Understand QoS Hardware Resources on Catalyst 9000 Switches

Contents

Introduction

This document describes how to understand an verify Quality of Service (QoS) hardware utilization on UADP ASIC based Catalyst 9000 Series Switches

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Cisco MQC QoS configuration; policy-maps, class-maps, access-control lists, access-control entries

Components Used

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

- Cisco Catalyst 9200L Cisco IOS®-XE 17.3.4

The general concepts, ideas, and various outputs can be seen in other Cisco Catalyst 9000 Series

Switches.

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 9300 - 9600 Series Switches
- Catalyst 9300X & 9400X
- Cisco IOS® XE 16.x & 17.x Software Versions

Background Information

- Various features on Catalyst 9000 Series Switches consume limited hardware resources. These resources exist to accelerate the performance of those features, and to deliver the expected high forward rates expected from a switch.
- The scale of these resources can vary from switch model to switch model, but the basic methodology to troubleshoot remains the same across Catalyst 9000 Series Switches with the UADP ASIC
- Commonly, the primary limited hardware resource with Switches is referred to as TCAM - Ternary Content Addressable Memory
- In Catalyst 9000 Series switches, multiple memory types are used beyond TCAM, suited to specific needs of a given feature

This document helps you to:

- **Understand** how Quality of Service (QoS) consumes hardware entries
- **Understand** logs or error messages that indicate a QoS hardware resource issue
- **Determine** what actions to take in order remediate hardware resource issues related to QoS

Terminology

QoS	Quality of Service	A concept / group of related features related to classify, mar queue, and schedule traffic in and out of a network device
TCAM	Ternary Content-Addressable Memory	A type of memory that stores and queries entries with three different inputs: 0, 1 and X. This type of memory is used in c where there are multiple matches to the same entry, and the resulting Hash for each would not be unique. This table inclu a mask or X value that allows it to know if it matches or does match this entry.
DSCP	Differentiated Services Code Point	A traffic classification mechanism contained in the IP Heade packet

CoS	Class of Service	A traffic classification mechanism contained in the Ethernet f 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
FED	Forward Engine Driver	Software component that programs the hardware of the devi

Review QoS Related Syslogs

If you run out of QoS related resources, SYSLOG messages are generated by the system:

QoS related Syslog Message	Definition	Recovery Actions
%FED_QOS_ERRMSG-4-TCAM_OVERFLOW: Switch 1 R0/0: fed: Failed to program TCAM for policy-map ingress_pmap2 on GigabitEthernet1/0/10.	Hardware (TCAM) reserved for QoS entires has run out of space	Ensure you have a valid / supp configuration. Then, review the remainder of this document to validate the current scale utiliz of your switch and possible ste reduce if it is overutilized.
%FED_QOS_ERRMSG-3-QUEUE_SCHEDULER_HW_ERROR: Switch 1 R0/0: fed: Failed to configure queue scheduler for GigabitEthernet1/0/27	Installation to hardware of QoS queue scheduler has failed	Verify your configuration is supported, review the QoS configuration guide for your sp platform and version of softwar For 9200L **ONLY**: Review Cisc ID CSCvz54607 and Cisco bug CSCvz76172
FED_QOS_ERRMSG-3-QUEUE_BUFFER_HW_ERROR: R0/0: fed: Failed to configure default queue buffer	Installation to hardware of QoS queue buffers has failed	Verify your configuration is supporte, review the QoS configuration guide for your sp platform and version of softwar Review Cisco bug ID CSCvs49

Validate Hardware Utilization and Policy Status

Verify current QoS TCAM utilization

```
show platform hardware fed switch active fwd-asic resource tcam utilization
```

