



De l'utilisation de traces de métrologie pour des simulations réalistes

– Philippe Owezarski, Nicolas Larrieu

LAAS-CNRS

Toulouse, France

{owe, nlarrieu}@laas.fr



- ▶ Introduction
- ▶ Les erreurs habituelles en simulation
- ▶ Éléments caractéristiques du trafic Internet
- ▶ Le rejeu de traces en simulation
- ▶ Perspectives

A circular inset image showing a modern office interior with a desk, a chair, and a window with blinds. The word 'HAUTE' is visible on a wall in the background.

Introduction

- ▶ La simulation est un outil essentiel pour la recherche et l'ingénierie réseau

- ▶ Permet une évaluation a priori
 - ▶ De l'absence de bugs
 - ▶ De sa vivacité
 - ▶ De son intégrité
 - ▶ De sa disponibilité
 - ▶ De ses performances
 - ▶ Etc.

Difficultés pour simuler l'Internet



- ▶ Simuler l'Internet n'est pas une tâche aisée à cause de :
 - ▶ Sa taille (nb utilisateurs, nb équipements)
 - ▶ Sa complexité (nb de protocoles, de technologies différentes, etc.)
 - ▶ Les comportements variés des utilisateurs et applications
 - ▶ Son évolution rapide en termes :
 - De technologie
 - D'usages

- ▶ Comment réaliser des simulations réalistes de l'Internet ?

- ▶ Projet dont sont issus NS et NS-2
- ▶ K. Fall, « Network emulation in the VINT/NS simulator », ISCC'99
- ▶ V. Paxson, S. Floyd, « difficulties in simulating the Internet », IEEE/ACM ToN, August 2001

- ▶ Recommande de travailler à partir des résultats d'analyse des traces capturées dans des projets de métrologie
 - « trace based simulation »
- ▶ Le « shaping » des paquets doit être fait par l'environnement de simulation en fonction de la topologie simulée
 - Travailler au niveau des flux

- ▶ Les sources de trafic simulées doivent respecter des propriétés invariantes du trafic
 - ▶ Corrélation à long terme entre paquets
 - ▶ Distribution log-normale des tailles de flux
 - ▶ Distributions à queue lourde des paramètres caractéristiques de l'activité réseau
 - ▶ Etc.



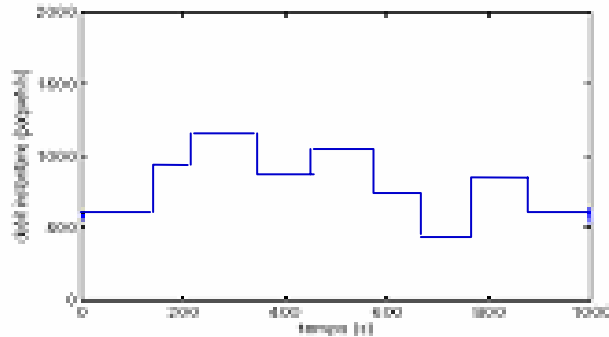
Les erreurs habituelles en simulation



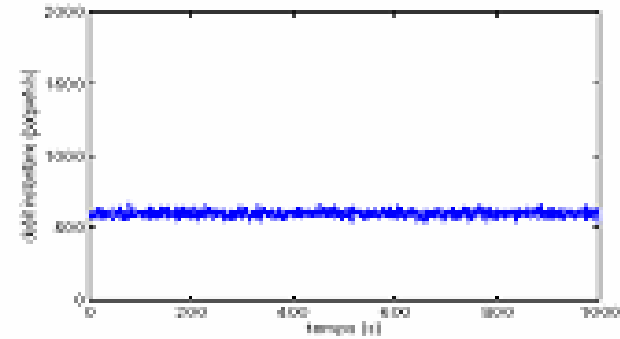
- ▶ Eviter les sources plus simples que le trafic réel
 - ▶ CBR
 - ▶ Poisson
 - ▶ Markov
 - ▶ ON-OFF avec des durées exponentielles
 - ▶ Etc.
- ▶ Les calculs seront plus simples / possibles
- ▶ Mais les résultats erronés !



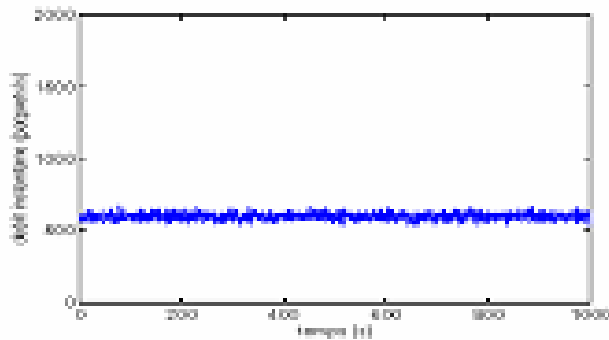
Quelques exemples de sources



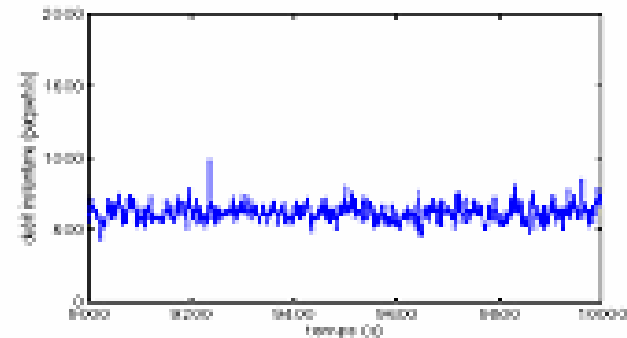
Constant



Poisson



ON-OFF avec durées exponentielles



Réel

- ▶ Pas/peu de contraintes d'instabilité
- Comportements faciles à analyser et comprendre (bon pour debug)

MAIS

- De nombreux cas possibles / fréquents dans la réalité ne sont pas rencontrés
- Les évaluations de performances sont optimistes



- ▶ Bien connaître les caractéristiques du trafic du réseau, ou mieux, son modèle !
- ▶ Pour effectuer des simulations dans des environnements réalistes



Éléments caractéristiques du trafic Internet (Projet METROPOLIS)

L'Internet est en mutation

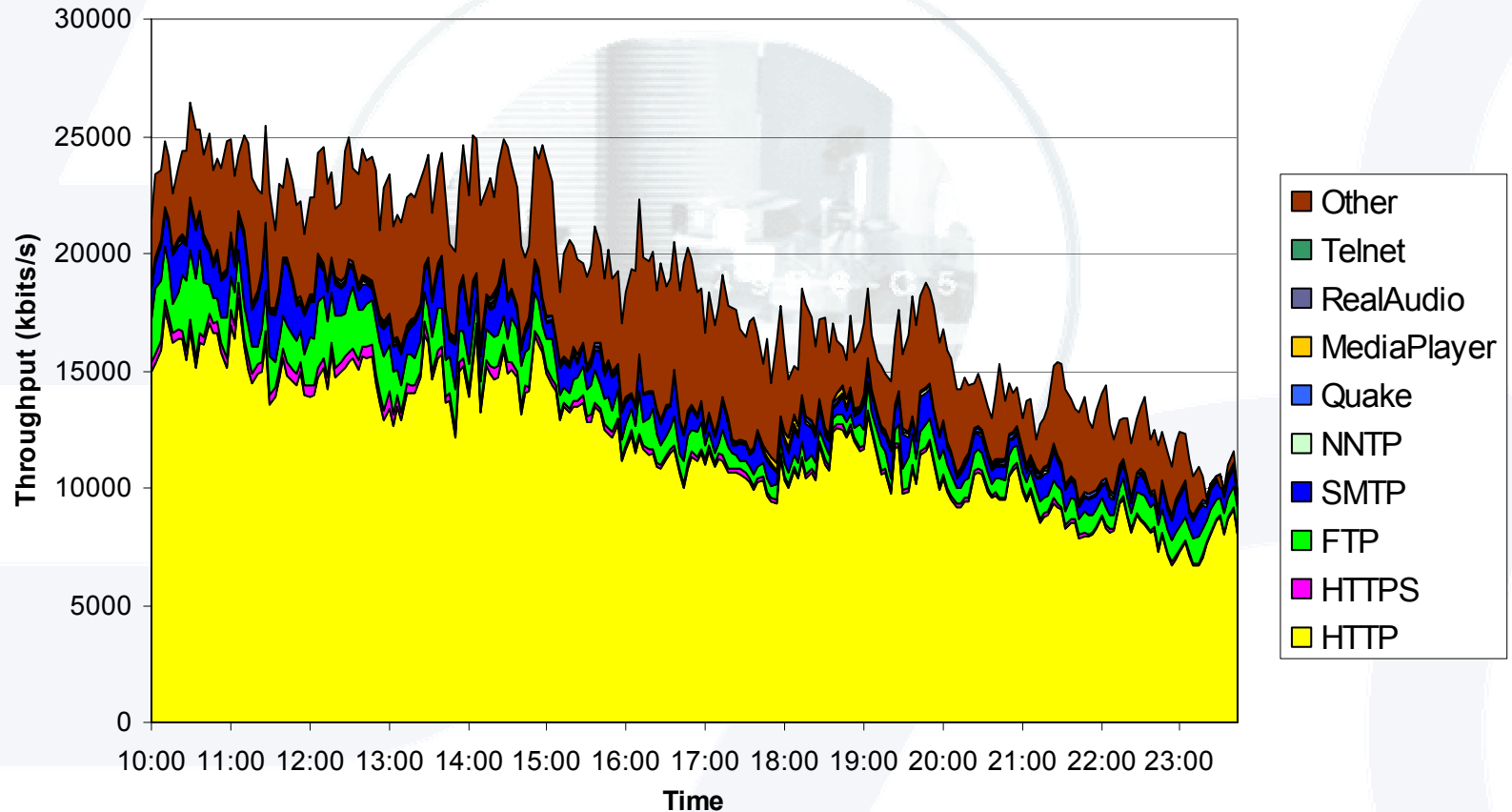
- ▶ Le trafic était presque exclusivement du trafic web
- ▶ Aujourd'hui, la part de trafic P2P ne cesse d'augmenter
 - ▶ mix entre trafic web et P2P, et de plus en plus de streaming



Trafic en mai 2000 (Sprint)



Main TCP applications throughputs

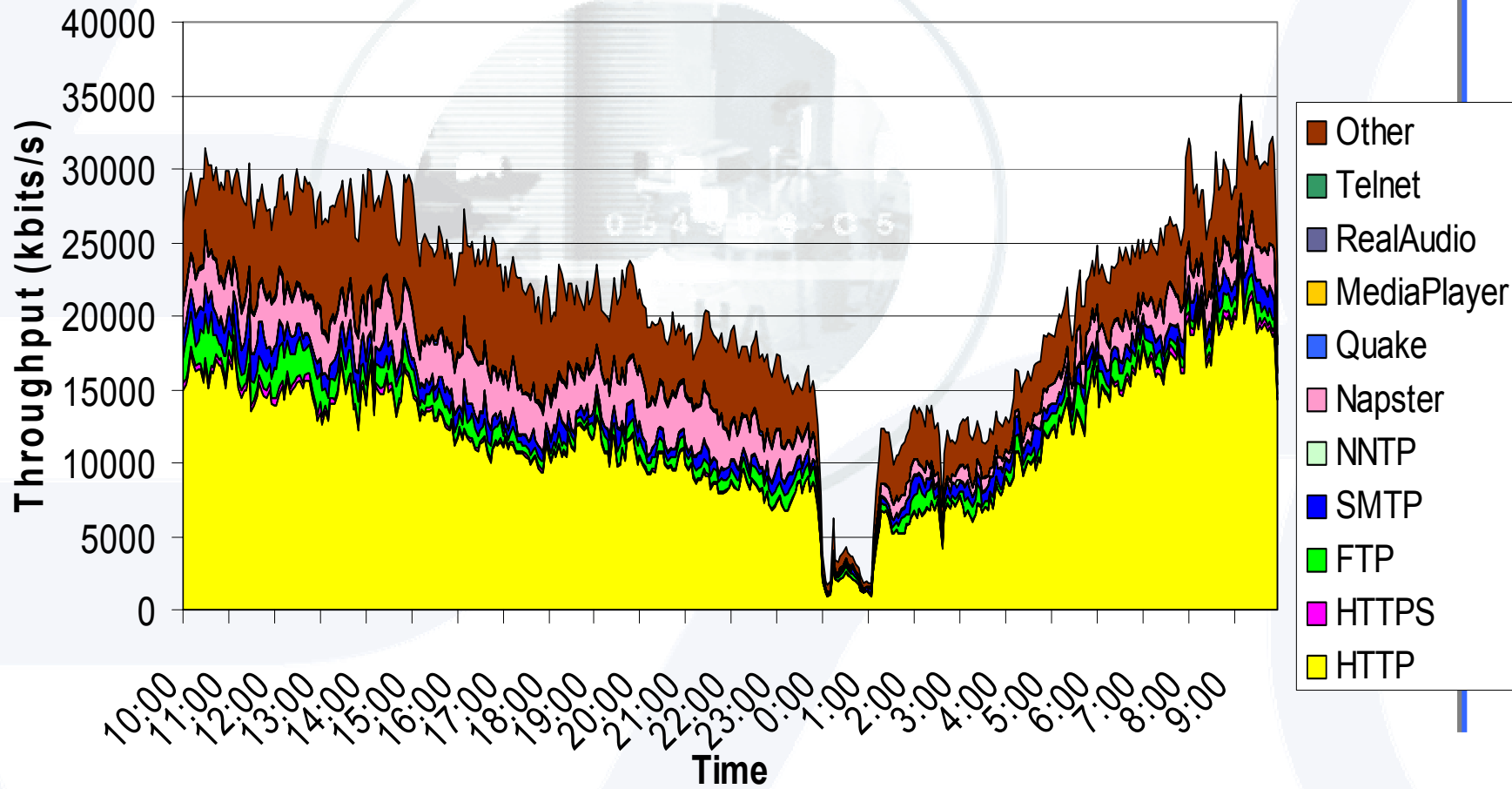




Trafic en août 2000 (Sprint)



Main TCP applications throughputs



Trafic en mai 2003 (Renater)

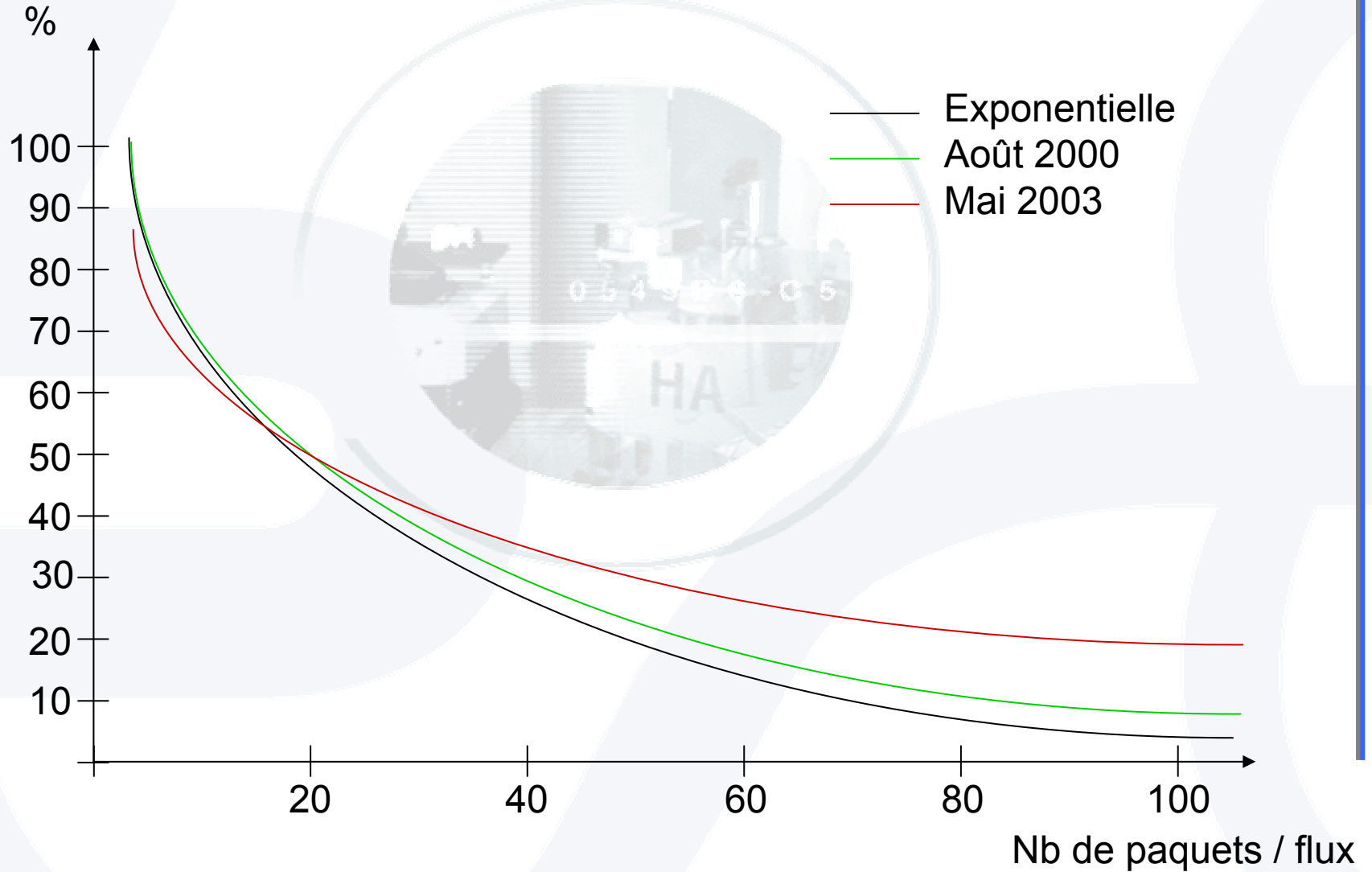


Impact du P2P sur le trafic

- ▶ Des milliers de souris
 - ▶ Un grand nombre d'éléphants
- Change la distribution des tailles de flux



