

Pianificazione dell'uscita QoS sugli switch Catalyst serie 6500/6000 con software Cisco IOS

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Convenzioni](#)

[Premesse](#)

[Rilasci coda di output](#)

[Capacità di accodamento dell'output di diverse schede di linea sugli switch Catalyst 6500/6000](#)

[Comprendere la funzionalità di coda di una porta](#)

[Pianificazione della configurazione, del monitoraggio e dell'output di esempio sugli switch Catalyst 6500/6000](#)

[Configurazione](#)

[Monitoraggio della programmazione dell'output e verifica delle configurazioni](#)

[Esempio di programmazione dell'output](#)

[Utilizzare la programmazione dell'output per ridurre il ritardo e l'instabilità](#)

[Riduci ritardo](#)

[Riduci variazione](#)

[Conclusioni](#)

[Informazioni correlate](#)

[Introduzione](#)

L'uso della pianificazione dell'output garantisce che il traffico importante non venga interrotto in caso di sovrassegnazione. In questo documento vengono descritte tutte le tecniche e gli algoritmi utilizzati per la pianificazione dell'output sugli switch Catalyst 6500/6000. Questo documento spiega anche come configurare e verificare il funzionamento della pianificazione dell'output sugli switch Catalyst 6500/6000 con software Cisco IOS®.

Per ulteriori informazioni sul protocollo WRED (Weighted Random Early Detection), WRR (Weighted Round Robin) e tail drop, fare riferimento alla [programmazione dell'output QoS sugli switch Catalyst serie 6500/6000 con software CatOS](#).

[Prerequisiti](#)

[Requisiti](#)

Nessun requisito specifico previsto per questo documento.

Componenti usati

Il documento può essere consultato per tutte le versioni software o hardware.

Convenzioni

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

Premesse

Rilasci coda di output

Le perdite di output sono causate da un'interfaccia congestionata. Una causa comune di questo problema può essere il traffico proveniente da un collegamento con larghezza di banda elevata che viene impostato su un collegamento con larghezza di banda inferiore o il traffico proveniente da più collegamenti in entrata che viene impostato su un singolo collegamento in uscita.

Ad esempio, se si invia una grande quantità di traffico bursty su un'interfaccia Gigabit e si passa a un'interfaccia a 100 Mbps, è possibile che si verifichino cali di output incrementali sull'interfaccia a 100 Mbps. Infatti, la coda di output su quell'interfaccia è sovraccarica dal traffico in eccesso a causa della mancata corrispondenza tra la larghezza di banda in entrata e in uscita. La velocità del traffico sull'interfaccia in uscita non può accettare tutti i pacchetti che devono essere inviati.

Per risolvere il problema, la soluzione migliore è aumentare la velocità della linea. Esistono tuttavia modi per impedire, ridurre o controllare le riduzioni di output quando non si desidera aumentare la velocità della linea. È possibile impedire la perdita di output solo se la perdita di output è una conseguenza di brevi picchi di dati. Se le perdite di output sono causate da un flusso costante ad alta velocità, non è possibile evitarle. Tuttavia, è possibile controllarli.

Capacità di accodamento dell'output di diverse schede di linea sugli switch Catalyst 6500/6000

Se non si è certi della funzionalità di accodamento di una porta, usare il comando **show queueing interface {gigabitethernet | fastethernet} mod/porta**. Di seguito vengono mostrate le prime righe di output di un comando **show queueing interface**. La porta si trova su una scheda di linea Supervisor Engine 1A:

```
cosmos#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy: Weighted Round-Robin
```

```
QoS is disabled globally
Trust state: trust DSCP
Default COS is 0
Transmit group-buffers feature is enabled
Transmit queues [type = 1p2q2t]:
  Queue Id      Scheduling  Num of thresholds
  -----
  1             WRR low    2
  2             WRR high   2
```

!--- Output suppressed.

L'output mostra che questa porta ha un tipo di coda di output noto come 1p2q2t.

Per verificare il tipo di coda disponibile su una porta specifica, usare anche il comando **show interface capabilities**:

```
la-orion#show interface gigabitethernet 6/2 capabilities
GigabitEthernet6/2
  Model: WS-SUP720-BASE
  Type: No GBIC
  Speed: 1000
  Duplex: full
  Trunk encap. type: 802.1Q,ISL
  Trunk mode: on,off,desirable,nonegotiate
  Channel: yes
  Broadcast suppression: percentage(0-100)
  Flowcontrol: rx-(off,on,desired),tx-(off,on,desired)
  Membership: static
  Fast Start: yes
  QoS scheduling: rx-(1p1q4t), tx-(1p2q2t)
  CoS rewrite: yes
  ToS rewrite: yes
  Inline power: no
  SPAN: source/destination
  UDLD: yes
  Link Debounce: yes
  Link Debounce Time: yes
  Ports on ASIC: 1-2
```

[Comprendere la funzionalità di coda di una porta](#)

Sugli switch Catalyst 6500/6000 sono disponibili diversi tipi di code. Nella tabella seguente viene illustrata la notazione dell'architettura QoS della porta:

Trasmissione (Tx)/Ricezione (Rx)	Notazione coda	N. di code	Coda di priorità	N. di code WRR	N. e tipo di soglia per le code WRR
Tx	2q2t	2	—	2	2 drop di coda configurabile
Tx	1p2q2t	3	1	2	2 WRED configurabili
Tx	1p3q1t	4	1	3	1 WRED configurabile
Tx	1p2q1t	3	1	2	1 WRED configurabile
Rx	1q4t	1	—	—	4 drop di coda configurabile
Rx	1p1q4t	2	1	1	4 drop di coda configurabile

Rx	1p1q0t	2	1	1	Non configurabile
Rx	1p1q8t	2	1	1	8 WRED configurabili
Tx	1p3q8t	4	1	3	8 cavi WRED o tail drop configurabili
Tx	1p7q8t	8	1	7	8 cavi WRED o tail drop configurabili
Rx	1q2t	1	—	—	1 drop configurabile = 1 non configurabile
Rx	1q8t	1	—	—	8 tail drop configurabile
Rx	2q8t	2	—	2	8 tail drop configurabile

Nella tabella seguente vengono elencati alcuni dei moduli e dei tipi di coda nei lati Rx e Tx dell'interfaccia o della porta. Se il modulo non è elencato qui, utilizzare il comando **show interface capabilities** per determinare la funzionalità della coda disponibile. Il comando **show interface capabilities** è descritto nella sezione [Output Queueing Capability of Different Line Card](#) sugli switch Catalyst 6500/6000.

Modulo	Code Rx	Code Tx
WS-X6K-S2-PFC2	1p1q4t	1p2q2t
WS-X6K-SUP1A-2GE	1p1q4t	1p2q2t
WS-X6K-SUP1-2GE	1q4t	2q2t
WS-X6501-10GEX4	1p1q8t	1p2q1t
WS-X6502-10GE	1p1q8t	1p2q1t
WS-X6516-GBIC	1p1q4t	1p2q2t
WS-X6516-GE-TX	1p1q4t	1p2q2t
WS-X6416-GBIC	1p1q4t	1p2q2t
WS-X6416-GE-MT	1p1q4t	1p2q2t
WS-X6316-GE-TX	1p1q4t	1p2q2t
WS-X6408A-GBIC	1p1q4t	1p2q2t
WS-X6408-GBIC	1q4t	2q2t
WS-X6524-100FX-MM	1p1q0t	1p3q1t
WS-X6324-100FX-SM	1q4t	2q2t
WS-X6324-100FX-MM	1q4t	2q2t
WS-X624-100FX-MT	1q4t	2q2t
WS-X6548-RJ-21	1p1q0t	1p3q1t
WS-X6548-RJ-45	1p1q0t	1p3q1t
WS-X6348-RJ-21	1q4t	2q2t
WS-X6348-RJ21V	1q4t	2q2t

WS-X6348-RJ-45	1q4t	2q2t
WS-X6348-RJ-45V	1q4t	2q2t
WS-X6148-RJ-45V	1q4t	2q2t
WS-X6148-RJ21V	1q4t	2q2t
WS-X6248-RJ-45	1q4t	2q2t
WS-X6248A-TEL	1q4t	2q2t
WS-X6248-TEL	1q4t	2q2t
WS-X6024-10FL-MT	1q4t	2q2t

[Pianificazione della configurazione, del monitoraggio e dell'output di esempio sugli switch Catalyst 6500/6000](#)

[Configurazione](#)

In questa sezione vengono descritti tutti i passaggi necessari per configurare la pianificazione dell'output su uno switch Catalyst 6500/6000 con software Cisco IOS. Per la configurazione predefinita di Catalyst 6500/6000, vedere la [richiesta 1: QoS è abilitato e viene utilizzato un parametro predefinito](#) nella sezione di questo documento.

La configurazione di Catalyst 6500/6000 implica i seguenti cinque passaggi:

1. [Abilita QoS](#)
2. [Mappare ogni possibile valore CoS \(Class of Service\) a una coda e a una soglia](#) (facoltativo)
3. [Configurazione del peso WRR](#) (facoltativo)
4. [Configurare i buffer assegnati a ciascuna coda](#) (facoltativo)
5. [Configurare il livello di soglia per ogni coda](#) (facoltativo)

Nota: ciascuno di questi passi è facoltativo, ad eccezione del passo 1. È possibile decidere di lasciare il valore predefinito per uno o più parametri.

[Passaggio 1: Abilita QoS](#)

Innanzitutto, abilitare QoS. Tenere presente che QoS è disabilitato per impostazione predefinita. Quando QoS è disabilitato, il mapping CoS configurato non influisce sul risultato. C'è una coda servita in modo FIFO (First In, First Out) e tutti i pacchetti vengono scartati lì.

```
cosmos#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
cosmos(config)#mls qos
```

QoS is enabled globally

```
Microflow policing is enabled globally
```

QoS global counters:

```
Total packets: 552638
IP shortcut packets: 0
Packets dropped by policing: 0
IP packets with TOS changed by policing: 0
IP packets with COS changed by policing: 0
Non-IP packets with CoS changed by policing: 0
```

Passaggio 2: Mapping di ogni valore CoS possibile a una coda e a una soglia

Per tutti i tipi di coda, assegnare il CoS a una coda e a una soglia. Il mapping definito per un tipo di porta 2q2t non viene applicato ad alcuna porta 1p2q2t. Inoltre, il mapping per 2q2t viene applicato a tutte le porte con un meccanismo di coda 2q2t. Utilizzare i seguenti comandi **cos-map** nell'interfaccia:

```
wrr-queue cos-map Q_number_(1-2) threshold_number_(1-2) cos_value_1 cos_value_2
priority-queue cos-map Q_number_(always 1) cos_value_1 cos_value_2
```

Nota: ognuno di questi comandi deve essere su una riga.

