

à l'Université Pierre et Marie Curie, le 19 octobre 2004

## M2 Informatique Réseaux

### Multimédia et Qualité de Service

# Cours 4 : Streaming et Signalisation

Timur FRIEDMAN

transparents adaptés de *Computer Networking*

copyright 1996-2002 J.F. Kurose et K.W. Ross

une partie des traductions grâce à Kavé SALAMATIAN

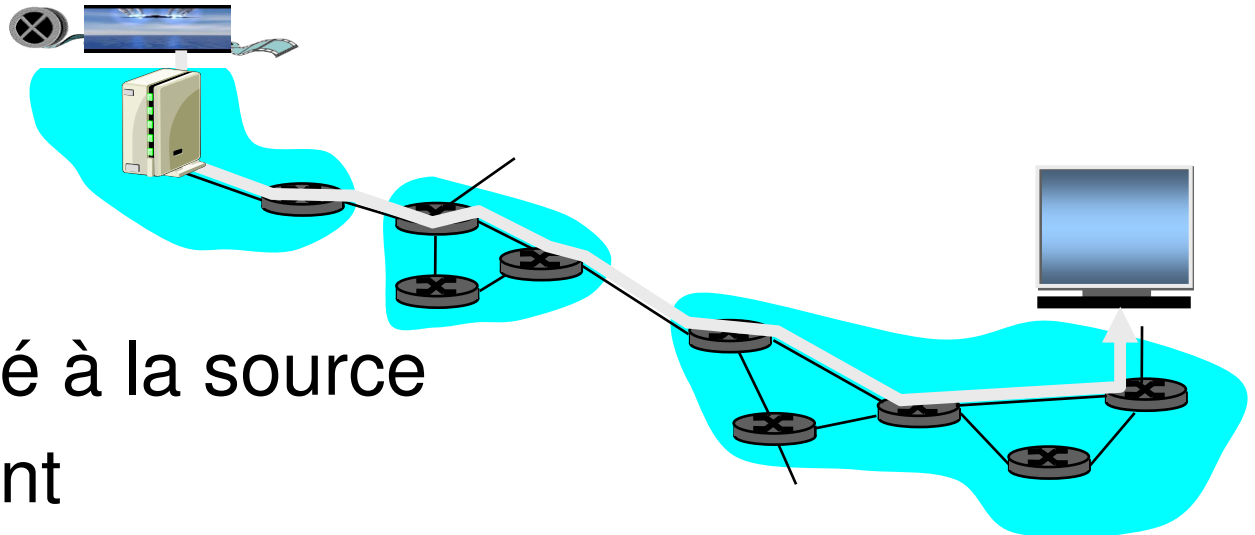
# Plan

- Applications multimédia
- Streaming d'audio et vidéo stockés
  - RTSP
- Exemple de l'interactive: téléphonie sur IP
- Applications en temps-réel
  - SIP

# Applications multimédia

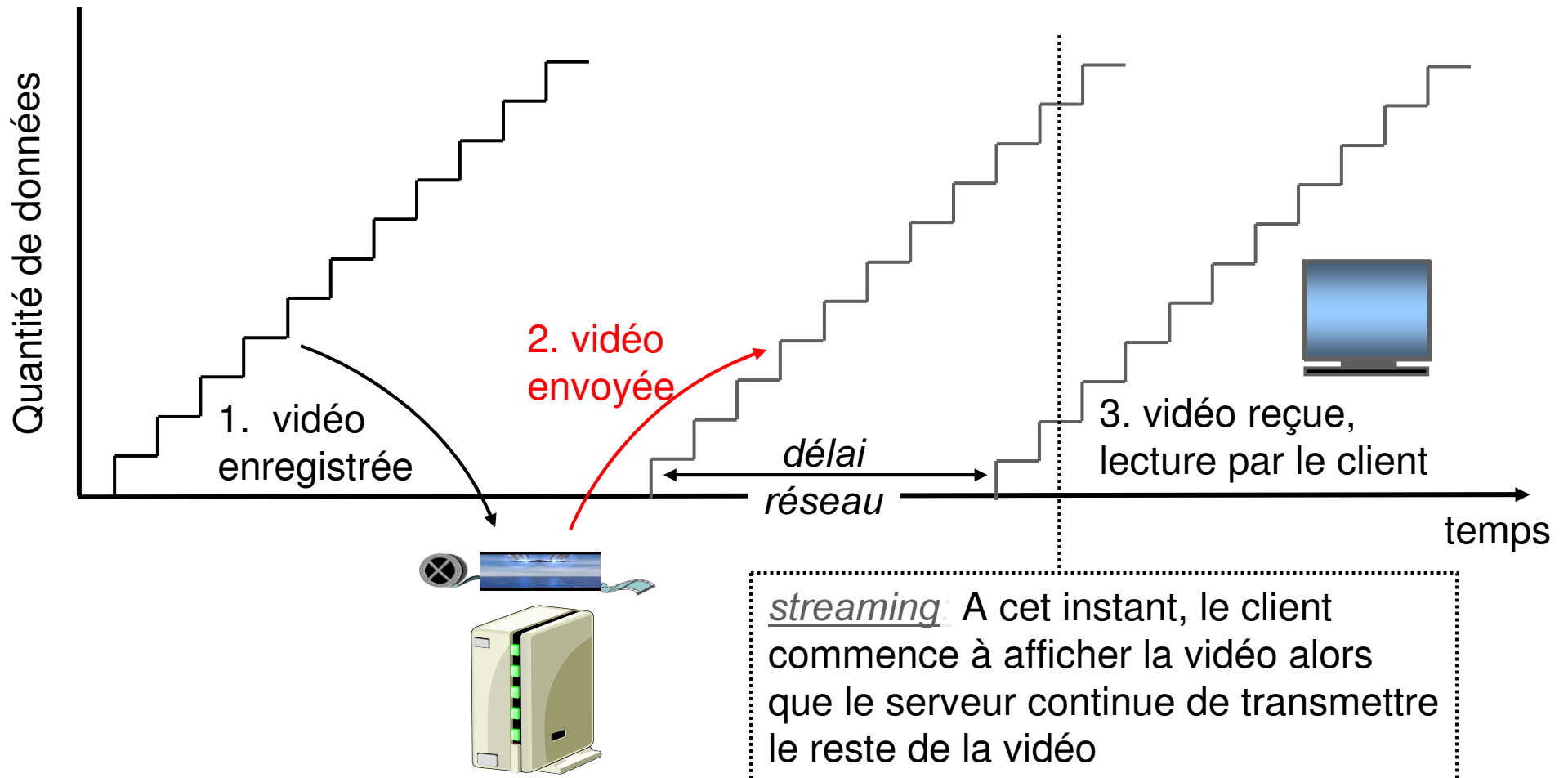
- Les classes d'applications
  - *streaming* d'audio et vidéo stockés
  - *streaming* d'audio et vidéo en direct
  - audio et vidéo interactives en temps réel
- *streaming* = la lecture en transit
  
- Les caractéristiques
  - sensible aux délais
    - de bout en bout
    - la gigue
  - résistant aux pertes

# Streaming d'audio et vidéo stockés

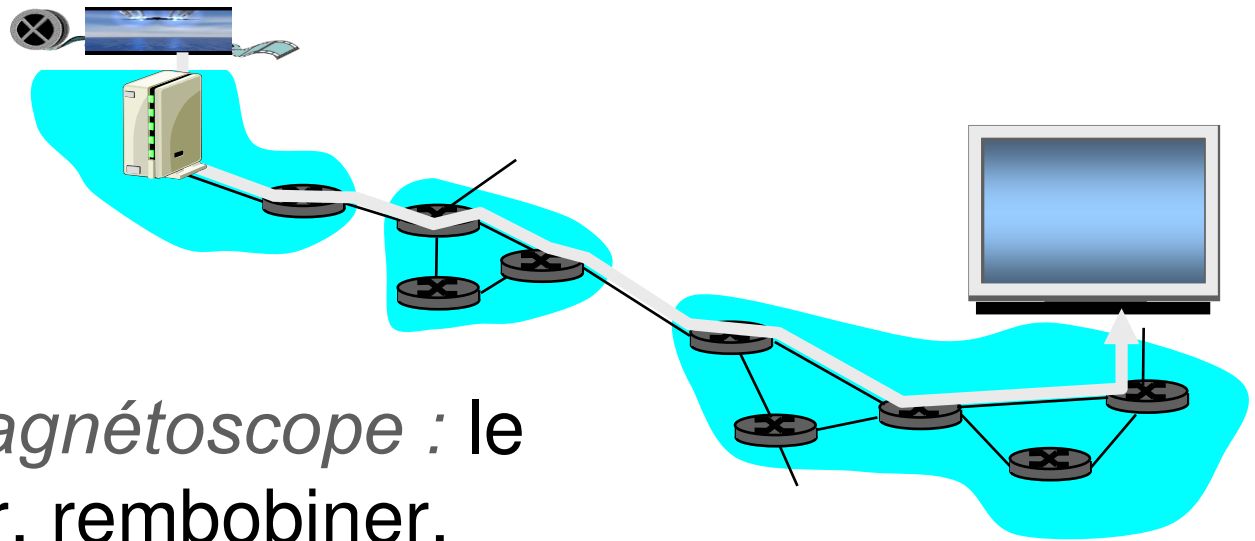


- media est stocké à la source
- transmis au client
- streaming : le client commence la lecture *avant* la fin de la réception
  - contrainte temporelle pour les données qui restent à transmettre :
    - qu'elles arrivent à temps pour la lecture

