

WRED et MDRR sur le routeur Internet de la gamme Cisco 12000 avec un exemple de mélange de configuration Unicast, Multicast et trafic vocal

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Conventions](#)

[Informations générales](#)

[Classes de priorité utilisées dans le test](#)

[Coloration des paquets IP](#)

[Résultats escomptés](#)

[Configuration du réseau](#)

[Mise en file d'attente du moteur GbE 2 à 3 ports](#)

[Algorithme WRED du moteur 2 dans le logiciel Cisco IOS](#)

[Configuration QoS](#)

[CoS Rx](#)

[QoS Tx](#)

[Mappage d'interface](#)

[À partir de No Drops](#)

[Quatre flux de données de 151 Mo chacun](#)

[Quatre flux de données de 160 Mo chacun](#)

[Quatre flux de données de 167 Mo chacun](#)

[Quatre flux de données de 191 Mo chacun](#)

[Quatre flux de données de 244 Mo chacun](#)

[QoS supprimée](#)

[Quatre flux de données de 153 Mo chacun](#)

[Quatre flux de données de 158 Mo chacun](#)

[Ajout de la charge entrante](#)

[12-RP#show interfaces g 6/0](#)

[Modification de la taille des flux](#)

[Vérification avec une carte de ligne 4 Gigabit Ethernet Engine 10 ports](#)

[Informations connexes](#)

[Introduction](#)

Ce document explique comment configurer une carte de ligne Cisco 12000 pour la détection WRED (Weighted Random Early Detection), décrite dans [RFC 2309](#) , dans un environnement multiservice.

Conditions préalables

Conditions requises

Les lecteurs de ce document doivent avoir une bonne connaissance de ce qui suit :

- [Présentation et configuration de MDRR et WRED sur les routeurs Internet de la gamme Cisco 12000](#)
- [Comment lire la sortie de la commande show controller frfab | Commandes tofab queue sur un routeur Internet de la gamme Cisco 12000](#)
- [Type de service dans la suite de protocoles Internet, priorité \(RFC-1349\)](#)
- [Détection précoce pondérée aléatoire](#)

Components Used

L'information fournie dans ce document est basée sur les versions logicielles et matérielles suivantes :

- Toute version du logiciel Cisco IOS® prenant en charge le routeur Internet de la gamme Cisco 12000. Il s'agit généralement des versions 12.0S et 12.0ST.
- Toutes les plates-formes Cisco 12000 sont couvertes par ce document. Il s'agit des numéros 12008, 12012, 12016, 12404, 12406, 12410 et 12416.

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

Pour plus d'informations sur les conventions des documents, référez-vous aux [Conventions utilisées pour les conseils techniques de Cisco](#).

Informations générales

La gamme Cisco 12000 est l'une des plates-formes les plus utilisées pour créer un réseau IP de coeur de réseau à bande passante élevée. Cette plate-forme offre la possibilité exclusive de configurer la qualité de service (QoS).

Comme il est de plus en plus courant de combiner différents types de trafic IP (tels que la voix sur IP - VoIP - et la multidiffusion) dans un même réseau, les exigences de hiérarchisation et de comportement de suppression contrôlé deviennent extrêmement importantes et, dans de nombreux cas, sont la différence entre la réussite et l'échec lors du lancement d'un nouveau service tel que la VoIP.

La configuration réseau requise pour différents types de trafic IP dépasse le cadre de ce

document. Ce document se concentre sur les tests de laboratoire effectués pour trouver une configuration applicable sur différentes cartes de ligne, y compris la carte de ligne Gigabit Ethernet 3 ports (GbE 3 ports) de la gamme Cisco 12000. Les résultats de ces tests montrent que la carte de ligne 3 ports GbE Engine 2 convient parfaitement à un environnement réseau impliquant un mélange de trafic voix, données et multidiffusion. Il prouve également que la configuration de la QoS fait une réelle différence dans un réseau encombré.

Classes de priorité utilisées dans le test

Les valeurs de priorité attribuées à différentes classes doivent être identiques sur l'ensemble du réseau. Vous devez déterminer une stratégie générique.

Classe	Priorité	Trafic
Trafic malveillant		Tout le trafic hors réseau non identifié
Sur réseau - sur réseau	1	Trafic qui reste dans le réseau SP (sur réseau)
Services de fournisseur d'accès Internet (FAI)	2	Trafic ISP, SMTP, POP, FTP, DNS, Telnet, SSH, www, https
PME (petites et moyennes entreprises)	3	Clients d'entreprise, un service Gold
Temps réel, non vocal	4	TV, jeux en temps réel
Voix	5	Trafic VOIP RTP
Messages de contrôle réseau	6-7	BGP (Border Gateway Protocol) et autres messages de contrôle

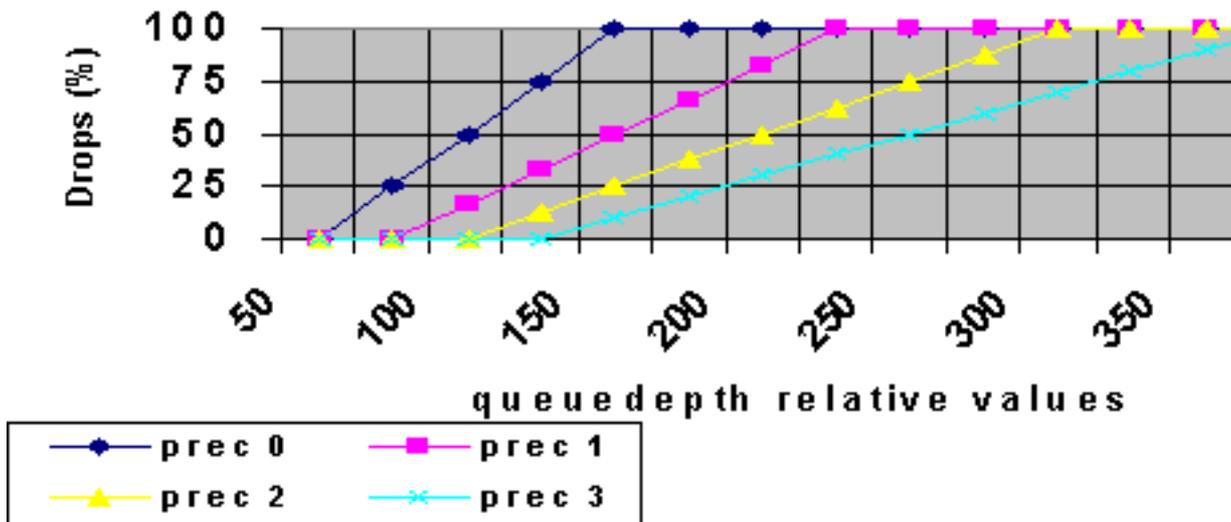
Coloration des paquets IP

Si la QoS doit être mise en oeuvre au coeur d'un réseau, une condition préalable est que le fournisseur de services soit en contrôle total de la valeur de priorité de *tous les* paquets IP transportés sur le réseau. La seule façon de procéder est de marquer tous les paquets lorsqu'ils pénètrent dans le réseau, sans distinction entre les paquets provenant du côté client/utilisateur final ou d'Internet. Aucun marquage ou coloration ne doit être effectué dans le coeur.

Résultats escomptés

L'objectif de cette conception est d'avoir un vrai comportement WRED dans les classes 0-3. Cela signifie que nous aimerions avoir une situation où nous commençons à supprimer les paquets de priorité 0 pendant la congestion. Après cela, nous devrions également commencer à supprimer la priorité 1 si l'encombrement continue, puis aussi la priorité 2 et 3. Tout cela est décrit dans le graphique ci-dessous.

