63138/63148 L2-Switch QoS Configuration

Version 1.0

Table of Contents

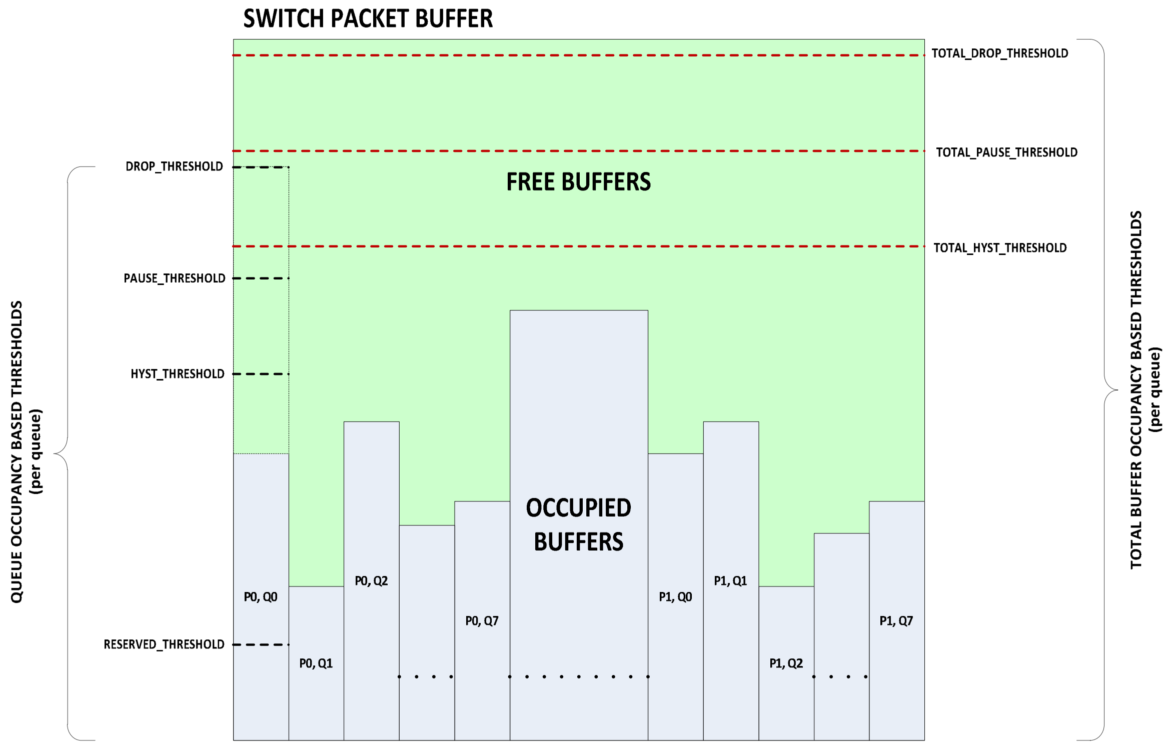
REVISION HISTORY

Revision Number	*Date*	*Change Description*
V1.0	03/31/2015	Initial version to describe TxQ threshold configuration. –Li Xu

This document contains information that is confidential and proprietary to Broadcom® Corporation (Broadcom) and may not be reproduced in any form without express written consent of Broadcom. No transfer or licensing of technology is implied by this document. Broadcom reserves the right to make changes without further notice to any products or data herein to improve reliability, function, or design. Information furnished by Broadcom is believed to be accurate and reliable. However, Broadcom does not assume any liability arising out of the application or use of this information, nor the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others.

1. INTRUDUCTION

In Runner Based DSL Router, each port has eight queues as below. The traffic from CPU is control by ACB flow control by default to guarantee no packet loss of WAN traffic. However, when LAN traffic and WAN traffic are all forwarded to the same queue and the same port causing congestion on that port, the LAN traffic will suppress the WAN traffic due to the ACB flow control well self regulated behavior by default. To maintain the WAN traffic in a higher privilege, eight queues in a port are divided into two groups for WAN and LAN. The WAN group has more resource assigned so it has higher privilege to pass through the port than LAN group.



