

# Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Conventions](#)

[Amorce de Class-Based Weighted Fair Queueing](#)

[Compréhension des limites et des suppressions de sortie de file d'attente](#)

[Classes définies par l'utilisateur configurées avec la commande « prioritaire »](#)

[Classes définies par l'utilisateur configurées avec la commande de « bande passante »](#)

[Comportement par défaut de classe](#)

[Informations connexes](#)

## [Introduction](#)

Ce document s'applique aux plates-formes logicielles de Cisco IOS® seulement, qui comporte généralement le 3800 de Cisco 7200VXR et de Cisco ISR, 2800, aux Routeurs de gamme 1800, aussi bien qu'aux Routeurs existants d'accès de Cisco comprenant 3700, 3600, 2600, et des Routeurs de gamme 1700.

## [Conditions préalables](#)

### [Conditions requises](#)

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- [Solutions de Qualité de service Cisco IOS](#)
- [QoS-----Cadre de queue hiérarchique \(HQF\)](#)

### [Composants utilisés](#)

Les informations contenues dans ce document sont basées sur les versions de logiciel suivantes :

- Pour Pre-HQF : Routeurs de Cisco qui exécutent la version du logiciel Cisco IOS 12.3(26)
- Pour HQF : Routeurs de Cisco qui exécutent le Logiciel Cisco IOS version 12.4(22)T

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

### [Conventions](#)

Pour plus d'informations sur les conventions utilisées dans ce document, reportez-vous à

## Amorce de Class-Based Weighted Fair Queueing

Dans des images IOS de pre-HQF, d'une façon générale, n'importe quelle classe avec une **commande bandwidth** sera généralement donnée la priorité contre des classes sans *bande passante* ou *priorité* basée sur les classes ? *Poids*. Afin de comprendre l'algorithme de planification de Mise en file d'attente pondérée basée sur les classes (CBWFQ), vous devez d'abord comprendre le concept d'un poids, qui est écoulement-particularité pour la particularité équitable de files d'attente basée par écoulement et de classe pour les files d'attente basées par classe individuelle dans la file d'attente de weighted fair basée par classe.

La formule pour dériver le poids pour une file d'attente équitable basée par écoulement est :

Les classes définies par l'utilisateur dans une file d'attente basée sur classe de weighted fair dérivent leurs poids respectifs en fonction de la **commande bandwidth** configurée dans la classe comme proportion de la SOMME de toutes les classes de *bande passante* dans la file d'attente basée par classe. La formule précise est de propriété industrielle.

Dans des images HQF, des foire-files d'attente, configurables basés sur écoulement dans les deux classes et par défaut définis par l'utilisateur de classe avec la foire-file d'attente, sont programmées également (au lieu d'en poids). En outre, dans HQF, une file d'attente basée par classe ? la priorité de planification s est déterminée basée sur le programmeur HQF et pas sur les classes ? formule existante de poids.

**Remarque:** Cette section n'est pas destinée pour être une analyse comportementale complète pour des exécutions de queue basées par classe. L'intention est une brève formation car CBWFQ s'applique aux limites et aux suppressions de sortie de compréhension de file d'attente.

## Compréhension des limites et des suppressions de sortie de file d'attente

### Classes définies par l'utilisateur configurées avec la commande « prioritaire »

Pour les classes définies par l'utilisateur MQC configurées avec toute variante de la commande **prioritaire**, y compris la **priorité**, le **<kbps> prioritaire**, et les **pour cent prioritaires**.

### Comportement de Pre-HQF

Techniquement, quoiqu'aucun CLI n'existe pour le voir, et lui n'est-il pas configurable, a ? masqué ? la file d'attente de système existe qui est partagée par toutes les données de classe prioritaire. Ceci agit en tant que pile d'attente pour des données prioritaires après qu'il ait été classifié et après qu'il a été permis par le régulateur encombrement-averti. Des paquets LLQ sont placés en cela ? masqué ? file d'attente de système s'ils ne peuvent pas être placés directement sur l'interface de sortie ? boucle de transmission s pendant l'interruption de réception, qui s'appelle autrement l'encombrement fonctionnel. Dans cette situation, parce que l'encombrement fonctionnel existe, le paquet est-il évalué contre le régulateur conditionnel LLQ pendant l'interruption de réception tandis que toujours possédé par l'interface de réception ? gestionnaire s. Si le paquet n'est pas lâché par le LLQ ? régulateur conditionnel s, il est placé en cela ?

masqué ? La file d'attente de système LLQ et l'interruption de réception est libérée. Par conséquent, tous les paquets placés en cela ? masqué ? la file d'attente de système se sont conformées au LLQ ? le régulateur averti d'encombrement s et sera retiré de la file d'attente à l'interface de sortie ? boucle de transmission s immédiatement par le programmeur LLQ/CBWFQ.

