MT7986 HW QoS Application Note

2022/01/20

Version History

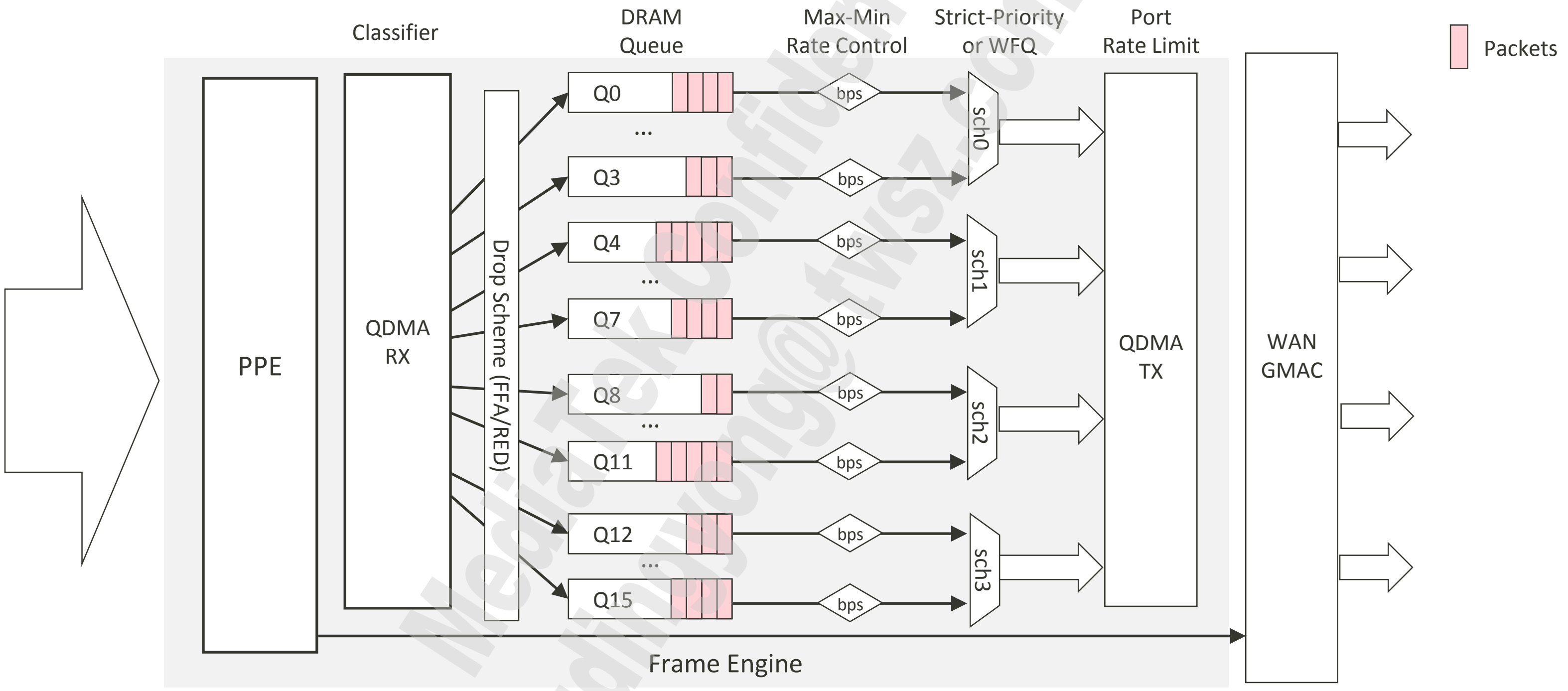
Version	Date	Author (Optional)	Description
0.1	2021-04-20	Henry Yen	Initial draft
1.0	2022-01-20	Michael Shih	Official release

