De betekenis van Weighted Fair Queuing op ATM

Inhoud

Inleiding

Dit document biedt een inleiding tot traffic wachting die gebruik maakt van de WFQ-technologie (Webex Fair Quing).

WFQ werd geïntroduceerd om trage snelheids-verbindingen mogelijk te maken, zoals seriële verbindingen om een eerlijke behandeling te bieden voor alle soorten verkeer. Om dit te doen, classificeert WFQ het verkeer in verschillende stromen (ook bekend als gesprekken) gebaseerd op laag drie en laag vier informatie, zoals IP adressen en TCP poorten. Dit gebeurt zonder de verplichting van u om toegangslijsten te definiëren. Dit betekent dat laagbandbreedteverkeer effectief prioriteit heeft boven hoogbandbreedteverkeer omdat het hoge bandbreedteverkeer de transmissiemedia in verhouding tot zijn toegewezen gewicht deelt.

Maar WFQ heeft bepaalde beperkingen:

- Het is niet schaalbaar indien het stroombedrag aanzienlijk stijgt.
- Native WFQ is niet beschikbaar op snelle interfaces zoals ATM-interfaces.

Op klasse gebaseerde weging in de wachtrij (CBWFQ) biedt een oplossing voor deze beperkingen.

In tegenstelling tot standaard WFQ staat CBWFQ u toe om verkeersklassen te definiëren. U kunt ook parameters, zoals bandbreedte en wachtrij-limieten, op hen toepassen. De bandbreedte die u aan een klasse toevoegt wordt gebruikt om het gewicht van die klasse te berekenen. Het gewicht van elk pakje dat aan de klassecriteria voldoet, wordt ook hieruit berekend. Het WFQ wordt vervolgens toegepast op de klassen, die meerdere stromen kunnen omvatten, in plaats van de stromen zelf.

Raadpleeg deze documenten voor meer informatie over de configuratie van CBWFQ:

- Op klasse gebaseerde, Weighted Fair Queuing (Per-VC CBWFQ) op Cisco 7200, 3600 en 2600 routers
- Op klasse 1-VC gebaseerde, gewogen wachtrij op RSP-gebaseerde platforms

ATM-interfaces ondersteunen geen native flow-gebaseerde WFQ die rechtstreeks op een interface zijn geconfigureerd met de opdracht **in een wachtrij**. Maar met de software die CBWFQ ondersteunt, kunt u op flow gebaseerde WFQ binnen de standaardklasse configureren, zoals in dit voorbeeld wordt getoond:

```
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
```

Voorwaarden

Vereisten

Er zijn geen specifieke vereisten van toepassing op dit document.

Gebruikte componenten

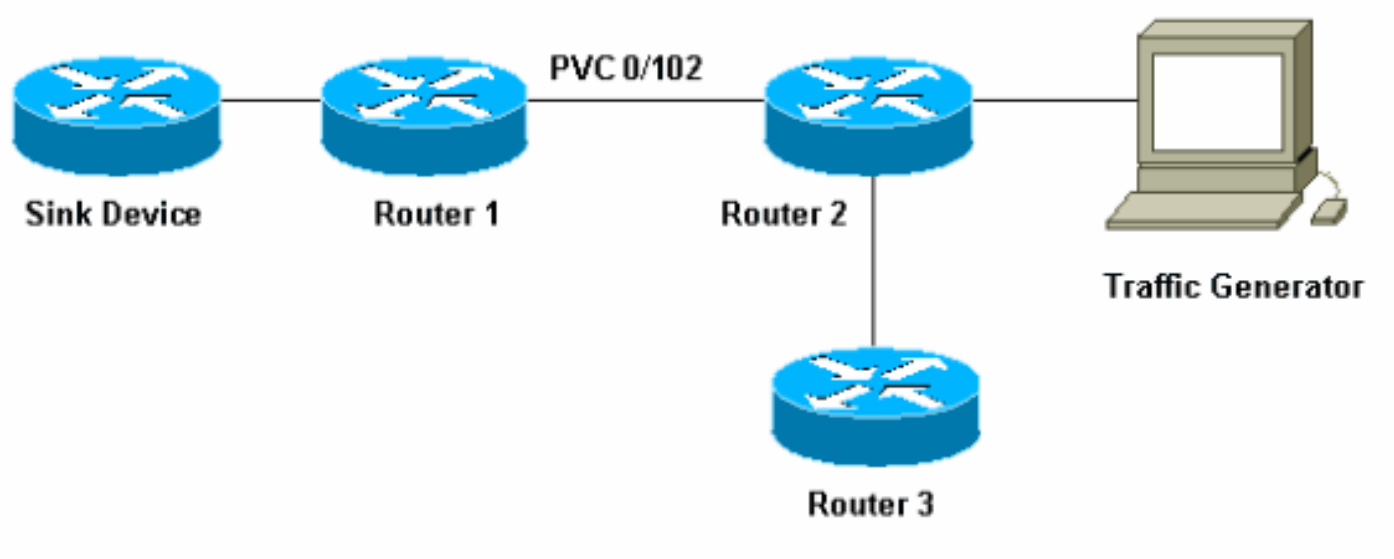
Dit document is niet beperkt tot specifieke software- en hardware-versies.