En dépit de l'existence de cette file d'attente, le comportement est les files d'attente différentes IOS créées pour des données de non-LLQ (telles que des files d'attente de foire-file d'attente et de bande passante) du fait la latence de queue pas supplémentaire (au-dessus de la latence de boucle de transmission) sera encourue puisque des paquets dans cette file d'attente seront immédiatement vidés à la boucle de transmission par le programmeur LLQ/CBWFQ. Si un paquet de classe prioritaire n'est pas lâché par le régulateur conditionnel pendant l'interruption de réception, alors ce paquet LLQ existera-t-il en cela ? masqué ? file d'attente de système brièvement avant de retirer de la file d'attente à l'interface de sortie ? boucle de transmission s. Dans ce cas, le programmeur LLQ/CBWFQ présentera immédiatement le paquet à l'interface de sortie ? boucle de transmission s. Le régulateur conditionnel a fonctionné avant d'admettre le paquet au LLQ/CBWFQ, limitant de ce fait le LLQ au débit configuré prioritaire.

En résumé, il est recommandé pour relâcher les données LLQ qui dépassent le débit prioritaire pendant des périodes d'encombrement que pour encourir la latence de queue supplémentaire, qui est un composant fondamental de LLQ. Ce mécanisme de maintien de l'ordre conditionnel permet une file d'attente prioritaire stricte sans permettre à la file d'attente prioritaire pour monopoliser l'interface entière PLIM, qui est une amélioration au-dessus d'IOS ? fonctionnalité de mise en file d'attente de priorité héritée s.

- **Limite de file d'attente de Pre-HQF** : NA
- **Pre-HQF ? priorité ? + ? random-detect ? comportement** : NA, WRED non permis dans LLQ.
- **Pre-HQF ? priorité ? + ? foire-file d'attente ? comportement** : NA, foire-file d'attente non permise dans LLQ.
- **Pre-HQF ? priorité ? + ? random-detect ? + ? foire-file d'attente ? comportement** : NA, ni foire-file d'attente ou random-detect pris en charge dans LLQ.

### Comportement HQF

Même que [Pre-HQF](#) à moins que la file d'attente masquée ne soit plus masquée et la queue-limit est maintenant configurable et se transfère sur 64 paquets.

- **Limite de file d'attente HQF** : 64 paquets
- **HQF ? priorité ? + ? random-detect ? comportement** : NA, WRED non permis dans LLQ.
- **HQF ? priorité ? + ? foire-file d'attente ? comportement** : NA, foire-file d'attente non permise dans LLQ.
- **HQF ? priorité ? + ? random-detect ? + ? foire-file d'attente ? comportement** : NA, ni foire-file d'attente ou random-detect pris en charge dans LLQ.

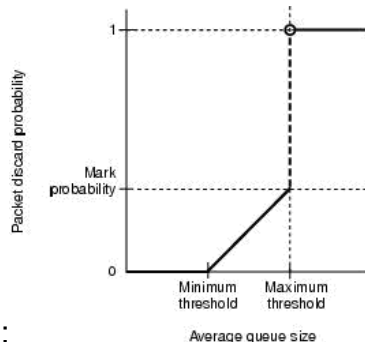
### Classes définies par l'utilisateur configurées avec la commande de « bande passante »

Pour les classes définies par l'utilisateur MQC configurées avec toute variante de la commande **bandwidth**, y compris le `<kbps>`, le **pourcentage de bande passante**, et le **pourcentage de bande passante restante de bande passante**.

## Comportement de Pre-HQF

La queue-limit par défaut est 64 paquets, qui est réglable. Si, pendant l'interruption de réception, vous devez mettre un paquet en file d'attente qui aurait comme conséquence > 64 paquets dans la file d'attente, le paquet est queue lâchée.