# Le streaming



# L'interactivité et les données stockées



- *Fonctionnalité magnétoscope* : le client peut arrêter, rembobiner, avancer
  - délai initial de 10 sec OK
  - 1-2 sec avant l'exécution de la commande OK

# Streaming d'audio et vidéo en direct

## Exemples

- radio par Internet
- transmission des matches sportives

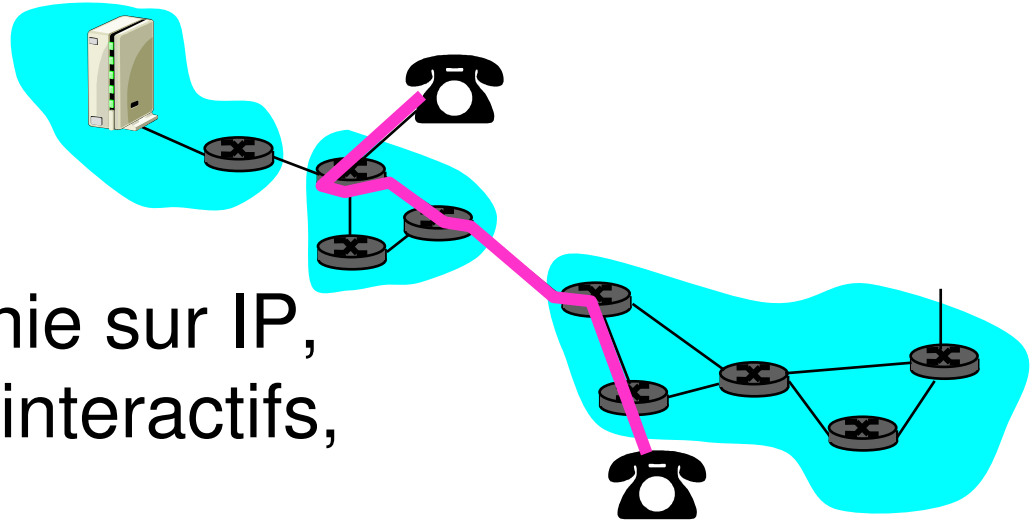
## Streaming

- mémoire tampon pour la lecture
- la lecture peut s'effectuer plusieurs dizaines de secondes après la transmission
- il y a toujours des contraintes temporelles

## Interactivité

- avancer impossible
- rembobiner, pause possible

# Audio et vidéo interactives en temps réel



- **applications** : téléphonie sur IP, visioconférence, jeux interactifs, mondes virtuels
- **délai de bout en bout requis** :
  - audio : < 150 ms bon, < 400 ms OK
  - vidéo : < 150 ms
- Le délai inclue le traitement par l'application :
  - mise en paquets
  - compression



# Plan

- Applications multimédia
- Streaming d'audio et vidéo stockés
  - RTSP
- Exemple de l'interactive: téléphonie sur IP
- Applications en temps-réel
  - SIP

# Streaming d'audio et vidéo stockés

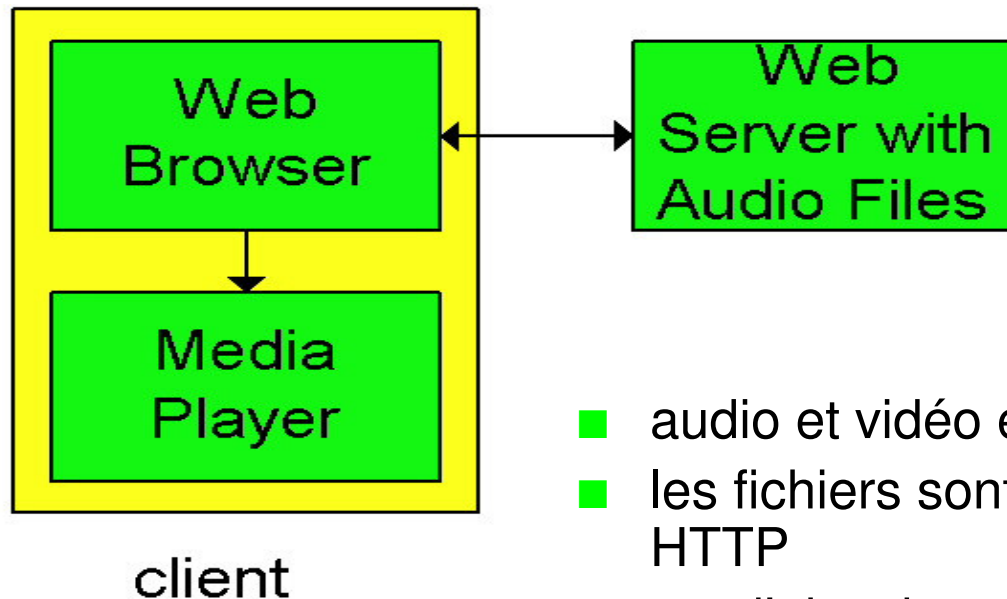
Techniques de streaming  
au niveau applicatif pour  
faire face à un Internet  
moindre effort :

- mise en mémoire  
tampon du côté client
- UDP et TCP mixtes
- plusieurs encodages  
possible pour le  
multimédia

## Lecteur Média

- suppression de la gigue
- décompression
- dissimulation d'erreurs
- interface graphique  
avec contrôles pour  
l'interactivité

# L'approche la plus simple

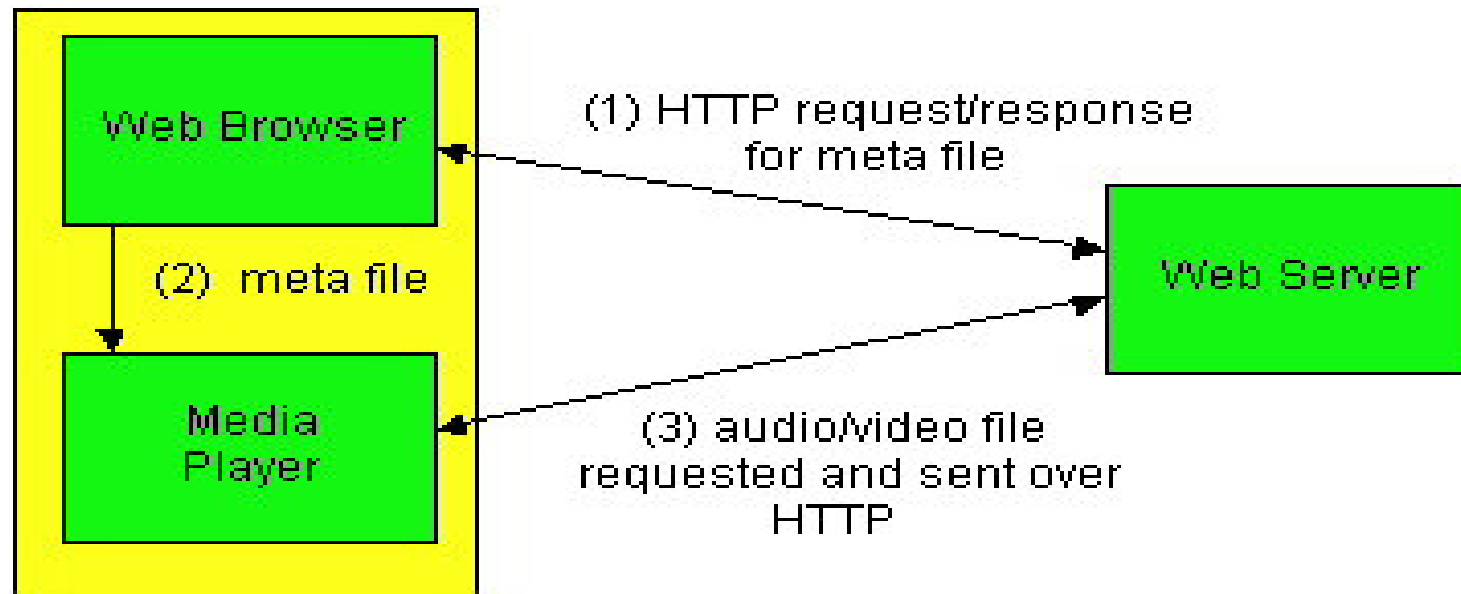


- audio et vidéo enregistré dans des fichiers
- les fichiers sont transmis en tant qu'objets HTTP
  - d'abord reçus complètement chez le client
  - transfert au lecteur après