2. SWITCH QUEUE PARTITIONING

By default, eight queues are partitioned into WAN and LAN groups. Even queue numbers are designed as LAN queues and odd number queues are designed as WAN queues.

The Ethernet driver will enforce this configuration by remapping the transmission even queue number into the next higher odd queue number. The current queue configuration can be checked by the quemap command:

sh> ethswctl -c quemap

WAN Queue Bit Map: 0xaa, Ethernet Driver Queue Remapping Map: 0x77553311

Success

WAN Queue Bit Map 0xaa is a bit map representing what queue numbers are used as WAN queue. 0xaa is 10101010 in binary meaning the queue 1,3,5 and 7 are designed as WAN queues. The rest queue numbers 0,2,4 and 6 are thus designed as LAN queues. Ethernet driver will configure each queue threshold properly based on this bit map and current link up ports number.

Ethernet Driver Queue Remapping Map values in output above is a queue map performed by Ethernet driver to remap all outgoing queues in transmission request to the new queues. For example, 0x77553311 will remap all even queue into odd queue number. This is the default remapping value in Ethernet driver, consistent with WAN Queue Bit Map configuration 0xaa which defines all odd queues as WAN queues. The definitions of WAN and LAN queues are only meaningful in terms of software. The difference is WAN queue will be configured with higher threshold values than LAN queues thus more resource available to guarantee lossless WAN to LAN traffic transmission.

