Flow Control

Table of contents

Priority Flow Control (PFC)

Priority Flow Control (PFC) IEEE 802.1Qbb applies pause functionality to specific classes of traffic on the Ethernet link. For example, PFC can provide lossless service for the RoCE traffic and best-effort service for the standard Ethernet traffic. PFC can provide different levels of service to specific classes of Ethernet traffic (using IEEE 802.1p traffic classes).

Configuring PFC on ConnectX-4 and above

1. Enable PFC on the desired priority:

   ```
   mlnx_qos -i <ethX> --pfc <0/1>,<0/1>,<0/1>,<0/1>,<0/1>,<0/1>,<0/1>,<0/1>
   ```

   Example (Priority=4):

   ```
   mlnx_qos -i eth1 --pfc 0,0,0,0,1,0,0,0
   ```

2. Create a VLAN interface:

   ```
   vconfig add <ethX> <VLAN_ID>
   ```

   Example (VLAN_ID=5):

   ```
   vconfig add eth1 5
   ```

3. Set egress mapping:

   1. For Ethernet traffic:

      ```
      vconfig set_egress_map <vlan_einterface> <skprio> <up>
      ```

      Example (skprio=3, up=5):

```
vconfig set_egress_map eth1.5 3 5
```

4. Create 8 Traffic Classes (TCs):

```
tc_wrap.py -i <interface>
```

5. Enable PFC on the switch.
   For information on how to enable PFC on your respective switch, please refer to Switch FC/PFC Configuration sections in the RDMA/RoCE Solutions Community page .

PFC Configuration Using LLDP DCBX

PFC Configuration on Hosts

PFC Auto-Configuration Using LLDP Tool in the OS

1. Start lldpad daemon on host.

```
lldpad -d Or
service lldpad start
```

2. Send lldpad packets to the switch.

```
lldptool set-lldp -i <ethX> adminStatus=rxtx ;
lldptool -T -i <ethX> -V sysName enableTx=yes;
lldptool -T -i <ethX> -V portDesc enableTx=yes ;
lldptool -T -i <ethX> -V sysDesc enableTx=yes
lldptool -T -i <ethX> -V sysCap enableTx=yess
lldptool -T -i <ethX> -V mngAddr enableTx=yess
lldptool -T -i <ethX> -V PFC enableTx=yes;
lldptool -T -I <ethX> -V CEE-DCBX enableTx=yes;
```

3. Set the PFC parameters.

- For the CEE protocol, use dcbtool:

  ```
  dcbtool sc <ethX> pfc pfcup:<xxxxxxxx>
  ```

  Example:

  ```
  dcbtool sc eth6 pfc pfcup:01110001
  ```

  where:

[pfcup:xxxxxxxx]	Enables/disables priority flow control. From left to right (priorities 0-7) - x can be equal to either 0 or 1. 1 indicates that the priority is configured to transmit priority pause.

- For IEEE protocol, use lldptool:

  ```
  lldptool -T -i <ethX> -V PFC enabled=x,x,x,x,x,x,x,x
  ```

  Example:

  ```
  lldptool -T -i eth2 -V PFC enabled=1,2,4
  ```

  where:

enabled	Displays or sets the priorities with PFC enabled. The set attribute takes a comma-separated list of priorities to enable, or the string none to disable all priorities.

PFC Auto-Configuration Using LLDP in the Firmware (for mlx5 driver)

There are two ways to configure PFC and ETS on the server:

1. **Local Configuration** - Configuring each server manually.
2. **Remote Configuration** - Configuring PFC and ETS on the switch, after which the switch will pass the configuration to the server using LLDP DCBX TLVs.

There are two ways to implement the remote configuration using mlx5 driver:

1. Configuring the adapter firmware to enable DCBX.
2. Configuring the host to enable DCBX.

For further information on how to auto-configure PFC using LLDP in the firmware, refer to the HowTo Auto-Config PFC and ETS on ConnectX-4 via LLDP DCBX Community post.

