Lossless Network for AI/ML/Storage/HPC with RDMA

Version 2

January 2024

BROADCOM

Table of contents

Table of contents

Introduction

Datacenter networking has evolved over the years and with the proliferation of AI/ML, disaggregated storage, and High-Performance Computing (HPC), today's data centers require a high performance, low-latency network. With ever increasing database sizes and demand for high bandwidth for the movement of data between processing nodes, a reliable transport is critical. As the future of metaverse applications evolve, the network needs to adapt to the humongous growth in data transfer due to data-intensive and compute-intensive applications. Broadcom's Ethernet Adapters (also referred to as Ethernet NICs) along with Arista Networks' switches (based on Broadcom's DNX and XGS family of ASICs) leverage RDMA (Remote Direct Memory Access) to eliminate any connectivity bottlenecks and facilitate a high-throughput, low latency transport.

RDMA Over Converged Ethernet (RoCE)

RoCE (RDMA over Converged Ethernet) is a network protocol that allows RDMA over an Ethernet network. RDMA helps to reduce the CPU workload as it offloads all transport communication tasks from the CPU to hardware and provides direct memory access for applications without involving the CPU. The second version of RoCE (RoCE-v2) enhances the protocol with UDP/IP header and enables a routable RoCE. Broadcom's Ethernet Adapters support RoCEv2 in hardware and allows for higher throughput, lower latency, and lower CPU utilization, which are critical for AI/ML, Storage, and High-Performance Compute (HPC) applications.

RoCEv2 provides three advantages:

- Operation on routed ethernet networks, ubiquitous in large data centers
- IP QoS – The DiffServ code point (DSCP), or alternatively VLAN PRI
- IP congestion control – The explicit congestion notification (ECN) signal

To eliminate potential packet loss and high latency on Ethernet networks, RoCEv2 uses congestion control mechanisms supported on Arista switches and Broadcom NICs such as Priority Flow Control (PFC), Explicit Congestion Notification (ECN) etc.

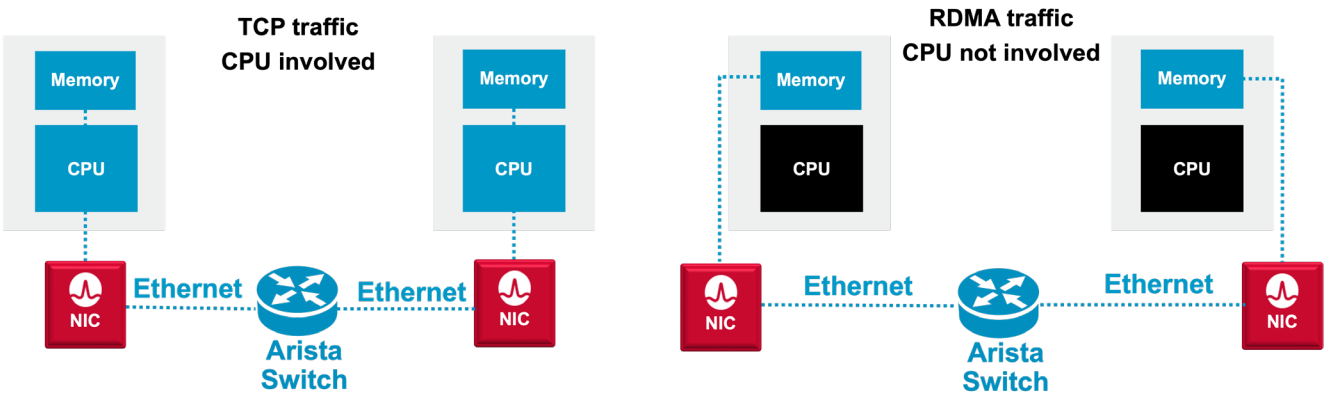

Figure 1: RoCE with Broadcom NICs and Arista Switches

RoCEv2 also defines a Congestion Notification Packet (CNP). RNICs send CNPs in response to ECN Congestion Experienced (CE) markings to indicate that the transmission rate should be reduced. ECN marking is done by switches along the path between source and destination or by the receiving NIC. CNPs are associated with RoCE connections, providing fine-grained, per-connection congestion notification information. RoCEv2 only specifies the mechanism for marking packets when congestion is experienced and

the format of the CNP response. It leaves the implementation of congestion control algorithm unspecified, including the following information:

- When packets are ECN marked (at which queue level, and at what probability)
- When CNPs are generated in response to ECN
- How sending rate is adjusted in response to CNPs

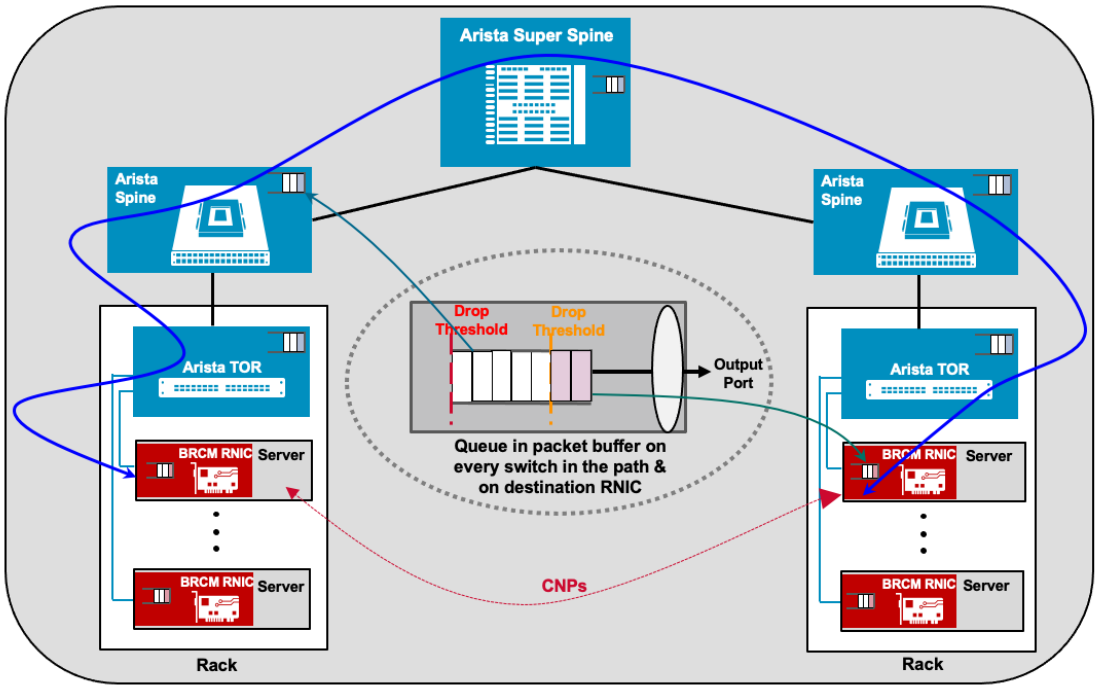

Figure 2: RoCE traffic in datacenter network

Deploying RoCE on Arista Switches

Arista Extensible Operating System (EOS®) is the core of Arista cloud networking solutions for next-generation data centers and high-performance computing networks. Arista EOS® provides all the necessary tools to achieve a premium lossless, high bandwidth low latency network. EOS® supports traffic management configuration, adjustable buffer allocation schemes and use of PFC and DCQCN to support RoCE deployments across Arista 7280R series, 7800R series, 7050X series, and 7060X series. The exact Arista Switch to be used depends on the specific use case.

Table 1: Arista Datacenter Switches for RoCE use cases

Arista Switch	Description
7800R3	Highest density 100G/400G Deep Buffer, Lossless Modular Super Spine Switch
7280R3	High Performance 10/40/100/400G Data Center switch with Dynamic Deep Buffer
7060X	High Performance 10/40/100/400G Fixed Configuration Switch
7050X3	1 RU 10/40G Multilayer Leaf Switches

The Arista 7280R and 7800R series are based on the Broadcom Jericho chipset families. Equipped with deep buffers and Virtual Output Queueing scheduling mechanisms, these ensure lossless transmission of end-to-end data. The 7280R series is the fixed configuration family of switches, while the 7800R series is the modular line of switches.

The Arista 7050X and 7060X series are based on the Broadcom Trident and Tomahawk chipset families respectively. Supporting rich feature support and low latency, the 7050X and 7060X series are perfect for highly efficient and robust deployments.

General installation and configuration of Arista switches is available here.

Once end-to-end network connectivity is established, Priority Flow Control (PFC) or Explicit Congestion Notification (ECN) can be enabled to ensure lossless transport for RoCE traffic.

Configuration and Verification of PFC on Arista Switches

PFC is one of the most important aspects of successful RoCE deployments. PFC specifies a link-layer flow control mechanism between directly connected peers. It uses the 802.3 PAUSE frames to implement flow-control measures for multiple classes of traffic. Switches can drop the less important traffic and notify the peer devices to pause traffic on specific classes so that critical data is not dropped and allowed to pass through the same port without any restrictions.

This Quality of Service (QoS) capability allows differentiated treatment of traffic based on the CoS/priority and eases congestion by ensuring that critical I/O is not disrupted and that other non-critical traffic that is more loss-tolerant can be dropped. Each priority is configured as either drop or no-drop. If a priority that is designated as no-drop is congested, the priority is paused. Drop priorities do not participate in pause.

PFC Configuration

The CLI command to enable PFC on the interface is "priority-flow-control mode on" and "priority-flow-control priority <TC> no-drop" enables PFC on that Transmit Queue.

- Enable PFC on the interface.

```
arista(config)#interface ethernet 3/1/1
arista(config-if-Et3/1/1)#priority-flow-control mode on
```

- Enable PFC for specific TCs.

```
arista(config-if-Et3/1/1)#priority-flow-control priority 0 no-drop
```

The above command should be issued for all the TC's that the user wants to enable PFC on.

Example Configuration

The following configuration shows how PFC can be configured for TC3 and TC4 under interface Ethernet 2/1/1 on an Arista switch.

```
interface Ethernet2/1/1
   mtu 9000
   speed 200G-4
   no switchport
   priority-flow-control mode on
   priority-flow-control priority 3 no-drop
   priority-flow-control priority 4 no-drop
   !
```

Show Commands

1. show priority-flow-control interfaces ethernet <>

```
arista#show priority-flow-control interfaces ethernet 3/1/1
The hardware supports PFC on priorities 0 1 2 3 4 5 6 7
PFC receive processing is enabled on priorities 0 1 2 3 4 5 6 7
The PFC watchdog timeout is 1.0 second(s)
The PFC watchdog recovery-time is 2.0 second(s) (auto)
The PFC watchdog polling-interval is 0.2 second(s)
The PFC watchdog action is drop
The PFC watchdog override action drop is false
Global PFC : Disabled
E: PFC Enabled, D: PFC Disabled, A: PFC Active, W: PFC Watchdog Active
Port       Status  Priorities  Action      Timeout  Recovery        Polling         Note
                                                    Interval/Mode   Config/Oper
-------------------------------------------------------------------------------
Et3/1/1    E - -   01          -           -        - / -           - / -           DCBX
disabled
Port                 RxPfc           TxPfc
Et3/1/1                  0               0
```

2. show priority-flow-control interfaces ethernet < > counters

```
arista#show priority-flow-control interfaces ethernet 3/1/1 counters

Port                 RxPfc           TxPfc
Et3/1/1                  0               0
```

3. show priority-flow-control interface ethernet < > status

```
arista#show priority-flow-control interfaces ethernet 3/1/1 status
The hardware supports PFC on priorities 0 1 2 3 4 5 6 7
PFC receive processing is enabled on priorities 0 1 2 3 4 5 6 7
The PFC watchdog timeout is 1.0 second(s)
The PFC watchdog recovery-time is 2.0 second(s) (auto)
The PFC watchdog polling-interval is 0.2 second(s)
The PFC watchdog action is drop
The PFC watchdog override action drop is false
Global PFC : Disabled
E: PFC Enabled, D: PFC Disabled, A: PFC Active, W: PFC Watchdog Active
Port       Status  Priorities  Action      Timeout  Recovery        Polling         Note
                                                    Interval/Mode   Config/Oper
-------------------------------------------------------------------------------
Et3/1/1    E - -   01          -           -        - / -           - / -           DCBX
disabled
```

Configuration and Verification of ECN on Arista Switches

Explicit Congestion Notification (ECN) is an extension to TCP/IP that provides end-to-end notification of impending network congestion prior to loss. Two Bits (bit 0 and bit 1) in the ToS byte of the IP header are used for ECN. That is, ECN bits in the ToS byte defines a packet in 4 different ways:

- 00 - (default) indicates packet is non-ECN capable
- 01 - indicates packet is ECN capable
- 10 - indicates packet is ECN capable
- 11 - indicates Congestion Occurred somewhere in the network

ECN is an optional feature that is only used when both endpoints support it. ECN should be considered complementary to PFC for lossless network behavior and is therefore an integral component of RoCE. ECN bits are marked on traffic in certain classes when the configured buffer thresholds are exceeded.

