Informazioni su Class Based Weighted Fair Queuing su ATM

Sommario

Introduzione

Questo documento offre un'introduzione alle code di traffico che utilizzano la tecnologia CBWFQ (Weighted Fair Queuing) basata su classi.

Il protocollo WFQ (Weighted Fair Queuing) consente ai collegamenti a bassa velocità, ad esempio i collegamenti seriali, di fornire un trattamento equo per tutti i tipi di traffico. Classifica il traffico in diversi flussi (noti anche come conversazioni) in base alle informazioni di livello tre e quattro, come gli indirizzi IP e le porte TCP. Questa operazione viene eseguita senza che sia necessario definire elenchi degli accessi. Ciò significa che il traffico a bassa larghezza di banda ha in effetti la priorità sul traffico a elevata larghezza di banda, perché il traffico a elevata larghezza di banda condivide i supporti di trasmissione in proporzione al loro peso assegnato. Tuttavia, WFQ presenta alcune limitazioni:

- Non è scalabile se il valore del flusso aumenta considerevolmente.
- Il modulo WFQ nativo non è disponibile sulle interfacce ad alta velocità come le interfacce ATM.

CBWFQ fornisce una soluzione per queste limitazioni. A differenza delle WFQ standard, CBWFQ consente di definire le classi di traffico e di applicare a tali classi parametri, quali la larghezza di banda e i limiti di coda. La larghezza di banda assegnata a una classe viene utilizzata per calcolare il "peso" della classe. Anche il peso di ogni pacchetto che soddisfa i criteri della classe viene calcolato da questo. WFQ viene applicato alle classi, che possono includere diversi flussi, anziché ai flussi stessi.

Per ulteriori informazioni sulla configurazione di CBWFQ, fare clic sui seguenti collegamenti:

Weighted Fair Queuing (Per-VC CBWFQ) basato su classi per-VC su router Cisco 7200, 3600 e 2600.

Weighted Fair Queuing basato su classi per-VC su piattaforme basate su RSP.

Operazioni preliminari

Convenzioni

Per ulteriori informazioni sulle convenzioni usate, consultare il documento Cisco sulle convenzioni nei suggerimenti tecnici.

Prerequisiti

Non sono previsti prerequisiti specifici per questo documento.

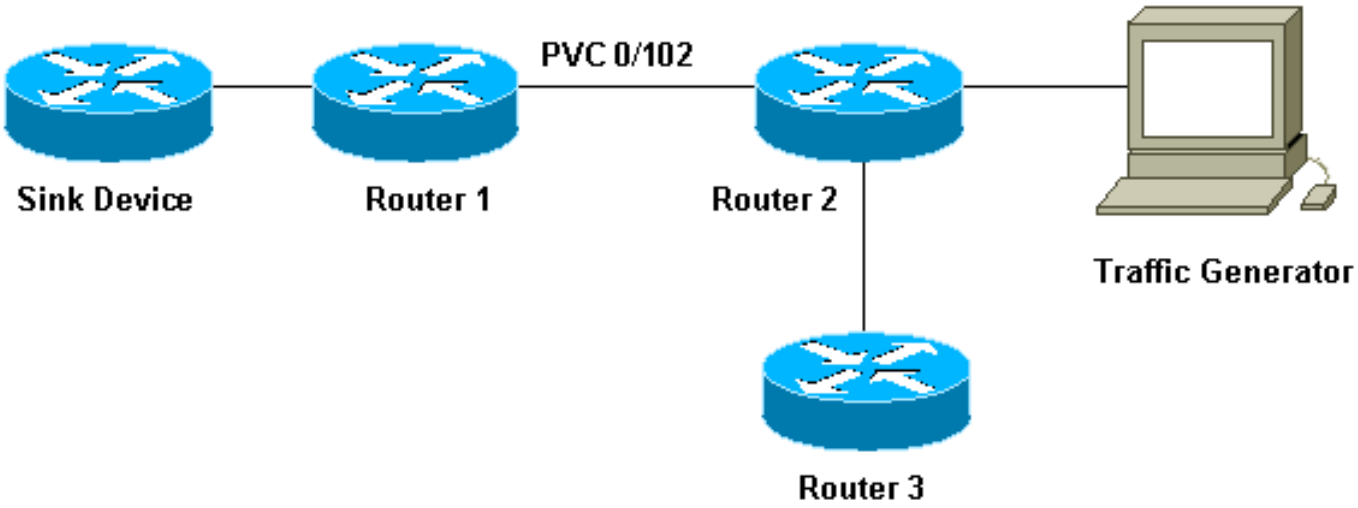
Componenti usati

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

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.

