

## Chapitre IV

### Multimédia et Qualité de Service

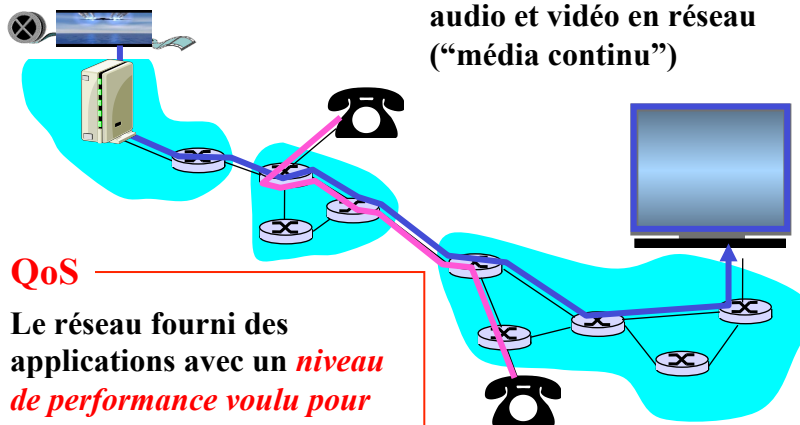


(ref: Kurose & Ross)

Cours chap. 4  
Tutorial IBM ch. 8  
(DiffServ + IntServ)

## Multimedia, Qualité de Service: Qu'est-ce?

**Applications multimédia:**  
audio et vidéo en réseau  
("média continu")



### QoS

Le réseau fournit des applications avec un *niveau de performance voulu pour l'application en question.*

## Chapitre 4: Buts

### Principes

- Classification des applications
- Identification des services réseau nécessaires aux applications
- Tirer le meilleur du service "best effort"
- Mécanismes pour assurer une certaine QoS

### Protocoles and Architectures

- Protocoles spécifiques pour le "best effort"
- Architectures pour la QoS

## Applications réseau MM

### Classes des applications

#### MM:

- 1) Streaming de données audio and video enregistrées
- 2) Streaming de données audio and video en direct
- 3) Audio and video en temps réel

**Le Jitter (gigue) est la variabilité des retards des paquets dans un même flux**

### Caractéristiques

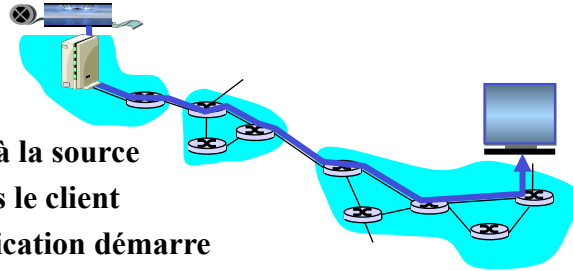
#### fondamentales:

- **Sensibilité au délai de transmission**
  - Délai de bout-en-bout
  - Jitter (gigue)
- Mais **tolérant aux pertes**: les pertes peu fréquentes causent des problèmes mineurs
- Contrairement aux données qui sont intolérantes aux pertes.

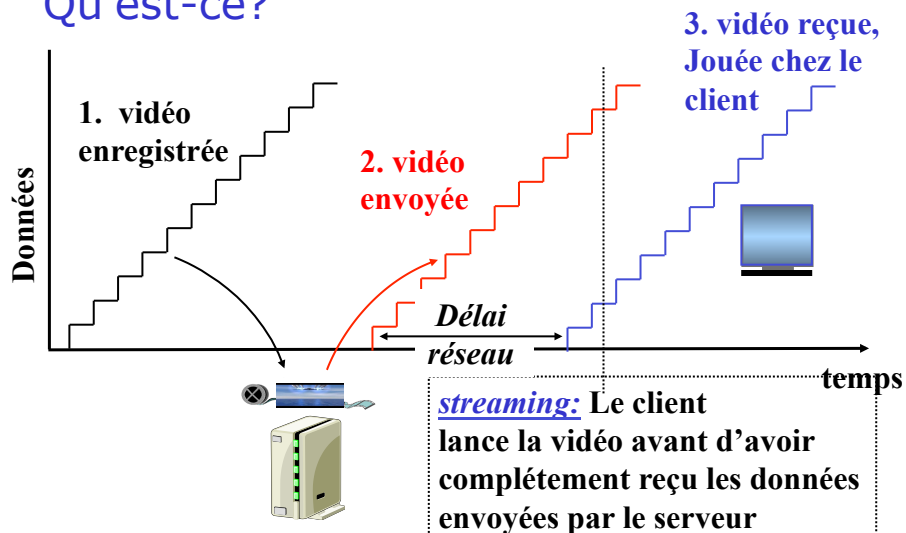
## Streaming de Multimedia enregistré

### Streaming:

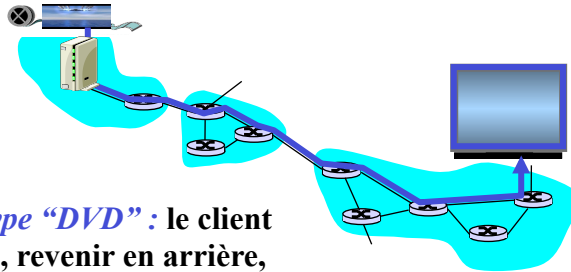
- ❑ media enregistré à la source
- ❑ Transmission vers le client
- ❑ streaming: l'application démarre du côté client *avant* que toutes les données soient arrivées
- ❑ Contraintes temporelles pour les données transmises: L'application ne doit pas être interrompue



## Streaming du multimedia enregistré: Qu'est-ce?



## Streaming Multimedia enregistré: Interactivité



- **Fonctionnalité de type “DVD”** : le client can faire une pause, revenir en arrière, aller en avant
  - Un délai initial de 10 sec. est OK
  - 1-2 sec pour une command : OK
  - RTSP souvent utilisé

## Streaming multimédia en temps réel

### Exemples:

- Radio Internet (talk show)
- Evènement sportif en direct

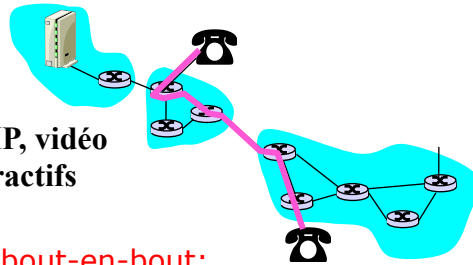
### Streaming

- Tempon (playback)
- Le playback peut être joué quelques dizaines de secondes après la transmission.
- Contraintes temporelles malgré tout

### Interactivité

- Impossible d’aller en avant (fast forward)
- Pause, rembobiner: Impossible!

## Multimédia interactif, en temps réel



□ **Applications:** téléphonie IP, vidéo conférences, mondes interactifs distribués

• **Exigences pour le délai bout-en-bout:**

- audio: < 150 msec bon, < 400 msec OK
  - inclu des délais au niveau des applications (*packetization*) et du réseau
  - Délais plus importants: interactivité difficile

• **Initialisation de session**

- Comment l'appelant donne son adresse IP, son numéro de port, ses algorithmes de codage?

## Multimedia sur Internet aujourd'hui

**TCP/UDP/IP: "service best-effort"**

- *aucune* garantie sur le délai, perte



Mais vous avez dit que les applications nécessitent de la QoS et un niveau De performance pour être efficace !



Les applications multimédia sur Internet utilisent des techniques au niveau de l'application pour limiter (au mieux) les effets des pertes et du délai

## Comment Internet devrait-il évoluer pour mieux supporter les apps multimédia?

### Philosophie des Services

#### Intégrés:

- Changements fondamentaux d'Internet pour que les apps puissent réserver de la LB de bout-en-bout.
- A besoin d'un software complexe dans les hosts et les routeurs

#### Laissez-faire

- Aucun changement majeur
- Plus de LB en cas de besoin
- Distribution de contenu, multicast.
  - Niveau applications

### Philosophie des services différenciés:

- Moins de changements dans l'infrastructure Internet, fournir des services de 1ère, 2ème classe.



