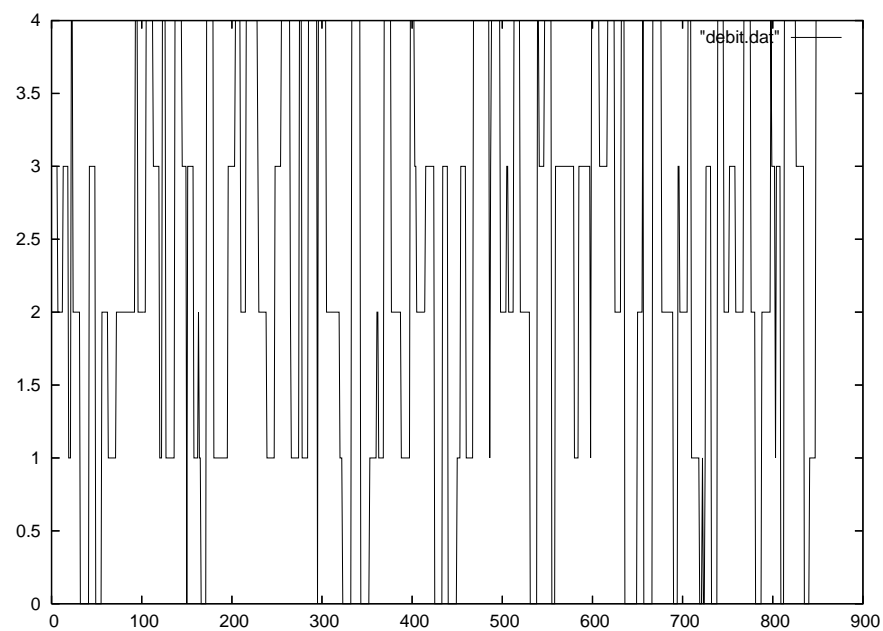


FIG.: Débit instantané r

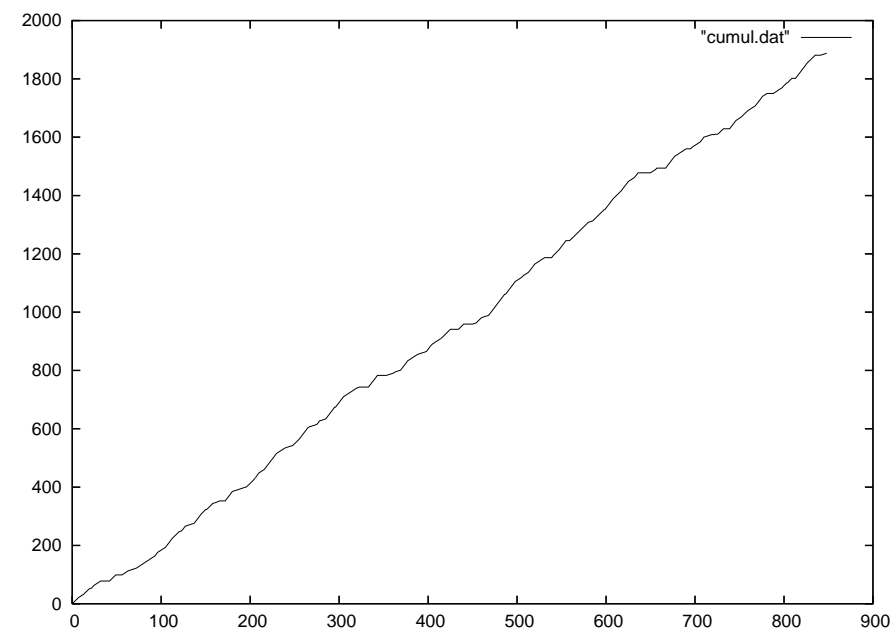


FIG.: Trafic cumulé R

Plan

Introduction

Concepts de base de calcul réseau

Flux

Courbe d'arrivée

Courbe de service

NC et le dioid $(\wedge, +)$

Le calcul réseau se plonge dans $(\wedge, +)$...

... et offre simplement des résultats importants

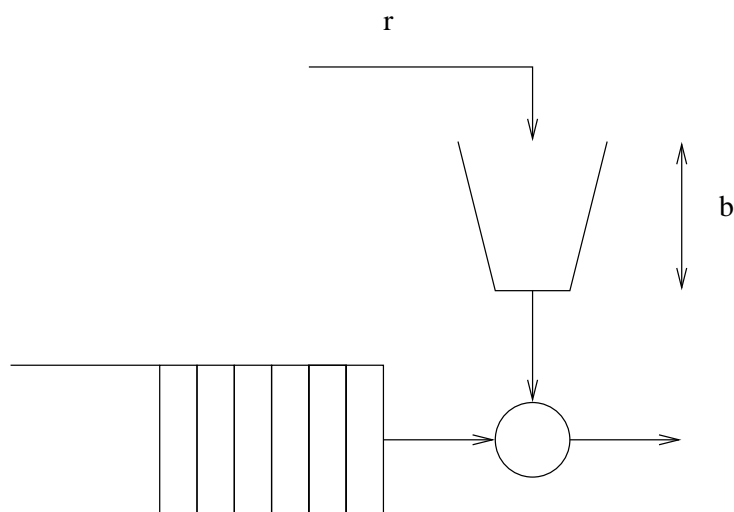
Agrégation

Caractériser un flux : courbe d'arrivée

- ▶ La courbe R représente exactement le flux.
 - ▶ On veut une approximation manipulable (ex : seau percé)
- ⇒ notion de *courbe d'arrivée* α

