Comprensión de Class Based Weighted Fair Queuing en ATM

Contenido

Introducción

Este documento provee una introducción para la cola de tráfico usando tecnología de class-based weighted fair queuing (CBWFQ)

El almacenamiento justo y ponderado en las colas (WFQ) posibilita los links de baja velocidad, tales como los links seriales, para proporcionar un tratamiento justo para todos los tipos de tráfico. Clasifica el tráfico en flujos diferentes (conocidos también como conversaciones) basadas en información de capa tres y capa cuatro, tales como direcciones IP y puertos TCP. Hace esto sin solicitarle que defina listas de acceso. Esto significa que el tráfico de ancho de banda bajo efectivamente tiene prioridad sobre el tráfico de ancho de banda alto porque el tráfico de ancho de banda alto comparte los medios de transmisión en proporción a su peso asignado. No obstante, WFQ tiene ciertas limitaciones:

- No hay posibilidad de ampliación si la cantidad de flujo aumenta considerablemente.
- WFQ nativo no está disponible en interfaces de alta velocidad como las interfaces ATM.

CBWFQ proporciona una solución para estas limitaciones. A diferencia de la WFQ estándar, CBWFQ le permite definir clases de tráfico y aplicar parámetros, tales como límites de cola y ancho de banda, a estas clases. El ancho de banda que se le asigna a una clase se utiliza para calcular el “peso” de esa clase. A partir de esto, también se calcula el peso de cada paquete que coincide con los criterios de clase. WFQ se aplica a las clases (que pueden incluir varios flujos) y no a los flujos mismos.

Para obtener más información sobre la configuración de CBWFQ, haga clic en los siguientes links:

Colocación en cola equilibrada ponderada por VC en función de la clase (CBWFQ por VC) en los routers Cisco 7200, 3600 y 2600.

Colocación en cola equilibrada ponderada por VC en función de la clase en plataformas con base RSP

Antes de comenzar

Convenciones

Para obtener más información sobre las convenciones del documento, consulte Convenciones de Consejos Técnicos de Cisco.

Prerequisites

No hay requisitos previos específicos para este documento.

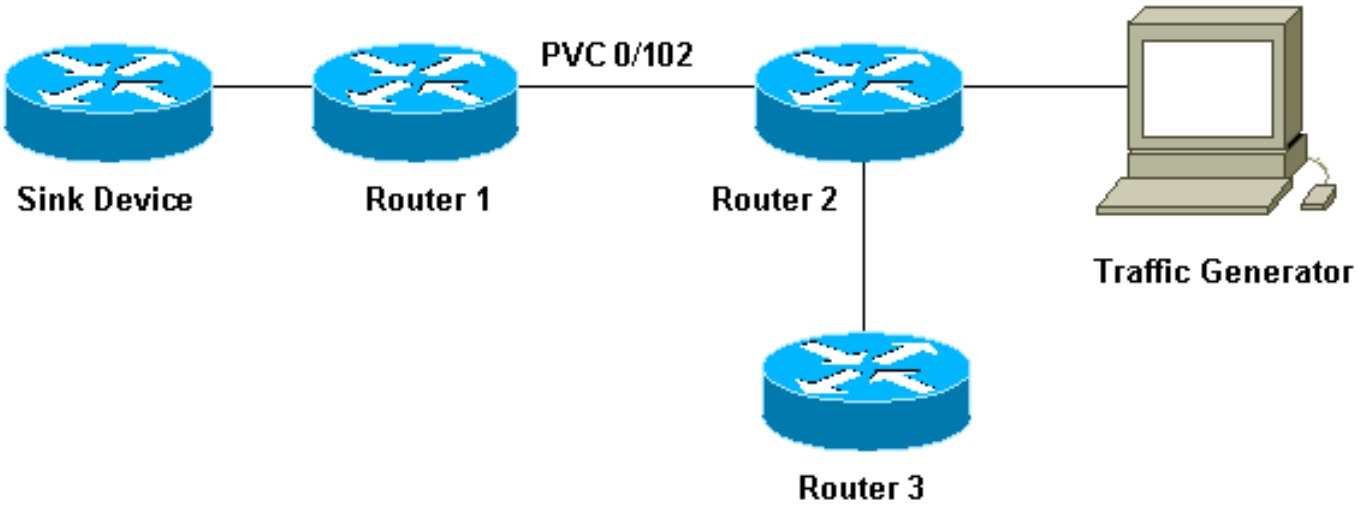
Componentes Utilizados

Este documento no tiene restricciones específicas en cuanto a versiones de software y de hardware.

La información que se presenta en este documento se originó a partir de dispositivos dentro de un ambiente de laboratorio específico. All of the devices used in this document started with a cleared (default) configuration. Si la red está funcionando, asegúrese de haber comprendido el impacto que puede tener un comando antes de ejecutarlo.

