Comprensione e configurazione di MDRR/WRED sui router serie 12000

Sommario

Introduzione

In questo documento viene descritto come configurare le funzionalità di gestione della congestione e prevenzione della congestione del software Cisco IOS® su Cisco serie 12000 Internet Router.

Dopo aver letto il documento, è necessario essere in grado di:

- Comprendere perché è importante configurare Modified Deficit Round Robin (MDRR) e Weighted Random Early Detection (WRED) nella rete principale.

- Comprendere l'implementazione alla base di MDRR e WRED su Cisco serie 12000.
- Configurare MDRR e WRED utilizzando la sintassi CoS (Class of Service) legacy e Modular QoS CLI (MQC).

Prerequisiti

Requisiti

Questo documento è utile per conoscere i seguenti argomenti:

- Una conoscenza generale dell'architettura di Cisco serie 12000 Internet Router.
- In particolare, una conoscenza dell'architettura di coda e questi termini:ToFab (Verso la struttura) - descrive le code lato ricezione su una scheda di linea in entrata.FrFab (dalla struttura) - descrive le code lato trasmissione su una scheda di linea in uscita.

Nota: si consiglia inoltre di cercare How to Read the Output of the show controller frame | tofab queue Comandi su un router Cisco serie 12000 Internet.

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

- Tutte le piattaforme Cisco 12000, che includono le versioni 12008, 12012, 12016, 12404, 12406, 12410 e 12416.
- Software Cisco IOS release 12.0(24)S1.

Nota: anche se le configurazioni descritte in questo documento sono state testate con il software Cisco IOS versione 12.0(24)S1, è possibile usare qualsiasi versione del software Cisco IOS che supporti Cisco serie 12000 Internet Router.

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.

Premesse

I metodi di coda definiscono il meccanismo di pianificazione dei pacchetti o l'ordine in cui i pacchetti vengono rimossi dalla coda per essere trasmessi sul cavo fisico. In base all'ordine e al numero di volte in cui una coda viene servita da una funzione di programmazione, i metodi di coda supportano anche le garanzie di larghezza di banda minima e le latenze basse.

Èimportante assicurarsi che un meccanismo di pianificazione dei pacchetti supporti l'architettura di commutazione su cui viene implementato. WFQ (Weighted Fair Queuing) è il noto algoritmo di

pianificazione per l'allocazione delle risorse sulle piattaforme router Cisco con un'architettura basata su bus. Tuttavia, non è supportato sui Cisco serie 12000 Internet Router. Non sono supportate neanche le code di priorità software Cisco IOS tradizionali e le code personalizzate. Al contrario, Cisco serie 12000 utilizza una forma speciale di coda chiamata MDRR (Modified Deficit Round Robin), che fornisce garanzie sulla larghezza di banda relativa e una coda a bassa latenza. M di MDRR significa "modificato"; aggiunge la coda di priorità rispetto a DRR in cui non è presente alcuna coda di priorità. Per ulteriori informazioni su MDRR, vedere la sezione Panoramica su MDRR.

Inoltre, Cisco serie 12000 supporta WRED (Weighted Random Early Detection) come regola di drop all'interno delle code MDRR. Questo meccanismo di prevenzione della congestione costituisce un'alternativa al meccanismo predefinito di perdita della coda. La congestione può essere evitata con cadute controllate.

La prevenzione della congestione e i meccanismi di gestione come WRED e MDRR sono particolarmente importanti nelle code FrFab di interfacce in uscita a velocità relativamente bassa, come le schede di linea canalizzate (LC). Il fabric dello switch ad alta velocità è in grado di consegnare i pacchetti ai gruppi di canali molto più rapidamente di quanto possano trasmetterli i gruppi di canali. Poiché le code e i buffer vengono gestiti a livello di porta fisica, la pressione di un canale può influire su tutti gli altri canali della porta. Pertanto, è molto importante gestire tale contropressione tramite WRED/MDRR, che limita l'impatto della contropressione ai canali in questione. Per i dettagli su come gestire l'oversubscription dell'interfaccia in uscita, vedere Risoluzione dei problemi relativi ai pacchetti ignorati e all'assenza di perdite di memoria sul router Internet Cisco serie 12000.

Panoramica di MDRR

Questa sezione fornisce una panoramica di Deficit Round Robin (MDRR) modificato.

Con la MDRR configurata come strategia di coda, le code non vuote vengono servite una dopo l'altra, in modo round robin. Ogni volta che viene servita una coda, viene rimossa una quantità fissa di dati. L'algoritmo quindi serve la coda successiva. Quando una coda viene servita, MDRR tiene traccia del numero di byte di dati rimossi dalla coda in eccesso rispetto al valore configurato. Nel passaggio successivo, quando la coda viene nuovamente servita, verrà rimossa una quantità minore di dati per compensare i dati in eccesso precedentemente serviti. Di conseguenza, la quantità media di dati rimossi dalla coda per coda sarà vicina al valore configurato. Inoltre, MDRR mantiene una coda di priorità che viene servita su base preferenziale. La funzione MDRR viene spiegata in dettaglio in questa sezione.

Ogni coda all'interno di MDRR è definita da due variabili:

- Valore quantum - Numero medio di byte serviti in ciascun arrotondamento.
- Contatore deficit - Viene utilizzato per tenere traccia del numero di byte trasmessi da una coda in ogni ciclo. Viene inizializzato sul valore quantistico.

I pacchetti in una coda vengono serviti finché il contatore di deficit è maggiore di zero. Ogni pacchetto fornito riduce il contatore di deficit di un valore uguale alla sua lunghezza in byte. Una coda non può più essere servita quando il contatore di deficit diventa zero o negativo. In ogni nuovo ciclo, il contatore di deficit di ogni coda non vuota viene aumentato del relativo valore quantico.

Nota: in generale, la dimensione quantistica di una coda non deve essere inferiore alla MTU

(Maximum Transmission Unit) dell'interfaccia. In questo modo lo scheduler serve sempre almeno un pacchetto da ciascuna coda non vuota.

A ogni coda MDRR può essere assegnato un peso relativo, con una delle code nel gruppo definita come coda di priorità. I pesi assegnano la larghezza di banda relativa per ciascuna coda quando l'interfaccia è congestionata. L'algoritmo MDRR rimuove i dati da ciascuna coda in modo round robin se nella coda sono presenti dati da inviare.

Se tutte le code MDRR normali contengono dati, vengono servite nel modo seguente:

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

Durante ogni ciclo, una coda può rimuovere dalla coda un quantum in base al peso configurato. Sulle schede di linea del motore 0 e del motore 2, il valore 1 equivale a dare all'interfaccia il peso della relativa MTU. Per ogni incremento superiore a 1, il peso della coda aumenta di 512 byte. Ad esempio, se l'MTU di una particolare interfaccia è 4470 e il peso di una coda è configurato su 3, ad ogni rotazione, è consentito rimuovere dalla coda 4470 + (3-1)*512 = 5494 byte. Se si utilizzano due code DRR normali, Q0 e Q1, Q0 viene configurato con un peso di 1 e Q1 viene configurato con un peso di 9. Se entrambe le code sono congestionate, ogni volta che si passa alla rotazione, Q0 può inviare 4470 byte e Q1 può inviare [4470 + (9-1)*512] = 8566 byte. In questo modo, il traffico che raggiunge Q0 equivale a circa 1/3 della larghezza di banda e il traffico che attraversa Q1 equivale a circa 2/3 della larghezza di banda.

Nota: la formula di rimozione dalla coda standard utilizzata per calcolare l'assegnazione della larghezza di banda MDRR è D = MTU + (weight-1)*512. Nelle versioni precedenti al software Cisco IOS versione 12.0(21) S/ST, le schede di linea del motore 4 utilizzavano una formula di rimozione dalla coda diversa. Per configurare il peso MDRR per un'assegnazione corretta della larghezza di banda, verificare di eseguire un software Cisco IOS versione successiva alla 12.0(21) S/ST.

Nota: la formula quantistica per le schede di linea Engine 4+ è (per la direzione toFab, non è valida per FrFab) Quantum = BaseWeight + {BaseWeight * (QueueWeight - 1) * 512} / MTU. BaseWeight viene ottenuto con la seguente formula: BaseWeight = {(MTU + 512 - 1) / 512} + 5

Nota: Tutti i calcoli sono arrotondati. ovvero tutti i decimali vengono ignorati.

Nota: per sapere se una scheda della linea del motore specifica supporta il supporto MDRR, vedere Supporto MDRR per tipo di motore.

Coda di priorità MDRR

Cisco serie 12000 supporta una coda di priorità (PQ) all'interno di MDRR. Questa coda fornisce il ritardo ridotto e il jitter ridotto richiesti dal traffico sensibile al tempo, ad esempio VoIP (Voice over IP).

Come accennato in precedenza, Cisco serie 12000 non supporta WFQ (Weighted Fair Queueing). Pertanto, il modulo PQ all'interno del modulo MDRR funziona in modo diverso dalla funzionalità LLQ (Low Latency Queueing) del software Cisco IOS disponibile per altre piattaforme.

Una differenza fondamentale è rappresentata dal modo in cui lo scheduler MDRR può essere configurato per la manutenzione di PQ in una delle due modalità elencate nella tabella 1:

Tabella 1 - Come configurare lo scheduler MDRR per la manutenzione di PQ in due modalità

	Modalità alternativa	Modalità di priorità rigorosa
Vantaggi	In questo caso, il PQ viene servito tra le altre code. In altre parole, l'utilità di pianificazione MDRR gestisce in alternativa la coda PQ e le altre code configurate.	In questo caso, il PQ viene servito ogni volta che non è vuoto. In questo modo si ottiene il minor ritardo possibile per la corrispondenza del traffico.
Svantaggi	Questa modalità può introdurre jitter e ritardo rispetto alla modalità di priorità rigorosa.	Questa modalità può determinare la morte di altre code, in particolare se i flussi corrispondenti sono mittenti aggressivi.

La modalità Alternativa consente di esercitare un minore controllo su variazione e ritardo. Se l'utilità di pianificazione MDRR inizia a servire frame di dimensioni MTU da una coda di dati e quindi arriva un pacchetto vocale nel PQ, l'utilità di pianificazione in modalità alternativa serve completamente la coda non prioritaria fino a quando il contatore del deficit raggiunge zero. Durante questo periodo, il PQ non viene servito e i pacchetti VoIP vengono ritardati.

Al contrario, in modalità di priorità rigorosa, l'utilità di pianificazione gestisce solo il pacchetto non prioritario corrente e quindi passa al PQ. L'utilità di pianificazione avvia la gestione di una coda non prioritaria solo dopo che la coda PQ è diventata completamente vuota.

