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

RÉPUBLIQUE FRANCAISE

CONFIGURATION DES SERVEURS POUR L'ENQUETE QoS MOBILE ARCEP 2024

Octobre 2023

ISSN n°2258-3106

Sommaire

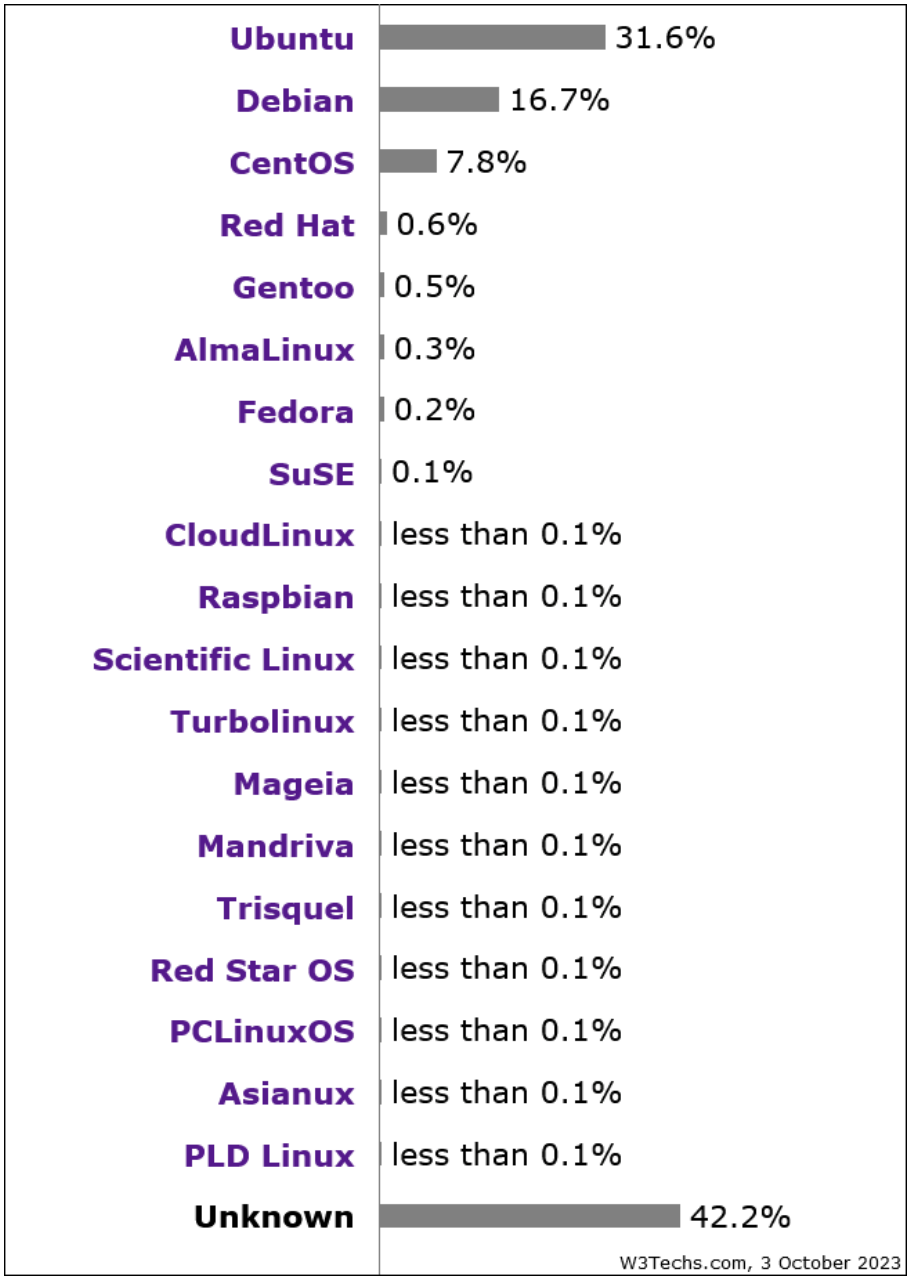
1 Choix du système d'exploitation et du noyau Linux

1.1 Système d'exploitation : Ubuntu Server 22.04 LTS

L'Arcep utilise Ubuntu Server 22.04 LTS pour l'enquête QoS 2024.
L'enquête QoS 2023 utilisait déjà Ubuntu Server 22.04 LTS.

