

## Contrôle de congestion

Cours 3

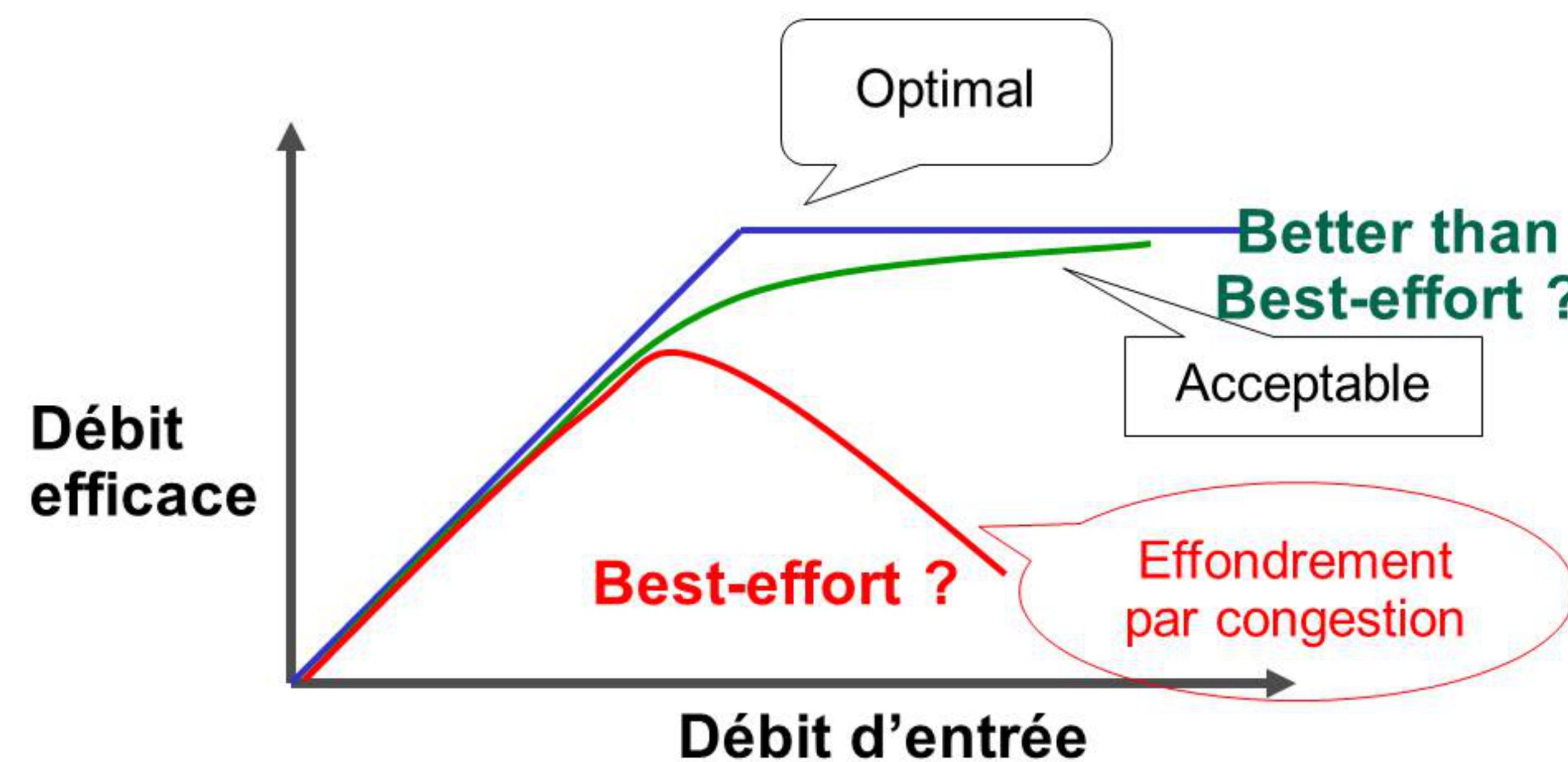
# CONTRÔLE DE CONGESTION ET GESTION DE BUFFER

- Contrôle en boucle fermée
- Contrôle de bout-en-bout
  - Au niveau transport souvent assisté par le réseau
- Services :
  - Best-effort et Better than best-effort
  - Pas de garantie stricte
- Qualité de service fournie :
  - Augmenter la vitesse de transfert et réduire les pertes de paquets
  - Augmenter l'utilisation du réseau

1

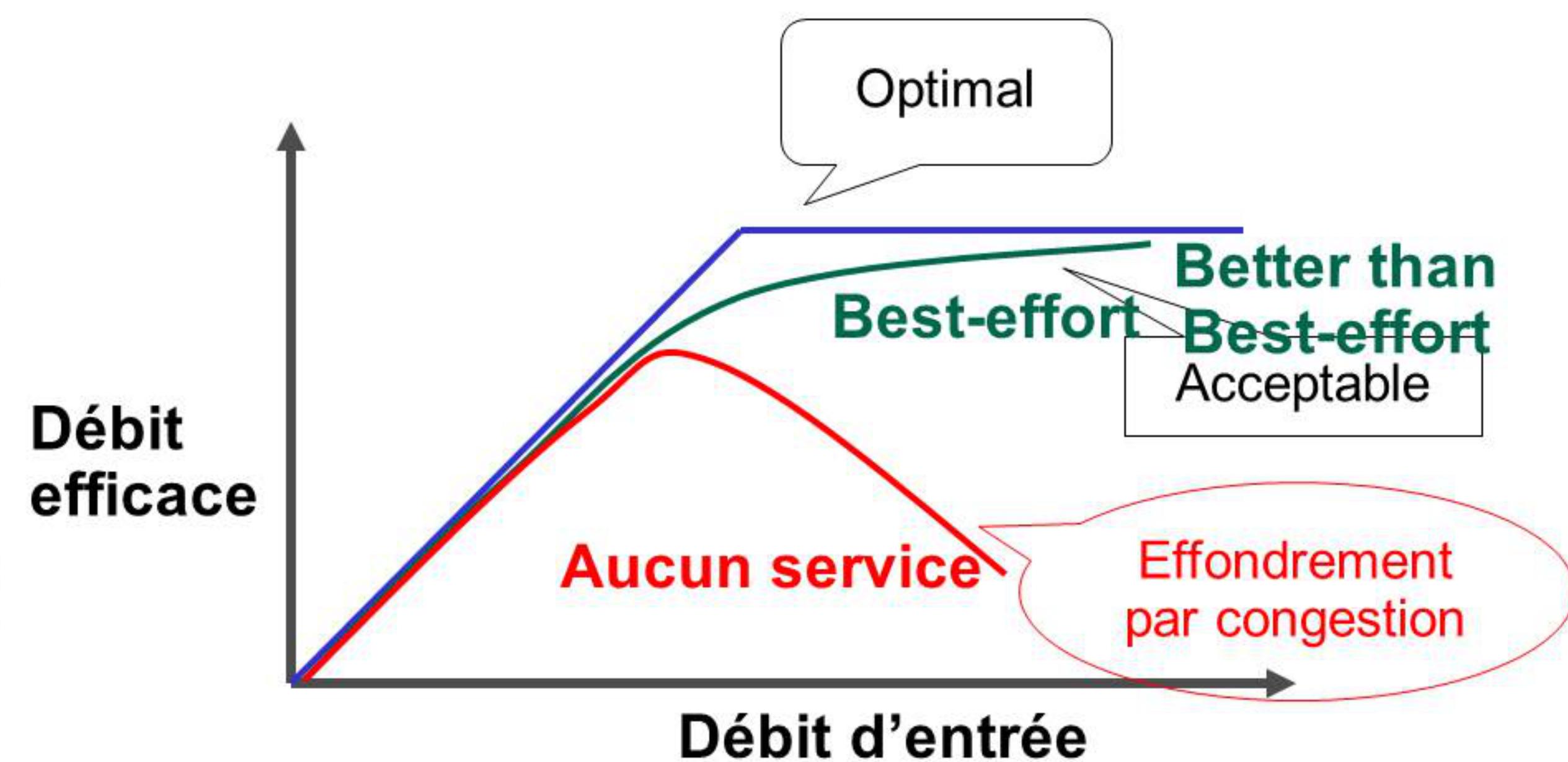
2

Congestion



1

Congestion



3

4

# Epistémologie

- Eviter les congestions permanentes
  - Faire en sorte que les congestions soient toujours transitoires (courte durée)



5

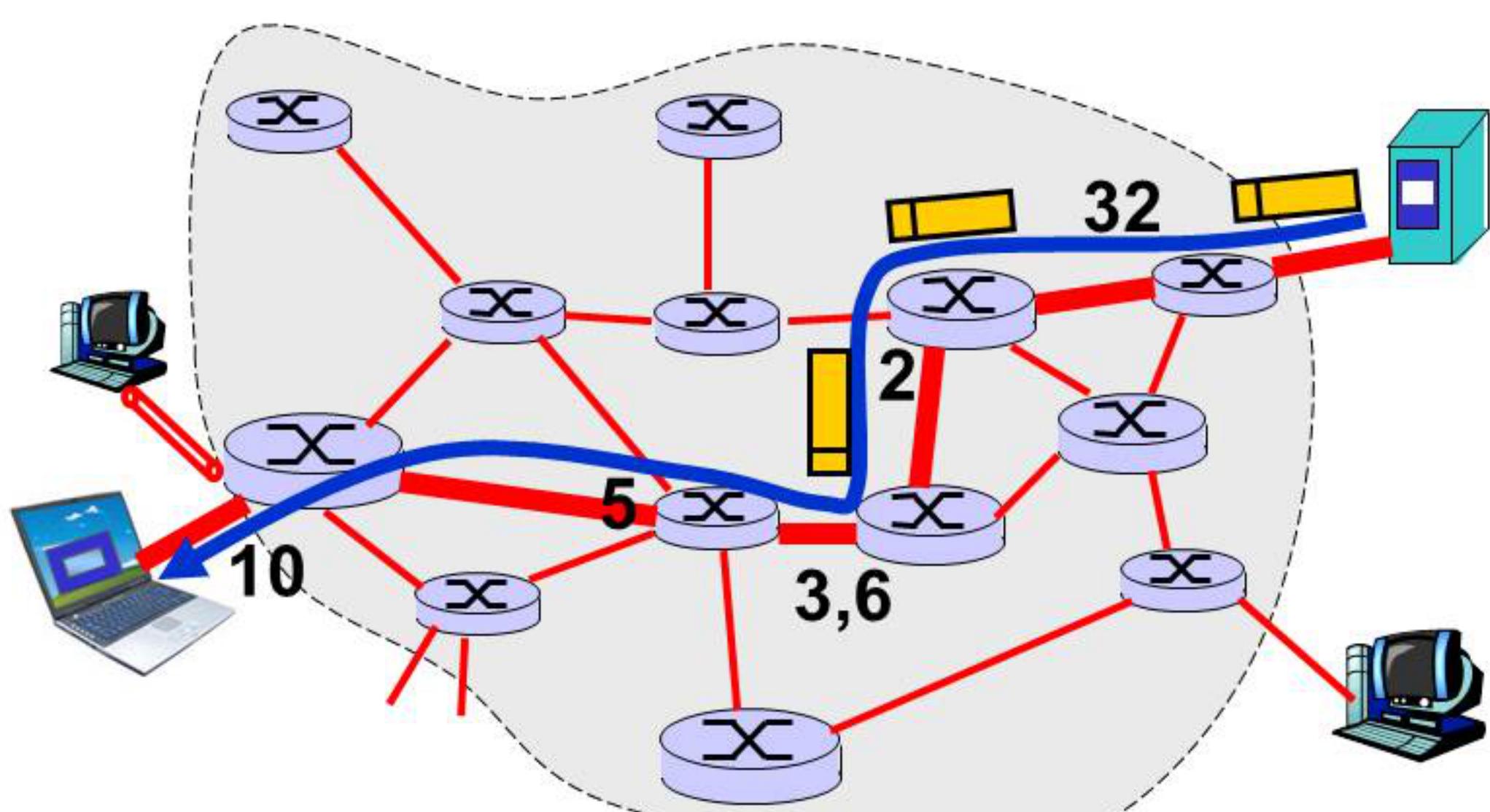
# Epistémologie

- Eviter les congestions permanentes
  - Faire en sorte que les congestions soient toujours transitoires (courte durée)

Transmettre au débit du goulot d'étranglement (« Bottleneck »)

Partager équitablement les liens traversés du réseau

6



Bande passante résiduelle minimale = 2 Mbit/s = Débit optimal de transmission

## Approches du contrôle de congestion

Deux approches principales :

### Contrôle de congestion de bout-en-bout :

- Pas de **feedback** du réseau (ou presque)
- La congestion est estimée grâce à l'observation des pertes et des délais de bout en bout.
- Approche suivie par le TCP de base

### Contrôle de congestion assisté par le réseau:

- Les routeurs fournissent des informations de retour (**feedback**) aux émetteurs
  - informations implicites
    - rejet
  - informations explicites
    - Un bit d'annonce de congestion
    - Débit d'émission explicite

8

# Contrôle de congestion dans TCP

## Contrôle de congestion de bout-en-bout

### Rappel TCP

- utilise une fenêtre pour contrôler le nombre de paquets que l'émetteur a le droit de « laisser » dans le réseau :
  - fenêtre de congestion : cwnd
- la fenêtre de transmission est calculée comme étant le minimum entre la fenêtre de congestion et la fenêtre de contrôle de flux (fenêtre conseillée par le récepteur)
  - $win = \min(cwnd, rwnd)$
- normalement, TCP envoie  $win$  paquets (segments) chaque RTT

