

Applications réseau

Cours 4 : Réseaux et multimédia

Florian Sikora
florian.sikora@dauphine.fr

LAMSADE

M1 apprentissage

Adapté des slides de Kurose & Ross

Cours 4 : Multimédia

Applications réseau multimédias

Streaming de données enregistrées

Voix sur IP

Streaming temps réel

Objectifs

- ▶ Comprendre le transfert de données multimédias.
- ▶ Développement récent.
- ▶ Q : Exemples ?

Objectifs

- ▶ Comprendre le transfert de données multimédias.
- ▶ Développement récent.
- ▶ Q : Exemples ?
 - ▶ Jeux en ligne.
 - ▶ Voix sur IP.
 - ▶ Vidéo en streaming.
 - ▶ Radio par Internet.
 - ▶ ...

Objectifs

- ▶ Comprendre le transfert de données multimédias.
- ▶ Développement récent.
- ▶ Q : Exemples ?
 - ▶ Jeux en ligne.
 - ▶ Voix sur IP.
 - ▶ Vidéo en streaming.
 - ▶ Radio par Internet.
 - ▶ ...
- ▶ Sensibles à la durée de transmission et délais.
- ▶ Tolérant à la perte.
- ▶ Enjeu car Internet est “best-effort” .

Enjeux

- ▶ La vidéo demande une très grosse bande passante !
- ▶ On compare :
 - ▶ Frank regarde sur Facebook une photo de 200Ko toutes les 10 secondes.
 - ▶ Mélanie écoute de la Musique sur deezer encodée à 128 Kbps.
 - ▶ Virginie regarde de la Vidéo encodée à 2Mbps.
- ▶ Après 1h07 :
 - ▶ F : 80 Mo.
 - ▶ M : 64 Mo.
 - ▶ V : 1 Go !!!
- ▶ La vidéo devrait représenter bientôt 90% du trafic internet...
- ▶ Compression possible...

Cours 4 : Multimédia

Applications réseau multimédias

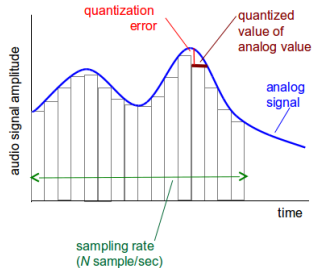
Streaming de données enregistrées

Voix sur IP

Streaming temps réel

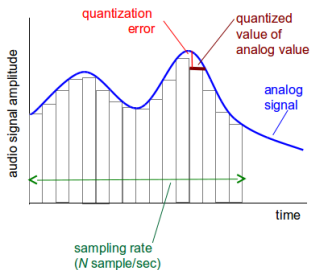
Audio

- ▶ Audio : signal analogique : doit être échantillonné.
- ▶ Échantillonnage à rythme fixe :
 - ▶ 8 000 échantillons par secondes pour le téléphone.
 - ▶ 44 100 pour un CD (44,1 kHz).
- ▶ Chaque échantillon est arrondi (quantification) à une valeur prédéfinie (perte).
 - ▶ Valeur prédéfinie est un nombre binaire, stocké sur un nombre fixé de bits.
 - ▶ Par ex : sur 8 bits, 256 valeurs possibles.
- ▶ Signal numérique : suite de représentations binaires.



Audio

- ▶ Par ex, 8 000 échantillons/sec, 256 valeurs (sur 1 octet) : 64 000 bit/sec.
- ▶ Signal numérique re-converti chez le receveur pour retrouver le signal analogique : perte.
 - ▶ Compromis taille du signal/qualité du signal.
- ▶ Exemples de taux d'échantillonnage :
 - ▶ CD : $44100 \cdot 16 \cdot 2$ (stéréo) = 1,411 Mbit/sec.
 - ▶ MP3 : 96, 128, 160... kbit/sec.
 - ▶ Téléphone sur Internet : > 5,3 kbit/sec.



Vidéo - Représentation numérique

- ▶ Vidéo : séquence d'images diffusées à un taux constant (par ex. 24 img/sec).
- ▶ Image numérique : tableau de pixels.
 - ▶ Un pixel : bits.
- ▶ Q : Comment encoder une vidéo ?



image i



image $i+1$

Vidéo - Encodage

- ▶ Utilisation de la redondance
 - ▶ Au sein d'une image (dans un ciel, deux pixels différent peu...).
 - ▶ Au lieu d'envoyer n pixels identiques, on envoie 2 valeurs : la couleur et le nombre de répétitions...
 - ▶ Entre deux images (peu de différences dans un même plan).
- ▶ Codage des différences pour limiter le nombre de bits nécessaires.

Encodage

- Compression : dire pixel identique si proche.



Encodage

- ▶ CBR (Constant Bit Rate) :
 - ▶ Taux d'échantillonnage fixe du début à la fin.
- ▶ VBR (Variable Bit Rate) :
 - ▶ Taux variable. Selon les besoins (plan dans le noir, images rapides, son tenu, changements de rythmes...)

Encodage vidéo

- ▶ Entrelacé (i) :
 - ▶ Doubler le nombre d'images par seconde perçues avec le même débit.
 - ▶ Une fois les lignes paires, une fois les lignes impaires.



