Codes correcteurs d'erreurs (FEC) dans les réseaux : principe et applications dans des réseaux pair-à-pair et sans fil.

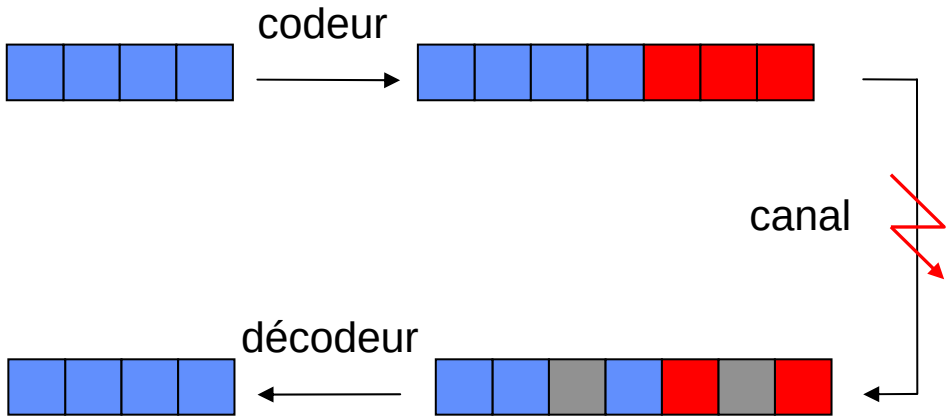
Jérôme Lacan, DMI, ENSICA, Toulouse

Plan de l'exposé

1. Codes correcteurs d'erreurs : principe et applications dans les réseaux.
2. Des codes dans les réseaux pair-à-pair
3. Mécanismes de contrôle d'erreurs pour ses transmissions multimédia multipoints sans fil.

1 – Codes correcteurs d'erreurs : principe et applications dans les réseaux.

Principe d'un code correcteur d'erreur



- **Exemples de codes :**
  - alphabet des aviateurs : a➔alpha, b➔bravo, c➔charlie, ...
  - code à répétition : 0➔000, 1➔111
  - Hamming, BCH, Reed-Solomon, convolutifs (Viterbi), LDPC,...

- **Idée fondamentale :**

  distribuer une quantité d'information de k symboles sur n symboles (n > k)

Applications des codes correcteurs d'erreurs

- **Stockage :**
  - Disques optiques (CD-ROM, DVD)
  - Disques durs + systèmes RAID
  - mémoire (SDRAM, RDRAM)
- **Transmissions couches basses :**
  - Satellite, sondes spatiales
  - GSM, UMTS, 802.11 (a et b+), HiperLAN,
  - Ethernet 10Gb ...

Fiabilisation des transmissions où la ré-émission de l'unité d'information n'est pas possible

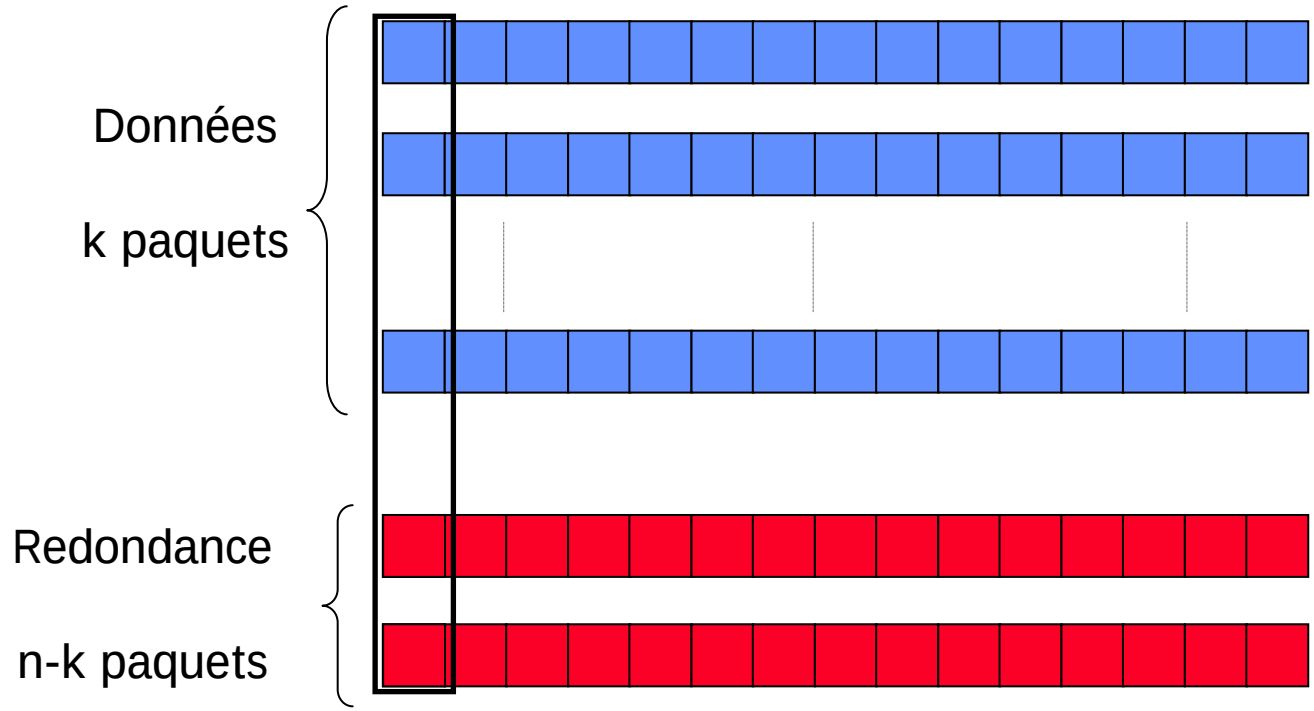
Des codes sur les couches hautes ?

- **Inconvénients :**
  - éloignés des erreurs de transmission
  - erreurs paquets (CRC)
  - codage/décodage logiciel
  - par rapport aux retransmissions :
    - **complexité de l'implémentation**
    - **en cas de mauvais dimensionnement du code, mauvaise utilisation de la bande passante**
- **Avantages :**
  - gestion des données au niveau sémantique
  - différentiation de service possible en fonction des utilisateurs
  - adaptabilité des mécanismes
  - par rapport aux retransmissions :
    - **évite les délais de retransmission (temps-réel)**
    - **compense avec un paquet de redondance des pertes distinctes chez plusieurs récepteurs (multipoint)**

Problématiques dans le domaine des réseaux

- **Grand nombre de contextes différents:**
  - différents types de canaux
  - communication : 1 vers 1, 1 vers n, n vers 1, ...
  - types d'applications :
    - **temps réel,**
    - **stockage distribué,**
    - **routage**
  - types de machines
- **Quelques constantes :**
  - Canal à effacement de paquets
  - nécessité d'algorithmes de codage/décodage performants en logiciel
  - nécessité d'une capacité de correction optimale: codes (quasi-) MDS

Schéma de codage en mode paquet



Calcul de la parité
du mot codé

Une solution classique : les codes MDS (1)

- **paramètres : [n, k, n-k+1] sur F_q où q > n.**

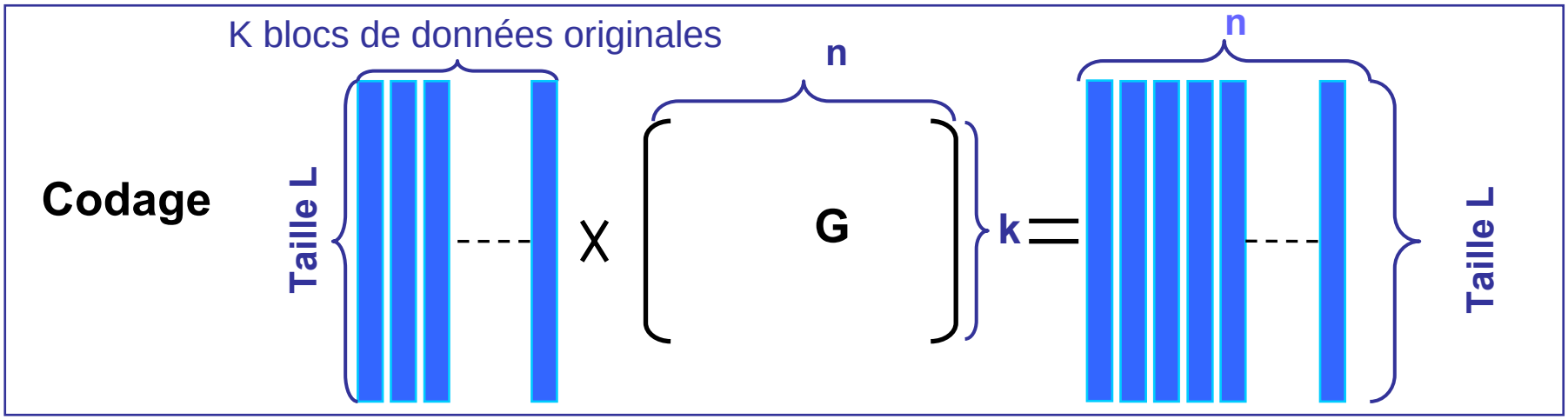
- **avantages de ces codes :**

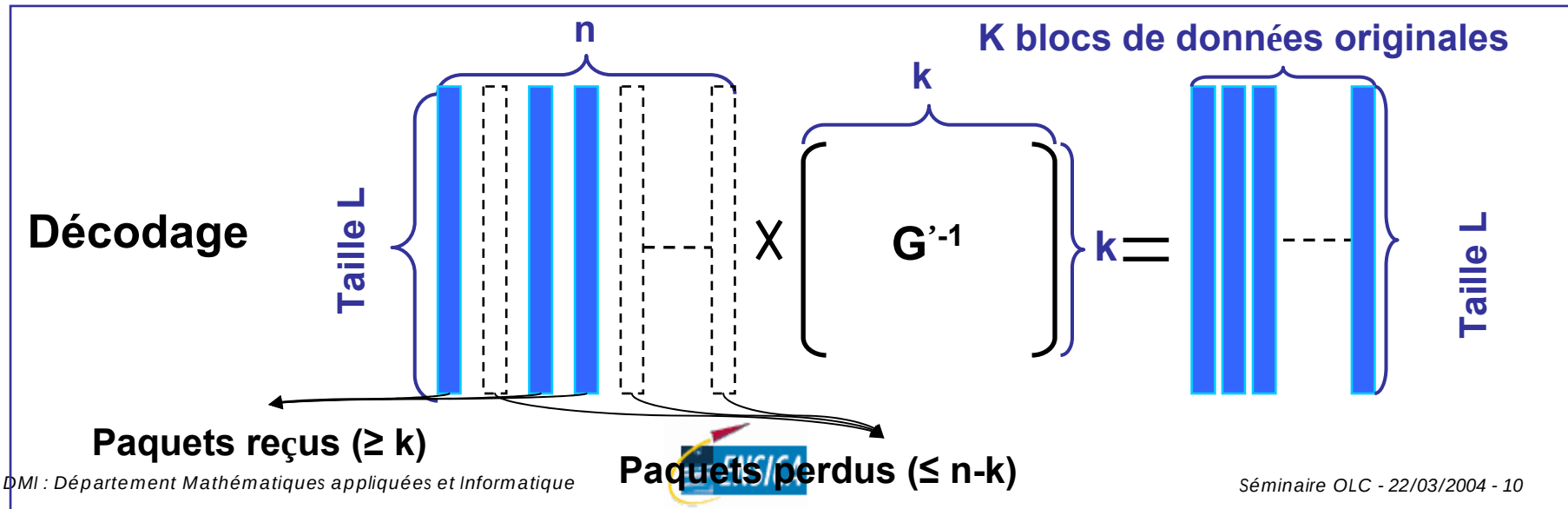
  - optimal pour le canal à effacement

    ➔ pour t symboles d'information perdus (localisés), t symboles de redondance sont nécessaires.