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

Pourcentages de sites Web utilisant diverses sous-catégories de Linux[1] :



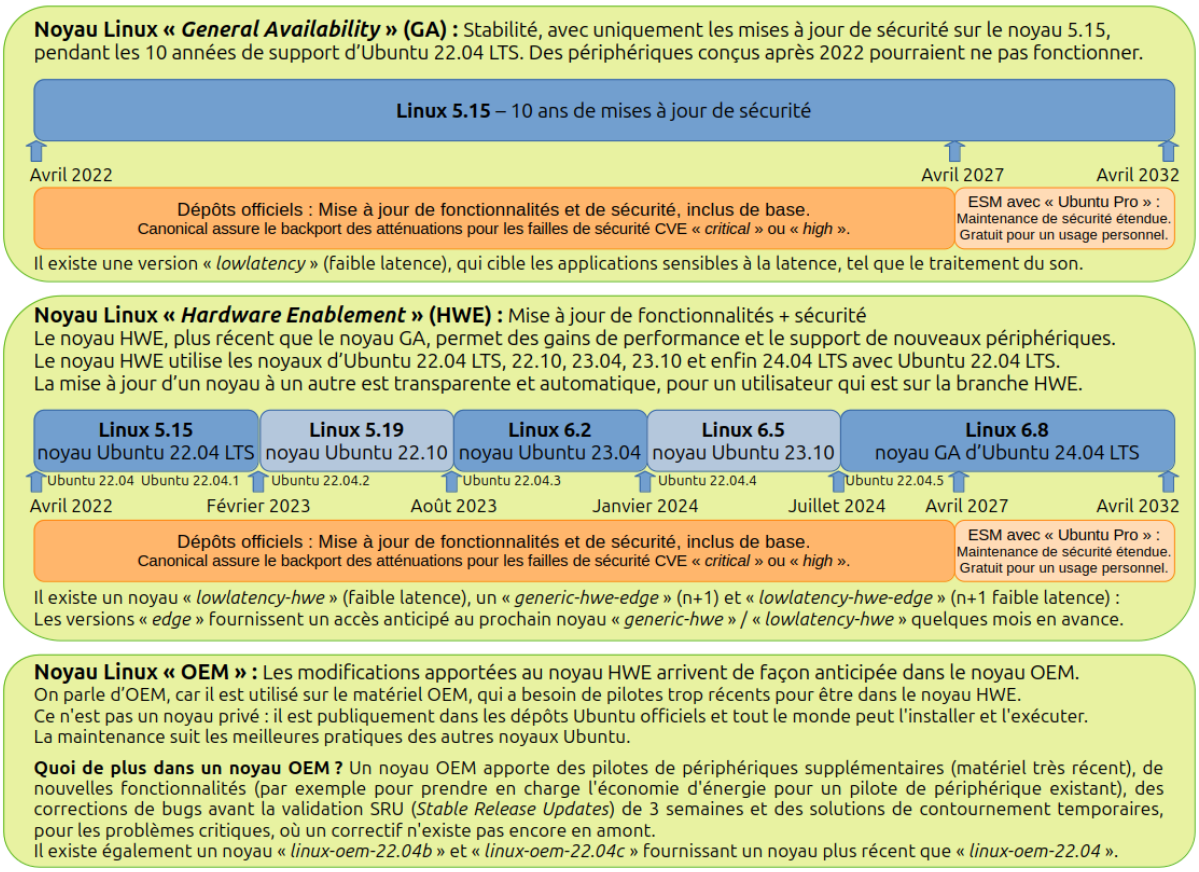
[1] Source : Statistiques proposées par https://w3techs.com/technologies/details/os-linux

1.2 Noyau Linux : noyau HWE

L'Arcep utilisera le noyau « *generic-hwe* » (HWE pour « *Hardware Enablement* ») proposé dans Ubuntu 22.04 LTS afin de bénéficier d'un noyau récent. C'est ce *rolling kernel model* qui est utilisé sur Ubuntu chez AWS[2].

