Comparatif des performances des systèmes d'exploitation serveurs en 2014

















Vivien Guéant – décembre 2024

Contexte de l'étude de 2014















Objectif : Sélectionner un système d'exploitation serveur efficace, mais également représentatif pour réaliser des tests de débits.

<u>Les 3 critères de choix pour choisir un système d'exploitation serveur :</u>

- **Les performances dans le cadre d'une connexion à 1 Gbit/s.**
- **La stabilité des résultats réalisée dans des conditions identiques.**
- **La représentativité de ce système d'exploitation pour l'hébergement de sites web.**

Les dernières versions des systèmes d'exploitation serveur ayant plus de 0,1% de part de marché pour l'hébergement de sites web ont été testées.

CentOS 7 étant strictement identique à RedHat 7 au nom et logo prés, seul ce dernier apparaît dans ce comparatif. (Tous les paquets de CentOS 7, à l'exception du logo, sont des paquets compilés à partir des sources de la distribution RHEL - Red Hat Enterprise Linux).

Méthodologie de test

Un réseau minimaliste est crée avec un client et un serveur, connectés entre eux directement par un switch 1 Gb/s.

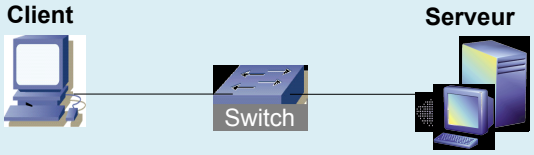
Latence A/R simulés (en ms) : 0, 5, 10, 20, 30, 50, 70, 100 et 140ms

Type de perte de paquet simulés : 0% , 0,001% , 0,01% et 0,1%

Taille du fichier test : 100Mo (utilisation de curl pour l'automatisation)

=> Pour chaque combinaison 100 téléchargements sont réalisés.

Schéma du réseau utilisé :

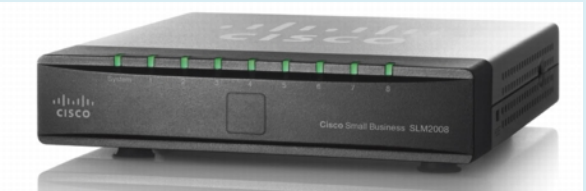


Logiciel pour simuler les pertes de paquets et la latence : NetEm « Network Emulation »

Serveur : Core i5 M560, 4 Go RAM.
Carte réseau serveur : Intel PRO/1000 82577LM Gigabit Ethernet.

Client : Core i5 M560, 4 Go RAM.
Carte réseau client : Intel PRO/1000 82577LM Gigabit Ethernet.
OS : Ubuntu 14.10, Debian 7 et RedHat ont été testés. Les résultats présentés ici sont ceux avec le client le plus performant : Ubuntu

Connexion entre serveur et client : Pas de box, uniquement un switch Cisco SLM2008 (8 ports 1 Gb/s)



Plus de 100 000 téléchargements effectués !

Systématiquement , c'est l'installation par défaut qui est utilisé, sans modification de paramètres.

Caractéristiques des 8 systèmes d'exploitations testés :

Microsoft Windows Server 2012 R2
Kernel : Win 6.2 Serveur Web : IIS 8.5
Sortie le 18 octobre 2013 (basé sur Windows 8.1)

Debian 7
Kernel : Linux 3.2 Serveur Web : Apache 2.2.22
Sortie le 4 mai 2013

Ubuntu Server 14.04 LTS
Kernel : Linux 3.13 Serveur Web : Apache 2.4.7
Sortie le 17 avril 2014

Ubuntu Server 14.10
Kernel : Linux 3.16 Serveur Web : Apache 2.4.10
Sortie le 23 octobre 2014

Red Hat Entreprise Linux Server 7
Kernel : Linux 3.10 Serveur Web : Apache 2.4.6
Sortie le 10 juin 2014

