

Comprendre et configurer MDRR/WRED sur les routeurs de la gamme 12000

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Conventions](#)

[Informations générales](#)

[Présentation de MDRR](#)

[File d'attente prioritaire de MDRR](#)

[Exemple MDRR](#)

[Support MDRR par type de moteur](#)

[Présentation WRED](#)

[Utiliser la syntaxe CoS existante pour la configuration](#)

[Utiliser l'interface de ligne de commande QoS modulaire \(MQC\) pour la configuration](#)

[Commandes de surveillance de la gestion et de l'évitement de la congestion](#)

[Commande show interfaces](#)

[Commande show interfaces {number} random](#)

[Commande show controller frfab queue {port} du logement exec \(y\)](#)

[Commande show controller frfab QM stat du logement exec \(y\)](#)

[Surveillance de la gestion des encombrements entrants](#)

[Commande show interfaces](#)

[Commande show controller tofab queue du logement exec \(x\)](#)

[Commande show controller tofab queue \(slot\) \(port\) du logement exec \(x\)](#)

[Commande show controller tofab QM stat du logement exec \(x\)](#)

[Étude de cas](#)

[Informations connexes](#)

[Introduction](#)

Ce document explique comment configurer les fonctions de gestion de la congestion et d'évitement de congestion du logiciel Cisco IOS® sur le routeur Internet de la gamme Cisco 12000.

Après avoir lu ce document, vous devez être en mesure de :

- Comprenez pourquoi il est important de configurer MDRR (Modified Deficit Round Robin) et WRED (Weighted Random Early Detection) dans votre réseau principal.

- Comprendre la mise en oeuvre qui sous-tend MDRR et WRED sur la gamme Cisco 12000.
- Configurez MDRR et WRED à l'aide de la syntaxe CoS (Class of Service) héritée et de la CLI QoS modulaire (MQC).

Conditions préalables

Conditions requises

Les lecteurs de ce document devraient avoir connaissance des sujets suivants :

- Connaissance générale de l'architecture des routeurs Internet de la gamme Cisco 12000.
- En particulier, une connaissance de l'architecture de mise en file d'attente et des termes suivants : ToFab (Vers le fabric) : décrit les files d'attente côté réception sur une carte de ligne entrante. FrFab (From the fabric) : décrit les files d'attente côté transmission sur une carte de ligne sortante.

Remarque : Il est également recommandé de rechercher [Comment lire la sortie de la fenêtre show controller | commandes tofab queue sur un routeur Internet de la gamme Cisco 12000](#).

Components Used

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

- Toutes les plates-formes Cisco 12000, dont 12008, 12012, 12016, 12404, 12406, 12410 et 12416.
- Logiciel Cisco IOS Version 12.0(24)S1.

Remarque : Bien que les configurations de ce document aient été testées sur le logiciel Cisco IOS Version 12.0(24)S1, toute version du logiciel Cisco IOS prenant en charge le routeur Internet de la gamme Cisco 12000 peut être utilisée.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

For more information on document conventions, refer to the [Cisco Technical Tips Conventions](#).

Informations générales

Les méthodes de mise en file d'attente définissent le mécanisme de planification des paquets ou l'ordre dans lequel les paquets sont retirés de la file d'attente à l'interface pour transmission sur le câble physique. En fonction de l'ordre et du nombre de fois où une file d'attente est traitée par une fonction de planification, les méthodes de mise en file d'attente prennent également en charge les garanties de bande passante minimale et les faibles latences.

Il est important de s'assurer qu'un mécanisme de planification des paquets prend en charge l'architecture de commutation sur laquelle il est mis en oeuvre. Weighted Fair Queuing (WFQ) est

l'algorithme de planification bien connu pour l'allocation des ressources sur les plates-formes de routeurs Cisco avec une architecture basée sur le bus. Cependant, il n'est pas pris en charge sur les routeurs Internet de la gamme Cisco 12000. La mise en file d'attente par priorité du logiciel Cisco IOS traditionnel et la mise en file d'attente personnalisée ne sont pas non plus prises en charge. Au lieu de cela, la gamme Cisco 12000 utilise une forme spéciale de mise en file d'attente appelée MDRR (Modified Deficit Round Robin), qui fournit des garanties de bande passante relative ainsi qu'une file d'attente à faible latence. Le M du MDRR signifie « modifié »; il ajoute la file d'attente prioritaire par rapport à DRR où aucune file d'attente prioritaire n'est présente. Pour plus d'informations sur MDRR, reportez-vous à la section [Vue d'ensemble de MDRR](#).

En outre, la gamme Cisco 12000 prend en charge la détection WRED (Weighted Random Early Detection) en tant que stratégie de rejet dans les files d'attente MDRR. Ce mécanisme d'évitement de congestion offre une alternative au mécanisme de suppression de queue par défaut. La congestion peut être évitée par des chutes contrôlées.

Les mécanismes d'évitement de congestion et de gestion tels que WRED et MDRR sont particulièrement importants sur les files d'attente FrFab des interfaces sortantes à faible débit, telles que les cartes de ligne multicanaux fractionnés (LC). La structure de commutation à haut débit peut livrer des paquets aux groupes de canaux beaucoup plus rapidement que les groupes de canaux ne peuvent les transmettre. Comme la mise en file d'attente / les tampons sont gérés au niveau du port physique, la contre-pression sur un canal peut affecter tous les autres canaux sur ce port. Il est donc très important de gérer cette contre-pression par le biais de WRED/MDRR, qui limite l'impact de la contre-pression sur le ou les canaux en question. Pour plus d'informations sur la gestion de la sursouscription d'interface sortante, consultez [Dépannage des paquets ignorés et des pertes de mémoire sur le routeur Internet de la gamme Cisco 12000](#).

[Présentation de MDRR](#)

Cette section fournit une vue d'ensemble de MDRR (Modified Deficit Round Robin).

Avec MDRR configuré comme stratégie de mise en file d'attente, les files d'attente non vides sont servies l'une après l'autre, de manière circulaire. Chaque fois qu'une file d'attente est servie, une quantité fixe de données est supprimée de la file d'attente. L'algorithme gère ensuite la file d'attente suivante. Lorsqu'une file d'attente est servie, MDRR suit le nombre d'octets de données retirés de la file d'attente au-delà de la valeur configurée. Au cours de la prochaine étape, lorsque la file d'attente est de nouveau servie, moins de données seront retirées de la file d'attente pour compenser l'excès de données qui a été servi précédemment. Par conséquent, la quantité moyenne de données retirées de la file d'attente par file d'attente sera proche de la valeur configurée. En outre, MDRR maintient une file d'attente prioritaire qui est desservie de façon préférentielle. Le RMR est expliqué plus en détail dans cette section.

Chaque file d'attente dans MDRR est définie par deux variables :

- Valeur quantique : nombre moyen d'octets servis dans chaque cycle.
- Compteur de déficit - Permet de suivre le nombre d'octets transmis par une file d'attente dans chaque cycle. Il est initialisé à la valeur quantique.

Les paquets d'une file d'attente sont servis tant que le compteur de déficit est supérieur à zéro. Chaque paquet servi réduit le compteur de déficit d'une valeur égale à sa longueur en octets. Une file d'attente ne peut plus être servie après que le compteur de déficit devient zéro ou négatif. Dans chaque nouveau cycle, le compteur de déficit de chaque file d'attente non vide est augmenté de sa valeur quantique.

Remarque : En général, la taille quantique d'une file d'attente ne doit pas être inférieure à l'unité de transmission maximale (MTU) de l'interface. Cela garantit que le planificateur sert toujours au moins un paquet de chaque file d'attente non vide.

Chaque file d'attente MDRR peut se voir attribuer un poids relatif, l'une des files d'attente du groupe étant définie comme file d'attente prioritaire. Les poids attribuent une bande passante relative à chaque file d'attente lorsque l'interface est congestionnée. L'algorithme MDRR supprime les données de chaque file d'attente de manière circulaire s'il y a des données dans la file d'attente à envoyer.

Si toutes les files d'attente MDRR normales contiennent des données, elles sont traitées comme suit :

0-1-2-3-4-5-6-0-1-2-3-4-5-6 ...

Au cours de chaque cycle, une file d'attente peut démettre une file d'attente quantique en fonction de son poids configuré. Sur les cartes de ligne Engine 0 et Engine 2, une valeur de 1 équivaut à donner à l'interface un poids de son MTU. Pour chaque incrément supérieur à 1, le poids de la file d'attente augmente de 512 octets. Par exemple, si la MTU d'une interface particulière est 4470 et que le poids d'une file d'attente est configuré sur 3, chaque fois que la rotation se produit, $4470 + (3-1)*512 = 5494$ octets sont autorisés à être retirés de la file d'attente. Si deux files d'attente DRR normales, Q0 et Q1, sont utilisées, Q0 est configuré avec un poids de 1 et Q1 avec un poids de 9. Si les deux files d'attente sont encombrées, à chaque fois que la rotation se produit, Q0 peut envoyer 4 470 octets et Q1 peut envoyer $[4\ 470 + (9-1)*512] = 8\ 566$ octets. Cela donnerait au trafic qui va à Q0 environ 1/3 de la bande passante, et au trafic qui passe par Q1 environ 2/3 de la bande passante.

Remarque : La formule standard de retrait de file d'attente utilisée pour calculer l'affectation de bande passante MDRR est $D = MTU + (weight-1)*512$. Dans les versions antérieures à la version 12.0(21) S/ST du logiciel Cisco IOS, les cartes de ligne du moteur 4 utilisaient une autre formule de retrait de file d'attente. Afin de configurer le poids MDRR pour une affectation correcte de bande passante, assurez-vous d'exécuter une version du logiciel Cisco IOS postérieure à 12.0(21) S/ST.

Remarque : La formule quantique pour les cartes de ligne Engine 4+ est (pour la direction toFab, ce n'est pas valide pour FrFab) $Quantum = BaseWeight + \{BaseWeight * (QueueWeight - 1) * 512\} / MTU$. La valeur BaseWeight est obtenue avec la formule suivante : $PoidsBase = \{(MTU + 512 - 1) / 512\} + 5$

Note : Tous les calculs sont arrondis ; autrement dit, tous les décimales sont ignorés.

Remarque : Pour savoir si une carte de ligne de moteur spécifique prend en charge MDRR, consultez [Support MDRR par type de moteur](#).

[File d'attente prioritaire de MDRR](#)

La gamme Cisco 12000 prend en charge une file d'attente prioritaire (PQ) dans MDRR. Cette file d'attente fournit le faible délai et la faible gigue requis par le trafic sensible au temps, tel que la voix sur IP (VoIP).