Note: See for more details on this command

16.X versions:

```
CAM Utilization for ASIC  [0]
 Table                                              Max Values        Used Values
 ----------------------------------------------------------------------------------
```

```
 Unicast MAC addresses                              16384/256          15/21
 L3 Multicast entries                               1024/256           0/7
 L2 Multicast entries                               1024                9
 Directly or indirectly connected routes            8192/3072          2/19
 QoS Access Control Entries                         1024                40 <<< QoS Entries
 Security Access Control Entries                    1408               125
 Ingress Netflow ACEs                                128                8
 Policy Based Routing ACEs                           512                9
 Egress Netflow ACEs                                 128                8
 Flow SPAN ACEs                                      256               13
 Control Plane Entries                               512               211
 Tunnels                                             128               17
 Lisp Instance Mapping Entries                       128                3
 SGT_DGT                                            2048/256           0/1
 CLIENT_LE                                          2048/64            0/0
 INPUT_GROUP_LE                                     1024                0
 OUTPUT_GROUP_LE                                    1024                0
 Macsec SPD                                          128                2
```

17.x Versions:

```
Codes: EM - Exact_Match, I - Input, O - Output, IO - Input & Output, NA - Not Applicable CAM
Utilization for ASIC [0] Table Subtype Dir Max Used %Used V4 V6 MPLS Other ----------------------
-------------------------------------------------------------------------------- Mac Address
Table EM I 16384 17 0.10% 0 0 0 17 Mac Address Table TCAM I 256 21 8.20% 0 0 0 21 L3 Multicast
EM I 1024 0 0.00% 0 0 0 0 L3 Multicast TCAM I 256 9 3.52% 3 6 0 0 L2 Multicast TCAM I 1024 11
1.07% 3 8 0 0 IP Route Table EM I 4096 3 0.07% 2 0 1 0 IP Route Table TCAM I 2048 19 0.93% 6 10
2 1  QOS ACL                TCAM          IO        1024        85    8.30%        28        38
0       19  <-- QoS Entries
 Security ACL            TCAM          IO        1408       129    9.16%        26        58         0
45
 Netflow ACL             TCAM          I          128         6    4.69%         2         2         0
2
 PBR ACL                 TCAM          I          512         9    1.76%         3         6         0
0
 Netflow ACL             TCAM          O          128         6    4.69%         2         2         0
2
 Flow SPAN ACL           TCAM          IO         256        13    5.08%         3         6         0
4
 Control Plane           TCAM          I          512       262   51.17%       114       106         0
42
 Tunnel Termination      TCAM          I          128        18   14.06%         8        10         0
0
 Lisp Inst Mapping       TCAM          I          128         1    0.78%         0         0         0
1
 CTS Cell Matrix/VPN
 Label                   EM            O         2048         0    0.00%         0         0         0
0
 CTS Cell Matrix/VPN
 Label                   TCAM          O          256         1    0.39%         0         0         0
1
 Client Table            EM            I         2048         0    0.00%         0         0         0
0
 Client Table            TCAM          I           64         0    0.00%         0         0         0
0
 Input Group LE          TCAM          I         1024         0    0.00%         0         0         0
0
 Output Group LE         TCAM          O         1024         0    0.00%         0         0         0
0
 Macsec SPD              TCAM          I          128         2    1.56%         0         0         0
2
```

Verify QoS policy is installed in hardware succesfully. Ensure the state is **VALID** and **SET_INHW**. Look at the bottom of the list for physical interface entries. In switch stacks or stackwise-virtual,

use the switch number, or active / standby to accurately reflect which switch you wish to validate hardware installation on.

```
C9200(config)#policy-map egress_pmap
C9200(config-pmap)#interface gi2/0/9
C9200(config-if)#service-policy output egress_pmap

C9200#show platform software fed switch 2 qos policy target status         <-- switch 2 is used
because the interface in question is Gi2/0/9 which is on switch 2

TCG status summary:

Loc Interface             IIF-ID           Dir State:(cfg,opr) Policy
--- --------------------- ---------------- --- --------------- --------------------
<snip> L:0 GigabitEthernet2/0/9 0x00000000000010 OUT VALID,SET_INHW  egress_pmap <-- VALID /
SET_INHW indicates the policy is understood by software and installed to hardware successfully
```

