



Accès aux vidéos en ligne : comment repousser les limites des tuyaux ?

vincent.roca@inria.fr

25 avril 2012

- Copyright © Inria 2012, tous droits réservés

○ contributeur : Vincent Roca

- distribué sous licence Creative Commons

**Paternité - Pas d'Utilisation Commerciale 2.0 France
(CC BY-NC 2.0)**

<http://creativecommons.org/licenses/by-nc/2.0/fr/deed.fr>

Plan

1. Introduction et contexte
2. Pourquoi et comment coder une vidéo numérique ?
3. Comment diffuser et accéder à une vidéo ?
4. Un exemple : le standard ISDB-Tmm

● 1^{ère} partie

Introduction et contexte

Un peu de vocabulaire

● VOD

- vidéo à la demande (« Video On Demand »)
 - l'utilisateur choisit son contenu, au lieu de « subir » les programmes choisis pour lui par les chaînes TV

● SVOD

- VOD avec abonnement (« Subscription VOD »)

● mode « streaming »

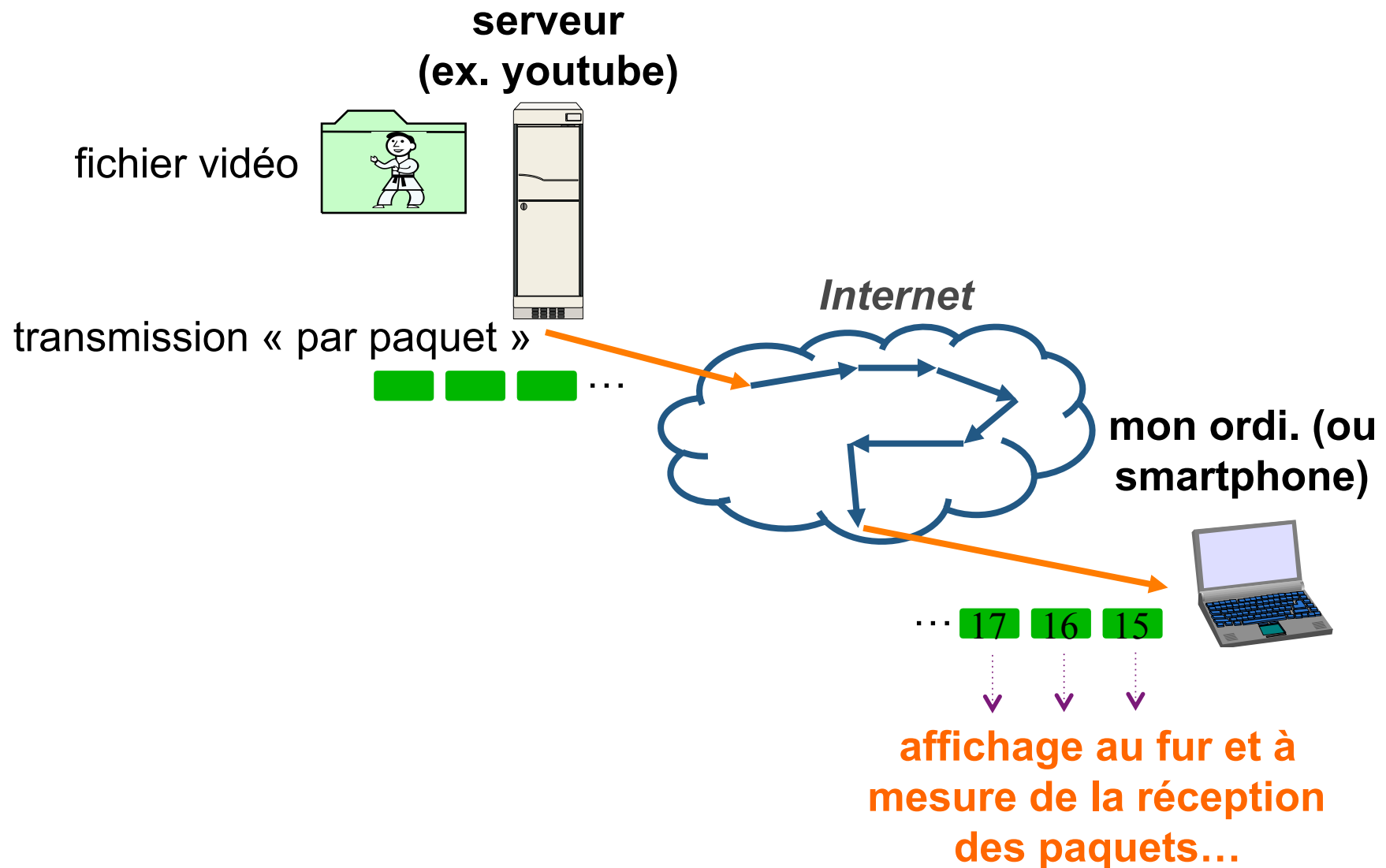
- lecture en continu de la vidéo/musique/...

● mode « téléchargement » (download)

- on télécharge la totalité, puis on lit la vidéo/musique/...

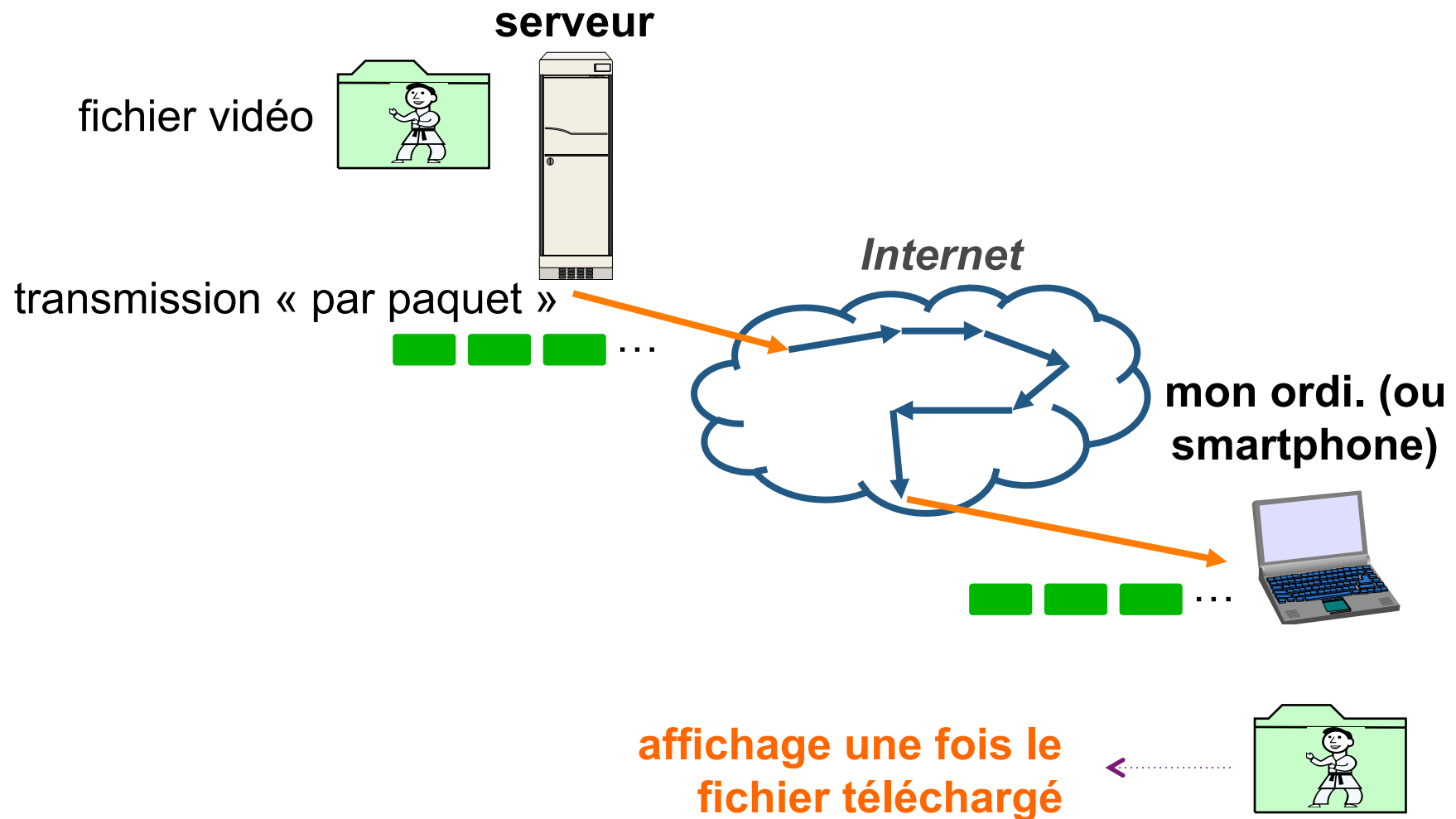
Un peu de vocabulaire... (suite)

- transmission en mode « streaming » ?



Un peu de vocabulaire... (suite)

- ou transmission en mode « téléchargement » ?



Un peu de vocabulaire... (suite)

- comparaison

mode streaming	téléchargement
on visionne rapidement	on attend la fin de réception du (gros) fichier
utilisable même si l'on n'a que peu d'espace de stockage libre	nécessite un espace de stockage suffisant
on peut zapper	zapper nécessite d'avoir téléchargé les deux fichiers... Bof !
la qualité dépend du réseau, qui varie dynamiquement	qualité optimale
visionner plusieurs fois nécessite plusieurs transmissions	on visionne autant de fois que voulu sans surcoût

Quelques statistiques pour fixer les idées...



