Western Digital.

Configuration Guide

Configuring a Mellanox® Network Adapter for Lossless Networking with OpenFlex™ Platforms

Abstract

This configuration guide provides an overview of how to configure lossless Ethernet settings on Mellanox based Ethernet network adapters with Western Digital® OpenFlex platforms.

April 2025

Table of Contents

Introduction

NVMe-oF™ based storage offers the promise of low latency shared storage. To obtain the performance potential of this technology, Ethernet Network Adapters in initiator hosts must be configured for lossless networking using standard Data Center Bridging (DCB) technologies.

Configuration Process Summary

The process of enabling lossless networking functionality on Mellanox® based Ethernet Network Adapters can be broken down into the following steps:

1. Download and Install Software Driver
2. Configure Priority Flow Control (PFC)
3. Configure Explicit Congestion Notification (ECN)
4. Show Pertinent Switch Counters
5. Configure Boot Time Scripts

At the end of each section will be included a single code block with all the CLI commands. This is for convenience to enable the reader to copy-and-paste the commands used in that section into their own script in a single action instead of requiring the reader to copy and paste each individual CLI command.

Example Hardware Specifications

In this guide, a ConnectX-6, with device driver version 5.8-1.1.2.1 and firmware version 20.35.2000 (MT_0000000236) was used. Terminal output shown in this guide may vary based on the product and firmware version. For additional information, see the Data24 Compatibility Matrix.

Configuration Table

Included below for convenience is a table to record the lossless configuration values for deployment.

Description	Variable	Example	Deployment Value
Ethernet Device Name	<INTERFACE>	ens7f0np0	
RoCE Device Name	<ROCE_DEVICE>	mlx5_0	
PFC Priority	<ROCE_PRIORITY>	3	
PFC DSCP	<ROCE_DSCP>	24	
CNP Priority	<CNP_PRIORITY>	6	
CNP DSCP	<CNP_DSCP>	48	
MST Device Path	<MST_DEVICE>	/dev/mst/mt4125_pciconf0	
Type of Service	<TOS>	96	
PCIe Device	<PCIE_DEVICE>	ca:00.0	

Download and Install Software Driver

1. First verify that the system has a Mellanox adapter installed. After the host has booted into the operating system (Linux®), run the following command to verify the Mellanox Network Adapter shows up on the PCIe bus.

```
$ lspci -v | grep -i Mellanox
```

2. Visit NVIDIA's website to obtain the Mellanox OFED: https://network.nvidia.com/products/infiniband-drivers/linux/mlnx_ofed/.

3. In the MLNX_OFED Download Center, select Archive Versions and search for an approved version that correlates to the certified list to the set-up. Then select the appropriate OS distribution, OS Version, and Architecture.

4. Download the resultant tgz driver file.

5. Copy the driver .tgz to the host.

   Note: If the host has access to the internet, the driver installation performs a firmware update. If the host does not have access to the internet, refer to Mellanox documentation to properly update the RNIC to the most current firmware.

6. Install the pre-requisite packages.

   Note: This is not a comprehensive list of pre-requisite packages. This is highly dependent on how the OS installation was performed. Attempt the installation of the driver and examine the output to discover any additional pre-requisite packages that are needed.

```
$ yum install python-devel kernel-devel redhat-rpm-config rpm-build gcc tcl gcc-gfortran tk
perl-File-Copy perl-File-Compare perl-sigtrap autoconf automake libtool kernel-rpm-macros
```

7. Extract the driver .tgz file on the host server.

```
$ tar zxvf MNX _ OFED _ LINUX-x-x.x.x.x-x86 _ 64.tgz
```

8. Run the installation script from the folder where the tar file was extracted. Make sure to use the --add-kernel-support, --with-nvmf, and --skip-repo command modifiers.

```
$ ./mlnxofedinstall --add-kernel-support --with-nvmf --skip-repo
```

9. Create or edit the file /etc/modules-load.d/modules.conf to contain the following.

```
#
# Mellanox nvmeOF
rdma _ cm
ib _ uverbs
rdma _ ucm
nvme
nvme-rdma
ib _ core
ib _ cm
ib _ umad
nvme _ rdma
nvme _ fabrics
```

10. Restart the driver.

```
$ /etc/init.d/openibd restart
```

11. Check ibv_devices and verify that the Mellanox Network Adapter is in the list.

```
$ ibv _ devices
```

Example Output:

```
root@pfedlr750-24:~# ibv _ devices
    device                 node GUID
    ------              ----------------
    mlx5 _ 0             b8cef60300076986
    mlx5 _ 1             b8cef60300076987
```

12. Configure IP Address and Subnet Mask on fabric ports.

13. Configure the appropriately sized MTU for communication with the NVMe over Fabrics device.

Note: The OpenFlex Data24 ships with a default MTU of 2200. The OpenFlex Data24 3200 ships with a default MTU of 5000.

