Pica8 White Paper

PicOS® Lossless Network for RDMA White Paper

1. RDMA Over Converged Ethernet (RoCE)

1.1 Principles of RDMA

In traditional TCP/IP communication, data transmission between two hosts involves the following core steps:

On the sender side:

- **User-to-kernel memory copy**: Application data is copied from user space to the socket send buffer in kernel space.
- **Protocol stack encapsulation**: The kernel protocol stack adds TCP/IP headers layer by layer to form a complete packet.
- **DMA transfer**: The data is transferred from the kernel buffer to the NIC queue via mechanisms like zero-copy (e.g., sendfile) or direct NIC DMA.

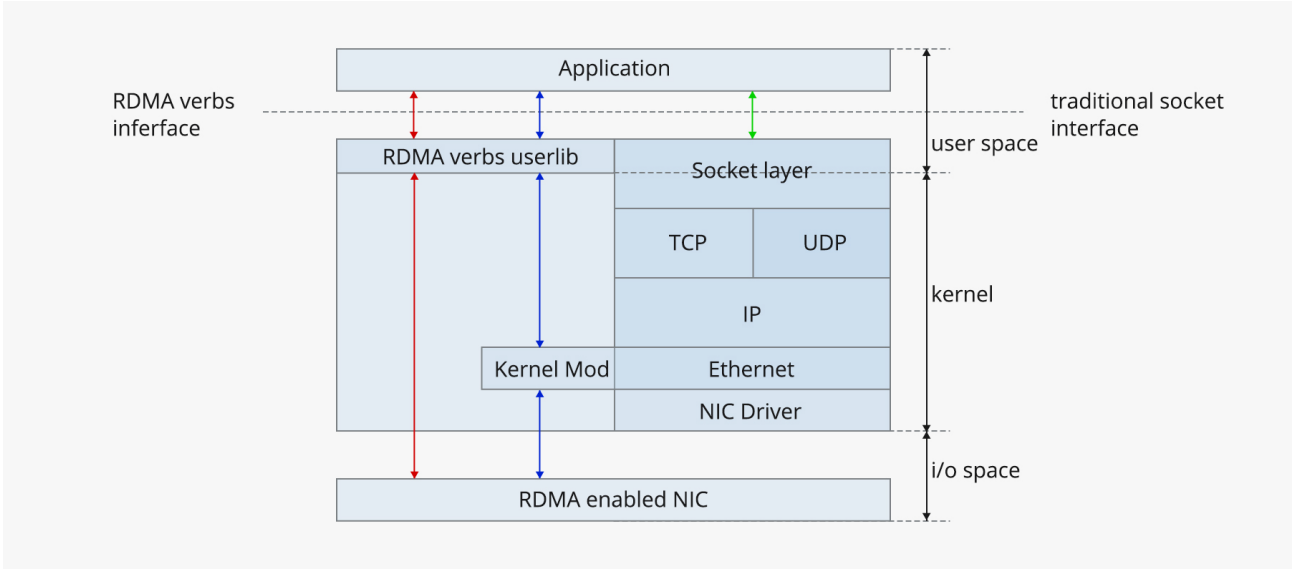
On the receiver side:

- **Hardware interrupt handling**: Upon packet arrival, the NIC triggers an interrupt, and the kernel uses the NAPI mechanism to store the data into the DMA ring buffer.
- **Protocol stack processing**: The kernel decapsulates the packet headers, performs checksum verification, and handles tasks such as out-of-order packet reassembly.
- **Ready data copy**: Complete message data is copied from the kernel socket receive buffer to user space memory.
- **Context switch**: The application is awakened from kernel mode to user mode to process the received data.

Performance bottlenecks include:

- Four context switches (two for send, two for receive)
- At least two full memory copies (user space ⇄ kernel space)
- Latency introduced by protocol stack processing (e.g., fragmentation/reassembly, ACK handling)
- Overhead from interrupt handling and software interrupt scheduling

This architecture causes significant performance degradation in high-speed networking environments. This is where RDMA (Remote Direct Memory Access)—with kernel bypass and zero-copy mechanisms—and kernel stack optimization techniques like XDP and eBPF bring their value.



1.2 Overview of RoCE (RDMA over Converged Ethernet)

RoCE is an Ethernet-based Remote Direct Memory Access (RDMA) protocol that enables applications to directly read and write memory between hosts without operating system intervention. This significantly reduces communication latency and CPU overhead. By offloading data transfer tasks to the network adapter, RoCE delivers higher throughput and lower system resource consumption.

The second-generation protocol, RoCEv2, builds upon the original architecture by introducing the UDP/IP protocol stack. This enhancement enables Layer 3 routing capability, making RoCEv2 well-suited for large-scale deployments in modern data centers.

Advantages of Broadcom RoCEv2 Solution

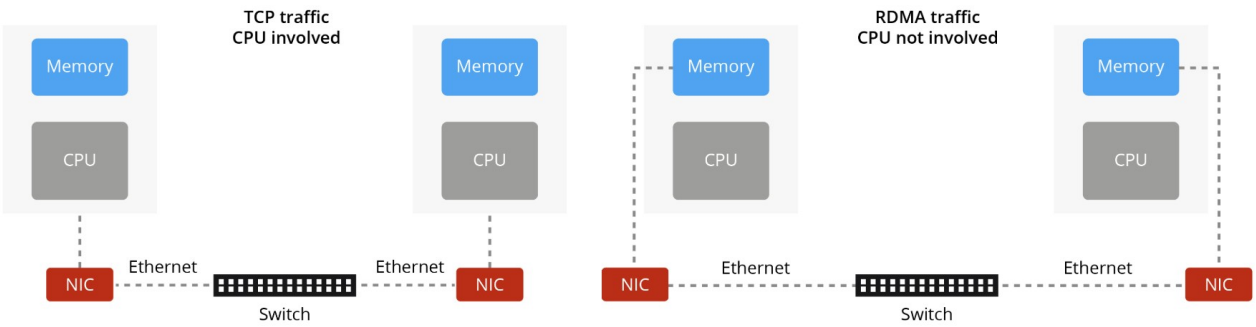
Broadcom's Ethernet adapters offer hardware-level support for RoCEv2, enabling high-performance, low-latency network communication across a wide range of scenarios, including AI/ML workloads, distributed storage, and high-performance computing (HPC). Key benefits include:

- **High Throughput**: Leverages NIC hardware acceleration with support for multi-queue, high-concurrency data transfers.
- **Low Latency**: Bypasses the kernel protocol stack, significantly reducing end-to-end latency for applications.
- **Low CPU Utilization**: The communication data path is offloaded from the CPU, freeing up host processing resources.

1.3 Three Key Features of RoCEv2

- **Ethernet-Based Routed Communication:** RoCEv2 utilizes UDP/IP encapsulation, enabling it to be routed across Layer 3 networks. This makes it well-suited for large-scale data center deployments based on Spine-Leaf architectures and cross-switch communication.
- **IP QoS Assurance:** RoCEv2 supports DiffServ (DSCP) and VLAN priority (802.1p), allowing network devices to perform priority-based traffic scheduling and control. This ensures stable performance for latency-sensitive and mission-critical workloads.
- **IP-Based Congestion Control:** RoCEv2 leverages ECN (Explicit Congestion Notification) to enable sender-side rate adjustment based on real-time network feedback. This mechanism effectively mitigates congestion and improves overall transmission efficiency.