ECN operates over an active queue management (AQM) algorithm - Weighted Random Early Detection (WRED) to detect congestion on the network device and mark ECN capable traffic with ECN flag.

Note: ECN is only used when both endpoints support it and are willing to use it.

Packets are ECN marked based on WRED as follows:

- If average queue size (ie. the number of packets in the queue) is below the minimum threshold, packets are queued as in normal operation without ECN.
- If average queue size is greater than the maximum threshold, packets are marked for congestion.
- If average queue size is between minimum and maximum queue threshold, packets are either queued or marked. The proportion of packets that are marked increases linearly from 0% at the minimum threshold to 100% at the maximum threshold.

ECN Configuration

- ECN is configured at the egress Tx-Queue of an Interface

```
arista(config)#interface ethernet 6/1/1
arista(config-if-Et6/1/1)#tx-queue 6
arista(config-if-Et6/1/1-txq-6)#random-detect ecn minimum-threshold 500 kbytes maximum-
threshold 1500 kbytes max-mark-probability 20
arista(config-if-Et6/1/1-txq-6)#
```

- Enable ECN Counters under the Tx-Queue

```
arista(config-if-Et6/1/1-txq-6)#random-detect ecn count
```

- Enable ECN Counter feature in Hardware

```
arista(config)#hardware counter feature ecn out
arista(config)#show hardware counter feature | grep -i ECN
ECN             out              Jericho2: 1                      up
```

- On DCS-7280R, DCS-7280R2, DCS-7500R, DCS-7500R2, DCS-7280R3, DCS-7500R3 and DCS-7800R3, the following CLI is required to allocate counter resources for ECN counters.

```
arista(config)# [no | default] hardware counter feature ecn out
```

PFC Watchdog

Priority Flow Control (PFC) Watchdog feature monitors the switch interfaces for priority-flow-control pause storms. If such storms are detected, it takes actions such as:

- Disable reacting to received Pause frames
- Stop sending packets to these interfaces and drop any incoming packets from these interfaces.

PFC Pause storm reception is usually an indication of a misbehaving node downstream, and propagating this congestion upstream is not desired. Note that the granularity of monitoring is per port and per priority.

Configuration

Configure the interval to poll the queues:

```
arista(config)# priority-flow-control pause watchdog default polling-interval ?
  <0.1 - 30> Polling interval in seconds
```

Configure the interval after which port should start dropping packets on congested priorities:

```
arista(config)# priority-flow-control pause watchdog default timeout ?
<0.2-60>  Timeout value in seconds
```

Configure the interval after which stuck ports, priorities when clear of PFC Pause storm should recover and start forwarding:

```
arista(config)# priority-flow-control pause watchdog default recovery-time ?
  <0.2-60>  Recovery time in seconds
```

Configure the PFC Watchdog action to be drop:

```
arista(config)# priority-flow-control pause watchdog action drop
```

If the drop action is not configured, the default action is to stop reacting to PFC Pause frames received on the (port, priority) experiencing the PFC Pause storm.

Show commands

```
# show priority-flow-control counters watchdog
Port        TxQ   Total times   Total times
                  stuck         recovered
-------    ----   -----------   -----------
Et3         UC1             6             6
```

Deploying RoCE on Broadcom Ethernet NIC Adapters

Designed for cloud scale and enterprise environments, Broadcom Ethernet NIC Adapters are the ideal solution for network connectivity for high performance computing, secure datacenter connectivity and AI/ML applications. Broadcom supports a broad portfolio of Ethernet NIC Adapters ranging from 1Gbps – 200Gbps port speeds and delivers best-in-class performance, hardware acceleration and offload capabilities that result in higher throughput, higher CPU efficiency, and lower workload latency for TCP/IP as well as RoCE traffic. RoCE is supported on Ethernet adapters based on BCM575xx (Thor) ASIC and the adapters support 10GE, 25GE, 100GE and 200GE speeds. The Broadcom Ethernet NIC adapters with RoCE support are available in both OCP and PCIE form factors and are summarized in Table 2 and Table 3 below.

Table 2: Broadcom OCP3.0 NIC Adapters with RoCE support			
Part Number	ASIC	Ports	I/O
BCM957504-N425G	BCM57504	4x 25G	SFP28
BCM957504-N1100G	BCM57504	1x 100G	QSFP56
BCM957504-N1100GD	BCM57504	1x 100G	DSFP
BCM957508-N2100G	BCM57508	2x 100G 1x 200G	QSFP56
BCM957508-N1200G	BCM57508	1x200G	QSFP56

Table 3: Broadcom PCIE NIC Adapters with RoCE support			
Part Number	ASIC	Ports	I/O
BCM957504-P425G	BCM57504	4x 25G	SFP28
BCM957508-P2100G	BCM57508	2x 100G 1x 200G	QSFP56
BCM957508-P1200G	BCM57508	1x 200G	QSFP56

RoCE (RDMA over converged Ethernet) is a complete hardware offload feature supported on Broadcom Ethernet NIC controllers, which allows RDMA functionality over an Ethernet network. RoCE helps to reduce CPU workload as it provides direct memory access for applications, bypassing the CPU.

RoCE Congestion Control on Broadcom Ethernet NIC Adapters

Broadcom Ethernet NIC adapters support two congestion control (CC) modes, DCQCN-P and DCQCN-D, where DCQCN-P utilizes Probabilistic ECN marking policy, with marking probability increasing linearly within a range of congested queue levels, while DCQCN-D utilizes Deterministic ECN marking policy as in DCTCP where 100% of the packets are marked when congested queue level rise above a configured threshold.

In both modes the NIC performs very similar operations and utilizes the same infrastructure to control the rate of each flow (Queue Pair, or QP, in RoCE terminology). But since the number of ECN marked packets and hence CNPs differ, the computation of congestion level is different.

In DCQCN-P there are fewer CNPs than in DCQCN-D since when congested queue level starts to rise, only a small percentage of packets traversing the switch are ECN marked. Some of the flows which do receive CNPs reduce their rate while others do not. If congestion persists, a higher percentage of packets are marked, and more flows possibly receive a signal from the network and reduce their rate. Thus, when there are many competing flows, the congested queue level may rise to higher level until stabilizing in comparison with DCQCN-D. On the other hand, since there are more CNPs with DCQCN-D, there is a higher load on the NIC in processing the stream of CNPs and accessing the associated flow context.

The CC algorithm in Broadcom Ethernet NIC adapters has been enhanced relative to the original DCQN paper due to several issues in the original algorithm. For more details, refer to the congestion control for RoCE whitepaper for Broadcom Ethernet adapters.

Installation guide for Broadcom Ethernet NIC Adapters

Broadcom Ethernet User Guide, available publicly, provides detailed instructions on how to install RoCE on Broadcom Ethernet Network Adapters.

This section talks about the procedures to install Broadcom Ethernet adapters and to configure RoCE.

Updating the Firmware on Broadcom Ethernet NIC Adapters

The following bnxtnvm command is used to update the adapter firmware on Broadcom Ethernet NIC adapters. Note that the bnxtnvm command requires sudo or root access.

```
sudo bnxtnvm -dev=<host network interface name> install <firmware package>
```

Example:

```
sudo bnxtnvm -dev=ens2f0np0 install BCM957508-N1200G.pkg
```

Configuring NVRAM

To update the NVRAM configuration, use the bnxtnvm utility provided with the release.

Run bnxtnvm version to check the version you are using.

- Ensure that RDMA is enabled for the specific PF.
- For RoCE performance, the performance profile NVM CFG must be set to RoCE (value 1).

NOTE: A host reboot is required for the new settings to take effect.

Verify the RDMA and performance settings with the following commands:

```
sudo bnxtnvm -dev=ens2f0np0 getoption=support_rdma:0
```

```
sudo bnxtnvm -dev=ens2f0np0 getoption=performance_profile
```

The output value for the support_rdma parameter should read Enabled and the value for performance_profile should read RoCE.

To enable RDMA for the specific PF and to set the performance profile to RoCE, use the following commands:

```
sudo bnxtnvm -dev=ens2f0np0 setoption=support_rdma:0#1
```

```
sudo bnxtnvm -dev=ens2f0np0 setoption=performance_profile#1
```

NOTE: The portion of the command that is dark red changes depending on the name of the host network interface.

Reboot the system after setting the NVRAM options.

Host Requirements for Driver/Library Compilation

Compiling the driver and library has dependencies on build packages such as automake, libtool, make, gcc, and so forth. The following packages are recommended based on the OS distribution being used.

» CentOS/Redhat/Fedora

See the following commands for CentOS, Redhat, and Fedora operating systems:

```
dnf group install “Development Tools”
```

```
dnf group install “Infiniband Support”
```

» Ubuntu/Debian

See the following commands for Ubuntu or Debian operating systems:

```
apt install autoconf automake bc bison build-essential flex libtool
```

```
apt install ibverbs-utils infiniband-diags libibverbs-dev perftest
```

Installing the Layer 2 and RoCE Driver

This section describes how to install the Layer 2 communication (L2) and RoCE driver. The installation tarball contains the netxtreme-bnxt_en-<version>.tar.gz file. This file includes both the L2 and RoCE drivers.

Install the drivers using the following commands:

```
BRCM_DRIVER_VERSION=1.10.2-227.0.130.0
```

```
tar xvf netxtreme-bnxt_en-${BRCM_DRIVER_VERSION}.tar.gz

cd netxtreme-bnxt_en-${BRCM_DRIVER_VERSION}.tar.gz
make
sudo make install
sudo depmod -a
```

Updating Initramfs

Most Linux distributions use a ramdisk image to store drivers for boot-up. These kernel modules take precedence, so the initramfs must be updated after installing the new bnxt_en/bnxt_re modules. For CentOS, Redhat, and Fedora operating systems, use sudo dracut -f and for Ubuntu/Debian operating systems use sudo update -initramfs -u.

Installing the RoCE Library

This section describes how to install the RoCE library. The installation tarball contains the libbnxt_re- <version>.tar.gz file. This file includes the libbnxt_re RoCE library.

Execute the following steps.

1. To avoid potential conflicting library files, remove or rename the libbnxt RoCE library from the Linux distribution using the following command. The command is a single command and tries to locate the inbox libbnxt_re library in one of the previous directories. It may be necessary to run it as a sudo user.

   ```
   find /usr/lib64 /usr/lib /lib64 -name "libbnxt_re-rdmav*.so" -exec mv {} {}.inbox \;
   ```

2. Build and install the userspace RDMA library from the source using the following commands. See Host Requirements for Driver/ Library Compilation for information regarding host package dependencies that are required for building the RoCE library from source. Note that the portion of the command that is dark red below is release specific.

   ```
   BRCM_LIB_VERSION=227.0.130.0
   tar xvf libbnxt_re-${BRCM_LIB_VERSION}.tar.gz
   cd libbnxt_re-${BRCM_LIB_VERSION}
   sh autogen.sh
   ./configure --sysconfdir=/etc
   make
   sudo make install all
   sudo sh -c "echo /usr/local/lib >> /etc/ld.so.conf"
   sudo ldconfig
   ```

3. Record the md5sum of the library that was built to verify that the correct library is running using the following command.

   ```
   find . -name "*.so" -exec md5sum {} \;
   ```

4. Use the following commands to identify the path of the libbnxt_re library being used on the host and then calculate its md5sum. The md5sum should match the md5sum of the built libraries in the previous step.

   ```
   strace ibv_devinfo 2>&1 | grep libbnxt_re | grep -v 'No such file'
   md5sum <path of the lbbnxt_re library> shown by the last command
   ```

Validating the RoCE Installation

After the drivers and libraries are installed, perform the following steps to validate RoCE installation.

Confirming the GUID for the RoCE Interface

The node GUID indicates that RoCE has been successfully configured on the system. There are two commands that can be used to confirm the GUID for the RoCE interface:

- ibv_devices – indicates if the GUID is available.

