

Présentation de la mise en file d'attente sur les interfaces de routeurs Frame Relay

Table des matières

[Introduction](#)

[Conditions préalables](#)

[Exigences](#)

[Composants utilisés](#)

[Conventions](#)

[Couches de files d'attente](#)

[Mise en file PVC](#)

[Mise En File D'Attente Au Niveau De L'Interface](#)

[Mise en file FIFO](#)

[Double FIFO](#)

[PIPQ](#)

[Réglage de la sonnerie TX](#)

[Informations connexes](#)

Introduction

Ce document passe en revue l'architecture de mise en file d'attente hiérarchique sur les interfaces en série configurées avec l'encapsulation de relayage de trames. Lorsqu'elles sont configurées avec FRTS (Frame Relay Traffic Shape), les interfaces de relayage de trames prennent en charge les couches de file d'attente suivantes :

- File PVC
- File d'attente au niveau interface

Conditions préalables

Exigences

Les lecteurs de ce document doivent être bien informés de :

- [Configuration Frame Relay](#)
- Routeurs des gammes Cisco 2600, 3600 et 7200
- [FRTS](#)

Composants utilisés

Les configurations utilisées dans ce document ont été capturées sur un routeur de la gamme Cisco 7200 avec le matériel et les logiciels suivants :

- Carte de ports T1 multicanal PA-MC-4T1
- Logiciel Cisco IOS® Version 12.2(6)

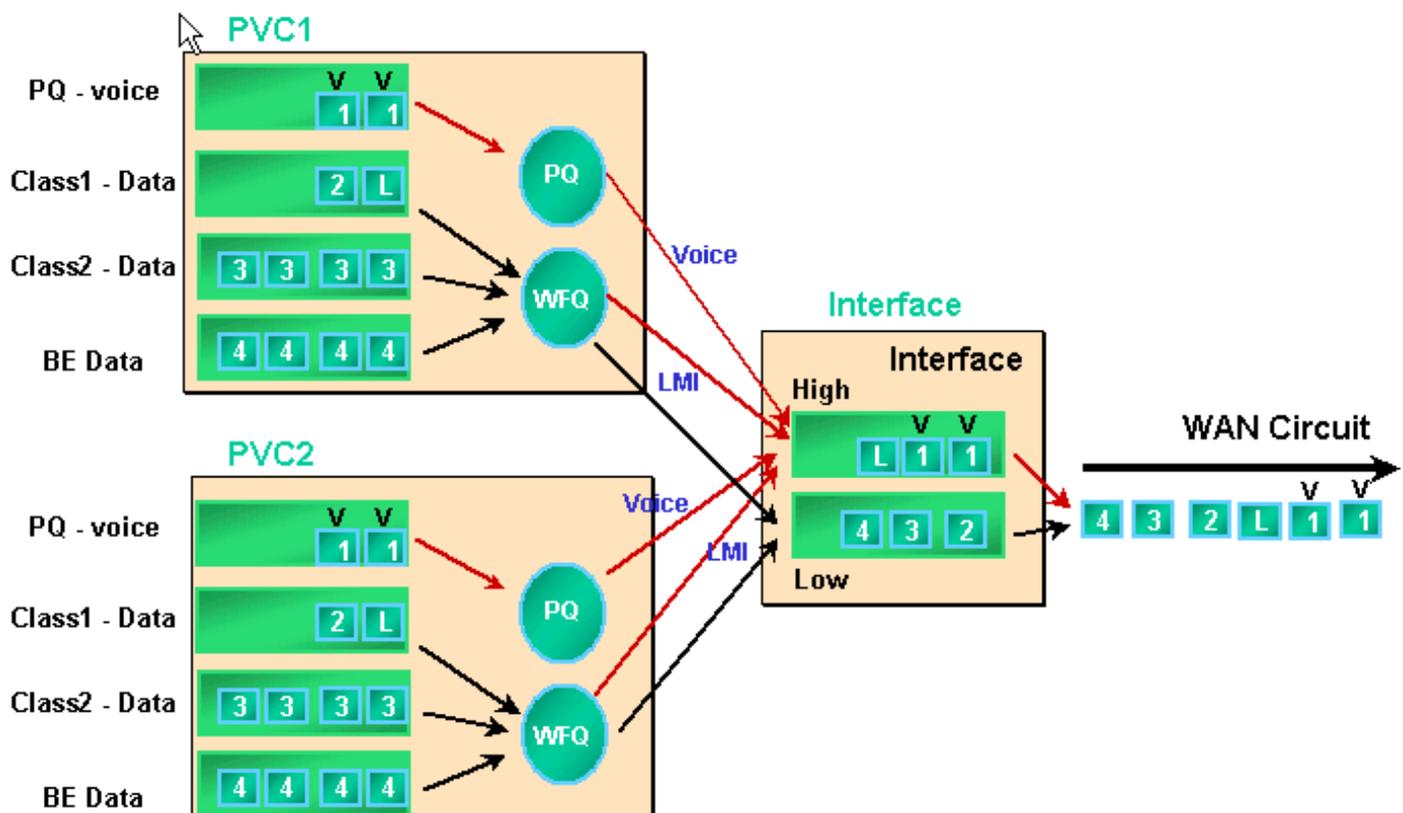
Les informations présentées dans ce document ont été créées à partir de périphériques dans un environnement de laboratoire spécifique. All of the devices used in this document started with a cleared (default) configuration. Si vous travaillez dans un réseau opérationnel, assurez-vous de bien comprendre l'impact potentiel de toute commande avant de l'utiliser.

