What Bytes Are Counted by IP to ATM CoS Queueing?

Document ID: 10420

Contents

Introduction

This document provides information to help you determine what bytes are counted by IP to Asynchronous Transfer Mode (ATM) queueing.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

This document is not restricted to specific software and hardware versions.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Determine the Value for the Bandwidth Statement in a QoS Service Policy

Q. I need to determine the value for the bandwidth statement in my QoS service policy. On ATM permanent virtual circuits (PVCs), how is this value measured? Does it count the entire 53 bytes of the ATM cells?

A. The **bandwidth** and **priority** commands configured in a service policy to enable class−based weighted fair queueing (CBWFQ) and low latency queueing (LLQ), respectively, use a kbps value that counts the same overhead bytes counted by the **show interface** command output. Specifically, the Layer 3 queueing system counts these:

Overhead Field	Length	Counted in show policy−map interface
		Yes

Logical Link Control / Subnetwork Access Protocol (LLC/SNAP)	8 (per packet)	
ATM Adaptation Layer 5 (AAL5) Trailer	4	No. AAL5 trailer and cyclic redundancy check (CRC) is added in the segmentation and reassembly (SAR), and hence never accounted in IOS. The 4 bytes that are counted are internal virtual circuit (VC) encapsulation bytes.
Padding to make last cell an even multiple of 48 bytes	Variable	No
ATM cell headers	5 (per cell)	No

This section shows you how to use the counters in the **show policy−map interface** command output in order to determine what overhead bytes are counted by the Layer 3 queueing system.

Traditionally, Cisco devices use these definitions of AAL5PDU bytes and ATM cell bytes:

- **ATM_cell_byte = roundup(aal5_pdu/48)*53**
- **aal5_pdu_byte = ip_size + snap(8)+aal5_ovh(8) = ether_size − 2**

In this test, 50 packets per second (pps) of 60−byte IP payload to PVC 0/3 are transmitted, which is configured for AAL5SNAP encapsulation:

```
r1#show policy-map interface
   ATM5/0.33: VC 0/33 −
    Service-policy output: llq (1265)

      Class-map: p5 (match−all) (1267/4)
        14349 packets, 1033128 bytes
        30 second offered rate 28000 bps, drop rate 0 bps
        Match: ip precedence 5  (1271)
        Weighted Fair Queueing
          Strict Priority
          Output Queue: Conversation 136
          Bandwidth 40 (kbps) Burst 1000 (Bytes)
          (pkts matched/bytes matched) 0/0
          (total drops/bytes drops) 0/0
```

1033128 bytes / 14349 packets = 72 bytes per packet

8 (SNAP header) + 60 IP paylod + 4 (first 4 bytes of AAL5 trailer) = 72

After the test, the **show policy−map int** command displays 14349 packets and 1033128 bytes. These values count the number of packets that match the criteria of the class. The `pkts matched/bytes matched` value increments only when the VC is congested or when the packet is process−switched. All process−switched packets are sent to the Layer 3 queueing engine.

Confirm that the **show interface atm** command counts the same overhead bytes. In this test, five pings of 100 bytes are sent:

```
7500-1#ping 192.168.66.70
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.66.70, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
7500-1#
```

The output of the **show interface atm** command displays five packets of input and 540 bytes. The extra 40 bytes above the 500 bytes of IP payload comes from this:

- 40 bytes / 5 packets = 8 bytes overhead per packet
- 8 bytes of LLC/SNAP header

```
7500-b#show interface atm 4/1/0
ATM4/1/0 is up, line protocol is up
  Hardware is cyBus ATM
  Internet address is 192.168.66.70/30
  MTU 4470 bytes, sub MTU 4470, BW 155520 Kbit, DLY 80 usec,
  rely 255/255, load 1/255
  NSAP address: BC.CDEF01234567890ABCDEF012.345678901334.13
  Encapsulation ATM, loopback not set, keepalive not supported
  Encapsulation(s): AAL5, PVC mode
  2048 maximum active VCs, 1024 VCs per VP, 1 current VCCs
  VC idle disconnect time: 300 seconds
  Last input 00:00:03, output 00:00:03, output hang never
  Last clearing of "show interface" counters 00:00:21
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 1 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     5 packets input, 560 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     5 packets output, 560 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

This is a test done over an Ethernet interface, which sends 100 packets of 74 bytes:

```
louve(TGN:OFF,Et3/0:2/2)#show pack
Ethernet Packet:  74 bytes
     Dest Addr: 0050.73d1.6938,   Source Addr: 0010.2feb.b854
     Protocol: 0x0800

IP   Version: 0x4,  HdrLen: 0x5,  TOS: 0x00
     Length: 60,   ID: 0x0000,  Flags-Offset: 0x0000
     TTL: 60,   Protocol: 1 (ICMP),   Checksum: 0x74B8 (OK)
     Source: 0.0.0.0,     Dest: 5.5.5.5

ICMP Type: 0,   Code: 0  (Echo Reply)
     Checksum: 0x0EFF (OK)
     Identifier: 0000,  Sequence: 0000
Echo Data:
   0 : 0001 0203 0405 0607 0809 0A0B 0C0D 0E0F 1011 1213  ....................
  20 : 1415 1617 1819 1A1B 1C1D 1E1F                       ............
```

Both the **show policy−map interface** command and the **show interface ethernet** command counted 740 bytes.

```
few#show policy-map interface ethernet 2/2
```

```
      Ethernet2/2
        Service-policy output: a-test

          Class-map: icmp (match-all)
            10 packets, 740 bytes

    few#show interface ethernet 2/2
          10 packets output, 740 bytes, 0 underruns(0/0/0)
```

60 IP payload + 2 * 6 (source and destination MAC address) + 2 (protocol type) = 74

From this calculation, you can see that the Ethernet CRC is not included in either the **show interface** or **show policy−map** command outputs. Importantly, both values are consistent in whether or not the CRC is included.

Finally, here are the bytes counted on a serial interface that uses high−level data link control (HDLC) encapsulation. In this test, five packets of 100 bytes are transmitted:

```
    r3#show policy interface
      Serial4/2:0
        Service-policy output: test

          Class-map: icmp (match-all)
            5 packets, 520 bytes
```

Here are the definitions of Cisco HDLC frames:

1	1	1	2	Variable	2	1
Flag 0x7E	Address	Ctrl 0x00	Protocol	Data	FCS	Flag 0x7E

- flag start or end of frame = 0x7E
- address frame type field:

  ♦ 0x0F Unicast Frame
  ♦ 0x80 Broadcast Frame
  ♦ 0x40 Padded Frame
  ♦ 0x20 Compressed Frame
- protocol Ethernet type of the encapsulated data, such as 0x0800 for IP

The output of the **show policy interface** command for the serial test displays 520 bytes. The additional four bytes per frame do not include the beginning and ending frame flags. Instead, the bytes include the address, control and protocol fields. Importantly, the bytes do not include the frame check sequence (FCS).

Conclusion

It is important to understand that there is a difference in the number of octets counted by the Layer 3 queueing system and the number of octets that actually are used by a packet once it reaches the physical layer. The real bandwidth used by 64−bytes packet is much greater on an ATM interface than on an Ethernet interface. Specifically, CBWFQ and LLQ do not account for these two sets of ATM−specific overhead:

- Padding Makes the last cell of a packet an even multiple of 48 bytes. This padding is added by the SAR once the packet reaches the ATM layer.
- 5−byte ATM cell header

In other words, CBWFQ and LLQ estimate 64 bytes at 64 bytes, but the packet actually occupies 106 bytes and uses two cells at the ATM and physical layers. On all interfaces, flags and a CRC are also present, but are not included by the Layer 3 queueing system.

Cisco bug ID CSCdt85156 (registered customers only) is a feature request to count the CRC. It argues that all fixed and predictable Layer 2 overhead, such as a CRC, should be included in the priority statement to make this configuration as accurate and close as possible to what actually is consumed by a flow when it hits the physical wire.

Related Information

- **Voice Over IP – Per Call Bandwidth Consumption**
- **Low Latency Queueing**
- **Asynchronous Transfer Mode (ATM) Resources**
- **LAN Product Support**
- **Technical Support & Documentation – Cisco Systems**

Updated: Nov 03, 2006   Document ID: 10420