- ibv_devinfo – indicates if the GUID is available and provides additional details about the RoCE interface.

```
#ibv_devices
    device                 node GUID
    ------              ----------------
   bnxt_re0             be97e1ffeda96d0

# ibv_devinfo
hca_id:                 bnxt_re0
transport:              InfiniBand (0)
fw_ver:                 227.1.111.0
node_guid:              be97:e1ff:feda:96d0
sys_image_guid:         be97:e1ff:feda:96d0
vendor_id:              0x14e4
vendor_part_id:         5968
hw_ver:                 0x1200
phys_port_cnt:          1
port:                   1
state:                  PORT_ACTIVE (4)
max_mtu:                4096 (5)
active_mtu:             4096 (5)
sm_lid:                 0
port_lid:               0
port_lmc:               0x00
link_layer:             Ethernet
```

Confirm Traffic Flow to the remote RoCE endpoint

If the RoCE endpoint is currently configured, traffic flow can be verified by using one of the perftest package utilities.

Command usage: `ib_write_bw -d bnxt_re0 -F -x 3--report_gbits <ip address of remote end point>` `ib_write_bw -d bnxt_re0 -F -x 3--report_gbits 192.168.2.30`

```
---------------------------------------------------------------------
                    RDMA_Write BW Test
Dual-port       : OFF                   Device         : bnxt_re0
Number of qps   : 1                     Transport type : IB
Connection type : RC                    Using SRQ      : OFF
TX depth        : 128
CQ Moderation   : 100
Mtu             : 4096[B]
Link type       : Ethernet
GID index       : 1
Max inline data : 0[B]
rdma_cm QPs     : OFF
Data ex. method : Ethernet
---------------------------------------------------------------------
local address: LID 0000 QPN 0x06c9 PSN 0xc5d95b RKey 0x2000212 VAddr 0x007fa3640ed000
GID: 00:00:00:00:00:00:00:00:00:00:255:255:192:168:02:20
```

```
remote address: LID 0000 QPN 0x06cb PSN 0x448da7 RKey 0x2000308 VAddr 0x007f9edfec1000
GID: 00:00:00:00:00:00:00:00:00:00:255:255:192:168:02:30
------------------------------------------------------------------
#bytes     #iterations    BW peak[Gb/sec]    BW average[Gb/sec] MsgRate[Mpps]
 65536      5000             196.92                193.53 0        .371852
```

Configuring Priority Flow Control on Broadcom NICs

Broadcom's RoCE driver enables three traffic classes (L2, RoCE, and Congestion Notification Packet (CNP)). Loading the driver automatically sets up the default RoCE/CNP Priority Code Point (PCP) priorities and Differentiated Services Code Point (DSCP) values. Priority Flow Control (PFC) and Congestion Control (CC) are enabled by default and the default values are listed in Table 3. No other configuration is required on the host if the upstream switches are configured with these default values.

Table 4: Broadcom NIC - Default PCP and DSCP Values

Traffic Type	Default PCP value	Default DSCP value
RoCE	3	26
CNP	7	48

The default traffic classes are:

- TC0 (L2 Traffic)
- TC1 (RoCE Traffic)
- TC2 (CNP Traffic)

In the absence of L2 traffic, the full bandwidth is allotted for RoCE traffic.

To change the default values of PCP and DSCP to match the user's network settings, bnxt_setupcc.sh script can be used. The script is provided as part of the binary package from Broadcom.

1. Install the bnxtqos RPM package using the following command:

   ```
   sudo rpm –ivh bnxtqos–<version>.rpm
   ```

   The bnxt_setupcc.sh script is installed in the /usr/bin directory.

2. Configure the RoCE and CNP priorities using the following command:

   Command usage:

   ```
   sudo bnxt_setupcc.sh –d <x> –i <RoCE interface> –m <x> –s <RoCE DSCP value> –p <CNP DSCP
   value> –r <RoCE PCP value> –c <CNP DSCP value>
   ```

   Example:

   ```
   sudo bnxt_setupcc.sh –d bnxt_re0 –i ens4f0np0 –m 3 –s 32 –p 36 –r 5 –c 6
   ```

The sample command shown in the previous example is configuring:

- RoCE PCP priority 5 and DSCP value 32
- CNP PCP priority 6 and DSCP value 36

Configuring Congestion Control on Broadcom NICs

To adjust the congestion control parameter, the Broadcom RoCE driver relies on the kernel configfs. The default congestion control algorithm is DCQCN-P. To change to DCQCN-D algorithm, use the following procedure.

NOTE: In addition to setting cc_mode to DCQCN, it is recommended to tune other advanced parameters for optimal performance.

- Configuring DCQCN-D
  To configure DCQCN-D, use the following commands:
  ```
  mkdir -p /sys/kernel/config/bnxt_re/bnxt_re0
  cd /sys/kernel/config/bnxt_re/bnxt_re0/ports/1/cc/
  echo -n 0 > cc_mode
  echo -n 1 > apply
  ```
- Configuring DCQCN-D
  To configure DCQCN-D, use the following commands:
  ```
  echo -n 1 > cc_mode
  echo -n 1 > apply
  ```
- Viewing the Current Congestion Control Parameters
  To view the currently configured congestion control parameters, use the following commands:
  ```
  mkdir -p /sys/kernel/config/bnxt_re/bnxt_re0
  cd /sys/kernel/config/bnxt_re/bnxt_re0/ports/1/cc/
  echo -n 1 > advanced
  echo -n 1 > apply
  cat apply
  ```

RoCE Performance Data

For measuring performance numbers on the cluster with Broadcom NICs (100GE/200GE) and Arista switches (200GE/400GE) for this user guide, OSU MPI Benchmark and GPCNet Benchmark tests have been used. The cluster configuration used for the benchmark tests is captured in Table 5 and 5.1 below.

Table 5: Cluster Configuration for performance tests on 100GE

Server	Switch	NIC	Benchmarks
Model: Dell R740 CPU: Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz Thread(s) per core: 2 Core(s) per socket: 16 Memory Type: DDR4 - 2666 MT/s Memory: 96 GB (48GB/Socket) Kernel: 5.11.0-44 (Ubuntu 20.04)	Model: Arista DCS-7060DX4-32S-F (TH3 TOR) Hardware Revision: 11.01 Software Version: 4.27.2F	Model: Broadcom P2100G Driver Version: 223.1.25.0 Firmware Version: 223.1.25.0 Congestion Control OSU: DCQCN-p Congestion Control GPCNet: DCQCN-d	UCX: 1.10.0 OpenMPI: 4.1.1 GPCNet: 1.1 OSU: 5.8

Table 5.1: Cluster Configuration for performance tests on 200GE

Server	Switch	NIC	Benchmarks
Model: Dell R750 CPU: Intel(R) Xeon(R) Gold 8358 CPU @ 2.60GHz Thread(s) per core: 1 Core(s) per socket: 32 Memory Type: DDR4 -3200 MT/s Memory: 128 GB (64GB/Socket) Kernel: 5.11.0-44 (Ubuntu 20.04.6 LTS)	Model: Arista DCS-7060DX4-32S-F (TH3 TOR) Hardware Revision: 11.01 Software Version: 4.28.1FX-7060DX4.1	Model: Broadcom P1200G Driver Version: 227.0.131.0 Firmware Version: 227.0.131.0 Congestion Control OSU: DCQCN-p Congestion Control GPCNet: DCQCN-d	UCX: 1.13.1 OpenMPI: 4.1.4 GPCNet: 1.1 OSU: 5.8

A summary of the performance numbers from various benchmark tests are captured in Table 6 and 6.1 below.

Table 6: Broadcom NIC / Arista switch ROCE Performance on 100GE		
Category	Test	Broadcom P2100 NIC
Baseline	OSU Benchmark osu_mbw_mr, 128KB, 4 nodes, 16 PPN	48.6 Gbps (389 Gbps)
Collectives	OSU Benchmark Blocking osu_alltoall latency, 16 nodes, 16 PPN, 128KB message	47.8 ms
	OSU Benchmark Blocking osu_allreduce latency, 16 nodes, 16 PPN, 128KB message	393 us
Congestion Control	GPCNet Benchmark Random Ring Two-sided Latency under congestion, 16 nodes, 16 PPN (99 percentile latency)	9.3 us
	GPCNet Benchmark Multiple Allreduce Latency under congestion, 16 nodes, 16 PPN (99 percentile latency)	39.2 us

Table 6.1: Broadcom NIC / Arista switch ROCE Performance on 200GE		
Category	Test	Broadcom P2100 NIC
Baseline	OSU Benchmark osu_mbw_mr, 128KB, 4 nodes, 16 PPN	95.8 GB/s (766 Gbps)
Collectives	OSU Benchmark Blocking osu_alltoall latency, 16 nodes, 16 PPN, 128KB message	28.12 ms
	OSU Benchmark Blocking osu_allreduce latency, 16 nodes, 16 PPN, 128KB message	369 us
Congestion Control	GPCNet Benchmark Random Ring Two-sided Latency under congestion, 16 nodes, 16 PPN (99 percentile latency)	31.40 us
	GPCNet Benchmark Multiple Allreduce Latency under congestion, 16 nodes, 16 PPN (99 percentile latency)	69.5 us

OSU MPI Multiple Bandwidth / Message Rate (osu_mbw_wr) Test

The focus of the multi-pair bandwidth and message rate test is to evaluate the aggregate uni-directional bandwidth and message rate between multiple pairs of processes. Each of the sending processes sends a fixed number of messages (the window size) back-to-back to the paired receiving process before waiting for a reply from the receiver. This process is repeated for several iterations. The objective of this benchmark is to determine the achieved bandwidth and message rate from one node to another node with a configurable number of processes running on each node.

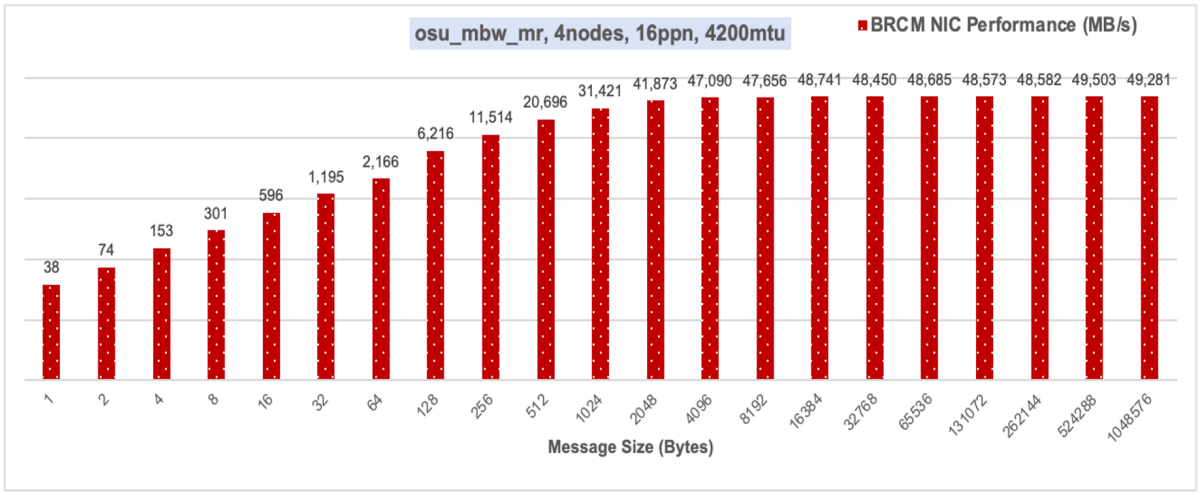

Figure 3: osu_mbw_wr benchmark test with Broadcom NICs and Arista Switches on 100GE

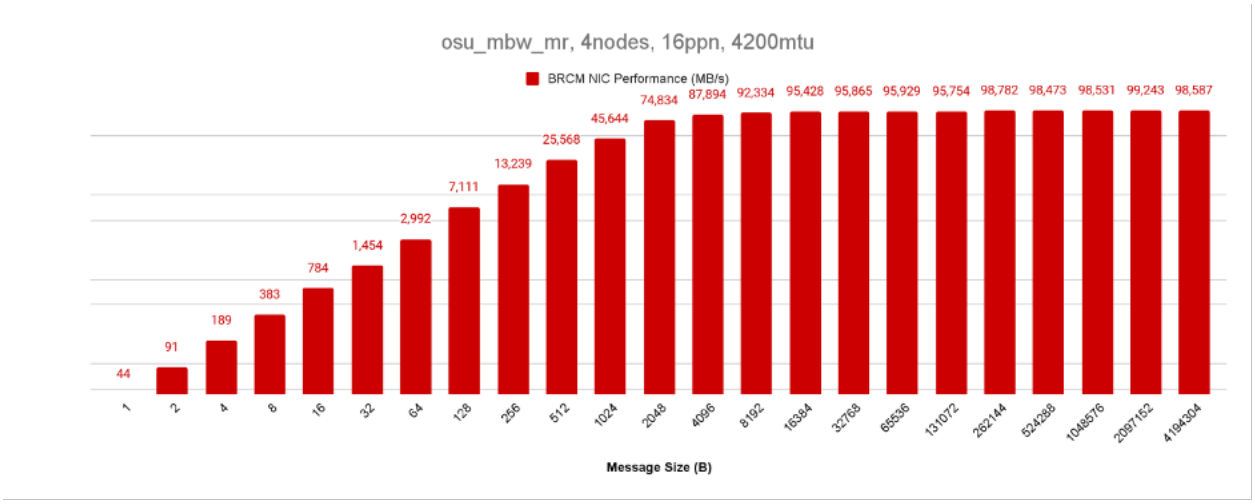

Figure 3.1: osu_mbw_wr benchmark test with Broadcom NICs and Arista Switches on 200GE

OSU MPI All to All (osu_alltoall) Latency Test

This benchmark test measures the min, max and the average latency of operation across N processes, for various message lengths, over many iterations and reports the average completion time for each message length.

