NetXtreme®-C/NetXtreme-E

User Guide

NetXtreme-UG101
April 2, 2019

Table of Contents

1 Regulatory and Safety Approvals 7
1.1 Regulatory 7
1.2 Safety 7
1.3 Electromagnetic Compatibility (EMC) 7
1.4 Electrostatic Discharge (ESD) Compliance 8
1.5 FCC Statement 8
2 Functional Description 8
3 Network Link and Activity Indication 8
4 Features 9
4.1 Software and Hardware Features 9
4.2 Virtualization Features 10
4.3 VXLAN 11
4.4 NVGRE/GRE/IP-in-IP/Geneve 11
4.5 Stateless Offloads 11
4.5.1 RSS 11
4.5.2 TPA 11
4.5.3 Header-Payload Split 11
4.6 UDP Fragmentation Offload 11
4.7 Stateless Transport Tunnel Offload 12
4.8 Multiqueue Support for OS 12
4.8.1 NDIS VMQ 12
4.8.2 VMware NetQueue 12
4.8.3 KVM/Xen Multiqueue 12
4.9 SR-IOV Configuration Support Matrix 12
4.10 SR-IOV 12
4.11 Network Partitioning (NPAR) 13
4.12 RDMA over Converge Ethernet – RoCE 13
4.13 Supported Combinations 13
4.13.1 NPAR, SR-IOV, and RoCE 13
4.13.2 NPAR, SR-IOV, and DPDK 14
4.13.3 Unsupported Combinations 14
5 Installing the Hardware 15
5.1 Safety Precautions 15
5.2 System Requirements 15
5.2.1 Hardware Requirements 15
5.2.2 Preinstallation Checklist 15
5.3 Installing the Adapter 16
5.4 Connecting the Network Cables 16

5.4.1 Supported Cables and Modules 16
5.4.1.1 Copper 16
5.4.1.2 SFP+ 16
5.4.1.3 SFP28 16
5.4.1.4 QSFP 16
6 Software Packages and Installation 17
6.1 Supported Operating Systems 17
6.2 Installing the Linux Driver 17
6.2.1 Linux Ethtool Commands 17
6.3 Installing the VMware Driver 18
6.4 Installing the Windows Driver 19
6.4.1 Driver Advanced Properties 19
6.4.2 Event Log Messages 20
7 Updating the Firmware 21
7.1 Linux 21
7.2 Windows/ESX 21
8 Teaming 22
8.1 Windows 22
8.2 Linux 22
9 System-Level Configuration 23
9.1 UEFI HII Menu 23
9.1.1 Main Configuration Page 23
9.1.2 Firmware Image Properties 23
9.1.3 Device-Level Configuration 23
9.1.4 NIC Configuration 23
9.1.5 iSCSI Configuration 23
9.2 Comprehensive Configuration Management 24
9.2.1 Device Hardware Configuration 24
9.2.2 MBA Configuration Menu 24
9.2.3 iSCSI Boot Main Menu 24
9.3 Auto-Negotiation Configuration 24
9.3.1 Operational Link Speed 27
9.3.2 Firmware Link Speed 27
9.3.3 Auto-negotiation Protocol 27
9.3.4 Windows Driver Settings 27
9.3.5 Linux Driver Settings 28
9.3.6 ESXi Driver Settings 28
10 iSCSI Boot 28
10.1 Supported Operating Systems for iSCSI Boot 28
10.2 Setting up iSCSI Boot 29
10.2.1 Configuring the iSCSI Target 29

10.2.2 Configuring iSCSI Boot Parameters 29
10.2.3 MBA Boot Protocol Configuration 30
10.2.4 iSCSI Boot Configuration 30
10.2.4.1 Static iSCSI Boot Configuration 30
10.2.4.2 Dynamic iSCSI Boot Configuration 31
10.2.5 Enabling CHAP Authentication 32
10.3 Configuring the DHCP Server to Support iSCSI Boot 33
10.3.1 DHCP iSCSI Boot Configurations for IPv4 33
10.3.1.1 DHCP Option 17, Root Path 33
10.3.1.2 DHCP Option 43, Vendor-Specific Information 34
10.3.1.3 Configuring the DHCP Server 34
10.3.2 DHCP iSCSI Boot Configuration for IPv6 34
10.3.2.1 DHCPv6 Option 16, Vendor Class Option 34
10.3.2.2 DHCPv6 Option 17, Vendor-Specific Information 34
10.3.2.3 Configuring the DHCP Server 35
11 VXLAN – Configuration and Use Case Examples 35
12 SR-IOV – Configuration and Use Case Examples 36
12.1 Linux Use Case Example 36
12.2 Windows Use Case Example 37
12.3 VMware SRIOV Use Case Example 38
13 NPAR – Configuration and Use Case Example 39
13.1 Features and Requirements 39
13.2 Limitations 40
13.3 Configuration 40
13.4 Notes on Reducing NIC Memory Consumption 42
14 RoCE – Configuration and Use Case Examples 43
14.1 Linux Configuration and Use Case Examples 43
14.1.1 Requirements 43
14.1.2 BNXT_RE Driver Dependencies 43
14.1.3 Installation 44
14.1.4 Limitations 45
14.1.5 Known Issues 45
14.2 Windows and Use Case Examples 45
14.2.1 Kernel Mode 45
14.2.2 Verifying RDMA 45
14.2.3 User Mode 46
14.3 VMware ESX Configuration and Use Case Examples 47
14.3.1 Limitations 47
14.3.2 BNXT RoCE Driver Requirements 47
14.3.3 Installation 47
14.3.4 Configuring Paravirtualized RDMA Network Adapters 47
14.3.4.1 Configuring a Virtual Center for PVRDMA 47

14.3.4.2 Tagging vmknic for PVRDMA on ESX Hosts 48
14.3.4.3 Setting the Firewall Rule for PVRDMA 48
14.3.4.4 Adding a PVRDMA Device to the VM 48
14.3.4.5 Configuring the VM on Linux Guest OS 48
15 DCBX – Data Center Bridging 50
15.1 QoS Profile – Default QoS Queue Profile 50
15.2 DCBX Mode – Enable (IEEE only) 51
15.3 DCBX Willing Bit 51
16 DPDK – Configuration and Use Case Examples 54
16.1 Compiling the Application 54
16.2 Running the Application 54
16.3 Testpmd Runtime Functions 55
16.4 Control Functions 55
16.5 Display Functions 55
16.6 Configuration Functions 56
17 Frequently Asked Questions 56
Revision History **57**

1 Regulatory and Safety Approvals

The following sections detail the Regulatory, Safety, Electromagnetic Compatibility (EMC), and Electrostatic Discharge (ESD) standard compliance for the NetXtreme®-C/NetXtreme-E Network Interface Card.

1.1 Regulatory

Table 1: Regulatory Approvals

Item	Applicable Standard	Approval/Certificate
CE/European Union	EN 60950-1	CB report and certificate
UL/USA	UL 60950-1 CTUVus UL	UL report and certificate.
CSA/Canada	CSA 22.2 No. 950	CSA report and certificate.
Taiwan	CNS14336 Class B	–

1.2 Safety

Table 2: Safety Approvals

Country	Certification Type/Standard	Compliance
International	CB Scheme ICES 003 - Digital Device UL 1977 (connector safety) UL 796 (PCB wiring safety) UL 94 (flammability of parts)	Yes

1.3 Electromagnetic Compatibility (EMC)

Table 3: Electromagnetic Compatibility

Standard/Country	Certification Type	Compliance
CE/EU	EN 55022:2010 + *AC:2011 Class B EN 55024 Class B	CE report and CE DoC
FCC/USA	CFR47, Part 15 Class B	FCC/IC DoC and EMC report referencing FCC and IC standards
IC/Canada	ICES-003 Class B	FCC/IC DoC and report referencing FCC and IC standards
ACA/Australia, New Zealand	EN 5022:2010 + *AC:2011	ACA certificate RCM Mark
BSMI/Taiwan	CNS13438 Class B	BSMI certificate
MIC/S. Korea	RRL KN22 Class B KN24 (ESD)	Korea certificate MSIP Mark
VCCI /Japan	V-3/2014/04	Copy of VCCI on-line certificate

1.4 Electrostatic Discharge (ESD) Compliance

Table 4: ESD Compliance Summary

Standard	Certification Type	Compliance
EN55024:2010	Air/Direct discharge	Yes

1.5 FCC Statement

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Consult the dealer or an experienced radio/TV technician for help.

NOTE: Changes or modifications not expressly approved by the manufacture responsible for compliance could void the user's authority to operate the equipment.

2 Functional Description

The Broadcom NetXtreme-C (BCM573XX) and NetXtreme-E (BCM574XX) family of Ethernet Controllers are highly-integrated, full-featured Ethernet LAN controllers optimized for data center and cloud infrastructures. Adapters support 100G/50G/40G/25G/10G/1G in both single and dual-port configurations. On the host side, these devices support sixteen lanes of a PCIe Generation 3 interface.

An extensive set of stateless offloads and virtualization offloads to enhance packet processing efficiency are included to enable low-overhead, high-speed network communications.

3 Network Link and Activity Indication

Ethernet connections, the state of the network link, and activity is indicated by the LEDs on the rear connector as shown in Table 5.

Refer to the individual board data sheets for specific media design.

Table 5: Network Link and Activity Indicated by Port LEDs

Port LED	LED Appearance	Network State
Link LED	Off	No link (cable disconnected)
	Continuously illuminated	Link
Activity LED	Off	No network activity
	Blinking	Network activity

4 Features

Refer to the following sections for device features.

4.1 Software and Hardware Features

Table 6 provides a list of host interface features.

Table 6: Host Interface Features

Feature	Details
Host Interface	PCIe 3.0 (Gen 3: 8 GT/s; Gen 2: 5 GT/s; Gen 1: 2.5 GT/s).
Number of PCIe lanes	PCIe Edge connector: x16.
Vital Product Data (VPD)	Supported.
Alternate Routing ID (ARI)	Supported.
Function Level Reset (FLR)	Supported.
Advanced Error Reporting	Supported.
PCIe ECNs	Support for TLP Processing Hints (TPH), Latency Tolerance Reporting (LTR), and Optimized Buffer Flush/Fill (OBFF).
MSI-X Interrupt vector per queue	1 per RSS queue, 1 per NetQueue, 1 per Virtual Machine Queue (VMQ).
IP Checksum Offload	Support for transmit and receive side.
TCP Checksum Offload	Support for transmit and receive side.
UDP Checksum Offload	Support for transmit and receive side.
NDIS TCP Large Send Offload	Support for LSOV1 and LSOV2.
NDIS Receive Segment Coalescing (RSC)	Support for Windows environments.
TCP Segmentation Offload (TSO)	Support for Linux and VMware environments.
Large Receive Offload (LRO)	Support for Linux and VMware environments.
Generic Receive Offload (GRO)	Support for Linux and VMware environments.
Receive Side Scaling (RSS)	Support for Windows, Linux, and VMware environments. Up to 8 queues/port supported for RSS.
Header-Payload Split	Enables the software TCP/IP stack to receive TCP/IP packets with header and payload data split into separate buffers. Supports Windows, Linux, and VMware environments.
Jumbo Frames	Supported.
iSCSI boot	Supported.
NIC Partitioning (NPAR)	Supports up to eight Physical Functions (PFs) per port, or up to 16 PFs per silicon. This option is configurable in NVRAM.

Table 6: Host Interface Features (Continued)

Feature	Details
RDMA over Converge Ethernet (RoCE)	The BCM5741X supports RoCE v1/v2 for Windows, Linux, and VMware.
Data Center Bridging (DCB)	The BCM5741X supports DCBX (IEEE and CEE specification), PFC, and AVB.
NCSI (Network Controller Sideband Interface)	Supported.
Wake on LAN (WOL)	Supported on designs with 10GBASE-T, SFP+, and SFP28 interfaces.
PXE boot	Supported.
UEFI boot	Supported.
Flow Control (Pause)	Supported.
Auto negotiation	Supported.
IEEE 802.1q VLAN	Supported.
Interrupt Moderation	Supported.
MAC/VLAN filters	Supported.

4.2 Virtualization Features

Table 7 lists the virtualization features of the NetXtreme-C/NetXtreme-E.

Table 7: Virtualization Features

Feature	Details
Linux KVM Multiqueue	Supported.
VMware NetQueue	Supported.
NDIS Virtual Machine Queue (VMQ)	Supported.
Virtual eXtensible LAN (VXLAN) – Aware stateless offloads (IP/UDP/TCP checksum offloads	Supported.
Generic Routing Encapsulation (GRE) – Aware stateless offloads (IP/UDP/TCP checksum offloads	Supported.
Network Virtualization using Generic Routing Encapsulation (NVGRE) – Aware stateless offloads	Supported.
IP-in-IP aware stateless offloads (IP/UDP/TCP checksum offloads	Supported
SR-IOV v1.0	128 Virtual Functions (VFs) for Guest Operating Systems (GOS) per device. MSI-X vector per VF is set to 16.

Table 7: Virtualization Features (Continued)

Feature	Details
MSI-X vector port	74 per port default value (two port configuration). 16 per VF and is configurable in HII and CCM.

4.3 VXLAN

A Virtual eXtensible Local Area Network (VXLAN), defined in IETF RFC 7348, is used to address the need for overlay networks within virtualized data centers accommodating multiple tenants. VXLAN is a Layer 2 overlay or tunneling scheme over a Layer 3 network. Only VMs within the same VXLAN segment can communicate with each other.

4.4 NVGRE/GRE/IP-in-IP/Geneve

Network Virtualization using GRE (NVGRE), defined in IETF RFC 7637, is similar to a VXLAN.

4.5 Stateless Offloads

4.5.1 RSS

Receive Side Scaling (RSS) uses a Toeplitz algorithm which uses 4 tuple match on the received frames and forwards it to a deterministic CPU for frame processing. This allows streamlined frame processing and balances CPU utilization. An indirection table is used to map the stream to a CPU.

Symmetric RSS allows the mapping of packets of a given TCP or UDP flow to the same receive queue.

4.5.2 TPA

Transparent Packet Aggregation (TPA) is a technique where received frames of the same 4 tuple matched frames are aggregated together and then indicated to the network stack. Each entry in the TPA context is identified by the 4 tuple: Source IP, destination IP, source TCP port, and destination TCP port. TPA improves system performance by reducing interrupts for network traffic and lessening CPU overhead.