Quel est votre opinion?

## Quelques mots sur la compression audio

- Signal analogique échantillonné à un débit constant
  - téléphone: 8'000 échant./sec
  - Musique CD : 44,100 échant./sec
- Chaque échantillon est quantifié, i.e., arrondi
  - e.g.,  $2^8=256$  valeurs quant. possible
- Chaque valeur quantifiée est représentée par des bits: 8 bits pour 256 valeurs
- Exemple:
  - 8,000 échant./sec, 256 valeurs quantifiées-> 64,000 b/s
- Le récepteur le convertit en signal analogique:
  - Réduction de la qualité

### Débits (exemples)

- CD: 1.411 Mb/s
- MP3: 96, 128, 160 kb/s
- Téléphonie internet: 5.3 - 13 kb/s

## Quelques mots sur la compression vidéo

- La video est une séquence d'images à un rythme constant
    - e.g. 24 images/sec
  - Les images digitales sont une rangée de pixels
  - Chaque pixel est représenté par des bits
  - Redondance
    - spatiale
    - temporelle
- Exemples:
- MPEG 1 (CD-ROM) 1.5 Mb/s
  - MPEG2 (DVD) 3-6 Mb/s
  - MPEG4 (souvent utilisé sur internet, < 1 Mbps)
- Recherche:
- Vidéo en couches (extensible)
    - S'adapte à la LB disponible

## Chapitre 4

- Multimedia Networking Applications
- **Streaming stored audio and video**
  - RTSP
- Real-time Multimedia: Internet Phone Case Study
- Protocols for Real-Time Interactive Applications
  - RTP, RTCP
  - SIP
- Beyond Best Effort
- Scheduling and Policing Mechanisms
- Integrated Services
- RSVP
- Differentiated Services

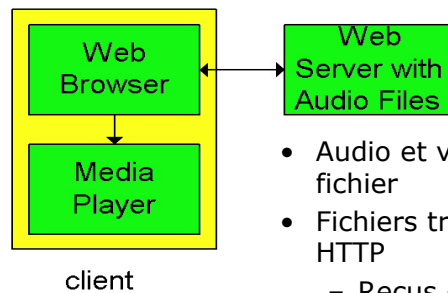
## Streaming de données multimedia enregistrées

- Niveau applicatif  
Techniques de streaming pour tirer le meilleur du service "best effort":
- Mémoriser du côté client
  - Utiliser UDP au lieu de TCP
  - Codages multiples pour le multimédia

### Media Player

- Élimination du jitter
- Décompression
- Dissimulation des erreurs
- GUI avec contrôle interactif

## Multimédia sur Internet: l'approche la plus simple



- Audio et vidéo enregistrés dans un fichier
- Fichiers transférés comme des objets HTTP
  - Reçus dans leur intégrité par le client
  - Joués par un "vidéo/audio player"

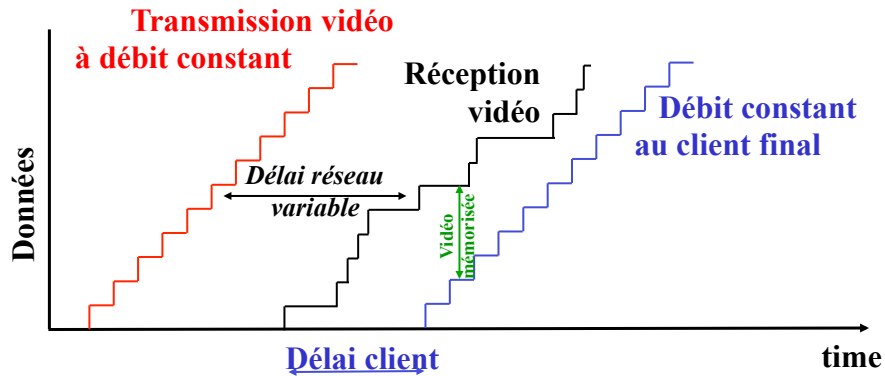
### Pas de streaming audio/video :

- Non, "pipelining," long délais avant de voir/entendre qqch!



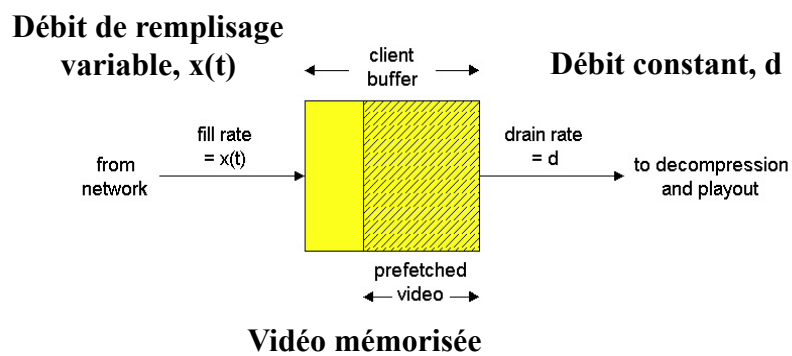


## Streaming Multimedia: Client Buffering



- Mémorisation du côté client, le délai de playout compense le délai du réseau

## Streaming Multimédia: Mémorisation du côté client



- Mémorisation côté client, le délai de playout compense le délai du réseau

## Streaming Multimédia: UDP ou TCP?

### UDP

- Le serveur envoie au débit voulu par le client (en ignorant les encombrements du réseau!)
  - Souvent le débit envoyé = débit de codage=débit constant
  - Alors, débit de remplissage = débit constant – pertes de paquets
- Délai de playout petit (2-5 secondes) pour compenser la gigue du réseau
- Correction des erreurs: si le temps le permet

### TCP

- Envoie au débit maximum
- Le débit de remplissage fluctue à cause du contrôle de l'encombrement de TCP.
- Délai de playout plus grand: adouci le débit délivré
- HTTP/TCP passe plus facilement à travers les firewalls

## Streaming Multimédia: débit client



**Q:** Comment faire avec des débits très différents à la réception ?

- 28.8 Kb/s dialup
- 100Mb/s Ethernet

**A:** Le serveur mémorise, transmet de multiples copies de vidéos, codées à des débits différents

## User Control of Streaming Media: RTSP

### HTTP

- Pas prévu pour les contenus MM
- Aucune commande pour la retransmission rapide

### RTSP: RFC 2326

- Application client-serveur.
- Permet à l'utilisateur de contrôler: rembobiner, aller en avant, pause, etc...

### Ce qu'il ne fait pas:

- Ne définit pas comment l'audio/vidéo est encapsulé pour le streaming à travers le réseau
- Ne dit pas comment le média est transporté (UDP ou TCP)
- Ne spécifie pas comment le media player mémorise l'audio/vidéo.

## RTSP: contrôle hors bande

### FTP utilise un canal de contrôle hors bande:

- Un fichier est transféré sur une connexion TCP
- L'information de contrôle (changement de répertoire, effacement de fichiers, renommage,...) est envoyée à travers une connexion TCP séparée.
- Les canaux "hors-bande" et "dans la bande" utilisent des numéros de ports différents.

### Les messages RTSP messages sont aussi envoyés hors bande :

- Les messages de contrôle utilisent différents numéros de port que le stream média: hors bande.
  - Port 554
- Le stream média est considéré "dans la bande".