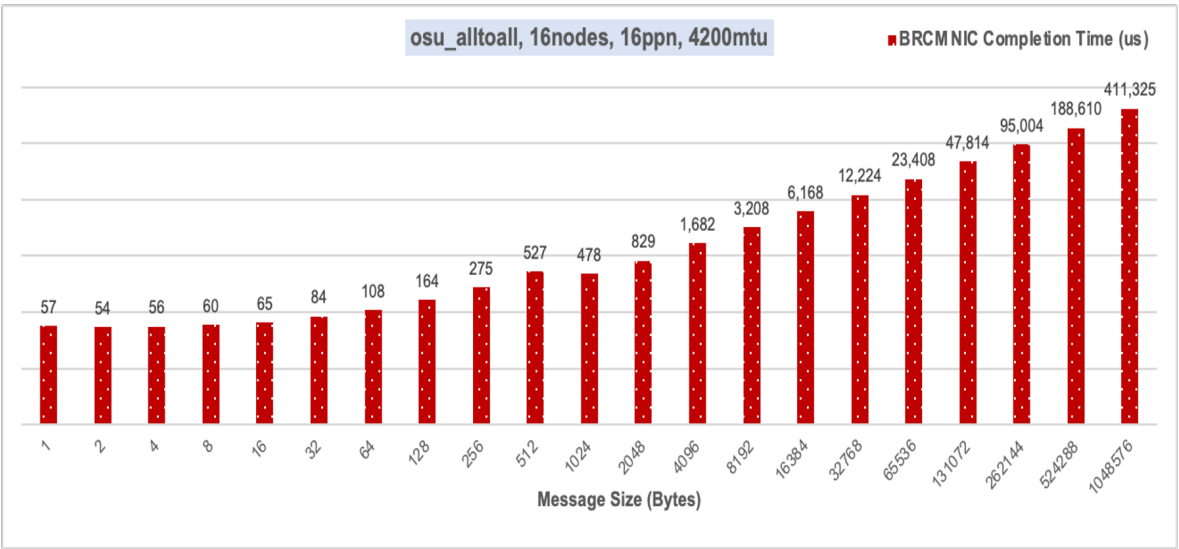

Figure 4: osu_alltoall latency test with Broadcom NICs and Arista Switches on 100GE

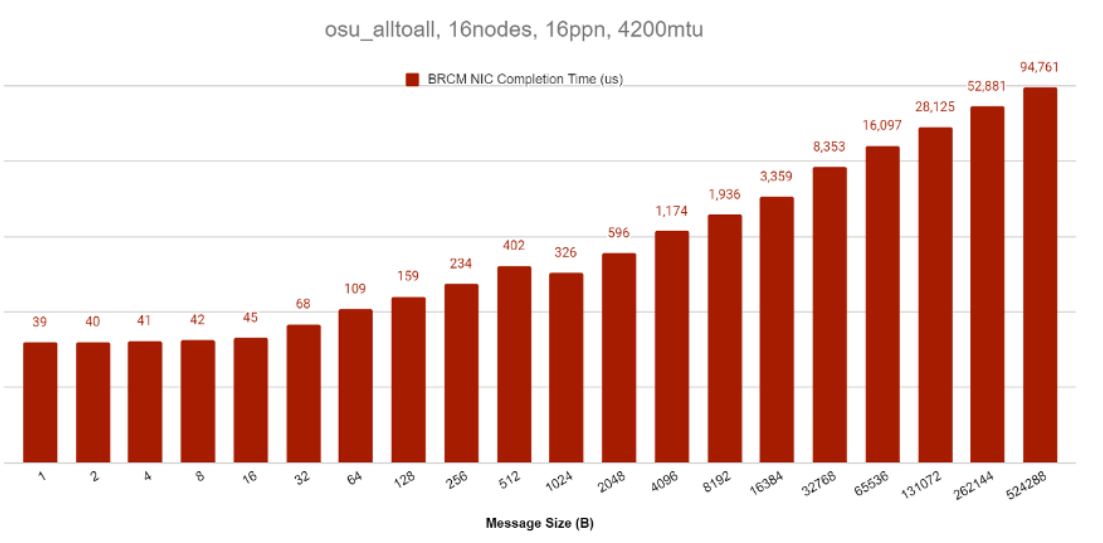

Figure 4.1: osu_alltoall latency test with Broadcom NICs and Arista Switches on 200GE

OSU All Reduce (osu_allreduce) Latency Test

Like osu_alltoall, osu_allreduce benchmark test measures the min, max and the average latency of operation across N processes, for various message lengths, over many iterations and reports the average completion time for each message length.

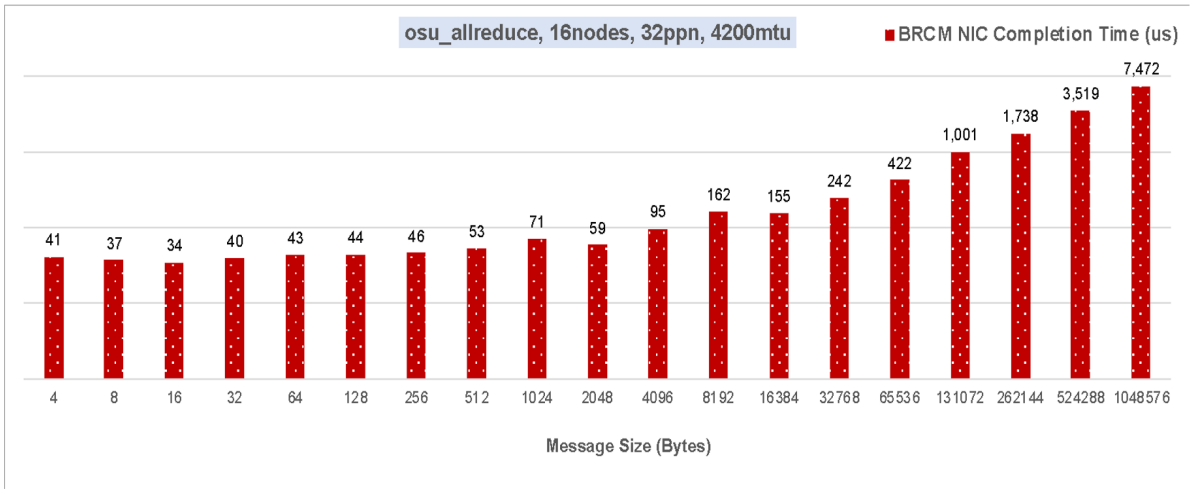

Figure 5: OSU All Reduce Latency Test with Broadcom NICs and Arista Switches on 100GE

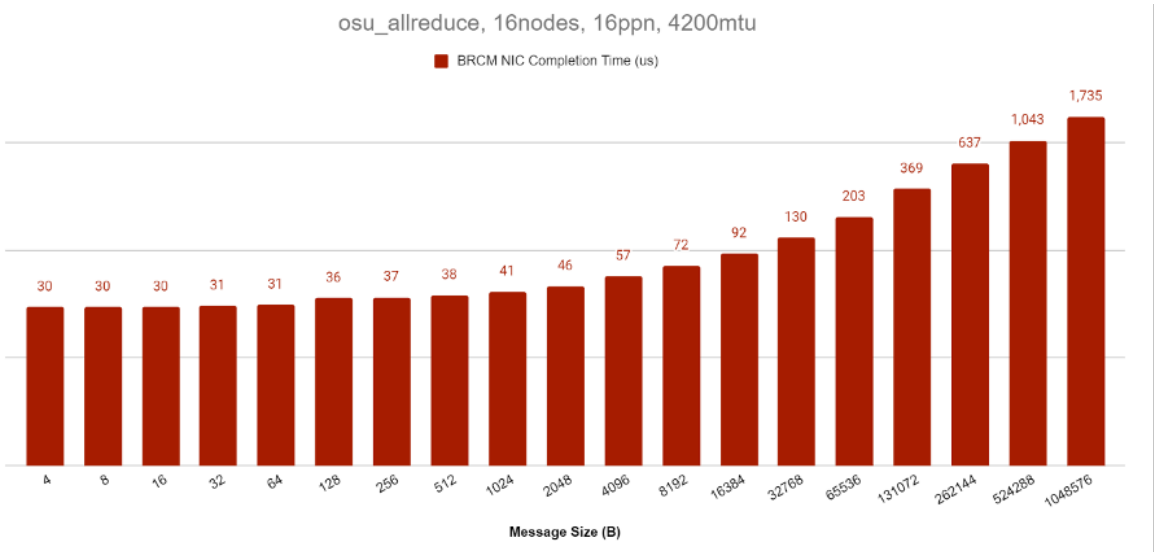

Figure 5.1: OSU All Reduce Latency Test with Broadcom NICs and Arista Switches on 200GE

GPCNet

Global Performance and Congestion Network Test (GPCNet) is a generic, topology agnostic bench-mark suite that captures the complex workloads anticipated on multitenant HPC networks. Broadcom NICs support different congestion control algorithms (dcqcn-d and dcqcn-p) for different applications. In this case, dcqcn-d maintains a shallow buffer leading to far lower completion time under congestion.

Table 7: GPCNet Benchmark test with Arista Switch and Broadcom NICs on 100GE			
gpcnet, 16 nodes, 16 ppn, 4200 mtu			
		Broadcom NIC dcqcn-d	Broadcom NIC dcqcn-p
Isolated / Unloaded	All Reduce: Lat:Avg	15.6 us	15.6 us
	All Reduce: Lat:99%	24.6 us	24.6 us
	RR Two-Sided Latency: BW:Avg	1144.7 MiB/s/rank	1245.2 MiB/s/rank
	RR Two-Sided Latency: BW:99%	748.2 MiB/s/rank	905.9 MiB/s/rank
	RR Two-Sided Latency: Lat:Avg	2.4 us	2.4 us
	RRTwos: Lat:99%	4.6 us	4.6 us
Congestion	All Reduce: Lat:Avg	23.3 us	41 us
	All Reduce: Lat:99%	39.2 us	99.3 us
	RR Two-Sided Latency: BW:Avg	734.9 MiB/s/rank	710.6 MiB/s/rank
	RR Two-Sided Latency: BW:99%	276.4 MiB/s/rank	214.2 MiB/s/rank
	RR Two-Sided Latency: Lat:Avg	4.1 us	10.1 us
	RR Two-Sided Latency: Lat:99%	9.3 us	46.4 us

Table 7.1: GPCNet Benchmark test with Arista Switch and Broadcom NICs on 200GE			
gpcnet, 16 nodes, 16 ppn, 4200 mtu			
		Broadcom NIC dcqcn-d	Broadcom NIC dcqcn-p
Isolated / Unloaded	All Reduce: Lat:Avg	14.0 us	14.0 us
	All Reduce: Lat:99%	23.0 us	23.0 us
	RR Two-Sided Latency: BW:Avg	2381.6 MiB/s/rank	2849.9 MiB/s/rank
	RR Two-Sided Latency: BW:99%	1704.9 MiB/s/rank	1328.2 MiB/s/rank
	RR Two-Sided Latency: Lat:Avg	2.1 us	2.1 us
	RRTwos: Lat:99%	4.2 us	4.2 us
Congestion	All Reduce: Lat:Avg	43.2 us	82.8 us
	All Reduce: Lat:99%	69.5 us	151.4 us
	RR Two-Sided Latency: BW:Avg	1375.6 MiB/s/rank	1352.0 MiB/s/rank
	RR Two-Sided Latency: BW:99%	595.4 MiB/s/rank	341.9 MiB/s/rank
	RR Two-Sided Latency: Lat:Avg	10.0 us	19.5 us
	RR Two-Sided Latency: Lat:99%	31.4 us	67.5 us

RR = Random Ring Communication Pattern

RoCE Applications

There are several new technologies that will allow data centers to benefit from performance improvements provided by RoCE. These include:

Peer Memory Direct

General Purpose GPU (GPU) and dedicated AI accelerators have been gaining a lot of traction in machine learning where it takes a massive amount of computing performance to train the sophisticated deep neural networks. This training can take days to weeks. Due to vast amount of data for training, it is desirable to distribute computations across multiple systems with accelerators, which result in increased data exchange for data loading to the accelerators prior to the computations and for data shuffling after the computations. This loading/shuffling process repeats as the training processes are iterative.

