QoS (Quality of Service)

QoS Classification
QoS Classification is used to classify the packets to assign a Traffic Class (TC) value based on service requirements. TC is the internal priority in switch (8 possible value) using for packet buffering and scheduling. TC-to-priority-group (PG) and TC-to-Queue maps depend on the TC maps to the ingress buffer (priority-group) and egress queue, respectively. TC is used as a priority field for packet re-marking on egress port. Determines the TC based on the priority of the incoming packet and trust mode of a port.

Dot1p-to-TC Mapping is assigned the TC based on PCP (Priority Code Point) in the VLAN tag, and this mapping table is configured per ingress physical port.
Default Dot1p-to-TC mapping on Broadcom platform:

Dot1P	0	1	2	3	4	5	6	7
TC	0	1	2	3	4	5	6	7

Default Dot1p-to-TC mapping on the Intel platform:

Dot1P	0~7
TC	0

DSCP-to-TC Mapping is assigned the TC based on DSCP field in the IP header, and this mapping table is configured per ingress physical port.
Default DSCP-to-TC mapping:

DSCP	0~63
TC	0

TC-to-Queue Mapping is assigned the Queue ID based on TC, and this mapping table is configured per ingress physical port.
Default TC-to-Queue mapping on Broadcom platform:

TC	0	1	2	3	4	5	6	7
Queue	0	1	2	3	4	5	6	7

Default TC-to-Queue mapping on Intel platform:

TC	0~7
Queue	0

Tested model & firmware version:
Switch model name:
DCS503 (AS9716-32D)
Edgecore SONiC version:
202111.1
202111.3
202111.8
Topology:
mceclip1.png

Pre-configuration:
Add VLAN 10 to Ethernet0, Ethernet8.

```
admin@sonic:~$ show vlan brief
+-----------+--------------+-----------+----------------+-------------
+---------------+--------------+-------------+
|   VLAN ID | IP Address   | Ports     | Port Tagging   | Proxy ARP   | DHCP
Helper   | DHCP Source   | DHCP Link   |
|           |              |           |                |             | Address
| Interface     | Selection   |
+===========+==============+===========+================+=============+=========
======+===============+==============+
|        10 |              | Ethernet0 | tagged         | disabled    |
|               |              |
|           |              | Ethernet8 | tagged         |             |
|               |              |
```

```
+-----------+--------------+-----------+----------------+-------------
+---------------+--------------+-------------+
Procedure:
Step 1: Create a profile for DOT1P/DSCP to TC(Traffic Class).

Example for Dot1P:

admin@sonic:~$ sudo config qos dot1p-tc add 1p_tc --dot1p 0 --tc 1
Example for DCSP:

admin@sonic:~$ sudo config qos dscp-tc add DSCP_TC --dscp 7 --tc 1
Note:

admin@sonic:~$ sudo config qos dot1p-tc add --help
Usage: config qos dot1p-tc add [OPTIONS] <profile>

  Add dot1p-tc map profile.

Options:
  --dot1p TEXT        Cos value  [required]
  --tc INTEGER RANGE  Traffic-class(TC) value  [required]
  -h, -?, --help      Show this message and exit.


Step 2: Modify the existing Dot1p/DSCP to TC profile.

Example for DOT1P:

admin@sonic:~$ sudo config qos dot1p-tc update 1p_tc --dot1p 1 --tc 2
Example for DCSP:

admin@sonic:~$ sudo config qos dscp-tc update DSCP_TC --dscp 8 --tc 2
Note:

admin@sonic:~$ sudo config qos dot1p-tc update --help
Usage: config qos dot1p-tc update [OPTIONS] <profile>

  Update dot1p-tc map profile.

Options:
  --dot1p TEXT        Cos value  [required]
  --tc INTEGER RANGE  Traffic-class(TC) value
  --remove            Delete the mapping entry
  -?, -h, --help      Show this message and exit.


Step 3: Check the profile of DOT1P/DSCP to TC.

DOT1P to TC:

admin@sonic:~$ show qos dot1p-tc
dot1p-tc policy: 1p_tc
  Dot1p    TC
-------  ----
      0     1
      1     2
DSCP to TC:

admin@sonic:~$ show qos dscp-tc
dscp-tc policy: DSCP_TC
  DSCP    TC
------  ----
     7     1
     8     2
```