## Exemple RTSP

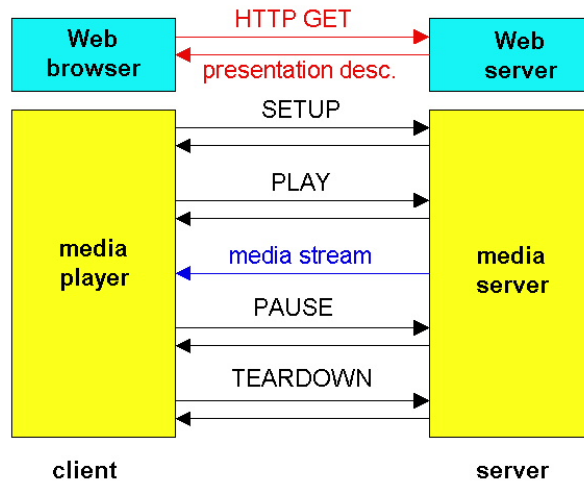
### Scénario:

- Le metafile communique avec un browser web
- Le browser lance un player
- Le player met sur pied un connexion de contrôle RTSP, une connexion de données au serveur de streaming
- QuickTime, RealNetworks

## Exemple de Metafile

```
<title>Twister</title>
<session>
  <group language=en lipsync>
    <switch>
      <track type=audio
        e="PCMU/8000/1"
        src = "rtsp://audio.example.com/twister/audio.en/lofi">
      <track type=audio
        e="DVI4/16000/2" pt="90 DVI4/8000/1"
        src="rtsp://audio.example.com/twister/audio.en/hifi">
    </switch>
    <track type="video/jpeg"
      src="rtsp://video.example.com/twister/video">
  </group>
</session>
```

## Opération RTSP



## Chapitre 4

- Multimedia Networking Applications
- Streaming stored audio and video
  - RTSP
- **Real-time, Interactive Multimedia: Internet Phone Case Study**
- Protocols for Real-Time Interactive Applications
  - RTP, RTCP
  - SIP
- Beyond Best Effort
- Scheduling and Policing Mechanisms
- Integrated Services
- RSVP
- Differentiated Services

## Applications interactives en temps réel

- PC-2-PC phone
  - Les services de messagerie instantanée les fournissent
- PC-2-phone
  - Dialpad
  - Net2phone
- Vidéoconférences avec des Webcams

On va regarder l'exemple du téléphone PC-2-PC en détail

## Multimédia *interactif*: Téléphonie Internet

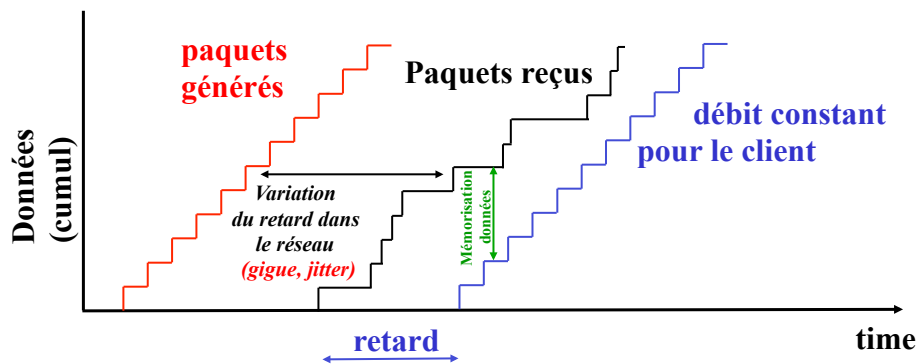
### Introduction par un exemple

- Audio de celui qui parle: alternativement des périodes de parole et de silence.
  - 64 kb/s durant les périodes de parole
- Les paquets sont générés pendant les périodes de parole
  - Morceaux de 20 msec à 8 KB/sec: 160 B de données
- Un entête "application" est ajouté à chaque morceau de 20 ms.
- Le tout est encapsulé dans un paquet UDP
- L'application envoie des paquets UDP dans un socket chaque 20 ms quand on parle

## Téléphonie internet: Perte et retard de paquets

- **Perte de paquets:** Perte de datagramme IP à cause d'un encombrement dans le réseau (mémoire d'un buffer qui déborde dans un routeur)
- **Retard:** Le datagramme IP arrive trop tard à destination pour pouvoir être utile
  - retards: processing, files d'attente dans le réseau, retards dans les systèmes terminaux (émetteurs, récepteurs)
  - Retards maximum tolérables: 400 ms
- Tolérance à la perte: dépend du codage de la voix, les pertes tolérables peuvent varier de 1% à 10%.

## Gigue (jitter)



- La différence de retard entre deux paquets consécutifs peut être de plus de 20 ms.

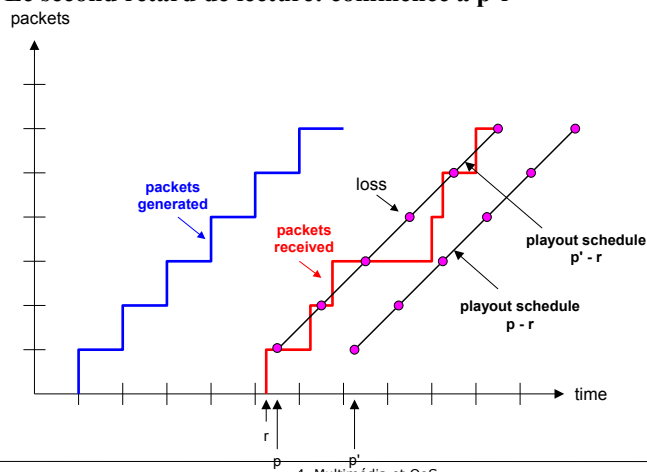


## Téléphonie internet: Retard de lecture fixe

- Le destinataire attend exactement  $q$  msec après la création d'un morceau de données avant de le lire..
  - $t$ : origine du temps du morceau de données. Lecture à  $t+q$  .
  - Si les données arrivent après  $t+q$ : les données sont perdues car elles arrivent trop tard
- Compromis pour  $q$ :
  - $q$  important: moins de perte de paquets
  - $q$  petit: meilleure interactivité

## Retard de lecture fixe

- L'émetteur génère des paquets chaque 20 msec.
- Le premier paquet arrive au temps  $r$
- Le premier retard de lecture est fixé à  $p'-r$
- Le second retard de lecture: commence à  $p-r$



## Retard de lecture adaptatif, I

- **But:** minimiser le retard de lecture, garder les pertes faibles
- **Approche:** ajuster le retard de lecture au fil de la conversation :
  - Estimer le retard dû au réseau, ajuster le délai de lecture au fil de la conversation.
  - Périodes de silence compressées et alongées.
  - Morceaux de données joués toutes les 20 msec dans le même discours.

$t_i$  = origine du temps du  $i$ ème paquet

$r_i$  = heure d'arrivée du paquet  $i$  au niveau du destinataire

$p_i$  = heure de lecture du paquet  $i$

$r_i - t_i$  = temps de transmission de bout-en-bout

$d_i$  = moyenne approximative du temps de transmission après avoir reçu le  $i$ ème paquet

$$d_i = (1 - u)d_{i-1} + u(r_i - t_i)$$

Où  $u$  est une constante fixe (e.g.,  $u = .01$ )

## Retard de lecture adaptatif, II

Il est utile d'estimer l'écart-type moyen du temps de transmission,  $v_i$  :

$$v_i = (1 - u)v_{i-1} + u |r_i - t_i - d_i|$$

Les estimations  $d_i$  et  $v_i$  sont calculées pour chaque paquet reçu, même si elles ne servent qu'à déterminer le moment de lecture du premier paquet d'un fragment de discours donné.

Pour le premier paquet d'un discours, son heure de lecture est:

$$p_i = t_i + d_i + Kv_i$$

où  $K$  est une constante positive.