9

10

## Contrôle de congestion dans TCP

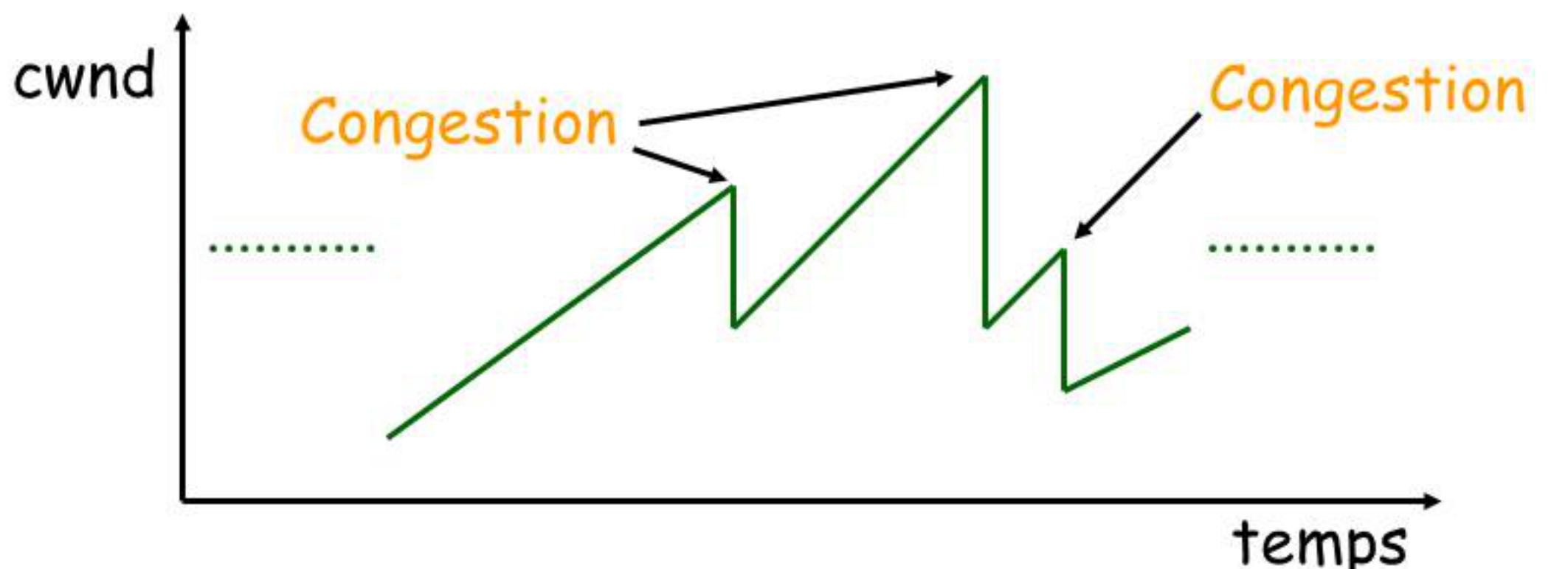
AIMD

### Principe

- Le débit de TCP est directement lié à la taille de la fenêtre cwnd
- Comment contrôler cwnd ?
  - Comment fixer les valeurs de cwnd ?
- TCP applique la méthode **AIMD**
  - Additive Increase Multiplicative Decrease

### Principe :

- AIMD change la fenêtre cwnd pendant la transmission pour s'adapter à l'état du réseau :
  - cwnd **augmente linéairement** si pas de congestion
  - cwnd **diminue multiplicativement** en cas de congestion



11

12

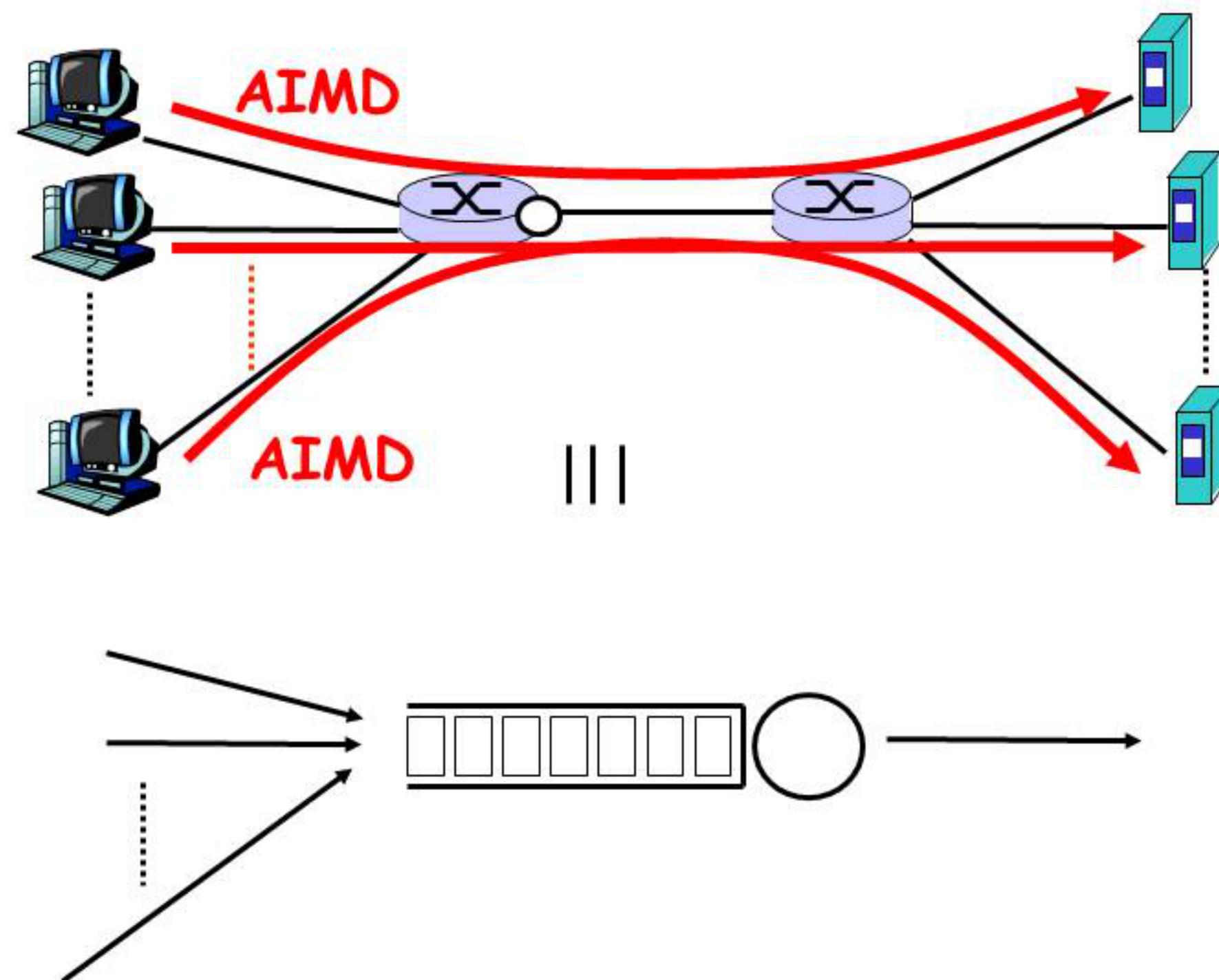
# AIMD dans TCP

- Algorithme TCP : Slow Start, **Congestion Avoidance**, Fast Retransmit, Fast Recovery

- $cwnd = cwnd + (1 / cwnd)$  pour chaque ACK reçu
- c'est à dire **+ 1** segment à chaque RTT
- jusqu'au prochain dépassement du seuil maximal ... ou la prochaine congestion
- si congestion alors :  $ssthresh = cwnd / 2$   
Après la retransmission de(s) segment(s) perdu(s) :  
 $cwnd \leftarrow ssthresh$   
et l'algorithme se répète

13

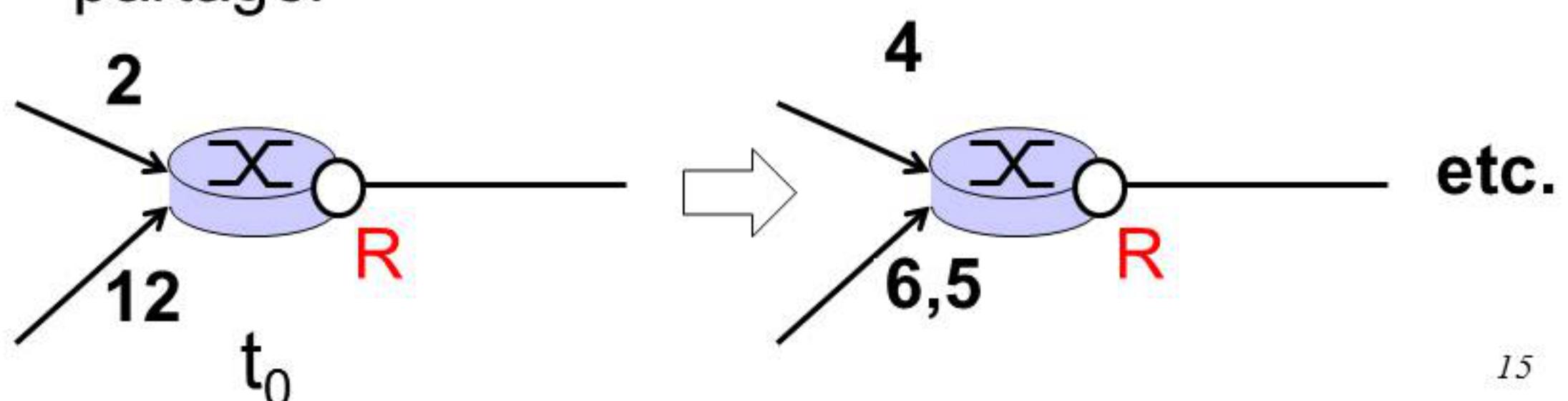
## AIMD : Propriétés



14

## AIMD : Propriétés

- Modèle AIMD simplifié :
- Deux connexions :
  - A  $t_0$  :  $D_1 = 2$ ,  $D_2 = 12$
  - Tant que  $D_1 + D_2 \leq 20$ , les  $D$  augmentent de 1
  - Quand  $D_1 + D_2 > 20$ , les  $D$  sont divisés par 2
  - $R = 20$  = capacité maximale de la ressource à partager

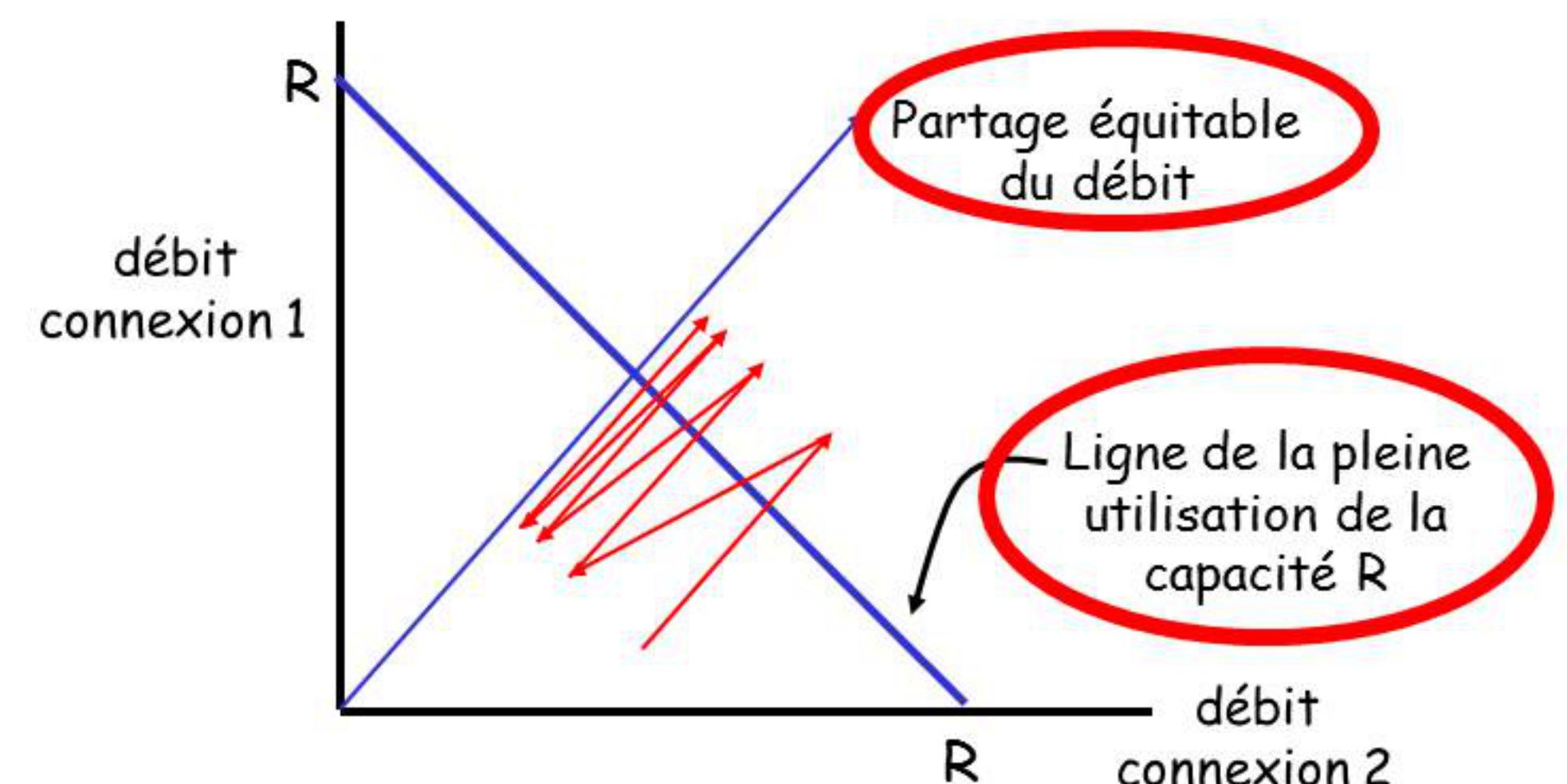


15

## AIMD : Propriétés

Deux connexions en parallèle :