- **Queue-limit de Pre-HQF** : 64 paquets, réglables par l'intermédiaire de la queue-limit.
- **Pre-HQF ? bande passante ? + ? random-detect ? comportement** : Exemple : Si n'importe quelle variante de bande passante est configurée avec n'importe quelle variante de random-detect, ignorez n'importe quelle queue-limit CLI, qui retire efficacement n'importe quelle limite de mémoire tampon dans la classe. En d'autres termes, le random-detect et la queue-limit sont mutuellement - exclusivité dans des images de pre-HQF. Utilisant le random-detect comme stratégie de baisse, la taille en cours de file d'attente est non restreinte et peut théoriquement occuper chaque mémoire tampon allouée à la foire-file d'attente basée par classe, où ce nombre de mémoires tampons allouées à la foire-file d'attente basée par classe est dérivé a basé sur le point d'attache de service-stratégie : Interface physique : 1000 paquets, réglables avec le hold-queue CLI d'interface PVC ATMOSPHERE : 500 paquets, réglables avec le vc-hold-queue PVC CLIClasse de mappage de relais de trame : 600 paquets, réglables avec le frame-relay holdq CLI de classe de mappage de relais de trameLa stratégie de mise en forme basée sur classe s'est appliquée (à la sous) interface (pre-HQF) : 1000 paquets, réglables avec le shape max-buffers CLI de classe MQC. **Remarque**: Tous les Relais de trames et classe basés formant des exemples supposent que la somme de modélisateurs ne dépasse pas le rythme d'horloge d'interface. **Remarque**: Dans des images de pre-HQF, quand une classe basée stratégie de mise en forme est appliquée à la sous) interface a (, prenez garde de l'interface physique sous-jacente ? la vitesse s, comme interfaces <2Mbps se transférera sur une file d'attente de weighted fair et les interfaces >2Mbps se transféreront sur le FIFO. Dans le pre-HQF, la file d'attente de formation alimentera la file d'attente d'attente d'interface, si la stratégie de mise en forme est reliée à la sous-interface ou au niveau d'interface physique. Pendant l'interruption de réception, chaque fois que un paquet va bien à un candidat pour une interface ? la file d'attente de sortie s, la taille de file d'attente de moyenne WRED est calculée utilisant cette formule : Si la taille moyenne résultante de file d'attente est : Plus petit que le minute-seuil WRED, mettez le paquet et libérez en file d'attente l'interruption de réception. Entre le minute-seuil WRED et le maximum-seuil WRED, relâchez probablement le paquet avec la probabilité accrue comme la taille moyenne de file d'attente obtient plus près du maximum-seuil WRED. Si la taille moyenne de file d'attente est exactement égale au maximum-seuil WRED, le paquet est lâché selon le dénominateur de probabilité de marque. Le dénominateur de probabilité de marque sert également de spécification de base pour déterminer quel pourcentage de paquets obtiendront relâché quand la limite moyenne de file d'attente n'est pas exactement égale au maximum-seuil WRED mais sont minute-seuil du supérieur à WRED. C'est un exemple



graphique :

Si le paquet est lâché, l'interruption de réception est

libérée et une baisse aléatoire est incrémentée. Si le paquet n'est pas lâché, le paquet est mis en file d'attente et l'interruption de réception est relâchée. Le supérieur au maximum-seuil WRED, relâchent le paquet, libèrent l'interruption de réception, et incrémentent une perte de destination.

- **Remarque:** WRED basés (par défaut) et basés sur dscp de Priorité IP permettent le minute-seuil, le maximum-seuil, et le dénominateur de probabilité de marque à définir différemment pour différentes valeurs. C'est où le composant pesé du dépistage précoce aléatoire est évident. Vous pouvez protéger certaines valeurs de tos relativement à d'autres valeurs par l'intermédiaire d'accorder leurs seuils relatifs et marquer des dénominateurs de probabilité. Si aléatoire le détectez et la bande passante sont configurées ensemble, la taille en cours de file d'attente peut être plus grande que le maximum-seuil WRED à n'importe quel moment indiqué. C'est parce que le minimum et les seuils maximaux WRED agissent seulement basé sur la taille moyenne de file d'attente (non en cours). Ceci fournit une occasion d'expirer toutes les mémoires tampons allouées à la file d'attente basée par classe qui pourrait avoir comme conséquence des baisses de NO--mémoire tampon se produisant n'importe où dans la file d'attente équitable basée par classe (référez-vous à l'ID de bogue Cisco CSCsm94757).
- **Pre-HQF ? bande passante ? + ? foire-file d'attente ? comportement :** NA, foire-file d'attente non permise dans la classe de bande passante.
- **Pre-HQF ? bande passante ? + ? random-detect ? + ? foire-file d'attente ? comportement :** NA, foire-file d'attente non permise dans la classe de bande passante

## [Comportement HQF](#)

Le comportement est identique comme décrit dans la section de [Pre-HQF](#).