Conventies

Raadpleeg Cisco Technical Tips Conventions (Conventies voor technische tips van Cisco) voor meer informatie over documentconventies.

Netwerkdiagram

Gebruik deze instelling om aan te geven hoe WFQ werkt:



In deze instelling kunnen pakketten in een van deze twee wachtrijen worden opgeslagen:

- De hardware eerst in first out (FIFO) wachtrij op de poortadapter en de netwerkmodule
- De rij in de software van Cisco IOS®, op het geheugen van de router in/uitvoer [I/O], waar Quality of Service (QoS) functies zoals CBWFQ kunnen worden toegepast

De FIFO-wachtrij op de poortadapter slaat de pakketten op voordat ze in cellen zijn gesegmenteerd voor verzending. Wanneer deze rij vol is, signalen van de poortadapter of van de netwerkmodule aan de IOS software die de rij wordt geblokkeerd. Dit mechanisme heet "tegendruk". Na ontvangst van dit signaal, stopt de router om pakketten naar de interface-FIFO-wachtrij te verzenden en slaat u de pakketten in de IOS-software op totdat de wachtrij opnieuw wordt onverstopt. Wanneer de pakketten in IOS worden opgeslagen, kan het systeem QoS toepassen.

Instellen van de transmissielimiet

Eén probleem met dit wachtend mechanisme is dat, hoe groter de FIFO-wachtrij op de interface, hoe langer de vertraging voordat pakketten aan het einde van deze wachtrij worden verzonden, kan worden uitgesteld. Dit kan ernstige prestatieproblemen veroorzaken voor vertragingsgevoelig verkeer zoals spraakverkeer.

Met de opdracht permanent virtueel circuit (PVC) **tx-ring** kunt u de grootte van de FIFO-wachtrij beperken.

```
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
```

<size> u kunt hier specificeren dat er een aantal pakketten is, voor Cisco 2600 en 3600 routers of hoeveelheid deeltjes, voor Cisco 7200 en 7500 routers.

De vermindering van de grootte van de verzendring heeft twee voordelen:

- De tijd dat pakketten in de FIFO-wachtrij staan, wordt verkort voordat deze wordt gesegmenteerd.

- Het versnelt het gebruik van QoS in de IOS-software.

Invloed van de transmissielimiet

Kijk naar het effect van de zend ring limiet die de instellingen gebruikt die in het vorige netwerkdiagram worden getoond. Ga ervan uit:

- De verkeersgenerator verstuurt verkeer (1500 byte-pakketten) naar het gootsteenapparaat en dit verkeer overbelast PVC 0/102 tussen router1 en router2.
- Router3 probeert router1 te pingelen.
- WFQ is geactiveerd op router2.

Bekijk twee configuraties die verschillende radiogolven gebruiken om de impact te zien die dit heeft.

Voorbeeld

In dit voorbeeld, stelt u de zendring in op drie (tx-ring-limit=3). Dit is wat u ziet wanneer u router1 van router3 pingelt:

```
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

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
   !--- ping (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
   !--- This is traffic from the traffic generator.
```

Voorbeeld B

In dit voorbeeld, stelt u de zendring in op 40 (tx-ring-limit=40). Dit is wat u ziet wanneer u dezelfde ping gebruikt als in Voorbeeld 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
```

Zoals u hier kunt zien, hoe groter de limiet van de verzendring, hoe groter de 'ping round-trip' tijd (RTT). U kunt hieruit concluderen dat een grote zendring limiet kan leiden tot aanzienlijke vertragingen bij de transmissie.

Hoe wordt het gewicht berekend

In de **uitvoer van ATM in de wachtrij** in Voorbeeld A ziet u dat aan elk gesprek een gewicht wordt toegewezen. Bekijk dit in detail:

```
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
```

Wanneer u WFQ gebruikt, kunt u het gewicht van elke conversatie berekenen met het gebruik van deze formule:

- **gewicht=32384/(voorrang+1)** - voor Cisco IOS-softwarerelease 12.0(5)T en hoger.
- **gewicht=4096/(voorrang+1)** - voor Cisco IOS-softwarereleases vóór 12.0(5)T.

Hoe de planningtijd te berekenen

U kunt deze gewichten nu gebruiken om de planningstijd van elk pakket te berekenen, wanneer het pakket van de IOS rij naar de poortadapter of de netwerkmodule FIFO-wachtrij wordt doorgestuurd.

U kunt de uitvoerplanningstijd met het gebruik van deze formule berekenen, waarbij **wachtrij_tail_time** de huidige planningtijd is:

uitvoerplanningstijd= wachtrij_staarttijd + tijdsgrootte*gewicht

Hoe WFQ werkt

In dit deel wordt uitgelegd hoe WFQ werkt. Het principe van WFQ is dat pakketten met een klein gewicht, of kleine pakketten, prioriteit moeten krijgen wanneer zij worden verzonden.

Maak een stroom die tien pakketten bestaat uit grote pakketten en vier kleinere pakketten (van 82 bytes) die een verkeersgenerator gebruikt om dit te controleren.

In dit voorbeeld is router2 een Cisco 7200 router met een PA-A3 (ATM poortadapter). Dit is belangrijk omdat de grootte van de FIFO-wachtrij op de poortadapter in deeltjes wordt uitgedrukt en niet in pakketten. Zie je Wat is een deeltje? voor meer informatie.

Wat is een Deeltaart?

In plaats van de toewijzing van één stuk aangrenzend geheugen voor een buffer, wijst het bufferen van deeltjes distigueuze (verspreid) stukken geheugen toe, noemde deeltjes, en koppelt ze vervolgens aan elkaar om één logische pakketbuffer te vormen. Dit heet een deeltjesbuffer. In zo'n schema kan een pakje dan over meerdere deeltjes worden verspreid.

In de 7200 router is de deeltjesgrootte 512 bytes.

Gebruik de opdracht **Show buffers** om te verifiëren of Cisco 7200 routers deeltjes gebruiken:

```
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

Dit zijn enkele testen om de WFQ-functie te illustreren. In deze eerste test, kijk of de bandbreedte tussen verschillende gesprekken kan worden gedeeld.

In deze test, maakte u de verkeersgenerator snel genoeg om het verkeer van PVC 0/102 tussen router1 en router2 te overladen. Voer een pingelen van router3 naar router1 over hetzelfde PVC uit:

```
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)!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!.[break]
Success rate is 98 percent (604/613), round-trip min/avg/max = 164/190/232 ms
```

Zoals u kunt zien op de uitvoer die wordt weergegeven, totdat WFQ op de interface is ingeschakeld, voorkomt het verkeer dat ander verkeer door de lijn gaat en sterft het. Zodra WFQ is ingeschakeld, is ping geslaagd.

U kunt hieruit zien dat, met het gebruik van WFQ, de bandbreedte tussen verschillende gesprekken kan worden gedeeld zonder dat de anderen worden geblokkeerd.

Test B

Dit is *hoe* de bandbreedte wordt gedeeld.

De stroom die de verkeersgenerator verstuurt is een breuk die uit tien grote pakketten bestaat, gevolgd door vier kleinere pakketten van 82 bytes. U stuurt dit op 100 Mbps naar router2. Wanneer u de burst verstuurt, is de uitvoerwachtrij op de router2 ATM-interface leeg. Router2 stuurt deze pakketten door een 10 KB PVC (dit is een zeer langzaam PVC) om ervoor te zorgen dat congestie in de uitvoerwachtrij voorkomt.

Deze test in verschillende fasen uitvoeren om dit proces te vereenvoudigen:

Niveau 1

Het grote verkeer bestaat uit tien pakketten van 482 bytes. Omdat de deeltjes op de PA-A3 512 bytes zijn, moet elk pakje, groot of klein, één deeltje nemen wanneer het wordt opgeslagen in de uitvoerwachtrij van de poortadapter. De router heeft een radiogrens van drie (tx-ring-limit=3). Dit is een voorbeeld van wat je kunt zien op het gootsteenapparaat:

```
   .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
```

U kunt de vier pakketten van 482 bytes zien die vóór de pakketten met 82 bytes worden verzonden, die normaal de prioriteit moeten krijgen. Daarom gebeurt dit.