  - souplesse : les MDS peuvent être très facilement tronqués et leurs paramètres peuvent être modifiés dynamiquement.

Une solution classique : les codes MDS (2)





Une solution classique : les codes MDS (3)

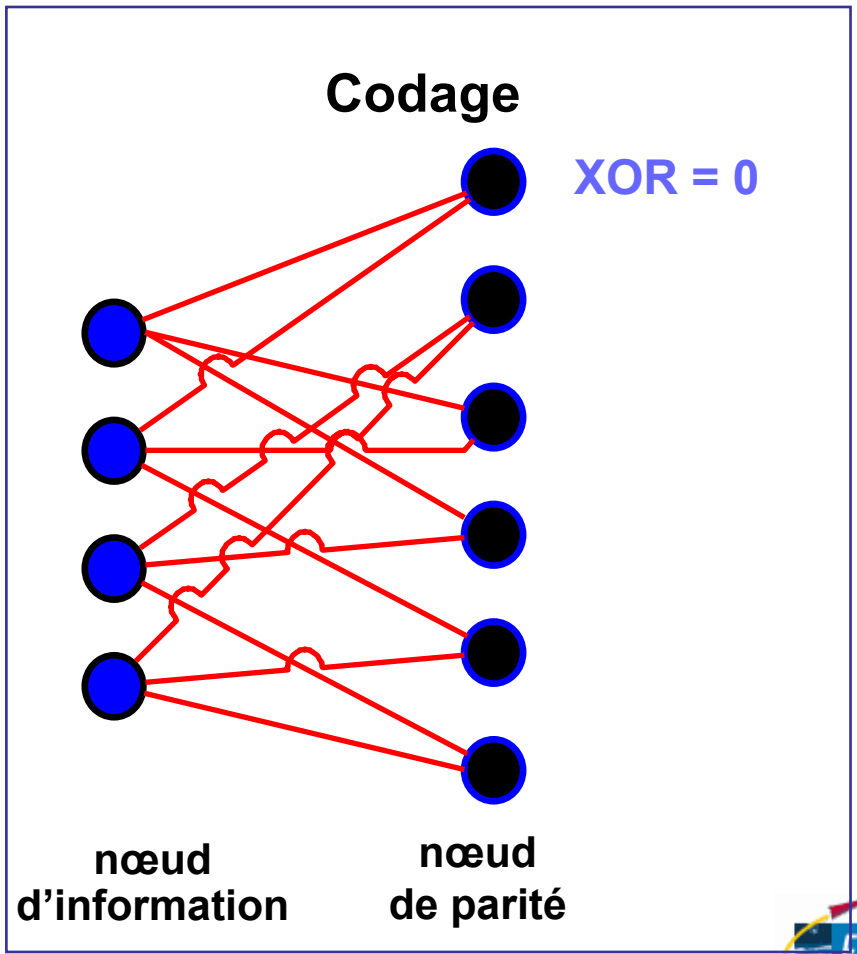
- **2 codes proposés :**
  - Rizzo 1997 : codes MDS systématiques dont la matrice génératrice est construite à partir d'une matrice de Vandermonde $(a_i{}^j)_{i,j}$
  - Luby 1995 : codes MDS systématiques dont la matrice génératrice est construite à partir d'une matrice de Cauchy: $(\frac{1}{a_i+b_j})_{i,j}$
- **inconvénients :**
  - temps de décodage :
    - **toutes les opérations se font dans un corps fini**
    - **accès à un tableau lié à la multiplication ou à l'addition dans le corps**
  - vitesses de codage/décodage (sur un P IV 2GHz) :
    - **[128, 256] sur $F_{2^{16}}$: 4.8 Mbits/secondes en décodage**
    - **[1024 , 2048] : 0.56 Mbits/s en décodage**

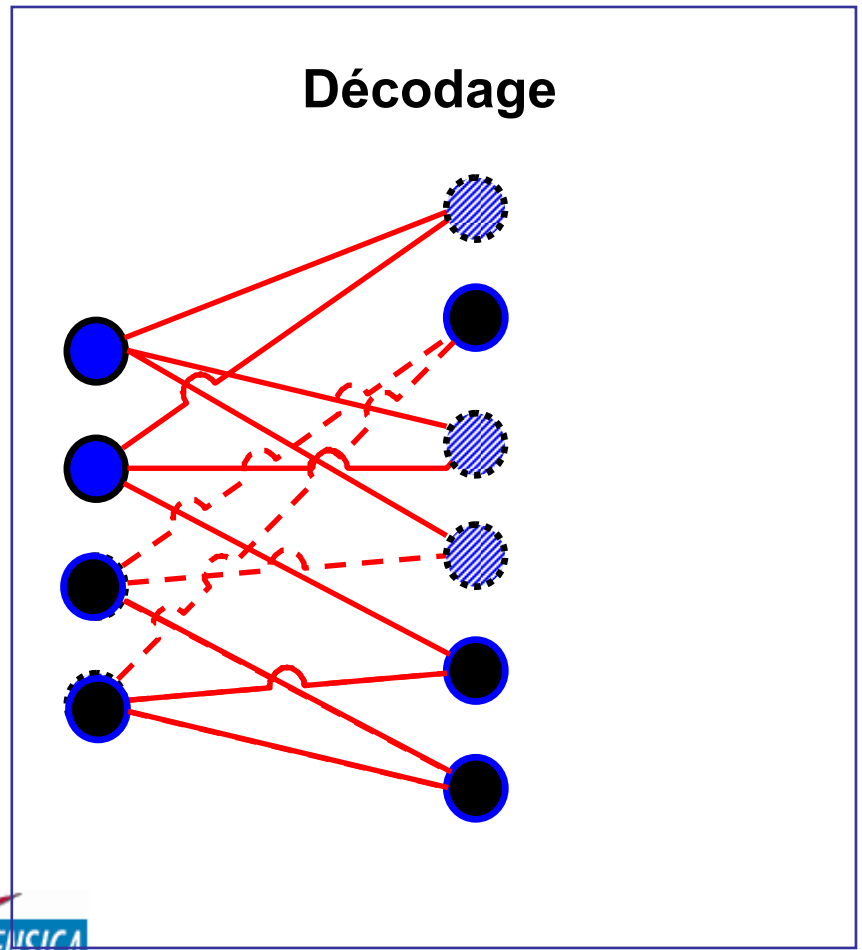
Résultats ENSICA : construction de codes MDS

- **Implémentation d'un nouveau code MDS :**
  - ♦ coefficient entiers (opérations modulo $2^{16}+1$)
  - ♦ coefficients représentés par des symboles codés sur 16 bits
  - ♦ ajout de 16 bits dans l'entête du paquet
  - ♦ utilisation d'une matrice de Cauchy
  - ♦ performances mesurées :
    - ▪ **par rapport à Rizzo : codage × 4 et décodage × 7**
    - ▪ **par rapport à Luby : codage et décodage × 1.3**
- **Proposition d'un code dont la complexité est optimale**
  - ♦ basé sur une nouvelle construction de matrice dont toute sous-matrice est inversible
  - ♦ utilisation de 2 matrices de vandermonde
  - ♦ complexités de codage et de décodage meilleures que les constructions existantes

L'autre solution : les codes LDPC

- **Principe : décodage basé sur le graphe construit à partir de la matrice de test creuse.**





L'autre solution : les codes LDPC (2)

- **Avantages :**
  - vitesses de codage/décodage nettement plus rapides que les MDS (complexités linéaires) ➔ exemple : pour un code LDPC [200,100], décodage de l'ordre du Gbits/s
  - Facilité pour construire des codes longs

- **Inconvénient :**
  - l'inefficacité du décodage :

$$\frac{nb\ symboles\ necessaires\ -\ k}{k}$$

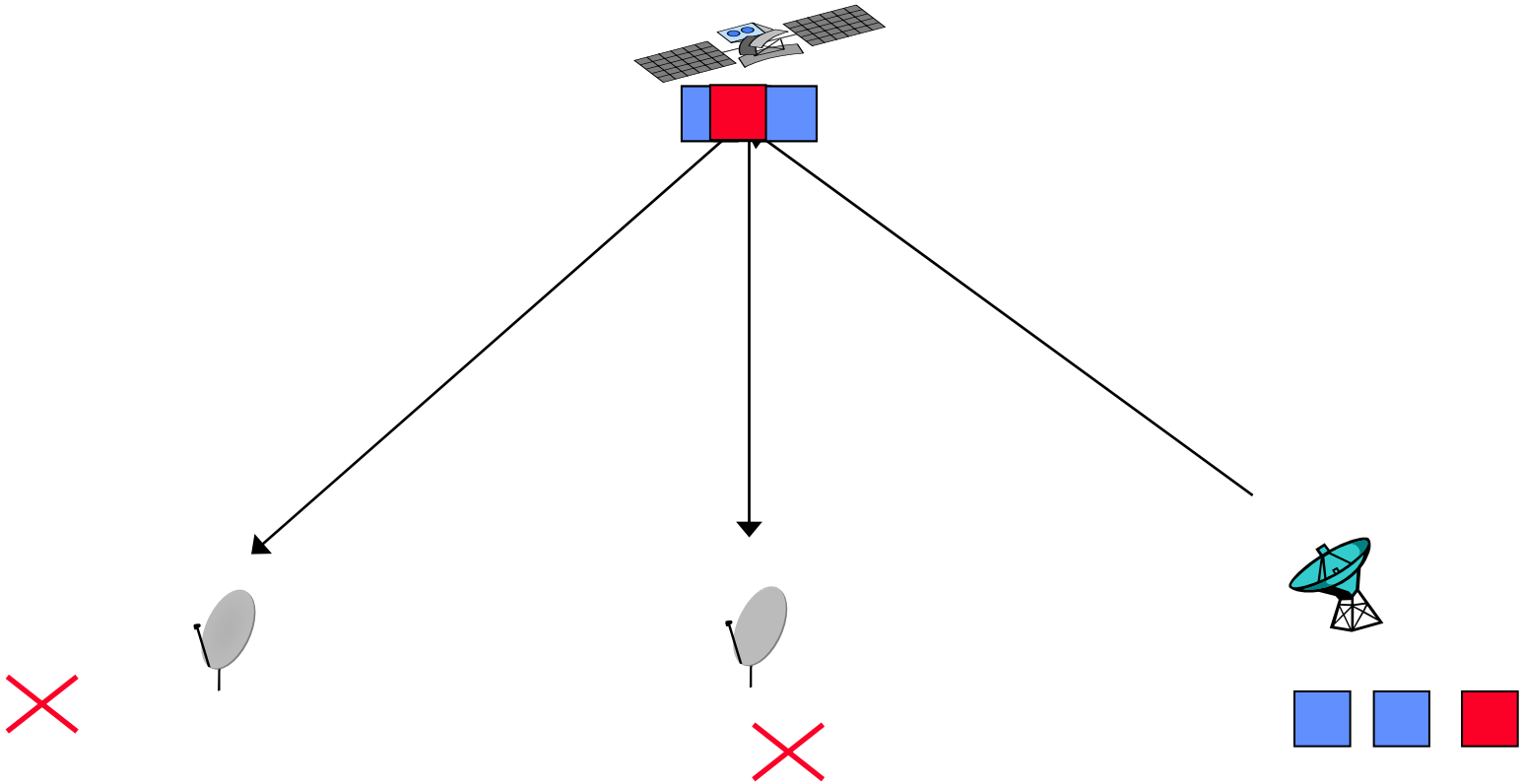
  - L'inefficacité tend vers 0 lorsque la longueur du code tend vers l'infini, mais avec des paramètres réalistes (longueur < 1000), la plupart des familles de codes LDPC ont une inefficacité de l'ordre de 10% ➔ impensable pour un grand nombre d'applications réseau

Applications des codes dans les réseaux (1)

- **transmissions multimédia temps-réel**
  - ♦ but : éviter les délais dus aux retransmissions
  - ♦ audio, téléphonie sur IP :
    - **paquets de taille fixe**
    - **délais inter-paquets constants**
    - **utilisation de codes avec un k petit (~10) et très peu de redondance (1 ou 2 paquets)**
  - ♦ vidéo :
    - **problème: les différents types d'images n'ont pas la même taille.**
    - **idée: protéger les paquets proportionnellement à leur importance ➔ codage à protection inégale (PET : Priority Encoding Transmission)**
  - ♦ à utiliser avec/à la place des mécanismes de "dissimulation (masquage) de pertes" des codages audio et vidéo.

