Understand Queue Buffer Allocation on Catalyst 9000 Switches

Contents

Introduction

Prerequisites

- Requirements
- Components Used
- Related Products

Background Information

- Terminology
- Buffers
- Switch Queue Buffer Allocation Algorithm

Configure

- Scenario 1: Determine Base Buffer Value for a Given Port Speed
- Scenario 2: Determine Soft and Hard buffer for Multiple Queues and Explicit Ratios
  - Scenario 2.1: Modify User Configured Multiplier from Scenario 2
- Scenario 3: Determine Soft and Hard Buffer for Multiple Queues and an Implicit Ratio
  - Scenario 3.1: Determine Soft and Hard Buffer for Multiple Queues and Multiple Implicit Ratios
- Scenario 4: Sum of Queue-buffers Ratio Across Queues is Less Than 100
- Scenario 5: Determine Soft and Hard Buffer of a Priority Level 2 Class
- Queue Limit

Related Information

Introduction

This document describes how to predict queue buffer allocation to traffic queues on Catalyst 9000 Series Switches.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Quality of Service (QoS) concepts and processes for how packets are Marked, Queued, and Scheduled
- Cisco MQC QoS Configuration

Components Used

The information in this document is based on these software and hardware versions:

- Cisco Catalyst 9500-25Y4C
- Cisco IOS® XE 17.4.1

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Related Products

This document can also be used with these hardware and software versions:

- Catalyst 9200 - 9600 Series Switches
- Catalyst 9300X & 9400X
- All Cisco IOS XE Software Versions

Note: This document does not apply to the 9500X or 9600X as these use a different ASIC and QoS architecture.

Background Information

For a technical overview of QoS on Catalyst 9000 Series Switches, see: Catalyst 9000 QoS and Queueing White Paper.

Often buffer allocation needs to be tuned as a response to undesired output drops for a particular class of traffic. More information on how to diagnose and troubleshoot output drops on Catalyst 9000 Series Switches can be found in this article: Troubleshoot Output Drops on Catalyst 9000 Switches

Terminology

QoS	Quality of Service	A concept / group of related features related to classify, mark, queue, and schedule traffic in and out of a network device

DSCP	Differentiated Services Code Point	A traffic classification mechanism contained in the IP Header of a packet
CoS	Class of Service	A traffic classification mechanism contained in the Ethernet frame header of a packet
ACE	Access Control Entry	A single rule or line within an Access Control List (ACL)
ACL	Access Control List	A group of Access Control Entries (ACEs) used by various features to match traffic and take an action
ASIC	Application Specific Integrated Circuit	A computer chip designed to perform a specific task or set of tasks with high efficiency.
UADP	Unified Access Data Plane	Cisco ASIC used in Catalyst 9000 Series Switches to perform many network packet processing tasks.
PBC	Packet Buffer Complex	Cisco UADP ASIC subsystem which serves as central packet buffer to process, queue, and schedule packets.
AQM	Active Queue Management	Cisco UADP ASIC subsystem which manages traffic queue and schedule actions for network ports.
DTS	Dynamic Threshold and Scale	Cisco UADP ASIC technology which dynamically adjusts and scales buffers across ports to optimize hardware utilization

Buffers

As a concept, buffers are memory used to absorb transient bursts of data, when the data switched or routed to a port exceeds that ports ability to put data on the wire. A port has a fixed rate at which it transmits and removes data from the queue. A buffer, conceptually, is simply a place to store, or queue, data until it is transmitted out of the interface.

On Catalyst 9000 Series Switches, the word buffer has two uses. The system buffer as a whole is otherwise known as the packet buffer complex (PBC) of the ASIC. The word buffers can also refer small unit of the PBC. A buffer is allocated to ports on a per-queue basis. In other words, a port queue is allocated a quantity

of small individual buffers from the overall system buffer.

On Cisco UADP ASIC based platforms, one buffer contains up to 256 Bytes of data, and buffers are linked together to represent frames larger than 256 Bytes.

Switch Queue Buffer Allocation Algorithm

- A port has a hard-coded base buffer value allocated it which can be scaled by the user or system (but the base number can not be changed, which is important to understand how to predict outcomes)
- The base buffer value for a given port type depends on specific combination of hardware (switch model), software version, and port speed. For example, a Gigabit Ethernet port on Catalyst C9500-25Y4C starts with a different base buffer value than a Gigabit Ethernet port on Catalyst 9200L
- This base value is split across *N* number of queues - based on the configuration.

Final calculation of available buffer per-queue is influenced by these factors:

- Number of queues configured
- The **queue-buffers ratio**, a command which allows you to adjust how base buffer is divided among configured port queues.
- A user configured base buffer multiplier (configured as **qos queue-softmax-multiplier <percent>**)
- A system defined base buffer multiplier (not user configurable, 4x or 400%) - while this is applied by default, there are some circumstances where its removed so its important to understand this 4x value exists to predict the allocation algorithm
- Both multipliers only affect what are called soft buffers

Soft buffers are buffers shared across ports. These buffers are called soft because they are not guaranteed to the port.