Outline

- Hardware-QoS (HQoS)
- HQoS Rate Limit Example
- QDMA Debug Command

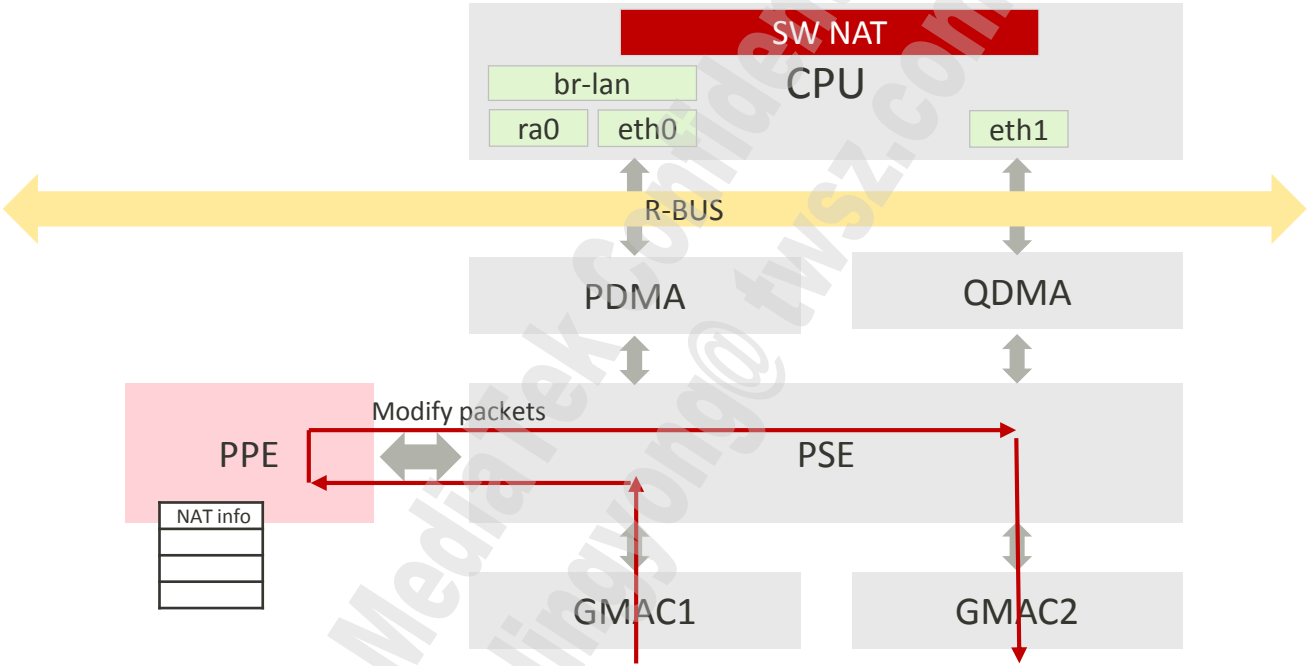
Hardware-QoS (HQoS)