L'enquête QoS 2024 comme celle de 2023 utilise le noyau HWE d'Ubuntu Server 22.04.

Les différents noyaux Linux disponibles avec Ubuntu 22.04 LTS :



Source : https://fr.archive.ubuntu.com/

Commande pour passer du noyau Linux 5.15 GA à un noyau HWE :
sudo apt install --install-recommends linux-generic-hwe-22.04

Après redémarrage sur le noyau Linux HWE, suppression du noyau 5.4, « linux-generic » :
sudo apt purge linux-generic linux-image-generic linux-headers-generic ; sudo apt autoremove

[2] Source : https://ubuntu.com/blog/introducing-the-ubuntu-aws-rolling-kernel-2

2 Configuration de la pile TCP/IP côté serveur

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

2.1 Algorithme d'évitement de congestion : 50% Cubic et 50% BBR

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

- **Cubic**, créé en 2006, s'appuie sur la perte de paquets comme signal pour réduire le débit. Cubic est l'algorithme d'évitement de congestion utilisé par défaut sous Linux (qui équipe la majorité des serveurs sur internet), mais aussi Android et macOS. C'est la valeur par défaut d'Ubuntu 22.04 LTS.

- **BBR** : Google a développé en 2016 **BBR** (pour « ***B**ottleneck **B**andwidth and **R**ound-trip propagation time* »), qui utilise un modèle différent, s'appuyant sur la bande passante maximale et le temps d'aller-retour (*RTT* ou « *round trip time* »). Cette approche permet à **BBR**, quand une connexion perd des paquets, de proposer un débit nettement plus élevé que ceux offerts par les algorithmes s'appuyant sur la perte de paquets, comme **Cubic**. Aujourd'hui, **BBR** est de plus en plus utilisé par certains grands acteurs de l'Internet, notamment sur les serveurs proposant HTTP/3, la nouvelle norme HTTP de troisième génération. Cependant, **BBR** n'est pas encore généralisé sur internet notamment en raison de problématiques d'équité des flux. En effet, sur un même lien où le débit est partagé entre utilisateurs (exemple : les fréquences du réseau mobile ou un lien fibre), les connexions **BBR** vont « prendre la place » des connexions **Cubic**. Une version « **BBRv3** » sera finalisé en 2024. Déjà utilisé sur YouTube et google.com, **BBRv3** améliore les performances de **BBR** et devrait permettre une meilleure cohabitation avec **Cubic,** en ce qu'il permet un partage de liens avec ce dernier. Note : « **BBRv2** » n'a jamais été finalisé.

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 fait évoluer sa méthodologie.

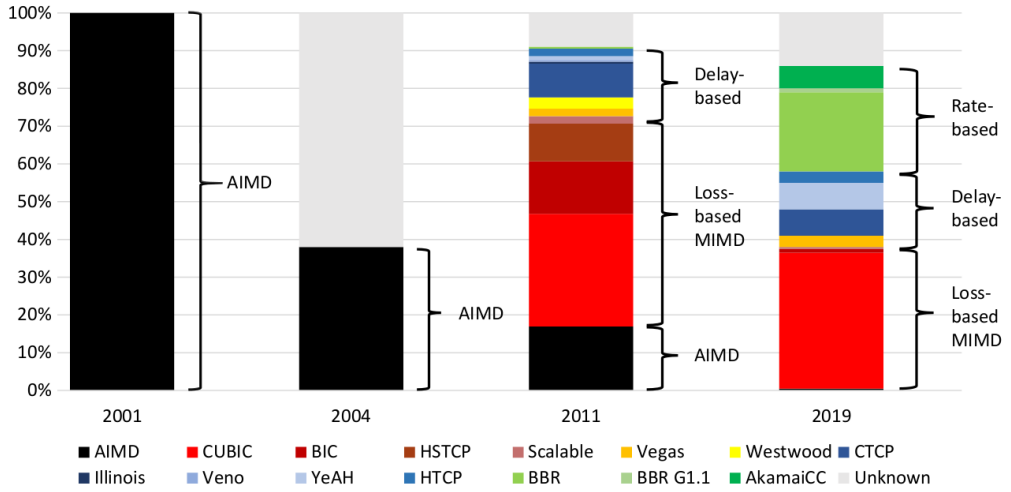
- 2021 : 100% des tests étaient réalisés avec l'algorithme d'évitement de congestion **Cubic** ;
- 2022 : 75% des tests utilisaient **Cubic** et 25% **BBR** ;
- 2023 : 62.5% des tests (5 tests sur 8) utilisaient **Cubic** et 37,5% (3 tests sur 8) **BBR** ;
- 2024 : 50% des tests utilisent **Cubic** et 50% **BBR**.