- ▶ Progressif (p) :
 - ▶ Toutes les lignes en même temps.
- ▶ 720p vs 1080i par exemple.

Applications

- ▶ Trois grands types d'applications possibles :
 1. Streaming stocké (audio, vidéo).
 - ▶ Peut être joué avant d'être entièrement téléchargé.
 - ▶ Stocké sur un serveur (peut être envoyé plus rapidement que reçu : nécessite un buffer côté client) (ou P2P (Spotify)).
 - ▶ Exemple : Youtube, Netflix (VOD de films par abonnement aux USA, 30% du trafic USA), télévision en replay...
 - ▶ 50% du trafic en Am. du Nord en 2011 (30% en 2009), 33% en Europe (Chiffres Sandvine).
 2. Conversations (audio/vidéo sur IP)
 - ▶ Exemple : Skype, Google...
 3. Streaming live audio/vidéo :
 - ▶ Exemple socrast, twitch...

Cours 4 : Multimédia

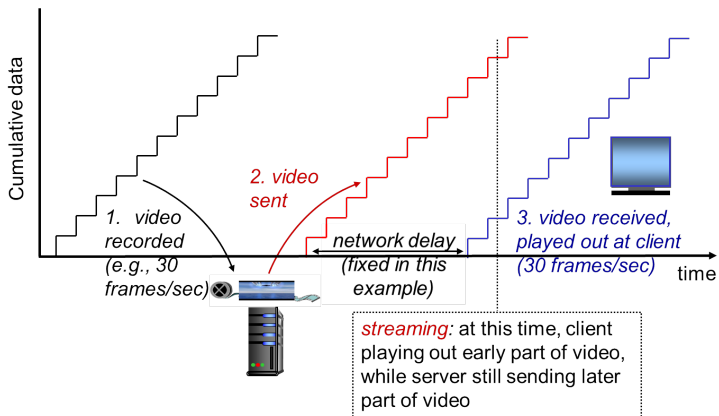
Applications réseau multimédias

Streaming de données enregistrées

Voix sur IP

Streaming temps réel

Streaming de vidéo enregistrée - Idéal

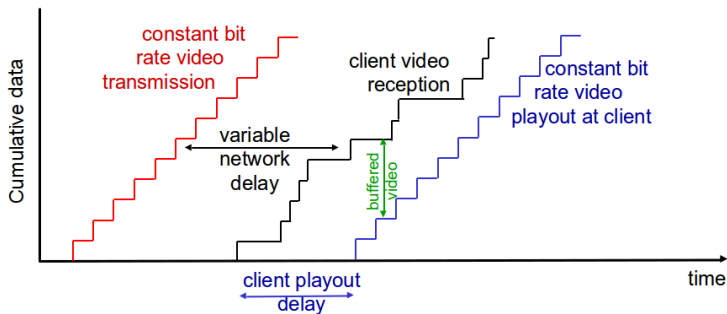


- ▶ Le client joue directement ce qu'il reçoit.

Streaming de vidéo enregistrée - Problèmes

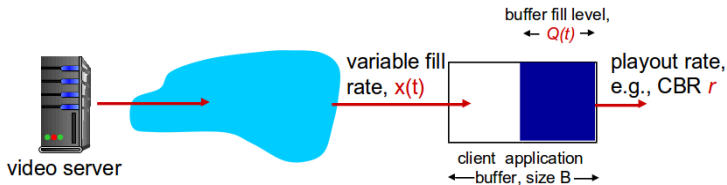
- ▶ Lecture continue :
 - ▶ Une fois que la lecture a commencée, doit lire comme sur la vidéo stockée.
 - ▶ Mais il peut y avoir des délais sur le réseau : besoin d'un buffer côté client.
- ▶ Demandes du client : pause, avance rapide, saut dans la vidéo, retour arrière...
- ▶ Paquets pouvant être perdus : retransmission.

Streaming de vidéo enregistrée - Revisité



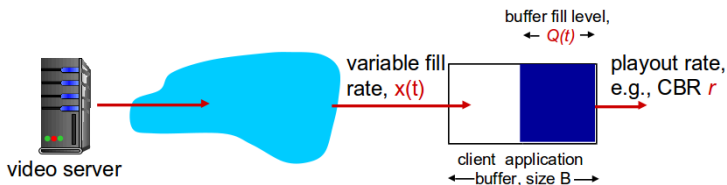
- ▶ Buffering côté client pour compenser.

Streaming de vidéo enregistrée - Buffering



- ▶ Commence à remplir un buffer jusqu'à ce que la lecture commence à un certain temps.
- ▶ Le buffer est rempli à un taux variable, mais la lecture est à taux constant.

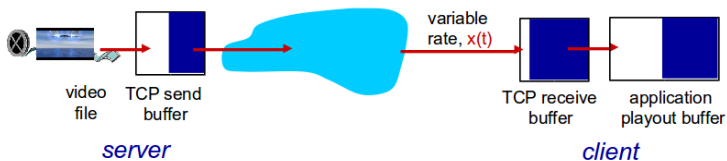
Streaming de vidéo enregistrée - Buffering



- ▶ Si $x < r$: buffer peut se vider entièrement et met la vidéo en pause jusqu'à remplissage du buffer.
- ▶ Si $x > r$: buffer ne se vide pas et absorbe les variabilités de x .
 - ▶ Compromis sur la taille du buffer : moins de problème si attend plus longtemps, mais début de lecture retardée.