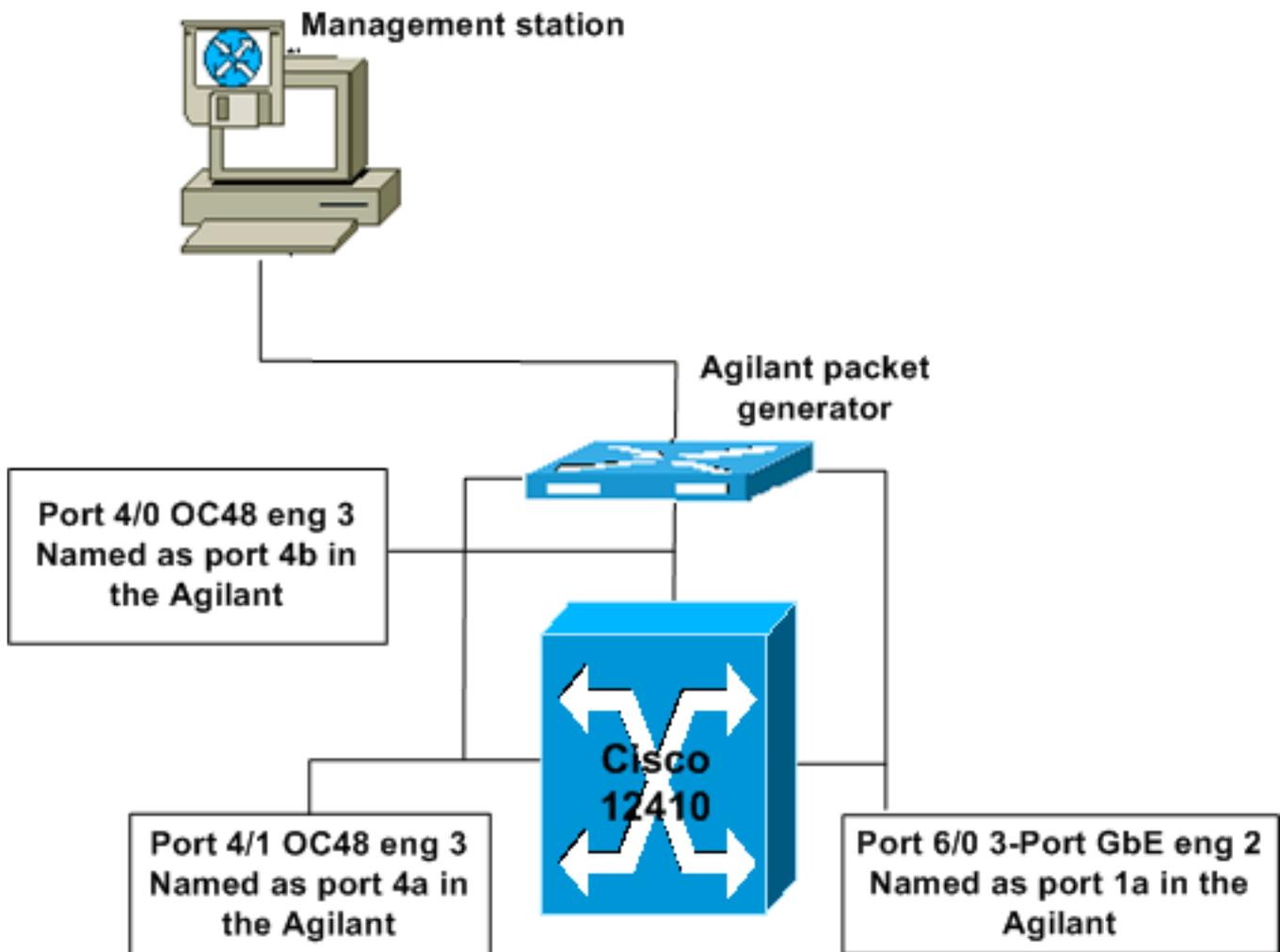
WRED behaviour



Vous devez disposer de la latence la plus faible possible pour les paquets voix, et aucune baisse du tout pour le trafic voix et multidiffusion.

Configuration du réseau

Pour tester et évaluer la configuration, nous avons utilisé un Cisco 12410 avec un générateur de paquets d'Agilant. Le routeur Cisco 12000 exécute une version d'ingénierie basée sur le logiciel Cisco IOS Version 12.0(21)S1.



Mise en file d'attente du moteur GbE 2 à 3 ports

Les cartes du moteur 2 comportent normalement huit files d'attente de sortie et une file d'attente à faible latence, ainsi que huit files d'attente de lancement par emplacement de destination. Il existe également une file d'attente de multidiffusion tofab distincte. Sur la carte GbE à 3 ports, il n'y a qu'une file d'attente de sortie par port physique. Dans le test, la configuration appliquée spécifie plus de files d'attente. Les résultats montrent que la carte GbE à 3 ports accepte cette configuration et que les files d'attente sont automatiquement mappées aux files d'attente disponibles.

Algorithme WRED du moteur 2 dans le logiciel Cisco IOS

Vous devez bien comprendre l'algorithme utilisé pour WRED dans la carte de ligne du moteur 2 lors de la configuration des valeurs de profondeur de file d'attente minimale et maximale. Le code ne se soucie pas de la valeur minimale configurée ; au lieu de cela, il utilise sa propre formule (basée sur la valeur maximale configurée) pour définir la valeur minimale.

Formule :

Valeur minimale = Valeur maximale - (puissance maximale de 2 qui ne génère pas de résultat négatif)

Les valeurs utilisées dans ce test ont abouti aux valeurs minimales suivantes programmées pour le circuit intégré spécifique à l'application (ASIC) :

Priorité	Minimum configuré	Maximum configuré	Puissance maximale de 2	Valeur minimale dans ASIC
0	50	5000	4096	5000-4096 = 904
1	60	6000	4096	6000-4096 = 1904
2	70	7000	4096	7000-4096 = 2904
3	80	8000	4096	8000-4096 = 3904

L'utilisation de cette formule pour calculer la valeur minimale signifie que vous pouvez vous retrouver avec un comportement de gestion de paquets incorrect si vous ne le prenez pas en compte lors de la configuration de vos paramètres WRED. Ceci est illustré dans l'exemple suivant :

Priorité	Minimum configuré	Maximum configuré	Puissance maximale de 2	Valeur minimale dans ASIC
0	50	150	128	150-128 = 22
1	75	225	128	225-128 = 97
2	100	300	256	300-256 = 44
3	125	375	256	375-256 = 119

Cela signifie que, même si les valeurs sont configurées pour commencer à chuter conformément à la règle 0 d'abord, puis 1, 2 et enfin 3 (ci-dessus), lorsque les valeurs sont écrites dans l'ASIC, vous commencez en fait à supprimer la priorité 0, puis la priorité 2, puis la priorité 1, et enfin la priorité 3. Il est impossible de voir quelles valeurs ont été configurées dans l'ASIC sur une carte du moteur 2. Si vous appliquez la configuration sur une carte du moteur 3, les valeurs affichées dans la configuration seront les valeurs réelles (la valeur minimale recalculée).

[Configuration QoS](#)

Pour plus de détails sur la configuration QoS, lisez [Présentation et configuration de MDRR et WRED sur le routeur Internet de la gamme Cisco 12000](#).

[CoS Rx](#)

```
rx-cos-slot 2 B2-Table
rx-cos-slot 3 B2-Table
rx-cos-slot 6 B2-Table
```

Dans la plupart des cas, vous pouvez utiliser la commande **rx-cos-slot all**. Dans notre cas de test, nous avons des cartes qui ne supportaient pas la file d'attente tofab, donc nous ne pouvions pas

toujours utiliser la commande **rx-cos-slot all**. Au lieu de cela, nous avons affecté notre table à logements aux cartes de ligne utilisées dans le test.

```
!  
slot-table-cos B2-Table  
destination-slot all B2  
multicast B2 !--- If you don't fulfill the steps above, you will not be able to reach the goal  
of 0 drops for Multicast traffic. With no rx-cos configured, multicast will be treated in the  
default queue, meaning it will drop as soon as there is congestion. !
```

QoS Tx

Vous pouvez maintenant configurer vos codes d'accès. Choisissez un nom pour votre qos tx, tel que « cos-queue-group B2 ».