Configuration du serveur qui utilise l'algorithme d'évitement de congestion **Cubic** :
net.ipv4.tcp_congestion_control=cubic

Configuration du serveur qui utilise l'algorithme d'évitement de congestion **BBR** :
net.ipv4.tcp_congestion_control=bbr

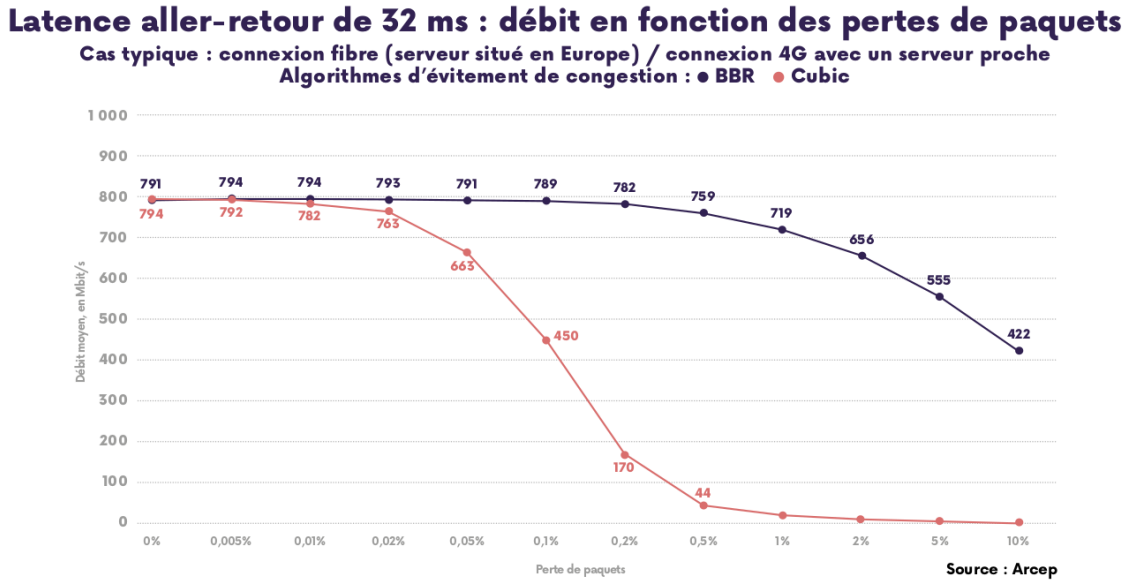
Part des algorithmes de contrôle de congestion déployés par nombre de sites web, dans le Top 20 000 Alexa[3] :»

Les algorithmes basés sur la perte de paquets (comme **Cubic**) sont légèrement devant ceux se basant sur la bande passante maximale (comme **BBR**).



Comparatif du débit moyen, obtenu avec BBR et Cubic, en fonction des pertes de paquets :

Les tests présentés ci-dessous ont été réalisés par les services de l'Arcep en environnement « contrôlé » : Un serveur mis en place pour l'occasion est dédié aux tests et relié directement au client par un câble Ethernet à 1 Gbit/s de 2 mètres. La latence et la perte de paquets sont rajoutées avec le logiciel NetEm, intégré au noyau Linux. Le protocole suivi est celui des campagnes de mesure de la QoS mobile de l'Arcep : un fichier de 250 Mio[4] (fichier utilisé pour l'enquête QoS mobile 2024) est téléchargé en HTTPS avec un serveur Ubuntu 22.04. Le test est arrêté une fois les 250 Mio atteints, ou après expiration du délai de 10 secondes, conformément au protocole QoS mobile 2024.