If you see an invalid policy or error instead of VALID / SET_INHW for a target interface, review the QoS policy and validate length and syntax. Also verify hardware utilization. Later sections of this document detail how to understand the resources a policy can consume.

```
C9200#show run policy-map egress_pmap
Current configuration : 624 bytes
!
policy-map egress_pmap
 class COS_DSCP6
  priority level 1
  queue-buffers ratio 5
 class COS_DSCP5
  bandwidth remaining percent 10
  queue-buffers ratio 5
<snip...>

C9200#show run class-map COS_DSCP6
Current configuration : 66 bytes
!
class-map match-any COS_DSCP6
match ip dscp ef
!
end
```

Understand Current Utilization of QoS Hardware Resources

Usage Example (9200L 17.3.4)

```
C9200#show platform hardware fed switch active fwd-asic resource tcam utilization | i
Codes|ASIC|-|QOS
Codes: EM - Exact_Match, I - Input, O - Output, IO - Input & Output, NA - Not Applicable CAM
Utilization for ASIC [0] Table Subtype Dir Max Used    %Used       V4       V6     MPLS    Other
 ---------------------------------------------------------------------------------------------
-------
 QOS ACL                TCAM         IO       1024       85    8.30%       28       38        0
19 <-- Baseline utilization with minimal configuration
```

Configure and **attach** a blank policy-map - no class-maps have been called in this policy-map, so this policy has no intended effect.

```
C9200(config)#policy-map egress_pmap
C9200(config-pmap)#interface gi1/0/9
C9200(config-if)#service-policy output egress_pmap

C9200#show platform hardware fed switch active fwd-asic resource tcam utilization | i
Codes|ASIC|-|QOS
Codes: EM - Exact_Match, I - Input, O - Output, IO - Input & Output, NA - Not Applicable CAM
Utilization for ASIC [0] Table Subtype Dir Max  Used    %Used       V4       V6     MPLS
Other
 ----------------------------------------------------------------------------------------------
-------
 QOS ACL                TCAM         IO       1024       89    8.69%       29       40        0
20 <-- 4 additional entries consumed
```

Observe that with even with zero class-maps attached or actions taken, 4 hardware entries are used, split across V4, V6, and Other.

In this example, a blank test class is added. In a normal scenario, this match-any class-map would allow multiple types of DSCP, CoS, or IPP labels to be matched. But for the example, no values have been called, so the class-map matches no traffic.

```
C9200(config)#class-map match-any TEST_CLASS
C9200(config-cmap)#policy-map egress_pmap
C9200(config-pmap)#class TEST_CLASS

C9200#show platform hardware fed switch active fwd-asic resource tcam utilization | i
Codes|ASIC|-|QOS
Codes: EM - Exact_Match, I - Input, O - Output, IO - Input & Output, NA - Not Applicable CAM
Utilization for ASIC [0] Table Subtype Dir Max Used    %Used       V4       V6     MPLS    Other
 ----------------------------------------------------------------------------------------------
-------
 QOS ACL                TCAM         IO       1024       92    8.92%       30       42        0
20 <-- 3 additional entries consumed
```

The example shows that for each additonal class called, even without any specific traffic matched, a baseline of one v4 entry and two v6 entries are consumed.