Comme indiqué ci-dessus, la gamme Cisco 12000 ne prend pas en charge la mise en file d'attente pondérée (WFQ). Ainsi, la PQ interne à MDRR fonctionne différemment de la fonctionnalité LLQ

(Low Latency Queueing) du logiciel Cisco IOS disponible pour d'autres plates-formes.

Une différence clé est la façon dont le planificateur MDRR peut être configuré pour traiter le PQ dans l'un des deux modes, comme indiqué dans le [tableau 1](#) :

Tableau 1 - Configuration du planificateur MDRR pour le service de la file d'attente de service en deux modes

	Mode alternatif	Mode Priorité stricte
Avantages	Ici, le PQ est traité entre les autres files d'attente. En d'autres termes, le planificateur MDRR gère alternativement la file d'attente PQ et toute autre file d'attente configurée.	Ici, le PQ est réparé chaque fois qu'il n'est pas vide. Ceci fournit le délai le plus faible possible pour le trafic correspondant.
Inconvénients	Ce mode peut introduire de la gigue et du retard par rapport au mode de priorité stricte.	Ce mode peut affamer d'autres files d'attente, en particulier si les flux correspondants sont des expéditeurs agressifs.

Le mode alternatif peut exercer moins de contrôle sur la gigue et le délai. Si le planificateur MDRR commence à traiter des trames de taille MTU à partir d'une file d'attente de données, puis qu'un paquet vocal arrive dans le PQ, le planificateur en mode alternatif sert complètement la file d'attente non prioritaire jusqu'à ce que son compteur de déficit atteigne zéro. Pendant ce temps, le PQ n'est pas traité et les paquets VoIP sont retardés.

En revanche, en mode de priorité stricte, le planificateur ne gère que le paquet non prioritaire actuel, puis passe au PQ. Le planificateur commence à traiter une file d'attente non prioritaire uniquement après que le PQ soit complètement vide.

Il est important de noter que la file d'attente prioritaire en mode de priorité alternative est traitée plusieurs fois dans un cycle, et prend donc plus de bande passante que les autres files d'attente avec le même poids nominal. Le nombre de files d'attente définies dépend du nombre de files d'attente. Par exemple, avec trois files d'attente, la file d'attente à faible latence est traitée deux fois plus souvent que les autres files d'attente et envoie deux fois son poids par cycle. Si huit files d'attente sont définies, la file d'attente à faible latence est traitée sept fois plus souvent et le poids effectif est sept fois plus élevé. Ainsi, la bande passante que la file d'attente peut prendre est liée à la fréquence à laquelle elle est servie par round robin, qui dépend à son tour du nombre de files d'attente définies dans l'ensemble. En mode de priorité alternative, la file d'attente prioritaire est généralement configurée avec un poids faible pour cette raison particulière.

Par exemple, supposons que quatre files d'attente sont définies : 0, 1, 2 et la file d'attente prioritaire. En mode de priorité alternative, si toutes les files d'attente sont congestionnées, elles sont traitées comme suit : 0, llq, 1, llq, 2, llq, 0, llq, 1, où llq signifie file d'attente à faible latence.

Chaque fois qu'une file d'attente est traitée, elle peut envoyer jusqu'à son poids configuré. Par conséquent, la bande passante minimale que peut avoir la file d'attente à faible latence est :

- WL = poids de la file d'attente à faible latence.
- W0, W1, ... Wn = poids des files d'attente DRR régulières.
- n = nombre de files d'attente DRR régulières utilisées pour cette interface.
- BW = bande passante de la liaison.

Pour le mode de priorité alternative, bande passante minimale de la file d'attente à faible latence = $BW * n * WL / (n * WL + \text{Sum}(W0, Wn))$.

Le poids est le seul paramètre de variable dans MDRR qui peut être configuré. Elle influence la quantité relative de bande passante qu'une classe de trafic peut utiliser et la quantité de trafic envoyée en un seul tour. L'utilisation de poids plus importants signifie que le cycle global prend plus de temps et peut-être augmente la latence.

Directives de configuration

- Il est préférable de configurer le poids de la classe qui a le besoin de bande passante le plus faible à 1 afin de maintenir le délai et la gigue aussi bas que possible parmi les autres classes.
- Sélectionnez les valeurs de poids les plus basses possible. Commencez par un poids de 1 pour la classe avec la bande passante la plus faible. Par exemple, lorsque vous utilisez deux classes avec une bande passante de 50 % pour chaque classe, vous devez configurer 1 et 1. Il n'est pas logique d'utiliser 10 et 10, car il n'y a aucun impact sur les performances lorsque vous choisissez 1. En outre, un poids plus élevé introduit plus de gigue.
- Une valeur de poids faible pour la LLQ est très importante, surtout en mode alternatif afin de ne pas ajouter trop de retard ou de gigue aux autres classes.

Exemple MDRR

L'exemple de cette section provient de *l'architecture logicielle Cisco IOS®*, Cisco Press.

Supposez que nous avons trois files d'attente :

- **File d'attente 0** - a un volume de 1 500 octets ; il s'agit de la file d'attente à faible latence, configurée pour fonctionner en mode alternatif.
- **File d'attente 1** - a un volume de 3000 octets.
- **File d'attente 2** - a un volume de 1 500 octets.

[La Figure 1](#) illustre l'état initial des files d'attente, ainsi que certains paquets reçus et mis en file d'attente.

Figure 1 : état initial de MDRR

Queues					Deficit Counters	
Queue 0			3/250	2/1500	1/250	0
Queue 1			6/1500	5/1500	4/1500	0
Queue 2	11/1500	10/250	9/250	8/250	7/250	0

La file d'attente 0 est traitée en premier, son volume est ajouté à son compteur de déficit, le paquet 1, qui est de 250 octets, est transmis et sa taille est soustraite du compteur de déficit. Comme le compteur de déficit de la file d'attente 0 est toujours supérieur à 0 ($1500 - 250 = 1250$), le paquet 2 est également transmis et sa longueur est soustraite du compteur de déficit. Le compteur de déficit de la file d'attente 0 est maintenant -250, de sorte que la file d'attente 1 est traitée ensuite. [La Figure 2](#) indique cet état.

Figure 2 - État suivant du MDRR

Queues					Deficit Counters	
Queue 0					3/250	-250
Queue 1						0
Queue 2	11/1500	10/250	9/250	8/250	7/250	0

Le compteur de déficit de la file d'attente 1 est défini sur 3 000 ($0 + 3\ 000 = 3\ 000$) et les paquets 4 et 5 sont transmis. Avec chaque paquet transmis, soustrayez la taille du paquet du compteur de déficit, de sorte que le compteur de déficit de la file d'attente 1 est réduit à 0. [La figure 3](#) illustre cet état.

Figure 3 - État MDRR lorsque le compteur de déficit de la file d'attente 1 est zéro

Queues					Deficit Counters	
Queue 0					3/250	-250
Queue 1					6/1500	0
Queue 2	11/1500	10/250	9/250	8/250	7/250	0

Nous devons revenir du mode de priorité secondaire à la file d'attente de service 0. Là encore, le quantique est ajouté au compteur de déficit actuel, et le compteur de déficit de la file d'attente 0 est défini sur le résultat ($-250 + 1500 = 1250$). Le paquet 3 est maintenant transmis, car le compteur de déficit est supérieur à 0 et la file d'attente 0 est vide. Lorsqu'une file d'attente est vidée, son compteur de déficit est défini sur 0, comme illustré à la [Figure 4](#).

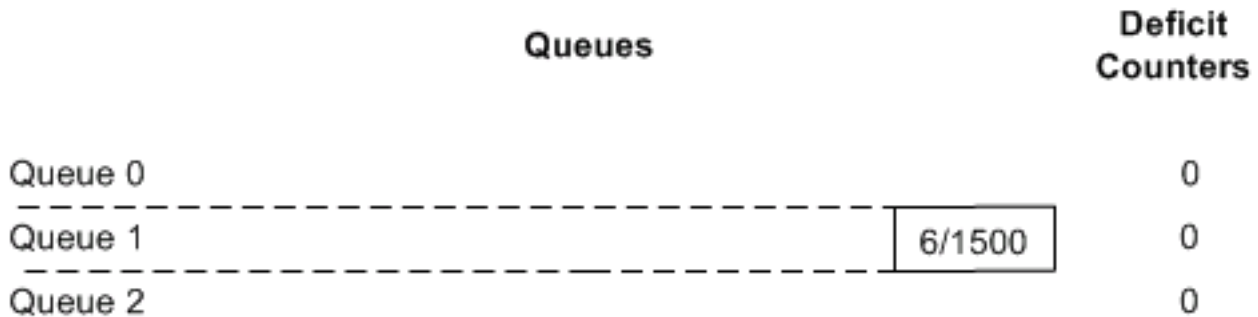
Figure 4 - État MDRR lorsqu'une file d'attente est vidée

Queues					Deficit Counters	
Queue 0						0
Queue 1					6/1500	0
Queue 2	11/1500	10/250	9/250	8/250	7/250	0

La file d'attente 2 est traitée ensuite ; son compteur de déficit est réglé sur 1 500 (0 + 1 500 = 1 500). Les paquets 7 à 10 sont transmis, ce qui laisse le compteur de déficit à 500 (1500 - (4*250) = 500). Comme le compteur de déficit est toujours supérieur à 0, le paquet 11 est également transmis.

Lorsque le paquet 11 est transmis, la file d'attente 2 est vide et son compteur de déficit est défini sur 0, comme illustré à la [Figure 5](#).

Figure 5 - État MDRR lors de la transmission du paquet 11



La file d'attente 0 est de nouveau traitée ultérieurement (car nous sommes en mode de priorité alternative). Comme il est vide, nous servons ensuite la file d'attente 1 et transmettons le paquet 6.

[Support MDRR par type de moteur](#)

La gamme Cisco 12000 prend en charge cinq modèles de cartes de ligne avec des architectures de moteur de transfert de couche 3 (L3) uniques. La prise en charge de MDRR varie selon le type de moteur L3 et le type de carte. Par exemple, il n'existe aucune prise en charge de MDRR et WRED sur les cartes de ligne ATM du moteur 0. Vous pouvez utiliser la commande **show diag** pour déterminer le type de moteur L3 de vos cartes de ligne installées :

```
router#show diags | include (SLOT | Engine)
!--- The regular expression is case-sensitive. ... SLOT 1 (RP/LC 1 ): 1 port ATM Over SONET
OC12c/STM-4c Multi Mode L3 Engine: 0 - OC12 (622 Mbps) SLOT 3 (RP/LC 3 ): 3 Port Gigabit
Ethernet L3 Engine: 2 - Backbone OC48 (2.5 Gbps)
```

[Files d'attente MDRR on ToFab \(Rx\)](#)

Vous pouvez utiliser la syntaxe CoS existante ou l'interface de ligne de commande QoS modulaire pour configurer MDRR sur la gamme Cisco 12000. Les sections suivantes de ce document expliquent comment configurer MDRR avec les anciennes CoS ou QoS modulaires. Vous devez configurer les files d'attente ToFab avec la syntaxe CoS héritée uniquement car elles ne prennent pas en charge la syntaxe plus récente de MQC. Voir [le tableau 2](#) pour plus de détails.