HQoS Architecture

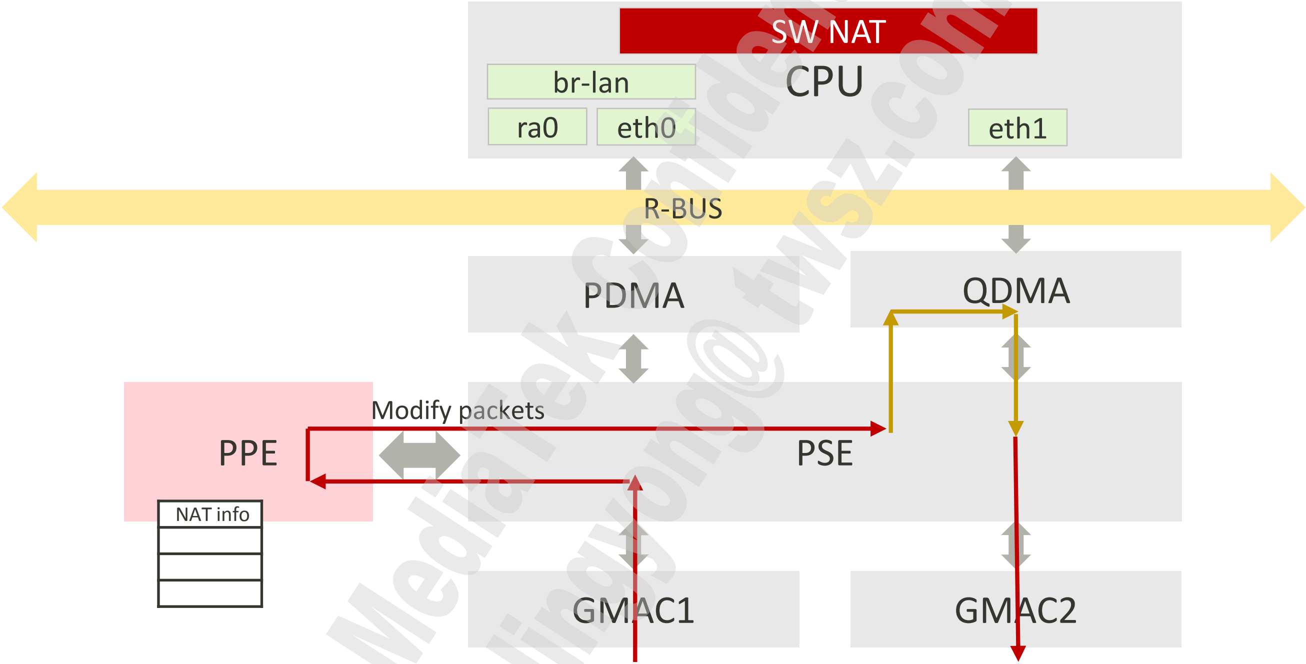


The mapping of packet-to-queue and queue-to-scheduler are configurable.

HNAT Enabled : LAN → WAN



HNAT + HQoS Enabled : LAN -> WAN



entry->ipv4_hnapt.iblk2.**fqos=1** indicates enabling QoS scheduling after BINDING
entry->ipv4_hnapt.iblk2.**qid=x** indicates which QoS queue to be in

Panther HQoS Specification

- **128 physical queues**
- **4 schedulers**
- **Queue scheduling policy**
  - **Strict-Priority (SP)**
    - For the queue with higher priority, its min rate is satisfied first.
    - For the queue with lower priority, its min rate is satisfied later.
    - Queue number as the priority, e.g. Queue 0 > Queue 1 > Queue 2…
  - **Weighted Round-Robin (WRR)**
    - After the min rate of each queue is satisfied, the remaining resource is arranged according to the weighting while not exceed the max rate of each queue.
    - Weight as the priority.

HQoS Codebase

- **Directory**
  - kernel/drivers/net/ethernet/mediatek/mtk_hnat/

- **Dependency between Ethernet & HQoS**
  - Search "**CONFIG_NET_MEDIATEK_HW_QOS**"
  - At eth Tx path: fill skb->mark into TXD

```
#if defined(CONFIG_NET_MEDIATEK_HW_QOS)
        qid = skb->mark & (MTK_QDMA_TX_MASK);
#endif
```

Ways to Enable HQoS Feature (1)

- **make kernel_menuconfig → enable NET_MEDIATEK_HW_QOS**

```
Symbol: NET_MEDIATEK_HW_QOS [=y]
Type  : bool
Prompt: Mediatek HW QoS support
  Location:
    -> Device Drivers
      -> Network device support (NETDEVICES [=y])
        -> Ethernet driver support (ETHERNET [=y])
          -> MediaTek ethernet driver (NET_VENDOR_MEDIATEK [=y])
            -> MediaTek SoC Gigabit Ethernet support (NET_MEDIATEK_SOC [=y])
(1)           -> MediaTek HW NAT support (NET_MEDIATEK_HNAT [=m])
  Defined at drivers/net/ethernet/mediatek/Kconfig:31
  Depends on: NETDEVICES [=y] && ETHERNET [=y] && NET_VENDOR_MEDIATEK [=y] && NET_MEDIATEK_HNAT [=m]
```

Ways to Enable HQoS Feature (2)

- **make menuconfig → enable iptables & iptables-mod-ipopt**

```
Symbol: DEFAULT_iptables [=y]
Type  : bool
Defined at tmp/.config-target.in:194818
Selected by [y]:
  - TARGET_DEVICE_mediatek_mt7986_DEVICE_mediatek_mt7986-evb [=y] && TARGET_MULTI_PROFILE [=y]
  - TARGET_DEVICE_mediatek_mt7986_DEVICE_mediatek_mt7986-fpga [=y] && TARGET_MULTI_PROFILE [=y
  - TARGET_DEVICE_mediatek_mt7986_DEVICE_mediatek_mt7986-fpga-sb [=y] && TARGET_MULTI_PROFILE
Selected by [n]:
  - DEFAULT_TARGET_sunxi_cortexa8 [=n] && TARGET_PER_DEVICE_ROOTFS [=n]
```

```
Symbol: PACKAGE_iptables-mod-ipopt [=y]
Type  : tristate
Defined at tmp/.config-package.in:105262
  Prompt: iptables-mod-ipopt......................... IP/Packet option extensions
  Depends on: PACKAGE_iptables [=y]
  Location:
    -> Network
(1)   -> Firewall
Selects: PACKAGE_libc [=y] && PACKAGE_libpthread [=y] && PACKAGE_kmod-ipt-ipopt [=y] && PACKAGE_librt [=n]
Selected by [n]:
  - PACKAGE_qos-scripts [=n]
  - PACKAGE_sqm-scripts [=n]
```

