Scheduling

Christophe Deleuze
Laboratoire d'Informatique de Paris 6
Université Pierre et Marie Curie
Paris - France

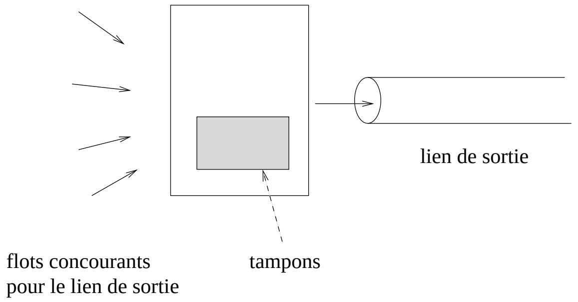
Christophe.Deleuze@lip6.fr

16 février 1998

Plan

Rôle du scheduler

- Devant un lien de sortie
- rôle : multiplexage
- réalise une allocation
  - en temps
  - en espace



Commutation de circuits/de paquets

Commutation de circuits

- *Time Division Multiplexing* (TDM)
- partage statique
- isolation complète

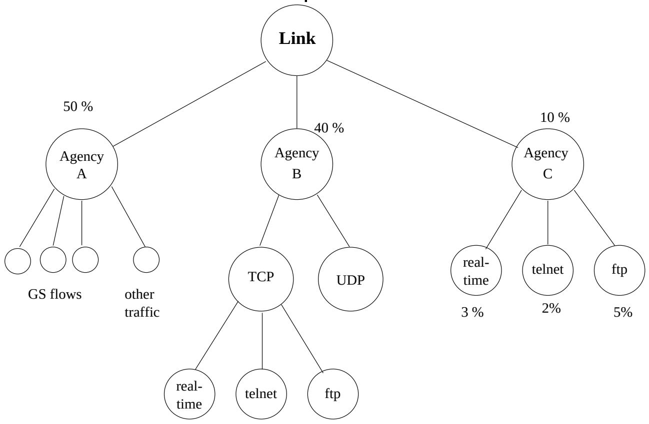
Commutation de paquets

- très grande flexibilité
- partage dynamique (au niveau du paquet)
- pas d'isolation

(+ Commutation de cellules)

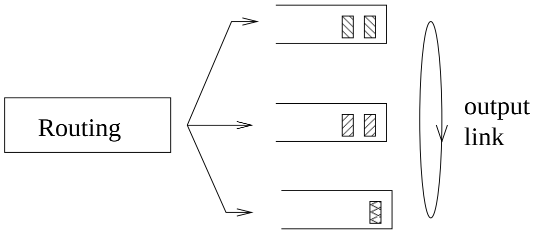
Intégration de services

- plusieurs classes de service
- Garanties dures
- partage de lien (*link-sharing*)
- $\rightarrow$ modèle hiérarchique



Un début de solution

- proposition de Nagle (1987)
- files séparées $\rightarrow$ service *round robin*
- sémantique d'un flot
- pb : équité comptée en paquets



Fair Queueing

- émulation d'un round robin *bit-à-bit*
- paquets triés sur la date d'émission de leur dernier bit

$$t_d = t_p + L_{paquet} \times t_{round}$$

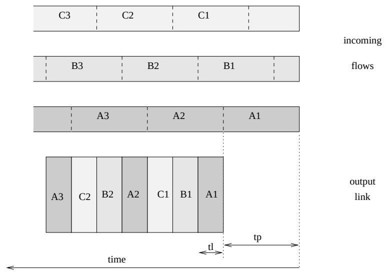
- $t_{round} = \frac{N(t)}{C_l}$ unité du "temps virtuel"
- en temps virtuel

$$t_d = t_p + L_{paquet}$$

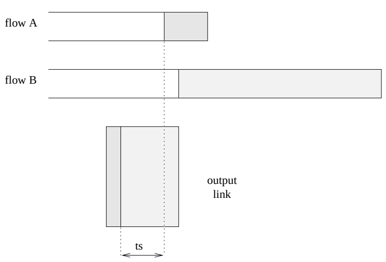
- en temps virtuel, le service reçu par chaque flot est constant (1 bit par unité de temps)
- pb : calcul de t_{round} en $\theta(nbflots)$
- approche du *modèle fluide*