Diagrama de la red

Para ilustrar el funcionamiento de WFQ, utilizaremos la siguiente configuración:



En la configuración que utilizamos aquí, los paquetes pueden almacenarse en una de las siguientes dos colas:

- La cola Primero en entrar, primero en salir (FIFO) de hardware en el módulo de red y adaptador de puerto.
- La cola en el software Cisco IOS® (en la memoria de entrada/salida [E/S] del router) donde se pueden aplicar funciones de calidad de servicio (QoS) como CBWFQ.

La cola FIFO en el adaptador de puerto almacena los paquetes antes de que se segmenten en celdas para su transmisión. Cuando esta cola se llena, el adaptador del puerto o el módulo de la red señala al software IOS que la cola se ha congestionado. El mecanismo se llama contrapresión. Al recibir esta señal, el router deja de enviar paquetes a la cola FIFO de la interfaz y almacena los paquetes en el software IOS hasta que la cola se descongestione nuevamente. Cuando se almacenan los paquetes en el IOS, el sistema puede aplicar las funciones de calidad de servicio (QoS) como, por ejemplo, CBWFQ.

Configuración del límite del anillo de transmisión

Un problema de este mecanismo de almacenamiento en cola es que cuanto más grande es la cola FIFO en la interfaz, más prolongada puede ser la demora antes de transmitir los paquetes al final de esta cola. Esto puede producir problemas graves de rendimiento en el tráfico sensible al retardo como el tráfico de voz.

El comando tx-ring-limit del circuito virtual permanente (PVC) le permite reducir el tamaño de la cola 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
```

El límite (x) que puede especificar aquí es una cantidad de paquetes (para routers Cisco 2600 y 3600) o una cantidad de partículas (para routers Cisco 7200 y 7500).

La reducción del tamaño del anillo de transmisión brinda dos beneficios:

- Reduce la cantidad de tiempo que aguardan los paquetes en la cola FIFO antes de ser segmentados.
- Acelera el uso de QoS en el software IOS.

Impacto del límite del anillo de transmisión

Observemos el efecto del límite del anillo de transmisión mediante la configuración mostrada en el diagrama de red anterior. Podemos suponer lo siguiente:

- El generador de tráfico envía tráfico (paquetes de 1500 byte) al dispositivo receptor y este tráfico sobrecarga el PVC 0/102 entre el router1 y el router2.
- El router 3 intenta hacer un ping al router 1.
- CBWFQ está habilitado en el router 2.

Para ver el impacto que esto tiene, veamos dos configuraciones que utilizan diferentes límites de anillo de transmisión.

Ejemplo A:

En este ejemplo, configuramos el anillo de transmisión en tres (tx-ring-limit=3). Esto es lo que vemos cuando hacemos ping al router1 desde el 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
```