Conventions

Pour plus d'informations sur les conventions utilisées dans ce document, reportez-vous aux [Conventions relatives aux conseils techniques Cisco](#).

Couches de files d'attente

La figure suivante illustre les deux couches de files d'attente lorsque la fonction FRTS est appliquée à l'interface. L'application des FRTS et des FRF.12 (Frame Relay Forum Implementation Agreements) entraîne le passage de la file d'attente au niveau de l'interface à une file d'attente FIFO double, selon les plates-formes qui prennent en charge cette technique de mise en file d'attente. Les deux files d'attente comprennent une file d'attente à priorité élevée pour transporter la voix sur IP (VoIP) et certains paquets de contrôle et une file d'attente à priorité faible pour transporter tous les autres paquets. Pour plus d'informations sur la mise en file d'attente FIFO double, consultez la section [FIFO double](#).



Les interfaces Frame Relay prennent en charge les files d'attente d'interface ainsi que les files d'attente PVC lorsque les files d'attente FRTS et PVC sont activées. Chaque file d'attente PVC prend également en charge un système WFQ (Weighted Fair Queuing) distinct, si la file d'attente PVC est configurée comme WFQ.

Mise en file PVC

Les interfaces Frame Relay et ATM peuvent prendre en charge plusieurs circuits virtuels (VC). Selon le matériel, ces interfaces prennent en charge les files d'attente de circuits virtuels permanents, ce qui garantit qu'un circuit virtuel encombré n'utilise pas toutes les ressources de mémoire et a un impact sur les autres circuits virtuels (non encombrés).

La commande [frame-relay traffic-shaping](#) active à la fois le formatage du trafic et la mise en file d'attente de circuit virtuel permanent pour tous les circuits virtuels sur une interface Frame Relay. La mise en forme du trafic PVC permet de mieux contrôler le flux de trafic sur un circuit virtuel individuel. Le formatage du trafic combiné à la mise en file d'attente VC limite la consommation de bande passante de l'interface pour un seul VC. Sans mise en forme, un circuit virtuel peut consommer toute la bande passante de l'interface et priver d'autres circuits virtuels.

Si vous ne spécifiez pas de valeurs de mise en forme, les valeurs par défaut pour le débit moyen et la taille de rafale sont appliquées. Lorsque la charge offerte au circuit virtuel dépasse les valeurs de mise en forme, les paquets en excès sont stockés dans la file d'attente de mise en mémoire tampon des paquets du circuit virtuel. Une fois les paquets mis en mémoire tampon, vous pouvez appliquer un mécanisme de mise en file d'attente et contrôler efficacement l'ordre des paquets retirés de la file d'attente du circuit virtuel vers la file d'attente de l'interface. Par

défaut, les files d'attente PVC utilisent la mise en file d'attente « premier arrivé, premier servi » avec une limite de 40 paquets. Utilisez la commande [frame-relay holdq](#) en mode de configuration map-class pour modifier cette valeur. Vous pouvez également appliquer une file d'attente à faible latence (LLQ) ou une file d'attente pondérée basée sur les classes (CBWFQ) à l'aide d'une politique de qualité de service (QoS) configurée avec les commandes de l'interface de ligne de commande (CLI) QoS modulaire (MQC). En outre, vous pouvez appliquer WFQ directement à l'intérieur de la map-class avec la commande [fair queue](#). Cette commande configure votre routeur pour classer le trafic en fonction du flux et place ces flux dans leurs propres sous-files d'attente. Ainsi, la commande fair queue crée un système WFQ par VC.

Les mécanismes de mise en file d'attente détaillés des files PVC sont décrits ci-dessous.

1. Exécutez la commande [show frame-relay pvc](#) 20. L'identificateur de connexion de liaison de données (DLCI) Frame Relay est identifié par le 20. Le résultat suivant ne montre aucune information de mise en file d'attente car FRTS n'est pas activé.

```
<#root>
```

```
Router#
```

```
show frame pvc 20
```

```
PVC Statistics for interface Serial16/0:0 (Frame Relay DTE)
```

```
DLCI = 20, DLCI USAGE = LOCAL, PVC STATUS = DELETED, INTERFACE = Serial16/0:0.1
```

input pkts 0	output pkts 0	in bytes 0
out bytes 0	dropped pkts 0	in FECN pkts 0
in BECN pkts 0	out FECN pkts 0	out BECN pkts 0
in DE pkts 0	out DE pkts 0	out bcast pkts 0
out bcast bytes 0		

```
PVC create time 00:00:38, last time PVC status changed 00:00:25
```

2. Configurez FRTS à l'aide de la commande frame-relay traffic-shaping en mode de configuration d'interface sous l'interface physique. Exécutez à nouveau la commande show frame-relay PVC [dlci].

```
<#root>
```

```
Router#
```

```
show frame-relay pvc 20
```

```
PVC Statistics for interface Serial16/0:0 (Frame Relay DTE)
```

```
DLCI = 20, DLCI USAGE = LOCAL, PVC STATUS = DELETED, INTERFACE = Serial16/0:0.1
```

input pkts 0	output pkts 0	in bytes 0
out bytes 0	dropped pkts 0	in FECN pkts 0

```
in BECN pkts 0          out FECN pkts 0          out BECN pkts 0
in DE pkts 0            out DE pkts 0
out bcast pkts 0        out bcast bytes 0
PVC create time 00:04:59, last time PVC status changed 00:04:46
cir 56000    bc 7000    be 0        byte limit 875    interval 125
```

!--- Shaping parameters.

```
mincir 28000          byte increment 875      Adaptive Shaping none
pkts 0                bytes 0                pkts delayed 0          bytes delayed 0
shaping inactive
traffic shaping drops 0
Queueing strategy: fifo
```

!--- Queue mechanism.

```
Output queue 0/40, 0 drop, 0 dequeued
```

!--- Queue size.

