Texas Instruments CPSW ethernet driver

Multiqueue & CBS & MQPRIO

The cpsw has 3 CBS shapers for each external ports. This document describes MQPRIO and CBS Qdisc offload configuration for cpsw driver based on examples. It potentially can be used in audio video bridging (AVB) and time sensitive networking (TSN).

The following examples were tested on AM572x EVM and BBB boards.

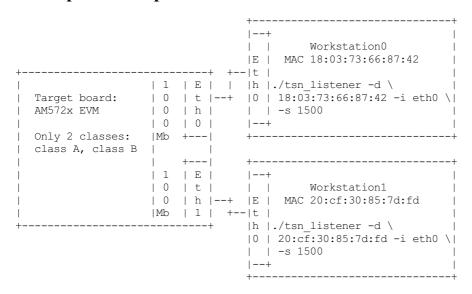
Test setup

Under consideration two examples with AM572x EVM running cpsw driver in dual emac mode.

Several prerequisites:

- TX queues must be rated starting from txq0 that has highest priority
- Traffic classes are used starting from 0, that has highest priority
- CBS shapers should be used with rated queues
- The bandwidth for CBS shapers has to be set a little bit more then potential incoming rate, thus, rate of all incoming tx queues has to be a little less
- Real rates can differ, due to discreetness
- Map skb-priority to txq is not enough, also skb-priority to 12 prio map has to be created with ip or vconfig tool
- Any 12/socket prio (0 7) for classes can be used, but for simplicity default values are used: 3 and 2
- only 2 classes tested: A and B, but checked and can work with more, maximum allowed 4, but only for 3 rate can be set.

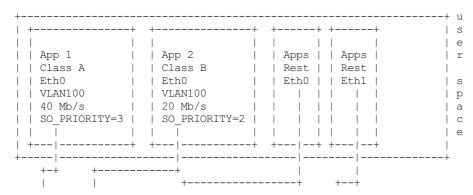
Test setup for examples



Example 1: One port tx AVB configuration scheme for target board

(prints and scheme for AM572x evm, applicable for single port boards)

- tc traffic class
- txq transmit queue
- p priority
- f fifo (cpsw fifo)
- S shaper configured



```
рO
                                                                        k
                                                                        е
                                                                        n
                                                                        1
                                                                        S
|tc0
       |tc1
               |tc2
                                                  Itc0
                                                                        р
                                                                        а
                                                                        С
                       |txq3
                                                  |txq4|
                            | Eth0.100
                                                          Eth1
                            <- L2 priority
|dma7|
       |dma6|
               |dma5|
                                     |dma3|
         \S
                                                                        S
                                                                        d
                                                                        r
                    10
                                   0|
 f3
       | f2
               | f0 |r
                                   r| f0
                                                                        V
|tc0
       |tc1
               |tc2 |t
                                   t|tc0
\CBS /
       \CBS /
               \CBS /1
                                   2\CBS
 \S /
         \S /
     // Add 4 tx queues, for interface Eth0, and 1 tx queue for Eth1
     $ ethtool -L eth0 rx 1 tx 5
     rx unmodified, ignoring
```

```
1.
```

```
2.
        // Check if num of queues is set correctly:
        $ ethtool -1 eth0
        Channel parameters for eth0:
        Pre-set maximums:
        RX:
        TX:
                        8
        Other:
                        0
        Combined:
                        0
        Current hardware settings:
        RX:
        TX:
        Other:
        Combined:
                        0
```

```
3.
        // TX queues must be rated starting from 0, so set bws for tx0 and tx1
        // Set rates 40 and 20 Mb/s appropriately.
        // Pay attention, real speed can differ a bit due to discreetness.
        // Leave last 2 tx queues not rated.
        \  \  \, echo 40 > /sys/class/net/eth0/queues/tx-0/tx_maxrate
        $ echo 20 > /sys/class/net/eth0/queues/tx-1/tx_maxrate
```

```
4.
        // Check maximum rate of tx (cpdma) queues:
        $ cat /sys/class/net/eth0/queues/tx-*/tx maxrate
        40
        20
        0
```

