

ATM での均等化キューイングについて

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[表記法](#)

[ネットワーク図](#)

[送信リング制限の設定方法](#)

[送信リングの制限の影響](#)

[例 A](#)

[例 B](#)

[重みの計算方法](#)

[スケジュール時間の計算方法](#)

[WFQ の動作の仕組み](#)

[パーティクルとは何か](#)

[テスト A](#)

[テスト B](#)

[第 1 段階](#)

[第 2 段階](#)

[第 3 段階](#)

[第 4 段階](#)

[要約](#)

[関連情報](#)

概要

このドキュメントでは、重み付け均等化キューイング (WFQ) テクノロジーを使用するトラフィック キューイングの概要について説明します。

WFQ は、シリアルリンクなどの低速リンクを有効にして、すべてのタイプのトラフィックに対して公平な処理を提供するために導入されました。これを実現するために、WFQ では、IP アドレスや TCP ポートなどのレイヤ 3 およびレイヤ 4 の情報をベースとして、トラフィックをさまざまなフローに分類します (会話とも呼ばれます)。アクセスリストを定義する必要はありません。このことは、高帯域幅トラフィックは、割り当てられた重みに比例する伝送メディアを共有しているため、事実上、低帯域幅トラフィックは高帯域幅トラフィックよりも優先度が高いことを意味します。

ただし、WFQ には特定の制限があります。

- フローの量がかなり多くなると、拡張性がなくなります。

- ATM インターフェイスなどの高速インターフェイスでは、ネイティブ WFQ は使用できません。

Class-based weighted fair queuing (CBWFQ; クラスベース重み付け均等化キューイング) では、これらの制限に対する解決策が提供されています。

標準の WFQ とは異なり、CBWFQ ではトラフィック クラスを定義できます。帯域幅やキュー制限などのパラメータを適用することもできます。クラスに割り当てる帯域幅は、そのクラスの重みを計算するために使用されます。また、この値からクラスの基準を満たす各パケットの重みも計算されます。その後、WFQ はクラスに適用されます。クラスには、フロー自体ではなく、複数のフローを含めることができます。

CBWFQ の設定方法の詳細については、次のドキュメントを参照してください。

- [Cisco 7200、3600、および2600ルータ上のVC単位クラスベース、重み付け均等化キューイング \(VC単位CBWFQ \)](#)
- [RSP ベースのプラットフォームでの、VC 単位、クラスベースの重み付け均等化キューイング](#)

ATM インターフェイスは、**fair-queue** コマンドを使用してインターフェイス上で直接設定されたネイティブのフローベースの WFQ をサポートしません。ただし、CBWFQ をサポートするソフトウェアでは、次の例に示すように、デフォルトクラス内でフローベースの WFQ を設定できます。

```
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
```

[前提条件](#)

[要件](#)

このドキュメントに特有の要件はありません。

[使用するコンポーネント](#)

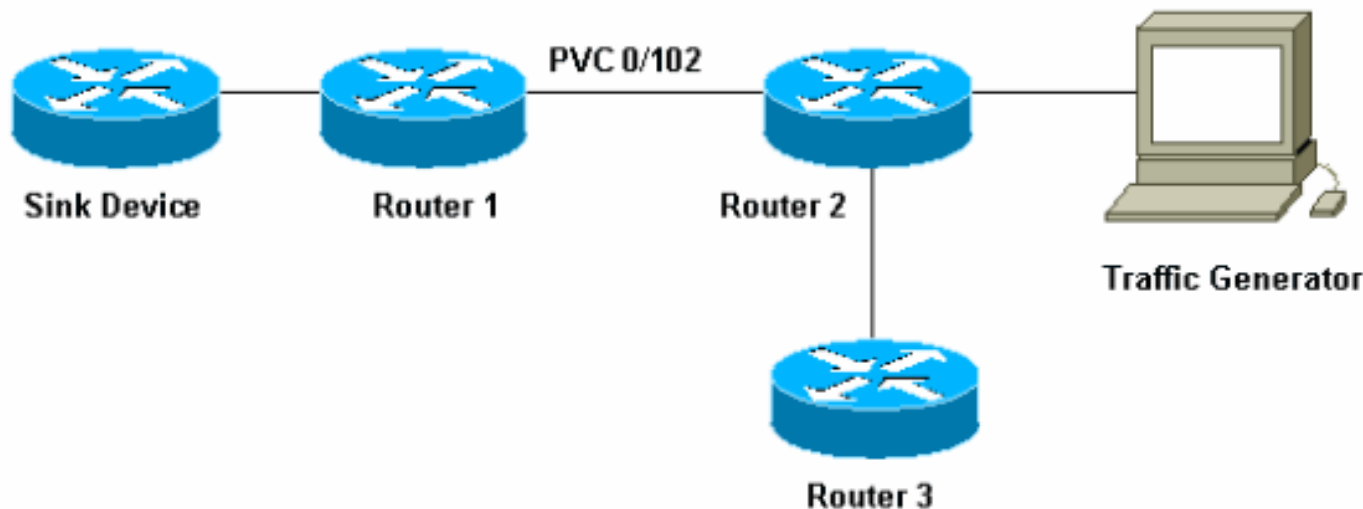
このドキュメントの内容は、特定のソフトウェアやハードウェアのバージョンに限定されるものではありません。