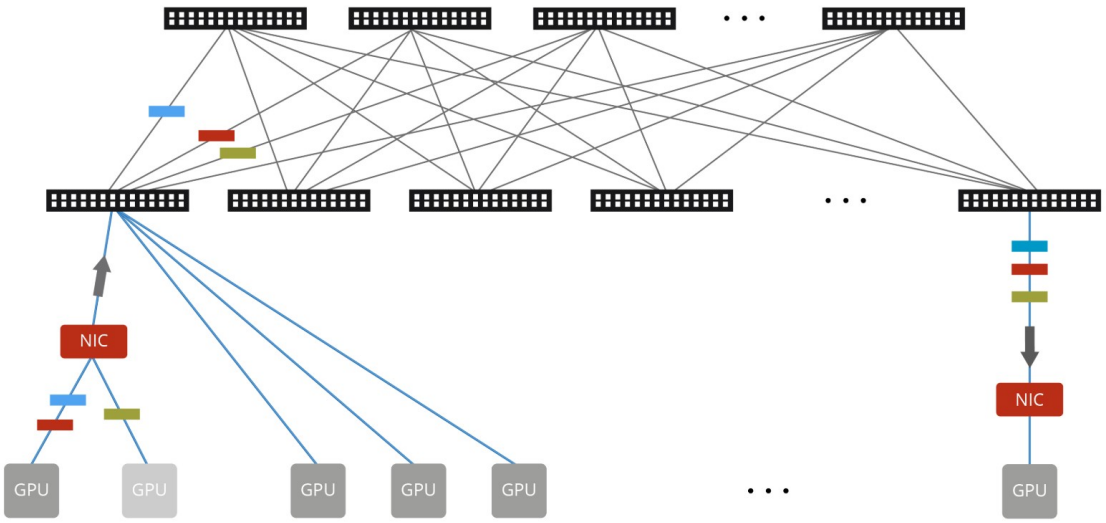
To further enhance a RoCEv2 network architecture, DCB (Data Center Bridging) configurations can be integrated to enable coordinated optimization of priority-based flow control (PFC), ECN-based congestion management, and queue scheduling strategies.



RoCEv2 introduces a network congestion detection and control mechanism based on Explicit Congestion Notification (ECN) and Congestion Notification Packets (CNP) to dynamically adjust data transmission rates and alleviate congestion along the transport path.

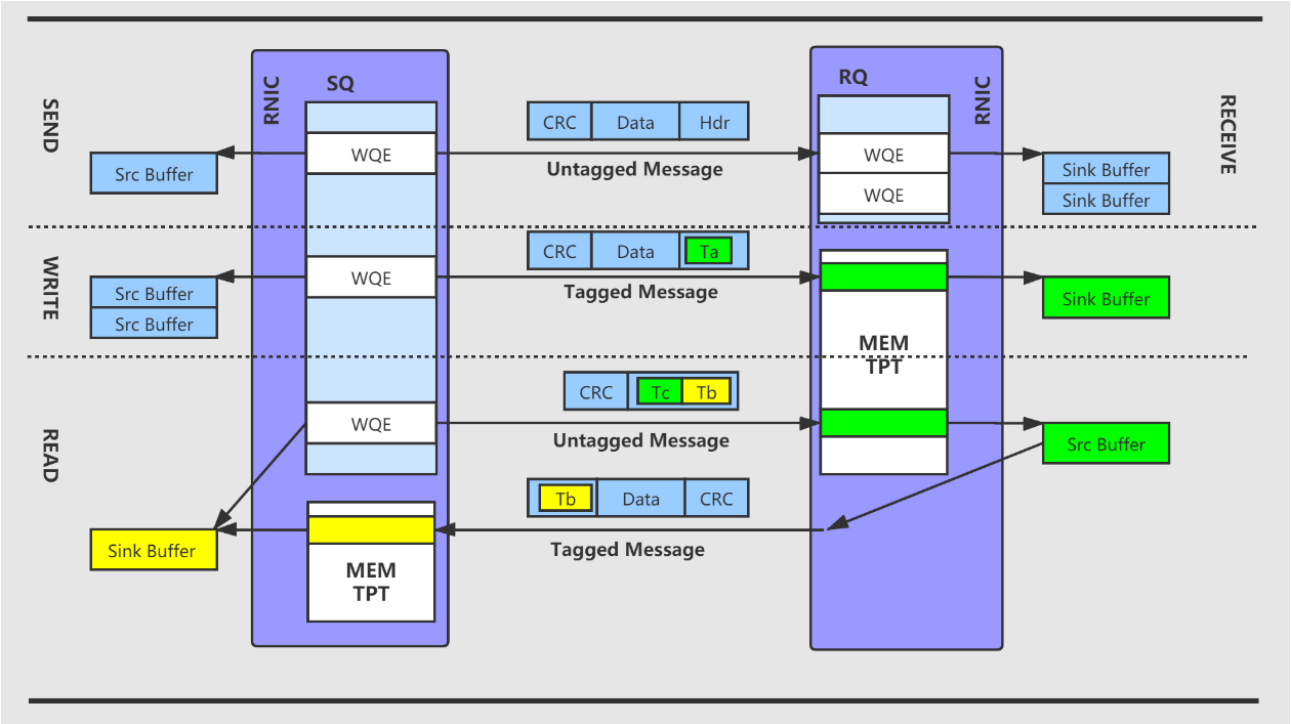
Core Mechanism:

- **ECN Marking (Congestion Experienced, CE):** When a switch or the receiver-side RNIC detects network congestion, it sets the ECN field in the IP header of a packet to indicate that congestion has occurred.
  - **Who marks it?** Primarily the switches (or ECN-capable routers) in the network.
  - **How is it marked?** The ECN field in the IP header is set to 11 (CE) to indicate congestion.
  - **When is it marked?** When the internal buffer or queue length of the switch exceeds a predefined threshold, signaling congestion on the link.
- **Congestion Notification Packet (CNP):** Upon receiving a packet with an ECN mark, the receiver-side RNIC generates a special CNP and sends it back to the sender to request a reduction in transmission rate.
  - **Who sends it?** The receiver's RNIC, upon detecting ECN-marked packets.
  - **Who receives it?** The original sender's RNIC.
  - **Purpose:** To notify the sender that congestion has occurred along the transmission path and that it should reduce its sending rate.



The RDMA data flow typically involves the following key steps:

- **Application Memory Registration:** The application registers a user-space memory region with the RDMA driver using an RDMA API (e.g., ibv_reg_mr). This grants the RDMA Network Interface Card (RNIC) direct access to the memory via DMA and sets local and remote access control policies.
- **RDMA Connection Establishment:** A connection is established using a Queue Pair (QP) and an RDMA transport protocol such as InfiniBand, RoCE, or iWARP.
- **Data Transmission:**
  - The sender-side NIC performs a direct memory read (via DMA) from the registered local memory.
  - Data is transferred over the PCIe bus to the local RDMA NIC (RNIC).
  - The NIC transmits the data over the InfiniBand/RoCE/iWARP network directly to the remote RNIC.
  - The remote RNIC writes the data via PCIe into the receiver's registered memory region.
- **Completion and Notification:** Once the transmission is complete, the RDMA device updates the relevant status. The application can immediately access the data in memory without additional copying.



1.4 Recommended RoCE RDMA Configuration Guidelines (Example)

Item	Recommended Setting
RoCE Mode	Use RoCEv2 (UDP encapsulation, ECN supported)
ECN	Enable ECN marking on switches when buffer thresholds are exceeded
CNP Response	Driver supports it by default; DCQCN needs to be enabled
PFC	Enable PFC only for RDMA-specific traffic classes (e.g., TC1)
Buffer	Allocate at least 512KB to 1MB per enabled RDMA traffic class (TC)
MTU	Recommend enabling Jumbo Frames (e.g., MTU 9000)
DSCP / PCP	Ensure consistent mapping: DSCP → TC → Priority → PFC

2. RoCE Deployment on FS Switches

> ⓘ This solution is designed to support low-latency, lossless Ethernet environments based on RoCEv2, especially for application scenarios requiring high bandwidth and low latency—such as AI training, high-performance computing (HPC), and distributed storage.