[3] Source Ayush Mishra, IETF 109 du 20 novembre 2020. Mesures réalisées entre juillet et octobre 2019 depuis des serveurs localisés à Singapour, Bombay, Paris, Sao Paulo et Ohio. Diapositives IETF : https://datatracker.ietf.org/meeting/109/materials/slides-109-iccrg-the-great-internet-tcp-congestion-control-census-00

[4] Mio, symbole d'unité du mébioctet valant 1024 Kio (Kibioctet) = 1024 x 1024 octets soit 1 048 576 octets. Un Mo (mégaoctet) vaut 1000 Ko soit 1 000 000 octets.

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 22.04 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 « tcp_no_metrics_save » : 1

L'Arcep conserve, pour l'enquête QoS 2024, la valeur « **tcp_no_metrics_save=1** » utilisée en 2023 pour décorréler 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 puissent les utiliser pour définir les conditions initiales. Habituellement, cela augmente les performances globales, mais cela 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 le paramétrage « **tcp_no_metrics_save=1** » afin d'éviter que le serveur bride tous les tests, à la suite d'une performance limitée.

2.4 Paramétrage « vm.swappiness » : 1

Le paramètre « **vm.swappiness = 1** » ne concerne pas TCP/IP, mais demande au système de limiter l'utilisation de l'espace d'échange situé sur disque (*swap*). Cet espace est utilisé 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 le *swap*. Afin de fiabiliser le flux de données généré par le serveur, l'Arcep utilise le paramétrage « **vm.swappiness = 1** », ce qui permet de limiter l'utilisation du *swap* aux cas où c'est nécessaire. Le paramétrage « **vm.swappiness = 1** », utilisé en 2024, était déjà utilisé pour l'enquête QoS 2023.

2.5 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 possible des serveurs présents sur Internet. Ce choix utilisé pour l'enquête QoS 2024 est conservé pour l'enquête QoS 2023.

2.6 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 défini par le paramètre **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 **initcwnd** (*TCP Initial Congestion Window*) est de **10 paquets** sur Ubuntu 22.04 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 **initcwnd** sous Ubuntu 22.04 LTS (10 paquets) pour l'enquête QoS 2024. Ce choix, le même que pour l'enquête QoS 2023, permet d'être représentatif de la majorité d'Internet.

2.7 Paramétrage de la fenêtre TCP : maximum de 32 Mio

TCP utilise un mécanisme de fenêtrage glissante 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. 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).**

Une fenêtre TCP trop petite peut 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 côté serveur (32 Mio[5]), de façon à ne pas limiter artificiellement les débits. C'est particulièrement utile dans le cadre des tests relatifs aux enquêtes QoS ultramarines, pour lesquelles le serveur est en métropole, entraînant une latence importante. Cette valeur de 32 Mio pour 2024 est la même que celle utilisée en 2023.

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

- **`net.ipv4`[6]`.tcp_rmem`**
- **`net.ipv4.tcp_wmem`**
- **`net.core.rmem_max`**
- **`net.core.wmem_max`**

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 2023 et 2024 : **net.ipv4.tcp_rmem=4096 131072 33554432**

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 2023 et 2024 : **net.ipv4.tcp_wmem=4096 87380 33554432**

Taille maximale de la mémoire tampon de réception du système d'exploitation pour tous les types de connexions :
Valeur pour l'enquête QoS 2023 et 2024 : **net.core.rmem_max= 33554432**

Taille maximale de la mémoire tampon d'envoi du système d'exploitation pour tous les types de connexions :
Valeur pour l'enquête QoS 2023 et 2024 : **net.core.wmem_max= 33554432**

---

[5] Mio, symbole d'unité du mébioctet valant 1 048 576 (1 024 × 1 024 = 2^{20}) octets.