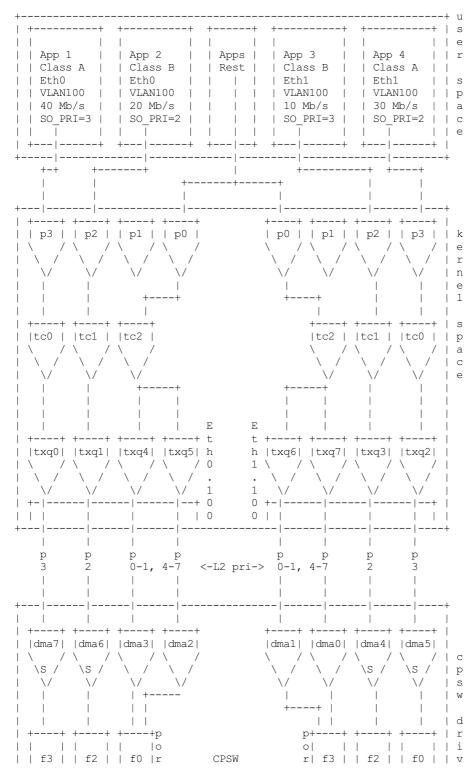
```
5.
          // Map skb->priority to traffic class:
          // 3pri -> tc0, 2pri -> tc1, (0,1,4-7)pri -> tc2
          // Map traffic class to transmit queue:
          // tc0 -> txq0, tc1 -> txq1, tc2 -> (txq2, txq3)
          \$ to qdisc replace dev eth0 handle 100: parent root mqprio num to 3 \setminus
          5a)
   // As two interface sharing same set of tx queues, assign all traffic
   // coming to interface Eth1 to separate queue in order to not mix it
   // with traffic from interface \mbox{Eth0,} so use separate txq to send
   // packets to Eth1, so all prio \rightarrow tc0 and tc0 \rightarrow txq4
   // Here hw 0, so here still default configuration for eth1 in hw \,
   \ tc qdisc replace dev eth1 handle 100: parent root mqprio num tc 1 \
   map 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 queues 104 hw 0
          // Check classes settings
          $ tc -g class show dev eth0
          +---(100:ffe2) maprio
               +---(100:3) maprio
               +---(100:4) maprio
          +--- (100:ffe1) maprio
               +---(100:2) maprio
          +---(100:ffe0) maprio
              +---(100:1) maprio
          $ tc -g class show dev eth1
          +--- (100:ffe0) maprio
              +---(100:5) maprio
  7.
          // Set rate for class A - 41 Mbit (tc0, txq0) using CBS Qdisc
          // Set it +1 Mb for reserve (important!)
          // here only idle slope is important, others arg are ignored
          \ensuremath{//} Pay attention, real speed can differ a bit due to discreetness
          $ tc qdisc add dev eth0 parent 100:1 cbs locredit -1438 \
          hicredit 62 sendslope -959000 idleslope 41000 offload 1
          net eth0: set FIFO3 bw = 50
  8.
          // Set rate for class B - 21 Mbit (tc1, txq1) using CBS Qdisc:
          // Set it +1 Mb for reserve (important!)
          \ tc qdisc add dev eth0 parent 100:2 cbs locredit -1468 \
          hicredit 65 sendslope -979000 idleslope 21000 offload 1
          net eth0: set FIFO2 bw = 30
          // Create vlan 100 to map sk->priority to vlan qos
          $ ip link add link eth0 name eth0.100 type vlan id 100
          8021q: 802.1Q VLAN Support v1.8
          8021q: adding VLAN 0 to HW filter on device eth0
          8021q: adding VLAN 0 to HW filter on device eth1
          net eth0: Adding vlanid 100 to vlan filter
 10.
          // Map skb->priority to L2 prio, 1 to 1
          $ ip link set eth0.100 type vlan \
          egress 0:0 1:1 2:2 3:3 4:4 5:5 6:6 7:7
 11.
          // Check egress map for vlan 100
          $ cat /proc/net/vlan/eth0.100
          [...]
          INGRESS priority mappings: 0:0 1:0 2:0 3:0 4:0 5:0 6:0 7:0
          EGRESS priority mappings: 0:0 1:1 2:2 3:3 4:4 5:5 6:6 7:7
 12.
          // Run your appropriate tools with socket option "SO PRIORITY"
          // to 3 for class A and/or to 2 for class B
          // (I took at https://www.spinics.net/lists/netdev/msq460869.html)
          ./tsn_talker -d 18:03:73:66:87:42 -i eth0.100 -p3 -s 1500&
          ./tsn talker -d 18:03:73:66:87:42 -i eth0.100 -p2 -s 1500&
 13.
          // run your listener on workstation (should be in same vlan)
          // (I took at https://www.spinics.net/lists/netdev/msg460869.html)
          ./tsn listener -d 18:03:73:66:87:42 -i enp5s0 -s 1500
          Receiving data rate: 39012 kbps
          Receiving data rate: 39012 kbps
```