Ways to Enable HQoS Feature (3)

- **make menuconfig → enable ebtables & ebtables-ipv4**

```
Symbol: PACKAGE_kmod-ebtables [=y]
Type  : tristate
Defined at tmp/.config-package.in:13261
  Prompt: kmod-ebtables.............................. Bridge firewalling modules
  Location:
    -> Kernel modules
(3)   -> Netfilter Extensions
Selects: PACKAGE_kmod-ipt-core [=y]
Selected by [y]:
  - PACKAGE_kmod-ebtables-ipv4 [=y]
  - PACKAGE_ebtables [=y]
```

```
Symbol: PACKAGE_kmod-ebtables-ipv4 [=y]
Type  : tristate
Defined at tmp/.config-package.in:13272
  Prompt: kmod-ebtables-ipv4............................. ebtables: IPv4 support
  Location:
    -> Kernel modules
(4)   -> Netfilter Extensions
Selects: PACKAGE_kmod-ebtables [=y]
Selected by [n]:
  - PACKAGE_phantap [=n]
```

Ways to Enable HQoS Feature (4)

- **Enable HQoS**
  - **Enable HQoS :**
    CMD> echo 1 > /sys/kernel/debug/hnat/qos_toggle
  - **Disable HQoS :**
    CMD> echo 0 > /sys/kernel/debug/hnat/qos_toggle
- **Configure HQoS**
  - **Edit /etc/config/mtkhnat**
  - **Active Settings:**
    CMD> sh /sbin/mtkhnat
- **Configure Queueing Policy**
  - **Use iptables command to mark skb**
  - **For example: iptables -t mangle -A FORWARD -d 192.168.1.6 -j MARK --set-mark 1**

HQoS Configurations: Rate Limit

- **/etc/config/mtkhnat.config**

```
config global global
        option enable 1
        option hqos 1
        option txq_num 16
        option scheduling 'wrr'
        option sch0_bw 1000000
        option sch1_bw 1000000

config queue
        option id 0
        option minrate 30
        option maxrate 100
        option weight 4
        option resv 4

config queue
        option id 1
        option minrate 30
        option maxrate 100
        option weight 2
        option resv 4
```

- HW QoS ON/OFF (1:ON, 0:OFF)
- txq_num (128 only for Panther)
- Scheduling policy ('wrr' or 'sp')
- sch0 rate limit (unit: Kbps)
- sch1 rate limit (unit: Kbps)
- Queue Id
- Percentage of min rate limit
- Percentage of max rate limit
- Weight for queue schedule
- Buffer reserved for HW/SW path

The weighted value for TX queue
0: Weight value = 16
1: Weight value = 1
2: Weight value = 2
n: Weight value = n
15: Weight value = 15

HQoS Configurations: SCH-Q Mapping

- /sbin/mtkhnat
- For example: change Queue_id=1 mapping to Scheduler=1
  - CMD> echo ${sch_ebl} ${sch_policy} ${sch_bw} > /sys/kernel/debug/hnat/qdma_sch1
    - sch_ebl : whether or not enable scheduler
    - sch_policy : decide scheduling policy among “sp” or “wrr”
    - sch_bw : scheduler bandwidth
  - CMD> echo 1 ${queue_minebl} ${minrate} ${queue_maxebl} ${maxrate} ${queue_weight} ${queue_resv} > /sys/kernel/debug/hnat/qdma_txq1
    - queue_minebl : whether or not enable min rate limit function
    - queue_maxebl : whether or not enable max rate limit function