Streaming de vidéo enregistrée

- ▶ Certains utilisent UDP, d'autres TCP.



- ▶ Si TCP, fichier multimédia obtenu via un GET HTTP (donc via TCP)
- ▶ Taux variable à cause des contrôles (de congestion) TCP...

Streaming de vidéo enregistrée - Adaptivité

- ▶ Streaming HTTP : tous les clients reçoivent la même vidéo.
 - ▶ Mais les clients ont des **bandes passantes différentes** !
- ▶ Dynamic Adaptive Streaming over HTTP (DASH).
- ▶ La même vidéo est encodée avec différentes qualités (youtube...) sur le serveur, en morceaux (chunks).
- ▶ Périodiquement, le client mesure sa bande passante et récupère le chunk adapté (grâce à un fichier manifest stocké sur le serveur donnant les URL).
 - ▶ C'est le client qui fait le travail : choisi quand, à quel encodage et où (peut préférer un serveur près de lui).

Streaming de vidéo enregistrée - Stockage

- ▶ Problème : comment proposer du contenu (millions de vidéos) à des centaines de milliers d'utilisateurs simultanés ?

Streaming de vidéo enregistrée - Stockage

- ▶ Problème : comment proposer du contenu (millions de vidéos) à des centaines de milliers d'utilisateurs simultanés ?
- ▶ Option 1 : un “mega serveur”.
 - ▶ Un seul point d'attaque (liaison serveur).
 - ▶ Point de congestion de réseau.
 - ▶ Très éloigné de clients potentiels.
 - ▶ Plusieurs copies d'une même vidéo envoyé sur une même liaison.

Streaming de vidéo enregistrée - Stockage

- ▶ Problème : comment proposer du contenu (millions de vidéos) à des centaines de milliers d'utilisateurs simultanés ?
- ▶ Option 1 : un "mega serveur".
 - ▶ Un seul point d'attaque (liaison serveur).
 - ▶ Point de congestion de réseau.
 - ▶ Très éloigné de clients potentiels.
 - ▶ Plusieurs copies d'une même vidéo envoyé sur une même liaison.
- ▶ Pas applicable à l'échelle !!!

Streaming de vidéo enregistrée - Stockage

- ▶ Problème : comment proposer du contenu (millions de vidéos) à des centaines de milliers d'utilisateurs simultanés ?

Streaming de vidéo enregistrée - Stockage

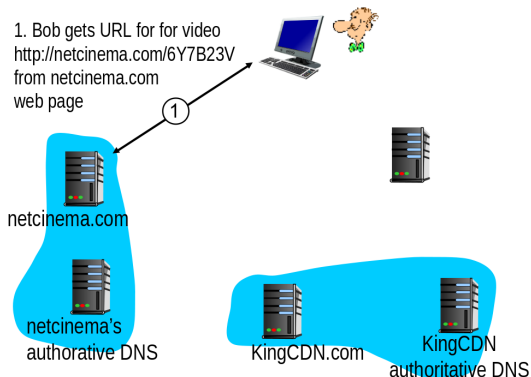
- ▶ Problème : comment proposer du contenu (millions de vidéos) à des centaines de milliers d'utilisateurs simultanés ?
- ▶ Option 2 : stockage de copies d'une même vidéo sur plusieurs sites géographiquement distribués (CDN Content Distribution Networks).
 - ▶ Peuvent être privés (Google...) ou tiers (Akamai...).

Streaming de vidéo enregistrée - CDN

- ▶ Bob (le client) demande une vidéo du type `http://netcinema.com/6Y7B23V`.
- ▶ Cette vidéo est stockée sur un CDN à l'adresse `http://KingCDN.com/NetC6y&B23V`.

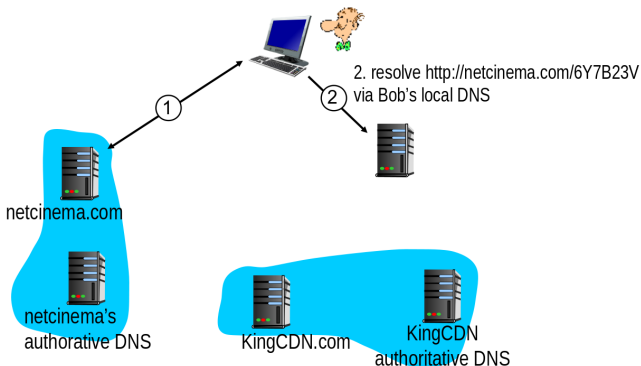
Streaming de vidéo enregistrée - CDN

- ▶ Bob (le client) demande une vidéo du type `http://netcinema.com/6Y7B23V`.
- ▶ Cette vidéo est stockée sur un CDN à l'adresse `http://KingCDN.com/NetC6y&B23V`.