Tableau 2 - Détails sur MDRR dans les files d'attente ToFab (Rx)

	Lorsque mis en oeuvre	ToFab MDRR	ToFab PQ alternatif	PQ stricte ToFab	ToFab WRED
Eng0	le	Non**	Non**	Oui	Oui

	logiciel Cisco IOS				
Eng1	-	Non	Non	Non	Non
Eng2	Matériel	Oui	Oui	Oui	Oui
Eng3	Matériel	Non	Oui	Oui	Oui
Eng4	Matériel	Oui	Oui	Oui	Oui
Eng4+	Matériel	Oui	Oui	Oui	Oui

** MDRR est pris en charge sur les LC du moteur 0 dans la direction ToFab (Rx), mais seulement le mode de priorité stricte, pas le mode de priorité alternative. Les sept files d'attente restantes sont prises en charge comme d'habitude.

Les interfaces entrantes gèrent une file d'attente de sortie virtuelle distincte par LC de destination. La façon dont ces files d'attente sont implémentées dépend du type de moteur L3.

- Moteur 0 : les LC entrantes gèrent huit files d'attente par emplacement de destination. Ainsi, le nombre maximal de files d'attente est $16 \times 8 = 128$. Chaque file d'attente peut être configurée séparément.
- Moteurs 2, 3, 4 et 4+ - Les LC entrants gèrent huit files d'attente par interface de destination. Avec 16 emplacements de destination et 16 interfaces par emplacement, le nombre maximal de files d'attente est de $16 \times 16 \times 8 = 2048$. Toutes les interfaces d'un emplacement de destination doivent utiliser les mêmes paramètres.

MDRR sur les files d'attente FrFab (Tx)

MDRR sur les files d'attente FrFab fonctionne de manière cohérente, qu'elle soit implémentée dans le matériel ou le logiciel. Tous les types de moteur de couche 3 prennent en charge huit files d'attente de classe pour chaque interface sortante. Voir [le tableau 3](#) pour plus de détails.

Tableau 3 - Détails sur MDRR dans les files d'attente FrFab (Tx)

	Lorsque mis en oeuvre	FrFab PQ alternatif	FrFab PQ strict	FrFab WRED
Eng0	Logiciel ¹	Non	Oui	Oui
Eng1	-	Non	Non	Non
Eng2	Matériel	Oui ²	Oui	Oui
Eng3	Matériel	Non	Oui	Oui
Eng4	Matériel	Oui	Oui	Oui
Eng4+	Matériel	Oui	Oui	Oui

¹ La prise en charge de MDRR sur les files d'attente FrFab des LC du moteur 0 est introduite dans ces versions du logiciel Cisco IOS :

- Logiciel Cisco IOS Version 12.0(10)S - 4xOC3 et 1xOC12 POS, 4xOC3 et 1xCHOC12/STM4.
- Logiciel Cisco IOS Version 12.0(15)S - 6xE3 et 12xE3.
- Logiciel Cisco IOS Version 12.0(17)S - 2xCHOC3/STM1.

² Vous devez configurer un autre MDRR dans la direction FrFab avec la syntaxe CoS héritée.

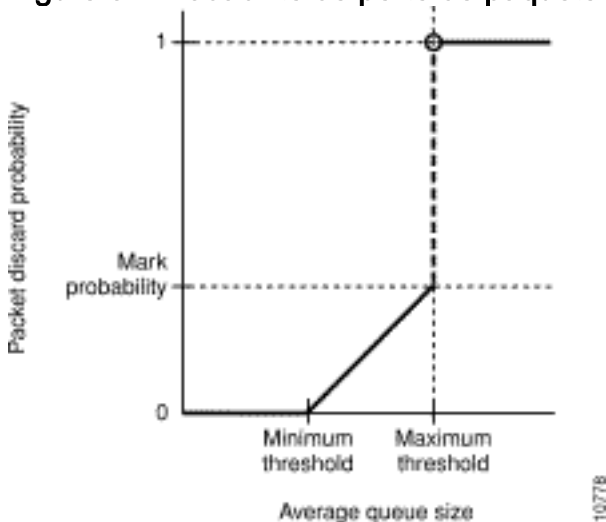
Remarque : La LC 3xGE prend en charge MDRR sur les files d'attente ToFab et, à partir de la version 12.0(15)S du logiciel Cisco IOS, sur les files d'attente FrFab avec deux restrictions, à savoir une quantité fixe et une file d'attente CoS unique pour chaque interface. La file d'attente prioritaire prend en charge un nombre quantique qui peut être configuré, ainsi que des modes de priorité stricte et alternative. Les trois interfaces partagent une seule PQ.

Remarque : Reportez-vous aux notes de version des routeurs de la gamme Cisco 12000 pour obtenir les dernières informations sur les fonctions QoS prises en charge sur les LC de la gamme Cisco 12000.

Présentation WRED

La détection WRED (Weighted Random Early Detection) est conçue pour éviter les effets néfastes de la congestion des interfaces sur le débit du réseau.

Figure 6 - Probabilité de perte de paquets WRED



Reportez-vous à [Détection précoce pondérée aléatoire sur le routeur de la gamme Cisco 12000](#) pour obtenir une explication des paramètres WRED. Les paramètres de probabilité minimale, maximale et de marque décrivent la courbe de détection précoce aléatoire (RED) réelle. Lorsque la moyenne pondérée de la file d'attente est inférieure au seuil minimal, aucun paquet n'est abandonné. Lorsque la moyenne pondérée de la file d'attente est supérieure au seuil de file d'attente maximum, tous les paquets sont abandonnés jusqu'à ce que la moyenne tombe en dessous du seuil maximum. Lorsque la moyenne est comprise entre les seuils minimum et maximum, la probabilité que le paquet soit abandonné peut être calculée par une ligne droite du seuil minimum (probabilité de perte égale à 0) au seuil maximum (probabilité de perte égale au dénominateur de probabilité 1/point).

La différence entre RED et WRED est que WRED peut rejeter sélectivement le trafic de priorité inférieure lorsque l'interface commence à être congestionnée et peut fournir des caractéristiques de performances différenciées pour différentes classes de service (CoS). Par défaut, WRED utilise un profil RED différent pour chaque poids (priorité IP - 8 profils). Il abandonne les paquets moins importants de manière plus agressive que les paquets plus importants.

Il est difficile d'ajuster les paramètres WRED pour gérer la profondeur de la file d'attente et dépend de nombreux facteurs, notamment :

- Charge de trafic et profil offerts.
- Rapport charge/capacité disponible.
- Comportement du trafic en présence de congestion.

Ces facteurs varient d'un réseau à l'autre et dépendent, à leur tour, des services offerts et des clients qui utilisent ces services. Nous ne pouvons donc pas formuler de recommandations qui s'appliquent à des environnements client spécifiques. Cependant, [le tableau 4](#) décrit les valeurs généralement recommandées en fonction de la bande passante de la liaison. Dans ce cas, nous ne différencions pas les caractéristiques de décrochage entre les différentes classes de service.

Tableau 4 - Valeurs recommandées basées sur la bande passante de la liaison

Bande passante	BW théorique (Kbits/s)	BW physique ¹ (Kbits/s)	Seuil minimal	Seuil maximal
OC3	155000	149760	94	606
OC12	622000	599040	375	2423
OC48	2400000	239616	1498	9690
OC192	10000000	9584640	5991	38759

¹ Taux SONET physique

Plusieurs contraintes sont prises en compte pour calculer les valeurs de seuil ci-dessus. Par exemple, la maximisation de l'utilisation de la liaison tout en minimisant la profondeur moyenne de la file d'attente, la différence entre les valeurs Maximum et Minimum doit être une puissance de deux (en raison de la limitation matérielle). D'après l'expérience et la simulation, la profondeur instantanée maximale d'une file d'attente contrôlée par RED est inférieure à 2 MaxTh. Pour OC48 et plus, 1 MaxTh, etc. Cependant, la détermination exacte de ces valeurs dépasse le cadre de ce document.

Remarque : Il n'est pas nécessaire de configurer la valeur constante de pondération exponentielle sur les cartes de ligne du moteur 2 et au-dessus, car l'échantillonnage de file d'attente matérielle est utilisé à la place. Pour les LC du moteur 0, ces valeurs peuvent être configurées :

- ds3 - 9
- OC3 - 10
- 12 - 12

Remarque : WRED n'est pas pris en charge sur les LC du moteur 1.

Comme l'expliquent les sections suivantes, vous pouvez utiliser la syntaxe CoS héritée et la syntaxe MQC plus récente pour configurer WRED.

[Utiliser la syntaxe CoS existante pour la configuration](#)

La syntaxe CoS héritée de la gamme Cisco 12000 utilise un modèle **cos-queue-group** pour définir un ensemble de définitions CoS. Vous appliquez ensuite le modèle aux files d'attente ToFab et FrFab sur les interfaces entrantes ou sortantes, respectivement.

[Étape 1 : Définir un cos-queue-group](#)

La commande **cos-queue-group** crée un modèle nommé de paramètres MDRR et WRED. Voici

les paramètres de configuration disponibles dans l'interface de ligne de commande :

```
Router(config)#cos-queue-group oc12
Router(config-cos-que)#?
Static cos queue commands:
```

default	Set a command to its defaults
dscp	Set per DSCP parameters, Engine 3 only
exit	Exit from COS queue group configuration mode
exponential-weighting-constant	Set group's RED exponential weight constant. (Not used by engine 0, 1 or 2 line cards)
no	Negate a command or set its defaults
precedence	Set per precedence parameters
queue	Set individual queue parameters
random-detect-label	Set RED drop criteria
traffic-shape	Enable Traffic Shaping on a COS queue group

Avec MDRR, vous pouvez mapper la priorité IP aux files d'attente MDRR et configurer la file d'attente spéciale à faible latence. Vous pouvez utiliser le paramètre de priorité sous la commande **cos-queue-group** pour ceci :

```
precedence <0-7> queue [ <0-6> | low-latency]
```

Vous pouvez mapper une priorité IP particulière à une file d'attente MDRR normale (file d'attente 0 à 6) ou vous pouvez la mapper à la file d'attente prioritaire. La commande ci-dessus vous permet de mapper plusieurs précédents IP à la même file d'attente.

