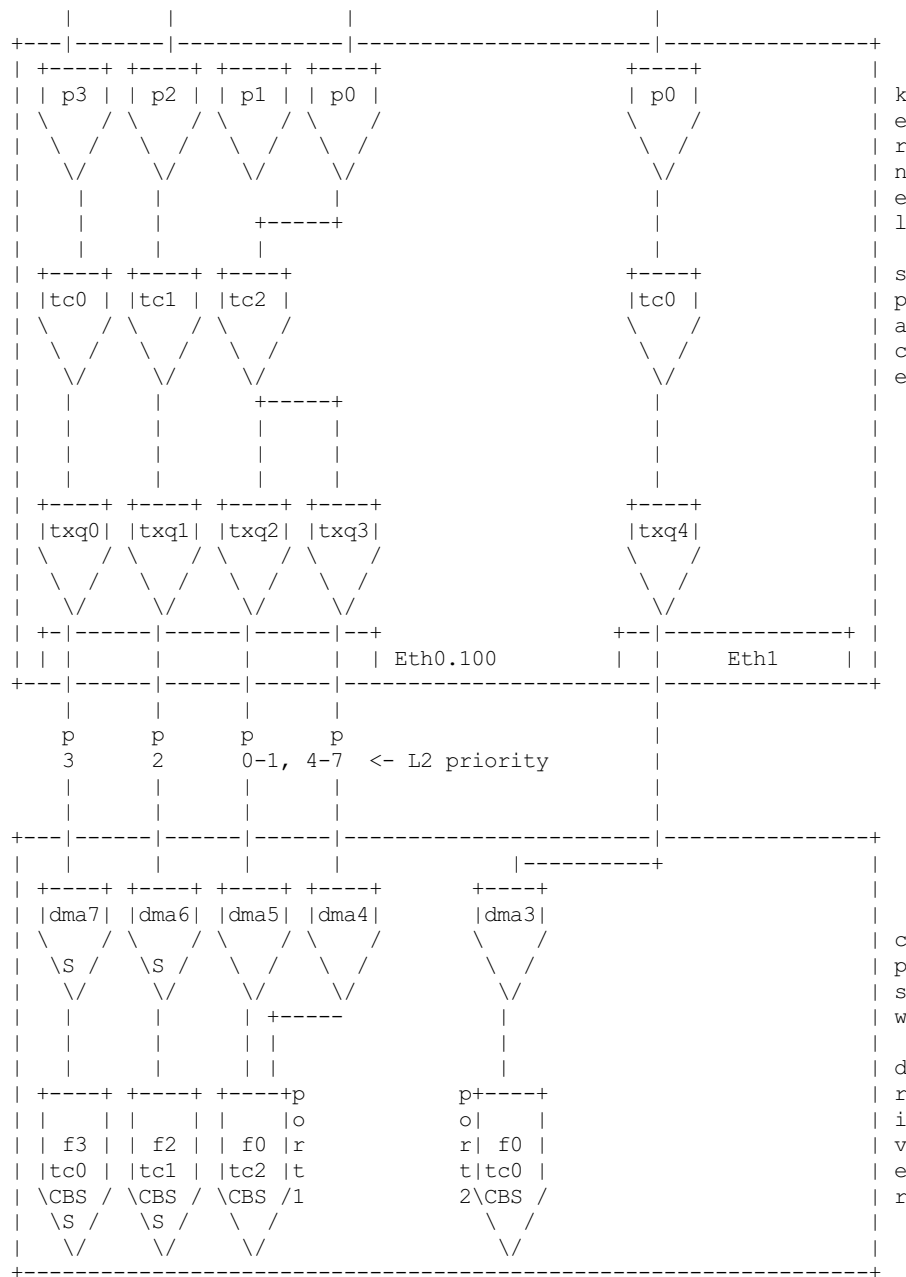


App 1	App 2	Apps	Apps
Class A	Class B	Rest	Rest
Eth0	Eth0	Eth0	Eth1
VLAN100	VLAN100		
40 Mb/s	20 Mb/s		
SO_PRIORITY=3	SO_PRIORITY=2		



1.

```
// Add 4 tx queues, for interface Eth0, and 1 tx queue for Eth1
$ ethtool -L eth0 rx 1 tx 5
rx unmodified, ignoring
```
2.

```
// Check if num of queues is set correctly:
$ ethtool -l eth0
Channel parameters for eth0:
Pre-set maximums:
RX:          8
TX:          8
Other:       0
Combined:    0
Current hardware settings:
RX:          1
TX:          5
Other:       0
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 discreteness.
// 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
```

```
0
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)
$ tc qdisc replace dev eth0 handle 100: parent root mqprio num_tc 3 \
map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 queues 1@0 1@1 2@2 hw 1
```

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 Eth0, so use separate txq to send
// packets to Eth1, so all prio -> tc0 and tc0 -> 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 queues 1@4 hw 0
```
6. 