4.5.3 Header-Payload Split

Header-payload split is a feature that enables the software TCP/IP stack to receive TCP/IP packets with header and payload data split into separate buffers. The support for this feature is available in both Windows and Linux environments. The following are potential benefits of header-payload split:

- The header-payload split enables compact and efficient caching of packet headers into host CPU caches. This can result in a receive side TCP/IP performance improvement.
- Header-payload splitting enables page flipping and zero copy operations by the host TCP/IP stack. This can further improve the performance of the receive path.

4.6 UDP Fragmentation Offload

UDP Fragmentation Offload (UFO) is a feature that enables the software stack to offload fragmentation of UDP/IP datagrams into UDP/IP packets. The support for this feature is only available in the Linux environment. The following is a potential benefit of UFO:

- The UFO enables the NIC to handle fragmentation of a UDP datagram into UDP/IP packets. This can result in the reduction of CPU overhead for transmit side UDP/IP processing.

4.7 Stateless Transport Tunnel Offload

Stateless Transport Tunnel Offload (STT) is a tunnel encapsulation that enables overlay networks in virtualized data centers. STT uses IP-based encapsulation with a TCP-like header. There is no TCP connection state associated with the tunnel and that is why STT is stateless. Open Virtual Switch (OVS) uses STT.

An STT frame contains the STT frame header and payload. The payload of the STT frame is an untagged Ethernet frame. The STT frame header and encapsulated payload are treated as the TCP payload and TCP-like header. The IP header (IPv4 or IPv6) and Ethernet header are created for each STT segment that is transmitted.

4.8 Multiqueue Support for OS

4.8.1 NDIS VMQ

The NDIS Virtual Machine Queue (VMQ) is a feature that is supported by Microsoft to improve Hyper-V network performance. The VMQ feature supports packet classification based on the destination MAC address to return received packets on different completion queues. This packet classification combined with the ability to DMA packets directly into a virtual machine's memory allows the scaling of virtual machines across multiple processors.

See Driver Advanced Properties for information on VMQ.

4.8.2 VMware NetQueue

The VMware NetQueue is a feature that is similar to Microsoft's NDIS VMQ feature. The NetQueue feature supports packet classification based on the destination MAC address and VLAN to return received packets on different NetQueues. This packet classification combined with the ability to DMA packets directly into a virtual machine's memory allows the scaling of virtual machines across multiple processors.

4.8.3 KVM/Xen Multiqueue

KVM/Multiqueue returns the frames to different queues of the host stack by classifying the incoming frame by processing the received packet's destination MAC address and or IEEE 802.1Q VLAN tag. The classification combined with the ability to DMA the frames directly into a virtual machine's memory allows scaling of virtual machines across multiple processors.

4.9 SR-IOV Configuration Support Matrix

- Windows VF over Windows hypervisor
- Windows VF and Linux VF over VMware hypervisor
- Linux VF over Linux KVM

4.10 SR-IOV

The PCI-SIG defines optional support for Single-Root IO Virtualization (SR-IOV). SR-IOV is designed to allow access of the VM directly to the device using Virtual Functions (VFs). The NIC Physical Function (PF) is divided into multiple virtual functions and each VF is presented as a PF to VMs.

SR-IOV uses IOMMU functionality to translate PCIe virtual addresses to physical addresses by using a translation table.

The number of Physical Functions (PFs) and Virtual Functions (VFs) are managed through the UEFI HII menu, the CCM, and through NVRAM configurations. SRIOV can be supported in combination with NPAR mode.

4.11 Network Partitioning (NPAR)

The Network Partitioning (NPAR) feature allows a single physical network interface port to appear to the system as multiple network device functions. When NPAR mode is enabled, the NetXtreme-E device is enumerated as multiple PCIe physical functions (PF). Each PF or “partition” is assigned a separate PCIe function ID on initial power on. The original PCIe definition allowed for eight PFs per device. For Alternative Routing-ID (ARI) capable systems, Broadcom NetXtreme-E adapters support up to 16 PFs per device. Each partition is assigned its own configuration space, BAR address, and MAC address allowing it to operate independently. Partitions support direct assignment to VMs, VLANs, and so on, just as any other physical interface.

4.12 RDMA over Converge Ethernet – RoCE

Remote Direct Memory Access (RDMA) over Converge Ethernet (RoCE) is a complete hardware offload feature in the BCM5741X that allows RDMA functionality over an Ethernet network. RoCE functionality is available in user mode and kernel mode application. RoCE Physical Functions (PF) and SRIOV Virtual Functions (VF) are available in single function mode and in mutli-function mode (NIC Partitioning mode). Broadcom supports RoCE in Windows, Linux, and VMware.

Refer to the following links for RDMA support for each operating system:

Windows

https://technet.microsoft.com/en-us/library/jj134210(v=ws.11).aspx

Redhat Linux

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/networking_guide/part-infiniband_and_rdma_networking

VMware

https://docs.vmware.com/en/VMware-vSphere/6.7/com.vmware.vsphere.networking.doc/GUID-E4ECDD76-75D6-4974-A225-04D5D117A9CF.html

4.13 Supported Combinations

The following sections describe the supported feature combinations for this device.

4.13.1 NPAR, SR-IOV, and RoCE

Table 8 provides the supported feature combinations of NPAR, SR-IOV, and RoCE.

Table 8: NPAR, SR-IOV, and RoCE

SW Feature	Notes
NPAR	Up to 8 PFs or 16 PFs
SR-IOV	Up to 128 VFs (total per chip)

Table 8: NPAR, SR-IOV, and RoCE (Continued)

SW Feature	Notes
RoCE on PFs	Up to 4 PFs
RoCE on VFs	Valid for VFs attached to RoCE-enabled PFs
Host OS	Linux, Windows, ESXi (no vRDMA support)
Guest OS	Linux and Windows
DCB	Up to two COS per port with non-shared reserved memory

4.13.2 NPAR, SR-IOV, and DPDK

Table 9 provides the supported feature combinations of NPAR, SR-IOV, and DPDK.

Table 9: NPAR, SR-IOV, and DPDK

SW Feature	Notes
NPAR	Up to 8 PFs or 16 PFs
SR-IOV	Up to 128 VFs (total per chip)
DPDK	Supported only as a VF
Host OS	Linux
Guest OS	DPDK (Linux)

4.13.3 Unsupported Combinations

The combination of NPAR, SR-IOV, RoCE, and DPDK is not supported.

5 Installing the Hardware

5.1 Safety Precautions

CAUTION! The adapter is being installed in a system that operates with voltages that can be lethal. Before removing the cover of the system, observe the following precautions to protect yourself and to prevent damage to the system components:

- Remove any metallic objects or jewelry from your hands and wrists.
- Make sure to use only insulated or nonconducting tools.
- Verify that the system is powered OFF and unplugged before you touch internal components.
- Install or remove adapters in a static-free environment. The use of a properly grounded wrist strap or other personal antistatic devices and an antistatic mat is strongly recommended.

5.2 System Requirements

Before installing the Broadcom NetXtreme-E Ethernet adapter, verify that the system meets the requirements listed for the operating system.

5.2.1 Hardware Requirements

Refer to the following list of hardware requirements:

- One open PCIe Gen 3 x8 or x 16 slot.
- 4 GB memory or more (32 GB or more is recommended for virtualization applications and nominal network throughput performance).

5.2.2 Preinstallation Checklist

Refer to the following list before installing the NetXtreme-C/NetXtreme-E device.

1. Verify that the server meets the hardware and software requirements listed in “System Requirements”.
2. Verify that the server is using the latest BIOS.
3. If the system is active, shut it down.
4. When the system shutdown is complete, turn off the power and unplug the power cord.
5. Holding the adapter card by the edges, remove it from its shipping package and place it on an antistatic surface.
6. Check the adapter for visible signs of damage, particularly on the card edge connector. Never attempt to install a damaged adapter.

5.3 Installing the Adapter

The following instructions apply to installing the Broadcom NetXtreme-E Ethernet adapter (add-in NIC) into most servers. Refer to the manuals that are supplied with the server for details about performing these tasks on this particular server.

1. Review the "Safety Precautions" on page 15 and "Preinstallation Checklist" before installing the adapter. Ensure that the system power is OFF and unplugged from the power outlet, and that proper electrical grounding procedures have been followed.
2. Open the system case and select any empty PCI Express Gen 3 x8 or x16 slot.
3. Remove the blank cover-plate from the slot.
4. Align the adapter connector edge with the connector slot in the system.
5. Secure the adapter with the adapter clip or screw.
6. Close the system case and disconnect any personal antistatic devices.

5.4 Connecting the Network Cables

Broadcom Ethernet switches are productized with SFP+/SFP28/QSFP28 ports that support up to 100 Gb/s. These 100 Gb/s ports can be divided into 4 x 25 Gb/s SFP28 ports. QSFP ports can be connected to SFP28 ports using 4 x 25G SFP28 breakout cables.

5.4.1 Supported Cables and Modules

5.4.1.1 Copper

The BCM957406AXXXX, BCM957416AXXXX, and BCM957416XXXX adapters have two RJ-45 connectors used for attaching the system to a CAT 6E Ethernet copper-wire segment.

5.4.1.2 SFP+

The BCM957302/402AXXXX, BCM957412AXXXX, and BCM957412MXXXX adapters have two SFP+ connectors used for attaching the system to a 10 Gb/s Ethernet switch.

5.4.1.3 SFP28

The BCM957404AXXXX, BCM957414XXXX, and BCM957414AXXXX adapters have two SFP28 connectors used for attaching the system to a 100 Gb/s Ethernet switch.

5.4.1.4 QSFP

The BCM957454XXXXXX, BCM957414AXXXX, and BCM957304XXXXX adapters have single QSFP connectors used for attaching the system to a 100 Gb/s Ethernet switch.

6 Software Packages and Installation

Refer to the following sections for information on software packages and installation.

6.1 Supported Operating Systems

Table 10 provides a list of supported operating systems.

Table 10: Supported Operating System List

OS Flavor	Distribution
Windows	Windows 2012 R2 or above
Linux	Redhat 6.9, Redhat 7.1 or above SLES 11 SP 4, SLES 12 SP 2 or above
VMware	ESXi 6.0 U3 or above

6.2 Installing the Linux Driver

Linux drivers can be downloaded from the Broadcom public website: https://www.broadcom.com/support/download-search/?pg=Ethernet+Connectivity,+Switching,+and+PHYs&pf=Ethernet+Network+Adapters+-+NetXtreme&pa=Driver.

See the package readme.txt files for specific instructions and optional parameters.

6.2.1 Linux Ethtool Commands

NOTE: In Table 11, ethX should be replaced with the actual interface name.

Table 11: Linux Ethtool Commands

Command	Description
ethtool -s ethX speed 25000 autoneg off	Set the speed. If the link is up on one port, the driver does not allow the other port to be set to an incompatible speed.
ethtool -i ethX	Output includes driver, firmware and package version.
ethtool -k ethX	Show offload features.
ethtool -K ethX tso off	Turn off TSO.
ethtool -K ethX gro off lro off	Turn off GRO/LRO.
ethtool -g ethX	Show ring sizes.
ethtool -G ethX rx N	Set Ring sizes.
ethtool -S ethX	Get statistics.
ethtool -l ethX	Show number of rings.
ethtool -L ethX rx 0 tx 0 combined M	Set number of rings.
ethtool -C ethX rx-frames N	Set interrupt coalescing. Other parameters supported are: rx-usecs, rx-frames, rx-usecs-irq, rx-frames-irq, tx-usecs, tx-frames, tx-usecs-irq, tx-frames-irq.
ethtool -x ethX	Show RSS flow hash indirection table and RSS key.
ethtool -s ethX autoneg on speed 10000 duplex full	Enable Autoneg (see “Auto-Negotiation Configuration” on page 24 for more details)
ethtool --show-eee ethX	Show EEE state.
ethtool --set-eee ethX eee off	Disable EEE.

Table 11: Linux Ethtool Commands (Continued)

Command	Description
ethtool --set-eee ethX eee on tx-lpi off	Enable EEE, but disable LPI.
ethtool -L ethX combined 1 rx 0 tx 0	Disable RSS. Set the combined channels to 1.
ethtool -K ethX ntuple off	Disable Accelerated RFS by disabling ntuple filters.
ethtool -K ethX ntuple on	Enable Accelerated RFS.
Ethtool -t ethX	Performs various diagnostic self-tests.
echo 32768 > /proc/sys/net/core/rps_sock_flow_entries echo 2048 > /sys/class/net/ethX/queues/rx-X/rps_flow_cnt	Enable RFS for Ring X.
sysctl -w net.core.busy_read=50	This sets the time to busy read the device's receive ring to 50 usecs. For socket applications waiting for data to arrive, using this method can decrease latency by 2 or 3 usecs typically at the expense of higher CPU utilization.
echo 4 > /sys/bus/pci/devices/0000:82:00.0/sriov_numvfs	Enable SR-IOV with four VFs on bus 82, Device 0 and Function 0.
ip link set ethX vf 0 mac 00:12:34:56:78:9a	Set VF MAC address.
ip link set ethX vf 0 state enable	Set VF link state for VF 0.
ip link set ethX vf 0 vlan 100	Set VF 0 with VLAN ID 100.

6.3 Installing the VMware Driver

The ESX drivers are provided in VMware standard VIB format and can be downloaded from VMware.com.

To install the Ethernet and RDMA driver, issue the following commands:

```
$ esxcli software vib install -v <bnxtnet>-<driver version>.vib

$ esxcli software vib install -v <bnxtroce>-<driver version>.vib
```

7. A system reboot is required for the new driver to take effect.

Other useful VMware commands shown in Table 12.

NOTE: In Table 12, vmnicX should be replaced with the actual interface name.

NOTE: `$ kill -HUP $(cat /var/run/vmware/vmkdevmgr.pid)` This command is required after `vmkload_mod bnxtnet` for successful module bring up.

Table 12: VMware Commands

Command	Description
esxcli software vib list |grep bnx	List the VIBs installed to see whether the bnxt driver installed successfully.
esxcfg-module –l bnxtnet	Print module info on to screen.
esxcli network get –n vmnicX	Get vmnicX properties.
esxcfg-module –g bnxtnet	Print module parameters.
esxcfg-module –s ‘multi_rx_filters=2 disable_tap=0 max_vfs=0,0 RSS=0’	Set the module parameters.
vmkload_mod –u bnxtnet	Unload bnxtnet module.
vmkload_mod bnxtnet	Load bnxtnet module.
esxcli network nic set –n vmnicX –D full –S 25000	Set the speed and duplex of vmnicX.