Remarque : Il est recommandé d'utiliser la priorité 5 pour la file d'attente à faible latence. La priorité 6 est utilisée pour les mises à jour de routage.

Vous pouvez attribuer un poids relatif à chaque file d'attente MDRR, l'une des files d'attente du groupe étant définie comme file d'attente prioritaire. Vous pouvez utiliser la commande **queue** sous **cos-queue-group** pour effectuer ceci :

```
queue <0-6> <1-2048>
queue low-latency [alternate-priority | strict-priority] <1-2048>
!--- The weight option is not available with strict priority.
```

Utilisez la commande **cos-queue-group** pour définir les paramètres WRED :

```
random-detect-label
```

Voici un exemple de **cos-queue-group** nommé **oc12**. Il définit trois classes de trafic : file d'attente 0, 1 et faible latence. Il mappe les valeurs de priorité IP 0 à 3 à la file d'attente 0, les valeurs de priorité 4, 6 et 7 à la file d'attente 1 et la priorité 5 à la file d'attente à faible latence. La file d'attente 1 reçoit plus de bande passante.

Exemple de configuration

```

cos-queue-group oc12
!--- Creation of cos-queue-group called "oc12".

precedence 0 queue 0
  !--- Map precedence 0 to queue 0. precedence 0 random-
detect-label 0 !--- Use RED profile 0 on queue 0.
precedence 1 queue 0 precedence 1 random-detect-label 0
precedence 2 queue 0 precedence 2 random-detect-label 0
precedence 3 queue 0 precedence 3 random-detect-label 0
!--- Precedence 1, 2 and 3 also go into queue 0.
precedence 4 queue 1 precedence 4 random-detect-label 1
precedence 6 queue 1 precedence 6 random-detect-label 1
precedence 7 queue 1 precedence 7 random-detect-label 1
precedence 5 queue low-latency !--- Map precedence 5 to
special low latency queue. !--- We do not intend to drop
any traffic from the LLQ. We have an SLA !--- that
commits not to drop on this queue. You want to give it
all !--- the bandwidth it requires. Random-detect-label
0 375 2423 1 !--- Minimum threshold 375 packets, maximum
threshold 2423 packets. !--- Drop probability at maximum
threshold is 1. random-detect-label 1 375 2423 1 queue 1
20 !--- Queue 1 gets MDRR weight of 20, thus gets more
Bandwidth. queue low-latency strict-priority !--- Low
latency queue runs in strict priority mode.

```

Étape 2 : création d'une table de créneaux horaires pour les files d'attente ToFab

Pour éviter le blocage de tête de ligne, les interfaces entrantes de la gamme Cisco 12000 gèrent une file d'attente de sortie virtuelle par emplacement de destination. Accédez à une carte de ligne à l'aide de la commande attachement et exécutez la commande `execute-on show controller tofab queue` (ou entrez directement la commande `execute-on slot 0 show controllers tofab queue`) pour afficher ces files d'attente de sortie virtuelles. Un exemple de sortie capturé directement à partir de la console LC est fourni ci-dessous. Voir [Comment lire le résultat de la commande show controller tofab | commandes tofab queue sur un routeur Internet de la gamme Cisco 12000.](#)

```

LC-Slot1#show controllers tofab queues
Carve information for ToFab buffers
  SDRAM size: 33554432 bytes, address: 30000000, carve base: 30029100
  33386240 bytes carve size, 4 SDRAM bank(s), 8192 bytes SDRAM pagesize, 2 carve(s)
  max buffer data size 9248 bytes, min buffer data size 80 bytes
  40606/40606 buffers specified/carved
  33249088/33249088 bytes sum buffer sizes specified/carved
      Qnum      Head      Tail      #Qelem      LenThresh
      ----      ----      ----      -
5 non-IPC free queues:

20254/20254 (buffers specified/carved), 49.87%, 80 byte data size
1          17297    17296    20254      65535
12152/12152 (buffers specified/carved), 29.92%, 608 byte data size
2          20548    20547    12152      65535
6076/6076 (buffers specified/carved), 14.96%, 1568 byte data size
3          32507    38582    6076       65535
1215/1215 (buffers specified/carved), 2.99%, 4544 byte data size
4          38583    39797    1215       65535
809/809 (buffers specified/carved), 1.99%, 9248 byte data size
5          39798    40606    809        65535

IPC Queue:
100/100 (buffers specified/carved), 0.24%, 4112 byte data size
30      72      71      100      65535

Raw Queue:

```

```
31      0      17302      0      65535
```

ToFab Queues:

Dest

Slot

0	0	0	0	65535
1	0	0	0	65535
2	0	0	0	65535
3	0	0	0	65535
4	0	0	0	65535
5	0	17282	0	65535
6	0	0	0	65535
7	0	75	0	65535
8	0	0	0	65535
9	0	0	0	65535
10	0	0	0	65535
11	0	0	0	65535
12	0	0	0	65535
13	0	0	0	65535
14	0	0	0	65535
15	0	0	0	65535
Multicast	0	0	0	65535

LC-Slot1#

Utilisez la commande **slot-table-cos** pour mapper un **cos-queue-group** nommé à une file d'attente de sortie virtuelle de destination. Vous pouvez configurer un modèle unique **cos-queue-group** pour chaque file d'attente de sortie

```
Router(config)#slot-table-cos table1
Router(config-slot-cos)#destination-slot ?
<0-15> Destination slot number
all      Configure for all destination slots
Router(config-slot-cos)#destination-slot 0 oc48
Router(config-slot-cos)#destination-slot 1 oc48
Router(config-slot-cos)#destination-slot 2 oc48
Router(config-slot-cos)#destination-slot 3 oc48
Router(config-slot-cos)#destination-slot 4 oc12
Router(config-slot-cos)#destination-slot 5 oc48
Router(config-slot-cos)#destination-slot 6 oc48
Router(config-slot-cos)#destination-slot 9 oc3
Router(config-slot-cos)#destination-slot 15 oc48
```

Remarque : La configuration ci-dessus utilise trois modèles, nommés oc48, oc12 et oc3. La configuration pour le **groupe-file-d'attente** cos nommé oc12 est comme indiqué à l'étape 1. De même, configurez oc3 et oc48. Il est recommandé d'appliquer un modèle unique à un ensemble d'interfaces en fonction de la bande passante et de l'application.

[Étape 3 - Appliquez un cos de table de logements à une interface entrante](#)

Utilisez la commande **rx-cos-slot** pour appliquer un **slot-table-cos** à un LC.

```
Router(config)#rx-cos-slot 0 ?
WORD Name of slot-table-cos
Router(config)#rx-cos-slot 0 table1
Router(config)#rx-cos-slot 2 table1
```

Étape 4 - Appliquez un groupe cos-queue à une interface sortante

La gamme Cisco 12000 gère une file d'attente distincte par interface de sortie. Pour afficher ces files d'attente, connectez-vous à l'interface de ligne de commande de la carte de ligne. Utilisez la commande **attachement**, puis exécutez la commande **show controller frfab queue**, comme illustré ici :

```
LC-Slot1#show controller frfab queue
===== Line Card (Slot 2) =====
Carve information for FrFab buffers
SDRAM size: 16777216 bytes, address: 20000000, carve base: 2002D100
16592640 bytes carve size, 0 SDRAM bank(s), 0 bytes SDRAM pagesize, 2 carve(s)
max buffer data size 9248 bytes, min buffer data size 80 bytes
20052/20052 buffers specified/carved
16581552/16581552 bytes sum buffer sizes specified/carved
      Qnum      Head      Tail      #Qelem  LenThresh
      ----      -
5 non-IPC free queues:
  9977/9977 (buffers specified/carved), 49.75%, 80 byte data size
  1         101       10077      9977     65535
  5986/5986 (buffers specified/carved), 29.85%, 608 byte data size
  2        10078      16063      5986     65535
  2993/2993 (buffers specified/carved), 14.92%, 1568 byte data size
  3        16064      19056      2993     65535
  598/598 (buffers specified/carved), 2.98%, 4544 byte data size
  4        19057      19654      598      65535
  398/398 (buffers specified/carved), 1.98%, 9248 byte data size
  5        19655      20052      398      65535
IPC Queue:
  100/100 (buffers specified/carved), 0.49%, 4112 byte data size
  30       77         76         100      65535
Raw Queue:
  31       0         82         0        65535
Interface Queues:
  0         0         0         0        65535
  1         0         0         0        65535
  2         0         0         0        65535
  3         0         0         0        65535
```

Utilisez la commande **tx-cos** pour appliquer un modèle **cos-queue-group** à une file d'attente d'interface. Comme indiqué ici, vous appliquez le jeu de paramètres directement à l'interface ; aucune table n'est nécessaire. Dans cet exemple, *pos48* est le nom d'un jeu de paramètres.

```
Router(config)#interface POS 4/0
Router(config-if)#tx-cos ?
      WORD  Name of cos-queue-group
Router(config-if)#tx-cos pos48
```

Utilisez la commande **show cos** pour confirmer votre configuration :

```
Router#show cos
!--- Only some of the fields are visible if MDRR is configured on Inbound !--- or Outbound
interfaces. Interface Queue cos Group Gi4/0 eng2-frfab !--- TX-cos has been applied. Rx Slot
Slot Table 4 table1 !--- rx-cos-slot has been applied. Slot Table Name - table1 1 eng0-tofab 3
eng0-tofab !--- slot-table-cos has been defined. cos Queue Group - eng2-tofab !--- cos-queue-
group has been defined. Prec Red Label [min, max, prob] Drr Queue [deficit] 0 0 [6000, 15000,
```



```
1/1] 0 [10] 1 1 [10000, 20000, 1/1] 1 [40] 2 1 [10000, 20000, 1/1] 1 [40] 3 1 [10000, 20000,
1/1] 0 [10] 4 2 [15000, 25000, 1/1] 2 [80] 5 2 [15000, 25000, 1/1] 2 [80] 6 no drop low latency
7 no drop low latency
```

Remarque : L'interface de ligne de commande héritée utilise également la syntaxe de priorité pour le trafic MPLS (Multiprotocol Label Switching). Le routeur traite les bits MPLS comme s'ils étaient des bits ToS (IP Type of Service) et place les paquets appropriés dans les files d'attente correctes. Ce n'est pas du tout vrai pour MQC. La QoS MPLS n'entre pas dans le champ d'application de ce document.