Distributions des tailles de flux

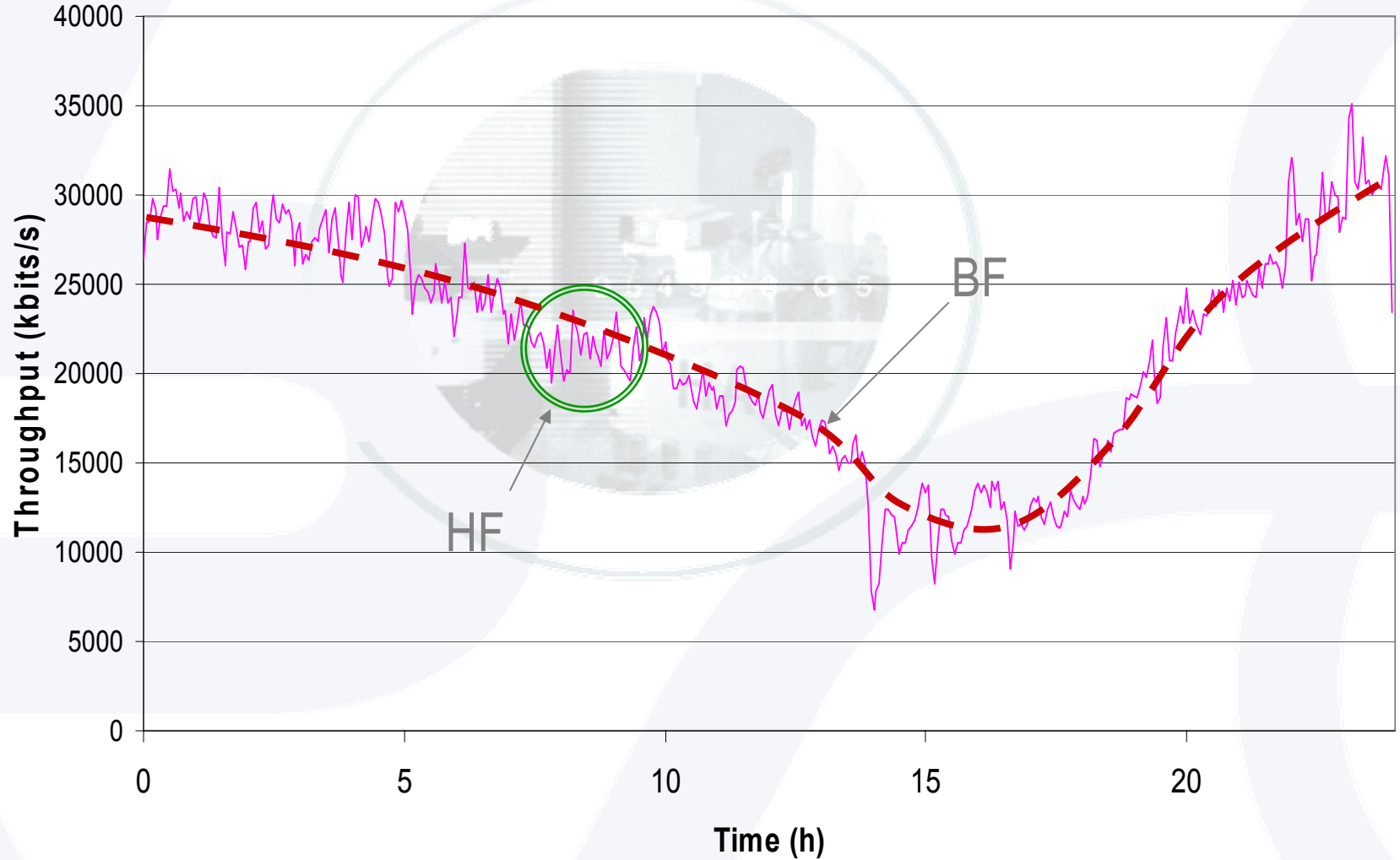


- ▶ Le trafic a des caractéristiques d'**auto-similarité**
- ▶ Dues à:
 - ▶ Distributions à queues lourdes des tailles de flux
 - ▶ Contrôle de congestion à la mode TCP
 - ▶ Routeurs
 - ▶ Comportements humains et des applications





Trafic réel



► QoS, Stabilité, Performance

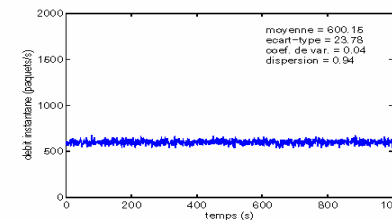
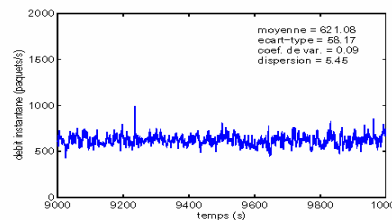
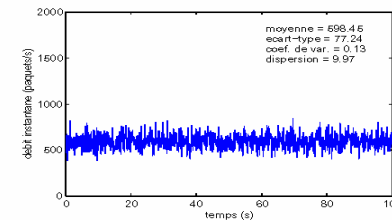
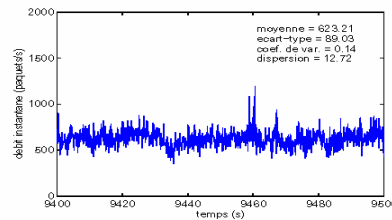
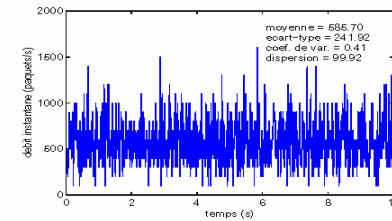
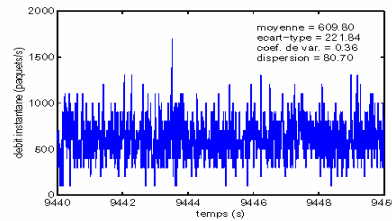
Trafic observé trafic Poissonnien simulé

Débit mesuré avec
comme granularité :

0.01 s

0.1 s

1 s



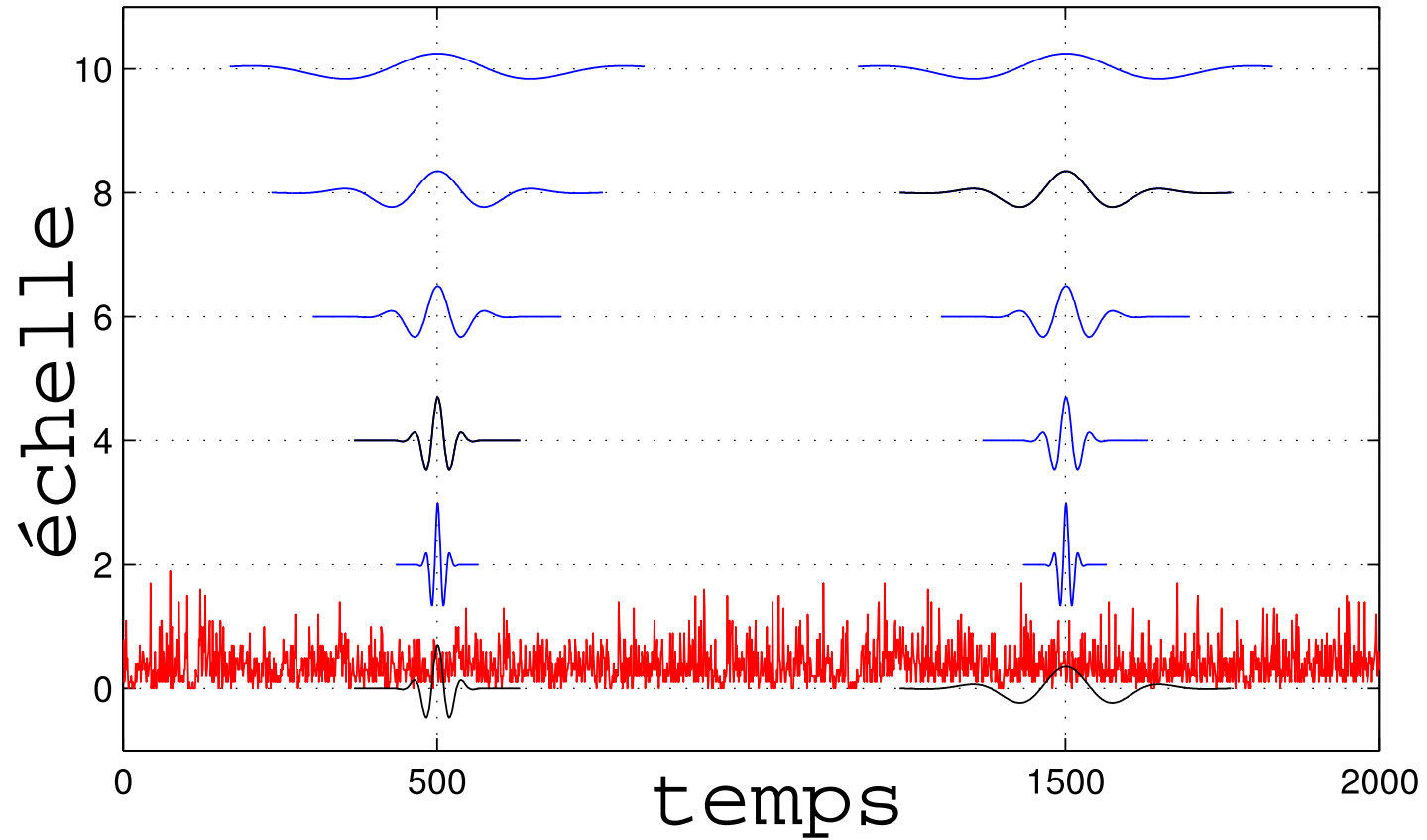
- ▶ Les processus auto-similaires sont des processus LRD (la réciproque est fausse)
- ▶ La LRD s'évalue à partir de la courbe de variance en fonction de l'échelle de temps
- ▶ Quantitativement, la caractéristique d'un processus auto-similaire est la valeur du facteur de Hurst (loi de puissance marquée dans un diagramme $LRD = f(\text{échelle})$)



- ▶ Des outils basés sur des méthodes semi-paramétriques existent :
 - ▶ Periodogramme
 - ▶ Basés sur une analyse spectrale
 - ▶ Basés sur une analyse par ondelettes
(→ LDestimate)

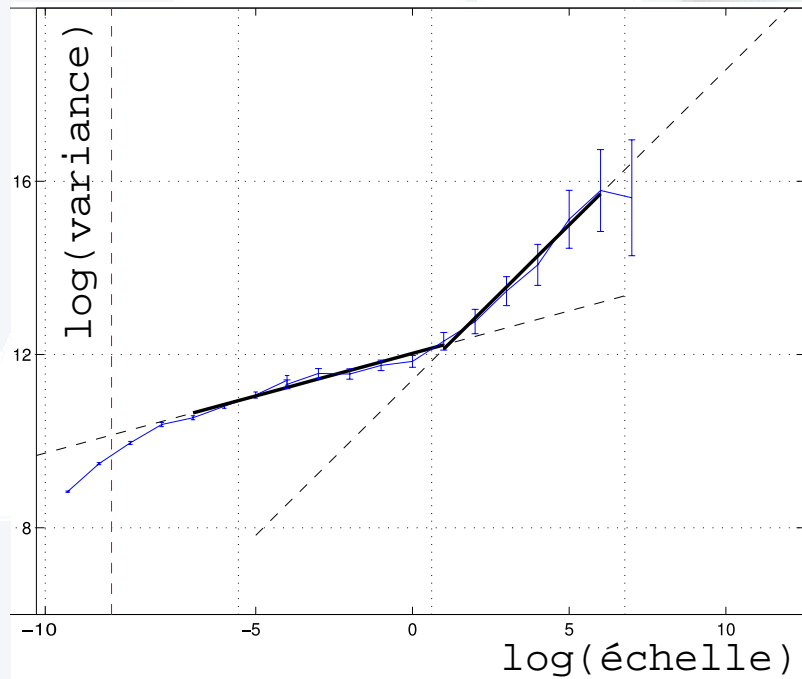


Décomposition en ondelettes





Quantifier l'invariance d'échelle



▶ « loi de puissance = droite dans un diagramme log-log »

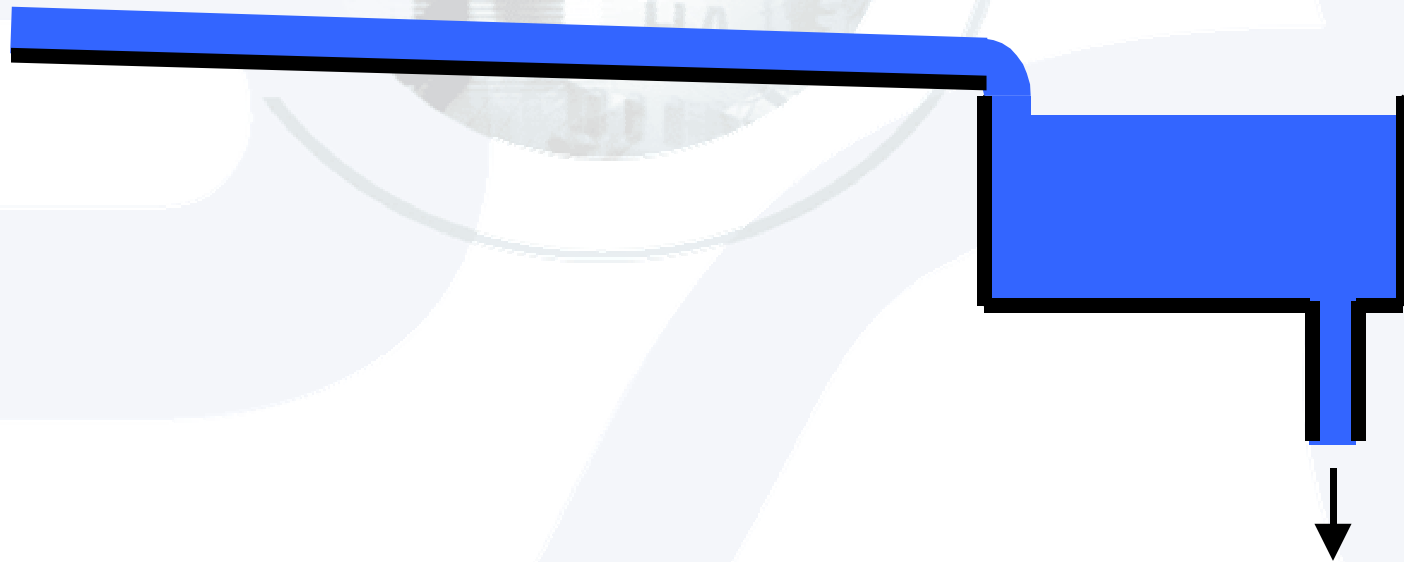
▶ 2 régimes d'invariance d'échelle

▶ Grandes échelles :

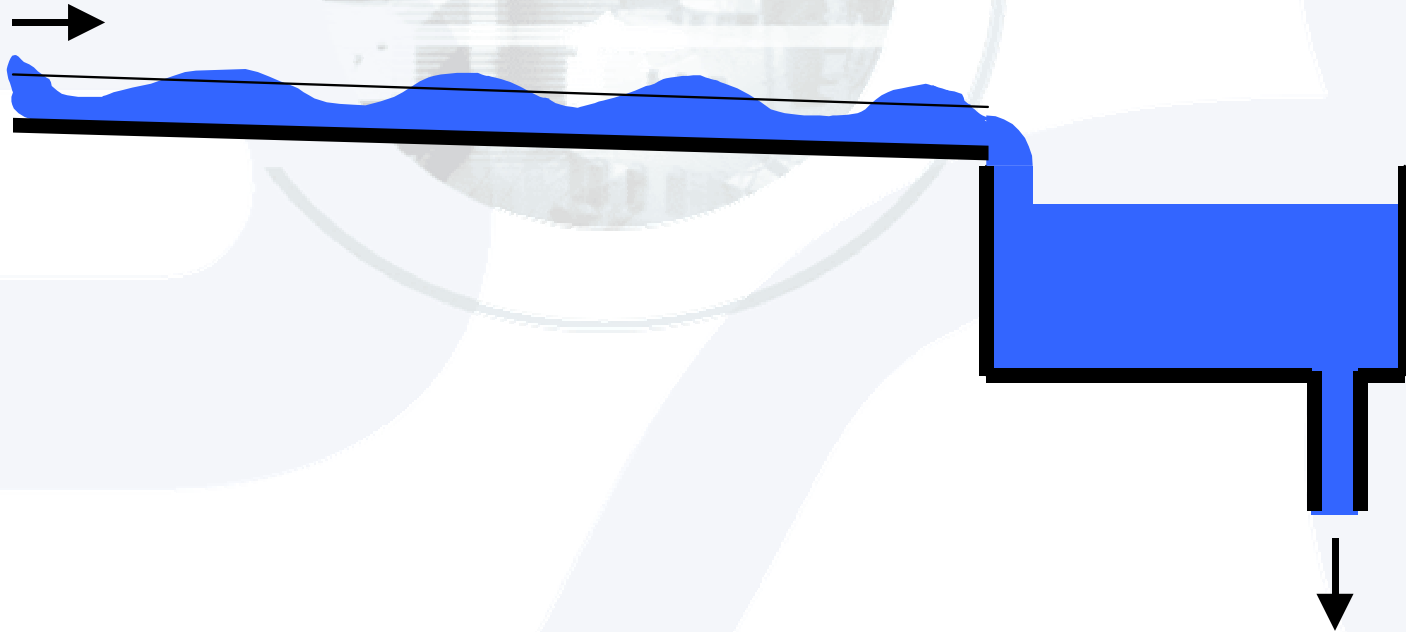
$$H = 0.7$$

(longue mémoire)