- YouTube, c'est :

- 8 ans de contenu mis en ligne chaque jour
 - soit 48h par minute
- 3 milliards de vidéos regardées par **jour**
 - dont > 400 millions pour YouTube Mobile par jour
 - > 700 milliards de vidéos regardés sur l'année 2010
- 800 millions d'utilisateurs uniques par mois
- en France :
 - 39.6% de part de marché
 - 214 min/mois de visionnage en moyenne

- source : http://www.youtube.com/t/press_statistics

Quelques statistiques... (suite)



- DailyMotion, c'est :
 - 1,2 milliards de vidéos regardées par **mois**
 - 114 millions de visiteurs uniques par mois
 - en France :
 - 2,7% de part de marché
 - 82 min/mois de visionnage en moyenne

- source : <http://www.dailymotion.com/fr/about>

Quelques statistiques... (suite)

NETFLIX

leader US de la vidéo à la demande par abonnement

NETFLIX

Member Sign In

Watch unlimited movies & TV episodes over the Internet!
For one low monthly price.

- ✓ Watch right on your TV or computer
- ✓ Cancel online anytime



Sorry, Netflix is not available in your country... yet
Enter your name and email address below and we'll email you when Netflix is available.

First Name

Last Name

Email

Confirm Email

[Let me know](#)

Secure Server
We will not sell or rent your email address.
See our [Privacy Policy](#).

Press play and watch in seconds via the Internet.
Watch what you want, when you want. Unlimited!

Watch online on your computer



Watch online on your TV;
connect your laptop directly to your TV



Watch online on your TV;
connect your game consoles to your Netflix account



[About Us](#) [Affiliates](#) [Blog](#) [Contact Us](#) [Developers](#) [Investor Relations](#) [Jobs](#) [Media Center](#)

I have a US mailing address

Use of the Netflix service and this Web site constitutes acceptance of our Terms of Use and Privacy Policy.
© 1997-2012 Netflix, Inc. All rights reserved. U.S. Patent Nos. 6,584,450; 7,024,381; 7,631,323; 7,403,910; and 7,617,127. (eu-west-1 581b4011 FR) []

Quelques statistiques... (suite)

- évolution de l'offre au cours du temps
 - de l'envoi de cassettes VHS/DVD par **courrier**...



- ... au **streaming sur Internet**
 - pour 7.99\$/mois, accès illimité aux films/séries en streaming
 - recommandation : 1.5 Mb/s min, 3 Mb/s en qualité DVD
 - facilement accessible avec un accès ADSL

Quelques statistiques... (suite)

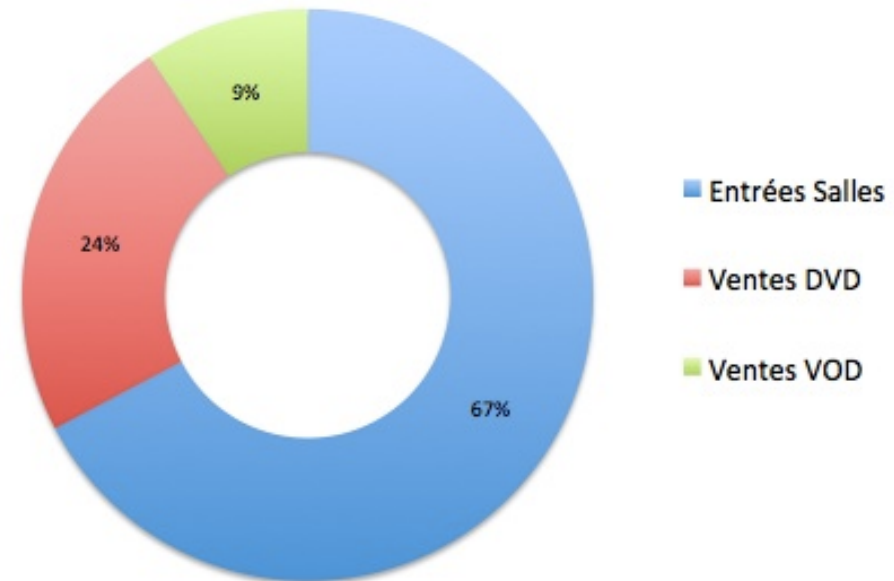
- Netflix, aux US, c'est :
 - 22 millions d'abonnements avec streaming
 - 3 millions d'abonnements DVD seuls



- 20% du trafic Internet aux heures de pointe (2010)
- Cela va augmenter inévitablement...

Quelques statistiques... (suite)

- « La VOD redonne le sourire à l'industrie du cinéma »
 - ZDNet, 13 février 2012
 - le poids de la VOD dans le chiffre d'affaire d'un film augmente...
 - exemple « Bridesmaids »:
 - 9% (40 Millions \$) du CA réalisé sur l'offre VOD aux US



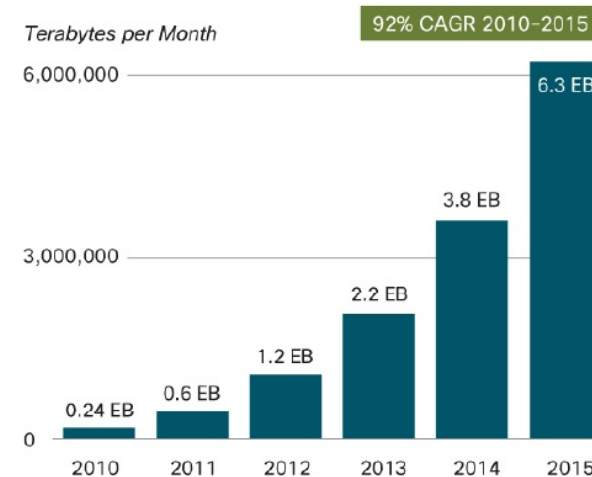
Quelques statistiques... (suite)

- Et le réseau là dedans ?

« Le trafic de données mobile est aujourd'hui **trois fois plus important** que tout le trafic Internet en 2000. Il devrait bondir de **26 fois** entre 2010 et 2015, pour atteindre 6,3 exaoctets (10^{18}) par mois. Rien qu'en 2010, il a triplé.

La vidéo sur mobile devrait être le principal consommateur de bande passante : il représente déjà 50% de tous le trafic data mobile observé... »

Figure 1. Cisco Forecasts 6.3 Exabytes per Month of Mobile Data Traffic by 2015



Source: Cisco VNI Mobile, 2011

○ Les opérateurs misent sur la disponibilité de nouveaux réseaux pour mobiles (amélioration avec 3G+, puis 4G/LTE)

- source : ZDNet, 1^{er} février 2011

En résumé...

- on a vu...

- différences entre les « modes streaming » et « téléchargement » de diffusion de contenus

- importance croissante de :

- YouTube offre VOD

- Netflix offre SVOD

- « la VOD, une chance pour l'industrie du cinéma »

- impacts sur le réseau

- Netflix : 20% du trafic Internet en heures de pointe !

- préoccupant pour « l'Internet mobile »

- course vers les réseaux 3G+ et 4G

● 2^{ème} partie

Aspects techniques

- pourquoi et comment coder une vidéo numérique ?

Pourquoi compresser une vidéo ?

- vidéo

- 25 ou 30 images/s pour une bonne illusion de mouvement

- une vidéo non compressée (les images sont indépendantes les unes des autres) a une **taille prohibitive**

- exemple :

- 1H de vidéo, format 1024x768, 25 image/s, 32 bits/pixel

- fichier de **283 GO**, streaming à **629 Mb/s**

- NB: ADSL-2 $\Rightarrow$ 18 Mb/s max

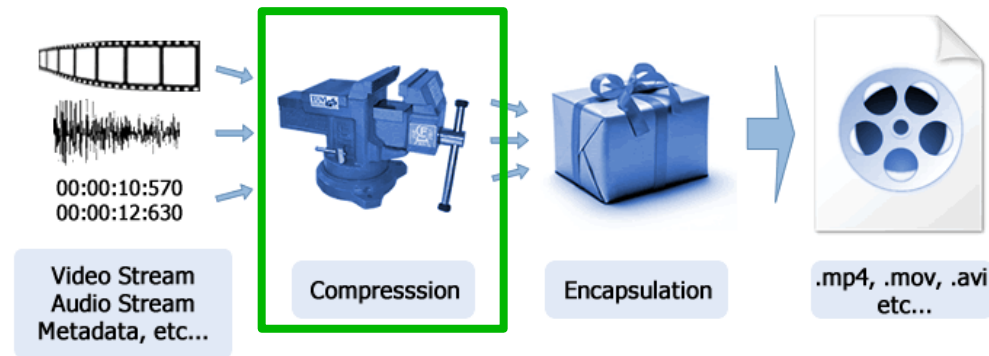
- coûts de stockage prohibitifs



- coûts de transmission prohibitifs



Comment **compresser** une vidéo ?