The system over allocates soft buffers intentionally. This allows any one port to use a large number of buffers if needed, but as more ports need buffers, all ports and queues are scaled down dynamically and fairly as part of the Cisco UADP ASIC DTS process.

In summary, soft buffers - referred to in outputs as softmax, is an opportunistic maximum value. A port only uses the full softmax if that amount of buffer is available from the overall system buffer. As buffer demand rises across other ports and queues, the maximum buffer available to the port is less.

Hard buffers are buffers explicitly reserved for a port, and are not affected by the DTS process. Because hard buffers are guaranteed buffers, sum of total hard buffers allocated to ports never exceeds the PBC segment dedicated to these hard buffers.

The mechanisms which governs the active scale of soft buffer is known as DTS (Dynamic Threshold and Scale), described in the in the Catalyst 9000 series QoS White Paper.

The size of the PBC segments dedicated to queue hard and soft buffers changes dynamically as you configure the system, and can be seen as AQM GlobalSoftLimit and GlobalHardLimit in this output:

<#root>

C9500#

show platform hardware fed active qos queue stats interface twe1/0/1

```
-----------------------------------------------------------------------------------------------
AQM Global counters
GlobalHardLimit:
```

18072

```
 | GlobalHardBufCount: 0
GlobalSoftLimit:
```

37224

```
 | GlobalSoftBufCount: 0
```

C9500#

show platform hardware fed active qos queue config interface tw1/0/1

```
Asic:0 Core:1 DATA Port:20 GPN:101 LinkSpeed:0x12
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 160 - 167
  DrainFast:Disabled PortSoftStart:2 - 4320 BufferSharing:Disabled
   DTS
```

Hardmax

Softmax

```
   PortSMin  GlblSMin  PortStEnd   QEnable
  ----- --------  --------  --------  --------  ---------  -------
 0   1  2
```

480

```
   3
```

1920

```
  16   960   0     0   4  5760      En
```

<--- default configuration has a mix of hard buffer and soft buffer in queue 0

```
 1   1  0
0
   4
2880
  16  1440   8   720   4  5760      En
<--- default configuration has two queues so some buffers are seen in queue 1


<snip>


C9500(config)#
policy-map test

C9500(config-pmap)#
class class-default

C9500(config-pmap-c)#
priority level 1


<--- Priority level 1 queue configuration on first queue, which is queue 0 in the next output

C9500(config-pmap-c)#
exit

C9500(config-pmap)#
exit

C9500(config)#
int tw1/0/1
```

```
C9500(config-if)#
service-policy output test

C9500(config-if)#
end

C9500#
show platform hardware fed active qos queue config interface twe1/0/1

Asic:0 Core:1 DATA Port:20 GPN:101 LinkSpeed:0x12
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 160 - 167
  DrainFast:Disabled PortSoftStart:4 - 1800 BufferSharing:Disabled
   DTS
Hardmax

Softmax
  PortSMin GlblSMin PortStEnd QEnable
  ----- -------- -------- -------- -------- --------- -------
0    1  4
1200
  7
1200
  0      0 0      0 3   2400    En
<--- Hardmax increased to 1200 from 480 in queue 0, softmax reduced to 1200 from 1920

1    1  0     0  0     0  0      0 0      0 3   2400    En
<--- queue 1 now no longer has any values, as no second queue is configured

C9500#
show platform hardware fed active qos queue stats interface twe1/0/1
```

```
-------------------------------------------------------------------------------------------
AQM Global counters
GlobalHardLimit:

18792

 | GlobalHardBufCount: 0

<--- GlobalHardLimit increased to 18792 from 18072, or by 720


GlobalSoftLimit:

36504

 | GlobalSoftBufCount: 0

<---



GlobalSoftLimit decreased from 37224 to 36504, or by 720
```



Note: Notice the increase in GlobalHardLimit and proportional decrease in GlobalSoftLimit.

Further, when you configure **priority level 1**, the softmax for that queue is statically set to be exactly equal to the hardmax. You can only modify the hard buffer for a **priority level 1** queue.

The change in GlobalHardLimit and GlobalSoftLimit is equal to 720. This is also equal to the change in hardmax after configuration.

The scenarios in this document explain how to compute and predict softmax and hardmax allocations across multiple policy-map configurations.

Configure

Scenario 1: Determine Base Buffer Value for a Given Port Speed

A queues final buffer value is in part a function of a base value that is first allocated across queues. This is

then later multiplied in the case of soft buffers.

The multiplication factors, in combination with other implicit behaviors, makes determination of a final value for a given queue, with a given configuration, a challenge.

The first step to clarify resultant queue buffer allocation is to determine the base buffer value.

To do this, leverage a priority queue, which receives hard buffer directly proportional either to number of queues or configured queue-buffers ratio.

With a specific configuration, you can explicitly derive the amount of base buffer allocated to a given port speed.

Configure and assign all buffer to a single, non multiplied queue (a **priority level 1** queue)

In this example, the class-default class is used to match all traffic, because no other classes are configured.