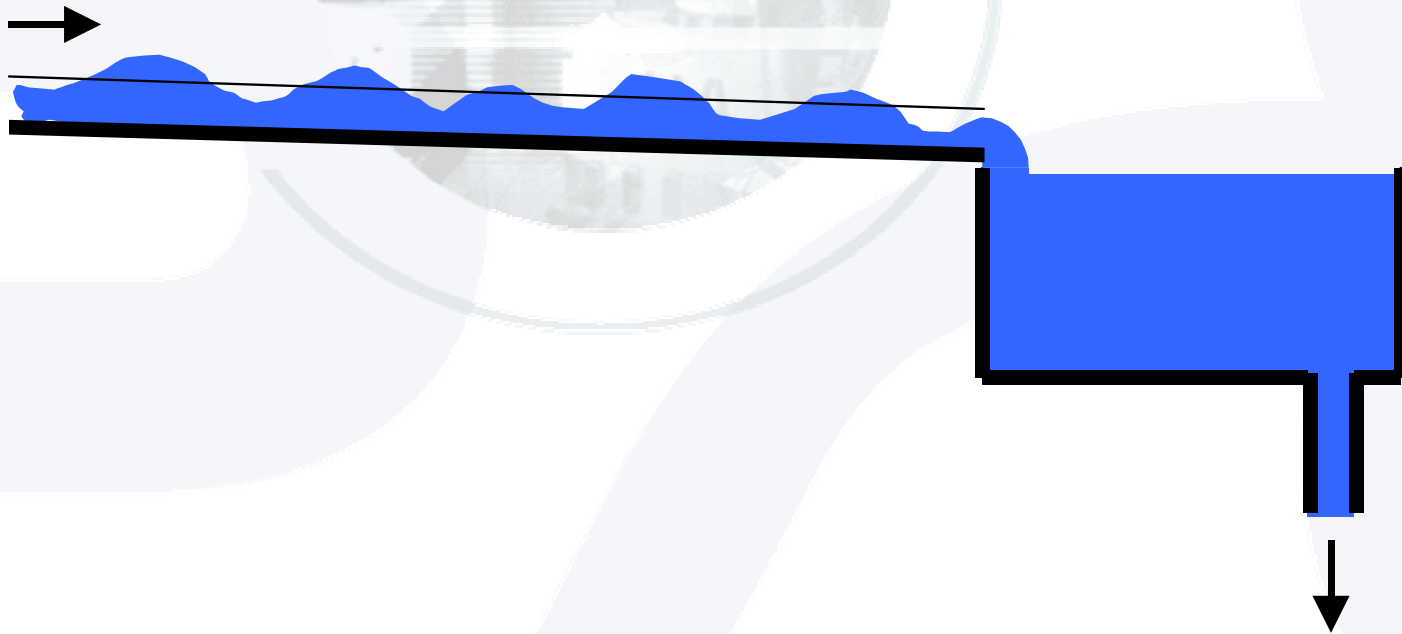
Longue mémoire et pertes



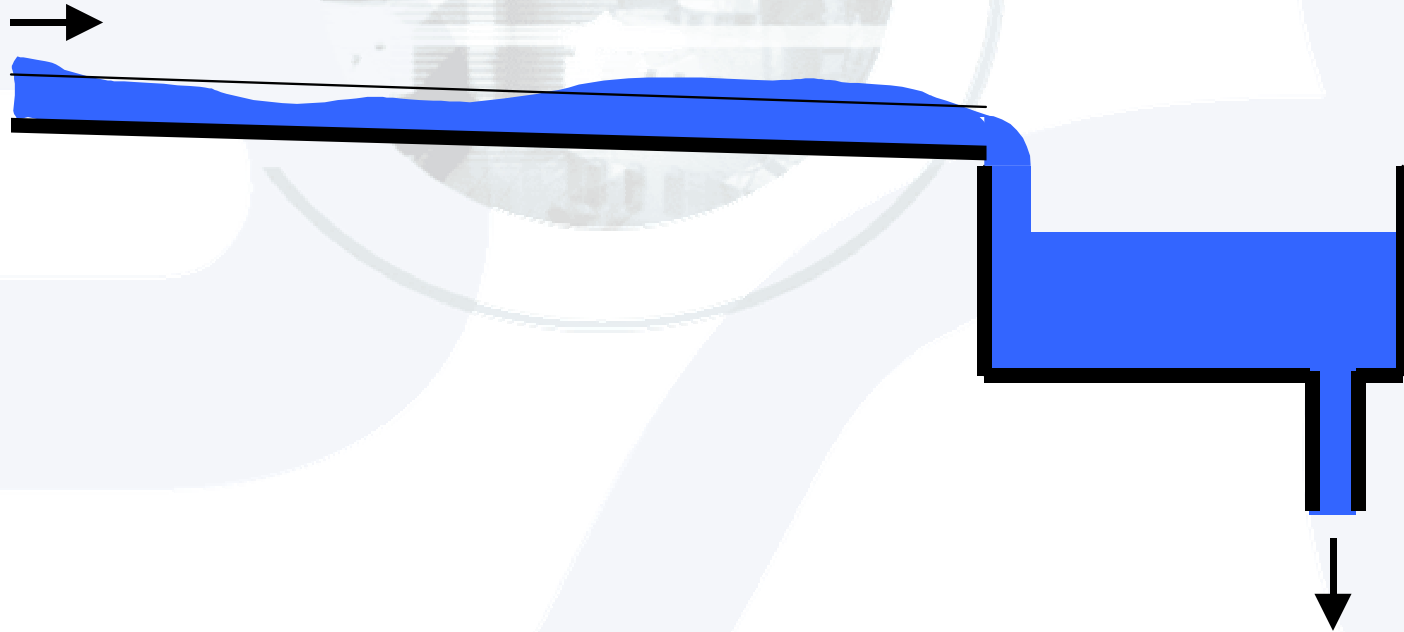
Longue mémoire et pertes



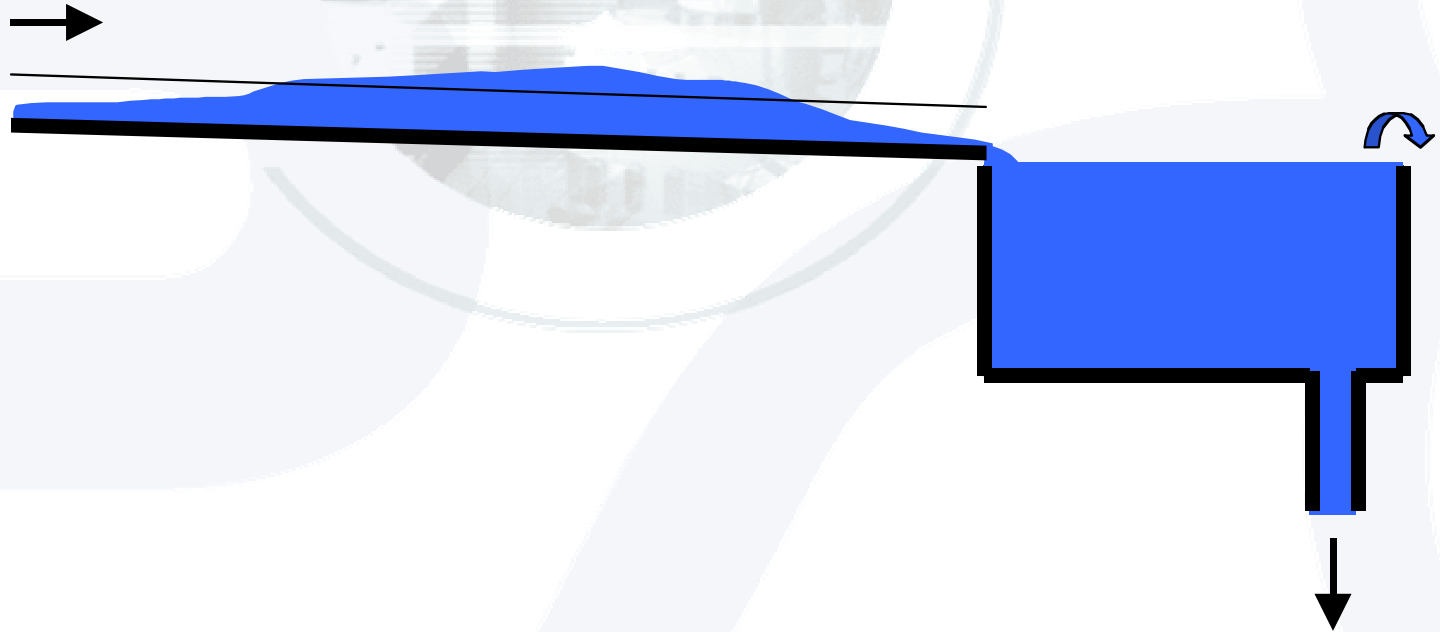
Longue mémoire et pertes



Longue mémoire et pertes



Longue mémoire et pertes

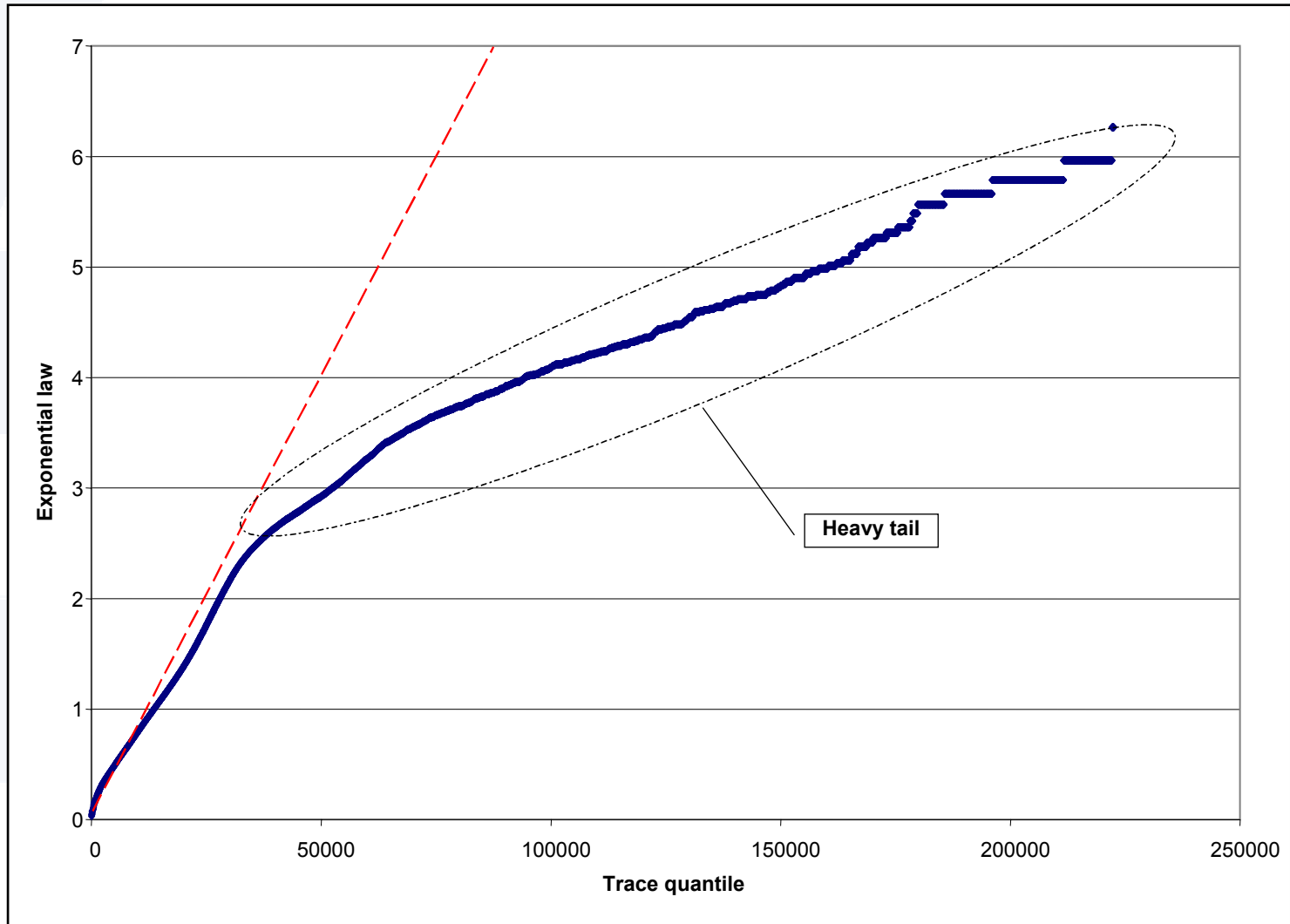




Exemple d'un réseau d'accès

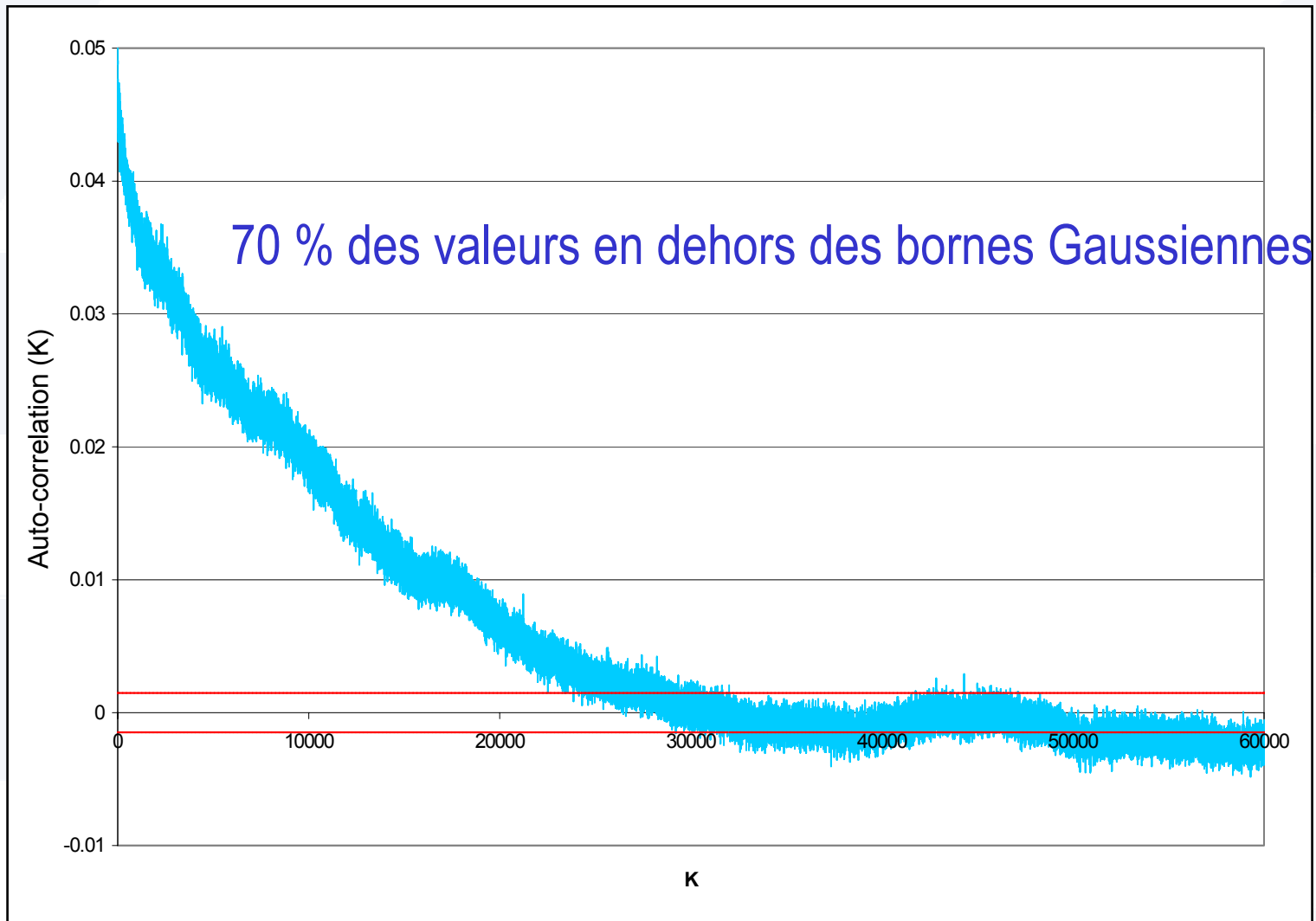


QQ-Plot des arrivées de paquets



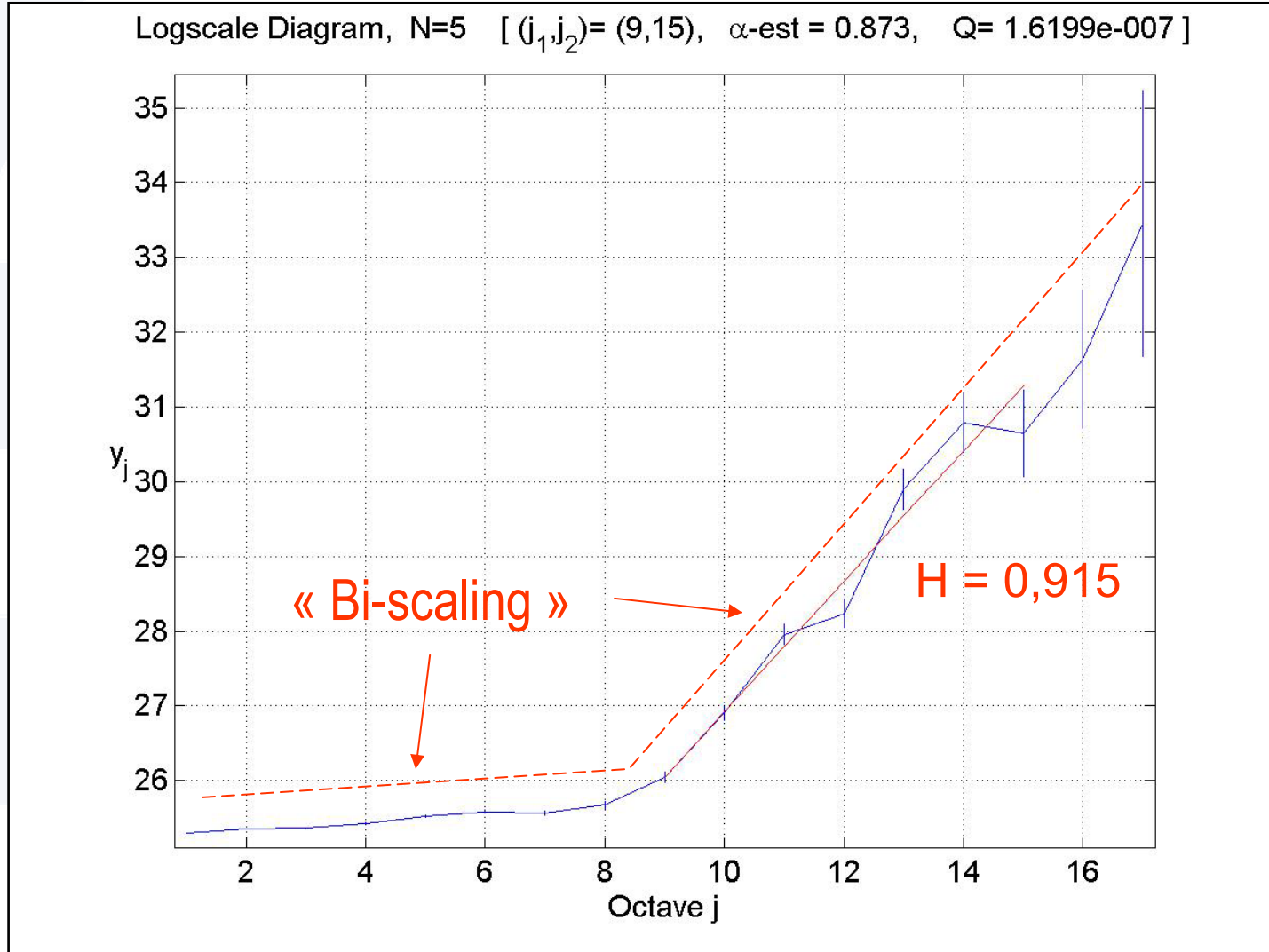


Corrélation des arrivées de paquets





H (LRD) measurements



- ▶ Inter-dépendantes et corrélées
 - Contrôle de congestion de TCP
 - ▶ Dépendance entre fenêtres CWND (même à long terme)
 - ▶ Corrélation due aux pertes