- la compression vidéo est « à perte »
 - il y a dégradation de qualité par rapport à l'original
 - challenge : limiter cette dégradation !
- technique 1 (compression spatiale) :
 - on triche en retirant d'une image les informations que l'œil distingue mal
 - on est plus sensible à la luminance qu'à la chrominance
 - permet une réduction de taille significative...
 - ... sans trop dégrader la qualité perçue

Comment compresser une vidéo... (suite)

○ technique 2 (compression temporelle) :

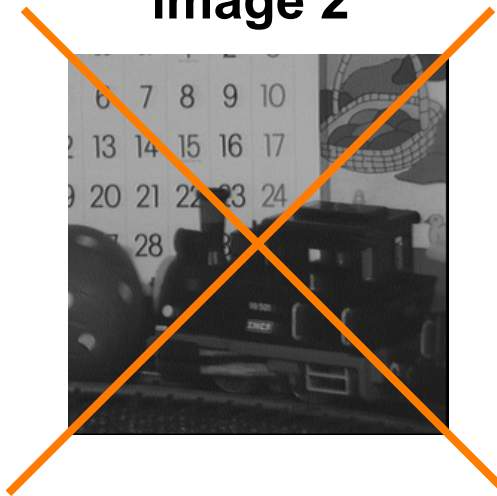
- On ne transmet pas deux fois les mêmes informations
- deux images successives étant voisines, on transmet la 1^{ère}, puis la différence 2^{ème} – 1^{ère} et ainsi de suite...

image 1
(référence)



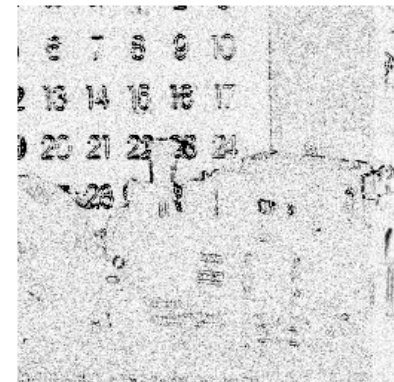
transmise

image 2



recalculée par
le récepteur

différence



transmise

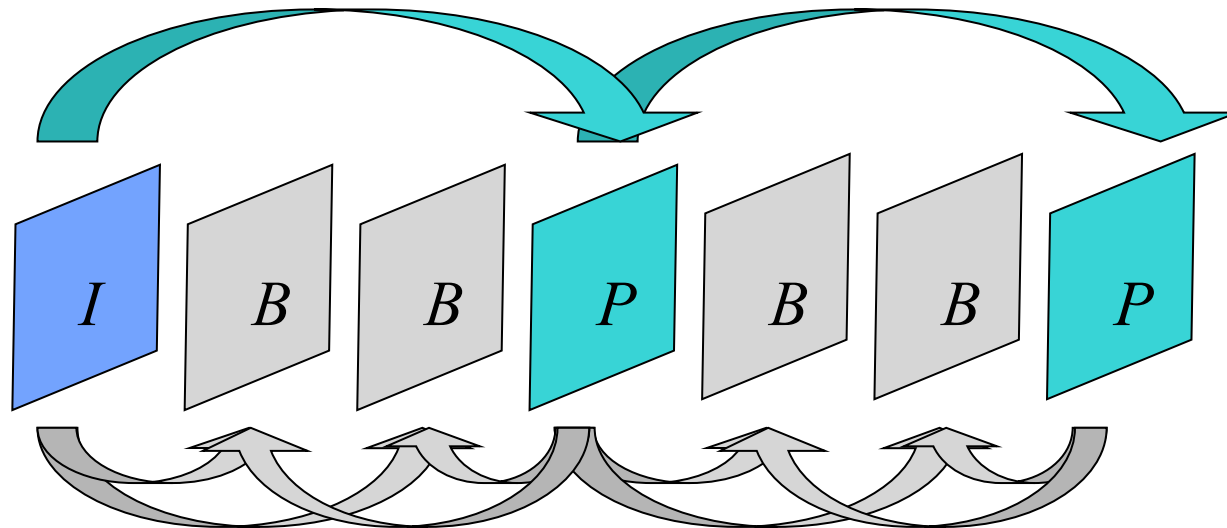
Comment compresser une vidéo... (suite)

- on utilise les 2 techniques conjointement (MPEG)

- chaque image est compressée

- on code les différences entre images en respectant une hiérarchie **I, P, B**

- **I** intra (image complète)
- **P** prédite: différence avec la trame I ou P précédente
- **B** prédite dans les deux sens: différence avec les trames I et P les plus proches



Comment compresser une vidéo... (suite)

● de multiples évolutions...

○ **Overs une meilleure compression à qualité égale**

- on réduit le « poids » de la vidéo (taille en octets)

○ **Overs de plus hautes définitions à « poids » égale**

- on améliore la qualité perçue avec la HD (1920x1080)
- autour de 5 Mb/s

○ **Overs la 3D**

De la théorie à la pratique

- on ne **peut** pas toujours compresser

- car cela exige une puissance de calcul importante, pas toujours compatible avec un fonctionnement en temps-réel
- exemple : format DV/HDV des caméscopes numériques
 - chaque image est indépendante

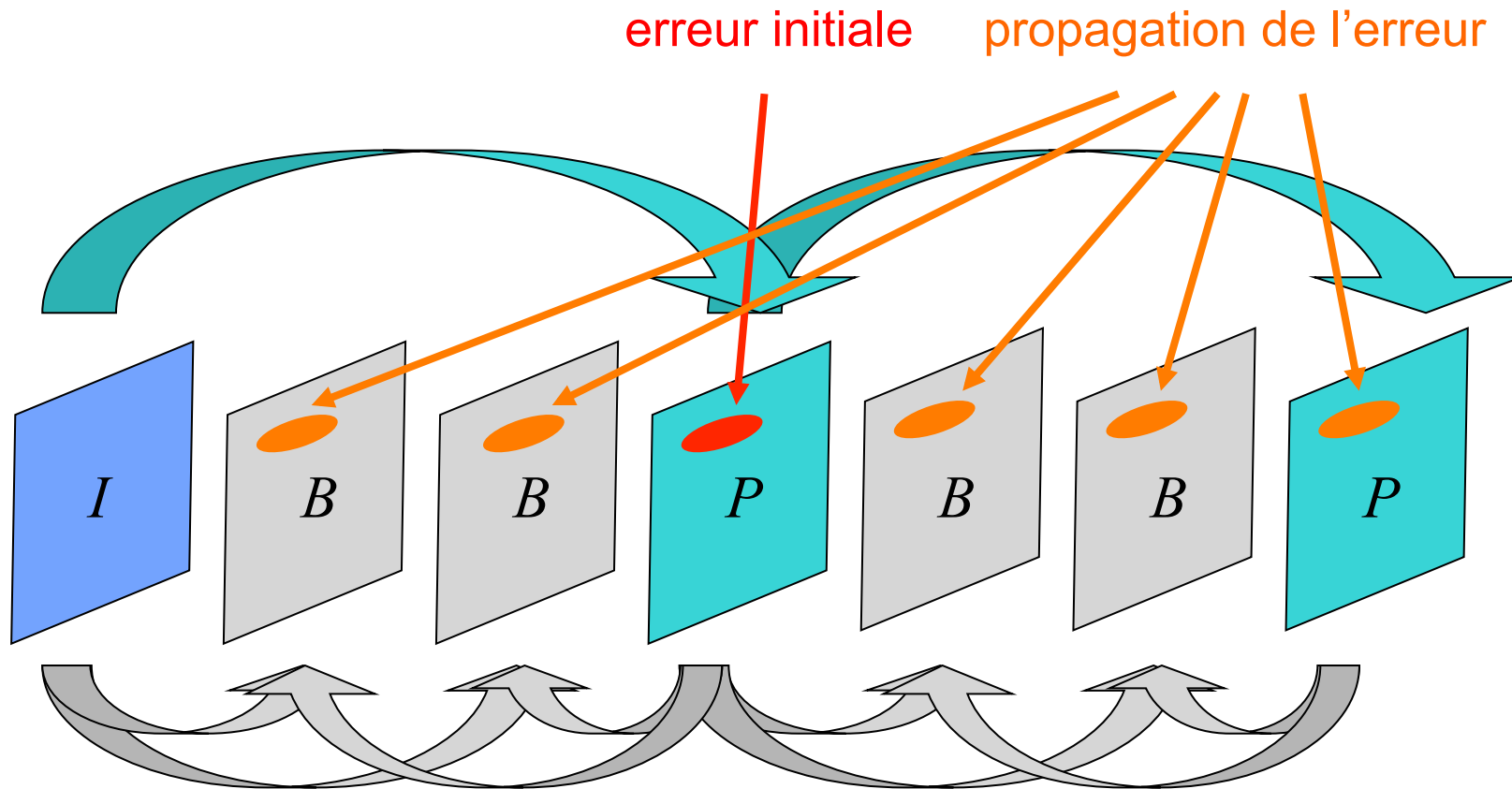
- on ne **veut** pas toujours compresser

- pour avoir une qualité optimale et pouvoir éditer facilement la vidéo lors du montage
- exemple : format MJPEG2000 du cinéma HD
 - la vidéo est une succession d'images indépendantes au format JPEG2000 (avec ou sans pertes)

De la théorie à la pratique... (suite)

- compresser n'est pas toujours bénéfique

- avec une compression MPEG, les erreurs sont propagées au sein du GOP (group of pictures)



De la théorie à la pratique... (suite)

Pour limiter les problèmes on va utiliser :

- **codage robuste** aux erreurs

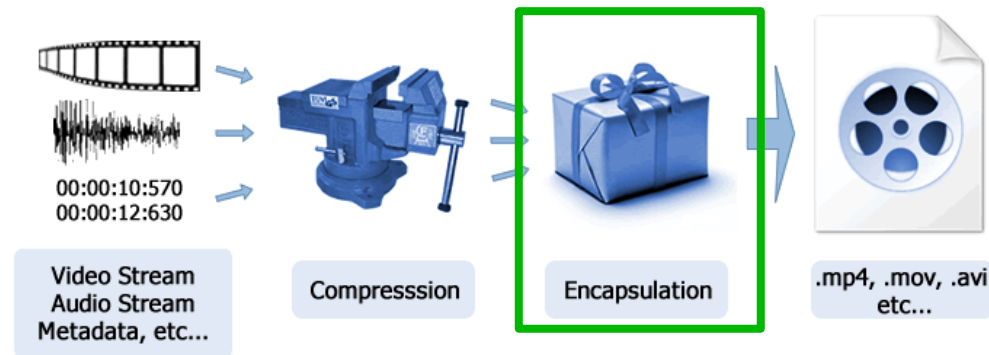
- **exemple : on ajoute des points de synchronisation dans le flux**

- techniques de **masquage d'erreurs**

- **le décodeur estime l'information manquante**

- en réutilisant des informations d'images précédentes
 - au sein d'une image par estimation de la zone manquante

Comment compresser une vidéo... (suite)



- à la compression s'ajoute le type de « conteneur »

- rassemble en un seul fichier le(s) **flux vidéo(s)**, audio(s), sous-titrage(s), méta-données, etc.