$$\forall t, s : R(t + s) - R(t) \leq \alpha(s)$$

- ▶ Indépendante de l'instant considéré ($\forall t$)
- ▶ Exemple : le seau percé



$$\gamma_{r,b}(t) = \begin{cases} 0 & \text{si } t \leq 0 \\ rt + b & \text{sinon} \end{cases}$$

FIG.: Le seau percé

Plan

Introduction

Concepts de base de calcul réseau

Flux

Courbe d'arrivée

Courbe de service

NC et le dioid $(\wedge, +)$

Le calcul réseau se plonge dans $(\wedge, +)$...

... et offre simplement des résultats importants

Agrégation

Serveur, élément réseau

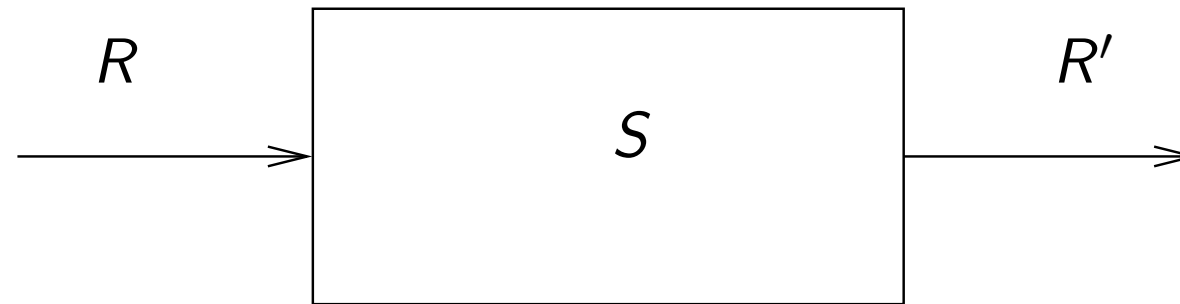


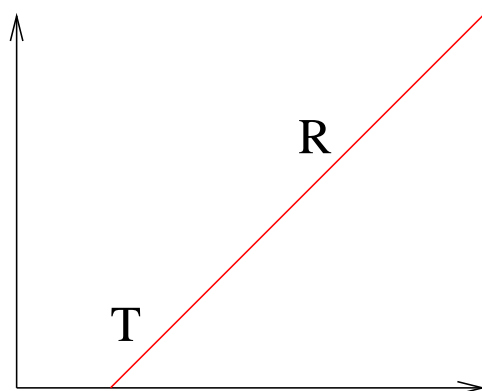
FIG.: Notion de serveur

- ▶ Un serveur S offre une *courbe de service* β ssi

$$\forall R, \forall t, \exists s : R'(t + s) - R(t) \geq \beta(s) \quad (1)$$

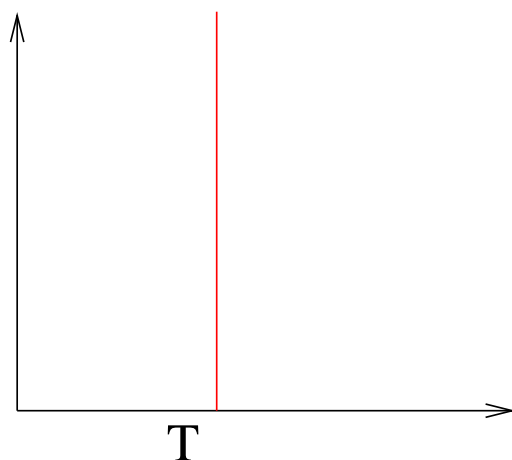
Exemples de serveur classique

► *rate-latency*



$$\beta_{R,T}(t) = R[t - T]^+ = \begin{cases} 0 & \text{si } t \leq T \\ R(t - T) & \text{sinon} \end{cases}$$