## Retard de lecture adaptatif, III

**Q:** Comment est-ce que le destinataire détermine si un paquet est le premier d'une conversation?

- Sans perte, le destinataire observe les temps d'arrivée.
  - Différence de temps entre deux paquets > 20 msec --> un nouveau discours commence.
- Avec des pertes, le destinataire doit observer les temps d'arrivée et le numéros de séquence.
  - Différence de temps entre deux paquets > 20 msec **et** numéros de séquence sans trous de séquence --> un nouveau discours commence.

## Problèmes 5,6,7,8,9b

## Récupération de paquets perdus (1)

### Correction d'erreurs sans voie de retour (FEC):

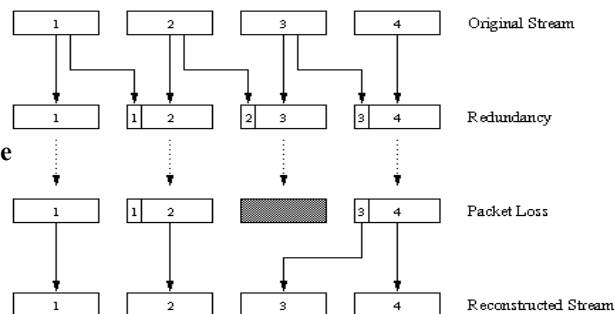
- Pour chaque groupe de  $n$  morceaux créer un OU exclusif à partir des morceaux originaux
- Envoyer les  $n+1$  morceaux, augmentant la LB d'un facteur de  $1/n$ .
- On peut reconstruire l'original ( $n$  morceaux) s'il y a au plus un morceau parmi les  $n+1$  qui est perdu
- Le retard de lecture doit être fixé pour recevoir les  $n+1$  paquets
- Compromis:
  - augmenter  $n$ : moins de perte de LB
  - augmenter  $n$ : plus de retard de lecture
  - augmenter  $n$ : il y a une plus grande probabilité que 2 ou plus de morceaux soient perdus

## Récupération de paquets perdus (2)

### 2ème mécanisme FEC

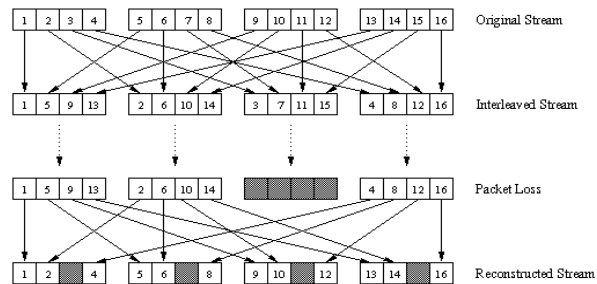
- envoyer les données avec une résolution inférieure avec redondance

Par exemple:  
 Au lieu du flux original PCM à 64 kb/s  
 utiliser un flux GSM à 13 kb/s.



- Quand il n'y a pas de perte consécutive le récepteur peut cacher les pertes.
- On peut adapter la méthode et associer plusieurs morceaux entre eux à chaque morceau original.

## Récupération de paquets perdus (3)



### Entrelacement

- Les morceaux sont séparés en plus petites unités
- Par exemple, des unités de 4\*5 msec par morceau
- Les paquets contiennent des petites unités de chaque morceau
- Si un paquet est perdu il nous reste la grande partie du message
- Pas d'overhead dû à la redondance
- Mais ajoute un retard de lecture

## Résumé: Multimedia sur Internet : ensemble d'astuces

- utilisation d'UDP pour éviter les délais de TCP dûs au contrôle de congestion pour les applications sensibles au retard
- Du côté client **retard de lecture adaptatif**: à compenser pour le délai
- Du côté serveur: **adapter le débit** à la largeur de bande disponible entre le client et le serveur
  - Choisi parmi les débits pré-encodés
  - Débit de codage dynamique par le serveur
- Récupération des erreurs (sur UDP)
  - FEC, entrelacement
  - retransmissions, si nous avons le temps
  - Cacher les erreurs

## Problèmes 10,11

## Chapitre 4

- Multimedia Networking Applications
- Streaming stored audio and video
  - RTSP
- Real-time, Interactive Multimedia: Internet Phone Case Study
- **Protocols for Real-Time Interactive Applications**
  - RTP, RTCP
  - SIP
- Beyond Best Effort
- Scheduling and Policing Mechanisms
- Integrated Services
- RSVP
- Differentiated Services

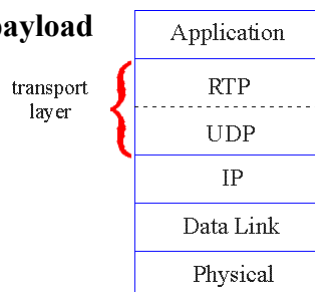
## Real-Time Protocol (RTP)

- RTP spécifie une structure de paquet pour les paquets audio et vidéo
- RFC 1889.
- Les paquets RTP fournissent
  - Identification du type de payload
  - Numéros de séquence
  - Référence de temps
- RTP fonctionne sur les systèmes terminaux.
- Les paquets RTP sont encapsulés dans des segments UDP
- Interopérabilité: Si deux téléphones internet utilisent RTP, alors ils devraient être capables de communiquer ensemble.

## RTP fonctionne au-dessus de UDP

**Les bibliothèques RTP fournissent un interface qui “étend” UDP: provide a transport-layer interface that extend UDP:**

- **numéros de ports, adresses IP**
- **identification du type de payload**
- **numérotation des paquets**
- **référence de temps**



## Exemple RTP

- Considérez un flux de voix codé (PCM) de 64 kb/s sur RTP.
- L'application collecte les données codées en messages, e.g., chaque 20 msec = 160 bytes.
- Les messages audio + entête RTP forment un paquet RTP, qui est encapsulé dans un segment UDP.
- L'entête RTP indique le type de codage audio utilisé dans chaque paquet
  - L'émetteur peut changer de codage durant la conférence.
- L'entête RTP contient des numéros de séquence et une référence de temps.

## RTP et QoS

- RTP ne fournit **pas** les mécanismes pour assurer la transmission des données en un temps défini. Il ne fournit d'ailleurs aucune garantie de qualité de service.
- L'encapsulation est seulement vue dans les systèmes terminaux: Elle n'est pas examinée dans les routeurs intermédiaires.
  - Les routeurs fournissant un service "best effort" ne font pas un effort spécial pour permettre aux paquets RTP d'arriver à temps à destination.



## Entête RTP



RTP Header

**Payload Type (7 bits):** Indique le type de codage utilisé. Si l'émetteur change le codage au milieu d'une conférence, il va informer le récepteur avec ce champ.

- Payload type 0: PCM mu-law, 64 kb/s
- Payload type 3, GSM, 13 kb/s
- Payload type 7, LPC, 2.4 kb/s
- Payload type 26, Motion JPEG
- Payload type 31. H.261
- Payload type 33, MPEG2 video

**Numéro de séquence (16 bits):** Incrémente de un pour chaque paquet RTP envoyé, peut être utilisé pour détecter des pertes de paquets et restaurer une séquence de paquets.

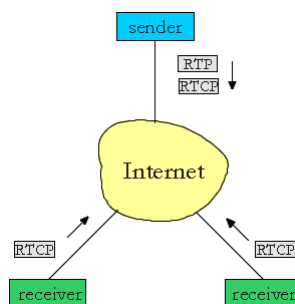
## Entête RTP (2)

- **Timestamp field (32 bytes long).** Référence de temps. Donne l'heure d'échantillonnage du premier octet du paquet.
  - Pour l'audio la valeur affichée par cette horloge augmente d'une unité à chaque période d'échantillonnage (par exemple, chaque 125 µsecs pour une horloge à 8 KHz)
  - Si les morceaux générés par l'application contiennent 160 échantillons, la référence de temps augmente cette valeur de 160 à chaque nouveau paquet RTP lorsque la source est active. Si la source est inactive l'horloge augmente à un rythme constant.
- **SSRC field (32 bits long).** Identifiant de la source de synchronisation. Chaque flux d'une session RTP est généralement doté d'un SSRC particulier.

