



autorité de régulation  
des communications électroniques,  
des postes et de la distribution de la presse

RÉPUBLIQUE FRANÇAISE

# CONFIGURATION DES SERVEURS POUR L'ENQUETE QoS MOBILE ARCEP 2022

28 mars 2022



ISSN n°2258-3106

## Sommaire

1	Choix du système d'exploitation et du noyau Linux.....	3
1.1	Système d'exploitation : Ubuntu 20.04 LTS .....	3
1.2	Noyau Linux : noyau HWE (Linux 5.13 S1 2022).....	4
2	Configuration de la pile TCP/IP coté serveur .....	5
2.1	Algorithme d'évitement de congestion : 75% Cubic et 25% BBR.....	5
2.2	Paramétrage « qdisc » (Protocole de Queuing Discipline) : FQ_CoDel (Cubic) FQ (BBR) .....	5
2.3	Paramétrage du TCP Offload Engine : Laisser la valeur par défaut.....	6
2.4	Paramétrage « initcwnd » : 10 paquets (valeur par défaut) .....	6
2.5	Paramétrage « tcp_no_metrics_save » : 1.....	6
2.6	Paramétrage des buffers : maximum de 16 Mo.....	7
2.7	Paramétrage « vm.swappiness » : 1.....	8
2.8	Résumé des configurations TCP/IP coté serveur .....	9
3	Configuration du serveur web.....	10
3.1	Choix du serveur web : Apache 2.4.41 avec le MPM event .....	10
3.2	Hypertext Transfer Protocol (http) : HTTP/1.x et HTTP/2 .....	10
3.3	Transport Layer Security (TLS) : TLS 1.2 et TLS 1.3.....	10
3.4	Nom de domaine : un IPv4 only et un IPv4+IPv6 .....	10

# 1 Choix du système d'exploitation et du noyaux Linux

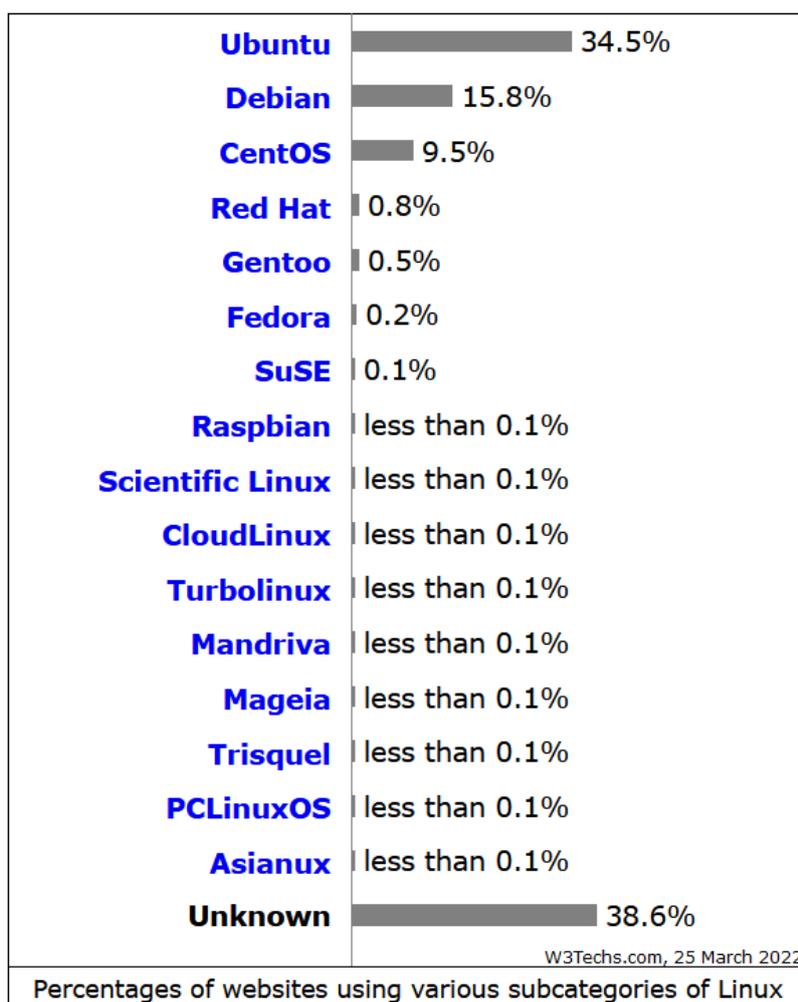
## 1.1 Système d'exploitation : Ubuntu 20.04 LTS

L'Arcep utilise Ubuntu server 20.04 LTS pour l'enquête QoS 2022.