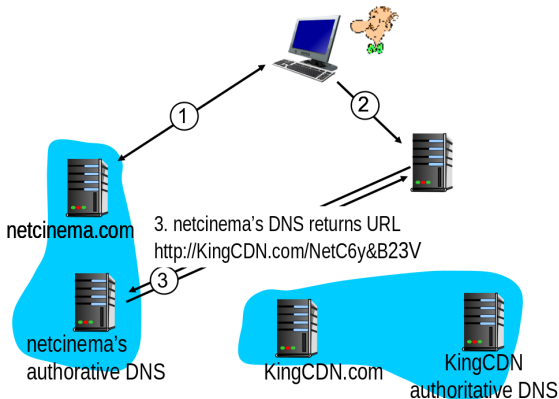
Streaming de vidéo enregistrée - CDN

- ▶ Bob (le client) demande une vidéo du type `http://netcinema.com/6Y7B23V`.
- ▶ Cette vidéo est stockée sur un CDN à l'adresse `http://KingCDN.com/NetC6y&B23V`.



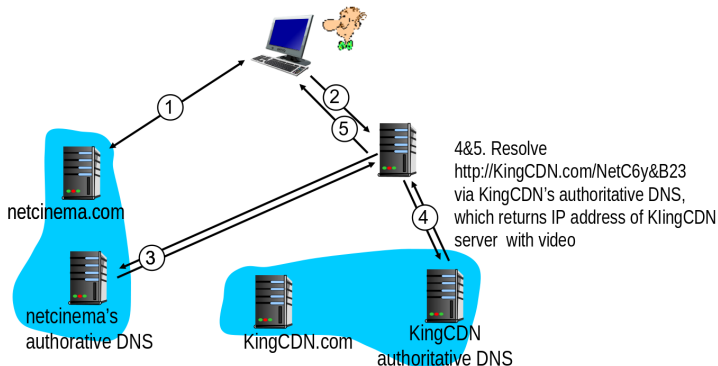
Streaming de vidéo enregistrée - CDN

- ▶ Bob (le client) demande une vidéo du type `http://netcinema.com/6Y7B23V`.
- ▶ Cette vidéo est stockée sur un CDN à l'adresse `http://KingCDN.com/NetC6y&B23V`.



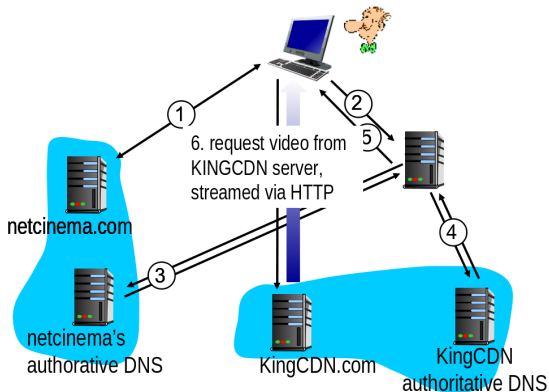
Streaming de vidéo enregistrée - CDN

- ▶ Bob (le client) demande une vidéo du type `http://netcinema.com/6Y7B23V`.
- ▶ Cette vidéo est stockée sur un CDN à l'adresse `http://KingCDN.com/NetC6y&B23V`.



Streaming de vidéo enregistrée - CDN

- ▶ Bob (le client) demande une vidéo du type `http://netcinema.com/6Y7B23V`.
- ▶ Cette vidéo est stockée sur un CDN à l'adresse `http://KingCDN.com/NetC6y&B23V`.



CDN - étude de cas Google

- ▶ Beaucoup de services “dans le cloud” (gmail, youtube, recherche, maps...).
- ▶ Grand réseau privé (2012).
 - ▶ 8 mega data centers (6 USA, 2 europe), ayant de l'ordre de 100 000 serveurs chacun. Pour le contenu dynamique (gmail, recherches...).
 - ▶ Trentaine de clusters (ordre de 100-500 serveurs chacuns), partout dans le monde près des ISP tier 1. Pour le contenu statique (youtube...).
 - ▶ Des centaines de clusters proches des FAI.

Étude de cas : Youtube

- ▶ Chiffres 2012 :
 - ▶ 3ème site mondial, 4% des pages vues.
 - ▶ 72h de vidéo uploadée chaque minute (plusieurs années par jour).
 - ▶ 4 milliards de vues par jour, 1 milliard unique par mois.
- ▶ Large utilisation du CDN de Google.
- ▶ Redirection par DNS du client vers le cluster le plus proche (en terme de RTT).
- ▶ Si le cluster n'a pas la vidéo, client renvoyé vers un autre cluster.
- ▶ Streaming HTTP.
- ▶ Conversions en différentes qualités sur les data center google (et pas chez le client).

Cours 4 : Multimédia

Applications réseau multimédias

Streaming de données enregistrées

Voix sur IP

Streaming temps réel

Voix sur IP (VoIP)

- ▶ La VoIP doit maintenir une conversation.
 - ▶ Délais importants préjudiciables.
 - ▶ < 150 ms : bien.
 - ▶ > 400 ms : pas bien.