2.1 Device Selection Guidelines (Recommended Models)

Broadcom-based Switching ASICs Ensure RoCE Feature Support

- Support for lossless Ethernet technologies, including IEEE 802.1Qbb PFC, ECN, and QoS scheduling
- Built-in support for ECMP, DCQCN, and other congestion control enhancements at the ASIC level
- Broad compatibility with mainstream RDMA NIC vendors such as Mellanox, Broadcom NetXtreme-E and Intel

PicOS: A Flexible and Stable Network OS

- Supports hybrid CLI (Cisco-like syntax) and Linux Shell for versatile operations

- Open network OS with support for Ansible, ZTP, SNMP, and custom scripting
- Unified configuration of PFC, ECN, and priority queues, ideal for large-scale RDMA traffic environments

Model	Port Configuration	Switching Capacity	Feature Support	Recommended Use Cases
N8650-64OD	64 × 800G OSFP	51.2 Tbps	RoCEv2, GLB/DLB	Cloud computing, AI/ML clusters
N8650-32OD	32 × 800G OSFP	25.6 Tbps	RoCEv2, GLB/DLB	Cloud computing, AI/ML clusters
N8520-32D	32 × 400G QSFP-DD	12.8 Tbps	RoCEv2, MLAG, EVPN-VXLAN	Edge/DC interconnect (DCI)
N8610-32D	32 × 400G QSFP-DD	12.8 Tbps	RoCEv2, DLB	Deep learning networks
N5680-32D	32 × 400G QSFP-DD	12.8 Tbps	PFC, ECN, QoS, M-LAG	AI clusters, large-scale RoCE deployments
N9550-64D	64 × 400G QSFP-DD	25.6 Tbps	RoCEv2, DLB	Hyperscale data centers, RDMA storage clusters
N8610-64D*	64 × 400G QSFP-DD	25.6 Tbps	RoCEv2, DLB	Hyperscale data centers, RDMA storage clusters
N8550-24C8D	24 × 200G QSFP56 + 8 × 400G QSFP-DD	2 Tbps	RoCEv2, MLAG, EVPN-VXLAN	Edge/DC interconnect (DCI)
N5580-48Y8C	48 × 25G SFP28 + 8 × 100G QSFP28	2 Tbps	EVPN-VXLAN, MLAG, Multihoming	Medium-scale cluster edge access

*** Indicates the device is under development**

Data Rate	Model & Features				
800G	N8650-64OD (2U) • 64 x 800G OSFP • TH5 BCM78900 	N8650-32OD (1U) • 32 x 800G OSFP • TH5 BCM78902 			
400G	N9550-64D (4U) • 64 x 400G QSFP-DD • TH4 BCM56990 	N8610-64D* (2U) • 64 x 400G QSFP-DD • TH4 BCM56990 	N5680-32D (1U) • 32 x 400G QSFP-DD • TH3 BCM56980 	N8610-32D (1U) • 32 x 400G QSFP-DD • TH4 BCM56993 	N8520-32D (1U) • 32 x 400G QSFP-DD • TD4-X11 BCM56880 

100/200G	N8550-24CD8D (1U) • 24 x 200G QSFP56, 8 x 400G QSFP-DD • TD4-X9 BCM56780 	N5630-64C (2U) • 64 x 100G QSFP28 • TH2 BCM56970 	N5580-32C (1U) • 32 x 100G QSFP28 • TD3-X7 BCM56870 	N8560-32C (1U) • 32 x 100G QSFP28 • TD3-X7 BCM56870 	
10/25G	N5580-48Y8C (1U) • 48 x 25G SFP28, 8 x 100G QSFP28, 2 x 10G SFP+ • TD3-X7 BCM56873 	N5570-48S6C (1U) • 48 x 10G SFP+, 6 x 100G QSFP28 • TD3-X5 BCM56771 	N5850-48X6C (1U) • 48 x 10G RJ45, 6 x 100G QSFP28 • TD3-X5 BCM56771 	N5850-48S6Q (1U) • 48 x 10G SFP+, 6 x 40G QSFP+ • TD2+ BCM56864 	

2.2 Switch Configuration Recommendations for RoCE Networks

Feature	Recommended Setting	Description
Port MTU	9216	Must be larger than RoCE packet size to prevent fragmentation
RoCE Traffic Priority	Priority 3	Recommended fixed priority for PFC traffic control
Enable PFC	priority-flow-control priority 3 enable	Enable PFC on the specified priority; ensure consistency with server settings
Enable ECN	ecn enable	Enable ECN marking for congestion feedback
DSCP/COS Mapping	DSCP 4791 → CoS 3 → TC 3	Ensure RoCEv2 traffic matches the configured lossless queue
QoS Queue Binding	Bind each priority to a dedicated queue	Guarantee isolation and determinism for RoCE traffic
Lossless Queue Routing	Based on UDP port 4791 or VLAN/QoS tag	Ensure RoCE traffic enters the lossless priority queue
Broadcast/Unknown Multicast	Disable if unnecessary	Reduce interference and preserve RDMA path stability

2.3 RoCE Configuration Steps on FS Switches (Example: N8560/N9550 Series)

Step 1: Enable PFC (Priority Flow Control)

Configure PFC on all relevant ports of the switch to prevent packet loss by pausing traffic during periods of congestion.

The following steps complete the configuration:

- Create a PFC configuration profile named pfc1.
- Apply the PFC profile to port te-1/1/1.

```
admin@PICOS# set class-of-service pfc-profile pfc1
admin@PICOS# set class-of-service interface te-1/1/1 pfc-profile pfc1
admin@PICOS# commit
```

- Display service statistics for the specified interface.
  In the PFC framework, classes 0 through 7 correspond to the 802.1p priority values. If port te-1/1/1 receives a PFC frame, the RxPFC counter will increase by 1. If port te-1/1/1 sends a PFC frame, the TxPFC counter will increase by 1.

```
admin@PICOS# run show class-of-service interface te-1/1/1
Interface : te-1/1/1
802.1P Priority Flow Control RxPFC TxPFC
0 true 0 500
1 true 0 0
2 true 0 71
3 true 0 0
4 true 0 0
5 true 0 0
6 true 0 102
7 true 0 0
trust mode : ieee-802.1
Default ieee-802.1 : 0
Default dscp : 0
Default inet-precedence : 0
Local-priority Queue-Schedule Code-points
0 SP,0kbps
1 SP,0kbps
2 SP,0kbps
3 SP,0kbps
4 SP,0kbps
5 SP,0kbps
6 SP,0kbps
7 SP,0kbps
```

Step 2: Configure PFC Buffers

Once PFC is enabled, default buffer space is allocated for priority queues. To make optimal use of available buffer resources, you can fine-tune buffer thresholds for specific queues as needed.

The following commands complete the configuration:

- Set the MTU of interface te-1/1/1 to 9000.
- Set the guaranteed buffer limit for PFC queue 3 on interface te-1/1/1 to 24,000 cells.
- Set the static threshold for the shared buffer of PFC queue 1 on interface te-1/1/1 to 10,000 cells.
- Set the offset for the shared buffer of PFC queue 1 on interface te-1/1/1 to 3,000 cells.

```
admin@PICOS# set interface gigabit-ethernet xe-1/1/1 mtu 9000
admin@PICOS# set interface gigabit-ethernet te-1/1/1 ethernet-switching-options buffer ingress-queue 3 guaranteed 24000
```

```
admin@PICOS# set interface gigabit-ethernet te-1/1/1 ethernet-switching-options buffer ingress-queue 1 threshold 10000
admin@PICOS# set interface gigabit-ethernet te-1/1/1 ethernet-switching-options buffer ingress-queue 1 reset-offset 3000
admin@PICOS# commit
```

Step 3: Configure PFC Watchdog

The PFC Watchdog helps detect and recover from PFC deadlock conditions.

Use the following commands to complete the configuration:

- Enable PFC Watchdog on queue 5 of interface te-1/1/1.
- Set the deadlock detection interval to 10 × 100 ms, where 100 ms is the default granularity of the detection timer.
- Set the recovery timeout to 10 × 100 ms, where 100 ms is the default granularity of the recovery timer.

```
admin@PICOS# set class-of-service interface te-1/1/1 pfc-watchdog code-point 5 enable true
admin@PICOS# set class-of-service pfc-watchdog code-point 5 detect-interval 10
admin@PICOS# set class-of-service pfc-watchdog code-point 5 restore-interval 10
admin@PICOS# commit
```

After configuration:

- Use the command show pfc-watchdog config to view the PFC Watchdog configuration.

```
admin@PICOS# run show pfc-watchdog config
 PORT ACTION QUEUE DETECTION TIME RESTORATION TIME
---------- ----------- ------------ ---------------- ------------------
te-1/1/1 forward 5 1000 1000
```

- Use the command show pfc-watchdog statistics to view statistics, including:
  - The number of PFC pause storms detected and recovered on each queue of the interface.
  - The number of packets dropped due to PFC deadlock.

```
admin@PICOS# run show pfc-watchdog stats
 QUEUE STATUS STORM DETECTED/RESTORED TX OK/DROP TX LAST OK/DROP
------------ ----------- ------------------------ ---------------- -----------------
te-1/1/1:5 stormed 9/8 82072626556/0 32053822365/0
```

Step 4: Configure ECN (Explicit Congestion Notification)

Users should continuously monitor network performance and ECN marking rates. ECN threshold values should be adjusted dynamically as needed to adapt to changing network conditions.

