RoCE & QoS

Jan 05, 2015

V-1.0.0

Changqing Li
licq@mellanox.com

Agenda

- Background
- What is RoCE
- RoCE Packet Format
- RoCE packet encapsulation
- RoCE QoS with Verbs API
- RoCE QoS with RDMA_CM API
- Differences between RDMA programming over IB and RoCE
- Performance
- RoCEv2
- Reference
- Q&A

Background

- Advantage of RDMA(Zero-copy, full protocol offload, high performance, good latency, low CPU utilization).
- Lots of Ethernet device and equipment deployed in data center
- Legecy TCP/IP stack becoming a problem with the growth of BW
- IEEE 802.1 DCB ensures lossless Ethernet network.

What is RoCE

- RoCE - RDMA over Converged Ethernet
  - 802.1Qbb - PFC
  - 802.1az - ETS
  - 802.1AB - LLDP, DCBx
  - 802.1Qau - CN

- Defined by IBTA
  - Annex A16: RDMA over Converged Ethernet (RoCE) of “InfiniBand Architecture Specification Volume 1 Release 1.2.1”

RoCE Packet Format

Legacy RoCE Packet

0x8951

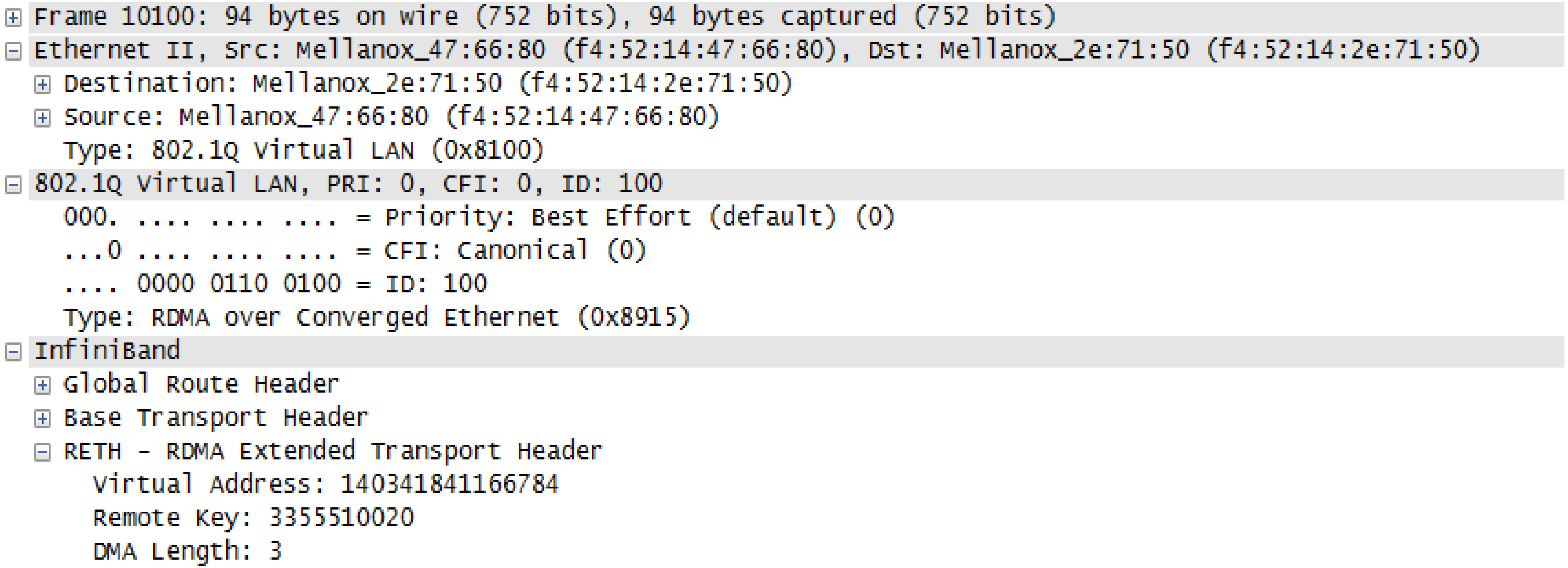
D_MAC						S_MAC						ET		GRH	BTH	ETH	PayLoad	CRC/FCS			
0	1	2	3	4	5	0	1	2	3	4	5	0	1					0	1	2	3