Table 12: VMware Commands (Continued)

Command	Description
esxcli network nic down –n vmnicX	Disable vmnicX.
esxcli network nic up –n vmnic6	Enable vmnicX.
bnxtnetcli –s –n vmnic6 –S “25000”	Set the link speed. Bnxtnetcli is needed for older ESX versions to support the 25G speed setting.

6.4 Installing the Windows Driver

To install the Windows drivers:

1. Download the Windows driver installation package can be downloaded from: https://www.broadcom.com/support/download-search/?pg=Ethernet+Connectivity,+Switching,+and+PHYs&pf=Ethernet+Network+Adapters+-+NetXtreme&pa=Driver.
2. Unzip the Wiin20xx_2xx.xx.x.zip file.
3. Launch the Device Manager.
4. Right-click on the Broadcom devices under Network Adapters.
5. Select **Update Driver**.
6. Select **Browse My Computer For Driver Software** and navigate to the folder where the driver files located. The driver l updates automatically.
7. Reboot the system to ensure that the driver is running.

6.4.1 Driver Advanced Properties

The Windows driver advanced properties are shown in Table 13.

Table 13: Windows Driver Advanced Properties

Driver Key	Parameters	Description
Encapsulated Task offload	Enable or Disable	Used for configuring NVGRE encapsulated task offload.
Energy Efficient Ethernet	Enable or Disable	EEE enabled for Copper ports and Disabled for SFP+ or SFP28 ports. This feature is only enabled for the BCM957406A4060 adapter.
Flow control	TX or RX or TX/RX enable	Configure flow control on RX or TX or both sides.
Interrupt Moderation	Enable or Disable	Default Enabled. Allows frames to be batch processed by saving CPU time.
Jumbo packet	1514, 4088, or 9014	Jumbo packet size.
Large Send offload V2 (IPv4)	Enable or Disable	LSO for IPv4.
Large Send offload V2 (IPv6)	Enable or Disable	LSO for IPv6.
Locally Administered Address	User entered MAC address.	Override default hardware MAC address after OS boot.
Max Number of RSS Queues	2, 4, or 8.	Default is 8. Allows user to configure Receive Side Scaling queues.

Table 13: Windows Driver Advanced Properties (Continued)

Driver Key	Parameters	Description
Priority and VLAN	Priority and VLAN Disable, Priority enabled, VLAN enabled, Priority and VLAN enabled.	Default Enabled. Used for configuring IEEE 802.1Q and IEEE 802.1P.
Receive Buffer (0=Auto)	Increments of 500.	Default is Auto.
Receive Side Scaling	Enable or Disable.	Default Enabled
Receive Segment Coalescing (IPv4)	Enable or Disable.	Default Enabled
Receive Segment Coalescing (IPv6)	Enable or Disable.	Default Enabled
RSS load balancing profile	NUMA scaling static, Closest processor, Closest processor static, conservative scaling, NUMA scaling.	Default NUMA scaling static.
Speed and Duplex	1 Gb/s, or 10 Gb/s, or 25 Gb/s, or Auto Negotiation.	10 Gb/s Copper ports can Auto negotiate speeds, whereas 25 Gb/s ports are set to forced speeds.
SR-IOV	Enable or Disable.	Default Enabled. This parameter works in conjunction with HW configured SR-IOV and BIOS configured SR-IOV setting.
TCP/UDP checksum offload IPv4	TX/RX enabled, TX enabled or RX Enabled or offload disabled.	Default RX and TX enabled.
TCP/UDP checksum offload IPv6	TX/RX enabled, TX enabled or RX Enabled or offload disabled.	Default RX and TX enabled.
Transmit Buffers (0=Auto)	Increment of 50.	Default Auto.
Virtual Machine Queue	Enable or Disable.	Default Enabled.
VLAN ID	User configurable number.	Default 0.

6.4.2 Event Log Messages

Table 14 provides the Event Log messages logged by the Windows NDIS driver to the event logs.

Table 14: Windows Event Log Messages

Message ID	Comment
0x0001	Failed Memory allocation.
0x0002	Link Down Detected.
0x0003	Link up detected.
0x0009	Link 1000 Full.
0x000A	Link 2500 Full.
0x000b	Initialization successful.
0x000c	Miniport Reset.
0x000d	Failed Initialization.
0x000E	Link 10GbE successful.
0x000F	Failed Driver Layer Binding.
0x0011	Failed to set Attributes.
0x0012	Failed scatter gather DMA.
0x0013	Failed default Queue initialization.
0x0014	Incompatible firmware version.
0x0015	Single interrupt.

Table 15: Event Log Messages

0x0016	Firmware failed to respond within allocated time.
0x0017	Firmware returned failure status.
0x0018	Firmware is in unknown state.
0x0019	Optics Module is not supported.
0x001A	Incompatible speed selection between Port 1 and Port 2. Reported link speeds are correct and might not match Speed and Duplex setting.
0x001B	Incompatible speed selection between Port 1 and Port 2. Link configuration became illegal.
0x001C	Network controller configured for 25 Gb full-duplex link.
0x001D	Network controller configured for 40 Gb full-duplex link.
0x001E	Network controller configured for 50 Gb full-duplex link.
0x001F	Network controller configured for 100 Gb full-duplex link.
0x0020	RDMA support initialization failed.
0x0021	Device's RDMA firmware is incompatible with this driver.
0x0022	Doorbell BAR size is too small for RDMA.
0x0023	RDMA restart upon device reset failed.
0x0024	RDMA restart upon system power up failed
0x0025	RDMA startup failed. Not enough resources.
0x0026	RDMA not enabled in firmware.
0x0027	Start failed, a MAC address is not set.
0x0028	Transmit stall detected. TX flow control is disabled from now on.

7 Updating the Firmware

7.1 Linux

To update the Linux firmware:

1. Download the firmware upgrade package from: https://www.broadcom.com/support/download-search/?pg=Ethernet+Connectivity,+Switching,+and+PHYs&pf=Ethernet+Network+Adapters+-+NetXtreme&pn=All&pa=Firmware&po=&dk= .
2. Execute the following commands:

   ```
   tar zxvf nxe_fw_upgrade.tgz
   ```

3. Execute the following command:

   ```
   ./fw_install.sh
   ```

4. Follow the install script steps.
5. Reboot the system for new firmware to take affect.

7.2 Windows/ESX

The NIC firmware can be upgraded using the NVRAM packages provided in the same link in the Linux session. Refer to the readme.txt for specific instructions for your adapter.

8 Teaming

8.1 Windows

The Broadcom NetXtreme-C/NetXtreme-E devices can participate in NIC teaming functionality using the Microsoft teaming solution. For more information on the NIC teaming functionality, see the Microsoft public documentation on Microsoft.com.

Microsoft LBFO is a native teaming driver that can be used in the Windows OS. The teaming driver also provides VLAN tagging capabilities.

8.2 Linux

Linux bonding is used for teaming under Linux. The concept is loading the bonding driver and adding team members to the bond which would load-balance the traffic.

Use the following steps to setup Linux bonding:

1. Execute the following command:

   ```
   modprobe bonding mode=”balance-alb”. This will create a bond interface.
   ```

2. Add bond clients to the bond interface. An example is shown below:

   ```
   ifenslave bond0 ethX; ifenslave bond0 ethY
   ```

3. Assign an IPv4 address to bond the interface using `ifconfig bond0 IPV4Address netmask NetMask up`. The IPV4Address and NetMask are an IPv4 address and the associated network mask.

NOTE: The IPv4 address should be replaced with the actual network IPv4 address. `NetMask` should be replaced by the actual IPv4 network mask.

4. Assign an IPv6 address to bond the interface using `ifconfig bond0 IPV6Address netmask NetMask up`. The IPV6Address and NetMask are an IPv6 address and the associated network mask.

NOTE: The IPv6 address should be replaced with the actual network IPv6 address. `NetMask` should be replaced by the actual IPv6 network mask.

Refer to the Linux Bonding documentation for advanced configurations.

9 System-Level Configuration

The following sections provide information on system-level NIC configuration.

9.1 UEFI HII Menu

Broadcom NetXtreme-E series controllers can be configured for pre-boot, iSCSI and advanced configuration such as SR-IOV using HII (Human Interface) menu.

To configure the settings, during system boot, enter BIOS setup and navigate to the network interface control menus.

9.1.1 Main Configuration Page

This page displays the current network link status, PCIe Bus:Device:Function, MAC address of the adapter and the Ethernet device.

The 10GBASE-T card allows the user to enable or disable Energy Efficient Ethernet (EEE).

9.1.2 Firmware Image Properties

Main configuration page – The firmware Image properties displays the Family version, which consists of version numbers of the controller BIOS, Multi Boot Agent (MBA), UEFI, iSCSI, and Comprehensive Configuration Management (CCM) version numbers.

9.1.3 Device-Level Configuration

Main configuration page – The device level configuration allows the user to enable SR-IOV mode, number of virtual functions per physical function, MSI-X vectors per Virtual function, and the Max number of physical function MSI-X vectors.

9.1.4 NIC Configuration

NIC configuration – Legacy boot protocol is used to select and configure PXE, iSCSI, or disable legacy boot mode. The boot strap type can be Auto, int18h (interrupt 18h), int19h (interrupt 19h), or BBS.

MBA and iSCSI can also be configured using CCM. Legacy BIOS mode uses CCM for configuration. The hide setup prompt can be used for disabling or enabling the banner display.

VLAN for PXE can be enabled or disabled and the VLAN ID can be configured by the user. See the Auto-Negotiation Configuration for details on link speed setting options.

9.1.5 iSCSI Configuration

iSCSI boot configuration can be set through the Main configuration page -> iSCSI configuration. Parameters such as IPv4 or IPv6, iSCSI initiator, or the iSCSI target are set through this page.

Refer to iSCSI Boot for detailed configuration information.

9.2 Comprehensive Configuration Management

For adapters that include CCM firmware (legacy boot only), preboot configuration can be configured using the Comprehensive Configuration Management (CCM) menu option. During the system BIOS POST, the Broadcom banner message is displayed with an option to change the parameters through the Control-S menu. When Control-S is pressed, a device list is populated with all the Broadcom network adapters found in the system. Select the desired NIC for configuration.

NOTE: Some adapters may not have CCM firmware in the NVRAM image and must use the HII menu to configure legacy parameters.

9.2.1 Device Hardware Configuration

Parameters that can be configured using this section are the same as the HII menu Device level configuration.

9.2.2 MBA Configuration Menu

Parameters that can be configured using this section are the same as the HII menu NIC configuration.

9.2.3 iSCSI Boot Main Menu

Parameters that can be configured using this section are the same as the HII menu iSCSI configuration.

9.3 Auto-Negotiation Configuration

NOTE: In NPAR (NIC partitioning) devices where one port is shared by multiple PCI functions, the port speed is preconfigured and cannot be changed by the driver.

The Broadcom NetXtreme-E controller supports the following auto-negotiation features:

- Link speed auto-negotiation
- Pause/Flow Control auto-negotiation

NOTE: Regarding link speed AN, when using SFP+, SFP28 connectors, use DAC or Multimode Optical transceivers capable of supporting AN. Ensure that the link partner port has been set to the matching auto-negotiation protocol. For example, if the local Broadcom port is set to IEEE 802.3by AN protocol, the link partner must support AN and must be set to IEEE 802.3by AN protocol.

NOTE: For dual ports NetXtreme-E network controllers, 10 Gb/s and 25 Gb/s are not a supported combination of link speed.

The supported combination of link speed settings for two ports NetXtreme-E network controller are shown in Table 16.

Table 16: Supported Combination of Link Speed Settings

Port1 Link Speed Setting	Port 2 Link Setting									
	Forced 1G	**Forced 10G**	**Forced 25G**	**AN Enabled {1G}**	**AN Enabled {10G}**	**AN Enabled {25G}**	**AN Enabled {1/10G}**	**AN Enabled {1/25G}**	**AN Enabled {10/25G}**	**AN Enabled {1/10/25G}**
Forced 1G	P1: no AN	P1: no AN	P1: no AN	P1: no AN	P1: no AN	P1: no AN	P1: no AN	P1: no AN	P1: no AN	P1: no AN
	P2: no AN	P2: no AN	P2: no AN	P2: {1G}	P2: AN {10G}	P2: AN {25G}	P2: AN {1/10G}	P2: AN {1/25G}	P2: AN {10/25G}	P2: AN {1/10/25G}
Forced 10G	P1: no AN	P1: no AN	Not supported	P1: no AN	P1: no AN	Not supported	P1: no AN	P1: no AN	P1: no AN	P1: no AN
	P2: no AN	P2: no AN		P2: {1G}	P2: {10G}		P2: AN {1/10G}	P2: AN {1G}	P2: AN {10G}	P2: AN {1/10G}
Forced 25G	P1: no AN	Not supported	P1: no AN	P1: no AN	P1: no AN	P1: no AN	P1: no AN	P1: no AN	P1: no AN	P1: no AN
	P2: no AN		P2: no AN	P2: no AN	P2: no AN	P2: no AN	P2: AN {1G}	P2: AN {1/25G}	P2: AN {25G}	P2: AN {1/25G}
AN Enabled {1G}	P1: {1G}	P1: {1G}	P1: {1G}	P1: AN {1G}	P1: AN {1G}	P1: AN {1G}	P1: AN {1G}	P1: AN {1G}	P1: AN {1G}	P1: AN {1G}
	P2: no AN	P2: no AN	P2: no AN	P2: AN {1G}	P2: AN {10G}	P2: AN {25G}	P2: AN {1/10G}	P2: AN {1/25G}	P2: AN {10/25G}	P2: AN {1/10/25G}
AN Enabled {10G}	P1: AN {10G}	P1: AN {25G}	Not supported	P1: AN {10G}	P1: AN {10G}	Not supported	P1: AN {25G}	P1: AN {10G}	P1: AN {10G}	P1: AN {10G}
	P2: no AN	P2: no AN		P2: AN {1G}	P2: AN {10G}		P2: AN {1G}	P2: AN {1G}	P2: AN {10G}	P2: AN {1/10G}
AN Enabled {25G}	P1: AN {25G}	Not supported	P1: AN {25G}	P1: AN {25G}	Not supported	P1: AN {25G}	P1: AN {1/10G}	P1: AN {25G}	P1: AN {25G}	P1: AN {25G}
	P2: no AN		P2: no AN	P2: AN {1G}		P2: AN {25G}	P2: AN {1/10G}	P2: AN {1/25G}	P2: AN {25G}	P2: AN {1/25G}
AN Enabled {1/10G}	P1: AN {1/10G}	P1: AN {1/10G}	P1: AN {1G}	P1: AN {1/10G}	P1: AN {1/10G}	P1: AN {1/10G}	P1: AN {1/25G}	P1: AN {1G}	P1: AN {1/10G}	P1: AN {1/10G}
	P2: no AN	P2: no AN	P2: no AN	P2: AN {1G}	P2: AN {10G}	P2: AN {25G}	P2: AN {1/10G}	P2: AN {1G}	P2: AN {10G}	P2: AN {1/10G}
AN Enabled {1/25G}	P1: AN {1/25G}	P1: {1G}	P1: AN {1/25G}	P1: AN {1/25G}	P1: AN {1G}	P1: AN {1/25G}	P1: AN {10/25G}	P1: AN {1/25G}	P1: AN {1/25G}	P1: AN {1/25G}
	P2: no AN	P2: no AN	P2: no AN	P2: AN {1G}	P2: AN {10G}	P2: AN {25G}	P2: AN {1/10G}	P2: AN {1/25G}	P2: AN {25G}	P2: AN {1/25G}
AN Enabled {10/25G}	P1: AN {10/25G}	P1: {10G}	P1: AN {25G}	P1: AN {10/25G}	P1: AN {10G}	P1: AN {25G}	P1: AN {1/10/25G}	P1: AN {25G}	P1: AN {10/25G}	P1: AN {10/25G}
	P2: no AN	P2: no AN	P2: no AN	P2: AN {1G}	P2: AN {10G}	P2: AN {25G}	P2: AN {1/10G}	P2: AN {25G}	P2: AN {10/25G}	P2: AN {1/10/25G}
AN Enabled {1/10/25G}	P1: AN {1/10/25G}	P1: {1/10G}	P1: AN {1/25G}	P1: AN {1/10/25G}	P1: AN {1/10G}	P1: AN {1/25G}	P1: AN {1/10/25G}	P1: AN {1/25G}	P1: AN {1/10/25G}	P1: AN {1/10/25G}
	P2: no AN	P2: no AN	P2: no AN	P2: AN {1G}	P2: AN {10G}	P2: AN {25G}	P2: AN {1/10G}	P2: AN {1/25G}	P2: AN {10/25G}	P2: AN {1/10/25G}