Utiliser l'interface de ligne de commande QoS modulaire (MQC) pour la configuration

L'objectif de l'interface de ligne de commande QoS modulaire (MQC) de Cisco est de connecter toutes les différentes fonctionnalités QoS de manière logique, afin de simplifier la configuration des fonctionnalités QoS (Quality of Service) du logiciel Cisco IOS. Par exemple, la classification est effectuée séparément de la mise en file d'attente, de la réglementation et du formatage. Il fournit un cadre de configuration unique pour la QoS basé sur des modèles. Voici quelques points à retenir sur la configuration de MQC :

- Il peut être facilement appliqué et supprimé d'une interface.
- Il peut être facilement réutilisé (la même stratégie peut être appliquée à plusieurs interfaces).
- Il offre un cadre de configuration unique pour la QoS qui vous permet de provisionner, surveiller et dépanner facilement.
- Il fournit un niveau supérieur d'abstraction.
- Il est indépendant de la plate-forme.

Sur la gamme Cisco 12000, les commandes MQC peuvent être utilisées à la place de la syntaxe CoS (Class of Service) existante.

La prise en charge de MQC sur la gamme Cisco 12000 n'implique pas que le même ensemble de fonctionnalités QoS disponible sur une autre plate-forme, telle que la gamme Cisco 7500, soit désormais disponible sur le Cisco 12000. Le MQC fournit une syntaxe commune dans laquelle une commande génère une fonction ou un comportement partagé. Par exemple, la commande **bandwidth** implémente une garantie de bande passante minimale. La gamme Cisco 12000 utilise MDRR comme mécanisme de planification pour réserver la bande passante, tandis que la gamme Cisco 7500 utilise WFQ. L'algorithme principal complète la plate-forme particulière.

Surtout, seules les files d'attente FrFab prennent en charge la configuration des fonctionnalités QoS via le MQC. Les files d'attente ToFab étant des files d'attente de sortie virtuelles et non de vraies files d'attente d'entrée, elles ne sont pas prises en charge par le MQC. Ils doivent être configurés avec des commandes CoS héritées.

[Le tableau 5](#) répertorie la prise en charge du MQC par type de moteur L3.

Tableau 5 - Prise en charge de MQC pour les types de moteurs de couche 3

Type de moteur de couche 3	Moteur 0	Moteur 1	Moteur 2	Moteur 3	Moteur 4	Moteur 4+

Support MQC	Oui	Non	Oui	Oui	Oui	Oui
Versio n IOS	12.0(15)S	-	12.0(15)S ¹	12.0(21)S	12.0(22)S	12.0(22)S

¹ Souvenez-vous de ces exceptions avec la prise en charge de MQC sur les cartes de ligne (LC) des moteurs 0 et 2 :

- 2xCHOC3/STM1 - Introduit dans 12.0(17)S.
- 1xOC48 DPT - Introduit dans la version 12.0(18)S.
- ATM 8xOC3 - Prévu pour 12.0(22)S.

Le MQC utilise ces trois étapes pour créer une stratégie QoS :

1. Définissez une ou plusieurs classes de trafic avec la commande **class-map**.
2. Créez une stratégie QoS avec la commande **policy-map** et affectez des actions QoS telles que **bande passante** ou **priorité** à une classe de trafic nommée.
3. Utilisez la commande **service-policy** pour attacher une carte-politique à la file d'attente FrFab d'une interface sortante.

Utilisez la commande **show policy-map interface** pour surveiller votre stratégie.

Pour plus d'informations, consultez [Vue d'ensemble de l'interface de ligne de commande Qualité de service modulaire](#).

[Étape 1 - Définissez les class-maps](#)

La commande **class-map** est utilisée pour définir des classes de trafic. En interne, sur la gamme Cisco 12000, la commande **class-map** attribue une classe à une file d'attente CoS spécifique sur la carte de ligne (voir l'[étape 4](#) pour plus de détails).

La commande **class-map** prend en charge « match-any », qui place les paquets qui correspondent à n'importe quelle instruction match dans la classe, et « match-all », qui place les paquets dans cette classe uniquement lorsque toutes les instructions sont vraies. Ces commandes créent une classe nommée "Δ_5" et classent tous les paquets ayant une priorité IP de 5 dans cette classe :

```
Router(config-cmap)#match ?
  access-group      Access group
  any                Any packets
  class-map         Class map
  destination-address Destination address
  fr-dlci           Match on fr-dlci
  input-interface   Select an input interface to match
  ip                IP specific values
  mpls              Multi Protocol Label Switching specific values
  not                Negate this match result
  protocol          Protocol
  qos-group         Qos-group
  source-address    Source address
Router(config-cmap)#match ip precedence 5
```

[Le tableau 6](#) répertorie les critères de correspondance pris en charge pour chaque type de moteur L3.

Tableau 6 - Critères de correspondance pris en charge pour les moteurs de couche 3

	Moteur 0, 2	Moteur 3	Moteur 4	Moteur 4+
priorité ip	Oui	Oui	Oui	Oui ¹
access-group	Non	Oui	Non	Non
mpls exp	Non	Oui	Non	Oui (12.0.26S)
ip dscp	Non	Oui	Non	Oui (12.0.26S)
qos-group	Non	Oui	Non	Non
match input-interface POS x/y	Non	Oui (comme stratégie de réception uniquement)	Non	Non

¹ entrée/sortie depuis 12.0.26S

Étape 2 - Créez une carte de stratégie

La commande **policy-map** est utilisée pour affecter des politiques ou des actions de gestion de paquets à une ou plusieurs classes définies. Par exemple, lorsque vous affectez une réservation de bande passante ou appliquez un profil de suppression aléatoire.

La gamme Cisco 12000 prend en charge un sous-ensemble de fonctionnalités MQC, basées sur l'architecture haut débit des moteurs L3. [Le tableau 7](#) répertorie les commandes prises en charge :

Tableau 7 - Commandes prises en charge

Comman de	Description
bande passante	Garantit une bande passante minimale en période de congestion. Il est spécifié en pourcentage de la vitesse de liaison ou en valeur absolue. Si une classe n'utilise pas ou n'a pas besoin de bande passante égale à la bande passante réservée, la bande passante disponible peut être utilisée par d'autres classes de bande passante.
police, forme	Limite la quantité de trafic qu'une classe peut transmettre. Ces commandes sont légèrement différentes en fonction. La commande police identifie le trafic qui dépasse la bande passante configurée et la supprime ou la remarque. La commande shape met en

	mémoire tampon tout trafic excédentaire et la planifie pour transmission à un rythme constant, mais ne supprime ni ne remarque.
Limite de la file d'attente	Attribue une longueur de file d'attente fixe à une classe de trafic donnée. Vous pouvez le spécifier en nombre de paquets pouvant être conservés dans la file d'attente.
priorité	Identifie une file d'attente comme file d'attente à faible latence. MQC prend en charge le mode strict uniquement pour un PQ. Le mode alternatif n'est pas pris en charge par MQC. Utilisez la commande priority sans valeur de pourcentage pour activer le mode de priorité stricte. Remarque : La mise en oeuvre de la commande priority sur la gamme Cisco 12000 diffère de celle des autres routeurs qui exécutent le logiciel Cisco IOS. Sur cette plate-forme, le trafic prioritaire n'est pas limité à la valeur de Kbits/s configurée pendant les périodes d'encombrement. Ainsi, vous devez également configurer la commande police pour limiter la quantité de bande passante qu'une classe prioritaire peut utiliser et garantir une bande passante adéquate pour les autres classes. Pour le moment, la commande police n'est prise en charge que sur les cartes de ligne du moteur 3. Sur les autres cartes de ligne du moteur, seule la classe par défaut est autorisée lorsque vous configurez une classe prioritaire.
random-detect	Attribue un profil WRED. Utilisez la commande de priorité de détection aléatoire pour configurer des valeurs WRED non par défaut par valeur de priorité IP.

Sur les LC du moteur 3, vous devez configurer les files d'attente FrFab avec l'interface de ligne de commande QoS modulaire (MQC) ; l'interface de ligne de commande (CLI) héritée n'est pas prise en charge.

Lorsque vous configurez la commande **bandwidth**, notez que les LC des moteurs 0 et 2 prennent en charge six classes de bande passante uniquement. Une septième classe peut être utilisée pour le service à faible latence et une huitième classe, qui est la classe par défaut, est utilisée pour tout le trafic non correspondant. Par conséquent, vous avez un total de huit files d'attente. La classe par défaut n'est pas utilisée comme classe prioritaire.

Sur les LC du moteur 3, la commande **bandwidth percent** est traduite en kbits/s, qui varie selon le débit de liaison sous-jacent, puis configurée directement sur la file d'attente. La précision de cette garantie de bande passante minimale est de 64 kbits/s.

Bien qu'aucune conversion en valeur quantique ne soit effectuée avec la commande **bandwidth**, toutes les files d'attente ont une valeur quantique. Sur les LC du moteur 3, la valeur quantique est

définie en interne en fonction de l'unité de transmission maximale (MTU) de l'interface et est définie de la même manière pour toutes les files d'attente. Il n'existe aucun mécanisme CLI MQC pour modifier cette valeur quantique, directement ou indirectement. La valeur quantique doit être supérieure ou égale au MTU de l'interface. En interne, la valeur quantique est exprimée en unités de 512 octets. Ainsi, avec un MTU de 4470 octets, la valeur quantique minimale de MTU doit être de 9.

MDRR sur LC du moteur 3

Cette section fournit des notes de configuration pour implémenter WRED et MDRR sur les LC du moteur 3.

- La bande passante MDRR configurée dans l'interface de ligne de commande est traduite en une quantité correspondant à L2 (par exemple, la surcharge L1 est supprimée). Cette quantité est ensuite arrondie aux 64 kbits/s suivants et programmée en matériel.
- Trois profils WRED différents sont pris en charge pour une classe.
- Le WRED (seuil maximum - seuil minimum) est approximativement à la puissance la plus proche de 2. Le seuil minimal est ensuite ajusté automatiquement, tandis que le seuil maximal reste inchangé.
- La valeur de probabilité de marque 1 est prise en charge.
- La configuration de la constante de pondération exponentielle n'est pas prise en charge.
- La priorité IP, les bits EXP MPLS et les valeurs DSCP sont pris en charge.

Remarque : chaque port ou canal sur les cartes de ligne Tetra (4GE-SFP-LC=) ou CHOC12/DS1-IR-SC= Frostbite a quatre files d'attente allouées par défaut. Les quatre files d'attente sont les suivantes :

- Une classe LLQ (Priority Queue)
- Une classe de file d'attente par défaut
- Deux classes normales non prioritaires

Lors de l'application d'une stratégie de service contenant plus de ces quatre classes (1 HPQ, 2 LPQ et classe-default) à l'interface, l'erreur suivante est signalée :

```
Router(config-if)#service-policy output mdr-policy
```

% Ressources de mise en file d'attente insuffisantes pour satisfaire la demande.