Voix sur IP (VoIP)

- ▶ Signal (personne parlant dans son micro relié à l'ordinateur) constitué de :
 - ▶ Moments actifs (paroles).
 - ▶ Silences.
- ▶ Par exemple, génération de paquets uniquement pour les moments de paroles (à 8000 octets/s (ou 64 kBits/s)), regroupés en morceaux toutes les 20 ms (donc 160 octets).
- ▶ Ajout d'un en-tête à ces 160 octets.
- ▶ Encapsulé dans un paquet UDP (ou TCP) : ajout d'en-têtes supplémentaires.

Voix sur IP (VoIP)

- ▶ Dans le meilleur des mondes :
 - ▶ Envoi toutes les 20 ms.
 - ▶ Réception toutes les 20ms, dans l'ordre.
- ▶ Le destinataire peut lire directement ce qu'il reçoit.

Voix sur IP (VoIP)

- ▶ Dans le meilleur des mondes :
 - ▶ Envoi toutes les 20 ms.
 - ▶ Réception toutes les 20ms, dans l'ordre.
- ▶ Le destinataire peut lire directement ce qu'il reçoit.
- ▶ Mais :
 - ▶ Paquets perdus.
 - ▶ Différence de vitesses.

Voix sur IP (VoIP)

- ▶ Dans le meilleur des mondes :
 - ▶ Envoi toutes les 20 ms.
 - ▶ Réception toutes les 20ms, dans l'ordre.
- ▶ Le destinataire peut lire directement ce qu'il reçoit.
- ▶ Mais :
 - ▶ Paquets perdus.
 - ▶ Différence de vitesses.
- ▶ Difficulté de déterminer quel morceau lire, quand...

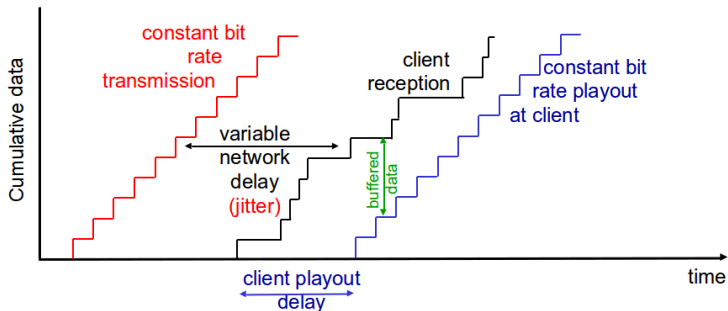
Gigue de paquets

- ▶ Temps de transmission d'un paquet influencé par délais d'attente au sein des routeurs rencontrés sur le parcours.
- ▶ Temps aléatoire, variable de paquet en paquet : gigue (jitter).
- ▶ Par exemple, deux paquets envoyés à 20ms d'écart.
- ▶ Le premier rencontre des routeurs sans files d'attentes.
- ▶ Le second est dans les embouteillages.
- ▶ Intervalle supérieur à 20ms à l'arrivée.
- ▶ Inverse possible.

Gigue de paquets

- ▶ Temps de transmission d'un paquet influencé par délais d'attente au sein des routeurs rencontrés sur le parcours.
- ▶ Temps aléatoire, variable de paquet en paquet : gigue (jitter).
- ▶ Par exemple, deux paquets envoyés à 20ms d'écart.
- ▶ Le premier rencontre des routeurs sans files d'attentes.
- ▶ Le second est dans les embouteillages.
- ▶ Intervalle supérieur à 20ms à l'arrivée.
- ▶ Inverse possible.
- ▶ Résolution à l'aide de numéros de séquences, marqueur et buffer de lecture.

Gigue : buffering



Gigue : buffering à retard fixe

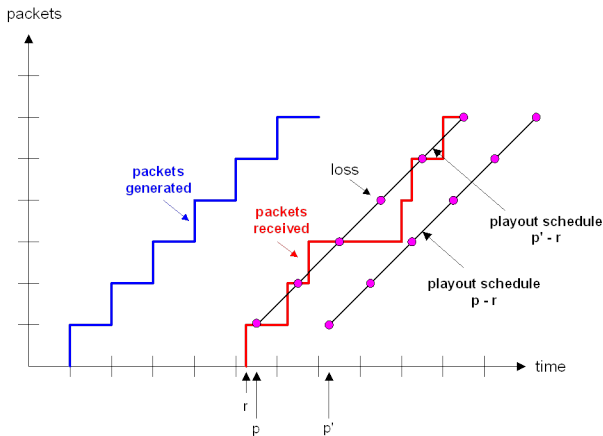
- ▶ Choix d'un délai fixe de p ms.
- ▶ Chaque morceau est tenté d'être lu p ms après sa génération.
- ▶ S'il n'est pas arrivé à cet instant : paquet perdu.

Gigue : buffering à retard fixe

- ▶ Choix d'un délai fixe de p ms.
- ▶ Chaque morceau est tenté d'être lu p ms après sa génération.
- ▶ S'il n'est pas arrivé à cet instant : paquet perdu.
- ▶ Compromis sur le choix de q :
 - ▶ Petit p : meilleure interaction.
 - ▶ Grand p : moins de pertes de morceaux.