Chaque valeur de priorité pour laquelle vous souhaitez configurer un comportement de suppression doit être mappée à une étiquette de détection aléatoire distincte.

```
precedence 0 random-detect-label 0  
  
precedence 1 random-detect-label 1  
  
precedence 2 random-detect-label 2  
  
precedence 3 random-detect-label 3
```

Pour MDRR (Modified Deficit Round Robin), mappez chaque priorité à une file d'attente MDRR. Dans notre exemple, nous avons mappé la priorité 0-3 à la même file d'attente MDRR afin de réserver de la bande passante pour la vidéo (trafic de multidiffusion). Ce mappage fournit le comportement demandé.

```
precedence 0 queue 0  
  
precedence 1 queue 0  
  
precedence 2 queue 0  
  
precedence 3 queue 0  
  
precedence 4 queue 4
```

La voix est marquée avec la priorité 5, c'est pourquoi la priorité 5 est mappée à la file d'attente à faible latence.

```
precedence 5 queue low-latency  
  
precedence 6 queue 6  
  
precedence 7 queue 6
```

Maintenant, vous devez configurer le comportement de suppression pour chacun des labels de détection aléatoire. Au cours du test, ces nombres ont été modifiés jusqu'à ce que les valeurs obtenues donnent le modèle de chute souhaité. Pour plus de détails, voir la section [Résultats attendus](#). La profondeur de la file d'attente est mesurée sur la file d'attente physique, et non sur la

file d'attente MDRR ou RED-Label.

```
random-detect-label 0 50 5000 1
random-detect-label 1 60 6000 1
random-detect-label 2 70 7000 1
random-detect-label 3 80 8000 1
```

Sur le Cisco 12000, il est possible de créer un comportement CBWFQ (Class-Based, Weighted Fair Queuing) en donnant un poids aux différentes files d'attente MDRR. Le poids par défaut est de 10 par file d'attente. Le nombre d'octets transmis à chaque cycle MDRR est fonction de la valeur de poids. Une valeur de 1 signifie 1 500 octets par cycle. Une valeur de 10 signifie plus de 1 500 octets (9*512) par cycle. »

```
queue 0 20
queue 4 20
queue 6 20
!
```

Mappage d'interface

Chaque interface doit être configurée pour WRED. Pour ce faire, utilisez les commandes suivantes :

- configurer le terminal
- interface gig 6/0
- tx-cos B2

À partir de No Drops

Le flux généré utilise les valeurs suivantes, sauf indication contraire :

```
MTU all three data streams 300byte, MTU voice 80byte, MTU MC 1500byte
```

```
126Mb MC, 114Mb voip
```

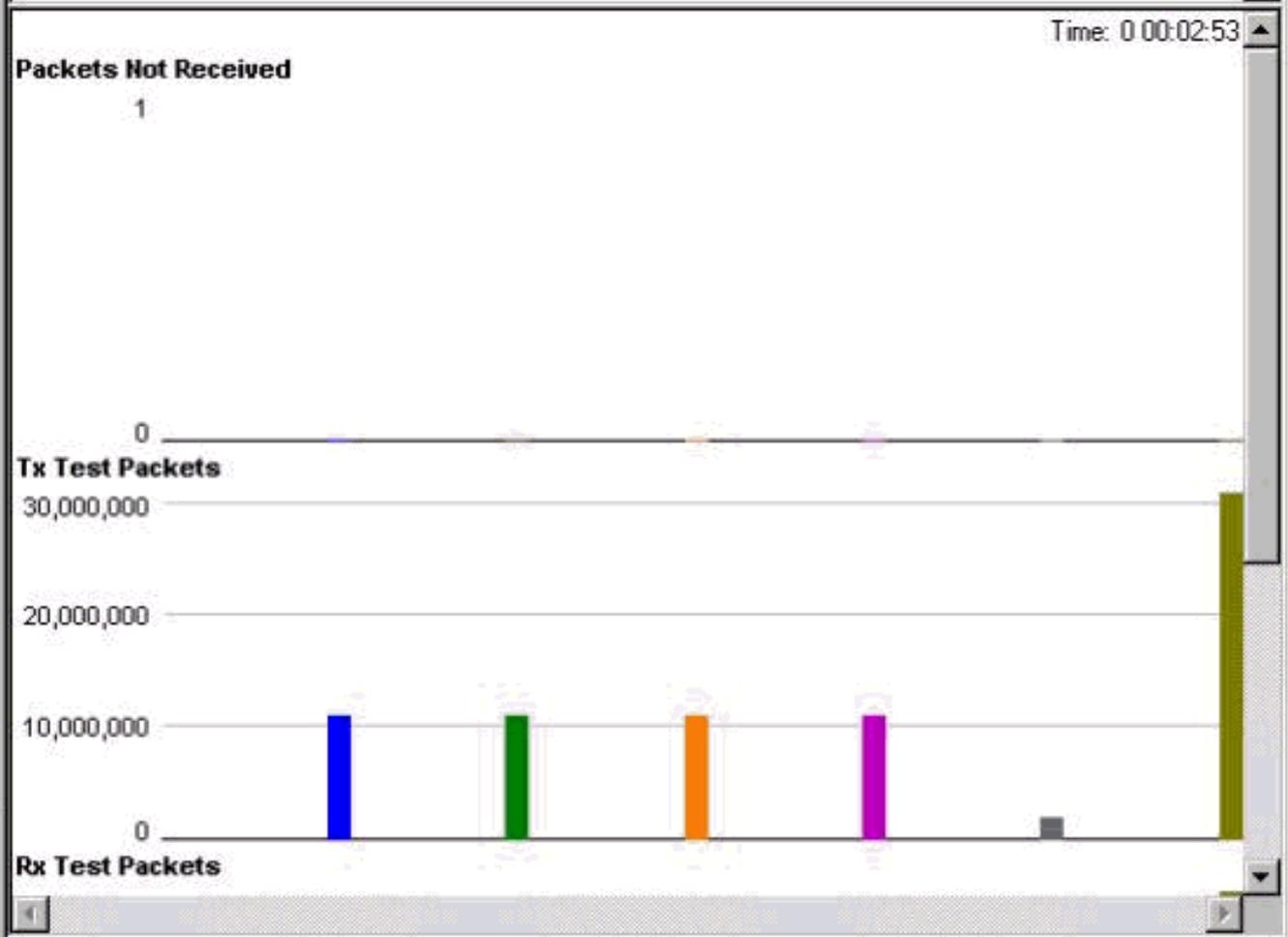
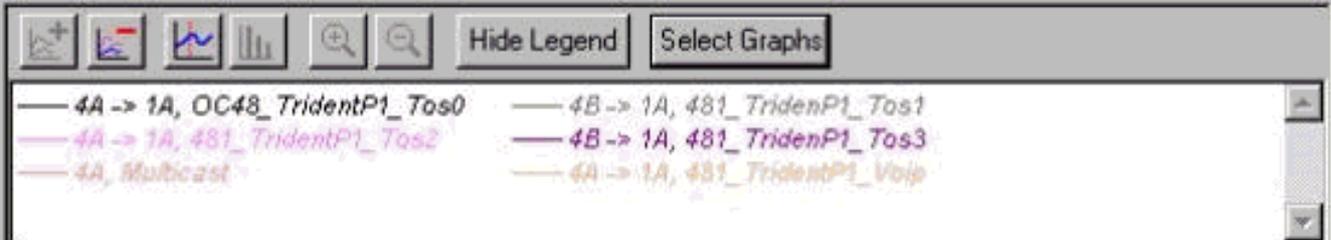
Cela donne un flux d'arrière-plan de 240 Mo (VoIP et multidiffusion).

Nous ajoutons ensuite quatre flux de données de même taille, mais avec priorité 0-3 (une valeur de priorité par flux).

Quatre flux de données de 151 Mo chacun

Cette configuration fournit une bande passante totale de 844 Mo. Le graphique ci-dessous montre qu'il y a 0 pertes de paquets et une latence très faible (environ 50 us - microsecondes - pour chaque flux, voix comprise).