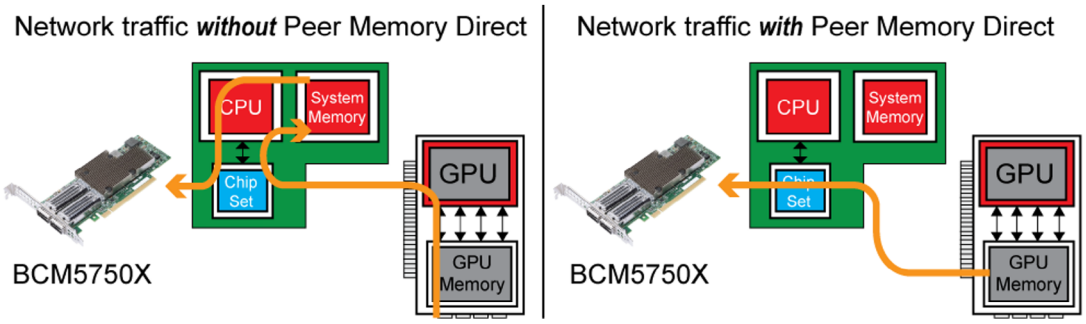

Figure 6: Peer Memory Direct Data Transfer

A closer look at these data flows reveals that the net data exchanges are moved between the accelerators on the same server or across different servers. Both types of flow indicate that a better data movement isolation can be put into place to isolate memory pressure to the memory fabrics, memory controllers and DDR memory. This is the motivation of Peer Memory Direct or AI Direct.

For the exchanges between different servers, RDMA capability on NICs is critical to offer a low latency and highly efficient and yet CPU bypass transport. Traditional RDMA software stack enables direct memory exchanges between process' memories which are server system DDR memory. Then the CPU is involved in moving data from system memory to accelerator memory. These additional movements add latencies and are subject to the limit of system memory bandwidth. Peer Memory Direct technology circumvents this by taking advantage of PCIe peer-to-peer (P2P) transfer and thereby eliminates CPU bandwidth and latency bottle necks. It also eliminates system memory copies and CPU overhead for transferring data to/from GPU memory. For configuring Peer Memory Direct with Broadcom NICs, refer to the link.

SMB Direct

Server Message Block (SMB) is an application-layer network protocol that provides shared access to files, printers, and serial ports. Microsoft provides support for high performance storage networks using RoCE and Microsoft uses SMB in this scenario.

This enables a remote file server to work like local storage with applications that use Microsoft SQL Server and Microsoft Storage Server.

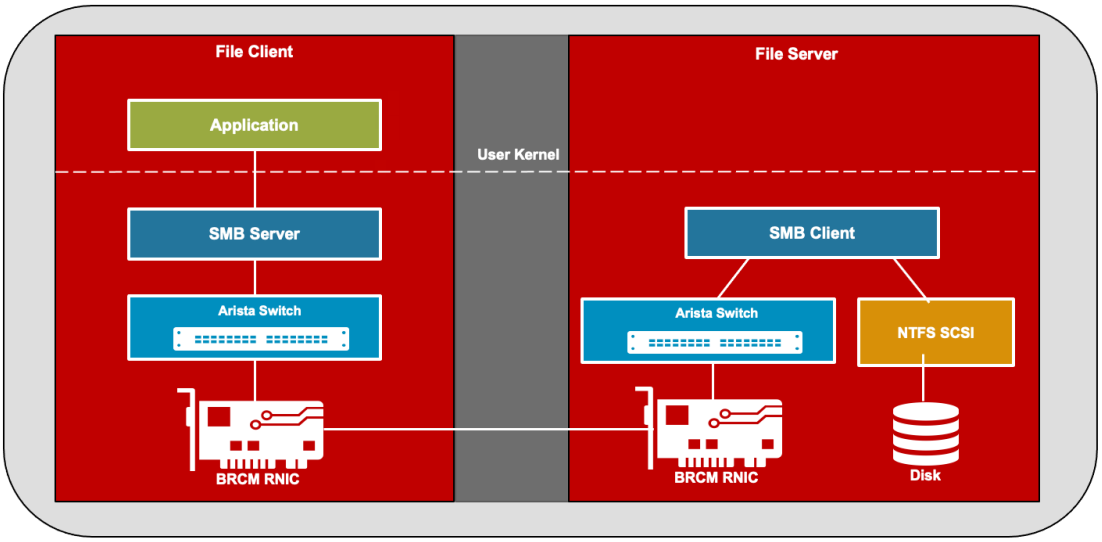

Figure 7: SMB Direct with Arista Switch and Broadcom NICs

SMB 3.0 added the SMB Direct feature that works with network adapters that support RoCE. This capability provides high-performance remote file access for servers and is ideal for use cases such as virtualization and databases. SMB Direct high-performance capabilities are also very beneficial for live migrations with Hyper-V deployments.

The combination of SMB Direct and RoCE adapters provide:

- Scalable, fast and efficient storage access
- High throughput with low latency
- Minimal CPU utilization for I/O processing
- Load balancing, automatic failover and bandwidth aggregation using SMB Multichannel

iSCSI Extensions for RDMA

Performance for Internet Small Computer System Interface (iSCSI) storage has also been enhanced with iSCSI extensions for RDMA (iSER). The iSER protocols are defined in RFCs 5047 and 7145 and enable RDMA to be used to transfer data directly between memory buffers for computers and storage devices.

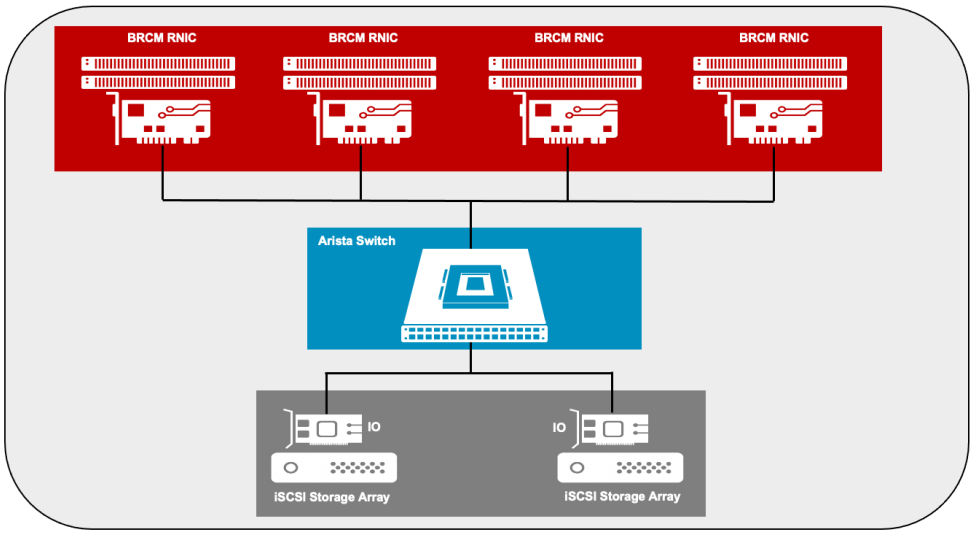

Figure 8: iSER with Arista Switch and Broadcom NICs

iSER promises to provide significant performance improvements over iSCSI due to eliminating the TCP/IP processing overhead, this becomes significant with increased Ethernet speeds of 10/25/100/200GbE and beyond. iSER will provide higher throughput for storage applications, lower latency and more efficient use of server and storage controller processing resources.

NFS over RDMA

Network file system (NFS) is a distributed file system protocol that allows users on client computers to access files over a network as if it was local storage. NFS is an open standard defined with request for comments (RFCs) that enable ongoing development and implementation of new technologies. One focus area has been the remote procedure call (RPC) layer for NFS that provides communication between the client and server. RDMA support has been added to the RPC layer with RFCs 5532, 5666 and 5667 to provide enhanced data transfer performance.

Using RoCE for NFS over RDMA has the potential for similar performance benefits as SMB Direct for increasing performance of applications servers that use network file storage. NFS clients and servers can expect higher throughput at smaller data block sizes as well as increased I/O operations per second (IOPS), lower latency and reduced NFS client and server CPU consumption.

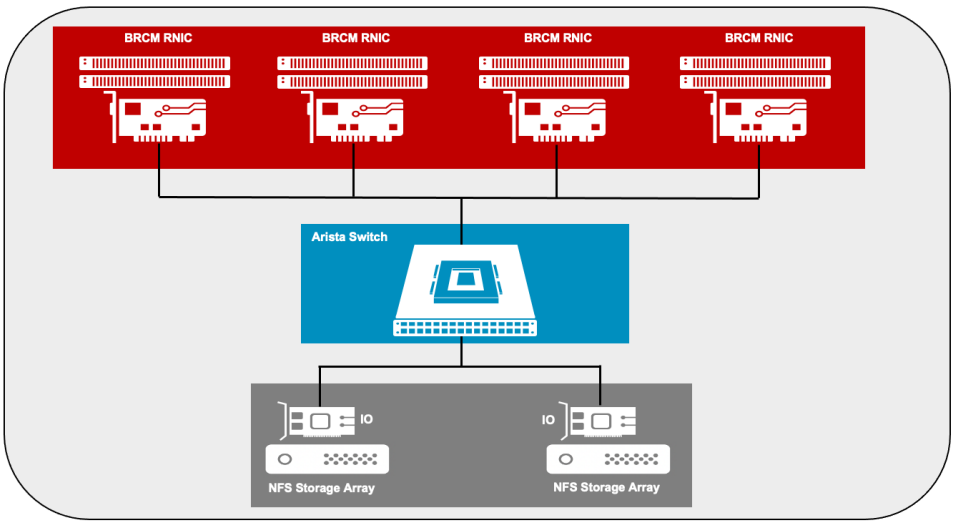

Figure 9: NFS over RDMA with Arista Switch and Broadcom NIC

NVMe-oF over RDMA

The NVMe protocol is not limited to simply connecting a local flash drive inside a server, it may also be used over a network. When used in this context, a network "fabric" enables any-to-any connections among storage and server elements. NVMe over Fabrics (NVMe-oF) is enabling organizations to create a very high-performance storage network with latency that rival direct attached storage. As a result, fast storage devices can be shared, when needed, among servers. NVMe over fabric is an alternative to SCSI on Fibre Channel or iSCSI, with the benefit of a lower latency, a higher I/O rate and an improved productivity.