RoCE Packet with Vlan Tag

0x8951

D_MAC						S_MAC						802.1Q/p				ET		GRH	BTH	ETH	PayLoad	CRC/FCS			
0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	0	1					0	1	2	3

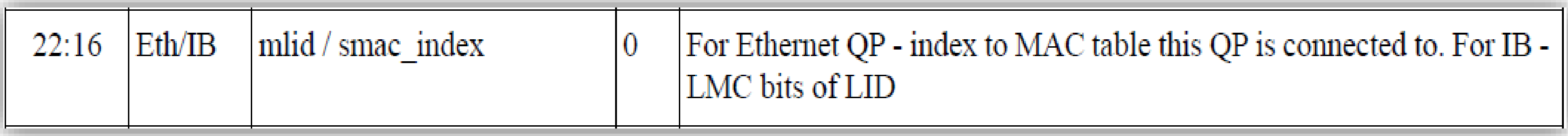
RoCE packet captured by ibdump



```
⊞ Frame 10100: 94 bytes on wire (752 bits), 94 bytes captured (752 bits)
⊟ Ethernet II, Src: Mellanox_47:66:80 (f4:52:14:47:66:80), Dst: Mellanox_2e:71:50 (f4:52:14:2e:71:50)
  ⊞ Destination: Mellanox_2e:71:50 (f4:52:14:2e:71:50)
  ⊞ Source: Mellanox_47:66:80 (f4:52:14:47:66:80)
    Type: 802.1Q Virtual LAN (0x8100)
⊟ 802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 100
    000. .... .... .... = Priority: Best Effort (default) (0)
    ...0 .... .... .... = CFI: Canonical (0)
    .... 0000 0110 0100 = ID: 100
    Type: RDMA over Converged Ethernet (0x8915)
⊟ InfiniBand
  ⊞ Global Route Header
  ⊞ Base Transport Header
  ⊟ RETH - RDMA Extended Transport Header
      Virtual Address: 140341841166784
      Remote Key: 3355510020
      DMA Length: 3
```

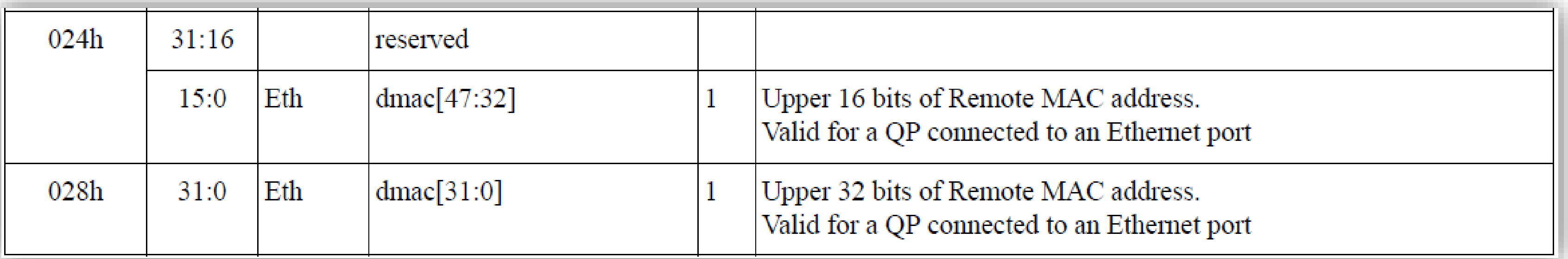
RoCE Packet Encapsulation – PRM description

Source MAC Address



22:16	Eth/IB	mlid / smac_index	0	For Ethernet QP - index to MAC table this QP is connected to. For IB - LMC bits of LID