Esempio di rete

Per illustrare il funzionamento di WFQ, utilizzare la configurazione seguente:



Nella configurazione utilizzata, i pacchetti possono essere archiviati in una delle due code seguenti:

- La coda FIFO (First In First Out) dell'hardware sulla scheda di porta e sul modulo di rete.
- La coda nel software Cisco IOS® (nella memoria [I/O] di input/output del router) a cui è possibile applicare le funzionalità Quality of Service (QoS), ad esempio CBWFQ.

La coda FIFO sull'adattatore porta archivia i pacchetti prima che vengano segmentati in celle per la trasmissione. Quando la coda è piena, l'adattatore di porta o il modulo di rete segnala al software IOS che la coda è congestionata. Questo meccanismo è chiamato contropressione. Alla ricezione del segnale, il router interrompe l'invio dei pacchetti alla coda FIFO dell'interfaccia e archivia i pacchetti nel software IOS finché la coda non viene nuovamente congestionata. Quando i pacchetti sono archiviati in IOS, il sistema può applicare le funzionalità QoS, ad esempio CBWFQ.

Impostazione del limite dell'anello di trasmissione

Uno dei problemi è che, maggiore è la dimensione della coda FIFO sull'interfaccia, maggiore è il ritardo prima della trasmissione dei pacchetti alla fine della coda. Ciò può causare gravi problemi di prestazioni per il traffico sensibile al ritardo, ad esempio il traffico vocale.

Il comando **tx-ring-limit** del circuito virtuale permanente (PVC) consente di ridurre le dimensioni della coda FIFO.

```
interface ATMx/y.z point-to-point
      ip address a.b.c.d M.M.M.M
      PVC A/B
       TX-ring-limit
       service-policy output test
```

Il limite (x) che è possibile specificare qui è il numero di pacchetti (per router Cisco 2600 e 3600) o la quantità di particelle (per router Cisco 7200 e 7500).

La riduzione della dimensione della ghiera di trasmissione presenta due vantaggi:

- Riduce il tempo di attesa dei pacchetti nella coda FIFO prima di essere segmentati.
- Accelera l'uso di QoS nel software IOS.

Impatto del limite dell'anello di trasmissione

Esaminiamo l'impatto del limite dell'anello di trasmissione utilizzando la configurazione mostrata nel diagramma di rete sopra. Si può supporre quanto segue:

- Il generatore di traffico sta inviando del traffico (pacchetti da 1500 byte) al dispositivo sink e questo traffico sta sovraccaricando il PVC 0/102 tra il router1 e il router2.
- Il router3 tenta di eseguire il ping tra il router1.
- CBWFQ è abilitato sul router 2.

Esaminiamo ora due configurazioni che utilizzano limiti diversi degli anelli di trasmissione per verificare l'impatto.

Esempio A

Nell'esempio, l'anello di trasmissione è impostato su tre (**TX-ring-limit=3**). Ecco cosa succede quando si esegue il ping tra router1 e router3.

```
POUND#ping ip
     Target IP address: 6.6.6.6
     Repeat count [5]: 100000
     Datagram size [100]:
     Timeout in seconds [2]:
     Extended commands [n]:
     Sweep range of sizes [n]:
     Type escape sequence to abort.
     Sending 100000, 100-byte ICMP Echos to 6.6.6.6, timeout is 2 seconds:
     !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
     !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
     !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
     !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
     !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
     !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
     !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
     !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!![snip]
     Success rate is 98 percent (604/613), round-trip min/avg/max = 164/190/232 ms
```