NOTE: 1 Gb/s link speed for SFP+/SFP28 is currently not support in this release.

- P1 – port 1 setting
- P2 – port 2 setting
- AN – auto-negotiation
- No AN – forced speed
- {link speed} – expected link speed
- AN {link speeds} – advertised supported auto-negotiation link speeds.

The expected link speeds based on the local and link partner settings are shown in Table 17.

Table 17: Expected Link Speeds

Local Speed Settings	Link Partner Speed Settings									
	Forced 1G	**Forced 10G**	**Forced 25G**	**AN Enabled {1G}**	**AN Enabled {10G}**	**AN Enabled {25G}**	**AN Enabled {1/10G}**	**AN Enabled {1/ 25G}**	**AN Enabled {10/25G}**	**AN Enabled {1/10/25G}**
Forced 1G	1G	No link	No link	No link	No link	No link	No link	No link	No link	No link
Forced 10G	No link	10G	No link	No link	No link	No link	No link	No link	No link	No link
Forced 25G	No link	No link	25G	No link	No link	No link	No link	No link	No link	No link
AN {1G}	No link	No link	No link	1G	No link	No link	1G	1G	No link	1G
AN {10G}	No link	No link	No link	No link	10G	No link	10G	No link	10G	10G
AN {25G}	No link	No link	No link	No link	No link	25G	No link	25G	25G	25G
AN {1/10G}	No link	No link	No link	1G	10G	No link	10G	1G	10G	10G
AN {1/25G}	No link	No link	No link	1G	No link	25G	1G	25G	25G	25G
AN {10/25G}	No link	No link	No link	No link	10G	25G	10G	25G	25G	25G
AN {1/10/25G}	No link	No link	No link	1G	10G	25G	10G	25G	25G	25G

To enable link speed auto-negotiation, the following options can be enabled in system BIOS HII menu or in CCM:

System BIOS->NetXtreme-E NIC->Device Level Configuration

9.3.1 Operational Link Speed

This option configures the link speed used by the OS driver and firmware. This setting is overridden by the driver setting in the OS present state.

9.3.2 Firmware Link Speed

This option configures the link speed used by the firmware when the device is in D3.

9.3.3 Auto-negotiation Protocol

This is the supported auto-negotiation protocol used to negotiate the link speed with the link partner. This option must match the AN protocol setting in the link partner port. The Broadcom NetXtreme-E NIC supports the following auto-negotiation protocols: IEEE 802.3by, 25G/50G consortiums and 25G/50G BAM. By default, this option is set to IEEE 802.3by.

Link speed and Flow Control/Pause must be configured in the driver in the host OS.

9.3.4 Windows Driver Settings

To access the Windows driver settings:

Open **Windows Manager -> Broadcom NetXtreme E Series adapter -> Advanced Properties -> Advanced tab**

Flow Control = Auto-Negotiation

This enables Flow Control/Pause frame AN.

Speed and Duplex = Auto-Negotiation

This enables link speed AN.

9.3.5 Linux Driver Settings

NOTE: For 10GBASE-T NetXtreme-E network adapters, auto-negotiation must be enabled.

NOTE: 25G and 50G advertisements are newer standards first defined in the 4.7 kernel's ethtool interface. To fully support these new advertisement speeds for auto-negotiation, a 4.7 (or newer) kernel and a newer ethtool utility (version 4.8) are required.

- ethtool -s eth0 speed 25000 autoneg off – This command turns off auto-negotiation and forces the link speed to 25 Gb/s.
- ethtool -s eth0 autoneg on advertise 0x0 – This command enables auto-negotiation and advertises that the device supports all speeds: 1G, 10G, 25G (and 40G, 50G if applicable).

  The following are supported advertised speeds.
  - 0x020 – 1000BASE-T Full
  - 0x1000 – 1000BASE-T Full
  - 0x80000000 – 25000BASE-CR Full
- ethtool -A eth0 autoneg on|off – Use this command to enable/disable pause frame auto-negotiation.
- ethtool -a eth0 – Use this command to display the current flow control auto-negotiation setting.

9.3.6 ESXi Driver Settings

NOTE: For 10GBASE-T NetXtreme-E network adapters, auto-negotiation must be enabled. Using forced speed on a 10GBASE-T adapter results in esxcli command failure.

NOTE: VMware does not support 25G/50G speeds in ESX6.0. In this case, use the second utility (BNXTNETCLI) to set 25G/50G speed. For ESX6.0U2, the 25G/50G speed is supported.

- $ esxcli network nic get -n <iface> – This command shows the current speed, duplex, driver version, firmware version and link status.
- $ esxcli network nic set -S 10000 -D full -n <iface> – This command sets the forced speed to 10 Gb/s.
- $ esxcli network nic set -a -n <iface> – This enables linkspeed auto-negotiation on interface <iface>.
- $ esxcli network nic pauseParams list – Use this command to get pause Parameters list.
- $ esxcli network nic pauseParams set --auto <1/0> --rx <1/0> --tx <1/0> -n <iface> – Use this command to set pause parameters.

NOTE: Flow control/pause auto-negotiation can be set only when the interface is configured in link speed auto-negotiation mode.

10 iSCSI Boot

Broadcom NetXtreme-E Ethernet adapters support iSCSI boot to enable the network boot of operating systems to diskless systems. iSCSI boot allows a Windows, Linux, or VMware operating system to boot from an iSCSI target machine located remotely over a standard IP network.

10.1 Supported Operating Systems for iSCSI Boot

The Broadcom NetXtreme-E Gigabit Ethernet adapters support iSCSI boot on the following operating systems:

- Windows Server 2012 and later 64-bit
- Linux RHEL 7.1 and later, SLES11 SP4 or later
- VMware 6.0 U2

10.2 Setting up iSCSI Boot

Refer to the following sections for information on setting up iSCSI boot.

10.2.1 Configuring the iSCSI Target

Configuring the iSCSI target varies per the target vendor. For information on configuring the iSCSI target, refer to the documentation provided by the vendor. The general steps include:

1. Create an iSCSI target.
2. Create a virtual disk.
3. Map the virtual disk to the iSCSI target created in Step 1 on page 29.
4. Associate an iSCSI initiator with the iSCSI target.
5. Record the iSCSI target name, TCP port number, iSCSI Logical Unit Number (LUN), initiator Internet Qualified Name (IQN), and CHAP authentication details.
6. After configuring the iSCSI target, obtain the following:
   - Target IQN
   - Target IP address
   - Target TCP port number
   - Target LUN
   - Initiator IQN
   - CHAP ID and secret

10.2.2 Configuring iSCSI Boot Parameters

Configure the Broadcom iSCSI boot software for either static or dynamic configuration. Refer to Table 18 for configuration options available from the General Parameters menu. Table 18 lists parameters for both IPv4 and IPv6. Parameters specific to either IPv4 or IPv6 are noted.

Table 18: Configuration Options

Option	Description
TCP/IP parameters via DHCP	This option is specific to IPv4. Controls whether the iSCSI boot host software acquires the IP address information using DHCP (Enabled) or use a static IP configuration (Disabled).
IP Autoconfiguration	This option is specific to IPv6. Controls whether the iSCSI boot host software configures a stateless link-local address and/or stateful address if DHCPv6 is present and used (Enabled). Router Solicit packets are sent out up to three times with 4-second intervals in between each retry. Or use a static IP configuration (Disabled).
iSCSI parameters via DHCP	Controls whether the iSCSI boot host software acquires its iSCSI target parameters using DHCP (Enabled) or through a static configuration (Disabled). The static information is entered through the iSCSI Initiator Parameters Configuration screen.
CHAP Authentication	Controls whether the iSCSI boot host software uses CHAP authentication when connecting to the iSCSI target. If CHAP Authentication is enabled, the CHAP ID and CHAP Secret are entered through the iSCSI Initiator Parameters Configuration screen.
DHCP Vendor ID	Controls how the iSCSI boot host software interprets the Vendor Class ID field used during DHCP. If the Vendor Class ID field in the DHCP Offer packet matches the value in the field, the iSCSI boot host software looks into the DHCP Option 43 fields for the required iSCSI boot extensions. If DHCP is disabled, this value does not need to be set.

Table 18: Configuration Options (Continued)

Option	Description
Link Up Delay Time	Controls how long the iSCSI boot host software waits, in seconds, after an Ethernet link is established before sending any data over the network. The valid values are 0 to 255. As an example, a user may need to set a value for this option if a network protocol, such as Spanning Tree, is enabled on the switch interface to the client system.
Use TCP Timestamp	Controls if the TCP Timestamp option is enabled or disabled.
Target as First HDD	Allows specifying that the iSCSI target drive appears as the first hard drive in the system.
LUN Busy Retry Count	Controls the number of connection retries the iSCSI Boot initiator attempts if the iSCSI target LUN is busy.
IP Version	This option specific to IPv6. Toggles between the IPv4 or IPv6 protocol. All IP settings are lost when switching from one protocol version to another.

10.2.3 MBA Boot Protocol Configuration

To configure the boot protocol:

1. Restart the system.
2. From the PXE banner, select **CTRL+S**. The **MBA Configuration Menu** displays.
3. From the **MBA Configuration Menu**, use the **up arrow** or **down arrow** to move to the Boot Protocol option. Use the **left arrow** or **right arrow** to change the Boot Protocol option for iSCSI.
4. Select **iSCSI Boot Configuration** from **Main Menu**.

10.2.4 iSCSI Boot Configuration

There are two ways to configure iSCSI boot:

- Static iSCSI Boot Configuration.
- Dynamic iSCSI Boot Configuration.

10.2.4.1 Static iSCSI Boot Configuration

In a static configuration, you must enter data for the system's IP address, the system's initiator IQN, and the target parameters obtained in “Configuring the iSCSI Target” on page 29. For information on configuration options, see Table 18 on page 29.

To configure the iSCSI boot parameters using static configuration:

1. From the **General Parameters** menu, set the following:
   - TCP/IP parameters via DHCP – Disabled. (For IPv4).
   - IP Autoconfiguration – Disabled. (For IPv6, non-offload).
   - iSCSI parameters via DHCP – Disabled.
   - CHAP Authentication – Disabled.
   - DHCP Vendor ID – BRCM ISAN.
   - Link Up Delay Time – 0.
   - Use TCP Timestamp – Enabled (for some targets, it is necessary to enable Use TCP Timestamp).
   - Target as First HDD – Disabled.
   - LUN Busy Retry Count – 0.
   - IP Version – IPv6. (For IPv6, non-offload).

2. Select **ESC** to return to the **Main** menu.
3. From the **Main** menu, select **Initiator Parameters**.
4. From the **Initiator Parameters** screen, enter values for the following:
   – IP Address (unspecified IPv4 and IPv6 addresses should be "0.0.0.0" and "::", respectively)
   – Subnet Mask Prefix
   – Default Gateway
   – Primary DNS
   – Secondary DNS
   – iSCSI Name (corresponds to the iSCSI initiator name to be used by the client system)

NOTE: Enter the IP address. There is no error-checking performed against the IP address to check for duplicates or incorrect segment/network assignment.

5. Select **Esc** to return to the **Main** menu.
6. From the **Main** menu, select **1st Target Parameters**.

NOTE: For the initial setup, configuring a second target is not supported.

7. From the **1st Target Parameters** screen, enable **Connect** to connect to the iSCSI target. Type values for the following using the values used when configuring the iSCSI target:
   – IP Address
   – TCP Port
   – Boot LUN
   – iSCSI Name
8. Select **Esc** to return to the **Main** menu.
9. Select **Esc** and select **Exit** and **Save Configuration**.
10. Select **F4** to save the MBA configuration.

10.2.4.2 Dynamic iSCSI Boot Configuration

In a dynamic configuration, specify that the system's IP address and target/initiator information are provided by a DHCP server (see IPv4 and IPv6 configurations in "Configuring the DHCP Server to Support iSCSI Boot" on page 33). For IPv4, with the exception of the initiator iSCSI name, any settings on the Initiator Parameters, 1st Target Parameters, or 2nd Target Parameters screens are ignored and do not need to be cleared. For IPv6, with the exception of the CHAP ID and Secret, any settings on the Initiator Parameters, 1st Target Parameters, or 2nd Target Parameters screens are ignored and do not need to be cleared. For information on configuration options, see Table 18 on page 29.