<#root>

```
Switch(config)#
```

policy-map test1

```
Switch(config-pmap)#
```

class class-default

```
Switch(config-pmap-c)#
```

priority level 1

<--- Assign hard buffer to the port, which is not affected by multipliers

```
Switch(config-pmap-c)#
```

queue-buffers ratio 100

<--- Assign all buffers to this queue only

The configuration in the previous example takes these actions::

- The **priority level 1** tells the system to perform a strict schedule for this queue / class, and allocate hard buffer to this queue / class.
    - How packets are scheduled is beyond the scope of this document, but a strict queue generally is serviced before all others.
- The **queue-buffers ratio 100** allocates 100/100 or 100% of available base buffer to this queue / class.

In a **policy-map** with more than one class, you can not allocate 100% of the buffer to a single class. You are required to allocate 1/100 or 1% at minimum to any class.

In a policy with only one class, you have only one class and you can allocate all of the buffer to it.

As noted previously, a priority queue gets hard buffers equal to its distribution of base buffer as per the configured **queue-buffers ratio**. A hard buffer is not subject to any multiplier.

Hard buffer is observed in outputs under a column titled **Hardmax**.

Now you have a single traffic class with buffers not subject to any multiplier. With this, you can explicitly derive base buffer allocation for this port speed (and only this port speed on this platform, others differ), because base buffer and hardmax are equal.

Base Buffer = ?

Queue Ratio 1 = 100/100 = 1

Hardmax for this queue = Base Buffer x Queue Ratio 1

X = Y x 1

X / 1 = Y

X = Y

X = Y = Hardmax = base buffer = 1200 (see Example 2).

In this example, policy-map test1 is applied to an interface as an output service-policy

<#root>

```
9500H(config)#
```

```
int tw1/0/3


9500H(config-if)#

service-policy output test1                    <--- service policy that assigns all buffer to the first



9500H#

show platform hardware fed active qos queue config interface tw1/0/3

Asic:0 Core:1 DATA Port:22 GPN:103 LinkSpeed:0x12
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 176 - 183
  DrainFast:Disabled PortSoftStart:4 - 1800 BufferSharing:Disabled
   DTS

Hardmax

  Softmax   PortSMin  GlblSMin  PortStEnd   QEnable
  ----- --------  --------  --------  --------  ---------  -------
 0   1  4

1200

   7  1200   0     0   0     0   3  2400      En

<--- hardmax 1200 - the maximum amount of buffer this port can use without multiplication

 1   1  0     0   0     0   0     0   0     0   3  2400      En
 2   1  0     0   0     0   0     0   0     0   3  2400      En
 3   1  0     0   0     0   0     0   0     0   3  2400      En
 4   1  0     0   0     0   0     0   0     0   3  2400      En
 5   1  0     0   0     0   0     0   0     0   3  2400      En
 6   1  0     0   0     0   0     0   0     0   3  2400      En
 7   1  0     0   0     0   0     0   0     0   3  2400      En
<snip>
```

As shown, the hardmax for this priority queue with 100% of buffer allocated to it, is 1200.

As hardmax is a fully unmultiplied / unscaled value, and 100% of buffer is configured to this queue, the base buffer allocation for this specific model of switch, software version, and specific port speed is 1200.

Other port speeds on this same switch, and other models of switches for the same port speed, receive different base buffer allocations. This base allocation is not user configurable and must be derived via observation.

Further scenarios in this document start all use the same switch, software, and port speed. So, they all assume a base allocation of 1200 for calculations to determine final buffer allocation.

Note: Notice that the softmax in the previous example is also 1200.

By design, a priority level 1 queue has softmax equal exactly to its hardmax. This is intended and not user configurable.

Further, this specific case of softmax allocation is not affected by softmax multipliers shown later. Only a **priority level 1** queue has this behavior for softmax, which is intended.

Scenario 2: Determine Soft and Hard buffer for Multiple Queues and Explicit Ratios

In this scenario, an additional queue is added. This queue does not use **priority level 1**, and so softmax scales with multipliers.

One multiplier is user configured and the other a hidden / notconfigurable multiplier.

Combine these multipliers with the base buffer derived for this port, in this case 1200 as per the Scenario 1.

Algorithmically:

Current queue ratio = **queue-buffers ratio** for the queue / class to be predicted
Hidden Multiplier = 400%

User Multiplier = Percent value you configure in **qos queue-softmax-multiplier <percent>**. Default is 100%

Softmax = (Base Buffer x (Current Queue Ratio / 100)) x Hidden Multiplier x (User Multiplier / 100)

<#root>

```
9500H(config)#
```

policy-map test2

```
9500H(config-pmap)#
```

class class1

```
9500H(config-pmap-c)#
```

```
priority level 1


9500H(config-pmap-c)#

queue-buffers ratio 50



<-- class 1 / first queue gets 50% of base buffer


9500H(config-pmap-c)#

class class-default


9500H(config-pmap-c)#

bandwidth remaining percent 100 <-- required configuration due to priority queue, can be ignored for thi


9500H(config-pmap-c)#

queue-buffers ratio 50



<-- class 2 / first queue gets 50% of base buffer
```

Summary of values:

- Base Buffer = 1200 - How to find this number is shown in Scenario 1
- User Multiplier = 100% - Default value, this has not been modified yet
- Hidden Multiplier = 4 - default and not user configurable
- Current Queue Ratio(class1) = 50% - Configured in the policy map with **queue-buffers ratio 50**
- Current Queue Ratio(class-default) = 50% - Configured in the policy map with **queue-buffers ratio 50**

Determine Class1 buffer allocation:

As class1 is a priority queue, it receives hardmax (hard buffer), and a special case of softmax not affected by multipliers.