Using the following command can see actual queue configured results.

```
# ethswctl -c swprioctrl
Current Thredshold Configuration Mode: Dyanmic
Total Ports         : 6
Link Up LAN Ports   : 1
Link Up WAN Ports   : 0
Max Stream Number   : 40
Global Flow Threshold Mode.
All Ports, Queue 0, Type:0-SwitchTxQHiReserveThreshold     : 0x0010 (16)
All Ports, Queue 0, Type:1-SwitchTxQHiHysteresisThreshold  : 0x01ad (429)
All Ports, Queue 0, Type:2-SwitchTxQHiPauseThreshold       : 0x01bd (445)
All Ports, Queue 0, Type:3-SwitchTxQHiDropThreshold        : 0x0211 (529)
All Ports, Queue 0, Type:4-SwitchTotalHysteresisThreshold  : 0x048e (1166)
All Ports, Queue 0, Type:5-SwitchTotalPauseThreshold       : 0x049d (1181)
All Ports, Queue 0, Type:6-SwitchTotalDropThreshold        : 0x0579 (1401)
--------------------------------------------------------------------------
All Ports, Queue 1, Type:0-SwitchTxQHiReserveThreshold     : 0x0010 (16)
All Ports, Queue 1, Type:1-SwitchTxQHiHysteresisThreshold  : 0x01c5 (453)
All Ports, Queue 1, Type:2-SwitchTxQHiPauseThreshold       : 0x01d5 (469)
All Ports, Queue 1, Type:3-SwitchTxQHiDropThreshold        : 0x022d (557)
All Ports, Queue 1, Type:4-SwitchTotalHysteresisThreshold  : 0x04c8 (1224)
All Ports, Queue 1, Type:5-SwitchTotalPauseThreshold       : 0x04d8 (1240)
All Ports, Queue 1, Type:6-SwitchTotalDropThreshold        : 0x05c1 (1473)
--------------------------------------------------------------------------
All Ports, Queue 2, Type:0-SwitchTxQHiReserveThreshold     : 0x0010 (16)
All Ports, Queue 2, Type:1-SwitchTxQHiHysteresisThreshold  : 0x01b2 (434)
All Ports, Queue 2, Type:2-SwitchTxQHiPauseThreshold       : 0x01c2 (450)
All Ports, Queue 2, Type:3-SwitchTxQHiDropThreshold        : 0x0217 (535)
All Ports, Queue 2, Type:4-SwitchTotalHysteresisThreshold  : 0x049a (1178)
All Ports, Queue 2, Type:5-SwitchTotalPauseThreshold       : 0x04a9 (1193)
All Ports, Queue 2, Type:6-SwitchTotalDropThreshold        : 0x0588 (1416)
--------------------------------------------------------------------------
All Ports, Queue 3, Type:0-SwitchTxQHiReserveThreshold     : 0x0010 (16)
All Ports, Queue 3, Type:1-SwitchTxQHiHysteresisThreshold  : 0x01ca (458)
All Ports, Queue 3, Type:2-SwitchTxQHiPauseThreshold       : 0x01da (474)
All Ports, Queue 3, Type:3-SwitchTxQHiDropThreshold        : 0x0233 (563)
All Ports, Queue 3, Type:4-SwitchTotalHysteresisThreshold  : 0x04d5 (1237)
All Ports, Queue 3, Type:5-SwitchTotalPauseThreshold       : 0x04e5 (1253)
All Ports, Queue 3, Type:6-SwitchTotalDropThreshold        : 0x05d0 (1488)
--------------------------------------------------------------------------
All Ports, Queue 4, Type:0-SwitchTxQHiReserveThreshold     : 0x0010 (16)
All Ports, Queue 4, Type:1-SwitchTxQHiHysteresisThreshold  : 0x01b7 (439)
All Ports, Queue 4, Type:2-SwitchTxQHiPauseThreshold       : 0x01c7 (455)
All Ports, Queue 4, Type:3-SwitchTxQHiDropThreshold        : 0x021d (541)
All Ports, Queue 4, Type:4-SwitchTotalHysteresisThreshold  : 0x04a6 (1190)
All Ports, Queue 4, Type:5-SwitchTotalPauseThreshold       : 0x04b5 (1205)
All Ports, Queue 4, Type:6-SwitchTotalDropThreshold        : 0x0597 (1431)