Problème avec les oscillations (2)



- ▶ Leur amplitude augmente :
 - ▶ avec la longueur des éléphants
 - Plus l'éléphants est gros plus CWND peut devenir grand, entraînant des variations d'autant plus fortes en cas de pertes
 - ▶ Avec l'agressivité des protocoles
 - L'objectif est de pouvoir consommer beaucoup de ressources rapidement pour exploiter les énormes capacités des réseaux actuels



- ▶ Les caractéristiques d'auto-similarité et de dépendances longue mémoire dans le trafic sont très difficiles à modéliser
- d'où l'importance de laisser l'environnement de simulation gérer la mise en forme des paquets.
- Les simulateurs devront donc être capables de recréer ces caractéristiques sur le trafic simulé



Le rejeu de traces en simulation

- ▶ Trouver un modèle réaliste pour les flux
 - ▶ Loi d'arrivée
 - ▶ Durée
 - ▶ Taille
 - ▶ Taille des paquets
 - ▶ etc.
- ▶ Créer une topologie créant des conditions réalistes pour la mise en forme des paquets

(NS sera utilisé dans la suite)

Besoin 1 : modèle d'arrivée des flux



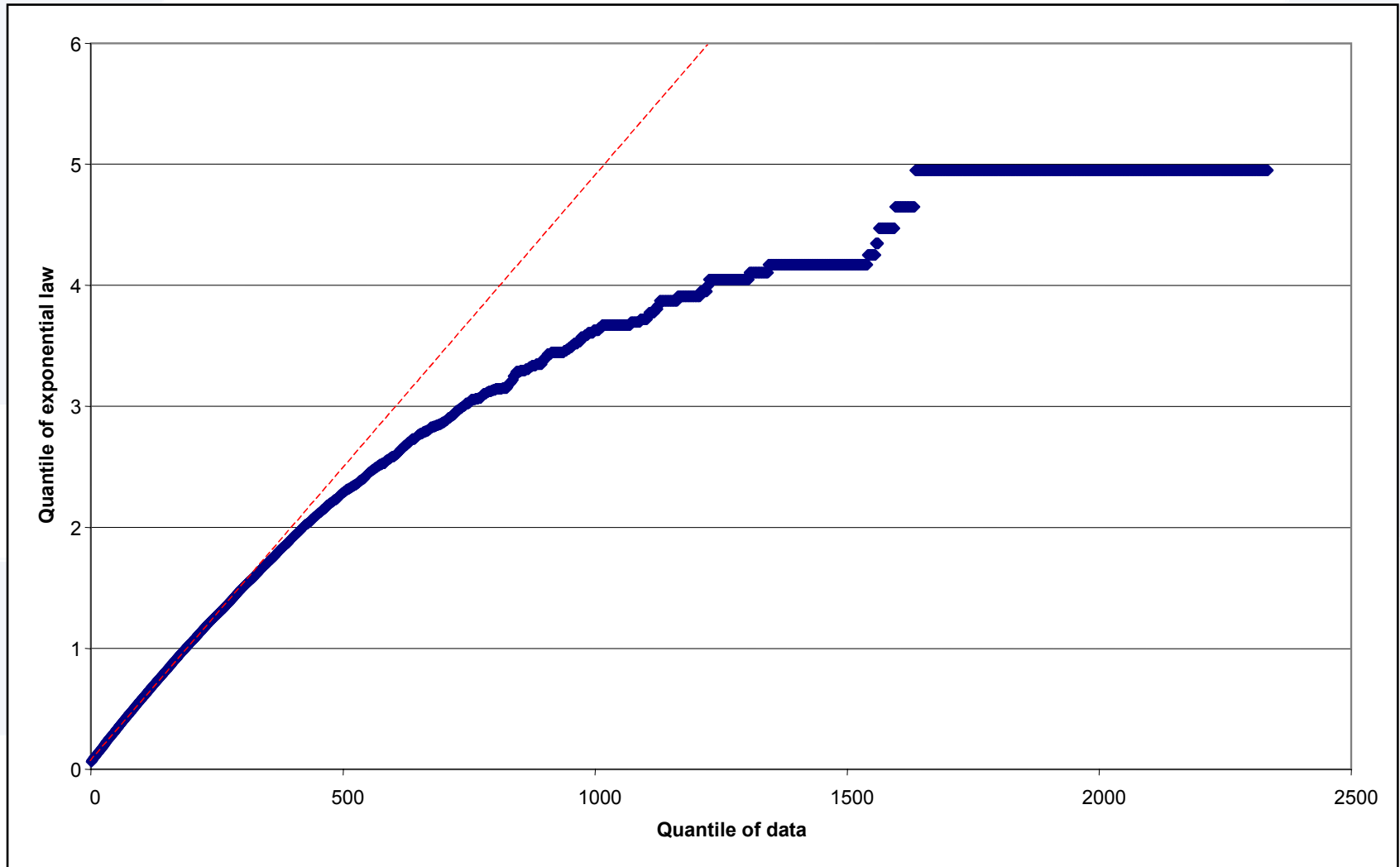
▶ Comme on va simuler du trafic au niveau flux :

→ Besoin d'un modèle d'arrivée des flux

→ Mais quel modèle ?

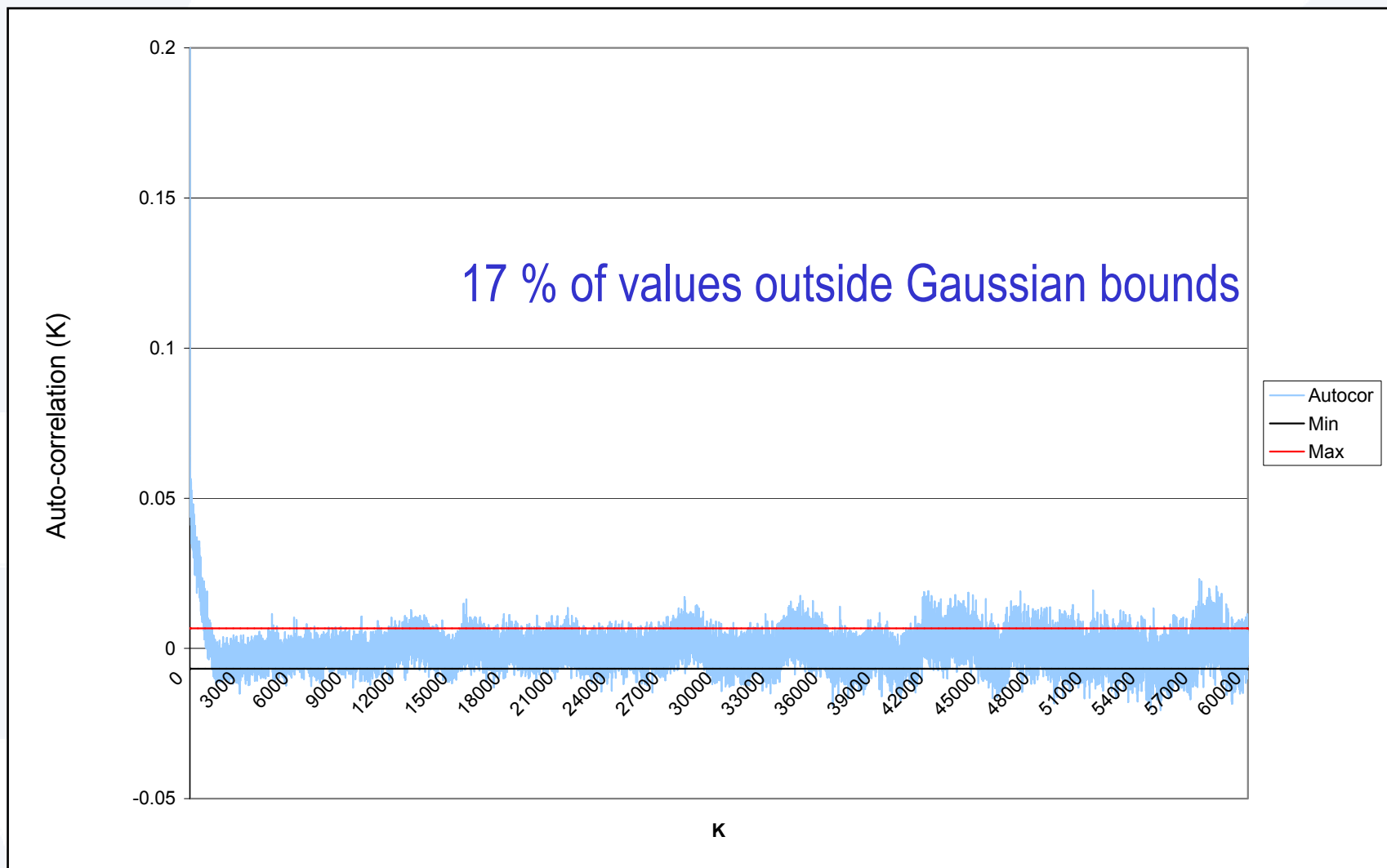


QQ-Plot des arrivées de flux (accès)





Correlation des arrivées de flux (accès)



- ▶ Ce processus d'arrivée des flux ne s'apparente pas à un processus connu (de nous)

- ▶ Rejouer dans le simulateur les flux tels qu'ils apparaissent dans une trace
 - ▶ Respecter les intervalles inter-arrivées des flux
 - ▶ Respecter le nombre et la taille des paquets du flux
 - ▶ Respecter le protocole initial
 - UDP : respecter les dates (relatives) d'émission des paquets du flux
 - TCP : l'agent TCP new Reno de NS fait le shaping des paquets

Besoin 2 : topologie adaptée

- ▶ Importance capitale car elle conditionne le shaping des paquets des flux
- ▶ 2 approches vont être proposées pour respecter 2 paramètres essentiels du fonctionnement de TCP
 - ▶ Les RTT des connexions TCP
 - ▶ Les taux de pertes

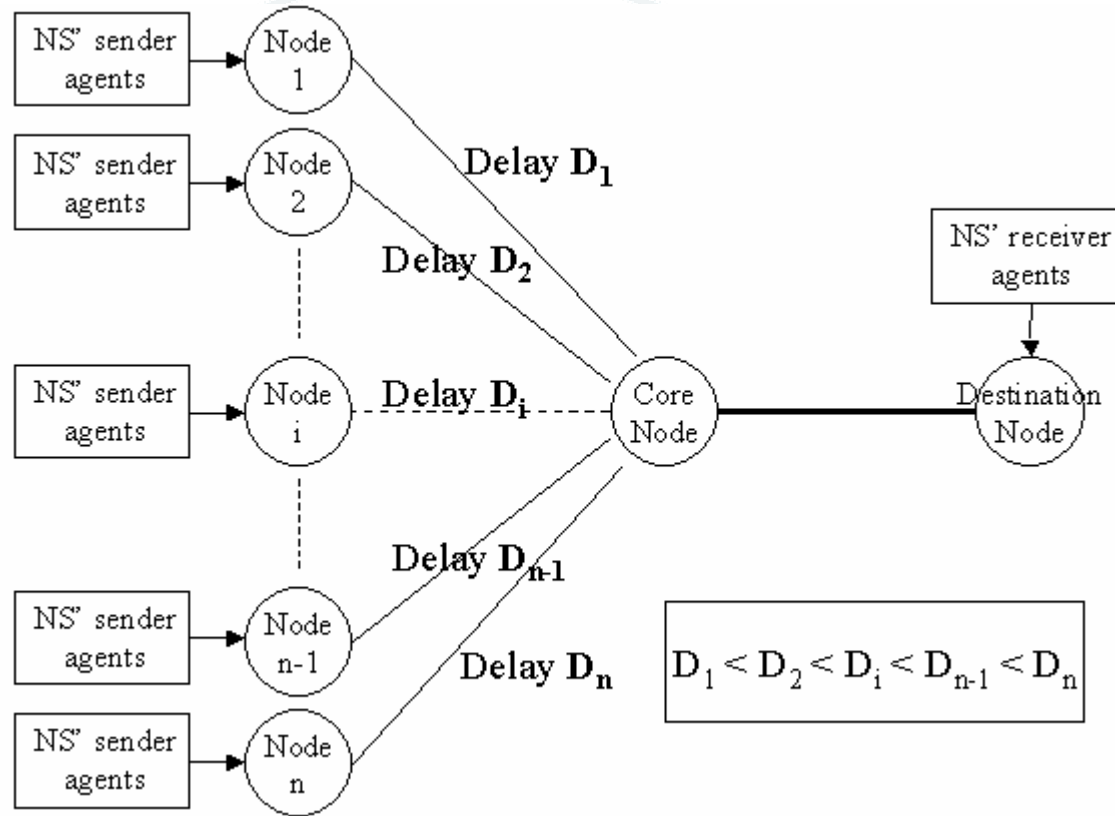
(chaque fois, une topologie minimale sera proposée)

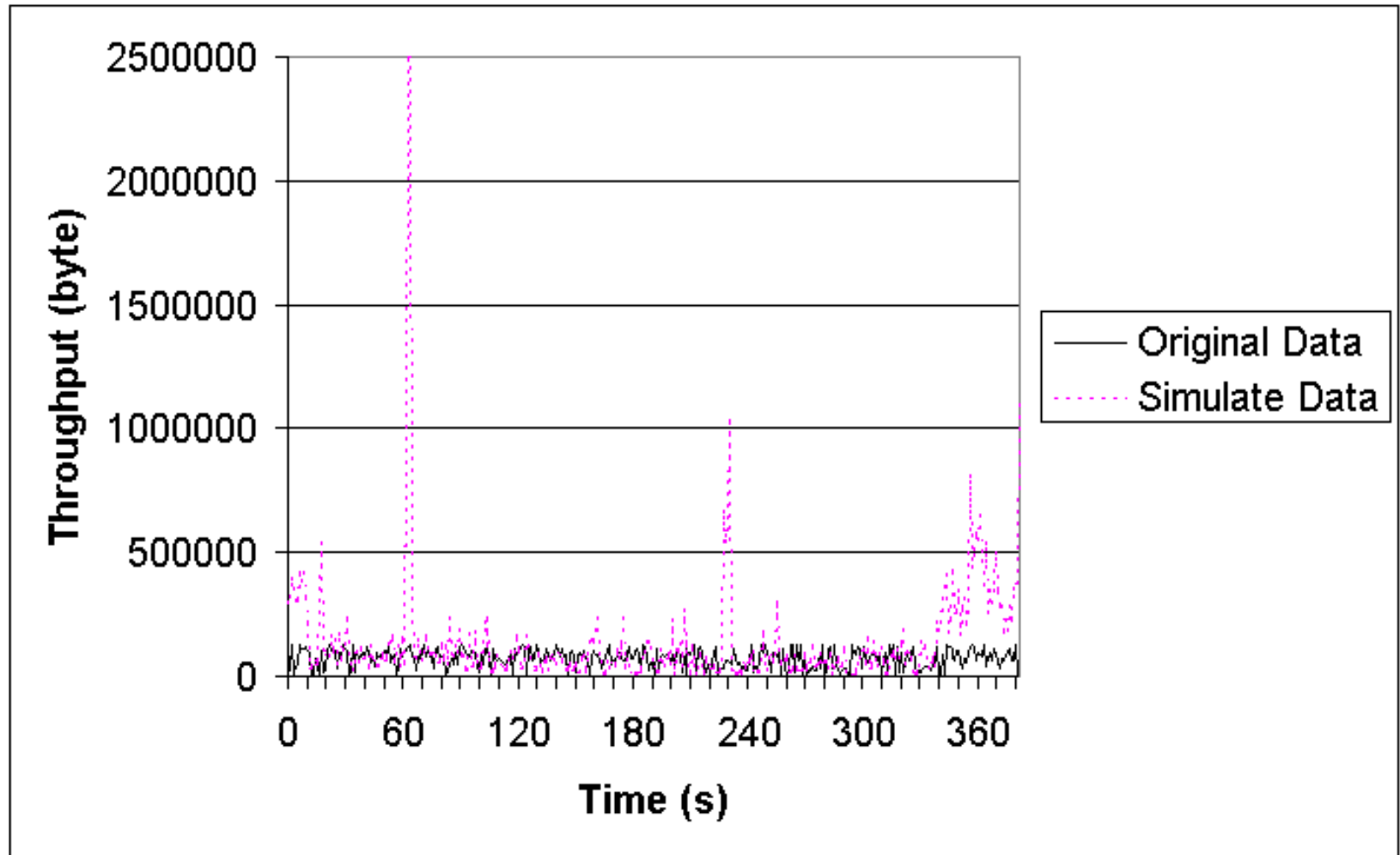


Topologie 1 : construite sur les RTT



Topologie « RTT »