Class1 hardmax = (Base Buffer x Current Queue Ratio(class1) / 100)
Class1 hardmax = 1200 x (50/100) = 600 - due to special case of a priority queue, stop all math, assign your

result to hardmax. Softmax equals Hardmax as a rule for **priority level 1**.

Determine class-default buffer allocation:

Class class-default = (Base Buffer x (Current Queue Ratio(class-default / 100)) x Hidden Multiplier x (User Multiplier / 100)

Class class-default = [

[base buffer] 1200 x [Current Queue Ratio] (50/100) = 600
[previous result] 600 x [Hidden multiplier] 4 x [User Multipiler] (100/100) = 2400

]

<#root>

```
9500H(config)#
```

int tw1/0/3

```
9500H(config-if)#
```

service-policy output test2

<-- apply the policy

```
9500H#
```

show platform hardware fed active qos queue config interface tw1/0/3

```
Asic:0 Core:1 DATA Port:22 GPN:103 LinkSpeed:0x12
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 176 - 183
  DrainFast:Disabled PortSoftStart:5 - 3600 BufferSharing:Disabled
   DTS  Hardmax  Softmax   PortSMin  GlblSMin  PortStEnd   QEnable
  ----- --------  --------  --------  --------  ---------  -------
 0   1  5
```

600

```
   9
```

600

```
   0     0   0     0   1  4800      En

 <-- Hardmax is 600 as predicted, Softmax is set equal to Hardmax due to priority level 1

 1   1  0     0  10

2400

  16  1200   8   600   1  4800      En

<--



Softmax is 2400 as predicted


<snip>
```

Final result: Q0 - Hardmax: 600 Softmax: 600. Q1 - Softmax: 2400

Scenario 2.1: Modify User Configured Multiplier from Scenario 2

This scenario starts the same as Scenario 2, except now you configure **qos queue-softmax-multiplier 1200**.

This multiplies the softmax buffers in the current configuration by 1200%, or a factor of 12.

Summary of values:

- Base Buffer = 1200 - How to find this number is shown in Scenario 1
- User Multiplier = 1200% - This is the value changed in this scenario, via **qos queue-softmax-multiplier 1200**
- Hidden Multiplier = 4 - default and not user configurable
- Current Queue Ratio(class1) = 50% - Configured in the policy map with **queue-buffers ratio 50**
- Current Queue Ratio(class-default) = 50% - Configured in the policy map with **queue-buffers ratio 50**

Determine Class1 buffer allocation:

As class1 is a priority queue, it receives hardmax (hard buffer), and a special case of softmax not affected by multipliers.

Class1 hardmax = (Base Buffer x Current Queue Ratio(class1) / 100)
Class1 hardmax = 1200 x (50/100) = 600 - due to special case of a priority queue, stop all math, assign your result to hardmax. Softmax equals Hardmax as a rule for **priority level 1**.

Determine class-default buffer allocation:

Class class-default = (Base Buffer x (Current Queue Ratio(class-default / 100)) x Hidden Multiplier x (User Multiplier / 100)

Class class-default =[

[base buffer] 1200 x [Current Queue Ratio] (50/100) = 600
[previous result] 600 x [Hidden multiplier] 4 x [User Multipiler] (1200/100) = 28800

]

Configure qos **queue-softmax-multiplier 1200** and observe changes to softmax (softmax is a maximum buffer value for that queue, scaled dynamically based on current overall buffer usage):

```
<#root>

9500H(config)#

qos queue-softmax-multiplier 1200



9500H#

show platform hardware fed active qos queue config interface tw1/0/3



Asic:0 Core:1 DATA Port:22 GPN:103 LinkSpeed:0x12
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 176 - 183
  DrainFast:Disabled PortSoftStart:3 - 31500 BufferSharing:Disabled
   DTS  Hardmax  Softmax   PortSMin  GlblSMin  PortStEnd   QEnable
  ----- --------  --------  --------  --------  ---------  -------
 0   1  5

600

   5

600

   0     0   0     0   6 42000      En
```

```
<-- Queue 0 does not change as its configured with priority level 1

 1   1  0     0   6

28800

   1   900   1   900   6 42000      En

<-- Softmax increases by 12x to 28800 from 1200 due to queue-softmax-multiplier 1200


<snip>
```

Final result: Q1 - Hardmax: 600, Softmax: 600. Q2 - Softmax: 28800

Scenario 3: Determine Soft and Hard Buffer for Multiple Queues and an Implicit Ratio

In this scenario, five queues are configured, but only four have **queue-buffers ratio** explicitly defined. The buffer allocated to those queues is the same as the previous examples.

The queue that is not configured receives the difference between the sum of all configured queue buffers, and 100.

Sum of Explicitly Configured Ratios = (Q0 buffer ratio) + (Q1 buffer ratio) ... (final buffer ratio) - up to 8 queues are supported on Catalyst 9000 Series Switches, so you can add up to 8 ratios

Implicit Ratio Leftover = (100 - Sum of Explicitly Configured Ratios).