[6] Le paramétrage net.ipv4.xxx s'applique également au protocole IPv6.

2.8 Résumé des configurations TCP/IP coté serveur

Afin de s'assurer que les configurations sont bien présentes même après un *reboot* du serveur, l'Arcep propose de les inclure dans un fichier de configuration :

/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 de la fenêtre TCP à 32 Mio
net.ipv4.tcp_rmem=4096 131072 33554432
net.ipv4.tcp_wmem=4096 87380 33554432
net.core.rmem_max=33554432
net.core.wmem_max=33554432

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

Serveur BBR :

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

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

# Paramétrage de la fenêtre TCP à 32 Mio
net.ipv4.tcp_rmem=4096 131072 33554432
net.ipv4.tcp_wmem=4096 87380 33554432
net.core.rmem_max=33554432
net.core.wmem_max=33554432

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

3 Configuration du serveur web

3.1 Choix du serveur web : Apache 2.4.52 avec OpenSSL 3.0.2

- Le serveur web utilisé est Apache HTTP Server, dans sa version proposée dans Ubuntu Server 22.04 LTS : **Apache 2.4.52**
- La version d'OpenSSL est la version proposée dans Ubuntu 22.04 LTS : **OpenSSL 3.0.2**

Ces versions sont identiques avec l'enquête QoS 2023. Les évolutions des versions par rapport à l'enquête 2022 sont liées à l'évolution en termes de système d'exploitation (Ubuntu Server 20.04 LTS pour l'enquête QoS 2022 contre Ubuntu Server 22.04 LTS pour l'enquête QoS 2023 et 2024).

3.2 Choix du *MPM* utilisé par Apache : *MPM event*

Les *MPM* (*Multi-Processing Modules*) sont des modules utilisés par le serveur web Apache qui définissent la manière d'accepter les requêtes des clients, ainsi que la façon de se les répartir entre les différents processus Apache.

Comme pour l'enquête QoS 2023, en 2024 le *MPM* utilisé est le ***MPM event*** pour sa représentativité. C'est le MPM utilisé par défaut par Apache quand il n'y a pas PHP d'installé sur le serveur.

Pour installer PHP avec le *MPM event*, il faut installer le paquet php-fpm et non libapache2-mod-php, comme indiqué ici :

```
apt install apache2 php-fpm apache2-utils
```

Attention : Dans le cas où libapache2-mod-php a été précédemment installé, il faut faire une petite manipulation pour désactiver le PHP intégré à Apache pour changer de *MPM* :
`a2dismod php8.1` (Il faut désactiver php pour désactiver mpm_prefork)
`a2dismod mpm_prefork`
`a2enmod mpm_event proxy_fcgi setenvif`

Activation de *PHP FPM* :
`systemctl start php8.1-fpm` (pour le démarrer immédiatement)
`systemctl enable php8.1-fpm` (pour le démarrer au démarrage du serveur)
`a2enconf php8.1-fpm`
Le fichier de configuration est situé ici : /etc/php/8.1/fpm/pool.d/www.conf
Le fichier php.ini se trouve ici : /etc/php/8.1/fpm/php.ini

3.3 Hypertext Transfer Protocol (http) : HTTP/1.x uniquement

Les serveurs de l'enquête QoS 2023 et 2024 écoutent en **HTTP/1.x** uniquement.
C'est la configuration par défaut d'Apache.

HTTP/2 n'est pas mis en place sur le serveur à cause de suspicions de dégradation de performances avec certains logiciels clients **HTTP/2**.

3.4 *Transport Layer Security* (TLS) : TLS 1.2 et TLS 1.3

Les serveurs de la campagne QoS 2024 écoutent uniquement en **TLS 1.2** et **TLS 1.3** (comme pour l'enquête QoS 2023).

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, identique à celui fait l'année dernière, 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/

3.5 Nom de domaine : un domaine IPv4 only et un domaine IPv4+IPv6 only

Les mobiles utilisés pour la l'enquête QoS 2024 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. C'était déjà le cas pour l'enquête QoS 2023.

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.