[表記法](#)

ドキュメント表記の詳細は、『[シスコテクニカルティップスの表記法](#)』を参照してください。

[ネットワーク図](#)

WFQ の動作を説明するには、次の設定を使用します。



この設定では、パケットは次の2つのキューのいずれかに格納できます。

- ポートアダプタとネットワークモジュールのハードウェア先入れ先出し(FIFO)キュー
- ルータの入出力(I/O)メモリ上のCisco IOS[®]ソフトウェアのキュー。CBWFQなどのQuality of Service(QoS)機能を適用できます

ポートアダプタにあるFIFOキューには、パケットが送信用にセルにセグメント化される前に格納されます。このキューがいっぱいになると、ポートアダプタまたはネットワークモジュールからIOSソフトウェアに対して、キューが輻輳しているという信号が出されます。このメカニズムは、バックプレッシャと呼ばれます。この信号を受信すると、ルータはインターフェイスのFIFOキューへのパケットの送信を停止し、キューが再び輻輳していないまで、パケットをIOSソフトウェアに保存します。パケットがIOSに保存されているときには、システムはQoSを適用できません。

送信リング制限の設定方法

このキューイングメカニズムに関連する問題の1つは、インターフェイス上のFIFOキューが大きいほど、このキューの最後にあるパケットが送信されるまでの遅延時間が長くなることです。これにより、音声トラフィックなどの遅延の影響を受けやすいトラフィックには、重大なパフォーマンス問題が生じます。

Permanent Virtual Circuit (PVC; 相手先固定接続) の tx-ring-limit コマンドを使用すると、FIFOキューのサイズを小さくすることができます。

```
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
```

ここで指定できる <size> は、Cisco 2600および3600ルータのパケット数、またはCisco 7200および7500ルータのパーティクル数です。

送信リングのサイズを小さくすると、次の2つの利点があります。

- パケットがセグメント化される前にFIFOキューで待機する時間が短縮されます。
- IOSソフトウェアでのQoSの使用が促進されます。


```
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
```

上記の出力からわかるように、送信リングの制限が大きくなれば、pingの Round-Trip Time (RTT; ラウンドトリップ時間) も長くなります。これにより、送信リングの制限が大きくなると、送信に大幅な遅延が発生する可能性があることが推測できます。

重みの計算方法

例Aのshow queue atmの出力では、各会話に重みが割り当てられています。詳細については、次を参照してください。

```
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
```

WFQを使用する場合は、次の式を使用して各会話の重みを計算できます。

- $\text{weight} = 32384 / (\text{precedence} + 1)$: Cisco IOSソフトウェアリリース12.0(5)T以降の場合。
- $\text{重み} = 4096 / (\text{優先順位} + 1)$ - Cisco IOSソフトウェアリリース12.0(5)T以前の場合。

スケジューリング時間の計算方法

これらの重み付けを使用して、IOSキューからポートアダプタまたはネットワークモジュールのFIFOキューにパケットが転送されたときの各パケットのスケジューリング時間を計算できます。

次の式を使用して、出力スケジューリング時間を計算できます。ここで、queue_tail_timeは現在のスケジューリング時間です。

出力スケジューリング時間 = queue_tail_time + パケットサイズ * 重み

WFQの動作の仕組み

このセクションでは、WFQの動作について説明します。WFQの原則は、小さな重み付けのパケ

Success rate is 98 percent (604/613), round-trip min/avg/max = 164/190/232 ms

図の出力からわかるように、インターフェイスでWFQが有効になるまで、トラフィックは他のトラフィックの通過を阻止し、回線を起動します。WFQが有効にされると、直ちに ping は成功します。

このことから、WFQを使用すると、帯域幅を異なる会話の間で共有でき、他の会話をブロックする必要はありません。

テスト B

これは帯域幅の共有方法です。

トラフィックジェネレータが送信するフローは、10個の大きなパケットで構成されるバーストであり、その後に82バイトの4個の小さなパケットが続きます。これは100 Mbpsでrouter2に送信します。バーストを送信すると、router2 ATMインターフェイスの出力キューは空になります。Router2は、出力キューで輻輳が発生することを確認するために、これらのパケットを10 KBのPVC (これは非常に遅いPVC) を介して送信します。