Applications des codes dans les réseaux (2)

- **Transmissions multicast, intérêt principal :**



Applications des codes dans les réseaux (2)

- **Transmissions multicast (suite) :**
  - problème : optimiser la quantité de redondance par rapport au taux de perte de paquets
  - Utilisation de retours pour ajuster cette quantité de redondance (mécanismes hybrides FEC-ARQ)
  - de nombreux systèmes ont été proposés utilisant :
    - **la structure de l'arbre multicast (reprises d'erreurs locales, ...)**
    - **une transmission en couches à débits variables pour mettre en place un mécanisme de contrôle de congestion multicast.**
    - **des variations de la redondance adaptées au type des erreurs (pertes de paquets groupées dans des transmissions sans fil).**
    - **...**

Applications des codes dans les réseaux (2)

- **Travaux ENSICA sur les transmissions multicast par satellite :**
  - Laisser remonter les erreurs bits pour réduire les "gaspillages" d'informations utiles par les CRC (thèse F. Arnal-Alcatel).
    - **évaluation de cette proposition sur une architecture UDP/IP/MPE/MPEG2-TS .**
    - **proposition 1 : MPHP (protection des entêtes)**
    - **proposition 2 : codes correcteurs au niveau transport corrigeant des erreurs bits et des effacements ou UDP-Lite**
  - Diffusion de contenus multimédia (non temps-réel) par satellite vers des mobiles connectés simultanément à un réseau 3G-4G (thèse J. Fimes-Alcatel)
    - **1 seule antenne : 1 seul émetteur à la fois➔ perte d'une proportion importante des données émis par le satellite.**
    - **Utilisation de codes au niveau transport pour compenser ces pertes**

Applications des codes dans les réseaux (3)

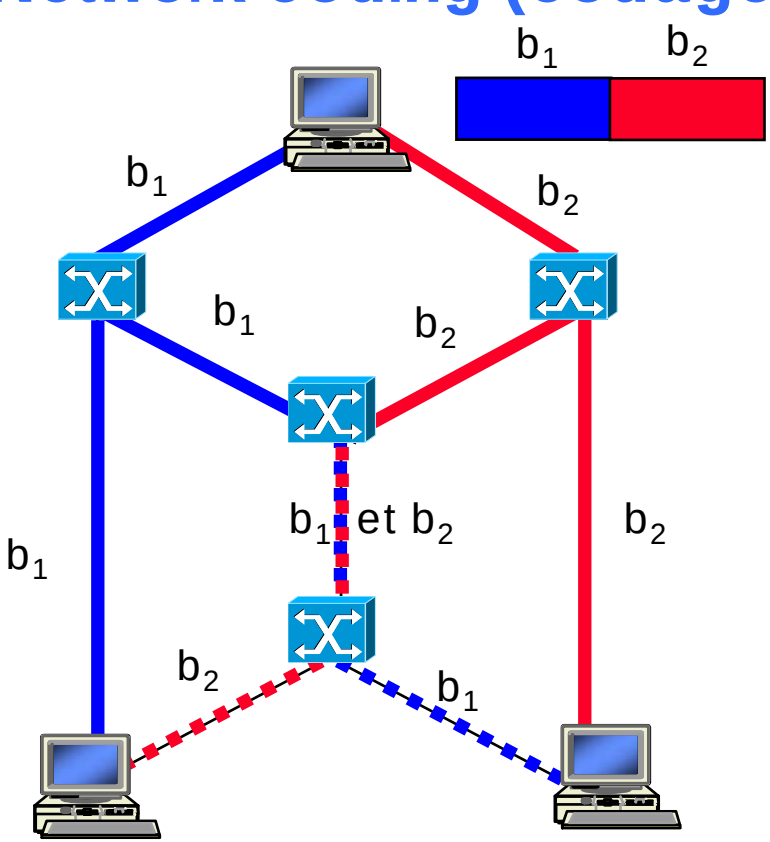
- **Stockage distribué: idée de base : un fichier est scindé en k blocs qui sont codés en n blocs et dispersés sur le réseau.**

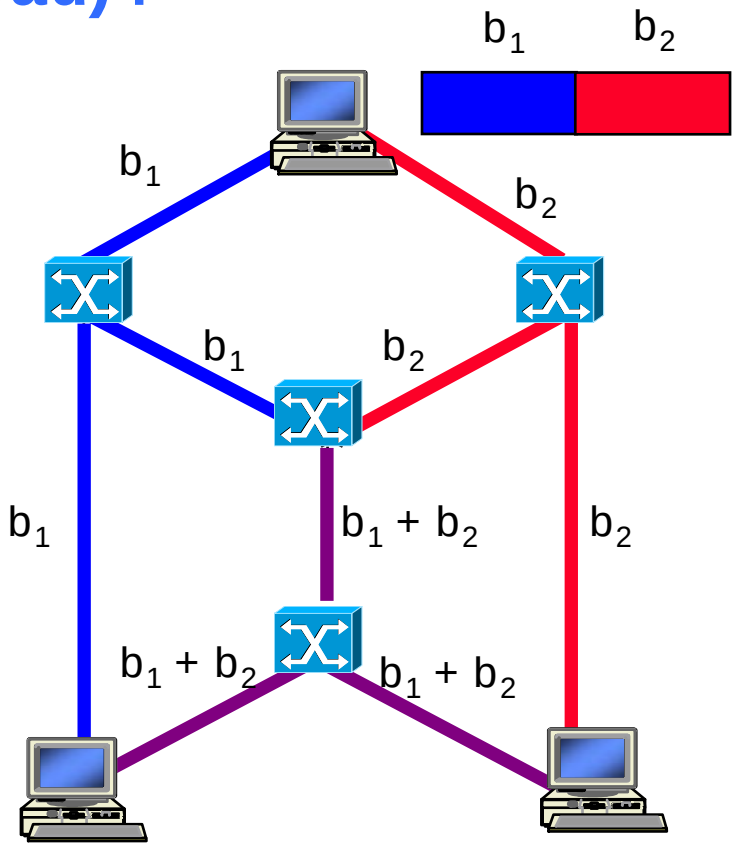
➔ **amélioration de la tolérance aux fautes du système**

- **Applications :**
  - ♦ systèmes de stockage RAID (versions > 1)
  - ♦ IDA: Information dispersal algorithm (Rabin 89),
    - **amélioration de la tolérance aux fautes**
    - **partage du secret : chaque serveur de stockage ne peut interpréter seul les données**
    - **distribution de la charge**

Applications des codes dans les réseaux (4)

- **Network coding (codage réseau) :**





Applications des codes dans les réseaux (5)

- **Encore d'autres applications :**
  - ♦ liens sans voie de retour (e.g. satellite)
  - ♦ transmissions de m vers 1 :
    - **le fichier à transmettre est scindé en k blocs qui sont codés en mk blocs.**
    - **Chaque émetteur prend k blocs différents et les transmet.**
    - **Le récepteur arrête la transmission dès qu'il a reçu k blocs.**
  - ♦ transmissions utilisant des chemins multiples
    - **le fichier à transmettre est scindé en k blocs et n-k blocs de redondance sont calculés.**
    - **Ces blocs sont transmis de l'émetteur vers le récepteur via n chemins différents (!).**
    - **La réception est OK dès que le récepteur reçoit k blocs.**
  - ♦ …

2 - Des codes dans des réseaux Pair-à-Pair

- **avec Laurent Dairaine et Laurent Lancérica**

Plan de cette partie

- **Rappels sur les réseaux Pair-à-pair (P2P).**
- **Accélération des téléchargements avec des codes à effacement.**
- **Comparaison**
- **Choix du code à effacement**
- **Conclusion**

Systèmes de stockage distribués et codes correcteurs d'erreurs

- **tolérance aux fautes : RAID [Paterson et al. 88], Information Dispersal Algorithm [Rabin 89], ...**
- **Partage du secret : Information Dispersal Algorithm [Rabin 89], ...**
- **Accès aux données efficace : [Naor / Roth 91] , [Jiang / Bruck 02], ...**

Réseaux pair-à-pair (P2P)

- **Réseau logique de haut niveau**
  - Noeuds (pairs) interconnectés par un réseau physique
  - Dissémination de l'information sur l'ensemble des pairs
  - Distribution de la charge sur le réseau physique et sur les utilisateurs.

- **Thèmes de recherche**
  - Passage à l'échelle, routage, tolérance aux fautes
  - Authentification, sécurité
  - Recherche des données et des pairs les plus "proches" contenant ces données
  - Évaluation des performances

- **Exemples de systèmes P2P :**
  - calculs distribués
  - réseaux de mobiles (ad hoc)
  - Systèmes de partage de fichiers (Kazaa, edonkey, Gnutella, ...)

