Analysis of Forward Error Correction in Packet Networks

Alain Jean-Marie

Eitan Altman, Omar Ait-Hellal, Parijat Dube, Yvan Calas, Tigist Alemu

ajm@lirmm.fr

INRIA/LIRMM CNRS University of Montpellier 2

Contents

Forward Error Correction at the Packet Level

- Definition
- Properties
- Examples

Computing the Efficient Throughput

- Bernoulli model
- Gilbert model
- Queuing models

Contents (ctd)

FEC and Queue Management Schemes

- Tail Drop, RED
- *A priori* Analysis
- Experiments
- *A posteriori* Analysis
- Model

Forward Error Correction at the Packet Level

Error correcting codes

Error detection/correction consists in adding redundancy bits to a message so that a certain number of transmission errors can be detected and/or corrected, up to a point.

Example: parity bits, CRC.

1 1 0 0 1 1 1 0 1 0 1 1

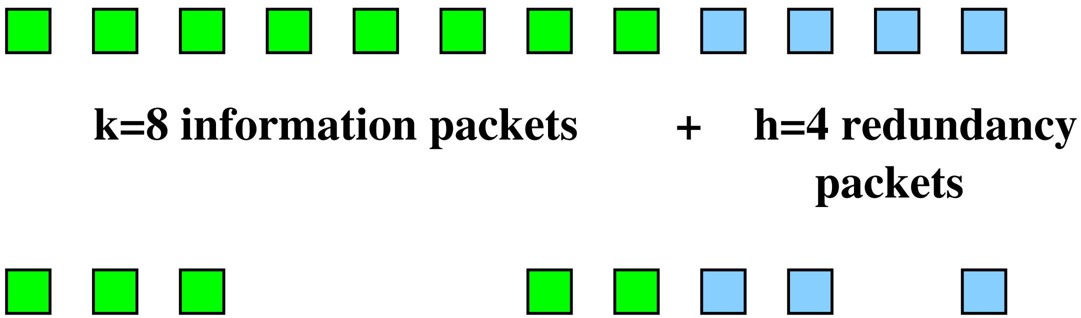
1 1 0 0 1 0 1 0 1 0 1 1 ✗

1 1 1 0 1 0 1 0 1 0 1 1 ✓

FEC at the Packet Level

When used at the packet level, there are no errors, only losses.

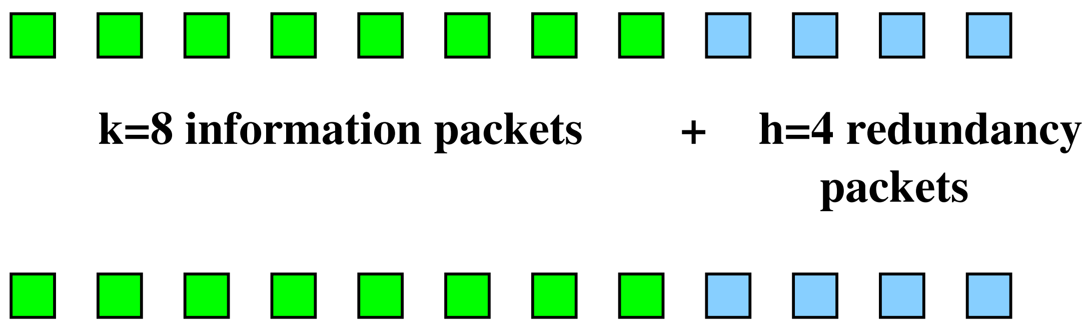
Reed-Solomon codes, among others, have the capacity to repair up to h lost packets, using h packets of redundancy.



FEC at the Packet Level

When used at the packet level, there are no errors, only losses.

Reed-Solomon codes, among others, have the capacity to repair up to h lost packets, using h packets of redundancy.



FEC Matrices

Iterative process for repairing matrices with line-and-column redundancy.

1	1	0	0
0	1	0	0
0	0	0	0
1	0	1	0

FEC Matrices

Iterative process for repairing matrices with line-and-column redundancy.

1	1	0	0
0	1	0	0
0	0	0	0
1	0	1	0

FEC Matrices

Iterative process for repairing matrices with line-and-column redundancy.

1	1	0	0
0	1	0	0
0	0	0	0
1	0	1	0

FEC Matrices

Iterative process for repairing matrices with line-and-column redundancy.

1	1	0	0
0	0	0	0
0	0	0	0
1	0	1	0

FEC Matrices

Iterative process for repairing matrices with line-and-column redundancy.

1	1	0	0
0	0	0	0
0	0	0	0
1	0	1	0

FEC Matrices

Iterative process for repairing matrices with line-and-column redundancy.

1	1	0	0
0	0	0	0
0	0	0	0
1	0	1	0

FEC Matrices

Iterative process for repairing matrices with line-and-column redundancy.

1	0	0	0
0	0	0	0
0	0	0	0
1	0	1	0

FEC Matrices

Iterative process for repairing matrices with line-and-column redundancy.

1	0	0	0
0	0	0	0
0	0	0	0
1	0	1	0

FEC Matrices

Iterative process for repairing matrices with line-and-column redundancy.

1	0	0	0
0	0	0	0
0	0	0	0
1	0	1	0

FEC Matrices

Iterative process for repairing matrices with line-and-column redundancy.