- ▶ Pour les comparer, on essaie de les faire correspondre à une distribution Weibull

$$f(T) = \frac{\beta}{\eta} \left(\frac{T}{\eta} \right)^{\beta-1} e^{-\left(\frac{T}{\eta} \right)^{\beta}}$$

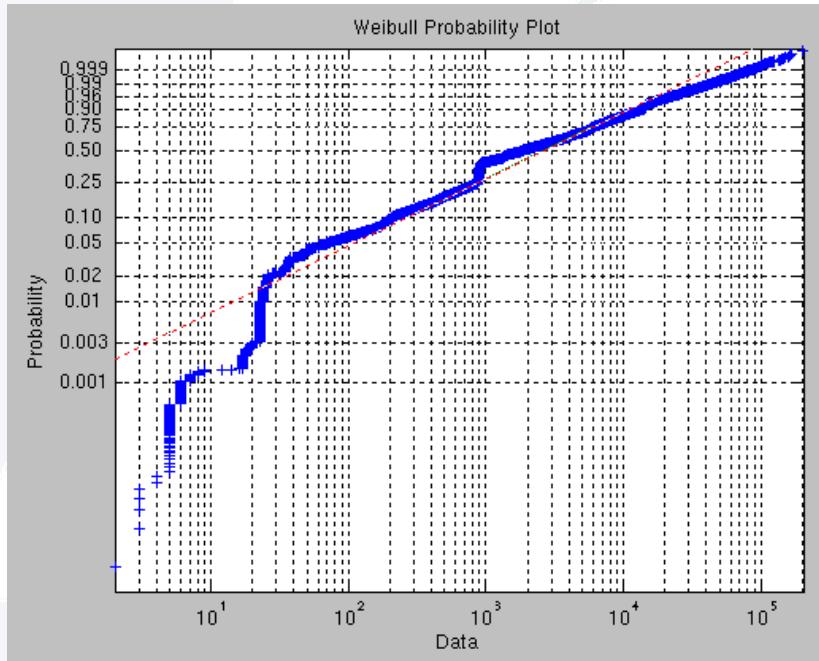
où: $f(T) \geq 0, T \geq 0$ _ ou _ $\beta > 0, \eta > 0$

$\eta =$ paramètre d'échelle

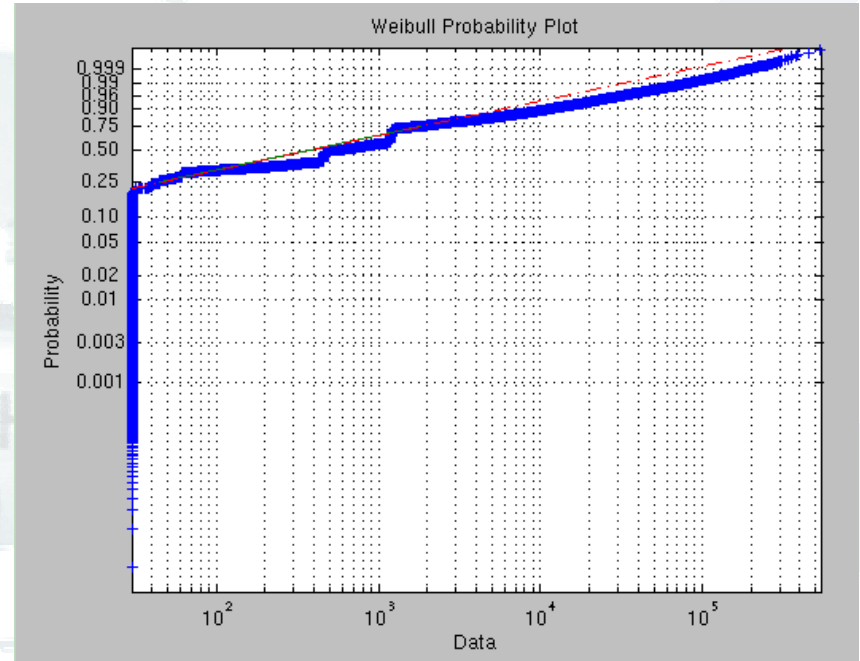
$\beta =$ paramètre de forme (ou pente)



Inter-arrivées des paquets



Trace



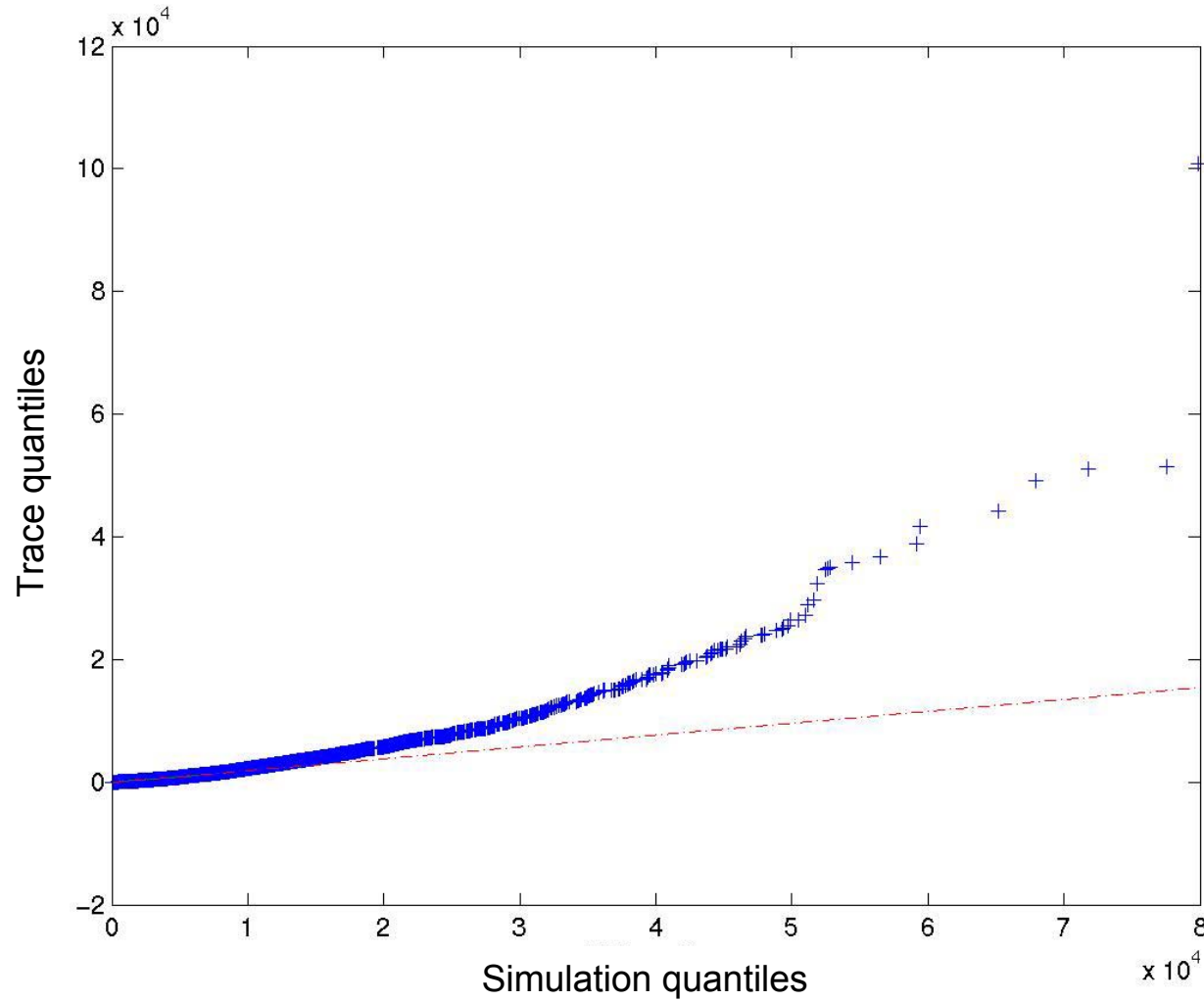
Simulation



- + les deux courbes correspondent à des distributions Weibull
- les paramètres ne sont pas les mêmes

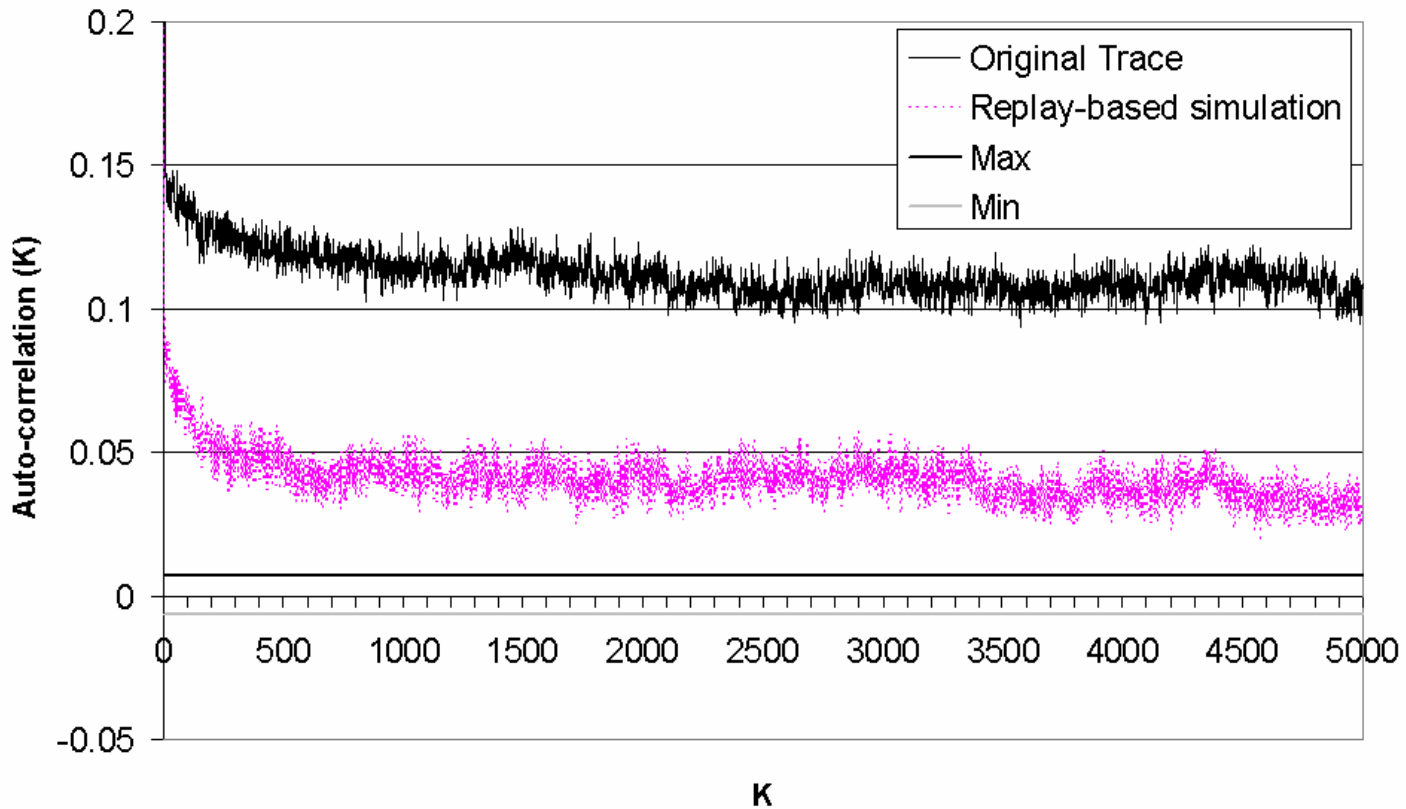


QQ-plot des arrivées de paquets





Corrélation des arrivées de paquets

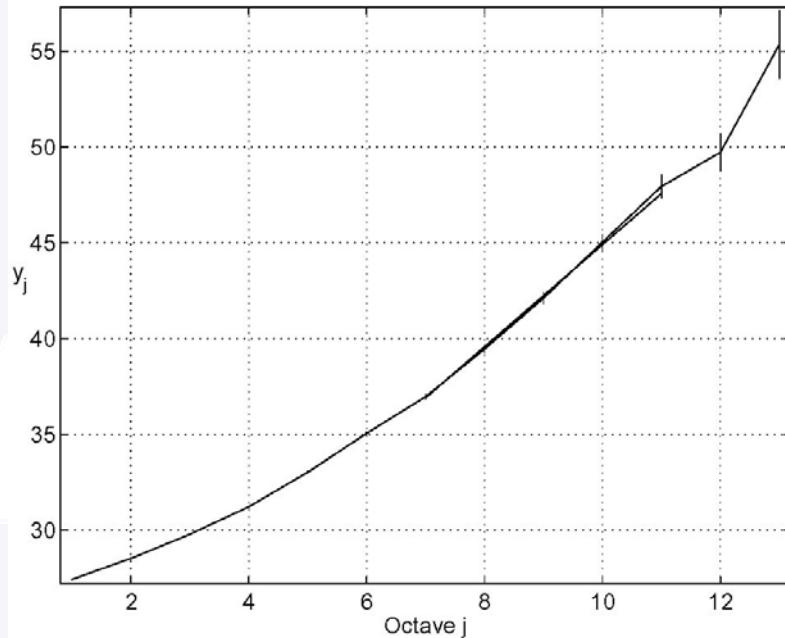




Diagrammes « Ldestimate »

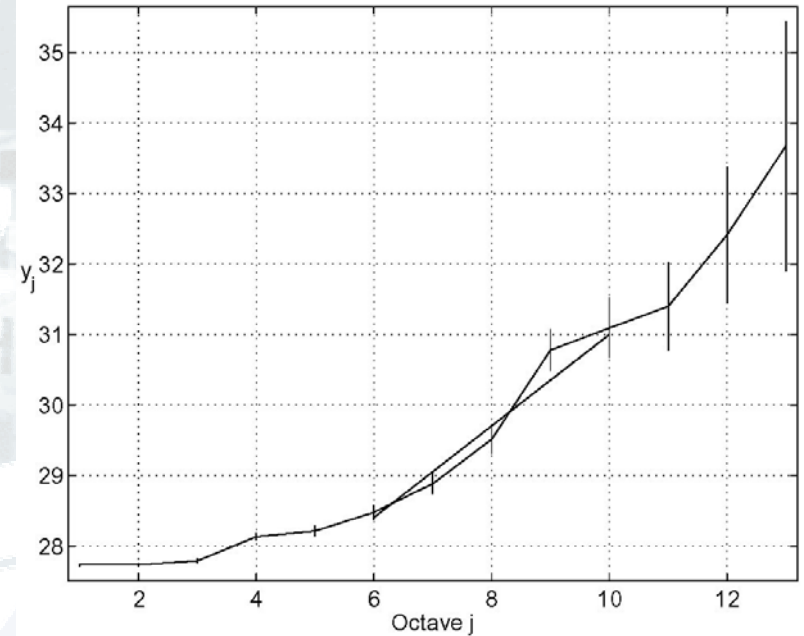


Logscale Diagram, N=3 [(j₁,j₂)= (7,11), α-est = 2.66, Q= 0.23715]



Trace : H=0,87

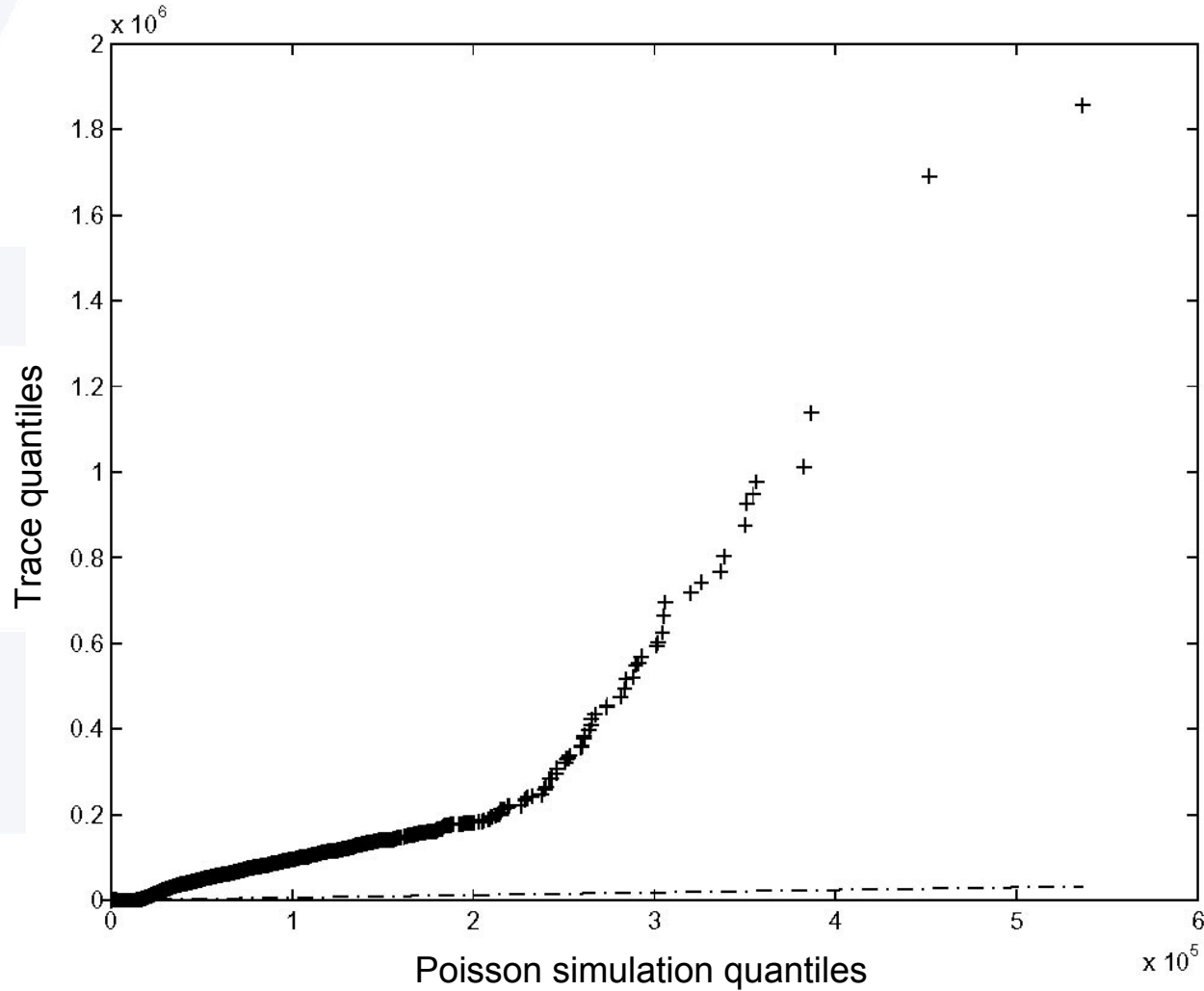
Logscale Diagram, N=3 [(j₁,j₂)= (6,10), α-est = 0.649, Q= 0.00041134]