Step 4: Create a profile for TC to Queue.

```
admin@sonic:~$ sudo config qos tc-queue add TC_Q --tc 1 --queue 2
Note:

admin@sonic:~$ sudo config qos tc-queue add --help
Usage: config qos tc-queue add [OPTIONS] <profile>

  Add tc-queue map profile.

Options:
  --tc TEXT              Traffic-class(TC) value  [required]
  --queue INTEGER RANGE  Queue value  [required]
  -?, -h, --help         Show this message and exit.
```

Step 5: Modify the existing TC to Queue profile.

```
admin@sonic:~$ sudo config qos tc-queue update TC_Q --tc 2 --queue 3
Note:

admin@sonic:~$ sudo config qos tc-queue update --help
Usage: config qos tc-queue update [OPTIONS] <profile>

  Update tc-queue map profile.

Options:
  --tc TEXT              Traffic-class(TC) value  [required]
  --queue INTEGER RANGE  Queue value
  --remove               Delete the mapping entry
  -?, -h, --help         Show this message and exit.
```

Step 6: Check the profile of TC to Queue.

```
admin@sonic:~$ show qos tc-queue
tc-queue policy: TC_Q
  TC    Queue
----  -------
   1        2
   2        3
```

Step 7: Binding the mapping table to the specified interface.

```
admin@sonic:~$ sudo config interface qos dot1p-tc bind Ethernet0 1p_tc
admin@sonic:~$ sudo config interface qos dscp-tc bind Ethernet0 DSCP_TC
admin@sonic:~$ sudo config interface qos tc-queue bind Ethernet0 TC_Q
Note:

admin@sonic:~$ sudo config interface qos dot1p-tc --help
Usage: config interface qos dot1p-tc [OPTIONS] <op> <interface_name> <profile>

  dot1p-tc policy configuration

Options:
  -?, -h, --help  Show this message and exit.
"op": "bind/unbind"
```

Step 6: Check the binding table.

```
admin@sonic:~$ show interfaces qos
Ethernet0:
```

Dot1p to TC: 1p_tc
DSCP to TC: DSCP_TC
TC to Queue: TC_Q

Result: Check the queue counters

When the VLAN 10 packet with priority 0 into the switch, it will follow the mapping table that we set.
Dot1p ------ > TC ------ > Queue

Based on the above setting, the VLAN 10 packet with priority 0 will be forwarded to Queue2.

Step 1. Clear the queue counter

```
admin@sonic:~$ sonic-clear queuecounters
```

Step 2. Check Ethernet8 (egress port) queue counter

```
admin@sonic:~$ show queue counters Ethernet8
Last cached time was 2022-09-13 21:39:53.822559
     Port    TxQ    Counter/pkts    Counter/bytes    Drop/pkts    Drop/bytes
---------  -----  --------------  ---------------  -----------  ------------
Ethernet8    UC0               0                0            0             0
Ethernet8    UC1               0                0            0             0
Ethernet8    UC2          37,846        2,422,144            0             0
Ethernet8    UC3               0                0            0             0
Ethernet8    UC4               0                0            0             0
Ethernet8    UC5               0                0            0             0
Ethernet8    UC6               0                0            0             0
Ethernet8    UC7               0                0            0             0
Ethernet8    UC8               0                0            0             0
Ethernet8    UC9               0                0            0             0
Ethernet8   MC10               0                0            0             0
Ethernet8   MC11               0                0            0             0
```

QoS Marking
Once the packets have been classified by QoS Classification, QoS Marking allows to set or change the priority value in the packets before sending them. The new priority value is brought to the next switches for fine-tuning or uniform the class of the traffic in the network.

QoS Marking enables to rewrite the Dot1p or DSCP of incoming packets based on Traffic Class (TC) value.