È possibile configurare separatamente la coda WRR. Se è presente una coda di priorità, è possibile configurarla con il comando **priority-queue**.

Nota: le code sono sempre numerate a partire dalla coda con la priorità più bassa possibile fino alla coda con priorità rigorosa disponibile. Ad esempio:

- La coda 1 è la coda WRR a bassa priorità.
- La coda 2 è la coda WRR ad alta priorità.
- La coda 3 è la coda con priorità rigorosa.

Ripetere l'operazione per tutti i tipi di coda, altrimenti l'assegnazione CoS predefinita rimane. Questa è una configurazione di esempio per 1p2q2t:

```
cosmos#configure terminal
cosmos(config)#interface gigabitethernet 1/1
cosmos(config-if)#priority-queue cos-map 1 5
!--- Assign a CoS of 5 to priority queue. cos-map configured on: Gi1/1 Gi1/2 cosmos(config-
if)#wrr-queue cos-map 1 1 0 1
!--- Assign CoS 0 and 1 to the first threshold of low-priority WRR queue. cos-map configured on:
Gi1/1 Gi1/2 cosmos(config-if)#wrr-queue cos-map 1 2 2 3
!--- Assign CoS 2 and 3 to the second threshold of low-priority WRR queue. cos-map configured
on: Gi1/1 Gi1/2 cosmos(config-if)#wrr-queue cos-map 2 1 4 6
!--- Assign CoS 4 and 6 to the first threshold of high-priority WRR queue. cos-map configured
on: Gi1/1 Gi1/2 cosmos(config-if)#wrr-queue cos-map 2 2 7
!--- Assign CoS 7 to the first threshold of high-priority WRR queue. cos-map configured on:
Gi1/1 Gi1/2
```

Controllare la configurazione:

```
cosmos#show queueing interface gigabitethernet 1/1
!--- Output suppressed. queue thresh cos-map ----- 1 1 0 1 1 2
2 3 2 1 4 6 2 2 7 3 1 5 !--- Output suppressed.
```

Passaggio 3: Configurazione del peso WRR

Configurare il peso WRR per le due code WRR. Utilizzare questo comando di interfaccia:

```
wrr-queue bandwidth weight_for_Q1 weight_for_Q2
```

Il peso 1 si riferisce alla coda 1, che deve essere la coda WRR a bassa priorità. Mantieni sempre

questo peso a un livello inferiore rispetto al peso 2. Il peso può assumere qualsiasi valore compreso tra 1 e 255. Utilizzare le seguenti formule per assegnare la percentuale:

- Alla coda 1— $[\text{peso } 1 / (\text{peso } 1 + \text{peso } 2)]$
- Alla coda 2— $[\text{peso } 2 / (\text{peso } 1 + \text{peso } 2)]$

È necessario definire il peso per tutti i tipi di code. Questi tipi di peso non devono essere necessariamente gli stessi. Questo è un esempio per 2q2t, dove la coda 1 viene servita il 20% del tempo e la coda 2 viene servita l'80% del tempo:

```
cosmos#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
cosmos(config)#interface gigabitethernet 1/1
cosmos(config-if)#wrr-queue bandwidth ?
<1-255> enter bandwidth weight between 1 and 255
cosmos(config-if)#wrr-queue bandwidth 20 80
!--- Queue 1 is served 20% of the time, and queue 2 is served !--- 80% of the time.
cosmos(config-if)#
```

Controllare la configurazione:

```
cosmos#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy: Weighted Round-Robin
Port QoS is enabled
Port is untrusted
Default cos is 0
Transmit queues [type = 1p2q2t]:
  Queue Id      Scheduling  Num of thresholds
  -----
   1            WRR low    2
   2            WRR high   2
   3            Priority   1

  WRR bandwidth ratios:  20[queue 1]  80[queue 2]
  queue-limit ratios:   90[queue 1]  5[queue 2]
!--- Output suppressed.
```

Nota: quando non è possibile usare il software CatOS, è possibile configurare pesi WRR diversi per ciascuna interfaccia.

[Passaggio 4: Configurare i buffer assegnati a ciascuna coda](#)

È necessario definire il rapporto della coda di trasmissione. Determina la modalità di divisione dei buffer tra le diverse code.

```
wrr-queue queue-limit percentage_WRR_Q1 percentage_WRR_Q2
cosmos(config)#interface gigabitethernet 1/2
cosmos(config-if)#wrr-queue queue-limit 70 15
!--- Queue 1 has 70% of the buffers. !--- Queues 2 and 3 both have 15% of the buffers. queue-
limit configured on: Gi1/1 Gi1/2
```

Nota: se la funzionalità di accodamento della porta Gigabit è 1p1q2t, è necessario utilizzare lo stesso livello per la coda con priorità rigida e per la coda WRR con priorità alta. Questi livelli non possono essere diversi per motivi hardware. È configurata solo la larghezza di banda per le due code WRR. Lo stesso valore viene utilizzato automaticamente per la coda WRR con priorità alta e per la coda con priorità rigida, se presente.