Èimportante notare che la coda di priorità in modalità di priorità alternativa viene servita più di una volta in un ciclo e richiede pertanto una larghezza di banda maggiore rispetto alle altre code con lo stesso peso nominale. Quanto altro è una funzione del numero di code definite. Ad esempio, con tre code, la coda a bassa latenza viene servita due volte più spesso delle altre code e invia il doppio del suo peso per ciclo. Se vengono definite otto code, la coda a bassa latenza viene servita sette volte più spesso e il peso effettivo è sette volte superiore. Pertanto, la larghezza di banda che la coda può assumere dipende dalla frequenza con cui viene servita per round robin, che a sua volta dipende dal numero complessivo di code definite. In modalità di priorità alternativa, la coda di priorità è in genere configurata con un peso ridotto per questo particolare motivo.

Si supponga, ad esempio, che siano definite quattro code: 0, 1, 2 e la coda di priorità. In modalità di priorità alternativa, se tutte le code sono congestionate, verranno servite nel modo seguente: **0, llq, 1, llq, 2, llq, 0, llq, 1, ...** dove llq indica la coda a bassa latenza.

Ogni volta che una coda viene servita, può inviare fino al peso configurato. Pertanto, la larghezza di banda minima che la coda a bassa latenza può avere è:

- WL = peso della coda a bassa latenza.
- W0, W1, ... Wn = pesi delle code DRR normali.
- n = numero di code DRR regolari utilizzate per questa interfaccia.
- BW = larghezza di banda del collegamento.

Per la modalità di priorità alternativa, la larghezza di banda minima della coda a bassa latenza = **BW * n * WL / (n * WL + Sum(W0,Wn))**.

Il valore weight è l'unico parametro variabile in MDRR che può essere configurato. Influisce sulla quantità relativa di larghezza di banda che una classe di traffico può utilizzare e sulla quantità di traffico inviato in un turno. L'uso di pesi maggiori significa che il ciclo complessivo richiede più tempo e può aumentare la latenza.

Linee guida per la configurazione

- Èpreferibile configurare il peso della classe con il requisito di larghezza di banda più basso a 1 in modo da mantenere il ritardo e l'effetto jitter il più basso possibile tra le altre classi.
- Selezionare valori di rilevanza il più bassi possibile. Iniziare con un peso di 1 per la classe con la larghezza di banda più bassa. Ad esempio, quando si utilizzano due classi con una larghezza di banda del 50% per ciascuna classe, è necessario configurare 1 e 1. Non ha senso utilizzare 10 e 10, perché non vi è alcun impatto sulle prestazioni quando si sceglie 1. Inoltre, un peso maggiore introduce una maggiore variazione.
- Un valore di peso basso per l'LLQ è molto importante, soprattutto in modalità alternativa per non aggiungere troppo ritardo o jitter alle altre classi.

Esempio di MDRR

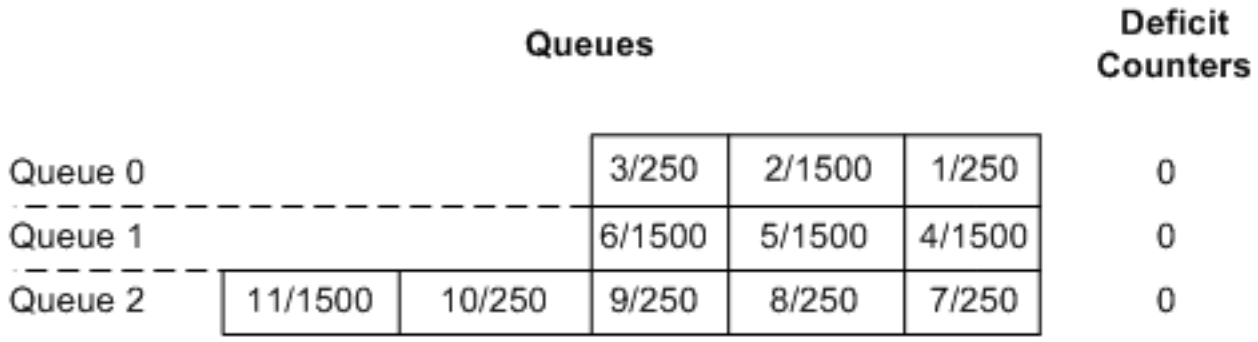
L'esempio in questa sezione è tratto da *Inside Cisco IOS® Software Architecture*, Cisco Press.

Si supponga di avere tre code:

- **Coda 0** - ha un quantum di 1500 byte; si tratta della coda a bassa latenza, configurata per funzionare in modalità alternativa.
- **Coda 1**: ha un quantum di 3000 byte.
- **Coda 2**: ha un quantum di 1500 byte.

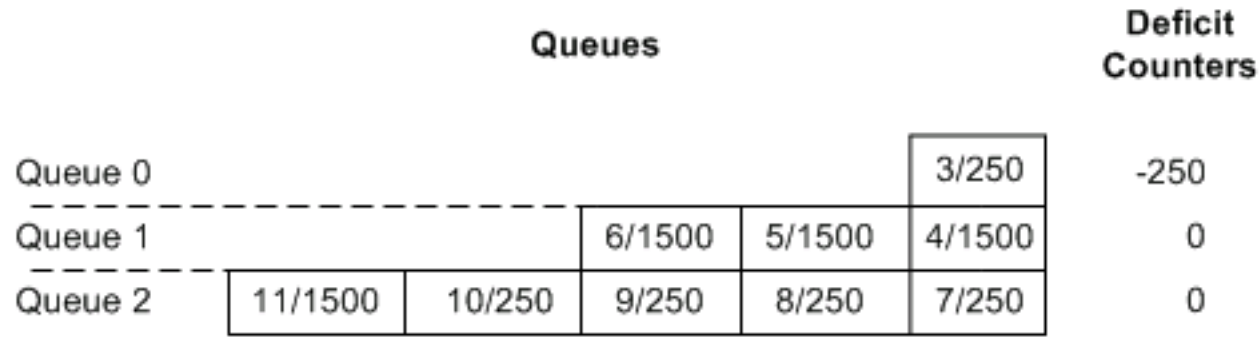
La Figura 1 mostra lo stato iniziale delle code e di alcuni pacchetti ricevuti e accodati.

Figura 1 - Stato iniziale MDRR



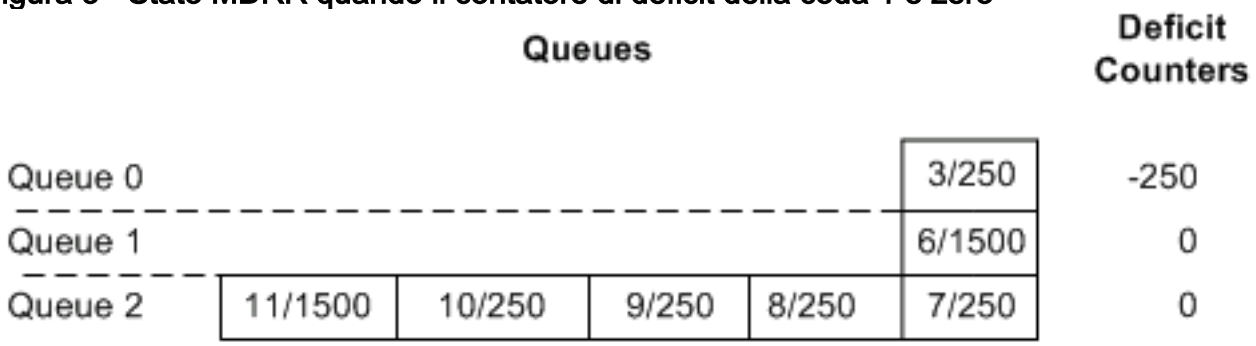
La coda 0 viene servita per prima, il suo quantum viene aggiunto al suo contatore deficit, il pacchetto 1, che è 250 byte, viene trasmesso e la sua dimensione viene sottratta dal contatore deficit. Poiché il contatore di deficit della coda 0 è ancora maggiore di 0 (1500 - 250 = 1250), viene trasmesso anche il pacchetto 2 e la sua lunghezza sottratta dal contatore di deficit. Il contatore di deficit della coda 0 è ora -250, quindi la coda 1 verrà servita successivamente. La Figura 2 indica questo stato.

Figura 2 - Stato successivo MDRR



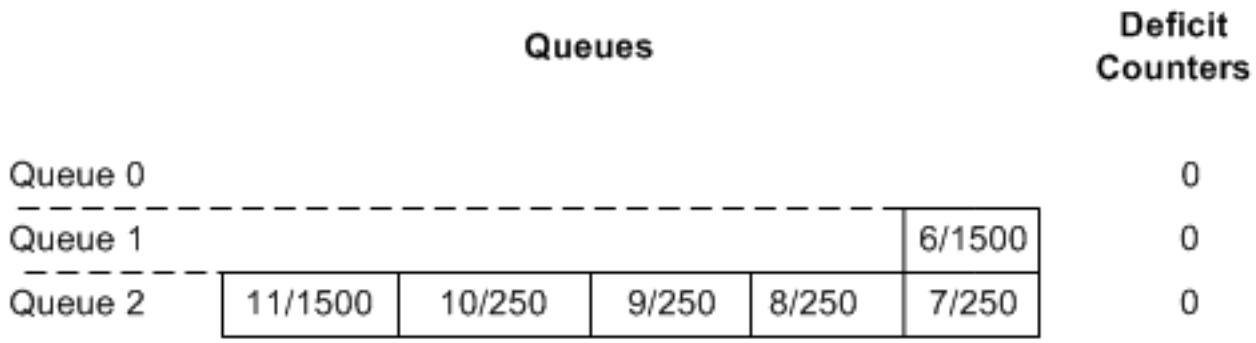
Il contatore di deficit della coda 1 è impostato su 3000 (0 + 3000 = 3000) e vengono trasmessi i pacchetti 4 e 5. Con ciascun pacchetto trasmesso, sottrarre le dimensioni del pacchetto dal contatore del deficit, in modo che il contatore del deficit della coda 1 venga ridotto a 0. La Figura 3 illustra questo stato.

Figura 3 - Stato MDRR quando il contatore di deficit della coda 1 è zero



Ènecessario tornare dalla modalità di priorità alternativa alla coda di servizio 0. Di nuovo, il quantum viene aggiunto al contatore di deficit corrente e il contatore di deficit della coda 0 viene impostato sul risultato (-250 + 1500 = 1250). Il pacchetto 3 è ora trasmesso perché il contatore del deficit è maggiore di 0 e la coda 0 è ora vuota. Quando una coda viene svuotata, il relativo contatore di deficit è impostato su 0, come mostrato nella Figura 4.

