

# Présentation et configuration de MDRR/WRED sur le routeur Internet de la gamme Cisco 12000

## Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Conventions](#)

[Informations générales](#)

[Aperçu MDRR](#)

[La file d'attente prioritaire de MDRR](#)

[Exemple MDRR](#)

[Support MDRR par le type de moteur](#)

[Aperçu WRED](#)

[Syntaxe existante de cos d'utilisation pour la configuration](#)

[Interface MQC \(Modular QoS CLI\) d'utilisation pour la configuration](#)

[Commandes de surveiller la Gestion et la manière d'éviter d'encombrement](#)

[L'exposition relie la commande](#)

[L'exposition relie {nombre} la commande aléatoire](#)

[La commande de file d'attente de frfab de show controller de l'emplacement d'exécutif \(y\) {port}](#)

[La commande stat du frfab QM de show controller de l'emplacement d'exécutif \(y\)](#)

[Surveillez la Gestion d'encombrement d'arrivée](#)

[L'exposition relie la commande](#)

[La commande de tofab queue de show controller de l'emplacement d'exécutif \(x\)](#)

[La commande de tofab queue de show controller de l'emplacement d'exécutif \(x\) \(emplacement\) \(port\)](#)

[La commande stat du tofab QM de show controller de l'emplacement d'exécutif \(x\)](#)

[Étude de cas](#)

[Informations connexes](#)

## [Introduction](#)

Examens de ce document comment configurer des caractéristiques de Gestion d'encombrement de logiciel de Cisco IOS® et de manière d'éviter d'encombrement sur le Routeur Internet de la série Cisco 12000.

Après que vous lisiez ce document, vous devez pouvoir :

- Comprenez pourquoi il est important de configurer le Mécanisme MDRR (Modified Deficit

- Round Robin) et le Détection précoce directe pondérée (WRED) dans votre principal réseau.
- Comprenez l'implémentation qui est à la base de MDRR et de WRED sur la gamme Cisco 12000.
  - Configurez MDRR et WRED utilisant la syntaxe existante et l'Interface MQC (Modular QoS CLI) de Classe de service (Cos).

## Conditions préalables

### Conditions requises

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

- Une connaissance générale de l'architecture de Routeur Internet de la série Cisco 12000.
- En particulier, une connaissance de l'architecture de queue et ces termes :ToFab (vers la matrice) – qui décrit le recevoir-side s'aligne sur un linecard d'arrivée.FrFab (de la matrice) – qui décrit le transmettre-side s'aligne sur un linecard sortant.

**Remarque:** On le recommande également que vous consultation [comment lire la sortie du frfab de show controller | commandes de tofab queue sur un Routeur Internet de la série Cisco 12000](#).

### Composants utilisés

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

- Toutes les Plateformes de Cisco 12000, qui incluent les 12008, 12012, 12016, 12404, 12406, 12410, et les 12416.
- Version du logiciel Cisco IOS 12.0(24)S1.

**Remarque:** Bien que les configurations dans ce document aient été testées sur la version du logiciel Cisco IOS 12.0(24)S1, n'importe quelle version logicielle de Cisco IOS qui prend en charge le Routeur Internet de la série Cisco 12000 peut être utilisée.

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 de documents, reportez-vous à [Conventions relatives aux conseils techniques Cisco](#).

## Informations générales

Les méthodes de mise en file d'attente définissent le mécanisme de planification de paquet ou la commande en lesquels des paquets sont retirés de la file d'attente à l'interface pour la transmission sur le fil physique. Basé sur la commande et le nombre de fois qu'une file d'attente est entretenu par une fonction de programmeur, les méthodes de mise en file d'attente prennent en charge également des garanties de bande passante minimale et des latences de bas.

Il est important de s'assurer qu'un mécanisme de planification de paquet prend en charge l'architecture de commutation sur laquelle il est mis en application. Le Mise en file d'attente pondérée (WFQ) est l'algorithme de planification réputé pour l'allocation de ressources sur des Plateformes de routeur de Cisco avec une architecture basée sur bus. Cependant, il n'est pas pris en charge sur le Routeur Internet de la série Cisco 12000. La queue traditionnelle et la coutume prioritaire de logiciel de Cisco IOS s'alignant également ne sont pas prises en charge. Au lieu de cela, la gamme Cisco 12000 utilise un formulaire spécial d'aligner le Mécanisme MDRR (Modified Deficit Round Robin) appelé, qui fournit des garanties relatives de bande passante aussi bien qu'une file d'attente à faible latence. Le M de MDRR signifie « modifié » ; il ajoute la file d'attente prioritaire comparée à DRR où aucune file d'attente prioritaire n'est présente. Pour des détails sur MDRR, voyez la vue d'ensemble [MDRR](#).

En outre, la gamme Cisco 12000 prend en charge le Détection précoce directe pondérée (WRED) comme stratégie de baisse dans les files d'attente MDRR. Ce mécanisme de manière d'éviter d'encombrement fournit une alternative au mécanisme par défaut de perte de destination. L'encombrement peut être évité par des baisses commandées.

Les mécanismes de manière d'éviter et de Gestion d'encombrement tels que WRED et MDRR sont particulièrement importants sur les files d'attente de FrFab des interfaces sortantes relativement à vitesse réduite, telles que les linecards canalisés (LCS). La matrice de commutation à haut débit peut livrer des paquets aux groupes de canaux beaucoup plus rapides que les groupes de canaux peuvent les transmettre. Pendant que s'alignant/mémoires tampons sont gérés au niveau de port physique, la contre-pression sur un canal peut affecter tous autres canaux sur ce port. Ainsi, il est très important de gérer cette contre-pression par WRED/MDRR, que les limites à la contre-pression affectent aux canaux en question. Pour des informations sur la façon gérer le surabonnement en partance d'interface, ne voir des [paquets ignorés de dépannage et l'aucune baisse de mémoire sur le Routeur Internet de la série Cisco 12000](#).

## [Aperçu MDRR](#)

Cette section fournit un aperçu de Mécanisme MDRR (Modified Deficit Round Robin).

Le MDRR étant configuré comme stratégie de queue, des files d'attente non vides sont servies l'un après l'autre, d'une permutation circulaire. Chaque fois que une file d'attente est servie, une quantité déterminée de données sont retirées de la file d'attente. L'algorithme entretient alors la prochaine file d'attente. Quand une file d'attente est servie, MDRR maintient le nombre d'octets de données qui a été retiré de la file d'attente au-dessus de la valeur configurée. Dans le prochain passage, quand la file d'attente est servie de nouveau, moins de données seront retirées de la file d'attente pour compenser les données excessives qui ont été servies précédemment. En conséquence, la quantité moyenne de données retirées de la file d'attente par file d'attente sera proche de la valeur configurée. En outre, MDRR met à jour une file d'attente prioritaire qui obtient servi sur une base préférentielle. MDRR est expliqué plus en détail dans cette section.