Esempio B

Nell'esempio, l'anello di trasmissione è impostato su 40 (**TX-ring-limit=40**). Di seguito è riportato ciò che si verifica quando si utilizza lo stesso ping dell'esempio A:

```
POUND#ping ip
     Target IP address: 6.6.6.6
     Repeat count [5]: 10000
     Datagram size [100]: 36
     Timeout in seconds [2]: 10
     Extended commands [n]:
     Sweep range of sizes [n]:
     Type escape sequence to abort.
     Sending 10000, 36-byte ICMP Echos to 6.6.6.6, timeout is 10 seconds:
     !!!!!!!!!!!!.
     Success rate is 92 percent (12/13), round-trip min/avg/max = 6028/6350/6488
```

Come si può vedere, maggiore è il limite degli anelli di trasmissione, maggiore è il tempo di andata e ritorno del ping (RTT). Da ciò si può dedurre che un limite elevato degli anelli di trasmissione può causare ritardi significativi nella trasmissione.

Funzionamento di CBWFQ

Ora che abbiamo visto l'impatto delle dimensioni della coda FIFO hardware, vediamo esattamente come funziona CBWFQ.

La WFQ nativa assegna un peso a ogni conversazione e quindi pianifica il tempo di trasmissione per ogni pacchetto dei diversi flussi. Il peso è una funzione della precedenza IP di ciascun flusso e la durata della programmazione dipende dalle dimensioni del pacchetto. Fare clic qui per ulteriori dettagli su WFQ.

CBWFQ assegna un peso a ciascuna classe configurata anziché a ciascun flusso. Questo peso è proporzionale alla larghezza di banda configurata per ogni classe. Più precisamente, il peso è una funzione della larghezza di banda dell'interfaccia divisa per la classe larghezza di banda. Pertanto, maggiore è il parametro della larghezza di banda, minore sarà il peso.

Èpossibile calcolare l'ora di programmazione dei pacchetti utilizzando la seguente formula:

```
scheduling tail_time= queue_tail_time + pktsize * weight
```

Divisione larghezza di banda interfaccia totale

Esaminiamo come il router divide la larghezza di banda totale dell'interfaccia tra le diverse classi. Per servire le classi, il router usa le code di calendario. Ognuna di queste code di calendario memorizza i pacchetti che devono essere trasmessi nello stesso orario di pianificazione. Il router quindi gestisce queste code di calendario una alla volta. Esaminiamo questo processo:

1. Se l'adattatore della porta riceve un pacchetto sull'interfaccia di output, si verifica una congestione, il pacchetto viene inserito in coda nel sistema operativo IOS (in questo caso, CBWFQ).
2. Il router calcola l'ora di pianificazione per il pacchetto in arrivo e lo memorizza nella coda del calendario corrispondente a questa ora di pianificazione. In una coda di calendario specifica è possibile archiviare un solo pacchetto per classe.
3. Quando è il momento di servire la coda del calendario in cui il pacchetto è stato archiviato, IOS svuota questa coda e invia i pacchetti alla coda FIFO sull'adattatore della porta stesso. La dimensione della coda FIFO è determinata dal limite degli anelli di trasmissione descritto sopra.
4. Se la coda FIFO è troppo piccola per contenere tutti i pacchetti contenuti nella coda del calendario servita, il router riprogramma i pacchetti che non possono essere memorizzati per l'ora di programmazione successiva (corrispondente al loro peso) e li inserisce nella coda del calendario corrispondente.
5. Al termine, l'adattatore della porta tratta i pacchetti nella coda FIFO e invia le celle in transito e il sistema operativo IOS passa alla coda di calendario successiva.Grazie a questo meccanismo, ogni classe riceve statisticamente una parte della larghezza di banda dell'interfaccia corrispondente ai parametri configurati per essa.

Meccanismo della coda del calendario e dimensione dell'anello di trasmissione

Esaminiamo la relazione tra il meccanismo della coda del calendario e le dimensioni dell'anello di trasmissione. Un piccolo anello di trasmissione consente al QoS di avviarsi più rapidamente e riduce la latenza per i pacchetti in attesa di essere trasmessi (importante per il traffico sensibile al ritardo, come la voce). Tuttavia, se è troppo piccolo, può causare un throughput inferiore per determinate classi. Infatti, se la ghiera di trasmissione non è in grado di contenere molti pacchetti, potrebbe essere necessario riprogrammarli.

Sfortunatamente, non esiste un valore ideale per le dimensioni dell'anello di trasmissione e l'unico modo per trovare il valore migliore è sperimentare.