Ports/Streams	Seq Errors	Packets Not R...	Avera...	Max Latency...	T: ▲
- 4B -> 1A, 481_TridenP1_Tos1	0		30.39	44.36	
- 4B -> 1A, 481_TridenP1_Tos3	0		35.29	48.94	
- 4A -> 1A, 481_TridenP1_Tos2	0		33.73	48.98	
- 4A -> 1A, 0C48_TridenP1_Tos0	0		28.90	44.50	
- 4A -> 1A, 481_TridenP1_Voip	0		32.99	52.84	
+ Port 1B			51.04	56.58	
+ Port 1C			45.12	49.80	
+ Port 1D		0			



[Quatre flux de données de 160 Mo chacun](#)

Cette configuration fournit une bande passante totale de 880 Mo. Le graphique ci-dessous montre que les paquets commencent à tomber de la classe de priorité 0 et que la latence de la voix a augmenté à 6 ms - millisecondes.

Ports/Streams	Seq Errors	Packets Not R...	Avera...	Max Latency...	T: ▲
- 4B -> 1A, 481_TridentP1_Tos1	0		1566.55	13117.44	
- 4B -> 1A, 481_TridentP1_Tos3	0		1571.17	13118.92	
- 4A -> 1A, 481_TridentP1_Tos2	0		1570.53	13124.36	
- 4A -> 1A, DC48_TridentP1_Tos0	114207	162429	1408.37	13117.96	
- 4A -> 1A, 481_TridentP1_Voip	0		815.62	6337.22	
+ Port 1B			847.10	6474.38	
+ Port 1C			834.87	6395.62	
+ Port 1D		0			







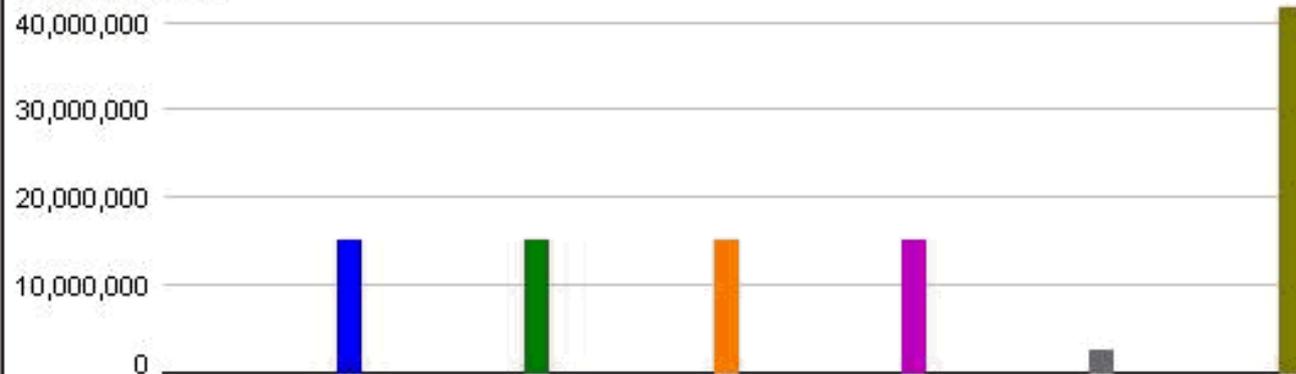

— 4A -> 1A, OC48_TridentP1_Tos0 — 4B -> 1A, 481_TridentP1_Tos1
 — 4A -> 1A, 481_TridentP1_Tos2 — 4B -> 1A, 481_TridentP1_Tos3
 — 4A, Multicast — 4A -> 1A, 481_TridentP1_Voip