**audio et vidéo ne sont pas lise en transit :**

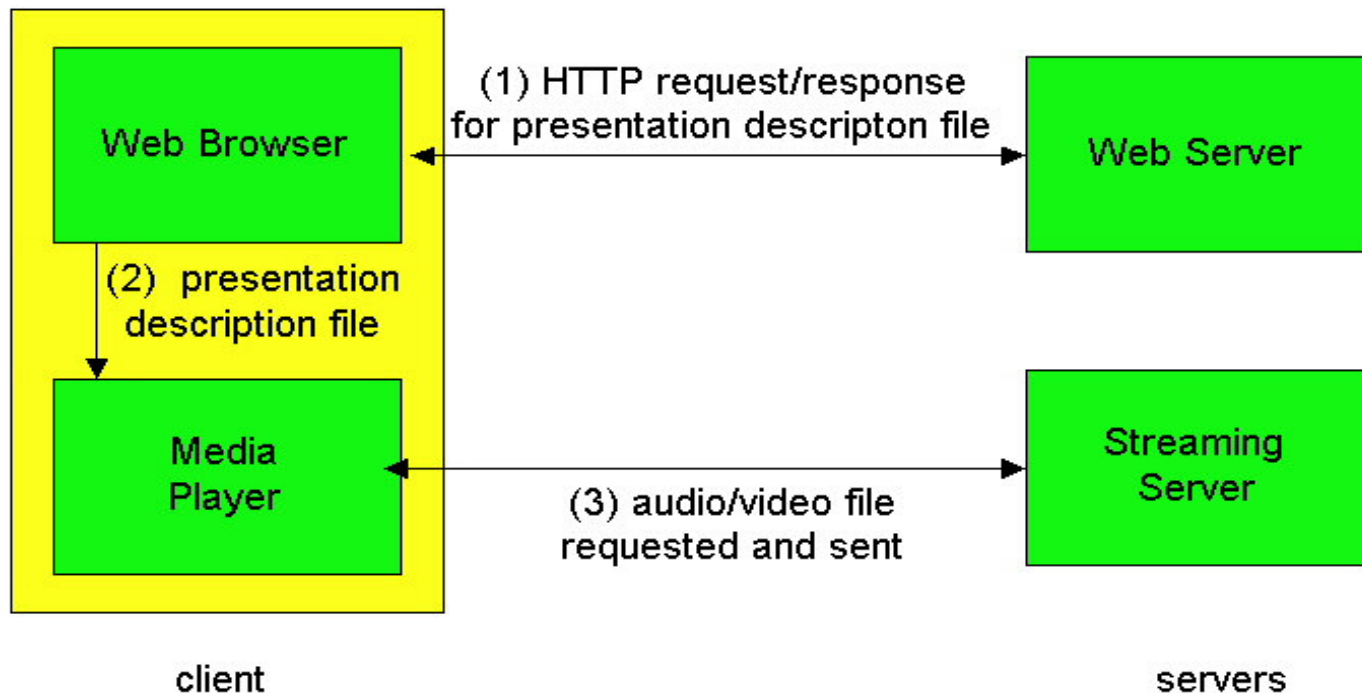
- pas de pipeline, délai de réception important avant l'affichage

# Une approche streaming



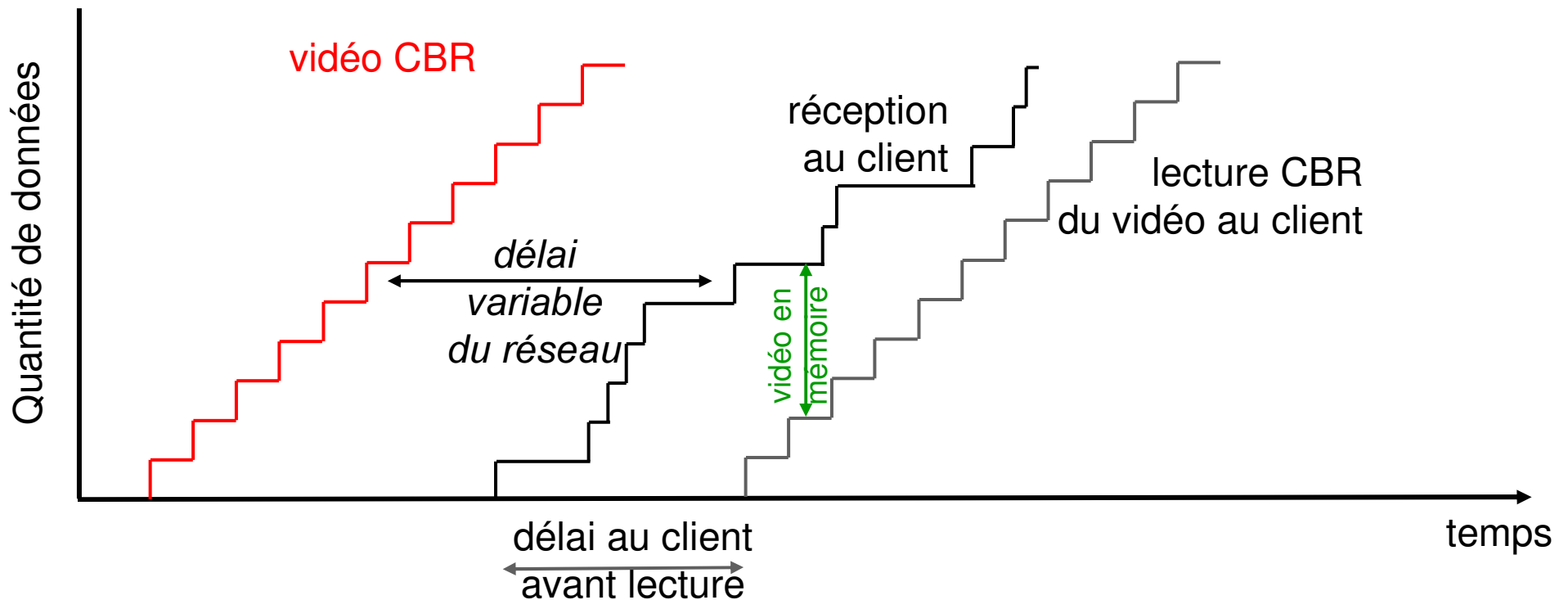
- le navigateur reçoit un **metafichier** par GET
- le navigateur exécute le lecteur « plug-in » et lui passe le metafichier
- le lecteur contacte le serveur
- le serveur envoie le flot audio/vidéo au lecteur en **streaming**