## Real-Time Control Protocol (RTCP)

- Fonctionne avec RTP.
- Chaque participant, dans une session RTP, transmet périodiquement des paquets de contrôle à tous les autres participants.
- Chaque paquet RTCP contient des rapports de réception/émission
  - Les rapports statistiques sont utiles pour les applications
- Les statistiques incluent le nombre de paquets envoyés, perdus, gigue etc,.
- Le feedback peut être utilisé pour contrôler la performance
  - L'émetteur peut modifier sa transmission basé sur les feedbacks reçus

## RTCP - suite



- **Au cours d'une session RTP chaque participant transmet des paquets multicast. Tous les paquets RTP et RTCP appartenant à une session utilisent des adresses multicast**
- **Les paquets RTP et RTCP sont différenciés les uns des autres par l'usage de numéros de ports distincts**
- **Pour limiter le trafic, chaque participant réduit son trafic RTCP Au fur et à mesure que le nombre de participants augmente**

## Paquets RTCP

### Rapports des paquets de la destination:

- Fraction des paquets perdus, du dernier numéro de séquence, de la gigue moyenne.

### Rapport des paquets de l'émetteur:

- SSRC du flux RTP, temps courant, nombre de paquets envoyés, nombre d'octets envoyés.

### Description des paquets de l'expéditeur:

- Adresse email de l'émetteur, nom de l'émetteur, SSRC du flux RTP associé.
- Fourni une correspondance entre le SSRC et le nom de l'utilisateur/host.

## Synchronisation des flux

- RTCP peut synchroniser différents flux multimédia dans une session RTP
- Quand une application de vidéoconférence génère un flux pour la vidéo et un flux pour l'audio.
- Les timestamps dans RTP sont liés aux horloges d'échantillonnage de l'audio et de la vidéo
  - Pas lié à un temps de référence
- Chaque paquet RTCP source contient:
  - timestamp du paquet RTP
  - Temps de référence pour quand le paquet a été créé.
- Les récepteurs peuvent utiliser cette association pour synchroniser l'audio et la vidéo.

## Adaptation au débit RTCP

- RTCP essaie de limiter son trafic à 5% de la largeur de bande de la session.
  - Les 75 kb/s sont également répartis parmi les destinataires :
    - Avec R destinataires, chacun dispose de  $75/R$  kb/s pour son trafic RTCP.
  - L'expéditeur envoie du trafic RTCP à 25 kb/s.
  - Un participant détermine le temps de transmission des paquets RTCP en estimant la taille moyenne des paquets RTCP divisé par le débit dont ils disposent.
- Exemple**
- Supposez qu'un expéditeur envoie une vidéo à 2 Mb/s. Alors RTCP va essayer de limiter son trafic à 100 kb/s.
  - RTCP accorde 75% de son débit aux destinataires et les 25% restant à l'expéditeur.

## Problèmes 12,16

## SIP

- Session Initiation Protocol
- Vient de l'IETF

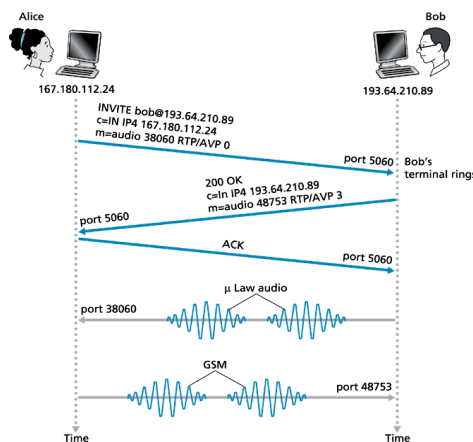
### Vision de SIP à long terme

- Tous les appels téléphoniques et conférences vidéo passeront par Internet
- Les gens sont identifiés par des noms ou des adresses email, plutôt que par des numéros de téléphone.
- Vous pouvez atteindre quelqu'un où qu'il soit physiquement. Son adresse IP peut également être quelconque.

## Services SIP

- Initiation de session
  - Fourni les mécanismes pour un appelant (auteur appel) d'aviser l'appelé son souhait d'engager une conversation
  - Permet aux participants de s'entendre sur le codage à utiliser.
  - Fourni les mécanismes pour terminer un appel.
- Détermine l'adresse IP "instantanée" de l'appelé.
  - L'adresse IP change constamment
- Gestion d'appel
  - Ajout d'un nouveau flux de données pendant l'appel
  - Changement du codage pendant l'appel
  - Invitation de nouveaux participants
  - Transfère et maintient les appels

## Etablissement d'un appel avec une adresse IP connue



- Alice envoie un message INVITE à Bob en indiquant son adresse IP et son numéro de port. Préférence de codage: loi  $\mu$  PCM

- L'accusé de réception de Bob (200 OK) indique le numéro de port, adresse IP et codage préféré (GSM)

- Les messages SIP peuvent être envoyés sur TCP ou UDP; Ici sur RTP/UDP.

- Numéro de port SIP par défaut: 5060.

## Etablissement d'un appel (suite)

- Négociation du codec:
  - Supposons que Bob n'ait pas de codeur avec la loi  $\mu$ .
  - Bob va répondre avec un "606 Not Acceptable" et lister les codeurs dont il dispose.
  - Alice peut envoyer un nouveau message INVITE en choisissant un codeur approprié.
- Rejet de l'appel
  - Bob peut rejeter l'appel en des messages tels que "occupé," "absent," "payement exigé" "interdit".
- Utilisation de RTP ou d'un autre protocole.

## Exemple de message SIP

```
INVITE sip:bob@domain.com SIP/2.0
Via: SIP/2.0/UDP 167.180.112.24
From: sip:alice@hereway.com
To: sip:bob@domain.com
Call-ID: a2e3a@pigeon.hereway.com
Content-Type: application/sdp
Content-Length: 885

c=IN IP4 167.180.112.24
m=audio 38060 RTP/AVP 0
```

• Ici nous ne connaissons pas l'adresse IP de Bob. Des serveurs intermédiaires SIP sont nécessaires

• Alice envoie et reçoit des messages SIP en utilisant le port de défaut 506.

• Alice spécifie avec "Via": entête que le client SIP envoie et reçoit, adresse IP, messages sur UDP.

### Notes:

- Syntaxe HTTP
- sdp = session description protocol
- Identifiant d'appel unique pour chaque appel.