L'enquête QoS 2021 utilisait déjà Ubuntu server 20.04 LTS.

Ce choix permet d'être représentatif d'une partie de l'Internet.

Statistiques proposées par <https://w3techs.com/technologies/details/os-linux>





## 2 Configuration de la pile TCP/IP coté serveur

Note : Les paramètres qui ne sont pas listés sont conservés dans leur valeur par défaut par le système d'exploitation.

### 2.1 Algorithme d'évitement de congestion : 75% Cubic et 25% BBR

Afin d'être toujours plus représentatif de la majorité d'Internet et suite à l'augmentation significative de serveurs utilisant l'algorithme d'évitement de congestion BBR, l'Arcep utilise pour l'enquête QoS 2022 deux types de serveurs :

- Un Serveur qui utilise l'algorithme d'évitement de congestion **Cubic** (75% des mesures en mono-connexion) : `net.ipv4.tcp_congestion_control=cubic`
- Un Serveur qui utilise l'algorithme d'évitement de congestion **BBR** (25% des mesures en mono-connexion) : `net.ipv4.tcp_congestion_control=bbr`

En 2021 toutes les mesures QoS étaient réalisées sur un serveur Cubic.

**Cubic** et **BBR** sont les deux algorithmes les plus utilisés coté serveur pour décider de la vitesse d'envoi des paquets.

Aujourd'hui la majorité de l'internet utilise **Cubic**, créé en 2006, qui s'appuie sur la perte de paquets comme signal pour réduire le débit. **Cubic** est l'implémentation par défaut sous Linux, Android et MacOS. C'est la valeur par défaut d'Ubuntu 20.04 LTS.

Google a développé en 2016 **BBR** pour "**B**ottleneck **B**andwidth and **R**ound-trip propagation time", qui utilise un modèle différent se basant sur la bande passante maximale et le temps d'aller-retour. Cette approche permet à **BBR** de proposer un débit nettement plus élevé que **Cubic** sur un réseau qui perd des paquets sans lien avec une congestion.

### 2.2 Paramétrage « qdisc » (Protocole de Queuing Discipline) : FQ\_CoDel (Cubic) FQ (BBR)

Pour être représentatif, le paramétrage du protocole de Queuing Discipline est lié à l'algorithme d'évitement de congestion.

- **Serveur Cubic** : Le protocole de Queuing Discipline utilisé est **FQ\_CoDel** (`net.core.default_qdisc=fq_codel` - c'est la valeur par défaut dans Ubuntu 20.04 LTS). Cette valeur est la même que pour l'enquête QoS 2021 et permet d'être représentatif de la majorité des serveurs sur Internet utilisant l'algorithme d'évitement de congestion Cubic.
- **Serveur BBR** : Le protocole de Queuing Discipline utilisé est **FQ** (`net.core.default_qdisc=fq`). FQ est recommandé par Google avec BBR et permet d'être représentatif de la majorité des serveurs sur Internet utilisant l'algorithme d'évitement de congestion BBR.

### 2.3 Paramétrage du TCP Offload Engine : Laisser la valeur par défaut

Le **TCP Offload Engine** (TOE) est une technologie utilisée dans les cartes d'interface réseau pour décharger le traitement d'opération liée à la pile TCP/IP vers le contrôleur de réseau.

Les différentes fonctions proposées par TOE sont laissées à leur **valeur par défaut**, afin d'être le plus représentatif des serveurs présents sur Internet. Ce choix utilisé pour l'enquête QoS 2021 est conservé pour l'enquête QoS 2022.

### 2.4 Paramétrage « `initcwnd` » : 10 paquets (valeur par défaut)

Le protocole TCP possède un algorithme nommé « slow start » pour découvrir le débit maximum que supporte une connexion. Avec le « slow start », TCP va commencer par envoyer un nombre de paquets (paquets) défini par l'`initcwnd` (c'est le « TCP Initial Congestion Window ») et va doubler la cadence jusqu'à la saturation du lien afin de s'adapter aux conditions réelles du réseau.

La valeur par défaut de l'`initcwnd` (TCP Initial Congestion Window) est de **10 paquets** sur Ubuntu 20.04 et c'est également la valeur médiane utilisée par le top 50 des sites les plus visités en France (top 50 des sites les plus visités en France selon Alexa).