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

- Valeur de quantum – C'est le nombre d'octets moyen servis dans chacun rond.
- Compteur de déficit – Ceci est utilisé pour dépister combien d'octets une file d'attente a transmis dans chacun rond. Il est initialisé à la valeur de quantum.

Des paquets dans une file d'attente sont servis tant que le compteur de déficit est plus grand que zéro. Chaque paquet servi diminue le compteur de déficit par une valeur égale à sa longueur dans les octets. Une file d'attente peut plus n'être servie après que le compteur de déficit devienne zéro

ou négatif. Dans chacun le nouveau rond, le compteur de déficit de chaque file d'attente non vide est augmenté par sa valeur de quantum.

**Remarque:** Généralement la taille de tranche de temps pour une file d'attente ne doit pas être plus petite que le Maximum Transmission Unit (MTU) de l'interface. Ceci s'assure que le programmeur sert toujours au moins un paquet de chaque file d'attente non vide.

Chaque file d'attente MDRR peut être donnée un poids relatif, avec une des files d'attente dans le groupe défini comme file d'attente prioritaire. Les poids assignent la bande passante relative pour chaque file d'attente quand l'interface est congestionnée. L'algorithme MDRR retire des données de la file d'attente de chaque file d'attente d'une permutation circulaire s'il y a des données dans la file d'attente à envoyer.

Si toutes les files d'attente du militaire de carrière MDRR ont des données dans eux, elles sont entretenues comme suit :

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

Pendant chaque cycle, une file d'attente peut retirer une tranche de temps de la file d'attente basée sur son poids configuré. Sur des linecards de l'engine 0 et de l'Engine 2, une valeur de 1 est équivalente à donner à l'interface par poids de son MTU. Pour chaque incrément au-dessus de 1, le poids de la file d'attente augmente de 512 octets. Par exemple, si le MTU d'une interface spécifique est 4470 et le poids d'une file d'attente est configurés pour être 3, chaque fois par la rotation,  $4470 + (3-1)*512 = 5494$  octets pour être retiré de la file d'attente. Si deux files d'attente normales DRR, Q0 et Q1, sont utilisées, Q0 est configuré avec un poids de 1 et Q1 est configuré avec un poids de 9. Si les deux files d'attente sont congestionnées, chaque fois par la rotation, Q0 seraient laissées envoyer 4470 octets et Q1 serait permis pour envoyer  $[4470 + (9-1)*512] = 8566$  octets. Ceci donnerait le trafic qui va à Q0 approximativement 1/3 de la bande passante, et le trafic qui passe par Q1 environ 2/3 de la bande passante.

**Remarque:** La norme retirent la formule de la file d'attente utilisée pour calculer l'affectation de bande passante MDRR est  $D = MTU + (weight-1)*512$ . Dans les versions plus tôt que le Logiciel Cisco IOS version 12.0(21) S/ST, les linecards de l'engine 4 ont utilisé un différent retirent la formule de la file d'attente. Afin de configurer le poids MDRR pour une affectation correcte de bande passante, assurez-vous que vous exécutez une version du logiciel Cisco IOS plus tard que 12.0(21) S/ST.

**Remarque:** La formule de tranche de temps pour les linecards de l'engine 4+ est (pour la direction de toFab, c'est non valide pour FrFab)  $Quantum = BaseWeight + \{BaseWeight * (QueueWeight - 1) * 512\}/MTU$ . Le BaseWeight est obtenu avec cette formule :  $BaseWeight = \{(MTU + 512 - 1)/512\} + 5$

**Remarque:** Tous les calculs sont arrondis hors fonction ; c'est-à-dire, toutes les décimales sont ignorées.

**Remarque:** Pour savoir si un linecard spécifique d'engine prend en charge MDRR, voir le [support MDRR par le type de moteur](#).

## [La file d'attente prioritaire de MDRR](#)

La gamme Cisco 12000 prend en charge un intérieur MDRR de la file d'attente prioritaire (PQ). Cette file d'attente fournit le bas retard et gigue faible exigés par le trafic sensible à la durée tel

que la voix sur ip (VoIP).

Comme remarquable ci-dessus, la gamme Cisco 12000 ne prend en charge pas la mise en file d'attente pondérée (WFQ). Ainsi, le PQ à l'intérieur de MDRR fonctionne différemment de la caractéristique de queue de basse latence de logiciel de Cisco IOS (LLQ) disponible pour d'autres Plateformes.

Une différence principale est comment le programmeur MDRR peut être configuré pour entretenir le PQ dans un de deux modes, comme répertorié dans le [tableau 1](#) :

**Tableau 1 – Comment configurer le programmeur MDRR pour entretenir le PQ en deux modes**

	Mode alternatif	Mode de priorité stricte
Avantages	Ici, le PQ est entretenu entre les autres files d'attente. En d'autres termes, le programmeur MDRR entretient alternativement le PQ et toutes les autres files d'attente configurées.	Ici le PQ est entretenu toutes les fois qu'il est non vide. Ceci fournit le plus bas possible retard pour le trafic assorti.
Inconvénients	Ce mode peut introduire le jitter et retarder une fois comparé au mode de priorité stricte.	Ce mode peut mourir de faim d'autres files d'attente, en particulier si les écoulements assortis sont les expéditeurs agressifs.

Le mode alternatif peut exercer moins de contrôle sur le jitter et retarder. Si les débuts de programmeur MDRR pour entretenir les trames de taille d'une mtu à partir des données s'alignent et alors un paquet vocal arrive dans le PQ, le programmeur dans le mode alternatif sert complètement la file d'attente non-prioritaire jusqu'à ce que ses portées zéro de compteur de déficit. Pendant ce temps, le PQ n'est pas entretenu, et les paquets VoIP sont retardés.

En revanche, dans le mode de priorité stricte, le programmeur entretient seulement le paquet non-prioritaire en cours et puis commute au PQ. Le programmeur commence à entretenir une file d'attente non-prioritaire seulement après que le PQ devient complètement vide.

Il est important de noter que la file d'attente prioritaire en mode alternatif prioritaire est entretenue plus d'une fois dans un cycle, et prend ainsi plus de bande passante que d'autres files d'attente avec le même poids nominal. Combien plus coûte une fonction de combien de files d'attente sont définis. Par exemple, avec trois files d'attente, la file d'attente à faible latence est entretenue deux fois plus souvent que les autres files d'attente, et elle envoie deux fois son poids par cycle. Si huit files d'attente sont définies, la file d'attente à faible latence est entretenue sept fois plus souvent et le poids efficace est sept fois plus élevé. Ainsi, la bande passante que la file d'attente peut prendre est liée à combien de fois elle est servie par recherche séquentielle, qui dépend consécutivement de combien de files d'attente sont définies globalement. En mode alternatif prioritaire, la file d'attente prioritaire est habituellement configurée avec un petit poids pour cette raison particulière.

Comme exemple, supposez que quatre files d'attente sont définies : 0, 1, 2 et la file d'attente prioritaire. En mode alternatif prioritaire, si toutes les files d'attente sont congestionnées, elles

seraient entretenues comme suit : **0, llq, 1, llq, 2, llq, 0, llq, 1,....** là où le llq signifie la file d'attente à faible latence.

