

Introducción a Weighted Fair Queuing en ATM

Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Convenciones](#)

[Diagrama de la red](#)

[Cómo Establecer el Límite del Anillo de Transmisión](#)

[Impacto del límite del anillo de transmisión](#)

[Ejemplo A:](#)

[Ejemplo B](#)

[Cómo calcular el peso](#)

[Cómo calcular el tiempo de programación](#)

[Cómo funciona WFQ](#)

[¿Qué es una partícula?](#)

[Test A](#)

[Test B](#)

[Etapa 1](#)

[Etapa 2](#)

[Etapa 3](#)

[Etapa 4](#)

[Summary](#)

[Información Relacionada](#)

Introducción

Este documento proporciona una introducción a la cola de tráfico que utiliza la tecnología de cola equilibrada ponderada (WFQ).

WFQ se introdujo para habilitar links de baja velocidad, como links seriales para proporcionar un tratamiento justo para todos los tipos de tráfico. Para hacerlo, WFQ clasifica el tráfico en diferentes flujos (también conocidos como conversaciones) basándose en información de capa tres y capa cuatro, como direcciones IP y puertos TCP. Esto se hace sin el requisito de que defina las 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.

Pero 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. Class-based weighted fair queuing (CBWFQ) provee una solución para estas limitaciones.

A diferencia de WFQ estándar, CBWFQ le permite definir clases de tráfico. También puede aplicarles parámetros, como ancho de banda y límites de cola. El ancho de banda que 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 entonces a las clases, que pueden incluir varios flujos, en lugar de los propios flujos.

Consulte estos documentos para obtener más información sobre cómo configurar CBWFQ:

- [Cola equilibrada ponderada basada en clase por VC \(CBWFQ por VC\) en routers Cisco 7200, 3600 y 2600](#)
- [Cola equilibrada ponderada por VC basada en la clase en plataformas basadas en RSP](#)

Las interfaces ATM no soportan WFQ nativo basado en flujo configurado directamente en una interfaz con el comando **fair-queue**. Pero, con el software que soporta CBWFQ, puede configurar WFQ basado en flujo dentro de la clase predeterminada, como se muestra en este ejemplo:

```
policy-map test
  class class-default
    fair-queue
!
interface ATMx/y.z point-to-point
 ip address a.b.c.d M.M.M.M
 pvc A/B
  service-policy output test
```

[Prerequisites](#)

[Requirements](#)

No hay requisitos específicos para este documento.

[Componentes Utilizados](#)

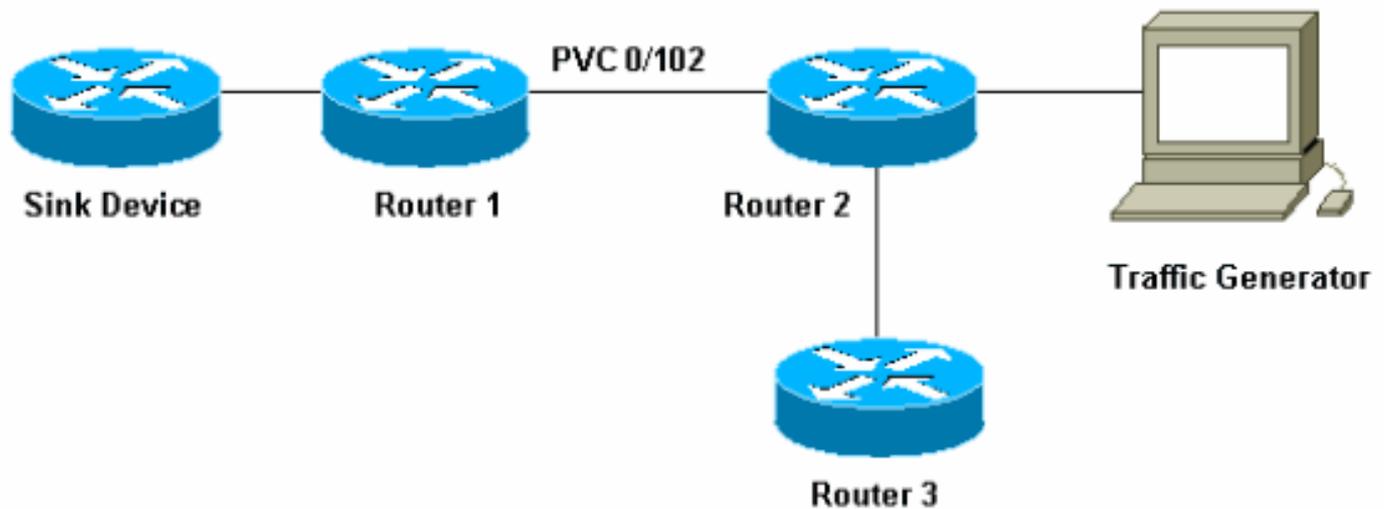
Este documento no tiene restricciones específicas en cuanto a versiones de software y de hardware.