NOTE: When using a DHCP server, the DNS server entries are overwritten by the values provided by the DHCP server. This occurs even if the locally provided values are valid and the DHCP server provides no DNS server information. When the DHCP server provides no DNS server information, both the primary and secondary DNS server values are set to 0.0.0.0. When the Windows OS takes over, the Microsoft iSCSI initiator retrieves the iSCSI Initiator parameters and configures the appropriate registries statically. It overwrites whatever is configured. Since the DHCP daemon runs in the Windows environment as a user process, all TCP/IP parameters have to be statically configured before the stack comes up in the iSCSI Boot environment.

- If DHCP Option 17 is used, the target information is provided by the DHCP server, and the initiator iSCSI name is retrieved from the value programmed from the Initiator Parameters screen. If no value was selected, then the controller defaults to the name:

```
iqn.1995-05.com.broadcom.<11.22.33.44.55.66>.iscsiboot
```

where the string **11.22.33.44.55.66** corresponds to the controller's MAC address.

- If DHCP option 43 (IPv4 only) is used, then any settings on the Initiator Parameters, 1st Target Parameters, or 2nd Target Parameters screens are ignored and do not need to be cleared.

To configure the iSCSI boot parameters using a dynamic configuration:

1. From the **General Parameters** menu screen, set the following parameters:
   - TCP/IP parameters via DHCP – Enabled. (For IPv4.)
   - IP Autoconfiguration – Enabled. (For IPv6, non-offload.)
   - iSCSI parameters via DHCP – Enabled
   - CHAP Authentication – Disabled
   - DHCP Vendor ID – BRCM ISAN
   - Link Up Delay Time – 0
   - Use TCP Timestamp – Enabled (for some targets, it is necessary to enable Use TCP Timestamp)
   - Target as First HDD – Disabled
   - LUN Busy Retry Count – 0
   - IP Version – IPv6. (For IPv6, non-offload.)
2. Select **ESC** to return to the **Main** menu.

NOTE: Information on the Initiator Parameters, and 1st Target Parameters screens are ignored and do not need to be cleared.

3. Select **Exit** and **Save Configurations**.

10.2.5 Enabling CHAP Authentication

Ensure that CHAP authentication is enabled on the target.

To enable CHAP authentication:

1. From the **General Parameters** screen, set **CHAP Authentication** to **Enabled**.
2. From the **Initiator Parameters** screen, enter the parameters for the following:
   - CHAP ID (up to 128 bytes)
   - CHAP Secret (if authentication is required, and must be 12 characters in length or longer)
3. Select **ESC** to return to the **Main** menu.
4. From the **Main** menu, select the **1st Target Parameters**.
5. From the **1st Target Parameters** screen, type values for the following using the values used when configuring the iSCSI target:
   - CHAP ID (optional if two-way CHAP)
   - CHAP Secret (optional if two-way CHAP, and must be 12 characters in length or longer)
6. Select **ESC** to return to the **Main** menu.
7. Select **ESC** and select **Exit** and **Save Configuration**.

10.3 Configuring the DHCP Server to Support iSCSI Boot

The DHCP server is an optional component and it is only necessary for dynamic iSCSI Boot configuration setup (see "Dynamic iSCSI Boot Configuration" on page 31).

Configuring the DHCP server to support iSCSI boot is different for IPv4 and IPv6. Refer to the following sections:

10.3.1 DHCP iSCSI Boot Configurations for IPv4

The DHCP protocol includes a number of options that provide configuration information to the DHCP client. For iSCSI boot, Broadcom adapters support the following DHCP configurations:

10.3.1.1 DHCP Option 17, Root Path

Option 17 is used to pass the iSCSI target information to the iSCSI client. The format of the root path as defined in IETC RFC 4173 is:

```
iscsi:"<servername>":"<protocol>":"<port>":"<LUN>":"<targetname>
```

The parameters are defined in Table 19.

Table 19: DHCP Option 17 Parameter Definition

Parameter	Definition
"iscsi:"	A literal string.
<servername>	The IP address or FQDN of the iSCSI target
":"	Separator.
<protocol>	The IP protocol used to access the iSCSI target. Currently, only TCP is supported so the protocol is 6.
<port>	The port number associated with the protocol. The standard port number for iSCSI is 3260.
<LUN>	The Logical Unit Number to use on the iSCSI target. The value of the LUN must be represented in hexadecimal format. A LUN with an ID OF 64 would have to be configured as 40 within the option 17 parameter on the DHCP server.
<targetname>	The target name in either IQN or EUI format (refer to RFC 3720 for details on both IQN and EUI formats). An example IQN name would be iqn.1995-05.com.broadcom:iscsi-target.

10.3.1.2 DHCP Option 43, Vendor-Specific Information

DHCP option 43 (vendor-specific information) provides more configuration options to the iSCSI client than DHCP option 17. In this configuration, three additional suboptions are provided that assign the initiator IQN to the iSCSI boot client along with two iSCSI target IQNs that can be used for booting. The format for the iSCSI target IQN is the same as that of DHCP option 17, while the iSCSI initiator IQN is simply the initiator's IQN.

NOTE: DHCP Option 43 is supported on IPv4 only.

The suboptions are listed below.

Table 20: DHCP Option 43 Suboption Definition

Suboption	Definition
201	First iSCSI target information in the standard root path format iscsi:"<servername>":"<protocol>":"<port>":"<LUN>":"<targetname>
203	iSCSI initiator IQN

Using DHCP option 43 requires more configuration than DHCP option 17, but it provides a richer environment and provides more configuration options. Broadcom recommends that customers use DHCP option 43 when performing dynamic iSCSI boot configuration.

10.3.1.3 Configuring the DHCP Server

Configure the DHCP server to support option 17 or option 43.

NOTE: If Option 43 is used, configure Option 60. The value of Option 60 should match the DHCP Vendor ID value. The DHCP Vendor ID value is BRCM ISAN, as shown in General Parameters of the iSCSI Boot Configuration menu.

10.3.2 DHCP iSCSI Boot Configuration for IPv6

The DHCPv6 server can provide a number of options, including stateless or stateful IP configuration, as well as information to the DHCPv6 client. For iSCSI boot, Broadcom adapters support the following DHCP configurations:

NOTE: The DHCPv6 standard Root Path option is not yet available. Broadcom suggests using Option 16 or Option 17 for dynamic iSCSI Boot IPv6 support.

10.3.2.1 DHCPv6 Option 16, Vendor Class Option

DHCPv6 Option 16 (vendor class option) must be present and must contain a string that matches the configured DHCP Vendor ID parameter. The DHCP Vendor ID value is BRCM ISAN, as shown in General Parameters of the iSCSI Boot Configuration menu.

The content of Option 16 should be <2-byte length> <DHCP Vendor ID>.

10.3.2.2 DHCPv6 Option 17, Vendor-Specific Information

DHCPv6 Option 17 (vendor-specific information) provides more configuration options to the iSCSI client. In this configuration, three additional suboptions are provided that assign the initiator IQN to the iSCSI boot client along with two iSCSI target IQNs that can be used for booting.

The suboptions are listed in Table 21 on page 35.

Table 21: DHCP Option 17 Suboption Definition

Suboption	Definition
201	First iSCSI target information in the standard root path format "iscsi:"[<servername>]":"<protocol>":"<port>":"<LUN>":"<targetname>"
203	iSCSI initiator IQN

NOTE: In Table 21, the brackets [ ] are required for the IPv6 addresses.

The content of option 17 should be <2-byte Option Number 201|202|203> <2-byte length> <data>.

10.3.2.3 Configuring the DHCP Server

Configure the DHCP server to support Option 16 and Option 17.

NOTE: The format of DHCPv6 Option 16 and Option 17 are fully defined in RFC 3315.

11 VXLAN – Configuration and Use Case Examples

VXLAN encapsulation permits many layer 3 hosts residing on one server to send and receive frames by encapsulating on to single IP address associated with the NIC card installed on the same server.

This example discusses basic VXLAN connectivity between two RHEL servers. Each server has one physical NIC enabled with outer IP address set to 1.1.1.4 and 1.1.1.2.

A VXLAN10 interface with VXLAN ID 10 is created with multicast group 239.0.0.10 and is associated with physical network port pxp1 on each server.

An IP address for the host is created on each server and associated that to VXLAN interface. Once the VXLAN interface is brought up, the host present in system 1 can communicate with host present in system 2. The VLXAN format is shown in Table 22.

Table 22: VXLAN Frame Format

MAC header	Outer IP header with proto = UDP	UDP header with Destination port= VXLAN	VXLAN header (Flags, VNI)	Original L2 Frame	FCS

Table 23 provides VXLAN command and configuration examples.

Table 23: VXLAN Command and Configuration Examples

System 1	System 2
PxPy: ifconfig PxPy 1.1.1.4/24	PxPy: ifconfig PxPy 1.1.1.2/24
IP LINK ADD VXLAN10 TYPE VXLAN ID 10 GROUP 239.0.0.10 DEV PXPY DSTPORT 4789	IP link add vxlan10 type vxlan id 10 group 239.0.0.10 dev PxPy dstport 4789
IP addr add 192.168.1.5/24 broadcast 192.168.1.255 dev vxlan10	IP addr add 192.168.1.10/24 broadcast 192.168.1.255 dev vxlan10
IP link set vxlan10 up	IP link set vxlan10 up
ip –d link show vxlan10	
Ping 192.168.1.10	ifconfig vxlan10 (MTU 1450) (SUSE and RHEL)
NOTE: X represents the PCIe bus number of the physical adapter found in the system. Y represents the port number on the physical adapter.	

12 SR-IOV – Configuration and Use Case Examples

SR-IOV can be configured, enabled, and used on 10-Gb and 25-Gb Broadcom NetExtreme-E NICs.

12.1 Linux Use Case Example

1. Enable SR-IOV in the NIC cards:
   a. SR-IOV in the NIC card can be enabled using the **HII** menu. During system boot, access the system **BIOS -> NetXtreme-E NIC -> Device Level Configuration**.
   b. Set the Virtualization mode to SR-IOV.
   c. Set the number of virtual functions per physical function.
   d. Set the number of MSI-X vectors per the VF and Max number of physical function MSI-X vectors. If the VF is running out of resources, balance the number of MSI-X vectors per VM using CCM.
2. Enable virtualization in the BIOS:
   a. During system boot, enter the system **BIOS -> Processor settings -> Virtualization Technologies** and set it to **Enabled.**
   b. During system boot, enter the system **BIOS -> SR-IOV Global** and set it to **Enabled**.
3. Install the desired Linux version with Virtualization enabled (libvirt and Qemu).
4. Enable the iommu kernel parameter.
   a. The IOMMU kernel parameter is enabled by editing `/etc/default/grub.cfg` and running `grub2-mkconfig -o /boot/grub2/grub.cfg` for legacy mode. For UEFI mode, edit `/etc/default/grub.cfg` and run `grub2-mkconfig -o /etc/grub2-efi.cfg`. Refer to the following example:

```
Linuxefi /vmlinuz-3.10.0-229.el7.x86_64 root=/dev/mapper/rhel-root ro rd.lvm.lv=rhel/swap
crashkernel=auto rd.lvm.lv=rhel/root rhgb intel_iommu=on quiet LANG=en_US.UTF.8
```

5. Install bnxt_en driver:
   a. Copy the `bnxt_en` driver on to the OS and run `make; make install; modprobe bnxt_en`.

NOTE: Use `netxtreme-bnxt_en<version>.tar.gz` to install both `bnxt_re` and `bnxt_en` for RDMA functionality on SRIOV VFs.

6. Enable Virtual Functions through Kernel parameters:
   a. Once the driver is installed, lspci displays the NetXtreme-E NICs present in the system. Bus, device, and Function are needed for activating Virtual functions.
   b. To activate Virtual functions, enter the command shown below:

```
echo X >/sys/bus/pci/device/0000\:Bus\:Dev.Function/sriov_numvfs
```

NOTE: Ensure that the PF interfaces are up. VFs are only created if PFs are up. X is the number of VFs that are exported to the OS.

A typical example would be:

```
echo 4 > /sys/bus/pci/devices/0000\:04\:00.0/sriov_numvfs
```

7. Check the PCIe virtual functions:

   The `lspci` command displays the virtual functions with DID set to 16D3 for BCM57402/BCM57404/BCM57406, 16DC for non-RDMA BCM57412/BCM57414/BCM57416, and 16C1 or RDMA enabled BCM57412/BCM57414/BCM57416.

8. Use the Virtual Manager to install a Virtualized Client system (VMs).

   Refer to the Linux documentation for Virtual Manager installation. Ensure that the hypervisor's built in driver is removed. An example would be `NIC:d7:73:a7 rtl8139`. Remove this driver.

9. Assign a virtual function to the guest VMs.

   Assign this adapter to a guest VM as a physical PCI Device. Refer to the Linux documentation for information on assigning virtual functions to a VM guest.

10. Install `bnxt_en` drivers on VMs:

    On the guest VMs, copy the `netxtreme-bnxt_en-<version>.tar.gz` source file and extract the `tar.gz` file. Change directory to each driver and run `make; make install; modprobe bnxt_en` (and `bnxt_re` if enabling RDMA). Make sure that the driver loads properly by checking the interface using modinfo command. The user may need to run `modprobe -r bnxt_en` to unload existing or inbox `bnxt_en` module prior to loading the latest built module.

11. Test the guest VM connectivity to external world:

    Assign proper IP address to the adapter and test the network connectivity.

12.2 Windows Use Case Example

1. Enable SR-IOV in the NIC cards:
   a. SR-IOV in the NIC card can be enabled using the HII menu. During the system boot, access the system **BIOS -> NetXtreme-E NIC -> Device Level Configuration**.
   b. Set the Virtualization mode to SR-IOV.
   c. Set the number of virtual functions per physical function.
   d. Set the number of MSI-X vectors per the VF and Max number of physical function MSI-X vectors. If the VF is running out of resources, balance the number of MSI-X vectors per VM using CCM.
2. Enable virtualization in the BIOS:
   a. During system boot, enter the system **BIOS -> Processor settings -> Virtualization Technologies** and set it to **Enabled**.
   b. During system boot, enter the system **BIOS -> SR-IOV Global** and set it to **Enabled**.
3. Install the latest KB update for your Windows 2012 R2 or Windows 2016 OS.
4. Install the appropriate Virtualization (Hyper-V) options. For more detail requirements and steps on setting up Hyper-V, Virtual Switch, and Virtual Machine, visit Microsoft.com.
5. Install the latest NetXtreme-E driver on the Hyper-V.
6. Enable SR-IOV in the NDIS miniport driver advanced properties.
7. In Hyper-V Manager, create your Virtual Switch with the selected NetXtreme-E interface.
8. Check the **Enable Single-Root I/O Virtualization (SR-IOV)** box while creating the Hyper-V Virtual Adapter.
9. Create a Virtual Machine (VM) and add the desired number of Virtual Adapters.
10. Under the Virtual Machine's Network Adapter settings for each Virtual Adapter, check **Enable SR-IOV** under the **Hardware Acceleration** section.
11. Launch your VM and install the desired guest OS.
12. Install the corresponding NetXtreme-E driver for each guest OS.