► délais garanti



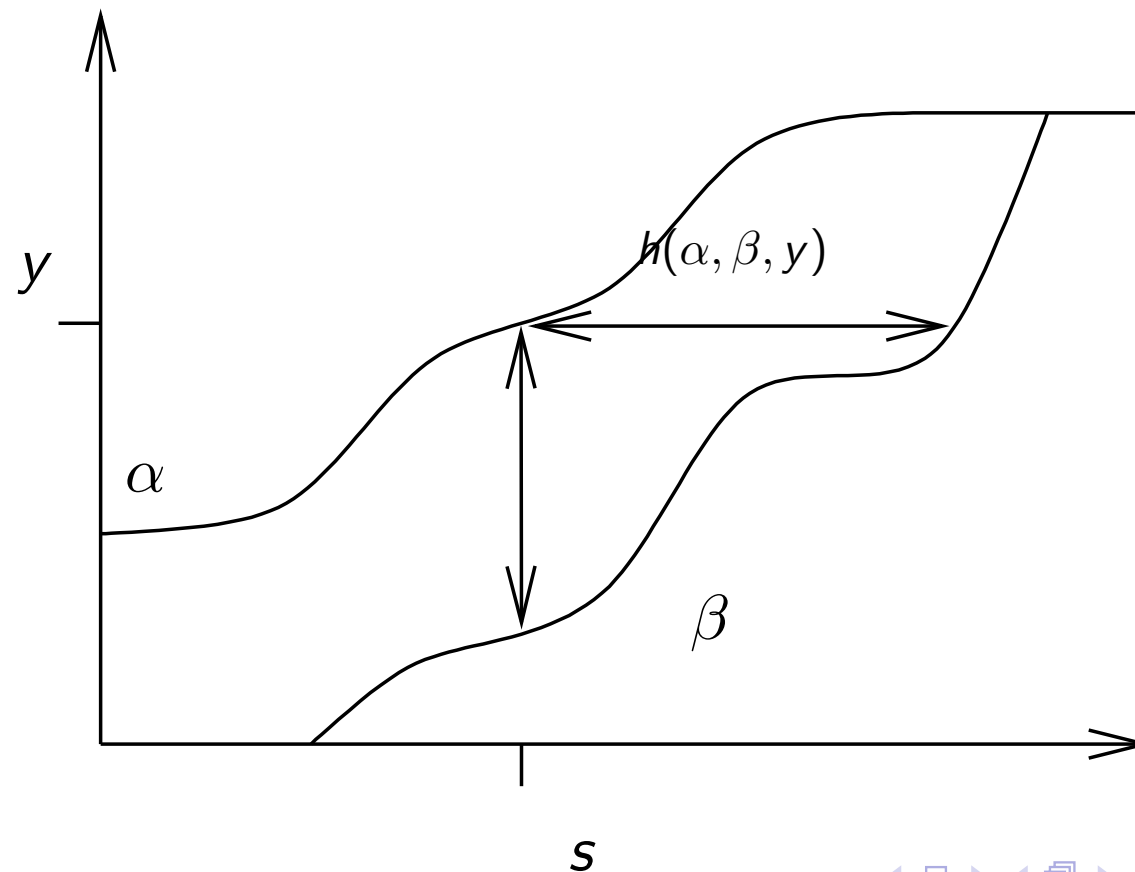
$$\delta_T(t) = \begin{cases} 0 & \text{si } t \leq T \\ \infty & \text{sinon} \end{cases}$$

Bornes du calcul réseau

Si on a un flux de courbe d'arrivée α traversant un serveur de courbe de service β , on peut borner le délais d et la file d'attente x .

$$d \leq \sup_y h(\alpha, \beta, y)$$

$$x \leq \sup_s \alpha(s) - \beta(s)$$



Plan

Introduction

Concepts de base de calcul réseau

Flux

Courbe d'arrivée

Courbe de service

NC et le dioid $(\wedge, +)$

Le calcul réseau se plonge dans $(\wedge, +)$...

... et offre simplement des résultats importants

Agrégation

Plan

Introduction

Concepts de base de calcul réseau

Flux

Courbe d'arrivée

Courbe de service

NC et le dioid $(\wedge, +)$

Le calcul réseau se plonge dans $(\wedge, +)$...

... et offre simplement des résultats importants

Agrégation

Le dioid $(\wedge, +)$

- ▶ Ensemble considéré : $\mathbb{R} \cup \{\infty\}$
- ▶ Deux opérateurs : \wedge , minimum et $+$, addition
- ▶ Trois autres opérateurs : convolution \otimes , déconvolution \oslash et fermeture sous-additive $\bar{\cdot}$

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

$$(f \oslash g)(t) = \sup_{0 \leq s} f(t+s) - g(s)$$

$$\bar{f} = \delta_0 \wedge f \wedge (f \otimes f) \wedge (f \otimes f \otimes f) \wedge \dots$$

Les résultats de calcul réseau se traduisent dans $(\wedge, +)$

- ▶ α est une courbe d'arrivée pour R

$$R \leq R \otimes \alpha$$

- ▶ le serveur S offre une courbe de service β

$$R' \geq R \otimes \beta$$

- ▶ les bornes se calculent

$$x \leq (\alpha \oslash \beta)(0)$$

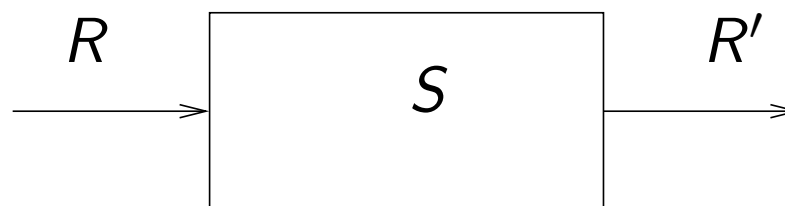
$$d \leq (\beta^{-1}(\alpha) \oslash \lambda_1)(0)$$

avec $\lambda_R = \beta_{R,0}$.