L'Arcep utilise la **valeur par défaut** de l'`initcwnd` sous Ubuntu 20.04 LTS (10 paquets) pour l'enquête QoS 2022. Ce choix, le même que pour l'enquête QoS 2021, permet d'être représentatif de la majorité d'Internet.

### 2.5 Paramétrage « `tcp_no_metrics_save` » : 1

L'Arcep conserve, pour l'enquête QoS 2022, la valeur « `tcp_no_save_metrics=1` » utilisée en 2021 pour décorréliser les tests successifs.

Par défaut, TCP enregistre diverses métriques de connexion dans le cache de routage lorsque la connexion se ferme, de sorte que les connexions établies dans un proche avenir peuvent les utiliser pour définir les conditions initiales. Habituellement, cela augmente les performances globales, mais peut parfois entraîner une dépendance du test N au test N-1.

Dans le cadre des tests réalisés par l'Arcep, et afin d'assurer une décorrélation des tests successifs, la mémorisation des tests précédents a été désactivée sur le serveur via « `tcp_no_save_metrics=1` » afin d'éviter que le serveur bride tous les tests, à la suite d'une performance limitée.

## 2.6 Paramétrage des buffers : maximum de 16 Mo

Les buffers TCP peuvent limiter artificiellement le débit pour les connexions à très haut débit ou forte latence. L'Arcep modifie la configuration par défaut des serveurs, par une valeur non limitante coté serveur (16 Mo), de façon à ne pas limiter artificiellement les débits, notamment pour les tests relatifs aux enquêtes QoS ultramarines pour lesquelles le serveur est en métropole, entraînant une latence importante.

TCP utilise un mécanisme de fenêtrage pour empêcher qu'un émetteur rapide ne surcharge un récepteur plus lent. Le récepteur annonce la quantité de données que l'émetteur doit envoyer avant d'attendre l'actualisation de la fenêtre par le récepteur. Par conséquent, si une application réceptrice ne peut pas recevoir de données sur la connexion, la quantité de données pouvant être mise en file d'attente est limitée.

Le délai le plus rapide d'actualisation d'une fenêtre correspond à un aller-retour, ce qui conduit à la formule suivante pour calculer l'une des limites de performance du transfert de données groupées sur une connexion TCP : **Débit <= taille de la fenêtre / latence du délai aller-retour (DAR)**

La taille de la fenêtre TCP est affectée par les 4 paramètres suivants :

- `net.ipv4.tcp_rmem`
- `net.ipv4.tcp_wmem`
- `net.core.rmem_max`
- `net.core.wmem_max`

Note : le paramétrage `net.ipv4.xxx` s'applique également au protocole IPv6.

Les deux premiers paramètres configurables affectent la taille de fenêtre TCP pour les applications qui laissent la fonction de réglage automatique de Linux se charger de cette tâche.

Les deux derniers paramètres configurables affectent la taille maximale de fenêtre TCP pour les applications qui tentent de contrôler directement la taille de la fenêtre TCP, en limitant la requête des applications à ces valeurs seulement.

**Paramètres de la mémoire tampon de réception TCP (`tcp_rmem`).** Les 3 valeurs sont :

- Taille minimale du tampon de réception pouvant être allouée à un socket TCP.
- Taille par défaut du tampon de réception.
- Taille maximale de la mémoire tampon de réception pouvant être allouée à un socket TCP.

Valeur pour l'enquête QoS 2021 et 2022 : [net.ipv4.tcp\\_rmem=4096 131072 16777216](#)

**Paramètres de la mémoire tampon d'envoi TCP (`tcp_wmem`).** Les 3 valeurs sont :

- Espace minimal du tampon d'envoi TCP disponible pour un socket TCP.
- Espace par défaut du tampon autorisé pour un socket TCP.
- Espace maximal du tampon d'envoi TCP.

Valeur pour l'enquête QoS 2021 et 2022 : [net.ipv4.tcp\\_wmem=4096 87380 16777216](#)

**Taille maximale de la mémoire tampon de réception du système d'exploitation pour tous les types de connexions :**

Valeur pour l'enquête QoS 2021 et 2022 : [net.core.rmem\\_max=16777216](#)

**Taille maximale de la mémoire tampon d'envoi du système d'exploitation pour tous les types de connexions :**

Valeur pour l'enquête QoS 2021 et 2022 : [net.core.wmem\\_max=16777216](#)

## 2.7 Paramétrage « vm.swappiness » : 1

Le paramètre « **vm.swappiness = 1** » ne concerne pas TCP/IP, mais demande au système de limiter l'utilisation de la swap.