## Conversion d'adresses de site de l'utilisateur

- Un appelant désire appeler quelqu'un mais il ne connaît que son nom ou son adresse email.
  - On a besoin d'obtenir l'adresse IP actuelle de l'appelé:
    - L'utilisateur voyage
    - Protocole DHCP
    - L'utilisateur a différents appareils (PC, PDA,...)
  - Les résultats peuvent être basés sur:
    - Moment du jour (travail, maison)
    - appelant (le boss qui appelé à la maison)
    - status de l'appelant (appels envoyés sur une boîte vocale quand on est déjà en conversation avec quelqu'un)
- Service fourni par des serveurs SIP :
- Enregistrement à un serveur SIP
  - Serveur proxy SIP

## SIP Registrar

- ❑ **Quand Bob démarre un client, le client envoie un message SIP REGISTER au serveur d'enregistrement de Bob (fonction similaire voulue par Instant Messaging)**

### Register Message:

```
REGISTER sip:domain.com SIP/2.0
Via: SIP/2.0/UDP 193.64.210.89
From: sip:bob@domain.com
To: sip:bob@domain.com
Expires: 3600
```

## Proxy SIP

- Alice envoie le message d'invitation à son serveur proxy
  - Contient l'adresse sip:bob@domain.com
- Le proxy est responsable pour le routage des messages vers le destinataire
  - Éventuellement avec de multiples proxys.
- Le destinataire envoie la réponse à travers le même ensemble de proxys.
- Le proxy retourne le message à Alice
  - Contient l'adresse IP de Bob
- Note: le proxy is analogue à un serveur DNS local



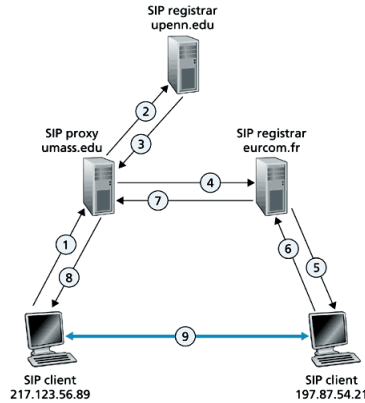
## Exemple

**jim@umass.edu**  
**veut appeler**  
**keith@upenn.edu**

(1) Jim envoie un message INVITE au proxy SIP de umass. (2) Le Proxy Transmet la requête au serveur de upenn (3) Le serveur de upenn retourne une réponse de redirection indiquant qu'il devrait essayer keith@eurecom.fr

(4) Le proxy de umass envoie un INVITE au serveur d'Eurécom. (5) Le Serveur d'Eurécom transmet l'INVITE à l'adresse 197.87.54.21, qui est l'adresse client SIP de Keith. (6-8) La réponse SIP est renvoyée (9) les deux clients peuvent s'échanger les données directement.

Note: il y a des messages SIP ACK, qui ne sont pas montrés ici.



## Comparaison avec H.323

- H.323 est un autre protocole de signalisation pour des applications vidéo/voix interactives
- H.323 est complète. C'est un suite de protocoles pour les conférences multimédia: signalisation, enregistrement, admission, contrôle, transports, codecs, ..
- SIP est un seul composant. Travaille de préférence avec RTP. Peut être combiné avec d'autres services/protocoles.
- H.323 vient de l'ITU (téléphonie).
- SIP vient de l'IETF: Emprunte beaucoup de ses concepts de HTTP. SIP a un "goût" d'internet, alors que H.323 en a un de la téléphonie.
- SIP utilise le principe "KISS" : Keep it simple stupid.

## Chapitre 4

- Multimedia Networking Applications
- Streaming stored audio and video
  - RTSP
- Real-time, Interactive Multimedia: Internet Phone Case Study
- Protocols for Real-Time Interactive Applications
  - RTP, RTCP
  - SIP
- **Beyond Best Effort**
- Scheduling and Policing Mechanisms
- Integrated Services
- RSVP
- Differentiated Services

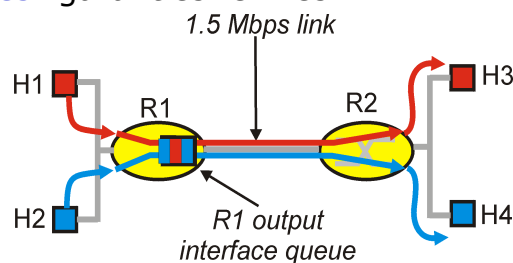
## Amélioration de la QoS dans les réseaux IP

**Jusqu'à présent:** "faire au mieux avec le best effort"

**Dans le futur:** la prochaine génération d'Internet avec des garanties de QoS

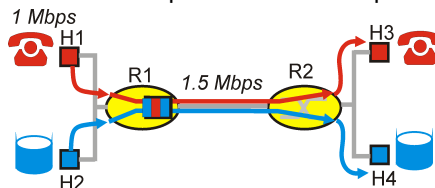
- **RSVP:** signalisation pour la réservation des ressources
- **Services différenciés:** garanties différentielles
- **Services intégrés:** garanties fermes

- Modèle simple  
Pour les études de  
partage et  
d'encombrement:



## Principes des garanties pour la QoS

- Exemple: 1Mb/s téléphone IP, FTP partagent une ligne de 1.5 Mb/s.
  - Des bursts FTP peuvent encombrer un routeur, causer des pertes audio
  - On désire donner la priorité à l'audio plutôt qu'à FTP

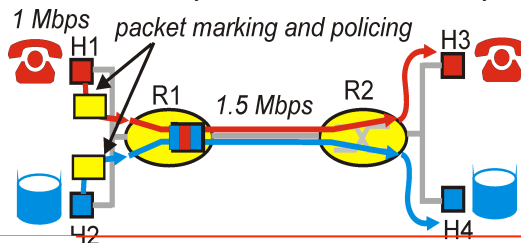


### 1er Principe

**Le marquage des paquets permet aux routeurs de différencier les paquets appartenant à des catégories de trafic différentes.**

## Principes pour des garanties de QoS (suite)

- Que se passe-t-il si les applications se comportent mal (si les application audio envoient à un débit plus élevé que le débit déclaré)
  - policing: forcer la source à adhérer aux allocations de LB
- Marquer et policer dans les bords du réseau:
  - similaire à ATM UNI (User Network Interface)

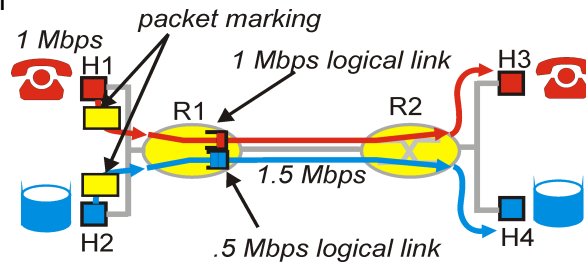


### 2ème principe

**Fournir une protection (isolation) pour les différents flux**

## Principes pour des garanties de QoS (suite)

- Allouer une largeur de bande fixe par flux: utilisation inefficace si le flux n'utilise pas cette allocation

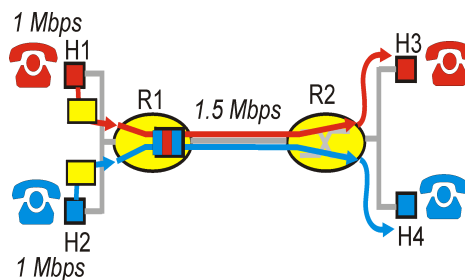


### 3ème principe

Tout en isolant les flux les uns des autres, il est souhaitable de laisser les ressources réseau libres de se répartir de la manière la plus efficace

## Principes pour des garanties de QoS (suite)

- *Basic fact of life:* On ne peut pas supporter des demandes de trafic au delà de la capacité de la ligne

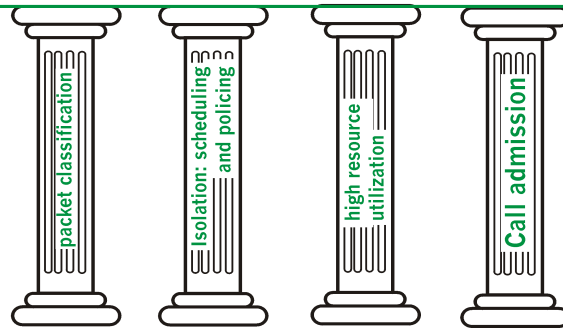


### Principe 4

Admission d'appel: le flux déclare ses besoins. Le réseau peut bloquer (occupé) s'il ne peut pas les satisfaire

## Résumé des principes pour la QoS

### QoS for networked applications



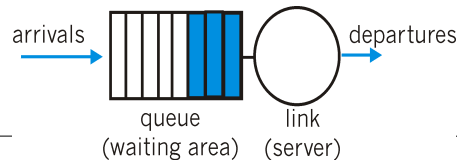
Ayons un aperçu des mécanismes pour les atteindre ...