- L'incrémentation additive génère une pente de 1.
- La réduction multiplicative réduit le débit proportionnellement



16

## AIMD : Propriétés

- Le routeur rejette des paquets quand il est saturé, donc en cas de congestion
- Les connexions AIMD réagissent en diminuant leur débit
  - → Diminue la différence de débit entre les connexions
  - → **Convergence vers l'équité**
- Les connexions AIMD augmentent leur débit si pas de congestion
  - → **Convergence vers la pleine utilisation**

17

## AIMD : Propriétés

- Propriétés négatives ?
  - Oscillation de débit
    - Risque d'instabilité
    - Périodes de sous-utilisation et sur-utilisation

18

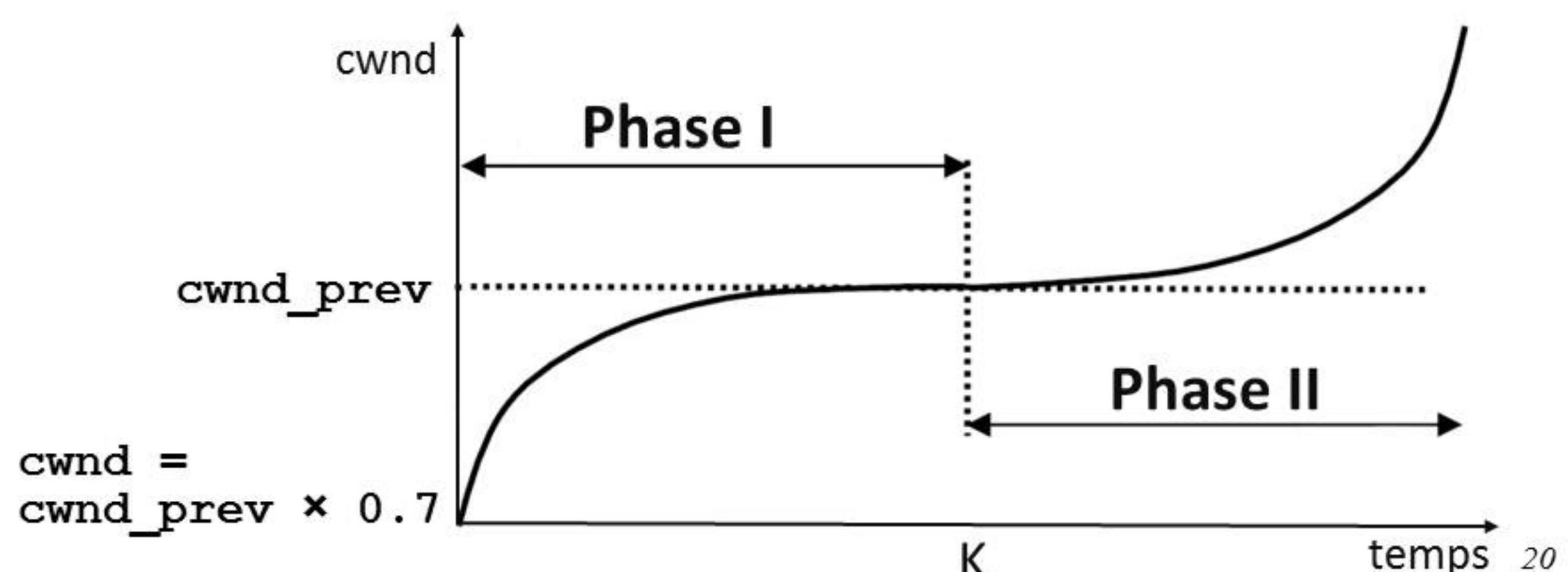
## AIMD ou AIMD

- Au lieu de **Additive IMD**
  - **Any IMD (Autre IMD) ?**
  - Si l'augmentation n'est pas linéaire mais identique, alors les propriétés d'équité et de convergence vers la pleine utilisation devraient être conservé

19

## TCP CUBIC

- **Principe :**
  - CubicIMD change la fenêtre cwnd pendant la transmission pour s'adapter à l'état du réseau :
    - cwnd **augmente via une fonction cubique** si pas de congestion
    - cwnd **diminue multiplicativement** en cas de congestion
      - Le facteur de diminution  $\beta = 0,7$  au lieu de 0,5



20

## TCP CUBIC

- Principe :

- CubicIMD change la fenêtre cwnd pendant la transmission pour s'adapter à l'état du réseau :
  - cwnd **augmente via une fonction cubique si pas de congestion**
  - cwnd **diminue multiplicativement** en cas de congestion
    - Le facteur de diminution  $\beta$

$$cwnd(t) = cwnd_{prev} + C \times (t - K)^3$$

$$K = \sqrt[3]{\frac{cwnd_{prev} \times (1 - \beta)}{C}}$$

- Phase I :  $t < K$  : augmentation rapide puis “observation”
- Phase II :  $t > K$  : “observation” puis augmentation rapide

21

## TCP CUBIC

- Principe :

- CubicIMD change la fenêtre cwnd pendant la transmission pour s'adapter à l'état du réseau :
  - cwnd **augmente via une fonction cubique si pas de congestion**
  - cwnd **diminue multiplicativement** en cas de congestion
    - Le facteur de diminution  $\beta$

### À la réception d'un ack :

$$cwnd = cwnd + (cwnd(t+rtt) - cwnd)/cwnd$$

- Phase I :  $t < K$  : augmentation rapide puis “observation”
- Phase II :  $t > K$  : “observation” puis augmentation rapide

22

## AIMD

- AIMD est appliquée dans d'autres protocoles/architectures
- Exemples
  - TCP-LP
    - TCP Low Priority
    - Background file transfer, file backup, software update
  - ABR d'ATM
  - QCN de l'IEEE 802.1
  - ...
- Mais elle n'est pas la seule technique de contrôle de congestion e2e

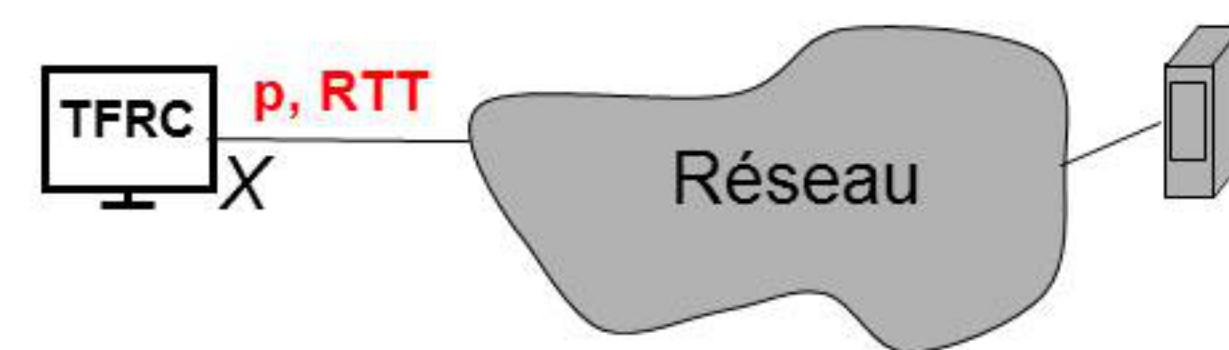
## Approches du contrôle de congestion

- Autre classification :
  - AIMD-based congestion control
  - Equation-based congestion control
  - Measurement-based congestion control

# Equation-based congestion control

- Utilisation d'équations reliant les paramètres décrivant l'état du réseau et les variables de l'application
- Exemple :
  - TFRC : TCP-Friendly Rate Control
  - Objectifs :
    - partager le débit équitablement (comme TCP)
    - déetecter le goulot d'étranglement (comme TCP)
    - réaliser le même débit qu'une connexion TCP dans les mêmes conditions.
    - éviter les oscillations de débit

- Solution
  - estimer le débit  $X$  d'une connexion TCP dans les mêmes conditions
  - limiter le débit à  $X$



The throughput equation is:

$$X = \frac{s}{R * \sqrt{2 * b * p / 3} + (t_{RTO} * (3 * \sqrt{3 * b * p / 8} * p * (1 + 32 * p^2)))}$$

- RFC 5348

25

26

## TFRC

- Principe de fonctionnement
  - L'émetteur commence la transmission par une phase de Slow Start assez similaire à celle de TCP.
  - Le récepteur mesure la probabilité de pertes, puis l'envoie à l'émetteur.
  - L'émetteur utilise les paquets envoyés par le récepteur pour mesurer un temps d'aller-retour moyen. Il estime aussi le temporisateur de retransmission.
  - L'émetteur utilise la formule PFTK pour ajuster son débit de transmission :
    - si**  $X_{courant} > X$  **alors**
      - $X_{courant} = X$
    - sinon**
      - augmenter  $X_{courant}$

X est calculé tels que :  
 $X \approx \text{Débit TCP}$ ,  
 $X$  est stable.

## TFRC

- Entête de paquets de données:
  - A sequence number.
  - A timestamp
  - The sender's current estimate of the round trip time
- Entêtes de paquet de "feedback" :
  - The timestamp of the last data packet received.
  - The amount of time elapsed between the receipt of the last data packet at the receiver, and the generation of this feedback report.
  - The rate at which the receiver estimates that data was received since the last feedback report was sent.
  - The receiver's current estimate of the loss event rate,  $p$
- TFRC peut être implanté au niveau application sur UDP ou au niveau transport avec DCCP

27

28

# TFRC

- DCCP : Datagram Congestion Control Protocol (RFC 4340)
  - Protocole générique,
    - Orienté connexion : initiation, fermeture (quelques similarités avec TCP)
    - messages : DCCP-Request, DCCP-Response, DCCP-Data, DCCP-Ack, DCCP-DataAck, DCCP, ...
  - L'émetteur et le récepteur négocient un mode de contrôle de congestion → CCID
    - CCID = 2, TCP-like (RFC 4341)
    - CCID = 3, TFRC** (RFC 4342)
    - exemples de messages : change R(CCID, 2), change L(CCID, 3)

29

## Measurement-based congestion control

- Exemple :
  - Augmenter/réduire le débit en se basant sur les changements dans le temps d'aller-retour (RTT)
- TCP Vegas et FastTCP utilisent une méthode hybride AIMD-based measurement-based:
- Les changements dans le temps d'aller-retour indiquent les congestions
  - Mesure le minimum et la moyenne de RTT
  - Si RTT augmente alors même réaction en cas de perte
- Objectif: Réaction plus rapide avant les pertes de paquets

30

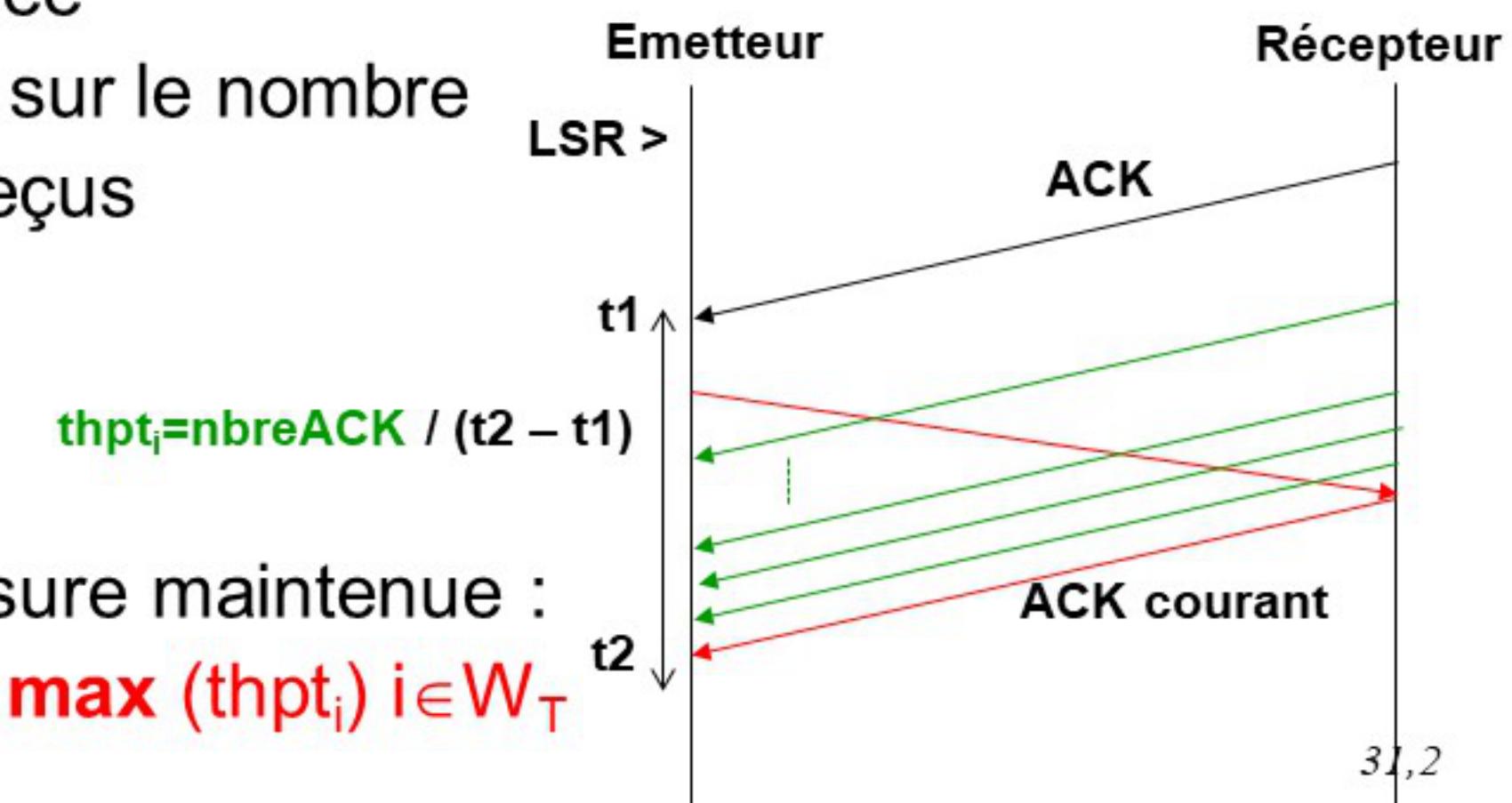
## Measurement-based congestion control

- TCP BBR
  - Mesure la bande passante offerte par le réseau à la connexion TCP
    - Diminue le débit en s'alignant avec ces mesures
  - Tente d'**augmenter** le débit périodiquement (Sondage de bande passante)
  - Estime le nombre de paquets pouvant être envoyés sans recevoir d'acquittement
    - Arrête l'envoi de paquets si ce nombre est atteint
- Objectif: Réaction plus rapide avant les pertes de paquets