14. When directly connecting to an OpenFlex Data24, without a switch, auto negotiation must be disabled on the fabric port. This can be accomplished with the ethtool command.

```
$ ethtool -s <INTERFACE> speed 100000 autoneg off
```

Note: This is only required when directly connecting to the OpenFlex Data24.

15. Ensure the file /etc/security/limits.conf contains the following lines:

```
* soft memlock unlimited
* hard memlock unlimited
* hard nofile 1048000
* soft nofile 1048000
```

16. Rebuild initrd.

```
$ dracut -f
```

Configuration PFC

The following is a list of things to know before following this process:

- All these commands must be executed for each port on the RDMA Network Adapter that needs to be configured.
- Many of these commands are not persistent through reboot and will have to be scripted to run at every boot.

PFC set up process:

1. Use MST to acquire PCIe address to Ethernet Device to IB Device mapping as well as the MST device path.

Note: These commands are informational and not required to be run at boot.

a. Start MST

```
$ mst start
```

b. Get MST device path

```
$ mst status -v
```

Example Output:

```
root@pfedlr750-24:~# mst status -v
```

MST modules:

```
------------
    MST PCI module is not loaded
    MST PCI configuration module loaded
PCI devices:
------------
DEVICE_TYPE           MST                           PCI       RDMA     NET                NUMA
ConnectX6DX(rev:0)    /dev/mst/mt4125_pciconf0.1    ca:00.1   mlx5_1   net-ens7f1np1      1
ConnectX6DX(rev:0)    /dev/mst/mt4125_pciconf0      ca:00.0   mlx5_0   net-ens7f0np0      1
```

2. Ensure LLDP and DCBx are disabled.

Note: This command is persistent and is not required to be run at boot.

```
$ mlxconfig -y -d <MST_DEVICE> set LLDP_NB_DCBX_P1=FALSE LLDP_NB_TX_MODE_P1=2 LLDP_NB_RX_MODE_P1=2 LLDP_NB_DCBX_P2=FALSE LLDP_NB_TX_MODE_P2=2 LLDP_NB_RX_MODE_P2=2
```

3. Reboot the host server for changes to take effect.

4. Verify LLDP and DCBx are disabled.

   Note: This command is informational and not required to be run at boot.

```
$ mlxconfig -d <MST _ DEVICE> q | grep LLDP
```

Example Output:

```
root@pfedlr750-24:~# mlxconfig -d mlx5 _ 0 q | grep LLDP
        LLDP _ NB _ DCBX _ P1                            False(0)
        LLDP _ NB _ RX _ MODE _ P1                        ALL(2)
        LLDP _ NB _ TX _ MODE _ P1                        ALL(2)
        LLDP _ NB _ DCBX _ P2                            False(0)
        LLDP _ NB _ RX _ MODE _ P2                        ALL(2)
        LLDP _ NB _ TX _ MODE _ P2                        ALL(2)
```

5. Disable RoCE slow restart.

   Note: Slow restart is a new feature introduced in recent firmware that allows for a lossy network configuration. Unfortunately, this impacts a lossless networking configuration's performance and should be disabled.

```
$ mlxreg -d ${pcidev} --reg _ name ROCE _ ACCL --set "roce _ adp _ retrans _ en=0x0,roce _ tx _ window _
en=0x0,roce _ slow _ restart _ en=0x0,roce _ slow _ restart _ idle _ en=0x0" -y
```

6. Configure the trust state for DSCP.

```
$ mlnx _ qos -i <INTERFACE> --trust dscp
```

7. Enable PFC on desired Priority.

   Note: Parameter --pfc takes eight comma separated values. The values passed are Boolean (0 or 1) to indicate whether PFC is enabled while the position indicates priority. The priorities are 0 - 7 from left to right. In this example we are enabling PFC on priority "3".

```
$ mlnx _ qos -i <INTERFACE> --pfc 0,0,0,1,0,0,0,0
```