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

- Plan horizontal = poids de la file d'attente à faible latence.
- $W_0, W_1, \dots, W_n$  = poids du militaire de carrière DRR s'aligne.
- $n$  = nombre de files d'attente régulières DRR utilisées pour cette interface.
- $BW$  = bande passante du lien.

Pour le mode alternatif prioritaire, la bande passante minimale de la file d'attente à faible latence = de  $BW * n * \text{plan horizontal} / (n * \text{plan horizontal} + \text{Sum}(W_0, W_n))$ .

Le poids est le seul paramètre variable dans MDRR qui peut être configuré. Il influence la quantité de bande passante relative qu'une classe du trafic peut utiliser, et combien de trafic est introduit un tour. L'utilisation de plus grands poids signifie que le cycle global prend plus long, et augmente probablement la latence.

### Directives de configuration

- Il vaut mieux de configurer le poids de la classe qui a la plus basse bande passante nécessaire à 1 afin de maintenir le retard et le jitter aussi bas comme possible parmi les autres classes.
- Sélectionnez les valeurs de poids qui sont aussi basses comme possible. Début avec un poids de 1 par la classe avec la plus basse bande passante. Par exemple, quand vous utilisez deux classes avec la bande passante de 50% pour chaque classe, vous devez configurer 1 et 1. Il ne semble pas raisonnable d'utiliser 10 et 10, parce qu'il n'y a aucune incidence sur la représentation quand vous choisissez 1. En outre, un poids plus élevé introduit plus de jitter.
- Une valeur basse de poids pour le LLQ est très importante, particulièrement dans le mode alternatif pour ne pas ajouter trop de retard ou se trémousser aux autres classes.

### Exemple MDRR

L'exemple dans cette section est pris de l'intérieur de l'*architecture logicielle de Cisco IOS®*, Cisco appuient sur.

Supposez que nous avons trois files d'attente :

- **La file d'attente 0** – a une tranche de temps de 1500 octets ; c'est la file d'attente à faible latence, configurée pour fonctionner dans le mode alternatif.
- **La file d'attente 1** – a une tranche de temps de 3000 octets.
- **La file d'attente 2** – a une tranche de temps de 1500 octets.

[La figure 1](#) montre l'état initial des files d'attente, avec quelques paquets qui ont été reçus et alignés.

### Figure 1 – État initial 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 entretenue d'abord, sa tranche de temps est ajoutée à son compteur de déficit, le paquet 1, qui est de 250 octets, est transmise, et sa taille est soustraite du compteur de déficit. Puisque le compteur de déficit de la file d'attente 0 est encore plus grand que 0 ( $1500 - 250 = 1250$ ), le paquet 2 est aussi bien transmis, et sa longueur soustraite du compteur de déficit. Le compteur de déficit de la file d'attente 0 est maintenant  $-250$ , ainsi la file d'attente 1 est entretenue ensuite. [La figure 2](#) indique cet état.

Figure 2 – État ultérieur MDRR

Queues					Deficit Counters			
Queue 0					3/250	-250		
Queue 1					6/1500	5/1500	4/1500	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 placé à 3000 ( $0 + 3000 = 3000$ ), et les paquets 4 et 5 sont transmis. Chaque paquet étant transmis, soustrayez la taille du paquet du compteur de déficit, ainsi le compteur de déficit de la file d'attente 1 est réduit à 0. [figures 3](#) illustre cet état.

Figure 3 – État MDRR quand 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 retourner du mode alternatif prioritaire pour entretenir la file d'attente 0. De nouveau, la tranche de temps est ajoutée au compteur de déficit en cours, et le compteur de déficit de la file d'attente 0 est placé au résultat ( $-250 + 1500 = 1250$ ). Le paquet 3 est maintenant transmis, parce que le compteur de déficit est plus grand que 0, et la file d'attente 0 est maintenant vide. Quand une file d'attente est vidée, son compteur de déficit est placé à 0, suivant les indications de la [figure 4](#).

Figure 4 – État MDRR quand 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 entretenue ensuite ; son compteur de déficit est placé à 1500 ( $0 + 1500 = 1500$ ). Les paquets 7 à 10 sont transmis, qui laisse le compteur de déficit à 500 ( $1500 - (4 \times 250) = 500$ ). Puisque le compteur de déficit est encore plus grand que 0, le paquet 11 est également transmis.

Quand le paquet 11 est transmis, la file d'attente 2 est vide, et son compteur de déficit est placé à 0, suivant les indications de la [figure 5](#).

Figure 5 – État MDRR quand le paquet 11 est transmis

Queues		Deficit Counters
Queue 0		0
Queue 1	6/1500	0
Queue 2		0

La file d'attente 0 est entretenue de nouveau ensuite (parce que nous sommes en mode alternatif prioritaire). Puisqu'il est vide, nous entretenons la file d'attente 1 ensuite, et transmettons le paquet 6.

## [Support MDRR par le type de moteur](#)

La gamme Cisco 12000 prend en charge cinq modèles de linecard avec la seule couche 3 (L3) expédiant des architectures d'engine. Le soutien de MDRR varie avec le type de moteur L3, et le type de carte. Par exemple, il n'y a aucun soutien de MDRR et de WRED sur des cartes de ligne ATM de l'engine 0. Vous pouvez utiliser la commande de **show diag** de déterminer le type de moteur L3 de vos cartes de ligne installée :

```
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)
```

## [MDRR sur des files d'attente de ToFab \(Rx\)](#)

Vous pouvez employer « la syntaxe existante de cos » ou « l'interface de ligne de commande modulaire de QoS » pour configurer MDRR sur la gamme Cisco 12000. Les sections postérieures



dans ce document discutent comment configurer MDRR avec le cos existant ou le QoS modulaire. Vous devez configurer les tofabs queue avec la syntaxe existante de cos seulement car ils ne prennent en charge pas la syntaxe plus nouvelle du MQC. Voir le [tableau 2](#) pour des détails.

**Tableau 2 – Détails sur MDRR sur des files d'attente de ToFab (Rx)**

	Là où mis en application	ToFab MDRR	Remplaçant PQ de ToFab	ToFab PQ strict	ToFab WRED
Eng0	Logiciel	Non **	Non **	Oui	Oui
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 n'est pris en charge sur le LCS de l'engine 0 dans la direction de ToFab (Rx), mais sur seulement le mode de priorité stricte, pas le mode alternatif prioritaire. Les sept files d'attente restantes sont prises en charge comme d'habitude.

Les interfaces d'arrivée mettent à jour une file d'attente de sortie virtuelle distincte par destination LC. Comment ces files d'attente sont mises en application dépend du type de moteur L3.

- Engine 0 – Les LCS d'arrivée mettent à jour 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.
- Engine 2, 3, 4, et 4+ – les LCS d'arrivée mettent à jour 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  $16 \times 16 \times 8 = 2048$ . Toutes les interfaces sur un emplacement de destination doivent utiliser les mêmes paramètres.