31,1

## Measurement-based congestion control

- TCP BBR
  - Mesure la bande passante offerte par le réseau à la connexion TCP
    - Diminue le débit en s'alignant avec ces mesures
    - A la réception de chaque ACK, une mesure est effectuée
    - Basée sur le nombre d'ACK reçus
- La mesure maintenue :  $B_{Prés} = \max_i (thpt_i) i \in W_T$

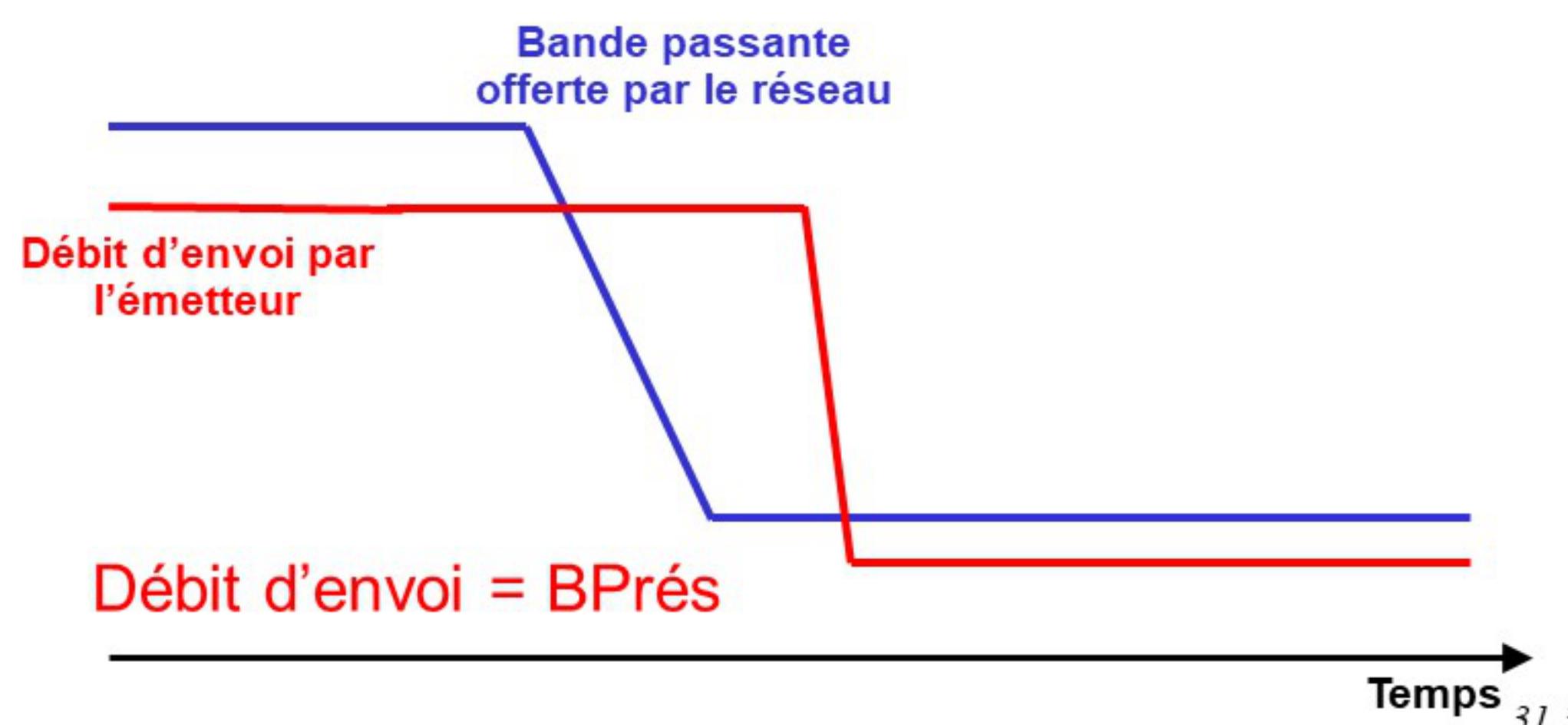


31,2

## Measurement-based congestion control

- TCP BBR

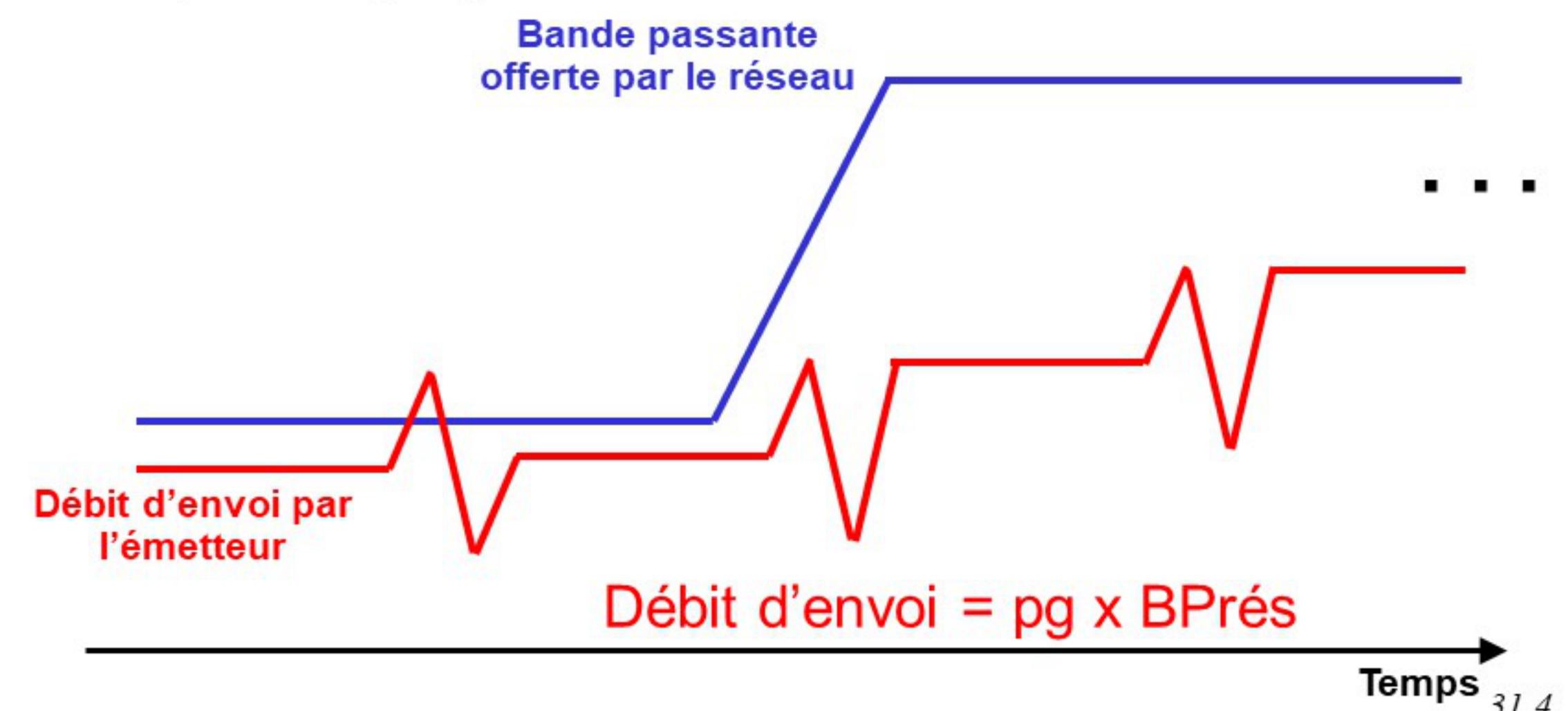
- Mesure la bande passante offerte par le réseau à la connexion TCP
  - Diminue le débit en s'alignant avec ces mesures



## Measurement-based congestion control

- TCP BBR

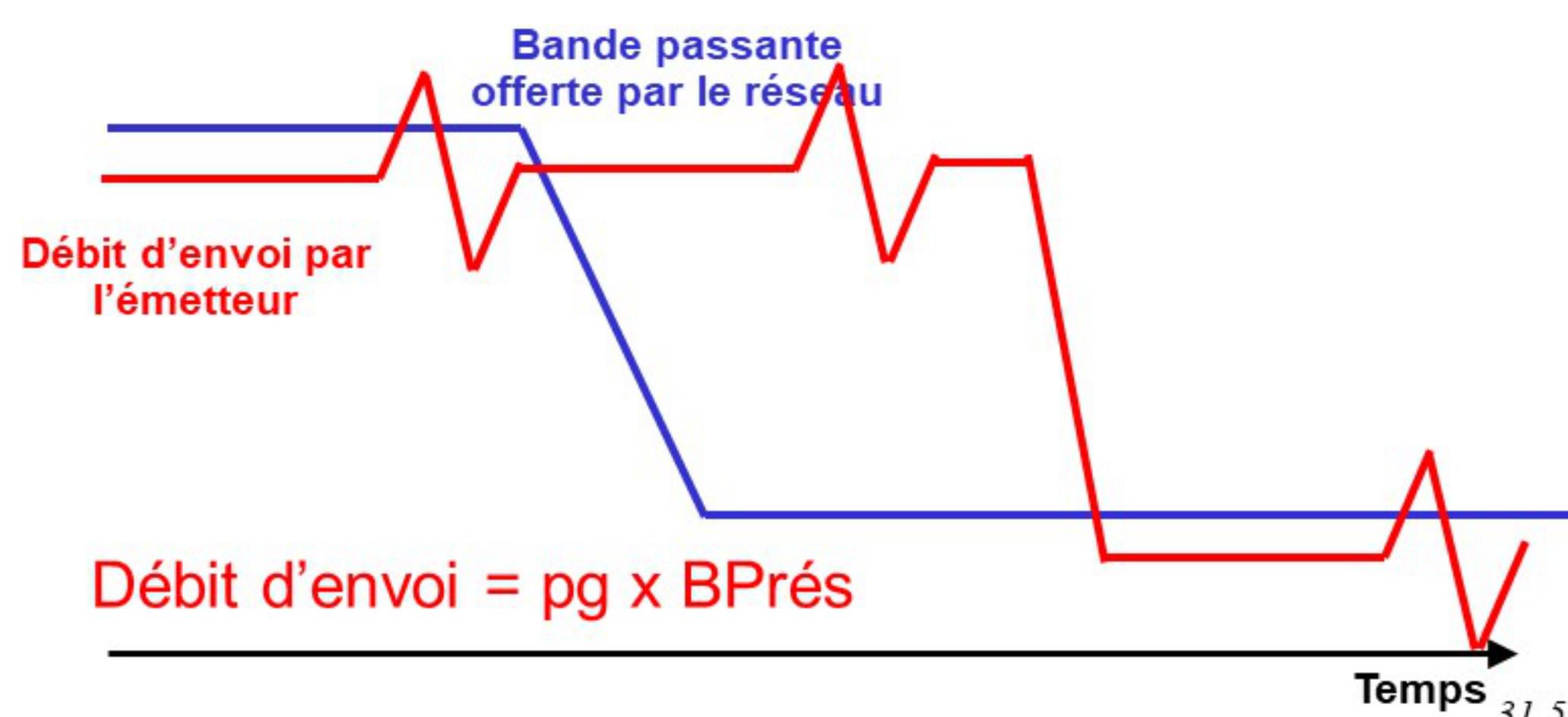
- Tente d'augmenter le débit périodiquement (Sondage de bande passante, « bandwidth probing »)