Enable Dot1p rewrite by configuring a TC-to-Dot1p mapping on an egress port. On Broadcom switches, the Dot1p rewrite is always enabled.
Default TC-to-Dot1p mapping.

```
TC    0     1     2     3     4     5     6     7
Dot1P 0     1     2     3     4     5     6     7
```

Enable DSCP rewrite by configuring a TC-to-DSCP mapping on an egress port. The DSCP rewrite is disabled by default on all ports.
Tested model & firmware version:
Switch model name:
DCS503 (AS9716-32D)
Edgecore SONiC version:
202111.1
202111.3
Topology:

mceclip0.png

Pre-configuration:
Add VLAN 10 to Ethernet0, Ethernet8.

```
admin@sonic:~$ show vlan brief
+-----------+-------------+-----------+----------------+-------------
+---------------+--------------+-------------+
|   VLAN ID | IP Address   | Ports     | Port Tagging   | Proxy ARP   | DHCP
Helper   | DHCP Source   | DHCP Link   |
|           |              |           |                |             | Address
| Interface     | Selection   |
+===========+==============+===========+================+=============+=========
======+===============+=============+
|        10 |              | Ethernet0 | tagged         | disabled    |
|               |              |
|           |              | Ethernet8 | tagged         |             |
|               |              |
+-----------+-------------+-----------+----------------+-------------
+---------------+--------------+-------------+
```

Procedure:
Step 1: Create a profile for DOT1P remarking.

Example for Dot1P:

```
admin@sonic:~$ sudo config qos remark dot1p add remark_dot1p --tc 0 --dot1p 1
```

Example for DSCP:

```
admin@sonic:~$ sudo config qos remark dscp add remark_dscp --tc 0 --dscp 7
```

Note:

```
admin@sonic:~$ sudo config qos remark dot1p add --help
Usage: config qos remark dot1p add [OPTIONS] <profile>

  Add tc-dot1p map profile.

Options:
  --tc TEXT              Traffic-class(TC) value  [required]
  --dot1p INTEGER RANGE  Cos value  [required]
  -?, -h, --help         Show this message and exit.
```

Step 2: Modify the existing remark profile.

Example for Dot1P:

```
admin@sonic:~$ sudo config qos remark dot1p update remark_dot1p --tc 1 --dot1p 2
```

Example for DSCP:

```
admin@sonic:~$ sudo config qos remark dscp update remark_dscp --tc 1 --dscp 8
```

Note:

```
admin@sonic:~$ sudo config qos remark dot1p update --help
Usage: config qos remark dot1p update [OPTIONS] <profile>

  Update tc-dot1p map profile.

Options:
  --tc TEXT              Traffic-class(TC) value  [required]
  --dot1p INTEGER RANGE  Cos value
  --remove               Delete the mapping entry
  -h, -?, --help         Show this message and exit.
```

Step 3: Check the remark profile.

```
admin@sonic:~$ show qos remark dot1p
dot1p policy: remark_dot1p
  TC    Dot1p
```

```
----  -------
   0        1
   1        2
```
Step 4: Binding the remark table to the egress interface.

```
admin@sonic:~$ sudo config interface qos remark dot1p bind Ethernet8
remark_dot1p
```
Note:

```
admin@sonic:~$ sudo config interface qos remark dot1p --help
Usage: config interface qos remark dot1p [OPTIONS] <op> <interface_name>
                                         <profile>

  tc-dot1p policy configuration

Options:
  -?, -h, --help  Show this message and exit.
```
"op": "bind/unbind"
Step 5: Check the binding table.

```
admin@sonic:~$ show interfaces qos
Ethernet8:
  Dot1p remark: remark_dot1p
```
Result:

Since this example didn't set the QoS Classification, when the VLAN 10 packet with priority 0 into switch, it will base on the default QoS Classification.
The priority 0 will map to TC 0.
Base on the remark profile, the TC 0 will map to the Dot1p 1 which means the traffic through the TC 0 will be remarked to priority 1.

Here's the packet.

mceclip2.png

QoS Scheduler
Queue scheduling provides preferential treatment of traffic classes mapped to specific egress queues. SONiC supports SP, WRR, and DWRR scheduling disciplines.