0	0	0	0
0	0	0	0
0	0	0	0
1	0	1	0

FEC Matrices

Iterative process for repairing matrices with line-and-column redundancy.

0	0	0	0
0	0	0	0
0	0	0	0
1	0	1	0

FEC Matrices

Iterative process for repairing matrices with line-and-column redundancy.

0	0	0	0
0	0	0	0
0	0	0	0
1	0	1	0

FEC Matrices

Iterative process for repairing matrices with line-and-column redundancy.

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

FEC Matrices

Iterative process for repairing matrices with line-and-column redundancy.

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Computing the Efficient Throughput

The Bernoulli model

Assumption: losses occur independently, with probability p:
Given a block of size k packets + h packets of redundancy, the probability to lose the whole block is:

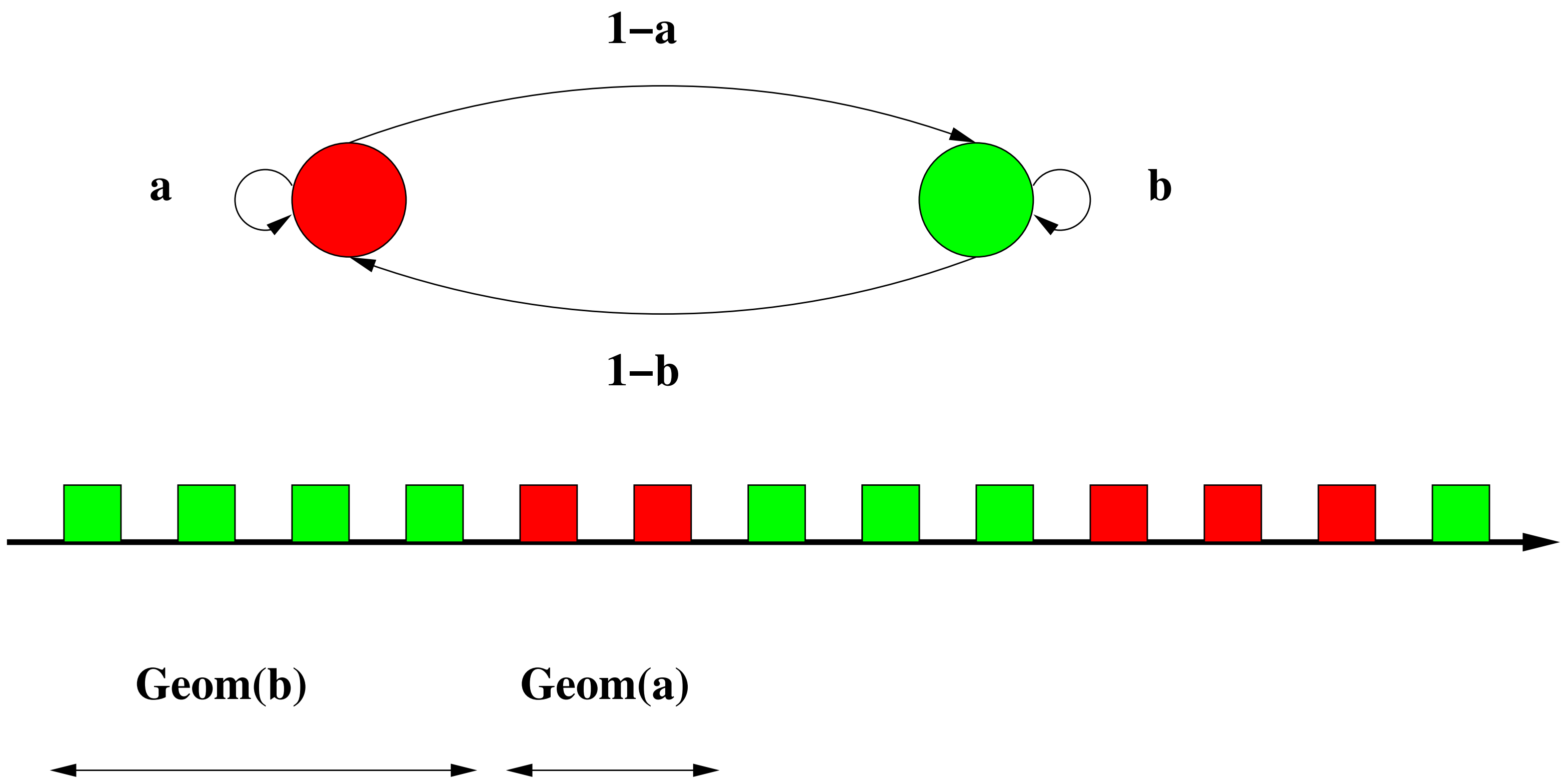
$$\pi_\ell = P(>h \text{ losses among } k+h \text{ packets}) = \sum_{\ell=h+1}^{h+k} \binom{k+h}{\ell} p^\ell (1-p)^{h+k-\ell}$$

Efficient throughput (goodput):

$$\lambda_{\text{eff}} = \lambda_{\text{in}} \times \frac{k}{k+h} \times (1-\pi_\ell)$$

The Gilbert model (1)

Assumption: losses occur according to the state of a (two-state) markov chain.