## Measurement-based congestion control

- TCP BBR

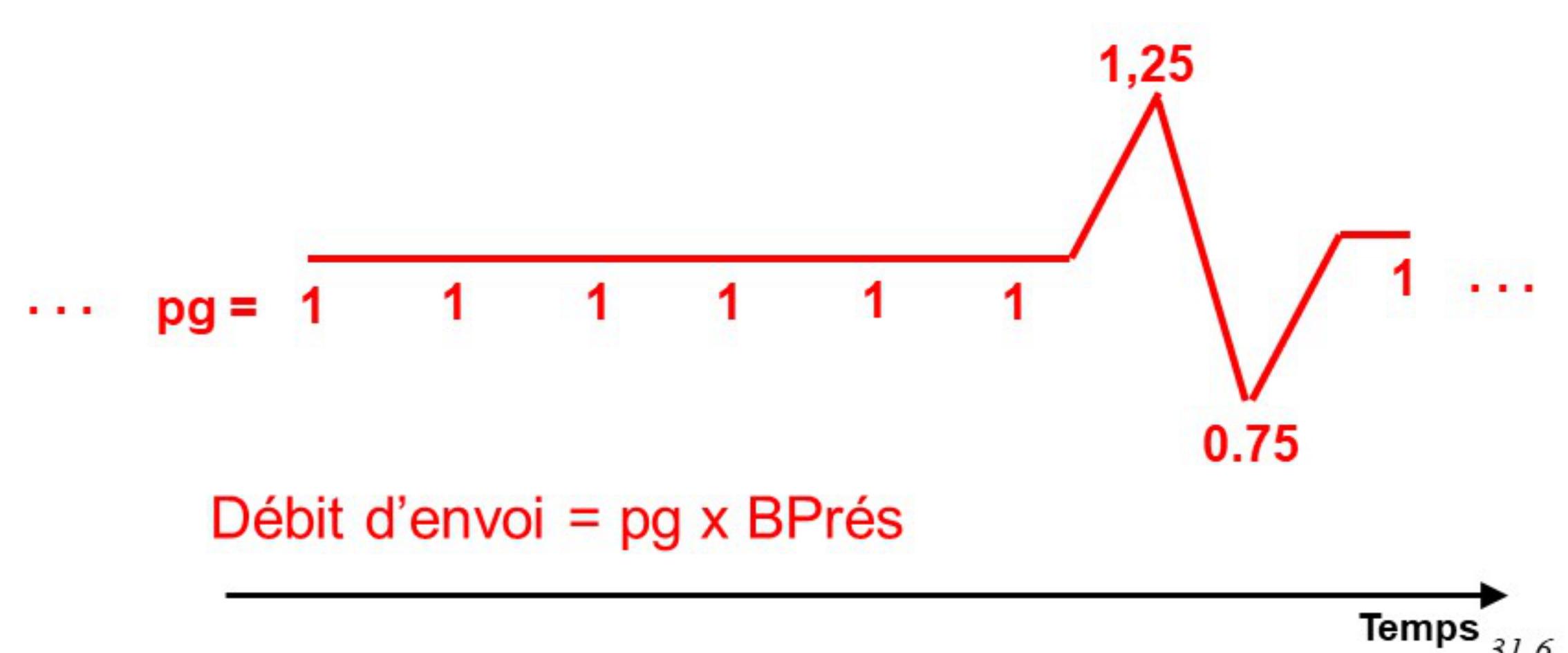
- Mesure la bande passante offerte par le réseau à la connexion TCP
  - Diminue le débit en s'alignant avec ces mesures



## Measurement-based congestion control

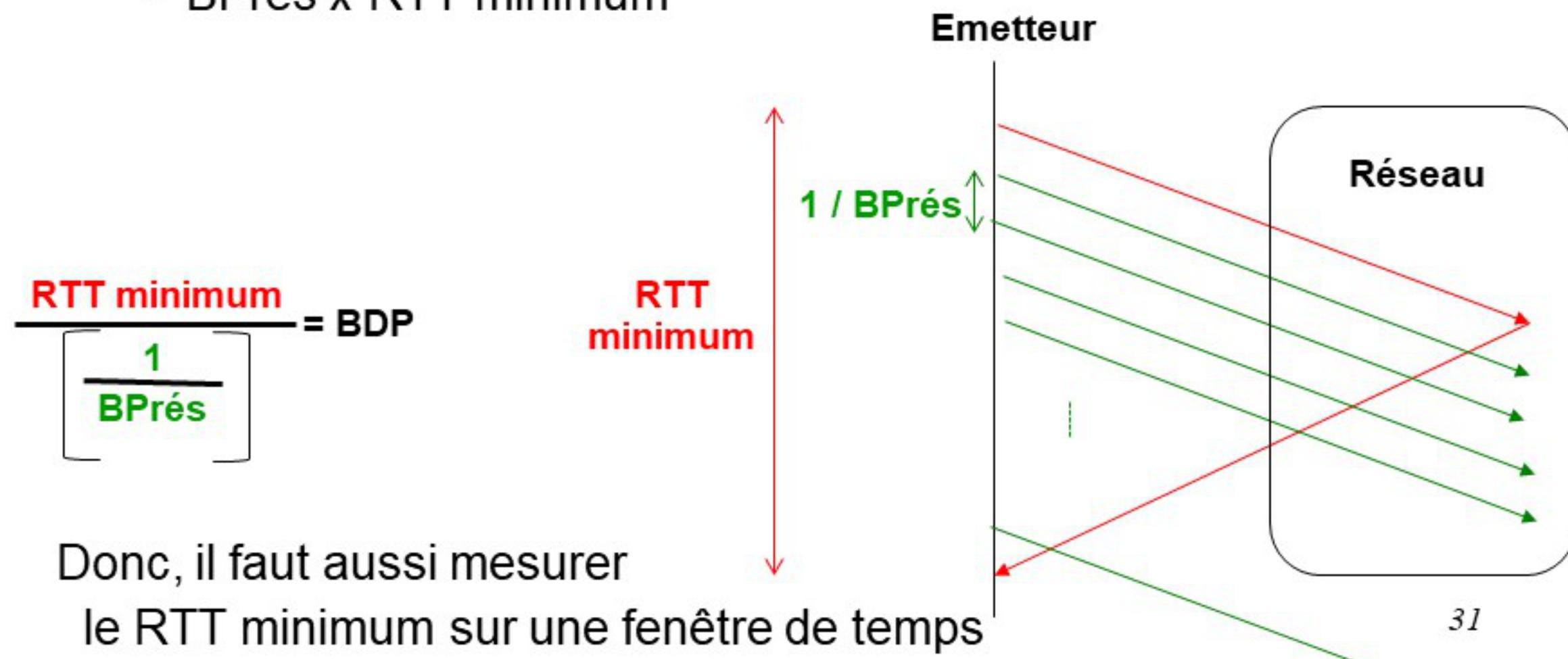
- TCP BBR

- pg est le « pacing gain » qui change de manière périodique
  - Sondge périodique de la bande passante résiduelle



## Measurement-based congestion control

- TCP BBR
  - Estime le nombre de paquets pouvant être envoyés sans recevoir d'acquittement
    - Arrête l'envoi de paquets si ce nombre est atteint
  - Il s'agit d'estimer le BDP : **Bandwidth Delay Product**
    - $BPrés \times RTT$  minimum



## Contrôle de congestion assisté par le réseau

- Nécessaire afin d'offrir des niveaux de service largement supérieur à Best-Effort
  - Les éléments du réseau (routeurs, ...) peuvent mieux estimer l'état de leur congestion.
- Souvent implanté via les mécanismes de gestion active de buffer (« Active Queue Management : AQM »).

## Contrôle de congestion de bout-en bout

- TCP NewReno (AIMD)
- TCP CUBIC
- Fast TCP
- TCP BBR
- . . .
- Pas suffisant pour offrir une qualité vraiment « better than best effort »
- Une assistance du réseau est nécessaire
  - Le réseau (les routeurs) connaît mieux son état (l'état des buffers des interfaces de transmission)<sup>32,1</sup>

## Gestion de buffer

- A chaque interface de sortie le routeur associe un buffer (mémoire tampon)
- Les paquets qui arrivent pendant la transmission d'un paquet en cours sont mis en attente dans le buffer
- Gestion du buffer :
  - Principalement : **Politique de rejet**
  - Quel paquet rejeter quand le buffer est rempli ?
- Interagit directement avec le contrôle de congestion et la gestion de trafic

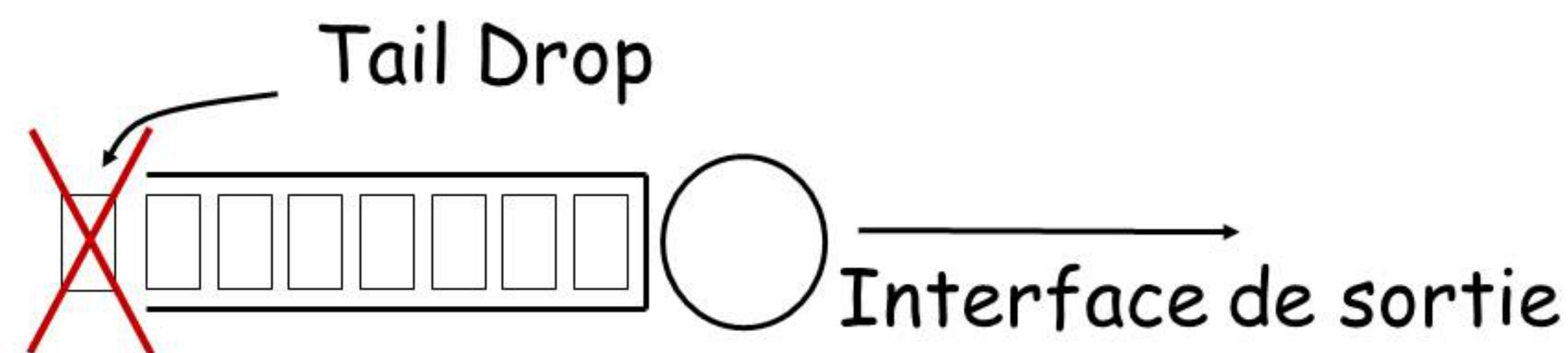
## Gestion de buffer

- A chaque interface de sortie le routeur associe un buffer (mémoire tampon)
- Les paquets qui arrivent pendant la transmission d'un paquet en cours sont mis en attente dans le buffer
- Gestion du buffer :
  - Principalement : **Politique de rejet** :
    - **Quand rejeter les paquets ?**
    - **Quel paquet rejeter ?**
- Interagit directement avec le contrôle de congestion et la gestion de trafic

34

## Gestion de buffer

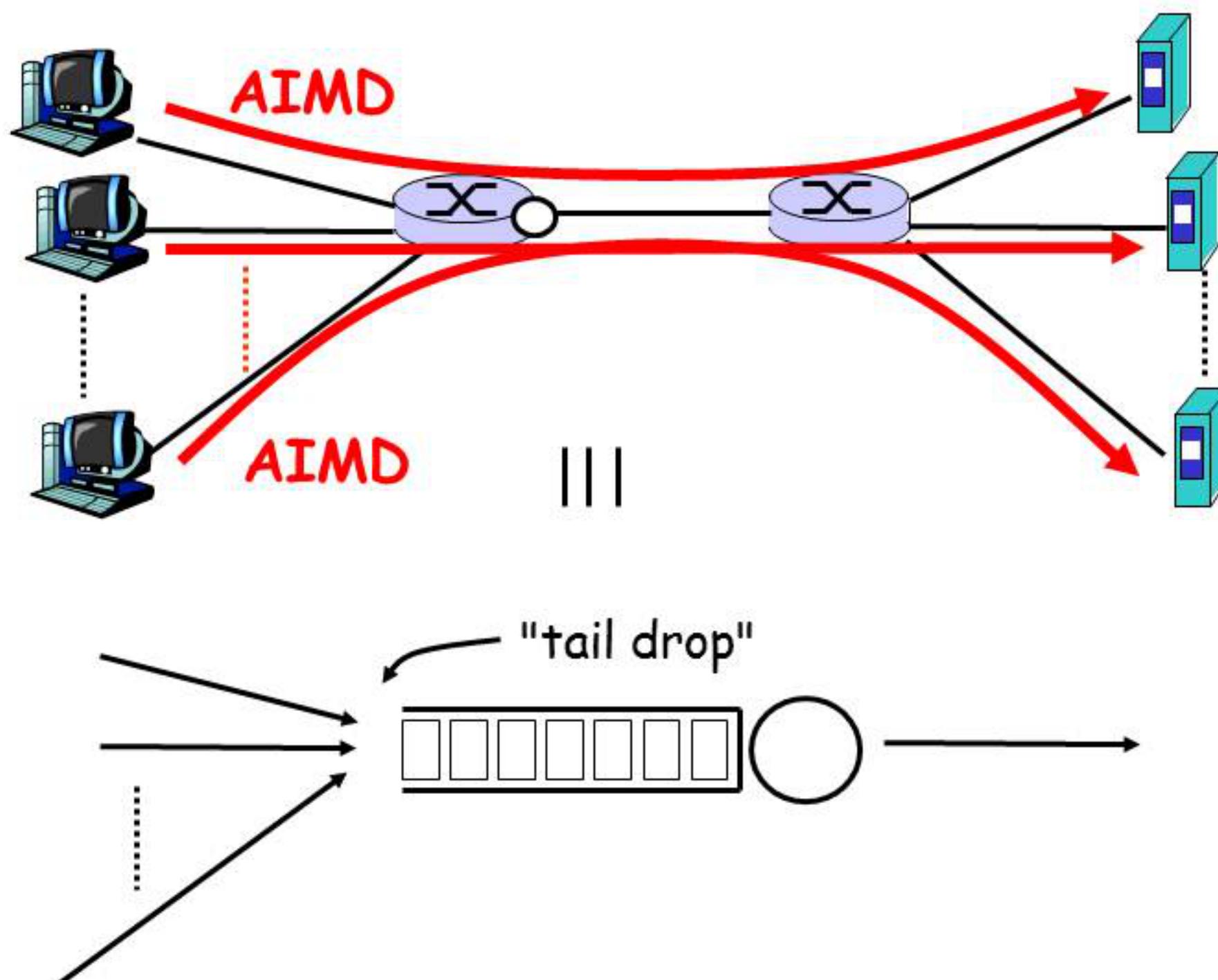
- Politique par défaut : Tail Drop (TD)
  - Le paquet qui vient d'arriver est rejeté.