As you **add** a match statement to each class, further entries are used:

```
C9200(config)#class-map match-any TEST_CLASS
C9200(config-cmap)#match precedence 0

C9200(config-cmap)#do show platform hardware fed switch ac fwd resource tcam utilization | i QOS
 QOS ACL                TCAM         IO       1024       96    9.38%       31       44        0
21 <-- 4 additional entries

C9200(config-cmap)#match precedence 1

C9200(config-cmap)#do show platform hardware fed switch ac fwd resource tcam utilization | i QOS
 QOS ACL                TCAM         IO       1024       99    9.67%       32       46        0
21 <-- 3 additional entries

C9200(config-cmap)#match cos 1

C9200(config-cmap)#do show platform hardware fed switch ac fwd resource tcam utilization | i QOS
 QOS ACL                TCAM         IO       1024      100    9.77%       32       46        0
22 <-- 1 additional entry

C9200(config-cmap)#match dscp 21

C9200(config-cmap)#do show platform hardware fed switch ac fwd resource tcam utilization | i QOS
```

```
 QOS ACL                 TCAM         IO       1024      103   10.06%       33       48        0
22 <-- 3 addditional entries

C9200(config-cmap)#match dscp 22

C9200(config-cmap)#do show platform hardware fed switch ac fwd resource tcam utilization | i QOS
 QOS ACL                 TCAM         IO       1024      103   10.06%       33       48        0
22 <-- 0 additional entries

C9200(config-cmap)#match dscp 23

C9200(config-cmap)#do show platform hardware fed switch ac fwd resource tcam utilization | i QOS
 QOS ACL                 TCAM         IO       1024      106   10.35%       34       50        0
22 <-- 3 additional entries

C9200(config-cmap)#match dscp 31

C9200(config-cmap)#do show platform hardware fed switch ac fwd resource tcam utilization | i QOS
 QOS ACL                 TCAM         IO       1024      109   10.64%       35       52        0
22 <-- 3 additional entries

C9200(config-cmap)#match dscp 32

C9200(config-cmap)#do show platform hardware fed switch ac fwd resource tcam utilization | i QOS
 QOS ACL                 TCAM         IO       1024      109   10.64%       35       52        0
22 <-- 3 additional entries

C9200(config-cmap)#match dscp 33

C9200(config-cmap)#do show platform hardware fed switch ac fwd resource tcam utilization | i QOS
 QOS ACL                 TCAM         IO       1024      112   10.94%       36       54        0
22 <-- 3 additional entries
```

Observe that in some instances, a single match statement consumes no further entries. Further observe that subseqent match statements consume multple entries.

Before you implement a policy network wide, test the policy as you develop it periodically, and make optimizations as you proceed.

> **Note**: For QoS related hardware utilization, the hardware usage does not always scale one-to-one with match statements or Access Control Entries (ACEs). The hardware operates in terms of Value Mask Result, or VMR. In some scenarios, more than one VMR can be needed to fully classify the range of data necessary to fulfill an ACE. Catalyst 9000 Series Switches UADP Family ASICs contain hardware to optimize these sceanrios, such as for those ACEs with port range operations (L4OPs), to reduce the need for expansion.

Troubleshoot Hardware Utilization

This section presents multiple scenarios with this combination of hardware and software to help illustarate a problem scenario and remediation.

- Platform - C9200L-48T-4X
- Cisco IOS®-XE 17.3.4

The presented scenarios illustrate:

- A small policy which addes a relatively small amount of entries to overall utilization
- A large policy which adds a relatively large amount of entries to overall utilization

- A second large policy which causes a failure to install that policy
- Remediation of the failure to install

Scenario: QoS TCAM Scale Estimation

> **Note**: These examples use Object-Group based ACLs. Object groups efficiently represent much larger traditional access-lists. They do not inherently consume more or less TCAM. Rather, they are a simplified and modular way to represent what would otherwise be very long, patterned lists of ACEs.

This example uses an ingress policy to mark packets. It involves Object-Groups, IP Access-Lists, and TCP/UDP Port based matches.