3. Par défaut, les files d'attente PVC utilisent une limite de file d'attente de sortie de 40 paquets. Utilisez la commande `frame-relay holdq` pour configurer une valeur autre que la valeur par défaut.

```
<#root>
```

```
Router(config)#
```

```
map-class frame-relay shaping
```

```
Router(config-map-class)#
```

```
no frame-relay adaptive-shaping
```

```
Router(config-map-class)#
```

```
frame-relay holdq 50
```

```
Router(config)#
```

```
interface serial 6/0:0.1
```

```
Router(config-subif)#
```

```
frame-relay interface-dlci 20
```

```
%PVC is already defined
```

```
Router(config-fr-dlci)#
```

```
class shaping
```

```
Router(config-fr-dlci)#
```

```
end
```

```
Router#
```

```
sh frame PVC 20
```

```
PVC Statistics for interface Serial6/0:0 (Frame Relay DTE)
```

DLCI = 20, DLCI USAGE = LOCAL, PVC STATUS = DELETED, INTERFACE = Serial6/0:0.1

```
input pkts 0          output pkts 0          in bytes 0
out bytes 0          dropped pkts 0         in FECN pkts 0
in BECN pkts 0       out FECN pkts 0       out BECN pkts 0
in DE pkts 0         out DE pkts 0
out bcast pkts 0     out bcast bytes 0
PVC create time 00:11:06, last time PVC status changed 00:10:53
cir 56000   BC 7000   be 0       byte limit 875   interval 125
mincir 28000   byte increment 875   Adaptive Shaping none
pkts 0       bytes 0           pkts delayed 0       bytes delayed 0
shaping inactive
traffic shaping drops 0
Queueing strategy: FIFO
Output queue 0/50, 0 drop, 0 dequeued
```

!--- Queue size.

4. Les files d'attente PVC prennent également en charge [CBWFQ](#) et [LLQ](#), que vous pouvez configurer à l'aide d'une stratégie de service et des commandes du MQC. L'exemple de sortie suivant a été capturé sur le circuit virtuel permanent Frame Relay après l'application d'une stratégie de service QoS.

<#root>

```
Router(config)#
```

```
class-map gold
```

```
Router(config-cmap)#
```

```
match ip dscp 46
```

```
Router(config-cmap)#
```

```
class-map silver
```

```
Router(config-cmap)#
```

```
match ip dscp 26
```

```
Router(config-cmap)#
```

```
policy-map sample
```

```
Router(config-pmap)#
```

```
class gold
```

```
Router(config-pmap-c)#
```

```
priority 64
```

```
Router(config-pmap-c)#
```

```
class silver
```

```
Router(config-pmap-c)#
```

```
bandwidth 32
```

```
Router(config)#
map-class frame-relay map1
Router(config-map-class)#
service-policy output sample
```

```
Router(config-if)#
frame-relay interface-dlci 20
Router(config-fr-dlci)#
class map1
```

```
Router#
show frame-relay PVC 20
```

PVC Statistics for interface Serial6/0:0 (Frame Relay DTE)

DLCI = 20, DLCI USAGE = LOCAL, PVC STATUS = DELETED, INTERFACE = Serial6/0:0.1

```
input pkts 0          output pkts 0          in bytes 0
out bytes 0           dropped pkts 0         in FECN pkts 0
in BECN pkts 0       out FECN pkts 0       out BECN pkts 0
in DE pkts 0         out DE pkts 0
out bcast pkts 0     out bcast bytes 0
PVC create time 00:12:50, last time PVC status changed 00:12:37
```

```
cir 56000    bc 7000    be 0        byte limit 875    interval 125
```

```
mincir 28000    byte increment 875    Adaptive Shaping none
pkts 0        bytes 0    pkts delayed 0    bytes delayed 0
shaping inactive
traffic shaping drops 0
service policy sample
```

Service-policy output: sample

```
Class-map: gold (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 BPS
  Match: ip dscp 46
  Weighted Fair Queueing
  Strict Priority
  Output Queue: Conversation 24
  Bandwidth 64 (kbps) Burst 1600 (Bytes)
  (pkts matched/bytes matched) 0/0
  (total drops/bytes drops) 0/0

Class-map: silver (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 BPS, drop rate 0 BPS
  Match: ip dscp 26
  Weighted Fair Queueing
  Output Queue: Conversation 25
  Bandwidth 32 (kbps) Max Threshold 64 (packets)
!--- Queue information.
```

```
(pkts matched/bytes matched)
(depth/total drops/no-buffer drops) 0/0/0
```

```
Class-map: class-default (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0 BPS, drop rate 0 BPS
  Match: any
  Output queue size 0/max total 600/drops 0
```

!--- Queue size.