# Une approche avec un serveur streaming



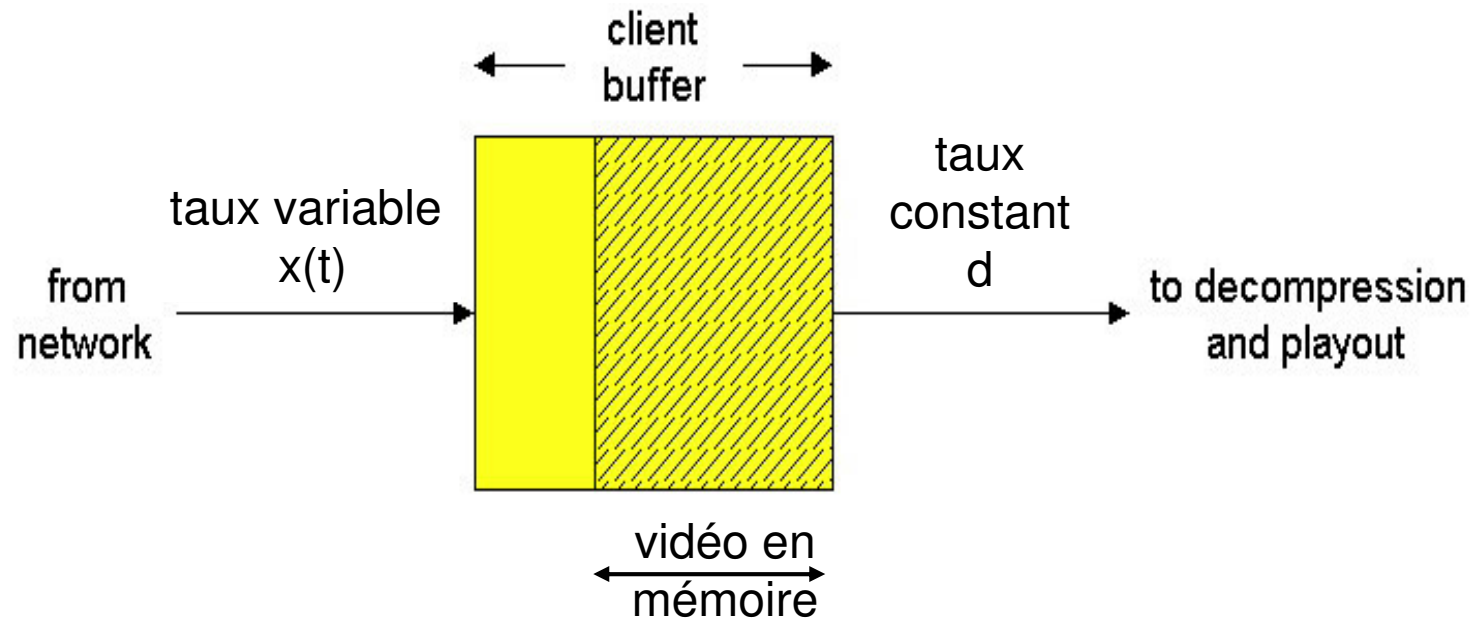
- Cet architecture permet l'utilisation d'un protocole non-HTTP entre le serveur et le lecteur multimédia.
- Permet l'utilisation de UDP au lieu de TCP.

# Mise en mémoire tampon du client



- Compensation pour le délai du réseau, la gigue :
  - mise en mémoire
  - délai avant la lecture au client

# Détails de la mémoire



- La mémoire tampon compense la gigue du délai

# Streaming du multimédia par UDP

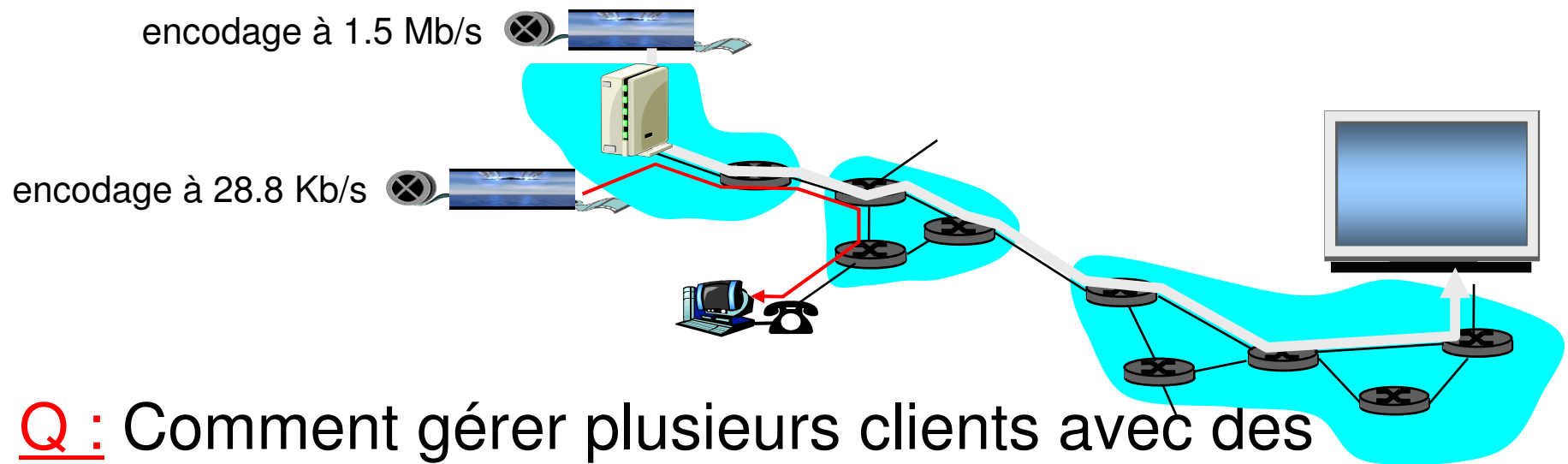
- Le serveur émet avec un débit adéquat pour le client
  - spécialement s'il ignore les problèmes de congestion
  - débit émetteur = taux d'encodage (CBR)
  - débit récepteur = CBR – pertes de paquets
- Délais de démarrage courts (2-5 seconds) pour compenser la gigue du délai
- Compensation de pertes : si la contrainte temporelle le permet



# Streaming du multimédia par TCP

- Le serveur émet au débit maximal permis par TCP
  - Le débit récepteur varie à cause du contrôle de congestion TCP
- Plus qu'on ajoute du délai au client, plus la lecture du vidéo devient lisse
- HTTP/TCP passent plus facilement à travers les pare-feux

# Clients hétérogènes



Q : Comment gérer plusieurs clients avec des débits de réception hétérogènes ?

- 28.8 Kb/s connexion par modem
- 100Mb/s sur Ethernet

R : Le serveur garde et transmet de multiples copies de la vidéo encodé, avec des débits différents

# RTSP « Real Time Streaming Protocol »

## HTTP

- Pas ciblé vers le multimédia
- Pas de commandes pour avancer, rembobiner, etc.

## RTSP : RFC 2326

- Protocole client-serveur au niveau applicatif.
- Fonctionnalités fournies au client: avancer, rembobiner, pause, repositionnement, etc....

## Limites de RTSP :

- pas de définition du codage audio/vidéo
- pas de définition de la couche transport : le media peut être transporté par UDP ou par TCP
- pas de définition de la manière dont le client doit faire la mise en mémoire tampon d'audio ou vidéo

# RTSP: un contrôle hors bande

## Comme avec FTP :

- Dans FTP, un fichier est transmis sur une connexion TCP.
- La signalisation de contrôle (changement de répertoire, effacement de fichier, renommement de fichier, etc.) est transmise sur une connexion TCP indépendante.
- Les canaux hors-bande et intrabande utilisent des numéros de ports différents.

## RTSP :

- Le flux multimédia est l'intrabande
- Les messages de contrôle RTSP utilisent un numéro de port hors-bande réservé.
  - Port 554