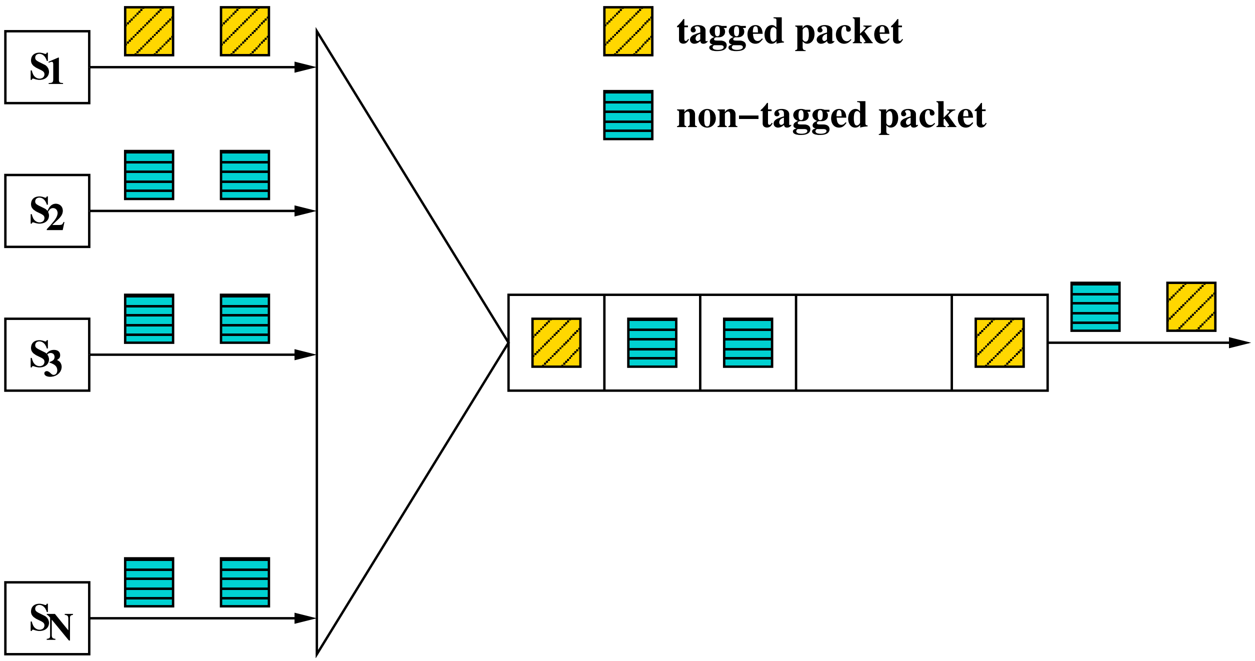
The Gilbert model (2)

Computation of probabilities: by recurrence

$$
\begin{aligned}
&P(\ h \text{ losses among } n \text{ packets} | X = \textcolor{red}{\bullet}) \\
&\quad = \quad a \times P(\ h-1 \text{ losses among } n-1 \text{ packets} | X = \textcolor{red}{\bullet}) \\
&\qquad\qquad +(1-a) \times P(\ h-1 \text{ losses among } n-1 \text{ packets} | X = \textcolor{green}{\bullet})
\end{aligned}
$$

$$
\begin{aligned}
&P(\ h \text{ losses among } n \text{ packets} | X = \textcolor{green}{\bullet}) \\
&\quad = \quad b \times P(\ h \text{ losses among } n-1 \text{ packets} | X = \textcolor{green}{\bullet}) \\
&\qquad\qquad +(1-b) \times P(\ h \text{ losses among } n-1 \text{ packets} | X = \textcolor{red}{\bullet})
\end{aligned}
$$

Queueing Model



- Markovian sources
- Computation by recurrences (Markov-modulated loss process)

Dimensioning Problem (1)

Given:

- a block size k
- an individual loss probability p for each packet
- a loss probability ε,

Find the smallest h such that:

$$\begin{aligned} & P(\text{ the message is lost }) \\ = \quad & P(> h \text{ losses among } k + h \text{ packets}) \\ < \quad & \varepsilon \, . \end{aligned}$$

Dimensioning Problem (2)