8. Allocate memory to secondary buffer. See the following table for the <mem_size> value to substitute in the command.

Model	Recommended Buffer Size
ConnectX-5	130048
ConnectX-6	261120
ConnectX-6Dx	500688

```
$ mlnx _ qos -i <INTERFACE> --buffer _ size <mem _ size>,<mem _ size>,0,0,0,0,0,0
```

9. Map desired priority to newly created buffer 1.

   Note: Parameter --prio2buffer takes eight comma separated values. The values passed are the buffer to assign while the position indicates priority. The priorities are 0 - 7 from left to right. In this example we are assigning priority "3" to use the newly created buffer 1.

```
$ mlnx _ qos -i <INTERFACE> --prio2buffer 0,0,0,1,0,0,0,0
```

10. Set traffic class transmission algorithm and bandwidth allocation to match the default on the Mellanox switch.

    Note: Parameter --tsa takes eight comma separated values. The values passed are the egress scheduling method to assign while the position indicates priority. The priorities are 0 - 7 from left to right. In this example we are assigning priority "6" to use the strict scheduling method.

    Note: Parameter --tcbw takes eight comma separated values. The values passed are a percentage of the total bandwidth while the position indicates priority. The priorities are 0 - 7 from left to right. In this example we are mimicking the default bandwidth assignments done at a Mellanox switch adjusting for the strict scheduling on priority 6.

```
$ mlnx _ qos -i <INTERFACE> --tsa ets,ets,ets,ets,ets,ets,strict,ets --tcbw 14,15,14,15,14,14,0,14
```

11. Set the default RoCE mode for RNICs.

```
$ cma _ roce _ mode -d <ROCE _ DEVICE> -m 2
```

12. Verify that the RoCE mode is now set to RoCE v2.

    Note: This command is informational and not required to be run at boot.

```
$ cma _ roce _ mode -d <ROCE _ DEVICE>
```

13. Determine Type of Service (TOS) value to assign to RoCE v2 traffic by using the DSCP value and perform a bitwise left shift by 2 or multiply by 4.

- TOS is an 8-bit construct.
- DSCP is a 6-bit field residing in TOS (6 most significant bits).
- Priority is a 3-bit field residing in DSCP (3 most significant bits).

For example: DSCP 24: 011000 with corresponding TOS 96: 01100000

14. Assign RoCE v2 traffic to use derived TOS value.

```
$ cma _ roce _ tos -d <ROCE _ DEVICE> -t <TOS>
```

Summary of CLI Commands

```
mlxconfig -y -d <MST _ DEVICE> set LLDP _ NB _ DCBX _ P1=FALSE LLDP _ NB _ TX _ MODE _ P1=2 LLDP _ NB _ RX _ MODE _ P1=2
LLDP _ NB _ DCBX _ P2=FALSE LLDP _ NB _ TX _ MODE _ P2=2 LLDP _ NB _ RX _ MODE _ P2=2
mlnx _ qos -i <INTERFACE> --trust dscp
mlnx _ qos -i <INTERFACE> --pfc 0,0,0,1,0,0,0,0
mlnx _ qos -i <INTERFACE> --buffer _ size 130944,130944,0,0,0,0,0,0
mlnx _ qos -i <INTERFACE> --prio2buffer 0,0,0,1,0,0,0,0
mlnx _ qos -i <INTERFACE> --tsa ets,ets,ets,ets,ets,ets,strict,ets --tcbw 14,15,14,15,14,14,0,14
cma _ roce _ mode -d <ROCE _ DEVICE> -m 2
cma _ roce _ tos -d <ROCE _ DEVICE> -t <TOS>
```

Configuration PFC

- If the OpenFlex Data24 is the target NVMe over Fabrics storage device this section does not apply.
- Mellanox has configured ECN to be enabled by default. This section is required if it is desired to reconfigure the default setting for CNP network priority. Mellanox configures CNPs to priority 6 using a DSCP value of 48.
- Many of these commands are not persistent through reboot and will have to be scripted to run at every boot.