Condivisione della larghezza di banda

Possiamo esaminare il concetto di condivisione della larghezza di banda usando l'impostazione mostrata nel nostro diagramma di rete, sopra. Il generatore di pacchetti genera flussi diversi e li invia al dispositivo di sink. La quantità totale di traffico rappresentata da questi flussi è sufficiente per sovraccaricare il PVC. CBWFQ è stato implementato sul router2. Di seguito è riportata la configurazione:

```
access-list 101 permit ip host 7.0.0.200 any
   access-list 101 permit ip host 7.0.0.201 any
   access-list 102 permit ip host 7.0.0.1 any
   !
   class-map small
     match access-group 101
   class-map big
     match access-group 102
   !
   policy-map test
   policy-map test
     small class
      bandwidth <x>
     big class
      bandwidth <y>
    interface atm 4/0.102
     pvc 0/102
       TX-ring-limit 3
       service-policy output test
       vbr-nrt 64000 64000
```

Nell'esempio, Router2 è un router Cisco 7200. Questo è importante perché il limite dell'anello di trasmissione è espresso in particelle, non in pacchetti. I pacchetti vengono accodati nella coda FIFO dell'adattatore di porta non appena è disponibile una particella libera, anche se è necessaria più di una particella per memorizzare il pacchetto.

Cos'è una particella?

Anziché allocare un pezzo di memoria contigua per un buffer, il buffering delle particelle alloca pezzi di memoria discontigui (sparsi), chiamati particelle, e poi li collega per formare un buffer di pacchetto logico. Questo si chiama buffer di particelle. In uno schema di questo tipo, un pacchetto può quindi essere distribuito su più particelle.

Nel router 7200 che stiamo usando qui, la dimensione delle particelle è 512 byte.

Èpossibile verificare se i router Cisco 7200 utilizzano particelle usando il comando **show buffers**:

```
router2#show buffers
     [snip]
     Private particle pools:
     FastEthernet0/0 buffers, 512 bytes (total 400, permanent 400):
          0 in free list (0 min, 400 max allowed)
          400 hits, 0 fallbacks
          400 max cache size, 271 in cache
     ATM4/0 buffers, 512 bytes (total 400, permanent 400):
          0 in free list (0 min, 400 max allowed)
          400 hits, 0 fallbacks
          400 max cache size, 0 in cache
```

Test A

Le classi "Small" e "Big" utilizzate per questo test sono popolate nel modo seguente:

- Classe di piccole dimensioni: i parametri della larghezza di banda sono stati configurati a 32 kbps. Questa classe memorizza dieci pacchetti di 1500 byte da 7.0.0.200, seguiti da dieci pacchetti di 1500 byte da 7.0.0.201
- Grande classe: il parametro della larghezza di banda è stato configurato a 16 kbps. Questa classe archivia un flusso di dieci pacchetti da 1500 byte da 7.0.0.1.

Il generatore di traffico invia un'esplosione di traffico destinata al dispositivo sink a 100 Mbps al router2 nell'ordine seguente:

1. Dieci pacchetti dalla versione 7.0.0.1.
2. Dieci pacchetti dal 7.0.0.200.
3. Dieci pacchetti dal 7.0.0.2010.

Verifica del peso del flusso

Guardiamo il peso applicato ai diversi flussi. A tale scopo, è possibile utilizzare il comando **show queue ATM x/y.z**.

```
alcazaba#show queue ATM 4/0.102
     Interface ATM4/0.102 VC 0/102
     Queueing strategy: weighted fair
     Total output drops per VC: 0
     Output queue: 9/512/64/0 (size/max total/threshold/drops)
        Conversations  2/3/16 (active/max active/max total)
        Reserved Conversations 2/2 (allocated/max allocated)

  (depth/weight/total drops/no-buffer drops/interleaves) 7/128/0/0/0
     Conversation 25, linktype: ip, length: 1494
     source: 7.0.0.201, destination: 6.6.6.6, id: 0x0000, ttl: 63, prot: 255

  (depth/weight/total drops/no-buffer drops/interleaves) 2/256/0/0/0
     Conversation 26, linktype: ip, length: 1494
     source: 7.0.0.1, destination: 6.6.6.6, id: 0x0000, ttl: 63, prot: 255
```