Per alcuni tipi di coda non è possibile regolare le dimensioni della coda. Un esempio è 1p3q1t, disponibile su WS-X6548RJ45. Questi tipi di coda sono fissi e non possono essere modificati.

Controllare la configurazione:

```
cosmos#show queueing interface gigabitethernet 1/2
Interface GigabitEthernet1/2 queueing strategy: Weighted Round-Robin
Port QoS is enabled
Port is untrusted
Default cos is 0
Transmit queues [type = 1p2q2t]:
  Queue Id      Scheduling  Num of thresholds
  -----
      1          WRR low           2
      2          WRR high           2
      3          Priority           1

WRR bandwidth ratios:      5[queue 1] 255[queue 2]
queue-limit ratios:       70[queue 1] 15[queue 2]
```

Nota: è consigliabile lasciare la parte più grande dei buffer per la coda WRR a bassa priorità. Questa è la coda in cui è necessario abilitare il buffer aggiuntivo. Le altre code vengono servite con priorità più alta.

[Passaggio 5: Configurare il livello di soglia per ciascuna coda](#)

Come passo finale, configurare il livello di soglia per la coda WRED o per la coda di rilascio. Nell'elenco sono disponibili i comandi seguenti:

- Per le code che utilizzano WRED come meccanismo di rilascio per la soglia, eseguire i seguenti comandi:

```
wrr-queue random-detect min-threshold Q_number threshold_1_value threshold_2_value
wrr-queue random-detect max-threshold Q_number threshold_1_value threshold_2_value
```

Nota: ognuno di questi comandi deve essere su una riga.

- Per le code che utilizzano tail drop come meccanismo di rilascio, eseguire questo comando:

```
wrr-queue threshold Q_number threshold_1_value threshold_2_value
```

Nota: questo comando deve essere su una riga.

Configurazione per una coda WRED:

```
cosmos(config)#interface gigabitethernet 1/1
cosmos(config-if)#wrr-queue random-detect min-threshold 1 20 50
!--- This sets the threshold of queue 1 to 20 and 50% minimum threshold !--- configured on Gi1/1
Gi1/2. cosmos(config-if)#wrr-queue random-detect min-threshold 2 20 50
!--- This sets the threshold of queue 2 to 20 and 50% minimum threshold !--- configured on Gi1/1
Gi1/2. cosmos(config-if)#wrr-queue random-detect max-threshold 1 50 80
!--- This sets the threshold of queue 1 to 50 and 80% maximum threshold !--- configured on Gi1/1
Gi1/2. cosmos(config-if)#wrr-queue random-detect max-threshold 2 40 60
!--- This sets the threshold of queue 2 to 49 and 60% maximum threshold !--- configured on Gi1/1
Gi1/2.
```

Configurazione per una coda di rilascio:

```

cosmos(config)#interface fastethernet 3/1
cosmos(config-if)#wrr-queue threshold ?
<1-2> enter threshold queue id (1-2)
cosmos(config-if)#wrr-queue threshold 1 ?
<1-100> enter percent of queue size between 1 and 100
cosmos(config-if)#wrr-queue threshold 1 50 100
!--- This sets the tail drop threshold for this 2q2t interface for !--- queue 1 (low-priority)
to 50 and 100% of the buffer. threshold configured on: Fa3/1 Fa3/2 Fa3/3 Fa3/4 Fa3/5 Fa3/6 Fa3/7
Fa3/8 Fa3/9 Fa3/10 Fa3/11 Fa3/12 cosmos(config-if)# cosmos(config-if)# cosmos(config-if)#wrr-
queue threshold 2 40 100
!--- This sets the tail drop threshold for this 2q2t interface for !--- queue 2 (high-priority)
to 40 and 100% of the buffer. threshold configured on: Fa3/1 Fa3/2 Fa3/3 Fa3/4 Fa3/5 Fa3/6 Fa3/7
Fa3/8 Fa3/9 Fa3/10 Fa3/11 Fa3/12 cosmos(config-if)#

```

Controllare la configurazione:

```

cosmos#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy: Weighted Round-Robin
Port QoS is enabled
Port is untrusted
Default cos is 0
Transmit queues [type = 1p2q2t]:
  Queue Id      Scheduling  Num of thresholds
  -----
  1             WRR low     2
  2             WRR high    2
  3             Priority    1

WRR bandwidth ratios:  20[queue 1]  80[queue 2]
queue-limit ratios:    70[queue 1]  15[queue 2]

queue random-detect-min-thresholds
-----
  1    20[1] 50[2]
  2    20[1] 50[2]

queue random-detect-max-thresholds
-----
  1    50[1] 80[2]
  2    40[1] 60[2]