# RTSP : un exemple

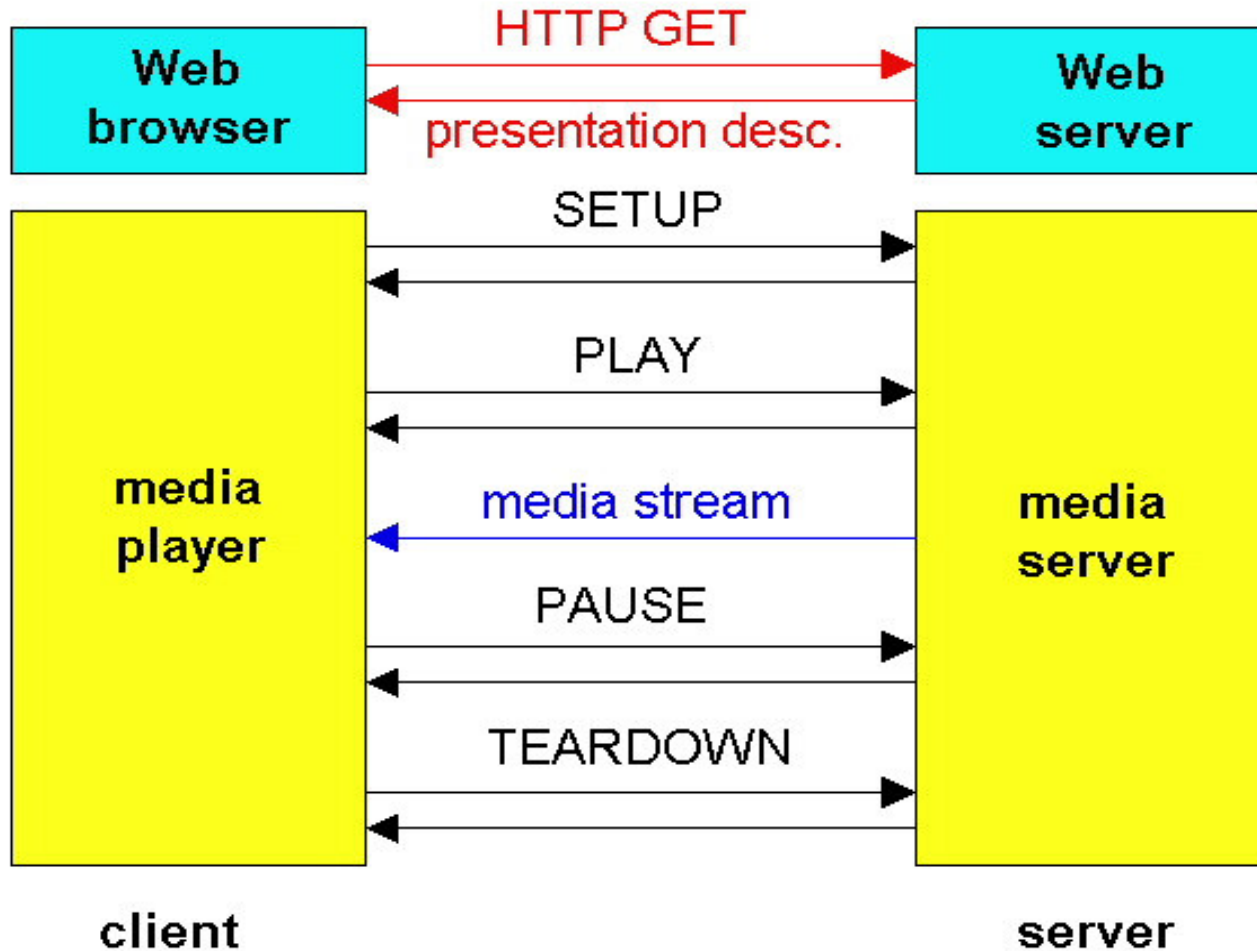
## Scénario:

- un metafichier est transmis au navigateur
- le navigateur exécute le lecteur
- le lecteur construit :
  - une connexion de contrôle RTSP
  - une connexion pour les données multimédia

# Metafichier : exemple

```
<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>
```

# Fonctionnement du RTSP



# RTSP : exemple d'un échange client-serveur

C: SETUP rtsp://audio.example.com/twister/audio RTSP/1.0  
Transport: rtp/udp; compression; port=3056; mode=PLAY

S: RTSP/1.0 200 1 OK  
Session 4231

C: PLAY rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0  
Session: 4231  
Range: npt=0-

C: PAUSE rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0  
Session: 4231  
Range: npt=37

C: TEARDOWN rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0  
Session: 4231

S: 200 3 OK



# Plan

- Applications multimédia
- Streaming d'audio et vidéo stockés
  - RTSP
- Exemple de l'interactive: téléphonie sur IP
- Applications en temps-réel
  - SIP

# Applications interactives en temps réel

- téléphonie PC à PC
  - fourni par les services de messagerie instantanée
- PC à téléphone
  - Dialpad
  - Net2phone
- visioconférence avec Webcam

Examinons l'exemple  
de téléphonie PC à  
PC

# Multimédia interactif : téléphonie sur IP

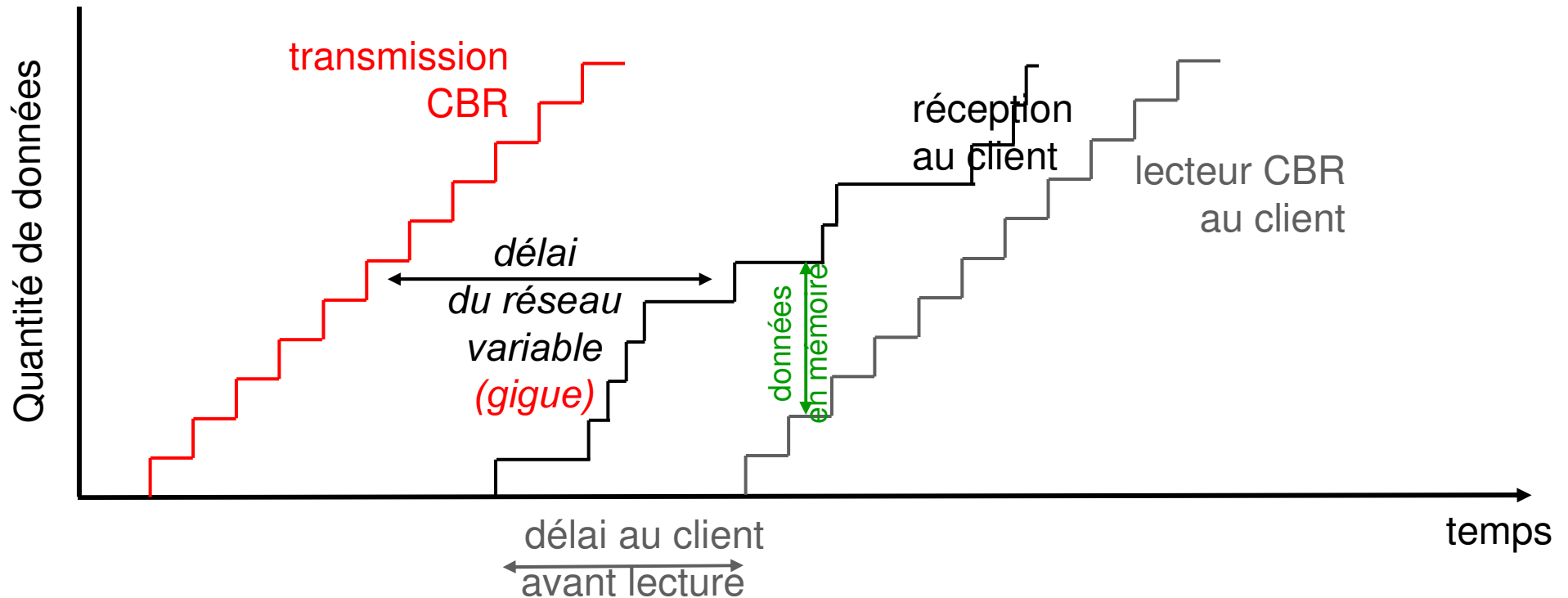
## Notre exemple :

- La voix alterne des périodes de son et de silence.
  - 64 kb/s pendant une période de son
- Des paquets ne sont générés que durant des périodes de son
  - échantillons de 20 msec à 8 Kbytes/sec: 160 bytes par trame
- Entête applicative ajouté à chaque trame.
- Trame+entête sont encapsulés dans des segments UDP.
- L'application envoi des segments UDP par le biais des sockets chaque 20 msec pendant une période de son.