Depuis 12.0(26)S, une commande a été ajoutée pour la carte de ligne 4GE-SFP-LC= Tetra qui permet de configurer huit files d'attente/VLAN au lieu de quatre. Les huit files d'attente sont les suivantes :

- Un LLQ
- Une file d'attente par défaut de classe
- Six files d'attente normales

L'utilisation de cette commande nécessitera un rechargement de microcode de la carte de ligne et permettra de configurer uniquement 508 VLAN au lieu de 1022. La syntaxe de la commande est comme suit :

```
[no] hw-module slot <slot#> files d'attente d'interface qos 8
```

Exemple :

Router(config)#hw-module slot 2 files d'attente d'interface qos 8

Avertissement : Veuillez micro-recharger la carte de ligne pour que cette commande prenne effet

Router(config)#microcode reload 2

Cette commande sera disponible pour la carte de ligne CHOC12/DS1-IR-SC= Frostbite dans 12.0(32)S

Exemple n° 1 : Commande pourcentage de bande passante

Cet exemple attribue 20 % de la bande passante disponible au trafic de classe Δ_4 et 30 % au trafic de classe Δ_3 . Il laisse les 50 % restants à la classe par défaut.

En outre, il configure WRED comme mécanisme de suppression sur toutes les classes de données.

Exemple n° 1 : pourcentage de bande passante

```
policy-map GSR_EXAMPLE
  class Prec_4
    bandwidth percent 20
    random-detect
    random-detect precedence 4 1498 packets 9690 packets 1
  !--- All data classes should have WRED configured. class
  Prec_3 bandwidth percent 30 random-detect random-detect
  precedence 3 1498 packets 9690 packets 1 class class-
  default !--- Class-default uses any leftover bandwidth.
  random-detect random-detect precedence 2 1498 packets
  9690 packets 1 random-detect precedence 1 1498 packets
  9690 packets 1 random-detect precedence 0 1498 packets
  9690 packets 1
```

Exemple n° 2 - Commande bandwidth {kbps}

Cet exemple montre comment appliquer la commande bandwidth en tant que valeur absolue en kbits/s au lieu d'un pourcentage.

Exemple n° 2 - bande passante {kbps}

```
policy-map GSR_EXAMPLE
  class Prec_4
    bandwidth 40000
  !--- Configures a minimum bandwidth guarantee of 40000
  kbps or 40 Mbps in !--- times of congestion. Random-
  detect random-detect precedence 4 1498 packets 9690
  packets 1 class Prec_3 bandwidth 80000 !--- Configures a
  minimum bandwidth guarantee of 80000 kbps or 80 Mbps in
  !--- times of congestion. Random-detect random-detect
  precedence 3 1498 packets 9690 packets 1 class class-
  default !--- Any remaining bandwidth is given to class-
  default. Random-detect random-detect precedence 2 1498
  packets 9690 packets 1 random-detect precedence 1 1498
  packets 9690 packets 1 random-detect precedence 0 1498
  packets 9690 packets 1
```

Exemple 3 - Commande priority

Cet exemple est conçu pour les fournisseurs de services qui utilisent le routeur de la gamme Cisco 12000 comme routeur de périphérie (PE) du fournisseur MPLS et qui doivent configurer une stratégie de service QoS sur la liaison entre le routeur PE et le routeur de périphérie client (CE). Il place les paquets de priorité IP 5 dans une file d'attente prioritaire et limite la sortie de cette file d'attente à 64 Mbits/s. Il attribue ensuite une partie de la bande passante restante aux classes de bande passante.

Toutes les files d'attente de classe non prioritaires sont configurées avec la commande **aléatoire-detect** pour activer WRED en tant que stratégie d'abandon. WRED doit être configuré explicitement pour toutes les classes de bande passante et les classes par défaut.

Exemple 3 - priorité

```
policy-map foo
  class Prec_5
    police 64000000 conform-action transmit exceed-
action drop
!--- The police command is supported on Engine 3 line
cards. priority class Prec_4 bandwidth percent 30
random-detect random-detect precedence 4 1498 packets
9690 packets 1 class Prec_3 bandwidth percent 10 random-
detect random-detect precedence 3 1498 packets 9690
packets 1 class Prec_2 bandwidth percent 10 random-
detect random-detect precedence 2 1498 packets 9690
packets 1 class Prec_1 bandwidth percent 10 random-
detect random-detect precedence 1 1498 packets 9690
packets 1 class Prec_0 bandwidth percent 25 random-
detect random-detect precedence 0 1498 packets 9690
packets 1 class class-default random-detect random-
detect precedence 6 1498 packets 9690 packets 1 random-
detect precedence 7 1498 packets 9690 packets 1
```

Étape 3 - Attribuez une carte de stratégie à une file d'attente d'interface sortante

Comme mentionné ci-dessus, le MQC fonctionne uniquement avec les files d'attente FrFab sur une interface de sortie. Pour appliquer une carte-politique définie, utilisez la commande **service-policy output**, comme indiqué ici :

```
Router(config)#interface POS 0/0
Router(config-if)#service-policy ?
  history  Keep history of QoS metrics
  input    Assign policy-map to the input of an interface
  output   Assign policy-map to the output of an interface
Router(config-if)#service-policy output ?
  WORD    policy-map name
Router(config-if)#service-policy output GSR_EXAMPLE
```

Étape 4 - Surveillez et vérifiez la stratégie de service

Utilisez la commande **show policy-map interface** pour afficher l'application d'une stratégie. La commande **show policy-map interface** affiche les informations suivantes :

- Les classes de bande passante et de priorité configurées et les critères de correspondance.
- Tous les profils WRED.

- Paramètres de forme et de police.
- Comptabilité et taux du trafic.
- File d'attente CoS interne à laquelle une classe particulière est mappée. Ces files d'attente sont référencées par le même index que celui utilisé dans la sortie de la commande **show controller frfab queue**.

Voici un exemple de configuration complète et des commandes **show** pour surveiller la stratégie :

```

Configuration complète

class-map match-all class1
  match ip precedence 1
class-map match-all class2
  match ip precedence 2
!--- Step 1 - Configure traffic classes. ! policy-map
policy1e Class class1 bandwidth percent 10 random-detect
random-detect precedence 1 375 packets 2423 packets 1
Class class2 bandwidth percent 20 random-detect !---
Step 2 - Configure a policy-map. ! interface POS6/0 ip
address 12.1.1.1 255.255.255.0 no ip directed-broadcast
no keepalive service-policy output policy1e !--- Step 3-
Attach policy-map to the interface.

```

Utilisez la commande **show policy-map interface** pour afficher la stratégie configurée sur l'interface, ainsi que toutes les classes configurées. Voici le résultat de la commande :

```
Router#show policy-map int pos6/0
```

```
POS6/0
```

```
Service-policy output: policy1e (1071)
```

```
Class-map: class1 (match-all) (1072/3)
```

```
0 packets, 0 bytes
5 minute offered rate 0 bps, drop rate 0 bps
```

```
Match: ip precedence 1 (1073)
```

```
Class of service queue: 1
```

```
Tx Queue (DRR configured)
```

bandwidth percent	Weight
10	1

```
Tx Random-detect:
```

```
Exp-weight-constant: 1 (1/2)
```

Precedence	RED Label	Min	Max	Mark
1	1	375	2423	1

```
Class-map: class2 (match-all) (1076/2)
```

```
0 packets, 0 bytes
5 minute offered rate 0 bps, drop rate 0 bps
```

```
Match: ip precedence 2 (1077)
```

```
Class of service queue: 2
```

```
Tx Queue (DRR configured)
```

bandwidth percent	Weight
20	9

```
Tx Random-detect:
```

```
Exp-weight-constant: 1 (1/2)
```

Precedence	RED Label	Min	Max	Mark

```
Class-map: class-default (match-any) (1080/0)
```

```
0 packets, 0 bytes
```

```
5 minute offered rate 0 bps, drop rate 0 bps
```

```
Match: any (1081)
0 packets, 0 bytes
5 minute rate 0 bps
```

Commandes de surveillance de la gestion et de l'évitement de la congestion

Cette section répertorie les commandes que vous pouvez utiliser pour surveiller votre stratégie de gestion et d'évitement de congestion.

[Le tableau 8](#) répertorie les commandes appropriées pour les cartes de ligne d'entrée et de sortie.

Tableau 8 - Commandes des cartes de ligne

Carte de ligne d'entrée	Carte de ligne de sortie
<ul style="list-style-type: none">• show interfaces• exec slot <x> sh controller tofab queue• exec slot <x> show controller tofab queue <slot> <port>• exec slot <x> show controller tofab qm stat	<ul style="list-style-type: none">• show interfaces• show interfaces <y> aléatoire• exec slot <y> show controller frfab queue• exec slot <y> show controller frfab queue <port>• exec slot <y> show controller frfab QM stat

Ces commandes sont expliquées dans cette section.

Commande show interfaces

Avant d'utiliser cette commande, confirmez la bonne stratégie de mise en file d'attente. Si le résultat affiche First In, First Out (FIFO), assurez-vous que la commande **service-policy** apparaît dans la configuration en cours (si MQC a été utilisé pour configurer MDRR).

Surveillez le nombre de pertes de sortie, qui représente le nombre total de pertes WRED FrFab qui se sont produites pour le trafic sortant sur cette interface. Le nombre de pertes de sortie dans la sortie de commande **show interfaces** doit être égal ou supérieur au nombre de pertes de sortie dans la sortie de commande **show interfaces <nombre> aléatoire**.