- exemple :

- .mpg (MPEG1/2/4, MPEG), .mov (QuickTime, Apple), .flv (Flash, Adobe), .webm (webm, Google), .avi (Microsoft), .rm (RealMedia, RealNetworks)

- source de confusion, complexité et incompatibilités !

En résumé...

- on a vu...

- nécessité de compresser

- pour stocker

- pour transmettre

- compression vidéo = ne conserver que l'information la plus importante visuellement, et ne pas transmettre deux fois des informations proches

- la compression « intra » est pratique, facilite l'édition et rend le flux robuste... au prix d'une compression moindre

- ne pas confondre standard de compression vidéo et type conteneur

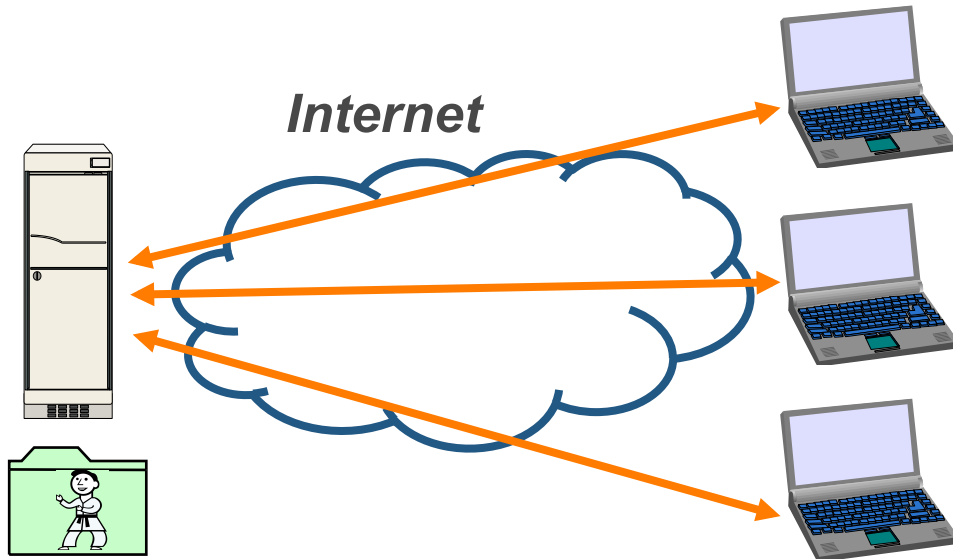
● 3^{ème} partie

Aspects techniques

- comment diffuser et accéder à une vidéo ?
 1. mode de transmission
 2. bufferisation
 3. pertes
 4. gros plan sur les codes LDPC-Staircase
 5. gros plan sur FECFRAME

1- Point à point ou multipoints ?

- comment sont diffusés les contenus ?

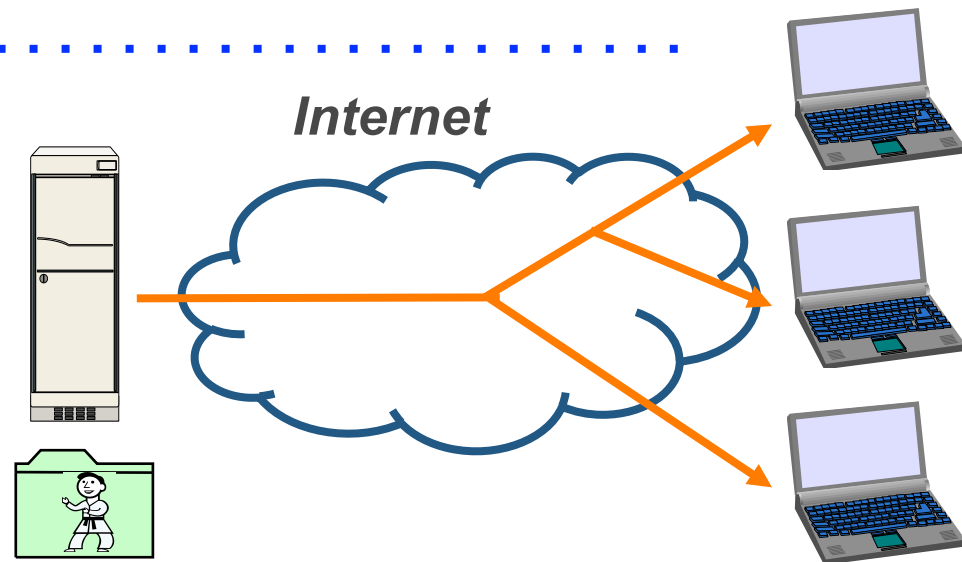


Point à point : $n \times (1 \Leftrightarrow 1)$

autant de connexions et de transmissions qu'il y a de clients

Multipoints : $1 \Rightarrow n$

un paquet est **transmis une fois** puis répliqué et acheminé par le réseau comme il se doit.



Point à point ou multipoints... (suite)

- le choix dépend de l'usage

- chaque client choisit à tout moment dans un catalogue important
⇒ **point à point**
- diffusion simultanée du même contenu à un grand nombre
⇒ **multipoints**

- **seule l'approche multipoints passe à l'échelle**

- c'est à dire supporte un nombre de clients très élevé

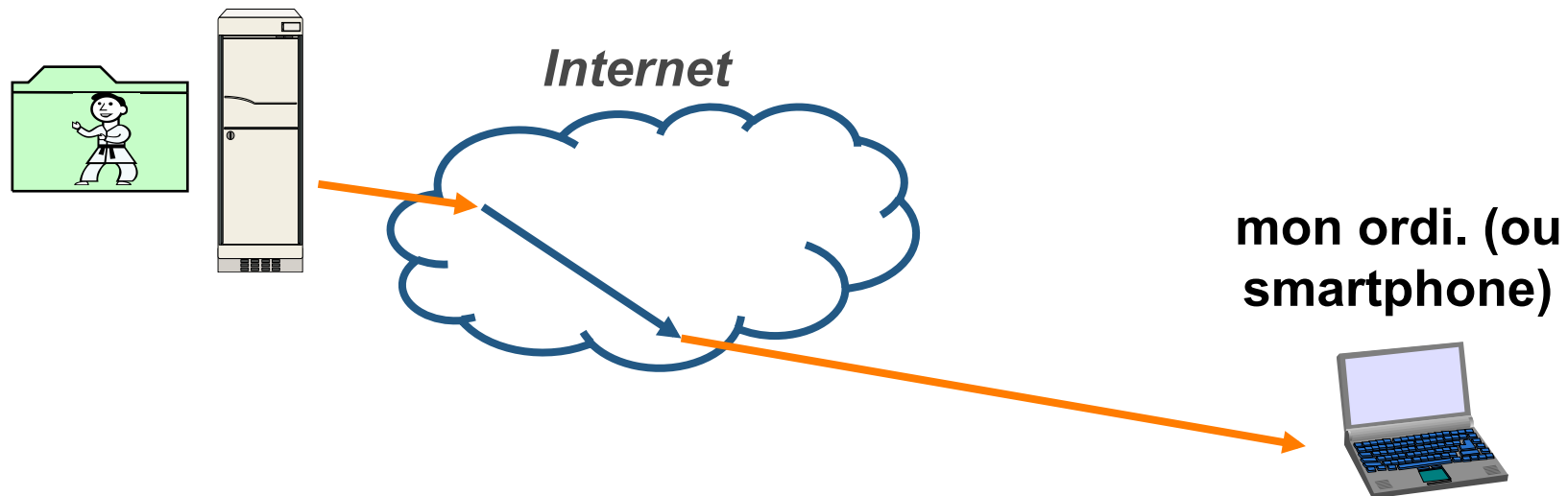
- **Questions :**

- **qu'utilise t-on pour le service VOD / YouTube ?**

- **qu'utilise t-on pour le service TV sur ADSL ?**

2- Bufferiser est nécessaire au streaming !

- le problème...



paquet reçus contenant les image successives...

... 17 16 15 14 plus rien... 13 12 11 10 ...
(blocage temporaire)

affichage au fur et à mesure de la réception des paquets...

↓ ↓ ↓ ↓ ok ok ok ok image figée 😞 ok ok ok ok

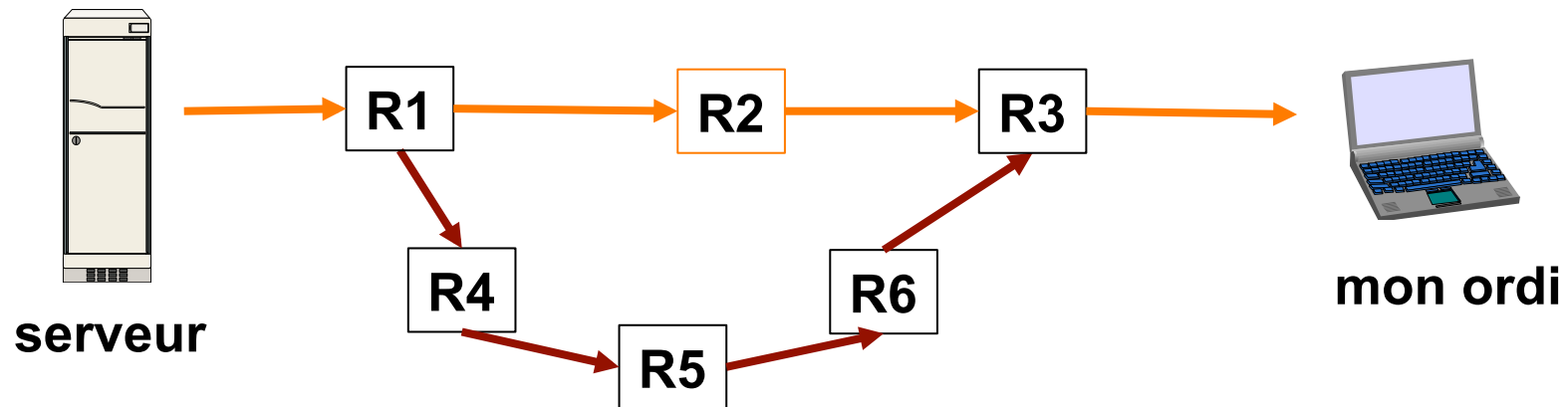
Bufferiser est nécessaire... (suite)

- question : pourquoi les délais d'acheminement sur Internet ne sont ils pas stables ?