--------------------------------------------------------------------------
All Ports, Queue 5, Type:0-SwitchTxQHiReserveThreshold     : 0x0010 (16)
All Ports, Queue 5, Type:1-SwitchTxQHiHysteresisThreshold  : 0x01cf (463)
All Ports, Queue 5, Type:2-SwitchTxQHiPauseThreshold       : 0x01df (479)
All Ports, Queue 5, Type:3-SwitchTxQHiDropThreshold        : 0x0239 (569)
All Ports, Queue 5, Type:4-SwitchTotalHysteresisThreshold  : 0x04e2 (1250)
All Ports, Queue 5, Type:5-SwitchTotalPauseThreshold       : 0x04f2 (1266)
All Ports, Queue 5, Type:6-SwitchTotalDropThreshold        : 0x05df (1503)
--------------------------------------------------------------------------
All Ports, Queue 6, Type:0-SwitchTxQHiReserveThreshold     : 0x0010 (16)
All Ports, Queue 6, Type:1-SwitchTxQHiHysteresisThreshold  : 0x01bc (444)
All Ports, Queue 6, Type:2-SwitchTxQHiPauseThreshold       : 0x01cc (460)
All Ports, Queue 6, Type:3-SwitchTxQHiDropThreshold        : 0x0223 (547)
All Ports, Queue 6, Type:4-SwitchTotalHysteresisThreshold  : 0x04b2 (1202)
All Ports, Queue 6, Type:5-SwitchTotalPauseThreshold       : 0x04c2 (1218)
All Ports, Queue 6, Type:6-SwitchTotalDropThreshold        : 0x05a6 (1446)
--------------------------------------------------------------------------
All Ports, Queue 7, Type:0-SwitchTxQHiReserveThreshold     : 0x0010 (16)
All Ports, Queue 7, Type:1-SwitchTxQHiHysteresisThreshold  : 0x01d4 (468)
All Ports, Queue 7, Type:2-SwitchTxQHiPauseThreshold       : 0x01e4 (484)
All Ports, Queue 7, Type:3-SwitchTxQHiDropThreshold        : 0x023f (575)
All Ports, Queue 7, Type:4-SwitchTotalHysteresisThreshold  : 0x04ef (1263)
All Ports, Queue 7, Type:5-SwitchTotalPauseThreshold       : 0x04ff (1279)
All Ports, Queue 7, Type:6-SwitchTotalDropThreshold        : 0x05ef (1519)
==========================================================================
IMP0, Queue 0, Type:0-SwitchTxQHiReserveThreshold     : 0x0010 (16)
```

```
IMP0, Queue 0, Type:1-SwitchTxQHiHysteresisThreshold  : 0x01b1 (433)
IMP0, Queue 0, Type:2-SwitchTxQHiPauseThreshold       : 0x01c1 (449)
IMP0, Queue 0, Type:3-SwitchTxQHiDropThreshold        : 0x0215 (533)
IMP0, Queue 0, Type:4-SwitchTotalHysteresisThreshold  : 0x0497 (1175)
IMP0, Queue 0, Type:5-SwitchTotalPauseThreshold       : 0x04a5 (1189)
IMP0, Queue 0, Type:6-SwitchTotalDropThreshold        : 0x0585 (1413)
--------------------------------------------------------------------------
IMP0, Queue 1, Type:0-SwitchTxQHiReserveThreshold     : 0x0010 (16)
IMP0, Queue 1, Type:1-SwitchTxQHiHysteresisThreshold  : 0x01b6 (438)
IMP0, Queue 1, Type:2-SwitchTxQHiPauseThreshold       : 0x01c6 (454)
IMP0, Queue 1, Type:3-SwitchTxQHiDropThreshold        : 0x021b (539)
IMP0, Queue 1, Type:4-SwitchTotalHysteresisThreshold  : 0x04a3 (1187)
IMP0, Queue 1, Type:5-SwitchTotalPauseThreshold       : 0x04b1 (1201)
IMP0, Queue 1, Type:6-SwitchTotalDropThreshold        : 0x0594 (1428)
--------------------------------------------------------------------------
IMP0, Queue 2, Type:0-SwitchTxQHiReserveThreshold     : 0x0010 (16)
IMP0, Queue 2, Type:1-SwitchTxQHiHysteresisThreshold  : 0x01bb (443)
IMP0, Queue 2, Type:2-SwitchTxQHiPauseThreshold       : 0x01cb (459)