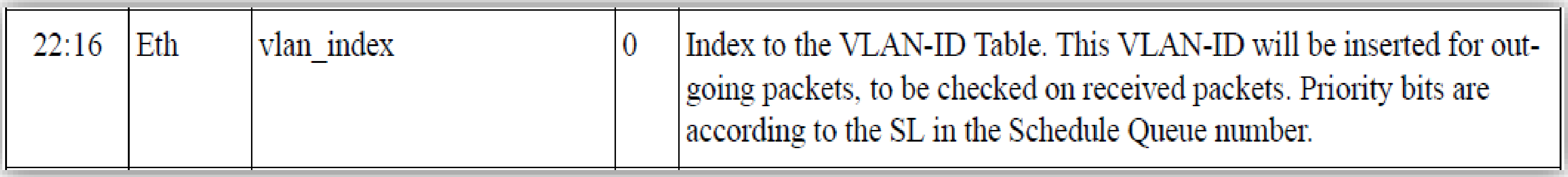
Destination MAC Address



024h	31:16		reserved		
	15:0	Eth	dmac[47:32]	1	Upper 16 bits of Remote MAC address. Valid for a QP connected to an Ethernet port
028h	31:0	Eth	dmac[31:0]	1	Upper 32 bits of Remote MAC address. Valid for a QP connected to an Ethernet port

RoCE Packet Encapsulation – PRM description

Vlan ID



22:16	Eth	vlan_index	0	Index to the VLAN-ID Table. This VLAN-ID will be inserted for out-going packets, to be checked on received packets. Priority bits are according to the SL in the Schedule Queue number.

UP (Vlan Priority)