Améliorer l'accès aux données

- **Classiquement, optimisation de la localisation des données**

- **Notre proposition: utiliser des codes à effacement dans les réseaux de partage de fichiers pour améliorer les perfrmances de téléchargement:**
  - Idée de base : pour la même quantité de données stockées, télécharger des parties de fichiers codées est plus efficace que télécharger des parties de fichiers répliquées
  - Comparaison
    - **Schéma classique: Réplication de fichiers entiers**
      - ➔ le fichier est répliqué m fois sur le réseau P2P
    - **Dissémination de blocs de fichiers non codés**
      - ➔ Chaque bloc est répliqué m fois
    - **Dissémination de blocs de fichiers codés**
      - ➔ Les blocs sont codés avec un code [u k, k] et répliqués m/u fois
  - Téléchargements séquentiels / parallèles

Exemple

Fichier : 4 blocs

Fichier

1 2 3 4 5 6 7 8

Fichier entier répliqué: coût = 4*2 = 8

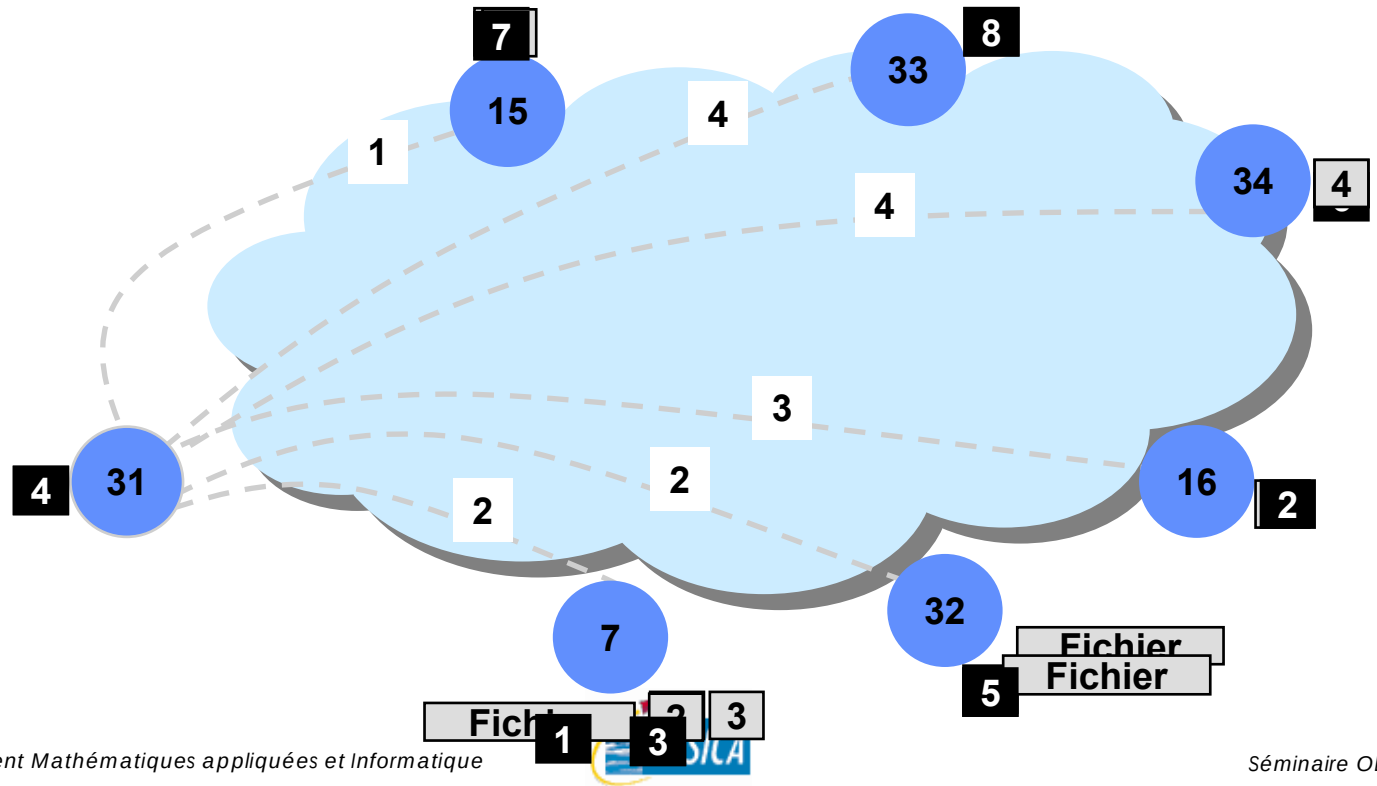
2 copies Nombre total de blocs 8

Dissémination de blocs non codés : coût =1*0+1*1+1*2+1*4 = 7

2 copies Nombre total de blocs 8

Dissémination de blocs codés: coût = 1*0+1*1+2*2 = 5

r = 2 1 copie Nombre total de blocs 8



Soit F un fichier de taille S_F

- **Réplication du fichier entier**
  - r copies du fichier avec les bandes passantes associées: $Bw_1, ..., Bw_r$

$$C = \min_{j=1..r}(S_F / Bw_j)$$

- **Réplication des blocs non codés**
  - F découpé en k blocs
  - r copies de chaque bloc avec les bandes passantes: $Bw_{1,1}, ..., Bw_{1,r}, Bw_{2,1} ..., Bw_{k,r}$
  - Soit $Bw_{i,j_i} = \max_{j=1,...,r} Bw_{i,j}$

$$C = \sum_{i=1,...,k} S_F /(k * Bw_{i,j_i})$$

Modèle (suite)

- **Réplication des blocs codés**
  - F découpé en k blocs qui sont ensuite codés en n blocs (avec un code MDS)
  - r'=r*k/n copies de chaque bloc avec les bandes passantes:

$$Bw_{1,1}, \ldots, Bw_{n,r'}$$

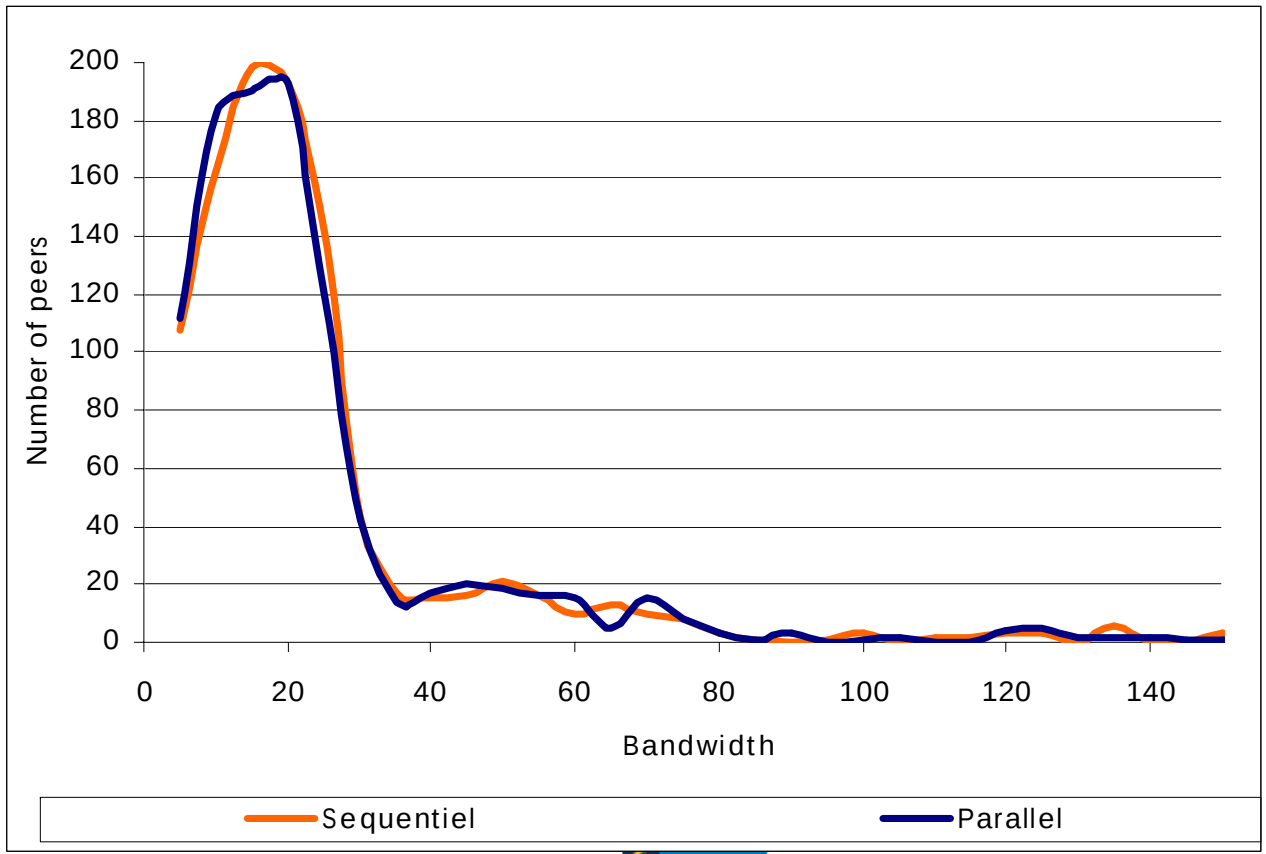
  - Soit $Bw_{i,j_i} = \max_{j=1,\ldots,r'} Bw_{i,j}$
  - Soit f une fonction de

$$\{1,\ldots,n\} \to \{1,\ldots,n\} : Bw_{f(1),j_{f(1)}} \geq \ldots \geq Bw_{f(r'),j_{f(r')}}$$

$$C = \sum_{i=1,\ldots,k} S_F / (k * Bw_{f(i),j_{f(i)}})$$

Distribution des débits

- **Campagne de mesures sur le réseau P2P Gnutella**



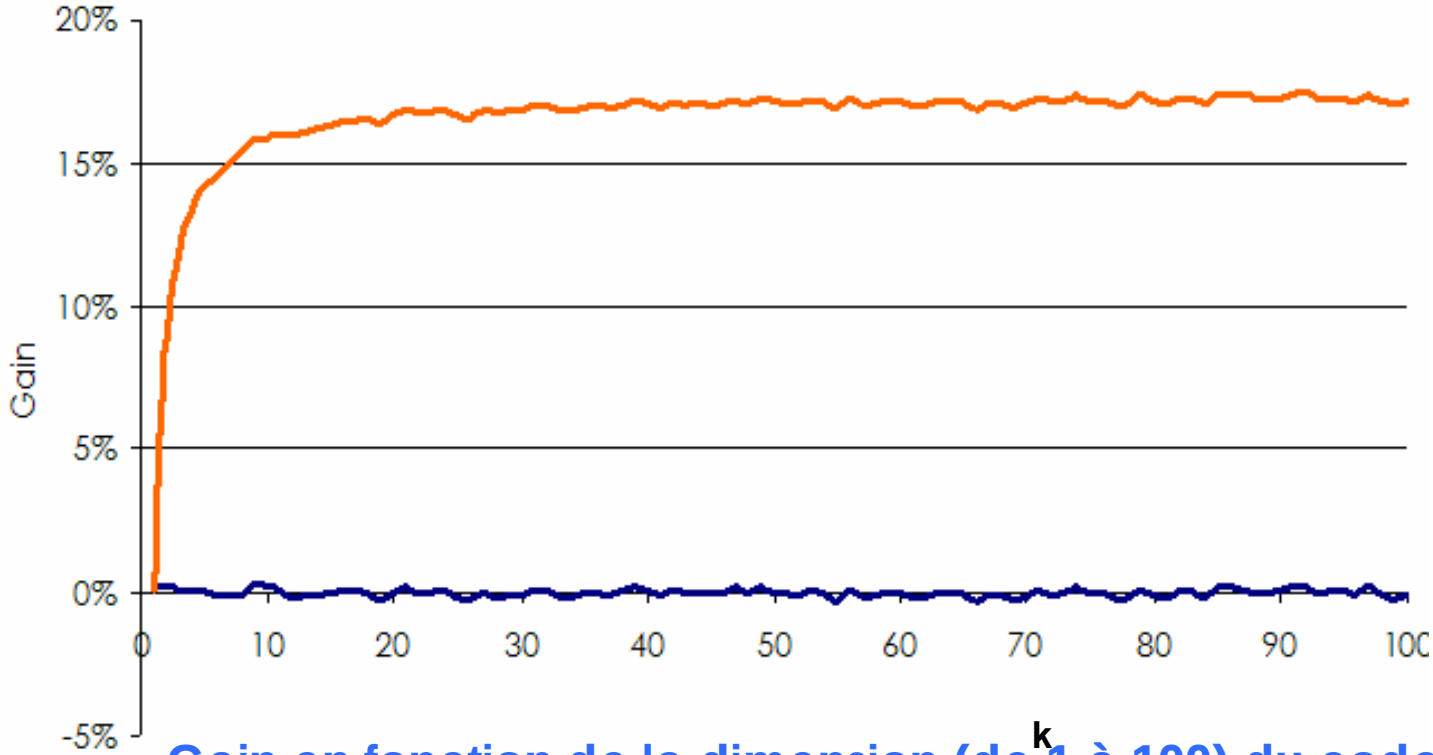
Simulation

- **Simulations**

  Fichier : k blocs
  - Réplication du fichier entier : r copies
  - Dissémination des blocs non codés : k blocs, r copies
  - Dissémination des blocs codés : n=uk blocks et r'=r/u copies (volume stocké constant)