Quando tutti i pacchetti della versione 7.0.0.200 sono stati inseriti nella coda all'esterno del router, è possibile verificare quanto segue:

```
alcazaba#show queue ATM 4/0.102
     Interface ATM4/0.102 VC 0/102
```

```
     Queueing strategy: weighted fair
     Total output drops per VC: 0
     Output queue: 9/512/64/0 (size/max total/threshold/drops)
        Conversations  2/3/16 (active/max active/max total)
        Reserved Conversations 2/2 (allocated/max allocated)

  (depth/weight/total drops/no-buffer drops/interleaves) 7/128/0/0/0
     Conversation 25, linktype: ip, length: 1494
     source: 7.0.0.201, destination: 6.6.6.6, id: 0x0000, ttl: 63, prot: 255

  (depth/weight/total drops/no-buffer drops/interleaves) 2/256/0/0/0
     Conversation 26, linktype: ip, length: 1494
     source: 7.0.0.1, destination: 6.6.6.6, id: 0x0000, ttl: 63, prot: 255
```

Come potete vedere qui, i flussi da 7.0.0.200 e 7.0.0.201 hanno lo stesso peso (128). Questo peso è la metà del peso assegnato al flusso da 7.0.0.1 (256). Ciò corrisponde al fatto che la nostra larghezza di banda ridotta è il doppio della dimensione della nostra grande classe.

Verifica della distribuzione della larghezza di banda

Come possiamo verificare la distribuzione della larghezza di banda tra i diversi flussi? In ogni classe viene utilizzato il metodo di coda FIFO. La nostra piccola classe è piena di dieci pacchetti dal primo flusso e dieci pacchetti dal secondo flusso. Il primo flusso viene rimosso dalla classe piccola a 32 kbps. Non appena sono stati inviati, vengono inviati anche i dieci pacchetti provenienti dall'altro flusso. Nel frattempo, i pacchetti della nostra classe vengono rimossi a 16 kbps.

Possiamo notare che, dal momento che il generatore di traffico sta inviando un'esplosione a 100 Mbps, il PVC sarà sovraccarico. Tuttavia, poiché all'avvio del test non è presente traffico sul PVC e, poiché i pacchetti della versione 7.0.0.1 sono i primi a raggiungere il router, alcuni pacchetti della versione 7.0.0.1 verranno inviati prima dell'avvio del CBWFQ a causa della congestione (in altre parole, prima che il ring di trasmissione sia pieno).

Poiché la dimensione delle particelle è di 512 byte e la dimensione dell'anello di trasmissione è di tre particelle, possiamo notare che due pacchetti da 7.0.0.1 vengono inviati prima che si verifichi una congestione. Una viene immediatamente inviata sul filo e la seconda viene memorizzata nelle tre particelle che formano la coda FIFO dell'adattatore di porta.

Di seguito sono riportati i debug sul dispositivo sink (che è semplicemente un router):

```
Nov 13 12:19:34.216: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, len 1482, rcvd 4
   Nov 13 12:19:34.428: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4

   !--- congestion occurs here. Nov 13 12:19:34.640: IP: s=7.0.0.200 (FastEthernet0/1),
d=6.6.6.6, Len 1482, rcvd 4 Nov 13 12:19:34.856: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6,
Len 1482, rcvd 4 Nov 13 12:19:35.068: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd
4 Nov 13 12:19:35.280: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 13
12:19:35.496: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 13
12:19:35.708: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 13 12:19:35.920:
IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 13 12:19:36.136: IP:
s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 13 12:19:36.348: IP: s=7.0.0.1
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 13 12:19:36.560: IP: s=7.0.0.200
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 13 12:19:36.776: IP: s=7.0.0.200
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 13 12:19:36.988: IP: s=7.0.0.1
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 13 12:19:37.200: IP: s=7.0.0.200
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 13 12:19:37.416: IP: s=7.0.0.200
```

```
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 13 12:19:37.628: IP: s=7.0.0.1
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 13 12:19:37.840: IP: s=7.0.0.201
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 13 12:19:38.056: IP: s=7.0.0.201
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 13 12:19:38.268: IP: s=7.0.0.1
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 13 12:19:38.480: IP: s=7.0.0.201
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 13 12:19:38.696: IP: s=7.0.0.201
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 13 12:19:38.908: IP: s=7.0.0.1
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 13 12:19:39.136: IP: s=7.0.0.201
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 13 12:19:39.348: IP: s=7.0.0.201
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 13 12:19:39.560: IP: s=7.0.0.1
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 13 12:19:39.776: IP: s=7.0.0.201
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 13 12:19:39.988: IP: s=7.0.0.201
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 13 12:19:40.200: IP: s=7.0.0.201
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 13 12:19:40.416: IP: s=7.0.0.201
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4
```