NOTE: The Virtual Function (VF) driver for NetXtreme-E is same driver as the base driver. For example, if the guest OS is Windows 2012 R2, the user needs to install Bnxtnd64.sys in VM. The user can do this by running the NetXtreme-E Driver Installer executable. Once the driver has been installed in the guest OS, the user can see the VF driver interface(s) appearing in Device Manager of the guest OS in the VM.

12.3 VMware SRIOV Use Case Example

1. Enable SR-IOV in the NIC cards:
   a. SR-IOV in the NIC card can be enabled using the HII menu. During the system boot, access the system **BIOS -> NetXtreme-E NIC -> Device Level Configuration**.
   b. Set the Virtualization mode to SR-IOV.
   c. Set the number of virtual functions per physical function.
   d. Set the number of MSI-X vectors per the VF and Max number of physical function MSI-X vectors. If the VF is running out of resources, balance the number of MSI-X vectors per VM using CCM.
2. Enable virtualization in the BIOS:
   a. During system boot, enter the system **BIOS -> Processor settings -> Virtualization Technologies** and set it to **Enabled**.
   b. During system boot, enter the system **BIOS -> SR-IOV Global** and set it to **Enabled**.
3. On ESXi, install the Bnxtnet driver using the following steps:
   a. Copy the <bnxtnet>-<driver version>.vib file in /var/log/vmware.

```
$ cd /var/log/vmware.
$ esxcli software vib install --no-sig-check -v <bnxtnet>-<driver version>.vib.
```

   b. Reboot the machine.
   c. Verify that whether drivers are correctly installed:

```
$ esxcli software vib list | grep bnxtnet
```

4. Install the Broadcom provided BNXTNETCLI (esxcli bnxtnet) utility to set/view the miscellaneous driver parameters that are not natively supported in esxcli, such as: link speed to 25G, show driver/firmware chip information, show NIC configuration (NPAR, SRIOV). For more information, see the bnxtnet driver README.txt.

   To install this utility:

   a. Copy BCM-ESX-bnxtnetcli-<version>.vib in /var/log/vmware.

```
$ cd /var/log/vmware
$ esxcli software vib install --no-sig-check -v /BCM-ESX-bnxtnetcli-<version>.vib
```

   b. Reboot the system.
   c. Verify whether vib is installed correctly:

```
$ esxcli software vib list | grep bcm-esx-bnxtnetcli
```

   d. Set speed to 10/20/25/40/50G:

```
$ esxcli bnxtnet link set -S <speed> -D <full> -n <iface>
```

   This returns an OK message if the speed is correctly set.

   Example:

```
$ esxcli bnxtnet link set -S 25000 -D full -n vmnic5
```

e. Show the link stats

```
$ esxcli bnxtnet link get -n vmnic6
```

f. Show the driver/firmware/chip information

```
$ esxcli bnxtnet drvinfo get -n vmnic4
```

g. Show the NIC information (for example, BDF; NPAR, SRIOV configuration)

```
$ esxcli bnxtnet nic get -n vmnic4
```

5. Enabling SRIOV VFs:

   Only the PFs are automatically enabled. If a PF supports SR-IOV, the PF(vmknicX) is part of the output of the command shown below.

   ```
   esxcli network sriovnic list
   ```

   To enable one or more VFs, the driver uses the module parameter max_vfs to enable the desired number of VFs for PFs. For example, to enable four VFs on PF1:

   ```
   esxcfg-module -s 'max_vfs=4' bnxtnet (reboot required)
   ```

   To enable VFs on a set of PFs, use the command format shown below. For example, to enable four VFs on PF 0 and 2 VFs on PF 2:

   ```
   esxcfg-module -s 'max_vfs=4,2' bnxtnet (reboot required)
   ```

   The required VFs of each supported PF are enabled in order during the PF bring up. Refer to the VMware documentation for information on how to map a VF to a VM.

NOTE: When using NPAR+SRIOV, every NPAR function (PF) is assigned a maximum of eight VFs.

13 NPAR – Configuration and Use Case Example

13.1 Features and Requirements

- OS/BIOS Agnostic – The partitions are presented to the operating system as real network interfaces so no special BIOS or OS support is required like SR-IOV.
- Additional NICs without requiring additional switch ports, cabling, PCIe expansion slots.
- Traffic Shaping – The allocation of bandwidth per partition can be controlled so as to limit or reserve as needed.
- Can be used in a Switch Independent manner – The switch does not need any special configuration or knowledge of the NPAR enablement.
- Can be used in conjunction with RoCE and SR-IOV.
- Supports stateless offloads such as, LSO, TPA, RSS/TSS, and RoCE (two PFs per port only).
- Alternative Routing-ID support for greater than eight functions per physical device.

NOTE: In the **UEFI HII Menu** page, the NXE adapters support up to 16 PFs per device on an ARI capable system. For a 2 port device, this means up to 8 PFs for each port.

13.2 Limitations

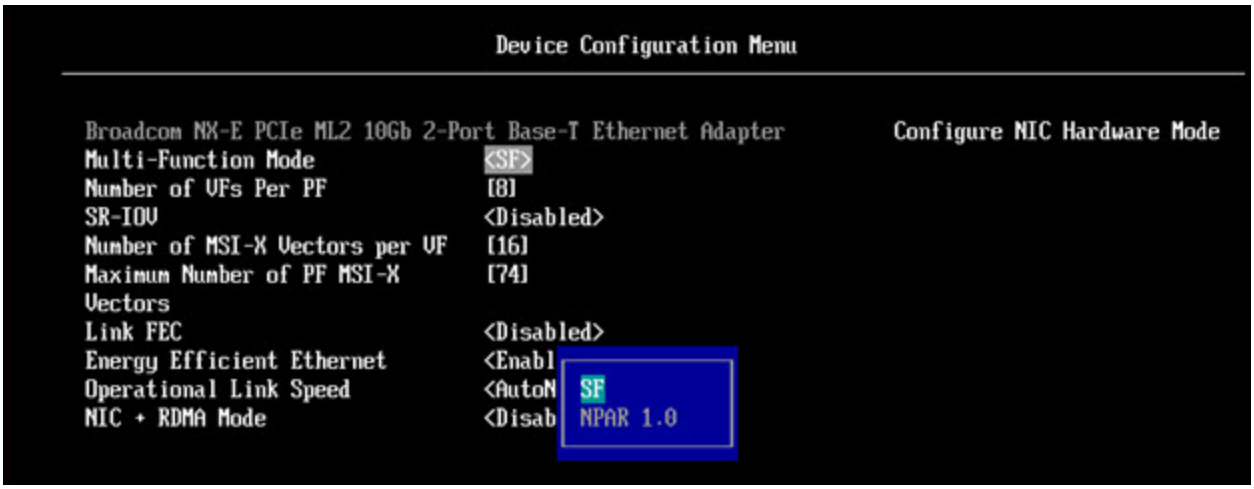
- Shared settings must be suppressed to avoid contention. For example: Speed, Duplex, Flow Control, and similar physical settings are hidden by the device driver to avoid contention.
- Non-ARI systems enable only eight partitions per physical device.
- RoCE is only supported on the first two partitions of each physical port, or a total of four partitions per physical device. In NPAR + SRIOV mode, only two VFs from each parent physical port can enable RDMA support, or total of four VFs + RDMA per physical device.

13.3 Configuration

NPAR can be configured using BIOS configuration HII menus or by using the Broadcom CCM utility on legacy boot systems. Some vendors also expose the configuration via additional proprietary interfaces.

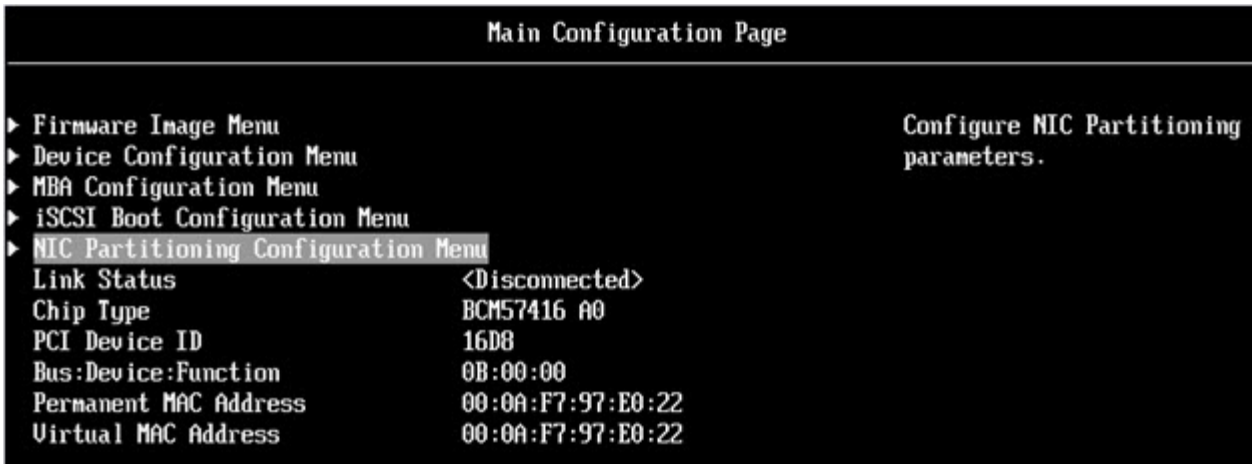
To enable NPAR:

1. Select the target NIC from the BIOS HII Menu or CCM interface and set the Multi-Function Mode or Virtualization Mode option. The choice of options affects the whole NIC instead of the individual port.

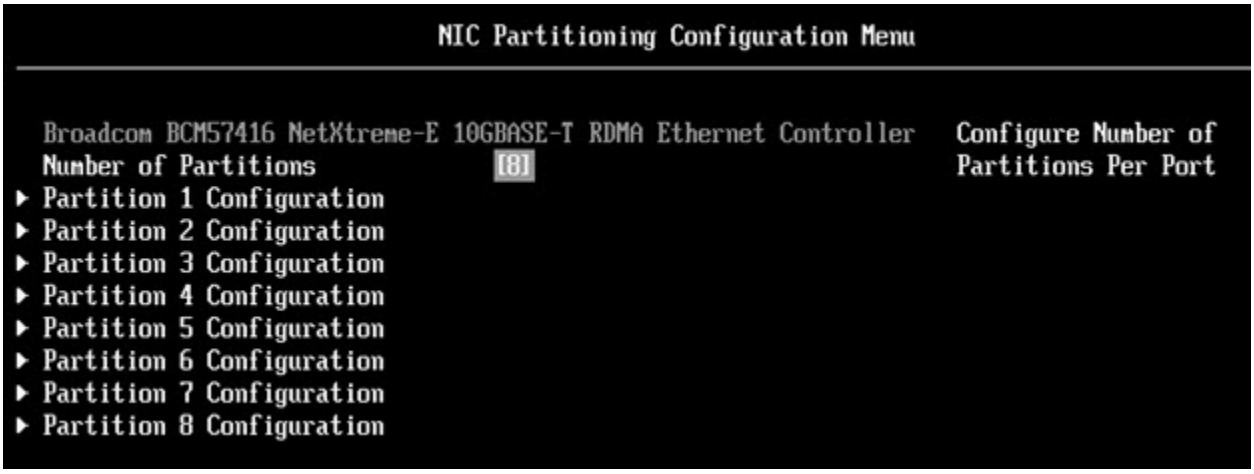


NPAR is enabled in combination with SR-IOV. For some ARI capable OEM systems, the **NParEP** button is available to explicitly allow the BCM5741X to support up to 16 partitions. Switching from Single Function mode to Multifunction mode, the device needs to be re-enumerated, therefore changes do not take effect until a system reboot occurs.

2. Once NPAR is enabled, the NIC Partitioning Main Configuration Menu option is available from the main NIC Configuration Menu associated with each physical port.



3. The NIC Partition Configuration Menu (shown below) allows the user to choose the number of partitions that should be allocated from the selected physical port. Each BCM5741X NIC can support a maximum of 16 partitions on an ARI capable server. By default, dual-port adapters are configured for eight partitions per physical port. Configuration options for each partition are also accessible from this menu. For some OEM systems, the HII menu also includes a Global Bandwidth Allocation page where the minimum (reserved) and maximum (limit) TX Bandwidth for all partitions can be configured simultaneously.



4. Set the NIC Partition Configuration parameters (see Table 24 on page 42).

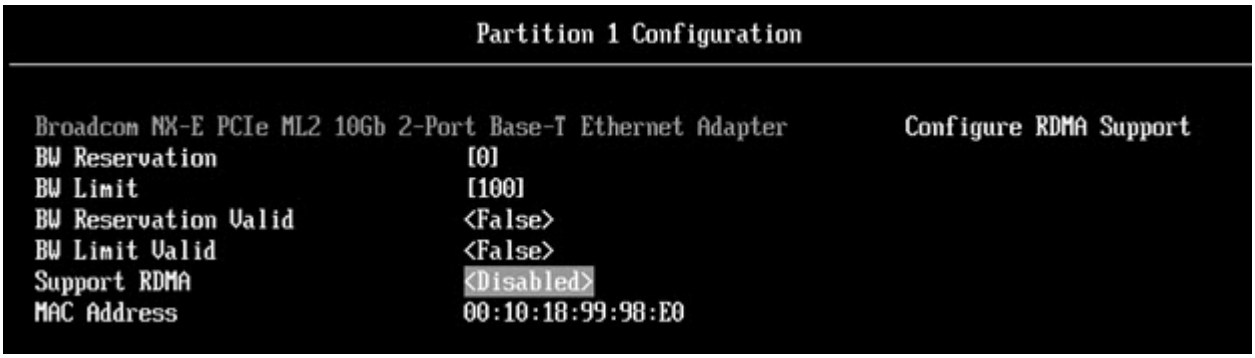


Table 24: NPAR Parameters

Parameter	Description	Valid Options
BW Reservation	Percentage of total available bandwidth that should be reserved for this partition. 0 indicates equal division of bandwidth between all partitions.	Value 0-100
BW Limit	Maximum percentage of available bandwidth this partition is allowed.	Value 0-100
BW Reservation Valid	Functions as an on/off switch for the BW Reservation setting.	True/False
BW Limit Valid	Functions as an on/off switch for the BW Limit setting.	True/False
Support RDMA	Functions as an on/off switch for RDMA support on this partition. **NOTE:** Only two partitions per physical port can support RDMA. For a dual-port device, up to 4 NPAR partitions can support RDMA.	Enabled/Disabled
MAC Address	MAC Address for this partition.	–

13.4 Notes on Reducing NIC Memory Consumption

Because of the faster link speeds supported in this NIC, the default number of receive buffers is larger. More packets can arrive within a given time interval when the link speed is higher, and if the host system is delayed in processing the receive interrupts, the NIC must drop packets if all available receive buffers are in use.