- **Évaluation du coût moyen pour obtenir le fichier avec les 3 stratégies**
  - 1.000.000 tirages effectués chaque cas

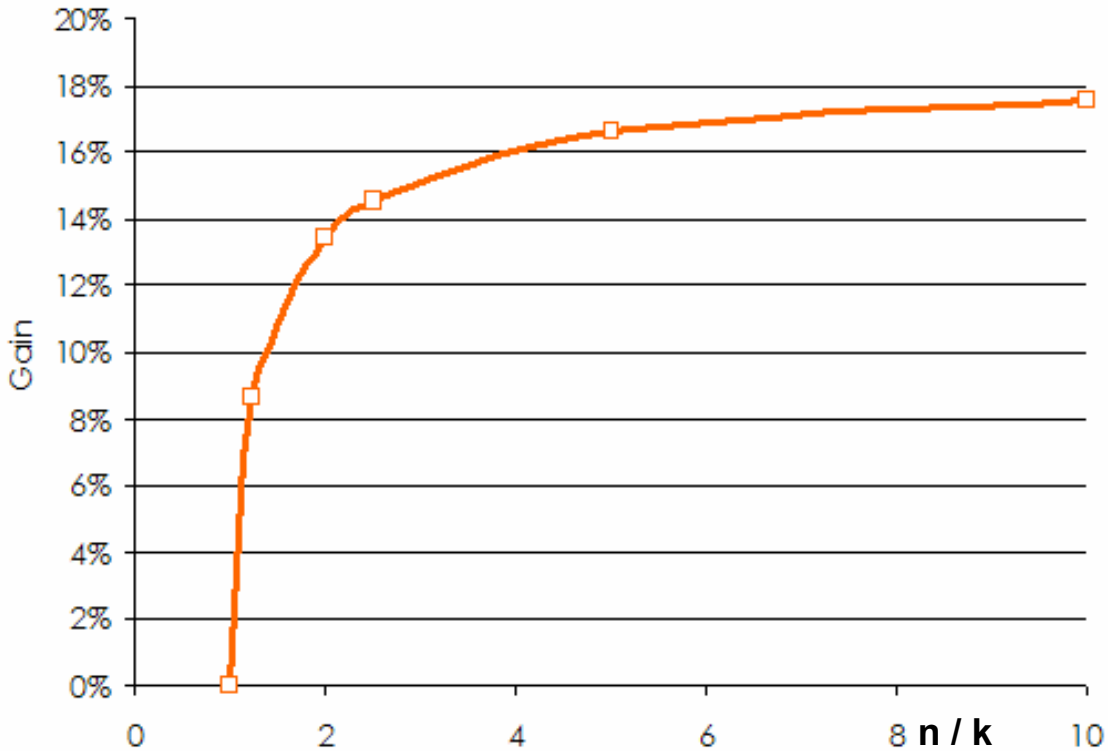
Résultats pour des téléchargements séquentiels (1)



Gain en fonction de la dimension (de 1 à 100) du code

- Référence : 10 copies du fichier entier
- Blocs non codés : 10 copies de chaque bloc
- blocs codés : (u = 5) 2 copies de chaque bloc

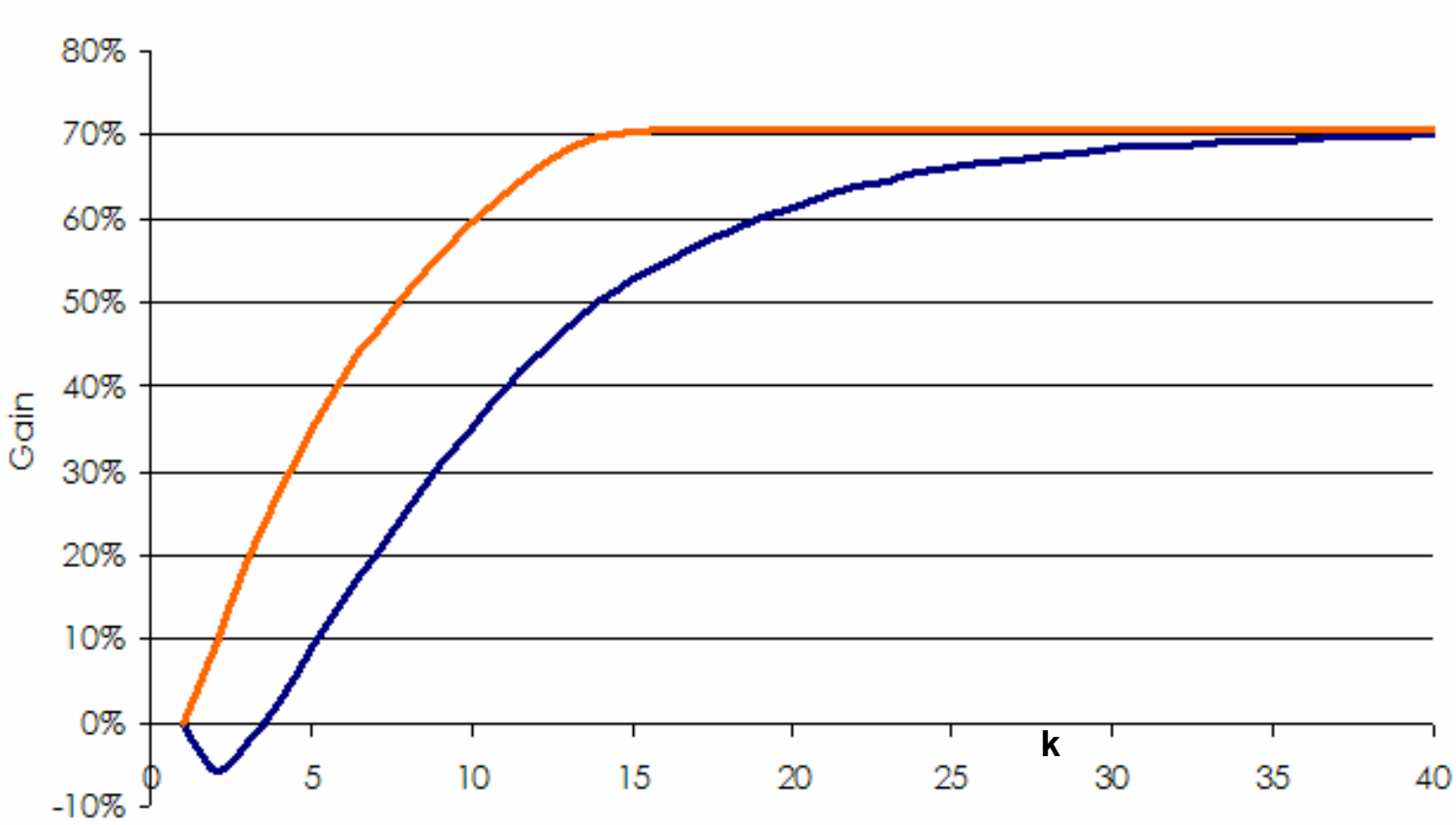
Résultats pour des téléchargements séquentiels (1)



Gain en fonction du taux de codage

- k = 20, chaque bloc est répliqué 40 fois
- Référence : Réplication du fichier entier
- Variation du taux de codage

Résultats pour des téléchargements parallèles (1)



Gain en fonction de la dimension du code.

Contraintes supportées par le code

- **k relativement faible (entre10 et 20)**
- **longueur du code : potentiellement grande (>1000).**
- **codage/décodage rapide en logiciel**
- **Dans la plupart des cas, les vitesses de codage et de décodage sont plus rapides que les taux de transmission**

➔ minimisation de la quantité de données transmises sur le réseau : le code à effacement doit être MDS.

Conclusion de cette partie

- **L'utilisation de codes à effacement permet de diminuer la durée moyenne des téléchargements**
- **Les gains dépendent des paramètres du réseau (distribution des débits, nombres de noeuds,...)**
  - Les paramètres du code et le nombre de copies doivent être adaptés au réseau.
- **Travaux futurs :**
  - évaluation de ce système dans d'autres réseaux P2P (réseaux de mobiles)
  - définition d'un algorithme de dissémination

3 – Évaluation de mécanismes de contrôle d'erreurs dans des transmissions multicast sans fil