- Logique, intuitive
- Les paquets sont traités de la même façon

35

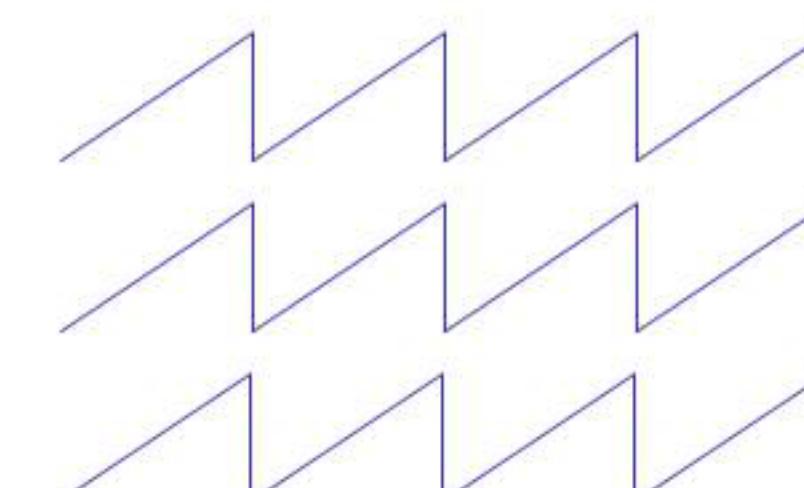
## TD et AIMD



36

## TD et AIMD

- Phénomène de **synchronisation globale** :
  - Avec Tail Drop les pertes sont groupées
  - Les connexions AIMD réagissent presque simultanément
- Conséquence :
  - Le réseau oscille entre période de sur-utilisation et période de sous-utilisation

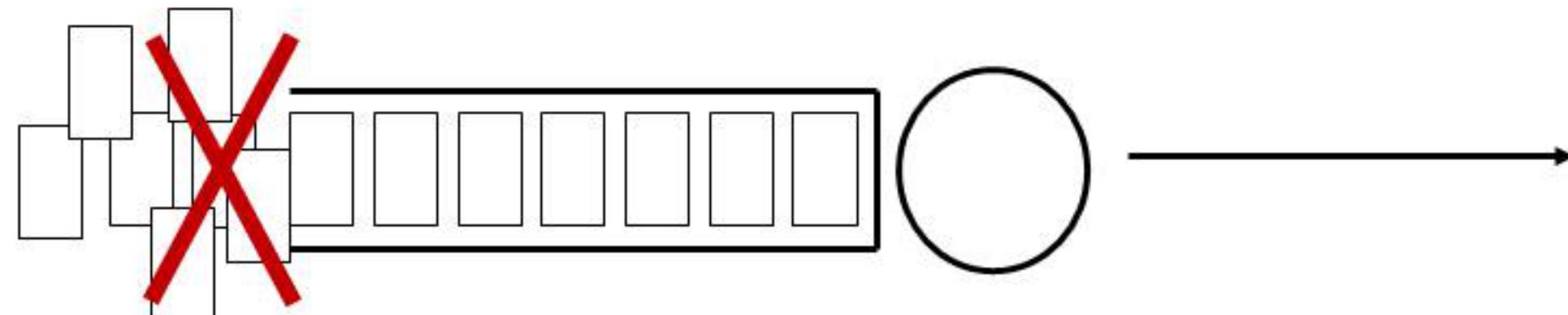


37

## TD et les rafales

### Injustice envers le trafic en rafale :

- Les connexions avec des rafales perdent plus de paquets que les autres connexions
- Quand une rafale de paquets trouve le buffer du routeur plein, plusieurs paquets sont perdus simultanément

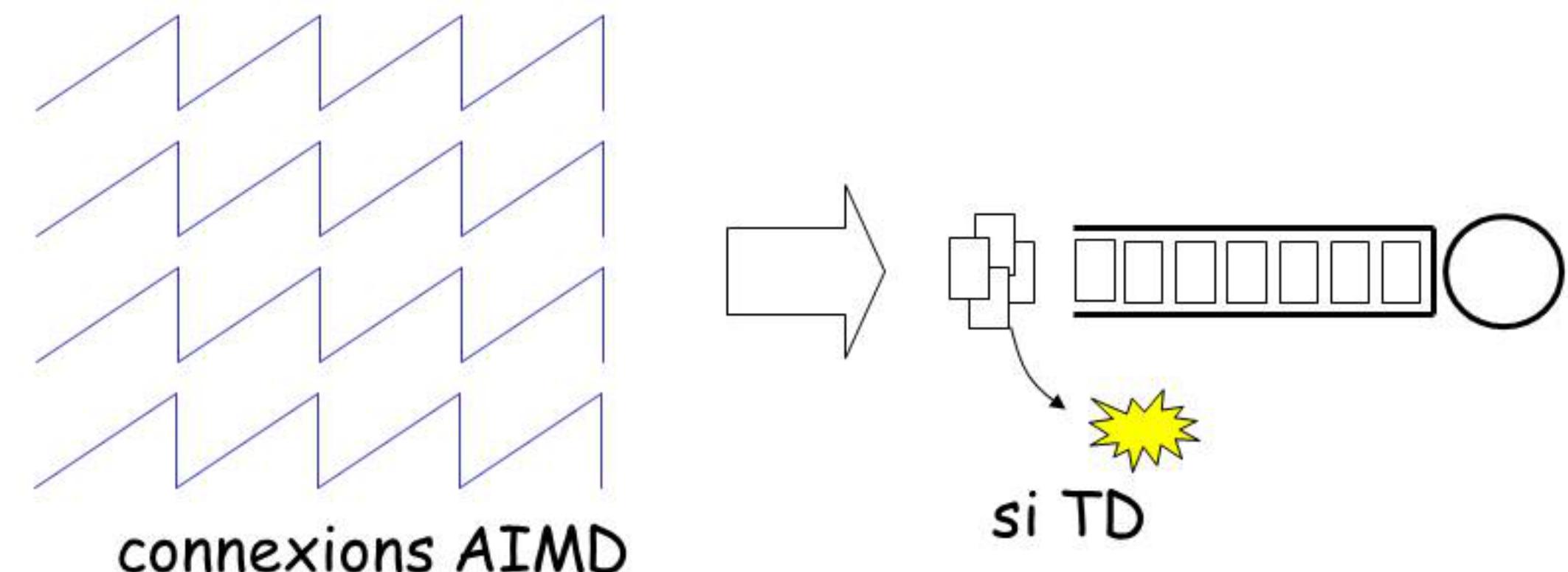


38

## La gestion de buffer RED

### RED : Random Early Detection

- Jeter les paquets à l'avance, avant d'atteindre la limite du buffer
- Rejet aléatoire → améliore le multiplexage

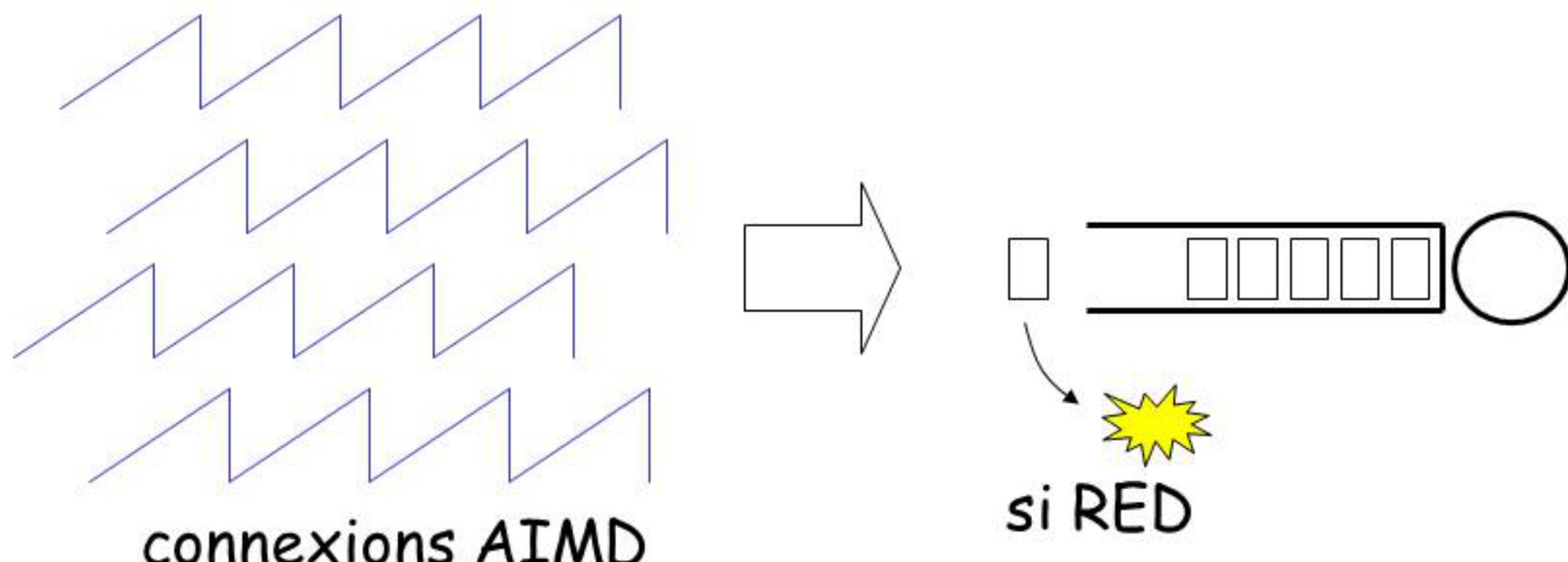


39

## La gestion de buffer RED

### RED : Random Early Detection

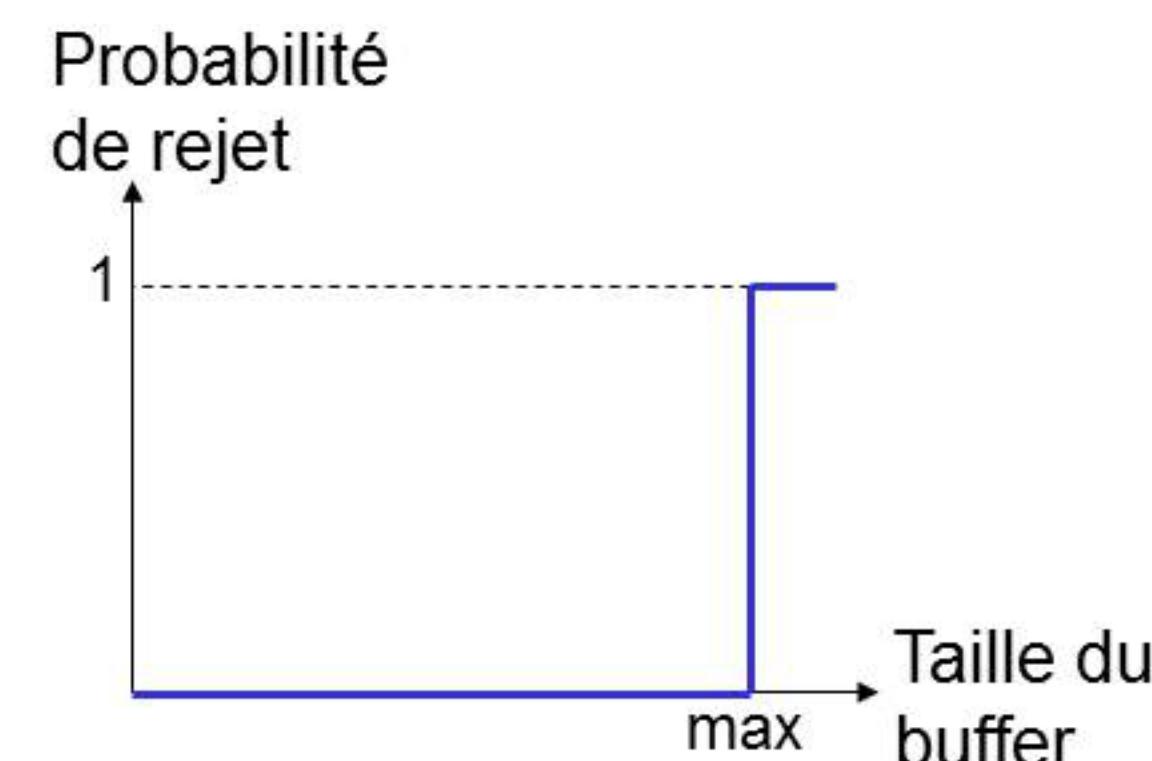
- Jeter les paquets à l'avance, avant d'atteindre la limite du buffer
- Rejet aléatoire → améliore le multiplexage