Remarque : Sur le routeur de la gamme Cisco 12000, les pertes de sortie de l'interface sont mises à jour après la mise à jour des pertes WRED. Il y a peu de chances que si vous utilisez un outil pour interroger les deux compteurs de perte, les pertes d'interface ne soient pas encore mises à jour.

```
Router#show interfaces POS 4/0
POS4/0 is up, line protocol is up
  Hardware is Packet over SONET
  Description: link to c12f9-1
  Internet address is 10.10.105.53/30
  MTU 4470 bytes, BW 622000 Kbit, DLY 100 usec, rely 255/255, load 82/255
  Encapsulation PPP, crc 32, loopback not set
```

```

Keepalive set (10 sec)
Scramble enabled
LCP Open
Open: IPCP, CDPCP, OSICP, TAGCP
Last input 00:00:02, output 00:00:05, output hang never
Last clearing of "show interface" counters 00:04:54
Queueing strategy: random early detection (WRED)
Output queue 0/40, 38753019 drops; input queue 0/75, 0 drops
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 200656000 bits/sec, 16661 packets/sec
  135 packets input, 6136 bytes, 0 no buffer
  Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
    0 parity
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  7435402 packets output, 11182627523 bytes, 0 underruns
  0 output errors, 0 applique, 0 interface resets
  0 output buffer failures, 0 output buffers swapped out
  0 carrier transitions

```

[Commande show interfaces {number} random](#)

Lorsque vous utilisez cette commande, vous devez :

- Vérifiez que le modèle **cos-queue-group** correct est appliqué à cette interface.
- Vérifiez les poids MDRR. Pour chaque file d'attente MDRR, vous pouvez vérifier la moyenne pondérée pour la longueur de la file d'attente et la valeur la plus élevée atteinte (en paquets). Les valeurs sont calculées comme une moyenne pondérée et ne doivent pas nécessairement refléter la profondeur de file d'attente maximale réelle jamais atteinte.
- Vérifiez les seuils WRED minimum et maximum.
- Vérifiez le nombre de pertes aléatoires et de pertes de seuil pour chaque étiquette RED (les pertes « To Fabric » indiquent le nombre total de pertes pour cette étiquette sur toutes les cartes de ligne).
- Le compteur « TX-queue-limit drops » est utilisé uniquement sur les LC du moteur 1, qui ne prennent pas en charge WRED. Les cartes du moteur 1 vous permettent de définir la limite des files d'attente MDRR à l'aide de la commande **d'interface TX-queue-limit**. Lorsque WRED est pris en charge, les seuils WRED déterminent la profondeur des files d'attente MDRR.

```

Router#show interfaces POS 4/0 random
POS4/0
cos-queue-group: oc12
RED Drop Counts

```

TX Link

To Fabric

RED Label	Random	Threshold	Random	Threshold
0	29065142	73492	9614385	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0

```
TX-queue-limit drops: 0
```

```
Queue Lengths
```

TX Queue (DRR configured) oc12

Queue	Average	High Water Mark	Weight
0	0.000	2278.843	1
1	0.000	0.000	73
2	0.000	0.000	10
3	0.000	0.000	10
4	0.000	0.000	10
5	0.000	0.000	10
6	0.000	0.000	10
Low latency	0.000	0.000	10

TX RED config

Precedence 0: **375 min threshold, 2423 max threshold, 1/1 mark weight**
Precedence 1: not configured for drop
Precedence 2: not configured for drop
Precedence 3: not configured for drop
Precedence 4: 375 min threshold, 2423 max threshold, 1/1 mark weight
Precedence 5: not configured for drop
Precedence 6: 375 min threshold, 2423 max threshold, 1/1 mark weight
Precedence 7: not configured for drop weight 1/2

Commande show controller frfab queue {port} du logement exec (y)

Cette commande affiche la profondeur de file d'attente instantanée d'un port donné sur un emplacement donné. L'exemple de résultat de cette section affiche la file d'attente MDRR sur l'interface POS 4/1. Vous voyez une profondeur de file d'attente pour la file d'attente MDRR 1 des paquets 1964. Le poids correspond au nombre d'octets pouvant être servis dans cette file d'attente. Ce poids détermine le pourcentage de bande passante que vous voulez donner à cette file d'attente. Le déficit est la valeur qui indique à l'algorithme DRR combien de paquets doivent encore être servis. Vous pouvez voir qu'aucun paquet n'est mis en file d'attente LLQ (file d'attente DRR 7).

```
Router#execute-on slot 4 show controllers frfab queue 1
```

```
===== Line Card (Slot 4) =====
```

```
FrFab Queue
```

```
Interface 1
```

DRR#	Head	Tail	Length	Average	Weight	Deficit
0	95330	40924	0	0.000	4608	0
1	211447	233337	1964	1940.156	41472	35036
2	0	0	0	0.000	9216	0
3	0	0	0	0.000	9216	0
4	0	0	0	0.000	9216	0
5	0	0	0	0.000	9216	0
6	0	0	0	0.000	9216	0
7	0	0	0	0.000	9216	0

Cette commande est utilisée, en particulier, pour surveiller la profondeur de la file d'attente prioritaire de la carte de ligne de sortie. Lorsque vous voyez que les paquets commencent à attendre sur cette LLQ, cela indique bien que vous devez détourner un certain trafic Voix sur IP (VOIP) vers d'autres cartes de ligne de sortie. Dans un bon design, la longueur doit toujours être égale à 0 ou 1. Dans un réseau réel, le trafic est dense, même pour les données vocales. Le délai supplémentaire devient plus grave lorsque la charge vocale totale dépasse 100 % de la bande passante de sortie pendant une courte période. Le routeur ne peut pas placer plus de trafic sur le câble que ce qui est autorisé, et par conséquent le trafic vocal est mis en file d'attente sur sa propre file d'attente prioritaire. Cela crée la latence vocale et la gigue vocale introduites par la rafale du trafic vocal lui-même.

```

Router#execute-on slot 4 show controllers frfab queue 0
===== Line Card (Slot 4) =====
FrFab Queue
Interface 0
DRR#      Head      Tail      Length  Average          Weight  Deficit
0         181008    53494    2487    2282.937        4608   249
1         16887    45447    7        0.000          41472  0
2         0         0         0        0.000          9216   0
3         0         0         0        0.000          9216   0
4         0         0         0        0.000          9216   0
5         0         0         0        0.000          9216   0
6         0         0         0        0.000          9216   0
7         107818   142207   93       0.000          9216   -183600

```

La file d'attente 7 est la file d'attente LLQ et la longueur indique le nombre de paquets dans cette file d'attente LLQ.

[Commande show controller frfab QM stat du logement exec \(y\)](#)

Utilisez cette commande lorsque vous soupçonnez que la mémoire de paquet d'un LC commence à approcher la pleine capacité. Une valeur croissante pour le compteur « no mem drop » suggère que WRED n'est pas configuré ou que les seuils WRED sont trop élevés. Ce compteur ne doit pas être incrémenté dans des conditions normales. Reportez-vous à [Dépannage des paquets ignorés et des pertes de mémoire sur le routeur Internet de la gamme Cisco 12000](#) pour plus d'informations.

```

Router#execute-on slot 4 show controllers frfab QM stat
===== Line Card (Slot 4) =====
68142538 no mem drop, 0 soft drop, 0 bump count
0 rawq drops, 8314999254 global red drops, 515761905 global force drops
0 no memory (ns), 0 no memory hwm (Ns)
no free queue
0         0         1968    88
0         0         0         0
0         0         0         0
0         0         0         0
0 multicast drops
TX Counts
Interface 0
859672328848 TX bytes, 3908130535 TX pkts, 75431 kbps, 6269 pps
Interface 1
86967615809 TX bytes, 57881504 TX pkts, 104480 kbps, 8683 PPS
Interface 2
0 TX bytes, 0 TX pkts, 0 kbps, 0 PPS
Interface 3
0 TX bytes, 0 TX pkts, 0 kbps, 0 PPS

```

[Surveillance de la gestion des encombrements entrants](#)

Cette section décrit les commandes utilisées pour surveiller la gestion de la congestion entrante.

[Commande show interfaces](#)

Avant d'exécuter cette commande, vérifiez si la valeur du compteur ignoré est en augmentation.

Vous verrez des paquets ignorés si vous n'avez plus de mémoire sur le côté ToFab ou si la carte de ligne n'accepte pas les paquets assez rapidement. Pour plus d'informations, consultez [Dépannage des pertes d'entrée sur le routeur Internet de la gamme Cisco 12000](#).

```
Router#show interfaces POS 14/0
POS14/0 is up, line protocol is up
  Hardware is Packet over SONET
  Description: agilent 3b for QOS tests
  Internet address is 10.10.105.138/30
  MTU 4470 bytes, BW 2488000 Kbit, DLY 100 usec, rely 234/255, load 1/255
  Encapsulation HDLC, crc 32, loopback not set
  Keepalive not set
  Scramble disabled
  Last input never, output 00:00:03, output hang never
  Last clearing of "show interface" counters 00:34:09
  Queueing strategy: random early detection (WRED)
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 2231000 bits/sec, 4149 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    563509152 packets input, 38318622336 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
      0 parity
    166568973 input errors, 0 CRC, 0 frame, 0 overrun, 166568973 ignored, 0 abort
    35 packets output, 12460 bytes, 0 underruns
    0 output errors, 0 applique, 0 interface resets
    0 output buffer failures, 0 output buffers swapped out
    0 carrier transitions
```

[Commande show controller tofab queue du logement exec \(x\)](#)

Cet exemple de sortie de la commande **show controller tofab queue <x>** du logement **exec** a été capturé lorsqu'il n'y avait pas de congestion sur une carte de ligne de sortie dans le logement 3.

```
Router#execute-on slot 13 show controllers tofab queue
===== Line Card (Slot 13) =====
Carve information for ToFab buffers
!--- Output omitted. ToFab Queues: Dest Slot 0 0 0 0 9690 1 0 0 0 9690 2 0 0 0 9690 3 11419
16812 0 9690 4 0 0 0 2423 5 0 0 0 9690 6 0 0 0 9690 7 0 0 0 262143 8 0 0 0 262143 9 0 0 0 606 10
0 0 0 262143 11 0 0 0 262143 12 0 0 0 262143 13 0 0 0 262143 14 0 0 0 262143 15 0 0 0 9690
Multicast 0 0 0 262143
```

La sortie suivante a été capturée en cas d'encombrement sur le logement 3 :

```
Router#execute-on slot 13 show controllers tofab queue
===== Line Card (Slot 13) =====
Carve information for ToFab buffers
!--- Output omitted. ToFab Queues: Dest Slot 0 0 0 0 9690 1 0 0 0 9690 2 0 0 0 9690 3 123689
14003 1842 9690 4 0 0 0 2423 5 0 0 0 9690 6 0 0 0 9690 7 0 0 0 262143 8 0 0 0 262143 9 0 0 0 606
10 0 0 0 262143 11 0 0 0 262143 12 0 0 0 262143 13 0 0 0 262143 14 0 0 0 262143 15 0 0 0 9690
Multicast 0 0 0 262143
```

[Commande show controller tofab queue \(slot\) \(port\) du logement exec \(x\)](#)

Utilisez cette commande pour déterminer la quantité de mémoire utilisée du côté ToFab. En particulier, notez le numéro dans la colonne '#Qelem'. Notez que :

- Lorsqu'aucune mémoire n'est utilisée, les valeurs sont les plus élevées.