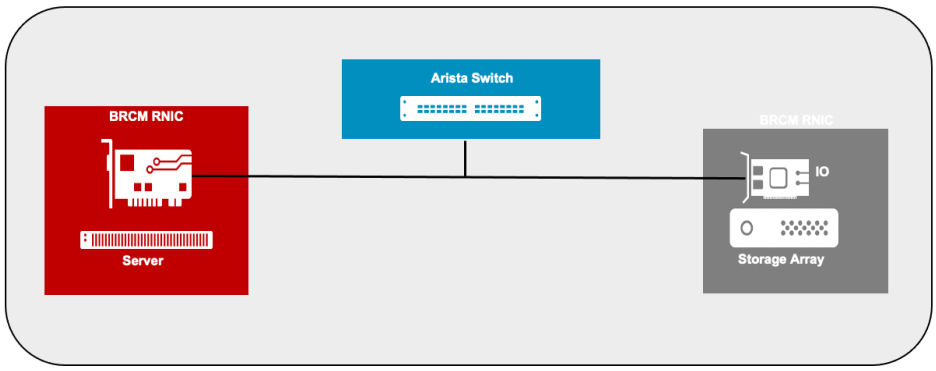

Figure 10: NVMe-oF over RDMA with Arista Switch and Broadcom NICs

Summary

RDMA is a proven technology to provide greater performance and scalability. With the heavy compute requirements associated with intensive AI/ML & storage workloads, RoCE is a fundamental component in ensuring a reliable, end-to-end transport solution for modern data centers. Arista and Broadcom are committed to support open standards-based congestion control mechanisms on the switches and NICs and are at the forefront to meet the requirements of datacenter applications to deliver a reliable, high throughput, low latency network. Deployments leveraging Arista switches and Broadcom NICs also benefit from the power efficient ethernet solutions which are extremely critical for data centers today and contribute to TCO savings.

References

- Arista Cloud Grade Routing Products
- Arista Hyper Scale Data Center Platforms
- Arista EOS Quality of Service
- Arista Priority Flow Control (PFC) and Explicit Congestion Notification (ECN)
- Arista Configuration Guide
- Arista EOS Software Downloads
- Arista AI Networking
- Arista CloudVision
- Arista Broadcom RoCE Datasheet
- Broadcom Ethernet Network Adapters
- Broadcom Ethernet NIC Configuration Guide
- Broadcom Ethernet NIC Firmware and Drivers Downloads
- Broadcom RoCE Configuration Guide
- Broadcom Ethernet NIC Congestion Control
- Congestion Control for Large-Scale RDMA Deployments
- Configuring Peer Memory Direct with Broadcom NICs

Santa Clara—Corporate Headquarters
5453 Great America Parkway,
Santa Clara, CA 95054

Phone: +1-408-547-5500
Fax: +1-408-538-8920
Email: info@arista.com

Ireland—International Headquarters
3130 Atlantic Avenue
Westpark Business Campus
Shannon, Co. Clare
Ireland

Vancouver—R&D Office
9200 Glenlyon Pkwy, Unit 300
Burnaby, British Columbia
Canada V5J 5J8

San Francisco—R&D and Sales Office 1390
Market Street, Suite 800
San Francisco, CA 94102

India—R&D Office
Global Tech Park, Tower A, 11th Floor
Marathahalli Outer Ring Road
Devarabeesanahalli Village, Varthur Hobli
Bangalore, India 560103

Singapore—APAC Administrative Office
9 Temasek Boulevard
#29-01, Suntec Tower Two
Singapore 038989

Nashua—R&D Office
10 Tara Boulevard
Nashua, NH 03062

Appendix A - RoCE Traffic Example

In this example, RDMA traffic is sent from SUT to Client 2, Client 3 and Client 4 and back, all with Broadcom Ethernet NICs and with Arista switches in the transit path as shown in the topology below. The commands and their respective outputs to verify statistics on each of the nodes are captured below.

SUT with Broadcom THOR NIC:

The ethtool utility displays the associated statistics for the interface (enp8s0f0np0 in the below example is the interface name). RoCE-specific statistics, including congestion control statistics, can be viewed using the Linux sysfs interface of the RoCE interface. The following example shows an example of RoCE statistics from the sysfs interface using bnxt_re0 as the RoCE interface name. Note that for brevity, counters which are zero are omitted in the outputs below and the important counters are highlighted in blue color in the outputs.

```
# ethtool -S ens7f0np0
NIC statistics:
     [0]: tx_ucast_packets: 946
     [0]: tx_ucast_bytes: 136862
     [1]: rx_ucast_packets: 1417
     [1]: rx_ucast_bytes: 205215
     [1]: tx_ucast_packets: 5
     [1]: tx_ucast_bytes: 270
     [2]: rx_ucast_packets: 492
     [2]: rx_ucast_bytes: 69703
     [2]: tx_ucast_packets: 482
     [2]: tx_ucast_bytes: 69031
     [3]: rx_ucast_packets: 913
     [3]: rx_ucast_bytes: 134714
     [3]: tx_ucast_packets: 596
     [3]: tx_ucast_bytes: 76567
     [4]: rx_ucast_packets: 878
     [4]: rx_ucast_bytes: 132404
     [4]: tx_ucast_packets: 2162
     [4]: tx_ucast_bytes: 291616
     [5]: tx_ucast_packets: 1
     [5]: tx_ucast_bytes: 54
     [6]: rx_ucast_packets: 567
     [6]: rx_ucast_bytes: 74653
     [6]: tx_ucast_packets: 544
     [6]: tx_ucast_bytes: 73135
     [7]: tx_ucast_packets: 1
     [7]: tx_ucast_bytes: 54
     rx_64b_frames: 2
     rx_65b_127b_frames: 1200
     rx_128b_255b_frames: 3098
     rx_1024b_1518b_frames: 669333143
     rx_total_frames: 669337443
     rx_ucast_frames: 669337410
     rx_mcast_frames: 33
     rx_good_frames: 669337443
rx_bytes: 737605761628
```

```
root@hpca4009:~# cat /sys/kernel/debug/bnxt_re/bnxt_
re0/info
bnxt_re debug info:
=====[ IBDEV bnxt_re0 ]============================
        link state: UP
        Max QP:          65537
        Max SRQ:         4096
        Max CQ:          65536
        Max MR:          262144
        Max MW:          262144
        Max AH:          65536
        Max PD:          65536
        Active QP:       1
        Active SRQ:      0
        Active CQ:       1
        Active MR:       1
        Active MW:       1
        Active AH:       0
        Active PD:       1
        QP Watermark:    3070
        SRQ Watermark:   33
        CQ Watermark:    131
        MR Watermark:    2950
        MW Watermark:    1
        AH Watermark:    284
        PD Watermark:    34
        Rx Pkts: 669333143
        Rx Bytes: 734927791014
        Tx Pkts: 1320298148
        Tx Bytes: 37785112622
        CNP Tx Pkts: 668830689
        RoCE Only Rx Pkts: 669333143
        RoCE Only Rx Bytes: 734927791014
        RoCE Only Tx Pkts: 651467459
        RoCE Only Tx Bytes: 37785112622
        rx_write_req: 669333143
        rx_good_pkts: 669333143
        rx_good_bytes: 734927791014
```

```
    tx_64b_frames: 11
    tx_65b_127b_frames: 1320299778
    tx_128b_255b_frames: 3096
    tx_good_frames: 1320302885
    tx_total_frames: 1320302885
    tx_ucast_frames: 1320302885
    tx_bytes: 95166312639
    rx_bytes_cos0: 734927791014
    rx_packets_cos0: 669333143
    rx_bytes_cos4: 620842
    rx_packets_cos4: 4300
    tx_bytes_cos0: 84603260852
    tx_packets_cos0: 1320298148
    tx_bytes_cos4: 647589
    tx_packets_cos4: 4737
    tx_bytes_cos5: 46818148230
    tx_packets_cos5: 668830689
    rx_bytes_pri0: 620842
    rx_bytes_pri1: 620842
    rx_bytes_pri2: 620842
    rx_bytes_pri3: 734927791014
    rx_bytes_pri4: 620842
    rx_bytes_pri5: 620842
    rx_bytes_pri6: 620842
    rx_packets_pri0: 4300
    rx_packets_pri1: 4300
    rx_packets_pri2: 4300
    rx_packets_pri3: 669333143
    rx_packets_pri4: 4300
    rx_packets_pri5: 4300
    rx_packets_pri6: 4300
    tx_bytes_pri0: 647589
    tx_bytes_pri1: 647589
    tx_bytes_pri2: 647589
    tx_bytes_pri3: 84603260852
    tx_bytes_pri4: 647589
    tx_bytes_pri5: 647589
    tx_bytes_pri6: 647589
    tx_bytes_pri7: 46818148230
    tx_packets_pri0: 4737
    tx_packets_pri1: 4737
    tx_packets_pri2: 4737
    tx_packets_pri3: 1320298148
    tx_packets_pri4: 4737
    tx_packets_pri5: 4737
    tx_packets_pri6: 4737
    tx_packets_pri7: 668830689
    rx_ecn_marked_pkts: 668830689
    seq_err_naks_rcvd:
    latency_slab [0 - 1] sec = 139902
```

TOR (Arista 7060X) Counters:

Arista_th3_g07 TOR1

```
arista-th3-g07#show interface ethernet 18/1 counters
| json | no-more
{
    “interfaces”: {
        “Ethernet18/1”: {
            “outBroadcastPkts”: 71,
            “outUcastPkts”: 669337412,
            “inMulticastPkts”: 0,
            “lastUpdateTimestamp”:
1660778460.931363,
            “inBroadcastPkts”: 0,
            “inOctets”: 95166312767,
            “outDiscards”: 0,
            “outOctets”: 737606103799,
            “inUcastPkts”: 1320302887,
            “outMulticastPkts”: 2536,
            “inDiscards”: 0
        }
    }
}
arista-th3-g07#show priority-flow-control interfaces
Ethernet 18/1 counters | json | no-more
2/1 counters | json | no-more{
    “interfaceCounters”: {
        “Ethernet18/1”: {
            “rxFrames”: 0,
            “txFrames”: 0
        }
    }
}
arista-th3-g07#show interface ethernet 18/5 counters
| json | no-more
{
   “interfaces”: {
        “Ethernet18/5”: {
            “outBroadcastPkts”: 71,
            “outUcastPkts”: 49312609,
            “inMulticastPkts”: 0,
            “lastUpdateTimestamp”:
1660778460.931363,
            “inBroadcastPkts”: 0,
            “inOctets”: 27495833865,
            “outDiscards”: 0,
            “outOctets”: 3548607559,
            “inUcastPkts”: 24952251,
            “outMulticastPkts”: 2536,
            “inDiscards”: 0
        }
    }
}
arista-th3-g07#show priority-flow-control interfaces
Ethernet 18/5 counters | json | no-more
{
    “interfaceCounters”: {
        “Ethernet18/5”: {
            “rxFrames”: 0,
            “txFrames”: 1934
```