1. Ensure that the debugfs is mounted.

```
$ mount | grep debugfs
```

2. If the path isn't available, a mount is required.

```
$ mount -t debugfs none /sys/kernel/debug
```

3. Set CNP mode to manual configuration.

```
$ echo 0 > /sys/kernel/debug/mlx5/<PCIE _ DEVICE>/cc _ params/np _ cnp _ prio _ mode
```

4. Verify CNP mode is set to manual configuration.

```
$ cat /sys/kernel/debug/mlx5/<PCIE _ DEVICE>/cc _ params/np _ cnp _ prio _ mode
```

5. Change the CNP DSCP value.

```
$ echo <CNP _ DSCP> > /sys/kernel/debug/mlx5/<PCIE _ DEVICE>/cc _ params/np _ cnp _ dscp
```

6. Change the CNP Class of Service.

```
$ echo <CNP _ PRIORITY > > /sys/kernel/debug/mlx5/<PCIE _ DEVICE>/cc _ params/np _ cnp _ prio
```

7. Verify the values were set correctly.

```
$ cat /sys/kernel/debug/mlx5/<PCIE _ DEVICE>/cc _ params/np _ cnp*
```

Summary of CLI Commands

```
mount -t debugfs none /sys/kernel/debug
echo 0 > /sys/kernel/debug/mlx5/<PCIE _ DEVICE>/cc _ params/np _ cnp _ prio _ mode
cat /sys/kernel/debug/mlx5/<PCIE _ DEVICE>/cc _ params/np _ cnp _ prio _ mode
echo <CNP _ DSCP> > /sys/kernel/debug/mlx5/<PCIE _ DEVICE>/cc _ params/np _ cnp _ dscp
echo <CNP _ PRIORITY> > /sys/kernel/debug/mlx5/<PCIE _ DEVICE>/cc _ params/np _ cnp _ prio
cat /sys/kernel/debug/mlx5/<PCIE _ DEVICE>/cc _ params/np _ cnp*
```

Show Pertinent Network Counters

As of August 2020, the CNP counters on the Mellanox RNICs do not appear to be accurate.

1. Show PFC counters for a specific interface.

```
$ ethtool -S <INTERFACE> | grep 'pause.*prio\|prio.*pause'
```

2. Show Traffic Class counters on a specific interface.

```
$ ethtool -S <INTERFACE> | grep prio
```

Example Output:

```
root@pfedlr750-24:~# ethtool -S ens7f0np0 | grep prio
     rx_prio0_bytes: 0
     rx_prio0_packets: 0
     rx_prio0_discards: 0
     tx_prio0_bytes: 0
     tx_prio0_packets: 0
     rx_prio1_bytes: 0
     rx_prio1_packets: 0
     rx_prio1_discards: 0
     tx_prio1_bytes: 0
     tx_prio1_packets: 0
     rx_prio2_bytes: 0
     rx_prio2_packets: 0
     rx_prio2_discards: 0
     tx_prio2_bytes: 0
     tx_prio2_packets: 0
     rx_prio3_bytes: 0
     rx_prio3_packets: 0
     rx_prio3_discards: 0
     tx_prio3_bytes: 0
     tx_prio3_packets: 0
     .
     .
     .
     rx_prio7_buf_discard: 0
     rx_prio7_cong_discard: 0
     rx_prio7_marked: 0
```

3. Show drop, discard, pause, and abort counters on a specific interface. This command only shows non-zero counts.

```
$ grep "" /sys/class/infiniband/mlx5_*/ports/1/counters/* | grep -i -e drop -e dis -e err -e pau -e abort
| awk -F: '$NF != 0 { print $0 }'
```

4. Show ECN CNP counters on specific interface

```
$ grep "" /sys/class/infiniband/mlx5_*/ports/1/hw_counters/* | grep -i -e cnp -e ecn
```

Example Output:

```
root@pfedlr750-24:~# grep "" /sys/class/infiniband/mlx5_*/ports/1/hw_counters/* | grep -i -e cnp -e ecn
/sys/class/infiniband/mlx5_0/ports/1/hw_counters/np_cnp_sent:0
/sys/class/infiniband/mlx5_0/ports/1/hw_counters/np_ecn_marked_roce_packets:0
/sys/class/infiniband/mlx5_0/ports/1/hw_counters/roce_slow_restart_cnps:0
/sys/class/infiniband/mlx5_0/ports/1/hw_counters/rp_cnp_handled:0
/sys/class/infiniband/mlx5_0/ports/1/hw_counters/rp_cnp_ignored:0
/sys/class/infiniband/mlx5_1/ports/1/hw_counters/np_cnp_sent:0
/sys/class/infiniband/mlx5_1/ports/1/hw_counters/np_ecn_marked_roce_packets:0
/sys/class/infiniband/mlx5_1/ports/1/hw_counters/roce_slow_restart_cnps:0
/sys/class/infiniband/mlx5_1/ports/1/hw_counters/rp_cnp_handled:0
/sys/class/infiniband/mlx5_1/ports/1/hw_counters/rp_cnp_ignored:0
```