```

```

cosmos#show queueing interface fastethernet 3/1
Interface FastEthernet3/1 queueing strategy: Weighted Round-Robin
Port QoS is enabled
Port is untrusted
Default cos is 0
Transmit queues [type = 2q2t]:
  Queue Id      Scheduling  Num of thresholds
  -----
  1             WRR low     2
  2             WRR high    2

WRR bandwidth ratios:  100[queue 1] 255[queue 2]
queue-limit ratios:    90[queue 1]  10[queue 2]

queue tail-drop-thresholds
-----
  1    50[1] 100[2]
  2    40[1] 100[2]

```

Non è possibile configurare la soglia e assegnare il CoS alla coda per porta. Tutte le modifiche vengono applicate a un set di porte contigue:

- Quattro porte per schede di linea gigabit: le porte da 1 a 4 sono unite, mentre le porte da 5 a 8 sono unite.
- Dodici porte per porte 10/100 o 100 porte in fibra in base all'accodamento 1q4t/2q2t: da 1 a 12, da 13 a 24, da 25 a 36 e da 36 a 48.
- Per determinare l'esatta porta appartenente allo stesso ASIC, usare il comando **show interface capabilities**.

Monitoraggio della programmazione dell'output e verifica delle configurazioni

Il comando più semplice da eseguire per verificare la configurazione corrente in fase di esecuzione di una porta per quanto riguarda la pianificazione dell'output è **show queueing interface {gigabitethernet | fastethernet}** comando *slot/port*. Questo comando visualizza il tipo di coda sulla porta, il mapping del CoS alle diverse code e soglie, la condivisione del buffer e il peso del WRR. In questo caso, il valore WRR è del 20% per la coda 1 e l'80% per la coda 2. Il comando visualizza inoltre tutte le informazioni configurate per la pianificazione dell'output e il numero di pacchetti scartati in ciascuna coda per ciascuna soglia:

```
cosmos#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy: Weighted Round-Robin
Port QoS is enabled
Port is untrusted
Default COS is 0
Transmit queues [type = 1p2q2t]:
  Queue Id      Scheduling  Num of thresholds
  -----
  1             WRR low     2
  2             WRR high    2
  3             Priority    1
WRR bandwidth ratios:  20[queue 1]  80[queue 2]
queue-limit ratios:    70[queue 1]  15[queue 2]

queue random-detect-max-thresholds
-----
  1    50[1] 80[2]
  2    40[1] 60[2]

queue thresh cos-map
-----
  1    1    0 1
  1    2    2 3
  2    1    4 6
  2    2    7
  3    1    5

Receive queues [type = 1p1q4t]:
  Queue Id      Scheduling  Num of thresholds
  -----
  1             Standard    4
  2             Priority    1

queue tail-drop-thresholds
-----
  1    100[1] 100[2] 100[3] 100[4]
```

```

queue thresh cos-map
-----
1      1      0 1
1      2      2 3
1      3      4
1      4      6 7
2      1      5

```

Packets dropped on Transmit:

BPDU packets: 0

```

queue thresh      dropped  [cos-map]
-----
1      1              0 [0 1 ]
1      2              0 [2 3 ]
2      1              0 [4 6 ]
2      2              0 [7 ]
3      1              0 [5 ]

```

Packets dropped on Receive:

BPDU packets: 0

```

queue thresh      dropped  [cos-map]
-----
1      1              0 [0 1 ]
1      2              0 [2 3 ]
1      3              0 [4 ]
1      4              0 [6 7 ]
2      1              0 [5 ]

```

Esempio di programmazione dell'output

Questo traffico viene iniettato sugli switch Catalyst 6500/6000:

- Porta gigabit 1/2: un gigabit di traffico con precedenza pari a zero
- Porta gigabit 5/2: 133 MB di traffico con precedenza di sette133 MB di traffico con precedenza di sei133 MB di traffico con precedenza di cinque133 MB di traffico con precedenza di quattro133 MB di traffico con precedenza di tre133 MB di traffico con precedenza di due133 MB di traffico con precedenza di uno

Tutto il traffico unicast esce dallo switch per porta gigabit 1/1, con una sovrascrittura notevole.

Caso 1: QoS è abilitato e viene utilizzato un parametro predefinito

Il comando **show queueing interface gigabitethernet 1/1** configura tutto l'output di questo esempio. Il comando fornisce informazioni aggiuntive sulla pianificazione dell'input. Tuttavia, poiché questo documento copre solo la programmazione dell'output, elimina tale output.

Quando QoS è abilitato a livello globale e tutti i parametri predefiniti sono in uso, questo output viene restituito dopo alcuni minuti:

```

nelix#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy: Weighted Round-Robin
Port QoS is enabled
Trust state: trust DSCP
Default cos is 0

```

```

Transmit queues [type = lp2q2t]:
Queue Id      Scheduling  Num of thresholds
-----
   1          WRR low           2
   2          WRR high           2
   3          Priority          1

WRR bandwidth ratios: 100[queue 1] 255[queue 2]
queue-limit ratios:   90[queue 1]   5[queue 2]

queue random-detect-max-thresholds
-----
   1    40[1] 100[2]
   2    40[1] 100[2]

queue thresh cos-map
-----
   1    1    0 1
   1    2    2 3
   2    1    4
   2    2    6 7
   3    1    5

Packets dropped on Transmit:
BPDU packets: 0

