

# Informazioni sull'accodamento sulle interfacce router Frame Relay

## Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Convenzioni](#)

[Livelli di coda](#)

[Accodamento PVC](#)

[Accodamento a livello di interfaccia](#)

[Accodamento FIFO](#)

[FIFO doppio](#)

[PIPQ](#)

[Sintonizzazione dell'anello TX](#)

[Informazioni correlate](#)

## Introduzione

In questo documento viene esaminata l'architettura di coda gerarchica sulle interfacce seriali configurate con l'incapsulamento Frame Relay. Se configurate con Frame Relay Traffic Shaping (FRTS), le interfacce Frame Relay supportano i seguenti livelli di coda:

- coda PVC
- Coda a livello di interfaccia

## Prerequisiti

### Requisiti

I lettori di questo documento devono essere a conoscenza di:

- [Configurazione Frame Relay](#)
- Cisco serie 2600, 3600 e 7200 router
- [FRTS](#)

### Componenti usati

Le configurazioni utilizzate in questo documento sono state acquisite su un router Cisco serie 7200 con i seguenti hardware e software:

- PA-MC-4T1 Multichannel T1 Port Adapter
- Cisco IOS® Software Release 12.2(6)

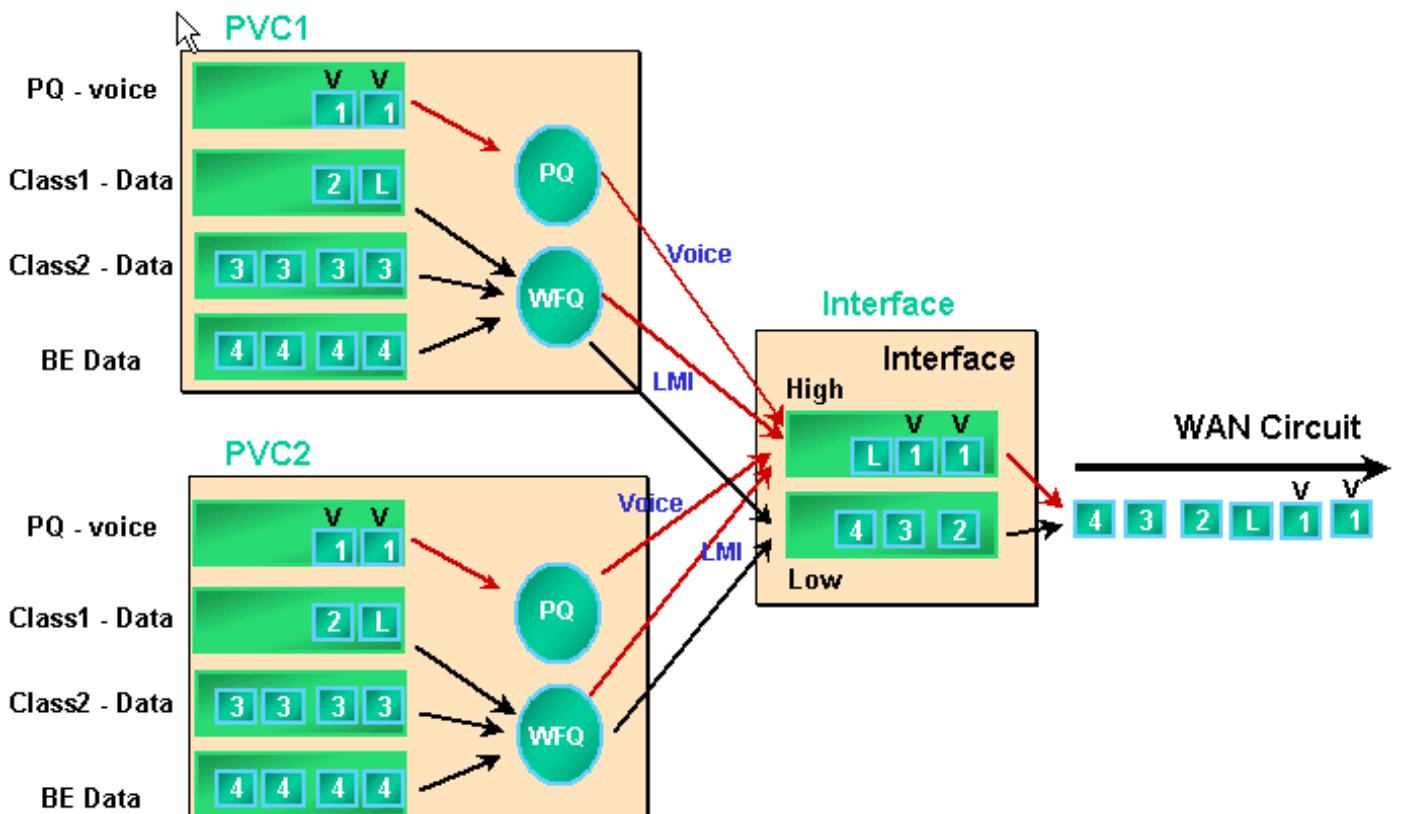
Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

### Convenzioni

Per ulteriori informazioni sulle convenzioni usate, consultare il documento [Cisco sulle convenzioni nei suggerimenti tecnici](#).