Object Groups	**Access List which uses the Object Group**	**Class Map**	**Policy Map**
object-group network **RFC1918-Private-IPv4** 10.0.0.0 255.0.0.0 172.16.0.0 255.240.0.0 192.168.0.0 255.255.0.0 object-group network **app_1** group-object **RFC1918-Private-IPv4**	ip access-list extended **APP_1_PORTS_1** 10 permit udp any object-group **app_1** range 1433 1434 20 permit udp object-group **app_1** range 1433 1434 any 30 permit tcp any object-group **app_1** range 1433 1434 40 permit tcp object-group **app_1** range 1433 1434 any 50 permit tcp any object-group **app_1** range 14300 14400 60 permit tcp object-group **app_1** range 14300 14400 any	class-map match-any **BigClass** match access-group name **APP_1_PORTS_1**	policy-map ingress_pma class **BigCl** set dscp cs

Review the chart, and note there are 3 subnets in *object-group network RFC1918-Private-IPv4*

```
object-group network app_1
 group-object RFC1918-Private-IPv4

object-group network RFC1918-Private-IPv4
10.0.0.0 255.0.0.0
172.16.0.0 255.240.0.0
192.168.0.0 255.255.0.0
```

Further, there 6 match statements in *ip access-list extended APP_1_PORTS_1.*

```
ip access-list extended APP_1_PORTS_1
 10 permit udp any object-group app_1 range 1433 1434 <-- permits any source, to group app_1 on
UDP ports 1433 - 1434
 20 permit udp object-group app_1 range 1433 1434 any <-- reverse of previous line, reminder
that app_1 is made up of RFC1918-Private-IPv4, which is 3 separate subnets
 30 permit tcp any object-group app_1 range 1433 1434
 40 permit tcp object-group app_1 range 1433 1434 any
 50 permit tcp any object-group app_1 range 14300 14400
 60 permit tcp object-group app_1 range 14300 14400 any
```

object-group network app_1 applies every entry in *object-group network RFC1918-Private-IPv4* to

every entry in *ip access-list extended APP_1_PORTS_1*

This has has a multiplicative effect, because for each ACE in *APP_1_PORTS_1*, it references object-group app_1 which itself representes 3 additional ACEs from RFC1918-Private-IPv4

Total utilization estimate for ip access-list extended APP_1_PORTS_1, when attached to a class-map and policy-map is:

APP_1 used 6 times x 3 object-group ACEs = 18

Apply the policy and observe TCAM utilization:

```
C9200#show platform hardware fed switch 2 fwd-asic resource tcam utilization | i Codes|ASIC|-|QOS
Codes: EM - Exact_Match, I - Input, O - Output, IO - Input & Output, NA - Not Applicable CAM
Utilization for ASIC [0] Table Subtype Dir Max Used    %Used       V4       V6     MPLS    Other
 ---------------------------------------------------------------------------------------------
-------
 QOS ACL                TCAM         IO       1024      85    8.69%       29       40        0
20 <-- baseline utilization

C9200(config-pmap)#interface gi1/0/9
C9200(config-if)#service-policy input ingress_pmap

C9200#show platform hardware fed switch active fwd-asic resource tcam utilization | i Codes|ASIC|-|QOS
Codes: EM - Exact_Match, I - Input, O - Output, IO - Input & Output, NA - Not Applicable CAM
Utilization for ASIC [0] Table Subtype Dir Max Used    %Used       V4       V6     MPLS    Other
 ---------------------------------------------------------------------------------------------
-------
 QOS ACL                TCAM         IO       1024     107   10.45%      47       40        0
20 <-- 22 entries consumed
```

Summary

- The ACLs define object groups which expand to consume **18** additional entries, due to the multiplicative effect of object groups
- The policy map consumes **4** entries by default
- This adds to **22 entries consumed**

Scenario: QoS TCAM Scale Increased (not exceeded)

This example is a continuation of the previous with a larger policy. This establishes how you can quickly consume a large amount of TCAM.