IMP0, Queue 2, Type:3-SwitchTxQHiDropThreshold        : 0x0221 (545)
IMP0, Queue 2, Type:4-SwitchTotalHysteresisThreshold  : 0x04af (1199)
IMP0, Queue 2, Type:5-SwitchTotalPauseThreshold       : 0x04be (1214)
IMP0, Queue 2, Type:6-SwitchTotalDropThreshold        : 0x05a3 (1443)
--------------------------------------------------------------------------
IMP0, Queue 3, Type:0-SwitchTxQHiReserveThreshold     : 0x0010 (16)
IMP0, Queue 3, Type:1-SwitchTxQHiHysteresisThreshold  : 0x01c0 (448)
IMP0, Queue 3, Type:2-SwitchTxQHiPauseThreshold       : 0x01d0 (464)
IMP0, Queue 3, Type:3-SwitchTxQHiDropThreshold        : 0x0227 (551)
IMP0, Queue 3, Type:4-SwitchTotalHysteresisThreshold  : 0x04bb (1211)
IMP0, Queue 3, Type:5-SwitchTotalPauseThreshold       : 0x04cb (1227)
IMP0, Queue 3, Type:6-SwitchTotalDropThreshold        : 0x05b2 (1458)
--------------------------------------------------------------------------
IMP0, Queue 4, Type:0-SwitchTxQHiReserveThreshold     : 0x0010 (16)
IMP0, Queue 4, Type:1-SwitchTxQHiHysteresisThreshold  : 0x01c5 (453)
IMP0, Queue 4, Type:2-SwitchTxQHiPauseThreshold       : 0x01d5 (469)
IMP0, Queue 4, Type:3-SwitchTxQHiDropThreshold        : 0x022d (557)
IMP0, Queue 4, Type:4-SwitchTotalHysteresisThreshold  : 0x04c8 (1224)
IMP0, Queue 4, Type:5-SwitchTotalPauseThreshold       : 0x04d8 (1240)
IMP0, Queue 4, Type:6-SwitchTotalDropThreshold        : 0x05c1 (1473)
--------------------------------------------------------------------------
IMP0, Queue 5, Type:0-SwitchTxQHiReserveThreshold     : 0x0010 (16)
IMP0, Queue 5, Type:1-SwitchTxQHiHysteresisThreshold  : 0x01ca (458)
IMP0, Queue 5, Type:2-SwitchTxQHiPauseThreshold       : 0x01da (474)
IMP0, Queue 5, Type:3-SwitchTxQHiDropThreshold        : 0x0233 (563)
IMP0, Queue 5, Type:4-SwitchTotalHysteresisThreshold  : 0x04d5 (1237)
IMP0, Queue 5, Type:5-SwitchTotalPauseThreshold       : 0x04e5 (1253)
IMP0, Queue 5, Type:6-SwitchTotalDropThreshold        : 0x05d0 (1488)
--------------------------------------------------------------------------
IMP0, Queue 6, Type:0-SwitchTxQHiReserveThreshold     : 0x0010 (16)
IMP0, Queue 6, Type:1-SwitchTxQHiHysteresisThreshold  : 0x01cf (463)
IMP0, Queue 6, Type:2-SwitchTxQHiPauseThreshold       : 0x01df (479)
IMP0, Queue 6, Type:3-SwitchTxQHiDropThreshold        : 0x0239 (569)
IMP0, Queue 6, Type:4-SwitchTotalHysteresisThreshold  : 0x04e2 (1250)
IMP0, Queue 6, Type:5-SwitchTotalPauseThreshold       : 0x04f2 (1266)
IMP0, Queue 6, Type:6-SwitchTotalDropThreshold        : 0x05df (1503)
```

```
--------------------------------------------------------------------------
IMP0, Queue 7, Type:0-SwitchTxQHiReserveThreshold     : 0x0010 (16)
IMP0, Queue 7, Type:1-SwitchTxQHiHysteresisThreshold  : 0x01d4 (468)
IMP0, Queue 7, Type:2-SwitchTxQHiPauseThreshold       : 0x01e4 (484)
IMP0, Queue 7, Type:3-SwitchTxQHiDropThreshold        : 0x023f (575)
IMP0, Queue 7, Type:4-SwitchTotalHysteresisThreshold  : 0x04ef (1263)
IMP0, Queue 7, Type:5-SwitchTotalPauseThreshold       : 0x04ff (1279)
IMP0, Queue 7, Type:6-SwitchTotalDropThreshold        : 0x05ef (1519)
==========================================================================
Success
#
```