### Livelli di coda

La figura seguente illustra i due livelli di coda quando FRTS viene applicato all'interfaccia. L'applicazione di FRTS e Frame Relay Forum Implementation Agreements (FRF.12) determina il passaggio della coda a livello di interfaccia a una doppia coda FIFO, a seconda delle piattaforme che supportano questa tecnica di coda. Le due code includono una coda ad alta priorità per il trasporto del protocollo VoIP (Voice over IP) e di alcuni pacchetti di controllo, e una coda a bassa priorità per il trasporto di tutti gli altri pacchetti. Per ulteriori informazioni sull'accodamento FIFO doppio, vedere la sezione [FIFO doppio](#).



Quando sono abilitate le code FRTS e PVC, le interfacce Frame Relay supportano le code di interfaccia e le code PVC. Ogni coda PVC supporta inoltre un sistema WFQ (Weighted Fair Queuing) distinto, se la coda PVC è configurata come WFQ.

## Accodamento PVC

Entrambe le interfacce Frame Relay e ATM possono supportare circuiti virtuali multipli (VC). A seconda dell'hardware, queste interfacce supportano le code PVC, che assicurano che un VC congestionato non consumi tutte le risorse di memoria e abbia un impatto sugli altri VC (non congestionati).

Il comando [frame-relay traffic-shaping](#) abilita sia il traffic shaping sia la coda PVC per tutti i VC su un'interfaccia Frame Relay. Il traffic shaping PVC consente un maggiore controllo del flusso del traffico su un singolo VC. Il traffic shaping, combinato con le code VC, limita il consumo della larghezza di banda dell'interfaccia per un singolo VC. Senza alcuna forma, un VC può consumare tutta la larghezza di banda dell'interfaccia e affamare altri VC.

Se non si specificano valori di shaping, vengono applicati i valori predefiniti per la velocità media e le dimensioni della frammentazione. Quando il carico offerto al VC supera i valori di shaping, i pacchetti in eccesso vengono memorizzati nella coda di buffering dei pacchetti del VC. Una volta memorizzato il buffer dei pacchetti, è possibile applicare un meccanismo di coda e controllare in modo efficace l'ordine dei pacchetti rimossi dalla coda VC alla coda dell'interfaccia. Per impostazione predefinita, le code PVC utilizzano la coda First Come, prima servita con un limite di 40 pacchetti. Per modificare questo valore, usare il comando [frame-relay holdq](#) in modalità di configurazione map-class. In alternativa, è possibile applicare LLQ (Low Latency Queueing) o CBWFQ (Weighted Fair Queueing) basato su classi utilizzando un criterio QoS (Quality of Service) configurato con i comandi dell'interfaccia della riga di comando (CLI) (MQC) QoS modulare. Inoltre, è possibile applicare WFQ direttamente all'interno della classe map con il comando [fair queue](#). Con questo comando il router viene configurato in modo da classificare il traffico in base al flusso e i flussi vengono inseriti nelle rispettive code secondarie. Di conseguenza, il comando fair queue crea un sistema WFQ per ciascuna VC.

Di seguito sono descritti i meccanismi di coda dettagliati per le code PVC.

1. Eseguire il comando [show frame-relay pvc 20](#). Il valore DLCI (Frame Relay Data Link Connection Identifier) è identificato dal valore 20. Nell'output seguente non vengono visualizzate informazioni sulla coda perché FRTS non è abilitato.

```
<#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. Configurare FRTS utilizzando il comando frame-relay traffic-shaping in modalità di configurazione interfaccia nell'interfaccia fisica. Eseguire di nuovo il comando show frame-relay PVC [dlci].