Gigue : buffering à retard fixe

- ▶ Envoi de paquets toutes les 20ms durant la parole.
- ▶ 2 cas de figure :
 - ▶ Choix d'un p court : perte de paquets.
 - ▶ Choix d'un p' plus long : lecture toute les 20ms OK.



Gigue : buffering à retard variable

- ▶ Faire **varier le délai** plutôt que de choisir un temps fixe pour avoir un délai le plus petit possible !
- ▶ But : ajuster le délai de lecture en fonction du temps de transmission du réseau.
- ▶ Estimation du délai pour le paquet i en fonction du temps moyen pris par les paquets précédents et de sa variance...

Perte de paquets

- ▶ But : récupérer des paquets perdus selon un certain délai.
- ▶ Temps réel :
 - ▶ Retransmettre un paquet ayant manqué son créneau de lecture est inutile (trop tard).
 - ▶ Retransmettre un paquet rejeté au niveau d'un routeur est voué à l'échec.

Perte de paquets

- ▶ But : récupérer des paquets perdus selon un certain délai.
- ▶ Temps réel :
 - ▶ Retransmettre un paquet ayant manqué son créneau de lecture est inutile (trop tard).
 - ▶ Retransmettre un paquet rejeté au niveau d'un routeur est voué à l'échec.
- ▶ Systèmes d'anticipations :
 - ▶ Correction d'erreurs sans voie de retour (FEC).
 - ▶ Entrelacement (interleaving).

Correction d'erreurs sans voie de retour : algorithme simple

- ▶ But : envoyer suffisamment de bits pour retrouver un paquet perdu sans exiger sa retransmission.

Correction d'erreurs sans voie de retour : algorithme simple

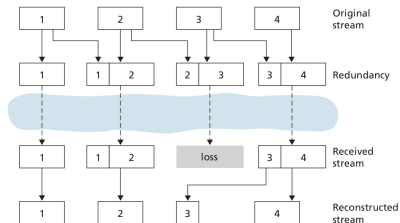
- ▶ But : envoyer suffisamment de bits pour retrouver un paquet perdu sans exiger sa retransmission.
- ▶ Algorithme simple :
 - ▶ Pour n morceaux, créer un nouveau morceau qui est le XOR des n morceaux.
 - ▶ Envoyer $n + 1$ morceaux : augmentation de la quantité de données par $1/n$.
 - ▶ Peut reconstruire les n morceaux originaux si au plus un morceau est perdu parmi les $n + 1$, sans délai.

Correction d'erreurs sans voie de retour : algorithme simple

- ▶ But : envoyer suffisamment de bits pour retrouver un paquet perdu sans exiger sa retransmission.
- ▶ Algorithme simple :
 - ▶ Pour n morceaux, créer un nouveau morceau qui est le XOR des n morceaux.
 - ▶ Envoyer $n + 1$ morceaux : augmentation de la quantité de données par $1/n$.
 - ▶ Peut reconstruire les n morceaux originaux si au plus un morceau est perdu parmi les $n + 1$, sans délai.
- ▶ Ne fonctionne pas si **plus d'une erreur**.
- ▶ Augmente beaucoup si n petit (33% pour 3).
- ▶ Destinataire doit **attendre** de recevoir les $n + 1$ avant de lire.

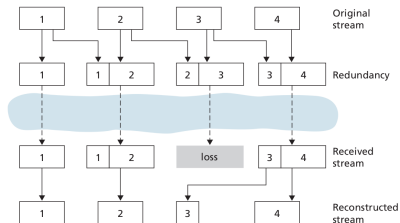
Correction d'erreurs sans voie de retour : algorithme piggyback

- ▶ Idée : envoyer en plus un flux de données à une **résolution inférieure** à celle de l'original.
 - ▶ Par exemple, flux de données au débit MIC 64kbit/s à un flux GSM 13kbit/s.
- ▶ Paquet i : paquet i original + **paquet $i - 1$** basse résolution.



Correction d'erreurs sans voie de retour : algorithme piggyback

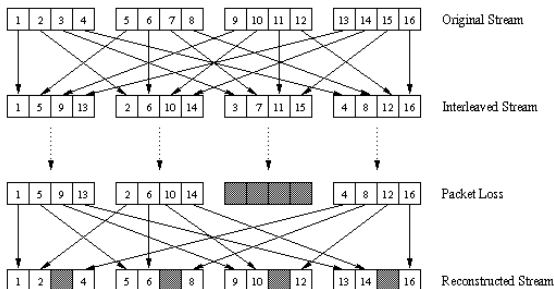
- ▶ Idée : envoyer en plus un flux de données à une **résolution inférieure** à celle de l'original.
 - ▶ Par exemple, flux de données au débit MIC 64kbit/s à un flux GSM 13kbit/s.
- ▶ Paquet i : paquet i original + **paquet $i - 1$** basse résolution.



- ▶ Si pas 2 pertes consécutives : OK
- ▶ Lecture dès le 2ème paquet.