Implicit Ratio Leftover is the value that gets assigned to a queue that does not have **queue-buffers ratio** configured.

Policy-map in use for this scenario:

<#root>

```
9500H(config)#

policy-map test3


9500H(config-pmap)#

class class1
```

```
9500H(config-pmap-c)#
priority level 1

9500H(config-pmap-c)#
queue-buffers ratio 20

9500H(config-pmap-c)#
class class2

9500H(config-pmap-c)#
bandwidth remaining percent 10 <-- no queue-buffers ratio statement for this class

9500H(config-pmap-c)#
class class3

9500H(config-pmap-c)#
bandwidth remaining percent 10

9500H(config-pmap-c)#
queue-buffers ratio 10


<-- rest of queues have an explicit queue-buffers ratio

9500H(config-pmap-c)#
class class4

9500H(config-pmap-c)#
bandwidth remaining percent 10

9500H(config-pmap-c)#
```

```
queue-buffers ratio 10


9500H(config-pmap-c)#

class class-default


9500H(config-pmap-c)#

bandwidth remaining percent 70


9500H(config-pmap-c)#

queue-buffers ratio 40
```

Summary of values:

- Base Buffer = 1200 - How to find this number is shown in Scenario 1
- User Multiplier = 100% - Default value, this has not been modified yet
- Hidden Multiplier = 4 - default and not user configurable
- Current Queue Ratio(class1) = 20% - Configured in the policy map with **queue-buffers ratio 20**
- Current Queue Ratio(class2) = Unknown, you must solve for this - **queue-buffers ratio** is not configured for this class
- Current Queue Ratio(class3) = 10% - Configured in the policy map with **queue-buffers ratio 10**
- Current Queue Ratio(class4) = 10% - Configured in the policy map with **queue-buffers ratio 10**
- Current Queue Ratio(class-default) = 40% - Configured in the policy map with **queue-buffers ratio 40**

Calculate queue buffer ratio that remains:

Implicit Ratio Leftover = (100 - Sum of Explicitly Configured Ratios).

100 - (20) - (10) - (10) - (50) = 20

Current Queue Ratio(class2) = 20

Calculate final queue buffer allocation
Class1 = [Base Buffer] 1200 x [Current Queue Ratio(class1)] (20/100) = 240 - priority queue, no further calculation

Class2 = [

[Base Buffer] 1200 x [Current Queue Ratio(class2)] (10/100) = 240 - base buffer allocation for this queue,

but it must be multiplied to get softmax for a non-priority queue
[Base Buffer allocation for this queue] 120 x [Hidden multiplier] 4 x [User Multiplier] (100/100) = 960

]

Repeat for queues that remain:

Class 3 = [

1200 x (10/100) = 120
120 x 4 x (100/100) = 480
]

Class 4 = [

1200 x (10/100) = 120
120 x 4 x (100/100) = 480
]

Class class-default = [

1200 x (40/100) = 480
600 x 4 x (100/100) = 1920
]

Result of the test3 policy-map applied compared to the prediction:

<#root>

```
9500H(config)#
```

int tw1/0/3

```
9500H(config-if)#
```

service-policy output test3

```
9500H#
```

show platform hardware fed active qos queue config interface tw1/0/3

```
Asic:0 Core:1 DATA Port:22 GPN:103 LinkSpeed:0x12
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 176 - 183
```

```
  DrainFast:Disabled PortSoftStart:4 - 2880 BufferSharing:Disabled
   DTS  Hardmax  Softmax   PortSMin  GlblSMin  PortStEnd   QEnable
  ----- --------  --------  --------  --------  ---------  -------
 0   1  5   240   8   240   0     0   0     0   6  3840      En
 1   1  0     0   9
```

960

```
  16   480   8   240   6  3840      En
```

<-- queue without queue buffers ratio configured receives any leftover ratio, as predicted

```
 2   1  0     0  11   480  16   240   8   120   6  3840      En
 3   1  0     0  11   480  16   240   8   120   6  3840      En
 4   1  0     0   4  1920  16   960   8   480   6  3840      En
<snip>
```

Final result: Q0 - Hardmax: 240, Softmax: 240. Q1 - Softmax: 960, Q2 - Softmax: 480, Q3 - Softmax: 480, Q4 - Softmax: 480

Scenario 3.1: Determine Soft and Hard Buffer for Multiple Queues and Multiple Implicit Ratios

In this scenario, five queues are configured and two queues do not have **queue-buffers ratio** configured.