Use the following commands to complete the configuration:

- Enable WRED on queue 0 of interface te-1/1/1.
- Set the maximum threshold to 400 and the minimum threshold to 200 on queue 0 of interface te-1/1/1.
- Set the packet drop probability to 50% on queue 0 of interface te-1/1/1.
- Enable ECN on queue 0 of interface te-1/1/1.

```
admin@PICOS# set interface gigabit-ethernet te-1/1/3 wred queue 0 enable true
```

```
admin@PICOS# set interface gigabit-ethernet te-1/1/3 wred queue 0 max_thresh 400
admin@PICOS# set interface gigabit-ethernet te-1/1/3 wred queue 0 min_thresh 200
admin@PICOS# set interface gigabit-ethernet te-1/1/3 wred queue 0 drop_probability 50
admin@PICOS# set interface gigabit-ethernet te-1/1/3 wred queue 0 ecn_thresh 1
admin@PICOS# commit
```

After configuration:

- Use the command to display WRED configuration for the specified interface.

```
admin@PICOS# run show interface gigabit-ethernet te-1/1/1 wred
Queue Num Min Thresh Max Thresh Drop Probability ECN Thresh Status
---------- ---------- ---------- ---------------- ------------ ----------
0 200 400 50% Enabled Enabled
1 0 0 0% Disabled Disabled
2 0 0 0% Disabled Disabled
3 0 0 0% Disabled Disabled
4 0 0 0% Disabled Disabled
5 0 0 0% Disabled Disabled
6 0 0 0% Disabled Disabled
7 0 0 0% Disabled Disabled
```

Step 5: Enable Dynamic Load Balancing for ECMP

By enabling dynamic load balancing, Equal-Cost Multi-Path (ECMP) routing can distribute traffic more evenly across multiple member links. This maximizes load balancing efficiency among ECMP paths.

```
admin@PICOS# set interface ecmp hash-mapping dlb-normal

admin@PICOS# commit
```

2.4 RoCE EasyDeploy Initialization

RoCE EasyDeploy is a feature designed to simplify the deployment and configuration of RoCE (RDMA over Converged Ethernet) on switches. It enables seamless integration with servers to optimize network performance. This feature allows users to easily switch between lossless and lossy modes, ensuring optimal performance in diverse network environments.

Features & Benefits

- Simplifies RoCE deployment and configuration on switches
- Enables fast switching between lossless and lossy transmission modes
- Fewer configuration steps with optimized default parameters
- Flexible interface-level control
- Enhances network stability and ease of maintenance

Limitations & Guidelines

- Supported only on Tomahawk2, Trident3-X7, and Tomahawk3 platforms
- The number of ECN and PFC queues on the switch must align with server-side configuration
- Post-deployment fine-tuning is supported and recommended based on network conditions
- Continuous monitoring of RoCE statistics is required for performance assurance

Configuration Example

- Set RoCE mode to lossless
- Apply RoCE mode to all interfaces

```
admin@PICOS# set class-of-service roce mode lossless
admin@PICOS# set class-of-service roce apply all
admin@PICOS# commit
```

- Verify RoCE configuration after deployment

```
admin@PICOS# run show class-of-service roce
status applied
mode lossless
congestion-control
 congestion-mode ECN
 enabled-queue 3
 max-threshold 1500000 bytes
 min-threshold 150000 bytes
 probability 100
pfc
 pfc-priority 3
 rx-enabled enabled
 tx-enabled enabled
trust
 trust-mode dscp

RoCE PCP/DSCP->LP mapping configurations
========================================
local-priority dscp
------------ --------------------
0 0,1,2,3,4,5,6,7
1 8,9,10,11,12,13,14,15
2 16,17,18,19,20,21,22,23
3 24,25,26,27,28,29,30,31
4 32,33,34,35,36,37,38,39
5 40,41,42,43,44,45,46,47
6 48,49,50,51,52,53,54,55
7 56,57,58,59,60,61,62,63

 RoCE LP->FC mapping and ETS configurations
 ==========================================
local-priority forwarding-class scheduler-weight
-------------- ----------------- -----------------
 0 default WRR-8
 1 default WRR-8
 2 default WRR-8
 3 roce WRR-8
 4 default WRR-8
 5 default WRR-8
 6 cnp SP
 7 default WRR-8
```

3 RoCE Configuration on Ethernet NICs

3.1 Hardware Requirements

> ⓘ A low-frequency CPU can increase RDMA polling latency. Since RDMA relies on Direct Memory Access (DMA) for high-speed data transfers, proper hardware provisioning is essential.

CPU Configuration

- **PCIe 5.0 Support:** The CPU must support a PCIe 5.0 controller with at least 40 PCIe 5.0 lanes to ensure sufficient bandwidth.
- **High-Frequency, Multi-Core:** At least 16 cores / 32 threads are required to handle high RDMA concurrency.
- **NUMA Optimization:** Use NUMA binding to ensure RDMA test processes and NICs are located on the same NUMA node. Use tools like numactl --cpunodebind, numactl --membind, or taskset to reduce latency caused by cross-node access.

Category	Low-Latency RDMA(HPC, Small Packet)	High-Throughput RDMA(AI/Storage, Large Packet)
CPU Frequency	≥ 3.5 GHz (Ideal: 4.0 GHz+)	≥ 2.5 GHz (Recommended: 3.0 GHz+)
Core Count	16–32 cores (Single-core performance priority)	32–128 cores (Multi-core scaling priority)
PCIe Version	PCIe 5.0 x16 (Required)	PCIe 5.0 x16 (Required)
Arch Optimization	Disable C-States, pin to high-performance cores	NUMA binding, optimized for parallelism
Recommended CPUs	Intel Xeon 8468H (3.8GHz, 48C)AMD EPYC 9554 (3.35GHz, 64C)	AMD EPYC 9754 (3.1GHz, 128C)Intel Xeon 8490H (3.5GHz, 60C)

Memory Configuration

- **High Bandwidth:** Use DDR5 4800 MT/s or higher. For example, DDR5 at 4400 MT/s (2DPC) delivers ~35.2 GB/s per channel (≈281.6 Gbit/s).

> ⓘ Note: Intel 4th Gen Silver CPUs support up to 4000 MT/s.

- **Sufficient Capacity:** RDMA requires a large amount of pre-pinned memory.
  - 128 GB is the minimum recommended.
  - 1 TB or more may be needed for large-scale tests.
  - Ensure ulimit -l unlimited is set to enable large locked memory via mlock(), avoiding swap-related performance degradation.
- **NUMA Affinity:** Bind memory to the same NUMA node as the RDMA process. Use numactl -N or taskset -c

Memory Category	Low-Latency RDMA (HPC, Small Packet)	High-Throughput RDMA (AI/Storage, Large Packet)
Memory Capacity	≥ 64 GB (128 GB+ recommended)	≥ 128 GB (512 GB+ recommended)
Memory Channels	≥ 4 channels (8+ recommended)	≥ 8 channels
Memory Speed	DDR5 4800 MT/s or higher	DDR5 4800 MT/s or higher
NUMA Binding	Required (bind to same NUMA node)	Required (bind to same NUMA node)
HugePages	Enable 2MB/1GB HugePages	Enable 2MB/1GB HugePages

BIOS Configuration

(Refer to vendor documentation for exact settings)

- Disable virtualization
- Set CPU mode to performance
- Disable C-States and P-States to reduce frequency fluctuations and latency
- Disable IOMMU, or set it to pass-through (PT) mode

UEFI/BIOS Area	Value
BIOS -> Processor Settings	Logical Processor = Disable
	Virtualization Technology = Disable
	SubNumaCluster = Disable
	MADt Core cluster = Linear
BIOS -> Integrated Devices	Global SRIOV = Disable
BIOS -> System Profile Setting	Server System Profile = Performance
	Workload = Not Configured
BIOS -> System Security	AC Recovery Delay = Random (highly recommended)

3.2 NIC Selection

3.2.1 Broadcom

Part Number	ASIC Chip	Ports	Connector
BCM957504-P425G	BCM57504	4x 25G	SFP28
BCM957508-P2100G	BCM57508 (Thor1)	2x 100G 1x 200G	QSFP56
BCM957508-P1200G	BCM57508 (Thor1)	1x 200G	QSFP56
BCM957608-P2200G	BCM57608 (Thor2)	2x 200G 1x 400G	QSFP-DD
BCM957608-P1400G	BCM57608 (Thor2)	1x 400G	QSFP112-DD

Step 1: Install IP and RoCE Drivers

> ⓘ All validation and testing are based on Ubuntu 22.04 LTS, using the Broadcom BCM957608-P1400G network adapter. The driver package version used is 232.1.132.8.