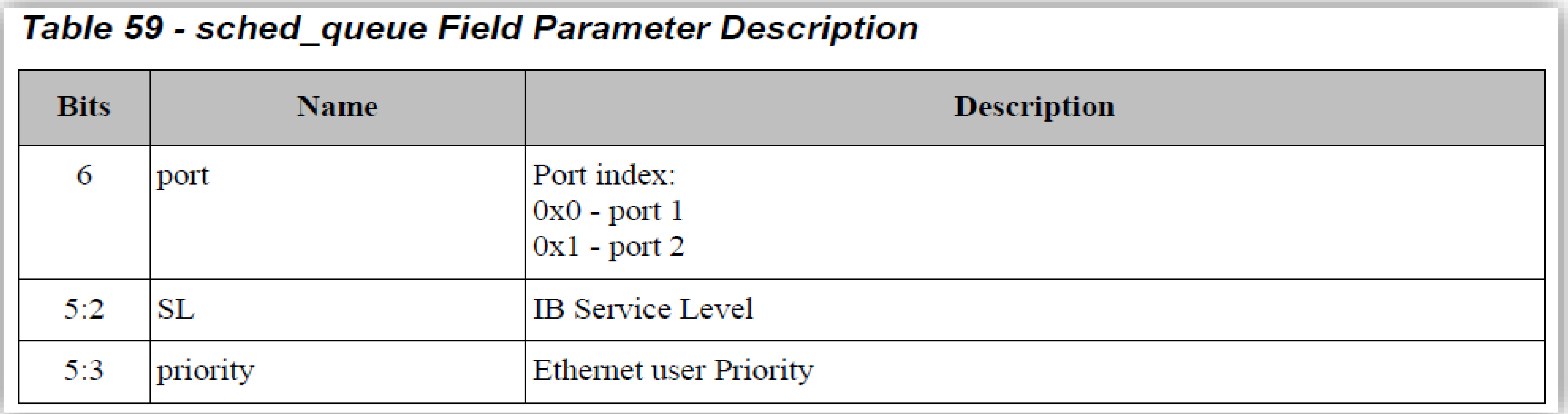


Table 59 - sched_queue Field Parameter Description

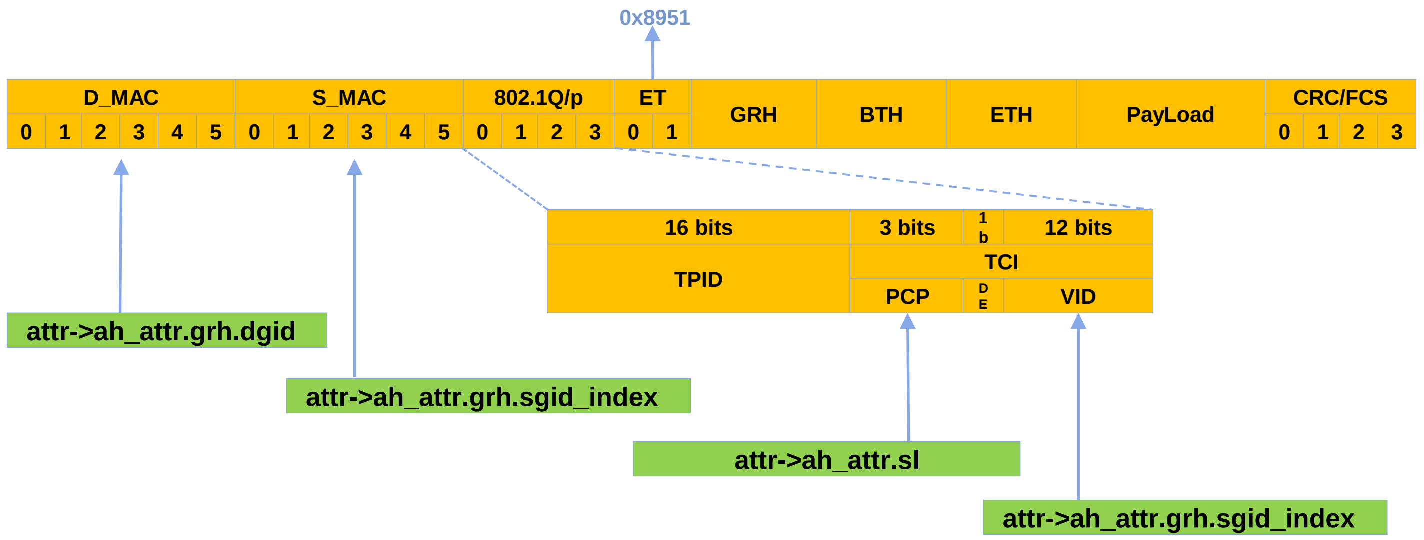
Bits	Name	Description
6	port	Port index: 0x0 - port 1 0x1 - port 2
5:2	SL	IB Service Level
5:3	priority	Ethernet user Priority

RoCE Packet Encapsulation – Application Behavior

User space RoCE application needs to do:

```
int flags           = IBV_QP_STATE;
attr->qp_state = IBV_QPS_RTR;

attr->ah_attr.is_global          = 1;
attr->ah_attr.grh.dgid           = <Destination GID>;
attr->ah_attr.grh.sgid_index  = <Local GID index>;
attr->ah_attr.grh.hop_limit     = 1;
attr->ah_attr.sl                       = <SL>;

ibv_modify_qp(qp, attr, flags);
```