# Multimédia interactif : les pertes et les délais

- **pertes de paquets** : Les paquets IP peuvent être perdu par cause de congestion (débordement des files d'attente)
- **délai important** : Un paquet IP peut être considéré comme perdu s'il arrive trop tard au récepteur
  - délais dus au traitement, files d'attente, terminaux (récepteur, émetteur)
  - délai maximal tolérable : 400 ms audio, vidéo
- **tolérance aux pertes** : dépend du type de compression, de la compensation, de l'existence de FEC.
  - taux de pertes entre 1% et 10% peuvent être toléré

# Gigue de délai



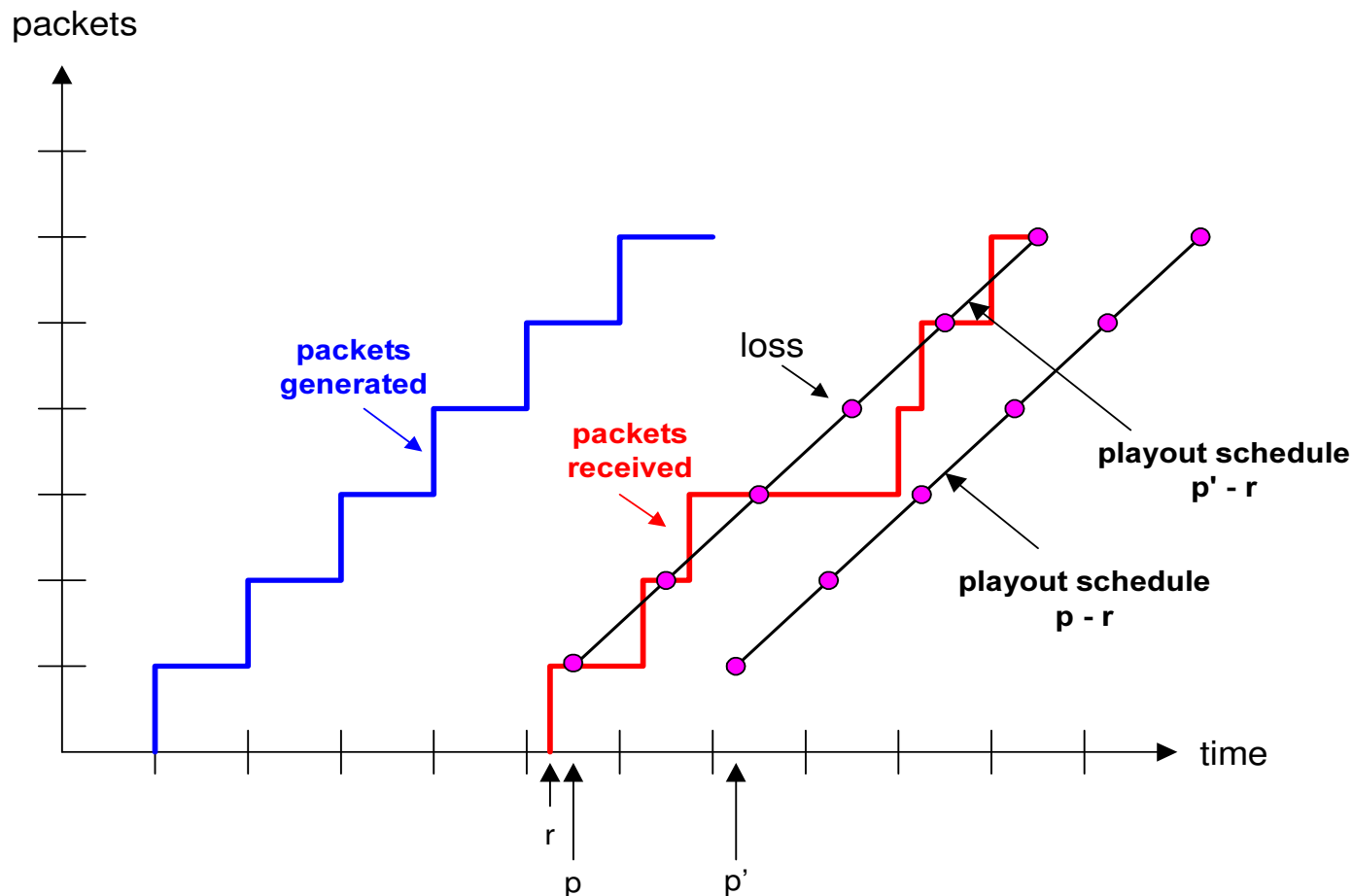
- La différence entre les délais de deux paquets consécutives peut être plus ou moins que 20 msec

# Tampons de réception fixe

- Le récepteur diffuse chaque paquet exactement  $q$  msec après que la trame a été généré.
  - Si la trame a une estampille à l'instant  $t$  : il est diffusé à l'instant  $t+q$  .
  - Si la trame arrive après  $t+q$  : la trame est considérée perdue
- Balance pour  $q$ :
  - grand  $q$ : moins de pertes
  - petit  $q$ : meilleur interactivité

# Délai de diffusion fixe

- L'émetteur génère un paquet toutes les 20 msec durant la période de son.
- Le premier paquet est reçu à l'instant  $r$
- Un programme de diffusion : commencer à  $p$
- Deuxième programme de diffusion: commencer à  $p'$



# Tampon adaptatif

## ■ Buts :

- minimiser le délai de diffusion
- maintien un taux de pertes bas

## ■ Stratégie : adapter le délai de diffusion

- Estimer le délai réseau, régler le délai de diffusion au début de chaque période de son.
- Les périodes de silence sont compressées et élongées.
- Les trames sont toujours diffusés chaque 20 msec pendant les périodes de son.



# Tampon adaptatif : calcul

$t_i$  = timestamp of the  $i$ th packet

$r_i$  = the time packet  $i$  is received by receiver

$p_i$  = the time packet  $i$  is played at receiver

$r_i - t_i$  = network delay for  $i$ th packet

$d_i$  = estimate of average network delay after receiving  $i$ th packet

Estimation dynamique du délai moyen au récepteur :

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

$u$  est constant (e.g.,  $u = .01$ ).

# Tampon adaptatif : calcul bis

L'écart type,  $v_i$ , peut aussi être estimé :

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

$d_i$  et  $v_i$  sont calculés pour chaque paquet, mais sont appliqués au début d'une période de parole.

Pour le premier paquet d'une période de parole l'instant de diffusion est :

$$p_i = t_i + d_i + Kv_i$$

pour  $K$  un constant positif.

Les autres paquets de cette même période de parole, sont diffusés dans une manière périodique.

# Tampon adaptatif

Q : Comment le récepteur détermine que le paquet est le premier d'une période de parole ?

- Sans pertes, le récepteur scrute les estampilles successive.
  - Si la différence entre deux estampilles successive  $> 20$  msec --> début de période de parole.
- Avec des pertes, il faut faire attention au numéros de séquence aussi.
  - différence entre deux estampilles  $> 20$  msec **et** numéros de séquence successive --> début de période.
- Voir RTP/RTCP

# Résistance contre les pertes (1)

## « forward error correction » (FEC) : procédure simple

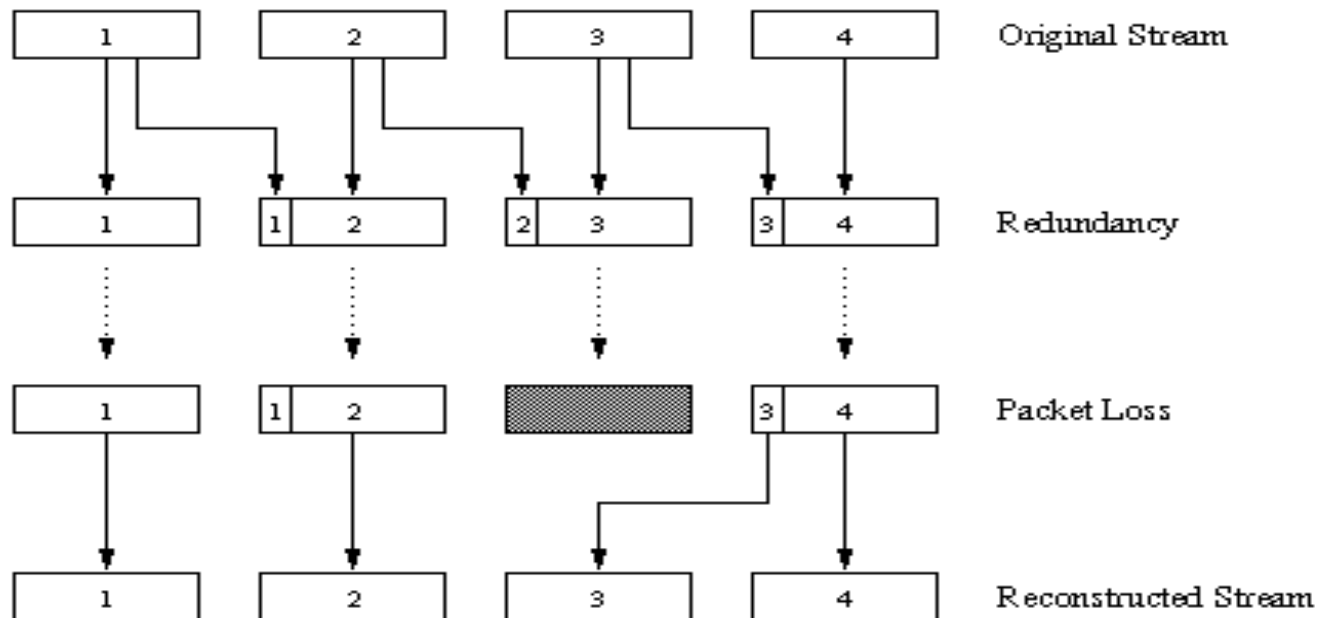
- pour chaque groupe de  $n$  trames, on crée une trame redondante
  - XOR sur les  $n$  trames
- on envoie  $n+1$  trames
  - le débit augmente par  $1/n$
- on peut reconstruire les  $n$  trames originales si on reçoit  $n$  des  $n+1$  trames
- Le délai de diffusion est le temps de recevoir toutes les  $n+1$  trames
- Echange :
  - $n$  plus grand
    - moins on gaspille la bande passante
    - plus de délai de diffusion
    - plus grande la probabilité de perdre 2 trames sur  $n+1$