SP (Strict Priority)- Higher priority egress queues get scheduled for transmission over lower priority queues.
WRR (Weighted Round Robin) - Egress queues receive bandwidth proportional to the configured weight. The scheduling granularity is per packet which causes large and small packets to be treated the same. Flows with large packets have an advantage over flows with smaller packets.
DWRR (Deficit Weighted Round Robin) - Similar to WRR but uses deficit counter scheduling granularity to account for packet size variations and provide a more accurate proportion of bandwidth.
Tested model & firmware version:
Switch model name:
DCS203 (AS7326-56X)
Edgecore SONiC version:
202111.1
202111.3
Topology:
mceclip3.png

Pre-configuration:
Add VLAN 10 to Ethernet0, Ethernet1, and Ethernet5.

```
admin@sonic:~$ show vlan brief
+-----------+--------------+-----------+----------------+-------------
```

```
+---------------+---------------+-------------+
|   VLAN ID | IP Address    | Ports     | Port Tagging   | Proxy ARP   | DHCP
Helper   | DHCP Source   | DHCP Link   |
|           |               |           |                |             | Address
| Interface     | Selection   |
+===========+===============+===========+================+=============+=========
======+===============+=============+
|        10 |               | Ethernet0 | tagged         | disabled    |
|                |             |
|           |               | Ethernet1 | tagged         |             |
|                |             |
|           |               | Ethernet5 | tagged         |             |
|                |             |
+-----------+---------------+-----------+----------------+-------------
+---------------+---------------+-------------+
```

The QoS Classification uses the default setting.

SP (Strict Priority)
Procedure:
Step 1: Set the scheduler mode.

```
admin@sonic:~$ sudo config scheduler add strict_mode --sched_type STRICT
```

Note:

```
admin@sonic:~$ sudo config scheduler add --help
Usage: config scheduler add [OPTIONS] <profile_name>

  Add QoS-Scheduler profile.

Options:
  --sched_type [WRR|DWRR|STRICT]  Scheduler type. WRR is not supported on
                                  Intel platform.
  --weight INTEGER RANGE          weight
  --shaper_type [bytes|packets]   Shaper type
  --bandwidth TEXT                Maximum bandwidth rate in kbps or pps. If
                                  shaper-type is bytes, user can specify a
                                  decimal number followed by the abbreviation
                                  k (1000), m (1,000,000), or g
                                  (1,000,000,000)
  -h, -?, --help                  Show this message and exit.
```

Step 2: Check status.

```
admin@sonic:~$ show scheduler
Name         Scheduling Type    Weight    Shaper Type    Bandwidth
-----------  -----------------  --------  -------------  -----------
strict_mode  STRICT             N/A       N/A            N/A
```

Step 3: Binding the scheduler to the interface.

```
admin@sonic:~$ sudo config interface scheduler bind queue Ethernet5 3
strict_mode
```

Note:

```
admin@sonic:~$ sudo config interface scheduler bind queue --help
Usage: config interface scheduler bind queue [OPTIONS] <interface_name>
                                             <queue> <sched_policy_name>

Options:
  -h, -?, --help  Show this message and exit.
```

Here's the command to unbind the scheduler.

```
admin@sonic:~$ sudo config interface scheduler unbind queue Ethernet5 3
```