```
<#root>
```

```
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   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. Per impostazione predefinita, le code PVC utilizzano un limite di 40 pacchetti per le code di output. Usare il comando frame-relay holdq per configurare un valore non predefinito.

```
<#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. Le code PVC supportano anche [CBWFQ](#) e [LLQ](#), che è possibile configurare utilizzando una policy per i servizi e i comandi di MQC. L'output di esempio seguente è stato acquisito sul PVC Frame Relay dopo l'applicazione di un criterio del servizio 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.*

In origine, il comando `frame-relay holdq <size>map-class` è stato usato per configurare le dimensioni delle sole code FIFO di traffic shaping. La dimensione massima era 512. Nel software Cisco IOS versione 12.2 e nella versione 12.2(4), questo comando influisce anche sul numero massimo di buffer nelle code CBWFQ per il traffic shaping, in quanto abilitato dal comando [service-policy](#) output map-class. La dimensione massima è ora 1024. I valori predefiniti, che rimangono invariati, sono 40 per FIFO e 600 per CBWFQ.

## Accodamento a livello di interfaccia

Dopo essere stati accodati in una coda PVC, i frame Frame Relay vengono rimossi dalle code a livello di interfaccia. Il traffico proveniente da tutti i VC passa attraverso le code a livello di interfaccia.

A seconda delle funzionalità configurate, la coda a livello di interfaccia di Frame Relay utilizza uno dei meccanismi seguenti.

Funzionalità	Meccanismo di accodamento predefinito
--------------	---------------------------------------

FRTS	FIFO
FRF.12	FIFO doppio
PIPQ	PIPQ

Nota: PIPQ (PVC Interface Priority Queueing) esegue l'override di FIFO e FIFO doppio. In altre parole, se si abilita FRF.12, la strategia di accodamento dell'interfaccia rimarrà PIPQ.

## Accodamento FIFO

Nei passaggi seguenti viene illustrato come la configurazione FRTS alteri il meccanismo di coda applicato a FIFO.

1. Creare un'interfaccia canalizzata utilizzando il comando 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. Eseguire il comando show interface serial 6/0:0 e confermare che l'interfaccia T1 utilizza la strategia di coda predefinita "Weighting Strategy: Weighted Fair". In primo luogo, un pacchetto viene accodato a una coda elaborata a livello di VC. e quindi inviato alla coda di interfaccia. In questo caso, verrebbe applicato WFQ.

```
<#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. Quando la strategia di accodamento è WFQ, è possibile utilizzare i comandi [show queueing](#) e [show queue](#) per confermare.

```
<#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. Applicare FRTS utilizzando il comando frame-relay traffic-shaping nella modalità di configurazione interfaccia.

```
<#root>
Router(config)#
interface serial 6/0:0
Router(config-if)#
frame-relay traffic-shaping
```

5. Quando si applica il comando FRTS, il router deve modificare la strategia di coda sulle code a livello di interfaccia in 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. Poiché la strategia di accodamento è ora FIFO, l'output del comando show queue e show queueing cambia.

```
<#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.
```

Il software Cisco IOS versione 12.2(4)T introduce la funzione [Adaptive Frame Relay Traffic Shaping for Interface Congestion](#), progettata per ridurre al minimo gli effetti del ritardo e delle perdite di pacchetti causati dalla congestione dell'interfaccia. La funzione Adaptive Frame Relay Traffic Shaping for Interface Congestion aiuta a garantire che il pacchetto venga scartato alle code VC.

Quando questa nuova funzionalità è abilitata, il meccanismo di traffic shaping controlla la congestione dell'interfaccia. Quando il livello di congestione supera un valore configurato denominato profondità di coda, la velocità di invio di tutti i PVC viene ridotta alla velocità minima di commit delle informazioni (minCIR). Quando la congestione dell'interfaccia scende al di sotto della profondità della coda, il meccanismo di traffic shaping riporta la velocità di invio dei PVC al CIR (Committed Information Rate). Questo processo garantisce il funzionamento di minCIR per i PVC in caso di congestione dell'interfaccia.

## FIFO doppio

Frame Relay Queueing, che appare nell'output del comando show interface serial come Dual FIFO, usa due livelli di priorità. La coda ad alta priorità gestisce pacchetti voce e pacchetti di controllo, ad esempio LMI (Local Management Interface). La coda a bassa priorità gestisce pacchetti frammentati (pacchetti dati o non voce).

Il meccanismo di coda a livello di interfaccia passa automaticamente a FIFO doppio quando si attiva una delle seguenti funzioni:

- FRF.12 Fragmentation: questo comando viene abilitato con il comando [frame-relay fragment](#) in modalità di configurazione map-class. I pacchetti di dati di dimensioni superiori a quelle specificate nel comando frame-relay fragment vengono prima accodati in una coda secondaria WFQ. Vengono quindi rimossi dalla coda e frammentati. Dopo la frammentazione, viene trasmesso il primo segmento. I segmenti rimanenti attendono il successivo tempo di trasmissione disponibile per la videoconferenza, come determinato dall'algoritmo di shaping. A questo punto, piccoli pacchetti voce e pacchetti dati frammentati

vengono interlacciati da altri PVC.

- Assegnazione di priorità al protocollo RTP (Real-time Transport Protocol): in origine, anche i pacchetti di dati di piccole dimensioni venivano classificati come appartenenti alla coda ad alta priorità per le loro dimensioni. Il software Cisco IOS versione 12.0(6)T ha modificato questo comportamento utilizzando la funzione RTP Prioritization (VoIPoFR). La coda ad alta priorità viene riservata solo ai pacchetti di controllo voce e LMI. VoIPoFR classifica i pacchetti VoIP in base all'intervallo di porte UDP RTP definito in una classe mappa Frame Relay. Tutto il traffico RTP all'interno di questo intervallo di porte viene accodato a una coda di priorità per la VC. Inoltre, i pacchetti voce entrano nella coda ad alta priorità a livello di interfaccia. Tutti gli altri pacchetti vengono inseriti nella coda non prioritaria a livello di interfaccia.

Nota: questa funzionalità presuppone che FRF.12 sia configurato.

Utilizzare il comando `show interface` per visualizzare le dimensioni delle due code. I passaggi seguenti mostrano le code FIFO doppie e descrivono come modificare le dimensioni delle code.

1. Eseguire il comando `show interface serial`. La coda con priorità alta utilizza un limite di coda pari al doppio delle dimensioni del limite della coda con priorità bassa.