La SWAP est une partie de la mémoire de masse d'un ordinateur utilisée par le système d'exploitation pour déplacer des données peu utilisées des programmes en cours d'exécution, afin d'utiliser l'espace libéré comme cache du système de fichiers.

Ce comportement risque de dégrader les performances réseau au moment où des données peu utilisées en mémoire vive sont mises dans la SWAP. Afin de fiabiliser le flux de données généré par le serveur « **vm.swappiness = 1** », ce qui permet de limiter l'utilisation de la swap aux cas où c'est nécessaire.

Le paramétrage « **vm.swappiness = 1** » était déjà utilisé pour l'enquête QoS 2021.

## 2.8 Résumé des configurations TCP/IP coté serveur

Afin d'être certains que les configurations sont bien présentes même après un reboot du serveur, l'Arcep propose de les inclure dans un fichier [/etc/sysctl.d/90-server-optimization.conf](#)

Les paramètres seront ainsi appliqués au redémarrage du serveur.

### Serveur Cubic :

```
# Algorithme d'évitement de congestion TCP et qdisc
net.ipv4.tcp_congestion_control=cubic
net.core.default_qdisc=fq_codel

# décorrélation des tests successifs
net.ipv4.tcp_no_metrics_save=1

# Paramétrage des buffers max à 16 Mo
net.ipv4.tcp_rmem=4096 131072 16777216
net.ipv4.tcp_wmem=4096 87380 16777216
net.core.rmem_max=16777216
net.core.wmem_max=16777216

# Limiter l'utilisation de la swap
vm.swappiness = 1
```

### Serveur BBR :

```
# Algorithme d'évitement de congestion TCP et qdisc
net.ipv4.tcp_congestion_control=bbp
net.core.default_qdisc=fq

# décorrélation des tests successifs
net.ipv4.tcp_no_metrics_save=1

# Paramétrage des buffers max à 16 Mo
net.ipv4.tcp_rmem=4096 131072 16777216
net.ipv4.tcp_wmem=4096 87380 16777216
net.core.rmem_max=16777216
net.core.wmem_max=16777216

# Limiter l'utilisation de la swap
vm.swappiness = 1
```

## 3 Configuration du serveur web

### 3.1 Choix du serveur web : Apache 2.4.41 avec le MPM event

Le serveur web utilisé est Apache 2 dans sa version proposée pour Ubuntu 20.04 LTS : **Apache 2.4.41**

Le MPM (Modules Multi-Processus) utilisé est le **MPM event** pour sa représentativité.

La version d'OpenSSL est la version proposée pour Ubuntu 20.04 LTS : **OpenSSL 1.1.1f**

### 3.2 Hypertext Transfer Protocol (http) : HTTP/1.x et HTTP/2

Les serveurs de l'enquête QoS 2022 écoutent en **HTTP/1.x** et **HTTP/2**.

C'est le client qui choisit le protocole utilisé.

Ce choix permet d'être représentatif de la majorité d'Internet.

L'activation de HTTP/2 avec Apache (mpm event) se fait avec la commande `sudo a2enmod http2`

### 3.3 Transport Layer Security (TLS) : TLS 1.2 et TLS 1.3

Les serveurs de la campagne QoS 2022 écoutent uniquement en **TLS 1.2** et **TLS 1.3**.

La configuration utilisée est la configuration « Intermediate » décrite sur le site de référence pour la configuration d'un serveur web : <https://ssl-config.mozilla.org/>

Ce choix permet d'être représentatif de la majorité d'Internet.

La configuration https des serveurs utilisés est de grade « **A** » sur <https://www.ssllabs.com/ssltest/>

### 3.4 Nom de domaine : un IPv4 only et un IPv4+IPv6

Les mobiles utilisés pour la l'enquête QoS 2022 en métropole sont configurés avec un APN IPv6 (IPv6 only ou IPv4+IPv6 selon les choix techniques des opérateurs) et l'option IPv6 est activée dans l'espace client quand l'opérateur à ce type de configuration.

Deux noms de domaines sont configurés sur chaque serveur :

- **Un nom de domaine IPv4 only** (50% des tests) : Force le trafic à utiliser IPv4 et la plateforme NAT64 pour un mobile configuré en IPv6 only.
- **Un nom de domaine IPv4+IPv6** proposant les deux protocoles (50% des tests).

Réaliser un test sur deux sur un nom de domaine IPv4 only et un test sur deux sur un nom de domaine proposant IPv4 et IPv6 permet d'être représentatif de la majorité d'Internet.