Le multimédia dans l'Internet

—————

La qualité de service (QoS)

Christophe Deleuze

`Christophe.Deleuze@activia.net`

DESS Télécom

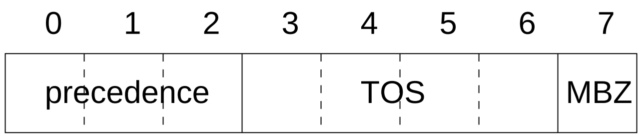
14 janvier 2003

La qualité de service

- QoS "buzzword"
- chacun y met ce qu'il veut
- Internet ➠ QoS ≠ *best-effort*
- qualité perçue
- multimédia
  - ✓ délais
  - ✓ pertes
  - ✓ débits trop faibles

Aux débuts de l'Internet

- octet TOS IPv4
  - ✓ priorité
  - ✓ type de service



1000 délai
0100 débit
0010 fiabilité
0001 coût

Valeurs recommandées

protocole	valeur	
TELNET	1000	délai
FTP		
contrôle	1000	délai
données	0100	débit
SMTP		
phase commande	1000	délai
phase données	0100	débit
DNS		
requête UDP	1000	délai
requête TCP	0000	
transfert de zone	0100	débit
NNTP	0001	coût

Avec IPv6

- Traffic class, 8 bits
  - ✓ TOS IPv4, non spécifié
  - ✓ applications peuvent le fixer, valeur défaut 0
  - ✓ routeurs peuvent le modifier
- Flow label, 20 bits
  - ✓ source identifie une séquence de paquets
  - ✓ si non supporté, mettre à zéro/passer inchangé
  - ✓ expérimental
    - ☞ flot = adresse src + flow label non zéro
    - ☞ à 0 si pas de flot

V	traffic class	Flow label

Prise en compte de la QoS dans le routage

Plan

- la QoS dans le routage
- techniques d'ordonnancement
- IntServ/RSVP
- DiffServ
- la QoS dans Linux

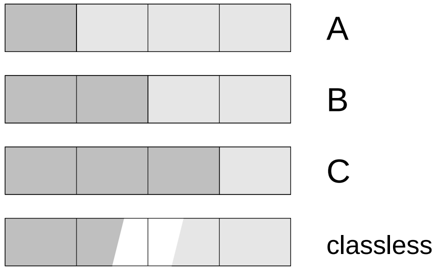
Rappel sur le routage

routage actuel “best effort”

- indique la connectivité
- une seule métrique
- un seul chemin (sauf *equal cost multipath routing*)
- hiérarchie intra/inter domaine

Rappel sur le relayage

- avant 1993, classes
- après, CIDR
- adresse/préfixe
- 192.168.10.0/18
- 192.168.12.0/24
- ➠ *longest prefix match*



Besoins et objectifs pour le routage QoS

- plus de métriques
- chemins multiples

- choix dynamique
- optimisation de l'usage du réseau *(traffic engineering)*
- dégradation de performance contenue

Questions...

- déterminer la QoS d'un lien
- granularité de la décision de routage
  - ✓ IP destination
  - ✓ IP src/dest
  - ✓ flot
- choix des métriques
- multi-métrique peut être très complexe
  ➠ *sequential filtering*

Routage sur le TOS : routage

- forme limitée de QoS routing
- routage à état des liens
  - ✓ collecter plusieurs métriques
  - ✓ calculer plusieurs routes
- possible avec OSPF
- granularité : destination
- pas de calculs complexes

Routage sur le TOS : relayage

une entrée = (`destination,TOS`) ➠ `next-hop`

1. recherche classique donne un ensemble de routes
2. si vide, destination injoignable
3. si TOS exact, terminé
4. si une route avec TOS 0, terminé
5. destination injoignable

Problèmes

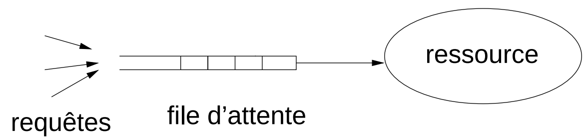
- multi-métrique
- dynamicité des états
- le multicast
- ...