RoCE Packet Encapsulation – Hardware Behavior

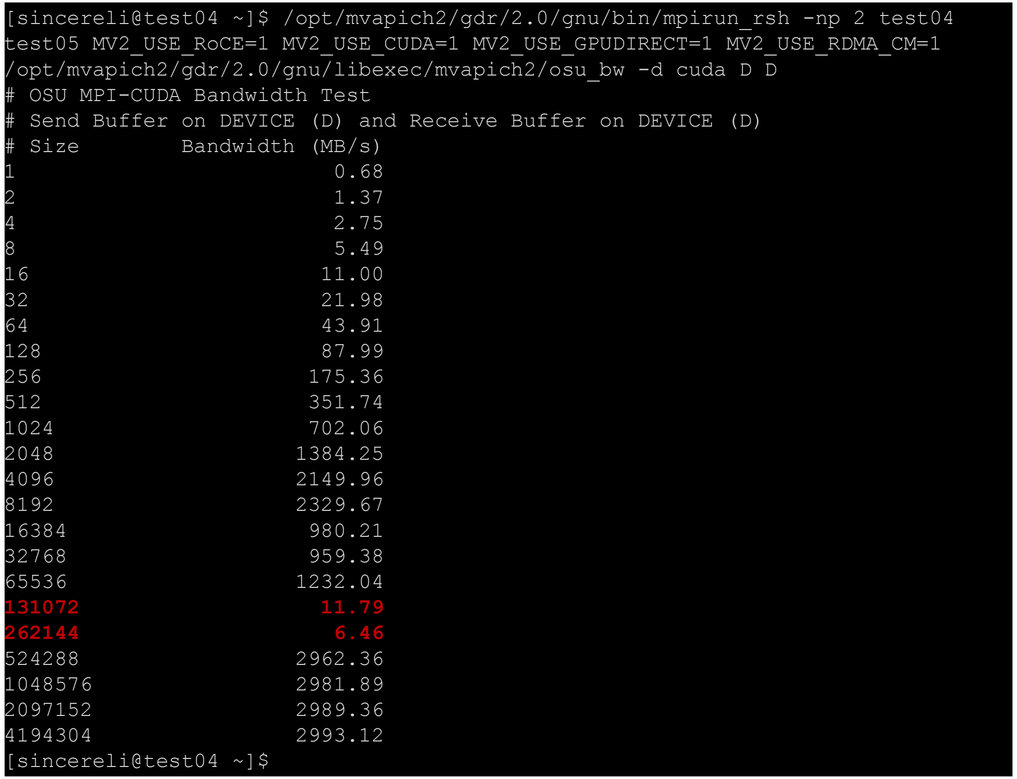


RoCE Packet Encapsulation – Driver Behavior

```
-> __uverbs_modify_qp
        -> ib_query_gid(sgid_index, &sgid)
        -> rdma_addr_find_dmac_by_grh(sgid, dgid, (char*)dmac, &vlan_id)
                -> sip = rdma_gid2ip(sgid)
                -> dip = rdma_gid2ip(dgid)
                -> rdma_resolve_ip(sip, dip, &dev_addr)
                -> memcpy(dmac, dev_addr.dst_dev_addr, ETH_ALEN);     // Got dmac
                -> dev = dev_get_by_index(bound_dev_if)
                -> vlan_id = rdma_vlan_dev_vlan_id(dev)               // Got Vlan ID

        -> rdma_addr_find_smac_by_sgid(sgid, (char*)smac)
                -> sip = rdma_gid2ip(sgid)
                -> rdma_translate_ip(sip, &dev_addr)
                -> memcpy(smac, dev_addr.src_dev_addr, ETH_ALEN);     // Got dmac

        -> ib_modify_qp
                -> mlx4_ib_modify_qp
                        -> __mlx4_ib_modify_qp
                                -> mlx4_set_path
                                        -> path->sched_queue = (ah->sl & 7) << 3;    // Set Prio
```

RoCE & PFC

- What will happen without PFC?



```
[sincereli@test04 ~]$ /opt/mvapich2/gdr/2.0/gnu/bin/mpirun_rsh -np 2 test04
test05 MV2_USE_RoCE=1 MV2_USE_CUDA=1 MV2_USE_GPUDIRECT=1 MV2_USE_RDMA_CM=1
/opt/mvapich2/gdr/2.0/gnu/libexec/mvapich2/osu_bw -d cuda D D
# OSU MPI-CUDA Bandwidth Test
# Send Buffer on DEVICE (D) and Receive Buffer on DEVICE (D)
# Size        Bandwidth (MB/s)
1                         0.68
2                         1.37
4                         2.75
8                         5.49
16                       11.00
32                       21.98
64                       43.91
128                      87.99
256                     175.36
512                     351.74
1024                    702.06
2048                   1384.25
4096                   2149.96
8192                   2329.67
16384                   980.21
32768                   959.38
65536                  1232.04
131072                   11.79
262144                    6.46
524288                 2962.36
1048576                2981.89
2097152                2989.36
4194304                2993.12
[sincereli@test04 ~]$
```

