Western Digital

Configuration Guide

Configuring a Broadcom Network Adapter for Lossless Networking with OpenFlex Platforms

Abstract

This configuration guide provides an overview of how to configure lossless Ethernet settings on Broadcom 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 Broadcom® based Ethernet Network Adapters can be broken down into the following steps:

1. Download and Install Software Driver and Firmware
2. Configure the Adapter into RDMA mode
3. Configure Additional Initiator Network Settings
4. Update Broadcom Firmware
5. Rebuild initrd, reboot, and Verify
6. Configure Priority Flow Control (PFC) & Explicit Congestion Notification (ECN)
7. Configure PFC Only
8. Show Pertinent Switch Counters
9. Configure Boot Time Scripts

Example Hardware Specifications

In this guide, a Broadcom P2100G, with device driver version 226.0.141.0 and firmware version 226.1.107.1 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	<DEVICE>	ens7f0p0	
RoCE Device Name	<ROCE_DEVICE>	bnxt_re0	
PFC Priority	<ROCE_PRI>	3	
PFC DSCP	<ROCE_DSCP>	24	
CNP Priority	<CNP_PRI>	6	
CNP DSCP	<CNP_DSCP>	48	
RNIC Port	<RNIC_PORT>	0	

Download and Install Software Driver and Firmware

Please wait until the Rebuild initrd, Reboot, and Verify Section before performing a host reboot. References to RHEL 8.7 and/or rhel8u7 may need to be adjusted for the version of RHEL being used.

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

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

2. Visit Broadcom's website using the following link to locate the model of RNIC being used.

   Note: The Adapter Drivers and Firmware can be found by searching the manufacturer's website: https://www.broadcom.com/products/ethernet-connectivity/network-adapters.

3. Select your device, select "Downloads", open the "Driver" and "Firmware" sections and download the following packages:

   a. Broadcom NetXtreme-E Linux Installer

   b. Broadcom NetXtreme-E Dual-port 100Gb Ethernet PCIe Adapter Firmware Image (Optional – Firmware is now bundled with the driver package)

4. Copy the Broadcom NetXtreme-E Linux Installer bundle to the host.

5. Install the pre-requisite packages.

```
$ dnf groupinstall -y "Infiniband"
$ dnf install -y libibverbs-devel make gcc kernel kernel-devel autoconf libtool rdma-core-devel rpm-build
kernel-rpm-macros elfutils-libelf-devel kernel-abi-whitelists rhel-system-roles tuned
```

6. Extract the driver bundle "tar.gz" file on the host server.

```
$ tar zxvf bcm_x.x.x.x.tar.gz
```

7. Change directory to the bnxtnvm/Linux/ directory.

```
$ cd bcm_x.x.x.x/bnxtnvm/Linux/
```

8. Install the bnxtnvm utility and use bnxtnvm to set the RDMA Mode.

```
$ rpm -ivh bnxtnvm-x.x.x.x.x86_64.rpm
```

9. Verify the bnxtnvm version.

```
linux bcm_226.1.107.1a # bnxtnvm version
Broadcom NetXtreme-C/E/S firmware update and configuration utility
Version v226.0.121.0
```

10. Change directory to the bnxtqos folder.

```
$ cd bcm_x.x.x.x/bnxtqos/x86_64/
```

11. Install the bnxtqos rpm

```
$ rpm -Uhv bnxtqos-x.x.x.x.x86_64.rpm
```

12. Verify the bnxtqos version.

```
$ bnxtqos version
```

Example Output:

```
linux ~ # bnxtqos version
Broadcom NetXtreme-E QoS Configuration Utility
Version 226.0.103.0 for Linux built Mar 17 2023 14:22:55
```

13. Change directory to the libbnxt_re source RPM location.

```
$ cd Linux/KMP-RoCE-Lib/KMP/Redhat/rhel8.7/
```

14. Rebuild the bnxt_re library source RPM.

```
$ rpmbuild --rebuild libbnxt_re-x.x.x.x-rhel8u7.src.rpm
```

15. Install the newly created bnxt_re library RPM

```
$ rpm -ivh /root/rpmbuild/RPMS/x86_64/libbnxt_re-x.x.x.x-rhel8u7.x86_64.rpm
```