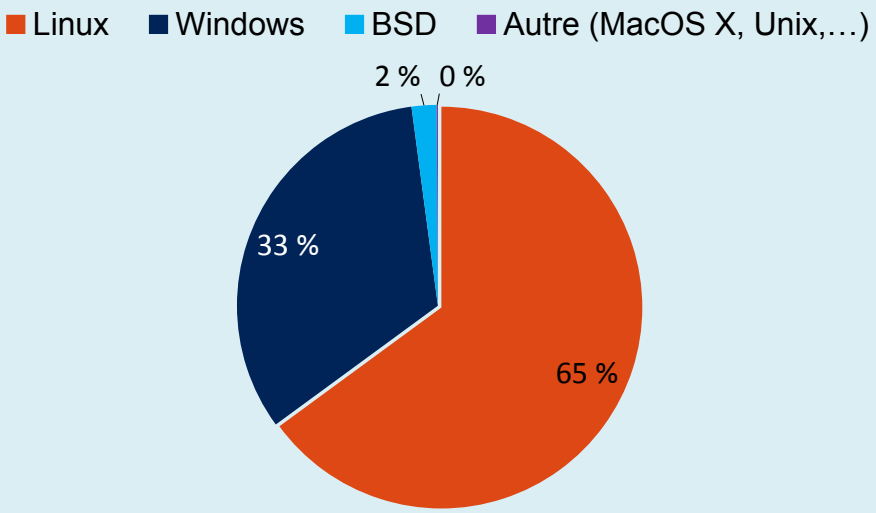
SUSE Linux Entreprise Server 12
Kernel : Linux 3.12 Serveur Web : Apache 2.4.10
Sortie le 28 octobre 2014

Gentoo Linux 30 octobre 2014
Kernel : Linux 3.16 Serveur Web : Apache 2.2.27
Sortie le 30 octobre 2014

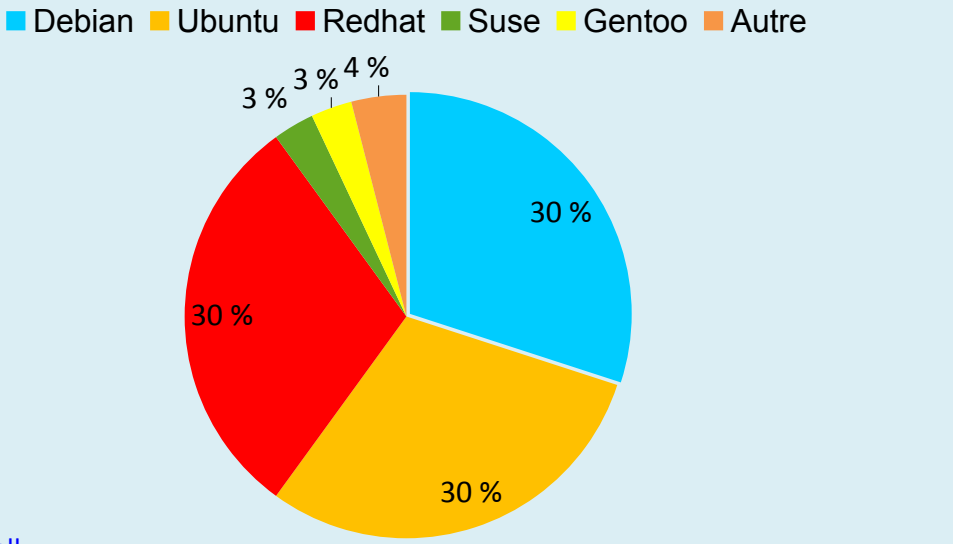
FreeBSD 10.0
Kernel : FreeBSD 10 Serveur Web : Apache 2.4.10
Sortie le 20 janvier 2014

Représentativité des systèmes d'exploitation serveur

Les 3 catégories de systèmes d'exploitation serveur :



Répartition approximative des parts de marché des Linux :



Source : https://w3techs.com/technologies/overview/operating_system/all

Les systèmes d'exploitation serveurs représentatifs d'un usage client :

Windows Server

ubuntu

debian

redhat

Les systèmes d'exploitation non représentatifs, étudiés à titre d'information :