RoCE QoS with Verbs API

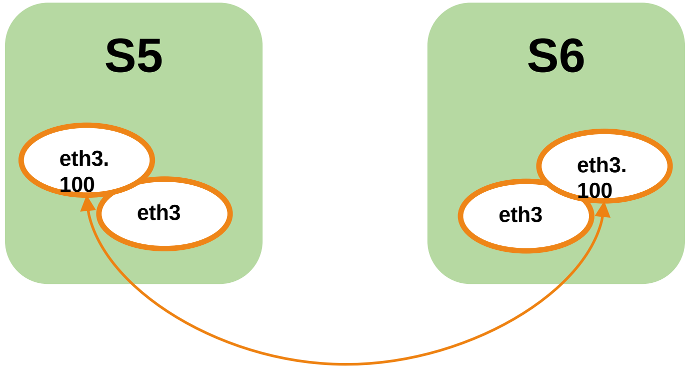
- SL (Service Level) to UP mapping

  UP = SL & 0x7

- Test RoCE QoS:

```
server$ ib_write_lat –x <GID_IDX> -S <SL> -a

client$  ib_write_lat –x <GID_IDX> -S <SL> <Server_IP_Addr>

watch –n 1 ‘ethtool –S | grep packets | grep prio’
```

RoCE QoS with Verbs API - Example



```
s6$ ib_write_lat –x 2 -S 2 -a

s5$ ib_write_lat –x 2 -S 2 -a 192.168.100.6

s5$ watch –n 1 ‘ethtool –S | grep packets | grep prio’
```

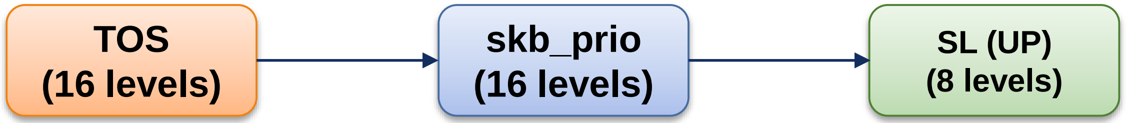
```
s5$ ifconfig eth3.100
eth3.100  Link encap:Ethernet  HWaddr 00:02:C9:18:97:40
          inet addr:192.168.100.5  Bcast:192.168.100.255  Mask:255.255.255.0

s6$ ifconfig eth3
eth3      Link encap:Ethernet  HWaddr 00:02:C9:A0:57:80
          inet addr:192.168.1.6  Bcast:192.168.1.255  Mask:255.255.255.0

s6$ ifconfig eth3.100
eth3.100  Link encap:Ethernet  HWaddr 00:02:C9:A0:57:80
          inet addr:192.168.100.6  Bcast:192.168.100.255  Mask:255.255.255.0

s6$ ibv_devinfo -v
hca_id: mlx4_0
        transport:                      InfiniBand (0)
        fw_ver:                         2.32.5100
        raw_eth_odp_caps:
                                        NO SUPPORT
        max_dct:                        0
                port:   1
                        state:                  PORT_ACTIVE (4)
                        max_mtu:                4096 (5)

                        active_width:           4X (2)
                        active_speed:           10.0 Gbps (4)
                        phys_state:             LINK_UP (5)
                        GID[  0]:               fe80:0000:0000:0000:0202:c9ff:fea0:5780
                        GID[  1]:               0000:0000:0000:0000:0000:ffff:c0a8:0106  ------> 192.168.1.6
                        GID[  2]:               0000:0000:0000:0000:0000:ffff:c0a8:6406  ------> 192.168.100.6
```

RoCE QoS with rdma_cm API

- Set QoS in user space application

```
rdma_set_option(rdma_cm_id, tos);
```

- TOS_2_SL mapping(2 steps)