PFC Configuration on Switches

1. In order to enable DCBX, LLDP should first be enabled:

   ```
   switch (config) # lldp
   show lldp interfaces ethernet remote
   ```

2. Add DCBX to the list of supported TLVs per required interface.

   For IEEE DCBX:

   ```
   switch (config) # interface 1/1
   switch (config interface ethernet 1/1) # lldp tlv-select dcbx
   ```

   For CEE DCBX:

   ```
   switch (config) # interface 1/1
   switch (config interface ethernet 1/1) # lldp tlv-select dcbx-cee
   ```

3. [**Optional**] Application Priority can be configured on the switch, with the required ethertype and priority. For example, IP packet, priority 1:

   ```
   switch (config) # dcb application-priority 0x8100 1
   ```

4. Make sure PFC is enabled on the host (for enabling PFC on the host, refer to PFC Configuration on Hosts section above). Once it is enabled, it will be passed in the

LLDP TLVs.

5. Enable PFC with the desired priority on the Ethernet port.

```
dcb priority-flow-control enable force
dcb priority-flow-control priority <priority> enable
interface ethernet <port> dcb priority-flow-control mode on force
```

Example - Enabling PFC with priority 3 on port 1/1:

```
dcb priority-flow-control enable force
dcb priority-flow-control priority 3 enable
interface ethernet 1/1 dcb priority-flow-control mode on force
```

Priority Counters

Several ingress and egress counters per priority are supported. Run ethtool -S to get the full list of port counters.

ConnectX-4 Counters

- Rx and Tx Counters:
  - Packets
  - Bytes
  - Octets
  - Frames
  - Pause
  - Pause frames
  - Pause Duration

    - Pause Transition

ConnectX-4 Example

```
# ethtool -S eth35 | grep prio4
prio4_rx_octets: 62147780800
prio4_rx_frames: 14885696
prio4_tx_octets: 0
prio4_tx_frames: 0
prio4_rx_pause: 0
prio4_rx_pause_duration: 0
prio4_tx_pause: 26832
prio4_tx_pause_duration: 14508
prio4_rx_pause_transition: 0
```

Note: The Pause counters in ConnectX-4 are visible via ethtool only for priorities on which PFC is enabled.

PFC Storm Prevention

PFC storm prevention enables toggling between default and auto modes.
The stall prevention timeout is configured to 8 seconds by default. Auto mode sets the stall prevention timeout to be 100 msec.
The feature can be controlled using sysfs in the following directory:
/sys/class/net/eth*/settings/ pfc_stall_prevention

- To query the PFC stall prevention mode:

  ```
  cat /sys/class/net/eth*/settings/pfc_stall_prevention
  ```

  Example

```
$ cat /sys/class/net/ens6/settings/pfc_stall_prevention
default
```

- To configure the PFC stall prevention mode:

```
Echo "auto"/"default" > /sys/class/net/eth*/settings/pfc_stall_prevention
```

The following two counters were added to the ethtool -S:

- **tx_Pause_storm_warning_events** - when the device is stalled for a period longer than a pre-configured watermark, the counter increases, allowing the debug utility an insight into current device status.
- **tx_pause_storm_error_events** - when the device is stalled for a period longer than a pre-configured timeout, the pause transmission is disabled, and the counter increase.

Dropless Receive Queue (RQ)

Dropless RQ feature enables the driver to notify the FW when SW receive queues are overloaded. This scenario takes place when the handling of SW receive queue is slower than the handling of the HW receive queues.
When this feature is enabled, a packet that is received while the receive queue is full will not be immediately dropped. The FW will accumulate these packets assuming posting of new WQEs will resume shortly. If received WQEs are not posted after a certain period of time, out_of_buffer counter will increase, indicating that the packet has been dropped.
This feature is disabled by default. In order to activate it, ensure that Flow Control feature is also enabled.

```
ethtool --set-priv-flags ens6 dropless_rq on
```

To get the feature state, run:

```
ethtool --show-priv-flags DEVNAME
```

Output example:

```
Private flags for DEVNAME:
rx_cqe_moder : on
rx_cqe_compress: off
sniffer : off
dropless_rq : off
hw_lro : off
```

To disable the feature, run:

```
ethtool --set-priv-flags ens6 dropless_rq off
```