À l'origine, la commande `frame-relay holdq <size> map-class` était utilisée pour configurer la taille des files d'attente de mise en forme du trafic FIFO uniquement. La taille maximale était de 512. Dans le logiciel Cisco IOS Version 12.2, et à partir du logiciel IOS Version 12.2(4), cette commande affecte également les tampons maximaux dans les files d'attente de formatage de trafic CBWFQ, comme activé par la commande [service-policy](#) `output map-class`. La taille maximale est désormais de 1024. Les valeurs par défaut, inchangées, sont 40 pour FIFO et 600 pour CBWFQ.

Mise En File D'Attente Au Niveau De L'Interface

Une fois les trames Frame Relay placées dans une file d'attente PVC, elles sont retirées de la file d'attente au niveau de l'interface. Le trafic de tous les circuits virtuels passe par les files d'attente au niveau de l'interface.

Selon les fonctions configurées, la file d'attente au niveau de l'interface Frame Relay utilise l'un des mécanismes suivants.

Fonctionnalité	Mécanisme de mise en file par défaut
FRTS	FIFO
FRF.12	Double FIFO
PIPQ	PIPQ

Remarque : PIPQ (PVC Interface Priority Queueing) remplace FIFO et double FIFO. En d'autres termes, si vous activez FRF.12, la stratégie de mise en file d'attente d'interface reste PIPQ.

Mise en file FIFO

Les étapes suivantes expliquent comment la configuration FRTS modifie le mécanisme de mise en file d'attente appliqué en FIFO.

1. Créez une interface multicanaux fractionnés à l'aide de la commande `channel-group`.

```
<#root>
```

```
Router(config)#
```

```

controller t1 6/0

Router(config-controller)#
channel-group 0 ?

    timeslots  List of timeslots in the channel group

Router(config-controller)#
channel-group 0 timeslots ?

    <1-24> List of timeslots which comprise the channel

Router(config-controller)#
channel-group 0 timeslots 12

```

2. Exécutez la commande `show interface serial 6/0:0` et vérifiez que l'interface T1 utilise la stratégie de mise en file d'attente par défaut « Queueing strategy: Weighted Fair ». Tout d'abord, un paquet est placé dans une file d'attente de fantaisie au niveau du circuit virtuel. Il est ensuite envoyé à la file d'attente d'interface. Dans ce cas, WFQ serait appliqué.

```

<#root>

Router#
show interface serial 6/0:0

Serial6/0:0 is up, line protocol is up (looped)
  Hardware is Multichannel T1
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
    reliability 253/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, Data non-inverted
  Keepalive set (10 sec)
  Last input 00:00:08, output 00:00:08, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops:
  Queueing strategy: weighted fair
  !--- Queue mechanism.

  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
  !--- Queue size.

  Conversations 0/1/16 (active/max active/max total)
  !--- Queue information.

  Reserved Conversations 0/0 (allocated/max allocated)
  !--- Queue information.

  Available Bandwidth 48 kilobits/sec
  !--- Queue information.

  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    5 packets input, 924 bytes, 0 no buffer
    Received 0 broadcasts, 14 runts, 0 giants, 0 throttles
    14 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    17 packets output, 2278 bytes, 0 underruns

```

```
0 output errors, 0 collisions, 0 interface resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions
no alarm present
Timeslot(s) Used:12, subrate: 64Kb/s, transmit delay is 0 flags
!--- Queue information.
```

3. Lorsque la stratégie de mise en file d'attente est WFQ, vous pouvez utiliser les commandes [show queueing](#) et [show queue](#) pour confirmer.

```
<#root>
```

```
Router#
```

```
show queueing interface serial 6/0:0
```

```
Interface Serial6/0:0 queueing strategy: fair
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
    Conversations 0/1/16 (active/max active/max total)
    Reserved Conversations 0/0 (allocated/max allocated)
    Available Bandwidth 48 kilobits/sec
```

```
Router#
```

```
show queue serial 6/0:0
```

```
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: weighted fair
Output queue: 0/1000/64/0 (size/max total/threshold/drops)
  Conversations 0/1/16 (active/max active/max total)
  Reserved Conversations 0/0 (allocated/max allocated)
  Available Bandwidth 48 kilobits/sec
```

4. Appliquez FRTS à l'aide de la commande `frame-relay traffic-shaping` en mode de configuration d'interface.

```
<#root>
```

```
Router(config)#
```

```
interface serial 6/0:0
```

```
Router(config-if)#
```

```
frame-relay traffic-shaping
```

5. L'application de FRTS invite le routeur à changer la stratégie de mise en file d'attente sur les files d'attente au niveau de l'interface en FIFO.

```
<#root>
```

```
Router#
```

```
show interface serial 6/0:0
```

```
Serial6/0:0 is up, line protocol is down (looped)
  Hardware is Multichannel T1
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY, crc 16, Data non-inverted
  Keepalive set (10 sec)
  LMI enq sent 13, LMI stat recvd 0, LMI upd recvd 0, DTE LMI down
  LMI enq recvd 19, LMI stat sent 0, LMI upd sent 0
  LMI DLCI 1023 LMI type is CISCO frame relay DTE
  Broadcast queue 0/64, broadcasts sent/dropped 0/0, interface broadcasts 0
  Last input 00:00:06, output 00:00:06, output hang never
  Last clearing of "show interface" counters 00:02:16
  Queueing strategy: FIFO
```

```
!--- queue mechanism
```

```
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
  19 packets input, 249 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 abort
  19 packets output, 249 bytes, 0 underruns
  0 output errors, 0 collisions, 0 interface resets
  0 output buffer failures, 0 output buffers swapped out
  0 carrier transitions
no alarm present
Timeslot(s) Used:12, subrate: 64Kb/s, transmit delay is 0 flags
```