[Convenciones](#)

Consulte [Convenciones de Consejos Técnicos Cisco para obtener más información sobre las convenciones del documento.](#)

[Diagrama de la red](#)

Utilice esta configuración para ilustrar cómo funciona WFQ:



En esta configuración, los paquetes se pueden almacenar en una de estas dos colas:

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

La cola FIFO en el adaptador del puerto guarda los paquetes antes de que éstos sean segmentados en celdas para su transmisión. Cuando esta cola está completa, el módulo de red o adaptador del puerto le indica al software del IOS que la cola está congestionada. El mecanismo se llama contrapresión. Al recibir esta señal, el router se detiene para enviar paquetes a la cola FIFO de la interfaz y almacena los paquetes en el software IOS hasta que la cola se descongestionó nuevamente. Cuando los paquetes se almacenan en el IOS, el sistema puede aplicar QoS.

Cómo Establecer el 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 *<tamaño>* que puede especificar aquí es un número de paquetes, para los routers Cisco 2600 y 3600, o cantidad de partículas, para los routers Cisco 7200 y 7500.

La reducción del tamaño del anillo de transmisión tiene dos ventajas:

- Reduce la cantidad de tiempo que los paquetes esperan en la cola FIFO antes de que se segmente.

En este ejemplo, el anillo de transmisión se establece en 40 (tx-ring-limit=40). Esto es lo que ve cuando utiliza 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 ms
```

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). A partir de esto, puede deducir que un límite de anillo de transmisión grande puede provocar retrasos significativos en la transmisión.

Cómo calcular el peso

En el resultado `show queue atm` en el Ejemplo A, verá que se asigna un peso a cada conversación. Observe esto con más detalle:

```
router2#show queue ATM 4/0.102
Interface ATM4/0.102 VC 0/102
Queuing strategy: weighted fair
Total output drops per VC: 1505772
Output queue: 65/512/64/1505772 (size/max total/threshold/drops)
Conversations 2/3/16 (active/max active/max total)
Reserved Conversations 0/0 (allocated/max allocated)

(depth/weight/discards/tail drops/interleaves) 1/32384/0/0/0
Conversation 2, linktype: ip, length: 58
source: 8.0.0.1, destination: 6.6.6.6, id: 0x2DA1, ttl: 254, prot: 1