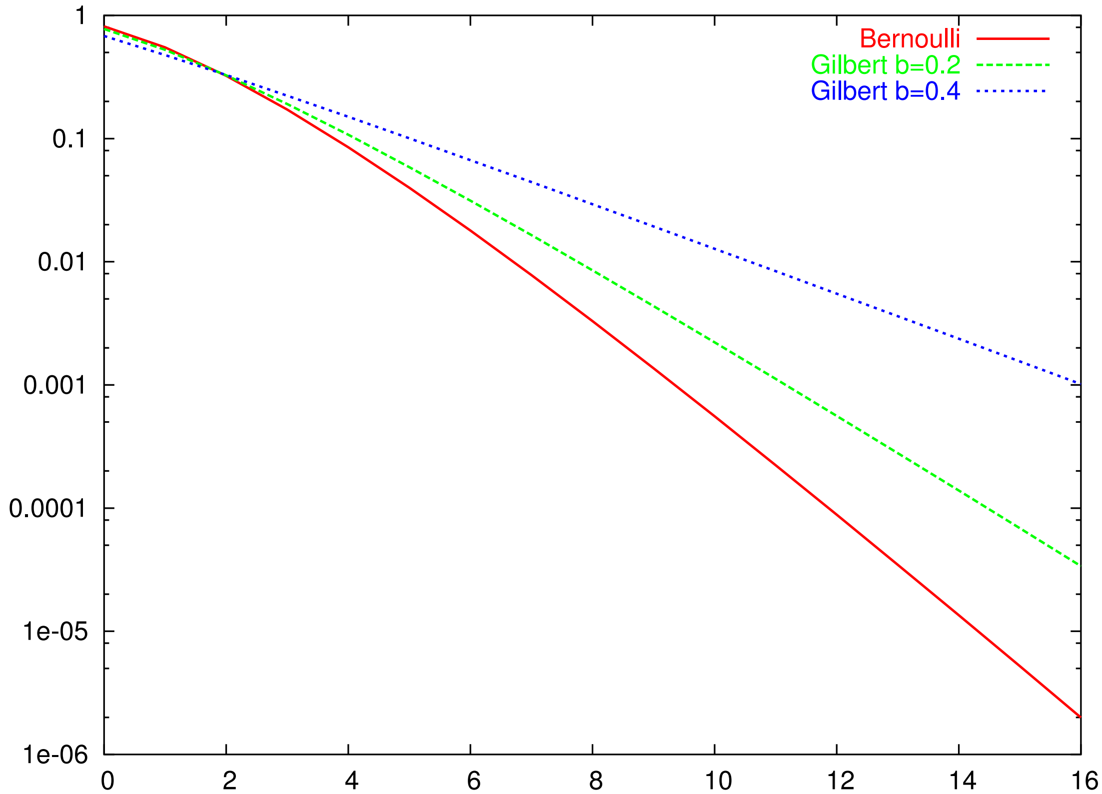
Two variants:

- the throughput of packets does not change, p is constant
- the throughput of information does not change, p increases.

General conclusions:

- Sometimes, it is not advantageous to add redundancy
- The value of h is larger for the models with bursts than with the Bernoulli model.

Comparison Bernoulli/Gilbert



Loss probability of a block of size $k = 16$, depending on h.

FEC and Queue Management Schemes

Queue Management (1)

Packets arrive to the buffer of a router. Is the packet enqueued? It depends on the Queue Management scheme.

Tail Drop

- if the buffer is full, the incoming packet is dropped
- if not, the packet is enqueued.

It is a “passive” queue management.

Queue Management (2)