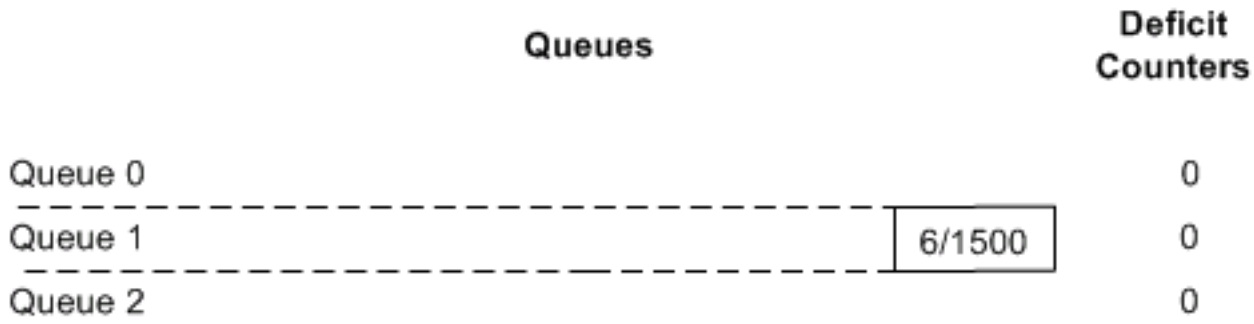
Figura 4 - Stato MDRR quando una coda viene svuotata



La coda 2 viene servita successivamente; il contatore del deficit è impostato su 1500 (0 + 1500 = 1500). Vengono trasmessi i pacchetti da 7 a 10, lasciando il contatore del deficit a 500 (1500 - (4*250) = 500). Poiché il contatore del deficit è ancora maggiore di 0, viene trasmesso anche il pacchetto 11.

Quando si trasmette il pacchetto 11, la coda 2 è vuota e il relativo contatore del deficit è impostato su 0, come mostrato nella Figura 5.

Figura 5 - Stato MDRR durante la trasmissione del pacchetto 11



La coda 0 verrà servita di nuovo in seguito (perché è attiva la modalità di priorità alternativa). Poiché è vuoto, verrà eseguita la coda di servizio 1 e il pacchetto 6 verrà trasmesso.

Supporto MDRR per tipo di motore

Cisco serie 12000 supporta cinque modelli di schede di linea con architetture motore di inoltro Layer 3 (L3) esclusive. Il supporto per MDRR varia in base al tipo di motore L3 e al tipo di scheda. Ad esempio, non è supportato MDRR e WRED sulle schede di linea Engine 0 ATM. È possibile utilizzare il comando **show diag** per determinare il tipo di motore L3 delle schede di linea installate:

```
router#show diags | include (SLOT | Engine)
!--- The regular expression is case-sensitive. ... SLOT 1 (RP/LC 1 ): 1 port ATM Over SONET
OC12c/STM-4c Multi Mode L3 Engine: 0 - OC12 (622 Mbps) SLOT 3 (RP/LC 3 ): 3 Port Gigabit
Ethernet L3 Engine: 2 - Backbone OC48 (2.5 Gbps)
```

MDRR su code ToFab (Rx)

Per configurare MDRR su Cisco serie 12000, è possibile usare la "Sintassi CoS legacy" o l'"Interfaccia della riga di comando QoS modulare". Nelle sezioni più recenti di questo documento viene descritto come configurare MDRR con CoS legacy o QoS modulare. È necessario configurare le code ToFab con la sintassi CoS legacy solo perché non supportano la sintassi più recente di MQC. Per maggiori informazioni, vedere la tabella 2.

Tabella 2 - Dettagli su MDRR nelle code ToFab (Rx)

	Dove implementato	ToFab MDRR	PQ alternativo ToFab	ToFab Strict PQ	ToFab WRED
Ing0	Software	No**	No**	Sì	Sì
Ing1	-	No	No	No	No
Ing2	Hardware	Sì	Sì	Sì	Sì
Ing3	Hardware	No	Sì	Sì	Sì
Ing4	Hardware	Sì	Sì	Sì	Sì

Ing4+	Hardware	Sì	Sì	Sì	Sì

** La modalità MDRR è supportata sulle schede LC del motore 0 in direzione ToFab (Rx), ma solo in modalità di priorità rigorosa e non in modalità di priorità alternativa. Le sette code rimanenti sono supportate normalmente.

Le interfacce in entrata mantengono una coda di output virtuale separata per LC di destinazione. La modalità di implementazione delle code dipende dal tipo di motore L3.

- Motore 0 - I LC in entrata mantengono otto code per slot di destinazione. Pertanto, il numero massimo di code è 16x8 = 128. Ogni coda può essere configurata separatamente.
- Motori 2, 3, 4 e 4+: i LC in entrata mantengono otto code per interfaccia di destinazione. Con 16 slot di destinazione e 16 interfacce per slot, il numero massimo di code è 16x16x8 = 2048. Tutte le interfacce in uno slot di destinazione devono utilizzare gli stessi parametri.

MDRR su code FrFab (Tx)

La funzione MDRR sulle code FrFab funziona in modo coerente sia nell'hardware che nel software. Tutti i tipi di motore L3 supportano otto code di classe per ogni interfaccia in uscita. Per ulteriori informazioni, vedere la tabella 3.

Tabella 3 - Dettagli su MDRR nelle code FrFab (Tx)

	Dove implementato	PQ alternativo FrFab	PQ FrFab Strict	FrFab WRED
Ing0	Software[1]	No	Sì	Sì
Ing1	-	No	No	No
Ing2	Hardware	Sì[2]	Sì	Sì
Ing3	Hardware	No	Sì	Sì
Ing4	Hardware	Sì	Sì	Sì
Ing4+	Hardware	Sì	Sì	Sì

[1]Il supporto di MDRR sulle code FrFab delle licenze CLI Engine 0 è introdotto nelle seguenti versioni del software Cisco IOS:

- Software Cisco IOS release 12.0(10)S - 4xOC3 e 1xOC12 POS, 4xOC3 e 1xCHOC12/ STM4.
- Software Cisco IOS release 12.0(15)S - 6xE3 e 12xE3.
- Software Cisco IOS release 12.0(17)S - 2xCHOC3/STM1.

[2]È necessario configurare un MDRR alternativo nella direzione FrFab con la sintassi CoS legacy.

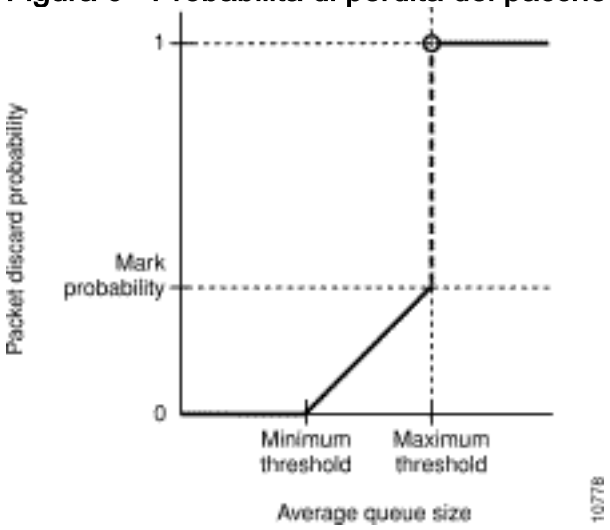
Nota: il controller di dominio 3xGE supporta MDRR sulle code ToFab e, a partire dal software Cisco IOS versione 12.0(15)S, sulle code FrFab con due restrizioni, ovvero un quantum fisso e una singola coda CoS per ciascuna interfaccia. La coda di priorità supporta un quantum configurabile e modalità di priorità sia rigida che alternativa. Tutte e tre le interfacce condividono un unico PQ.

Nota: per le informazioni più recenti sulle funzionalità QoS supportate sui Cisco serie 12000 LC, vedere le note di versione dei router Cisco serie 12000.

Panoramica di WRED

Weighted Random Early Detection (WRED) è progettato per prevenire gli effetti dannosi della congestione dell'interfaccia sul throughput della rete.

Figura 6 - Probabilità di perdita del pacchetto WRED



Per una spiegazione dei parametri WRED, vedere Weighted Random Early Detection sul router Cisco serie 12000. I parametri di probabilità minimum, maximum e mark descrivono la curva RED (Random Early Detection) effettiva. Quando la media ponderata della coda è inferiore alla soglia minima, non viene scartato alcun pacchetto. Quando la media ponderata della coda è superiore alla soglia massima, tutti i pacchetti vengono scartati finché la media non scende sotto la soglia massima. Quando la media è compresa tra la soglia minima e la soglia massima, la probabilità che il pacchetto venga scartato può essere calcolata su una linea retta che va dalla soglia minima (la probabilità di caduta è 0) alla soglia massima (la probabilità di caduta è uguale al denominatore di probabilità 1/mark).

La differenza tra RED e WRED è che WRED può eliminare selettivamente il traffico con priorità inferiore quando l'interfaccia inizia a diventare congestionata e può fornire caratteristiche di prestazioni differenziate per classi di servizio (CoS) diverse. Per impostazione predefinita, WRED utilizza un profilo RED diverso per ogni peso (IP precedence - 8 profili). Perde pacchetti meno importanti in modo più aggressivo rispetto ai pacchetti più importanti.

Èdifficile regolare i parametri WRED per gestire la profondità della coda e dipende da molti fattori, tra cui:

- Carico di traffico e profilo offerti.
- Rapporto tra carico e capacità disponibile.
- Comportamento del traffico in presenza di congestione.

Questi fattori variano a seconda della rete e, a loro volta, dipendono dai servizi offerti e dai clienti che li utilizzano. Di conseguenza, non possiamo fornire suggerimenti applicabili a specifici ambienti dei clienti. Tuttavia, la tabella 4 descrive i valori generalmente consigliati in base alla larghezza di banda del collegamento. In tal caso, non distinguiamo le caratteristiche di rilascio tra le diverse classi di servizio.

Tabella 4 - Valori consigliati in base alla larghezza di banda del collegamento

Larghezza di	BW teorico	Physical BW[1]	Soglia minima	Soglia massima

banda	(kbps)	(kbps)		
OC3	155000	149760	94	606
OC12	622000	599040	375	2423
OC48	2400000	239616	1498	9690
OC192	10000000	9584640	5991	38759

[1]Velocità fisica SONET

Per calcolare i valori di soglia sopra indicati vengono presi in considerazione diversi vincoli. Ad esempio, se si massimizza l'utilizzo del collegamento riducendo al minimo la profondità media della coda, la differenza tra Massimo e Minimo deve essere una potenza di due (a causa di limitazioni hardware). In base all'esperienza e alla simulazione, la profondità istantanea massima di una coda controllata da RED è inferiore a 2 MaxTh. Per 0C48 e versioni successive, 1 MaxTh e così via. Tuttavia, l'esatta determinazione di questi valori esula dalle finalità del presente documento.