Correction : entrelacement

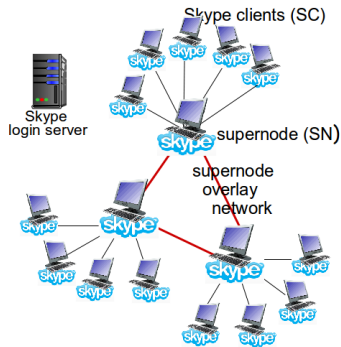
- ▶ Morceaux divisés en morceaux plus petits (par ex. 5ms).
- ▶ Paquet contenant des petits morceaux de différents morceaux originaux.



- ▶ Paquet perdu : contient la majorité d'un morceau original.
- ▶ Pas de données supplémentaires.
- ▶ Mais plus grande latence...

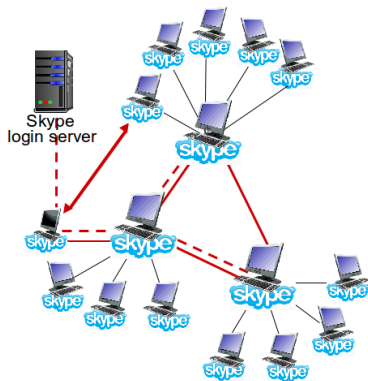
Architecture Skype avant 2012

- ▶ Application propriétaire :
rétro-ingénierie.
- ▶ Messages chiffrés.
- ▶ Composants P2P :
 - ▶ Clients : pairs connectés directement pour un appel VoIP.
 - ▶ Super-nœuds : pairs avec une fonction spéciale.
 - ▶ Réseau de super-nœuds.
 - ▶ Serveur de login.



Architecture Skype avant 2012

- ▶ Opération d'un client Skype :
 - ▶ Rejoindre le réseau en TCP via un SN (difficile de savoir comment. DHT possible).
 - ▶ Login via le serveur Skype.
 - ▶ Obtenir l'IP du correspondant via le réseau de SN.
 - ▶ Appel VoIP.
 - ▶ SN également utile en cas d'utilisation de routeurs et NAT.
- ▶ Depuis 2012 (et rachat par Microsoft) : SN remplacés par des serveurs.
 - ▶ De 800 utilisateurs / SN : peut gérer plusieurs milliers.
 - ▶ Meilleure vie privée, évite des failles de sécurité dues aux utilisateurs...
 - ▶ Après 2014 : abandon du P2P (car + de terminaux mobiles avec autonomie limitée...).



Skype

- ▶ Conférence à plusieurs :
 - ▶ Chacun envoie son stream audio à l'hôte de la conférence.
 - ▶ L'hôte combine les streams audio en un seul, et l'envoie aux $n - 1$ autres.
 - ▶ $2(n - 1)$ streams au lieu de $n(n - 1)$.

Skype

- ▶ Conférence à plusieurs :
 - ▶ Chacun envoie son stream audio à l'hôte de la conférence.
 - ▶ L'hôte combine les streams audio en un seul, et l'envoie aux $n - 1$ autres.
 - ▶ $2(n - 1)$ streams au lieu de $n(n - 1)$.
 - ▶ Pour la vidéo, combinaison impossible : envoi à un serveur (un seul envoi vs $n - 1$ envois : upload limité).

Skype

- ▶ Qualité audio souvent meilleure que par téléphone (16 000 samples/s vs 8 000 samples/s).
- ▶ Audio et vidéo par UDP (TCP si routeurs bloquent), contrôles par TCP.
- ▶ Corrections sans voie de retour (FEC) pour audio et vidéo.
- ▶ Qualité audio/vidéo et quantité FEC s'adapte selon les conditions du réseau.
- ▶ Vie privée : Bob peut sniffer l'IP d'Alice si conversation.
 - ▶ Utilisation de service de géo-localisation.
 - ▶ Même IP que BitTorrent.
 - ▶ ...

Cours 4 : Multimédia

Applications réseau multimédias

Streaming de données enregistrées

Voix sur IP

Streaming temps réel

Protocoles

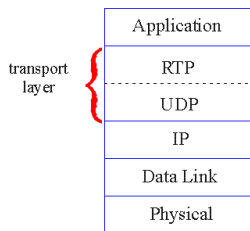
- ▶ Les applications vues précédemment ajoutent des champs d'en-tête aux morceaux de données audio/vidéo envoyées.
 - ▶ Numéro de séquence.
 - ▶ Référence de temps.
 - ▶ ...
- ▶ Différentes applications peuvent faire usage de telles informations.
- ▶ Définition d'une structure de paquet regroupant des champs utiles.

RTP

- ▶ Real-Time Protocol (RTP). RFC 3550.
- ▶ Regroupe des champs utiles.
- ▶ Pour le transport de données audio et vidéos, sous différents formats (même propriétaires).
- ▶ Utilisé par beaucoup d'applications.
 - ▶ Deux applications utilisant RTP peuvent théoriquement communiquer.

RTP sur UDP

- ▶ Donne une interface de transport qui étend UDP.
- ▶ Ajout d'en-têtes aux données de l'application, puis encapsulé dans une trame UDP.



RTP exemple

- ▶ Envoi de voix (audio) encodée en PCM à un débit de 64 kbit/s par RTP.
- ▶ Application collecte les données en morceaux, par exemple toutes les 20 ms : 160 octets par morceau.
- ▶ Le morceau audio + les en-têtes RTP forment un paquet RTP.
- ▶ Paquet RTP encapsulé dans un segment UDP.