このプロセスを簡素化するには、次のテストを複数の段階で実行します。

第 1 段階

大きなトラフィックには、482 バイトのパケットが 10 個含まれています。PA-A3 のパーティクルは 512 バイトであるため、各パケットは、大きなものでも小さなものでも、ポート アダプタの出力キューに格納されるときには 1 つのパーティクルを使用します。ルータの送信リングの制限は 3 です (tx-ring-limit=3)。次に、シンクデバイスに表示される内容の例を示します。

```
.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
```

82バイトのパケットの前に送信された482バイトの4つのパケットを確認できます。これは通常、優先順位を取得する必要があります。これが原因です。

まず、このバーストは 482 バイトのパケット 10 個で構成されているため、ルータに最初に到達し、その後に 82 バイトのパケットが続きます。482バイトのパケットは輻輳が発生していない時間に到着するため、トラフィックがないため、1つのパケットがポートアダプタセグメンテーションリアセンブリ(SAR)に即時にキューイングされ、セルに分割されて回線上に送信されます。すなわち、送信リングはまだ空のままです。

482バイトのパケットを1つ送信するのに必要な時間が、合計バーストを送信するためにトラフィックジェネレータに必要な時間よりも長いことを計算できます。したがって、最初の482バイトのパケットがポートアダプタにキューイングされると、バーストの482バイトのパケットがルータにすでに存在していると仮定できます。よって、その他の 482 バイトのパケットは、送信リングにキューされます。482バイトの3つのパケットは、そこに存在する3つの空きパーティクルを使用してキューに入れられます。

注：パケットは、複数のパーティクルを格納する必要がある場合でも、空きパーティクルが存在するとすぐに送信リングにキューイングされます。

この時点で3つのパーティクルがいっぱいになったため、輻輳状態になります。したがって、IOSでのキューイングが始まります。4つの82バイトのパケットがようやくルータに到達した時点では、輻輳が発生しています。これらの4つのパケットはキューイングされて、この2つのフローに対してWFQが使用されます。これを確認するには、**show queue ATM**コマンドを使用する**ATM**キューを調べます。

```
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
```

デバッグでは、482バイトの最初の4つのパケットの後に82バイトのパケットが続いていることがわかります。これらの小さなパケットは、大きなパケットより先にルータから送信されます。このことは、輻輳が発生すると、直ちに小さなパケットに対して大きなパケットより高い優先順位が与えられることを意味しています。

これを確認するには、「**重みの計算**」セクションで指定した**重み付けとスケジュール付け時間の式**を使用してください。

第2段階

送信リングの制限を5に増やし、大きなパケットが482バイトの場合、前の出力に従って、輻輳が発生する前に482バイトの6個のパケットが表示され、その後に82バイトの4個のパケットが表示されます。


```
.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
```

ここでわかるように、これは実際に起こることです。

第3段階

パーティクルサイズは512バイトです。したがって、送信リングがパーティクルで表現され、パーティクルサイズよりわずかに大きいパケットを使用する場合、各パケットは2つのパーティクルを取ります。これは、582バイトのパケットと3つの送信リングを使用して示されています。これらのパラメータを使用すると、582バイトの3つのパケットが表示されます。1つはATMインターフェイスでトラフィックなしで送信され、3つのパーティクルは解放されます。したがって、その他の2つのパケットがキューイングされて、その後82バイトのパケット4つが続き、さらにその後582バイトのパケット7つが続きます。

```
.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
```

第4段階

パケットサイズを1482 (3つのパーティクル) とし、5つの送信リングを定義します。送信リングがパーティクルで定義されている場合は、次のように表示されます。

- 1つのパケットが即座に送信される
- 5つのパーティクルのうち3つを取る1つのパケット
- 2つのパーティクルが空いているため、1つのパケットがキューイングされます

```
.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
```

要約

実施したテストから、次の結論を得ることができます。

- WFQを使用しない低速PVCでは、大量のトラフィックが小さなトラフィックに影響を与えます。たとえば、WFQが有効になるまでpingが停止するなど影響は小さいものです。
- 送信リング(tx-ring-limit)のサイズによって、キューイングメカニズムがジョブを開始する速度が決まります。送信リングの制限が増加すると、ping RTTが増加し、この影響を確認できます。したがって、WFQまたはLLQを実装する必要がある場合は、送信リングの制限を小さ

くする方が合理的です。

- CBWFQを使用するWFQは、実際には小さなトラフィックをバルクトラフィックよりも優先します。

関連情報

- [ATM テクノロジーに関するサポート ページ](#)
- [輻輳管理の概要](#)
- [テクニカル サポートとドキュメント - Cisco Systems](#)