### [MDRR sur des files d'attente de FrFab \(Tx\)](#)

MDRR sur les files d'attente de FrFab fonctionne uniformément si mis en application en matériel ou logiciel. Tout le support huit de types de moteur L3 classent des files d'attente pour chaque interface sortante. Voir le [tableau 3](#) pour des détails.

**Tableau 3 – Détails sur MDRR sur des files d'attente de FrFab (Tx)**

	Là où mis en application	Remplaçant PQ de FrFab	FrFab PQ strict	FrFab WRED
Eng0	Logiciel <sup>1</sup>	Non	Oui	Oui
Eng1	-	Non	Non	Non
Eng2	Matériel	Oui <sup>2</sup>	Oui	Oui
Eng3	Matériel	Non	Oui	Oui
Eng4	Matériel	Oui	Oui	Oui

Eng4+	Matériel	Oui	Oui	Oui
-------	----------	-----	-----	-----

1Support pour MDRR sur des files d'attente de FrFab de LCS de l'engine 0 est introduit dans ces versions de logiciel de Cisco IOS :

- Logiciel Cisco IOS version 12.0(10)S - POS 4xOC3 et 1xOC12, 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.

2You doit configurer le MDRR alternatif dans la direction de FrFab avec la syntaxe existante de cos.

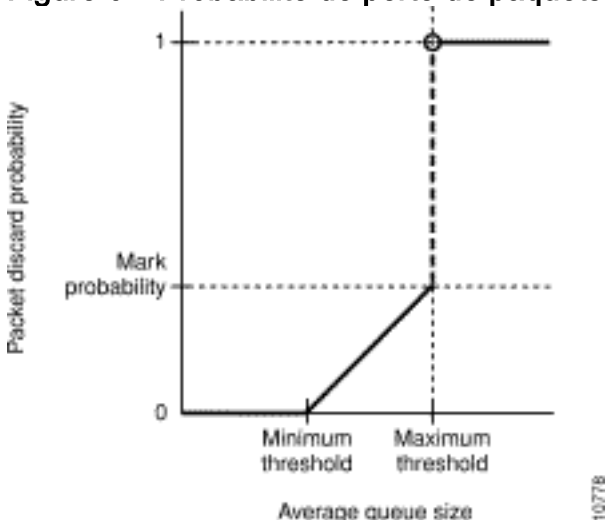
**Remarque:** Les supports MDRR 3xGE LC sur les tofabs queue et, comme du Logiciel Cisco IOS version 12.0(15)S, sur le FrFab s'aligne avec deux restrictions, à savoir, une tranche de temps fixe, et cos simple s'alignent pour chaque interface. La file d'attente prioritaire prend en charge une tranche de temps qui peut être configurée, et des modes stricts et alternatifs prioritaires. Chacune des trois interfaces partage un PQ simple.

**Remarque:** Voyez les notes en version de Routeurs de la gamme Cisco 12000 pour les dernières informations concernant les caractéristiques prises en charge de QoS sur le LCS de gamme Cisco 12000.

## Aperçu WRED

Le Détection précoce directe pondérée (WRED) est conçu pour empêcher les effets néfastes de la congestion d'interface sur le débit de réseau.

Figure 6 – Probabilité de perte de paquets WRED



Voir la [détection tôt aléatoire pesée sur le routeur de gamme Cisco 12000](#) pour une explication des paramètres WRED. Le minimum, le maximum, et les paramètres de probabilité de marque décrivent la courbe réelle de Détection précoce aléatoire (RED). Quand la moyenne pesée de file d'attente est au-dessous du seuil minimum, aucun paquet n'est lâché. Quand la moyenne pesée de file d'attente est au-dessus du seuil de file d'attente maximum, tous les paquets sont lâchés jusqu'aux basses moyennes au-dessous du seuil maximal. Quand la moyenne est entre le minimum et les seuils maximaux, la probabilité que le paquet sera lâché peut être calculée par une ligne droite du seuil minimum (la probabilité de la baisse sera 0) au seuil maximal (la probabilité de la baisse est égale au dénominateur de la probabilité 1/mark).

La différence entre le ROUGE et le WRED est que WRED peut sélectivement jeter le trafic de bas-priorité quand l'interface commence à obtenir congestionné, et peut fournir des caractéristiques du fonctionnement différenciées pour différentes classes de service (cos). Par défaut, WRED utilise un profil ROUGE différent pour chaque poids (Priorité IP - 8 profils). Il relâche les paquets moins importants plus agressivement que des paquets plus importants.

C'est un défi complexe pour accorder des paramètres WRED pour gérer la profondeur de la file d'attente, et dépend de beaucoup de facteurs, qui incluent :

- Charge et profil de la circulation offerts.
- Rapport de chargement à la capacité disponible.
- Comportement du trafic en présence de l'encombrement.

Ces facteurs varient le réseau par le réseau et, à leur tour, dépendent des services offerts et des clients qui utilisent ces services. Ainsi, nous ne pouvons pas émettre les recommandations qui s'appliquent aux environnements spécifiques de client. Cependant, le [tableau 4](#) décrit généralement des valeurs recommandées basées sur la bande passante du lien. Dans ce cas, nous ne différencions pas les caractéristiques chutantes entre les différentes classes de service.

**Tableau 4 – Valeurs recommandées basées sur la bande passante du lien**

Bande passante	BW théorique (Kbps)	BW physique <sup>1</sup> (Kbps)	Seuil minimum	Seuil maximal
OC3	155000	149760	94	606
OC12	622000	599040	375	2423
OC48	2400000	239616	1498	9690
OC192	10000000	9584640	5991	38759

débit <sup>1</sup>Physical SONET

Il y a plusieurs contraintes qui sont prises en considération pour calculer les valeurs seuil ci-dessus. Par exemple, la maximisation de l'utilisation de lien tout en réduisant la profondeur moyenne de file d'attente, la différence entre le maximum et le minimum doivent être un unité de deux (dû à la limitation matérielle). Basé sur l'expérience et la simulation, la profondeur instantanée maximum d'une file d'attente commandée par le ROUGE est moins de 2 MaxTh. Pour OC48 et en haut, 1 MaxTh, et ainsi de suite. Cependant, la détermination précise de ces valeurs est hors de portée de ce document.

**Remarque:** La valeur constante de coefficient exponentiel n'a pas besoin d'être configurée sur l'Engine 2 et au-dessus des linecards, puisque l'échantillonnage de file d'attente de matériel est utilisé à la place. Pour le LCS de l'engine 0, ces valeurs peuvent être configurées :

- DS3 – 9
- oc3 – 10
- oc12 – 12

**Remarque:** WRED n'est pas pris en charge sur le LCS de l'engine 1.

Pendant que les sections suivantes expliquent, vous pouvez employer la syntaxe existante de cos et la syntaxe plus nouvelle MQC pour configurer WRED.