Time: 0 00:03:54 ▲

Packets Not Received



Tx Test Packets

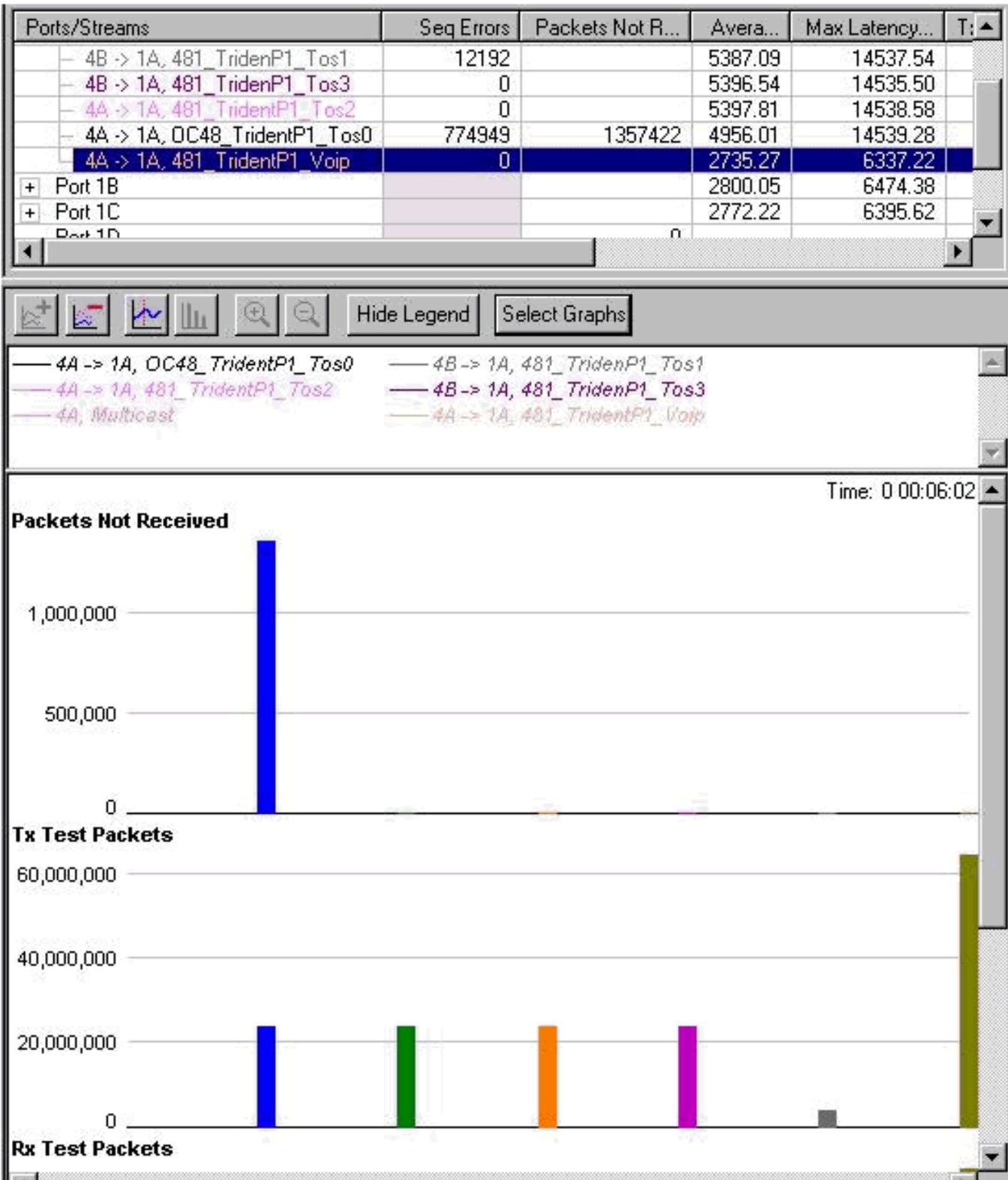


Rx Test Packets

Quatre flux de données de 167 Mo chacun

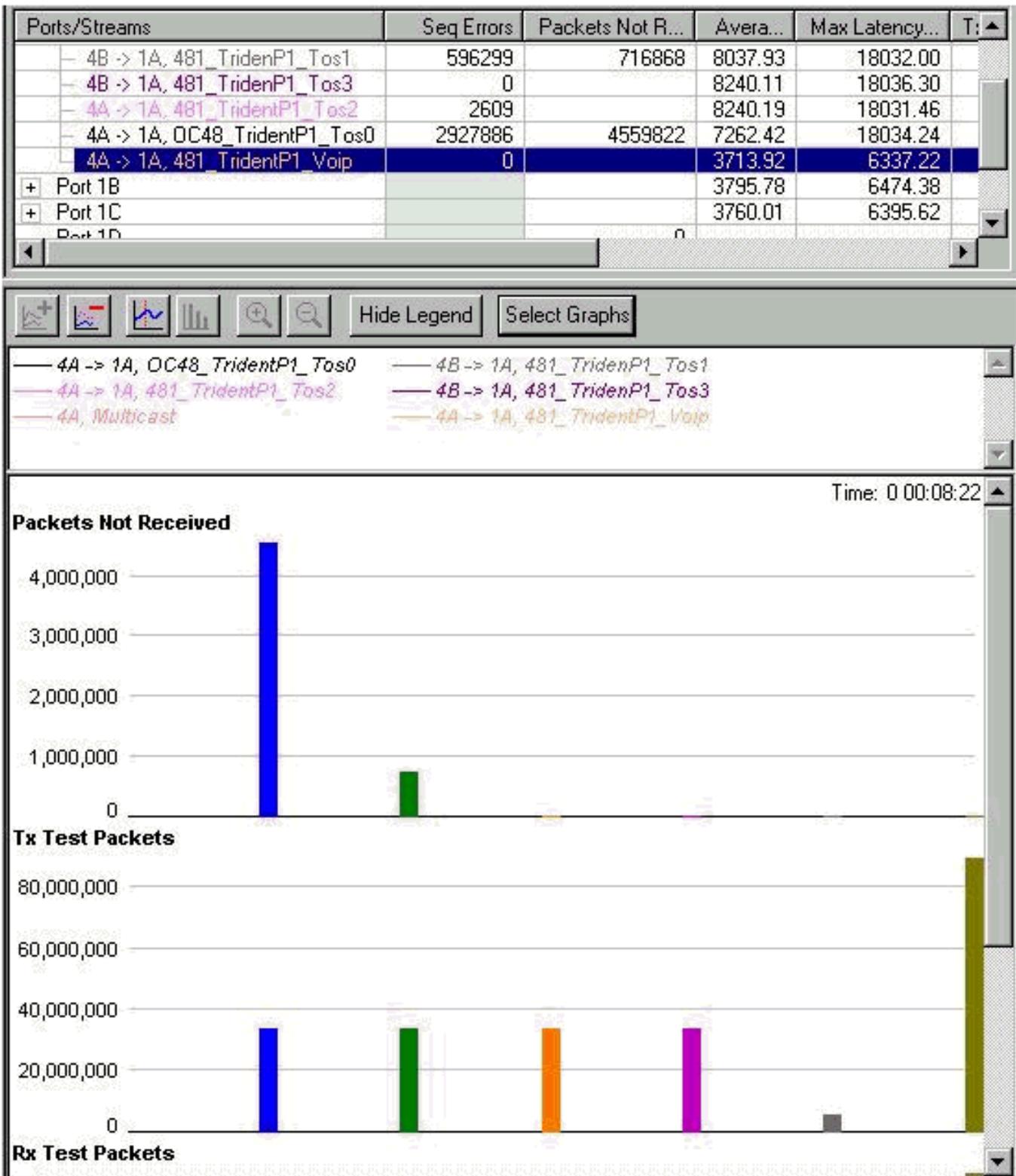
Cette configuration fournit une bande passante totale de 908 Mo. Les abandons commencent maintenant pour la classe de priorité 1 également. La latence du trafic vocal est toujours la même.

Note : Le flux n'a pas été arrêté avant d'être augmenté, de sorte que la différence entre le nombre de gouttes dans le flux 0 et 1 est cumulative.



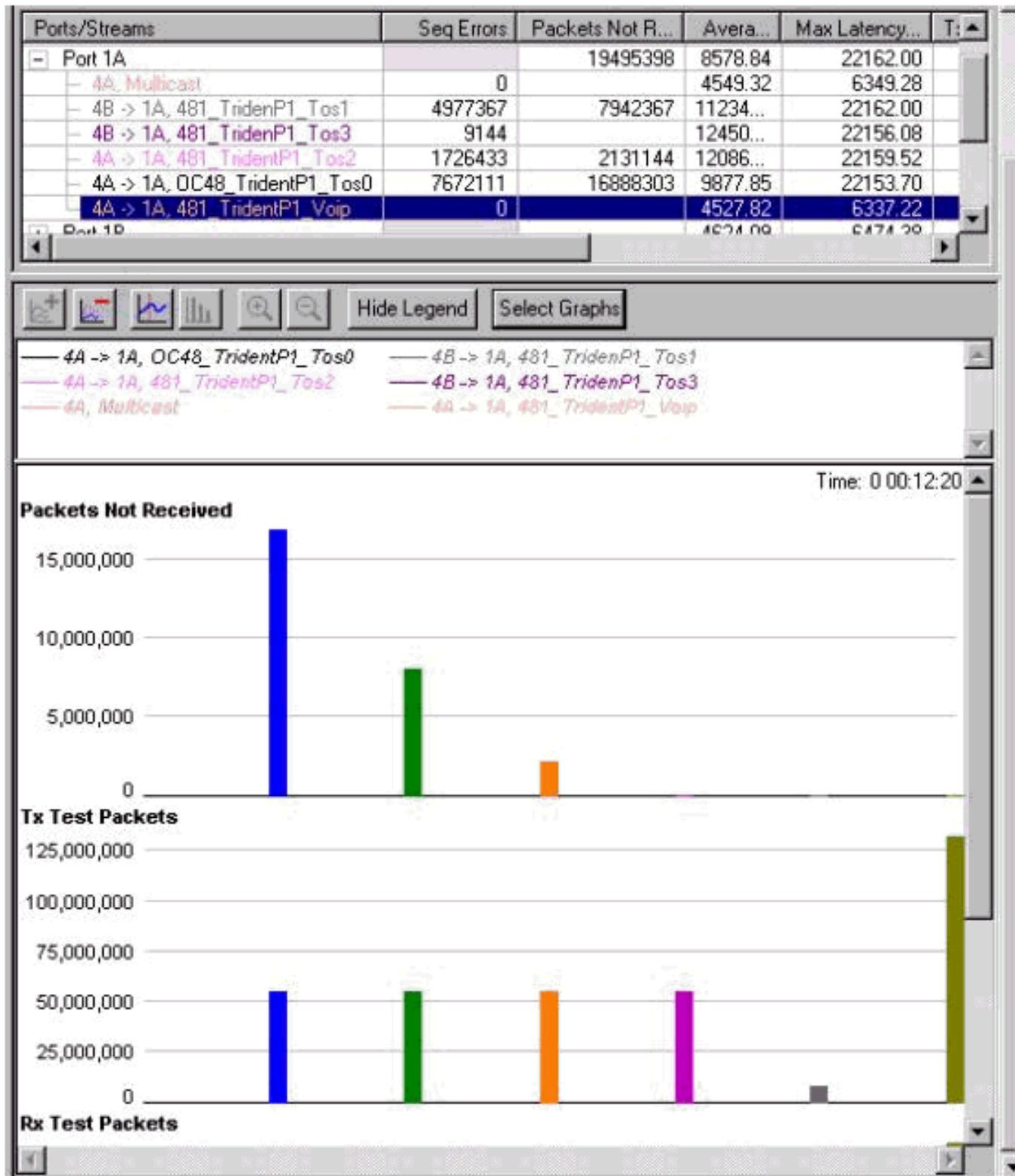
[Quatre flux de données de 191 Mo chacun](#)

Lorsque la bande passante totale augmente, les paquets commencent également à tomber de la file d'attente prioritaire 2. La bande passante totale que nous tentons d'atteindre pour l'interface Gigabit Ethernet est maintenant de 1 004 Mo. Ceci est illustré dans les compteurs d'erreur de séquence dans le graphique ci-dessous.