Et en offre d'autres

- ▶ Possibilité d'ajuster les contraintes : si α est une courbe d'arrivée de R , alors $\bar{\alpha}$ l'est aussi (et $\bar{\alpha} \leq \alpha$).
- ▶ Possibilité de *propager* les calculs.
Si R de courbe d'arrivée α traverse le serveur S de courbe service β , alors le flux de sortie R' a pour courbe d'arrivée α'

$$\alpha' = \alpha \otimes \beta$$



Plan

Introduction

Concepts de base de calcul réseau

Flux

Courbe d'arrivée

Courbe de service

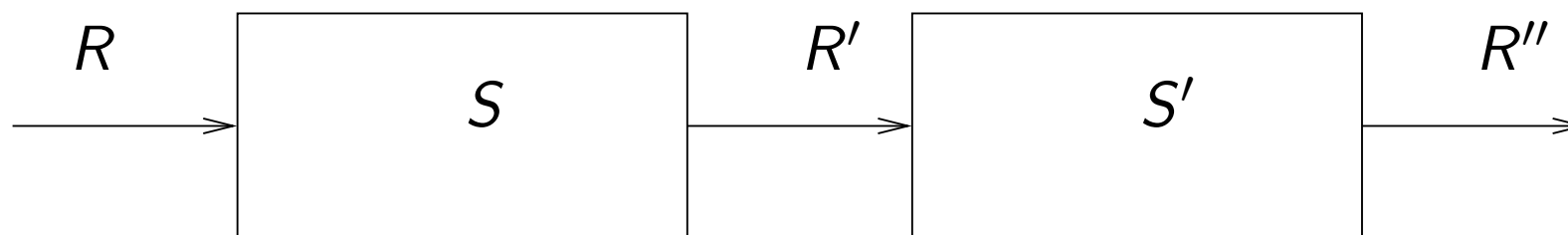
NC et le dioid $(\wedge, +)$

Le calcul réseau se plonge dans $(\wedge, +)$...

... et offre simplement des résultats importants

Agrégation

Pay bust only once



- ▶ Célèbre résultat de calcul réseau : la concaténation de serveurs de service β et β' est équivalente à un seul serveur de service $\beta \otimes \beta'$.
- ▶ Le délais de bout en bout est plus petit que la somme des délais locaux.
- ▶ Preuve triviale en calcul réseau.

$$R'' \geq R' \otimes \beta' \geq (R \otimes \beta) \otimes \beta' = R \otimes (\beta \otimes \beta')$$

Plan

Introduction

Concepts de base de calcul réseau

Flux

Courbe d'arrivée

Courbe de service

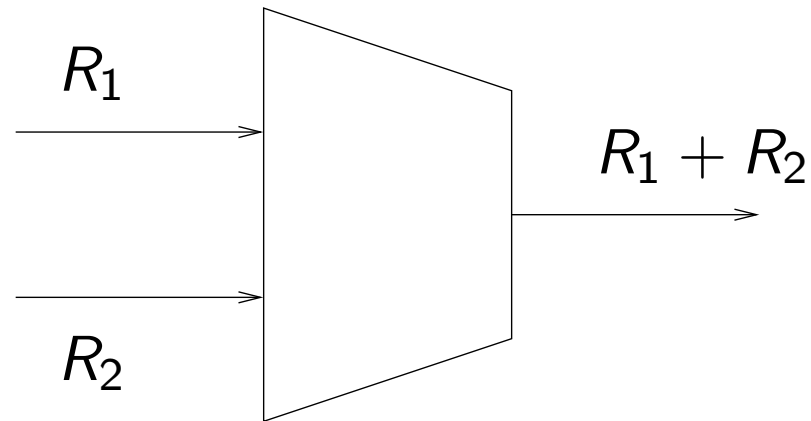
NC et le dioid $(\wedge, +)$

Le calcul réseau se plonge dans $(\wedge, +)$...