## [Syntaxe existante de cos d'utilisation pour la configuration](#)

La syntaxe existante de Classe de service (Cos) de gamme Cisco 12000 utilise un modèle de cos-file d'attente-**groupe** pour définir un ensemble de définitions de cos. Vous appliquez alors le modèle aux files d'attente de ToFab et de FrFab sur les interfaces d'arrivée ou sortantes, respectivement.

## Étape 1 : Définissez un cos-file d'attente-groupe

L'ordre de cos-file d'attente-**groupe** crée un modèle Désigné des paramètres MDRR et WRED. Voici les paramètres de configuration disponibles au CLI :

```
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 tracer la Priorité IP aux files d'attente MDRR et configurer la file d'attente à faible latence spéciale. Vous pouvez utiliser le paramètre de priorité sous l'ordre de cos-file d'attente-**groupe** pour ceci :

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

Vous pouvez tracer une Priorité IP particulière à une file d'attente du militaire de carrière MDRR (la file d'attente 0 6) ou vous peut la tracer à la file d'attente prioritaire. La commande ci-dessus te permet pour tracer plusieurs priorités IP à la même file d'attente.

**Remarque:** Il est recommandé que vous utilisez la priorité 5 pour la file d'attente à faible latence. La priorité 6 est utilisée pour conduire des mises à jour.

Vous pouvez donner à chacun la file d'attente MDRR par poids relatif, avec une des files d'attente dans le groupe défini comme file d'attente prioritaire. Vous pouvez utiliser la commande de **file d'attente** sous le cos-file d'attente-**groupe** de faire 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 l'ordre de cos-file d'attente-**groupe** de définir tous les paramètres WRED :