Qualité de l'émulation

Effet de *paquetisation* $t_p = \frac{L}{R}$



Effet de *serialisation* $t_s = \frac{L_{max}}{C_l}$



Garanties de QoS

Garantie de débit

- le poids w_i affecté au flot i correspond à un débit relatif $=$ *portion* de la bande passante

$$B_i = C_l \times \frac{w_i}{\sum w_j}$$

- la limitation de la somme des poids permet une garantie absolue

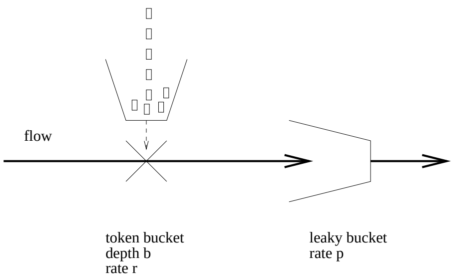
$$\sum w_j \leq 1 \Longrightarrow B_i \geq C_l \times w_i$$

La **garantie de délai** est directement liée à la garantie de débit.

Exemple : classe GS IntServ (1)

Caractérisation du flot

- TSPEC = (b, r, p)
- token bucket : débit moyen, taille max de rafale
- leaky bucket : débit crête



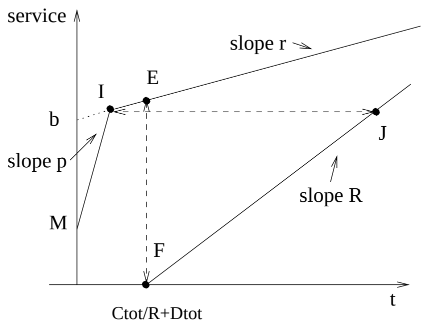
Exemple : classe GS IntServ (2)

Caractérisation du chemin

- Chaque nœud implémente une émulation du *modèle fluide*
- 2 termes d'erreur généraux :
  - C erreur "interne" au flot ($\rightarrow$ paquetisation)
  - D erreur "externe" au flot ($\rightarrow$ serialisation)
- Ces deux valeurs sont accumulées dans C_{tot} et D_{tot} qui caractérisent l'ensemble du chemin

Exemple : classe GS IntServ (3)

- enveloppe du flot
- enveloppe du service



- IJ délai max
- EF tampon max

Complexité

Algo de base

- calcul de l'estampille en θ(nb flots)
- tri des paquets en θ(log nb paquets)

La complexité est essentiellement liée à la fonction de temps virtuel

Autres algos

- on peut *estimer* la fonction
- meilleures solutions en $\theta(1)$!

Différentes implémentations

- WFQ est une famille nombreuse !
- quatre critères d'évaluation :
  - complexité
  - délais ajoutés (pour garanties dures)
  - déviation du modèle (pour partage de lien)
  - adéquation pour une utilisation hiérarchique
- l'algorithme idéal pour un réseau à intégration de service doit être performant dans les 4 aspects

Et la gigue ?

- *fair queueing* est un algo *work-conserving*
- *work-conserving* = pas de contrôle de gigue possible
- création de distortion
- algos *non work-conserving* peuvent conserver la forme d'un flot.
- mais augmentent les délais

Avec une garantie de délai, on peut corriger la gigue à l'arrivée (tampons).

Conclusions

- WFQ est un algo polyvalent
  - partage de liens
  - garantie de QoS
  - combinaison hiérarchique
- répond bien aux besoins d'un modèle à intégration de service
- sa complexité est maintenant raisonnable
- Problème de "scalabilité" $\rightarrow$ agrégation