0

```
Receiving data rate: 39012 kbps
        Receiving data rate: 39000 kbps
14.
        // Restore default configuration if needed
        $ ip link del eth0.100
        $ tc qdisc del dev eth1 root
        $ tc qdisc del dev eth0 root
        net eth0: Prev FIFO2 is shaped
        net eth0: set FIFO3 bw = 0
        net eth0: set FIFO2 bw = 0
        $ ethtool -L eth0 rx 1 tx 1
```

Example 2: Two port tx AVB configuration scheme for target board

(prints and scheme for AM572x evm, for dual emac boards only)



```
|tc0 | |tc1 | |tc2 |t
                                             t|tc0 | |tc1 | |tc2 | | e
   \CBS / \CBS / \CBS /1
                                            2\CBS / \CBS / \CBS / | r
                                              \s / \s / \ / |
         \S /
   \S /
          \/
    \/
       // Add 8 tx queues, for interface Eth0, but they are common, so are accessed
       // by two interfaces Eth0 and Eth1.
       $ ethtool -L eth1 rx 1 tx 8
       rx unmodified, ignoring
2.
       // Check if num of queues is set correctly:
       $ ethtool -1 eth0
       Channel parameters for eth0:
       Pre-set maximums:
       TX:
                       8
       Other:
                      0
                   0
       Combined:
       Current hardware settings:
              1
       TX:
                       8
       Other:
       Combined:
       // TX queues must be rated starting from 0, so set bws for tx0 and tx1 for Eth0
       // and for tx2 and tx3 for Eth1. That is, rates 40 and 20 Mb/s appropriately
       // for Eth0 and 30 and 10 Mb/s for Eth1.
       // Real speed can differ a bit due to discreetness
       // Leave last 4 tx queues as not rated
       $ echo 40 > /sys/class/net/eth0/queues/tx-0/tx_maxrate
       $ echo 20 > /sys/class/net/eth0/queues/tx-1/tx maxrate
       $ echo 30 > /sys/class/net/eth1/queues/tx-2/tx_maxrate
       $ echo 10 > /sys/class/net/eth1/queues/tx-3/tx maxrate
4.
       // Check maximum rate of tx (cpdma) queues:
       $ cat /sys/class/net/eth0/queues/tx-*/tx maxrate
       40
       20
       30
       10
       0
       0
       0
5.
       // Map skb->priority to traffic class for Eth0:
       // 3pri -> tc0, 2pri -> tc1, (0,1,4-7)pri -> tc2
       // Map traffic class to transmit queue:
       // tc0 -> txq0, tc1 -> txq1, tc2 -> (txq4, txq5)
       $ tc qdisc replace dev eth0 handle 100: parent root mqprio num_tc 3 \
       6.
       // Check classes settings
       $ tc -q class show dev eth0
       +---(100:ffe2) maprio
           +---(100:5) maprio
            +---(100:6) maprio
        +---(100:ffel) maprio
            +---(100:2) maprio
       +--- (100:ffe0) maprio
           +---(100:1) maprio
7.
       // Set rate for class A - 41 Mbit (tc0, txq0) using CBS Qdisc for Eth0
       // here only idle slope is important, others ignored
        // Real speed can differ a bit due to discreetness
       $ tc qdisc add dev eth0 parent 100:1 cbs locredit -1470 \
       hicredit 62 sendslope -959000 idleslope 41000 offload 1
       net eth0: set FIFO3 bw = 50
8.
       // Set rate for class B - 21 Mbit (tc1, txq1) using CBS Qdisc for Eth0
       \ tc qdisc add dev eth0 parent 100:2 cbs locredit -1470 \
       hicredit 65 sendslope -979000 idleslope 21000 offload 1
       net eth0: set FIFO2 bw = 30
9.
       // Create vlan 100 to map sk->priority to vlan qos for Eth0