To determine buffer allocation, the same logic from Scenario 2 continues, but you must also divide the Implicit Ratio Leftover by the total number implicit queues / queues that do not have **queue-buffers ratio**

Policy-map in use for this scenario:

<#root>

```
9500H(config)#
```

policy-map test4

```
9500H(config-pmap)#
```

class class1

```
9500H(config-pmap-c)#
```

priority level 1

```
9500H(config-pmap-c)#
```

```
queue-buffers ratio 20

9500H(config-pmap-c)#
class class2

9500H(config-pmap-c)#
bandwidth remaining percent 10 <-- no queue-buffers ratio statement for this class

9500H(config-pmap-c)#
class class3

9500H(config-pmap-c)#
bandwidth remaining percent 10 <-- no queue-buffers ratio statement for this class

9500H(config-pmap-c)#
class class4

9500H(config-pmap-c)#
bandwidth remaining percent 10

9500H(config-pmap-c)#
queue-buffers ratio 10

9500H(config-pmap-c)#
class class-default

9500H(config-pmap-c)#
bandwidth remaining percent 70

9500H(config-pmap-c)#
queue-buffers ratio 40
```

Summary of values:

- Base Buffer = 1200 - How to find this number is shown in Scenario 1
- User Multiplier = 100% - Default value
- Hidden Multiplier = 4 - Default and not user configurable
- Current Queue Ratio(class1) = 20% - Configured in the policy map with **queue-buffers ratio 20**
- Current Queue Ratio(class2) = Unknown, you must solve for this - **queue-buffers ratio** is not configured for this class
- Current Queue Ratio(class3) = Unknown, you must solve for this - **queue-buffers ratio** is not configured for this class
- Current Queue Ratio(class4) = 10% - Configured in the policy map with **queue-buffers ratio 10**
- Current Queue Ratio(class-default) = 40% - Configured in the policy map with **queue-buffers ratio 40**

Calculate queue buffer ratio that remains:

Implicit Ratio Leftover = (100 - Sum of Explicitly Configured Ratios).

Number of Implicit Queues = 2 (class2 and class3 have no **queue-buffers ratio** defined)

Sum of Configured Ratios = 20+40+10 = 7

Implicit Ratio Leftover = 100 - 70 = 30

Implicit queue ratio allocation = [Implicit Ratio Leftover] 30 / [Number of Implicit Queues] 2 = 15

Calculate final queue buffer allocation:

Class1 =

[Base Buffer] 1200 x [Current Queue Ratio(class1)] (20/100) = 240 - priority queue, no further calculation

Class2 =

[Base Buffer] 1200 x [Implicit queue ratio allocation] (15/100) = 180 - Because class 2 has no defined **queue-buffers ratio**, the remainder of queue-buffers ratio from explicit queues is shared among implicit queues.

[Base Buffer allocation for this queue] 180 x [Hidden multiplier] 4 x [User Multiplier] (100/100) = 720

Repeat for queues that remain:

Class 3 = [

1200 x (15/100) = 180
120 x 4 x (100/100) = 720
]

Class 4 = [

1200 x (10/100) = 120
120 x 4 x (100/100) = 480
]

Class class-default = [

1200 x (40/100) = 480
600 x 4 x (100/100) = 1920
]

Result of the test4 policy-map applied compared to the prediction:

<#root>

```
9500H(config)#
```

interface tw1/0/3

```
9500H(config-if)#
```

service-policy output test4

```
9500H#
```

show platform hardware fed active qos queue config interface tw1/0/3

```
Asic:0 Core:1 DATA Port:22 GPN:103 LinkSpeed:0x12
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 176 - 183
  DrainFast:Disabled PortSoftStart:4 - 2880 BufferSharing:Disabled
   DTS  Hardmax  Softmax   PortSMin  GlblSMin  PortStEnd   QEnable
  ----- --------  --------  --------  --------  ---------  -------
 0   1  5   240   8   240   0     0   0     0   6  3840      En
 1   1  0     0   9
```

720

```
  16   360   8   180   6  3840      En

<-- queue 1 and 2


were not configured with queue-buffers ratio

 2   1  0     0   9

720

  16   360   8   180   6  3840      En

<-- queue 1 and 2 get an equal share of leftover buffer ratio

 3   1  0     0  11   480  16   240   8   120   6  3840      En
 4   1  0     0   4  1920  16   960   8   480   6  3840      En
<snip>
```



Note: if the result of Implicit queue ratio allocation is not an integer, an equal share is not possible. The result rounds up for queues earlier in the policy-map, and round down for later queues. The final sum of queue buffers ratio allocated remains 100, but implicit queues do not always get equal allocation due to the integer result requirement just described.

Scenario 4: Sum of Queue-buffers Ratio Across Queues is Less Than 100

In this scenario, five queues are configured, all with queue-buffers ratio. The sum total of queue-buffers ratio across the classes is less than 100.

In this case, the unallocated buffer ratio is distributed across the classes evenly.

Similar to the previous scenario, if the divided result of the leftover queue-buffers ratio is not an integer, the final allocation to each queue is rounded up or rounded down and added to the queue buffers ratio you configured.