Bufferiser est nécessaire... (suite)

- plusieurs raisons :

- sur un câble unique donné, le délai varie peu... sauf si un grand nombre de stations veulent y **accéder simultanément** !
- ... mais sur Internet, le délai dépend de l'**état d'encombrement** des routeurs traversés qui varie dynamiquement et fortement
- ... mais aussi du **chemin suivi** qui peut changer en cours de route



Bufferiser est nécessaire... (suite)

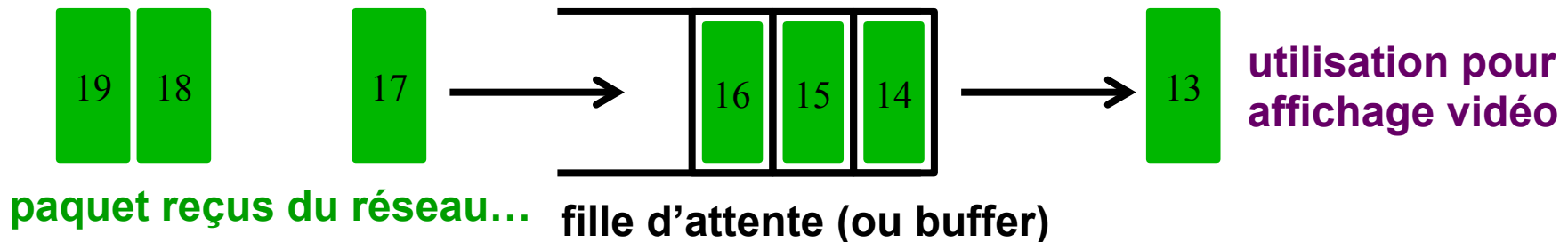
- **plusieurs raisons (suite) :**

- ... enfin, si la solution de streaming repose sur TCP (souvent le cas), les **retransmissions** TCP en cas de perte posent problème...

- cf plus loin le « recouvrement sur erreurs TCP »

Bufferiser est nécessaire... (suite)

- bufferiser est nécessaire pour éviter la famine ☹
- en pratique...
 - le client impose un délai avant de démarrer l'affichage
 - le client stocke des données d'avance pendant ce temps
 - c'est la « **bufferisation** »
 - le buffer est ensuite **consommé**
 - pour l'affichage vidéo
 - et **renouvelé** en permanence
 - avec l'arrivée de nouveaux paquets



Bufferiser est nécessaire... (suite)

- le buffer est visualisé dans la barre de lecture



○ Si la « zone grisée » disparaît, l'image se fige (il n'y a plus de données disponibles à afficher !)

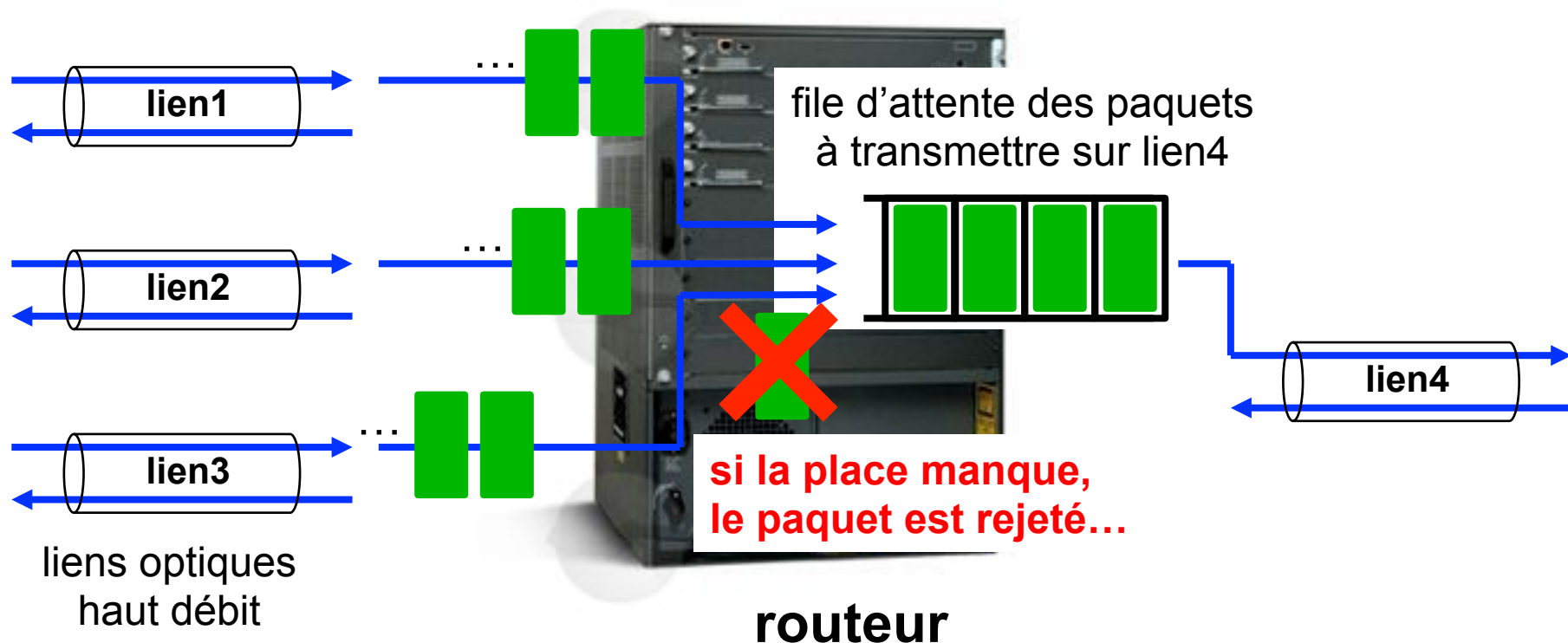
3- Quid des pertes ?

- question : peut-il y avoir des pertes sur Internet et pourquoi ?

« To be lost or not to be lost »

Quid des pertes... (suite)

- malheureusement oui, essentiellement dues à la présence d'un **routeur congestionné**
 - trop de trafic entrant pour le même lien de sortie
 - la file d'attente déborde... et des paquets perdus



Quid des pertes... (suite)

- on fait quoi ?

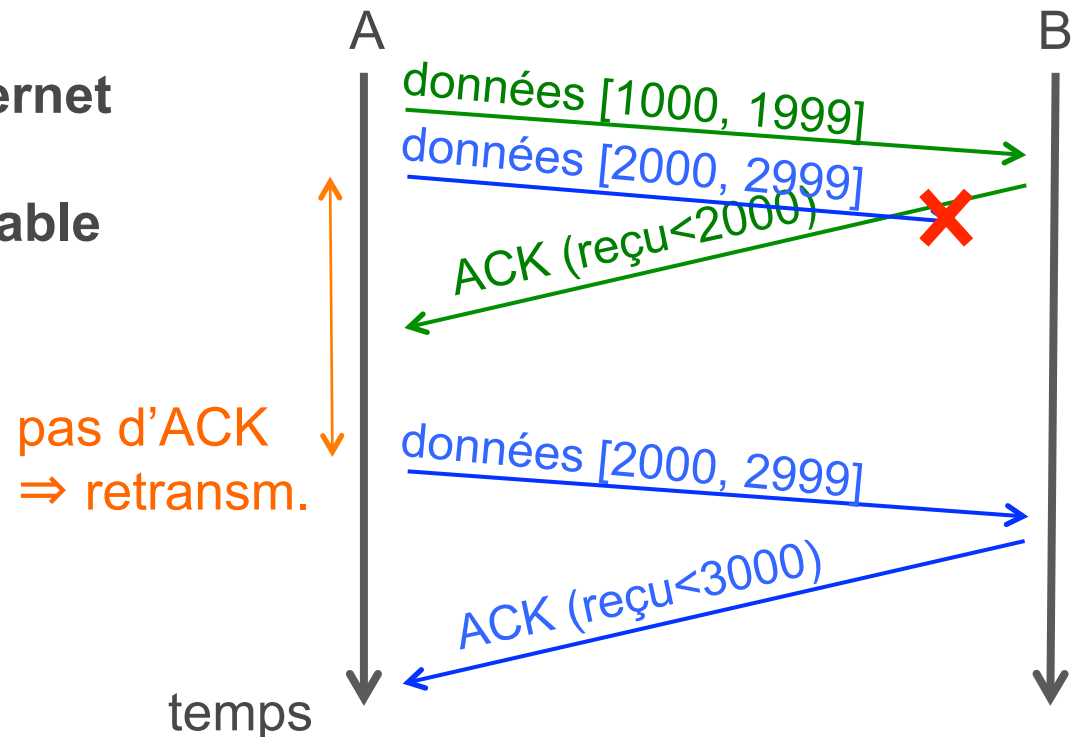
- cas 1 (simple) : **transmission point à point**

- le client informe le serveur et demande une retransmission

- le protocole TCP fait cela très bien, tout seul...