(depth/weight/discards/tail drops/interleaves) 64/32384/1505776/0/0
Conversation 15, linktype: ip, length: 1494
source: 7.0.0.1, destination: 6.6.6.6, id: 0x0000, ttl: 63, prot: 255
```

Cuando utiliza WFQ, puede calcular el peso de cada conversación con el uso de esta fórmula:

- $\text{weight} = \frac{32384}{(\text{precedence} + 1)}$ - para Cisco IOS Software Release 12.0(5)T y posterior.
- $\text{weight} = \frac{4096}{(\text{precedence} + 1)}$ - para las versiones del software Cisco IOS anteriores a 12.0(5)T.

Cómo calcular el tiempo de programación

Ahora puede utilizar estos pesos para calcular el tiempo de programación de cada paquete, cuando el paquete se reenvía de la cola del IOS al adaptador de puerto o a la cola FIFO del módulo de red.

Puede calcular el tiempo de programación de salida con el uso de esta fórmula, donde `queue_tail_time` es el tiempo de programación actual:

tiempo de programación de salida = `queue_tail_time + pktsize*weight`

Cómo funciona WFQ

Esta sección explica cómo funciona WFQ. El principio de WFQ es que los paquetes con un peso pequeño, o los paquetes pequeños, deben tener prioridad cuando se envían.

Cree un flujo que comprenda diez paquetes grandes y cuatro paquetes más pequeños (de 82 bytes) que utilicen un generador de tráfico para verificar esto.

En este ejemplo, el router 2 es un router Cisco 7200 con un PA-A3 (adaptador de puerto ATM). Esto es importante porque el tamaño de la cola FIFO de salida en el adaptador de puerto se expresa en partículas y no en paquetes. Vea la sección [¿Qué es una partícula?](#) para obtener más información.

¿Qué es una partícula?

En lugar de asignar un fragmento de memoria contigua para un búfer, el almacenamiento en búfer de partículas asigna partes de memoria discontinuas (dispersas), llamadas partículas, y luego las enlaza para formar un búfer de paquetes lógico. A esto se lo llama memoria intermedia de partículas. En tal esquema, un paquete puede esparcirse en partículas múltiples.

En el router 7200, el tamaño de partícula es de 512 bytes.

Utilice el comando **show buffers** para verificar si los routers Cisco 7200 utilizan partículas:

```
router#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
ATM2/0 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

Estas son algunas pruebas para ilustrar la funcionalidad de WFQ. En esta primera prueba, observe si el ancho de banda se puede compartir entre diferentes conversaciones.

En esta prueba, hizo que el generador de tráfico enviara el tráfico lo suficientemente rápido como para sobrecargar el PVC 0/102 entre el router1 y el router2. Realice un ping del router3 al router1 a través del mismo PVC:

```
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:
..... (WFQ is enabled here)!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!.
Success rate is 98 percent (604/613), round-trip min/avg/max = 164/190/232 ms

```

Como se puede ver en la salida mostrada, hasta que WFQ esté habilitado en la interfaz, el tráfico evita que otros tráficos atraviesen y deja inactiva la línea. Tan pronto como la WFQ se habilita, el ping es exitoso.

Puede ver de esto que, con el uso de WFQ, el ancho de banda se puede compartir entre diferentes conversaciones sin que una bloquee las otras.

Test B

Así es *como* se comparte el ancho de banda.

El flujo que envía el generador de tráfico es una ráfaga compuesta por diez paquetes grandes, seguidos de cuatro paquetes más pequeños de 82 bytes. Esto se envía a 100 Mbps al router 2. Cuando envía la ráfaga, la cola de salida en la interfaz ATM del router2 está vacía. El Router2 envía estos paquetes a través de un PVC de 10 KB (es un PVC muy lento) para asegurarse de que la congestión se produzca en la cola de salida.

Realice esta prueba en varias etapas para simplificar este proceso:

Etapa 1

El tráfico grande comprende diez paquetes de 482 bytes. Como las partículas en PA-A3 son 512 bytes, cada paquete, ya sea grande o pequeño, debe tomar una partícula cuando se lo almacena en la cola de salida del adaptador de puerto. El límite del anillo de transmisión del router es tres (tx-ring-limit=3). Este es un ejemplo de lo que se puede ver en el dispositivo receptor:

```

.Nov 7 15:39:13.776: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, len 482, rcvd 4
.Nov 7 15:39:13.776: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, unknown protocol
.Nov 7 15:39:14.252: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4
.Nov 7 15:39:14.252: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, unknown protocol
.Nov 7 15:39:14.732: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4
.Nov 7 15:39:14.732: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, unknown protocol
.Nov 7 15:39:15.208: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4
.Nov 7 15:39:15.208: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, unknown protocol

!--- Congestion occurs at this point. .Nov 7 15:39:15.512: IP: s=7.0.0.200 (FastEthernet0/1),
d=6.6.6.6, Len 82, rcvd 4 .Nov 7 15:39:15.516: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len
82, unknown protocol .Nov 7 15:39:15.644: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82,
rcvd 4 .Nov 7 15:39:15.644: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, unknown
protocol .Nov 7 15:39:15.776: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, rcvd 4 .Nov
7 15:39:15.776: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, unknown protocol .Nov 7
15:39:15.904: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, rcvd 4 .Nov 7 15:39:15.904:
IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, unknown protocol .Nov 7 15:39:16.384: IP:
s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4 .Nov 7 15:39:16.384: IP: s=7.0.0.1
(FastEthernet0/1), d=6.6.6.6, Len 482, unknown protocol .Nov 7 15:39:16.860: IP: s=7.0.0.1
(FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4 .Nov 7 15:39:16.860: IP: s=7.0.0.1
(FastEthernet0/1), d=6.6.6.6, Len 482, unknown protocol .Nov 7 15:39:17.340: IP: s=7.0.0.1
(FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4 .Nov 7 15:39:17.340: IP: s=7.0.0.1