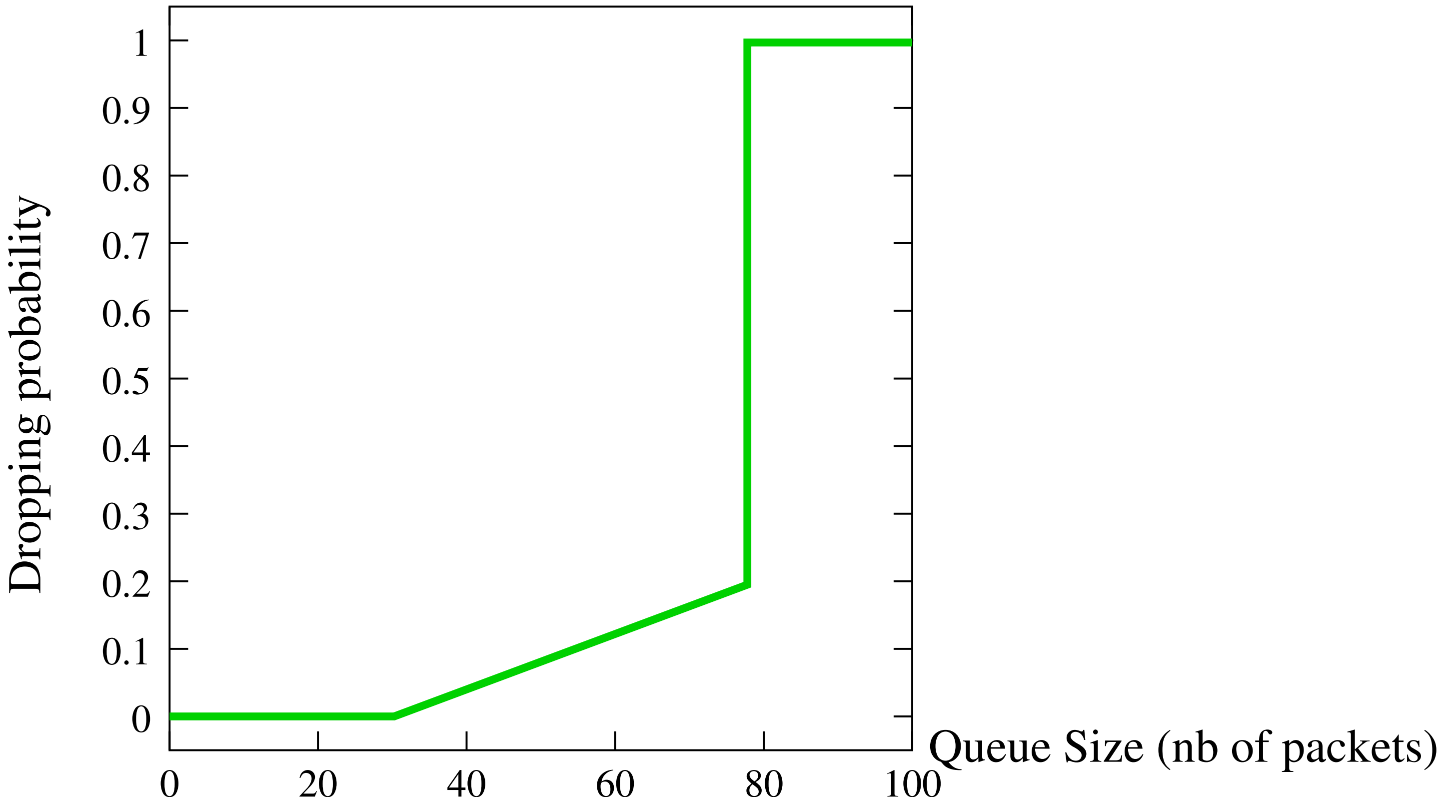
RED: Random Early Detection

- when the packet arrives, the average queue length is $\hat{L}$,
- if the buffer is full, the packet is dropped,
- if not, the packet is dropped with probability $d(\hat{L})$,
- otherwise, it is enqueued.
- the average queue length is updated:

$$\hat{L} \leftarrow (1-\omega)\hat{L} + \omega L$$

It is an Active Queue Management scheme.

Queue Management (3)



Typical dropping function $d(\hat{L})$ for RED.

Preliminary Analysis

The dropping process of TD and RED is known to have the following characteristics:

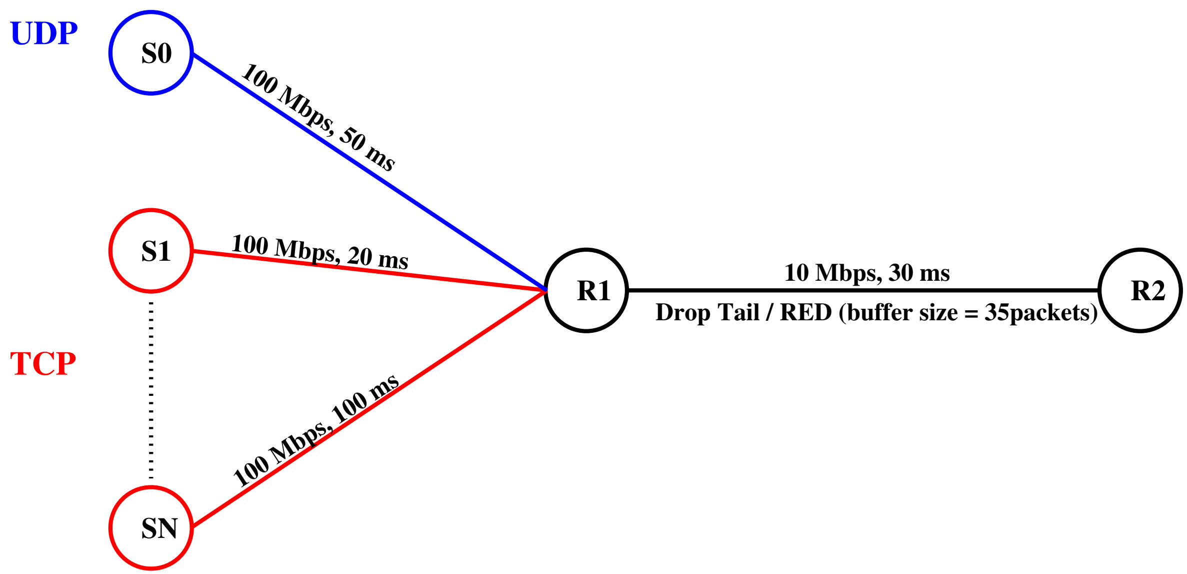
- TD drops packets more in bursts
- RED drops packets more randomly
- the loss rate of RED is larger than that of TD.

The fact that RED spreads losses randomly should favor RED. But the increase of loss probability should be moderate.

Experimental setup

Simulations with the `ns-2` program.

- Source of packets with the UDP protocol, 5-10% of the BW
- Background traffic of TCP flows, saturating the BW.