- **résolu** 😊

- TCP, clef de voute de l'Internet (avec IPv4/IPv6)
- service de transmission fiable et ordonné



Quid des pertes... (suite)

● cas 2 : transmission multipoints

○ Il faut **trouver autre chose** si le nombre de clients est $\gg 1$

- imaginez un serveur qui reçoit 1 million de requêtes de retransmission simultanément !

● technique : on ajoute de la **redondance** au flux

○ la redondance permet de récupérer une ou plusieurs pertes

○ notion de « **code correcteur d'effacement** » (FEC)

○ le même paquet de redondance permet de corriger une perte différente chez des récepteurs différents 😊

● exemple :



○ On reconstruit p13 par : $p13 = \text{somme}(p12, p13) - p12$

Contre les pertes, des codes *FEC*

- en pratique...

- c'est une « somme » au sens de l'opérateur binaire **XOR**

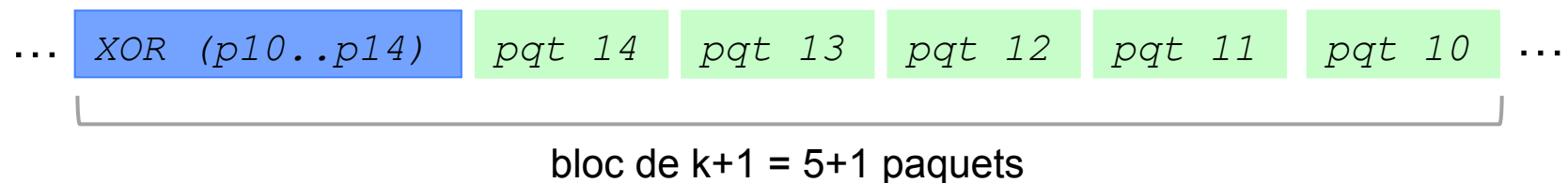
$$0 \oplus 0 = 0$$

$$1 \oplus 0 = 1$$

$$0 \oplus 1 = 1$$

$$1 \oplus 1 = 0$$

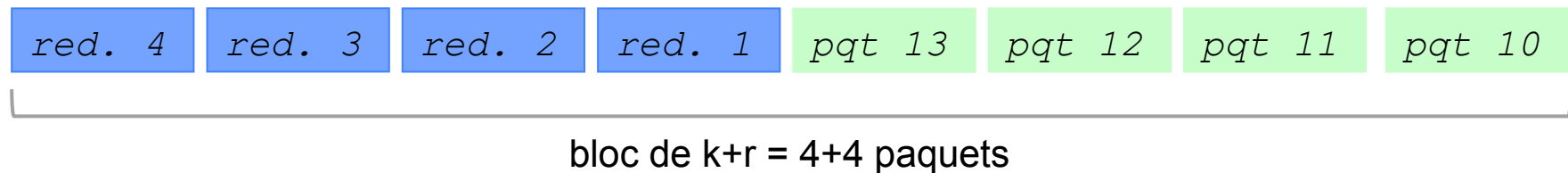
- on introduit une « somme XOR » tous les **k** paquets



- permet de récupérer une perte quelconque (mais unique) parmi les **k+1** paquets

Contre les pertes, des codes FEC... (suite)

- et on peut utiliser des codes plus « puissants »
 - on ajoute r paquets de redondance tous les k paquets

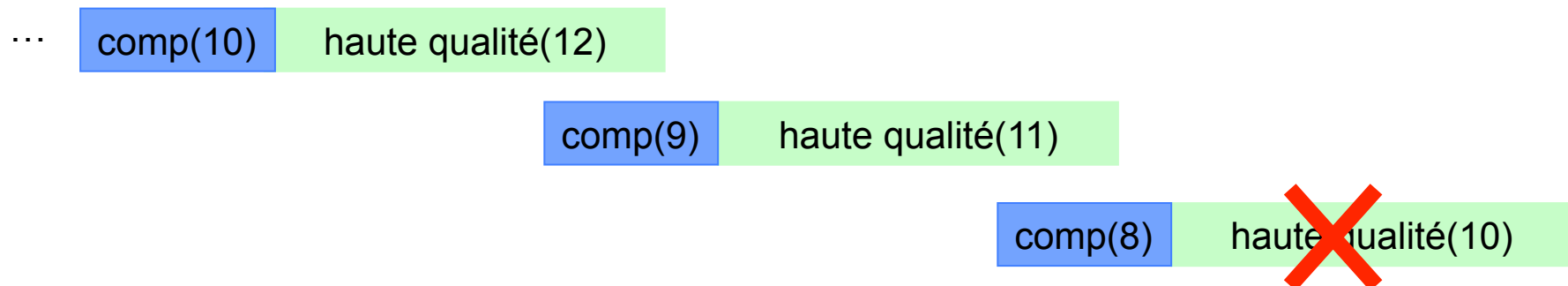


○ permet de récupérer r pertes quelconques parmi les $k+r$ paquets 😊

○ exemple : Reed-Solomon, LDPC-Staircase, etc.

Contre les pertes, des codes FEC... (suite)

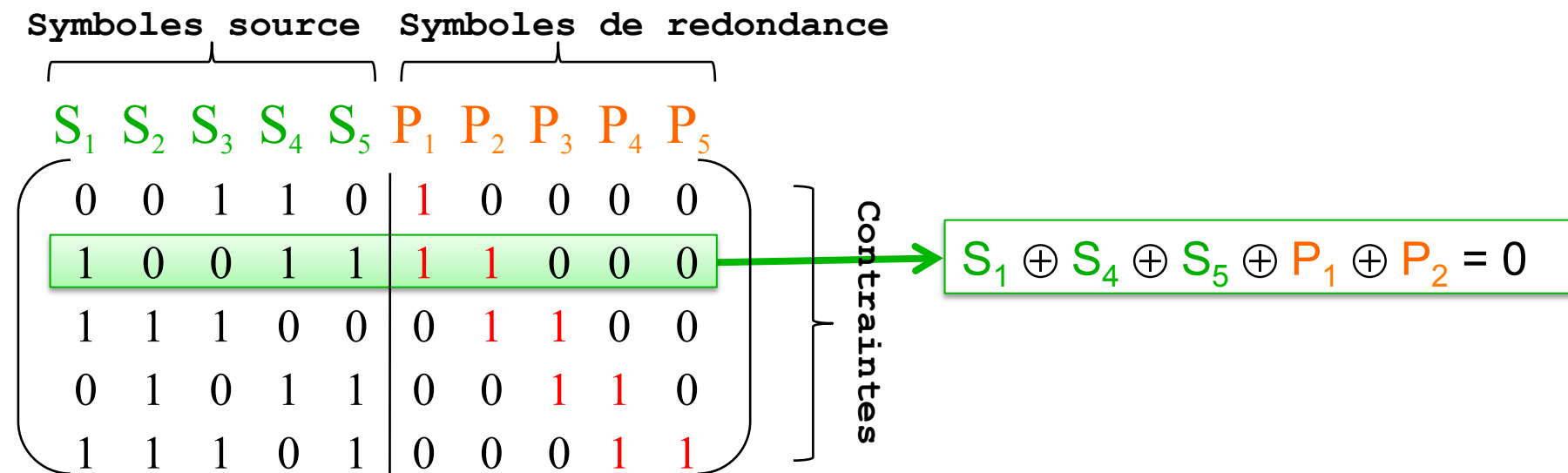
- et parfois une redondance toute simple suffit !
- exemple :
 - flux audio
 - un échantillon audio est encodé deux fois
 - bonne qualité
 - fortement compressé
 - la version compressée est transmise avec un peu de retard
 - en général attaché à un paquet existant
 - en cas de perte isolée, le récepteur pourra remplacer le paquet manquant par sa version compressée reçue plus tard



Gros plan 1: les codes LDPC-Staircase

● le cas des codes LDPC-Staircase

- un standard : <http://tools.ietf.org/rfc/rfc5170.txt>
- intégré au système ISDB-Tmm Japonais 😊
- une solution simple mais très efficace
 - capacités de correction et vitesse d'encodage/décodage
 - système linéaire entre symboles source et de redondance



N_1 "1"s par colonne

Gros plan sur LDPC-Staircase... (suite)

- 1 : encodage des codes LDPC-Staircase

- c'est trivial

- on produit les symboles de redondance en séquence: P1, puis P2, puis P3, etc.

- la vitesse d'encodage est très élevée

- exemple: objet de taille 1MO, 1024 symboles source et 512 symboles de redondance, Intel Xeon 5120/1.86GHz/64-bit Linux

k	code rate	N1	encoding speed	number XOR / k
1024	2/3	5	3.33 Gbps	5.499

Gros plan sur LDPC-Staircase... (suite)