Download the appropriate driver installation package from the Broadcom Ethernet network adapter Driver download page.

The package includes an automated tool for driver installation and system configuration. After extracting the ZIP file, the contents are placed in a directory, referred to below as Ubuntu_DIR.

The installer provides the following:

- Installation of included drivers, firmware, RDMA libraries, and utilities (niccli)
- Configuration of interface priorities for both IP and RoCE
- Sample switch configuration templates
- Installation of additional system packages: ansible, libibverbs-utils, rdmacm-utils, perftest, gcc, make, rpmbuild, kernel headers, etc.

```
root@FS1:~# sudo apt install -y automake autoconf libtool libibverbs-dev ibverbs-utils infiniband-diags perftest ethtool

root@FS1:~# sudo tar -zxvf bcm_232.1.132.8c.tar.gz -C Ubuntu_DIR

root@FS1:~# sudo cd Ubuntu_DIR/bcm_232.1.132.8c/utils/linux_installer

root@FS1:~# sudo bash install.sh -i  -i  -f -g # =bus id
```

- Verify if the specified version is installed

```
root@FS1:~# sudo dmesg | grep -i bnx | grep -i ver
[ 4.285625] bnxt_en: module verification failed: signature and/or required key missing - tainting kernel
[ 4.327736] Broadcom NetXtreme-C/E/S driver bnxt_en v1.10.3-232.0.155.5
[ 45.158387] bnxt_re: Broadcom NetXtreme-C/E RoCE Driver bnxt_re v232.0.155.5 (December 10, 2024)
[ 7655.824701] Broadcom NetXtreme-C/E/S driver bnxt_en v1.10.3-232.0.155.5
[ 7664.413538] bnxt_re: Broadcom NetXtreme-C/E RoCE Driver bnxt_re v232.0.155.5 (December 10, 2024)
[ 7676.806627] Broadcom NetXtreme-C/E/S driver bnxt_en v1.10.3-232.0.155.5
[ 7685.553099] bnxt_re: Broadcom NetXtreme-C/E RoCE Driver bnxt_re v232.0.155.5 (December 10, 2024)

root@FS1:~# modinfo bnxt_en | grep -i ver
version:  1.10.3-232.0.155.5
description: Broadcom NetXtreme-C/E/S network driver
srcversion: A88BAEC6211445D95464C78
vermagic: 6.8.0-57-generic SMP preempt mod_unload modversions

root@FS1:~# modinfo bnxt_re | grep -i ver
version: 232.0.155.5
description: Broadcom NetXtreme RoCE Driver
srcversion: B738B8F51148E0DA45D7322
depends: ib_uverbs,ib_core,bnxt_en
vermagic: 6.8.0-57-generic SMP preempt mod_unload modversions

root@FS1:~# dkms status|grep -i bnxt
bnxt_en/1.10.3.232.0.155.5, 6.8.0-57-generic, x86_64: installed

bnxt_re/232.0.155.5, 6.8.0-57-generic, x86_64: installed
```

Step 2: Manually Configure RoCE Settings

The default RoCE configuration is as follows:

- RoCEv2 (RDMA over IPv4) is enabled
- Congestion Control and PFC are enabled
- RoCE traffic is marked with DSCP 26 on priority 3
- RoCE CNP traffic is marked with DSCP 48 on priority 7
- MTU is set to 1500 bytes

```
root@FS1:~# cat /etc/bnxt_re/bnxt_re.conf

ENABLE_FC=1

FC_MODE=3

ROCE_PRI=3

ROCE_DSCP=26

CNP_PRI=7

CNP_DSCP=48
```

```
ROCE_BW=50

UTILITY=3
```

Switch ECN + PFC Synchronization Example

```
qos map dscp 26 to traffic-class 3
qos map dscp 48 to traffic-class 7
interface Ethernet1/1
 mtu 1500
 tx-queue 3
random-detect ecn minimum-threshold 500 kbytes maximum-threshold 1500 kbytes max-mark-probability 20
 priority-flow-control mode on
 priority-flow-control priority 3 no-drop
```

Manually Modify NIC RoCE Configuration

The NIC has RDMA enabled by default. You can verify and configure it using the NICCLI tool.

- Enable the RDMA option

```
To check the option value:
sudo niccli -i  nvm -getoption support_rdma -scope
# Example:
root@FS1:~# sudo niccli -i 1 nvm -getoption support_rdma -scope 0

--------------------------------------------------------------------------------
NIC CLI v232.0.153.0 - Broadcom Inc. (c) 2024 (Bld-106.52.39.138.16.0)
--------------------------------------------------------------------------------

support_rdma = True

To enable the option:
sudo niccli -i  nvm -setoption support_rdma -scope  -value 1
# value 0: Default
# value 1: RoCE
# Example:
root@FS1:~# sudo niccli -i 1 nvm -setoption support_rdma -scope 0 -value 1

--------------------------------------------------------------------------------
NIC CLI v232.0.153.0 - Broadcom Inc. (c) 2024 (Bld-106.52.39.138.16.0)
--------------------------------------------------------------------------------

support_rdma is set successfully
```

- Enable the RoCE performance profile on the NIC

```
To check the option value:
 sudo niccli -i  nvm -getoption performance_profile
# Example:
root@FS1:~# sudo niccli -i 1 nvm -getoption performance_profile
--------------------------------------------------------------------------------
```

```
NIC CLI v232.0.153.0 - Broadcom Inc. (c) 2024 (Bld-106.52.39.138.16.0)
-------------------------------------------------------------------------------

performance_profile = default

To enable the option:
 sudo niccli -i  nvm -getoption performance_profile
# value 0: Default
# value 1: RoCE
# Example:
root@FS1:~# sudo niccli -i 1 nvm -setoption performance_profile -value 1
-------------------------------------------------------------------------------
NIC CLI v232.0.153.0 - Broadcom Inc. (c) 2024 (Bld-106.52.39.138.16.0)
-------------------------------------------------------------------------------

performance_profile is set successfully
Please do the device reset to apply the configuration
```

- Enable PCIe relaxed ordering on the NIC

```
To check the option value
sudo niccli -i  nvm -getoption pcie_relaxed_ordering
# Example:
root@FS1:~# sudo niccli -i 1 nvm -getoption pcie_relaxed_ordering
-------------------------------------------------------------------------------
NIC CLI v232.0.153.0 - Broadcom Inc. (c) 2024 (Bld-106.52.39.138.16.0)
-------------------------------------------------------------------------------

pcie_relaxed_ordering = Disabled


To enable the option:
sudo niccli -i  nvm -setoption pcie_relaxed_ordering -value 1
# Example:
root@FS1:~# sudo niccli -i 1 nvm -setoption pcie_relaxed_ordering -value 1
-------------------------------------------------------------------------------
NIC CLI v232.0.153.0 - Broadcom Inc. (c) 2024 (Bld-106.52.39.138.16.0)
-------------------------------------------------------------------------------

pcie_relaxed_ordering is set successfully
Please reboot the system to apply the configuration
```

- Modify congestion control (cc) parameters

Option	Description
cc_mode	0 is used for DCQCN-D (deterministic marking); 1 is for interoperability with DCQCN-P (probabilistic marking). Default is 1.
ecn_enable	Enable ECN.
g	Weight for running average in rate calculation: weight is 2^g, where g is between 1 and 9. Default is 8.
init_cr	Initial rate when QP is created or before inactivity_th expires. Value between 0–1023, representing a percentage of link BW. Required when PF is enabled.

init_tr	Target rate after inactivity_th expires and QP is created. Value between 0–1023, representing a percentage of link BW.
rtt	Round-trip time (RTT) in nanoseconds, used to calculate the interval between CNP and actual packet processing. Default is 100000 ns.
inact_cp	If QP is inactive, the counter used for suppression logic will decay after the threshold expires. Default is 100000 microseconds.
cnp_dscp	DSCP value for CNP packets.
roce_dscp	DSCP value for RoCE traffic.
cnp_prio	Priority for CNP packets.
roce_prio	Priority for RoCE traffic.
apply	Apply the settings.

```
mkdir -p /sys/kernel/config/rdma_cm/bnxt_re0
echo RoCE v2 > /sys/kernel/config/rdma_cm/bnxt_re0/ports/1/default_roce_mode

mkdir -p /sys/kernel/config/bnxt_re/bnxt_re0
cd /sys/kernel/config/bnxt_re/bnxt_re0/ports/1/cc/

echo -n 0x1a > roce_dscp
echo -n 0x3 > roce_prio
echo -n 0x1 > disable_prio_vlan_tx
echo -n 0x1 > ecn_enable
echo -n 0x30 > cnp_dscp
echo -n 0x7 > cnp_prio
#After setting all the above parameters, apply the values to HW
echo -n 0x1 > apply
```