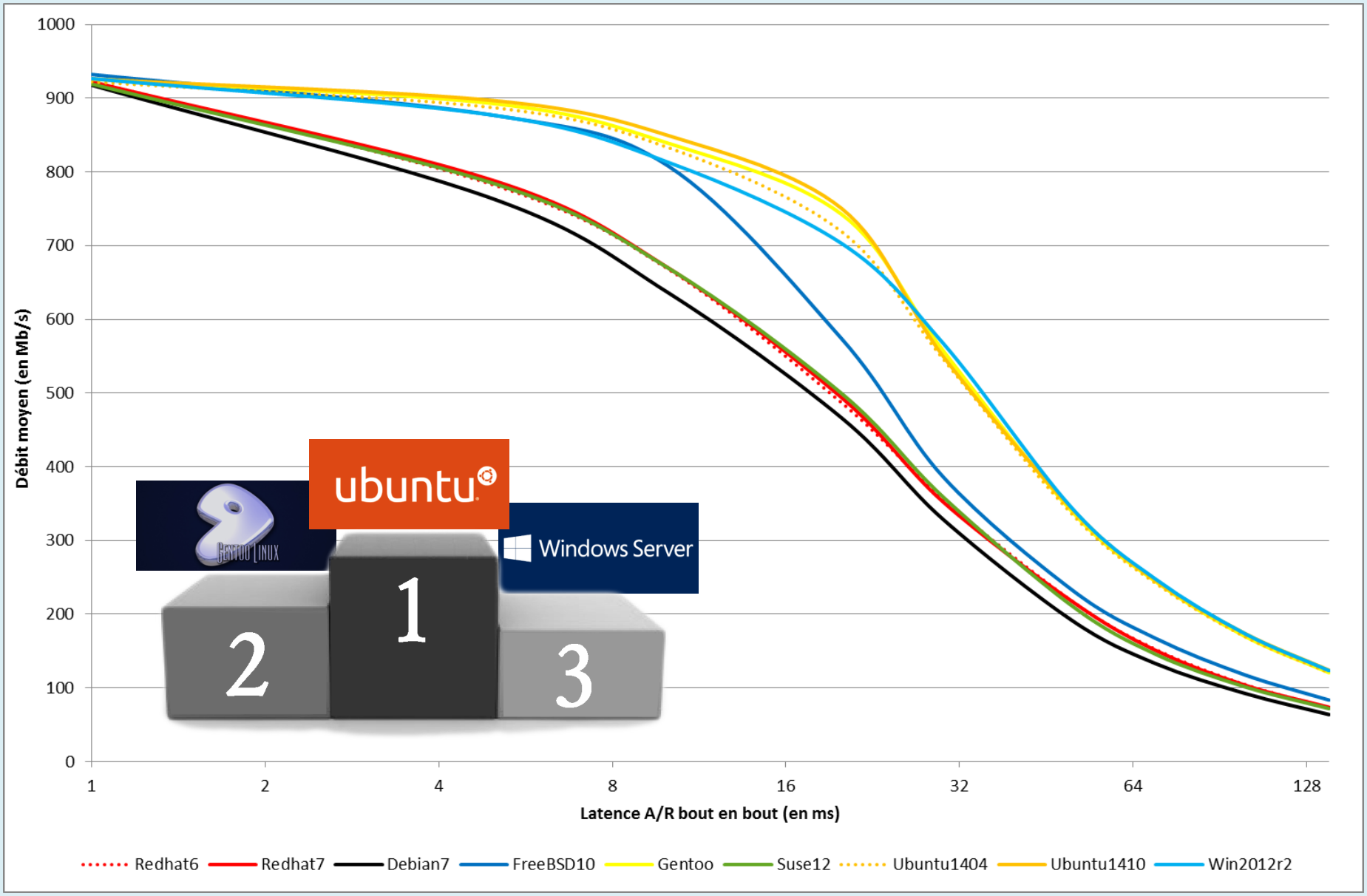
SUSE Linux Enterprise Server

Gentoo Linux

FreeBSD

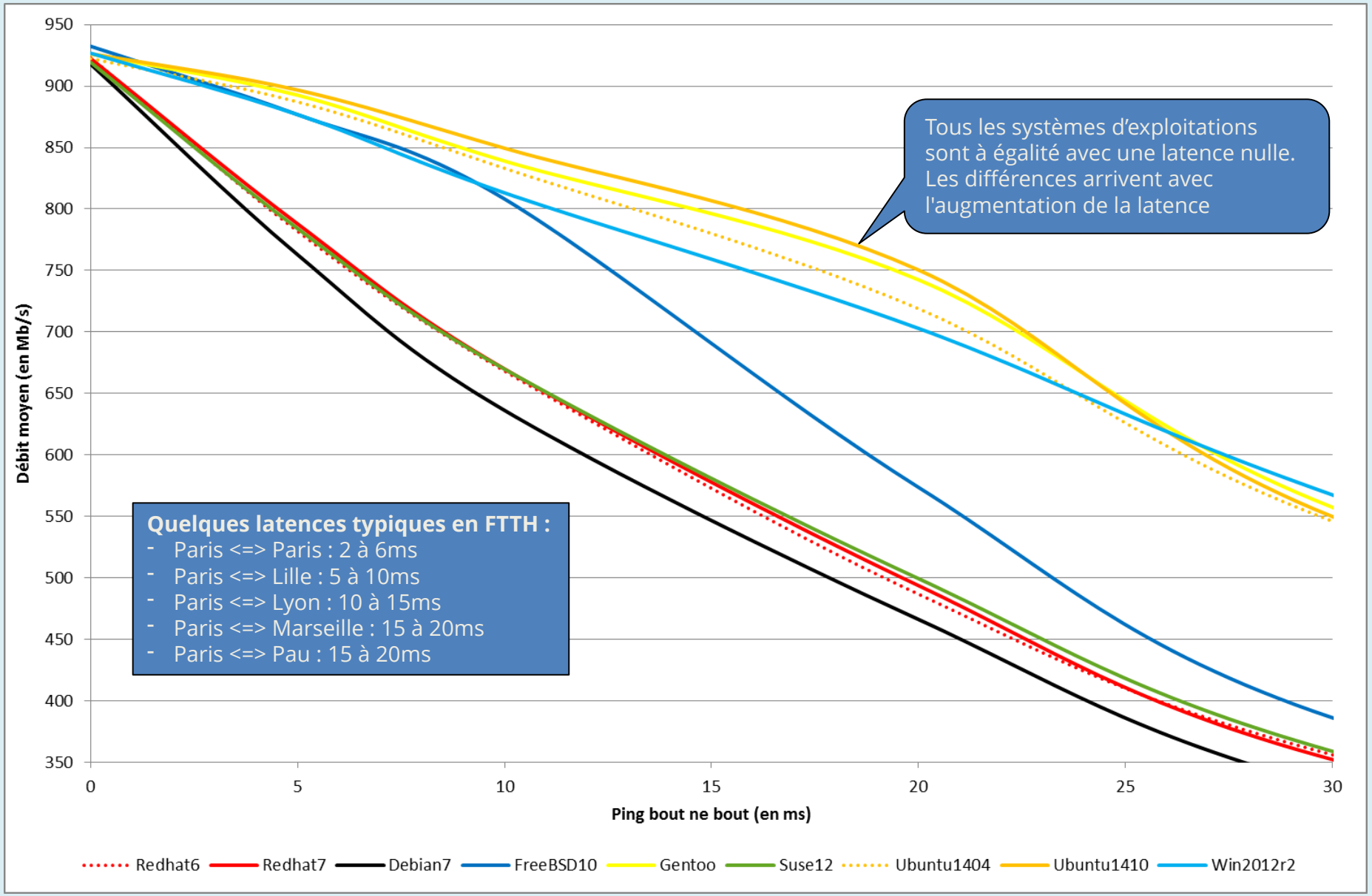
Débit en fonction de la latence A/R

Sans perte de paquets