Nota: non è necessario configurare il valore della costante di ponderazione esponenziale sulle schede di linea del motore 2 e superiori, poiché viene utilizzato il campionamento della coda hardware. Per i LC del motore 0, è possibile configurare i seguenti valori:

- ds3 - 9
- oc3-10
- oc12-12

Nota: WRED non è supportato sui LC del motore 1.

Come illustrato nelle sezioni seguenti, per configurare WRED è possibile utilizzare sia la sintassi CoS legacy che la sintassi MQC più recente.

Usa sintassi CoS legacy per la configurazione

La sintassi Cisco serie 12000 legacy Class of Service (CoS) utilizza un modello **cos-queue-group** per definire un insieme di definizioni CoS. Il modello verrà quindi applicato rispettivamente alle code ToFab e ForFab su interfacce in entrata o in uscita.

Passaggio 1: Definire un gruppo coda di costi

Il comando **cos-queue-group** crea un modello denominato di parametri MDRR e WRED. Di seguito sono riportati i parametri di configurazione disponibili nella CLI:

```
Router(config)#cos-queue-group oc12
Router(config-cos-que)#?
Static cos queue commands:

default                           Set a command to its defaults
dscp                              Set per DSCP parameters, Engine 3 only
exit                              Exit from COS queue group configuration mode
exponential-weighting-constant    Set group's RED exponential weight constant.
                                  (Not used by engine 0, 1 or 2 line cards)
no                                Negate a command or set its defaults
precedence                        Set per precedence parameters
queue                             Set individual queue parmeters
```

```
random-detect-label               Set RED drop criteria
traffic-shape                     Enable Traffic Shaping on a COS queue group
```

Con MDRR, è possibile mappare la precedenza IP alle code MDRR e configurare la coda speciale a bassa latenza. È possibile utilizzare il parametro precedence nel comando **cos-queue-group** per:

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

Èpossibile mappare una particolare precedenza IP a una coda MDRR regolare (coda da 0 a 6) oppure alla coda di priorità. Il comando precedente consente di mappare diverse precedenza IP alla stessa coda.

Nota: si consiglia di utilizzare la precedenza 5 per la coda a bassa latenza. La priorità 6 viene utilizzata per gli aggiornamenti del routing.

Èpossibile assegnare a ciascuna coda MDRR un peso relativo, con una delle code nel gruppo definita come coda di priorità. A tale scopo, è possibile utilizzare il comando **queue** nel gruppo **cos-queue-**group:

```
queue <0-6> <1-2048>
queue low-latency [alternate-priority | strict-priority] <1-2048>
!--- The weight option is not available with strict priority.
```

Utilizzare il comando **cos-queue-group** per definire eventuali parametri WRED:

```
random-detect-label
```

Di seguito è riportato un esempio di **cos-queue-group** denominato **oc12**. Vengono definite tre classi di traffico: coda 0, 1 e bassa latenza. I valori di precedenza IP 0 - 3 vengono mappati alla coda 0, i valori di precedenza 4, 6 e 7 alla coda 1 e la precedenza 5 alla coda a bassa latenza. Alla coda 1 viene assegnata una larghezza di banda maggiore.

Esempio di configurazione

```
cos-queue-group oc12
!--- Creation of cos-queue-group called "oc12".

precedence 0 queue 0
 !--- Map precedence 0 to queue 0. precedence 0 random-
detect-label 0 !--- Use RED profile 0 on queue 0.
precedence 1 queue 0 precedence 1 random-detect-label 0
precedence 2 queue 0 precedence 2 random-detect-label 0
precedence 3 queue 0 precedence 3 random-detect-label 0
!--- Precedence 1, 2 and 3 also go into queue 0.
precedence 4 queue 1 precedence 4 random-detect-label 1
precedence 6 queue 1 precedence 6 random-detect-label 1
precedence 7 queue 1 precedence 7 random-detect-label 1
precedence 5 queue low-latency !--- Map precedence 5 to
special low latency queue. !--- We do not intend to drop
any traffic from the LLQ. We have an SLA !--- that
commits not to drop on this queue. You want to give it
```

```
all !--- the bandwidth it requires. Random-detect-label
0 375 2423 1 !--- Minimum threshold 375 packets, maximum
threshold 2423 packets. !--- Drop probability at maximum
threshold is 1. random-detect-label 1 375 2423 1 queue 1
20 !--- Queue 1 gets MDRR weight of 20, thus gets more
Bandwidth. queue low-latency strict-priority !--- Low
latency queue runs in strict priority mode.
```

Passaggio 2 - Creare una tabella slot-cos per le code ToFab

Per evitare il blocco dell'head of line, le interfacce in entrata su Cisco serie 12000 mantengono una coda di output virtuale per slot di destinazione. Passare a una scheda di linea utilizzando il comando attach ed eseguire il comando execute-on show controller to fab queue (o immettere direttamente il comando execute-on slot 0 show controller to fab queue) per visualizzare queste code di output virtuali. Di seguito è riportato un esempio di output acquisito direttamente dalla console LC. Vedere Come leggere l'output del comando show controller frfab | tofab queue Comandi su un router Cisco serie 12000 Internet.

```
LC-Slot1#show controllers tofab queues
 Carve information for ToFab buffers
  SDRAM size: 33554432 bytes, address: 30000000, carve base: 30029100
  33386240 bytes carve size, 4 SDRAM bank(s), 8192 bytes SDRAM pagesize, 2 carve(s)
  max buffer data size 9248 bytes, min buffer data size 80 bytes
  40606/40606 buffers specified/carved
  33249088/33249088 bytes sum buffer sizes specified/carved
            Qnum    Head    Tail        #Qelem     LenThresh
            ----    ----    ----        ------     ---------
       5 non-IPC free queues:

            20254/20254    (buffers specified/carved), 49.87%, 80 byte data size
            1       17297   17296       20254      65535
            12152/12152    (buffers specified/carved), 29.92%, 608 byte data size
            2       20548   20547       12152      65535
            6076/6076    (buffers specified/carved), 14.96%, 1568 byte data size
            3       32507   38582       6076       65535
            1215/1215    (buffers specified/carved), 2.99%, 4544 byte data size
            4       38583   39797       1215       65535
            809/809    (buffers specified/carved), 1.99%, 9248 byte data size
            5       39798   40606       809        65535
       IPC Queue:
            100/100 (buffers    specified/carved), 0.24%, 4112 byte data size
            30      72      71          100        65535
       Raw  Queue:
            31      0       17302       0          65535

       ToFab Queues:
               Dest
            Slot
            0       0       0           0          65535
            1       0       0           0          65535
            2       0       0           0          65535
            3       0       0           0          65535
            4       0       0           0          65535
            5       0       17282       0          65535
            6       0       0           0          65535
            7       0       75          0          65535
            8       0       0           0          65535
            9       0       0           0          65535
            10      0       0           0          65535
            11      0       0           0          65535
```

```
           12      0       0           0          65535
           13      0       0           0          65535
           14      0       0           0          65535
           15      0       0           0          65535
    Multicast      0       0           0          65535
   LC-Slot1#
```

Utilizzare il comando **slot-table-cos** per mappare un **cos-queue-group** denominato a una coda di output virtuale di destinazione. È possibile configurare un modello **cos-queue-group** univoco per ogni coda di output

```
Router(config)#slot-table-cos table1
Router(config-slot-cos)#destination-slot ?
  <0-15>  Destination slot number
  all     Configure for all destination slots
Router(config-slot-cos)#destination-slot 0 oc48
Router(config-slot-cos)#destination-slot 1 oc48
Router(config-slot-cos)#destination-slot 2 oc48
Router(config-slot-cos)#destination-slot 3 oc48
Router(config-slot-cos)#destination-slot 4 oc12
Router(config-slot-cos)#destination-slot 5 oc48
Router(config-slot-cos)#destination-slot 6 oc48
Router(config-slot-cos)#destination-slot 9 oc3
Router(config-slot-cos)#destination-slot 15 oc48
```

Nota: la configurazione sopra riportata utilizza tre modelli, denominati oc48, oc12 e oc3. La configurazione per il **cos-queue-group** denominato oc12 è quella illustrata al passo 1. Analogamente, configurare oc3 e oc48. Si consiglia di applicare un modello univoco a un set di interfacce basato sulla larghezza di banda e sull'applicazione.

Passaggio 3 - Applicare uno slot-table-cos a un'interfaccia in entrata

Utilizzare il comando **rx-cos-slot** per applicare uno **slot-table-cos** a un LC.

```
Router(config)#rx-cos-slot 0 ?
  WORD  Name of slot-table-cos
Router(config)#rx-cos-slot 0 table1
Router(config)#rx-cos-slot 2 table1
```

Passaggio 4 - Applicare un gruppo di code di costo a un'interfaccia in uscita

Cisco serie 12000 mantiene una coda separata per ciascuna interfaccia in uscita. Per visualizzare queste code, collegarsi alla CLI della scheda di linea. Usare il comando **attach**, quindi eseguire il comando **show controller frame queue**, come mostrato di seguito:

```
LC-Slot1#show controller frfab queue
    ========= Line Card (Slot 2) =======
  Carve information for FrFab buffers
   SDRAM size: 16777216 bytes, address: 20000000, carve base: 2002D100
   16592640 bytes carve size, 0 SDRAM bank(s), 0 bytes SDRAM pagesize, 2 carve(s)
   max buffer data size 9248 bytes, min buffer data size 80 bytes
   20052/20052 buffers specified/carved
   16581552/16581552 bytes sum buffer sizes specified/carved
            Qnum    Head       Tail              #Qelem  LenThresh
```

```
     ----    ----       ----               ------  ---------
5 non-IPC free queues:
     9977/9977 (buffers    specified/carved), 49.75%, 80 byte data size
     1       101        10077              9977    65535
     5986/5986 (buffers    specified/carved), 29.85%, 608 byte data size
     2       10078      16063              5986    65535
     2993/2993 (buffers    specified/carved), 14.92%, 1568 byte data size
     3       16064      19056              2993    65535
     598/598 (buffers    specified/carved), 2.98%, 4544 byte data size
     4       19057      19654              598     65535
     398/398 (buffers    specified/carved), 1.98%, 9248 byte data size
     5       19655      20052              398     65535
IPC Queue:
     100/100 (buffers    specified/carved), 0.49%, 4112 byte data size
     30      77         76                 100     65535
Raw Queue:
     31      0          82                 0       65535
Interface Queues:
     0       0          0                  0       65535
     1       0          0                  0       65535
     2       0          0                  0       65535
     3       0          0                  0       65535
```