- Configure Congestion Control Using bnxt_setupcc.sh

  The bnxt_setupcc.sh tool uses niccli or lldptool to configure network parameters with a single command. An example is shown below:

```
root@FS1:~# sudo bnxt_setupcc.sh -d bnxt_re0 -i ens4f0np0 -m 3 -s 26 -p 48 -r 3 -c 7 -u 3
ENABLE_PFC = 1 ENABLE_CC = 1
ENABLE_DSCP = 1 ENABLE_DSCP_BASED_PFC = 1
L2 50 RoCE 50
Using Ethernet interface ens2np0 and RoCE interface bnxt_re0
Setting pfc/ets 0000:2a:00.0
--------------------------------------------------------------------------------
NIC CLI v232.0.153.0 - Broadcom Inc. (c) 2024 (Bld-106.52.39.138.16.0)
--------------------------------------------------------------------------------

BCM57608> AppTLV deleted successfully.
BCM57608> AppTLV deleted successfully.
BCM57608> AppTLV deleted successfully.
BCM57608> Enhanced transmission selection (ets) configured successfully.
BCM57608> pfc configured successfully.
BCM57608> AppTLV configured successfully.
BCM57608> AppTLV configured successfully.
BCM57608> AppTLV configured successfully.
BCM57608>
```

```
Settings Default to use RoCE-v2

root@FS1:~# sudo niccli -i 1 getqos
/opt/niccli/niccli.x86_64: /lib/x86_64-linux-gnu/libnl-3.so.200: no version information available (required by /opt/niccli/niccli.x86_64)

--------------------------------------------------------------------------------
NIC CLI v232.0.153.0 - Broadcom Inc. (c) 2024 (Bld-106.52.39.138.16.0)
--------------------------------------------------------------------------------

IEEE 8021QAZ ETS Configuration TLV:
 PRIO_MAP: 0:0 1:0 2:0 3:1 4:0 5:0 6:0 7:2
 TC Bandwidth: 50% 50% 0%
 TSA_MAP: 0:ets 1:ets 2:strict
IEEE 8021QAZ PFC TLV:
 PFC enabled: 3
IEEE 8021QAZ APP TLV:
 APP#0:
  Priority: 7
 Sel: 5
 DSCP: 48

 APP#1:
 Priority: 3
 Sel: 5
 DSCP: 26

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

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

Parameter Description：

Option	Description
-d	RoCE Device Name (for example, bnxt_re0, bnxt_re_bond).
-i	Ethernet Interface Name (for example, p1p1 or for bond, specify secondary interfaces like -i p6p1 -i p6p2).
-r [0-7]	RoCE packet priority.
-s VALUE	RoCE packet DSCP value.
-c [0-7]	RoCE CNP packet priority.
-p VALUE	RoCE CNP packet DSCP value.
-v VALUE	Use specified VLAN ID instead of DSCP.
-b VALUE	RoCE Bandwidth percentage for ETS configuration. Default is 80%.
-h	Display help.
-m [1-3]	1 – PFC only.2 – CC only.3 – PFC + CC mode.
-u [2-3]	Utility to configure QoS settings.2 – Use lldptool3 – Use niccli utility

3.2.2 NVIDIA

Part Number	ASIC Chip	Ports	Connector
MCX512A-ACAT	ConnectX-5	2x 25G	SFP28
MCX623106AN-VDAT	ConnectX-6	2x 100G	QSFP56
MCX653106A-HDAT	ConnectX-6	2x 200G	QSFP56
MCX755106AS-HEAT	ConnectX-7	2x 200G	QSFP112
MCX75310AAS-HEAT	ConnectX-7	1x 400G	OSFP

Step 1: Install IP and RoCE Drivers

> ⓘ All testing is conducted on Ubuntu 22.04 LTS, using the BCM957608-P1400G NIC. The installation package version used is 232.1.132.8.

For NVIDIA NICs, both RoCE and IP drivers are integrated into the MLNX_OFED package. Multiple installation methods are available. You can download the MLNX_OFED version that matches your system from the NVIDIA official driver page.

The MLNX_OFED package includes all necessary components for NVIDIA NICs (e.g., ConnectX-5/6/7), including:

- Kernel modules and drivers: mlx5_core, ib_uverbs, ib_core
- User-space protocol libraries and development APIs: for RDMA and RoCE application development
- Device management and debugging tools
- Performance testing and benchmarking tools

```
wget https://content.mellanox.com/ofed/MLNX_OFED-/MLNX_OFED_LINUX--ubuntu22.04-x86_64.tgz
tar -xvzf MLNX_OFED_LINUX-*.tgz
cd MLNX_OFED_LINUX-*
sudo ./mlnxofedinstall --add-kernel-support
```

- Verify if the specified version is installed

```
root@FS1:~# sudo dmesg | grep -i mlx5 | grep -i ver
[115345.338018] mlx5_core 0000:45:00.0: mlx5_pcie_event:304:(pid 3648990): PCIe slot advertised sufficient power (75W).
[116552.824757] mlx5_core 0000:45:00.0: firmware version: 28.43.1014
[116553.127338] mlx5_core 0000:45:00.0: mlx5_pcie_event:304:(pid 3732904): PCIe slot advertised sufficient power (75W).
[116578.352357] mlx5_core 0000:1f:00.0: firmware version: 28.43.1014
[116578.657001] mlx5_core 0000:1f:00.0: mlx5_pcie_event:304:(pid 3732904): PCIe slot advertised sufficient power (75W).

root@FS1:~# sudo modinfo mlx5_core | grep -i ver
version: 24.07-0.6.1
description: Mellanox 5th generation network adapters (ConnectX series) core driver
srcversion: DBC460DDA01C102939C89DC
vermagic: 6.8.0-52-generic SMP preempt mod_unload modversions

root@FS1:~# sudo modinfo mlx5_ib |grep -i ver
description: Mellanox 5th generation network adapters (ConnectX series) IB driver
srcversion: 84F601BAD872C13B202813E
depends: mlx5_core,ib_uverbs,ib_core,macsec,mlx_compat
vermagic: 6.8.0-52-generic SMP preempt mod_unload modversions

root@FS1:~# sudo dkms status |grep -i mlnx-ofed
mlnx-ofed-kernel/24.07.OFED.24.07.0.6.1.1, 6.8.0-52-generic, x86_64: installed
```

```
mlnx-ofed-kernel/24.07.OFED.24.07.0.6.1.1, 6.8.0-57-generic, x86_64: installed
```

Step 2: Manually Configure RoCE Settings

Zero Touch RoCE (ZTR) is a plug-and-play RDMA auto-optimization solution that automatically configures key parameters such as PFC, ECN, and DSCP without manual intervention. This ensures lossless RoCEv2 traffic over Ethernet.

The default RoCE configuration includes:

- RoCEv2 (RDMA over IPv4) is enabled
- PFC is enabled
- ECN is enabled on all priority queues
- DCQCN is configured based on ECN
- RoCEv2 CNP packets are marked with DSCP 46
- MTU is set to 1500 bytes