● 2 : décodage des codes LDPC-Staircase

○ consiste à résoudre un système linéaire

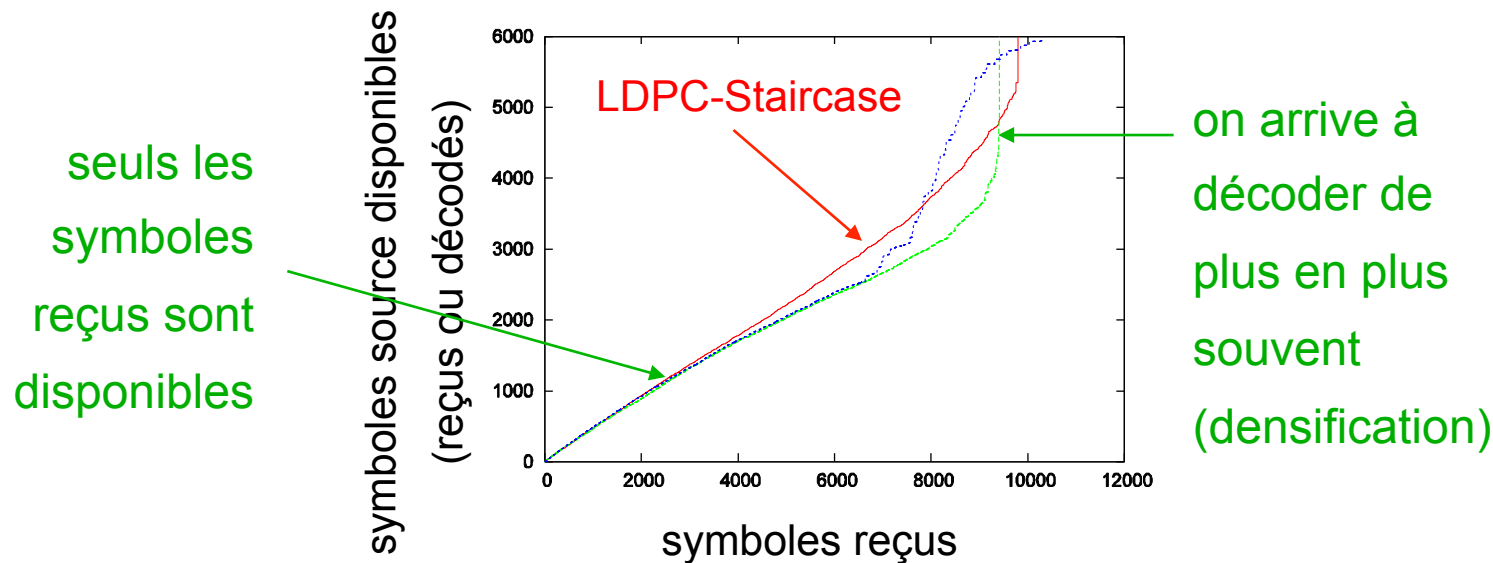
- les variables sont les symboles manquants

○ solution 1: **décodage Itératif** (IT)

○ Une approche triviale

- si une équation n'a qu'une variable, sa valeur est connue. On la substitue dans les autres équations où elle intervient, et ainsi de suite
- possible car le système est très **creux** !

Gros plan sur LDPC-Staircase... (suite)



○très rapide mais sous optimal

- certains systèmes, bien que non singuliers, ne pourront pas être résolus

○solution 2: **élimination de Gauss** (ML, Max. Likelihood)

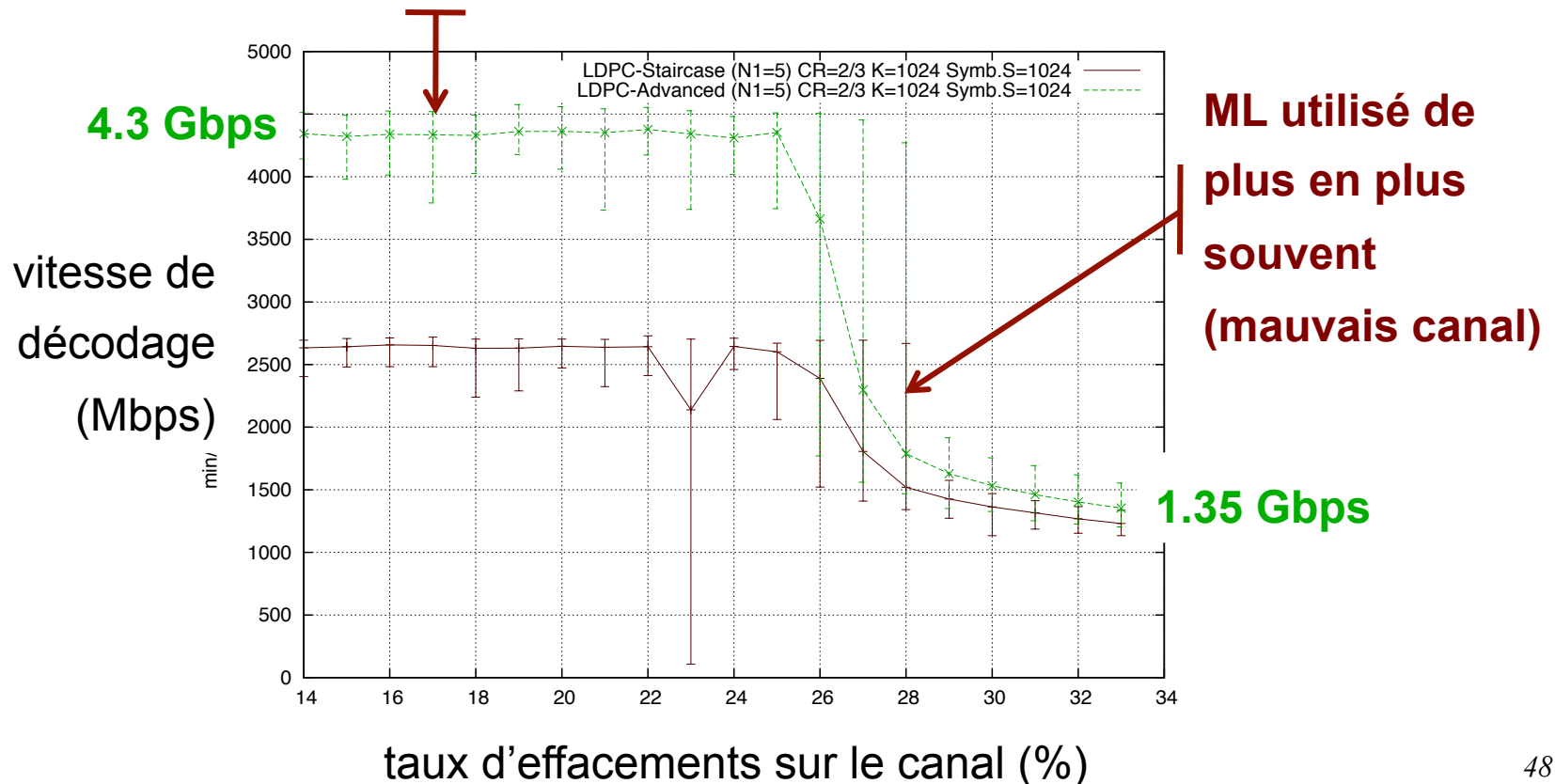
○optimal mais plus coûteux

- tout système linéaire non singulier sera résolu

Gros plan sur LDPC-Staircase... (suite)

- **exemple:** objet de taille 1MO, 1024 symboles source et 512 symboles de redondance, $N_1=5$, Intel Xeon 5120/1.86GHz/64-bit Linux
- **vitesse de décodage**

IT suffisant (bon canal)

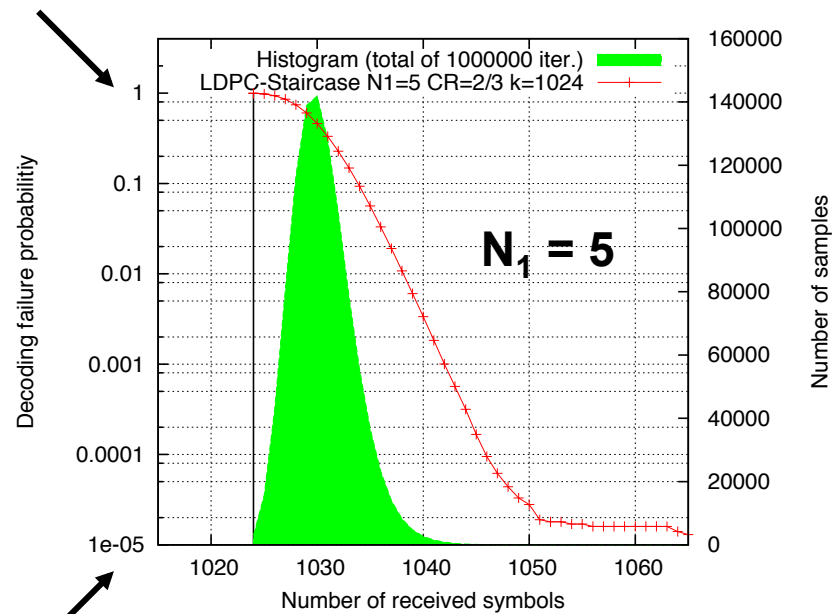


Gros plan sur LDPC-Staircase... (suite)

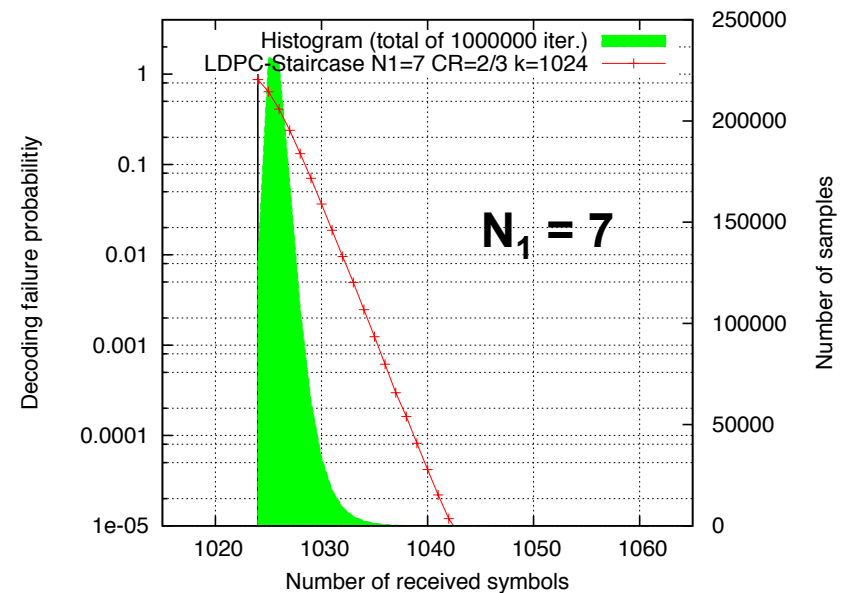
- capacités de correction

paramètres	moyenne	overhead une probabilité d'échec $\leq 10^{-4}$
$k=1024, N_1=5$	0.636%	dès 1046 symboles reçus (2.1% ou 22 symboles)
$k=1024, N_1=7$	0.238%	dès 1039 symboles reçus (1.5% ou 15 symboles)