6. Comme la stratégie de mise en file d'attente est maintenant FIFO, le résultat des commandes show queue et show queueing change.

```
<#root>
```

```
Router#
```

```
show queueing interface serial 6/0:0
```

```
Interface Serial6/0:0 queueing strategy: none
Router#
```

```
Router#
```

```
show queue serial 6/0:0
```

```
'Show queue' not supported with FIFO queueing.
```

La version 12.2(4)T du logiciel Cisco IOS introduit la fonctionnalité [Adaptive Frame Relay Traffic Shaping for Interface Congestion](#), qui est conçue pour minimiser les effets du retard et des pertes

de paquets provoqués par l'encombrement de l'interface. La fonction Adaptive Frame Relay Traffic Shaping for Interface Congestion permet de s'assurer que les paquets sont abandonnés dans les files d'attente de circuits virtuels.

Lorsque cette nouvelle fonctionnalité est activée, le mécanisme de mise en forme du trafic surveille l'encombrement de l'interface. Lorsque le niveau d'encombrement dépasse une valeur configurée appelée profondeur de file d'attente, le débit d'envoi de tous les circuits virtuels permanents est réduit au débit minimal garanti (minCIR). Dès que l'encombrement de l'interface passe en dessous de la profondeur de la file d'attente, le mécanisme de mise en forme du trafic redéfinit le débit d'envoi des circuits virtuels permanents sur le débit d'informations garanti (CIR). Ce processus garantit le minCIR pour les circuits virtuels permanents en cas d'encombrement de l'interface.

Double FIFO

La mise en file d'attente Frame Relay, qui apparaît dans le résultat de la commande `show interface serial` sous la forme Dual FIFO, utilise deux niveaux de priorité. La file d'attente à priorité élevée gère les paquets vocaux et les paquets de contrôle tels que l'interface de gestion locale (LMI). La file d'attente de faible priorité gère les paquets fragmentés (paquets de données ou non vocaux).

Le mécanisme de mise en file d'attente au niveau de l'interface passe automatiquement en mode FIFO double lorsque vous activez l'une des fonctionnalités suivantes :

- FRF.12 Fragmentation — Ceci est activé avec la commande [frame-relay fragment](#) en mode de configuration `map-class`. Les paquets de données supérieurs à la taille de paquet spécifiée dans la commande `frame-relay fragment` sont d'abord mis en file d'attente dans une sous-file d'attente WFQ. Ils sont ensuite retirés de la file d'attente et fragmentés. Après fragmentation, le premier segment est transmis. Les segments restants attendent le prochain temps de transmission disponible pour ce circuit virtuel, tel que déterminé par l'algorithme de mise en forme. À ce stade, les petits paquets vocaux et les paquets de données fragmentés sont entrelacés à partir d'autres circuits virtuels permanents.
- Hiérarchisation RTP (Real-time Transport Protocol) : à l'origine, les petits paquets de données étaient également classés comme appartenant à la file d'attente à priorité élevée simplement en raison de leur taille. La version 12.0(6)T du logiciel Cisco IOS a modifié ce comportement à l'aide de la fonctionnalité de hiérarchisation RTP (VoIPoFR). Il réserve la file d'attente de priorité élevée aux paquets de contrôle LMI et voix uniquement. VoIPoFR classe les paquets VoIP en les faisant correspondre à la plage de ports UDP RTP définie dans une classe de mappage Frame Relay. Tout le trafic RTP dans cette plage de ports est mis en file d'attente prioritaire pour le circuit virtuel. En outre, les paquets vocaux sont placés dans la file d'attente de priorité élevée au niveau de l'interface. Tous les autres paquets entrent dans la file d'attente non prioritaire au niveau de l'interface.

Remarque : cette fonctionnalité suppose que FRF.12 est configuré.

Utilisez la commande `show interface` pour afficher la taille des deux files d'attente. Les étapes ci-dessous montrent les files d'attente FIFO doubles et expliquent comment modifier la taille des files

d'attente.

1. Exécutez la commande show interface serial. La file d'attente de priorité élevée utilise une limite de file d'attente qui est le double de la limite de file d'attente de priorité faible.

```
<#root>
```

```
Router#
```

```
show interface serial 6/0:0
```

```
Serial6/0:0 is up, line protocol is down
  Hardware is Multichannel T1
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY, crc 16, Data non-inverted
  Keepalive set (10 sec)
  LMI enq sent 236, LMI stat recvd 0, LMI upd recvd 0, DTE LMI down
  LMI enq recvd 353, LMI stat sent 0, LMI upd sent 0
  LMI DLCI 1023 LMI type is CISCO frame relay DTE
  Broadcast queue 0/64, broadcasts sent/dropped 0/0, interface broadcasts 0
  Last input 00:00:02, output 00:00:02, output hang never
  Last clearing of "show interface" counters 00:39:22
  Queueing strategy: dual FIFO!
```

```
--- Queue mechanism.
```

```
Output queue: high size/max/dropped 0/256/0
```

```
!--- High-priority queue.
```

```
Output queue 0/128, 0 drops; input queue 0/75, 0 drops
```

```
!--- Low-priority queue.
```

```
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
  353 packets input, 4628 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 abort
  353 packets output, 4628 bytes, 0 underruns
  0 output errors, 0 collisions, 0 interface resets
  0 output buffer failures, 0 output buffers swapped out
  0 carrier transitions
no alarm present
Timeslot(s) Used:12, subrate: 64Kb/s, transmit delay is 0 flags
```