Note in examples above, all odd queues in switch LAN ports are configured with higher threshold values than even queues.

Using the following commands can check current port congestion status:

```
# ethswctl -c quemon -p 1
==========================================================================
Type:2-SysTotalPeakCount               : 0x0018 (24)
Type:3-SysTotalCurCount                : 0x0010 (16)
==========================================================================
Port 1:
 Type:4-PortPeakRxBuffer                : 0x000a (10)
 Type:6-PortPauseHistory                : No
 Type:7-PortQuantumPauseHistory         : No
 Type:8-PortRxBasedPauseHistory         : No
 Type:9-PortRxBufferErrorHitory         : No
--------------------------------------------------------------------------
Queue 0, Type: 0-QueCurCount                 : 0x0:(0)
Queue 0, Type: 1-QuePeakCount                : 0x2:(2)
Queue 0, Type: 5-QueFinalCongestedStatus     : 0x0:(0) - Below HW Reserved
Queue 0, Type:10-QueCongestedStatus          : 0x0:(0) - Below HW Reserved
Queue 0, Type:11-QueTotalCongestedStatus     : 0x0:(0) - Below HW Reserved
--------------------------------------------------------------------------
Queue 1, Type: 0-QueCurCount                 : 0x0:(0)
Queue 1, Type: 1-QuePeakCount                : 0x0:(0)
Queue 1, Type: 5-QueFinalCongestedStatus     : 0x0:(0) - Below HW Reserved
Queue 1, Type:10-QueCongestedStatus          : 0x0:(0) - Below HW Reserved
Queue 1, Type:11-QueTotalCongestedStatus     : 0x0:(0) - Below HW Reserved
--------------------------------------------------------------------------
Queue 2, Type: 0-QueCurCount                 : 0x0:(0)
Queue 2, Type: 1-QuePeakCount                : 0x0:(0)
Queue 2, Type: 5-QueFinalCongestedStatus     : 0x0:(0) - Below HW Reserved
Queue 2, Type:10-QueCongestedStatus          : 0x0:(0) - Below HW Reserved
Queue 2, Type:11-QueTotalCongestedStatus     : 0x0:(0) - Below HW Reserved
--------------------------------------------------------------------------
Queue 3, Type: 0-QueCurCount                 : 0x0:(0)
Queue 3, Type: 1-QuePeakCount                : 0x0:(0)
Queue 3, Type: 5-QueFinalCongestedStatus     : 0x0:(0) - Below HW Reserved
Queue 3, Type:10-QueCongestedStatus          : 0x0:(0) - Below HW Reserved
Queue 3, Type:11-QueTotalCongestedStatus     : 0x0:(0) - Below HW Reserved
--------------------------------------------------------------------------
Queue 4, Type: 0-QueCurCount                 : 0x0:(0)
Queue 4, Type: 1-QuePeakCount                : 0x0:(0)
Queue 4, Type: 5-QueFinalCongestedStatus     : 0x0:(0) - Below HW Reserved
Queue 4, Type:10-QueCongestedStatus          : 0x0:(0) - Below HW Reserved
Queue 4, Type:11-QueTotalCongestedStatus     : 0x0:(0) - Below HW Reserved
--------------------------------------------------------------------------
Queue 5, Type: 0-QueCurCount                 : 0x0:(0)
Queue 5, Type: 1-QuePeakCount                : 0x0:(0)
Queue 5, Type: 5-QueFinalCongestedStatus     : 0x0:(0) - Below HW Reserved
Queue 5, Type:10-QueCongestedStatus          : 0x0:(0) - Below HW Reserved
Queue 5, Type:11-QueTotalCongestedStatus     : 0x0:(0) - Below HW Reserved
--------------------------------------------------------------------------
Queue 6, Type: 0-QueCurCount                 : 0x0:(0)
Queue 6, Type: 1-QuePeakCount                : 0x0:(0)
Queue 6, Type: 5-QueFinalCongestedStatus     : 0x0:(0) - Below HW Reserved
Queue 6, Type:10-QueCongestedStatus          : 0x0:(0) - Below HW Reserved
Queue 6, Type:11-QueTotalCongestedStatus     : 0x0:(0) - Below HW Reserved
--------------------------------------------------------------------------
Queue 7, Type: 0-QueCurCount                 : 0x0:(0)
Queue 7, Type: 1-QuePeakCount                : 0x0:(0)
Queue 7, Type: 5-QueFinalCongestedStatus     : 0x0:(0) - Below HW Reserved
Queue 7, Type:10-QueCongestedStatus          : 0x0:(0) - Below HW Reserved
Queue 7, Type:11-QueTotalCongestedStatus     : 0x0:(0) - Below HW Reserved
Success
#
```