Pr = 1 : décodage impossible



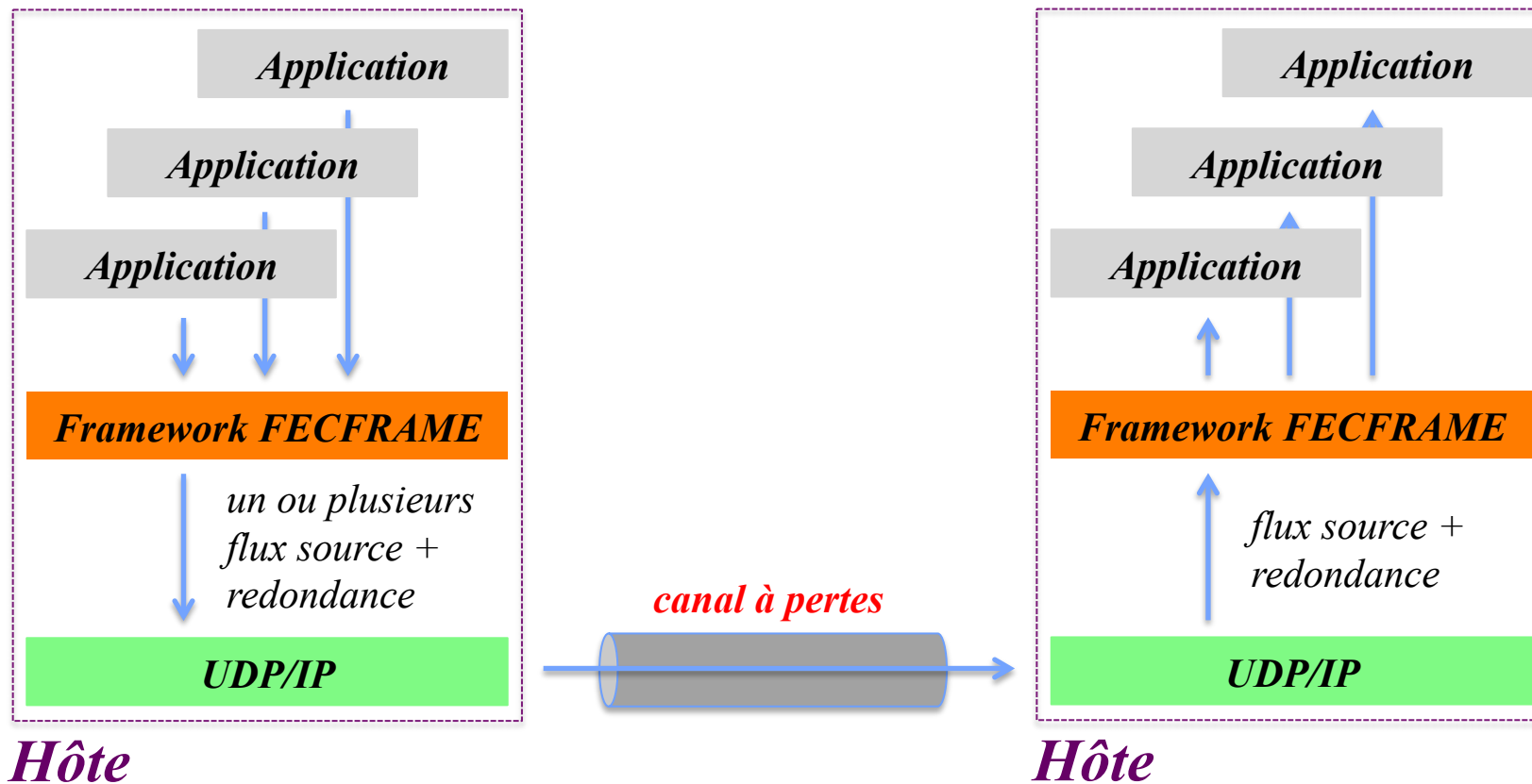
Pr << 1 : forte probabilité de décodage



Gros plan 2: FECFRAME

- FECFRAME, ou comment utiliser les codes FEC dans une solution de streaming

○RFC 6363, October 2011, <http://tools.ietf.org/rfc/rfc6363.txt>



Gros plan sur FECFRAME... (suite)

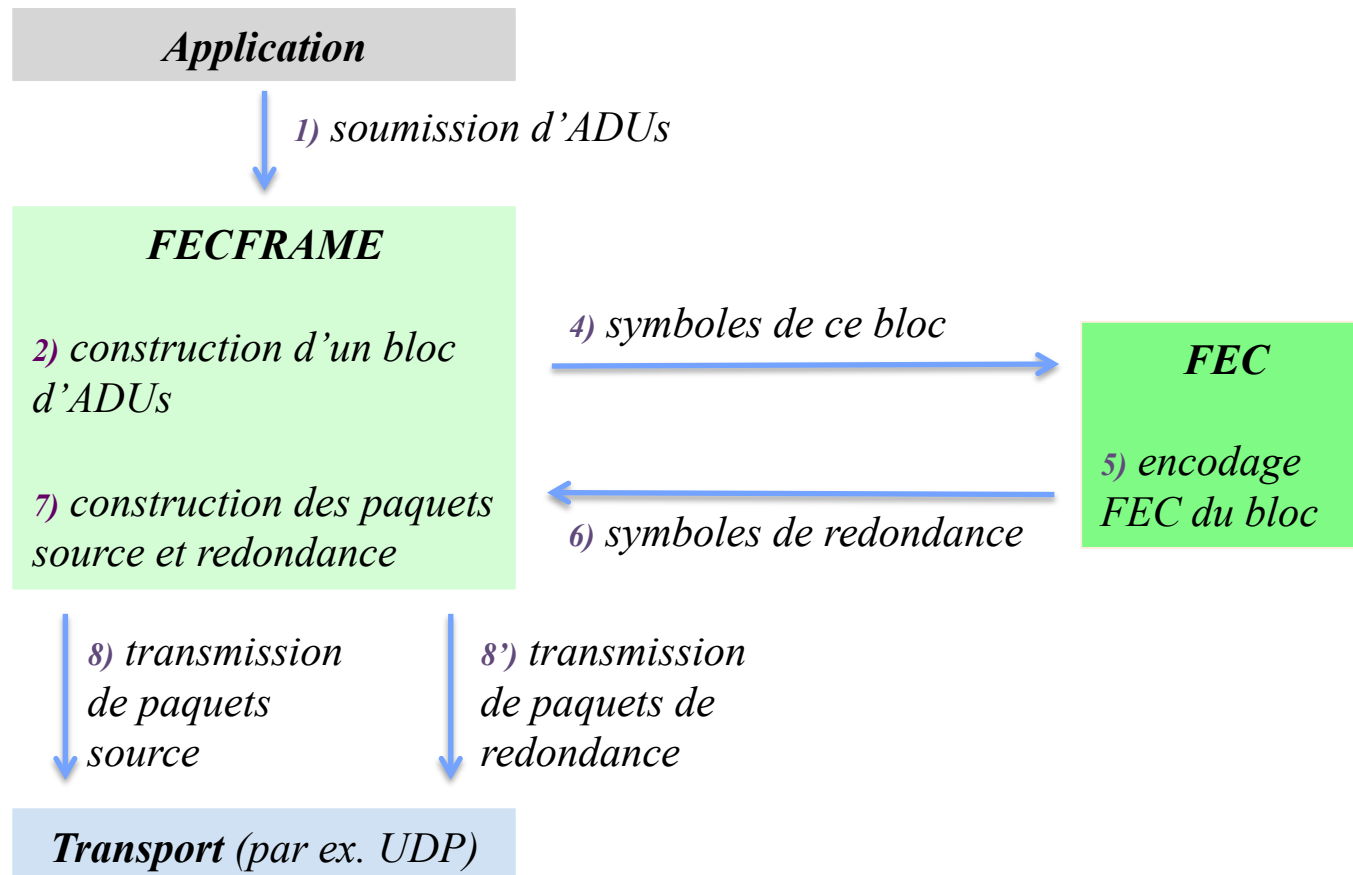
- dans le détail, à l'émetteur...

1. on accumule les données de l'application
2. lorsque l'on en a suffisamment ou que l'on est limité par les contraintes temps-réel, on découpe en symboles et on envoie le tout au codec FEC
3. on récupère les symboles de redondances
4. on transmet les symboles source et redondance sur le réseau

- idem coté récepteur

Gros plan sur FECFRAME... (suite)

○ schéma de fonctionnement à l'émetteur



En résumé...

- on a vu...

- les modèles de transmission point à point (ex. VOD / YouTube) et multipoints (ex. TV sur ADSL)
 - très impactant
- la nécessité de bufferiser dans le cas du streaming
 - aide à combattre les aléas de délai de transmission variable
- la gestion des erreurs
 - en point à point ⇒ pris en charge par TCP
 - en multipoints ⇒ pris en charge par les codes FEC
 - exemple de code FEC : LDPC-Staircase
 - exemple d'usage : FECFRAME

● 4^{ème} partie

Un exemple

- le standard japonais ISDB-Tmm

« Mobile Media Broadcasting: ISDB-Tmm »,
Shinichiro Tonooka, MMBI Inc., Mai 2011.

http://www.itu.int/ITU-D/asp/CMS/Events/2011/DigitalBroadcast-May2011/S5_MMBI_Japan.pdf

Questions ?

- vous aimez les sciences informatiques ? Allez voir le site Interstice, édité par l'Inria :

○ <http://interstices.info/>