Poiché le dimensioni dei pacchetti per entrambi i flussi sono uguali, in base alla formula del tempo di programmazione, dovremmo vedere due pacchetti della nostra piccola classe essere inviati per ogni pacchetto della nostra grande classe. Questo è esattamente quello che vediamo nei debug qui sopra.

Test B

Per il secondo test, popolare le classi nel modo seguente:

- Classe piccola: il parametro della larghezza di banda è stato configurato su 32 kbps. Vengono generati dieci pacchetti di 500 byte da 7.0.0.200, seguiti da dieci pacchetti di 1500 byte da 7.0.0.201.
- Grande classe: il parametro della larghezza di banda è stato configurato a 16 kbps. La classe archivia un flusso di pacchetti da 1500 byte provenienti da 7.0.0.1.

Il generatore di traffico invia una frammentazione del traffico a 100 Mbps al router2 nel seguente ordine:

1. Dieci pacchetti da 1500 byte da 7.0.0.1.
2. Dieci pacchetti da 500 byte da 7.0.0.200.
3. Dieci pacchetti da 1500 byte da 7.0.0.201.

FIFO è configurato in ciascuna classe.

Verifica del peso del flusso

La fase successiva consiste nel verificare il peso applicato ai flussi classificati:

```
alcazaba#show queue ATM 4/0.102
   Interface ATM4/0.102 VC 0/102
   Queueing strategy: weighted fair
   Total output drops per VC: 0
   Output queue: 23/512/64/0 (size/max total/threshold/drops)
      Conversations  2/3/16 (active/max active/max total)
      Reserved Conversations 2/2 (allocated/max allocated)

  (depth/weight/total drops/no-buffer drops/interleaves) 15/128/0/0/0
     Conversation 25, linktype: ip, length: 494
     source: 7.0.0.200, destination: 6.6.6.6, id: 0x0000, ttl: 63, prot: 255

  (depth/weight/total drops/no-buffer drops/interleaves) 8/256/0/0/0
```

```
     Conversation 26, linktype: ip, length: 1494
     source: 7.0.0.1, destination: 6.6.6.6, id: 0x0000, ttl: 63, prot: 255

   alcazaba#show queue ATM 4/0.102
   Interface ATM4/0.102 VC 0/102
   Queueing strategy: weighted fair
   Total output drops per VC: 0
   Output queue: 13/512/64/0 (size/max total/threshold/drops)
        Conversations  2/3/16 (active/max active/max total)
        Reserved Conversations 2/2 (allocated/max allocated)

  (depth/weight/total drops/no-buffer drops/interleaves) 8/128/0/0/0
     Conversation 25, linktype: ip, length: 1494
     source: 7.0.0.201, destination: 6.6.6.6, id: 0x0000, ttl: 63, prot: 255

  (depth/weight/total drops/no-buffer drops/interleaves) 5/256/0/0/0
     Conversation 26, linktype: ip, length: 1494
     source: 7.0.0.1, destination: 6.6.6.6, id: 0x0000, ttl: 63,
```

Come si può vedere nell'output precedente, i flussi da 7.0.0.200 e 7.0.0.201 hanno ricevuto lo stesso peso (128). Questo peso è la metà del peso assegnato al flusso da 7.0.0.1. Ciò corrisponde al fatto che la classe piccola ha una larghezza di banda doppia rispetto alla classe grande.