```
random-detect-label <label> <minimum-threshold> <maximum-threshold> <mark-probability
denominator>
```

Voici un exemple d'un cos-file d'attente-**groupe** nommé **oc12**. Il définit trois classes du trafic : file d'attente 0, 1, et faible latence. Il trace les valeurs de Priorité IP 0 - 3 pour aligner 0, priorité évalué 4, 6, et 7 pour aligner 1, et priorité 5 à la file d'attente à faible latence. La file d'attente 1 est assignée 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 - Créez l'emplacement-table-cos pour des tofabs queue

Pour éviter la tête de la ligne bloquant, les interfaces d'arrivée sur la gamme Cisco 12000 mettent à jour une file d'attente de sortie virtuelle par emplacement de destination. Allez à un linecard utilisant la commande d'attache et exécutez la commande de tofab queue de show controller d'exécute-on (ou sélectionnez directement la commande de tofab queue de shows controllers de l'emplacement 0 d'exécute-on) de visualiser ces files d'attente de sortie virtuelle. La sortie témoin capturée directement de la console LC est fournie ci-dessous. Voyez [comment lire la sortie du frfab de show controller | commandes de tofab queue sur un Routeur Internet de la série 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 l'emplacement-table-cos commandent de tracer un cos-file d'attente-groupe Désigné à une file d'attente de sortie virtuelle de destination. Vous pouvez configurer un seul modèle de cos-file d'attente-groupe 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, oc48 Désigné, oc12, et oc3. La configuration pour le cos-file d'attente-groupe nommé oc12 est suivant les indications de Step1. De même, configurez oc3 et oc48. Il est recommandé que vous appliquez un seul modèle à un ensemble d'interfaces basées sur la bande passante et l'application.

## Étape 3 - Appliquez l'emplacement-table-cos à une interface d'arrivée

Utilisez la commande de **rx-cos-emplacement** d'appliquer l'emplacement-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 cos-file d'attente-groupe à une interface sortante

La gamme Cisco 12000 met à jour une file d'attente séparée par interface en partance. Pour visualiser ces files d'attente, attache au linecard CLI. Utilisez la commande d'**attache**, et puis exécutez la commande de **file d'attente de frfab de show controller**, 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 le **tx-cos** commandent d'appliquer un modèle de cos-file d'attente-groupe à une file d'attente d'interface. Comme affiché ici, vous appliquez l'ensemble de paramètres directement à l'interface ; aucune table n'est nécessaire. Dans cet exemple, *pos48* est le nom d'un ensemble 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 le **cos d'exposition** commandent de 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:** Le legs CLI utilise également la syntaxe de priorité pour le trafic de Commutation multiprotocole par étiquette (MPLS). Le routeur traite les bits MPLS comme s'ils sont des bits de type de service IP (tos) et met les paquets appropriés dans les files d'attente correctes. Cela vaut pas du tout pour MQC. Le MPLS QoS est hors de portée de ce document.

## Interface MQC (Modular QoS CLI) d'utilisation pour la configuration

L'objectif de l'Interface MQC (Modular QoS CLI) de Cisco est de connecter toutes les différentes caractéristiques de QoS d'une certaine manière une façon logique, afin de simplifier la configuration de la qualité logicielle de Cisco IOS des caractéristiques de service (QoS). Par exemple, la classification est faite séparément de la queue, du maintien de l'ordre, et de la formation. Il fournit un cadre simple de configuration pour QoS qui est basé sur modèle. Voici

quelques points à se souvenir au sujet de la configuration MQC :

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

Sur la gamme Cisco 12000, des commandes MQC peuvent être utilisées au lieu de la syntaxe existante de Classe de service (Cos).

Le support MQC sur la gamme Cisco 12000 n'implique pas que le même disponible réglé de caractéristique de QoS sur une autre plate-forme, telle que la gamme Cisco 7500, est maintenant disponible sur le Cisco 12000. Le MQC fournit une syntaxe commune en laquelle une commande a comme conséquence une fonction ou un comportement partagée. Par exemple, la **commande bandwidth** implémente une garantie de bande passante minimale. La gamme Cisco 12000 emploie MDRR comme mécanisme de planification pour faire la réservation de bande passante, alors que la gamme Cisco 7500 utilise WFQ. L'algorithme de principal complète la plate-forme particulière.

D'une manière primordiale, seulement les files d'attente de FrFab prennent en charge la configuration des caractéristiques de QoS par le MQC. Puisque les tofabs queue sont des files d'attente de sortie virtuelle, et des files d'attente d'entrée non vraies, ils ne sont pas pris en charge par le MQC. Ils doivent être configurés avec des commandes existantes de cos.

Soutien de listes du [tableau 5 du](#) MQC par type de moteur L3.

**Tableau 5 – Soutien de MQC pour des types de moteur L3**

Type de moteur L3	Engine 0	Engine 1	Engine 2	Engine 3	Engine 4	Engine 4+
Support MQC	Oui	Non	Oui	Oui	Oui	Oui
L'IOS libèrent	12.0(15)S	-	12.0(15)S <sup>1</sup>	12.0(21)S	12.0(22)S	12.0(22)S

<sup>1</sup>Remember ces exceptions avec le support MQC sur les linecards de l'engine 0 et 2 (LC) s :

- 2xCHOC3/STM1 - Introduit dans 12.0(17)S.
- 1xOC48 DPT - Introduit dans 12.0(18)S.
- atmosphères 8xOC3 - Prévus pour 12.0(22)S.

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

1. Définissez un ou plusieurs classes du trafic avec la commande de **class-map**.
2. Créez une stratégie QoS avec la commande de **policy-map** et assignez les actions de QoS telles que la **bande passante** ou la **priorité à une** classe Désignée du trafic.

3. Utilisez la commande de service-**stratégie** de relier un policy-map à la file d'attente de FrFab d'une interface sortante.

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

Voir l'[aperçu de qualité de service modulaire d'interface de ligne de commande](#) pour de plus amples informations.

### Étape 1 - Définissez les class-map

La commande de **class-map** est utilisée de définir des classes du trafic. Intérieurement, sur la gamme Cisco 12000, la commande de **class-map** assigne une classe à une file d'attente spécifique de cos sur le linecard (voir l'[étape 4](#) pour des détails).

La commande de **class-map** prend en charge le « match any », qui place les paquets qui appartiennent des déclarations l'unes des de correspondance dans la classe, et « correspondance-tout », qui place des paquets dans cette classe seulement quand toutes les déclarations sont vraies. Ces commandes créent une classe nommée "Prec\_5", et classifient tous les paquets avec une Priorité IP de 5 à 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 le critère de correspondance pris en charge pour chaque type de moteur L3.

**Tableau 6 – Critère de correspondance pris en charge pour des moteurs L3**

	Engine 0, 2	Engine 3	Engine 4	Engine 4+
Priorité IP	Oui	Oui	Oui	Oui <sup>1</sup>
access-group	Non	Oui	Non	Non
exp de MPLS	Non	Oui	Non	Oui (12.0.26S)
ip dscp	Non	Oui	Non	Oui (12.0.26S)
qos-groupe	Non	Oui	Non	Non
POS x/y de match input-interface	Non	Oui (comme reçoivent stratégie seulement)	Non	Non

<sup>1</sup> d'entrée/de sortie depuis 12.0.26S



## [Étape 2 - Créez un policy-map](#)

La commande de **policy-map** est utilisée d'assigner des stratégies ou des actions de traitement des paquets à un ou plusieurs classes définies. Par exemple, quand vous assignez une réservation de bande passante, ou appliquez un profil aléatoire de baisse.

La gamme Cisco 12000 prend en charge un sous-ensemble de caractéristiques MQC, basé sur l'architecture à grande vitesse des moteurs L3. [Le tableau 7](#) répertorie les commandes qui sont prises en charge :

**Tableau 7 – Commandes prises en charge**

Comman de	Description
bande passante	Fournit une garantie de bande passante minimale au cours des périodes d'encombrement. Il est spécifié comme pourcentage de la vitesse de liaison ou comme valeur absolue. Si une classe n'utilise pas ou a besoin de la bande passante égale au Kbps réservé, la bande passante disponible peut être utilisée par d'autres classes de bande passante.
police, forme	Limite le niveau de trafic qu'une classe peut transmettre. Ces commandes sont légèrement différentes dans la fonction. <b>La police</b> commande identifie le trafic qui dépasse la bande passante configurée, et des baisses ou les remarque. Les mémoires tampons de commande de <b>forme</b> n'importe quel trafic excédentaire et le programme pour la transmission à un débit constant, mais ne chute pas ou remarque.
Queue- limit	Assigne une longueur fixe de file d'attente à une classe du trafic donnée. Vous pouvez spécifier ceci en terme des paquets qui peuvent être tenus 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 seulement pour un PQ. Le mode alternatif n'est pas pris en charge par MQC. Utilisez la commande <b>prioritaire</b> sans valeur de pourcentage d'activer le mode de priorité stricte. <b>Remarque:</b> L'implémentation de la commande <b>prioritaire</b> sur la gamme Cisco 12000 diffère de l'implémentation sur d'autres Routeurs qui exécutent le logiciel de Cisco IOS. Sur cette plate-forme, le trafic prioritaire n'est pas limité à la valeur configurée de Kbps au cours des périodes d'encombrement. Ainsi, vous devez également configurer la <b>police</b> commandes de

	limiter combien de bande passante une classe prioritaire peut utiliser-et assurer à bande passante adéquate pour d'autres classes. À ce moment, la <b>police</b> commande est seulement prise en charge sur des linecards de l'engine 3. Sur les autres linecards d'engine, on permet seulement le classe-par défaut quand vous configurez une classe prioritaire.
random-detect	Assigne un profil WRED. Utilisez la commande de <b>random-detect precedence</b> de configurer des valeurs du non-par défaut WRED par valeur de priorité IP.

Sur le LCS de l'engine 3, vous devez configurer les files d'attente de FrFab avec l'Interface MQC (Modular QoS CLI) ; l'interface de ligne de commande existante (CLI) n'est pas prise en charge.

Quand vous configurez la **commande bandwidth**, notez que des classes de bande passante du support six LCS de l'engine 0 et 2 seulement. Une septième classe peut être utilisée pour le service de faible latence et une huitième classe, qui est classe-par défaut, est utilisée pour tout le trafic non assorti. Par conséquent, vous avez un total de huit files d'attente. le Classe-par défaut n'est pas utilisé comme classe prioritaire.

Sur le LCS de l'engine 3, la commande de **pourcentage de bande passante** est traduite dans une valeur de Kbps, qui varie avec du débit de base de la liaison, et puis configurée directement sur la file d'attente. La précision de cette garantie de bande passante minimale est des 64 Kbits/s.

Bien qu'aucune conversion en valeur de quantum ne soit faite avec la **commande bandwidth**, toutes les files d'attente ont une tranche de temps. Sur le LCS de l'engine 3, la valeur de quantum est placée intérieurement basée sur le Maximum Transmission Unit (MTU) de l'interface, et est placée également pour toutes les files d'attente. Il n'y a aucun mécanisme MQC CLI pour modifier cette valeur de quantum, directement ou indirectement. La valeur de quantum doit être supérieur ou égal à l'interface MTU. Intérieurement, la valeur de quantum est dans les unités de 512 octets. Ainsi, avec un MTU de 4470 octets, la valeur de quantum minimum du MTU doit être 9.

## [MDRR sur l'engine 3 LC](#)

Cette section fournit des notes de configuration pour implémenter WRED et MDRR sur le LCS de l'engine 3.

- La bande passante MDRR configurée dans le CLI est traduite à une quantité correspondant à L2 (par exemple, le temps système L1 est enlevé). Que la quantité est alors arrondie jusqu'aux prochains 64 Kbits/s et prévue dans le matériel.
- Trois profils différents WRED sont pris en charge pour une classe.
- Le WRED (seuil maximal - seuil minimum) est approché du puissance de 2 le plus proche. Le seuil minimum est alors ajusté automatiquement tandis que le seuil maximal est maintenu inchangé.
- La valeur 1 de probabilité de marque est prise en charge.
- La configuration constante de coefficient exponentiel n'est pas prise en charge.
- La Priorité IP, les bits d'EXP MPLS, et les valeurs DSCP sont pris en charge.

**Remarque:** Chaque port ou canal sur linecards les tétra (4GE-SFP-LC=) ou CHOC12/DS1-IR-SC= de gelure ont quatre files d'attente allouées par défaut. Les quatre files d'attente comprennent ce

qui suit :

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

Quand s'appliquant une service-stratégie contenant plus que ces quatre classes (1 HPQ, 2 LPQs et classe-par défaut) à l'interface, l'erreur suivante sera signalée :

**La #service-stratégie de routeur (config-si) a sorti la mdrd-stratégie**

**% pas assez de ressources en Mise en file d'attente disponibles pour satisfaire la demande.**

En date de 12.0(26)S, une commande a été ajoutée pour le tétra linecard 4GE-SFP-LC= qui permet la configuration de huit queues/VLAN au lieu de quatre. Les huit files d'attente comprennent ce qui suit :

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

L'utilisation de cette commande exigera une recharge de microcode du linecard et aura comme conséquence la capacité de configurer seulement 508 VLAN au lieu de 1022. La syntaxe de la commande est comme suit :

**[non] files d'attente d'interface 8 de qos de <slot#> de hw-module slot**

Exemple :

**files d'attente d'interface 8 de qos de l'emplacement 2 de Router(config)#hw-module**

**Avertissement : S'il vous plaît micro-recharge le linecard pour que cette commande la prenne effet**

**Recharge 2 de Router(config)#microcode**

Cette commande sera disponible pour le linecard de gelure CHOC12/DS1-IR-SC= dans 12.0(32)S

**Exemple #1 - commande de pourcentage de bande passante**

Cet exemple alloue 20 pour cent de bande passante disponible pour classer le trafic Prec\_4 et 30 pour cent pour trafiquer du trafic de la classe Prec\_3. Il laisse les 50 pour cent demeurants à la classe par défaut.

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

#### **Exemple #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 #2 - commande de bande passante {Kbps}

Cet exemple montre comment appliquer la commande bandwidth comme une valeur absolue de Kbps au lieu d'un pourcentage.

### Exemple #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 prioritaire

Cet exemple est conçu pour les fournisseurs de services qui utilisent le routeur de gamme Cisco 12000 en tant que routeur de Provider Edge MPLS (PE) et doivent configurer une stratégie de service QoS sur le lien entre le routeur PE et le routeur de Customer Edge (CE). Il place la Priorité IP 5 paquets dans une file d'attente prioritaire, et limite la sortie de cette file d'attente à 64 Mbits/s. Il assigne alors une partie de la bande passante restante aux classes de bande passante.

Toutes les files d'attente non-prioritaires de classe sont configurées avec la commande de **random-detect** d'activer WRED comme stratégie de baisse. Tous les classe et classe-par défaut de bande passante doivent avoir WRED configuré explicitement.

### 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 - Assignez un policy-map à une file d'attente d'interface sortante

Comme mentionné ci-dessus, le MQC fonctionne seulement avec les files d'attente de FrFab sur une interface sortante. Pour appliquer un policy-map défini, utilisez la commande de **sortie de service-stratégie**, comme affiché 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 de **show policy-map interface** de visualiser l'application d'une stratégie. La commande de **show policy-map interface** affiche ce qui suit :

- Classes de bande passante configurée et prioritaires et correspondance-sur des critères.
- Tous profils WRED.
- Paramètres de forme et de police.
- Comptabilité et débits du trafic.
- La file d'attente interne de cos à laquelle une classe particulière est tracée. Ces files d'attente sont mises en référence par le même index qui est utilisé dans la sortie de la commande de **file d'attente de frfab de show controller**.

Voici un exemple d'une configuration complète et les **commandes show** de 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
policyle 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 policyle !--- Step 3-
Attach policy-map to the interface.
```