Step 4: Check status.

```
admin@sonic:~$ show interfaces scheduler
Ethernet5
Type      Queue  Profile      Scheduling Mode    Weight    Shaper Type
Bandwidth
------  -------  -----------  -----------------  --------  -------------
-----------
Queue         3  strict_mode  STRICT             N/A       N/A
0
Result:

Injecting the traffic with priority 0~7 from Ethernet0 and Ethernet1 to
Ethernet5 to trigger the congestion.
Queue 3 in Ethernet5 will be the highest priority.

admin@sonic:~$ show queue counters Ethernet5
Last cached time was 2022-09-14 06:40:04.548068
     Port    TxQ    Counter/pkts    Counter/bytes    Drop/pkts    Drop/bytes
---------  -----  --------------  ---------------  -----------  ------------
Ethernet5    UC0       1,824,146      233,490,688    2,407,299   308,134,272
Ethernet5    UC1       1,824,176      233,494,528    2,407,270   308,130,560
Ethernet5    UC2       1,824,179      233,494,912    2,407,292   308,133,376
Ethernet5    UC3       4,255,417      544,693,376            0             0
Ethernet5    UC4       1,824,185      233,495,680    2,407,305   308,135,040
Ethernet5    UC5       1,824,187      233,495,936    2,407,276   308,131,328
Ethernet5    UC6       1,824,189      233,496,192    2,407,272   308,130,816
Ethernet5    UC7       1,824,194      233,496,832    2,407,284   308,132,352
Ethernet5    UC8               0                0            0             0
Ethernet5    UC9               0                0            0             0
Ethernet5   MC10               0                0            0             0
Ethernet5   MC11               0                0            0             0
Ethernet5   MC12               0                0            0             0
Ethernet5   MC13               0                0            0             0
Ethernet5   MC14               0                0            0             0
Ethernet5   MC15               0                0            0             0
Ethernet5   MC16               0                0            0             0
Ethernet5   MC17               0                0            0             0
Ethernet5   MC18               0                0            0             0
Ethernet5   MC19               0                0            0             0
Notes:

8 dedicated queues (i.e UC0, UC1, ..., UC7) for unicast.
Other 8 dedicated queues (i.e MC10, MC11, ..., UC17) for multicast.
UC8, UC9, UC18, UC19 are not available.
WRR (Weighted Round Robin)
Procedure:
Step 1:  Set the scheduler mode.

admin@sonic:~$ sudo config scheduler add wrr_7 --sched_type WRR --weight 7
admin@sonic:~$ sudo config scheduler add wrr_3 --sched_type WRR --weight 3
Step 2: Check status.

admin@sonic:~$ show scheduler
Name    Scheduling Type      Weight  Shaper Type    Bandwidth
------  -----------------  --------  -------------  -----------
wrr_3   WRR                       3  N/A            N/A
wrr_7   WRR                       7  N/A            N/A
Step 3: Binding the scheduler to the interface.

admin@sonic:~$ sudo config interface scheduler bind queue Ethernet5 3 wrr_7
admin@sonic:~$ sudo config interface scheduler bind queue Ethernet5 4 wrr_3
Step 4: Check status.

admin@sonic:~$ show interfaces scheduler
```

```
Ethernet5
Type      Queue  Profile    Scheduling Mode      Weight  Shaper Type
Bandwidth
------  -------  ---------  -----------------  --------  -------------
-----------
Queue         3  wrr_7      WRR                       7  N/A
0
Queue         4  wrr_3      WRR                       3  N/A
0
Result:

Injecting the traffic with priority 3,4 from Ethernet0 and Ethernet1 to
Ethernet5 to trigger the congestion.
The egress traffic will be distributed based on weights.

admin@sonic:~$ show queue counters Ethernet5
Last cached time was 2022-09-23 05:53:03.667274
     Port    TxQ    Counter/pkts    Counter/bytes    Drop/pkts    Drop/bytes
---------  -----  --------------  ---------------  -----------  ------------
Ethernet5    UC0               1              409            0             0
Ethernet5    UC1               0                0            0             0
Ethernet5    UC2               0                0            0             0
Ethernet5    UC3       1,518,614      858,150,900      620,044   348,958,020
Ethernet5    UC4         671,393      379,639,721    1,467,263   827,705,931
Ethernet5    UC5               0                0            0             0
Ethernet5    UC6               0                0            0             0
Ethernet5    UC7               0                0            0             0
Ethernet5    UC8               0                0            0             0
Ethernet5    UC9               0                0            0             0
Ethernet5   MC10               0                0            0             0
Ethernet5   MC11               0                0            0             0
Ethernet5   MC12               0                0            0             0
Ethernet5   MC13               0                0            0             0
Ethernet5   MC14               0                0            0             0
Ethernet5   MC15               0                0            0             0
Ethernet5   MC16               0                0            0             0
Ethernet5   MC17               0                0            0             0
Ethernet5   MC18               0                0            0             0
Ethernet5   MC19               0                0            0             0
Note:

Total: 1,518,614 + 671,393 = 2,190,007
Queue 3: 1,518,614 / 2,190,007 = 0.6934
Queue 4: 671,393 / 2,190,007 = 0.3065



DWRR (Deficit Weighted Round Robin)
Procedure:
Step 1:  Set the scheduler mode.

admin@sonic:~$ sudo config scheduler add dwrr_7 --sched_type DWRR --weight 7
admin@sonic:~$ sudo config scheduler add dwrr_3 --sched_type DWRR --weight 3
Step 2: Check status.

admin@sonic:~$ show scheduler
Name    Scheduling Type      Weight  Shaper Type    Bandwidth
------  -----------------  --------  -------------  -----------
dwrr_3  DWRR                      3  N/A            N/A
dwrr_7  DWRR                      7  N/A            N/A
Step 3: Binding the scheduler to the interface.

admin@sonic:~$ sudo config interface scheduler bind queue Ethernet5 3 dwrr_7
admin@sonic:~$ sudo config interface scheduler bind queue Ethernet5 4 dwrr_3
```