Arista_th3_g03 TOR2

```
arista-th3-g03>show interface ethernet 5/1 counters
| json | no-more
{
    “interfaces”: {
        “Ethernet5/1”: {
            “outBroadcastPkts”: 10,
            “outUcastPkts”: 57511180,
            “inMulticastPkts”: 994,
            “lastUpdateTimestamp”:
1660778303.8212898,
            “inBroadcastPkts”: 0,
            “inOctets”: 97621969471,
            “outDiscards”: 0,
            “outOctets”: 59914226499,
            “inUcastPkts”: 71191980,
            “outMulticastPkts”: 1958651,
            “inDiscards”: 0
        }
    }
}

arista-th3-g03>show priority-flow-control interfaces
Ethernet 5/1 counters | json | no-more
{
    “interfaceCounters”: {
        “Ethernet5/1”: {
            “rxFrames”: 2,
            “txFrames”: 224
        }
    }
}
arista-th3-g03>show interface ethernet 6/1 counters
| json | no-more
{
“interfaces”: {
        “Ethernet6/1”: {
            “outBroadcastPkts”: 10,
            “outUcastPkts”: 91728740,
            “inMulticastPkts”: 997,
            “lastUpdateTimestamp”:
1660778303.8212895,
            “inBroadcastPkts”: 0,
            “inOctets”: 62233433035,
            “outDiscards”: 0,
            “outOctets”: 123065936920,
            “inUcastPkts”: 83198176,
            “outMulticastPkts”: 1958649,
           “inDiscards”: 0
        }
    }
}
arista-th3-g03>show priority-flow-control interfaces
Ethernet 6/1 counters | json | no-more
{
    “interfaceCounters”: {
        “Ethernet6/1”: {
            “rxFrames”: 4,
            “txFrames”: 0
        }
    }
```

```
        }
    }
}
arista-th3-g07#show interface ethernet 29/1 counters
| json | no-more
{
    "interfaces": {
        "Ethernet29/1": {
            "outBroadcastPkts": 0,
            "outUcastPkts": 317953396,
            "inMulticastPkts": 149,
            "lastUpdateTimestamp":
1660778460.931363,
            "inBroadcastPkts": 0,
            "inOctets": 177532609156,
            "outDiscards": 0,
            "outOctets": 23100619799,
            "inUcastPkts": 161101421,
            "outMulticastPkts": 149,
            "inDiscards": 0
        }
    }
}
arista-th3-g07#show priority-flow-control interfaces
Ethernet 29/1 counters | json | no-more
{
    "interfaceCounters": {
        "Ethernet29/1": {
            "rxFrames": 0,
            "txFrames": 2829897
        }
    }
}
arista-th3-g07#show interface ethernet 30/1 counters
| json | no-more
{
    "interfaces": {
        "Ethernet30/1": {
            "outBroadcastPkts": 0,
            "outUcastPkts": 302604358,
            "inMulticastPkts": 149,
            "lastUpdateTimestamp":
1660778460.931363,
            "inBroadcastPkts": 0,
            "inOctets": 177535132815,
            "outDiscards": 0,
            "outOctets": 21993000949,
            "inUcastPkts": 161103034,
            "outMulticastPkts": 149,
            "inDiscards": 0
        }
    }
}
arista-th3-g07#show priority-flow-control interfaces
Ethernet 30/1 counters | json | no-more
{
    "interfaceCounters": {
        "Ethernet30/1": {
            "rxFrames": 0,
           "txFrames": 2814146
        }
    }
}
arista-th3-g07#show interface ethernet 31/1 counters
| json | no-more
{
}
arista-th3-g03>show interface ethernet 9/1 counters
| json | no-more
{
    "interfaces": {
        "Ethernet9/1": {
            "outBroadcastPkts": 0,
            "outUcastPkts": 0,
            "inMulticastPkts": 0,
            "lastUpdateTimestamp":
1660778303.8212895,
            "inBroadcastPkts": 0,
            "inOctets": 0,
            "outDiscards": 0,
            "outOctets": 0,
            "inUcastPkts": 0,
            "outMulticastPkts": 0,
            "inDiscards": 0
        }
    }
}
arista-th3-g03>show priority-flow-control interfaces
Ethernet 9/1 counters | json | no-more
{
    "interfaceCounters": {
        "Ethernet9/1": {
            "rxFrames": 0,
            "txFrames": 0
        }
    }
}
arista-th3-g03>show interface ethernet 10/1 counters
| json | no-more
{
    "interfaces": {
        "Ethernet10/1": {
            "outBroadcastPkts": 0,
            "outUcastPkts": 0,
            "inMulticastPkts": 0,
            "lastUpdateTimestamp":
1660778303.8212895,
            "inBroadcastPkts": 0,
            "inOctets": 0,
            "outDiscards": 0,
            "outOctets": 0,
            "inUcastPkts": 0,
            "outMulticastPkts": 0,
            "inDiscards": 0
```

```
“interfaces”: {
        “Ethernet31/1”: {
            “outBroadcastPkts”: 0,
            “outUcastPkts”: 338847296,
            “inMulticastPkts”: 149,
            “lastUpdateTimestamp”:
1660778460.9313638,
            “inBroadcastPkts”: 0,
            “inOctets”: 177512971933,
            “outDiscards”: 0,
            “outOctets”: 24606791115,
            “inUcastPkts”: 161082914,
            “outMulticastPkts”: 149,
            “inDiscards”: 0
        }
    }
}
arista-th3-g07#show priority-flow-control interfaces
Ethernet 31/1 counters | json | no-more
{
    “interfaceCounters”: {
        “Ethernet31/1”: {
            “rxFrames”: 0,
            “txFrames”: 2832609
        }
    }
}
arista-th3-g07#show interface ethernet 32/1 counters
| json | no-more
{
    “interfaces”: {
        “Ethernet32/1”: {
            “outBroadcastPkts”: 0,
            “outUcastPkts”: 311585228,
            “inMulticastPkts”: 149,
            “lastUpdateTimestamp”:
1660778460.9313633,
            “inBroadcastPkts”: 0,
            “inOctets”: 177529325177,
            “outDiscards”: 0,
            “outOctets”: 22642480937,
            “inUcastPkts”: 161097792,
            “outMulticastPkts”: 149,
            “inDiscards”: 0
        }
    }
}
arista-th3-g07#show priority-flow-control interfaces
Ethernet 32/1 counters | json | no-more
{
    “interfaceCounters”: {
        “Ethernet32/1”: {
            “rxFrames”: 0,
            “txFrames”: 2844861
        }
    }
}
```

```
        }
    }
}
arista-th3-g03>show priority-flow-control interfaces
Ethernet 10/1 counters | json | no-more
{
    “interfaceCounters”: {
        “Ethernet10/1”: {
            “rxFrames”: 0,
            “txFrames”: 0
        }
    }
}
```

Spine (Arista 7280R3) Counters:

Arista_dnx_g06 – Spine1

```
arista-dnx-g06#show interface ethernet 5/1 counters
| json | no-more
{
    “interfaces”: {
        “Ethernet5/1”: {
            “outBroadcastPkts”: 0,
            “outUcastPkts”: 329914002,
            “inMulticastPkts”: 154,
            “lastUpdateTimestamp”:
1660778353.1140516,
            “inBroadcastPkts”: 0,
            “inOctets”: 178002539491,
            “outDiscards”: 0,
            “outOctets”: 23848769051,
            “inUcastPkts”: 161527895,
            “outMulticastPkts”: 154,
            “inDiscards”: 0
        }
    }
}
arista-dnx-g06#show priority-flow-control interfaces
Ethernet 5/1 counters | json | no-more
{
    “interfaceCounters”: {
        “Ethernet5/1”: {
            “rxFrames”: 0,
            “txFrames”: 1050096
        }
    }
}
arista-dnx-g06#show interface ethernet 6/1 counters
| json | no-more
{
    “interfaces”: {
        “Ethernet6/1”: {
            “outBroadcastPkts”: 0,
            “outUcastPkts”: 290643752,
            “inMulticastPkts”: 154,
           “lastUpdateTimestamp”:
1660778353.1140528,
            “inBroadcastPkts”: 0,
            “inOctets”: 177065217505,
            “outDiscards”: 0,
            “outOctets”: 21015128877,
            “inUcastPkts”: 160676560,
            “outMulticastPkts”: 154,
            “inDiscards”: 0
        }
    }
}
arista-dnx-g06#show priority-flow-control interfaces
Ethernet 6/1 counters | json | no-more
{
“interfaceCounters”: {
        “Ethernet6/1”: {
            “rxFrames”: 0,
            “txFrames”: 1004698
        }
    }
}
arista-dnx-g06#show interface ethernet 9/1 counters
| json | no-more
```

Arista_dnx_g05 – Spine2

```
arista-dnx-g05#show interface ethernet 5/1 counters
| json | no-more
{
    “interfaces”: {
        “Ethernet5/1”: {
            “outBroadcastPkts”: 0,
            “outUcastPkts”: 328639662,
            “inMulticastPkts”: 151,
            “lastUpdateTimestamp”:
1660778293.277825,
            “inBroadcastPkts”: 0,
            “inOctets”: 177477783028,
            “outDiscards”: 0,
            “outOctets”: 23756835738,
            “inUcastPkts”: 161051403,
            “outMulticastPkts”: 151,
            “inDiscards”: 0
        }
    }
}
arista-dnx-g05#show priority-flow-control interfaces
Ethernet 5/1 counters | json | no-more
{
    “interfaceCounters”: {
        “Ethernet5/1”: {
            “rxFrames”: 0,
            “txFrames”: 1049634
        }
    }
}
arista-dnx-g05#show interface ethernet 6/1 counters
| json | no-more
{
    “interfaces”: {
        “Ethernet6/1”: {
            “outBroadcastPkts”: 0,
            “outUcastPkts”: 321792862,
            “inMulticastPkts”: 151,
           “lastUpdateTimestamp”:
1660778293.2778263,
            “inBroadcastPkts”: 0,
            “inOctets”: 177564527542,
            “outDiscards”: 0,
            “outOctets”: 23263396560,
            “inUcastPkts”: 161129303,
            “outMulticastPkts”: 151,
            “inDiscards”: 0
        }
    }
}
arista-dnx-g05#show priority-flow-control interfaces
Ethernet 6/1 counters | json | no-more
{
    “interfaceCounters”: {
        “Ethernet6/1”: {
            “rxFrames”: 0,
            “txFrames”: 1049278
        }
    }
}
arista-dnx-g05#show interface ethernet 23/1 counters
| json | no-more
```

```
{
    “interfaces”: {
        “Ethernet9/1”: {
            “outBroadcastPkts”: 0,
            “outUcastPkts”: 161101421,
            “inMulticastPkts”: 155,
            “lastUpdateTimestamp”:
1660778353.1140516,
            “inBroadcastPkts”: 0,
            “inOctets”: 23100621215,
            “outDiscards”: 0,
            “outOctets”: 177532610121,
            “inUcastPkts”: 317953396,
            “outMulticastPkts”: 154,
            “inDiscards”: 0
        }
    }
}
arista-dnx-g06#show priority-flow-control interfaces
Ethernet 9/1 counters | json | no-more
{
    “interfaceCounters”: {
        “Ethernet9/1”: {
            “rxFrames”: 2829897,
            “txFrames”: 0
        }
    }
}
arista-dnx-g06#show interface ethernet 10/1 counters
| json | no-more
{
    “interfaces”: {
        “Ethernet10/1”: {
            “outBroadcastPkts”: 0,
            “outUcastPkts”: 161103034,
            “inMulticastPkts”: 155,
            “lastUpdateTimestamp”:
1660778353.1140523,
            “inBroadcastPkts”: 0,
            “inOctets”: 21993002365,
            “outDiscards”: 0,
            “outOctets”: 177535133785,
            “inUcastPkts”: 302604358,
            “outMulticastPkts”: 154,
            “inDiscards”: 0
        }
    }
}
arista-dnx-g06#show priority-flow-control interfaces
Ethernet 10/1 counters | json | no-more
{
    “interfaceCounters”: {
        “Ethernet10/1”: {
            “rxFrames”: 2814146,
            “txFrames”: 0
        }
    }
}
```

```
{
    “interfaces”: {
        “Ethernet23/1”: {
            “outBroadcastPkts”: 0,
            “outUcastPkts”: 161082914,
            “inMulticastPkts”: 151,
            “lastUpdateTimestamp”:
1660778293.2778249,
            “inBroadcastPkts”: 0,
            “inOctets”: 24606791587,
            “outDiscards”: 0,
            “outOctets”: 177512972321,
            “inUcastPkts”: 338847296,
            “outMulticastPkts”: 151,
            “inDiscards”: 0
        }
    }
}
arista-dnx-g05#show priority-flow-control interfaces
Ethernet 23/1 counters | json | no-more
{
    “interfaceCounters”: {
        “Ethernet23/1”: {
            “rxFrames”: 2832609,
            “txFrames”: 0
        }
    }
}
arista-dnx-g05#show interface ethernet 24/1 counters
| json | no-more
{
    “interfaces”: {
        “Ethernet24/1”: {
            “outBroadcastPkts”: 0,
            “outUcastPkts”: 161097792,
            “inMulticastPkts”: 151,
            “lastUpdateTimestamp”:
1660778293.2778265,
            “inBroadcastPkts”: 0,
            “inOctets”: 22642481409,
            “outDiscards”: 0,
            “outOctets”: 177529325565,
            “inUcastPkts”: 311585228,
            “outMulticastPkts”: 151,
            “inDiscards”: 0
        }
    }
}
arista-dnx-g05#show priority-flow-control interfaces
Ethernet 24/1 counters | json | no-more
{
    “interfaceCounters”: {
        “Ethernet24/1”: {
            “rxFrames”: 2844861,
            “txFrames”: 0
        }
    }
}
```