Utilisez la commande de **show policy-map interface** de visualiser la stratégie configurée sur l'interface, avec toutes les classes configurées. Voici la sortie de la commande :

```
Router#show policy-map int pos6/0 POS6/0 Service-policy output: policyle (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 surveiller la Gestion et la manière d'éviter d'encombrement

Cette section répertorie les commandes que vous pouvez employer pour surveiller votre stratégie de Gestion et de manière d'éviter d'encombrement.

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

Tableau 8 – Commandes pour les linecards

Carte de ligne d'entrée	Linecard de sortie
<ul style="list-style-type: none"> <li>• <a href="#">show interfaces</a></li> <li>• tofab queue SH de contrôleur de &lt;x&gt; d'emplacement d'exécutif</li> <li>• &lt;port&gt; de &lt;slot&gt; de tofab queue de show controller de &lt;x&gt; d'emplacement d'exécutif</li> <li>• stat de qm de tofab de show controller de &lt;x&gt; d'emplacement d'exécutif</li> </ul>	<ul style="list-style-type: none"> <li>• <a href="#">show interfaces</a></li> <li>• affichez le &lt;y&gt; d'interfaces aléatoire</li> <li>• file d'attente de frfab de show controller de &lt;y&gt; d'emplacement d'exécutif</li> <li>• &lt;port&gt; de file d'attente de frfab de show controller de &lt;y&gt; d'emplacement d'exécutif</li> <li>• stat du frfab QM de show controller de &lt;y&gt; d'emplacement d'exécutif</li> </ul>

Ces commandes est expliquées dans cette section.

### [L'exposition relie la commande](#)

Avant que vous utilisiez cette commande, confirmez la « stratégie correcte de Mise en file d'attente. » Si la sortie affiche le First In, First Out (FIFO), assurez-vous que la commande de service-**stratégie** apparaît en configuration en cours (si MQC a été utilisé pour configurer MDRR).

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

**Remarque:** Sur le routeur de gamme Cisco 12000, les baisses de sortie d'interface sont mises à jour après que les baisses WRED soient mises à jour. Il y a une petite occasion que si vous utilisez un outil pour questionner les deux compteurs de baisse, les baisses d'interface ne sont 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

## L'exposition relie {nombre} la commande aléatoire

Quand vous utilisez cette commande, vous devez :

- Vérifiez que le modèle correct de cos-file d'attente-**groupe** 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 file d'attente et la valeur la plus élevée atteint (en paquets). Les valeurs sont calculées comme moyenne pondérée, et n'ont pas besoin de refléter la profondeur de la file d'attente maximum réelle jamais atteinte.
- Vérifiez le minimum et les seuils maximaux WRED.
- Vérifiez le nombre de baisses aléatoires et de baisses de seuil pour chaque étiquette ROUGE (les baisses « à matrice » indiquent la quantité totale de baisses pour cette étiquette sur tous les linecards).
- Les « baisses de tx-queue-limit » contre- est utilisées seulement sur le LCS de l'engine 1, qui ne prennent en charge pas WRED. Les cartes de l'engine 1 te permettent de fixer la limite des files d'attente MDRR avec la **commande d'interface de tx-queue-limit**. Là où 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
```