16. Change directory to the bnxt_en driver source RPM.

```
$ cd Linux/KMP-L2-RoCE/KMP/Redhat/rhel8.7/
```

17. Rebuild the bnxt_en driver source RPM.

```
$ rpmbuild --rebuild bnxt_en-x.x.x-x.x.x.x.rhel8u7.src.rpm
```

18. Install the newly created driver RPM.

```
$ rpm -ivh /root/rpmbuild/RPMS/x86_64/kmod-bnxt_en-x.x.x-x.x.x.x.rhel8u7.x86_64.rpm
```

Configure the Adapter into RDMA Mode

Please wait until the Rebuild initrd, Reboot, and Verify Section before performing a host reboot. By default, Broadcom Network Adapters ship with RDMA functionality disabled. This section covers how to enable RDMA functionality using the bnxtnvm utility.

1.Enable RDMA Support using the following commands

```
$ bnxtnvm -dev=<DEVICE> setoption=support_rdma:<RNIC_PORT>#0x1
```

Example Output:

```
linux ~ # bnxtnvm -dev=ens7f0np0 setoption=support_rdma:0#0x1
support_rdma is set successfully
Please do the device reset to apply the configuration

linux ~ # bnxtnvm -dev=ens7f0np0 setoption=support_rdma:1#0x1
support_rdma is set successfully
Please do the device reset to apply the configuration
```

2. Check that RDMA Support is enabled using the following commands

```
$ bnxtnvm -dev=<DEVICE> getoption=support_rdma:<RNIC_PORT>
```

Example Output:

```
linux ~ # bnxtnvm -dev=ens7f0np0 getoption=support_rdma:0
support_rdma = Enabled

linux ~ # bnxtnvm -dev=ens7f0np0 getoption=support_rdma:1
support_rdma = Enabled
```

3. Disable DCBX Mode using the following commands.

```
$ bnxtnvm -dev=<DEVICE> setoption=dcbx_mode:<RNIC_PORT>#0x0
```

Example Output:

```
linux ~ # bnxtnvm -dev=ens7f0np0 setoption=dcbx_mode:0#0x0
dcbx_mode is set successfully
Please reboot the system to apply the configuration

linux ~ # bnxtnvm -dev=ens7f0np0 setoption=dcbx_mode:1#0x0
dcbx_mode is set successfully
Please reboot the system to apply the configuration
```

4. Check that DCBX Mode is disabled using the following command

```
$ bnxtnvm -dev=<DEVICE> getoption=dcbx_mode:<RNIC_PORT>
```

Example Output:

```
linux ~ # bnxtnvm -dev=ens7f0np0 getoption=dcbx_mode:0
dcbx_mode = Disabled

linux ~ # bnxtnvm -dev=ens7f0np0 getoption=dcbx_mode:1
dcbx_mode = Disabled
```

5. Disable LLDP Nearest Bridge using the following commands

```
$ bnxtnvm -dev=<DEVICE> setoption=lldp _ nearest _ bridge:<RNIC _ PORT>#0x0
```

Example Output:

```
linux ~ # bnxtnvm -dev=ens7f0np0 setoption=lldp _ nearest _ bridge:0#0x0
lldp _ nearest _ bridge is set successfully
Please reboot the system to apply the configuration

linux ~ # bnxtnvm -dev=ens7f0np0 setoption=lldp _ nearest _ bridge:1#0x0
lldp _ nearest _ bridge is set successfully
```

6. Check that LLDP Nearest Bridge is disabled using the following commands

```
$ bnxtnvm -dev=<DEVICE> getoption=lldp _ nearest _ bridge:<RNIC _ PORT>
```

Example Output:

```
linux ~ # bnxtnvm -dev=ens7f0np0 getoption=lldp _ nearest _ bridge:0
lldp _ nearest _ bridge = Disabled

linux ~ # bnxtnvm -dev=ens7f0np0 getoption=lldp _ nearest _ bridge:1
lldp _ nearest _ bridge = Disabled
```

Configure Additional Initiator Network Settings

Please wait until the Rebuild initrd, Reboot, and Verify Section before performing a host reboot.