```

```

queue thresh      dropped  [cos-map]
-----
   1    1    149606424  [0 1 ]
   1    2           0  [2 3 ]
   2    1    16551394  [4 ]
   2    2    4254446   [6 7 ]
   3    1           0  [5 ]

```

In questo output, i valori predefiniti sono:

- Peso WRR per coda 1— $100 / (100 + 255) = 28\%$
- Peso WRR per coda 2— $255 / (255 + 100) = 72\%$
- Condivisione buffer: —90% per la coda 1, 5% per la coda 2 e 5% per la coda con priorità rigorosa

La maggior parte dei pacchetti nella coda WRR a bassa priorità vengono scartati, ma alcuni vengono ancora scartati nella coda WRR ad alta priorità per entrambe le soglie. Vi è un totale di 170.412.264 cadute ($149.606.424 + 16.551.394 + 4.254.446$). Le gocce vengono suddivise come segue:

- $149.606.424 / 170.412.264 = 88\%$ delle cadute nella coda 1 (primo pacchetto di soglia con CoS 0 e 1)
- $16.551.394 / 170.412.264 = 10\%$ di cadute nella coda 2 (primo pacchetto di soglia con CoS 4)
- $4.254.446 / 170.412.264 = 2\%$ di cadute nella coda 2 (secondo pacchetto di soglia con CoS di 6 o 7)

Nota: nella coda con priorità rigorosa non viene visualizzata alcuna perdita.

Caso 2: Modifica peso WRR

Come osservato nel [caso 1: QoS è abilitato e viene utilizzato un parametro predefinito](#), i pacchetti nella coda 2 vengono ancora scartati. Modificare il peso WRR per assegnare una maggiore larghezza di banda alla coda 2. Ora, la coda 1 viene svuotata il 4% del tempo e la coda 2 viene

svuotata il 96% del tempo:

```
show run interface gigabitethernet 1/1
```

```
interface GigabitEthernet1/1
no ip address
wrr-queue bandwidth 10 255
mls qos trust dscp
switchport
switchport mode access
end
```

```
nelix#show queueing interface gigabitethernet 1/1
```

```
Interface GigabitEthernet1/1 queueing strategy: Weighted Round-Robin
```

```
Port QoS is enabled
```

```
Trust state: trust DSCP
```

```
Default cos is 0
```

```
Transmit queues [type = lp2q2t]:
```

Queue Id	Scheduling	Num of thresholds
1	WRR low	2
2	WRR high	2
3	Priority	1

```
WRR bandwidth ratios: 10[queue 1] 255[queue 2]
```

```
queue-limit ratios: 90[queue 1] 5[queue 2]
```

```
queue random-detect-max-thresholds
```

1	40[1]	100[2]
2	40[1]	100[2]

```
queue thresh cos-map
```

1	1	0	1
1	2	2	3
2	1	4	
2	2	6	7
3	1	5	

```
Packets dropped on Transmit:
```

```
BPDU packets: 0
```

```
queue thresh dropped [cos-map]
```

1	1	2786205	[0 1]
1	2	0	[2 3]
2	1	11363	[4]
2	2	69	[6 7]
3	1	0	[5]

Come si vede in questo output, la percentuale di cadute nella coda 2 è ora molto più bassa. Un totale di 2.797.637 gocce vengono suddivise nel modo seguente:

- $2.786.205 / 2.797.637 = 99,591\%$ delle cadute nella coda 1 (con pacchetto CoS 0 e 1)
- $11.363 / 2.797.637 = 0,408\%$ delle cadute nella coda 2 (prima soglia con pacchetto CoS 4)
- $69 / 2.797.637 = 0,001\%$ delle cadute nella coda 2 (seconda soglia per il pacchetto con CoS 6 e 7)

Se si utilizzano pesi WRR diversi, si assicura un maggiore QoS nella coda 2.

Caso 3: Ulteriore modifica del peso WRR

Con il peso WRR si può essere ancora più aggressivi. In questo output di esempio, solo lo 0,39% del peso viene assegnato alla coda 1:

```
show run interface gigabitethernet 1/1
interface GigabitEthernet1/1
  no ip address
  wrr-queue bandwidth 1 255
  mls qos trust dscp
  switchport
  switchport mode access
end

nelix#show queueing interface gigabitethernet 1/1
Interface GigabitEthernet1/1 queueing strategy: Weighted Round-Robin
  Port QoS is enabled
  Trust state: trust DSCP
  Default cos is 0
  Transmit queues [type = lp2q2t]:
    Queue Id      Scheduling  Num of thresholds
    -----
      1           WRR low      2
      2           WRR high     2
      3           Priority     1

  WRR bandwidth ratios:   1[queue 1] 255[queue 2]
  queue-limit ratios:     90[queue 1]  5[queue 2]

  queue random-detect-max-thresholds
  -----
      1   40[1] 100[2]
      2   40[1] 100[2]

  queue thresh cos-map
  -----
      1   1   0 1
      1   2   2 3
      2   1   4
      2   2   6 7
      3   1   5

  Packets dropped on Transmit:
    BPDU packets: 0

  queue thresh      dropped  [cos-map]
  -----
      1   1          2535315  [0 1 ]
      1   2              0  [2 3 ]
      2   1             705  [4 ]
      2   2              73  [6 7 ]
      3   1              0  [5 ]
```

Anche con il peso WRR aggressivo, i pacchetti vengono ancora scartati nella coda 2. Tuttavia, al confronto, non sono molti i pacchetti. La coda 2 contiene ora solo lo 0,03% dei pacchetti.