Utilizzare il comando **tx-cos** per applicare un modello **cos-queue-group** a una coda di interfaccia. Come mostrato di seguito, il parametro impostato viene applicato direttamente all'interfaccia. non sono necessarie tabelle. In questo esempio, *pos48* è il nome di un set di parametri.

```
Router(config)#interface POS 4/0
Router(config-if)#tx-cos ?
  WORD  Name of cos-queue-group
Router(config-if)#tx-cos pos48
```

Utilizzare il comando **show cos** per confermare la configurazione:

```
Router#show cos
!--- Only some of the fields are visible if MDRR is configured on Inbound !--- or Outbound
interfaces. Interface Queue cos Group Gi4/0 eng2-frfab !--- TX-cos has been applied. Rx Slot
Slot Table 4 table1 !--- rx-cos-slot has been applied. Slot Table Name - table1 1 eng0-tofab 3
eng0-tofab !--- slot-table-cos has been defined. cos Queue Group - eng2-tofab !--- cos-queue-
group has been defined. Prec Red Label [min, max, prob] Drr Queue [deficit] 0 0 [6000, 15000,
1/1] 0 [10] 1 1 [10000, 20000, 1/1] 1 [40] 2 1 [10000, 20000, 1/1] 1 [40] 3 1 [10000, 20000,
1/1] 0 [10] 4 2 [15000, 25000, 1/1] 2 [80] 5 2 [15000, 25000, 1/1] 2 [80] 6 no drop low latency
7 no drop low latency
```

Nota: la CLI precedente utilizza anche la sintassi di precedenza per il traffico Multiprotocol Label Switching (MPLS). Il router tratta i bit MPLS come se fossero bit del tipo di servizio IP (ToS) e mette i pacchetti appropriati nelle code corrette. Questo non è assolutamente vero per MQC. QoS MPLS non è compreso nell'ambito di questo documento.

Usa Modular QoS CLI (MQC) per la configurazione

L'obiettivo di MQC (Modular QoS CLI) di Cisco è quello di connettere tutte le diverse funzionalità QoS in modo logico, al fine di semplificare la configurazione delle funzionalità QoS (Quality of Service) del software Cisco IOS. Ad esempio, la classificazione viene eseguita separatamente dalle operazioni di accodamento, monitoraggio e shaping. Fornisce un'unica struttura di configurazione per QoS basata su modelli. Di seguito sono riportati alcuni punti da ricordare

relativi alla configurazione di MQC:

- Può essere facilmente applicato a un'interfaccia e rimosso.
- Può essere riutilizzato facilmente (lo stesso criterio può essere applicato a più interfacce).
- Offre un'unica struttura di configurazione per QoS che consente di eseguire facilmente il provisioning, il monitoraggio e la risoluzione dei problemi.
- Fornisce un livello più alto di astrazione.
- Èindipendente dalla piattaforma.

Sui Cisco serie 12000, è possibile usare i comandi MQC anziché la sintassi CoS (Class of Service) precedente.

Il supporto di MQC sulla serie 12000 non implica che la stessa funzionalità QoS disponibile su un'altra piattaforma, ad esempio la serie 7500, sia già disponibile su Cisco 12000. MQC fornisce una sintassi comune in cui un comando dà luogo a una funzione o a un comportamento condiviso. Ad esempio, il comando **bandwidth** implementa una garanzia di larghezza di banda minima. Cisco serie 12000 utilizza MDRR come meccanismo di pianificazione per riservare la larghezza di banda, mentre Cisco serie 7500 utilizza WFQ. L'algoritmo principale completa la piattaforma specifica.

Èimportante sottolineare che solo le code FrFab supportano la configurazione delle funzionalità QoS tramite MQC. Poiché le code ToFab sono code di output virtuali e non vere, non sono supportate da MQC. Devono essere configurati con comandi CoS legacy.

La tabella 5 elenca il supporto per il tipo di motore MQC per L3.

Tabella 5 - Supporto di MQC per i tipi di motore L3

Tipo di motore L3	Motore 0	Motore 1	Motore 2	Motore 3	Motore 4	Engine 4+
Supporto MQC	Sì	No	Sì	Sì	Sì	Sì
Release IOS	12.0(15)S	-	12.0(15)S[1]	12.0(21)S	12.0(22)S	12.0(22)S

[1]Ricordare le seguenti eccezioni con il supporto MQC sulle schede di linea Engine 0 e 2 (LC):

- 2xCHOC3/STM1 - Introdotto in 12.0(17)S.
- 1xOC48 DPT - Introdotto in 12.0(18)S.
- 8xOC3 ATM - Pianificato per la versione 12.0(22)S.

MQC utilizza i tre passaggi seguenti per creare un criterio QoS:

1. Definire una o più classi di traffico con il comando **class-map**.
2. Creare un criterio QoS con il comando **policy-map** e assegnare le azioni QoS, ad esempio la larghezza di **banda** o la **priorità**, a una classe di traffico denominata.
3. Utilizzare il comando **service-policy** per associare una mappa dei criteri alla coda FrFab di un'interfaccia in uscita.

Utilizzare il comando **show policy-map interface** per monitorare i criteri.

Per ulteriori informazioni, vedere Cenni preliminari sull'interfaccia della riga di comando Modular

Quality of Service.

Passaggio 1 - Definizione delle mappe di classe

Il comando **class-map** viene usato per definire le classi del traffico. Internamente, sui Cisco serie 12000, il comando **class-map** assegna una classe a una coda CoS specifica sulla scheda di linea (vedere il passaggio 4 per i dettagli).

Il comando **class-map** supporta "match-any", che inserisce nella classe i pacchetti che corrispondono a una delle istruzioni match, e "match-all", che inserisce i pacchetti in questa classe solo quando tutte le istruzioni sono true. Questi comandi creano una classe denominata "Prec_5" e classificano tutti i pacchetti con una precedenza IP pari a 5 per questa classe:

```
Router(config-cmap)#match ?
  access-group         Access group
  any                  Any packets
  class-map            Class map
  destination-address  Destination address
  fr-dlci              Match on fr-dlci
  input-interface      Select an input interface to match
  ip                   IP specific values
  mpls                 Multi Protocol Label Switching specific values
  not                  Negate this match result
  protocol             Protocol
  qos-group            Qos-group
  source-address       Source address
Router(config-cmap)#match ip precedence 5
```

Nella tabella 6 sono elencati i criteri di corrispondenza supportati per ogni tipo di motore L3.

Tabella 6 - Criteri di corrispondenza supportati per i motori L3

	Motore 0, 2	Motore 3	Motore 4	Engine 4+
precedenza ip	Sì	Sì	Sì	Sì [1]
access-group	No	Sì	No	No
mpls exp	No	Sì	No	Sì (12.0.26S)
ip dscp	No	Sì	No	Sì (12.0.26S)
qos-group	No	Sì	No	No
corrispondenza interfacce di ingresso POS x/y	No	Sì (solo come criterio di ricezione)	No	No

[1] ingresso/uscita da 12.0.26S

Passaggio 2 - Creare una mappa dei criteri

Il comando **policy-map** viene utilizzato per assegnare i criteri o le azioni di gestione dei pacchetti a una o più classi definite. Ad esempio, quando si assegna una prenotazione della larghezza di banda o si applica un profilo di rilascio casuale.

Cisco serie 12000 supporta un sottoinsieme di funzionalità MQC basate sull'architettura ad alta velocità dei motori L3. Nella tabella 7 sono elencati i comandi supportati:

Tabella 7 - Comandi supportati

Comando	Descrizione
larghezza di banda	Garantisce una larghezza di banda minima durante i periodi di congestione. Viene specificato come percentuale della velocità di collegamento o come valore assoluto. Se una classe non utilizza o non richiede una larghezza di banda uguale a quella riservata, la larghezza di banda disponibile può essere utilizzata da altre classi di larghezza di banda.
polizia, forma	Limita la quantità di traffico che una classe può trasmettere. Questi comandi hanno una funzione leggermente diversa. Il comando **Police** identifica il traffico che supera la larghezza di banda configurata e lo scarta o lo contrassegna. Il comando **shape** memorizza nel buffer l'eventuale traffico in eccesso e lo pianifica per la trasmissione a una velocità costante, ma non restituisce drop o note.
Queue-limit	Assegna una lunghezza di coda fissa a una determinata classe di traffico. È possibile specificare il numero di pacchetti che possono essere contenuti nella coda.
priority	Identifica una coda come coda a bassa latenza. MQC supporta la modalità rigorosa solo per PQ. Modalità alternativa non supportata tramite MQC. Utilizzare il comando **priority** senza un valore percentuale per abilitare la modalità di priorità rigida. **Nota:** l'implementazione del comando **priority** su Cisco serie 12000 è diversa dall'implementazione su altri router con software Cisco IOS. In questa piattaforma, il traffico con priorità non è limitato al valore configurato in kbps durante i periodi di congestione. Di conseguenza, è necessario anche configurare il comando **Police** per limitare la larghezza di banda che una classe prioritaria può utilizzare e garantire una larghezza di banda adeguata per altre classi. Al momento, il comando **Police** è supportato solo sulle schede di linea Engine 3.

	Nelle altre schede della linea del motore, quando si configura una classe di priorità è consentito utilizzare solo la classe predefinita.
rilevamento casuale	Assegna un profilo WRED. Utilizzare il comando **random-detect precedence** per configurare i valori WRED non predefiniti per ciascun valore di precedenza IP.

Sugli LC del motore 3, è necessario configurare le code FrFab con MQC (Modular QoS CLI); CLI (Command Line Interface) legacy non supportato.

Quando si configura il comando **bandwidth**, notare che i LC del motore 0 e 2 supportano solo sei classi di larghezza di banda. È possibile utilizzare una settima classe per il servizio a bassa latenza e un'ottava classe, che è la classe predefinita, per tutto il traffico non corrispondente. In totale, pertanto, sono presenti otto code. Class-default non viene utilizzato come classe di priorità.

Sugli LC del motore 3, il comando **bandwidth percent** viene convertito in un valore kbps, che varia con la velocità dei collegamenti sottostanti, e quindi configurato direttamente sulla coda. La precisione di questa garanzia di larghezza di banda minima è 64 kbps.