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

```
# Broadcom nvmeOF
ib _ core
ib _ cm
ib _ uverbs
ib _ umad
rdma _ cm
rdma _ ucm
nvme
nvme _ rdma
nvme _ fabrics
bnxt _ re
```

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

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

Configure Additional Initiator Network Settings – Direct Connect

Please wait until the Rebuild initrd, Reboot, and Verify Section before performing a host reboot.

The following ethtool and Reed-Solomon FEC items are only required when directly connecting to the OpenFlex Data24. The ethtool command must be scripted to run at every boot as the setting is not persistent across boots. Adjust the Broadcom RNIC to enable the Reed-Solomon FEC setting w/ bnxtnvm, which is persistent across boots.

1. 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
```

2. Use bnxtnvm to check and set the Reed-Solomon FEC settings on each port of the RNIC:

```
$ bnxtnvm -dev=<DEVICE> getoption=fwd_err_correct:<port>
$ bnxtnvm -dev=<DEVICE> getoption=fwd_err_correct:<port>#0x2
$ bnxtnvm -dev=<DEVICE> getoption=fwd_err_correct:<port>
```

Example Output:

```
Port 0:

linux ~ # bnxtnvm -dev=ens7f0np0 getoption=fwd_err_correct:0
fwd_err_correct = Disabled

linux ~ # bnxtnvm -dev=ens7f0np0 setoption=fwd_err_correct:0#0x2
fwd_err_correct is set successfully
Please reboot the system to apply the configuration

linux ~ # bnxtnvm -dev=ens7f0np0 getoption=fwd_err_correct:0
fwd_err_correct = CL91_Reed_solomon

Port 1:

linux ~ # bnxtnvm -dev=ens7f0np0 getoption=fwd_err_correct:1
fwd_err_correct = Disabled

linux ~ # bnxtnvm -dev=ens7f0np0 setoption=fwd_err_correct:1#0x2
fwd_err_correct is set successfully
Please reboot the system to apply the configuration

linux ~ # bnxtnvm -dev=ens7f0np0 getoption=fwd_err_correct:1
fwd_err_correct = CL91_Reed_solomon
```

Update Broadcom Firmware

Please wait until the Rebuild initrd, Reboot, and Verify Section before performing a host reboot.

1. Change directory to the NVRAM Image folder

```
$ cd bcm_x.x.x.x/NVRAM_Images
```

2. Execute the firmware upgrade using the bnxtnvm command:

```
$ bnxtnvm -dev=<DEVICE> install <BCMxxxxx-RNICModel.pkg>
```

3. Follow the instructions to update specific devices or simply choose “All”

Example Output:

```
linux bcm _ 226.1.107.1a # bnxtnvm -dev=ens7f0np0 install NVRAM _ Images/BCM957508-P2100G.pkg

Broadcom NetXtreme-C/E/S firmware update and configuration utility version v226.0.121.0

NetXtreme-E Controller #1 at PCI Domain:0000 Bus:ca Dev:00
        Firmware on NVM - v224.1.102.0

NetXtreme-E Controller #1 will be updated to firmware version v226.1.107.1

Do you want to continue (Y/N)?Y

NetXtreme-C/E/S Controller #1 is being updated.............................................

Firmware update is completed.
A system reboot is needed for firmware update to take effect.
```

Rebuild initrd, reboot, and Verify

1. Rebuild initrd.