- **Queue-limit HQF :** 64 paquets, réglables par l'intermédiaire de la queue-limit. Ce correspond celui dans le pre-HQF.
- **HQF ? bande passante ? + ? random-detect ? comportement :** Exemple : **Remarque:** La queue-limit par défaut est 64 paquets. Le comportement est identique que dans la section correspondante de pre-HQF, à une importante exception. Dans des images HQF, le random-detect et la queue-limit peuvent coexister dans la même classe définie par l'utilisateur (ou pour classer le classe-par défaut) et la queue-limit sera activée et accordée à 64 paquets en configuration par défaut. En soi, la queue-limit servira de taille en cours maximum de file d'attente dans une classe de random-detect, donc fournissant un mécanisme pour limiter des baisses de NO--mémoire tampon discutées dans la section correspondante de pre-HQF. En raison de cet ajout, la limite de file d'attente configurée doit être au moins aussi grande que le maximum-seuil de random-detect, où le maximum-seuil de random-detect se transférera sur 40 paquets, ou bien le programme d'analyse syntaxique rejettera la configuration. Ceci introduit une courant-file d'attente-limite signent des classes WRED, par lequel, même si le calcul de profondeur moyenne de file d'attente est plus petit que le maximum-seuil, si la taille en cours de file d'attente (non moyenne) est plus grande que la queue-limit, le paquet sera abandonné, l'interruption de réception seront libérées, et une perte de destination enregistrée. Souvenez-vous, si la queue-limit est suffisamment élevée accordé pour épuiser les mémoires tampons de queue d'agrégat pour la file d'attente basée sur classe, des baisses de NO--mémoire tampon peut se produire toujours. Des mémoires tampons de queue d'agrégat pour HQF sont définies ici : Interface physique : 1000 paquets, réglables avec le hold-queue CLI d'interface PVC ATMOSPHERE : 500 paquets, réglables avec le vc-hold-queue PVC CLIClasse de mappage de relais de trame : 600 paquets, réglables avec le frame-relay holdq

CLI de classe de mappage de relais de trame La stratégie de mise en forme basée sur classe s'est appliquée à l'interface physique en code HQF : 1000 paquets, réglables avec une combinaison de hold-queue CLI d'interface et de queue-limit de stratégie enfant où la queue-limit de stratégie enfant a une limite supérieure de hold-queue d'interface. La stratégie de mise en forme basée sur classe s'est appliquée à la sous-interface en code HQF : 512 paquets, non réglables (étudiez avec l'équipe de plate-forme NSSTG QoS, si elle est réglable) **Remarque:** Tous les Relais de trames et classe basés formant des exemples supposent que la somme de modélisateurs ne dépasse pas le rythme d'horloge d'interface. Voici un exemple de monde réel : Pendant cette sortie, aucun trafic n'est généré par l'interface : En ce moment, le trafic est commencé. Le flot est non-bursty à 400PPS, les trames consistantes de l'octet of 1000 : Avis comment, par la suite, avec un flot non-bursty, la profondeur moyenne de file d'attente WRED égalera la profondeur de la file d'attente en cours, qui est le comportement prévu.

- **HQF ? bande passante ? + ? foire-file d'attente ? comportement** : Quand la bande passante et la foire-file d'attente sont appliquées ensemble à une classe définie par l'utilisateur HQF, chaque file d'attente basée sur écoulement est allouée une queue-limit égale à  $.25 * \text{queue-limit}$ . Puisque la queue-limit par défaut est 64 paquets, chaque file d'attente basée par écoulement dans une foire-file d'attente sera allouée 16 paquets. Si quatre écoulements traversaient cette classe, par défaut chaque file d'attente de flux aurait 16 paquets, donc vous ne compteriez jamais voir les paquets totaux mis en file d'attente de  $>64 (4 * 16)$ . Toutes les pertes de destination d'une file d'attente de flux individuelle sont enregistrées comme flowdrops. Si le nombre de files d'attente de flux étaient sensiblement élevé de même que la queue-limit, alors une autre occasion pour la NO--mémoire tampon chute. Par exemple, la stratégie arrogante attache point est une interface physique, où 1000 mémoires tampons d'agrégat sont allouées : Dans cette configuration, le trafic appréciable dans toutes les files d'attente de flux peut mourir de faim les mémoires tampons d'agrégat et le résultat d'interface dans des baisses de NO--mémoire tampon dans d'autres classes définies par l'utilisateur (voir l'ID de bogue Cisco CSCsw98427). C'est parce que 1024 files d'attente de flux, chacun avec une queue-limit de 32 paquets peuvent facilement oversubscribe l'allocation de queue de mémoire tampon basée par classe d'agrégat de l'interface 1000.
- **HQF ? bande passante ? + ? random-detect ? + ? foire-file d'attente ? comportement** : Exemple : Même que la bande passante et la foire-file d'attente dans la section à moins que la taille de file d'attente de moyenne WRED soit calculée chaque fois un paquet arrive pour décider si le paquet devrait être relâchée aléatoire ou queue relâchée. Comme avec le pre-HQF, toutes les files d'attente de flux partageront un exemple des seuils WRED, signifiant tous les paquets mis en file d'attente à toutes les files d'attente de flux sont utilisés pour calculer la profondeur moyenne de file d'attente WRED, puis la décision de baisse applique le minimum et des seuils maximaux WRED contre les paquets d'agrégat dans toutes les files d'attente de flux. Cependant, un autre départ à la bande passante et une foire-file d'attente dans la section, parce qu'un exemple des seuils WRED s'appliquent à toutes les files d'attente basées sur écoulement, les différentes files d'attente de flux ? queue-limit ( $.25 * ? \text{queue-limit} ?$ ) est ignorée et honore à la place la queue-limit d'agrégat de classes pour un contrôle de limite en cours de file d'attente.