Sebbene non venga eseguita alcuna conversione in un valore quantum con il comando **bandwidth**, tutte le code dispongono di un quantum. Sugli LC del motore 3, il valore quantum viene impostato internamente in base alla MTU (Maximum Transmission Unit) dell'interfaccia ed è impostato in modo uguale per tutte le code. Non esiste alcun meccanismo MQC CLI per modificare questo valore quantistico, direttamente o indirettamente. Il valore quantum deve essere maggiore o uguale all'MTU dell'interfaccia. Internamente, il valore quantistico è in unità di 512 byte. Pertanto, con una MTU di 4470 byte, il valore quantistico minimo dell'MTU deve essere 9.

MDRR su LC Engine 3

In questa sezione vengono fornite le note di configurazione per implementare WRED e MDRR sui LC del motore 3.

- La larghezza di banda MDRR configurata nella CLI viene convertita in una quantità corrispondente a L2 (ad esempio, il sovraccarico L1 viene rimosso). Tale quantità viene quindi arrotondata ai successivi 64 kbps e programmata nell'hardware.
- Per una classe sono supportati tre diversi profili WRED.
- Il valore WRED (soglia massima - soglia minima) viene approssimato alla potenza più vicina di 2. La soglia minima viene quindi regolata automaticamente mentre la soglia massima rimane invariata.
- Il valore di probabilità contrassegno 1 è supportato.
- La configurazione delle costanti di ponderazione esponenziali non è supportata.
- Sono supportati i valori IP Precedence, MPLS EXP e DSCP.

Nota: per impostazione predefinita, a ciascuna porta o canale delle schede di linea Tetra (4GE-SFP-LC=) o CHOC12/DS1-IR-SC= Frostbite sono assegnate quattro code. Le quattro code sono costituite dalle seguenti:

- Classe LLQ (One Priority Queue)
- Una classe di coda predefinita
- Due classi normali non prioritarie

Quando si applica una policy di servizio contenente più di queste quattro classi (1 HPQ, 2 LPQ e class-default) all'interfaccia, viene segnalato il seguente errore:

Router(config-if)#service-policy output mdrr-policy

% Risorse di accodamento disponibili insufficienti per soddisfare la richiesta.

A partire dalla versione 12.0(26)S, è stato aggiunto un comando per la scheda di linea 4GE-SFP-LC= Tetra che consente la configurazione di otto code/VLAN invece di quattro. Le otto code sono le seguenti:

- One LLQ
- Una coda di classe predefinita
- Sei code normali

L'uso di questo comando richiederà un ricaricamento del microcodice della scheda di linea e consentirà di configurare solo 508 VLAN invece di 1022. La sintassi del comando è la seguente:

[no] slot hw-module <slot#> code interfaccia qos 8

Ad esempio:

Router(config)#hw-module slot 2 qos interface queue 8

Avviso: Ricaricare la scheda di linea per rendere effettivo il comando

Router(config)#microcode reload 2

Questo comando sarà disponibile per la scheda di linea CHOC12/DS1-IR-SC= Frostbite in 12.0(32)S

Esempio 1 - Comando bandwidth percent

In questo esempio viene allocato il 20% della larghezza di banda disponibile al traffico Prec_4 della classe e il 30% al traffico della classe Prec_3. Il restante 50% viene assegnato alla classe predefinita della classe.

Inoltre, configura WRED come meccanismo di rilascio su tutte le classi di dati.

Esempio 1 - percentuale larghezza di banda

```
policy-map GSR_EXAMPLE
 class Prec_4
  bandwidth percent 20
  random-detect
  random-detect precedence 4 1498 packets 9690 packets 1
!--- All data classes should have WRED configured. class
Prec_3 bandwidth percent 30 random-detect random-detect
precedence 3 1498 packets 9690 packets 1 class class-
default !--- Class-default uses any leftover bandwidth.
random-detect random-detect precedence 2 1498 packets
9690 packets 1 random-detect precedence 1 1498 packets
9690 packets 1 random-detect precedence 0 1498 packets
9690 packets 1
```

Esempio 2 - Comando larghezza di banda {kbps}

In questo esempio viene illustrato come applicare il comando bandwidth come valore kbps assoluto anziché come percentuale.

Esempio 2 - larghezza di banda {kbps}

```
policy-map GSR_EXAMPLE
 class Prec_4
  bandwidth 40000
!--- Configures a minimum bandwidth guarantee of 40000
kbps or 40 Mbps in !--- times of congestion. Random-
detect random-detect precedence 4 1498 packets 9690
packets 1 class Prec_3 bandwidth 80000 !--- Configures a
minimum bandwidth guarantee of 80000 kbps or 80 Mbps in
!--- times of congestion. Random-detect random-detect
precedence 3 1498 packets 9690 packets 1 class class-
default !--- Any remaining bandwidth is given to class-
default. Random-detect random-detect precedence 2 1498
packets 9690 packets 1 random-detect precedence 1 1498
packets 9690 packets 1 random-detect precedence 0 1498
packets 9690 packets 1
```

Esempio 3 - comando priority

Questo esempio è destinato ai provider di servizi che utilizzano il router Cisco serie 12000 come router MPLS provider Edge (PE) e devono configurare un criterio di servizio QoS sul collegamento tra il router PE e il router Customer Edge (CE). Inserisce i pacchetti IP Precedence 5 in una coda di priorità e limita l'output di tale coda a 64 Mbps. Assegna quindi una parte della larghezza di banda rimanente alle classi di larghezza di banda.

Tutte le code delle classi non prioritarie sono configurate con il comando **random-detect** per abilitare WRED come criterio di eliminazione. Per tutte le classi di larghezza di banda e le classi predefinite, WRED deve essere configurato in modo esplicito.

Esempio n. 3 - priorità

```
policy-map foo
  class Prec_5
    police 64000000 conform-action transmit exceed-
action drop
!--- The police command is supported on Engine 3 line
cards. priority class Prec_4 bandwidth percent 30
random-detect random-detect precedence 4 1498 packets
9690 packets 1 class Prec_3 bandwidth percent 10 random-
detect random-detect precedence 3 1498 packets 9690
packets 1 class Prec_2 bandwidth percent 10 random-
detect random-detect precedence 2 1498 packets 9690
packets 1 class Prec_1 bandwidth percent 10 random-
detect random-detect precedence 1 1498 packets 9690
packets 1 class Prec_0 bandwidth percent 25 random-
detect random-detect precedence 0 1498 packets 9690
packets 1 class class-default random-detect random-
detect precedence 6 1498 packets 9690 packets 1 random-
detect precedence 7 1498 packets 9690 packets 1
```

Passaggio 3 - Assegnare una mappa dei criteri a una coda di interfaccia in uscita

Come accennato in precedenza, MQC funziona solo con le code FrFab su un'interfaccia in uscita. Per applicare una mappa dei criteri definita, utilizzare il comando **service-policy output**, come mostrato di seguito:

```
Router(config)#interface POS 0/0
Router(config-if)#service-policy ?
  history  Keep history of QoS metrics
  input    Assign policy-map to the input of an interface
  output   Assign policy-map to the output of an interface
Router(config-if)#service-policy output ?
  WORD  policy-map name
Router(config-if)#service-policy output GSR_EXAMPLE
```

Fase 4 - Monitoraggio e verifica dei criteri del servizio

Utilizzare il comando **show policy-map interface** per visualizzare l'applicazione di un criterio. Il comando **show policy-map interface** visualizza quanto segue:

- Classi di larghezza di banda e priorità configurate e criteri di corrispondenza.
- Qualsiasi profilo WRED.
- Parametri relativi a forma e polizia.
- Contabilità e tariffe del traffico.
- Coda CoS interna a cui è mappata una determinata classe. A queste code fa riferimento lo stesso indice utilizzato nell'output del comando **show controller frfab queue**.

Di seguito è riportato un esempio di configurazione completa e dei comandi **show** per monitorare il criterio:

Configurazione completa

```
class-map match-all class1
   match ip precedence 1
class-map match-all class2
   match ip precedence 2
!--- Step 1 - Configure traffic classes. ! policy-map
policy1e Class class1 bandwidth percent 10 random-detect
random-detect precedence 1 375 packets 2423 packets 1
Class class2 bandwidth percent 20 random-detect !---
Step 2 - Configure a policy-map. ! interface POS6/0 ip
address 12.1.1.1 255.255.255.0 no ip directed-broadcast
no keepalive service-policy output policy1e !--- Step 3-
Attach policy-map to the interface.
```

Utilizzare il comando **show policy-map interface** per visualizzare i criteri configurati sull'interfaccia, insieme a tutte le classi configurate. Di seguito è riportato l'output del comando:

```
Router#show policy-map int pos6/0
 POS6/0

  Service-policy output: policy1e (1071)

    Class-map: class1 (match-all) (1072/3)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: ip precedence 1  (1073)
```

```
  Class of service queue: 1
  Tx Queue (DRR configured)
  bandwidth percent         Weight
    10                      1
  Tx Random-detect:
  Exp-weight-constant: 1 (1/2)
  Precedence        RED Label       Min             Max             Mark
  1                 1               375             2423            1

Class-map: class2 (match-all) (1076/2)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: ip precedence 2  (1077)
  Class of service queue: 2
  Tx Queue (DRR configured)
  bandwidth percent         Weight
    20                      9
  Tx Random-detect:
  Exp-weight-constant: 1 (1/2)
  Precedence        RED Label       Min             Max             Mark

Class-map: class-default (match-any) (1080/0)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any  (1081)
    0 packets, 0 bytes
    5 minute rate 0 bps
```

Comandi per monitorare la gestione e la prevenzione delle congestioni

In questa sezione vengono elencati i comandi che è possibile utilizzare per monitorare la gestione delle congestioni e i criteri di prevenzione.

La Tabella 8 elenca i comandi relativi alle schede di linea in ingresso e in uscita.

Tabella 8 - Comandi per le schede di linea

Scheda linea in ingresso	Scheda linea in uscita
• show interfaces • coda da controller sh a fab slot exec <x> • slot exec <x> show controller to fab queue <slot> <porta> • stato qm show controller tofab slot <x> exec	• show interfaces • show interfaces <y> random • coda frfab show controller slot <y> exec • exec slot <y> show controller frfab queue <porta> • stato QM frame controller show <y> exec slot

Questi comandi vengono spiegati in questa sezione.

Il comando show interfaces

Prima di usare questo comando, confermare la "strategia di accodamento" corretta. Se nell'output viene visualizzato First In, First Out (FIFO), verificare che il comando **service-policy** sia visualizzato nella configurazione in esecuzione (se MQC è stato utilizzato per configurare MDRR).

Monitorare il numero di rilasci di output, che rappresenta il numero totale di rilasci FrFab WRED che si sono verificati per il traffico in uscita su questa interfaccia. Il numero di rilasci di output nell'output del comando **show interfaces** deve essere maggiore o uguale al numero di rilasci di output nell'output del comando **show interfaces<number>random**.