```
<#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. Utilizzare il comando hold-queue {value} out per modificare le dimensioni della coda di interfaccia.

```
<#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. Eseguire nuovamente il comando show interface serial e notare come i valori massimi della "coda di output" siano stati modificati.

```
<#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
```

## PIPQ

[Il protocollo PIPQ Frame Relay](#) è progettato per configurazioni in cui sistemi VC separati trasportano un singolo tipo di traffico, ad esempio voce o dati. In questo modo è possibile assegnare una priorità a ciascun PVC. PIPQ riduce al minimo la serializzazione o il ritardo di accodamento a livello di interfaccia, garantendo che la videoconferenza ad alta priorità sia servita per prima. PIPQ classifica i pacchetti estraendo il DLCI e cercando la priorità nella struttura PVC appropriata. Il meccanismo PIPQ non controlla il contenuto del pacchetto. Pertanto, non prende decisioni in base al contenuto dei pacchetti.

Utilizzare i seguenti comandi per configurare PIPQ.

1. Abilitare PIPQ con il comando `frame-relay interface-queue priority` sull'interfaccia principale.

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

2. Utilizzare il comando `show interface serial` per confermare "Strategia di coda: priorità DLCI". Questo comando visualizza anche le dimensioni correnti e il numero di scarti per ciascuna coda.

```
<#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. Compilare una classe mappa Frame Relay e assegnare un livello di priorità a un VC utilizzando il comando [frame-relay interface-queue priority {high|medium|normal|low}](#) . La priorità predefinita del PVC è normale. Tutti i PVC con la stessa priorità condividono la stessa coda di priorità FIFO. Applicare la classe map al VC. Nell'output di esempio seguente, un PVC con numero DLCI 21 viene assegnato alla coda dell'interfaccia ad alta priorità.