The default value of receive buffers was selected to work well for typical configurations. If you have many NICs in a system, have enabled NPAR on multiple NICs, or if you have only a small amount of RAM, you may see a Code 12 yellow bang in the Device Manager for some of the NICs. Code 12 means that the driver failed to load because there were not enough resources. In this case, the resource is a specific type of kernel memory called Non-Paged Pool (NPP) memory.

If you are getting a Code 12, or for other reasons wish to reduce the amount of NPP memory consumed by the NIC:

- Reduce the number of RSS queues from the default of 8 to 4 or 2. Each RSS queue has its own set of receive buffers allocated, so reducing the number of RSS queues reduces the allocated NPP memory. There can be performance implications from reducing the number of RSS queues, as fewer cores participate in processing receive packets from that NIC. Per processor CPU utilization should be monitored to ensure that there are no “hot” processors after this change.
- Reduce memory allocation by reducing the number of receive buffers allocated. The default value of 0 means the driver should automatically determine the number of receive buffers. For typical configurations, a setting of 0 (=auto) maps to XXXX receive buffers per queue. You can choose a smaller value such as 1500, 1000, or 500. (The value needs to be in multiples of 500 between the range of 500 and 15000.) As mentioned above, a smaller number of receive buffers increases the risk of packet drop and a corresponding impact to packet retransmissions and decreased throughput.

The parameters “Maximum Number of RSS Queues” and “Receive Buffers (0=Auto)” can be modified using the **Advanced** properties tab for each NIC in the **Device Manager**. If you want to modify multiple NICs at the same time, it is faster to use the *Set-NetAdapterAdvancedProperty PowerShell cmdlet*. For example, to assign two RSS queues for all NICs in a system whose NIC name starts with “Sl”, run the following command:

```
Set-NetAdapterAdvancedProperty Sl* -RegistryKeyword *NumRSSQueues -RegistryValue 2
```

Similarly, to set the number of Receive buffers to 1500, run the following command:

```
Set-NetAdapterAdvancedProperty Sl* -RegistryKeyword *ReceiveBuffers -RegistryValue 1500
```

For an overview of how to use PowerShell to modify NIC properties, refer to Microsoft.com.

14 RoCE – Configuration and Use Case Examples

This section provides configuration and use case examples for RoCE.

To enable RoCE for PFs or VFs, the user must enable the RDMA selection in the HII menu in the BIOS before the RDMA option takes effect in the host or guest OS.

To enable RDMA in single function mode (if **Virtualization Mode** is **None** or **SR-IOV**):

During the system boot, access the **System Setup** → **NetXtreme-E NIC** → **Main Configuration Page** and set **NIC+ RMDA Mode** to **Enabled**.

To enable RDMA if Virtualization Mode is NPAR or NPAR+SR-IOV:

During the system boot, access the **System Setup** → **NetXtreme-E NIC** →**NIC Partitioning Configuration** → **Partition 1 (or 2) Configuration** and set **NIC+ RMDA Mode** to **Enabled**.

NOTE: If using NPAR+SRIOV mode, only two VFs from each parent physical port can enable RDMA support, or a total of four VFs+RDMA per physical device.

14.1 Linux Configuration and Use Case Examples

14.1.1 Requirements

To configure RoCE in Linux, the following items are required:

- bnxt_en-roce (RoCE supported bnxt_en driver which is part of released gzip compressed tar archive)
- bnxt_re (RoCE driver)
- libbnxtre (User mode RoCE library module)

14.1.2 BNXT_RE Driver Dependencies

The Bnxt_re driver requires a special RoCE enabled version of bnxt_en which is included in the netxtreme-bnxt_en-1.7.9.tar.gz (or newer) package. The bnxt_re driver compilation depends on whether IB stack is available along with the OS distribution or an external OFED is required.

NOTE: It is necessary to load the correct bnxt_en version that is included in the same netxtreme-bnxt_en-1.7.x.tar.gz package. Bnxt_re and Bnxt_en function as a pair to enable RoCE traffic. Using mismatching versions of these two drivers produces unreliable or unpredictable results.

- Distros that have an IB Stack available along with OS distribution:

  RH7.1/7.2/7.3/6.7/6.8, SLES12SP2 and Ubuntu 16.04

  If not already installed, the IB stack and useful utils can be installed in Redhat with the following commands prior to compiling bnxt_re:

  ```
  yum -y install libibverbs* inifiniband-diag perftest qperf librdmacm utils
  ```

  To compile bnxt_re:

  ```
  $make
  ```

- Distros that need external OFED to be installed:

  SLES11SP4

  Refer to the OFED release notes at the following link and install OFED before compiling bnxt_re driver.

  http://downloads.openfabrics.org/downloads/OFED/release_notes/OFED_3.18-2_release_notes

  To compile bnxt_re:

  ```
  $export OFED_VERSION=OFED-3.18-2
  $make
  ```

14.1.3 Installation

To install RoCE in Linux:

1. Upgrade the NIC NVRAM using the RoCE supported firmware packages from Software Release 20.06.04.01 or newer.
2. In the OS, uncompress, build, and install the BCM5741X Linux L2 and RoCE drivers.
   a. `# tar -xzf netxtreme-bnxt_en-1.7.9.tar.gz`
   b. `# cd netxtreme-bnxt_en-bnxt_re`
   c. `# make build && make install`
3. Uncompress, build, and install the NetXtreme-E Linux RoCE User Library.
   a. `# tar xzf libbnxtre-0.0.18.tar.gz`
   b. `# cd libbnxtre-0.0.18`
   c. `# sh autogen.sh`
   d. `# ./configure && make && make install.`
   e. `# cp bnxtre.driver /etc/libibverbs.d/`
   f. `# echo "/usr/local/lib" >> /etc/ld.so.conf`
   g. `# ldconfig -v`

Refer to the bnxt_re README.txt for more details on configurable options and recommendations.

14.1.4 Limitations

In dual-port NICs, if both ports are on same subnet, RDMA perftest commands may fail. The possible cause is due to an arp flux issue in the Linux OS. To workaround this limitation, use multiple subnets for testing or bring the second port/interface down.

14.1.5 Known Issues

`Bnxt_en` and `Bnxt_re` are designed to function in pair. Older `Bnxt_en` drivers prior to version 1.7.x do not support RDMA and cannot be loaded at the same time as the `Bnxt_re` (RDMA) driver. The user may experience a system crash and reboot if `Bnxt_re` is loaded with older `Bnxt_en` drivers. It is recommend that the user load the `Bnxt_en` and `Bnxt_re` module from the same `netxtreme-bnxt_en-<1.7.x>.tar.gz` bundle.

To prevent mismatching a combination of `bnxt_en` and `bnxt_re` from being loaded, the following is required:

- If RedHat/CentOS 7.2 OS was installed to the target system using PXEboot with `bnxt_en` DUD or a kernel module RPM, delete the file `bnxt_en.ko` found in `/lib/modules/$(uname -r)/extra/bnxt_en/bnxt_en.ko` or edit `/etc/depmod.d/`.
- bnxt_en.conf to override to use updated version. Users can also erase the current BCM5741X Linux kernel driver using the rpm -e kmod-bnxt_en command. RHEL 7.3/SLES 12 Sp2 has bnxt_en inbox driver (older than v1.7.x). This driver must be removed and the latest bnxt_en be added before applying the bnxt_re (RoCE drivers).

14.2 Windows and Use Case Examples

14.2.1 Kernel Mode

Windows Server 2012 and beyond invokes the RDMA capability in the NIC for SMB file traffic if both ends are enabled for RDMA. Broadcom NDIS miniport bnxtnd.sys v20.6.2 and beyond support RoCEv1 and RoCEv2 via the NDKPI interface. The default setting is RoCEv1.

To enable RDMA:

1. Upgrade the NIC NVRAM using the appropriate board packages. In CCM or in UEFI HII, enable support for RDMA.
2. Go to the adapter **Advanced Properties** page and set **NetworkDirect Functionality** to **Enabled** for each BCM5741X miniport, or using PowerShell window, run the following command:

   ```
   Set-NetAdapterAdvancedProperty -RegistryKeyword *NetworkDirect -RegistryValue 1
   ```

3. The following Powershell commands returns true if **NetworkDriect** is enabled.

   ```
   a. Get-NetOffLoadGlobalSetting
   b. Get-NetAdapterRDMA
   ```

14.2.2 Verifying RDMA

To verify RDMA:

1. Create a file share on the remote system and open that share using Windows Explorer. To avoid hard disk read/write speed bottleneck, a RAM disk is recommended as the network share under test.
2. From PowerShell, run the following commands:

   ```
   Get-SmbMultichannelConnection | fl *RDMA*
   ClientRdmaCapable : True
   ServerRdmaCapable : True
   ```

If both Client and Server show True, then any file transfers over this SMB connection use SMB.

3. The following commands can be used to enable/disable SMB Multichannel:

   Server Side:

   – Enable: Set-SmbServerConfiguration -EnableMultiChannel $true
   – Disable: Set-SmbServerConfiguration -EnableMultiChannel $false

   Client Side:

   – Enable: Set-SmbClientConfiguration -EnableMultiChannel $true
   – Disable: Set-SmbClientConfiguration -EnableMultiChannel $false

NOTE: By default, the driver sets up two RDMA connections for each network share per IP address (on a unique subnet). The user can scale up the number of RDMA connections by adding multiple IP addresses, each with different a subnet, for the same physical port under test. Multiple network shares can be created and mapped to each link partner using the unique IP addresses created.

For example:

```
On Server 1, create the following IP addresses for Network Port1.
172.1.10.1
172.2.10.2
172.3.10.3

On the same Server 1, create 3 shares.
Share1
Share2
Share3

On the network link partners,
Connect to \\172.1.10.1\share1
Connect to \\172.2.10.2\share2
Connect to \\172.3.10.3\share3
...etc
```

14.2.3 User Mode

Before running a user mode application written to NDSPI, copy and install the `bxndspi.dll` user mode driver. To copy and install the user mode driver:

1. Copy `bxndspi.dll` to `C:\Windows\System32.`
2. Install the driver by running the following command:

```
rundll32.exe .\bxndspi.dll,Config install|more
```

14.3 VMware ESX Configuration and Use Case Examples

14.3.1 Limitations

The current version of the RoCE supported driver requires ESXi-6.5.0 GA build 4564106 or above.

14.3.2 BNXT RoCE Driver Requirements

The BNXTNET L2 driver must be installed with the `disable_roce=0` module parameter before installing the driver.

To set the module parameter, run the following command:

```
esxcfg-module -s "disable_roce=0" bnxtnet
```

Use the ESX6.5 L2 driver version 20.6.9.0 (RoCE supported L2 driver) or above.

14.3.3 Installation

To install the RoCE driver:

1. Copy the `<bnxtroce>-<driver version>.vib` file in `/var/log/vmware` using the following commands:

   ```
   $ cd /var/log/vmware
   $ esxcli software vib install --no-sig-check -v <bnxtroce>-<driver version>.vib
   ```

2. Reboot the machine.
3. Verify that the drivers are correctly installed using the following command:

   ```
   esxcli software vib list | grep bnxtroce
   ```

4. To disable ECN (enabled by default) for RoCE traffic use the tos_ecn=0 module parameter for bnxtroce.

14.3.4 Configuring Paravirtualized RDMA Network Adapters

Refer to Vmware.com for additional information on setting up and using Paravirtualized RDMA (PVRDMA) network adapters.

14.3.4.1 Configuring a Virtual Center for PVRDMA

To configure a Virtual Center for PVRDMA:

1. Create DVS (requires a Distributed Virtual Switch for PVRDMA)
2. Add the host to the DVS.

14.3.4.2 Tagging vmknic for PVRDMA on ESX Hosts

To tag a vmknic for PVRDMA to use on ESX hosts:

1. Select the host and right-click on **Settings** to switch to the settings page of the **Manage** tabs.
2. In the **Settings** page, expand **System** and click **Advanced System Settings** to show the Advanced System Settings key-pair value and its summary.
3. Click **Edit** to bring up the **Edit Advanced System Settings**.
   Filter on **PVRDMA** to narrow all the settings to just `Net.PVRDMAVmknic`.
4. Set the `Net.PVRDMAVmknic` value to vmknic, as in example vmk0

14.3.4.3 Setting the Firewall Rule for PVRDMA

To set the firewall rule for PVRDMA:

1. Select the host and right-click on **Settings** to switch to the settings page of the **Manage** tabs.
2. In the **Settings** page, expand **System** and click **Security Profile** to show the firewall summary.
3. Click **Edit** to bring up the **Edit Security Profile**.
4. Scroll down to find pvrdma and check the box to set the firewall.

14.3.4.4 Adding a PVRDMA Device to the VM

To add a PVRDMA device to the VM:

1. Select the VM and right-click on **Edit Settings**.
2. Add a new Network Adapter.
3. Select the network as a **Distributed Virtual Switch** and **Port Group**.
4. For the **Adapter Type**, select **PVRDMA** and click **OK**.

14.3.4.5 Configuring the VM on Linux Guest OS

NOTE: The user must install the appropriate development tools including git before proceeding with the configuration steps below.