- **avec Tanguy Perennou**

introduction

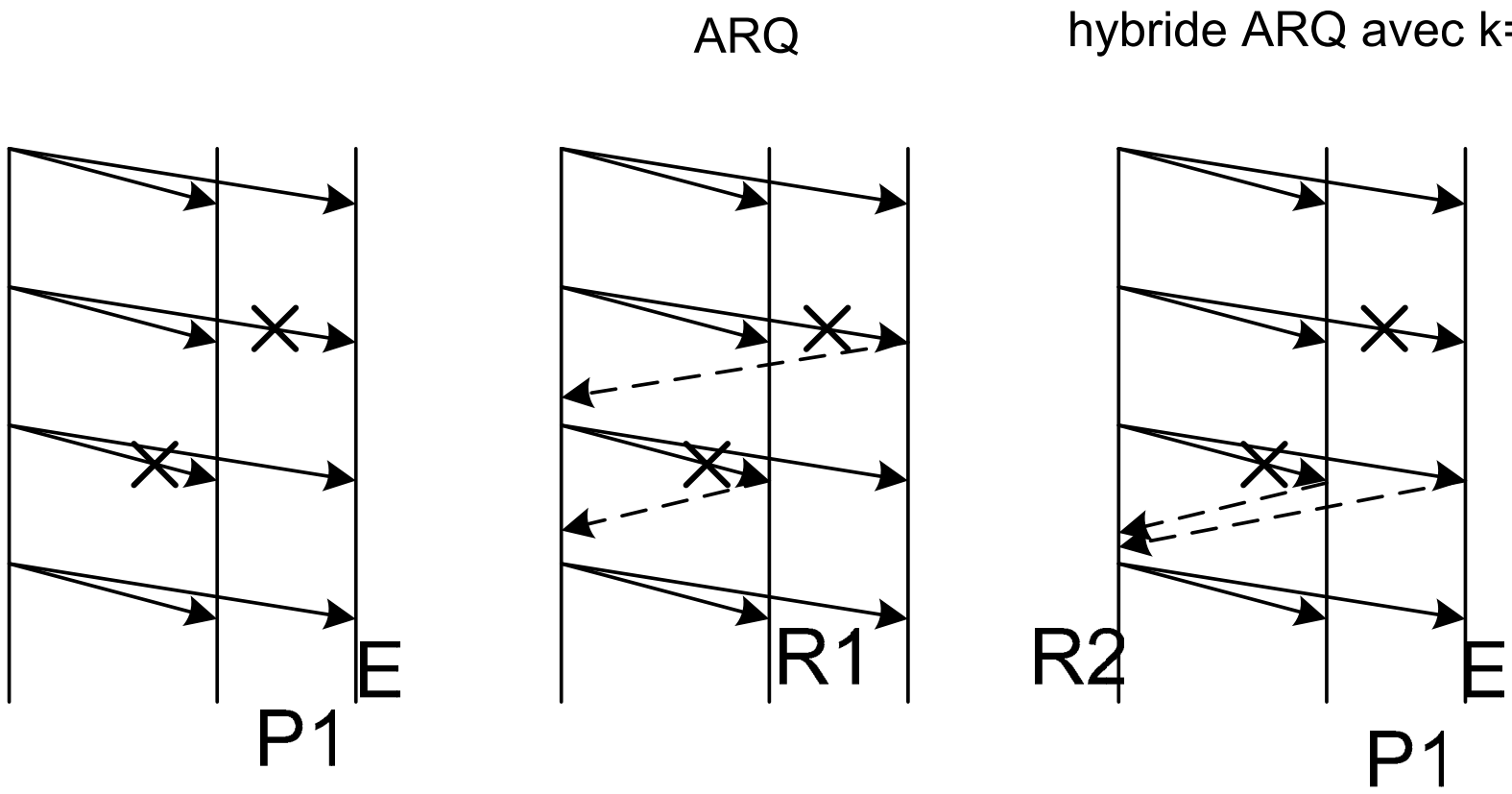
- **Pour des transmissions multicast dans des réseaux 802.11b, la fiabilité au niveau transport doit-elle être assurée par :**
  - des codes correcteurs (FEC)
  - des retransmissions (ARQ)
  - des mécanismes hybrides FEC-ARQ ?
- **Postulat : pour savoir quel(le famille de) mécanismes(s) choisir, il faut connaître le canal de transmission.**
- **Travail initié par le DEA de Hanaa El-Natour**
  - tests des performance des codes sur des machines mobiles légères (PDA) en terme de vitesse de codage/décodage et de dépense d'énergie (non présenté ici)
  - observation du canal 802.11b et choix des mécanismes

Méthodologie (1)

- **Constat : on ne peut pas implémenter tous les mécanismes et les tester sur le réseau réel.**
- **approche classique : modéliser, implémenter et tester certains mécanismes dans un simulateur à événements discrets comme ns-2**
- **notre approche : recueillir des traces réelles (sans aucun mécanisme de contrôle d'erreur) et simuler sur ces traces les différents mécanismes.**

Méthodologie (2)

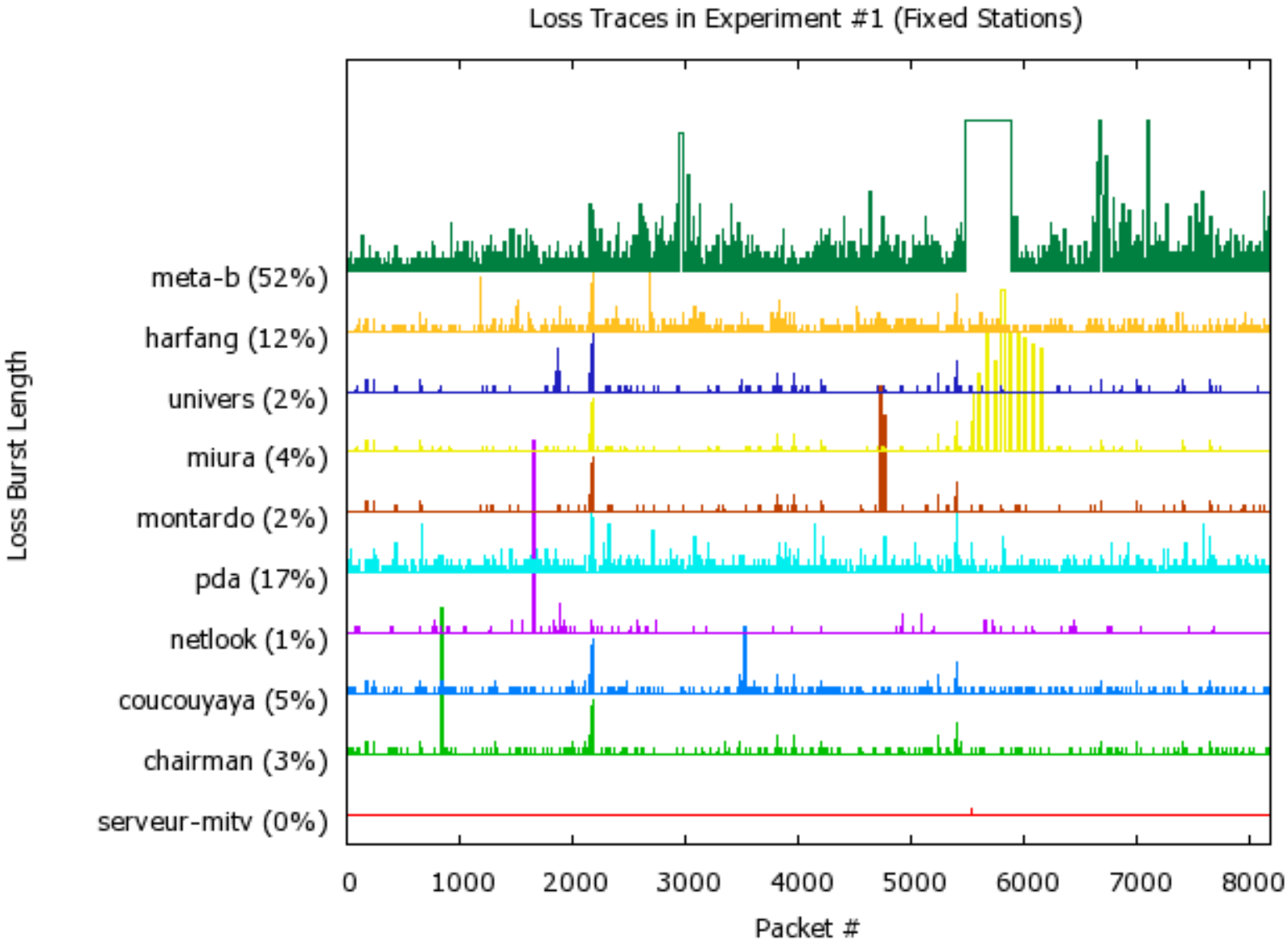
- **Exemple :**



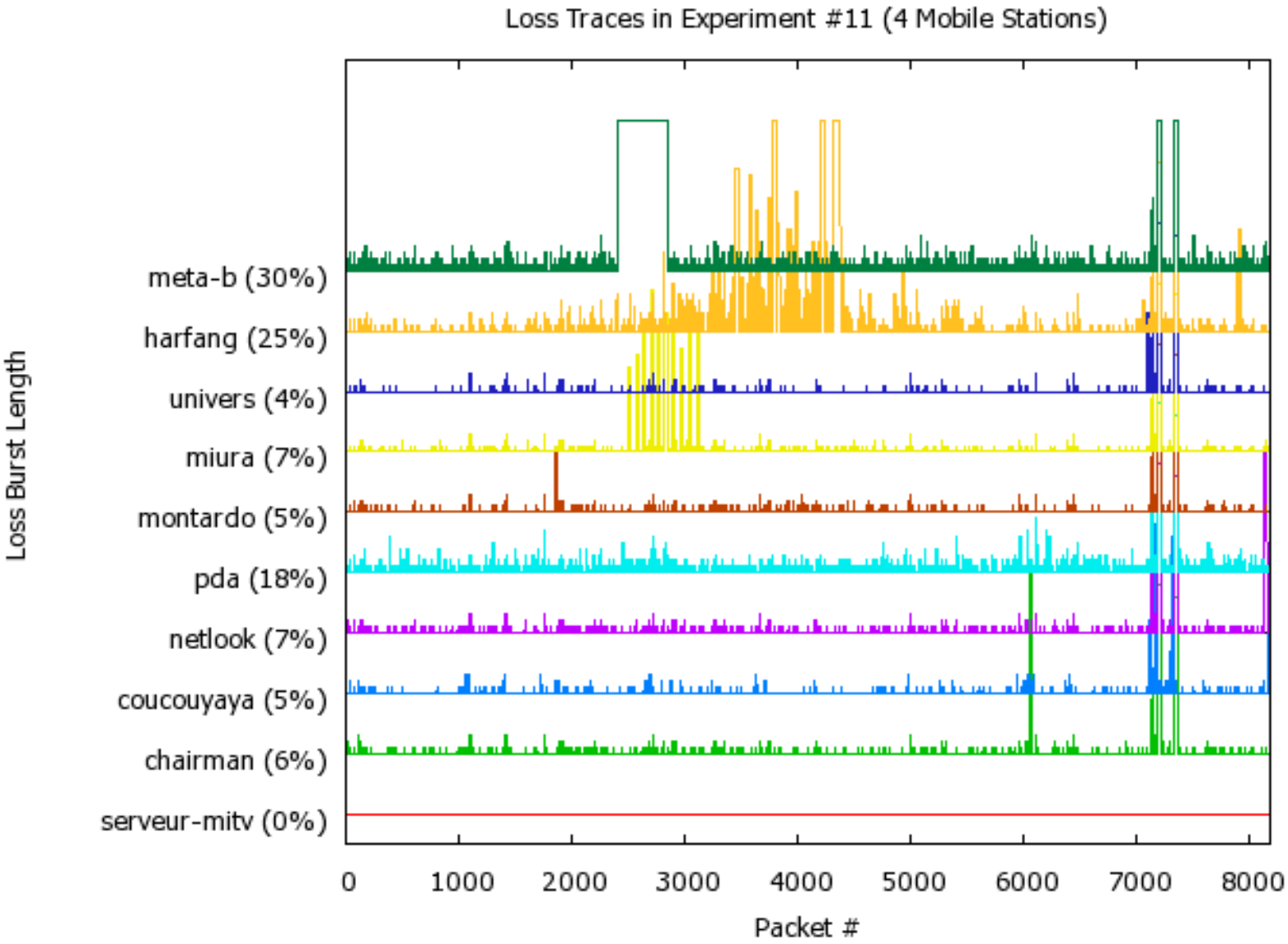
Recueil des traces

- **Architecture de tests:**
  - émission du flux video multipoint par un PC sur le WLAN du DMI/ENSICA
  - réception par 10 récepteurs :
    - **1ère partie des tests : tous les récepteurs sont fixes**
    - **2ème partie des tests : 40% des récepteurs sont mobiles**
- **caractéristiques de la transmission :**
  - flux vidéo MPEG/RTP multicast à 500 kb/s
  - 8196 paquets de taille variable
  - durée : environ 1 minute 30
- **Note : 802.11b utilise des mécanismes de retransmissions au niveau MAC pour les liaisons point-à-point, mais ils ne sont pas activés pour les liaisons multipoint.**

Observation des pertes (1)



Observation des pertes (2)



Premiers commentaires sur ces traces

- Taux de perte de paquets très variable suivant les récepteurs (de – de 1% à +de 20%).
- Pour chaque récepteur, les pertes sont groupées en burst (corrélation temporelle)
- Les pertes des différents récepteurs sont corrélées (corrélation "spatiale").