Simul : H=0,65

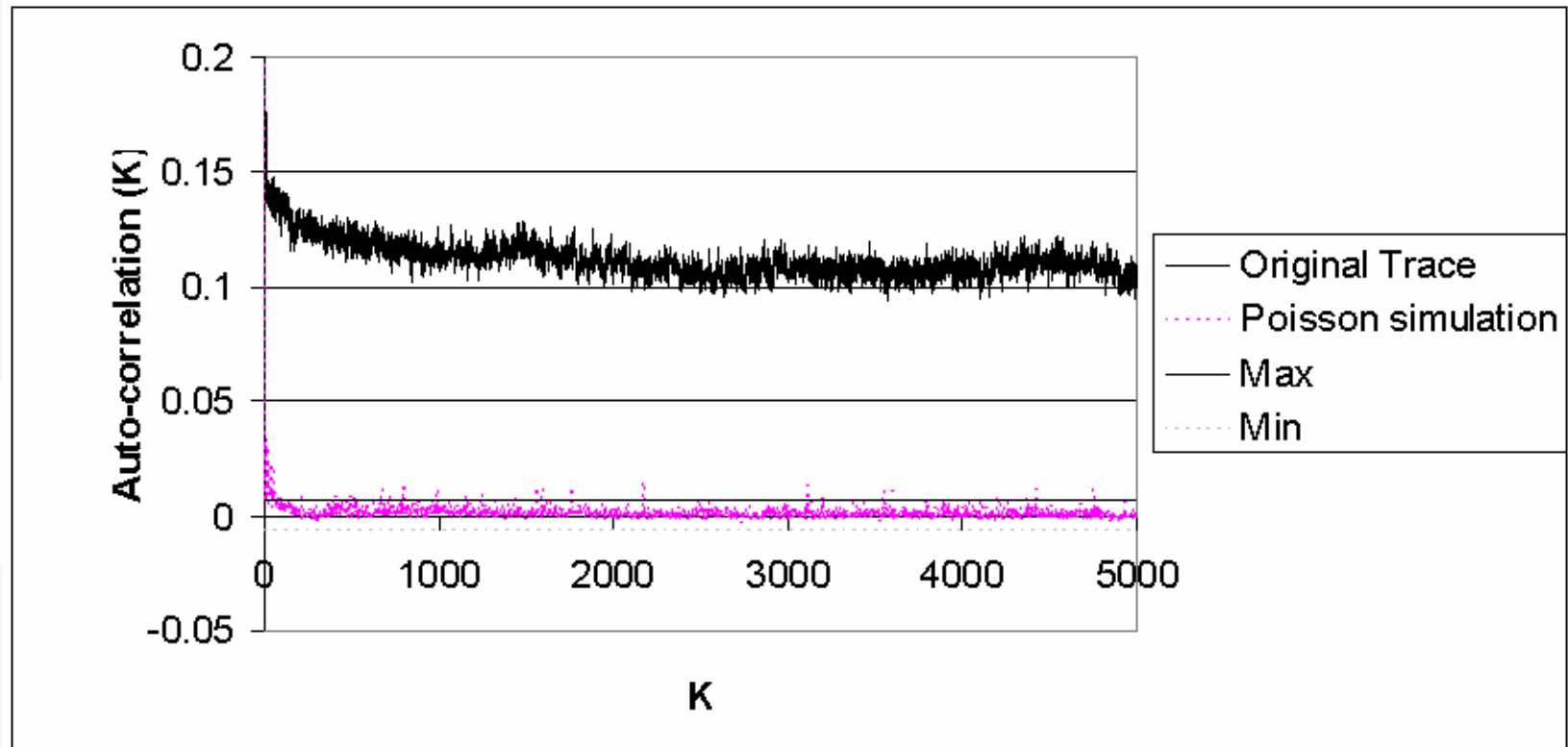
Rejeu vs. Poisson

QQ-plot des arrivées de paquets





Rejeu vs. Poisson





- ▶ La méthode basée sur le rejeu des flux permet de retrouver les caractéristiques qualitative de la trace originale
- ▶ Les résultats sont bien meilleurs que pour des simulations avec des flux Poissonniens

Mais

- ▶ Quantitativement on a des limitations et donc un trafic simulé moins complexe que le trafic original



Topologie 2: construite sur les taux de pertes

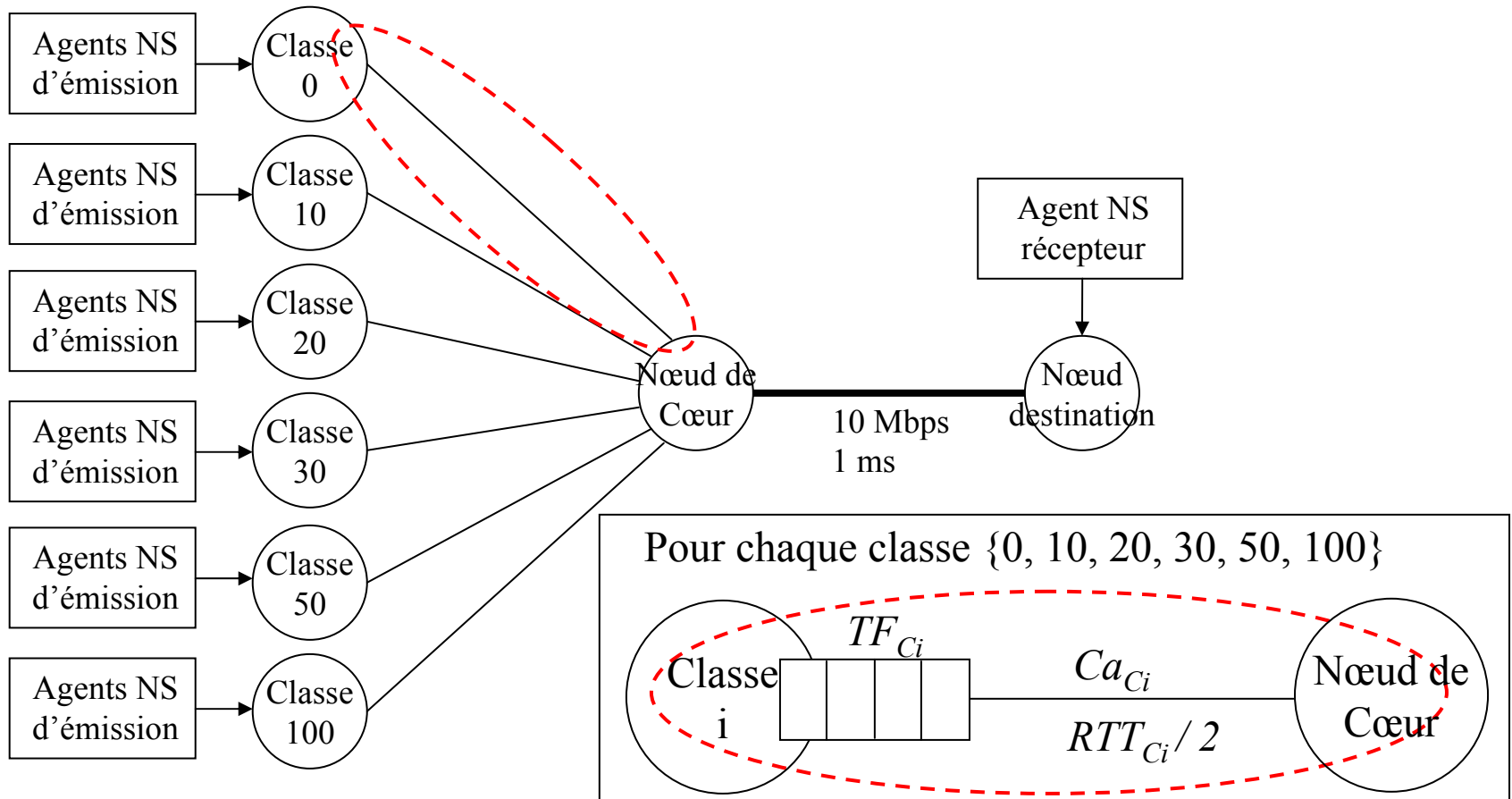


Classes de pertes

classe	Taux de perte des flux de la classe (%)
C0	0
C10	0-10
C20	10-20
C30	20-30
C50	30-50
C100	50-100



Topologie « taux de pertes »





► Capacité du lien

$$Ca_{Ci} = \frac{\sum_{n=1}^{N_{flux}} d_i \times D_i}{d_{trace}}$$

où : N_{flux} = nombre _ de _ flux _ appartenant _ à _ la _ classe _ i

d_i = durée _ du _ flux _ i

D_i = débit _ moyen _ du _ flux _ i

d_{trace} = durée _ de _ la _ trace



- ▶ Taille des files d'attente:

$$TF_{Ci} = Ca_{Ci} \times (100 - \text{taux}_{\text{perte}})$$

- ▶ $RTT_i = \text{moyenne}(RTT_{Ci})$

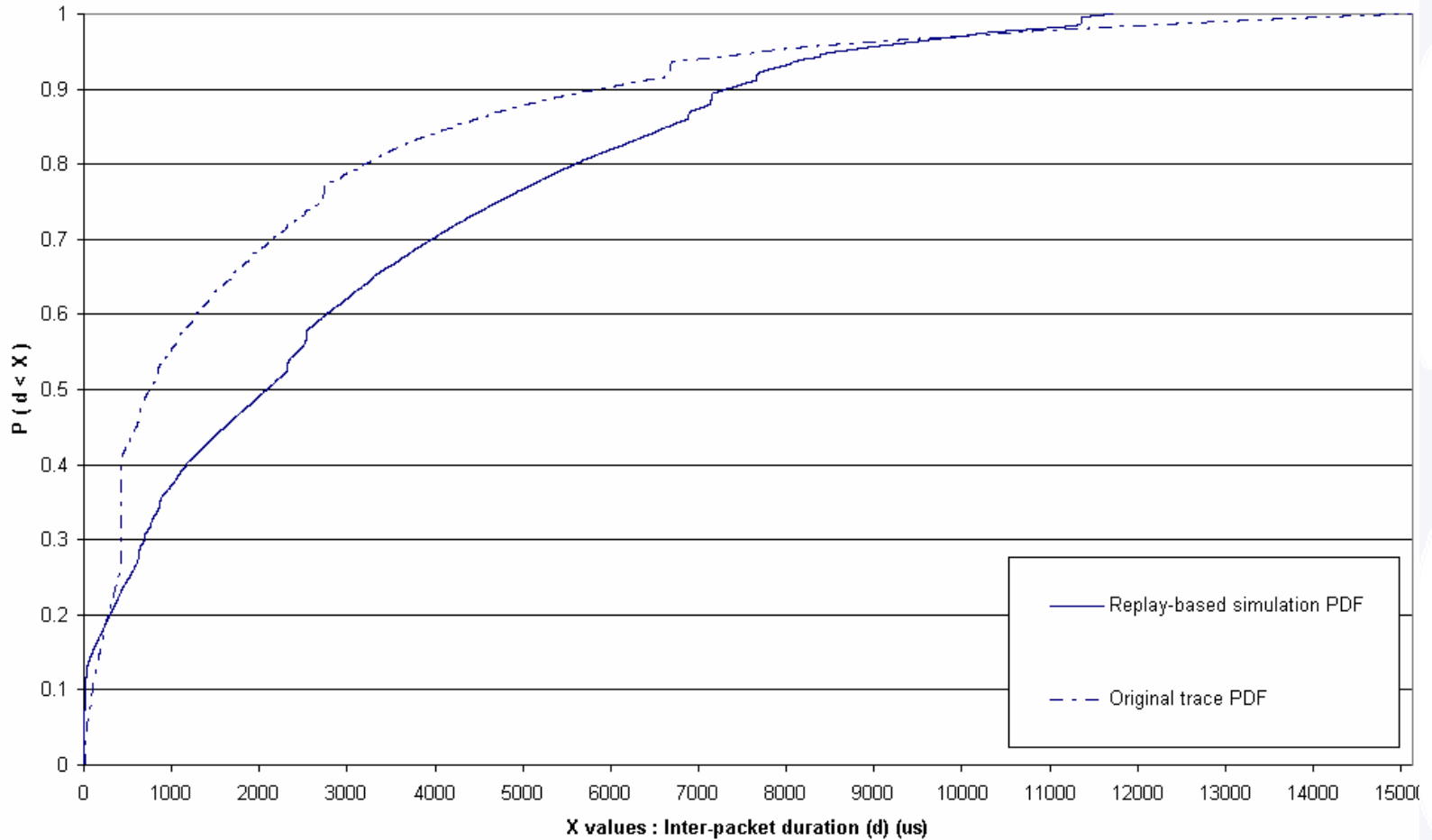


Attributs calculés

classe	Ca_{ci} (Mbps)	TF_{ci} (Octets)	RTT_{ci} (ms)	Nb de flux TCP dans la trace
C0	0,928802	116101	454	3742
C10	0,522106	65264	475	428
C20	0,244047	27456	380	1844
C30	0,053694	5370	711	340
C50	0,076173	6666	555	1425
C100	0,072219	4514	570	698