# Résistance contre les pertes (2)

## 2ème procédure FEC

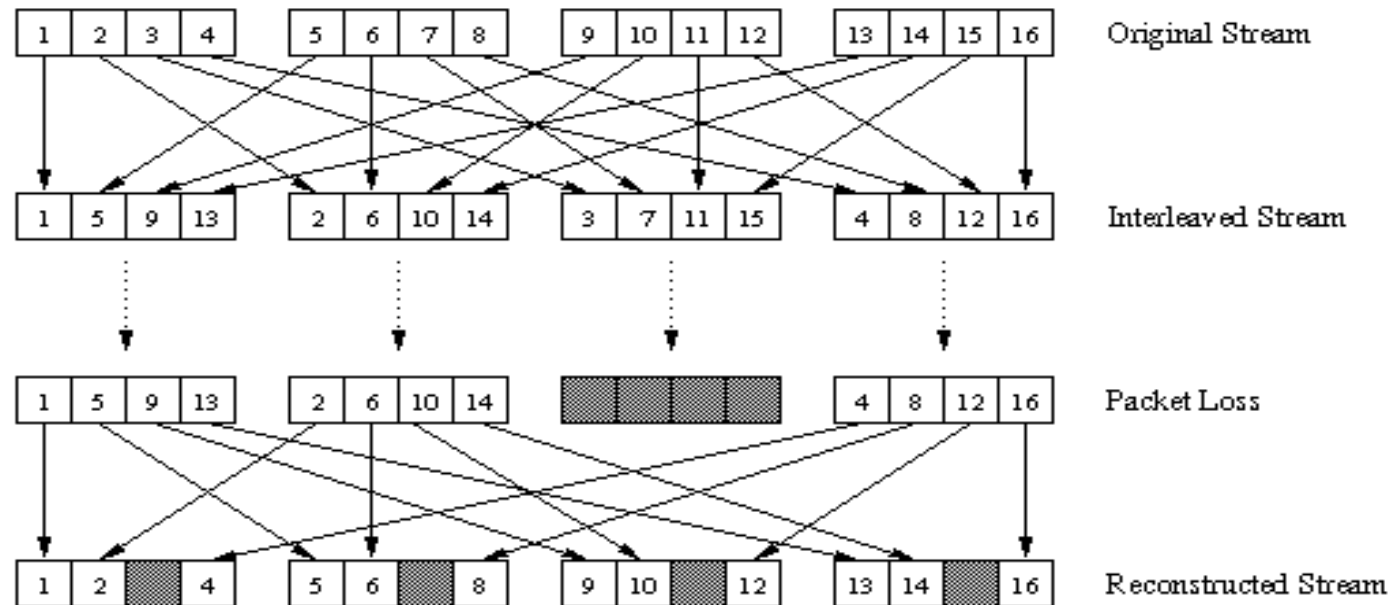
superposition d'un flux de qualité inférieur

- par exemple, flux PCM à 64 kb/s et flux redondant GSM à 13 kbps.



- Le récepteur peut cacher toute perte non-consecutive.

# Résistance contre les pertes (3)



## 3ème procédure : entrelacement

- les trames sont découpées en unités plus petits
- par exemple, 4 unités de 5 msec par trame
- chaque paquet contient des petits unités de plusieurs trames
- si un paquet est perdu, on reçoit toujours le plupart de chaque trame
- pas de surdébit de redondance
- le délai de diffusion augmente

# Bilan sur le multimédia interactif

- on utilise UDP
  - éviter les délais induits par contrôle de congestion TCP
- le récepteur adapte son délai de diffusion
  - compensation pour les délais aléatoires
- le serveur gère le débit du flux en fonction de la bande passante disponible sur le chemin client-serveur
  - choix parmi des encodages à différents débits
  - la choix peut être dynamique
- compensation pour les pertes
  - FEC, entrelacement
  - des retransmissions, si le temps permis
  - cacher des erreurs: redondances des flux

# Plan

- Applications multimédia
- Streaming d'audio et vidéo stockés
  - RTSP
- Exemple de l'interactive: téléphonie sur IP
- Applications en temps-réel
  - SIP



# SIP

- Session Initiation Protocol (RFC 3263)

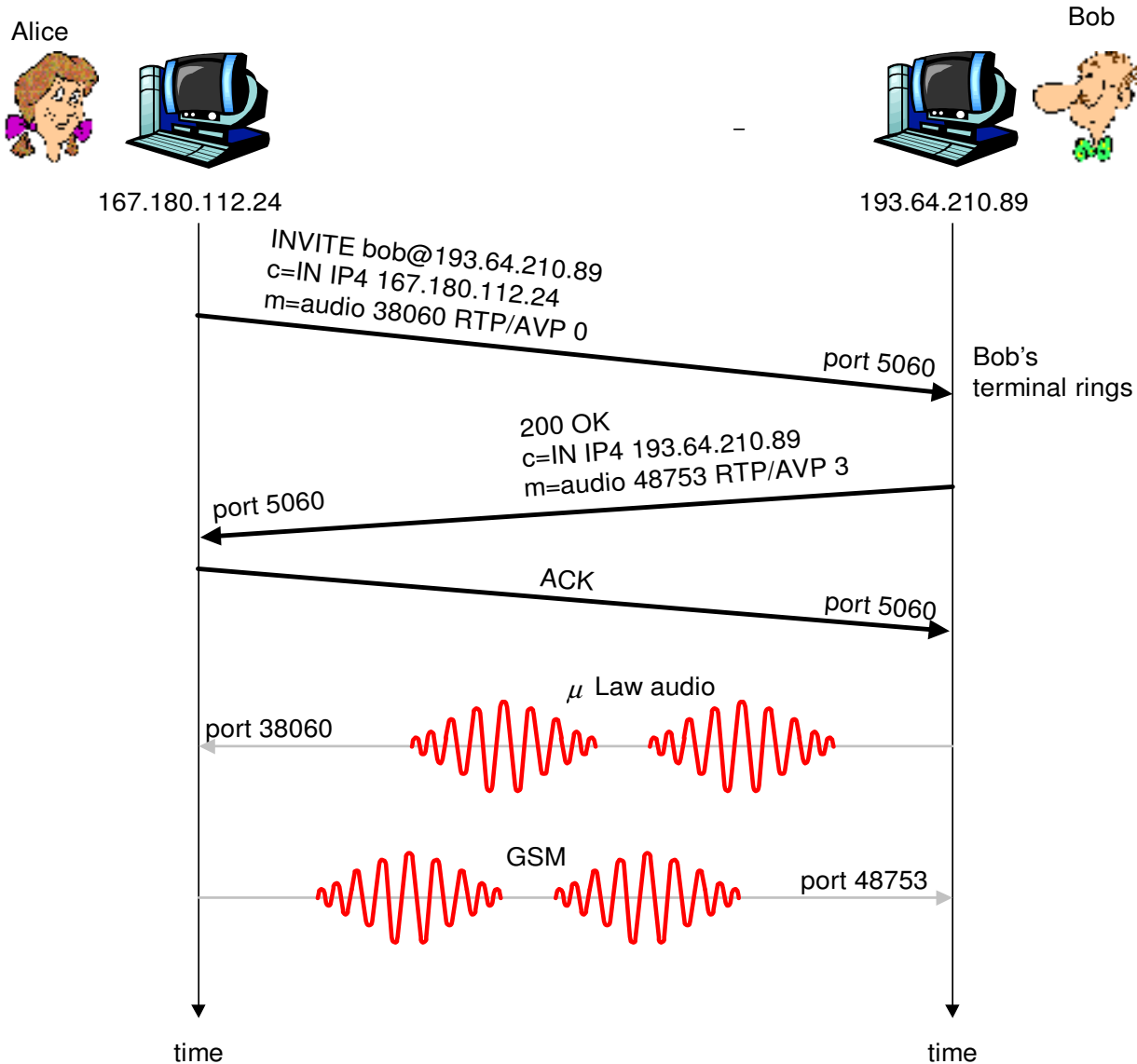
## La philosophie de SIP

- Tous les appels téléphoniques et tous les visioconférences vont avoir lieu dans l'Internet.
- On identifie les participants par nom ou par mél, non par numéro de téléphone.
- On peut toujours joindre son correspondant
  - n'importe où il se trouve
  - n'importe lequel adresse IP il utilise.