2. Utilisez la commande hold-queue {value} out pour modifier les tailles de file d'attente d'interface.

```
<#root>
```

```
Router(config)#
```

```
interface serial 6/0:0
```

```
Router(config-if)#
```

```
hold-queue ?
```

```
<0-4096> Queue length
```

```
Router(config-if)#
```

```
hold-queue 30 ?
```

```
in Input queue  
out Output queue
```

```
Router(config-if)#
```

```
hold-queue 30 out
```

3. Exécutez à nouveau la commande show interface serial et notez la modification des valeurs maximales de la file d'attente de sortie.

```
<#root>
```

```
Router#
```

```
show interface serial 6/0:0
```

```
Serial6/0:0 is up, line protocol is up
```

```
Hardware is Multichannel T1
```

```
MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
```

```
reliability 255/255, txload 1/255, rxload 1/255
```

```
Encapsulation FRAME-RELAY, crc 16, Data non-inverted
```

```
Keepalive set (10 sec)
```

```
LMI enq sent 249, LMI stat recvd 0, LMI upd recvd 0, DTE LMI down
```

```
LMI enq recvd 372, LMI stat sent 0, LMI upd sent 0
```

```
LMI DLCI 1023 LMI type is CISCO frame relay DTE
```

```
Broadcast queue 0/64, broadcasts sent/dropped 0/0, interface broadcasts 0
```

```
Last input 00:00:02, output 00:00:02, output hang never
```

```
Last clearing of "show interface" counters 00:41:32
```

```
Queueing strategy: dual FIFO
```

```
!--- Queue mechanism.
```

```
Output queue: high size/max/dropped 0/60/0
```

```
!--- High-priority queue.
```

```
Output queue 0/30, 0 drops; input queue 0/75, 0 drops
```

```
!--- Low-priority queue.
```

```
5 minute input rate 0 bits/sec, 0 packets/sec
```

```
5 minute output rate 0 bits/sec, 0 packets/sec
```

```
372 packets input, 4877 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 abort
```

```
372 packets output, 4877 bytes, 0 underruns
```

```
0 output errors, 0 collisions, 0 interface resets
```

```
0 output buffer failures, 0 output buffers swapped out
```

```
0 carrier transitions
```

```
no alarm present
```

```
Timeslot(s) Used:12, subrate: 64Kb/s, transmit delay is 0 flags
```

[Le protocole PIPQ Frame Relay](#) est conçu pour les configurations dans lesquelles des circuits virtuels distincts transportent un seul type de trafic, tel que la voix ou les données. Cela vous permet d'attribuer une valeur de priorité à chaque circuit virtuel permanent. PIPQ minimise le délai de sérialisation ou de mise en file d'attente au niveau de l'interface en s'assurant que le circuit virtuel de priorité élevée est traité en premier. PIPQ classe les paquets en extrayant le DLCI et en recherchant la priorité dans la structure PVC appropriée. Le mécanisme PIPQ n'examine pas le contenu du paquet. Par conséquent, il ne prend aucune décision en fonction du contenu du paquet.

Utilisez les commandes suivantes pour configurer PIPQ.

1. Activez PIPQ avec la commande `frame-relay interface-queue priority` sur l'interface principale.

```
<#root>
Router(config)#
interface serial 6/0:0
Router(config-if)#
frame-relay interface-queue priority
Router(config-if)#
end
```

2. Utilisez la commande `show interface serial` pour confirmer « Queueing strategy: DLCI priority ». Cette commande affiche également la taille actuelle et le nombre de abandons pour chaque file d'attente.

```
<#root>
Router#
show interface serial 6/0:0

Serial6/0:0 is up, line protocol is up
  Hardware is Multichannel T1
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY, crc 16, Data non-inverted
  Keepalive set (10 sec)
  LMI enq sent 119, LMI stat recvd 0, LMI upd recvd 0, DTE LMI down
  LMI enq recvd 179, LMI stat sent 0, LMI upd sent 0
  LMI DLCI 1023 LMI type is CISCO frame relay DTE
  Broadcast queue 0/64, broadcasts sent/dropped 0/0, interface broadcasts 0
  Last input 00:00:06, output 00:00:06, output hang never
  Last clearing of "show interface" counters 00:19:56
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: DLCI priority
  !--- Queue mechanism.
```

```
Output queue (queue priority: size/max/drops):
  high: 0/20/0, medium: 0/40/0, normal: 0/60/0, low: 0/80/0
!--- Queue size.
```

```
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
 179 packets input, 2347 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 abort
 179 packets output, 2347 bytes, 0 underruns
 0 output errors, 0 collisions, 0 interface resets
 0 output buffer failures, 0 output buffers swapped out
 0 carrier transitions
no alarm present
Timeslot(s) Used:12, subrate: 64Kb/s, transmit delay is 0 flags
```

3. Créez une classe de mappage Frame Relay et attribuez un niveau de priorité à un circuit virtuel à l'aide de la commande [frame-relay interface-queue priority {high|medium|normal|low}](#). La priorité PVC par défaut est normale. Tous les circuits virtuels permanents ayant la même priorité partagent la même file d'attente de priorité FIFO. Appliquez la map-class au circuit virtuel. Dans l'exemple de sortie suivant, un circuit virtuel permanent avec le numéro DLCI 21 est attribué à la file d'attente d'interface de priorité élevée.

```
<#root>

Router(config)#
map-class frame-relay high_priority_class

Router(config-map-class)#
frame-relay interface-queue priority high

Router(config-map-class)#
exit

Router(config)#
interface serial 6/0:0.2 point

Router(config-subif)#
frame-relay interface-dlci 21

Router(config-fr-dlci)#
class ?

WORD map class name

Router(config-fr-dlci)#
class high_priority_class
```