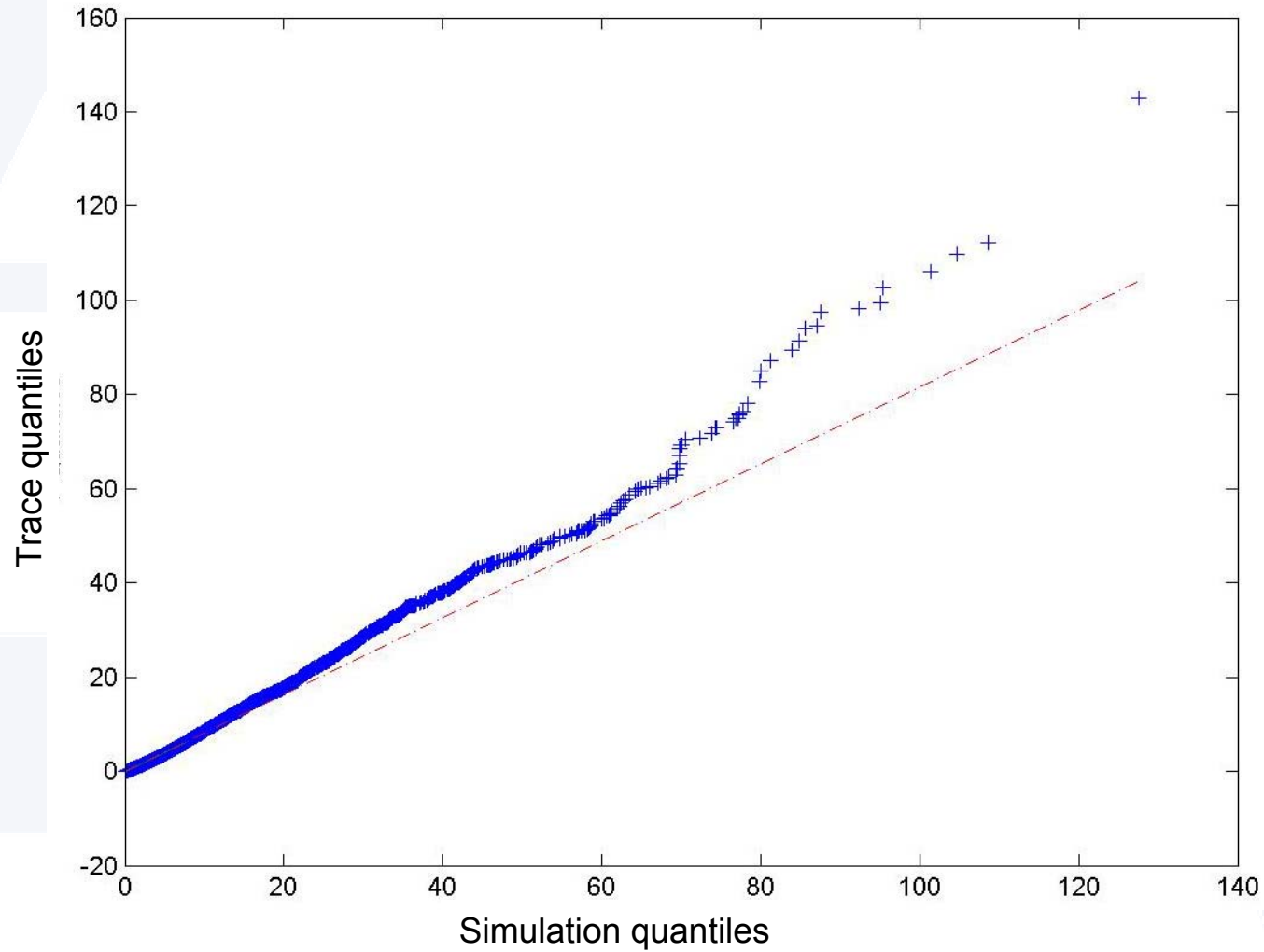


CDF des inter-arrivées de paquets



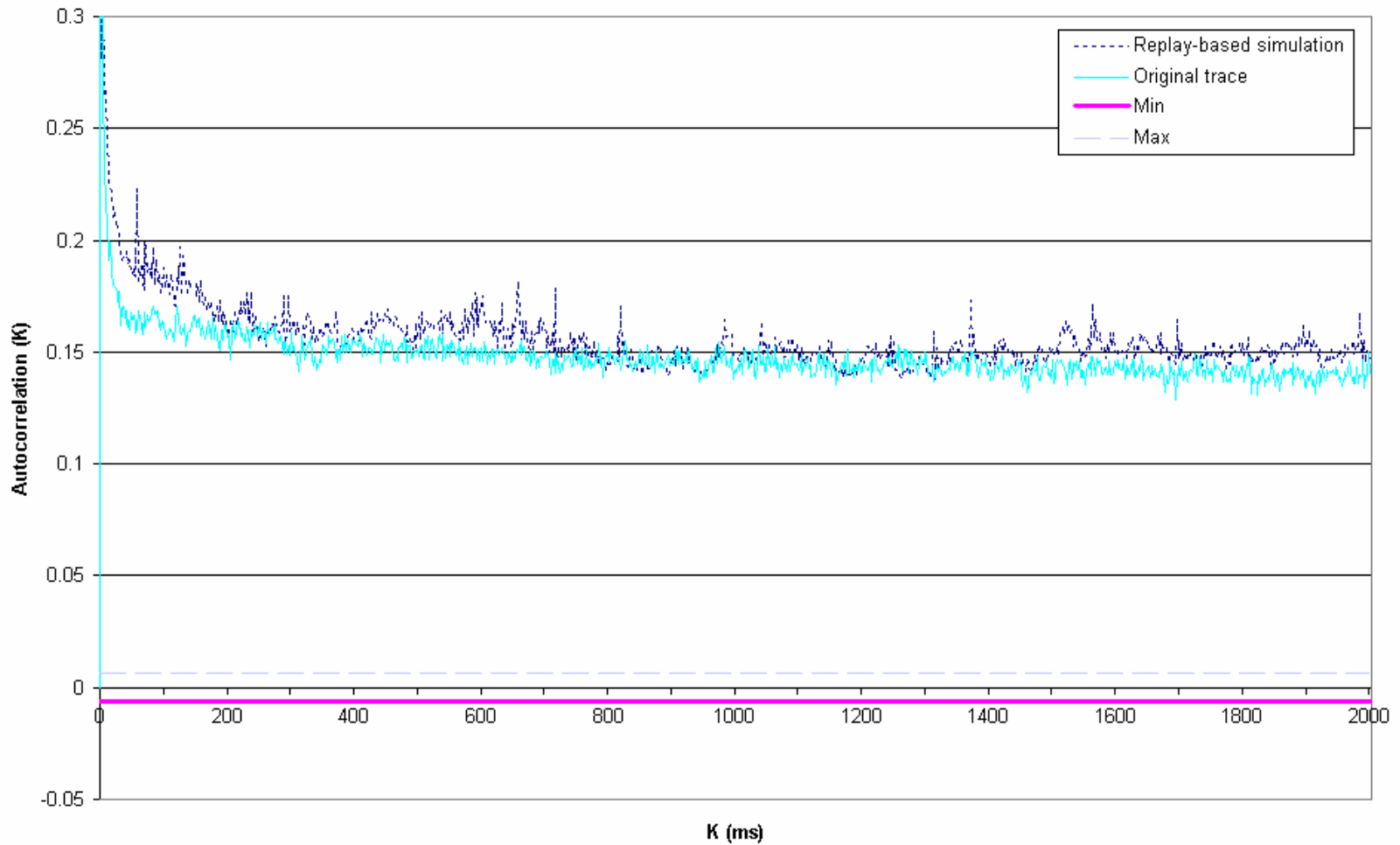


QQ-plot inter-arrivées de paquets

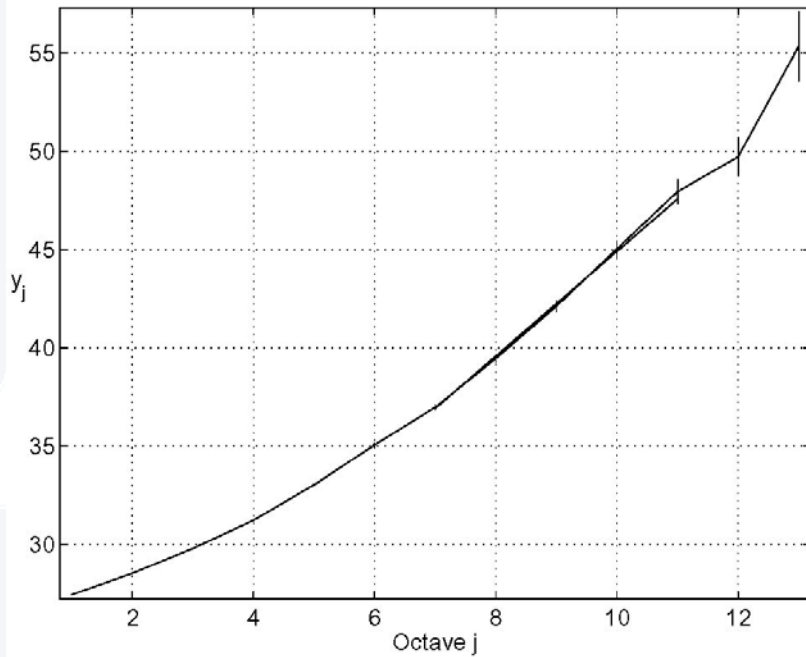




Auto-corrélation des paquets

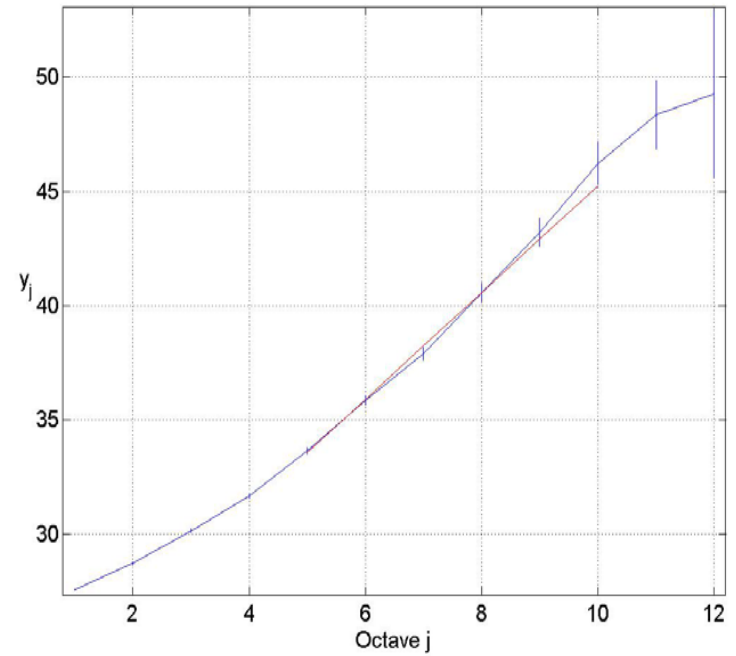


Logscale Diagram, N=3 [(j₁,j₂)=(7,11), α-est = 2.66, Q= 0.23715]



Trace : H=0,87

Logscale Diagram, N=3 [(j₁,j₂)=(5,10), α-est = 2.33, Q= 0.019919]



Simul : H=0,86

Bilan sur la topologie « Tx pertes »



- ▶ Très bons résultats qualitatifs et quantitatifs
- ▶ Les pertes sont bien à la source des phénomènes de dépendance

Mais

- ▶ Lissage des distributions inter-arrivées à cause de la moyenne prise sur les RTT (on fait disparaître les petits RTT)



Prospective



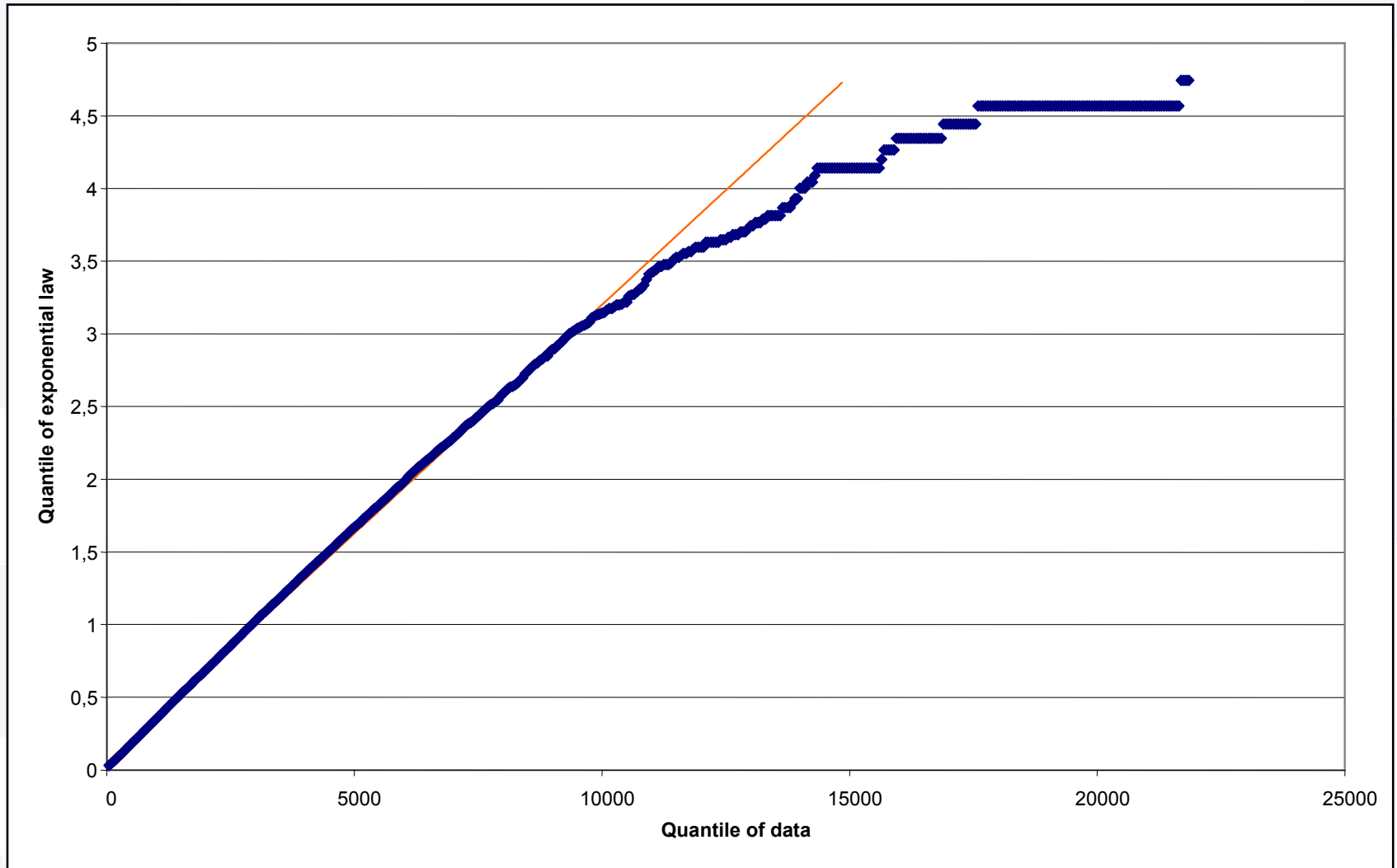
- ▶ Le réalisme obtenu en rejouant des traces est impressionnant (bien meilleur qu'avec des approches classiques)
 - ▶ Mais les derniers % de précision sont les plus difficiles à obtenir
- Les conditions de la simulation restent éloignées de la réalité !

1ère lacune observée

- ▶ Les propriétés du trafic considérées comme invariantes ne le sont pas forcément
- Le trafic de cœur du réseau n'a pas les mêmes propriétés que le trafic de bordure
- Le trafic de cœur est (presque) Poissonnien