```
root@FS1:~# sudo mlnx_qos -i ens2np0 -a
DCBX mode: OS controlled84
Priority trust state: dscp
dscp2prio mapping:
prio:0 dscp:07,06,05,04,03,02,01,00,
prio:1 dscp:15,14,13,12,11,10,09,08,
prio:2 dscp:23,22,21,20,19,18,17,16,
prio:3 dscp:31,30,29,28,27,26,25,24,
prio:4 dscp:39,38,37,36,35,34,33,32,
prio:5 dscp:47,46,45,44,43,42,41,40,
prio:6 dscp:55,54,53,52,51,50,49,48,
prio:7 dscp:63,62,61,60,59,58,57,56,
default priority:
Receive buffer size (bytes): 19872,243072,0,0,0,0,0,0,max_buffer_size=2069280
Cable len: 7
PFC configuration:
 priority 0 1 2 3 4 5 6 7
 enabled 0 0 0 0 0 0 0 0
 buffer 1 1 1 1 1 1 1 1
tc: 0 ratelimit: unlimited, tsa: vendor
 priority: 0
tc: 1 ratelimit: unlimited, tsa: vendor
 priority: 1
tc: 2 ratelimit: unlimited, tsa: vendor
 priority: 2
tc: 3 ratelimit: unlimited, tsa: vendor
 priority: 3
tc: 4 ratelimit: unlimited, tsa: vendor
 priority: 4
tc: 5 ratelimit: unlimited, tsa: vendor
 priority: 5
tc: 6 ratelimit: unlimited, tsa: vendor
 priority: 6
tc: 7 ratelimit: unlimited, tsa: vendor
 priority: 7
```

Manually Modify NIC RoCE Configuration

RDMA is enabled by default on the NIC. You can use the mlnx_qos tool to verify and configure RoCE-related settings.

- DCQCN-ECN Configuration

```
root@FS1:~# sudo mlxconfig -d mlx5_0 query|grep -i ROCE_CC_PRIO_MASK

 ROCE_CC_PRIO_MASK_P1 255

root@FS1:~# mlxconfig -d mlx5_0 set ROCE_CC_PRIO_MASK_P1 =0x08
Device #1:
----------

Device type: ConnectX7
Name: MCX75310AAS-NEA_Ax
Description: NVIDIA ConnectX-7 HHHL Adapter card; 400GbE / NDR IB (default mode); Single-port OSFP; PCIe 5.0 x16; Crypto Disabled; Secure Boot Enabled;
Device: mlx5_2

Configurations: Next Boot New
ROCE_CC_PRIO_MASK_P1 255 0x10(16)

 Apply new Configuration? (y/n) [n] :
```

Note: 255 in binary is 11111111, which corresponds to 8 priority queues.

This means **DCQCN is enabled for all priorities (Priority 0–7)** — RoCE congestion control will apply to all traffic.

To enable DCQCN **only on Priority 3**, convert the binary value 00001000 to hexadecimal, which equals 0x10

- RoCE_np Settings

```
To check cnp_dscp options
root@FS1:~# cat /sys/class/net/ens2np0/ecn/roce_np/cnp_802p_prio
6

root@FS1:~# cat /sys/class/net/ens2np0/ecn/roce_np/cnp_dscp
48

root@FS1:~# cat /sys/class/net/ens2np0/ecn/roce_np/enable/*
1
1
1
1
1
1
1
1
# Indicates that all queues have RoCE_np enabled by default. No modification is needed unless customization is required.


To temporary configuration
root@FS1:~# echo 5 >/sys/class/net/ens2np0/ecn/roce_np/cnp_802p_prio
```

```
5

root@FS1:~# echo 36 >/sys/class/net/ens2np0/ecn/roce_np/cnp_802p_prio
36

To persistent configuration
mlxconfig -d ibdev_name or /dev/mst/ -y s CNP_DSCP_P1= CNP_802P_PRIO_P1=
#example
root@FS1:~# sudo mlxconfig -d mlx5_0 -y set CNP_DSCP_P1=36 CNP_802P_PRIO_P1=5

Device #1:
----------

Device type: ConnectX7
Name: MCX75310AAS-NEA_Ax
Description: NVIDIA ConnectX-7 HHHL Adapter card; 400GbE / NDR IB (default mode); Single-port OSFP; PCIe 5.0 x16; Crypto
Disabled; Secure Boot Enabled;
Device: mlx5_0

Configurations: Next Boot New
 CNP_DSCP_P1 40 36
 CNP_802P_PRIO_P1 7 5

 Apply new Configuration? (y/n) [n] : y
Applying... Done!
-I- Please reboot machine to load new configurations.

Reboot machine or reset fw

root@FS1:~# mlxfwreset -d mlx5_0 -l 3 -y r
Requested reset level for device, /dev/mst/mt4129_pciconf2:
3: Driver restart and PCI reset
Continue with reset?[y/N] y
-I- Sending Reset Command To Fw -Done
-I- Stopping Driver -Done
-I- Resetting PCI -Done
-I- Starting Driver -Done
-I- Restarting MST -Done
-I- FW was loaded successfully.
```

- PFC Configuration

```
To check pfc configrution
root@FS1:~# sudo mlnx_qos -i ens2np0
DCBX mode: OS controlled
Priority trust state: dscp
dscp2prio mapping:
prio:0 dscp:07,06,05,04,03,02,01,00,
prio:1 dscp:15,14,13,12,11,10,09,08,
prio:2 dscp:23,22,21,20,19,18,17,16,
prio:3 dscp:31,30,29,28,27,26,25,24,
prio:4 dscp:39,38,37,36,35,34,33,32,
prio:5 dscp:47,46,45,44,43,42,41,40,
prio:6 dscp:55,54,53,52,51,50,49,48,
prio:7 dscp:63,62,61,60,59,58,57,56,
```

```
default priority:
Receive buffer size (bytes): 0,156096,0,0,0,0,0,0,max_buffer_size=4151520
Cable len: 7
PFC configuration:
priority 0 1 2 3 4 5 6 7
enabled 0 0 0 1 0 0 0 0
buffer 0 0 0 1 0 0 0 0
tc: 0 ratelimit: unlimited, tsa: vendor
priority: 1
tc: 1 ratelimit: unlimited, tsa: vendor
priority: 0
tc: 2 ratelimit: unlimited, tsa: vendor
priority: 2
tc: 3 ratelimit: unlimited, tsa: vendor
priority: 3
tc: 4 ratelimit: unlimited, tsa: vendor
priority: 4
tc: 5 ratelimit: unlimited, tsa: vendor
priority: 5
tc: 6 ratelimit: unlimited, tsa: vendor
priority: 6
tc: 7 ratelimit: unlimite
```

To enable PFC for priority

```
root@FS1:~# mlnx_qos -i ens2np0 --pfc 0,0,1,0,0,0,0,0
DCBX mode: OS controlled
Priority trust state: dscp
dscp2prio mapping:
prio:0 dscp:07,06,05,04,03,02,01,00,
prio:1 dscp:15,14,13,12,11,10,09,08,
prio:2 dscp:23,22,21,20,19,18,17,16,
prio:3 dscp:31,30,29,28,27,26,25,24,
prio:4 dscp:39,38,37,36,35,34,33,32,
prio:5 dscp:47,46,45,44,43,42,41,40,
prio:6 dscp:55,54,53,52,51,50,49,48,
prio:7 dscp:63,62,61,60,59,58,57,56,
default priority:
Receive buffer size (bytes): 19872,156096,0,0,0,0,0,0,max_buffer_size=4151520
Cable len: 7
PFC configuration:
 priority 0 1 2 3 4 5 6 7
 enabled 0 0 1 0 0 0 0 0
 buffer 0 0 1 0 0 0 0 0
tc: 0 ratelimit: unlimited, tsa: vendor
 priority: 1
tc: 1 ratelimit: unlimited, tsa: vendor
 priority: 0
tc: 2 ratelimit: unlimited, tsa: vendor
 priority: 2
tc: 3 ratelimit: unlimited, tsa: vendor
 priority: 3
tc: 4 ratelimit: unlimited, tsa: vendor
 priority: 4
tc: 5 ratelimit: unlimited, tsa: vendor
 priority: 5
tc: 6 ratelimit: unlimited, tsa: vendor
```