Mécanismes à évaluer

- **Mécanismes basés sur les retransmissions :**
  - ACK, NACK, SACK
  - NACK suppression
- **Mécanismes basés sur les codes**
  - type et paramètres des codes
- **Mécanismes hybrides : les retours des récepteurs sont utilisés pour ajuster au mieux le niveau de redondance.**

➔ **but : comparer des représentants de chaque famille**

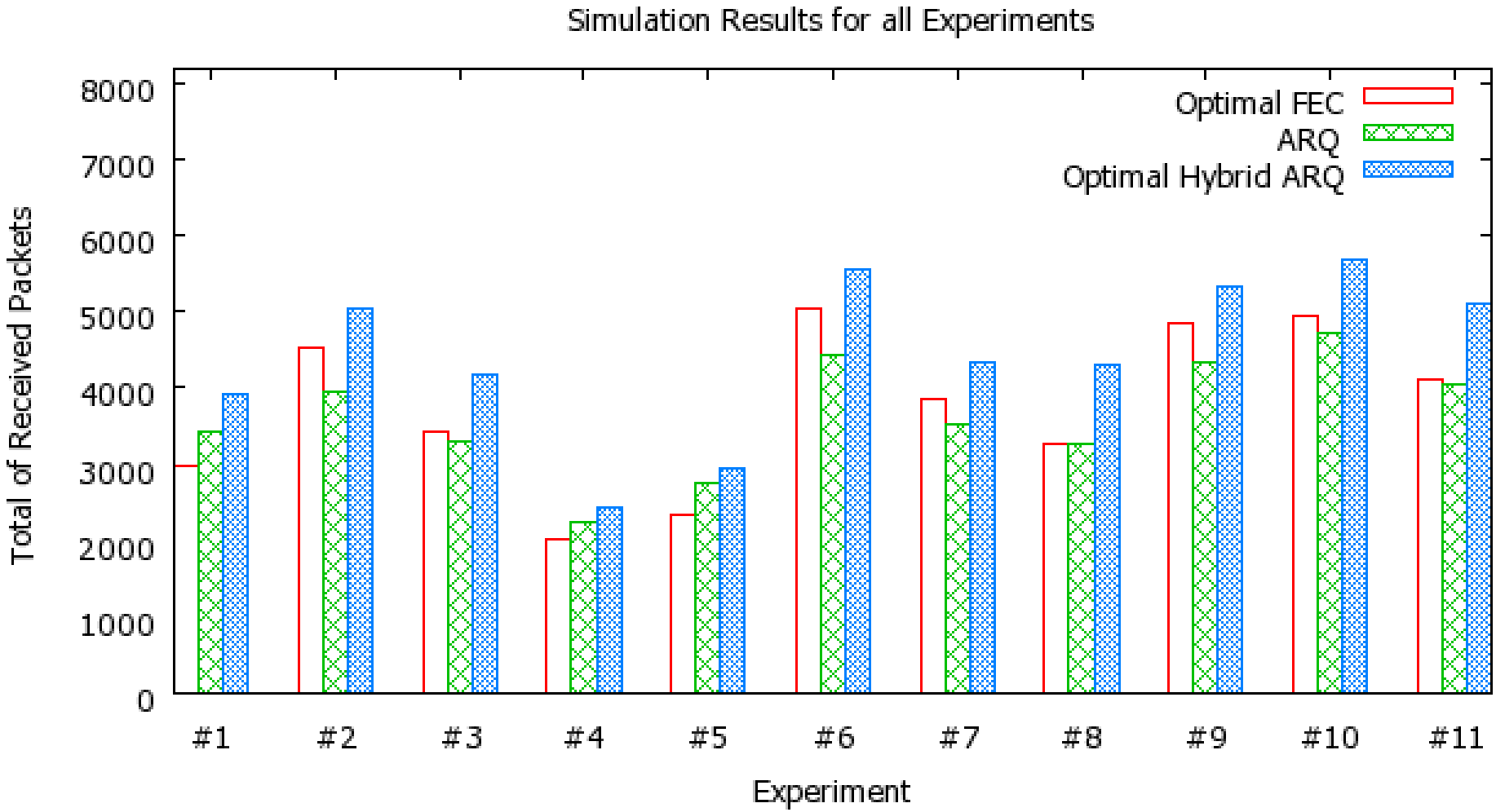
Mécanismes testés (1)

- **ARQ :**
  - chaque récepteur informe l'émetteur de la réception de chaque paquet (émission d'un ACK en unicast).
  - l'émetteur ré-emet le paquet tant que tous les récepteurs ne l'ont pas reçu au moins une fois.
  - on suppose que les retours ne sont pas perdus
  - à 500 kb/s, le timing (timeout du l'émetteur) est tout à fait réaliste.
- **FEC :**
  - l'émetteur utilise un k et un n fixé pour chaque bloc de paquets.
  - A la fin de l'émission de chaque bloc, les récepteurs indiquent si ils ont pu reconstituer les k paquets initiaux. Si au moins un des récepteurs n'a pu réussi cette opération le bloc de n paquets est de nouveau émis.

Mécanismes testés (2)

- **Hybride ARQ :**
  - l'émetteur envoie un bloc de k paquets (non codés)
  - tous les émetteurs indiquent **combien** de paquets ont été perdus
  - l'émetteur calcule le maximum *Max* de ces nombres et envoie *Max* nouveaux paquets de redondance.
  - cette opération est répétée jusqu'à ce que chaque récepteur ait reçu au moins k paquets.

Résultats


Simulation Results for all Experiments
Optimal FEC
ARQ
Optimal Hybrid ARQ
Total of Received Packets
8000
7000
6000
5000
4000
3000
2000
1000
0
#1
#2
#3
#4
#5
#6
#7
#8
#9
#10
#11
Experiment

Commentaires de ces résultats

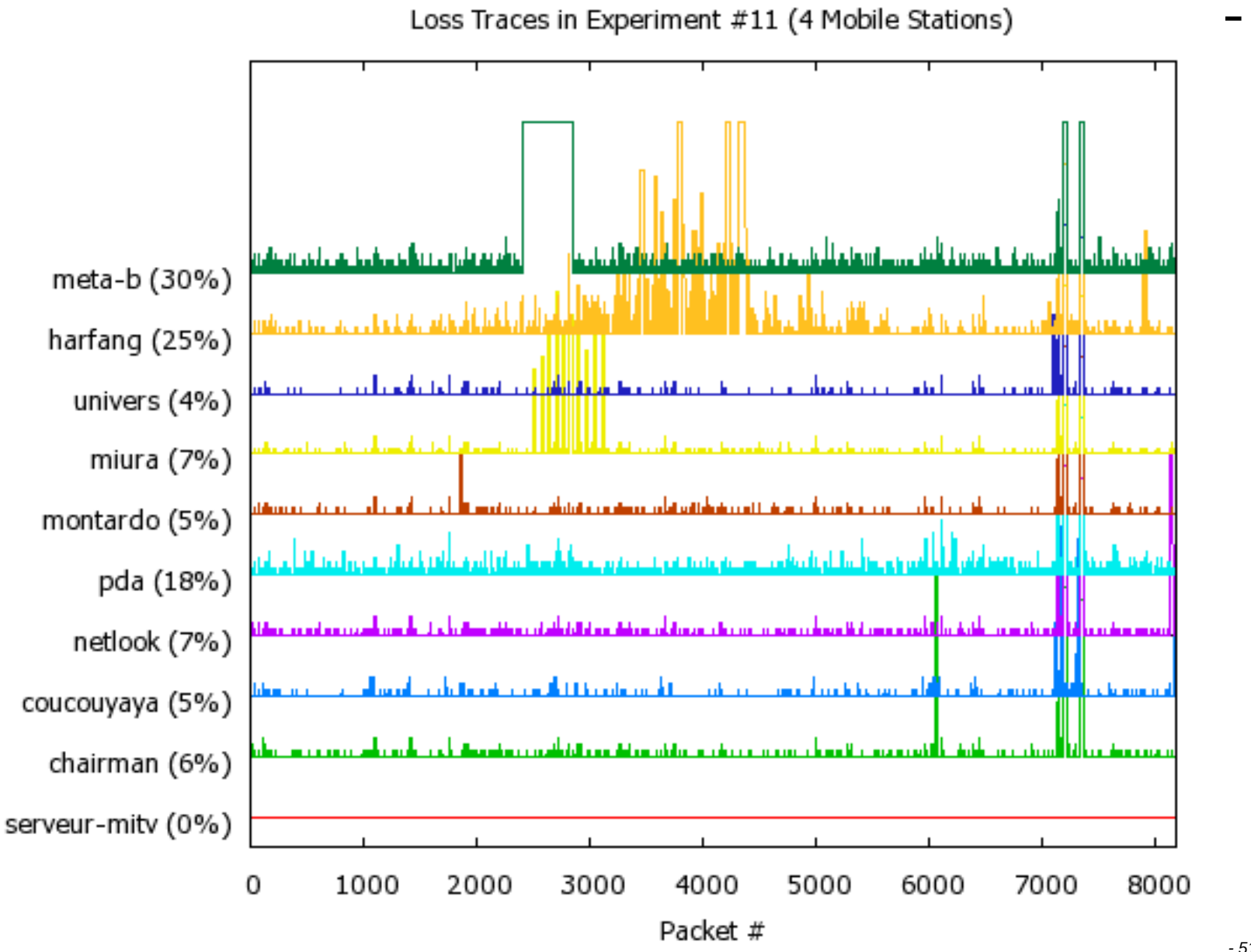
- **Hybride ARQ obtient toujours les meilleurs résultats**
- **Toutefois, les différences avec les autres mécanismes ne sont pas aussi importantes que dans d'autres travaux de référence.**
- **Ces travaux n'ont pas pris en compte les différentes corrélations et testé leur mécanismes avec des pertes uniformément réparties.**

➔ **Influence des différentes formes de corrélation de pertes sur les performances des mécanismes ?**