- OK si contrainte simple (TOS)
- peu déployé (réseaux homogènes)

Techniques d'ordonnancement

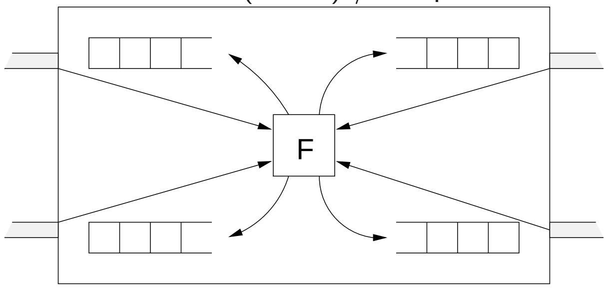
Multiplexage

- partage d'une ressource limitée
- discipline d'ordonnancement
  ✓ ordre de service (temps)
  ✓ file d'attente (espace)



Cas d'un routeur

- ressources multiplexées
  - ✓ bande passante
  - ✓ mémoire tampon
- First Come First Served (FCFS) / Drop Tail



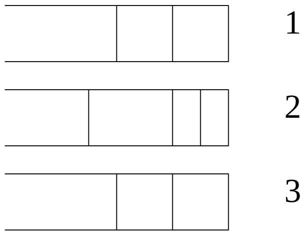
Conservation du travail

Work-conserving
- jamais au repos si des paquets attendent
- peut créer des déformations
- loi de conservation (délais moyens)

Non work-conserving
- peut être au repos même si des paquets attendent
- maintient l'aspect du trafic (gigue)
- augmente le délai moyen global
- limite le besoin de tampons

Priorités

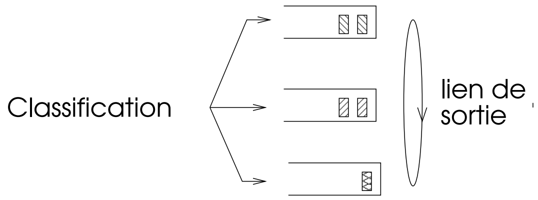
"multilevel priority with exhaustive service"



- protection unidirectionnelle
- famine ➠ CAC/poliçage aux priorités supérieures

Tourniquet (*Round robin*)

- proposition de Nagle (1987)
- files séparées ➠ service *round robin*
- pb : équité comptée en paquets
- définition d'un flot ?



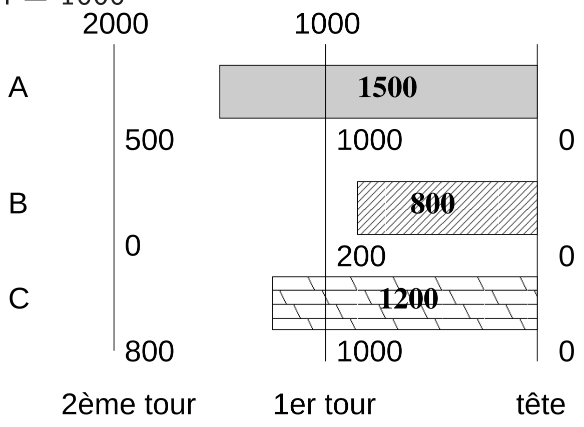
- isolation
- nb de flots limité ➠ garantie de BP

Deficit Round Robin

- compteur de déficit cd pour chaque flot
- initialisé à 0
- quantum de service q
- paquet de taille L
  - ✓ $L > q + cd$
    - ☞ $cd += q$
  - ✓ $L \leq q + cd$
    - ☞ sert le paquet
    - ☞ $cd += (q - L)$
- $cd = 0$ si pas de paquet dans la file
- équité à échelle de temps > cycle