Policy-map in use for this scenario:

```
<#root>

9500H(config)#

policy-map test5
```

```
9500H(config-pmap)#
class class1

9500H(config-pmap-c)#
priority level 1

9500H(config-pmap-c)#
queue-buffers ratio 10

9500H(config-pmap-c)#
class class2

9500H(config-pmap-c)#
bandwidth remaining percent 10

9500H(config-pmap-c)#
queue-buffers ratio 10

9500H(config-pmap-c)#
class class3

9500H(config-pmap-c)#
bandwidth remaining percent 10

9500H(config-pmap-c)#
queue-buffers ratio 10

9500H(config-pmap-c)#
class class4

9500H(config-pmap-c)#
```

```
bandwidth remaining percent 10


9500H(config-pmap-c)#

queue-buffers ratio 10


9500H(config-pmap-c)#

class class-default


9500H(config-pmap-c)#

bandwidth remaining percent 70


9500H(config-pmap-c)#

queue-buffers ratio 12
```

Summary of values:

- Base Buffer = 1200 - How to find this number is shown in Scenario 1
- User Multiplier = 100% - Default value
- Hidden Multiplier = 4 - Default and not user configurable
- Current Queue Ratio(class1) = 10% - Configured in the policy map with **queue-buffers ratio 10**
- Current Queue Ratio(class2) = 10% - Configured in the policy map with **queue-buffers ratio 10**
- Current Queue Ratio(class3) = 10% - Configured in the policy map with **queue-buffers ratio 10**
- Current Queue Ratio(class4) = 10% - Configured in the policy map with **queue-buffers ratio 10**
- Current Queue Ratio(class-default) = 12% - Configured in the policy map with **queue-buffers ratio 12**
- Total number of queues = 5

Sum of Configured Ratios = 10 + 10 + 10 + 10 + 12 = 52

Buffer Ratio Leftover = 100% - 52% = 48%

[Buffer Ratio Leftover] 48% / [Total Number of Queues] 5 = 9.6% added per queue - This is not an integer, so its final application to queues must be rounded up or down on a per-queue basis

To get the final queue-buffers ratio number the system uses, you must add 9 or 10 to the already configured queue buffers ratio.

Classes higher in the policy-map receive the rounded up value, 10. Classes lower in the policy-map

receive the rounded down value, 9.

Calculate final queue buffer allocation

Buffer Ratio Leftover = 48

Class1 = [Base Buffer] x ([Current Queue ratio(class1) + Rounded up value of shared buffer ratio leftover)]

Class1 = 1200 x ((10% + 10%)/100) = 240 - priority queue, no further calculation

Buffer Ratio leftover = (48 - 10) = 38

Class2 = [Base Buffer] x ([Current Queue ratio(class2) + Rounded up value of shared buffer ratio leftover)]

Class2 = 1200 x ((10% + 10%)/100) = 240 - Continue to multiply this by user and system multipliers as this is not a priority queue

Class2 = [Base Buffer allocation for this queue] 240 x [Hidden multiplier] 4 x [User Multiplier] (100/100) = 960 - softmax result for this queue

Buffer Ratio leftover = (38 - 10) - 28

Repeat for queues that remain:

Class 3 = [

1200 x ((10+10)/100) = 240
120 x 4 x (100/100) = 960
]

Buffer Ratio leftover = (28 - 10) = 18

Class 4 = [

1200 x ((10+9)/100) = 240
120 x 4 x (100/100) = 912
]

Buffer Ratio leftover = 9

Class class-default= [

1200 x ((12+9)/100) = 252
120 x 4 x (100/100) = 1008
]

Buffer Ratio Leftover = 0

Result of the test5 policy-map applied compared to the prediction:

<#root>

```
9500H#

show platform hardware fed active qos queue config interface tw1/0/3

Asic:0 Core:1 DATA Port:22 GPN:103 LinkSpeed:0x12
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 176 - 183
  DrainFast:Disabled PortSoftStart:5 - 1512 BufferSharing:Disabled
   DTS  Hardmax  Softmax   PortSMin  GlblSMin  PortStEnd   QEnable
  ----- --------  --------  --------  --------  ---------  -------
 0   1  5   240   9

240

   0     0   0     0   6  2016      En
 1   1  0     0  10

960

  16   480   8   240   6  2016      En
 2   1  0     0  10

960

  16   480   8   240   6  2016      En
 3   1  0     0  11

912

  16   456   8   228   6  2016      En
 4   1  0     0  12

1008

  16   504   8   252   6  2016      En
<snip>
```

Scenario 5: Determine Soft and Hard Buffer of a Priority Level 2 Class

In this scenario, a class in a policy-map is configured with **priority level 2**.

Unlike **priority level 1**, in which softmax is not affected by multipliers and is set equal to hardmax, **priority level 2** allows for softmax to be multiplied while it also has a hard buffer (hardmax) allocation.

Policy-map in use for this scenario:

```
<#root>
9500H(config)#
policy-map test6

9500H(config-pmap)#
class class1

9500H(config-pmap-c)#
priority level 1

9500H(config-pmap-c)#
queue-buffers ratio 50 <-- 50 / 50 split between both queues

9500H(config-pmap-c)#
class class-default

9500H(config-pmap-c)#
priority level 2


<-- Priority level 2 in use now

9500H(config-pmap-c)#
queue-buffers ratio 50 <-- 50 / 50 split between both queues
```

Result of the test6 policy-map applied:

```
<#root>

9500H#

show platform hardware fed active qos queue config interface tw1/0/3


Asic:0 Core:1 DATA Port:22 GPN:103 LinkSpeed:0x12
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 176 - 183
  DrainFast:Disabled PortSoftStart:5 - 3600 BufferSharing:Disabled
   DTS  Hardmax  Softmax   PortSMin  GlblSMin  PortStEnd   QEnable
  ----- --------  --------  --------  --------  ---------  -------
 0   1  5   600   9

600

   0     0   0     0   1  4800      En

<-- Softmax is equal to hardmax

 1   1  5   600  10

2400

  16  1200   0     0   1  4800      En

<-- Softmax is multiplied by Hidden Multiplier (400%) and User Multiplier (100% default)r


<snip>
```

In the output shown previously, the second queue softmax has 4 as much softmax of the first queue. This is because **priority-level 1** softmax specifically is not affected by system softmax multipliers, but **priority-level 2** is.