... et offre simplement des résultats importants

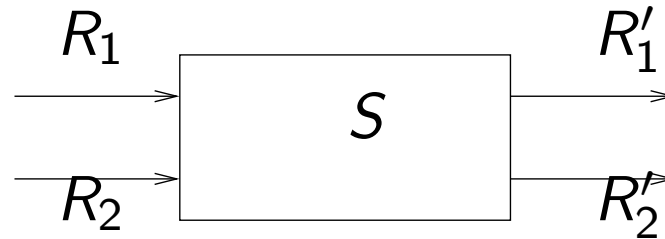
Agrégation

Multiplexeur



- ▶ Somme de deux flux : $R_1 + R_2$
- ▶ Courbe d'arrivée : $\alpha_{1,2} = \alpha_1 + \alpha_2$

Serveur partagé



- ▶ On peut facilement avoir $R'_1 + R'_2$.
- ▶ Mais comment connaître le service S_i offert à chaque flux ?
- ▶ Dépend bien sûr de la politique de service.

- ▶ Dans la suite, on suppose que R_i a pour courbe d'arrivée α_i et S offre la courbe de service β . On note $\{i, j\} = \{1, 2\}$.

Priorité non interruptible

On suppose que R_1 est strictement plus prioritaire que R_2 , et qu'on ne peut pas interrompre l'émission d'un message.

Si l_2^M est la taille maximale d'un message du flux R_2 , alors on a :

$$\beta_1 = [\beta - l_2^M]^+$$

$$\beta_2 = [\beta - \alpha_1]^+$$

sous réserve que β_2 soit une fonction croissante (ie $\beta > \alpha_1$).

Politique inconnue

Si on ne connaît pas la politique, et si β est une courbe de service stricte on peut tout de même déduire.

$$\beta_i = [\beta - \alpha_j]^+$$

sous réserve que β_i soit une fonction croissante.

Politique FIFO

Si le serveur applique une politique FIFO, on a, pour tout $\theta > 0$, le serveur S offre au flux R_i le service

$$\beta_i^\theta(t) = [\beta(t) - \alpha_j(t - \theta)]^+ 1_{t > \theta}$$


avec $1_{t > \theta} = 0$ si $t \leq \theta$, et $1_{t > \theta} = 1$ sinon.


De plus, on peut calculer plus finement la courbe d'arrivée en sortie.


$$\alpha'_i = \inf_{0 \leq \theta} \alpha_i \oslash \beta_i^\theta$$

Bibliographie I

 Cheng-Shang Chang.
Performance Guarantees in communication networks.
Telecommunication Networks and Computer Systems. Springer, 2000.

 Rene L. Cruz.
A calculus for network delay, part I : Network elements in isolation.
IEEE Transactions on information theory, 37(1) :114–131, January 1991.

 Rene L. Cruz.
A calculus for network delay, part II : Network analysis.
IEEE Transactions on information theory, 37(1) :132–141, January 1991.

 Jean-Yves Le Boudec and Patrick Thiran.
Network Calculus, volume 2050 of *LNCS*.
Springer Verlag, 2001.
[http ://lrcwww.epfl.ch/PS_files/NetCal.htm](http://lrcwww.epfl.ch/PS_files/NetCal.htm).

Deuxième partie II

Faire du calcul réseau pour les systèmes embarqués

Plan

Introduction

Pourquoi faire du NC ?

Exemple : AFDX

Prenons un peu de recul

Des résultats sans trop d'effort

Des résultats en réfléchissant un peu

Des résultats en contribuant

Conclusion

Plan

Introduction

Pourquoi faire du NC ?

Exemple : AFDX

Prenons un peu de recul

Des résultats sans trop d'effort

Des résultats en réfléchissant un peu

Des résultats en contribuant

Conclusion

Pourquoi faire du NC ?

- ▶ Médium ou serveur partagé
- ▶ Existence de non déterminisme
- ▶ Absence de synchronisation globale
- ▶ Besoin de borner les pires cas
- ▶ Flux de données connus ou constraints
- ▶ Performances du médium ou serveur connues

Sinon, autres méthodes :

- ▶ ordonnancement
- ▶ model-checking
- ▶ ...

Plan

Introduction

Pourquoi faire du NC ?

Exemple : AFDX

Prenons un peu de recul

Des résultats sans trop d'effort

Des résultats en réfléchissant un peu

Des résultats en contribuant

Conclusion

AFDX

▶ Principes

- ▶ Réseau fédérateur avionique
- ▶ Basé sur technologie Ethernet
- ▶ Liens *full-duplex* \Rightarrow plus de collision à l'accès
- ▶ Présence de *switch* \Rightarrow indéterminisme
- ▶ Notion de *Virtual Link* : chemin de niveau 2
 - ▶ mono-émetteur, multi-récepteur
 - ▶ taille min et max de trame
 - ▶ délais minimum entre trame

▶ Exemple : A380

- ▶ 8 switch, \approx 200 ports
- ▶ \approx 500 VL, \approx 3500 chemins

Plan

Introduction

Pourquoi faire du NC ?