# Les services SIP

- Etablissement de connexion
  - Signalisation au correspondant qu'on veut l'appeler
  - Signalisation pour se mettre d'accord sur les medias et l'encodage
  - Signalisation pour la terminaison d'un appel.
- Recherche de l'adresse IP du correspondant
  - Mappage entre l'identifiant mnémonique et l'adresse IP actuelle
- Gestion de l'appel
  - Ajout de nouveaux flux de media en cours d'appel
  - Changement d'encodage en cours d'appel
  - Invitations aux autres
  - Transfert d'appel, mise en garde

# Exemple : établissement d'appel



- Alice invite Bob à communiquer (port 5060)
  - son message SIP invite contient son numéro de port et son adresse IP
  - il indique que Alice préfère de recevoir l'encodage PCM ulaw
- Le message 200 OK de Bob indique
  - son adresse IP et numéro de port pour la communication
  - sa préférence pour l'encodage GSM
- Les messages SIP peuvent être TCP or UDP; ici RTP/UDP.

# Etablissement d'appel bis

- Négociation de l'encodage:
  - Supposons que Bob ne dispose pas de l'encodeur PCM ulaw.
  - Bob répond par 606 Not Acceptable Reply. Il fournit la liste des encodeurs dont il peut s'en servir.
  - Alice peut ensuite envoyer un nouveau message INVITE, proposant un encodeur adapté.
- Rejet d'appel
  - Bob peut rejeter des appels avec les réponses replies « occupé », « absent », « crédit insuffisant », ou « interdit ».

# Exemple d'un 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
```

A noter:

- syntaxe HTTP
- sdp = session description protocol
- unicité d'identifiants Call-ID

- Dans ce cas, Alice ne connaît pas l'adresse IP de Bob.

- Des serveurs intermediares sont nécessaires.

- Alice envoie et reçoit des messages SIP sur le numéro de port par défaut 5060.

- Alice indique dans l'entête Via que son client SIP envoie et reçoit par UDP.

# Annuaire et localisation

- Comment appeler si on connaît seulement le nom ou l'adresse mél ?
- Comment connaître l'adresse IP actuel du correspondant ?
  - le correspondant se déplace
  - il utilise le protocole DHCP
  - il a plusieurs appareils IP (PC, PDA, voiture)
- L'information fourni peut varier selon :
  - l'horaire (travail, maison)
  - celui qui appel (éviter qu'un appel professionnel est diriger vers le foyer)
  - l'état du correspondant (quand il est occupé, on dirige les appels vers le répondeur)

## Les serveurs SIP :

- SIP registrar
- SIP proxy

# SIP Registrar

- Quand Bob allume son client SIP, le client envoie un message SIP REGISTER message au Registrar Server (comme dans la messagerie instantanée)

message REGISTER :

```
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
```

# SIP Proxy

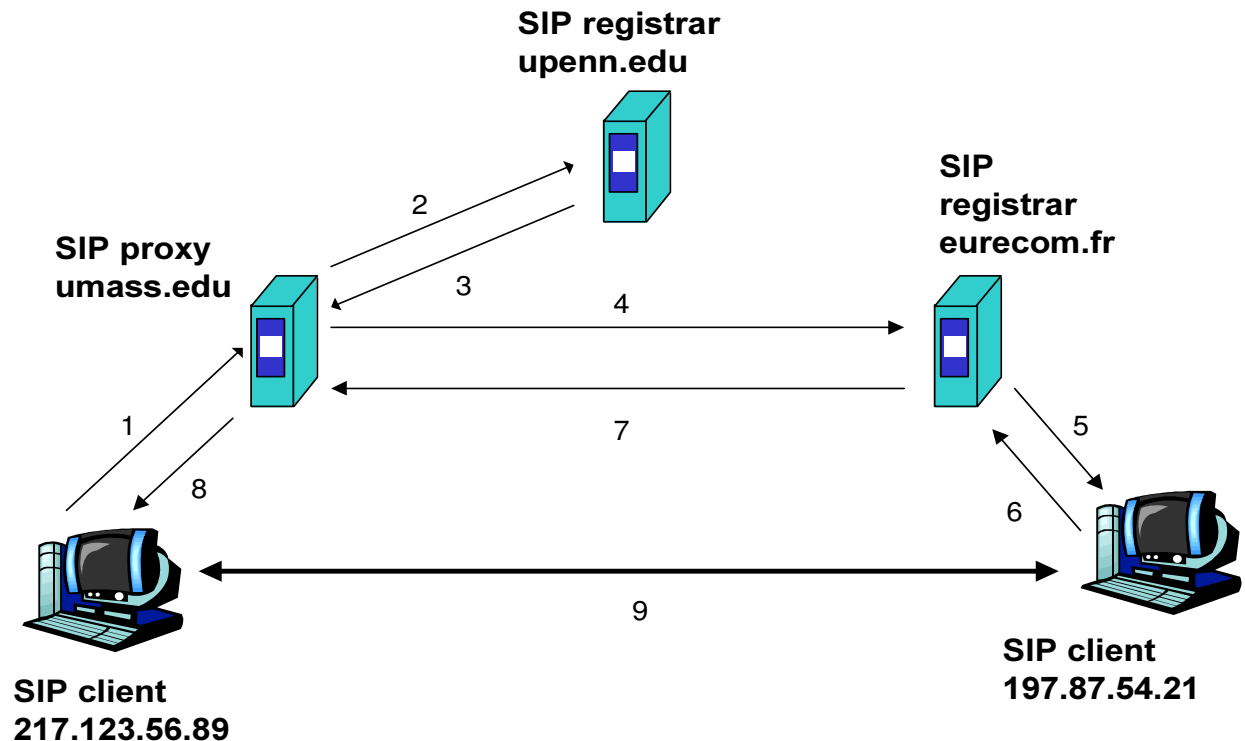
- Alice envoie un message INVITE à son serveur proxy
  - il contient l'adresse sip:bob@domain.com
- Le proxy est responsable de l'acheminement des messages SIP vers le correspondant
  - les messages peuvent passer par plusieurs serveurs proxy
- Le correspondant envoie sa réponse via le même ensemble de serveurs proxy.
- Le proxy fournit la réponse SIP à Alice
  - la réponse contient l'adresse IP de Bob
- A noter : un proxy se ressemble à un serveur DNS



# Exemple

Appel de jim@umass.edu  
vers keith@upenn.edu

(1) Jim envoie un message INVITE au SIP proxy de umass. (2) Le proxy renvoi le message vers le SIP registrar de upenn. (3) Le registrar de upenn répond avec un message REDIRECT, indiquant qu'il faut contacter keith@eurecom.fr



(4) Le proxy umass envoie un INVITE au registrar eurecom. (5) Le registrar eurecom renvoi le message INVITE à 197.87.54.21, sur lequel tourne le client SIP de keith. (6-8) réponse SIP (9) media envoyé directement entre les clients.

**A noter:** il y a aussi des acquittements SIP (pas montrés).

# Comparaison avec H.323

- H.323 est un autre protocole de signalisation pour l'interactive en temps réel
- H.323 est un ensemble monolithique de protocoles pour le visioconférence : signalisation, enregistrement, contrôle d'admission, transport, et encodage.
- SIP est un composant modulaire. Il est compatible avec RTP. Il peut fonctionner avec d'autres protocoles et services aussi.
- H.323 a été développé à l'UIT (téléphonie).
- SIP a été développé à l'IETF: Il est inspiré par HTTP.
- SIP est plus simple (et alors plus efficace ?)