## La commande de file d'attente de frfab de show controller de l'emplacement d'exécutif (y) {port}

Cette commande affiche la profondeur de la file d'attente instantanée pour un port donné sur un emplacement donné. La sortie témoin dans cette section affiche la file d'attente MDRR sur l'interface pos 4/1. Vous voyez une profondeur de la file d'attente pour la file d'attente 1 MDRR de 1964 paquets. Le poids est le nombre d'octets qui peuvent être servis sur cette file d'attente. Ce poids détermine le pourcentage de la 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 toujours être servis. Vous pouvez voir qu'il n'y a aucun paquet aligné dans le 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, de surveiller la profondeur de la file d'attente prioritaire du linecard de sortie. Quand vous voyez que le début de paquets à attendre sur ce LLQ, il est une bonne indication que vous devez détourner du trafic de voix sur ip (VOIP) à d'autres linecards de sortie. Dans une bonne conception, la longueur devrait toujours être 0 ou 1. Dans un réseau de vie réelle, vous éprouverez le trafic bursty, même pour des données vocales. Le retard supplémentaire obtient plus sérieux quand tout le chargement de Voix dépasse 100% de la bande passante de sortie pendant une courte période. Le routeur ne peut pas mettre plus de trafic sur le





```
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 quand il y avait d'encombrement sur l'emplacement 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
```

## [La commande de tofab queue de show controller de l'emplacement d'exécutif \(x\) \(emplacement\) \(port\)](#)

Utilisez cette commande de déterminer combien de mémoire est utilisée du côté de ToFab. En particulier, notez le nombre dans 'colonne de #Qelem la ». Notez cela :

- Quand aucune mémoire n'est utilisée, les valeurs sont à leur plus élevé.
- La valeur de la colonne de « #Qelem » diminue pendant que des paquets sont mis en mémoire tampon.
- Quand la colonne de « #Qelem » atteint zéro, toutes les mémoires tampons découpées sont en service. Sur l'Engine 2 LC, les petits paquets peuvent emprunter l'espace de mémoire tampon de plus grands paquets.

Vous pouvez également utiliser cette commande de déterminer le nombre de paquets en attente sur une file d'attente de sortie virtuelle. L'exemple ici affiche comment vérifier l'emplacement 14 pour le nombre instantané de paquets sur ces files d'attente pour l'emplacement 4, le port 1 (POS 4/1). Nous voyons 830 paquets alignés sur la file d'attente 1. MDRR.

```
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
```

## [La commande stat du tofab QM de show controller de l'emplacement d'exécutif \(x\)](#)

Utilisez cette commande de voir le nombre de baisses de ToFab par linecard. Vérifiez également un compteur de « aucune baisse de mémoire » que des incréments. Ce compteur incrémente quand le cos n'est pas configuré du côté de ToFab.

```
Router#execute-on slot 13 show controllers tofab QM stat ===== Line Card (Slot 13) =====
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
Q status errors 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

## [Étude de cas](#)

Cette étude de cas affiche comment configurer une stratégie typique pour le noyau de réseau d'un environnement de fournisseur de services. Il applique des commandes de file d'attente et te permet d'utiliser MDRR/WRED pour la Gestion de file d'attente active. Stratégies QoS dans de périphérie de Routeurs le marquage du trafic d'utilisation normalement, conditionnant, et ainsi de suite, pour permettre à des Routeurs au centre de trier le trafic dans des classes basées sur la Priorité IP ou les valeurs de point de code de DiffServ (DSCP). Cette étude de cas emploie des caractéristiques de QoS de logiciel de Cisco IOS pour atteindre les accords de niveau de service serrés (SLA) et les niveaux de service différent pour la Voix, le vidéo, et les services de données

sur le même circuit principal IP.

Dans l'approche, un fournisseur de services a mis en application trois classes du trafic. Le plus important est la classe de queue LLQ ou de basse latence. C'est la classe pour la Voix et le vidéo. Cette classe doit éprouver un retard et instabilité minimum, et doit ne jamais éprouver la perte de paquets ou les paquets commandés à nouveau tant que la bande passante de cette classe ne dépasse pas la bande passante de lien. Cette classe est connue en tant que trafic de transmission expédié du comportement de Par-saut (PHB E-F) en architecture de DiffServ. Le fournisseur de services Internet (ISP) a conçu le réseau d'une manière dont cette classe ne dépasse pas 30% sur la charge moyenne de la bande passante de lien. Les deux autres classes sont la classe d'affaires et la classe de meilleur effort.

Dans la conception, nous avons configuré les Routeurs de telle manière que la classe d'affaires obtienne toujours 90% de la bande passante restante et la classe de meilleur effort obtienne 10%. Ces deux classes ont moins de trafic sensible à la durée et peuvent éprouver la perte du trafic, le retard plus élevé, et le jitter. Dans la conception, le foyer est sur des linecards d'Engine 2 : 1xOC48 rev B, 4xOC12 rev les linecards B, et 8xOC3.

Des linecards de Rév B best suited pour porter le trafic VoIP en raison d'une architecture révisée ASIC et de matériel, qui introduit la latence très petite. Avec l'ASIC révisé, la file d'attente FIFO de transmission est redimensionnée par le gestionnaire de linecard à approximativement deux fois le plus grand MTU sur la carte. Recherchez « - B » s'est ajouté au numéro de pièce, tel qu'OC48E/POS-SR-SC-B=.

**Remarque:** Ne confondez pas la file d'attente FIFO de transmission avec les files d'attente de FrFab qui peuvent être accordées sur des linecards de l'engine 0 avec la commande d'interface de tx-queue-limit.

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

**Tableau 9 – Critères de correspondance pour chaque classe**

Nom de classe	Critères de correspondance
File d'attente prioritaire - Le trafic vocal	Priorité 5
File d'attente d'affaires	Priorité 4
File d'attente de meilleur effort	Priorité 0

Les linecards OC48 peuvent aligner un grand nombre de paquets dans les tofabs queue. Ainsi, il est important de configurer MDRR/WRED sur les tofabs queue, particulièrement quand l'interface de sortie est une interface à grande vitesse telle qu'OC48. La matrice peut seulement commuter le trafic au linecard de réception à un débit maximum théorique de 3 GBP (1500 paquets d'octets). Si le trafic total envoyé est plus grand que ce que la matrice de commutation peut porter à sa carte de réception, beaucoup de paquets seront alignés sur les tofabs queue.

Interface POS3/0

```
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 sis0 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 sls0 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 sls0 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

```

On le prévoit que le plus que vous avez trafic VoIP, plus le trafic d'affaires doit attendre avant qu'il obtient servi. Cependant, ce n'est pas un problème parce que SLA serré n'exige aucune perte de paquets, et latence et jitter très bas 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 tôt aléatoire pesée sur le routeur de gamme Cisco 12000](#)
- [Aperçu de qualité de service modulaire d'interface de ligne de commande](#)
- [Page de support de Routeurs d'Internet de gamme 12000](#)
- [Support technique - Cisco Systems](#)