       $ ip link add link eth0 name eth0.100 type vlan id 100
       net eth0: Adding vlanid 100 to vlan filter
```

```
10.
        // Map skb->priority to L2 prio for Eth0.100, one to one
         $ ip link set eth0.100 type vlan \
        egress 0:0 1:1 2:2 3:3 4:4 5:5 6:6 7:7
11.
        // Check egress map for vlan 100
        $ cat /proc/net/vlan/eth0.100
        [...]
        INGRESS priority mappings: 0:0 1:0 2:0 3:0 4:0 5:0 6:0 7:0
        EGRESS priority mappings: 0:0 1:1 2:2 3:3 4:4 5:5 6:6 7:7
12.
        // Map skb->priority to traffic class for Eth1:
        // 3pri -> tc0, 2pri -> tc1, (0,1,4-7)pri -> tc2
        // Map traffic class to transmit queue:
        // tc0 -> txq2, tc1 -> txq3, tc2 -> (txq6, txq7)
        $ tc qdisc replace dev eth1 handle 100: parent root mqprio num tc 3 \
        // Check classes settings
13.
        $ tc -g class show dev eth1
        +---(100:ffe2) maprio
             +---(100:7) maprio
             +---(100:8) maprio
        +---(100:ffe1) maprio
             +---(100:4) maprio
        +---(100:ffe0) maprio
            +---(100:3) maprio
14.
        // Set rate for class A - 31 Mbit (tc0, txq2) using CBS Qdisc for Eth1
        // here only idle slope is important, others ignored, but calculated
        // for interface speed - 100Mb for eth1 port.
        // Set it +1 Mb for reserve (important!)
        \ tc qdisc add dev eth1 parent 100:3 cbs locredit -1035 \
        hicredit 465 sendslope -69000 idleslope 31000 offload 1 net eth1: set FIFO3 bw = 31
15.
        // Set rate for class B - 11 Mbit (tc1, txq3) using CBS Qdisc for Eth1
        // Set it +1 Mb for reserve (important!)
        $ tc qdisc add dev eth1 parent 100:4 cbs locredit -1335 \
        hicredit 405 sendslope -89000 idleslope 11000 offload 1
        net eth1: set FIFO2 bw = 11
16.
        // Create vlan 100 to map sk->priority to vlan qos for Eth1
        \ ip link add link eth1 name eth1.100 type vlan id 100
        net ethl: Adding vlanid 100 to vlan filter
17.
        // Map skb->priority to L2 prio for Eth1.100, one to one
        $ ip link set eth1.100 type vlan \\
        egress 0:0 1:1 2:2 3:3 4:4 5:5 6:6 7:7
18.
        // Check egress map for vlan 100
        $ cat /proc/net/vlan/eth1.100
        INGRESS priority mappings: 0:0 1:0 2:0 3:0 4:0 5:0 6:0 7:0
        EGRESS priority mappings: 0:0 1:1 2:2 3:3 4:4 5:5 6:6 7:7
19.
        // Run appropriate tools with socket option "SO PRIORITY" to 3 \,
        // for class A and to 2 for class B. For both interfaces
         ./tsn_talker -d 18:03:73:66:87:42 -i eth0.100 -p2 -s 1500&
         ./tsn talker -d 18:03:73:66:87:42 -i eth0.100 -p3 -s 1500&
         ./tsn_talker -d 20:cf:30:85:7d:fd -i eth1.100 -p2 -s 1500&
         ./tsn talker -d 20:cf:30:85:7d:fd -i eth1.100 -p3 -s 1500&
20.
        // run your listener on workstation (should be in same vlan)
        // (I took at https://www.spinics.net/lists/netdev/msg460869.html)
         ./tsn listener -d 18:03:73:66:87:42 -i enp5s0 -s 1500
        Receiving data rate: 39012 kbps
        Receiving data rate: 39000 kbps
```

```
// Restore default configuration if needed
$ ip link del eth1.100
$ ip link del eth0.100
$ tc qdisc del dev eth1 root
net eth1: Prev FIFO2 is shaped
net eth1: set FIFO3 bw = 0
net eth1: set FIFO2 bw = 0
$ tc qdisc del dev eth0 root
net eth0: Prev FIFO2 is shaped
net eth0: set FIFO2 bw = 0
net eth0: set FIFO2 bw = 0
$ ethtool -L eth0 rx 1 tx 1
```