Exemple : AFDX

Prenons un peu de recul

Des résultats sans trop d'effort

Des résultats en réfléchissant un peu

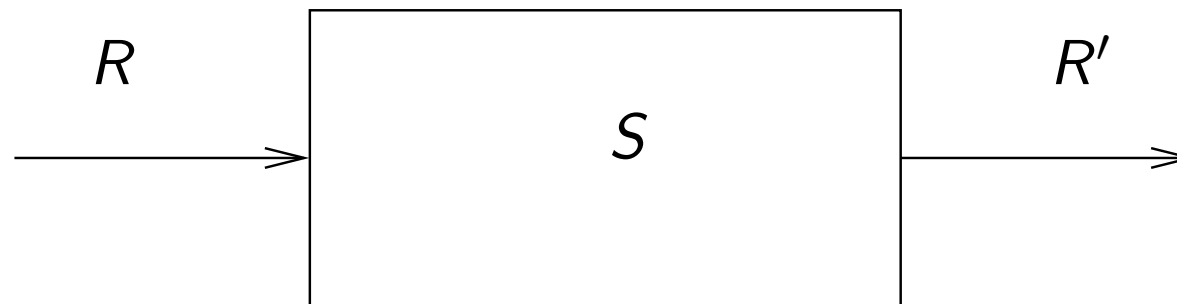
Des résultats en contribuant

Conclusion

Et si on veut en faire du NC...

- ▶ Le NC présenté par LE BOUDEC et THIRAN, c'est...
 - ▶ beau
 - ▶ facile
 - ▶ ça donne de bons résultats
- ▶ Mais quand il faut s'y mettre
 - ▶ il y a plein de formules
 - ▶ il y a deux bons bouquins [LBT01, Cha00]
 - ▶ il faut trouver la bonne modélisation

Retour sur les hypothèses



Ce schéma de base fonctionne si :

- ▶ on a un seul flux en entrée
- ▶ on est capable de dire quel service lui sera offert
- ▶ on est capable de lui donner une enveloppe de trafic
- ▶ on est capable de faire des calculs dans $min, +$ sur α et β ($\alpha \otimes \beta$)

Retour sur les résultats

Si on a tout ça, on peut calculer des bornes sur :

- ▶ le délais : $d(\alpha, \beta) \geq d(R, S)$,
- ▶ la taille des buffers : $x(\alpha, \beta) \geq x(R, S)$.

Ces bornes sont *les plus justes possibles* c-à-d : avec α et β donnés, on peut construire R_α et S_β tel que :

$$h(\alpha, \beta) = h(R_\alpha, S_\beta) \geq h(R, S)$$

$$x(\alpha, \beta) = x(R_\alpha, S_\beta) \geq x(R, S)$$

Donc, les bornes ne sont bonnes que si les courbes sont proches de la réalité.

Est-on oblig  de toucher aux formules ?

► Convolution :

$$(f \otimes g)(t) = \inf_{0 \leq u \leq t} (f(t - u) + g(u))$$

Se discr tise bien :

```
double delta= (t - u ) / 1000 ;  
double res= f(t) + g(0);  
for(double u= 0.0 ; u <= t ; u+= delta) {  
    res= min(res, f(t-u) + g(u) );  
}
```

Un peu de code en Scilab.

► D convolution :

$$(f \oslash g)(t) = \sup_{0 \leq u \leq \infty} (f(t + u) - g(u))$$

Nettement moins facile   coder...

Plan

Introduction

Pourquoi faire du NC ?

Exemple : AFDX

Prenons un peu de recul

Des résultats sans trop d'effort

Des résultats en réfléchissant un peu

Des résultats en contribuant

Conclusion

Première approche

Reprendre les résultats déjà étudiés :

- ▶ flux : seau percés
- ▶ serveur : *rate-latency*

Problèmes :

- ▶ le schéma de base ne s'applique pas (serveur partagé)
- ▶ on peut appliquer le “blind multiplexing”
- ▶ les résultats sont très mauvais ($\approx 10x$ plus que pire cas mesuré)
- ▶ problème d'ordonnancement des calculs

Mais on a quand même une borne, *certifiable*.

Plan

Introduction

Pourquoi faire du NC ?

Exemple : AFDX

Prenons un peu de recul

Des résultats sans trop d'effort

Des résultats en réfléchissant un peu

Des résultats en contribuant

Conclusion

Sans faire de calcul

- ▶ Il faut bien comprendre ce qu'on modélise.
- ▶ Comprendre d'où viennent les sur-approximations