If you configure a user softmax multiplier, only the **priority-level 2** queue is affected:

```
<#root>

9500H(config)#

qos queue-softmax-multiplier 200



<#root>

9500H#

show platform hardware fed active qos queue config interface tw1/0/3
```

```
Asic:0 Core:1 DATA Port:22 GPN:103 LinkSpeed:0x12
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 176 - 183
  DrainFast:Disabled PortSoftStart:5 - 7200 BufferSharing:Disabled
   DTS  Hardmax  Softmax   PortSMin  GlblSMin  PortStEnd   QEnable
  ----- --------  --------  --------  --------  ---------  -------
 0   1  5   600   9   600   0     0   0     0   5  9600       En
```

<--- priority-level 1 queue unaffected by softmax multiplier

```
 1   1  5   600  10
```

4800

```
   8  1200   0     0   5  9600       En
```

<--- User multiplier increased to 200%, softmax for this queue doubles

```
<snip>
```

Queue Limit

The queue-limit configuration affects the final queue buffer allocation

The primary mechanism to influence queue buffer allocation is the **queue buffers-ratio** configuration added per-queue in a MQC policy-map.

However, queue buffer allocation is affected is affected by other configurations.

Queue-limit defines thresholds by which you drop a particular class of traffic (via Weighted Tail Drop, WTD), which is not covered in this document.

In a specific circumstance, **queue-limit** modifies the system Hidden Multiplier for soft buffer queues - which affects overall soft buffer allocation for that queue that **queue-limit** is applied to.

First, understand that queue-limit can be configured up to 3 times per class. This defines up to 3 thresholds for WTD on a per DSCP, or CoS basis.

In the next output, only two thresholds are defined.

Apply two queue-limits to a policy-map:

<#root>

9500H(config)#

policy-map test7

9500H(config-pmap)#

class class1

9500H(config-pmap-c)#

priority level 1

9500H(config-pmap-c)#

queue-buffers ratio 50

9500H(config-pmap-c)#

class class-default

9500H(config-pmap-c)#

priority level 2

9500H(config-pmap-c)#

queue-buffers ratio 50

9500H(config-pmap-c)#

queue-limit dscp af11 percent 10 <-- Tells system to drop af11 traffic at 10% queue utilization

9500H(config-pmap-c)#

queue-limit dscp af12 percent 50 <-- Tells system to drop af12 traffic at 50% queue utilization

Observe the results of buffer allocation:

<#root>

```
9500H(config-pmap-c)#
```

interface tw1/0/3

```
9500H(config-if)#
```

service-policy output test7

```
9500H#
```

show platform hardware fed active qos queue config interface tw1/0/3

```
Asic:0 Core:1 DATA Port:22 GPN:103 LinkSpeed:0x12
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 176 - 183
  DrainFast:Disabled PortSoftStart:5 - 7200 BufferSharing:Disabled
   DTS  Hardmax  Softmax   PortSMin  GlblSMin  PortStEnd   QEnable
  ----- --------  --------  --------  --------  ---------  -------
 0   1  5   600   9   600   0     0   0     0   5  9600      En
 1   1  5   600  10
```

4800

```
   8  1200   0     0   5  9600      En
```

<--- final result for queue that contains 2 queue-limit statements is 4800

```
<snip>
```

In the next example, a third **queue-limit** configuration is added to class class-default.

Observe the results of buffer allocation:

<#root>

```
9500H(config)#
```

policy-map test7

```
9500H(config-pmap)#
```

class class-default

```
9500H(config-pmap-c)#
```

queue-limit dscp af13 percent 100

```
9500H#
```

show platform hardware fed active qos queue config interface tw1/0/3

```
Asic:0 Core:1 DATA Port:22 GPN:103 LinkSpeed:0x12
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 176 - 183
  DrainFast:Disabled PortSoftStart:5 - 1800 BufferSharing:Disabled
   DTS  Hardmax  Softmax   PortSMin  GlblSMin  PortStEnd   QEnable
  ----- --------  --------  --------  --------  ---------  -------
 0   1  5   600   9   600   0     0   0     0   5  2400       En
 1   1  5   600  10
```

1200

```
  32  1200   0     0   5  2400       En
```

<-- Softmax reduces by 400% from previous example

```
<snip>
```

When a third **queue-limit** configuration is added to a queue, the system hidden soft buffer multiplier of 400% is disabled for that queue. However, that queue still respects a user configured **qos queue-softmax-multiplier <percent>.**

Related Information

- Cisco Catalyst 9000 Switching Platforms: QoS and Queuing White Paper
- Cisco Technical Support & Downloads