```
$ dracut -f
```

2. Perform a reboot of the server to complete the firmware upgrade process, RNIC configuration changes, and other settings to apply.

3. Verify the firmware was installed correctly on the host server using the following command

```
$ ./bnxtnvm/Linux/bnxtnvm pkgver
```

Example Output:

```
linux ~ # bnxtnvm pkgver

Device #1
---------
Device: ens7f0np0
Active Package version    : 226.1.107.1
Package version on NVM    : 226.1.107.1
Primary SBI Version       : 222.0.24.0
Secondary SBI Version     : 222.0.24.0
Primary SRT Version       : 226.0.145.1
Secondary SRT Version     : 226.0.145.1
Primary CRT Version       : 226.0.145.1
Secondary CRT Version     : 226.0.145.1
```

4. Verify the new module version(s)

```
$ modinfo -F version bnxt_en ; modinfo -F version bnxt_re
```

Example Output:

```
linux ~ #  modinfo -F version bnxt_en ; modinfo -F version bnxt_re
1.10.2-226.0.141.0
226.0.141.0

linux ~ # ibv_devices
    device                 node GUID
    ------              ----------------
    bnxt_re0            86160cfffe6f2760
    bnxt_re1            86160cfffe6f2761
```

5. Check ibv_devices and verify that the Broadcom Network Adapter is present in the list

```
$ ibv_devices
```

Example Output:

```
linux ~ # ibv_devices
    device                 node GUID
    ------              ----------------
    bnxt_re0            86160cfffe6f2760
    bnxt_re1            86160cfffe6f2761
```

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

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

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

Configure PFC & ECN

- If the OpenFlex Data24 is the target NVMe over Fabrics storage device and in RoCE mode this section does not apply.
- Broadcom provides a bnxt_setupcc.sh script that can set the majority of these settings within their source RPM. It can be used in lieu of most of these instructions.

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

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

1. Disable global pause.

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

```
$ ethtool -A <DEVICE> rx off tx off
```

2. Verify Global Pause is disabled.

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

```
$ ethtool -a <DEVICE>
```

Example Output:

```
linux ~ # ethtool -a ens7f0np0
Pause parameters for ens7f0np0:
Autonegotiate:   off
RX:              off
TX:              off
```

3. Configure PFC and ECN using bnxt_setup.sh.

The bnxt_setupcc.sh script will setup everything needed for the following:

- RoCE v2 Mode using the "-t 2" option
- Enable PFC + ECN (CC) using the "-m 3" option
- Enable PFC DSCP using the "-s <ROCE_DSCP>" option
- Enable CNP DSCP using the "-p <CNP DSCP>" option
- Set the PFC Priority to 3 using the "-r <ROCE_PRI>" option
- Set the CNP Priority to 6 using the "-c <CNP_PRIO>" option

```
$ bnxt _ setupcc.sh -d <ROCE _ DEVICE> -i <DEVICE> -t 2 -m 3 -s <ROCE _ DSCP> -p <CNP _ DSCP> -b 50 -r <ROCE _
PRI> -c <CNP _ PRIO>
```

Example Output:

```
linux ~ # bnxt _ setupcc.sh -d bnxt _ re0 -i ens7f0np0 -t 2 -m 3 -s 24 -p 48 -b 50 -r 3 -c 6
ENABLE _ PFC = 1 ENABLE _ CC = 1
ENABLE _ DSCP = 1 ENABLE _ DSCP _ BASED _ PFC = 1
L2 50 RoCE 50
Using Ethernet interface ens7f0np0   and RoCE interface bnxt _ re0
Settings for ens7f0np0:
 Supported ports: [ FIBRE ]
 Supported link modes:   40000baseCR4/Full
                         25000baseCR/Full
                         50000baseCR2/Full
                         100000baseCR4/Full
                         10000baseCR/Full
                         200000baseCR4/Full
 Supported pause frame use: Symmetric Receive-only
 Supports auto-negotiation: Yes
 Supported FEC modes: RS        BASER   LLRS
 Advertised link modes:  40000baseCR4/Full
                         25000baseCR/Full
                         50000baseCR2/Full
                         100000baseCR4/Full
                         10000baseCR/Full
                         200000baseCR4/Full
 Advertised pause frame use: No
 Advertised auto-negotiation: Yes
 Advertised FEC modes: RS       BASER   LLRS
 Speed: 100000Mb/s
 Lanes: 4
 Duplex: Full
 Auto-negotiation: on
 Port: Direct Attach Copper
 PHYAD: 0
 Transceiver: internal
 Supports Wake-on: d
 Wake-on: d
        Current message level: 0x000020c1 (8385)
                               drv rx _ err tx _ err hw
 Link detected: yes
driver: bnxt _ en
version: 1.10.2-226.0.141.0
```