Sans faire de calcul

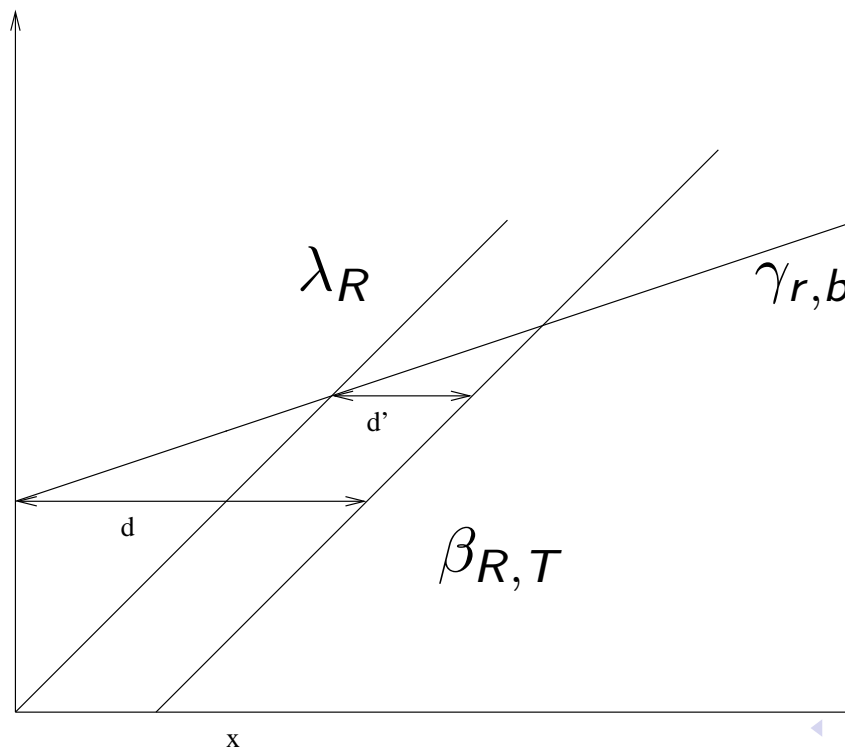
- ▶ Il faut bien comprendre ce qu'on modélise.
- ▶ Comprendre d'où viennent les sur-approximations
- ▶ ... ou tester au hasard

Sans faire de calcul

- ▶ Il faut bien comprendre ce qu'on modélise.
- ▶ Comprendre d'où viennent les sur-approximations
- ▶ ... ou tester au hasard

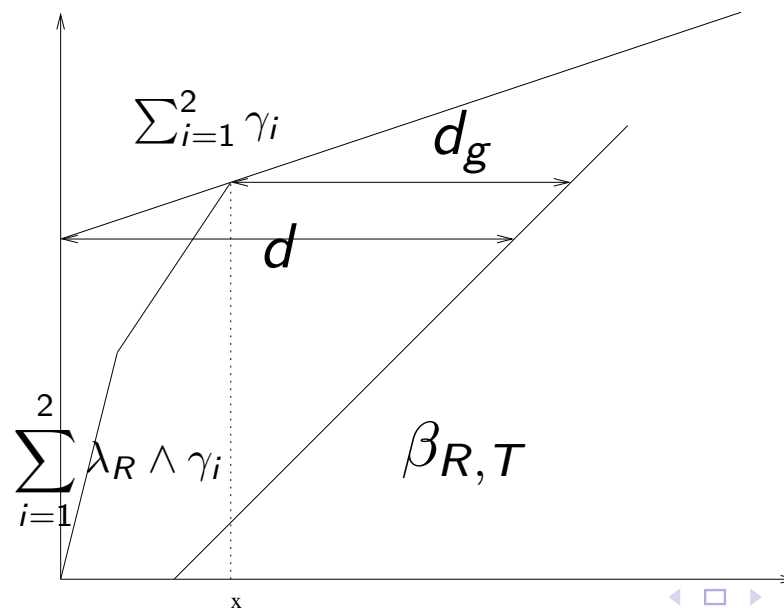
Exemple : AFDX (Thèse J. Grieu)

- ▶ Une rafale est toujours limitée par le débit du lien
- ▶ donne délais $d' \ll d$



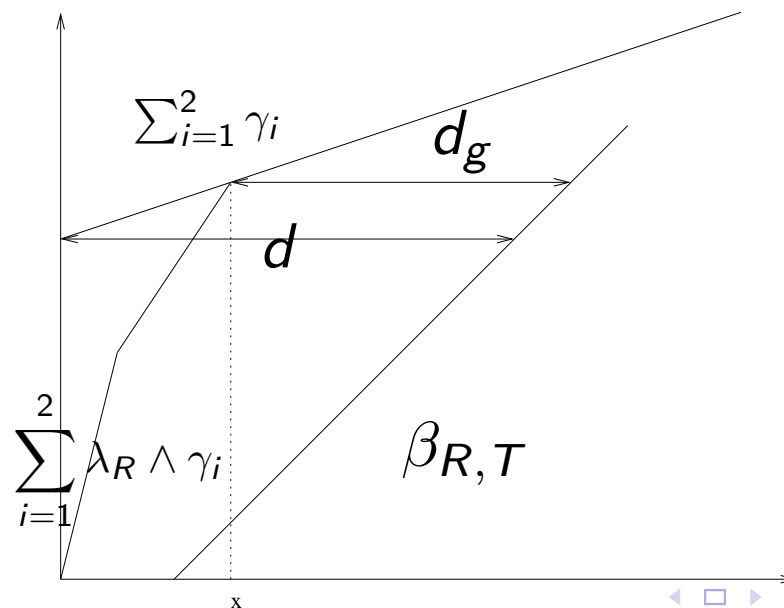
Mais un peu de code

- ▶ la somme de courbes de la forme $\lambda_R \wedge \gamma_{r,b}$ est de la forme $\bigwedge_{i=1}^n \gamma_{r_i,b_i}$
- ▶ on peut calculer ces sommes
- ▶ on peut calculer les délais d entre $\bigwedge_{i=1}^n \gamma_{r_i,b_i}$ et $\beta_{R,T}$ mais ...