Client 1 with Broadcom THOR NIC:

```
# ethtool -S ens7f0np0
NIC statistics:
     [2]: rx_ucast_packets: 475
     [2]: rx_ucast_bytes: 68575
     [3]: tx_ucast_packets: 1039
     [3]: tx_ucast_bytes: 143036
     [4]: rx_ucast_packets: 545
     [4]: rx_ucast_bytes: 73195
     [4]: tx_ucast_packets: 567
     [4]: tx_ucast_bytes: 74653
     [5]: rx_ucast_packets: 506
     [5]: rx_ucast_bytes: 70621
     rx_64b_frames: 1937
     rx_65b_127b_frames: 49311606
     rx_128b_255b_frames: 1034
     rx_total_frames: 49314577
     rx_ucast_frames: 49312608
     rx_mcast_frames: 35
     rx_ctrl_frames: 1934
     rx_pfc_frames: 1934
     rx_good_frames: 49312643
     rx_pfc_ena_frames_pri3: 1934
     rx_bytes: 3548265698
     tx_65b_127b_frames: 574
     tx_128b_255b_frames: 1032
     tx_1024b_1518b_frames: 24950644
     tx_good_frames: 24952250
     tx_total_frames: 24952250
     tx_ucast_frames: 24952250
     tx_bytes: 27495833801
     rx_bytes_cos0: 1541451750
     rx_packets_cos0: 24862125
     rx_bytes_cos4: 216782
     rx_packets_cos4: 1561
     rx_bytes_cos5: 1809222818
     rx_packets_cos5: 24448957
     pfc_pri3_rx_transitions: 967
     tx_bytes_cos0: 27296004536
     tx_packets_cos0: 24950644
    tx_bytes_cos4: 217689
     tx_packets_cos4: 1606
     rx_bytes_pri0: 216782
     rx_bytes_pri1: 216782
     rx_bytes_pri2: 216782
     rx_bytes_pri3: 1541451750
     rx_bytes_pri4: 216782
     rx_bytes_pri5: 216782
     rx_bytes_pri6: 216782
     rx_bytes_pri7: 1809222818
     rx_packets_pri0: 1561
     rx_packets_pri1: 1561
     rx_packets_pri2: 1561
     rx_packets_pri3: 24862125
     rx_packets_pri4: 1561
     rx_packets_pri5: 1561
     rx_packets_pri6: 1561
     rx_packets_pri7: 24448957
     tx_bytes_pri0: 217689
     tx_bytes_pri1: 217689
     tx_bytes_pri2: 217689
     tx_bytes_pri3: 27296004536
    tx_bytes_pri4: 217689
```

```
root@hpca4010:~# cat /sys/kernel/debug/bnxt_re/bnxt_
re0/info
bnxt_re debug info:
=====[ IBDEV bnxt_re0 ]============================
        link state: UP
        Max QP:          65537
        Max SRQ:         4096
        Max CQ:          65536
        Max MR:          262144
        Max MW:          262144
        Max AH:          65536
        Max PD:          65536
        Active QP:       1
        Active SRQ:      0
        Active CQ:       1
        Active MR:       1
        Active MW:       1
        Active AH:       0
        Active PD:       1
        QP Watermark:    3070
        SRQ Watermark:   33
        CQ Watermark:    131
        MR Watermark:    2950
        MW Watermark:    1
        AH Watermark:    284
        PD Watermark:    34
        Rx Pkts: 49311082
        Rx Bytes: 1541451750
        Tx Pkts: 24950644
        Tx Bytes: 27296004536
        CNP Rx Pkts: 24448957
        RoCE Only Rx Pkts: 24862125
        RoCE Only Rx Bytes: 1541451750
        RoCE Only Tx Pkts: 24950644
        RoCE Only Tx Bytes: 27296004536
        tx_write_req: 24950644
        rx_good_pkts: 24862125
        rx_good_bytes: 1541451750
        dbq_int_en: 30
        dbq_pacing_complete: 29
        latency_slab [0 - 1] sec = 139902
```

```
tx_bytes_pri5: 217689
     tx_bytes_pri6: 217689
     tx_packets_pri0: 1606
     tx_packets_pri1: 1606
     tx_packets_pri2: 1606
     tx_packets_pri3: 24950644
     tx_packets_pri4: 1606
     tx_packets_pri5: 1606
     tx_packets_pri6: 1606
```

Client 2 with Broadcom THOR NIC:

```
# ethtool -S ens7f0np0
NIC statistics:
     [1]: tx_ucast_packets: 411
     [1]: tx_ucast_bytes: 64357
     [4]: rx_ucast_packets: 483
     [4]: rx_ucast_bytes: 69103
     [4]: tx_ucast_packets: 378
     [4]: tx_ucast_bytes: 62179
     [5]: rx_ucast_packets: 471
     [5]: rx_ucast_bytes: 68311
     [6]: rx_ucast_packets: 597
     [6]: rx_ucast_bytes: 76627
     [7]: tx_ucast_packets: 451
     [7]: tx_ucast_bytes: 66997
     rx_64b_frames: 390574
     rx_65b_127b_frames: 643627388
     rx_128b_255b_frames: 1034
     rx_total_frames: 644018996
     rx_ucast_frames: 643628391
     rx_mcast_frames: 34
     rx_ctrl_frames: 390571
     rx_pfc_frames: 390571
     rx_good_frames: 643628425
     rx_pfc_ena_frames_pri3: 390571
     rx_bytes: 46418582019
     tx_65b_127b_frames: 208
     tx_128b_255b_frames: 1032
     tx_1024b_1518b_frames: 326166544
    tx_good_frames: 326167784
     tx_total_frames: 326167784
     tx_ucast_frames: 326167784
     tx_bytes: 359435729981
     rx_bytes_cos0: 19682585596
     rx_packets_cos0: 317461058
     rx_bytes_cos4: 218192
     rx_packets_cos4: 1584
     rx_bytes_cos5: 24136267868
     rx_packets_cos5: 326165782
     tx_bytes_cos0: 356826199136
     tx_packets_cos0: 326166544
     tx_bytes_cos4: 193533
     tx_packets_cos4: 1240
     rx_bytes_pri0: 218192
     rx_bytes_pri1: 218192
     rx_bytes_pri2: 218192
     rx_bytes_pri3: 19682585596
     rx_bytes_pri4: 218192
     rx_bytes_pri5: 218192
     rx_bytes_pri6: 218192
     rx_bytes_pri7: 24136267868
    rx_packets_pri0: 1584
     rx_packets_pri1: 1584
```

```
root@hpci5001:~# cat /sys/kernel/debug/bnxt_re/bnxt_
re0/info
bnxt_re debug info:
=====[ IBDEV bnxt_re0 ]============================
        link state: UP
        Max QP:          65537
        Max SRQ:         4096
        Max CQ:          65536
        Max MR:          262144
        Max MW:          262144
        Max AH:          65536
        Max PD:          65536
        Active QP:       1
        Active SRQ:      0
        Active CQ:       1
        Active MR:       1
        Active MW:       1
        Active AH:       0
        Active PD:       1
        QP Watermark:    3070
        SRQ Watermark:   33
        CQ Watermark:    131
        MR Watermark:    2950
        MW Watermark:    1
        AH Watermark:    284
        PD Watermark:    34
        Rx Pkts: 643626840
        Rx Bytes: 19682585596
        Tx Pkts: 326166544
        Tx Bytes: 356826199136
        CNP Rx Pkts: 326165782
        RoCE Only Rx Pkts: 317461058
        RoCE Only Rx Bytes: 19682585596
        RoCE Only Tx Pkts: 326166544
        RoCE Only Tx Bytes: 356826199136
        tx_write_req: 326166544
        rx_good_pkts: 317461058
        rx_good_bytes: 19682585596
       dbq_pacing_complete: 29
        latency_slab [0 - 1] sec = 139902
```

```
rx_packets_pri2: 1584
     rx_packets_pri3: 317461058
     rx_packets_pri4: 1584
     rx_packets_pri5: 1584
     rx_packets_pri6: 1584
     rx_packets_pri7: 326165782
     tx_bytes_pri0: 193533
     tx_bytes_pri1: 193533
     tx_bytes_pri2: 193533
     tx_bytes_pri3: 356826199136
     tx_bytes_pri4: 193533
     tx_bytes_pri5: 193533
     tx_bytes_pri6: 193533
     tx_packets_pri0: 1240
     tx_packets_pri1: 1240
     tx_packets_pri2: 1240
     tx_packets_pri3: 326166544
     tx_packets_pri4: 1240
     tx_packets_pri5: 1240
     tx_packets_pri6: 1240
```

Client 3 with Broadcom THOR NIC:

```
# ethtool -s ens7f0np0
NIC statistics:
     [0]: rx_ucast_packets: 483
     [0]: rx_ucast_bytes: 69097
     [0]: tx_ucast_packets: 958
     [0]: tx_ucast_bytes: 137690
     [2]: tx_ucast_packets: 461
     [2]: tx_ucast_bytes: 67657
     [3]: rx_ucast_packets: 702
     [3]: rx_ucast_bytes: 83557
     [3]: tx_ucast_packets: 1
     [3]: tx_ucast_bytes: 54
     [6]: tx_ucast_packets: 1
     [6]: tx_ucast_bytes: 54
     [7]: rx_ucast_packets: 475
     [7]: rx_ucast_bytes: 68569
     rx_64b_frames: 454212
     rx_65b_127b_frames: 627360881
     rx_128b_255b_frames: 1034
     rx_total_frames: 627816127
     rx_ucast_frames: 627361886
     rx_mcast_frames: 34
     rx_ctrl_frames: 454207
     rx_pfc_frames: 454207
     rx_good_frames: 627361920
     rx_pfc_ena_frames_pri3: 454207
     rx_bytes: 45253667833
     tx_64b_frames: 2
     tx_65b_127b_frames: 387
     tx_128b_255b_frames: 1032
     tx_1024b_1518b_frames: 318215955
     tx_good_frames: 318217376
     tx_total_frames: 318217376
     tx_ucast_frames: 318217376
     tx_bytes: 350674193561
     rx_bytes_cos0: 19166945112
     rx_packets_cos0: 309144276
     rx_bytes_cos4: 225493
     rx_packets_cos4: 1694
    rx_bytes_cos5: 23547980300
     rx_packets_cos5: 318215950
pfc_pri3_rx_transitions: 224678
     tx_bytes_cos0: 348128254770
     tx_packets_cos0: 318215955
     tx_bytes_cos4: 205455
```

```
root@hpca4009:~# cat /sys/kernel/debug/bnxt_re/bnxt_
re0/info
bnxt_re debug info:
=====[ IBDEV bnxt_re0 ]============================
        link state: UP
        Max QP:          65537
        Max SRQ:         4096
        Max CQ:          65536
        Max MR:          262144
        Max MW:          262144
        Max AH:          65536
        Max PD:          65536
        Active QP:       1
        Active SRQ:      0
        Active CQ:       1
        Active MR:       1
        Active MW:       1
        Active AH:       0
        Active PD:       1
        QP Watermark:    3070
        SRQ Watermark:   33
        CQ Watermark:    131
        MR Watermark:    2950
        MW Watermark:    1
        AH Watermark:    284
        PD Watermark:    34
        Rx Pkts: 627360226
        Rx Bytes: 19166945112
        Tx Pkts: 318215955
        Tx Bytes: 348128254770
        CNP Rx Pkts: 318215950
        RoCE Only Rx Pkts: 309144276
        RoCE Only Rx Bytes: 19166945112
        RoCE Only Tx Pkts: 318215955
        RoCE Only Tx Bytes: 348128254770
        tx_write_req: 318215955
        rx_good_pkts: 309144276
        rx_good_bytes: 19166945112
        fw_service_prof_type_sup : 1
        dbq_int_en: 30
        dbq_pacing_complete: 29
        latency_slab [0 - 1] sec = 139902
```

```
    tx_packets_cos4: 1421
    rx_bytes_pri0: 225493
    rx_bytes_pri1: 225493
    rx_bytes_pri2: 225493
    rx_bytes_pri3: 19166945112
    rx_bytes_pri4: 225493
    rx_bytes_pri5: 225493
    rx_bytes_pri6: 225493
    rx_bytes_pri7: 23547980300
    rx_packets_pri0: 1694
    rx_packets_pri1: 1694
    rx_packets_pri2: 1694
    rx_packets_pri3: 309144276
    rx_packets_pri4: 1694
    rx_packets_pri5: 1694
    rx_packets_pri6: 1694
    rx_packets_pri7: 318215950
    tx_bytes_pri0: 205455
    tx_bytes_pri1: 205455
    tx_bytes_pri2: 205455
    tx_bytes_pri3: 348128254770
    tx_bytes_pri4: 205455
    tx_bytes_pri5: 205455
    tx_bytes_pri6: 205455
    tx_packets_pri0: 1421
    tx_packets_pri1: 1421
    tx_packets_pri2: 1421
    tx_packets_pri3: 318215955
    tx_packets_pri4: 1421
    tx_packets_pri5: 1421
    tx_packets_pri6: 1421
```