Deficit Round Robin

- quantum = 1000



Partage équitable *(Fair queueing)*

- émulation d'un round robin *bit-à-bit*
- $R(t)$ nombre de cycles
- $N_{ac}(t)$ flots actifs
- paquet de taille L, commence à t_0 ➠ finit à $R(t) = R(t_0) + L$

$$F_i^k = S_i^k + L \quad \text{et} \quad S_i^k = max(F_i^{k-1}, R(t_i^k))$$

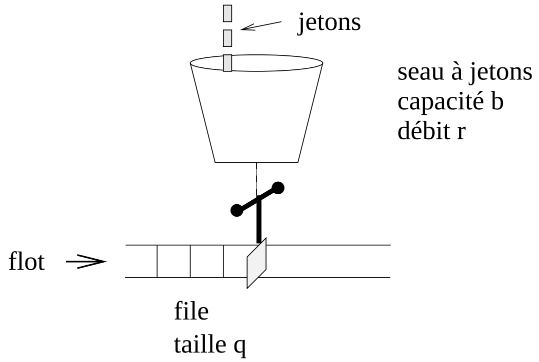
- paquets triés sur le numéro de cycle de leur dernier bit F_i^k
- retard max
  - ✓ délai $\frac{L_{max}}{C}$
  - ✓ service L_{max}

Implémenter WFQ

- on maintient **une seule** file
- un paquet P_i^k arrive :
  - ✓ classification
  - ✓ màj $R(t)$ – en $\theta(nb.flots)$
  - ✓ calcul de son estampille F_i^k – en $\theta(1)$
  - ✓ insertion dans la file – en $\theta(nb.paquets)$ ou $\theta(\log nb.paquets)$

Token bucket filter

- débit moyen
- capacité ➠ taille max de rafale
- taille de la file



Implémenter TB

- manière naive
- manière correcte

```
init : tokens = bucket_size, last_date = now

event : tokens += rate * (now - last_date)
        tokens = min(tokens, bucket_size)
        if packet_size < tokens, send packet,
                                 tokens-=packet_size
                                 last_date=now
        else alarm(now+(packet_size-tokens)*rate)

event = packet arrival on empty queue or alarm
```


Garanties

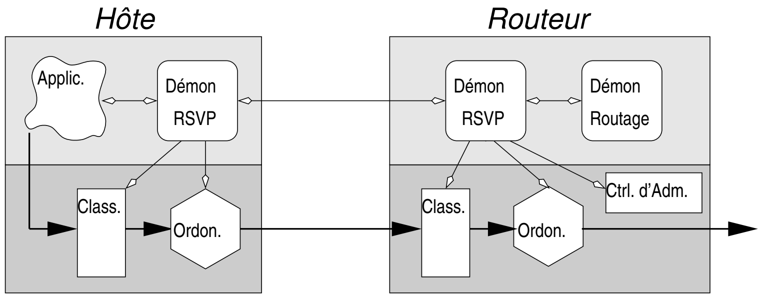
- isolation ➠ garanties
- débit
  - ✓ FQ garantit $\frac{1}{n}$ de la BP
  - ✓ si $n \leq n_{max}$, $b \geq \frac{1}{n_{max}}$
  - ✓ idem pour TB
- délai (dans les files) : débit consommé $<$ débit alloué
  ➠ OK !
- pertes (dans les files)
  - ✓ réservation d'un tampon

L'architecture IntServ/RSVP

IETF Integrated Services : IntServ

- intégration de services
  - ✓ réseaux spécialisés ➠ réseau intégré
- IETF ➠ (1992 – 2000)
- gestion **par flot**
  - ✓ caractériser le flot (TSPEC)
  - ✓ spécifier le service demandé
  - ✓ réserver les ressources dans chaque nœud (CAC)
  - ✓ pb : IP non connecté
  - ✓ RSVP

Architecture IntServ/RSVP

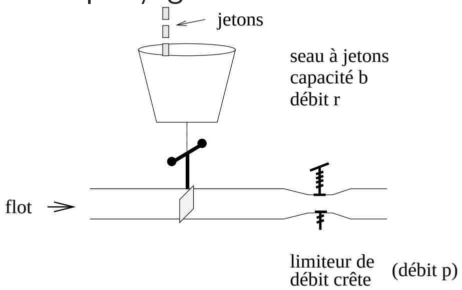


3 composants :
- RSVP (signalisation)
- classification
- ordonnancement

➠ traitement *par flot*

TSPEC : *traffic specification*

- contrat de trafic
  - ✓ p débit crête
  - ✓ r débit moyen
  - ✓ b rafale max *(burst)*
  - ✓ L taille max de datagramme
  - ✓ m unité min de poliçage