Quatre flux de données de 244 Mo chacun

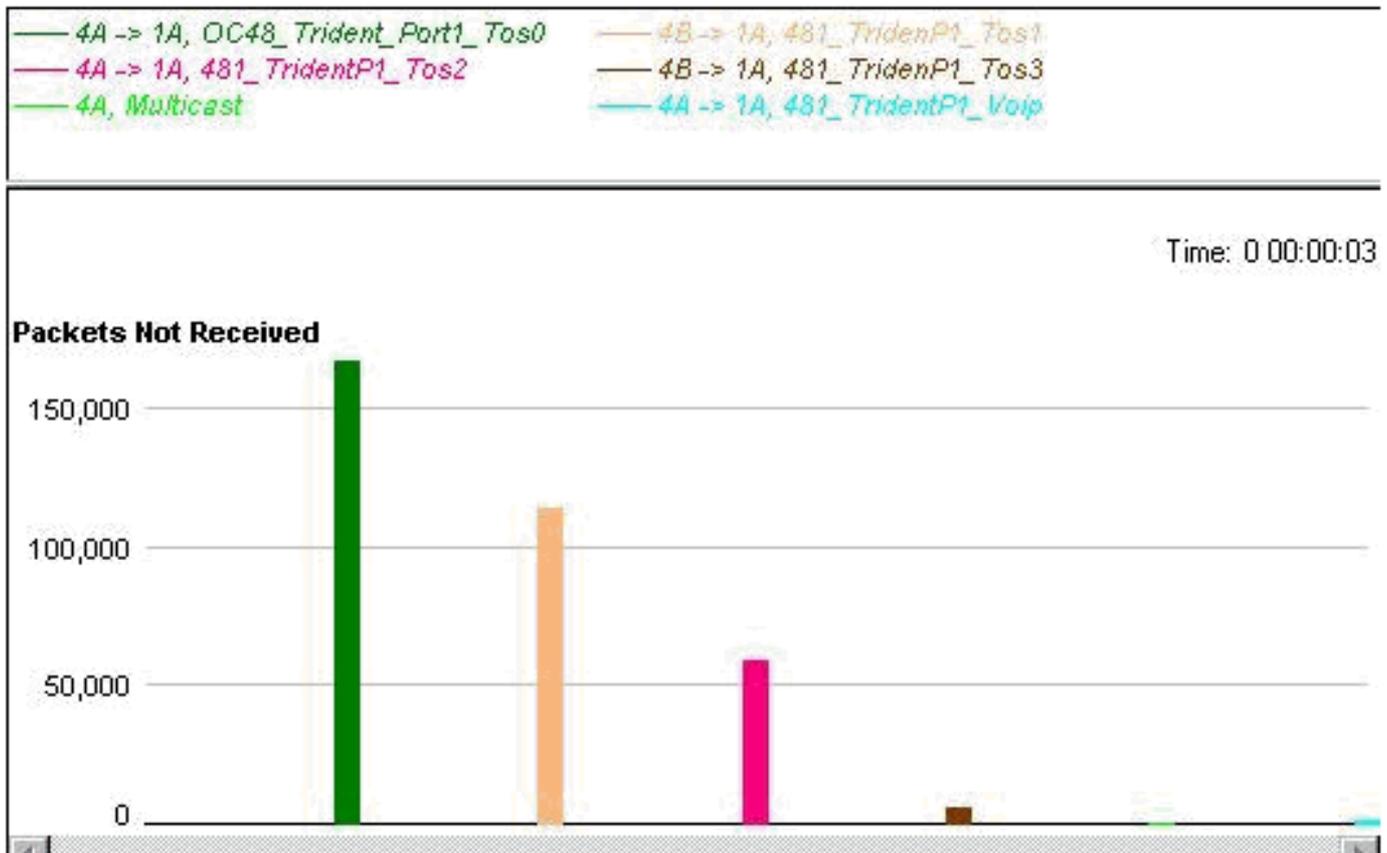
Les erreurs de séquence pour la priorité 3 commencent également à augmenter. C'est le premier signe que les abandons commenceront à partir de cette file d'attente. La bande passante totale que nous tentons d'envoyer à l'interface GbE est maintenant de 1 216 Mo. Notez que les pertes sur la multidiffusion (MC) et la file d'attente vocale sont toujours nulles et que la latence de la file d'attente vocale n'a pas augmenté.



Arrêt et démarrage

Tous les flux ont été arrêtés et ont commencé à générer un graphique qui a effacé les compteurs. Cela montre comment cela va se passer en cas de forte congestion. Comme vous pouvez le voir ci-dessous, le comportement est celui qui est souhaité.

Ports/Streams	Packets Not Received
All Ports	244885
[-] Port 1A	259941
[-] 4A -> 1A, OC48_Trident_Port1_Tos0	133621
[-] 4A -> 1A, 481_TridentP1_Tos2	45024
[-] 4A -> 1A, 481_TridentP1_Voip	
[-] 4A, Multicast	
[-] 4B -> 1A, 481_TridentP1_Tos1	89282
[-] 4B -> 1A, 481_TridentP1_Tos3	624



QoS supprimée

Pour prouver que la QoS améliore réellement les performances en cas d'encombrement, la QoS a été supprimée et l'interface a été congestionnée. Les résultats sont ci-dessous (le flux généré utilise les valeurs suivantes sauf indication contraire).