Step 4: Check status.

```
admin@sonic:~$ show interfaces scheduler
Ethernet5
Type      Queue  Profile    Scheduling Mode      Weight  Shaper Type
Bandwidth
------  -------  ---------  -----------------  --------  -------------
-----------
Queue         3  dwrr_7     DWRR                     7  N/A
0
Queue         4  dwrr_3     DWRR                     3  N/A
0
```

Result:

Injecting the traffic with priority 3,4 from Ethernet0 and Ethernet1 to Ethernet5 to trigger the congestion.
The egress traffic will be distributed based on weights.

```
admin@sonic:~$ show queue counters Ethernet5
Last cached time was 2022-09-28 07:07:32.975167
     Port    TxQ    Counter/pkts    Counter/bytes    Drop/pkts    Drop/bytes
---------  -----  --------------  ---------------  -----------  ------------
Ethernet5    UC0               3            1,227            0             0
Ethernet5    UC1               0                0            0             0
Ethernet5    UC2               0                0            0             0
Ethernet5    UC3       5,972,663      764,500,864    2,473,283   316,580,224
Ethernet5    UC4       2,605,564      333,512,192    5,840,382   747,568,896
Ethernet5    UC5               0                0            0             0
Ethernet5    UC6               0                0            0             0
Ethernet5    UC7               0                0            0             0
Ethernet5    UC8               0                0            0             0
Ethernet5    UC9               0                0            0             0
Ethernet5   MC10               0                0            0             0
Ethernet5   MC11               0                0            0             0
Ethernet5   MC12               0                0            0             0
Ethernet5   MC13               0                0            0             0
Ethernet5   MC14               0                0            0             0
Ethernet5   MC15               0                0            0             0
Ethernet5   MC16               0                0            0             0
Ethernet5   MC17               0                0            0             0
Ethernet5   MC18               0                0            0             0
Ethernet5   MC19               0                0            0             0
```

Note:

Total:5,972,663 + 2,605,564 = 8,578,227
Queue 3: 5,972,412 / 8,577,474 = 0.6962
Queue 4: 2,605,062 / 8,577,474 = 0.3037

Hybrid mode (SP + WRR)
Procedure:
Step 1: Set the scheduler mode.

```
admin@sonic:~$ sudo config scheduler add strict_mode --sched_type STRICT
admin@sonic:~$ sudo config scheduler add wrr_7 --sched_type WRR --weight 7
admin@sonic:~$ sudo config scheduler add wrr_3 --sched_type WRR --weight 3
```

Step 2: Check status.

```
admin@sonic:~$ show scheduler
Name         Scheduling Type      Weight  Shaper Type    Bandwidth
-----------  -----------------  --------  -------------  -----------
strict_mode  STRICT                  N/A  N/A            N/A
wrr_3        WRR                       3  N/A            N/A
```

```
wrr_7        WRR                        7  N/A            N/A
```
Step 3: Binding the scheduler to the interface.

```
admin@sonic:~$ sudo config interface scheduler bind queue Ethernet5 2
strict_mode
admin@sonic:~$ sudo config interface scheduler bind queue Ethernet5 3 wrr_7
admin@sonic:~$ sudo config interface scheduler bind queue Ethernet5 4 wrr_3
```
Step 4: Check status.

```
admin@sonic:~$ show interfaces scheduler
Ethernet5
Type      Queue  Profile      Scheduling Mode    Weight    Shaper Type
Bandwidth
------  -------  -----------  -----------------  --------  -------------
-----------
Queue         2  strict_mode  STRICT             N/A       N/A
0
Queue         3  wrr_7        WRR                7         N/A
0
Queue         4  wrr_3        WRR                3         N/A
0
```
Result:

Injecting the traffic with priority 2,3,4 from Ethernet0 and Ethernet1 to Ethernet5 to trigger the congestion.
Queue 2 in Ethernet5 will be the highest priority.
Queue 3, and 4 in Ethernet 5 will distribute the remaining traffic based on weights.

```
admin@sonic:~$ show queue counters Ethernet5
Last cached time was 2022-09-23 05:39:48.815061
     Port    TxQ    Counter/pkts    Counter/bytes    Drop/pkts    Drop/bytes
---------  -----  --------------  ---------------  -----------  ------------
Ethernet5    UC0               1              409            0             0
Ethernet5    UC1               0                0            0             0
Ethernet5    UC2       5,630,631      720,720,768            0             0
Ethernet5    UC3       2,035,749      260,575,872    3,594,882   460,144,896
Ethernet5    UC4         912,664      116,820,992    4,717,966   603,899,648
Ethernet5    UC5               0                0            0             0
Ethernet5    UC6               0                0            0             0
Ethernet5    UC7               0                0            0             0
Ethernet5    UC8               0                0            0             0
Ethernet5    UC9               0                0            0             0
Ethernet5   MC10               0                0            0             0
Ethernet5   MC11               0                0            0             0
Ethernet5   MC12               0                0            0             0
Ethernet5   MC13               0                0            0             0
Ethernet5   MC14               0                0            0             0
Ethernet5   MC15               0                0            0             0
Ethernet5   MC16               0                0            0             0
Ethernet5   MC17               0                0            0             0
Ethernet5   MC18               0                0            0             0
Ethernet5   MC19               0                0            0             0
```
Note:

```
Total: 2,035,749 + 912,664 = 2,948,413
Queue 3: 2,035,749 /  2,948,413 = 0.6904
Queue 4: 912,664 /  2,948,413 = 0.3095
```

Hybrid mode (SP + DWRR)
Procedure:
Step 1: Set the scheduler mode.

```
admin@sonic:~$ sudo config scheduler add strict_mode --sched_type STRICT
admin@sonic:~$ sudo config scheduler add dwrr_7 --sched_type DWRR --weight 7
admin@sonic:~$ sudo config scheduler add dwrr_3 --sched_type DWRR --weight 3
Step 2: Check status.

admin@sonic:~$ show scheduler
Name         Scheduling Type      Weight  Shaper Type    Bandwidth
-----------  -----------------  --------  -------------  -----------
dwrr_3       DWRR                      3  N/A            N/A
dwrr_7       DWRR                      7  N/A            N/A
strict_mode  STRICT                  N/A  N/A            N/A
Step 3: Binding the scheduler to the interface.

admin@sonic:~$ sudo config interface scheduler bind queue Ethernet5 2
strict_mode
admin@sonic:~$ sudo config interface scheduler bind queue Ethernet5 3 dwrr_7
admin@sonic:~$ sudo config interface scheduler bind queue Ethernet5 4 dwrr_3
Step 4: Check status.

admin@sonic:~$ show interfaces scheduler
Ethernet5
Type      Queue  Profile      Scheduling Mode    Weight    Shaper Type
Bandwidth
------  -------  -----------  -----------------  --------  -------------
-----------
Queue         2  strict_mode  STRICT             N/A       N/A
0
Queue         3  dwrr_7       DWRR               7         N/A
0
Queue         4  dwrr_3       DWRR               3         N/A
0
Result:

Injecting the traffic with priority 2,3,4 from Ethernet0 and Ethernet1 to
Ethernet5 to trigger the congestion.
Queue 2 in Ethernet5 will be the highest priority.
Queue 3, and 4 in Ethernet 5 will distribute the remaining traffic based on
weights.

admin@sonic:~$ show queue counters Ethernet5
Last cached time was 2022-09-29 02:36:49.112556
     Port    TxQ    Counter/pkts    Counter/bytes    Drop/pkts    Drop/bytes
---------  -----  --------------  ---------------  -----------  ------------
Ethernet5    UC0               8            2,989            0             0
Ethernet5    UC1               0                0            0             0
Ethernet5    UC2       1,898,110      242,958,080            0             0
Ethernet5    UC3       1,180,033      667,767,452      718,077   403,717,686
Ethernet5    UC4         521,926      295,427,164    1,376,184   776,243,178
Ethernet5    UC5               0                0            0             0
Ethernet5    UC6               0                0            0             0
Ethernet5    UC7               0                0            0             0
Ethernet5    UC8               0                0            0             0
Ethernet5    UC9               0                0            0             0
Ethernet5   MC10               0                0            0             0
Ethernet5   MC11               0                0            0             0
Ethernet5   MC12               0                0            0             0
Ethernet5   MC13               0                0            0             0
Ethernet5   MC14               0                0            0             0
Ethernet5   MC15               0                0            0             0
Ethernet5   MC16               0                0            0             0
Ethernet5   MC17               0                0            0             0
Ethernet5   MC18               0                0            0             0
Ethernet5   MC19               0                0            0             0
```