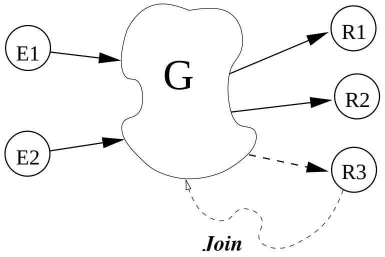


Services IntServ

- GS *guaranteed service*
  - ✓ débit garanti
  - ✓ délai max garanti
  - ✓ pas de pertes
- CL *controlled load*
  - ✓ définition vague
  - ✓ service "au mieux" sans congestion
  - ✓ applications adaptatives

Aparté : multicast IP

- many-to-many
- 2 concepts fondamentaux
  - ✓ abstraction de groupe
  - ✓ orienté récepteur



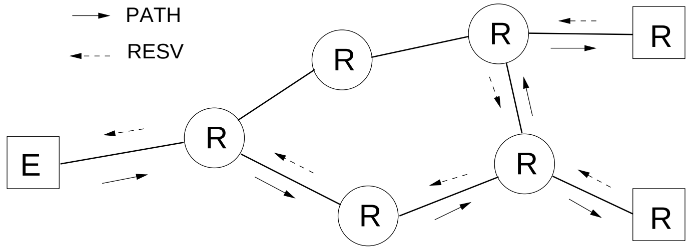
Resource reSerVation setup Protocol

signalisation pour IntServ

- multipoint N vers M
- unidirectionnel
- orienté récepteur
- hétérogénéité
- dynamicité
- réservations partagées
- états relâchés

Mécanismes RSVP

- messages PATH
  - ✓ arbre de parcours inverse
  - ✓ une passe avec collecte (OPWA)
- messages RESV
  - ✓ contrôle d'admission (CAC)
  - ✓ établissement de la réservation



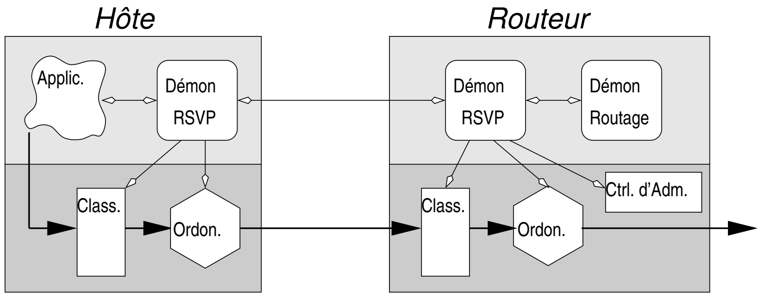
Classification

- traitement par flot

IPv4 IP src et dest + **port src/dest**
- pas "propre"
- relativement complexe

IPv6 IP src + flow label

Interfaces RSVP



- application
- *traffic control* (classificateur+ordonnanceur)
- routage
- ctrl. admission inclu dans l'interface ordonnanceur

Problèmes de l'approche IntServ

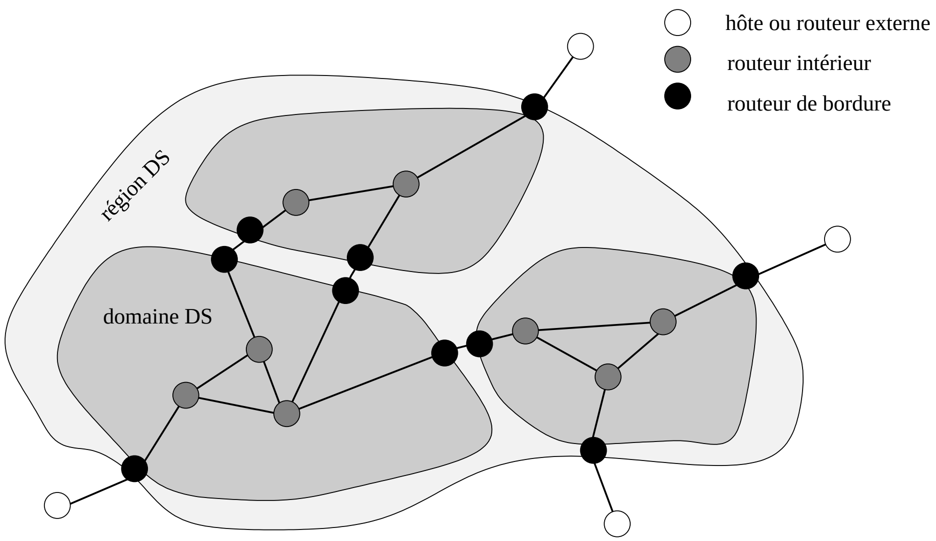
- scalabilité
  - ✓ pas dans le plan de données *(hardware)*
  - ✓ plan de contrôle
  - ✓ ➠ agrégation
- déploiement de bout en bout (ISP + applications)
- complexité de RSVP ? (spec. 112 pages)
- mauvaises implémentations ?
- *policy control*, sécurité