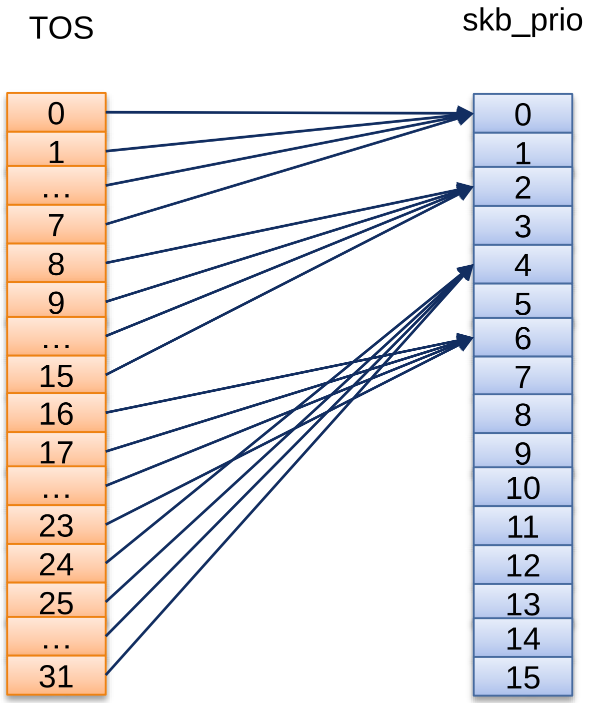


- Kernel driver handles the mapping

```
skb_prio = rt_tos2priority(id_priv->tos);
ndev      = ndev->priv_flags & IFF_802_1Q_VLAN ? vlan_dev_real_dev(ndev) : ndev;
sl        = netdev_get_prio_tc_map(ndev, skb_prio);
```

RoCE QoS with rdma_cm API – Mapping from TOS to skb_prio

- Static mapping



```
#define TC_PRIO_BESTEFFORT              0
#define TC_PRIO_FILLER                  1
#define TC_PRIO_BULK                    2
#define TC_PRIO_INTERACTIVE_BULK        4
#define TC_PRIO_INTERACTIVE             6
#define TC_PRIO_CONTROL                 7
#define TC_PRIO_MAX                     15
const __u8 ip_tos2prio[16] = {
                TC_PRIO_BESTEFFORT,
                TC_PRIO_BESTEFFORT,
                TC_PRIO_BESTEFFORT,
                TC_PRIO_BESTEFFORT,
                TC_PRIO_BULK,
                TC_PRIO_BULK,
                TC_PRIO_BULK,
                TC_PRIO_BULK,
                TC_PRIO_INTERACTIVE,
                TC_PRIO_INTERACTIVE,
                TC_PRIO_INTERACTIVE,
                TC_PRIO_INTERACTIVE),
                TC_PRIO_INTERACTIVE_BULK,
                TC_PRIO_INTERACTIVE_BULK,
                TC_PRIO_INTERACTIVE_BULK,
                TC_PRIO_INTERACTIVE_BULK
};
```

RoCE QoS with rdma_cm API – Mapping from skb_prio to SL

- Set&Check netdev skprio2up mapping

Example:

```
s6$ tc_wrap.py -i eth3 -u 3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3
UP  0
UP  1
UP  2
UP  3
                                        skprio: 0
                                        skprio: 1
                                        skprio: 2 (tos: 8)
                                        skprio: 3
                                        skprio: 4 (tos: 24)
                                        skprio: 5
                                        skprio: 6 (tos: 16)
                                        skprio: 7
                                        skprio: 8
                                        skprio: 9
                                        skprio: 10
                                        skprio: 11
                                        skprio: 12
                                        skprio: 13
                                        skprio: 14
                                        skprio: 15
UP  4
UP  5
UP  6
UP  7
s6$ cat /sys/class/net/eth3/qos/skprio2up
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
```

Reference

- <u>How To Run RoCE and TCP over L2 Enabled with PFC</u>
- RoCE Spec: Annex A16 RoCE.pdf

Q&A

THANK YOU