```
firmware-version: 226.0.145.1/pkg 226.1.107.1
expansion-rom-version:
bus-info: 0000:ca:00.0
supports-statistics: yes
supports-test: yes
supports-eeprom-access: yes
supports-register-dump: yes
supports-priv-flags: yes
check if lldpad service is running : no action needed
Setting pfc/ets on ens7f0np0
Command executed successfully.
Command executed successfully.
Command executed successfully.
Command executed successfully.
Command executed successfully.
Command executed successfully.
Command executed successfully.
IEEE 8021QAZ ETS Configuration TLV:
  PRIO_MAP: 0:0 1:0 2:0 3:1 4:0 5:0 6:2 7:0
  TC Bandwidth: 50% 50% 0% 0% 0% 0% 0% 0%
  TSA_MAP: 0:ets 1:ets 2:strict 3:strict 4:strict 5:strict 6:strict 7:strict
IEEE 8021QAZ PFC TLV:
  PFC enabled: 3
IEEE 8021QAZ APP TLV:
  APP#0:
   Priority: 6
   Sel: 5
   DSCP: 48
APP#1:
   Priority: 3
   Sel: 5
   DSCP: 24

  APP#2:
   Priority: 3
   Sel: 3
   UDP or DCCP: 4791

TC Rate Limit: 100% 100% 100% 0% 0% 0% 0% 0%
Settings Default to use RoCE-v2
Setting up CC Settings
Setting up DSCP/PRI
Complete

linux ~ # bnxtqos -dev=ens7f0np0 get_qos
IEEE 8021QAZ ETS Configuration TLV:
  PRIO_MAP: 0:0 1:0 2:0 3:1 4:0 5:0 6:2 7:0
  TC Bandwidth: 50% 50% 0% 0% 0% 0% 0% 0%
  TSA_MAP: 0:ets 1:ets 2:strict 3:strict 4:strict 5:strict 6:strict 7:strict
IEEE 8021QAZ PFC TLV:
  PFC enabled: 3
IEEE 8021QAZ APP TLV:
```

```
  APP#0:
   Priority: 6
   Sel: 5
   DSCP: 48

  APP#1:
   Priority: 3
   Sel: 5
   DSCP: 24

  APP#2:
   Priority: 3
   Sel: 3
   UDP or DCCP: 4791

TC Rate Limit: 100% 100% 100% 0% 0% 0% 0% 0%
```

4. Verify the settings.

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

   ```
   $ bnxtqos -dev=<DEVICE> get _ qos
   ```

Example Output:

```
linux ~ # bnxtqos -dev=ens7f0np0 get _ qos
IEEE 8021QAZ ETS Configuration TLV:
  PRIO _ MAP: 0:0 1:0 2:0 3:1 4:0 5:0 6:2 7:0
  TC Bandwidth: 50% 50% 0% 0% 0% 0% 0% 0%
  TSA _ MAP: 0:ets 1:ets 2:strict 3:strict 4:strict 5:strict 6:strict 7:strict
IEEE 8021QAZ PFC TLV:
  PFC enabled: 3
IEEE 8021QAZ APP TLV:
  APP#0:
   Priority: 6
   Sel: 5
   DSCP: 48

  APP#1:
   Priority: 3
   Sel: 5
   DSCP: 24

  APP#2:
   Priority: 3
   Sel: 3
   UDP or DCCP: 4791

TC Rate Limit: 100% 100% 100% 0% 0% 0% 0% 0%
```

Summary of CLI Commands

```
ethtool -A <DEVICE> rx off tx off
ethtool -a <DEVICE>
bnxt _ setupcc.sh -d <ROCE _ DEVICE> -i <DEVICE> -t 2 -m 3 -s <PFC _ DSCP> -p <CNP _ DSCP> -b 50 -r <ROCE _ PRI>
-c <CNP _ PRI>
bnxtqos -dev=<DEVICE> get _ qos
```

Configure PFC Only

Broadcom provides a bnxt_setupcc.sh script that can set the majority of these settings within their source RPM. It can be used in lieu of most of these instructions. “bnxt_setupcc.sh”

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

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

1. Disable global pause.

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