Using the following commands can check the actual Ethernet driver queue remapping statistics:

```
# ethctl eth1 stats

TxPkts:            72896
TxOctets:        9619200
TxDropPkts:            0

RxPkts:            89642
RxOctets:       11474176
RxDropPkts:            0

Device eth1:
  rx_packets_queue[0]                  :          0;   rx_packets_queue[1]                 :        0
  rx_packets_queue[2]                  :          0;   rx_packets_queue[3]                 :    89642
  rx_packets_queue[4]                  :          0;   rx_packets_queue[5]                 :        0
  rx_packets_queue[6]                  :          0;   rx_packets_queue[7]                 :        0
  tx_packets_queue_in[0]               :      63971;   tx_packets_queue_in[1]              :        0
  tx_packets_queue_in[2]               :          0;   tx_packets_queue_in[3]              :        0
  tx_packets_queue_in[4]               :          0;   tx_packets_queue_in[5]              :        0
  tx_packets_queue_in[6]               :       8923;   tx_packets_queue_in[7]              :        2
  tx_packets_queue_out[0]              :          0;   tx_packets_queue_out[1]             :    63971
  tx_packets_queue_out[2]              :          0;   tx_packets_queue_out[3]             :        0
  tx_packets_queue_out[4]              :          0;   tx_packets_queue_out[5]             :        0
  tx_packets_queue_out[6]              :          0;   tx_packets_queue_out[7]             :     8925
  rx_dropped_no_rxdev                  :          0;   rx_dropped_blog_drop                :        0
  rx_dropped_no_skb                    :          0;   rx_packets_blog_done                :    58283
  rx_dropped_skb_headinit              :          0;   rx_packets_netif_receive_skb        :    31359
  rx_errors_indicated_by_low_level     :          0;   rx_dropped_undersize                :        0
  rx_dropped_overrate                  :          0;   tx_enet_xmit                        :    72896
  tx_dropped_bad_nbuff                 :          0;   tx_enet_xmit_extra_chained_nbuff    :        0
  tx_dropped_no_lowlvl_resource        :          0;   tx_dropped_no_fkb                   :        0
  tx_dropped_no_skb                    :          0;   tx_dropped_no_gem_ids               :        0
  tx_dropped_bad_gem_id                :          0;   tx_dropped_misaligned_nbuff         :        0
  tx_drops_no_valid_gem_fun            :          0;   tx_drops_skb_linearize_error        :        0
  tx_dropped_runner_lan_fail           :          0;   tx_dropped_runner_wan_fail          :        0
  tx_dropped_no_gso_dsc                :          0;   tx_dropped_sid_tx_fail              :        0
  tx_dropped_no_rdpa_port_mapped       :          0;   tx_dropped_no_gem_tcount            :        0
  tx_dropped_no_epon_tx_fun            :          0;   tx_dropped_no_epon_oam_fun          :        0
  tx_dropped_gpon_tx_fail              :          0;   tx_dropped_epon_tx_fail             :        0
  tx_dropped_epon_oam_fail             :          0;   tx_dropped_xpon_lan_fail            :        0
  portCnt.rxIn                            :       89642;   portCnt.rx2Kernel                        :      31359
  portCnt.rx2Blog                         :       58283;   portCnt.rx2Kernel + portCnt.rx2Blog      :      89642
  portCnt.rxDrops                         :           0
  (portCnt.rx2Kernel + portCnt.rx2Blog + portCnt.rxDrops):      89642
  portCnt.rxIn == (portCnt.rx2Kernel + portCnt.rx2Blog + portCnt.rxDrops):       True
  portCnt.txIn                            :       72896;   portCnt.txOut                            :      72896
  portCnt.txDrops                         :          0;   portCnt.txExtraChain                     :          0
  (portCnt.txOut + portCnt.txDrops - portCnt.txExtraChain):      72896
  portCnt.txIn == (portCnt.txOut + portCnt.txDrops - portCnt.txExtraChain):       True

#
```