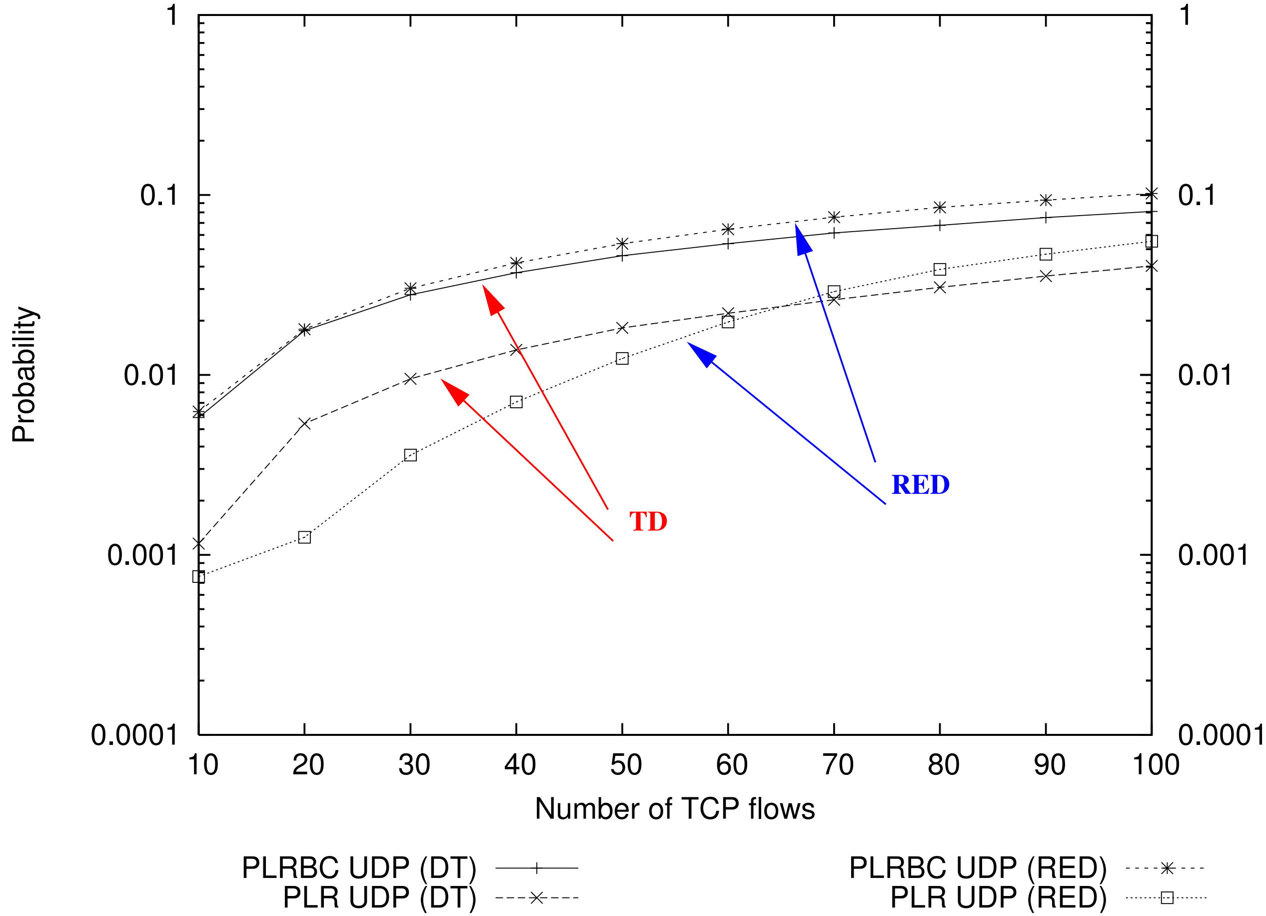
Measurements

Statistics collected about:

- agregate throughput,
- queueing delay,
- loss rate before correction
- loss rate after correction
- loss run length

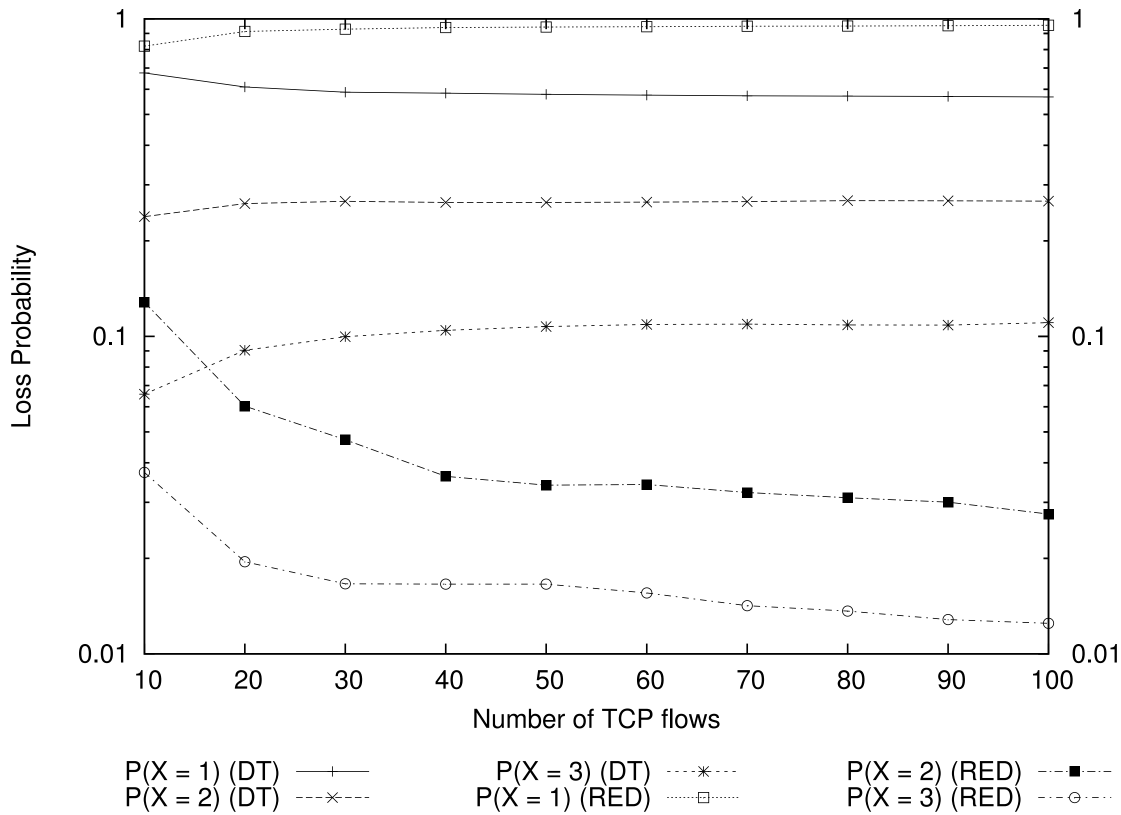
Results (1)

Loss rates, $k = 16$ packets per block + $h = 2$ FEC packets.