```
$ ethtool -A <DEVICE> rx off tx off
```

2. Verify Global Pause is disabled.

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

```
$ ethtool -a <DEVICE>
```

Example Output:

```
linux ~ # ethtool -a ens7f0np0
Pause parameters for ens7f0np0:
Autonegotiate:   off
RX:              off
TX:              off
```

3. Configure PFC using bnxt_setup.sh.

The bnxt_setupcc.sh script will setup everything needed for the following:

- RoCE v2 Mode using the “-t 2” option
- Enable PFC using the “-m 1” option
- Enable PFC DSCP using the “-s <ROCE_DSCP>” option
- Set the PFC Priority to 3 using the “-r <ROCE_PRI>” option

```
$ bnxt _ setupcc.sh -d <ROCE _ DEVICE> -i <DEVICE> -t 2 -m 1 -s <ROCE _ DSCP> -b 50 -r <ROCE _ PRI>
```

Example Output:

```
linux ~ # bnxt _ setupcc.sh -d bnxt _ re0 -i ens7f0np0 -t 2 -m 1 -s 24 -b 50 -r 3
ENABLE _ PFC = 1 ENABLE _ CC = 0
ENABLE _ DSCP = 1 ENABLE _ DSCP _ BASED _ PFC = 1
L2 50 RoCE 50
Using Ethernet interface ens7f0np0   and RoCE interface bnxt _ re0
Settings for ens7f0np0:
 Supported ports: [ FIBRE ]
 Supported link modes:   40000baseCR4/Full
                         25000baseCR/Full
                         50000baseCR2/Full
                         100000baseCR4/Full
                         10000baseCR/Full
                         200000baseCR4/Full
 Supported pause frame use: Symmetric Receive-only
 Supports auto-negotiation: Yes
 Supported FEC modes: RS        BASER   LLRS
 Advertised link modes:  40000baseCR4/Full
                         25000baseCR/Full
                         50000baseCR2/Full
                         100000baseCR4/Full
                         10000baseCR/Full
                         200000baseCR4/Full
```

```
Advertised pause frame use: No
 Advertised auto-negotiation: Yes
 Advertised FEC modes: RS       BASER   LLRS
 Speed: 100000Mb/s
 Lanes: 4
 Duplex: Full
 Auto-negotiation: on
 Port: Direct Attach Copper
 PHYAD: 0
 Transceiver: internal
 Supports Wake-on: d
 Wake-on: d
        Current message level: 0x000020c1 (8385)
                               drv rx _ err tx _ err hw
 Link detected: yes
driver: bnxt _ en
version: 1.10.2-226.0.141.0
firmware-version: 226.0.145.1/pkg 226.1.107.1
expansion-rom-version:
bus-info: 0000:ca:00.0
supports-statistics: yes
supports-test: yes
supports-eeprom-access: yes
supports-register-dump: yes
supports-priv-flags: yes
check if lldpad service is running : no action needed
Setting pfc/ets on ens7f0np0
Command executed successfully.
Command executed successfully.
Command executed successfully.
Command executed successfully.
Command executed successfully.
Command executed successfully.
Command executed successfully.
IEEE 8021QAZ ETS Configuration TLV:
  PRIO _ MAP: 0:0 1:0 2:0 3:1 4:0 5:0 6:0 7:0
  TC Bandwidth: 50% 50% 0% 0% 0% 0% 0% 0%
  TSA _ MAP: 0:ets 1:ets 2:strict 3:strict 4:strict 5:strict 6:strict 7:strict
IEEE 8021QAZ PFC TLV:
  PFC enabled: 3
IEEE 8021QAZ APP TLV:
  APP#0:
   Priority: 3
   Sel: 5
   DSCP: 24
```

```
  APP#1:
   Priority: 3
   Sel: 3
   UDP or DCCP: 4791

TC Rate Limit: 100% 100% 0% 0% 0% 0% 0% 0%
Settings Default to use RoCE-v2
Setting up DSCP/PRI
Complete
```

Summary of CLI Commands

```
ethtool -A <DEVICE> rx off tx off
bnxt_setupcc.sh -d <ROCE_DEVICE> -i <DEVICE> -t 2 -m 1 -s <ROCE_DSCP> -b 50 -r <ROCE_PRI>
bnxtqos -dev=<DEVICE> get_qos
```

Show Pertinent Network Counters

As of August 2020, the Traffic Class counters on the Broadcom RNICs do not appear to be accurate

1. Show PFC counters for a specific interface.