Note:

Total: 1,180,033 + 521,926 = 1,701,959
Queue 3: 1,180,033 / 1,701,959 = 0.6933
Queue 4: 521,926 / 1,701,959 = 0.3066

QoS Shaping
Queue shaping provides control of minimum and maximum bandwidth requirements per egress queue for more effective bandwidth utilization. (Note: There is no minimum bandwidth guarantee support for queue shaping on Broadcom and Intel switches.) Egress queues that exceed an average transmission rate beyond the shaper max bandwidth will stop being serviced. Additional ingress traffic will continue to be stored on the egress queue until the queue size is exceeded which results in tail drop.

Tested model & firmware version:
Switch model name:
DCS203 (AS7326-56X)
Edgecore SONiC version:
202111.1
202111.3
Topology:
mceclip4.png

Pre-configuration:
Add VLAN 10 to Ethernet0, Ethernet1, Ethernet5.

```
admin@sonic:~$ show vlan brief
+-----------+--------------+-----------+----------------+-------------
+---------------+---------------+-------------+
|   VLAN ID | IP Address   | Ports     | Port Tagging   | Proxy ARP   | DHCP
Helper   | DHCP Source   | DHCP Link   |
|           |              |           |                |             | Address
| Interface     | Selection   |
+===========+==============+===========+================+=============+=========
======+===============+=============+
|        10 |              | Ethernet0 | tagged         | disabled    |
|           |              |           |
|           |              | Ethernet1 | tagged         |             |
|           |              |           |
|           |              | Ethernet5 | tagged         |             |
|           |              |           |
+-----------+--------------+-----------+----------------+-------------
+---------------+---------------+-------------+
```

Procedure:
Step 1: Create the profile for QoS sharping.

```
admin@sonic:~$ sudo config scheduler add 400m_shaping --shaper_type bytes --
bandwidth 400m
```

Note:

```
admin@sonic:~$ sudo config scheduler add --help
Usage: config scheduler add [OPTIONS] <profile_name>

  Add QoS-Scheduler profile.

Options:
  --sched_type [WRR|DWRR|STRICT]  Scheduler type. WRR is not supported on
                                  Intel platform.
  --weight INTEGER RANGE          weight
```

```
--shaper_type [bytes|packets]   Shaper type
--bandwidth TEXT                Maximum bandwidth rate in kbps or pps. If
                                  shaper-type is bytes, user can specify a
                                  decimal number followed by the abbreviation
                                  k (1000), m (1,000,000), or g
                                  (1,000,000,000)
  -h, -?, --help                  Show this message and exit.
```

Step 2: Check the profile:

```
admin@sonic:~$ show scheduler
Name          Scheduling Type    Weight    Shaper Type    Bandwidth
------------  -----------------  --------  -------------  -----------
400m_shaping  N/A                N/A       bytes          400 Mbps
```

Step 3: Binding the QoS sharping to the egress interface.

```
admin@sonic:~$ sudo config interface scheduler bind queue Ethernet5 3
400m_shaping
```

Note:

```
Usage: config interface scheduler <bind/unbind> queue [OPTIONS] <interface_name>
<queue> <sched_policy_name>

Options:
  -?, -h, --help  Show this message and exit.
```

Or

```
Usage: config interface scheduler <bind/unbind> port [OPTIONS] <interface_name>
<sched_policy_name>

Options:
  -h, -?, --help  Show this message and exit.
```

Step 4: Check the binding table.

```
admin@sonic:~$ show interfaces scheduler
Ethernet5
Type      Queue  Profile       Scheduling Mode    Weight    Shaper Type
Bandwidth
------  -------  ------------  -----------------  --------  -------------
-----------
Queue         3  400m_shaping  N/A                N/A       bytes          400
Mbps
```

Result:

Since the QoS shaping, the traffic through to the Queue 3 will be down to about 400 Mbps.

mceclip5.png