Nota: sui router Cisco serie 12000, le perdite di output dell'interfaccia vengono aggiornate dopo le perdite di output WRED. Se si utilizza uno strumento per eseguire query su entrambi i contatori di rilascio, è possibile che le interfacce rilasciate non vengano ancora aggiornate.

```
Router#show interfaces POS 4/0
POS4/0 is up, line protocol is up
  Hardware is Packet over SONET
  Description: link to c12f9-1
  Internet address is 10.10.105.53/30
  MTU 4470 bytes, BW 622000 Kbit, DLY 100 usec, rely 255/255, load 82/255
  Encapsulation PPP, crc 32, loopback not set
  Keepalive set (10 sec)
  Scramble enabled
  LCP Open
  Open: IPCP, CDPCP, OSICP, TAGCP
  Last input 00:00:02, output 00:00:05, output hang never
  Last clearing of "show interface" counters 00:04:54
  Queueing strategy: random early detection (WRED)
  Output queue 0/40, 38753019 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 200656000 bits/sec, 16661 packets/sec
     135 packets input, 6136 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
              0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     7435402 packets output, 11182627523 bytes, 0 underruns
     0 output errors, 0 applique, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

Il comando show interfaces {number} random

Quando si utilizza questo comando, è necessario:

- Verificare che all'interfaccia sia applicato il modello **cos-queue-group** corretto.
- Controllare i pesi MDRR. Per ogni coda MDRR, è possibile controllare la media ponderata per la lunghezza della coda e il valore massimo raggiunto (in pacchetti). I valori vengono calcolati come media ponderata e non devono riflettere la profondità massima effettiva della coda mai raggiunta.
- Controllare le soglie minime e massime WRED.
- Controllare il numero di cadute casuali e di cadute di soglia per ogni etichetta RED (le cadute "To Fabric" indicano la quantità totale di cadute per questa etichetta su tutte le schede di linea).
- Il contatore "TX-queue-limit drops" viene utilizzato solo su LC del motore 1 che non supportano WRED. Le schede del motore 1 consentono di impostare il limite delle code

MDRR con il comando **TX-queue-limit interface**. Se WRED è supportato, le soglie WRED determinano la profondità delle code MDRR.

```
Router#show interfaces POS 4/0 random
POS4/0
cos-queue-group: oc12
RED Drop Counts

TX Link
To Fabric
RED Label          Random       Threshold          Random       Threshold
0                29065142           73492         9614385               0
1                       0               0               0               0
2                       0               0               0               0
3                       0               0               0               0
4                       0               0               0               0
5                       0               0               0               0
6                       0               0               0               0

TX-queue-limit drops: 0

Queue Lengths

TX Queue (DRR configured) oc12
Queue              Average              High Water Mark         Weight
0                    0.000                 2278.843                1
1                    0.000                    0.000               73
2                    0.000                    0.000               10
3                    0.000                    0.000               10
4                    0.000                    0.000               10
5                    0.000                    0.000               10
6                    0.000                    0.000               10
Low latency          0.000                    0.000               10

TX RED config
  Precedence 0: 375 min threshold, 2423 max threshold, 1/1 mark weight
  Precedence 1: not configured for drop
  Precedence 2: not configured for drop
  Precedence 3: not configured for drop
  Precedence 4: 375 min threshold, 2423 max threshold, 1/1 mark weight
  Precedence 5: not configured for drop
  Precedence 6: 375 min threshold, 2423 max threshold, 1/1 mark weight
  Precedence 7: not configured for drop weight 1/2
```

Il comando show controller frfab queue {port} dello slot di esecuzione (y)

Questo comando visualizza la profondità di coda istantanea per una determinata porta su un determinato slot. L'output di esempio in questa sezione visualizza la coda MDRR sull'interfaccia POS 4/1. Viene visualizzata una profondità per la coda MDRR 1 di 1964 pacchetti. Il peso è il numero di byte che possono essere serviti in questa coda. Questo valore determina la percentuale della larghezza di banda che si desidera assegnare alla coda. Il deficit è il valore che indica all'algoritmo DRR quanti pacchetti devono ancora essere forniti. Si noti che non vi sono pacchetti in coda nella coda LLQ (coda DRR 7).

```
Router#execute-on slot 4 show controllers frfab queue 1
========= Line Card (Slot 4) =======
FrFab Queue
```

```
Interface 1
DRR#    Head    Tail    Length  Average         Weight  Deficit
0       95330   40924   0            0.000      4608    0
1       211447  233337  1964      1940.156     41472    35036
2       0       0       0            0.000      9216    0
3       0       0       0            0.000      9216    0
4       0       0       0            0.000      9216    0
5       0       0       0            0.000      9216    0
6       0       0       0            0.000      9216    0
7       0       0       0            0.000      9216    0
```

Questo comando è usato in particolare per monitorare la profondità della coda di priorità della scheda di linea in uscita. Quando vedi che i pacchetti iniziano ad attendere su questo LLQ, è una buona indicazione che devi deviare del traffico Voice over IP (VOIP) verso altre schede di linea in uscita. In una buona progettazione, la lunghezza dovrebbe sempre essere 0 o 1. In una rete reale, si sperimenterà traffico bursty, anche per i dati vocali. Il ritardo aggiuntivo diventa più grave quando il carico vocale totale supera il 100% della larghezza di banda in uscita per un breve periodo di tempo. Il router non può mettere più traffico sul cavo di quello consentito, quindi il traffico vocale viene accodato sulla propria coda di priorità. Questo crea latenza e jitter vocali introdotti dalla frammentazione del traffico vocale stesso.

```
Router#execute-on slot 4 show controllers frfab queue 0
========= Line Card (Slot 4) =======
FrFab Queue
Interface 0
DRR#    Head    Tail    Length  Average         Weight  Deficit
0       181008  53494   2487       2282.937     4608    249
1       16887   45447   7            0.000      41472   0
2       0       0       0            0.000      9216    0
3       0       0       0            0.000      9216    0
4       0       0       0            0.000      9216    0
5       0       0       0            0.000      9216    0
6       0       0       0            0.000      9216    0
7       107818  142207  93           0.000      9216    -183600
```

La coda 7 è LLQ e la lunghezza indica il numero di pacchetti presenti in LLQ.

Il comando show controller frfab QM stat dello slot exec (y)

Utilizzare questo comando quando si sospetta che la memoria del pacchetto di un LC inizi a raggiungere la piena capacità. Un valore crescente per il contatore "nessuna perdita di memoria" suggerisce che WRED non è configurato o che le soglie WRED sono impostate su un valore troppo alto. In condizioni normali, questo contatore non deve aumentare. Per ulteriori informazioni, vedere Risoluzione dei problemi relativi ai pacchetti ignorati e all'assenza di perdite di memoria su Cisco Internet Router serie 12000.

```
Router#execute-on slot 4 show controllers frfab QM stat
========= Line Card (Slot 4) =======
68142538 no mem drop, 0 soft drop, 0 bump count
0 rawq drops, 8314999254 global red drops, 515761905 global force drops
0 no memory (ns), 0 no memory hwm (Ns)
no free queue
0       0       1968    88
0       0       0       0
0       0       0       0
0       0       0       0
```

```
0 multicast drops
TX Counts
 Interface 0
859672328848 TX bytes, 3908130535 TX pkts, 75431 kbps, 6269 pps
 Interface 1
86967615809 TX bytes, 57881504 TX pkts, 104480 kbps, 8683 PPS
 Interface 2
0 TX bytes, 0 TX pkts, 0 kbps, 0 PPS
 Interface 3
0 TX bytes, 0 TX pkts, 0 kbps, 0 PPS
```

Monitoraggio gestione congestione in ingresso

In questa sezione vengono descritti i comandi utilizzati per monitorare la gestione delle congestioni in entrata.

Il comando show interfaces

Prima di eseguire questo comando, verificare se il valore nel contatore ignorato è in aumento. I pacchetti ignorati vengono visualizzati se si esaurisce la memoria sul lato ToFab o se la scheda di linea non accetta i pacchetti abbastanza velocemente. Per ulteriori informazioni, vedere Risoluzione dei problemi di rilascio di input su Cisco serie 12000 Internet Router.

```
Router#show interfaces POS 14/0
POS14/0 is up, line protocol is up
  Hardware is Packet over SONET
  Description: agilent 3b for QOS tests
  Internet address is 10.10.105.138/30
  MTU 4470 bytes, BW 2488000 Kbit, DLY 100 usec, rely 234/255, load 1/255
  Encapsulation HDLC, crc 32, loopback not set
  Keepalive not set
  Scramble disabled
  Last input never, output 00:00:03, output hang never
  Last clearing of "show interface" counters 00:34:09
  Queueing strategy: random early detection (WRED)
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 2231000 bits/sec, 4149 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     563509152 packets input, 38318622336 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
              0 parity
     166568973 input errors, 0 CRC, 0 frame, 0 overrun, 166568973 ignored, 0 abort
     35 packets output, 12460 bytes, 0 underruns
     0 output errors, 0 applique, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

Il comando show controller to fab queue dello slot exec (x)

Questo output di esempio del comando **show controller to fab queue dello slot exec <x>** è stato acquisito quando non si è verificata alcuna congestione su una scheda di linea in uscita nello slot 3.

```
Router#execute-on slot 13 show controllers tofab queue
```

```
========= Line Card (Slot 13) =======
Carve information for ToFab buffers
!--- Output omitted. ToFab Queues: Dest Slot 0 0 0 0 9690 1 0 0 0 9690 2 0 0 0 9690 3 11419
16812 0 9690 4 0 0 0 2423 5 0 0 0 9690 6 0 0 0 9690 7 0 0 0 262143 8 0 0 0 262143 9 0 0 0 606 10
0 0 0 262143 11 0 0 0 262143 12 0 0 0 262143 13 0 0 0 262143 14 0 0 0 262143 15 0 0 0 9690
Multicast 0 0 0 262143
```

Il seguente output è stato acquisito in caso di congestione nello slot 3:

```
Router#execute-on slot 13 show controllers tofab queue
========= Line Card (Slot 13) =======
Carve information for ToFab buffers
!--- Output omitted. ToFab Queues: Dest Slot 0 0 0 0 9690 1 0 0 0 9690 2 0 0 0 9690 3 123689
14003 1842 9690 4 0 0 0 2423 5 0 0 0 9690 6 0 0 0 9690 7 0 0 0 262143 8 0 0 0 262143 9 0 0 0 606
10 0 0 0 262143 11 0 0 0 262143 12 0 0 0 262143 13 0 0 0 262143 14 0 0 0 262143 15 0 0 0 9690
Multicast 0 0 0 262143
```

Il comando exec slot (x) show controller to fab queue (slot) (porta)

Utilizzare questo comando per determinare la quantità di memoria utilizzata sul lato ToFab. In particolare, annotare il numero nella colonna '#Qelem'. Si noti che:

- Quando non viene utilizzata alcuna memoria, i valori sono ai massimi.
- Il valore della colonna "#Qelem" diminuisce man mano che i pacchetti vengono memorizzati nel buffer.
- Quando la colonna "#Qelem" raggiunge lo zero, tutti i buffer scolpiti sono in uso. Su Engine 2 LC, i piccoli pacchetti possono prendere in prestito lo spazio del buffer dai pacchetti più grandi.