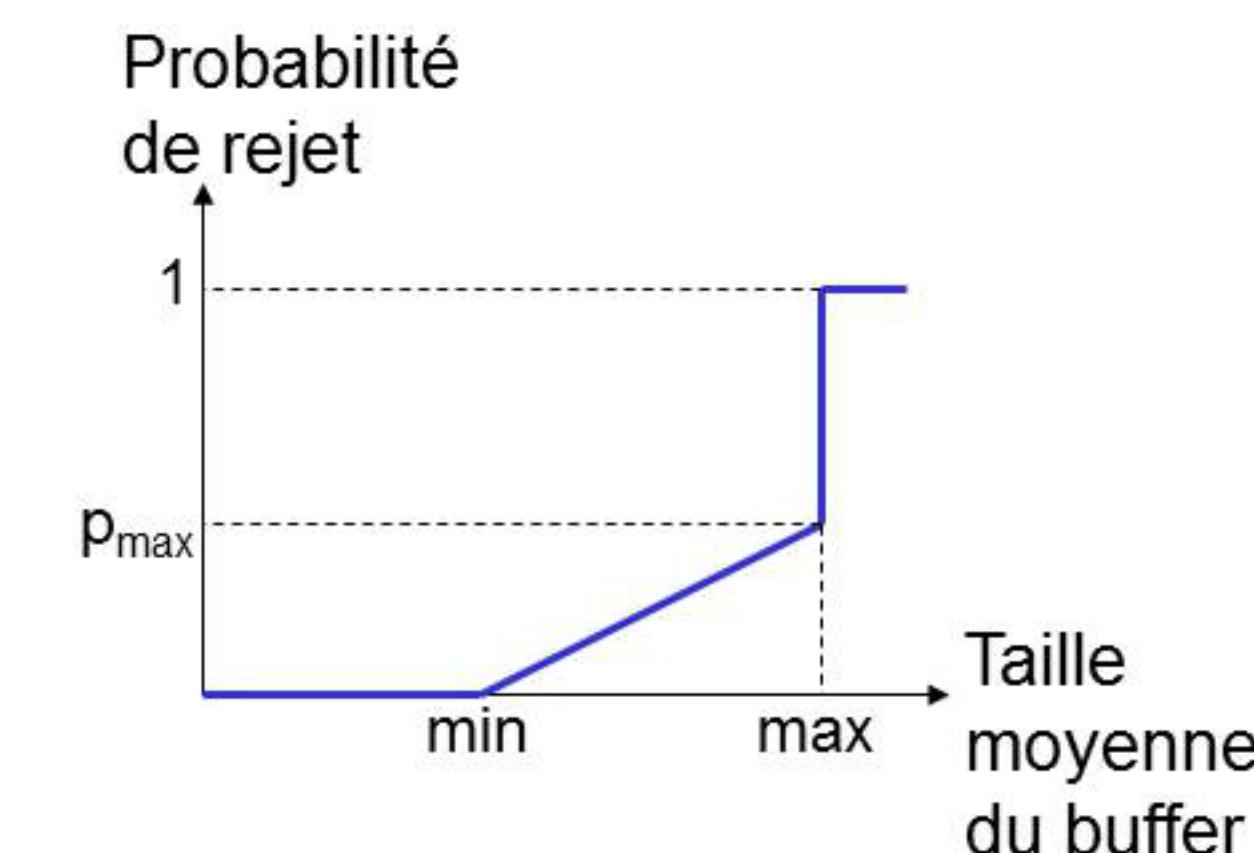
40

## Gestion de buffer : RED

### TD : Tail Drop



### RED : Random Early Detection



Taille du buffer = nombre de paquets dans le buffer ( $q$ )

Taille moyenne du buffer = nombre moyen de paquets dans le buffer (avg)

41

## RED

**Exponential  
Weighted  
Moving  
Average**

- L'algorithme (version simplifiée)

A l'arrivée d'un paquet :

mesurer  $q$  // la taille courante du buffer

$\text{avg} \leftarrow (1-w) * \text{avg} + w * q // w << 1$

si  $\text{min} \leq \text{avg} < \text{max}$

$p \leftarrow p_{\text{max}} * (\text{avg} - \text{min}) / (\text{max} - \text{min})$

jeter (ou marquer) le paquet avec  
la probabilité  $p$

sinon

si  $\text{avg} \geq \text{max}$

jeter le paquet

// (jeter avec probabilité 1)

42

## RED + ECN

- Explicit Congestion Notification :

- Marquer les paquets au lieu de les jeter

- Le routeur positionne le bit CE (Congestion Experienced) de l'en-tête du paquet IP à 1.

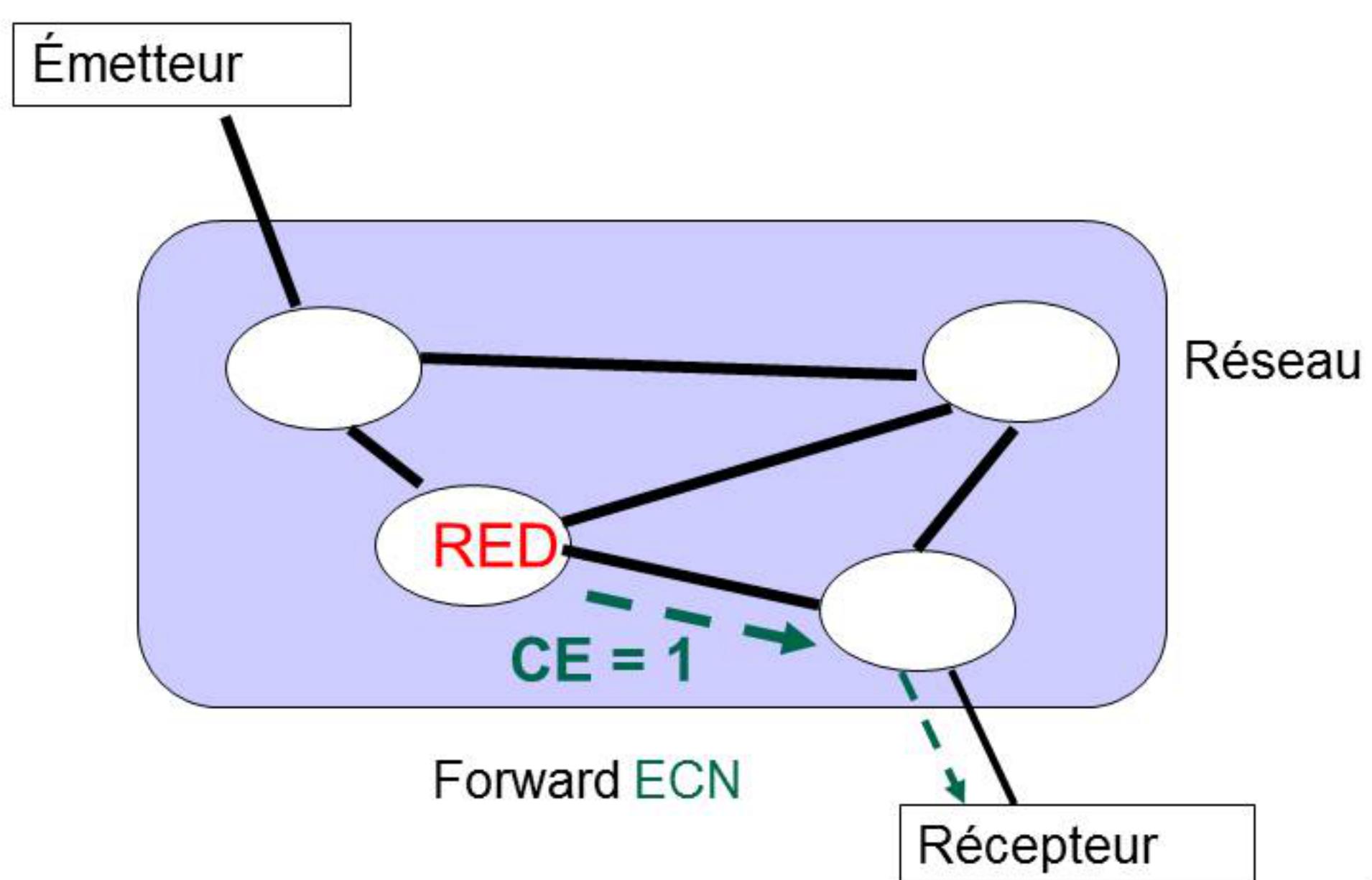
- Inutile si l'application ou le protocole de transport n'est pas modifié pour en tenir compte

43

## RED + ECN

## TCP + RED + ECN

- Explicit Congestion Notification :



44

- RFC 3168 :

- Pendant l'établissement de la connexion, la source positionne un bit ECT (ECN Capable Transport) à 1 pour indiquer qu'il supporte ECN

- Les bits ECN-echo et CWR de l'en-tête TCP sont utilisées pour la signalisation entre l'émetteur et le récepteur.

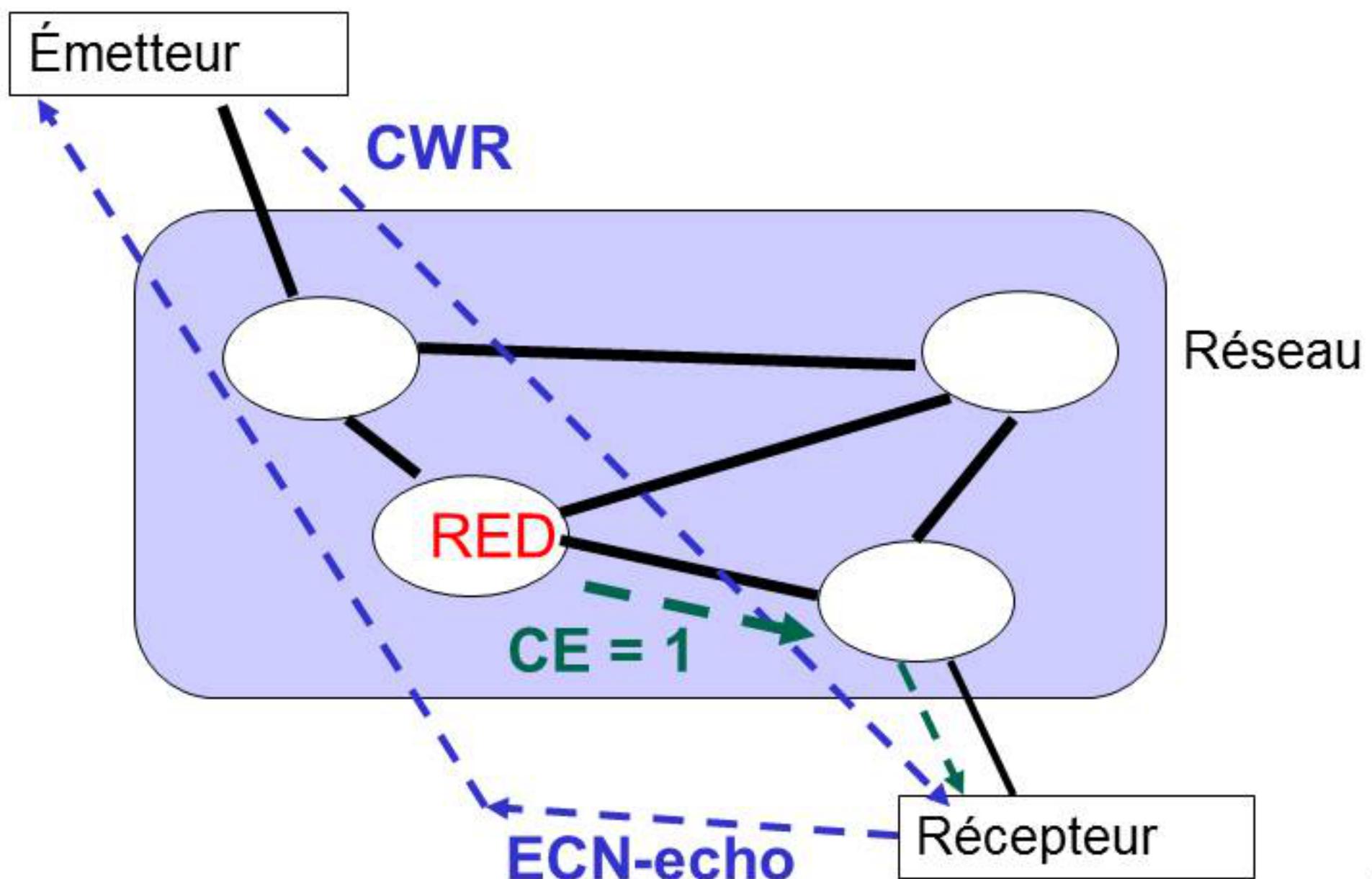
- ECN-echo : pour marquer tous les ACK suivant une indication de congestion (CE=1)

- CWR : utilisé par l'émetteur pour indiquer qu'il a bien réagi au signal de congestion

45

# RED + ECN

- Explicit Congestion Notification :



46

## Garanties

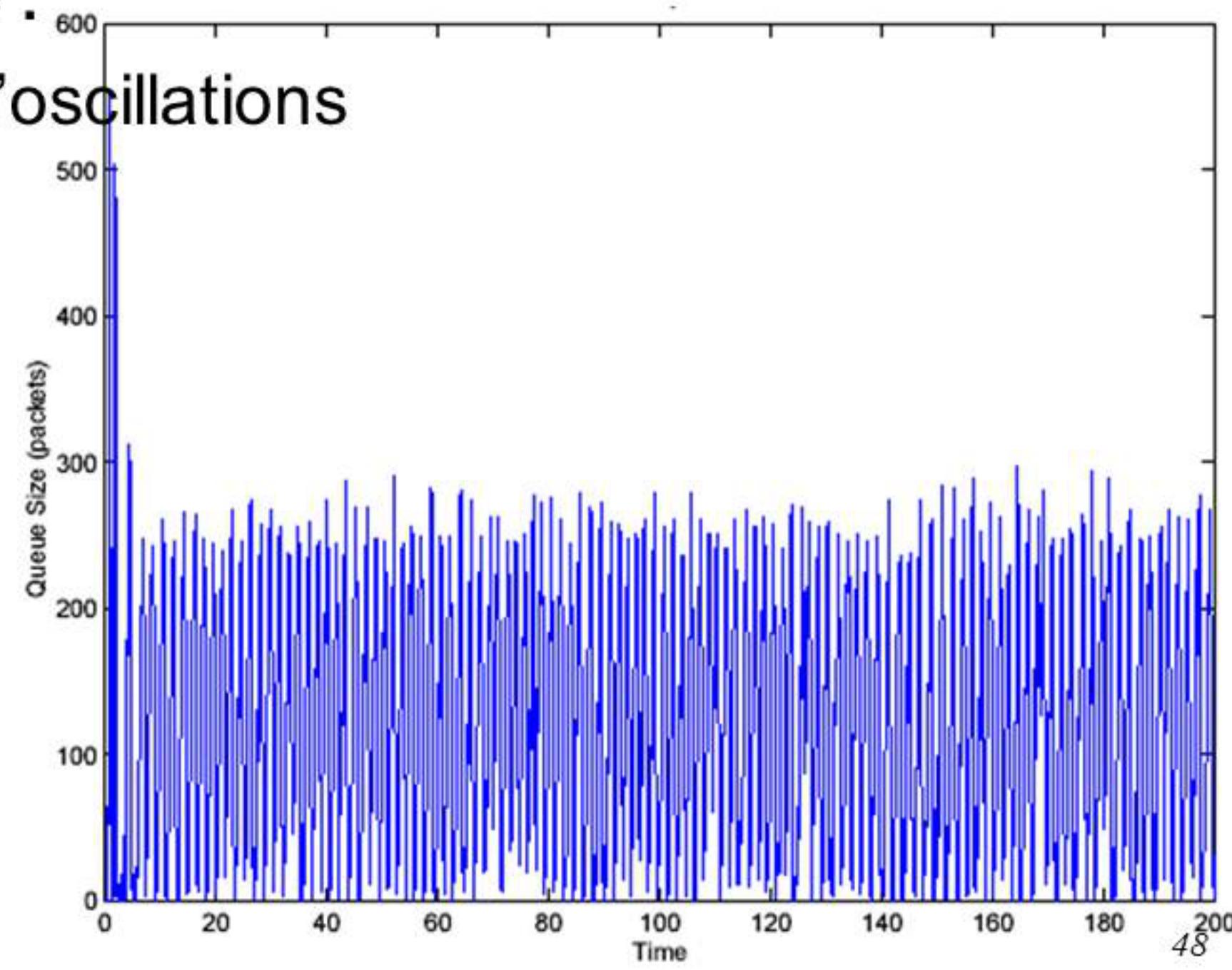
- Performance (non quantifiée)
  - Injustice envers le trafic à rafales, synchronisation globale ??
  - Comment choisir les paramètres
    - ( $p_{\max}$ , min, max) ?
  - Peuvent être obtenus à partir de tests sur des cas simples
- Tests possibles :
  - Comportement stationnaire :
    - débit moyen, utilisation moyenne du buffer, etc.
  - Comportement dynamique :
    - stabilité, utilisation instantanée du buffer, etc.