- ▶ En-têtes RTP indiquent le type de l'encodage de chaque paquet (encodage peut donc être modifié en cours).
- ▶ Numéro de séquence, référence de temps...

RTP et qualité de service

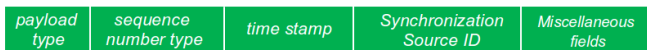
- ▶ RTP ne fournit aucune garantie de qualité de service.
 - ▶ Pas de garantie sur le temps pour délivrer le paquet.
 - ▶ Ni s'ils arriveront, ni dans l'ordre...
- ▶ RTP vu que par les hôtes, les routeurs ne font pas la distinction.

En-têtes RTP

<i>payload type</i>	<i>sequence number type</i>	<i>time stamp</i>	<i>Synchronization Source ID</i>	<i>Miscellaneous fields</i>
-------------------------	---------------------------------	-------------------	--------------------------------------	---------------------------------

- ▶ Payload (7bits) : type des données.
 - ▶ 0 : PCM, 64 kbit/s.
 - ▶ 3 : GSM, 13 kbit/s.
 - ▶ 31 : H.261 (vidéo).
 - ▶ 33 : MPEG2 (vidéo).
 - ▶ ...
- ▶ sequence # (16bits) : Numéro de séquence.
 - ▶ Incrémenté à chaque envoi de paquet.
 - ▶ Permet la détection de perte de paquets, et l'ordre.

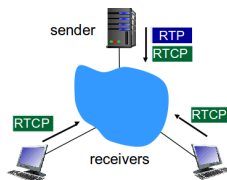
En-têtes RTP



- ▶ Timestamp (32 bits) : référence de temps.
 - ▶ Donne l'heure d'échantillonnage du premier octet du paquet.
 - ▶ Utilisé pour éliminer la gigue.
- ▶ SSRC (32 bits) : Identifiant de la source de synchro.
 - ▶ Identifie la source du stream.

RTCP

- ▶ RTP Control Protocol.
- ▶ Utilisé en conjonction avec RTP pour une session à plusieurs (multicast).
- ▶ Chaque participant à une session RTP envoie périodiquement des paquets de contrôle aux autres participants.
- ▶ Ces paquets contiennent des rapports statistique.
 - ▶ Nombre de paquets envoyés / perdus, la gigue...
- ▶ Utilisé pour contrôler la performance (rien n'oblige l'application à utiliser ces infos).



SIP

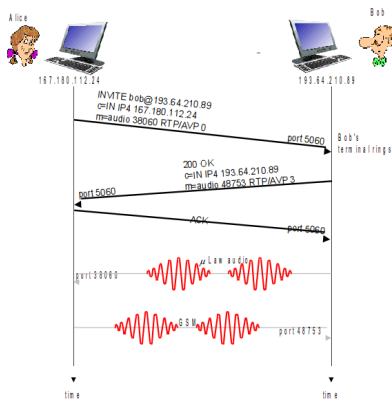
- ▶ Session Initiation Protocol (SIP) RFC 3261.
- ▶ Vision à long terme :
 - ▶ Les communications audio/vidéo seront toutes par Internet.
 - ▶ Entre un ordinateur, un téléphone connecté...
- ▶ Personnes identifiées par nom/adresse mail plutôt que par leur numéros.
- ▶ Destinataire joignable peut importe où il est, quel terminal il utilise...

SIP - services offerts

- ▶ Mécanismes nécessaires à l'établissement d'un appel entre deux interlocuteurs du réseau IP.
 - ▶ Destinataire avisé qu'il est appelé.
 - ▶ Accord sur l'encodage.
 - ▶ Terminaison de l'appel.
- ▶ Détermination de l'adresse IP (dynamique) du destinataire.
- ▶ Gestion des appels.
 - ▶ Changement d'encodage.
 - ▶ Ajout d'autres personnes.
 - ▶ Transferts, mise en attentes...

SIP - Exemple

- ▶ On suppose que Alice connaît l'IP de Bob.
- ▶ Le SIP d'Alice invite Bob. Indique son IP, son port, son encodage voulu.
- ▶ Bob répond 200 OK, envoie son IP, son port, son encodage voulu (autre message suivi d'un autre INVITE si encodage KO par ex., ou s'il ne veut pas répondre.).
- ▶ Messages pouvant être envoyés par TCP ou UDP (RTP/UDP).



Conclusion

- ▶ Internet est un service “best-effort” : aucune garantie.
- ▶ Solutions vue précédemment niveau application (buffering, qualité adaptée...).
- ▶ Multimédia en réseau une des applications les plus utilisées.
- ▶ Moins de temps devant la radio et la télévision : utilisation d’Internet pour des données audio et vidéo, en direct et enregistré.
- ▶ Devrait continuer à se développer (sans fil).
- ▶ Internet également pour les appels téléphoniques.
- ▶ VoIP moins chère, plus de services (conférence, vidéo...).

Pseudos

`http://www.quizzoodle.com/session/
9ddd27e5a3e040e3a31a3281ab572aeb`