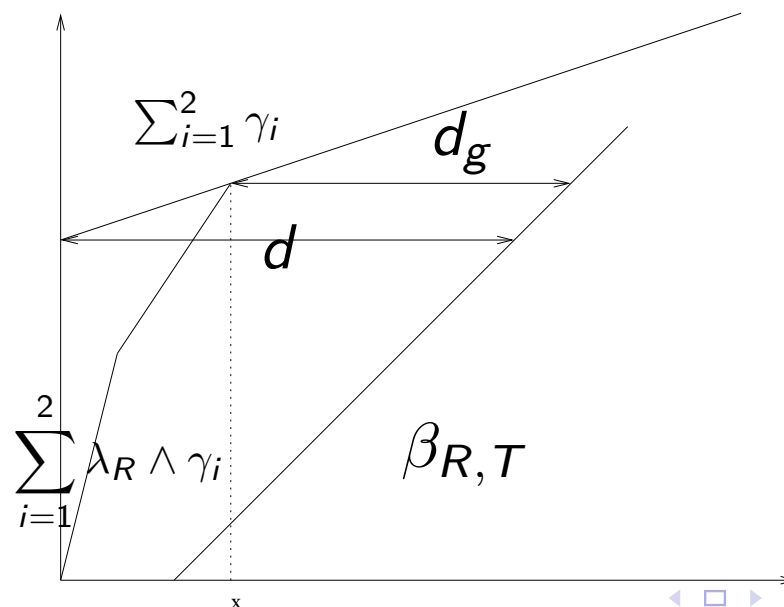
Mais un peu de code

- ▶ la somme de courbes de la forme $\lambda_R \wedge \gamma_{r,b}$ est de la forme $\bigwedge_{i=1}^n \gamma_{r_i,b_i}$
- ▶ on peut calculer ces sommes
- ▶ on peut calculer les délais d entre $\bigwedge_{i=1}^n \gamma_{r_i,b_i}$ et $\beta_{R,T}$
- ▶ on ne sait pas calculer $(\bigwedge_{i=1}^n \gamma_{r_i,b_i}) \ominus \beta_{R,T}$
ie on ne sait pas propager le calcul



Mais un peu de code

- ▶ la somme de courbes de la forme $\lambda_R \wedge \gamma_{r,b}$ est de la forme $\bigwedge_{i=1}^n \gamma_{r_i,b_i}$
- ▶ on peut calculer ces sommes
- ▶ on peut calculer les délais d entre $\bigwedge_{i=1}^n \gamma_{r_i,b_i}$ et $\beta_{R,T}$
- ▶ on ne sait pas calculer $(\bigwedge_{i=1}^n \gamma_{r_i,b_i}) \otimes \beta_{R,T}$
ie on ne sait pas propager le calcul
- ▶ mais on peut quand même dire (sous hypothèse FIFO) que chaque flux subit au pire un délais d , ie reçoit un service δ_d et donc, on calcule $(\lambda_R \wedge \gamma_{r,b}) \otimes \delta_T$.



Plan

Introduction

Pourquoi faire du NC ?

Exemple : AFDX

Prenons un peu de recul

Des résultats sans trop d'effort

Des résultats en réfléchissant un peu

Des résultats en contribuant

Conclusion

Mon expérience

- ▶ environ un an pour s'y mettre
- ▶ deux bouquins : [LBT01, Cha00]
- ▶ développement d'un embryon d'outil
- ▶ rattrapé par équipe INRIA : ARC COINC
- ▶ calcul de deux formules

$$\overline{\bigwedge \gamma_i \odot \beta_{R,T}}$$
$$\inf_{0 \leq \theta} \bigwedge \gamma_i \odot \beta_i^\theta$$

Plan

Introduction

Pourquoi faire du NC ?

Exemple : AFDX

Prenons un peu de recul

Des résultats sans trop d'effort

Des résultats en réfléchissant un peu

Des résultats en contribuant

Conclusion

Pour conclure

Le calcul réseau :

- ▶ une technique “nouvelle”
- ▶ initialement développés pour Internet, ATM, QoS
- ▶ adaptée aux systèmes embarqués “complexes”
- ▶ pas d’outil (pour le moment)
- ▶ (presque) facile d’avoir des résultats
- ▶ demande une peu de travail pour avoir de bons résultats
- ▶ communauté française qui se constitue