<#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. Utilizzare i comandi show frame-relay PVC [dlci] e show queueing interface per confermare la modifica della configurazione.

```
<#root>
```

```
Router#
```

```
show frame 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. Facoltativamente, configurare le dimensioni di ciascuna coda di interfaccia con il comando seguente. Le dimensioni predefinite delle code con priorità alta, media, normale e bassa sono rispettivamente 20, 40, 60 e 80 pacchetti. Per configurare un valore diverso, usare il comando `frame-relay interface-queue priority [<high limit><medium limit><normal limit><low limit>]` in modalità di configurazione interfaccia.

Una volta abilitato, PIPQ ignora qualsiasi altro meccanismo di coda dell'interfaccia Frame Relay, incluso il doppio FIFO. Se successivamente si abilita FRF.12 o FRTS, il meccanismo di coda a livello di interfaccia non tornerà a FIFO doppio. Inoltre, non è possibile abilitare PIPQ se sull'interfaccia è già configurato un meccanismo di coda fantasia non predefinito. Può essere abilitato in presenza di WFQ se WFQ è il metodo di coda dell'interfaccia predefinito. L'eliminazione della configurazione PIPQ comporta la modifica della coda a livello di interfaccia al valore di default o al valore FIFO doppio, se FRF.12 è abilitato.

PIPQ applica l'accodamento a priorità rigida. Se il traffico viene continuamente rimosso dalla coda ad alta priorità, l'utilità di pianificazione della coda pianificherà la coda ad alta priorità e potrebbe effettivamente far morire di fame le code a bassa priorità. Pertanto, prestare attenzione nell'assegnare i PVC alla coda ad alta priorità.

## Sintonizzazione dell'anello TX

L'anello TX è il buffer FIFO senza priorità usato per memorizzare i frame prima della trasmissione. Le interfacce Frame Relay utilizzano un singolo anello TX condiviso da tutti i VC. Per impostazione predefinita, le dimensioni dell'anello TX sono di 64 pacchetti per interfacce WAN seriali a velocità più elevata, inclusi PA-T3+, PA-MC-2T3+ e PA-H. Gli adattatori di porte WAN a bassa velocità ora sintonizzano automaticamente l'anello TX sul valore di 2 pacchetti. In altre parole, i driver di interfaccia impostano valori univoci degli anelli TX predefiniti in base alla quantità di larghezza di banda.

Coda	Posizione	Metodi di coda	Applicazione dei criteri dei servizi	Comando da ottimizzare
Accodamento hardware o anello di trasmissione per interfaccia	Adattatore porta o modulo di rete	Solo FIFO	No	<a href="#">tx-ring-limit</a>
Coda di layer 3 per VC	Buffer di interfaccia	FIFO, WFQ,	Sì	Varia a seconda del

	o sistema del processore di layer 3	CBWFQ o LLQ		metodo di coda: <ul style="list-style-type: none"> <li>• holdq frame relay con FIFO</li> <li>• <a href="#">queue-limit</a> con CBWFQ</li> </ul>
--	-------------------------------------	-------------	--	---

Nota: a differenza delle interfacce ATM come PA-A3, le interfacce Frame Relay usano un singolo anello di trasmissione per l'interfaccia. Non è necessario creare un anello separato per ogni sistema VC.

È importante sapere che l'anello TX è FIFO e non può supportare un meccanismo di coda alternativo. Pertanto, se si sintonizza l'anello TX su un valore di 2 sulle interfacce a bassa velocità, la maggior parte del buffer dei pacchetti viene spostata sulla coda PVC, dove vengono applicati i sofisticati meccanismi di coda e le policy sui servizi QoS.

Nella tabella seguente sono elencati gli adattatori delle porte seriali per la serie 7x00 per la regolazione automatica dell'anello di trasmissione.

N. parte adattatore porta	Sintonizzazione automatica limite ring TX
Adattatori porte seriali ad alta velocità	
PA-H e PA-2H	Sì
PA-E3 e PA-T3	Sì
PA-T3+	Sì
Adattatori porte seriali multicanale	
PA-MC-2T3+	Sì
PA-MC-2T1(=), PA-MC-4T1(=), PA-MC-8T1(=), PA-MC-8DSX1(=)	Sì
PA-MC-2E1/120(=), PA-MC-8E1/120(=)	Sì
PA-MC-T3, PA-MC-E3	Sì
PA-MC-8TE1+	Sì
PA-STM1	Sì

Adattatori porte seriali	
PA-4T, PA-4T+	Sì
PA-4E1G	Sì
PA-8T-V35, PA-8T-X21, PA-8T-232	Sì

La dimensione della ghiera di trasmissione viene sintonizzata automaticamente quando è abilitata una funzione di ottimizzazione della voce. Inoltre, applicando PIPQ l'anello di trasmissione viene sintonizzato automaticamente.

Il seguente output è stato acquisito su un router serie 7200 con software Cisco IOS versione 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, 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=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.*

## Informazioni correlate

- [Configurazione di CBWFQ sui PVC Frame Relay](#)
- [Accodamento a bassa latenza per Frame Relay](#)
- [Accodamento priorità interfaccia PVC Frame Relay](#)
- [Configurazione di Frame Relay Traffic Shaping su router 7200 e piattaforme inferiori](#)
- [Cisco serie 7500 Frame Relay Traffic Shaping con QoS distribuito](#)
- [Configurazione del contrassegno pacchetti sui PVC Frame Relay](#)
- [Accodamento a bassa latenza per Frame Relay](#)
- [Pagine di supporto Frame Relay](#)
- [Pagine di supporto QoS](#)
- [Supporto tecnico – Cisco Systems](#)

## Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l'accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).