Aangezien de burst voornamelijk uit tien pakketten van 482 bytes bestaat, bereiken deze eerst de router, gevolgd door de pakketten van 82 bytes. Aangezien de 482-byte-pakketten op een tijdstip aankomen waarop er geen stremming is, aangezien er geen verkeer is, wordt één pakje onmiddellijk in de wachtrij geplaatst voor de segmentering en hermontage van de poortadapter (SAR), die in cellen moet worden gehakt en op de draad moet worden verzonden. Met andere woorden, de verzendring is nog leeg.

U kunt berekenen dat de tijd die nodig is om een pakket met 482 bytes te verzenden, groter is dan de tijd die voor de verkeersgenerator nodig is om de totale burst te verzenden. U kunt daarom veronderstellen dat, wanneer het eerste pakket van 482 bytes aan de poortadapter wordt wachtrij wordt geplaatst, er al meer pakketten van 482 bytes van de uitbarsting in de router aanwezig zijn. Daarom kunnen meer pakketten van 482 bytes in de wachtrij worden geplaatst voor de verzendring. Drie extra pakketten van 482 bytes worden in de wachtrij geplaatst met het gebruik van de drie gratis deeltjes die er aanwezig zijn.

Opmerking: Packets worden in de verzendring geplaatst zodra er een vrij deeltje is, zelfs als er meer dan één deeltje moet worden opgeslagen.

Op dit moment is er opstopping, omdat de drie deeltjes vol zijn. Daarom begint een wachtrij in IOS. Wanneer de vier 82-byte pakketten definitief de router bereiken, is er opstopping. Deze vier pakketten worden in de wachtrij geplaatst en WFQ wordt op de twee stromen gebruikt. Kijk naar de ATM-wachtrij die de opdracht **in wachtrij** van **ATM** gebruikt om dit te zien:

```
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
```

U kunt in de media zien dat de eerste vier pakketten van 482 bytes worden gevolgd door de pakketten met 82 bytes. Deze kleine pakketten gaan uit de router vóór de grote pakketten. Dit betekent dat, zodra de congestie zich voordoet, kleine pakketten prioriteit hebben boven grote pakketten.

Gebruik de in het gedeelte Gewicht berekenen en schema's om dit te controleren.

Niveau 2

Als u de limiet van de verzendring verhoogt tot vijf bytes en de grote pakketten 482 bytes zijn, dient u, in overeenstemming met de vorige uitvoer, zes pakketten van 482 bytes te zien voordat er een stremming plaatsvindt, gevolgd door vier pakketten van 82 bytes, dan nog eens vier bytes van 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
```

Zoals je hier kunt zien, is dit inderdaad wat er gebeurt.

Niveau 3

De deeltjesgrootte is 512 bytes. Daarom, als de verzendring in deeltjes wordt uitgedrukt en u pakketten gebruikt die iets groter zijn dan de deeltjesgrootte, neemt elke twee deeltjes. Dit wordt geïllustreerd door het gebruik van pakketten van 582 bytes en een zendring van drie bytes. Met deze parameters moet u drie pakketten van 582 bytes zien. Een wordt verzonden zonder verkeer op de ATM-interface, waardoor drie deeltjes vrij blijven. Daarom kunnen nog twee pakketten in de wachtrij worden geplaatst, gevolgd door vier pakketten met 82 bytes en vervolgens zeven pakketten met 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
```

Niveau 4

Neem een pakketgrootte van 1482 (drie deeltjes) en definieer een verzendring van vijf. Als de verzendring is gedefinieerd in deeltjes, ziet u iets soortgelijks:

- Eén pakket direct verzonden
- Eén pakje dat drie van de vijf deeltjes bevat
- Eén pakje in de wachtrij omdat twee deeltjes vrij zijn

```
.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
```

Samenvatting

Aan de hand van de uitgevoerde tests kunt u de volgende conclusies trekken:

- Bij langzame PVC’s zonder WFQ beïnvloedt het bulkverkeer klein verkeer, zoals pings die

worden gestort totdat WFQ is ingeschakeld.
- De grootte van de zendring (tx-ring-limit) bepaalt hoe snel het wachtend mechanisme zijn baan begint te doen. U kunt het effect hiervan zien met de toename van de ping RTT als de limiet voor de verzendring toeneemt. Daarom is het, als WFQ of LLQ moet worden geïmplementeerd, zinvol om de limiet voor de verzendring te verlagen.
- WFQ dat CBWFQ gebruikt geeft voorrang aan kleine handel boven bulkverkeer.

Gerelateerde informatie

- Ondersteuning van ATM-technologie
- Overzicht van congestiebeheer
- Technische ondersteuning en documentatie – Cisco Systems