4. Utilisez les commandes show frame-relay PVC [dlci] et show queueing interface pour confirmer votre modification de configuration.

```
<#root>
```

```
Router#
```

```
show frame-relay PVC 21
```

```
PVC Statistics for interface Serial6/0:0 (Frame Relay DTE)
```

```
DLCI = 21, DLCI USAGE = LOCAL, PVC STATUS = INACTIVE, INTERFACE = Serial6/0:0.2
```

```
input pkts 0          output pkts 0          in bytes 0
out bytes 0           dropped pkts 0         in FECN pkts 0
in BECN pkts 0       out FECN pkts 0       out BECN pkts 0
in DE pkts 0         out DE pkts 0
out bcast pkts 0     out bcast bytes 0
PVC create time 00:00:17, last time PVC status changed 00:00:17
cir 56000    BC 7000    be 0    byte limit 875    interval 125
mincir 28000    byte increment 875    Adaptive Shaping none
pkts 0    bytes 0    pkts delayed 0    bytes delayed 0
shaping inactive
traffic shaping drops 0
Queueing strategy: FIFO
Output queue 0/40, 0 drop, 0 dequeued
```

```
!--- Size of the PVC queue.
```

```
priority high
```

```
!--- All frames from this PVC are dequeued to the high-priority queue
```

```
!--- at the interface.
```

```
Router#
```

```
show queueing interface serial 6/0:0
```

```
Interface Serial6/0:0 queueing strategy: priority
```

```
Output queue utilization (queue/count)
  high/13 medium/0 normal/162 low/0
```

5. Configurez éventuellement la taille de chaque file d'attente d'interface à l'aide de la commande suivante. Les tailles par défaut des files d'attente à priorité élevée, moyenne, normale et faible sont respectivement de 20, 40, 60 et 80 paquets. Pour configurer une valeur différente, utilisez la commande frame-relay interface-queue priority [<high limit><medium limit><normal limit><low limit>] en mode de configuration d'interface.

Une fois activée, la fonction PIPQ remplace tout autre mécanisme de mise en file d'attente d'interface Frame Relay, y compris la double FIFO. Si vous activez ensuite FRF.12 ou FRTS, le mécanisme de mise en file d'attente au niveau de l'interface ne reviendra pas à la

fonction FIFO double. En outre, PIPQ ne peut pas être activé si un mécanisme de mise en file d'attente de fantaisie autre que par défaut est déjà configuré sur l'interface. Il peut être activé en présence de WFQ si WFQ est la méthode de mise en file d'attente d'interface par défaut. Si FRF.12 est activé, la suppression de la configuration PIPQ modifie la mise en file d'attente au niveau de l'interface en mode FIFO par défaut ou double.

PIPQ applique une mise en file d'attente avec priorité stricte. Si le trafic est continuellement retiré de la file d'attente de priorité haute, le planificateur de mise en file d'attente planifiera la file d'attente de priorité haute et peut effectivement priver les files d'attente de priorité inférieure. Par conséquent, veillez à attribuer les circuits virtuels permanents à la file d'attente à priorité élevée.

Réglage de la sonnerie TX

L'anneau TX est la mémoire tampon FIFO non prioritaire utilisée pour stocker les trames avant leur transmission. Les interfaces Frame Relay utilisent un seul anneau TX partagé par tous les circuits virtuels. Par défaut, la taille de l'anneau TX est de 64 paquets pour les interfaces WAN série à haut débit, y compris PA-T3+, PA-MC-2T3+ et PA-H. Les cartes de ports WAN à faible débit réduisent désormais automatiquement l'anneau TX à une valeur de 2 paquets. En d'autres termes, les pilotes d'interface définissent des valeurs de sonnerie TX par défaut uniques en fonction de la quantité de bande passante.

File d'attente	Emplacement	Méthodes de mise en file d'attente	Les routages spécifiques de service s'appliquent	Commande pour ajuster
File d'attente matérielle ou anneau de transmission par interface	Adaptateur de port ou module réseau	FIFO uniquement.	Non	tx-ring-limit
File d'attente de couche 3 par circuit virtuel	Système de traitement de la couche 3 ou tampons d'interface	FIFO, WFQ, CBWFQ ou LLQ	Oui	Varie avec la méthode de mise en file d'attente: <ul style="list-style-type: none"> • frame-relay holdq avec FIFO • queue-

				limit avec CBWFQ
--	--	--	--	--

Remarque : contrairement aux interfaces ATM telles que PA-A3, les interfaces Frame Relay utilisent un seul anneau de transmission pour l'interface. Ils ne construisent pas un anneau distinct pour chaque circuit virtuel.

Il est important de savoir que l'anneau TX est FIFO et ne peut pas prendre en charge un autre mécanisme de mise en file d'attente. Par conséquent, si vous réduisez l'anneau TX à une valeur de 2 sur les interfaces à faible débit, la majeure partie de la mise en mémoire tampon des paquets est déplacée vers la file d'attente PVC, où les mécanismes de mise en file d'attente sophistiqués et les politiques de service QoS s'appliquent.