```
$ ethtool -S <DEVICE> | grep pfc.*pri3
```

Example Output:

```
linux ~ # ethtool -S ens7f0np0 | grep pfc.*pri3
     rx_pfc_ena_frames_pri3: 0
     tx_pfc_ena_frames_pri3: 0
     pfc_pri3_rx_transitions: 0
     pfc_pri3_tx_transitions: 0
```

2. Show Traffic Class counters on a specific interface.

```
$ ethtool -S <DEVICE> | grep pri
```

Example Output:

```
linux ~ # ethtool -S ens7f0np0 | grep pri
     rx_pfc_ena_frames_pri0: 0
     rx_pfc_ena_frames_pri1: 0
     rx_pfc_ena_frames_pri2: 0
     rx_pfc_ena_frames_pri3: 0
     rx_pfc_ena_frames_pri4: 0
     rx_pfc_ena_frames_pri5: 0
     rx_pfc_ena_frames_pri6: 0
     rx_pfc_ena_frames_pri7: 0
     tx_pfc_ena_frames_pri0: 0
     tx_pfc_ena_frames_pri1: 0
     tx_pfc_ena_frames_pri2: 0
     tx_pfc_ena_frames_pri3: 0
     tx_pfc_ena_frames_pri4: 0
     tx_pfc_ena_frames_pri5: 0
     tx_pfc_ena_frames_pri6: 0
     tx_pfc_ena_frames_pri7: 0
     pfc_pri0_rx_transitions: 0
     pfc_pri1_rx_transitions: 0
     pfc_pri2_rx_transitions: 0
     pfc_pri3_rx_transitions: 0
     pfc_pri4_rx_transitions: 0
```

```
     pfc _ pri5 _ tx _ transitions: 0
     pfc _ pri6 _ tx _ transitions: 0
     pfc _ pri7 _ tx _ transitions: 0
     rx _ bytes _ pri0: 122980
     rx _ bytes _ pri1: 122980
     rx _ bytes _ pri2: 122980
     rx _ bytes _ pri3: 122980
     rx _ bytes _ pri4: 122980
     rx _ bytes _ pri5: 122980
     rx _ bytes _ pri6: 19920
     rx _ bytes _ pri7: 0
     rx _ packets _ pri0: 1802
     rx _ packets _ pri1: 1802
     rx _ packets _ pri2: 1802
     rx _ packets _ pri3: 1802
     rx _ packets _ pri4: 1802
     rx _ packets _ pri5: 1802
     rx _ packets _ pri6: 60
     rx _ packets _ pri7: 0
     tx _ bytes _ pri0: 2434
     tx _ bytes _ pri1: 2434
     tx _ bytes _ pri2: 2434
     tx _ bytes _ pri3: 2434
     tx _ bytes _ pri4: 2434
     tx _ bytes _ pri5: 2434
     tx _ bytes _ pri6: 0
     tx _ bytes _ pri7: 0
     tx _ packets _ pri0: 41
     tx _ packets _ pri1: 41
     tx _ packets _ pri2: 41
     tx _ packets _ pri3: 41
     tx _ packets _ pri4: 41
     tx _ packets _ pri5: 41
     tx _ packets _ pri6: 0
     tx _ packets _ pri7: 0
```

3. Show drop, discard, pause, and abort counters on a specific interface.