Results (2)

Loss Run Length:
$k = 16$ packets per block + $h = 1$ FEC packets.



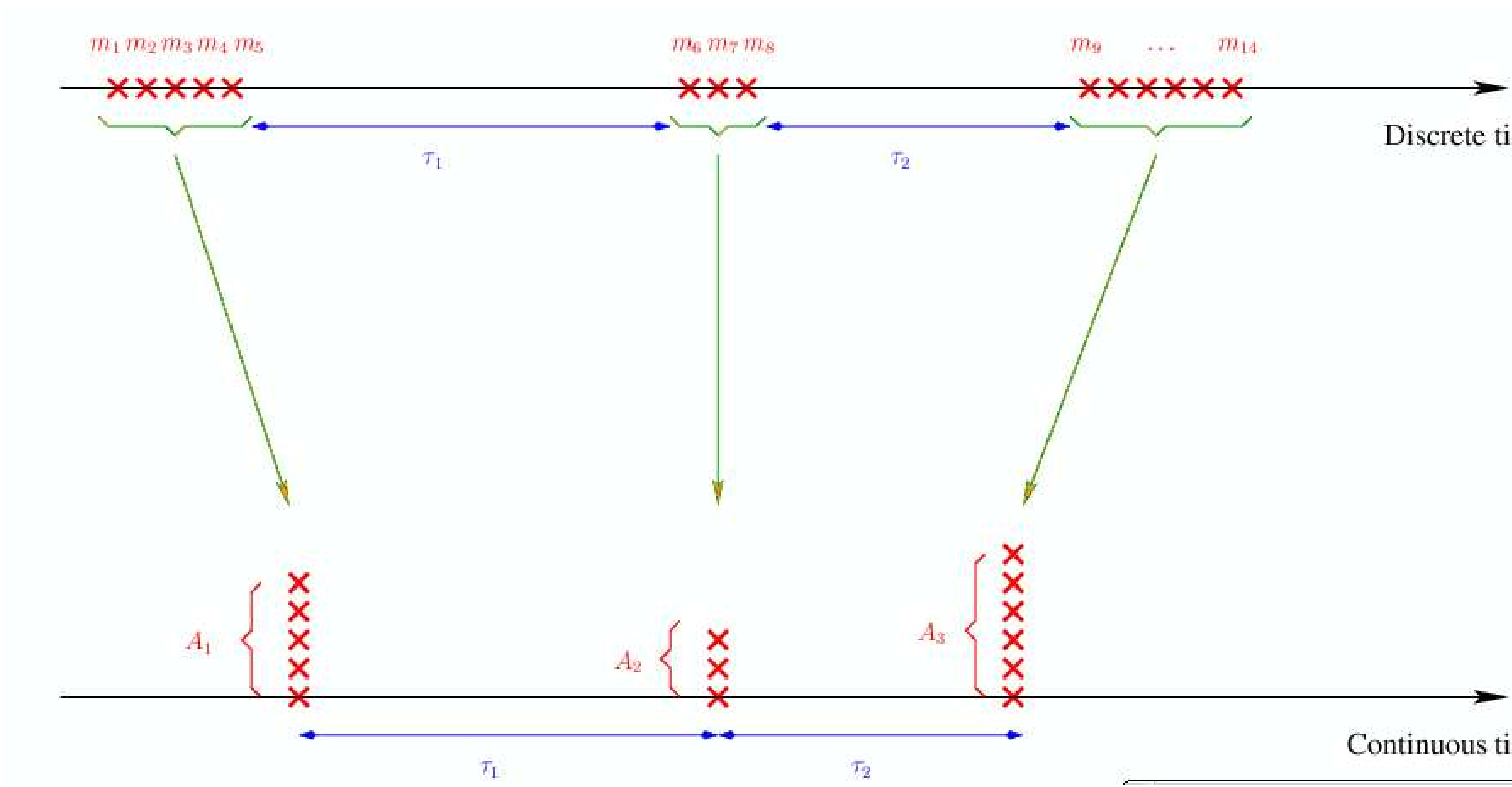
Analysis a posteriori

- Statistics on the loss run length confirm that losses of RED are mostly isolated.

#	RED	TD
1	95%	60%
2	3%	20%
3+	2%	20%

- Losses under RED are marginally superior to that of TD
- Nevertheless, RED is not always superior to TD.

A model (1)



m_1 m_2 m_3 m_4 m_5
m_6 m_7 m_8
m_9 ... m_{14}
τ_1
τ_2
Discrete ti
A_1
A_2
A_3
τ_1
τ_2
Continuous ti

A model (2)

Process of loss:

- groups of losses occur according to a Poisson process with rate λ,
- groups have random sizes with identical distribution and mean a.