L'architecture DiffServ

Services différenciés : DiffServ

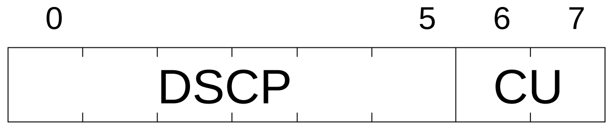
- groupe IETF démarré en août 1997
- poussé par les ISP (besoin d'offrir du service)
- plus simple qu'IntServ
- classes de trafic
- identification : DSCP (DiffServ CodePoint)
- deux types de nœuds :
  - ✓ de bordure
  - ✓ intérieurs
- conditionnement du trafic
- PHB : Per Hop Behavior

Architecture DiffServ



Champ DS

- IPv4 : octet TOS
- IPv6 : Traffic Class octet



- 6 bits non structurés
- si inconnu, utiliser le défaut
- subsume les 3 bits *precedence* d'IPv4

Services DiffServ *(PHBs)*

- EF *expedited forwarding*
  - ✓ BP garantie
  - ✓ pertes, délais et gigue faibles
  - ✓ "ligne louée virtuelle"
  - ✓ conditionnement de trafic en bordure
- AF *assured forwarding*
  - ✓ réduire la congestion à long terme
  - ✓ accepter les rafales (congestion à court terme)
  - ✓ gestion active de la file ➠ RED *(Random Early Detection)*

Random Early Dectection

- contrôle de congestion TCP (pertes)
- synchronisation
- RED
  - ✓ taille moyenne de la file
  - ✓ jette des paquets avec proba fct de la taille moyenne
- pb : en pratique, difficile à régler

Conclusion sur DiffServ

- + scalable dans le plan de données
  - ✓ nombre réduit de classes
  - ✓ complexité repoussée aux bornes du réseau
- – antagonisme flexibilité/fermeté des garanties
- – pb du plan de contrôle
  - ✓ sur provisionner
  - ✓ *bandwidth broker*
- – multicast ?

La qualité de service dans Linux

Historique

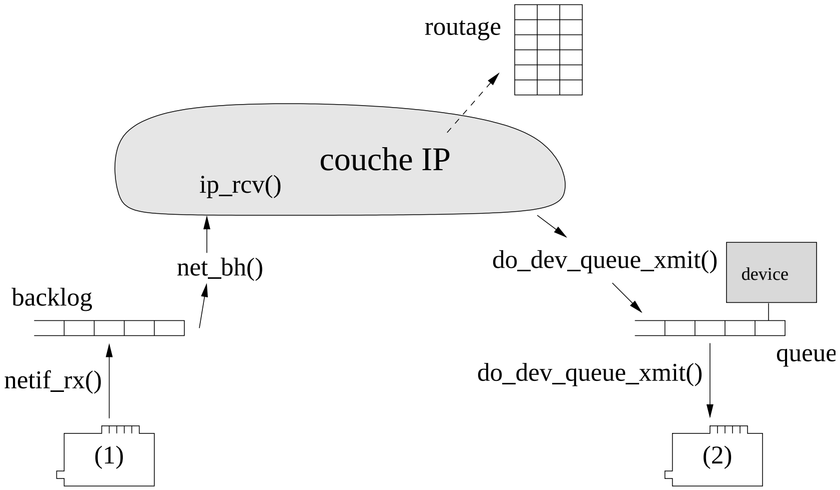
- noyau Unix libre commencé en 1991
- support QoS apparu dans les noyaux 2.1.xx (1998)
- Alexey Kuznetsov
- DiffServ depuis les derniers 2.3.xx