Ejemplo B

En este ejemplo, configuramos el anillo de transmisión en 40 (TX-ring-limit=40). Esto es lo que vemos cuando utilizamos el mismo ping que en el ejemplo 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
```

Como se puede observar, si el límite de tono de transmisión es mayor, mayor será el tiempo que insumirá el trayecto de ida y vuelta del ping (RTT). De esto podemos deducir que un límite del anillo de transmisión amplio puede resultar en importantes demoras en la transmisión.

Cómo funciona CBWFQ

Ahora que hemos visto el impacto que tiene el tamaño de la cola FIFO de hardware, veamos cómo funciona exactamente CBWFQ.

WFQ nativo asigna un peso a cada conversación y luego programa el tiempo de transmisión para cada paquete de los diferentes flujos. El peso es una función de la precedencia IP de cada flujo y el tiempo de programación depende del tamaño del paquete. Haga clic aquí para obtener más detalles sobre WFQ.

CBWFQ asigna un peso a cada clase configurada, en vez de a cada flujo. Este peso es proporcional al ancho de banda configurado para cada clase. Más precisamente, el peso es una función del ancho de banda de la interfaz dividido por el ancho de banda de la clase. Por lo tanto, cuanto mayor sea el parámetro de ancho de banda, menor será el peso.

Podemos calcular el tiempo de programación del paquete mediante la siguiente fórmula:

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

División de ancho de banda de interfaz total

Veamos cómo el router divide el ancho de banda total de la interfaz entre las diferentes clases. Para dar servicio a las clases, el router utiliza colas de calendario. Cada una de estas colas de calendario almacena paquetes que se deben transmitir en el mismo tiempo de cola planificado. El router luego se ocupa de estas colas del calendario de a una a la vez. Observemos este proceso:

1. Si se produce una congestión en el adaptador de puerto cuando llega un paquete en la interfaz de salida, esto provoca la colocación en cola en IOS (CBWFQ en este caso).
2. El router calcula un horario de programación para este paquete que llega y lo guarda en la cola del calendario correspondiente a este horario de programación. Sólo se puede almacenar un paquete por clase en una cola de calendario particular.
3. Cuando llega el momento de prestar servicio a la cola del calendario en la que ha sido almacenado el paquete, el IOS vacía esta cola y envía los paquetes a la cola primero en entrar, primero en salir en el adaptador de puerto. El tamaño de esta cola FIFO viene determinado por el límite del anillo de transmisión descrito arriba.
4. Si la cola FIFO es muy pequeña para acomodar a todos los paquetes que se mantienen en la cola del calendario de servicios, el router vuelve a programar los paquetes que no pueden almacenarse para la próxima programación (que corresponde a su peso) y los coloca en la cola del calendario correspondiente.
5. Cuando se hace todo esto, el adaptador de puerto trata los paquetes en su cola FIFO y envía las celdas en el cable y el IOS se mueve a la cola del calendario siguiente.Gracias a este mecanismo, cada clase recibe estadísticamente una parte del ancho de banda de

interfaz correspondiente a los parámetros configurados para ello.

Mecanismo de cola de calendario contra tamaño del anillo de transmisión

Observemos la relación entre el mecanismo de cola del calendario y el tamaño del anillo de transmisión. Un anillo de transmisión pequeño permite que QoS se inicie más rápido y reduce la latencia para los paquetes en espera de ser transmitidos (que es importante para el tráfico sensible al retardo como la voz). Sin embargo, si es muy pequeño, puede causar menor rendimiento en ciertas clases. Esto se debe a que es posible que haya que reprogramar muchos paquetes si el anillo de transmisión no puede darles cabida.

Desafortunadamente, no hay un valor ideal para el tamaño del anillo de transmisión y la única manera de encontrar el mejor valor es experimentando.

Uso compartido de ancho de banda

Podemos observar el concepto de ancho de banda compartido mediante la configuración presentada en el diagrama de red anterior. El generador de paquetes produce diferentes flujos y los envía al dispositivo receptor. La cantidad total de tráfico representado por estos flujos es suficiente para sobrecargar el PVC. Hemos implementado CBWFQ en el Router 2. Así es como se ve nuestra configuración:

```
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
```

En nuestro ejemplo, el Router2 es un Cisco 7200 Router. Esto es importante dado que el límite del anillo de transmisión se expresa en partículas, no en paquetes. Los paquetes se colocan en la cola primero en entrar, primero en salir del adaptador de puerto apenas haya una partícula disponible, incluso si se necesita más de una partícula para almacenar el paquete.

¿Qué es una partícula?

En vez de destinar una pieza de memoria contigua para un búfer, el almacenamiento de partículas destina piezas discontinuas de memoria, llamadas partículas, y luego las une para formar un búfer lógico de paquetes. A esto se lo llama memoria intermedia de partículas. En tal esquema, un paquete puede esparcirse en partículas múltiples.

En el router 7200 que usamos aquí, el tamaño de la partícula es de 512 bytes.

Podemos verificar si los routers Cisco 7200 utilizan partículas mediante el 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

Las clases “Pequeña” y “Grande” que usamos para esta prueba son alimentados de la siguiente manera.

- Clase chica - hemos configurado los parámetros del ancho de banda en 32 kbps. Esta clase almacena diez paquetes de 1500 bytes desde 7.0.0.200 seguidos de diez paquetes de 1500 bytes desde 7.0.0.201
- Clase grande – hemos configurado el parámetro del ancho de banda en 16 kbps. Esta clase almacena un flujo de diez paquetes de 1500 bytes de 7.0.0.1.

El generador de tráfico envía una ráfaga de tráfico destinada al dispositivo receptor a 100 Mbps al Router2 en el siguiente orden:

1. Diez paquetes desde 7.0.0.1.
2. Diez paquetes desde 7.0.0.200.
3. Diez paquetes desde 7.0.0.201.

Verificación del volumen de flujo

Debe mirarse el peso aplicado a los distintos flujos. Para hacer esto, podemos usar el 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
```

Cuando todos los paquetes de 7.0.0.200 se han puesto en cola fuera del router, podemos ver lo

siguiente:

```
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
```

Como puede observar aquí, los flujos desde 7.0.0.200 y 7.0.0.201 tienen el mismo peso (128). Este peso es la mitad del tamaño del peso asignado para el flujo desde 7.0.0.1 (256). Esto corresponde al hecho de que nuestro ancho de banda de clase pequeña duplica el tamaño de nuestra clase grande.

Verificación de la distribución de la banda ancha

Entonces, ¿cómo podemos comprobar la distribución del ancho de banda entre los diferentes flujos? El método de colocación en cola FIFO se usa en cada clase. Nuestra clase pequeña está llena con diez paquetes del primer flujo y diez paquetes del segundo flujo. El primer flujo se elimina de la clase pequeña a 32 kbps. En cuanto se hayan enviado, también se envían los diez paquetes desde el otro flujo. Mientras tanto, los paquetes de nuestra clase mayor se eliminan a 16 kbps.