```

```
(FastEthernet0/1), d=6.6.6.6, Len 482, unknown protocol .Nov 7 15:39:17.816: IP: s=7.0.0.1
(FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4 .Nov 7 15:39:17.820: IP: s=7.0.0.1
(FastEthernet0/1), d=6.6.6.6, Len 482, unknown protocol .Nov 7 15:39:18.296: IP: s=7.0.0.1
(FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4 .Nov 7 15:39:18.296: IP: s=7.0.0.1
(FastEthernet0/1), d=6.6.6.6, Len 482, unknown protocol .Nov 7 15:39:18.776: IP: s=7.0.0.1
(FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4 .Nov 7 15:39:18.776: IP: s=7.0.0.1
(FastEthernet0/1), d=6.6.6.6, Len 482, unknown protocol
```

Puede ver los cuatro paquetes de 482 bytes enviados antes de los paquetes de 82 bytes, que normalmente deberían obtener la prioridad. Por eso sucede esto.

Dado que la ráfaga se compone principalmente de diez paquetes de 482 bytes, éstos llegan primero al router, seguidos por los paquetes de 82 bytes. Dado que los paquetes de 482 bytes llegan en un momento en el que no hay congestión, ya que no hay tráfico, un paquete se coloca inmediatamente en la cola de la segmentación y reensamblado del adaptador de puerto (SAR) para que se fragmente en las celdas y se envíe en el cable. En otras palabras, el anillo de transmisión continúa vacío.

Puede calcular que el tiempo necesario para enviar un paquete de 482 bytes es mayor que el tiempo necesario para el generador de tráfico para enviar la ráfaga total. Por lo tanto, puede suponer que, cuando el primer paquete de 482 bytes se pone en cola en el adaptador de puerto, ya hay más paquetes de 482 bytes de la ráfaga en el router. Por lo tanto, se pueden enviar a la cola del anillo de transmisión más paquetes de 482 bytes. Tres paquetes más de 482 bytes se ponen en cola con el uso de las tres partículas libres presentes allí.

Nota: Los paquetes se ponen en cola en el anillo de transmisión tan pronto como hay una partícula libre, incluso si necesitan que se almacene más de una partícula.

En este punto, hay congestión, ya que las tres partículas están llenas. Entonces, se inicia el almacenamiento en cola en el IOS. Cuando los cuatro paquetes de 82 bytes finalmente alcanzan el router, hay congestión. Estos cuatro paquetes se colocan en la cola y se utiliza WFQ en los dos flujos. Observe la cola ATM que utiliza el comando **show queue ATM** para ver esto:

```
router2#show queue ATM 4/0.102 vc 0/102
  Interface ATM4/0.102 VC 0/102
  Queuing strategy: weighted fair
  Total output drops per VC: 0
  Output queue: 10/512/64/0 (size/max total/threshold/drops)
    Conversations 2/4/16 (active/max active/max total)
    Reserved Conversations 0/0 (allocated/max allocated)

(depth/weight/total drops/no-buffer drops/interleaves) 4/32384/0/0/0
  Conversation 6, linktype: ip, length: 82
  source: 7.0.0.200, destination: 6.6.6.6, id: 0x0000, ttl: 63, prot: 255

(depth/weight/total drops/no-buffer drops/interleaves) 6/32384/0/0/0
  Conversation 15, linktype: ip, length: 482
  source: 7.0.0.1, destination: 6.6.6.6, id: 0x0000, ttl: 63, prot: 255
```

Puede ver en las depuraciones que los primeros cuatro paquetes de 482 bytes son seguidos por los paquetes de 82 bytes. Estos paquetes pequeños salen del router antes que los paquetes más grandes. Esto indica que, en cuanto ocurre la congestión, los paquetes pequeños tienen prioridad sobre los más grandes.