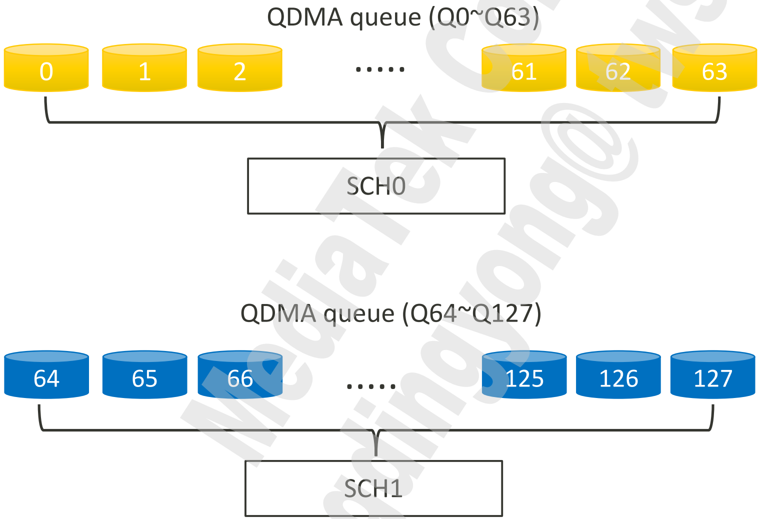
```
# set the queue of sch0 group(the lower half of total queues)
[ "${queue_id}" -le $(((txq_num / 2) - 1)) ] && \
echo 0 ${queue_minebl} ${minrate} ${queue_maxebl} ${maxrate} ${queue_weight} \
     ${queue_resv} > /sys/kernel/debug/hnat/qdma_txq${queue_id}
```

Scheduler ID

Queue ID

Queue Selection via skb->mark

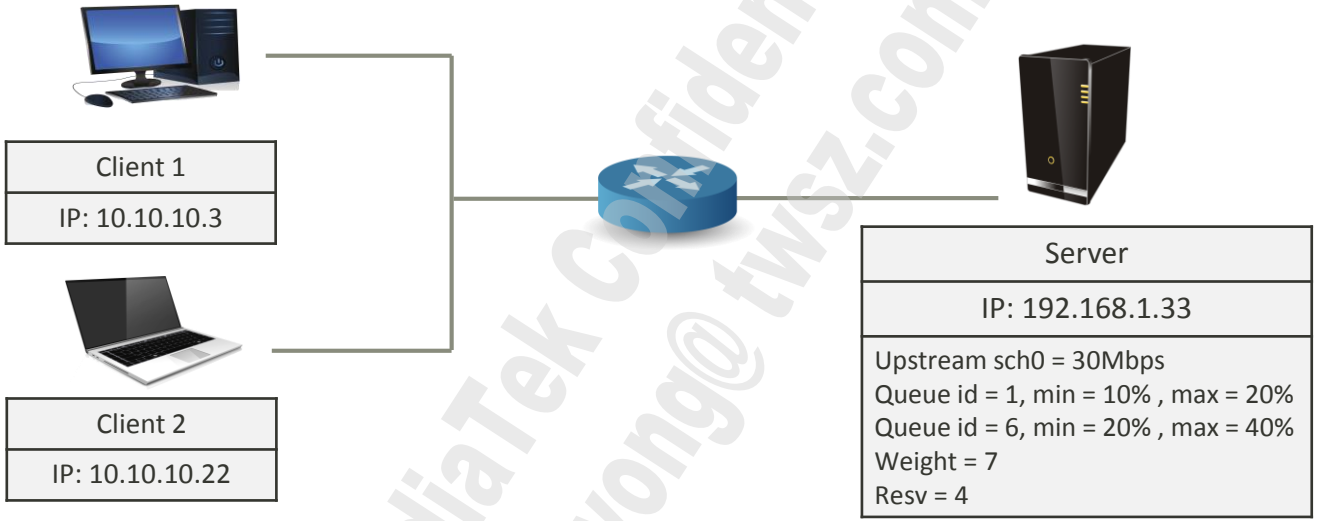
- **Customers can use iptables to direct packets into specified queue.**
  - entry->ipv4_hnapt.iblk2.qid = **skb->mark**
  - iptables -t mangle -A FORWARD -p tcp -j MARK **--set-mark x**



skb->mark	Queue
0	0
1	1
2	2
3	3
4	4
5	5
6	6
⋮	
111	111
122	122
123	123
124	124
125	125
126	126
127	127

HQoS Rate Limit Example

HQoS Rate Limit Example (1)



skb->mark settings
iptables -t mangle -A FORWARD -p tcp -j MARK **--set-mark 1** // for eth->eth iptables -t mangle -A FORWARD -p udp -j MARK **--set-mark 6** ebtables -t broute -A BROUTING -p ipv4 --ip-src 10.10.10.3 -j mark **--mark-set 1** // for wlan->eth ebtables -t broute -A BROUTING -p ipv4 --ip-src 10.10.10.22 -j mark **--mark-set 6**