Le tableau suivant répertorie les cartes de ports série de la gamme 7x00 pour le réglage automatique de l'anneau de transmission.

Référence de la carte de ports	TX Ring Limit Réglage automatique
Cartes de ports série haut débit	
PA-H et PA-2H	Oui
PA-E3 et PA-T3	Oui
PA-T3+	Oui
Cartes de ports série multicanaux	
PA-MC-2T3+	Oui
PA-MC-2T1(=), PA-MC-4T1(=), PA-MC-8T1(=), PA-MC-8DSX1(=)	Oui
PA-MC-2E1/120(=), PA-MC- 8E1/120(=)	Oui
PA-MC-T3, PA-MC-E3	Oui
PA-MC-8TE1+	Oui
PA-STM1	Oui
Cartes de ports série	
PA-4T, PA-4T+	Oui
PA-4E1G	Oui
PA-8T-V35, PA-8T-X21, PA-8T- 232	Oui

La taille de la sonnerie de transmission est automatiquement désactivée lorsqu'une fonction

d'optimisation vocale est activée. En outre, l'application de la fonction PIPQ entraîne la désactivation automatique de l'anneau de transmission.

Le résultat suivant a été capturé sur un routeur de la gamme 7200 exécutant le logiciel Cisco IOS version 12.2(6).

```
<#root>
```

```
7200-16#
```

```
show controller serial 6/0:0
```

```
Interface Serial6/0:0
```

```
f/w rev 1.2.3, h/w rev 163, PMC freedm rev 1 idb = 0x6382B984  
ds = 0x62F87C18, plx_devbase=0x3F020000, pmc_devbase=0x3F000000  
Enabled=TRUE, DSX1 linestate=0x0,  
Ds>tx_limited:1 Ds>tx_count:0 Ds>max_tx_count:20
```

```
alarm present
```

```
Timeslot(s) Used:1-24, substrate: 64Kb/s, transmit delay is 0 flags
```

```
Download delay = 0, Report delay = 0
```

```
IDB type=0xC, status=0x84208080
```

```
Pci shared memory = 0x4B16B200
```

```
Plx mailbox addr = 0x3F020040
```

```
RxFree queue=0x4B2FA280, shadow=0x62F9FA70
```

```
Rx freeq_wt=256, freeq_rd=256, ready_wt=1, ready_rd=0
```

```
TxFree queue=0x4B2FAAC0, shadow=0x62F8FA44
```

```
TX freeq_wt=4099, freeq_rd=4099, ready_wt=4, ready_rd=3
```

```
# of TxFree queue=4095
```

```
Freedm FIFO (0x6292BF64), hp=0x6292C034 indx=26, tp=0x6292CF5C indx=511
```

```
reset_count=0 resurrect_count=0
```

```
TX enqueued=0, throttled=0, unthrottled=0, started=10
```

```
tx_limited=TRUE tx_queue_limit=2
```

!--- Note "tx_limited=TRUE" when PIPQ is enabled. The "tx_queue_limit" value

!--- describes the value of the transmit ring.

```
7200-16(config)#
```

```
interface serial 6/0:0
```

```
7200-16(config-if)#
```

```
no frame-relay interface-queue priority
```

```
7200-16(config-if)#
```

```
end
```

```
7200-16#
```

```
show controller serial 6/0:0
```

```
Interface Serial6/0:0
```

```
f/w rev 1.2.3, h/w rev 163, PMC freedm rev 1 idb = 0x6382B984  
Ds = 0x62F87C18, plx_devbase=0x3F020000, pmc_devbase=0x3F000000  
Enabled=TRUE, DSX1 linestate=0x0,  
Ds>tx_limited:0 Ds>tx_count:0 Ds>max_tx_count:20  
alarm present
```

Timeslot(s) Used:1-24, subrate: 64Kb/s, transmit delay is 0 flags
Download delay = 0, Report delay = 0
IDB type=0xC, status=0x84208080
Pci shared memory = 0x4B16B200
Plx mailbox addr = 0x3F020040
RxFree queue=0x4B2FA280, shadow=0x62F9FA70
Rx freeq_wt=256, freeq_rd=256, ready_wt=1, ready_rd=0
TxFree queue=0x4B2FAAC0, shadow=0x62F8FA44
TX freeq_wt=4099, freeq_rd=4099, ready_wt=4, ready_rd=3
of TxFree queue=4095
Freedm FIFO (0x6292BF64), hp=0x6292C034 indx=26, tp=0x6292CF5C indx=511
reset_count=0 resurrect_count=0
TX enqueued=0, throttled=0, unthrottled=0, started=11
tx_limited=FALSE

!--- Transmit ring value has changed.

Informations connexes

- [Configuration CBWFQ sur circuits virtuels permanent \(PVC\) de relais de trame](#)
- [Mise en file d'attente à faible latence pour Frame Relay](#)
- [Mise en file d'attente prioritaire des interfaces PVC Frame Relay](#)
- [Configuration du formatage de trafic de relais de trame sur les routeurs 7200 et plates-formes inférieures](#)
- [Mise en forme du trafic de relais de trame avec QoS distribué sur Cisco 7500](#)
- [Configuration du marquage de paquet sur PVC de relais de trame](#)
- [Mise en file d'attente à faible latence pour Frame Relay](#)
- [Pages de support Frame Relay](#)
- [Pages de support QoS](#)
- [Support technique - Cisco Systems](#)

À propos de cette traduction

Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.