Verifica della distribuzione della larghezza di banda

Possiamo produrre i seguenti debug dal dispositivo sink:

```
Nov 14 06:52:01.761: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4
   Nov 14 06:52:01.973: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4

   !--- Congestion occurs here. Nov 14 06:52:02.049: IP: s=7.0.0.200 (FastEthernet0/1),
d=6.6.6.6, Len 482, rcvd 4 Nov 14 06:52:02.121: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6,
Len 482, rcvd 4 Nov 14 06:52:02.193: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 482, rcvd
4 Nov 14 06:52:02.269: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4 Nov 14
06:52:02.341: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4 Nov 14 06:52:02.413:
IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4 Nov 14 06:52:02.629: IP: s=7.0.0.1
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 14 06:52:02.701: IP: s=7.0.0.200
(FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4 Nov 14 06:52:02.773: IP: s=7.0.0.200
(FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4 Nov 14 06:52:02.849: IP: s=7.0.0.200
(FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4 Nov 14 06:52:02.921: IP: s=7.0.0.200
(FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4 Nov 14 06:52:03.149: IP: s=7.0.0.1
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 14 06:52:03.361: IP: s=7.0.0.201
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 14 06:52:03.572: IP: s=7.0.0.201
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 14 06:52:03.788: IP: s=7.0.0.1
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 14 06:52:04.000: IP: s=7.0.0.201
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 14 06:52:04.212: IP: s=7.0.0.201
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 14 06:52:04.428: IP: s=7.0.0.1
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 14 06:52:04.640: IP: s=7.0.0.201
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 14 06:52:04.852: IP: s=7.0.0.201
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 14 06:52:05.068: IP: s=7.0.0.1
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 14 06:52:05.280: IP: s=7.0.0.201
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 14 06:52:05.492: IP: s=7.0.0.201
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 14 06:52:05.708: IP: s=7.0.0.1
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 14 06:52:05.920: IP: s=7.0.0.201
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 14 06:52:06.132: IP: s=7.0.0.201
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 14 06:52:06.348: IP: s=7.0.0.1
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4 Nov 14 06:52:06.560: IP: s=7.0.0.1
(FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4
```

In questo scenario, i flussi della classe piccola non hanno le stesse dimensioni del pacchetto.

Quindi, la distribuzione dei pacchetti non è così banale come per il Test A di cui sopra.

Orari di programmazione

Esaminiamo più da vicino gli orari di pianificazione di ciascun pacchetto. Il tempo di programmazione dei pacchetti viene calcolato con la formula seguente:

```
scheduling tail_time= sub_queue_tail_time + pktsize *
     weight
```

Per le diverse dimensioni del pacchetto, l'ora di programmazione utilizza la seguente formula:

```
500 bytes (small class): scheduling tail_time = x + 494 * 128
     = x + 63232
     1500 bytes (small class): scheduling tail_time = x + 1494 *
     128 = x + 191232
     1500 bytes (big class): scheduling tail_time = x + 1494 *
     256 = x + 382464
```

Da queste formule, è possibile vedere che sei pacchetti di 500 byte della nostra piccola classe vengono trasmessi per ciascun pacchetto di 1500 byte della nostra grande classe (mostrato nell'output del debug di cui sopra).

Possiamo anche notare che due pacchetti di 1500 byte della nostra piccola classe vengono inviati per un pacchetto di 1500 byte della nostra grande classe (mostrato nell'output del debug di cui sopra).

Dai test sopra riportati, è possibile concludere quanto segue:

- La dimensione dell'anello di trasmissione (limite dell'anello TX) determina la velocità con cui il meccanismo di coda inizia a funzionare. Possiamo notare l'impatto con l'aumento del ping RTT quando il limite dell'anello di trasmissione aumenta. Pertanto, se si implementa CBWFQ o Low Latency Queueing [LLQ], è consigliabile ridurre il limite dell'anello di trasmissione.
- CBWFQ consente una condivisione equa della larghezza di banda dell'interfaccia tra diverse classi.

Informazioni correlate

- Weighted Fair Queuing (CBWFQ per VC) basato su classi per-VC su router Cisco 7200, 3600 e 2600
- Weighted Fair Queuing basato su classi per-VC su piattaforme basate su RSP
- Informazioni su Weighted Fair Queuing su ATM
- Supporto tecnologia IP to ATM Class of Service
- Supporto della tecnologia ATM
- Documentazione e supporto tecnico – Cisco Systems