Débit en fonction de la latence A/R

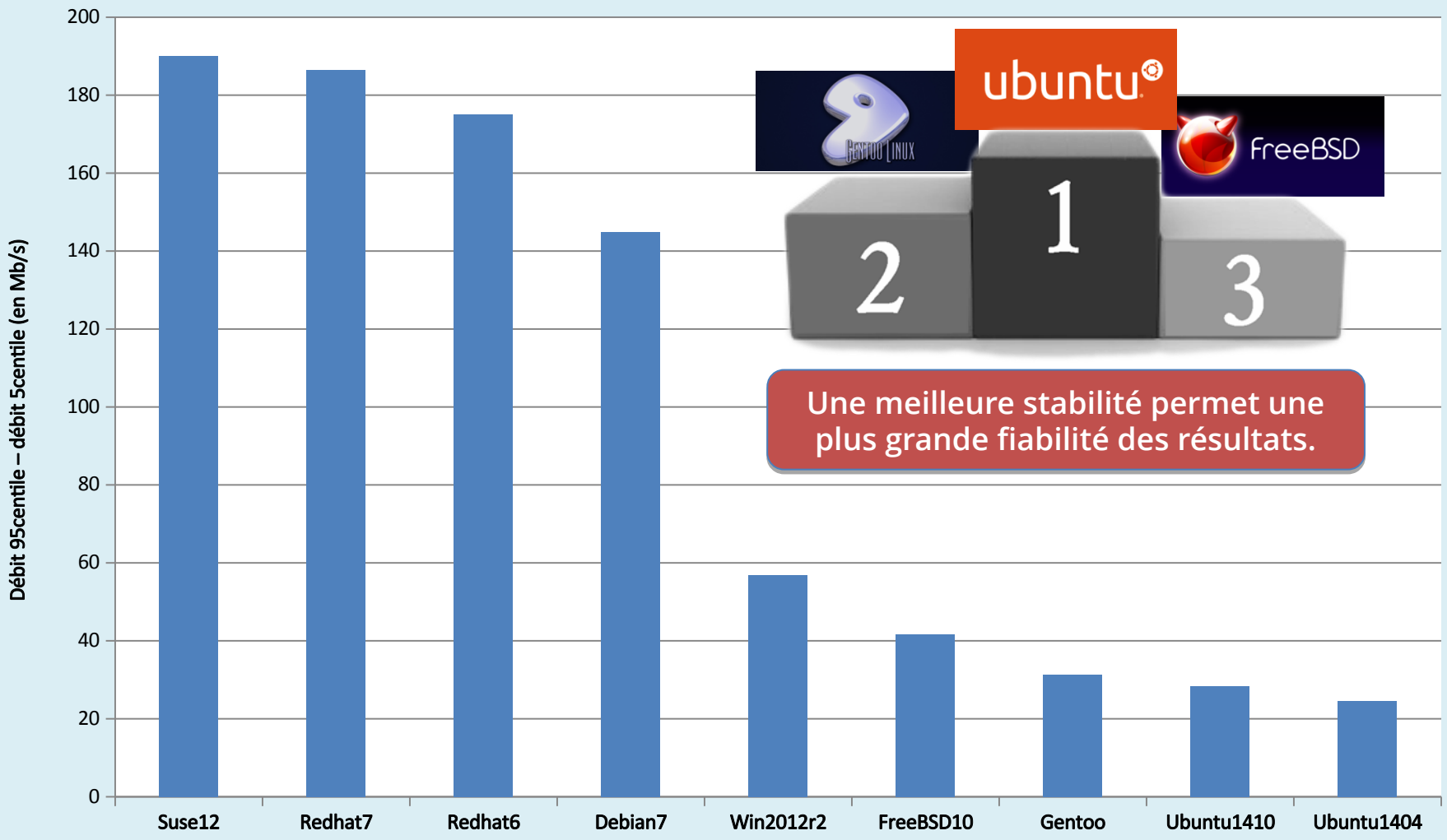
Sans perte de paquets



Stabilité des débits

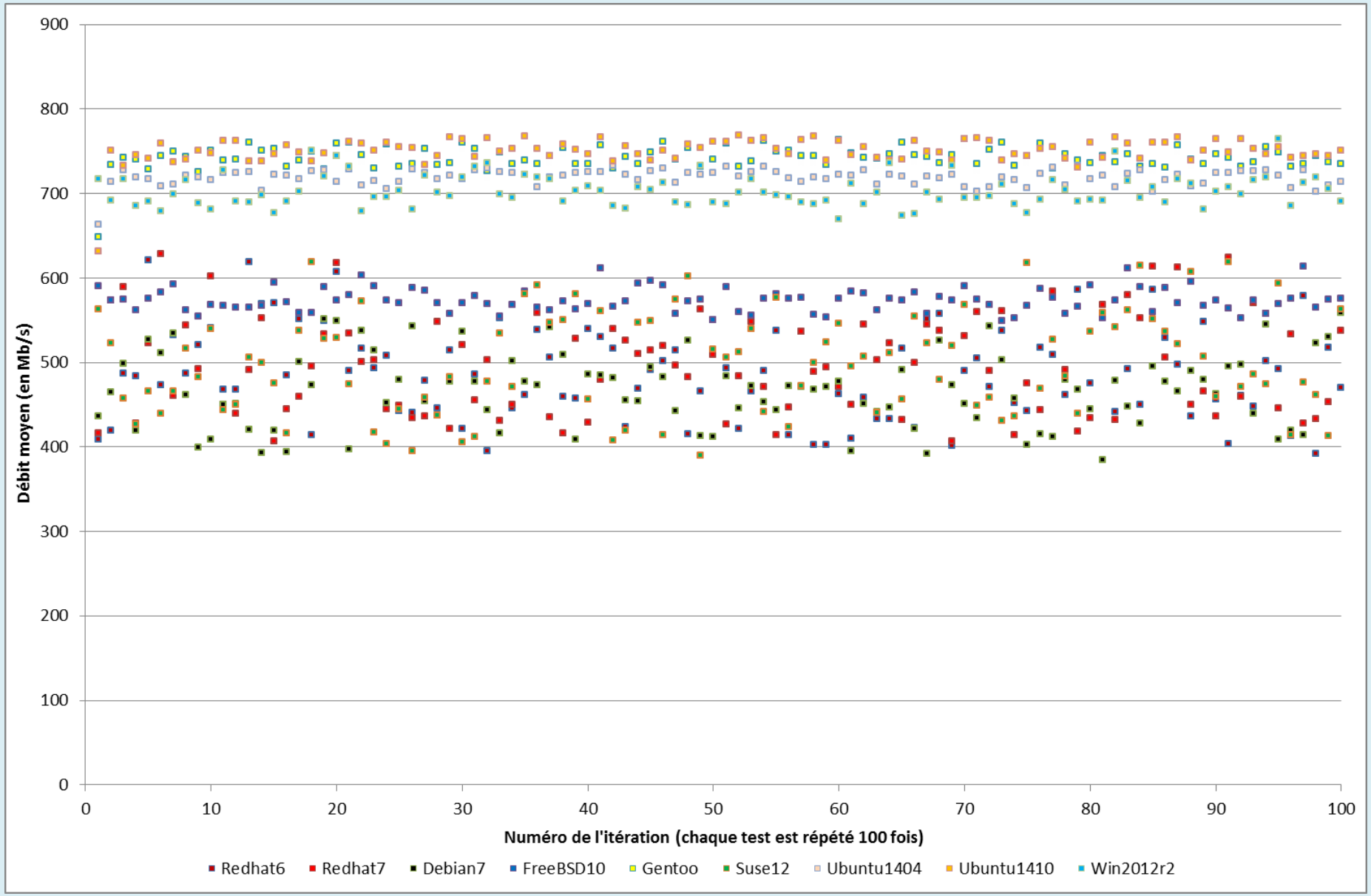
avec une latence A/R de 20ms

Pour la stabilité des résultats, on calcule la différence entre le débit 95centile et le débit 5centile pour 100 tests.



Stabilité des débits

avec une latence A/R de 20ms



Impact des pertes de paquets
avec une latence A/R de 20ms

Des pertes de paquets aléatoirement réparties sont introduites dans le sens serveur => client afin de simuler une dégradation de la connexion (utilisation de Netem pour générer les pertes de paquets).