Utilice las fórmulas de peso y de horario de programación proporcionadas en la sección [Cálculo del peso](#) para verificar esto.

Etapa 2

Si aumenta el límite del anillo de transmisión a cinco y los paquetes grandes son 482 bytes entonces, de acuerdo con la salida anterior, debería ver seis paquetes de 482 bytes antes de que se produzca la congestión, seguidos de cuatro paquetes de 82 bytes, luego otros cuatro paquetes de 482 bytes:

```
.Nov 7 15:49:57.365: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4
.Nov 7 15:49:57.365: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, unknown protocol
.Nov 7 15:49:57.841: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4
.Nov 7 15:49:57.845: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, unknown protocol
.Nov 7 15:49:58.321: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4
.Nov 7 15:49:58.321: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, unknown protocol
.Nov 7 15:49:58.797: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4
.Nov 7 15:49:58.801: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, unknown protocol
.Nov 7 15:49:59.277: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4
.Nov 7 15:49:59.277: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, unknown protocol
.Nov 7 15:49:59.757: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4
.Nov 7 15:49:59.757: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, unknown protocol
.Nov 7 15:49:59.973: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, rcvd 4
.Nov 7 15:49:59.973: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, unknown protocol
.Nov 7 15:50:00.105: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, rcvd 4
.Nov 7 15:50:00.105: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, unknown protocol
.Nov 7 15:50:00.232: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, rcvd 4
.Nov 7 15:50:00.232: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, unknown protocol
.Nov 7 15:50:00.364: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, rcvd 4
.Nov 7 15:50:00.364: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, unknown protocol
.Nov 7 15:50:00.840: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4
.Nov 7 15:50:00.844: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, unknown protocol
.Nov 7 15:50:01.320: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4
.Nov 7 15:50:01.320: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, unknown protocol
.Nov 7 15:50:01.796: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4
.Nov 7 15:50:01.800: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, unknown protocol
.Nov 7 15:50:02.276: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, rcvd 4
.Nov 7 15:50:02.276: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 482, unknown protocol
```

Como pueden ver aquí, esto es lo que sucede.

Etapa 3

El tamaño de la partícula es de 512 bytes. Por lo tanto, si el anillo de transmisión se expresa en partículas, y usted utiliza paquetes ligeramente más grandes que el tamaño de la partícula, cada uno toma dos partículas. Esto se ilustra con el uso de paquetes de 582 bytes y un anillo de transmisión de tres. Con estos parámetros, debe ver tres paquetes de 582 bytes. Uno se envía sin tráfico en la interfaz ATM, lo que deja tres partículas libres. Por lo tanto, dos o más paquetes pueden ser almacenados en cola, seguidos por cuatro paquetes de 82 bytes y luego siete paquetes de 582 bytes:

```
.Nov 7 15:51:34.604: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 582, rcvd 4
.Nov 7 15:51:34.604: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 582, unknown protocol
.Nov 7 15:51:35.168: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 582, rcvd 4
.Nov 7 15:51:35.168: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 582, unknown protocol
.Nov 7 15:51:35.732: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 582, rcvd 4
.Nov 7 15:51:35.736: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 582, unknown protocol
.Nov 7 15:51:35.864: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, rcvd 4
.Nov 7 15:51:35.864: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, unknown protocol
.Nov 7 15:51:35.996: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, rcvd 4
```

```

.Nov 7 15:51:35.996: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, unknown protocol
.Nov 7 15:51:36.124: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, rcvd 4
.Nov 7 15:51:36.124: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, unknown protocol
.Nov 7 15:51:36.256: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, rcvd 4
.Nov 7 15:51:36.256: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, unknown protocol
.Nov 7 15:51:36.820: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 582, rcvd 4
.Nov 7 15:51:36.820: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 582, unknown protocol
.Nov 7 15:51:37.384: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 582, rcvd 4
.Nov 7 15:51:37.388: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 582, unknown protocol
.Nov 7 15:51:37.952: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 582, rcvd 4
.Nov 7 15:51:37.952: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 582, unknown protocol
.Nov 7 15:51:38.604: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 582, rcvd 4
.Nov 7 15:51:38.604: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 582, unknown protocol
.Nov 7 15:51:39.168: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 582, rcvd 4
.Nov 7 15:51:39.168: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 582, unknown protocol
.Nov 7 15:51:39.732: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 582, rcvd 4
.Nov 7 15:51:39.736: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 582, unknown protocol
.Nov 7 15:51:40.300: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 582, rcvd 4
.Nov 7 15:51:40.300: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 582, unknown protocol