HQoS Rate Limit Example (2)

Upstream Limit * Max Rate = 30 * 0.2 = **6 Mbps**

Client 1
Queue id = 1

```
[  3] local 10.10.10.3 port 63698 connected with 192.168.1.33 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  6.88 MBytes  5.77 Mbits/sec
[  3] 10.0-20.0 sec  6.75 MBytes  5.66 Mbits/sec
[  3] 20.0-30.0 sec  6.75 MBytes  5.66 Mbits/sec
[  3] 30.0-40.0 sec  6.75 MBytes  5.66 Mbits/sec
[  3] 40.0-50.0 sec  6.88 MBytes  5.77 Mbits/sec
[  3] 50.0-60.0 sec  6.75 MBytes  5.66 Mbits/sec
[  3] 60.0-70.0 sec  6.75 MBytes  5.66 Mbits/sec
[  3] 70.0-80.0 sec  6.88 MBytes  5.77 Mbits/sec
[  3] 80.0-90.0 sec  6.75 MBytes  5.66 Mbits/sec
[  3] 90.0-100.0 sec  6.75 MBytes  5.66 Mbits/sec
[  3] 100.0-110.0 sec  6.75 MBytes  5.66 Mbits/sec
[  3] 110.0-120.0 sec  6.88 MBytes  5.77 Mbits/sec
```

Upstream Limit * Max Rate = 30 * 0.4 = **12 Mbps**

Client 2
Queue id = 6

```
[  3] local 10.10.10.22 port 53766 connected with 192.168.1.33 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  13.6 MBytes  11.4 Mbits/sec
[  3] 10.0-20.0 sec  13.6 MBytes  11.4 Mbits/sec
[  3] 20.0-30.0 sec  13.6 MBytes  11.4 Mbits/sec
[  3] 30.0-40.0 sec  13.5 MBytes  11.3 Mbits/sec
[  3] 40.0-50.0 sec  13.6 MBytes  11.4 Mbits/sec
[  3] 50.0-60.0 sec  13.6 MBytes  11.4 Mbits/sec
[  3] 60.0-70.0 sec  13.5 MBytes  11.3 Mbits/sec
[  3] 70.0-80.0 sec  13.5 MBytes  11.3 Mbits/sec
[  3] 80.0-90.0 sec  13.6 MBytes  11.4 Mbits/sec
[  3] 90.0-100.0 sec  13.6 MBytes  11.4 Mbits/sec
[  3] 100.0-110.0 sec  13.6 MBytes  11.4 Mbits/sec
[  3] 110.0-120.0 sec  13.5 MBytes  11.3 Mbits/sec
```

QDMA Debug Command

QDMA Debug Command (1)

- **Description:**
  - show specific QDMA scheduler info
- **Command format:**
  - *cat /sys/kernel/debug/hnat/qdma_sch[0~1]*
- **Example: (show QDMA sch0 info)**
  - cat /sys/kernel/debug/hnat/qdma_sch0

```
root@LEDE:/# cat /sys/kernel/debug/hnat/qdma_sch0
EN     Scheduling      MAX     Queue#
1      WRR             30000   0  1  2  3  4  5  6  7
root@LEDE:/# cat /sys/kernel/debug/hnat/qdma_sch1
EN     Scheduling      MAX     Queue#
1      WRR             30000   8  9  10  11  12  13  14  15
```

EN: Rate limit enable or not
Scheduling:
WRR: Use Weighted Round-Robin strategy to select the physical queue for the rate between MN and MAX rate.
SP: Use Strict Priority strategy to Select the physical queue for the rate between MN and MAX rate.
MAX: Max rate limit for sch0 or sch1
Queue#: QDMA queue in the scheduler

QDMA Debug Command (2)

- **Description:**
  - show specific QDMA queue info
- **Command format:**
  - *cat /sys/kernel/debug/hnat/qdma_txq[0~15]*
- **Example: (show QDMA txq6 info)**
  - cat /sys/kernel/debug/hnat/qdma_txq6

```
root@LEDE:/# cat /sys/kernel/debug/hnat/qdma_txq6
scheduler: 0
hw resv: 4
sw resv: 4
packet count: 0
packet drop: 0

      EN     RATE     WEIGHT
-------------------------
max    1    30000        7
min    1     7500
```