Configure Boot Time Scripts

The prior sections of this document provided instructions on how to configure RDMA capable Network Adapters for lossless networking. Many of the commands used do not persist during a reboot. This section details the method for configuring lossless settings to persist during a reboot cycle. To that end, a combination of a systemd service and a script that this service invokes is used to apply the lossless configuration to the host at boot time.

The following instructions are specific example using RedHat® based operating systems with systemd. If the system does not meet these specifications use the scripts below as a template to create operable services and scripts for the operating system.

1. Create a lossless configuration script named /usr/local/sbin/lossless_mlx.sh with the following contents:

Note: This is a script specific to Mellanox based network adapters. This script may require customization depending on the RNIC that is being used.

```
#!/bin/bash

PROGNAME="${0##*/}"
PATH=${PATH}:/root

# Mellanox specific lossless network configuration script

pcidev_from_nic ()
{
    local _nic=${1} _pcidev;
    if [ -z "${_nic}" ]; then
        echo "${FUNCNAME}(): NIC must be specified -- aborting" 1>&2;
        exit 1;
    fi;
    local _path="/sys/class/net/${_nic}/device";
    if [ ! -h "${_path}" ]; then
        echo "${FUNCNAME}(): \"${_path}\" not found -- aborting" 1>&2;
        exit 1;
    fi;
    _pcidev=`readlink ${_path} | sed -e 's/.*\///'`;
    if [ -n "${_pcidev}" ]; then
        echo ${_pcidev};
    else
        echo "none";
    fi
}

ibdev_from_nic ()
{
    local _nic=${1} _pcidev _lnks _ibdev;
    if [ -z "${_nic}" ]; then
        echo "${FUNCNAME}(): NIC must be specified -- aborting" 1>&2;
        exit 1;
    fi;
    _pcidev=`pcidev_from_nic ${_nic}`;
   if [ "${_pcidev}" == "none" -o -z "${_pcidev}" ]; then
        echo "${FUNCNAME}(): _pcidev=\"${_pcidev}\" not found -- aborting" 1>&2;
        exit 1;
    fi;
    _lnks=`find /sys/class/infiniband -maxdepth 1 -type l -print -exec readlink {} \; | grep ${_pcidev}`;
    _ibdev=`echo "${_lnks}" | head -1 | sed -e 's/.*\///'`;
if [ -n "${_ibdev}" ]; then
 echo ${_ibdev};
        else
        echo "none";
       echo ${_ibdev};
```

```
        else
        echo "none";
    fi
}

report_error ()
{
    if [ "${#}" -lt 1 ]; then
        echo "${FUNCNAME}(): requires at least 1 arguments" 1>&2;
        exit 1;
    fi;
    if [ -n "${PROGNAME}" ]; then
        echo "${PROGNAME}: ${*}" 1>&2;
        echo "${PROGNAME}: ${*}" > /dev/kmsg;
    else
        echo "${*}" 1>&2;
        echo "${*}" > /dev/kmsg;
    fi
}

set_roce ()
{
    local _ibdev="${1}" _mode="${2}" _tos="${3}";
    if [ "${#}" -lt 3 ]; then
        echo "${FUNCNAME}(): requires at least 3 arguments" 1>&2;
        exit 1;
    fi;
    mkdir -p /sys/kernel/config/rdma_cm/${_ibdev};
    [ -n "${_mode}" ] && echo "${_mode}" > /sys/kernel/config/rdma_cm/${_ibdev}/ports/1/default_roce_
mode;
    [ -n "${_tos}" ] && echo "${_tos}" > /sys/kernel/config/rdma_cm/${_ibdev}/ports/1/default_roce_tos;
    rmdir /sys/kernel/config/rdma_cm/${_ibdev}
}

set_global_pause ()
{
    local _nic="${1}";
    local _res _set=;
    if [ -z "${_nic}" ]; then
        echo "${FUNCNAME}(): NIC must be specified -- aborting" 1>&2;
        exit 1;