MTU all three data streams 300byte, MTU voice 80byte, MTU MC 1500byte

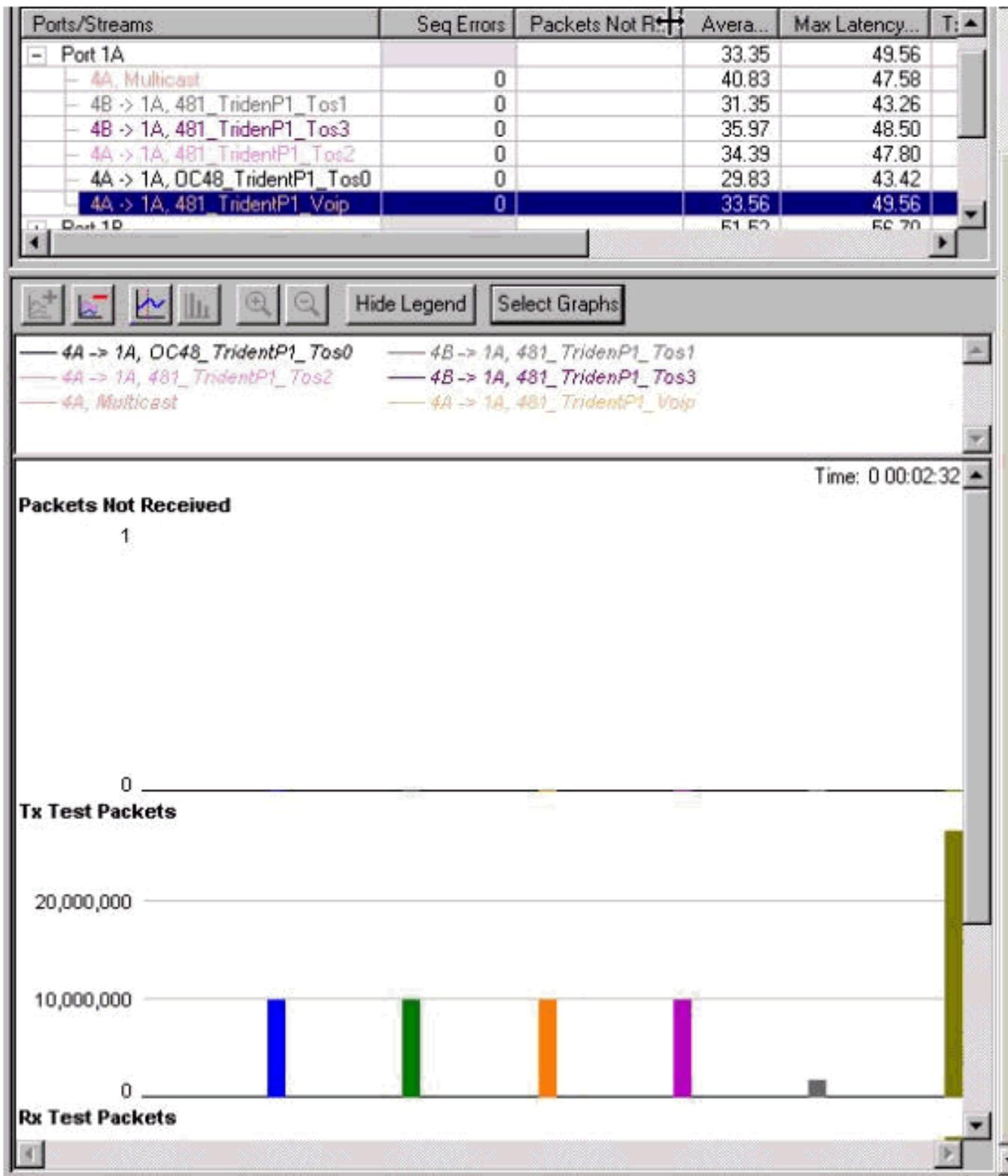
126Mb MC, 114Mb VoIP

Cela donne un flux d'arrière-plan de 240 Mo (VoIP et multidiffusion).

Nous ajoutons ensuite quatre flux de données de même taille, mais avec priorité 0-3 (une valeur de priorité par flux).

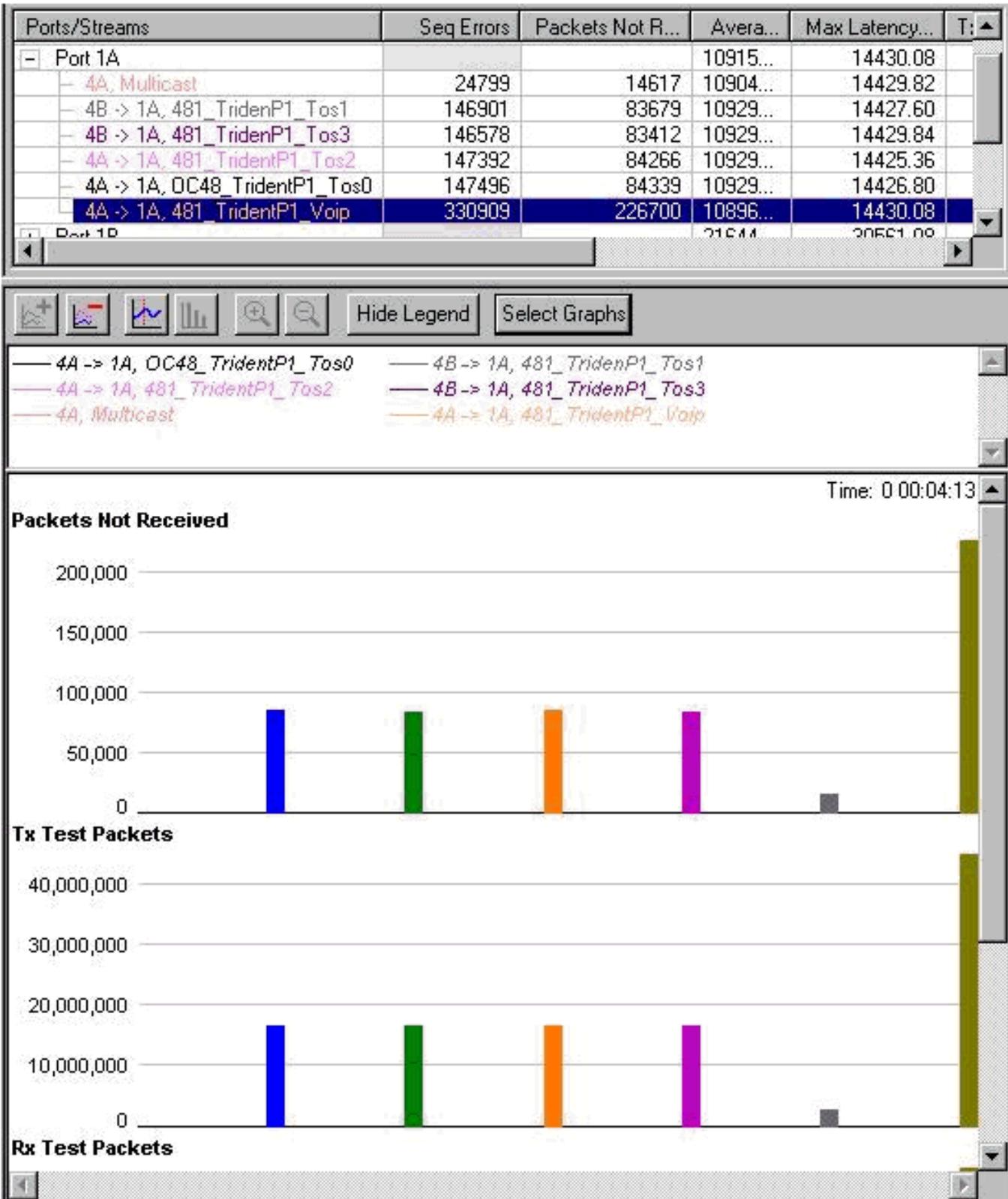
Quatre flux de données de 153 Mo chacun

Cela donne un total de 852 Mo. Il y a 0 gouttes, et une latence de moins de 50 us.



[Quatre flux de données de 158 Mo chacun](#)

Nous commençons à diminuer à environ la même utilisation que lorsque WRED est appliqué (872 Mo). La différence est maintenant qu'il y a une latence de paquets vocaux de 14 ms (plus de deux fois supérieure à celle du test WRED), et que les abandons se produisent également de toutes les classes, y compris VoIP et multicast.



[Ajout de la charge entrante](#)

Jusqu'à présent, tous les tests n'ont inclus que la transmission via les interfaces Gigabit Ethernet. Pour vérifier comment l'interface réagit dans une situation où nous congestionnons également l'interface dans l'autre direction, les tests suivants ont été effectués.

Pour ce test, nous avons chargé l'interface Gigabit Ethernet avec une quantité totale de 1 056 Mo. Cela a entraîné des pertes sur la priorité 0-2, aucune baisse sur le trafic de priorité 3. (MC et VOIP sont restés les mêmes, c'est-à-dire qu'il n'y a pas de chute). Nous avons ensuite ajouté du trafic

dans l'autre direction, autant de trafic que le générateur de paquets a pu envoyer via l'interface Gigabit Ethernet. Le résultat est assez impressionnant : la congestion de réception n'interfère pas du tout avec la file d'attente de transmission, et la latence du trafic de réception est extrêmement faible, moins de 13 us pour la voix.