Caso 4: Modifica assegnazione buffer limite coda

Come illustrato nel [caso 2: Modificare il peso WRR](#) e la [richiesta 3: Nelle](#) sezioni [aggiuntive per la](#)

[modifica del peso WRR](#), i pacchetti sono ancora in attesa nella coda 2, anche se la percentuale WRR assicura che il rilascio è minimo. Tuttavia, quando si raggiunge la seconda soglia (impostata sul 100%) nella coda 2, alcuni pacchetti continuano a essere scartati.

Per migliorare questo aspetto, modificare il limite della coda (dimensione del buffer assegnato a ciascuna coda). In questo esempio, il limite della coda è impostato su 70% per la coda 1, 15% per la coda 2 e 15% per la coda con priorità rigorosa:

```
show run gigabitethernet 1/1
```

```
interface GigabitEthernet1/1
  no ip address
  wrr-queue bandwidth 1 255
  wrr-queue queue-limit 70 15
  mls qos trust dscp
  switchport
  switchport mode access
end
```

```
nelix#show queueing interface gigabitethernet 1/1
```

```
Interface GigabitEthernet1/1 queueing strategy: Weighted Round-Robin
```

```
Port QoS is enabled
```

```
Trust state: trust DSCP
```

```
Default cos is 0
```

```
Transmit queues [type = 1p2q2t]:
```

Queue Id	Scheduling	Num of thresholds
1	WRR low	2
2	WRR high	2
3	Priority	1

```
WRR bandwidth ratios: 1[queue 1] 255[queue 2]
```

```
queue-limit ratios: 70[queue 1] 15[queue 2]
```

```
queue random-detect-max-thresholds
```

1	40[1]	100[2]
2	40[1]	100[2]

```
queue thresh cos-map
```

1	1	0	1
1	2	2	3
2	1	4	
2	2	6	7
3	1	5	

```
Receive queues [type = 1plq4t]:
```

Queue Id	Scheduling	Num of thresholds
1	Standard	4
2	Priority	1

```
queue tail-drop-thresholds
```

```
1 100[1] 100[2] 100[3] 100[4]
```

```
queue thresh cos-map
```

1	1	0	1
1	2	2	3

```

1      3      4
1      4      6 7
2      1      5

```

```

Packets dropped on Transmit:
BPDU packets: 0

```

```

queue thresh      dropped  [cos-map]
-----
1      1      154253046  [0 1 ]
1      2              0  [2 3 ]
2      1              0  [4 ]
2      2              0  [6 7 ]
3      1              0  [5 ]

```

Ora, le gocce si verificano solo nella coda 1.

Utilizzare la programmazione dell'output per ridurre il ritardo e l'instabilità

Gli studi di casi nella sezione [Esempio di pianificazione dell'output](#) dimostrano il vantaggio di implementare la pianificazione dell'output per evitare una caduta del traffico VoIP o mission-critical in caso di sovrasscrizione della porta di output. L'oversubscription non si verifica molto di frequente in una normale rete (in particolare su un collegamento Gigabit). L'abbonamento deve essere effettuato solo nelle ore di punta del traffico o in caso di picchi di traffico che si verificano entro un periodo di tempo molto breve.

Anche senza sottoscrizioni eccessive, la pianificazione dell'output può essere di grande aiuto in una rete in cui QoS è implementato end-to-end. In questa sezione vengono forniti esempi di come la programmazione dell'output può contribuire a ridurre il ritardo e l'instabilità.

Riduci ritardo

Il ritardo di un pacchetto aumenta a causa del tempo "perso" nel buffer di ciascuno switch durante l'attesa della trasmissione. Ad esempio, un piccolo pacchetto vocale con un CoS di 5 viene inviato da una porta durante un backup o un trasferimento di file di grandi dimensioni. Si supponga che non vi sia QoS per la porta di output e che il pacchetto vocale piccolo venga messo in coda dopo 10 pacchetti grandi da 1500 byte. In questo caso, è possibile calcolare facilmente il tempo di velocità in gigabit necessario per trasmettere i 10 pacchetti di grandi dimensioni:

- $(10 \times 1500 \times 8) = 120.000$ bit trasmessi in 120 microsecondi

Se il pacchetto deve attraversare otto o nove switch durante il passaggio sulla rete, può verificarsi un ritardo di circa 1 millisecondo. Ciò include solo i ritardi nella coda di output dello switch attraversato nella rete.