```
Linux ~ # grep "" /sys/kernel/debug/bnxt _ re/bnxt _ re*/info | grep -i -e drop -e dis -e err -e pau -e
abort -e IBDEV | awk '$NF != 0 || $2 == "IBDEV" { print $0 }'
/sys/kernel/debug/bnxt _ re/bnxt _ re0/info:=====[ IBDEV bnxt _ re0 ]=============================
/sys/kernel/debug/bnxt _ re/bnxt _ re0/info:        rx _ roce _ discard _ pkts: 69
/sys/kernel/debug/bnxt _ re/bnxt _ re0/info:        seq _ err _ naks _ rcvd: 5
/sys/kernel/debug/bnxt _ re/bnxt _ re0/info:        res _ oos _ drop _ count: 8
/sys/kernel/debug/bnxt _ re/bnxt _ re1/info:=====[ IBDEV bnxt _ re1 ]=============================
/sys/kernel/debug/bnxt _ re/bnxt _ re1/info:        rx _ roce _ error _ pkts: 52
/sys/kernel/debug/bnxt _ re/bnxt _ re1/info:        rx _ roce _ discard _ pkts: 89
/sys/kernel/debug/bnxt _ re/bnxt _ re1/info:        seq _ err _ naks _ rcvd: 7
/sys/kernel/debug/bnxt _ re/bnxt _ re1/info:        unrecoverable _ err: 2
/sys/kernel/debug/bnxt _ re/bnxt _ re1/info:        res _ rx _ range _ err: 2
/sys/kernel/debug/bnxt _ re/bnxt _ re1/info:        res _ oos _ drop _ count: 10
```

4. Show ECN CNP counters on specific interface

```
cat /sys/kernel/debug/bnxt_re/bnxt_re*/info | grep -i -e cnp
```

Example Output:

```
linux ~ # cat /sys/kernel/debug/bnxt_re/bnxt_re*/info | grep -i -e cnp
 CNP Tx Pkts: 0
 CNP Rx Pkts: 0
 CNP Tx Pkts: 0
 CNP Rx Pkts: 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 instructions below are specific example using RedHat® based operating systems with systemd. If your system does not meet these specifications use the scripts below as a template to create operable services and scripts for your operating system

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

Note: This is a script specific to Broadcom based network adapters. This script may require customization depending on the RNIC that is being used. Pay particular attention to the bold comments for instruction on how to customize

```
#!/bin/bash

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

# Broadcom specific lossless network configuration script

BNXT_SETUPCC="/usr/bin/bnxt_setupcc.sh"

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";
    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="${ _ set} rx off";
   fi;
    if ! echo "${ _ 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="ens7f0np0 ens7f1np1"
ECN=0
DIR=1

ROCE _ PRI=3
ROCE _ DSCP=24
CNP _ PRI=6
CNP _ DSCP=48
ROCE _ TOS=96

if [ ! -x "${BNXT _ SETUPCC}" ]; then
    report _ err "${PROGNAME}: \"${BNXT _ SETUPCC}\" NOT found and is required"
    exit 1
fi

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 ((ECN)); then
        res=`${BNXT_SETUPCC} -i ${nic} -m 3 -s ${ROCE_DSCP} -p ${CNP_DSCP} -b 50 -t 2 -r ${ROCE_PRI} -c ${CNP_
PRI} 2>&1`
    else
        res=`${BNXT_SETUPCC} -i ${nic} -m 1 -s ${ROCE_DSCP} -b 50 -t 2 -r ${ROCE_PRI} 2>&1`
    fi
    if [ "${?}" -ne 0 ]; then
        report_error "${res}"
    else
        echo "bnxtqos -dev=${nic} get_qos"
        bnxtqos -dev=${nic} get_qos
    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
    fi

done

exit 0
```

2. Set the permissions on the lossless_bnxt.sh script to 755.

```
$ chmod 755 /usr/local/sbin/lossless_bnxt.sh
```

3. Create a service control script /etc/systemd/system/lossless_bnxt.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

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

[Install]
WantedBy=default.target
```

4. Set the permissions on the lossless_bnxt.service file to 644.

```
$ chmod 644 /etc/system/system/lossless_bnxt.service
```

5. Register and start the new service.

```
# systemctl enable --now lossless_bnxt
```


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