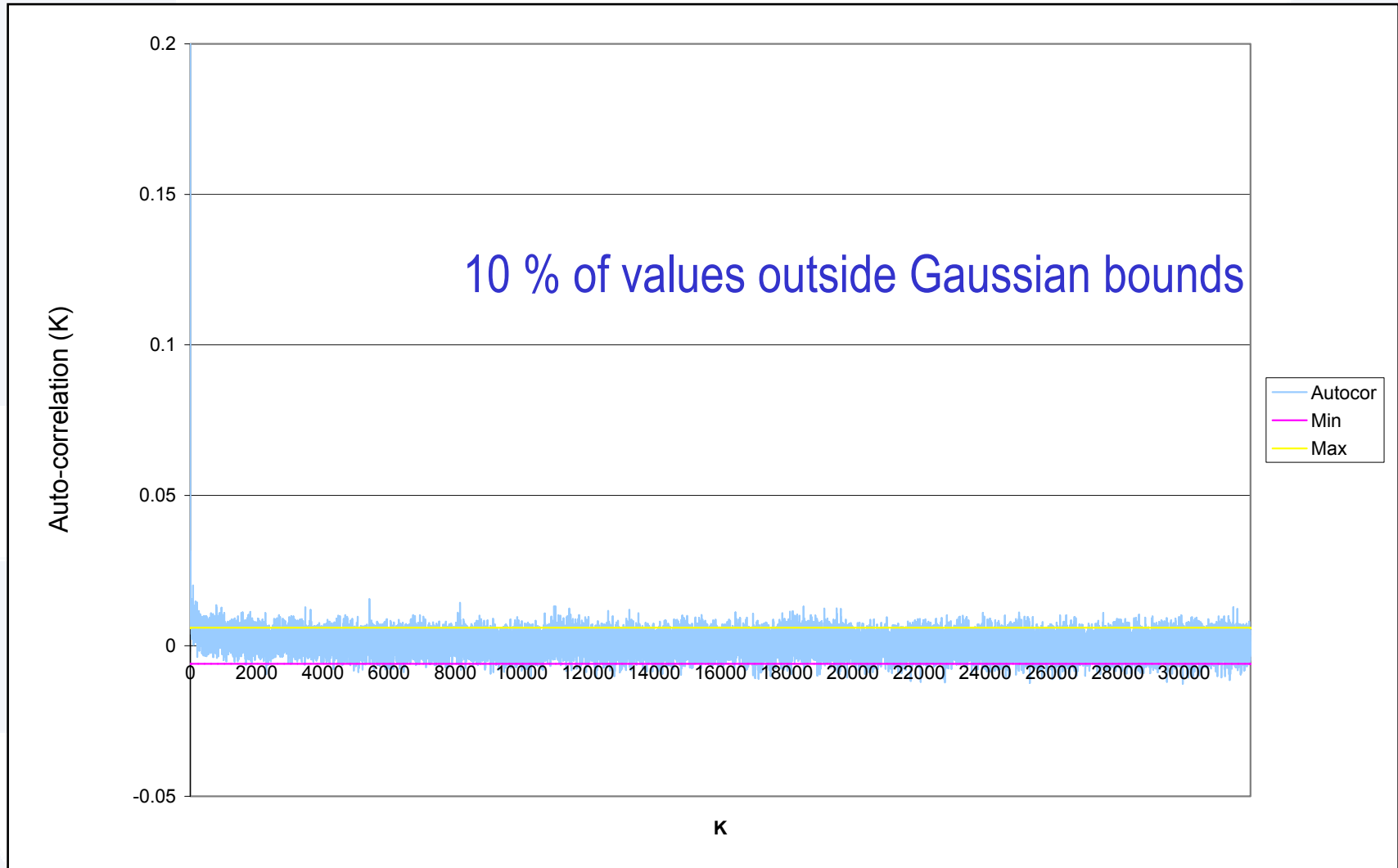


QQ-Plot des arrivées de flux



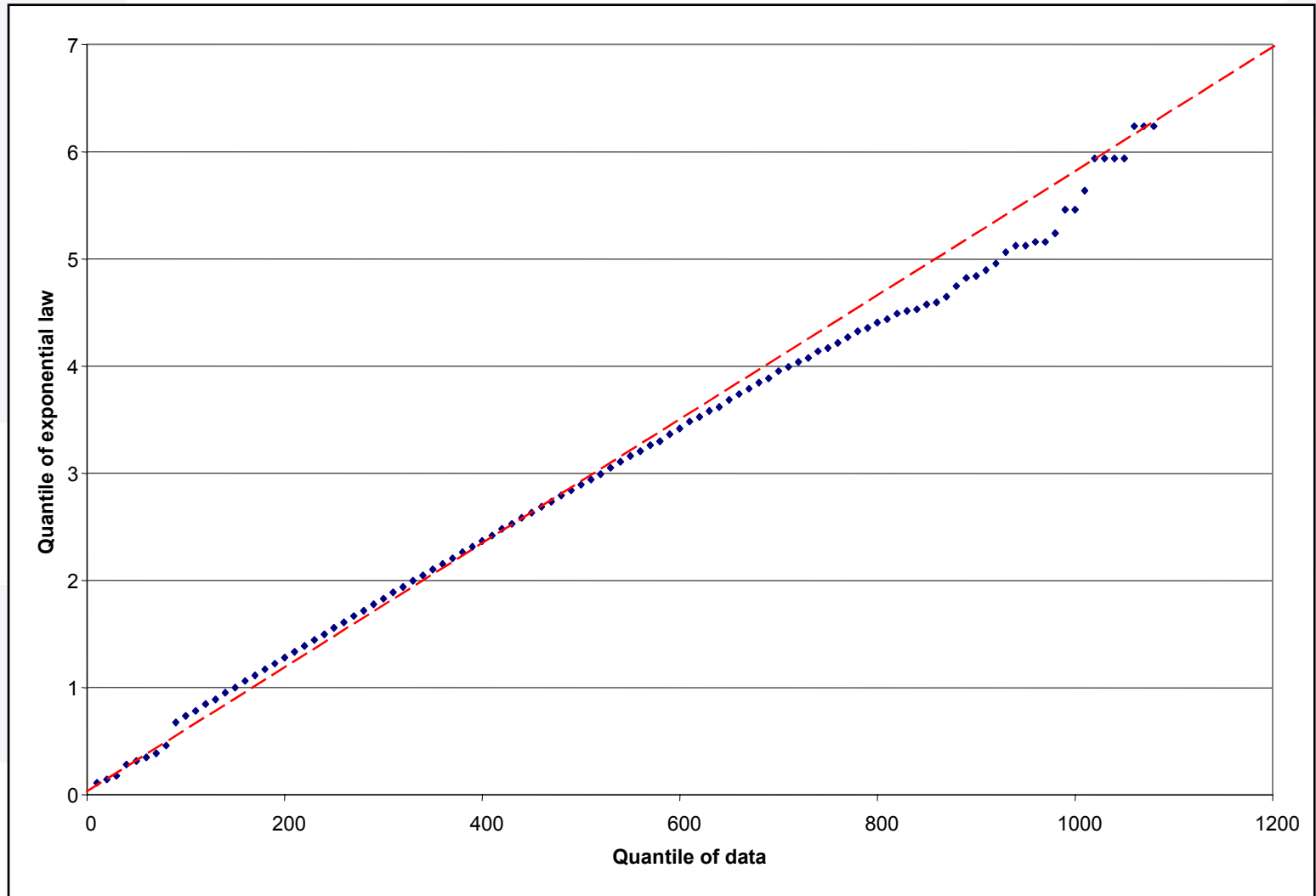


Corrélation des arrivées de flux



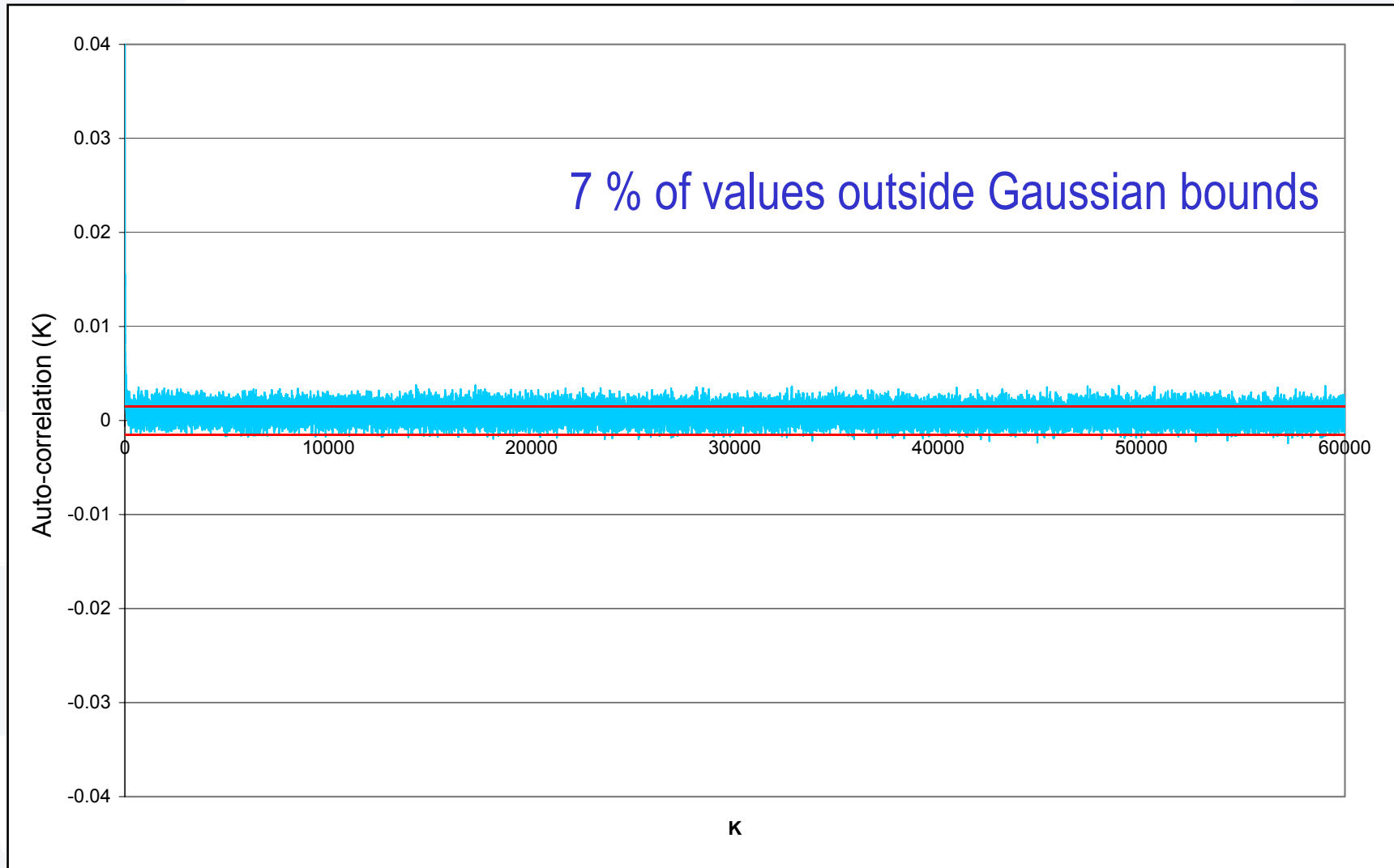


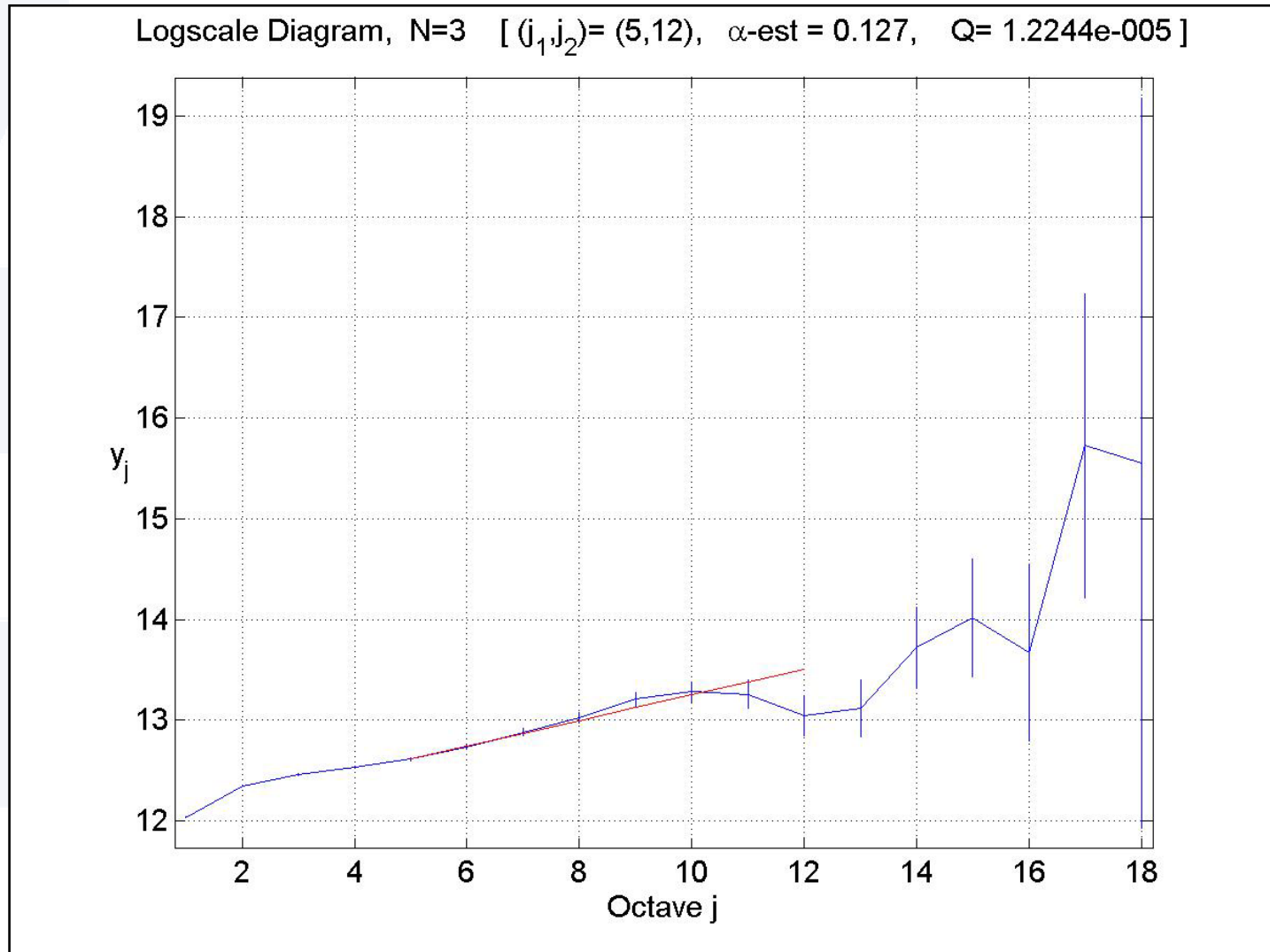
QQ-plot des arrivées de paquets





Corrélation des arrivées de paquets







	Trafic de bordure	Trafic de cœur
Paramètre de Hurst (H)	$H = 0,915$ [0.868, 0.962]	$H = 0,561$ [0.556, 0.565]

- Le trafic de bordure est très complexe
- Le trafic de cœur est plus lisse

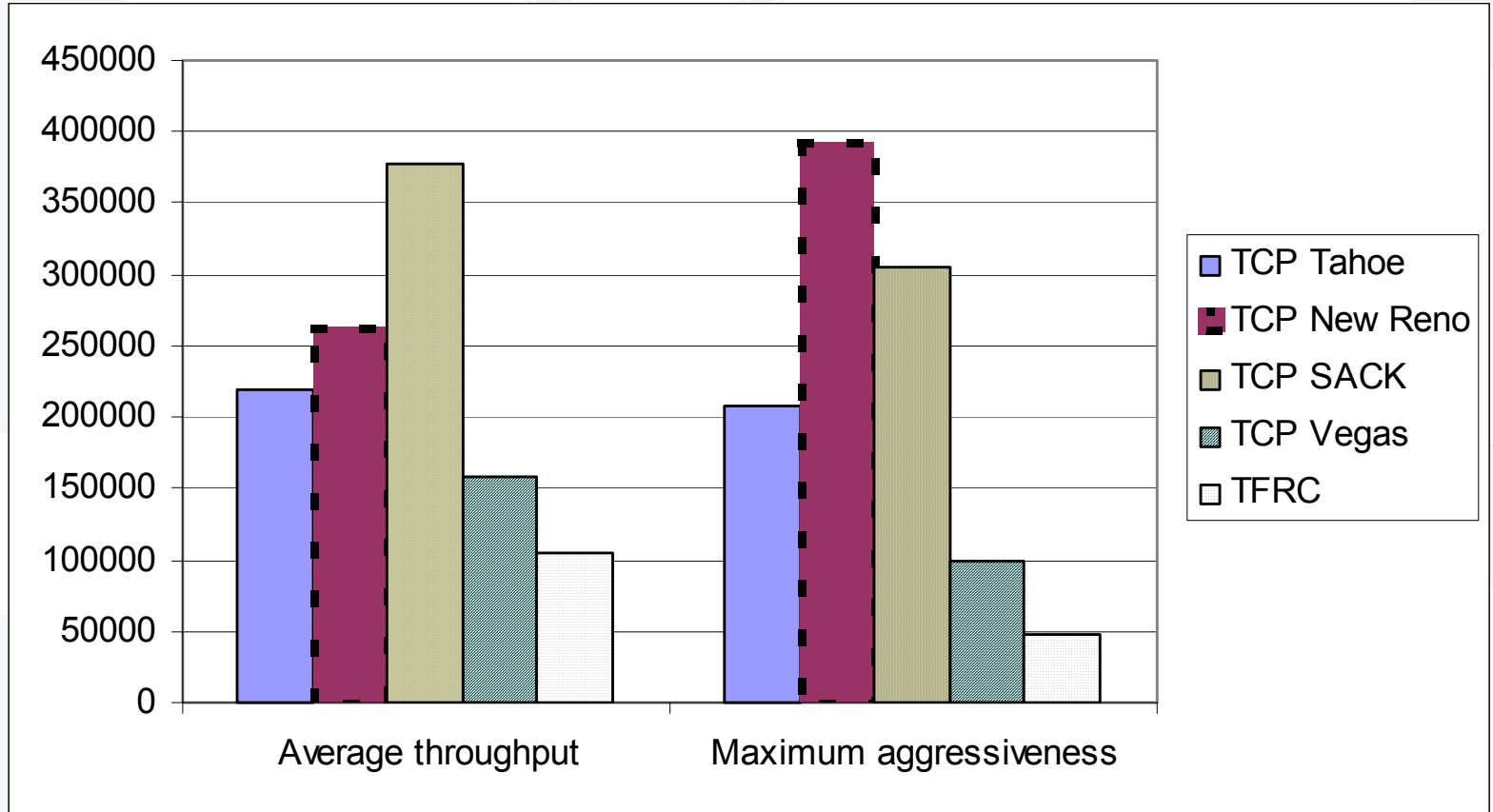
→ Quel sera l'impact sur la topologie à simuler pour combiner ces deux aspects ?



- ▶ Comment déterminer à partir des traces passives la version de TCP qui a été utilisée par un flux ?
 - ▶ Il n'existe pour ce faire que des techniques actives (TBIT)
- ▶ Or les versions de TCP ont des niveaux d'agressivité différents
 - ▶ Et donc un impact sur les oscillations du trafic



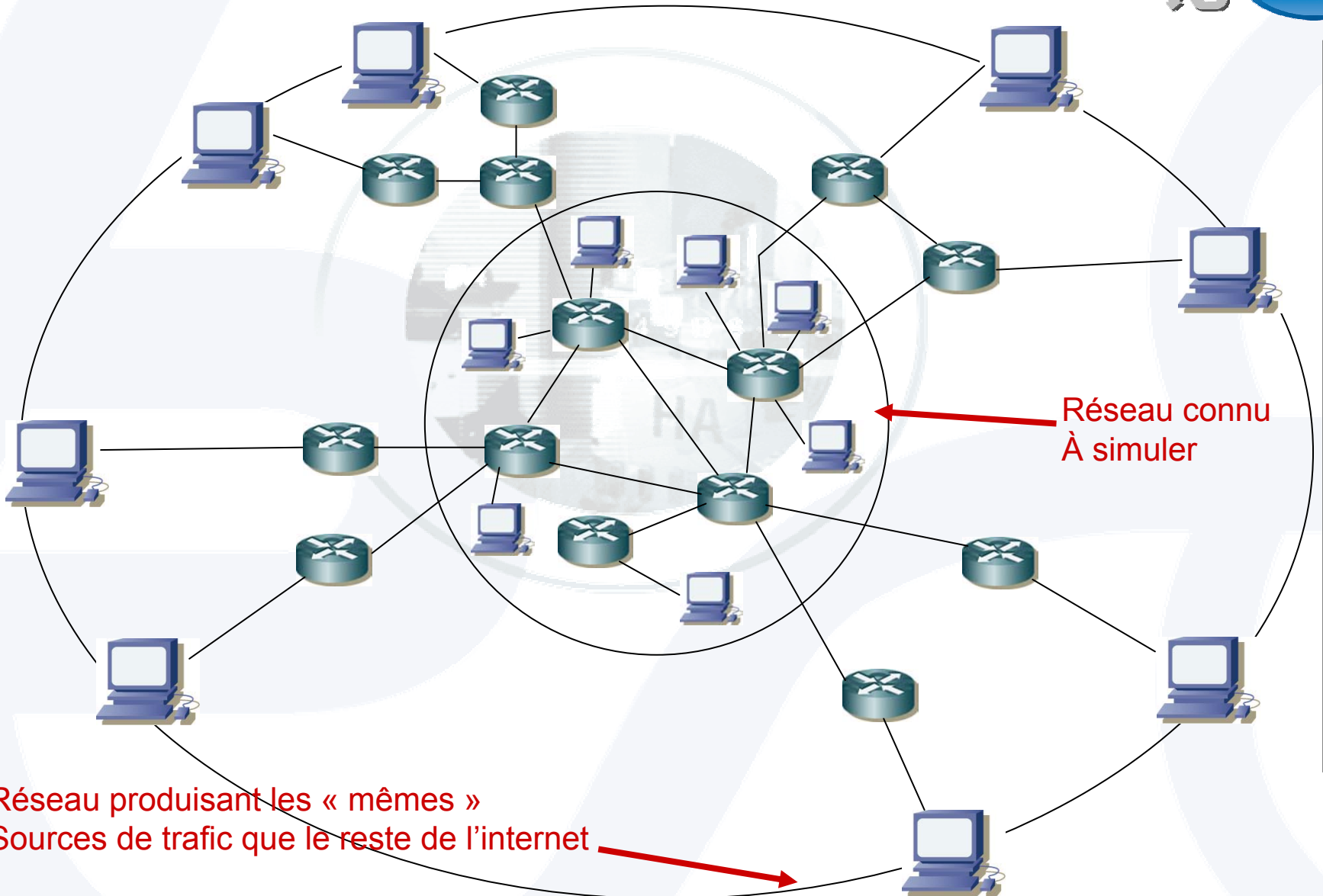
Agressivité des versions de TCP



- ▶ Utiliser T-RAT pour déterminer la version de TCP utilisée par un flux à partir d'une trace passive ?
 - T-RAT permet de déterminer le facteur limitant d'une connexion TCP ↔ lien avec l'agressivité ?

- ▶ Considérer des topologies moins simplistes pour :
 - ▶ Tenir compte des différents types de trafic (cœur/bordure/etc.)
 - ▶ Permettre de créer des contentions entre les flux qui étaient effectivement en contention et qui ont donc interagi l'un avec l'autre
- ▶ Pour cela :
 - ▶ Découvrir la topologie environnante (Magoni, pansiot) ?
 - ▶ Estimer les matrices de trafic (Vaton) ?

Améliorer le réalisme des simul



Réseau produisant les « mêmes »
Sources de trafic que le reste de l'internet



Conclusion générale



- ▶ Le rejeu de flux à partir de traces de métrologie passive donne déjà de très bon résultats
- ▶ Le rejeu est une solution disponible dès à présent (alors que les modèles formels du trafic risquent encore de demander de nombreuses années de recherche)
- ▶ Beaucoup reste à faire !
- ▶ **METROPOLIS va mettre des traces à disposition de la communauté internationale**