- Port 1A	21976403	12050.70	20237.92
- 4A, Multicast		6277.06	6402.88
- 4B -> 1A, 481_TridenP1_Tos1	7729260	16763.90	20232.04
- 4B -> 1A, 481_TridenP1_Tos3		17287.73	20237.92
- 4A -> 1A, 481_TridenP1_Tos2	2090730	17139.98	20233.44
- 4A -> 1A, OC48_TridenP1_Tos0	17210053	16519.08	20236.76
- 4A -> 1A, 481_TridenP1_Voip		6245.21	6376.02
+ Port 1B		6379.18	6512.30
+ Port 1C		6323.52	6432.94
Port 1D	0		
- Port 4A	7	13.09	15.02
- 1A -> 4A, 481_Triden_P3_tos3		13.37	15.02
- 1A -> 4A, 481_TridenP3_Voip		12.67	14.30
- 1A -> 4A, 48_TridenP3_tos2		13.23	14.94
- Port 4B	6	13.11	14.64
- 1A -> 4B, 48_Triden_P3_Tos0		13.11	14.62
- 1A -> 4B, 481_TridenP3_Tos1		13.11	14.64

Ports/Streams	Tx Test Throughput (Mb/s)	Rx Test Throughput (Mb/s)
All Ports		
- Port 1A	747.47	858.75
- 4A, Multicast	126.24	126.24
- 4B -> 1A, 481_TridenP1_Tos1	189.55	142.03
- 4B -> 1A, 481_TridenP1_Tos3	189.54	189.54
- 4A -> 1A, 481_TridenP1_Tos2	189.54	175.70
- 4A -> 1A, OC48_TridenP1_Tos0	189.54	95.24
- 4A -> 1A, 481_TridenP1_Voip	130.00	130.00
+ Port 1B	0.00	126.24
+ Port 1C	0.00	126.24
Port 1D	0.00	0.00
- Port 4A	635.32	424.24
- 1A -> 4A, 481_Triden_P3_tos3	161.62	161.62
- 1A -> 4A, 481_TridenP3_Voip	101.01	101.01
- 1A -> 4A, 48_TridenP3_tos2	161.62	161.62
- Port 4B	379.09	323.23
- 1A -> 4B, 48_Triden_P3_Tos0	161.62	161.62
- 1A -> 4B, 481_TridenP3_Tos1	161.62	161.62

[12-RP#show interfaces g 6/0](#)

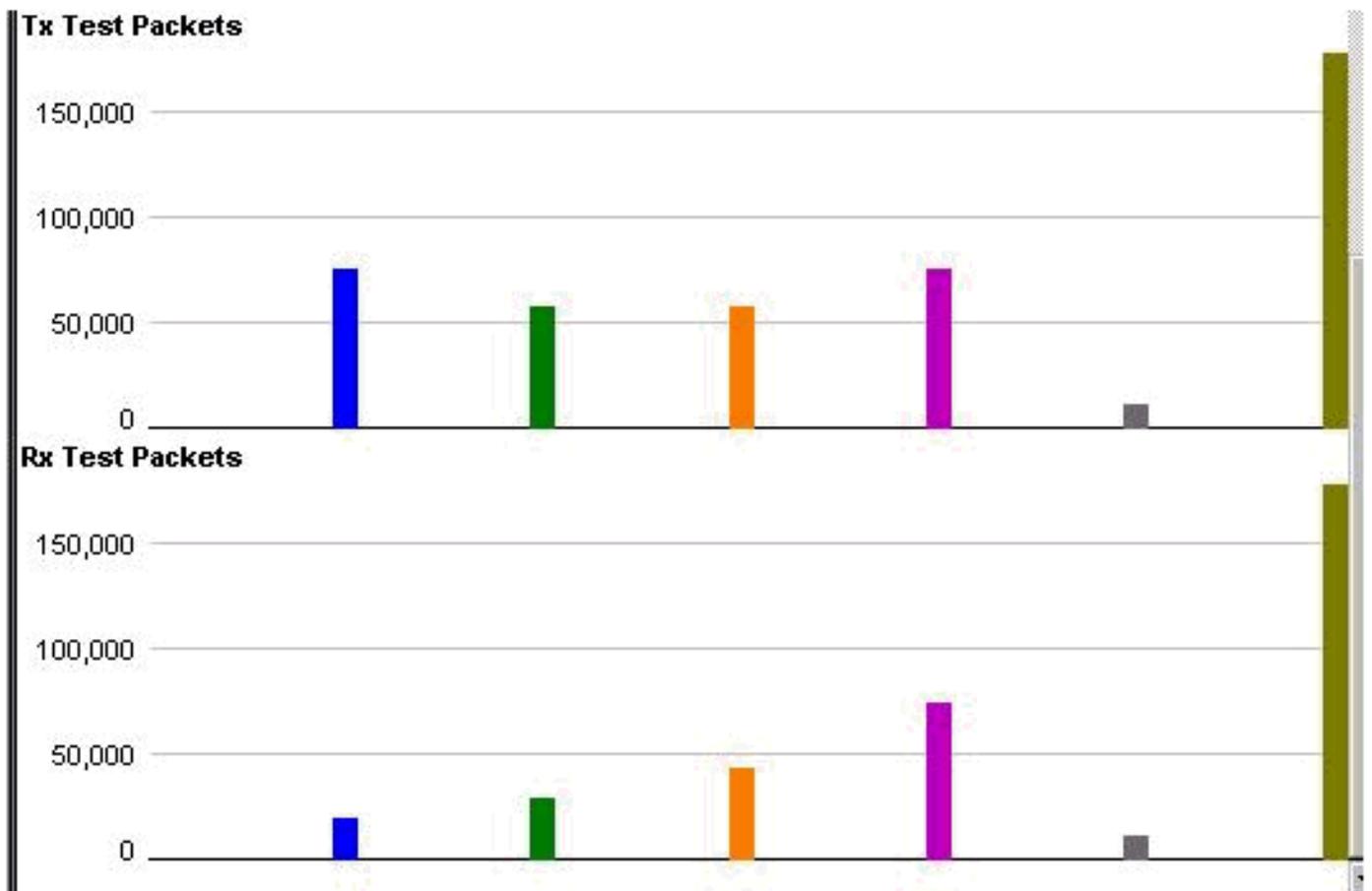
Vous pouvez surveiller la charge sur la liaison Gigabit à l'aide de la commande **show interfaces** :

```
Router#show interfaces gig 6/0
GigabitEthernet6/0 is up, line protocol is up
  Hardware is GigMac 3 Port GigabitEthernet, address is 0004.de56.c264
  (bia 0004.de56.c264)
  Internet address is 178.6.0.1/24
```

```
MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec, rely 255/255, load 232/255
Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
Full-duplex mode, link type is force-up, media type is SX
output flow-control is unsupported, input flow-control is off
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:00:05, output 00:00:05, output hang never
Last clearing of "show interface" counters 08:52:40
Queueing strategy: random early detection (WRED)
Output queue 0/40, 2174119522 drops; input queue 0/75, 0 drops
30 second input rate 838969000 bits/sec, 792079 packets/sec
30 second output rate 910819000 bits/sec, 464704 packets/sec
 7584351146 packets input, 1003461987270 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
0 watchdog, 514 multicast, 0 pause input
11167110605 packets output, 2241229569668 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 babbles, 0 late collision, 0 deferred
0 lost carrier, 0 no carrier, 0 pause output
0 output buffer failures, 0 output buffers swapped out
```

Modification de la taille des flux

Pour vérifier que les résultats du test ne sont pas dus au fait que la bande passante est identique pour tous les flux, nous avons modifié les flux de sorte qu'ils transmettent différentes quantités de données. Nous avons également essayé de modifier l'unité de transmission maximale (MTU) de sorte qu'elle était différente pour chaque flux. Avec les valeurs de file d'attente configurées, le résultat était toujours le même, en supprimant la priorité 0 d'abord, puis 1, puis 2, et enfin la priorité 3.



[Vérification avec une carte de ligne 4 Gigabit Ethernet Engine 10 ports](#)

Comme la latence de la file d'attente VoIP (file d'attente à faible latence) dans le test était relativement élevée, nous avons effectué le même test avec la carte de ligne Gigabit Ethernet Engine 4 à 10 ports. Comme prévu, le résultat avec cette carte de ligne était bien meilleur en ce qui concerne la latence dans la file d'attente à faible latence (LLQ). Les résultats ont été les mêmes en ce qui concerne la façon dont le déclin s'est produit. La latence de la LLQ était d'environ 10 us, soit 1:100 de la latence de la carte de ligne Gigabit Ethernet Engine 2 à 3 ports.

[Informations connexes](#)

- [Présentation et configuration de MDRR et WRED sur les routeurs Internet de la gamme Cisco 12000](#)
- [Comment lire la sortie de la commande show controller frfab | Commandes tofab queue sur un routeur Internet de la gamme Cisco 12000](#)
- [Type de service dans la suite de protocoles Internet, priorité \(RFC-1349\)](#)
- [Détection précoce pondérée aléatoire](#)
- [Support et documentation techniques - Cisco Systems](#)