47

## Garanties

- Stabilité :

- Trop d'oscillations

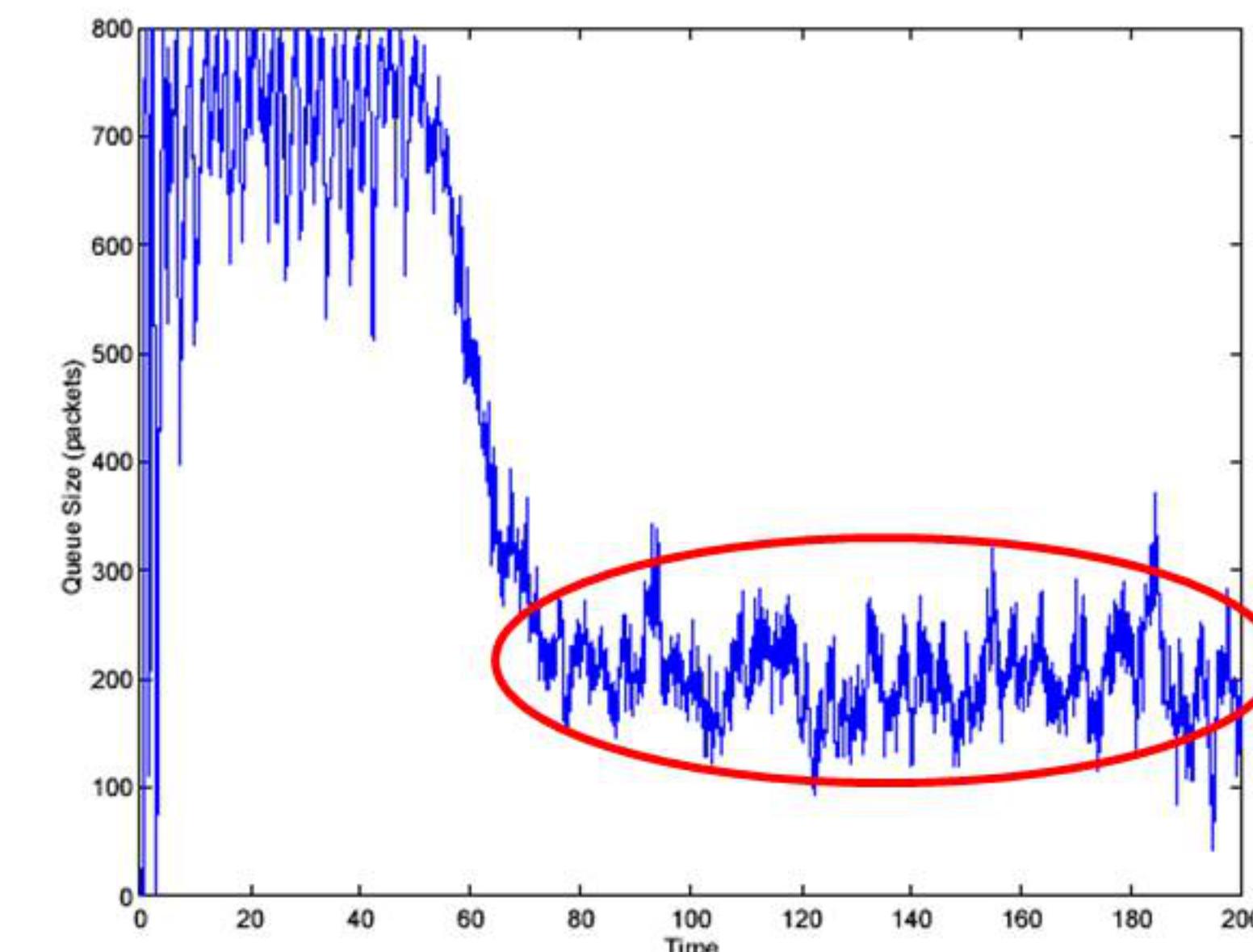


48

## Garanties

- Stabilité :

- Objectif à atteindre (bonne configuration)

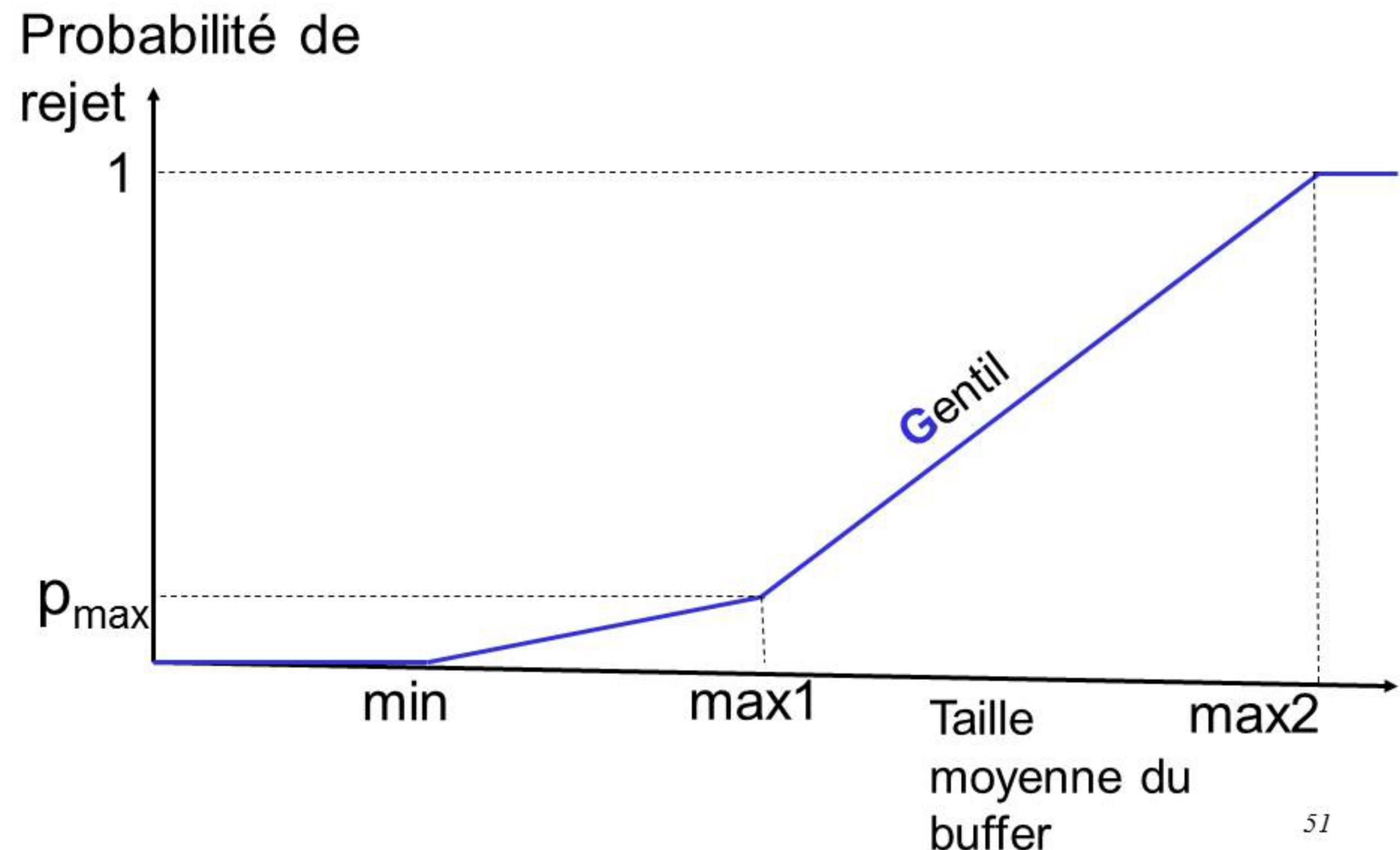


49

# Gestion de buffer autre que RED

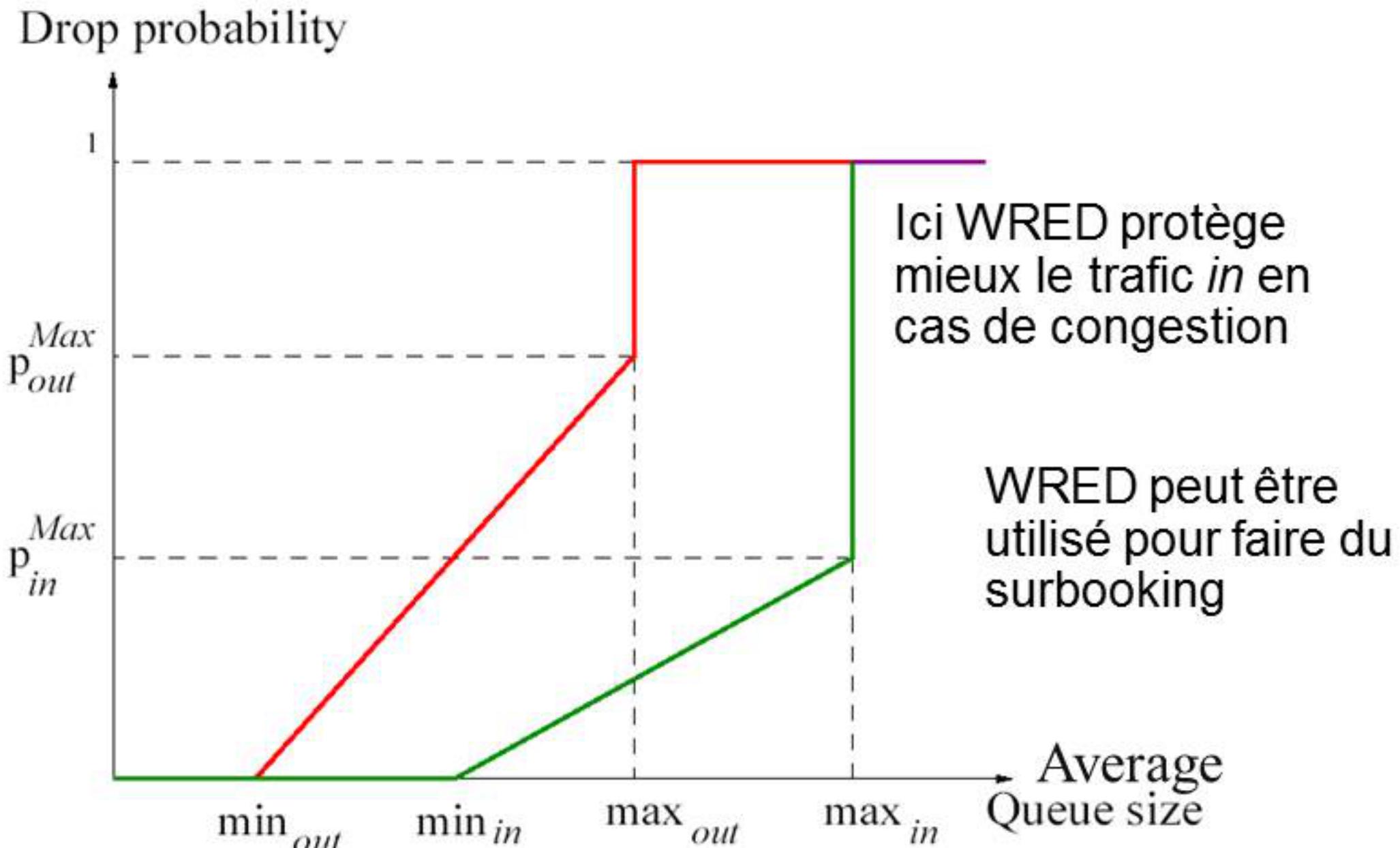
GRED

- La liste est longue !
- Exemples :
  - Gentle RED (GRED)
  - WRED
    - Class-based ou flow-based
  - BLUE



## WRED

- Weighted RED : Single average multiple thresholds
  - e.g. deux couleurs (*in*, *out*) partiellement imbriquées :



Les paquets ne sont plus traités de la même façon