    fi;
    _res=`ethtool -a ${_nic}`;
    if ! echo "${_res}" | grep '^Autonegotiate:' | awk '{print $NF}' | grep -q 'off'; then
        _set="autoneg off";
    fi;
    if ! echo "${_res}" | grep '^RX:' | awk '{print $NF}' | grep -q 'off'; then
            [ -z "${_set}" ] && _set="rx off" || _set="${_set} rx off";
    fi;
 [ -z "${_set}" ] && _set="rx off" || _set="${_set} rx off";
    fi;
    if ! echo "${_res}" | grep '^TX:' | awk '{print $NF}' | grep -q 'off'; then
        [ -z "${_set}" ] && _set="tx off" || _set="${_set} tx off";
    fi;
```

```
        [ -n "${ _ set}" ] && ethtool -A "${ _ nic}" ${ _ set}
}
        [ -z "${ _ set}" ] && _ set="rx off" || _ set="${ _ set} rx off";
    fi;
    if ! echo "${ _ res}" | grep '^TX:' | awk '{print $NF}' | grep -q 'off'; then
        [ -z "${ _ set}" ] && _ set="tx off" || _ set="${ _ set} tx off";
    fi;
    [ -n "${ _ set}" ] && ethtool -A "${ _ nic}" ${ _ set}
}

NICS="enp134s0f0 enp47s0f0"
ECN=0
DIR=0

ROCE _ PRI=3
ROCE _ DSCP=24
ROCE _ PORT=4791
CNP _ PRI=6
CNP _ DSCP=48
ROCE _ TOS=$((ROCE _ DSCP<<2))

# Calculate the total available Rx buffer size for all ports on each RNIC
declare -A nics nics _ c
for nic in ${NICS}; do
    if [[ "${nic}" =~ ^enp ]]; then
        this _ nic=`echo ${nic} | cut -ds -f1`
    elif [[ "${nic}" =~ ^ens ]]; then
        this _ nic=`echo ${nic} | cut -df -f1`
    else
        report _ error "${PROGNAME}: \"${nic}\" unsupported NIC name -- aborting"
        exit 1
    fi
    if [ -z "${nics[${this _ nic}]}" ]; then
        nics[${this _ nic}]=0
        nics _ c[${this _ nic}]=1
    else
        ((nics _ c[${this _ nic}]++))
    fi
    rbt=${nics[${this _ nic}]}
    pcidev=`pcidev _ from _ nic ${nic}`
    if [ -z "${pcidev}" ]; then
        report _ error "${PROGNAME}: \"${nic}\" pcidev not found -- aborting"
        exit 1
    fi
    nic _ type=`lspci -s ${pcidev} | sed -e 's/^.*\[//' -e 's/]//' | tr -d ' '`
    if [ -z "${nic _ type}" ]; then
        report _ error "${PROGNAME}: \"${nic}\" type not parsed correctly -- aborting"
        exit 1
    fi
    case "${nic _ type}" in
        *ConnectX-5*)
            # Max = 130944
            # Max = 130944
            rbt=130048
```

```
            ;;
            # Max = 130944
            rbt=130048
            ;;
        *ConnectX-6*)
            # Max = 262016
            rbt=261120
            ;;
        *ConnectX-6Dx*)
# Max = 500976
            rbt=500688
            ;;
        *ConnectX-7*)
            # Max = 524288
            rbt=524160
            ;;
        *)
            report_error "${PROGNAME}: NIC type \"${nic_type}\" not handled -- defaulting to ConnectX-5
setting"
            rbt=130048
            ;;
    esac
    # Store in hash for later use
    if [ "${nics_c[${this_nic}]}" -eq 1 ]; then
        nics[${this_nic}]=${rbt}
    else
        nics[${this_nic}]=$((rbt / nics_c[${this_nic}]))
    fi
done

for nic in ${NICS}; do
    mlnx_qos -i ${nic} --trust dscp --dcbx os >/dev/null 2>&1
    [ "${?}" -ne 0 ] && report_error "mlnx_qos failed to set trust and dcbx"
    mlnx_qos -i ${nic} --pfc 0,0,0,1,0,0,0,0 2>&1
    [ "${?}" -ne 0 ] && report_error "mlnx_qos failed to set pfc"
done
for nic in ${NICS}; do
    # Disable global pause on NIC
    set_global_pause "${nic}"

    # Set the RoCE mode and ToS
    ibdev=`ibdev_from_nic ${nic}`