## Chapitre 4

- Multimedia Networking Applications
- Streaming stored audio and video
  - RTSP
- Real-time, Interactive Multimedia: Internet Phone Case Study
- Protocols for Real-Time Interactive Applications
  - RTP, RTCP
  - SIP
- Beyond Best Effort
  - Scheduling and Policing Mechanisms
  - Integrated Services
  - RSVP
  - Differentiated Services

## Mécanismes d'ordonnancement et de contrôle

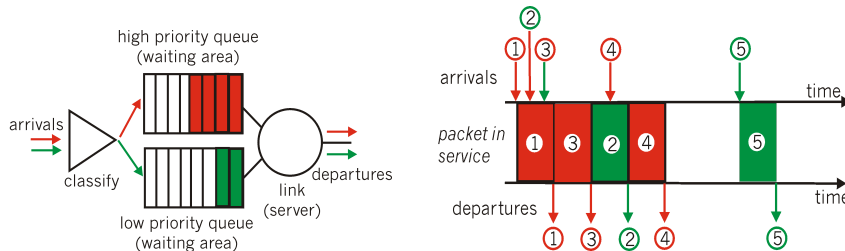
- **Mécanismes d'ordonnancement (scheduling)**: choix du paquet suivant à mettre sur la ligne
- **FIFO (first in first out) scheduling**: envoyer dans l'ordre d'arrivée
  - Exemple dans le monde réel?
  - **Rejet de paquets**: si un paquet arrive dans une file pleine: Qui éliminer?
    - Élimination de la queue: éliminer le paquet entrant
    - priorité: éliminer sur la base de la priorité
    - aléatoirement: éliminer aléatoirement



75

## Mécanismes d'ordonnancement et de contrôle (suite)

- Mise en attente avec respect des priorités:**  
transmettre le paquet avec la priorité la plus haute
- **Classes multiples, avec différentes priorités**
    - La classe peut dépendre du marquage ou des informations de l'entête (e.g. IP source/dest, no de port, etc..)
    - Exemple dans le monde réel?



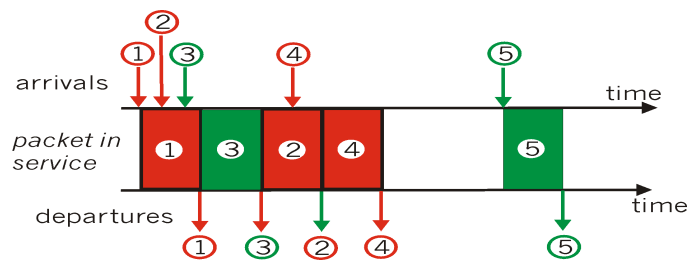
4. Multimédia et QoS

76

## Mécanismes d'ordonnancement et de contrôle (suite)

### Ordonnement "round robin" (cyclique):

- Classes multiples
- Scanne les classes des files de manière cyclique, servant un de chaque classe (si disponible)
- Exemple dans le monde réel?



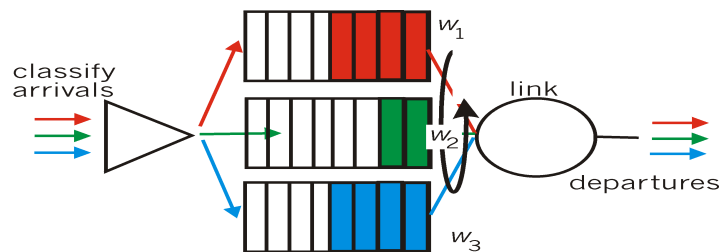
4. Multimédia et QoS

77

## Mécanismes d'ordonnancement et de contrôle (suite)

### Weighted Fair Queuing:

- Round Robin généralisé
- Chaque classe obtient un niveau de service (poids) à chaque cycle
- Exemples dans le monde réel?



4. Multimédia et QoS

78

## Problème 17

## Mécanismes de contrôle

**But:** limiter le trafic pour ne pas excéder les paramètres déclarés

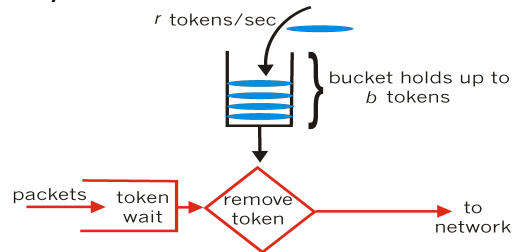
Trois points de contrôle importants:

- *(Long terme) Débit moyen:* combien de paquets peuvent être envoyés par unité de temps
  - Question cruciale: quelle est la longueur de l'intervalle? 100 paquets par sec ou 6000 packets par min ont la même moyenne!
- *Débit de pointe:* e.g., 6000 paquets par min. (ppm) en moyenne; 1500 paquets par seconde en débit de pointe
- *(Max.) Grandeur du burst:* Nombre max. de paquets envoyés consécutivement (avec aucun vide)



## Mécanismes de contrôle

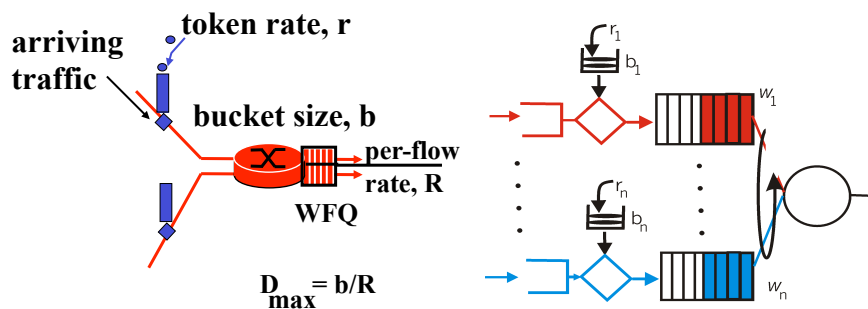
**Sceau percé:** limite l'entrée à une grandeur de burst et à un débit moyen.



- Le sceau peut contenir  $b$  jetons
- Les jetons sont générés au débit de  $r$  jetons/sec jusqu'à ce que le sceau soit plein
- *Sur l'intervalle de longueur  $t$ : nombre de paquets pouvant entrer dans le sceau est au plus égal à  $(r t + b)$ .*

## Mécanismes de contrôle (suite)

- Sceau percé, WFQ combinés pour fournir un temps maximal d'attente pour le délai, i.e., **garantie QoS!**



## Exercices 18, 19, 20

## Chapitre 4

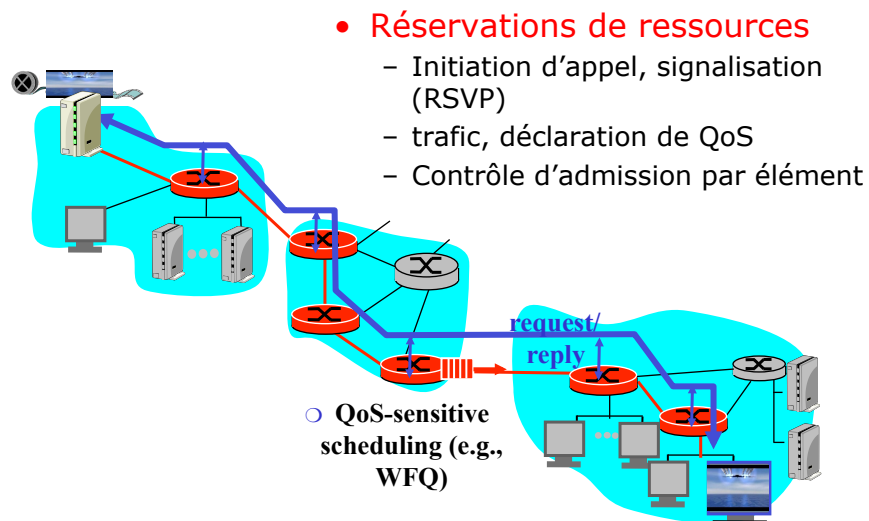
- Multimedia Networking Applications
- Streaming stored audio and video
  - RTSP
- Real-time, Interactive Multimedia: Internet Phone Case Study
- Protocols for Real-Time Interactive Applications
  - RTP, RTCP
  - SIP
- Beyond Best Effort
- Scheduling and Policing Mechanisms
- **Integrated Services**
- RSVP
- Differentiated Services

## Services intégrés (IETF)

- Architecture pour fournir des garanties de QoS dans des réseaux IP pour des sessions individuelles
- Réserveation de ressources: les routeurs maintiennent des informations sur les états des ressources allouées
- Admission/rejet de nouvelles requêtes d'appel:

**Question:** Est-ce qu'un nouveau flux peut être admis Avec les garanties de performance voulues tout en ne dégradant pas celles des flux déjà admis?