Global loss rate: $p = \lambda \times a$

Distribution of the number of losses:

$$\sum_k z^k P(k \text{ losses in } [0,t)) \;=\; e^{\lambda(A(z)-1)} \;.$$

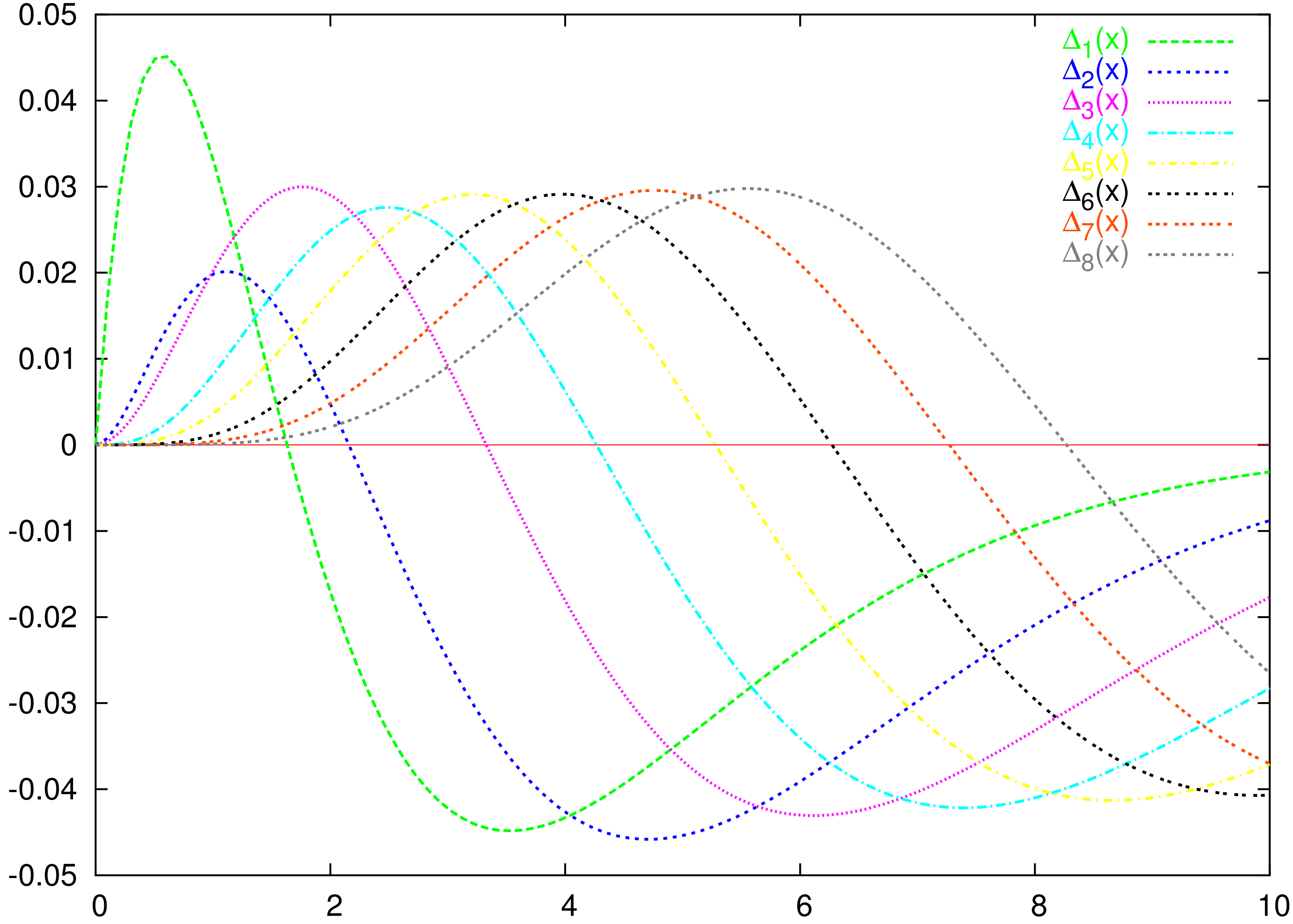
Comparison (1)

Comparison of two cases:

- Case “RED”: losses of
  1 with proba 0.9,
  2 with proba 0.1
- Case “Tail Drop”: losses of
  1 with proba 0.6,
  2 with proba 0.4
- Same average packet loss number $x = p \times (h + k)$

$$\begin{aligned}\Delta_h(x) &= P(\text{ message saved in case “RED” with } h \text{ FEC}) \\ &\quad - P(\text{ message saved in case “TD” with } h \text{ FEC})\end{aligned}$$

Comparison (2)



Comparison (3)

Empirical evidence (+ Analysis!) shows: RED is better if:

$$x \leq h + C$$

for some constant C.
Equivalently, RED better if:

$$\begin{aligned} k &\leq \frac{1-p}{p}h + \frac{C}{p} \\ \frac{h}{k} &\geq \frac{p}{1-p} - \frac{C}{1-p}\frac{1}{k} \\ p &\leq \frac{h+C}{h+k}. \end{aligned}$$