## [Comportement par défaut de classe](#)

## [Comportement de Pre-HQF](#)

Dans le pre-HQF, la foire-file d'attente par défaut de par défaut de classe, toutes les files d'attente de flux partagent la queue-limit pour la classe (le par défaut est 64), et il n'y a aucune réservation de bande passante. En d'autres termes, le nombre total de paquets mis en file d'attente dans toutes les files d'attente de flux ne dépassera jamais la queue-limit. La quantité de paquets mis en file d'attente dans chaque file d'attente de flux dépendra du poids calculé de la file d'attente de flux. Réciproquement, si la foire-file d'attente et le random-detect sont utilisés ensemble dans le classe-par défaut, la queue-limit sera ignorée et toutes les files d'attente de flux partageront les mêmes seuils WRED. En soi, tous les paquets actuellement mis en file d'attente dans toutes les files d'attente de flux seront utilisés pour calculer la taille de file d'attente de moyenne WRED. Puisque la taille en cours de file d'attente n'a aucune limite supérieure dans cette configuration, l'occasion pour des baisses de NO--mémoire tampon est élevée (référez-vous à l'ID de bogue Cisco CSCsm94757).

- Ajouter le classe-par défaut de bande passante aura comme conséquence le comportement tracé les grandes lignes dans le [comportement de Pre-HQF - les classes définies par l'utilisateur configurées avec la section de commande de « bande passante »](#).
- Ajouter la bande passante et le random-detect pour classer le classe-par défaut aura comme conséquence le comportement tracé les grandes lignes dans le *Pre-HQF ? bande passante ? + ? random-detect ?* section de *comportement de* [comportement de Pre-HQF - Classes définies par l'utilisateur configurées avec la commande de « bande passante »](#).

**Remarque:** Dans le pre-HQF, la bande passante ne peut pas coexister avec la foire-file d'attente dans le classe-par défaut.

## [Comportement HQF](#)

Dans HQF, des par défaut par défaut de classe à une file d'attente FIFO et est alloués une pseudo réservation de bande passante basée sur les allocations de surplus des classes définies par l'utilisateur. En soi, parce que comportement par défaut de classe-par défaut de classe HQF, voyez le [comportement HQF - les classes définies par l'utilisateur configurées avec la section de commande de « bande passante »](#). À tout moment, indépendamment de la configuration, le classe-par défaut de classe dans des images HQF aura toujours une réservation de bande passante implicite égale à la bande passante inutilisée d'interface non consommée par les classes définies par l'utilisateur. Par défaut, la classe par défaut reçoit un minimum de 1% de la bande passante de forme d'interface ou de parent. Il est également possible de configurer explicitement la bande passante CLI dans le par défaut de classe.

- Si la foire-file d'attente est configurée dans le Classe-par défaut de classe, le comportement apparie-t-il le *HQF ? bande passante ? + ? foire-file d'attente ?* section de *comportement de* [comportement HQF - Classes définies par l'utilisateur configurées avec la commande de « bande passante »](#).
- Si la foire-file d'attente et le random-detect sont configurés ensemble dans le Classe-par défaut, le comportement apparie-t-il le *HQF ? bande passante ? + ? random-detect ? + ? foire-file d'attente ?* section de *comportement de* [comportement HQF - Classes définies par l'utilisateur configurées avec la commande de « bande passante »](#).

## [Informations connexes](#)

- [Solutions guide de configuration de Qualité de service Cisco IOS, version 12.4T](#)
- [QoS-----Cadre de queue hiérarchique \(HQF\)](#)

- [Assistance technique sur la technologie QoS](#)
- [Assistance produit routeurs](#)
- [Support et documentation techniques - Cisco Systems](#)