Èinoltre possibile utilizzare questo comando per determinare il numero di pacchetti in coda in una coda di output virtuale. L'esempio mostra come controllare lo slot 14 per il numero istantaneo di pacchetti su queste code per lo slot 4, porta 1 (POS 4/1). Vengono visualizzati 830 pacchetti in coda nella coda MDRR 1.

```
Router# execute-on slot 14 show controllers tofab queue 4 1
========= Line Card (Slot 14) =======
ToFab Queue
Slot 4 Int 1
DRR#    Head    Tail    Length  Average         Weight  Deficit
0       0       0       0            0.000      4608    0
1       203005  234676  830        781.093      41472   37248
2       0       0       0            0.000      9216    0
3       0       0       0            0.000      9216    0
4       0       0       0            0.000      9216    0
5       0       0       0            0.000      9216    0
6       0       0       0            0.000      9216    0
7       0       0       0            0.000      9216    0
```

Il comando show controller to fab QM stat dello slot exec (x)

Utilizzare questo comando per visualizzare il numero di rilasci ToFab per scheda di linea. Verificare inoltre la presenza di un contatore di "nessuna perdita di memoria" incrementabile. Questo contatore aumenta quando CoS non è configurato sul lato ToFab.

```
Router#execute-on slot 13 show controllers tofab QM stat
```

```
========= Line Card (Slot 13) =======
0 no mem drop, 0 soft drop, 0 bump count
0 rawq drops, 1956216536 global red drops, 6804252 global force drops
0 no memory (Ns), 0 no memory hwm (Ns)
no free queue
0       0       0       0
0       0       0       0
0       0       0       0
0       0       0       0
Q status errors
0       0       0       0
0       0       0       0
0       0       0       0
0       0       0       0
```

Case study

In questo caso di studio viene illustrato come configurare un criterio tipico per il nucleo di rete di un ambiente di provider di servizi. Applica i comandi di coda e consente di utilizzare MDRR/WRED per la gestione attiva delle code. Le policy QoS nei router perimetrali in genere utilizzano il traffic marking, il condizionamento, ecc., per consentire ai router nel core di ordinare il traffico in classi in base alla precedenza IP o ai valori DiffServ Code Point (DSCP). Questo caso aziendale utilizza le funzionalità QoS del software Cisco IOS per soddisfare gli SLA (Service Level Agreement) e i diversi livelli di servizio per i servizi voce, video e dati sulla stessa backbone IP.

Nell'approccio, un provider di servizi ha implementato tre classi di traffico. Il più importante è la classe LLQ o Low Latency Queueing. Classe per voce e video. Questa classe deve subire un ritardo minimo e subire jitter, e non deve mai subire perdite di pacchetti o pacchetti riordinati, a condizione che la larghezza di banda di questa classe non superi la larghezza di banda del collegamento. Questa classe è nota come traffico EF PHB (Expedited Forwarding Per-Hop Behavior) nell'architettura DiffServ. Il provider di servizi Internet (ISP) ha progettato la rete in modo che questa classe non superi il 30% del carico medio della larghezza di banda del collegamento. Le altre due classi sono la classe business e la classe di impegno migliore.

In fase di progettazione, i router sono stati configurati in modo che la classe aziendale ottenga sempre il 90% della larghezza di banda rimanente e la classe di impegno migliore il 10%. Queste due classi hanno un traffico meno sensibile al tempo e possono subire perdite di traffico, ritardi più elevati e jitter. Nel progetto, l'attenzione è rivolta alle schede di linea del motore 2: schede di linea 1xOC48 rev B, 4xOC12 rev B e 8xOC3.

Le schede di linea Rev B sono più adatte a trasportare il traffico VoIP grazie a un ASIC rivisto e all'architettura hardware, che introduce una latenza molto ridotta. Con l'ASIC rivisto, la coda FIFO di trasmissione viene ridimensionata dal driver della scheda di linea al doppio dell'MTU più grande presente sulla scheda. Cercare un "-B" aggiunto al numero di parte, ad esempio OC48E/POS-SR-SC-B=.

Nota: non confondere la coda FIFO di trasmissione con le code FrFab che possono essere sintonizzate sulle schede di linea del motore 0 con il comando di interfaccia tx-queue-limit.

La Tabella 9 elenca i criteri di corrispondenza per ciascuna classe.

Tabella 9 - Criteri di corrispondenza per ciascuna classe

Nome classe	Criteri di corrispondenza

Coda prioritaria - Traffico vocale	Precedenza 5
Coda aziendale	Precedenza 4
Coda massimo sforzo	Precedenza 0

Le schede di linea OC48 possono accodare un numero elevato di pacchetti nelle code ToFab. Pertanto, è importante configurare MDRR/WRED sulle code ToFab, in particolare quando l'interfaccia in uscita è un'interfaccia ad alta velocità come OC48. L'infrastruttura può passare il traffico alla scheda di linea ricevente solo a una velocità teorica massima di 3 Gbps (pacchetti da 1500 byte). Se la quantità totale di traffico inviato è superiore a quella che il fabric di switching può trasportare alla scheda di ricezione, molti pacchetti verranno accodati sulle code ToFab.

```
Interface POS3/0
  description OC48 egress interface
 ip address 10.10.105.53 255.255.255.252
 no ip directed-broadcast
 ip router Isis encapsulation ppp
 mpls traffic-eng tunnels
 tag-switching ip
 no peer neighbor-route
 crc 32
 clock source internal
 POS framing sdh
 POS scramble-atm
 POS threshold sf-ber 4
 POS flag s1s0 2
 TX-cos oc48
 Isis metric 2 level-1
 Isis metric 2 level-2
 ip rsvp bandwidth 2400000 2400000
!
interface POS4/1
 description OC12 egress interface
 ip address 10.10.105.121 255.255.255.252
 no ip directed-broadcast
 ip router Isis encapsulation ppp
 mpls traffic-eng tunnels
 no peer neighbor-route
 crc 32
 clock source internal
 POS framing sdh
 POS scramble-ATM POS threshold sf-ber 4
 POS flag s1s0 2
 TX-cos oc12
 Isis metric 2 level-1
 Isis metric 2 level-2
 ip RSVP bandwidth 600000 60000
!
interface POS9/2
 description OC3 egress interface
 ip address 10.10.105.57 255.255.255.252
 no ip directed-broadcast
 ip router Isis crc 16
 POS framing sdh
 POS scramble-ATM POS flag s1s0 2
 TX-cos oc3
 Isis metric 200 level-1
 Isis metric 2 level-2
!
```

```
interface POS13/0
 description agilent 3a for QOS tests - ingress interface.
 ip address 10.10.105.130 255.255.255.252
 no ip directed-broadcast
 no ip route-cache cef
 no ip route-cache
 no ip mroute-cache
 no keepalive
 crc 32
 POS threshold sf-ber 4
 TX-cos oc48
!
interface POS14/0
 description agilent 3b for QOS tests - ingress interface.
 ip address 10.10.105.138 255.255.255.252
 no ip directed-broadcast
 no keepalive
 crc 32
 POS threshold sf-ber 4
 TX-cos oc48
!
interface POS15/0
 description agilent 4A for QOS tests - ingress interface
 ip address 10.10.105.134 255.255.255.252
 no ip directed-broadcast
 no ip mroute-cache
 no keepalive
 crc 32
 POS threshold sf-ber 4
 TX-CoS oc48
!
rx-cos-slot 3 StotTable
rx-cos-slot 4 StotTable
rx-cos-slot 9 StotTable
rx-cos-slot 13 StotTable
rx-cos-slot 14 StotTable
rx-cos-slot 15 StotTable
!
slot-table-cos StotTable
 destination-slot 0 oc48
 destination-slot 1 oc48
 destination-slot 2 oc48
 destination-slot 3 oc48
 destination-slot 4 oc12
 destination-slot 5 oc48
 destination-slot 6 oc48
 destination-slot 9 oc3
 destination-slot 15 oc48
!
cos-queue-groupoc3
 precedence 0 random-detect-label 0
 precedence 4 queue 1
 precedence 4 random-detect-label 1
 precedence 5 queue low-latency
 precedence 6 queue 1
 precedence 6 random-detect-label 1
 random-detect-label 0 94 606 1
 random-detect-label 1 94 606 1
 queue 0 1
 queue 1 73
 queue low-latency strict-priority
!--- Respect the tight SLA requirements. !--- No packets drop/low delay and jitter for the
priority queue. ! CoS-queue-groupoc12
 precedence 0 random-detect-label 0
```

```
 precedence 4 queue 1
 precedence 4 random-detect-label 1
 precedence 5 queue low-latency
 precedence 6 queue 1
 precedence 6 random-detect-label 1
 random-detect-label 0 375 2423 1
 random-detect-label 1 375 2423 1
 queue 0 1
 queue 1 73
 queue low-latency strict-priority
!
CoS-queue-groupoc48
 precedence 0 random-detect-label 0
 precedence 4 queue 1
 precedence 4 random-detect-label 1
 precedence 5 queue low-latency
 precedence 6 queue 1
 precedence 6 random-detect-label 1
 random-detect-label 0 1498 9690 1
 random-detect-label 1 1498 9690 1
 queue 0 1
 queue 1 73
 queue low-latency strict-priority
```

Più traffico VOIP si ha, maggiore sarà il traffico aziendale che dovrà attendere prima di essere servito. Tuttavia, non si tratta di un problema in quanto lo SLA stretto non richiede alcuna perdita di pacchetto e richiede una latenza e un jitter molto bassi per la coda di priorità.

Informazioni correlate

- Come leggere l'output del comando show controller frfab | comandi di coda tofab su un Cisco serie 12000 Internet Router
- Risoluzione dei problemi relativi ai pacchetti ignorati e all'assenza di perdite di memoria sul Cisco serie 12000 Internet Router
- Risoluzione dei problemi di input sul Cisco serie 12000 Internet Router
- Cisco serie 12000 Random Early Detection
- Panoramica dell'interfaccia della riga di comando Modular Quality of Service
- Pagina di supporto per i router Internet serie 12000
- Supporto tecnico – Cisco Systems