```
 priority: 6
tc: 7 ratelimit: unlimited, tsa: vendor
 priority: 7

root@FS1:~# mlnx_qos -i ens2np0 --pfc 0,0,1,0,0,0,0,0
DCBX mode: OS controlled
Priority trust state: dscp
dscp2prio mapping:
prio:0 dscp:07,06,05,04,03,02,01,00,
prio:1 dscp:15,14,13,12,11,10,09,08,
prio:2 dscp:23,22,21,20,19,18,17,16,
prio:3 dscp:31,30,29,28,27,26,25,24,
prio:4 dscp:39,38,37,36,35,34,33,32,
prio:5 dscp:47,46,45,44,43,42,41,40,
prio:6 dscp:55,54,53,52,51,50,49,48,
prio:7 dscp:63,62,61,60,59,58,57,56,
default priority:
Receive buffer size (bytes): 19872,156096,0,0,0,0,0,0,max_buffer_size=4151520
Cable len: 7
PFC configuration:
 priority 0 1 2 3 4 5 6 7
 enabled 0 0 1 0 0 0 0 0
 buffer 0 0 1 0 0 0 0 0
tc: 0 ratelimit: unlimited, tsa: vendor
 priority: 1
tc: 1 ratelimit: unlimited, tsa: vendor
 priority: 0
tc: 2 ratelimit: unlimited, tsa: vendor
 priority: 2
tc: 3 ratelimit: unlimited, tsa: vendor
 priority: 3
tc: 4 ratelimit: unlimited, tsa: vendor
 priority: 4
tc: 5 ratelimit: unlimited, tsa: vendor
 priority: 5
tc: 6 ratelimit: unlimited, tsa: vendor
 priority: 6
tc: 7 ratelimit: unlimited, tsa: vendor
 priority: 7
```

To set dscp for priority

```
root@FS1:~# mlnx_qos -i ens2np0 --dscp2prio=set,30,2
DCBX mode: OS controlled
Priority trust state: dscp
dscp2prio mapping:
 prio:0 dscp:07,06,05,04,03,02,01,00,
 prio:1 dscp:15,14,13,12,11,10,09,08,
 prio:2 dscp:30,23,22,21,20,19,18,17,
 prio:2 dscp:16,
 prio:3 dscp:31,29,28,27,26,25,24,
 prio:4 dscp:39,38,37,36,35,34,33,32,
 prio:5 dscp:47,46,45,44,43,42,41,40,
 prio:6 dscp:55,54,53,52,51,50,49,48,
 prio:7 dscp:63,62,61,60,59,58,57,56,
default priority:
Receive buffer size (bytes): 19872,156096,0,0,0,0,0,0,max_buffer_size=4151520
```

```
Cable len: 7
PFC configuration:
 priority 0 1 2 3 4 5 6 7
 enabled 0 0 1 0 0 0 0 0
 buffer 0 0 1 0 0 0 0 0
tc: 0 ratelimit: unlimited, tsa: vendor
 priority: 1
tc: 1 ratelimit: unlimited, tsa: vendor
 priority: 0
tc: 2 ratelimit: unlimited, tsa: vendor
 priority: 2
tc: 3 ratelimit: unlimited, tsa: vendor
 priority: 3
tc: 4 ratelimit: unlimited, tsa: vendor
 priority: 4
tc: 5 ratelimit: unlimited, tsa: vendor
 priority: 5
tc: 6 ratelimit: unlimited, tsa: vendor
 priority: 6
tc: 7 ratelimit: unlimited, tsa: vendor
 priority: 7

root@FS1:~# ethtool -S ens2np0 | grep pause
 rx_pause_ctrl_phy: 0
 tx_pause_ctrl_phy: 0
 rx_prio2_pause: 0
 rx_prio2_pause_duration: 0
 tx_prio2_pause: 0
 tx_prio2_pause_duration: 0
 rx_prio2_pause_transition: 0
 tx_pause_storm_warning_events: 0
 tx_pause_storm_error_events: 0
```

3.3 Verify RoCE Configuration

After driver installation and configuration, perform the following steps to verify the RoCE setup:

- Check the GUID of the RoCE Interface
  - Use ibv_devices to confirm whether a GUID is available
  - Use ibv_devinfo to confirm the presence of a GUID and obtain additional details about the RoCE interface

```
root@FS1:~# ibv_devices
 device node GUID
 ------ ----------------
 mlx5_1 a088c20300d938f8
 mlx5_0 a088c20300d92ec0
root@FS1:~# ibv_devices
hca_id: mlx5_1
 transport: InfiniBand (0)
 fw_ver: 28.43.2566
 node_guid: a088:c203:00d9:38f8
 sys_image_guid: a088:c203:00d9:38f8
 vendor_id: 0x02c9
 vendor_part_id: 4129
```

```
 hw_ver: 0x0
 board_id: MT_0000000838
 phys_port_cnt: 1
 port: 1
 state: PORT_DOWN (1)
 max_mtu: 4096 (5)
 active_mtu: 4096 (5)
 sm_lid: 0
 port_lid: 65535
 port_lmc: 0x00
 link_layer: Ethernet

hca_id: mlx5_0
 transport: InfiniBand (0)
 fw_ver: 28.43.2566
 node_guid: a088:c203:00d9:2ec0
 sys_image_guid: a088:c203:00d9:2ec0
 vendor_id: 0x02c9
 vendor_part_id: 4129
 hw_ver: 0x0
 board_id: MT_0000000838
 phys_port_cnt: 1
 port: 1
 state: PORT_DOWN (1)
 max_mtu: 4096 (5)
 active_mtu: 1024 (3)
 sm_lid: 0
 port_lid: 0
 port_lmc: 0x00
 link_layer: Ethernet
```

- Use the perftest package utilities to verify traffic

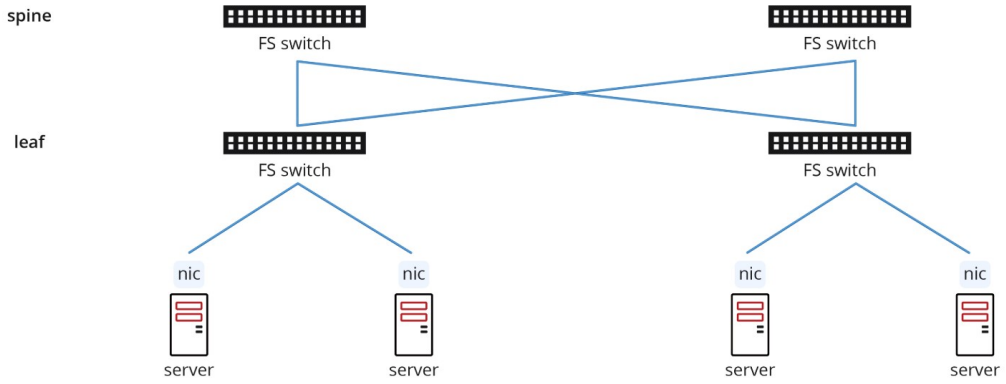
```
server#ib_write_bw -d mlx5_0 -F -x 3--report_gbits

client#ib_write_bw -d mlx5_0 -F -x 3--report_gbits 192.168.2.30 --run_i --qp 16
```

4 RoCE Performance Testing and Results

Performance measurements were taken on a cluster using Broadcom/NVIDIA NICs (200GE/400GE) and FS switches (200GE/400GE).

Server	Switch	NIC	Benchmarks
Model: Dell R860 CPU: 6448H Memory Type: DDR4 - 5600 MT/s Memory: 512 GB (128GB/Socket) Kernel: 6.8.0-57-generic (Ubuntu22.04)	Model: N5680-32D Hardware Revision: - Software Version: 4.5.0E/3b574830da	Model: Broadcom P1400G Driver Version: 1.10.3-232.0.155.5 Firmware Version: 230.2.36.0/pkg 230.2.37.0 Congestion Control OSU: DCQCN-p	Perftest: 6.23
		Model: MCX75310AAS-NEAT Driver Version: 24.10-2.1.8 Firmware Version: 28.43.2566 Congestion Control OSU: DCQCN-p	



The performance results from various benchmark tests are summarized below:

Category	Test	Broadcom	NVIDIA
RDMA Point-to-Point Bandwidth	Tool: ib_write_bw, ib_read_bw Unidirectional, Bidirectional Bandwidth	392.25 GB/sec 684 GB/sec	393 GB/sec 695 GB/sec
RDMA Point-to-Point Latency	Tool: ib_write_lat Small packet RTT latency test: 64B, 256B, 512B	3.58 t_avg [usec] 3.75 t_avg [usec] 3.75 t_avg [usec]	2.19 t_avg [usec] 2.92 t_avg [usec] 2.93 t_avg [usec]
RDMA Multi-Node Scalability	Tool: ib_write_bw Simulated multi-node RDMA access to a single target node	node1: 130 GB/sec node2: 131 GB/sec node3: 129 GB/sec total: 390 GB/sec	node1: 131 GB/sec node2: 128 GB/sec node3: 134 GB/sec total: 393 GB/sec
Long-Term Stability Test	Tool: ib_write_bw 12+ hours of sustained high-pressure traffic; packet loss, performance fluctuation recorded	392 GB/sec	392 GB/sec