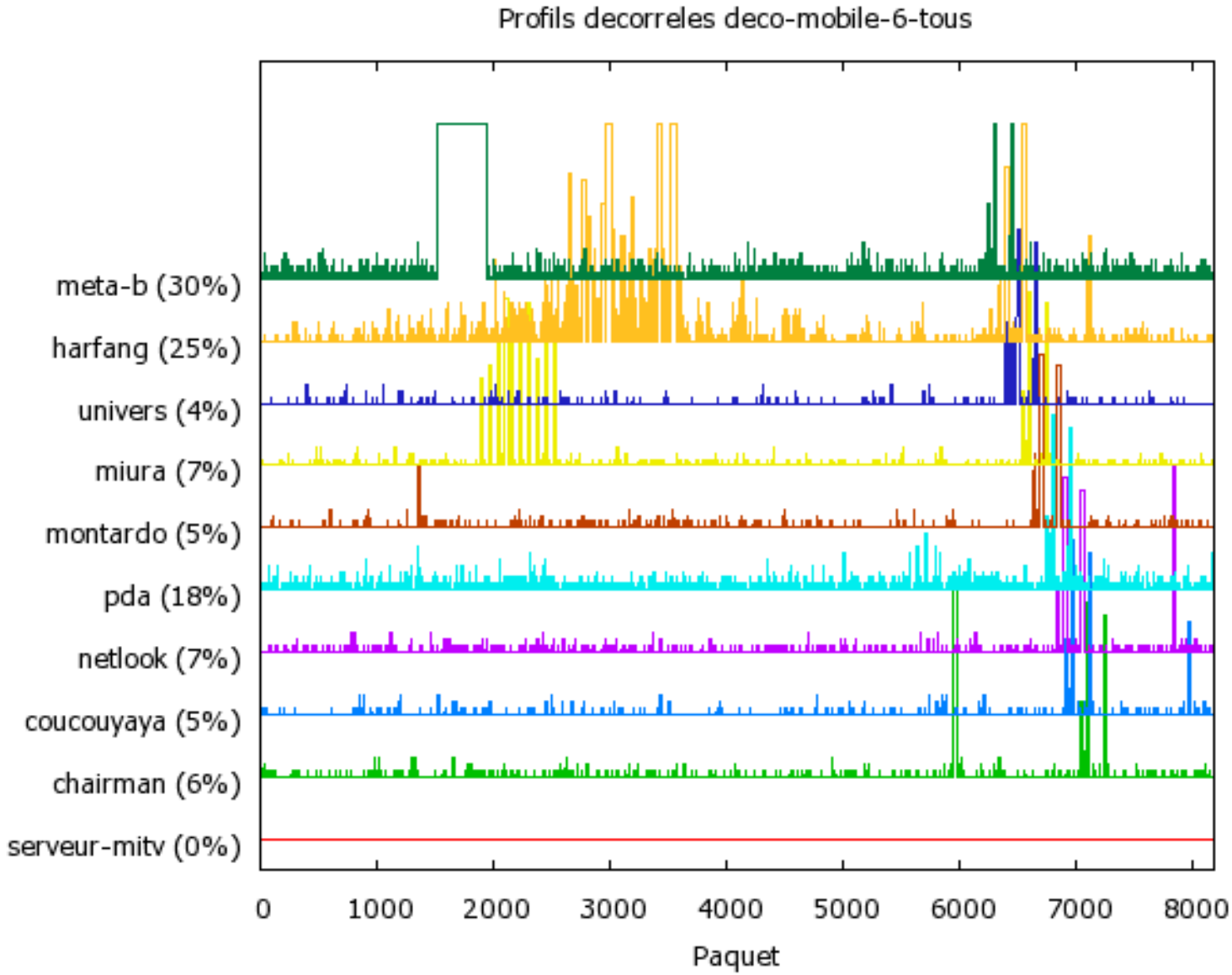
Génération de traces avec différents niveaux de corrélation

- **A partir de traces réelles, supprimer :**
  1. la corrélation spatiale : décalage cyclique des traces de chaque récepteur
  2. la corrélation temporelle : pour chaque récepteur, on génère des pertes uniformément réparties dont le taux correspond au taux de pertes observées.

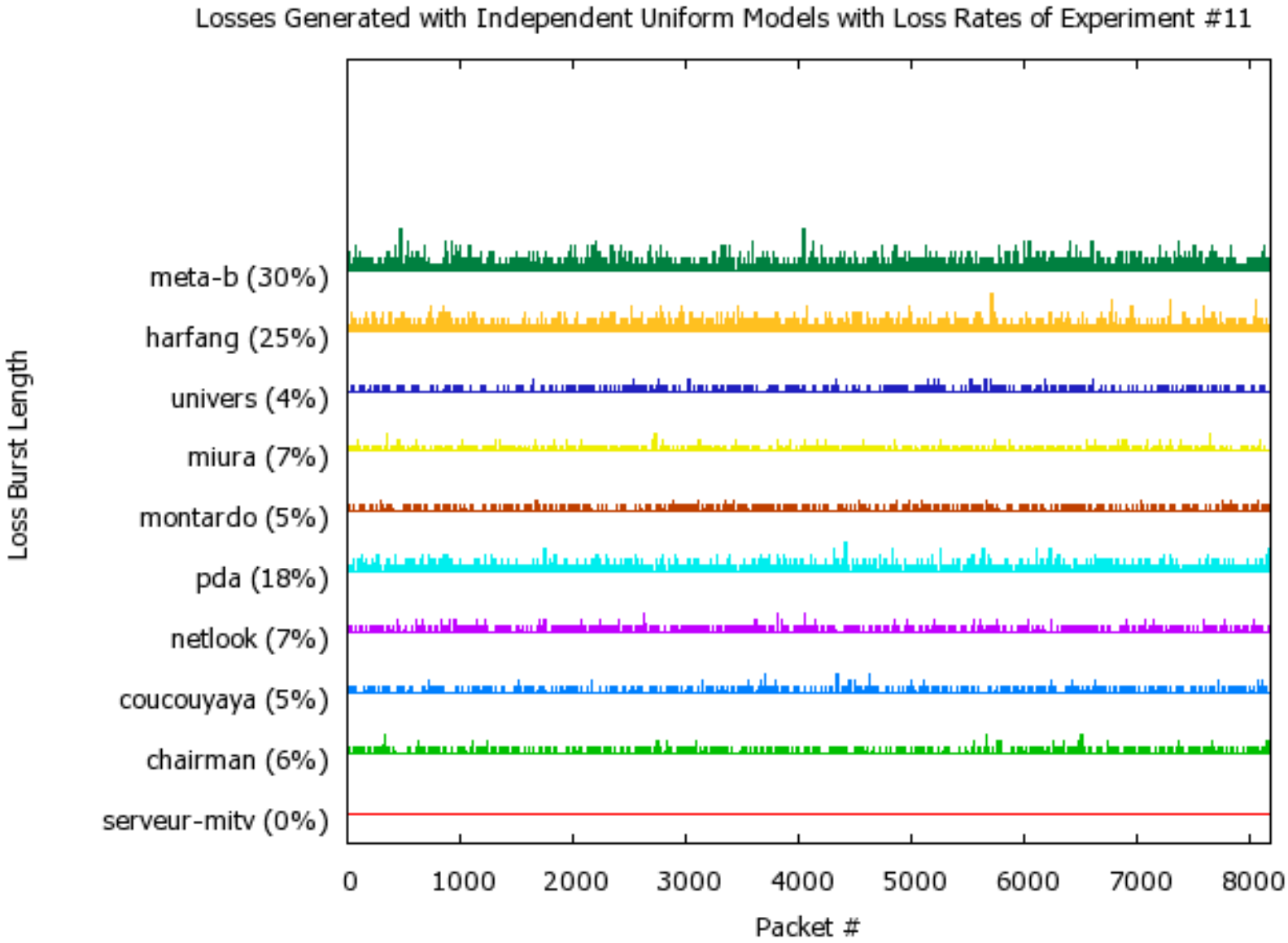
Trace initiale



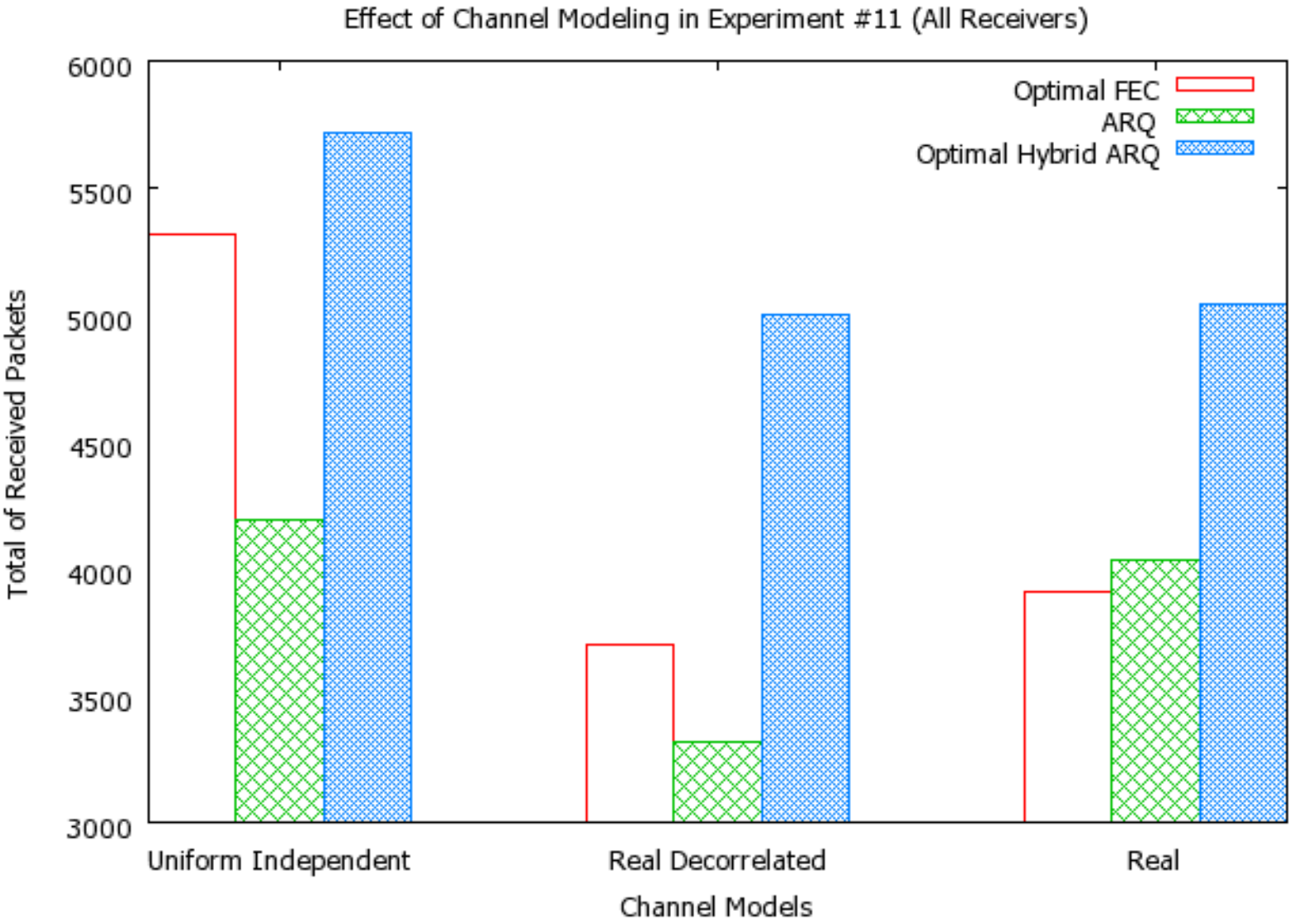
Sans corrélation spatiale



Sans aucune corrélation



Résultats


Effect of Channel Modeling in Experiment #11 (All Receivers)
Total of Received Packets
6000
5500
5000
4500
4000
3500
3000
Optimal FEC
ARQ
Optimal Hybrid ARQ
Uniform Independent
Real Decorrelated
Real
Channel Models

Conclusions et travaux futurs

- **Attention au modèle de perte !!**
- **Les mécanismes hybrides ARQ-FEC obtiennent les meilleurs résultats.**
- **Les différences avec les autres mécanismes ne sont pas énormes.**
- **Notre approche basée sur les traces réelles semble valide.**
- **Plusieurs points restent à étudier :**
  - ♦ quantité des retours (congestions possibles ?)
  - ♦ aller plus loin dans l'étude des mécanismes
  - ♦ prise en compte des délais pour les débits de transmission plus importants
  - ♦ modélisation du canal en termes de transmission de paquets