```

Etapa 4

Tome un tamaño de paquete de 1482 (tres partículas) y defina un anillo de transmisión de cinco. Si el anillo de transmisión se define en partículas, verá algo similar:

- Un paquete transmitido inmediatamente
- Un paquete que toma tres de las cinco partículas
- Un paquete en cola porque dos partículas están libres

```

.Nov 8 07:22:41.200: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4
.Nov 8 07:22:41.200: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, unknown protocol
.Nov 8 07:22:42.592: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4
.Nov 8 07:22:42.592: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, unknown protocol
.Nov 8 07:22:43.984: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4
.Nov 8 07:22:43.984: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, unknown protocol
.Nov 8 07:22:44.112: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, rcvd 4
.Nov 8 07:22:44.112: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, unknown protocol
.Nov 8 07:22:44.332: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, rcvd 4
.Nov 8 07:22:44.332: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, unknown protocol
.Nov 8 07:22:44.460: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, rcvd 4
.Nov 8 07:22:44.460: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, unknown protocol
.Nov 8 07:22:44.591: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, rcvd 4
.Nov 8 07:22:44.591: IP: s=7.0.0.200 (FastEthernet0/1), d=6.6.6.6, Len 82, unknown protocol
.Nov 8 07:22:45.983: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4
.Nov 8 07:22:45.983: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, unknown protocol
.Nov 8 07:22:47.371: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4
.Nov 8 07:22:47.375: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, unknown protocol
.Nov 8 07:22:48.763: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4
.Nov 8 07:22:48.767: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, unknown protocol
.Nov 8 07:22:50.155: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4
.Nov 8 07:22:50.155: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, unknown protocol
.Nov 8 07:22:51.547: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4
.Nov 8 07:22:51.547: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, unknown protocol
.Nov 8 07:22:53.027: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4
.Nov 8 07:22:53.027: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, unknown protocol
.Nov 8 07:22:54.415: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, rcvd 4
.Nov 8 07:22:54.419: IP: s=7.0.0.1 (FastEthernet0/1), d=6.6.6.6, Len 1482, unknown protocol

```

Summary

A partir de las pruebas realizadas, puede concluir lo siguiente:

- En los PVC lentos sin WFQ, el tráfico masivo impacta en el tráfico pequeño, como los pings que se están paralizando hasta que se habilita WFQ.
- El tamaño del anillo de transmisión (tx-ring-limit) determina la rapidez con la que el mecanismo de colocación en cola comienza a hacer su trabajo. Puede ver el impacto de esto con el aumento del ping RTT cuando aumenta el límite del anillo de transmisión. Por lo tanto, si se necesita implementar WFQ o LLQ, tiene sentido reducir el límite de anillo de transmisión.
- WFQ que utiliza CBWFQ prioriza los tráfico pequeños sobre el tráfico masivo.

[Información Relacionada](#)

- [Páginas de soporte de la tecnología ATM](#)
- [Descripción general de la administración de congestión](#)
- [Soporte Técnico y Documentación - Cisco Systems](#)