Podemos ver que, dado que el generador de tráfico está enviando una ráfaga a 100 Mbps, el PVC se sobrecargará. Sin embargo, como no hay tráfico en el PVC cuando comienza la prueba y, como los paquetes de 7.0.0.1 son los primeros que alcanzan el router, algunos paquetes de 7.0.0.1 serán enviados antes de que el CBWFQ comience debido a la congestión (en otras palabras, antes de que el anillo de transmisión esté lleno).

Dado que el tamaño de la partícula es 512 bytes y que el tamaño del anillo de transmisión es tres partículas, se puede observar que antes que se produzca la congestión, se envían dos paquetes desde 7.0.0.1. El primero se envía de inmediato por el cable y el segundo se almacena en las tres partículas que forman la cola FIFO del adaptador de puerto.

A continuación, puede ver las depuraciones en el dispositivo receptor (que es simplemente 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
```

Dado que los tamaños de los paquetes de ambos flujos son iguales y en base a la fórmula de horario de programación, deberíamos ver que dos paquetes de nuestra clase pequeña se envían para cada paquete de nuestra clase grande. Esto es exactamente lo que se ve en las depuraciones anteriores.

Test B

Para nuestra segunda prueba, completemos las clases de la siguiente manera:

- Clase chica: hemos configurado el parámetro del ancho de banda en 32 kbps. Se generan los paquetes de 500 bytes desde 7.0.0.200, seguidos por paquetes de 1500 bytes desde 7.0.0.201
- Clase grande – hemos configurado el parámetro del ancho de banda en 16 kbps. La clase almacena un flujo de paquetes de 1500 bytes que llegan desde 7.0.0.1.

El generador de tráfico envía una ráfaga de tráfico a 100 Mbps al Router2 en el orden siguiente:

1. Diez paquetes de 1500 bytes de 7.0.0.1.
2. Diez paquetes de 500 bytes desde 7.0.0.200.
3. Diez paquetes de 1500 bytes de 7.0.0.201.

Se configura FIFO en cada clase.

Verificación del volumen de flujo

El próximo paso es verificar el peso aplicado a los flujos clasificados:

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

Como puede observar en el resultado anterior, los flujos desde 7.0.0.200 y 7.0.0.201 recibieron el mismo peso (128). Este peso es la mitad del tamaño del peso asignado para el flujo desde 7.0.0.1. Esto corresponde al hecho que una clase pequeña tiene el doble de ancho de banda que el tamaño de nuestra clase grande.

Verificación de la distribución de la banda ancha

Podemos producir las siguientes depuraciones desde el dispositivo de pileta.

```
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
```

En este escenario, los flujos en nuestra pequeña clase no tienen el mismo tamaño de paquete. Por lo tanto, la distribución del paquete no es tan trivial como para la prueba A arriba.

Tiempos de planificación

Observemos en más detalle los horarios de programación para cada paquete. El tiempo de programación para los paquetes se calcula usando la siguiente fórmula:

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

Para distintos tamaños de paquetes, el tiempo de programación emplea la siguiente fórmula:

```
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
```

A partir de estas fórmulas, podemos ver que los seis paquetes de 500 bytes de nuestra clase pequeña son transmitidos por cada paquete de 1500 bytes de nuestra clase mayor (mostrada en la salida de depuración de arriba).

También podemos ver que dos paquetes de 1500 bytes de nuestra clase pequeña se envían para un paquete de 1500 bytes de nuestra clase mayor (mostrada en la salida de depuración de arriba).

De las pruebas anteriores, concluimos lo siguiente:

- El tamaño del anillo de transmisión (TX-ring-limit) determina qué tan rápido comienza a funcionar el mecanismo de colocación en cola. Podemos ver el impacto con el aumento de RTT de ping cuando aumenta el límite del anillo de transmisión. Por lo tanto, si implementa CBWFQ o la Puesta en cola de latencia baja [LLQ], considere reducir el límite del anillo de transmisión.
- CBWFQ permite compartir equitativamente el ancho de banda de la interfaz entre diferentes clases.

Información Relacionada

- Colocación en cola equilibrada ponderada por VC en función de la clase (CBWFQ por VC) en los routers Cisco 7200, 3600 y 2600
- Colocación en cola equilibrada ponderada basada en clase por VC en plataformas basadas en RSP

- Introducción a Weighted Fair Queuing en ATM
- Soporte de Tecnología de Clase de Servicio de IP a ATM
- Soporte de Tecnología ATM
- Soporte Técnico y Documentación - Cisco Systems