3. CUSTOMIZING QUEUE CONFIGURATION FROM DEFAULT VALUES

User can use following parameters to further customize switch output queue threshold configurations.

3.1 Queue Configuration Mode:

Ethswctl –c swprioctrl –m <mode>

will configure the queue configuration to Dynamic, Static or Manual mode. When queue threshold configuration is in manual mode, the same swprictrl command can be used to configure each threshold manually. We don't recommend user do configuration manually. When the Queue Configuration Mode is in Dynamic mode, each queue thresholds will be configured by Ethernet driver based on WAN queue assignment and link up port number. Under automatic mode, the Threshold queue was carefully calculated to reach the following goals:

- Each queue has a minimum guaranteed buffer to pass through traffic.
- Two or more congestion streams can be passed in proper ratio.
- WAN queues have more resource assigned than LAN queues.
- Higher priority queues have more resource assigned than lower priority queues.
- Buffer can be utilized efficiently.

When queue mode is configured as Static mode, each queue thresholds will be configured by Ethernet driver based on WAN queue assignment and total populated port number. The difference of Static mode from Dynamic mode is Dynamic mode will reconfigure thresholds with port link up and down activities while Static mode will not.

Under Dynamic or Static configuration mode, the following parameters can be configured furthermore based on application environment.

3.2 WAN Queue Map Configuration:

Etheswctl –c quemap –v <WAN queue bit map>

will configure the queue numbers assigned for WAN traffic. The queue thresholds will be reconfigured based on the new WAN queue assignments. By default, all odd number queues are assigned as WAN queues.

For example:

sh> ethswctl –c quemap –v 0xaa

will configure all odd number queues to be WAN queues, which is default values.

sh> ethswctl –c quemap –v 0x55

will configure all even number queues to be WAN queues.

3.3 WAN Traffic Outgoing Queue Remapping

To insure WAN traffic through the CPU host always go through the designed WAN queues, the queue mapping policy should be consistent across the whole system level. This can be achieved by traffic priority remap on kernel level through iptables/ebtabls chains, Ethernet driver provided queue remapping mechanism, or a combination of both mechanisms.

3.3.1 Kernel Level Queue Mapping Control

On kernel level, when WAN port is configured as routing with LAN ports, user can use iptables to mark traffic from CPU to LAN ports to specific queues. For example:

sh> iptables -t mangle -A FORWARD -o br0 -j MARK --set-mark 5

Will mark WAN to LAN traffic to send out in queue 5.

When WAN port is configured in bridging mode with LAN ports, user can use ebtables to mark traffic from CPU to LAN ports to specific queues. For example:

sh> ebtables -t filter -A FORWARD -o ! eth0 -j mark --mark-set 3

Will mark WAN to LAN traffic to send out in queue 3.

Note: when testing the iptables and ebtables, use *sh> fc disable* to disable flow cache, or you will need to do *sh> fc flush* every time when you change iptables or ebtables configurations. User can also configure this through Broadcom GUI, or call Broadcom CMS configuration APIs from customer's GUI operations, which in turn will update flow cache with new configurations automatically.

3.3.2 Ethernet Driver Queue Remapping

On Ethernet driver level, command

sh> ethswctl –c quemap -q <Ethernet Driver Queue Remapping Map>

will configure LAN outgoing traffic remapping.

sh> ethswctl –c quemap –q 0x77553311

will remap all even number queues into odd number queues. This is default queue mapping configuration and consistent with default WAN queue map configuration value 0xaa which configures all odd number queues as WAN traffic queues.

The WAN queue map and queue remapping need to maintain consistent or the configuration will be rejected. The remapped WAN queue numbers specified by –q need to be one of the WAN queues specified by -v.

For example:

sh> ethswctl –c quemap –v 0xaa –q 0x77553311

is valid configuration since queue 7,5,3,1 matches WAN queue number specified in –v 0xaa(7,5,3 and 1 as WAN queues) bit map. Where

sh> ethswctl –c quemap –v 0x2 –q 0x22222222

is invalid configuration since –v specifies queue 1 as WAN queue, while -q specifies mapping all queues into queue 2 which is not WAN queue.

This consistency checked will also be performed between the new configuration value and current existing value if not –v and –q are specified at the same time.

User can also use combination of these two ways to achieve sophistic queue mapping control across the whole system.