Policy 1:

Object Groups	Access Lists which use the Object Groups	Class Map	Policy Map
object-group network **experimental_1** 240.1.192.0 255.255.192.0	ip access-list extended **APP_1_PORTS_1** 10 permit udp any object-group **app_1** range 1433 1434	class-map match-any **BigClass_1** match access-group name **APP_3_PORTS_2**	policy-map big_ingress_p class

240.2.96.0 255.255.224.0
240.3.160.0
255.255.240.0
240.4.32.0 255.255.224.0
240.5.160.0
255.255.224.0
240.6.192.0
255.255.224.0
240.7.128.0
255.255.128.0
240.8.0.0 255.255.0.0
240.9.128.0
255.255.192.0
240.10.224.0
255.255.224.0
240.11.0.0 255.255.240.0
240.12.160.0
255.255.224.0
240.13.192.0
255.255.224.0
240.14.192.0
255.255.240.0
240.15.128.0
255.255.224.0
object-group network **experimental_2**
241.0.0.0 255.255.192.0
241.4.0.0 255.252.0.0
241.8.0.0 255.252.0.0
host 241.12.1.1
host 241.13.1.1
host 241.14.1.1
host 241.15.1.1
241.16.0.0 255.252.0.0
host 241.20.1.1
host 241.21.1.1
host 241.22.1.1
host 241.23.1.1
object-group network **RFC1918-Private-IPv4**
10.0.0.0 255.0.0.0
172.16.0.0 255.240.0.0
192.168.0.0 255.255.0.0

object-group **network app_1**
group-object **RFC1918-Private-IPv4**

object-group **network app_2**
group-object **RFC1918-Private-IPv4**

20 permit udp object-group **app_1** range 1433 1434 any
<4 more lines>
ip access-list extended **APP_1_PORTS_2**
10 permit udp any object-group **app_1** range 7750 7759
20 permit udp object-group **app_1** range 7750 7759 any
<18 more lines>
ip access-list extended **APP_1_PORTS_3**
10 permit udp any object-group **app_1** range 22030 22031
20 permit udp object-group **app_1** range 22030 22031 any
<6 more lines>
ip access-list extended **APP_2_PORTS_1**
10 permit udp any object-group **app_2** range 6000 9291
20 permit udp object-group **app_2** range 6000 9291 any
ip access-list extended **APP_3_PORTS_1**
10 permit tcp any object-group **app_3** eq 7563
20 permit tcp object-group **app_3** eq 7563 any
<4 more lines>
ip access-list extended **APP_3_PORTS_2**
10 permit udp any object-group **app_3** eq 554
20 permit udp object-group **app_3** eq 554 any
<2 more lines>
ip access-list extended **APP_3_PORTS_3**
10 permit udp any object-group **app_3** eq 22331
20 permit udp object-group **app_3** eq 22331 any
<2 more lines>
ip access-list extended **APP_3_PORTS_4**
10 permit tcp any object-group **app_3** eq 5432
20 permit tcp object-group **app_3** eq 5432 any
<6 more lines>
ip access-list extended **APP_4_PORTS_1**

class-map match-any **BigClass_2**
match access-group name **APP_4_PORTS_1**
class-map match-any **BigClass_3**
match access-group name **APP_1_PORTS_2**
match access-group name **APP_3_PORTS_3**
match access-group name **APP_2_PORTS_1**
class-map match-any **BigClass_4**
match access-group name **APP_1_PORTS_3**
match access-group name **APP_3_PORTS_4**
class-map match-any **BigClass_5**
match access-group name **APP_1_PORTS_1**
match access-group name **APP_3_PORTS_1**

BigClass_1
set dscp cs
class
BigClass_2
set dscp af
class
BigClass_3
set dscp cs
class
BigClass_4
set dscp af
class
BigClass_5
set dscp cs
class class-
default

object-group **network app_3**
 group-object **RFC1918-Private-IPv4**
object-group **network app_4**
 group-object **RFC1918-Private-IPv4**
 group-object **experimental_1**
 group-object **experimental_2**

10 permit udp any object-group app_4 range 1718 1719
20 permit udp object-group app_4 range 1718 1719 any
<14 more lines>