## Intserv: Scénario de garantie de QoS



## Admission d'appel

Une nouvelle session doit:

- déclarer ses exigences QoS
  - **R-spec**: défini la QoS demandée
- caractérise le trafic qu'il va envoyer dans le réseau
  - **T-spec**: défini les caractéristiques de trafic
- Protocole de signalisation: on en a besoin pour transporter les R-spec et T-spec aux routeurs (où la réservation est exigée)
  - **RSVP**

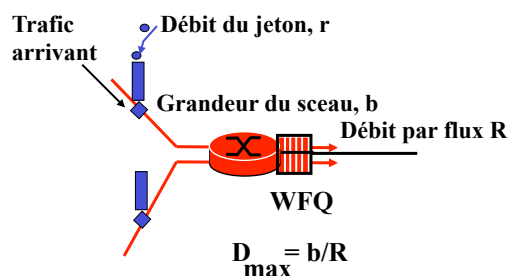
## Intserv QoS: Modèles de service [rfc2211, rfc 2212]

### Service garanti:

- Cas le pire pour les arrivées de trafic: policé par le sceau percé
- Simple (prouvé mathématiquement) *limite* sur le délai [Parekh 1992, Cruz 1988]

### Service à contrôle de charge:

- "une qualité de service approxime la QoS que le même flux recevrait d'un élément réseau non chargé"



## Chapitre 4

- Multimedia Networking Applications
- Streaming stored audio and video
  - RTSP
- Real-time, Interactive Multimedia: Internet Phone Case Study
- Protocols for Real-Time Interactive Applications
  - RTP, RTCP
  - SIP
- Beyond Best Effort
- Scheduling and Policing Mechanisms
- Integrated Services
- RSVP
- Differentiated Services

## Services différenciés de l'IETF

### Problèmes avec Intserv:

- **Etendue (Scalability):** signalisation, maintenir un état par flux est difficile pour un grand nombre de flux
- **Modèles de service flexibles:** Intserv a seulement deux classes. On aimerait des classes de service "qualitatives"
  - "comportement comme un câble"
  - Distinction relative des services relative service distinction: Platinum, Gold, Silver

### Approche de Diffserv:

- Fonctions simples dans le coeur du réseau, relativement complexes dans les bords (edge routeurs (périphérie), hosts)
- Ne définit pas des classes de service, fournit les composants fonctionnels pour construire les classes de service

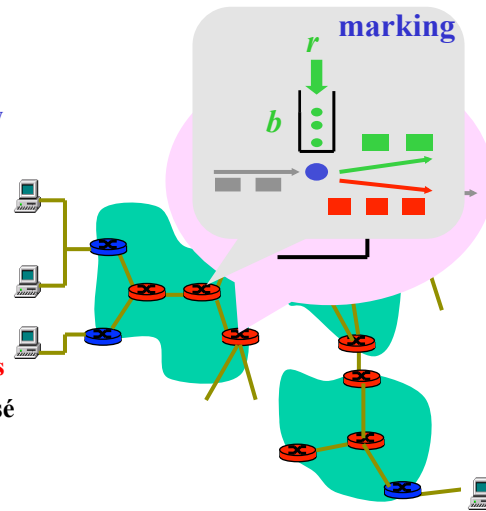
## Architecture Diffserv

### Edge router:

- Gestion du trafic par flux **per-flow**
- Paquets marqués comme **in-profile** et **out-profile**

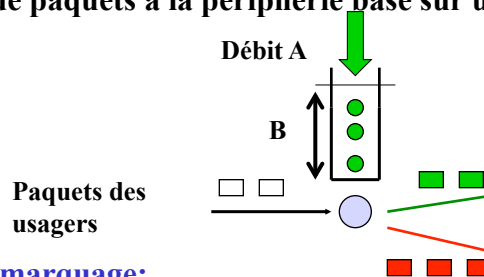
### Core router:

- Gestion du trafic par flux **per class**
- Mémorisation et séquençement basé sur le **marquage** sur les bords
- La préférence est donnée aux paquets **in-profile**
- Assured Forwarding



## Marquage dans les routeurs de périphérie (Edge-router)

- **profil**: débit A pre-négocié rate A, sceau de dimension B
- Marquage de paquets à la périphérie basé sur un profil **par-flux**



### Possibilités de marquage:

- Marquage basé sur la classe: les paquets de différentes classes sont marqués différemment
- Marquage intra classes: la portion du flux qui est conforme est marquée différemment de celle qui est non conforme

## Classification and Conditionnement

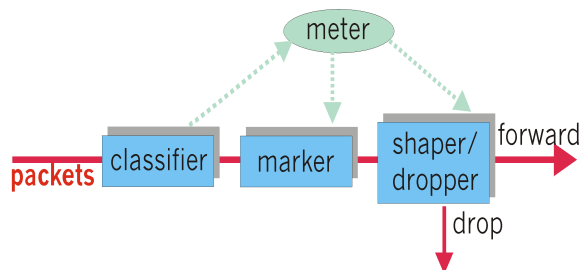
- Le paquet est marqué avec le champ Type of Service (TOS) dans IPv4, et Traffic Class dans IPv6
- 6 bits sont utilisés pour le Differentiated Service Code Point (DSCP). Il détermine quel PHB le paquet va recevoir
- 2 bits sont actuellement inutilisés (CU)



## Classification and Conditionnement

On peut vouloir limiter l'injection de trafic d'une certaine classe:

- L'utilisateur déclare son profil de trafic (e.g., débit, grandeur du burst)
- Le trafic est mesuré, et "profilé" si non conforme



## Forwarding (PHB)

- PHB permet de d'accorder à chaque catégorie de trafic le type de transmission.
- PHB ne spécifie pas quels mécanismes à utiliser pour s'assurer d'un comportement avec une certaine performance
- Exemples:
  - La classe A obtient  $x\%$  de la LB d'une ligne sortante sur un intervalle de temps spécifié
  - Les paquets de classe A partent en premier avant les paquets de classe B

## Forwarding (PHB)

Deux PHBs font actuellement l'objet de débats à l'IETF:

- **Expedited Forwarding**: le débit de départ d'une classe égale ou excède un débit spécifique
  - Ligne logique avec un débit minimum garanti
- **Assured Forwarding**: 4 classes de trafic
  - Chacun a une LB minimum garantie
  - Chacun avec trois catégories de "préférence de perte"



## Multimedia: Résumé

- Exigence des applications MM
- Faire au mieux avec le "best effort"
- Mécanismes d'ordonnancement et de contrôle
- Prochaine génération d'Internet:  
Intserv, RSVP, Diffserv

## Exercices supplémentaires