Mécanismes disponibles

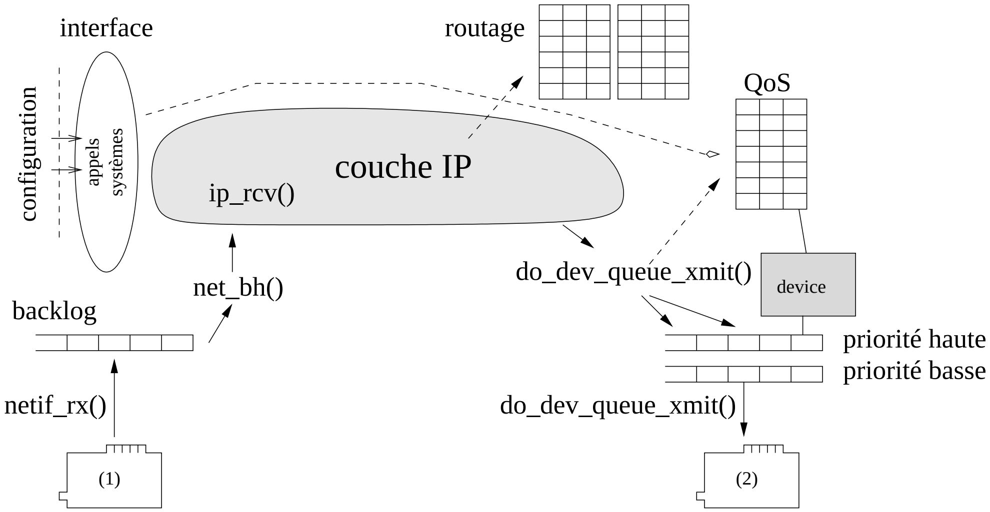
- *policing*
- ordonnanceurs
  - ✓ *fair queuing*
  - ✓ TBF
  - ✓ ...
- ports de RSVP
- DiffServ en cours
- Routage sur le TOS

```
$ ip route add 10.0.0.0/24 tos xx via
192.168.1.249
```

Linux sans la QoS



Linux avec la QoS



Limitations

- mécanismes peu testés
- implémentation peu testée
- système non temps réel
  - ✓ ex. granularité de l'horloge (e.g. pour TBF)

Conclusion

Utiliser la QoS

- mécanismes divers et non stabilisés
- routage TOS ou QoS
- IntServ
  - ✓ RAPI
  - ✓ SCRAPI
- DiffServ
  - ✓ fixer le champ DS (e.g. setsockopt)
- API générique ?
  - ✓ généricité
  - ✓ efficacité
  - ✓ ➠ contraintes opposées

Conclusion

- des problèmes subsistent
  - ✓ techniques
  - ✓ politiques
- solutions plausibles à court terme
  - ✓ routage intra-domaine sur le TOS
  - ✓ DiffServ

Références

- IETF DiffServ working group, `http://www.ietf.org/html.charters/diffserv-charter.html`
- IETF IntServ working group (terminé), `http://www.ietf.org/html.charters/OLD/intserv-charter.html`
- Internet Protocol – Quality of Service Page (aspects implémentation), `http://qos.ittc.ukans.edu`

[1] Almersberger (Werner). – *Differentiated Services on Linux.* – Web page, 2002. `http://diffserv.sourceforge.net/`.

[2] Deleuze (Christophe). – *Qualité de service dans l'Internet : problèmes liés au haut débit et au facteur d'échelle.* – Paris, France, Thèse de PhD, université Pierre et Marie Curie, janvier 2000.

Fin

Document réalisé avec LaTeX.
Généré le 13 janvier 2003.
Classe de document `foils`.
Dessins réalisés avec `xfig`.

Christophe Deleuze, `Christophe.Deleuze@activia.net`
`http://www.activia.net`

Ce document est disponible en postscript compressé avec gzip à
`http://rp.lip6.fr/~deleuze/activia.html`