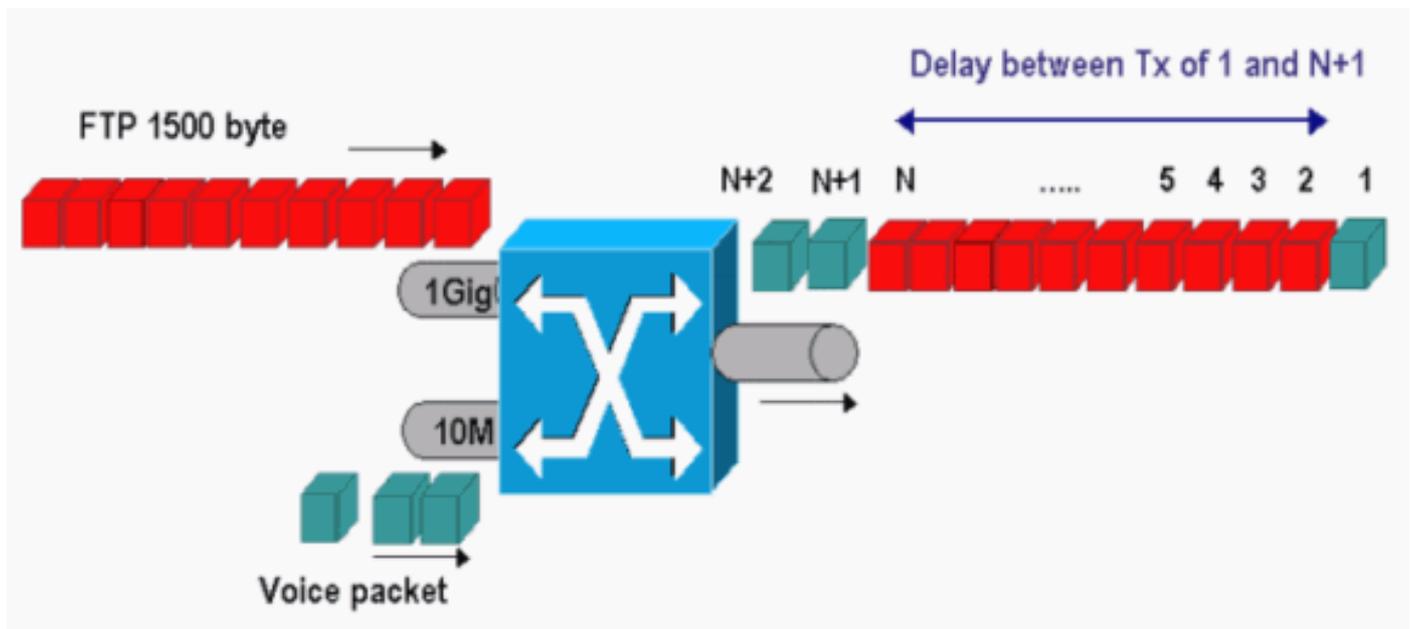
Nota: se è necessario mettere in coda gli stessi 10 pacchetti di grandi dimensioni su un'interfaccia da 10 MB (ad esempio, collegata a un telefono IP e a un PC), l'introduzione del ritardo è:

- $(10 \times 1500 \times 8) = 120.000$ bit trasmessi in 12 millisecondi

L'implementazione della programmazione dell'output assicura che i pacchetti voce con un CoS di 5 vengano inseriti nella coda con priorità assoluta e vengano inviati prima di qualsiasi pacchetto con un CoS inferiore a 5. Ciò riduce i ritardi.

Riduci variazione

Un altro vantaggio importante della programmazione dell'output è la riduzione dell'effetto jitter. Jitter è la variazione di ritardo dei pacchetti all'interno dello stesso flusso. In questo scenario di esempio viene illustrato come la programmazione dell'output può ridurre la variazione:



In questo scenario, la stessa porta di output deve inviare due flussi:

- Un flusso vocale in ingresso su una porta Ethernet da 10 MB.
- Un flusso FTP in ingresso su una porta uplink 1 Gigabit Ethernet.

Entrambi i flussi lasciano lo switch attraverso la stessa porta di output. In questo esempio viene mostrato ciò che può verificarsi senza l'utilizzo della programmazione dell'output. Tutti i pacchetti di dati di grandi dimensioni possono essere interlacciati tra due pacchetti voce. Questo crea un tremolio nella ricezione del pacchetto voce dallo stesso flusso. Tra la ricezione del pacchetto 1 e del pacchetto $n + 1$ si verifica un ritardo maggiore in quanto lo switch trasmette il pacchetto di dati di grandi dimensioni. Tuttavia, il ritardo tra $n + 1$ e $n + 2$ è trascurabile. Ciò determina uno jitter nel flusso del traffico vocale. È possibile evitare questo problema utilizzando una coda con priorità rigorosa. Accertarsi di mappare il valore CoS dei pacchetti voce alla coda a priorità rigorosa.

Conclusioni

In questo documento sono stati riportati alcuni casi di studio su come configurare e risolvere i problemi di pianificazione delle code di output su uno switch Catalyst 6500/6000 con software Cisco IOS. Nella maggior parte delle reti con traffico vocale sono stati inoltre illustrati i vantaggi della pianificazione dell'output:

- Evita la perdita di traffico critico in caso di sovrascrittura della porta di output.
- Riduce il ritardo.
- Riduce l'effetto jitter.

Informazioni correlate

- [Pianificazione dell'output QoS sugli switch Catalyst serie 6500/6000 con software CatOS](#)

- [Qualità del servizio sugli switch Catalyst serie 6000](#)
- [Pagine di supporto dei prodotti LAN](#)
- [Pagina di supporto dello switching LAN](#)
- [Documentazione e supporto tecnico – Cisco Systems](#)