- La valeur de la colonne "#Qelem » diminue à mesure que les paquets sont mis en mémoire tampon.
- Lorsque la colonne "#Qelem » atteint zéro, toutes les mémoires tampon sculptées sont utilisées. Sur le LC du moteur 2, les petits paquets peuvent emprunter de l'espace tampon à des paquets plus volumineux.

Vous pouvez également utiliser cette commande pour déterminer le nombre de paquets mis en file d'attente sur une file d'attente de sortie virtuelle. L'exemple ci-dessous montre comment vérifier le logement 14 pour le nombre instantané de paquets sur ces files d'attente pour le logement 4, port 1 (POS 4/1). Nous voyons 830 paquets mis en file d'attente sur la file d'attente MDRR 1.

```
Router# execute-on slot 14 show controllers tofab queue 4 1
===== Line Card (Slot 14) =====
ToFab Queue
Slot 4 Int 1
DRR#      Head      Tail      Length  Average          Weight  Deficit
0          0          0          0        0.000           4608    0
1        203005    234676    830      781.093         41472   37248
2          0          0          0        0.000           9216    0
3          0          0          0        0.000           9216    0
4          0          0          0        0.000           9216    0
5          0          0          0        0.000           9216    0
6          0          0          0        0.000           9216    0
7          0          0          0        0.000           9216    0
```

[Commande show controller tofab QM stat du logement exec \(x\)](#)

Utilisez cette commande pour voir le nombre de pertes ToFab par carte de ligne. Vérifiez également si le compteur « aucune perte de mémoire » s'incrémente. Ce compteur s'incrémente lorsque CoS n'est pas configuré du côté ToFab.

```
Router#execute-on slot 13 show controllers tofab QM stat
===== Line Card (Slot 13) =====
0 no mem drop, 0 soft drop, 0 bump count
0 rawq drops, 1956216536 global red drops, 6804252 global force drops
0 no memory (Ns), 0 no memory hwm (Ns)
no free queue
0      0      0      0
0      0      0      0
0      0      0      0
0      0      0      0
Q status errors
0      0      0      0
0      0      0      0
0      0      0      0
0      0      0      0
```

[Étude de cas](#)

Cette étude de cas montre comment configurer une stratégie type pour le coeur de réseau d'un environnement de fournisseur de services. Il applique les commandes de file d'attente et vous permet d'utiliser MDRR/WRED pour la gestion active des files d'attente. Les stratégies QoS des routeurs de périphérie utilisent généralement le marquage du trafic, le conditionnement, etc., pour

permettre aux routeurs du coeur de trier le trafic en classes en fonction de la priorité IP ou des valeurs DSCP (DiffServ Code Point). Cette étude de cas utilise les fonctionnalités QoS de la plateforme logicielle Cisco IOS pour respecter les contrats de niveau de service (SLA) serrés et différents niveaux de service pour les services voix, vidéo et données sur le même fédérateur IP.

Dans cette approche, un fournisseur de services a mis en oeuvre trois classes de trafic. La plus importante est la classe LLQ ou Low Latency Queueing. Il s'agit de la classe de voix et de vidéo. Cette classe doit subir un délai et une gigue minimaux et ne doit jamais subir de perte de paquets ou de réordonnement de paquets tant que la bande passante de cette classe ne dépasse pas la bande passante de liaison. Cette classe est connue sous le nom de trafic de transfert accéléré par comportement de tronçon (EF PHB) dans l'architecture DiffServ. Le fournisseur d'accès à Internet (FAI) a conçu le réseau de manière à ce que cette classe ne dépasse pas 30 % en charge moyenne de la bande passante de liaison. Les deux autres classes sont la classe affaires et la classe du meilleur effort.

Dans la conception, nous avons configuré les routeurs de manière à ce que la classe affaires obtienne toujours 90 % de la bande passante restante et que la classe du meilleur effort obtienne 10 %. Ces deux classes ont un trafic moins sensible au temps et peuvent subir des pertes de trafic, des retards plus élevés et de la gigue. Dans la conception, l'accent est mis sur les cartes de ligne du moteur 2 : 1 carte de ligne OC48 rev B, 4 cartes de ligne OC12 rev B et 8 cartes de ligne OC3.

Les cartes de ligne Rev B sont les mieux adaptées pour transporter le trafic VoIP en raison d'une architecture ASIC et matérielle révisée, qui introduit très peu de latence. Avec l'ASIC révisé, la file d'attente FIFO de transmission est redimensionnée par le pilote de la carte de ligne à environ deux fois la plus grande MTU de la carte. Recherchez un "-B » ajouté à la référence, tel qu'OC48E/POS-SR-SC-B=.

Remarque : Ne confondez pas la file d'attente FIFO de transmission avec les files d'attente FrFab qui peuvent être réglées sur les cartes de ligne du moteur 0 avec la commande d'interface tx-queue-limit.

[Le tableau 9](#) répertorie les critères correspondants pour chaque classe.

Tableau 9 - Critères correspondants pour chaque classe

Nom de la classe	Critères correspondants
File d'attente prioritaire - Trafic vocal	Priorité 5
File d'attente d'entreprise	Priorité 4
File d'attente du meilleur effort	Priorité 0

Les cartes de ligne OC48 peuvent mettre en file d'attente un grand nombre de paquets dans les files d'attente ToFab. Il est donc important de configurer MDRR/WRED sur les files d'attente ToFab, en particulier lorsque l'interface de sortie est une interface à haut débit telle qu'OC48. Le fabric ne peut commuter le trafic vers la carte de ligne de réception qu'à un débit maximal théorique de 3 Gbit/s (paquets de 1 500 octets). Si la quantité totale de trafic envoyée est supérieure à ce que la matrice de commutation peut transporter à sa carte de réception, de nombreux paquets seront mis en file d'attente sur les files d'attente ToFab.

```
description OC48 egress interface
ip address 10.10.105.53 255.255.255.252
no ip directed-broadcast
ip router Isis encapsulation ppp
mpls traffic-eng tunnels
tag-switching ip
no peer neighbor-route
crc 32
clock source internal
POS framing sdh
POS scramble-atm
POS threshold sf-ber 4
POS flag s1s0 2
TX-cos oc48
Isis metric 2 level-1
Isis metric 2 level-2
ip rsvp bandwidth 2400000 2400000
!
interface POS4/1
description OC12 egress interface
ip address 10.10.105.121 255.255.255.252
no ip directed-broadcast
ip router Isis encapsulation ppp
mpls traffic-eng tunnels
no peer neighbor-route
crc 32
clock source internal
POS framing sdh
POS scramble-ATM POS threshold sf-ber 4
POS flag s1s0 2
TX-cos oc12
Isis metric 2 level-1
Isis metric 2 level-2
ip RSVP bandwidth 600000 60000
!
interface POS9/2
description OC3 egress interface
ip address 10.10.105.57 255.255.255.252
no ip directed-broadcast
ip router Isis crc 16
POS framing sdh
POS scramble-ATM POS flag s1s0 2
TX-cos oc3
Isis metric 200 level-1
Isis metric 2 level-2
!
interface POS13/0
description agilent 3a for QOS tests - ingress interface.
ip address 10.10.105.130 255.255.255.252
no ip directed-broadcast
no ip route-cache cef
no ip route-cache
no ip mroute-cache
no keepalive
crc 32
POS threshold sf-ber 4
TX-cos oc48
!
interface POS14/0
description agilent 3b for QOS tests - ingress interface.
ip address 10.10.105.138 255.255.255.252
no ip directed-broadcast
no keepalive
crc 32
```

```

POS threshold sf-ber 4
TX-cos oc48
!
interface POS15/0
description agilent 4A for QOS tests - ingress interface
ip address 10.10.105.134 255.255.255.252
no ip directed-broadcast
no ip mroute-cache
no keepalive
crc 32
POS threshold sf-ber 4
TX-CoS oc48
!
rx-cos-slot 3 StotTable
rx-cos-slot 4 StotTable
rx-cos-slot 9 StotTable
rx-cos-slot 13 StotTable
rx-cos-slot 14 StotTable
rx-cos-slot 15 StotTable
!
slot-table-cos StotTable
destination-slot 0 oc48
destination-slot 1 oc48
destination-slot 2 oc48
destination-slot 3 oc48
destination-slot 4 oc12
destination-slot 5 oc48
destination-slot 6 oc48
destination-slot 9 oc3
destination-slot 15 oc48
!
cos-queue-groupoc3
precedence 0 random-detect-label 0
precedence 4 queue 1
precedence 4 random-detect-label 1
precedence 5 queue low-latency
precedence 6 queue 1
precedence 6 random-detect-label 1
random-detect-label 0 94 606 1
random-detect-label 1 94 606 1
queue 0 1
queue 1 73
queue low-latency strict-priority
!--- Respect the tight SLA requirements. !--- No packets drop/low delay and jitter for the
priority queue. ! CoS-queue-groupoc12
precedence 0 random-detect-label 0
precedence 4 queue 1
precedence 4 random-detect-label 1
precedence 5 queue low-latency
precedence 6 queue 1
precedence 6 random-detect-label 1
random-detect-label 0 375 2423 1
random-detect-label 1 375 2423 1
queue 0 1
queue 1 73
queue low-latency strict-priority
!
CoS-queue-groupoc48
precedence 0 random-detect-label 0
precedence 4 queue 1
precedence 4 random-detect-label 1
precedence 5 queue low-latency
precedence 6 queue 1
precedence 6 random-detect-label 1

```

```
random-detect-label 0 1498 9690 1
random-detect-label 1 1498 9690 1
queue 0 1
queue 1 73
queue low-latency strict-priority
```

Plus le trafic VOIP est important, plus le trafic professionnel doit attendre avant d'être desservi. Cependant, ce n'est pas un problème car le SLA étroit ne nécessite aucune perte de paquets, et une latence et une instabilité très faibles pour la file d'attente prioritaire.

[Informations connexes](#)

- [Comment lire la sortie de la commande show controller frfab | Commandes tofab queue sur un routeur Internet de la gamme Cisco 12000](#)
- [Dépannage des paquets ignorés et des pannes pour manque de mémoire sur routeur Internet de la gamme Cisco 12000](#)
- [Dépannage des suppressions d'entrées sur les routeurs Internet de la gamme Cisco 12000](#)
- [Détection précoce aléatoire pondérée sur les routeurs de la gamme Cisco 12000](#)
- [Présentation de l'interface de ligne de commande de qualité de service modulaire](#)
- [Page de support des routeurs Internet de la gamme 12000](#)
- [Support technique - Cisco Systems](#)