    if [ "${ibdev}" != "none" ]; then
        set_roce ${ibdev} "RoCE v2" ${ROCE_TOS}
    else
        report_error "ibdev_from_nic() returned \"none\" for \"${nic}\""
    fi

    if [[ "${nic}" =~ ^enp ]]; then
        this_nic=`echo ${nic} | cut -ds -f1`
           elif [[ "${nic}" =~ ^ens ]]; then
        this_nic=`echo ${nic} | cut -df -f1`
    else
```

```
        report _ error “${PROGNAME}: \”${nic}\” unsupported NIC name -- aborting”
        exit 1
    fi

    rbt=${nics[${this _ nic}]}
    # Set the Rx buffer size for 0 and 1
    mlnx _ qos -i ${nic} --buffer _ size ${rbt},${rbt},0,0,0,0,0,0 2>&1
    [ “${?}” -ne 0 ] && report _ error “mlnx _ qos failed to set buffer size”
    mlnx _ qos -i ${nic} --prio2buffer 0,0,0,1,0,0,0,0 2>&1
    [ “${?}” -ne 0 ] && report _ error “mlnx _ qos failed to set prio2buffer”
    mlnx _ qos -i ${nic} --dscp2prio=set,${ROCE _ DSCP},${ROCE _ PRI} 2>&1
    [ “${?}” -ne 0 ] && report _ error “mlnx _ qos failed to set dscp to prio map”
    mlnx _ qos -i ${nic} --tsa ets,ets,ets,ets,ets,ets,strict,ets --tcbw 14,15,14,15,14,14,0,14 2>&1
    [ “${?}” -ne 0 ] && report _ error “mlnx _ qos failed to set tsa and tcbw”

    pcidev=`pcidev _ from _ nic ${nic}`
    # If we decide to support “lossy” configurations, this will need to be revisited
    mlxreg -d ${pcidev} --reg _ name ROCE _ ACCL --get
    mlxreg -d ${pcidev} --reg _ name ROCE _ ACCL --set “roce _ adp _ retrans _ en=0x0,roce _ tx _ window _
en=0x0,roce _ slow _ restart _ en=0x0,roce _ slow _ restart _ idle _ en=0x0” -y
    [ “${?}” -ne 0 ] && report _ error “mlxreg failed to disable roce _ slow _ restart”
    # Turn on “lossy”
    # mlxreg -d ${pcidev} --reg _ name ROCE _ ACCL --set “roce _ adp _ retrans _ en=0x1,roce _ tx _ window _
en=0x0,roce _ slow _ restart _ en=0x1,roce _ slow _ restart _ idle _ en=0x0” -y

    if ((ECN)); then
        if ! grep -q debugfs /proc/mounts; then
            mount -t debugfs none /sys/kernel/debug
            [ “${?}” -ne 0 ] && report _ error “mount failed to mount debugfs”
        fi
        pushd /sys/kernel/debug/mlx5/${pcidev}/cc _ params/ >/dev/null
        echo 0 >  np _ cnp _ prio _ mode
        echo ${CNP _ DSCP} > np _ cnp _ dscp
        echo ${CNP _ PRI} > np _ cnp _ prio
        popd >/dev/null
fi

    if ((DIR)); then
        # Using direct connect, set speed to current speed and turn off autonegotiation
        eto=`ethtool ${nic}`
        speed=`echo “${eto}” | grep ‘Speed:’ | awk ‘{print $NF}’`
        nspeed=`echo ${speed} | tr -cd ‘0-9’`
        ethtool -s ${nic} speed ${nspeed} autoneg off
        [ “${?}” -ne 0 ] && report _ error “ethtool failed to fix speed and disable autoneg on \”${nic}\””
    fi

done

exit 0
```

2. Create a service control script /etc/systemd/system/lossless_mlx.service with the following contents:

```
# Service to configure lossless on system startup
[Unit]
Description=Script to set DSCP mode and default CMA TOS value
After=network-online.target
# Requires=openibd.service

[Service]
Type=simple
ExecStart=/usr/local/sbin/lossless_mlx.sh
TimeoutStartSec=0

[Install]
WantedBy=default.target
```

3. Register the new service.

```
$ systemctl enable lossless_mlx
```

4. Start the new service.

```
$ systemctl start lossless_mlx
```

5601 Great Oaks Parkway
San Jose, CA 95119, USA
www.westerndigital.com