```
// Check classes settings
$ tc -g class show dev eth0
+---(100:ffe2) mqprio
| +---(100:3) mqprio
| +---(100:4) mqprio
|
+---(100:ffe1) mqprio
| +---(100:2) mqprio
|
+---(100:ffe0) mqprio
| +---(100:1) mqprio

$ tc -g class show dev eth1
+---(100:ffe0) mqprio
| +---(100:5) mqprio
```
  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
// Pay attention, real speed can differ a bit due to discreteness
$ 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
```
  9. 

```
// 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/msg460869.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
Receiving data rate: 39012 kbps
Receiving data rate: 39012 kbps
Receiving data rate: 39012 kbps
```

```

Receiving data rate: 39012 kbps
Receiving data rate: 39012 kbps
Receiving data rate: 39012 kbps
Receiving data rate: 39012 kbps
Receiving data rate: 39012 kbps
Receiving data rate: 39012 kbps
Receiving data rate: 39012 kbps
Receiving data rate: 39000 kbps

```

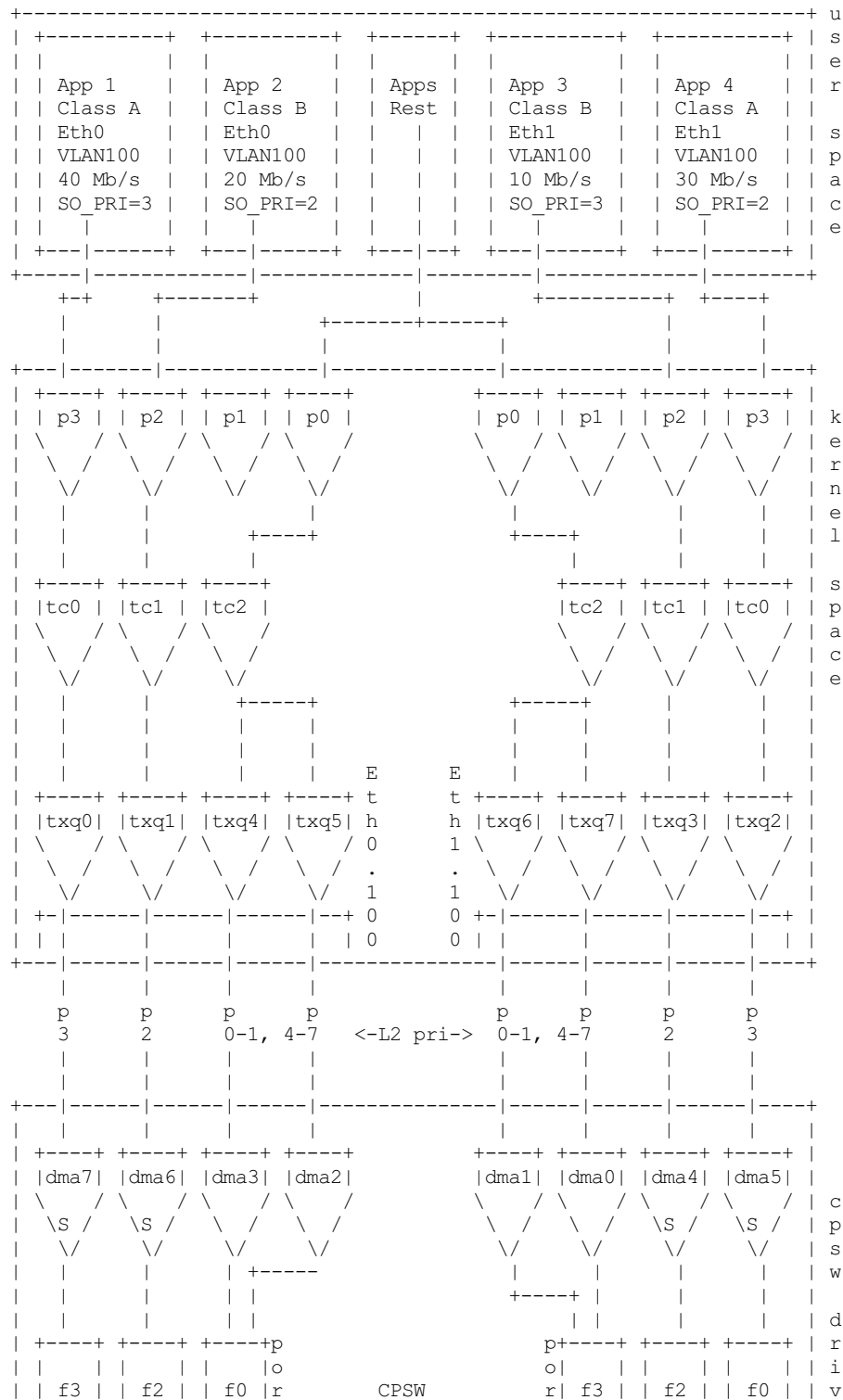
```

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)



| t tc0    tc1    tc2     e |
|---------------------------|
| \CBS / \CBS / \CBS /1     |
| \S / \S / \ /             |
| \ / \ / \ /               |

| t tc0    tc1    tc2     e |
|---------------------------|
| 2\CBS / \CBS / \CBS /   r |
| \S / \S / \ /             |
| \ / \ / \ /               |

```

+-----+
=====Eth=====>

```

1. // 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 -l eth0  
Channel parameters for eth0:  
Pre-set maximums:  
RX: 8  
TX: 8  
Other: 0  
Combined: 0  
Current hardware settings:  
RX: 1  
TX: 8  
Other: 0  
Combined: 0
3. // 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 discreteness  
// 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  
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 \  
map 2 2 1 0 2 2 2 2 2 2 2 2 2 2 2 2 queues 1@0 1@1 2@4 hw 1
6. // Check classes settings  
\$ tc -g class show dev eth0  
+---(100:ffe2) mqprio  
| +---(100:5) mqprio  
| +---(100:6) mqprio  
|  
+---(100:ffe1) mqprio  
| +---(100:2) mqprio  
|  
+---(100:ffe0) mqprio  
+---(100:1) mqprio
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 discreteness  
\$ 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

- [illegible]

21.       // 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 FIFO3 bw = 0  
net eth0: set FIFO2 bw = 0  
\$ ethtool -L eth0 rx 1 tx 1