1. Download the PVRDMA driver and library using the following commands:

```
git clone git://git.openfabrics.org/~aditr/pvrdma_driver.git
git clone git://git.openfabrics.org/~aditr/libpvrdma.git
```

2. Compile and install the PVRDMA guest driver and library.
3. To install the driver, execute `make && sudo insmod pvrdma.ko` in the directory of the driver.
   The driver must be loaded after the paired vmxnet3 driver is loaded.

NOTE: The installed RDMA kernel modules may not be compatible with the PVRDMA driver. If so, remove the current installation and restart. Then follow the installation instructions. Refer to the README in the driver's directory for more information about the different RDMA stacks.

4. To install the library, execute `./autogen.sh && ./configure --sysconfdir=/etc && make && sudo make install` in the directory of the library.

NOTE: The installation path of the library needs to be in the shared library cache. Follow the instructions in the INSTALL file in the library's directory.

NOTE: The firewall settings may need to be modified to allow RDMA traffic. Ensure the proper firewall settings are in place.

5. Add the `/usr/lib` in the `/etc/ld.so.conf` file and reload the `ldconf` by running `ldconfig`
6. Load ib modules using `modprobe rdma_ucm`.
7. Load the PVRDMA kernel module using `insmod pvrdma.ko`.
8. Assign an IP address to the PVRDMA interface.
9. Verify whether the IB device is created by running the `ibv_devinfo -v` command.

15 DCBX – Data Center Bridging

Broadcom NetXtreme-E controllers support IEEE 802.1Qaz DCBX as well as the older CEE DCBX specification. DCB configuration is obtained by exchanging the locally configured settings with the link peer. Since the two ends of a link may be configured differently, DCBX uses a concept of 'willing' to indicate which end of the link is ready to accept parameters from the other end. This is indicated in the DCBX protocol using a single bit in the ETS Configuration and PFC TLV, this bit is not used with ETS Recommendation and Application Priority TLV. By Default, the NetXtreme-E NIC is in 'willing' mode while the link partner network switch is in 'non-willing' mode. This ensures the same DCBX setting on the Switch propagates to the entire network.

Users can manually set NetXtreme-E NIC to non-willing mode and perform various PFC, Strict Priority, ETS, and APP configurations from the host side. Refer to the driver readme.txt for more details on available configurations. This section provides an example of how such a setting can be done in Windows with Windows PowerShell. Additional information on DCBX, QoS, and associated use cases are described in more details in a separate white paper, beyond the scope of this user manual.

The following settings in the UEFI HII menu are required to enable DCBX support:

System Setup→Device Settings→NetXtreme-E NIC→Device Level Configuration

15.1 QoS Profile – Default QoS Queue Profile

Quality of Server (QoS) resources configuration is necessary to support various PFC and ETS requirements where finer tunning beyond bandwidth allocation is needed. The NetXtreme-E allows the administrator to select between devoting NIC hardware resources to support Jumbo Frames and/or combinations of lossy and lossless Class of Service queues (CoS queues). Many combinations of configuration are possible and therefore can be complicated to compute. This option allows a user to select from a list of precomputed QoS Queue Profiles. These precomputed profiles are design to optimize support for PFC and ETS requirements in typical customer deployments.

The following is a summary description for each QoS Profile.

Table 25: QoS Profiles

Profile No.	Jumbo Frame Support	No. of Lossy CoS Queues/Port	No. of Lossless CoS Queues/Port	Support for 2-Port SKU
Profile #1	Yes	0	1 (PFC Supported)	Yes (25 Gb/s)
Profile #2	Yes	4	2 (PFC Supported)	No
Profile #3	No. (MTU <= 2 KB)	6	2 (PFC Supported)	Yes (25 Gb/s)
Profile #4	Yes	1	2 (PFC Supported)	Yes (25 Gb/s)
Profile #5	Yes	1	0 (No PFC Support)	Yes (25 Gb/s)
Profile #6	Yes	8	0 (No PFC Support)	Yes (25 Gb/s)
Profile #7	This configuration maximizes packet-buffer allocations to two lossless CoS Queues to maximize RoCE performance while trading off flexibility.			
	Yes	0	2	Yes (25 Gb/s)
Default	Yes	Same as Profile #4		Yes

15.2 DCBX Mode – Enable (IEEE only)

This option allows a user to enable/disable DCBX with the indicated specification. IEEE only indicates that IEEE 802.1Qaz DCBX is selected.

Windows Driver setting:

After enabling the indicated options in the UEFI HII menu to set firmware level settings, perform the follow selection in the Windows driver advanced properties.

Open **Windows Manager → Broadcom NetXtreme E Series adapter → Advanced Properties → Advanced tab**

Quality of Service = Enabled

Priority & VLAN = Priority& VLAN enabled

VLAN = <ID>

Set desired VLAN id

To exercise the DCB related command in Windows PowerShell, install the appropriate DCB Windows feature.

1. In the **Task Bar**, right-click the Windows PowerShell icon and then click **Run as Administrator**. Windows PowerShell opens in elevated mode.
2. In the Windows PowerShell console, type:

```
Install-WindowsFeature "data-center-bridging"
```

15.3 DCBX Willing Bit

The DCBX willing bit is specified in the DCB specification. If the Willing bit on a device is true, the device is willing to accept configurations from a remote device through DCBX. If the Willing bit on a device is false, the device rejects any configuration from a remote device and enforces only the local configurations.

Use the following to set the willing bit to True or False. 1 for enabled, 0 for disabled.

Example `set-netQoSdcbxSetting -Willing 1`

Use the following to create a Traffic Class.

```
C:\> New-NetQosTrafficClass -name "SMB class" -priority 4 -bandwidthPercentage 30 -Algorithm ETS
```

Note: By default, all IEEE 802.1p values are mapped to a default traffic class, which has 100% of the bandwidth of the physical link. The command shown above creates a new traffic class to which any packet tagged with eight IEEE 802.1p value 4 is mapped, and its Transmission Selection Algorithm (TSA) is ETS and has 30% of the bandwidth. It is possible to create up to seven new traffic classes. In addition to the default traffic class, there is at most eight traffic classes in the system.

Use the following in displaying the created Traffic Class:

```
C:\> Get-NetQoSTrafficClass
Name           Algorithm  Bandwidth(%)   Priority
-------------------------
[Default]          ETS                70         0-3,5-7
SMB class         ETS              30             4
```

Use the following in modifying the Traffic Class:

```
PS C:\> Set-NetQoSTrafficClass -Name "SMB class" -BandwidthPercentage 40
PS C:\> get-NetQosTrafficClass
Name Algorithm Bandwidth(%) Priority
-------------------------------------------------------------------
[Default] ETS            60 0-3,5-7
SMB class ETS          40       4
```

Use the following to remove the Traffic Class:

```
PS C:\> Remove-NetQosTrafficClass -Name "SMB class"
PS C:\> Get-NetQosTrafficClass
Name Algorithm Bandwidth(%) Priority
-------------------------------------------------------------------
[Default] ETS          100              0-7
```

Use the following to create Traffic Class (Strict Priority):

```
C:\> New-NetQosTrafficClass -name "SMB class" -priority 4 -bandwidthPercentage 30-Algorithm Strict
```

Enabling PFC:

```
PS C:\> Enable-NetQosFlowControl -priority 4
PS C:\> Get-NetQosFlowControl -priority 4
Priority Enabled
-------------------------------------------------------------------
4 True

PS C:\> Get-NetQosFlowControl
```

Disabling PFC:

```
PS C:\> disable-NetQosflowControl -priority 4
PS C:\> get-NetQosFlowControl -priority 4
Priority Enabled
-------------------------------------------------------------------
4 False
```

Use the following to create QoS Policy:

```
PS C:\> New-NetQosPolicy -Name "SMB policy" -SMB -PriorityValue8021Action 4

Name : SMB policy
Owner : Group Policy (Machine)
NetworkProfile : All
Precedence : 127
```

NOTE: The previous command creates a new policy for SMB. SMB is an inbox filter that matches TCP port 445 (reserved for SMB). If a packet is sent to TCP port 445 it is tagged by the operating system with IEEE 802.1p value of 4 before the packet is passed to a network miniport driver. In addition to SMB, other default filters include iSCSI (matching TCP port 3260), NFS (matching TCP port 2049), LiveMigration (matching TCP port 6600), FCOE (matching EtherType 0x8906) and NetworkDirect. NetworkDirect is an abstract layer created on top of any RDMA implementation on a network adapter. NetworkDirect must be followed by a Network Direct port. In addition to the default filters, a user can classify traffic by application's executable name (as in the first example below), or by IP address, port, or protocol.

Use the following to create QoS Policy based on the Source/Destination Address:

```
PS C:\> New-NetQosPolicy "Network Management" -IPDstPrefixMatchCondition 10.240.1.0/24 -
IPProtocolMatchCondition both -NetworkProfile all -PriorityValue8021Action 7
Name : Network Management
Owner : Group Policy (Machine)
Network Profile : All
Precedence : 127
IPProtocol : Both
IPDstPrefix : 10.240.1.0/24
PriorityValue : 7
```

Use the following to display QoS Policy:

```
PS C:\> Get-NetQosPolicy
Name : Network Management
Owner : (382ACFAD-1E73-46BD-A0A-6-4EE0E587B95)
NetworkProfile : All
Precedence : 127
IPProtocol : Both
IPDstPrefix : 10.240.1.0/24
PriorityValue : 7
Name : SMB policy
Owner : (382AFAD-1E73-46BD-A0A-6-4EE0E587B95)
NetworkProfile : All
Precedence : 127
Template : SMB
PriorityValue : 4
```

Use the following to modify the QoS Policy:

```
PS C:\> Set-NetqosPolicy -Name "Network Management" -IPSrcPrefixMatchCondition 10.235.2.0/24 -
IPProtocolMatchCondition both -PriorityValue802.1Action 7
PS C:\> Get-NetQosPolicy -name "network management"
Name : Network Management
Owner : {382ACFD-1E73-46BD-A0A0-4EE0E587B95}
NetworkProfile : All
Precedence : 127
IPProtocol : Both
IPSrcPrefix : 10.235.2.0/24
IPDstPrefix : 10.240.1.0/24
PriorityValue : 7
```

Use the following to remove QoS Policy:

```
PS C:\> Remove-NetQosPolicy -Name "Network Management"
```

16 DPDK – Configuration and Use Case Examples

The testpmd application can be used to test the DPDK in a packet forwarding mode and also to access NIC hardware features such as Flow Director. It also serves as an example of how to build a more fully-featured application using the DPDK SDK. The below chapters shows how to build and run the testpmd application and how to configure the application from the command line and the run-time environment.

16.1 Compiling the Application

To compile the application:

1. Set the required environmental variables and go to the source code with the following commands:

```
cd /Linux/Linux_DPDK
tar zxvf dpdk-*.gz
cd dpdk-*
make config T=x86_64-native-linuxapp-gcc && make
```

2. Allocate system resources and attach the UIO module with the following commands:

```
mkdir -p /mnt/huge
mount -t hugetlbfs nodev /mnt/huge
echo 2048 > /sys/devices/system/node/node0/hugepages/hugepages-2048kB/nr_hugepages
echo 2048 > /sys/devices/system/node/node1/hugepages/hugepages-2048kB/nr_hugepages
modprobe uio
insmod  ./build/kmod/igb_uio.ko
```

3. Bind the device with the following command:

```
usertools/dpdk-devbind.py -b igb_uio 3b:00.0
```

4. The PCI device information is displayed by lspci with the following commands:

```
[root@localhost /]# lspci |grep Eth
3b:00.0 Ethernet controller: Broadcom Limited BCM57414 NetXtreme-E 10Gb/25Gb RDMA Ethernet
Controller (rev 01)
3b:00.1 Ethernet controller: Broadcom Limited BCM57414 NetXtreme-E 10Gb/25Gb RDMA Ethernet
Controller (rev 01)
```

16.2 Running the Application

To run the application, execute the following command:

```
build/app/testpmd -l 0,1,2,3,4,5,6,7,8 -- -i --nb-ports=1 --nb-cores=8 --txq=2 --rxq=2
```

Options for this example:

- -l – CORELIST List of cores to run.
- -i – Interactive Run testpmd in interactive mode.
- --nb-cores=N – Sets the number of forwarding cores.
- --nb-ports=N – Sets the number of forwarding ports.
- --rxq=N – Sets the number of RX queues per port to N.
- --txq=N – Sets the number of TX queues per port to N.

16.3 Testpmd Runtime Functions

When the testpmd application is started in interactive mode, (-i|--interactive), it displays a prompt that can be used to start and stop forwarding, configure the application, display statistics (including the extended NIC statistics aka xstats), set the Flow Director, and other tasks.

```
testpmd>
```

16.4 Control Functions

This section contains the control functions.

start

Starts packet forwarding with current configuration:

```
testpmd> start
```

start tx_first

Starts packet forwarding with current configuration after sending specified number of bursts of packets:

```
testpmd> start tx_first ("|burst_num)
```

The default burst number is 1 when burst_num not presented.

stop

Stops packet forwarding and display accumulated statistics:

```
testpmd> stop
```

quit

Quits to prompt:

```
testpmd> quit
```

16.5 Display Functions

The functions in ths section are used to display information about the testpmd configuration or the NIC status.

- show port – Displays information for a given port or all ports.
- show port rss reta – Displays the RSS redirection table entry indicated by masks on port X.
- show port rss-hash – Displays the RSS hash functions and RSS hash key of a port.
- show (rxq|txq) – Displays information for a given port's RX/TX queue.
- show config – Displays the configuration of the application. The configuration comes from the command-line.
- set fwd – Sets the packet forwarding mode.
- read rxd – Displays an RX descriptor for a port RX queue.
- read txd – Displays a TX descriptor for a port TX queue.

16.6 Configuration Functions

The testpmd application can be configured from the runtime as well as from the command line. This section describes the available configuration functions that are available.

csum set – Selects the hardware or software calculation of the checksum when transmitting a packet using the csum forwarding engine: testpmd> csum set (ip|udp|tcp|sctp|outer-ip) (hw|sw) (port_id).

17 Frequently Asked Questions

- Does the device support AutoNeg at 25G speed?

  Yes. Refer to “Auto-Negotiation Configuration” on page 24 for more details.
- How would I connect SFP28 cable to QSFP ports?

  Breakout cables are available from QSFP to 4xSFP28 ports.
- What are the compatible port speeds?

  For BCM57404AXXXX/BCM57414 dual-port devices, the port speed of each port must be compatible with the port speed of the other port. 10 Gb/s and 25 Gb/s are not compatible speed. If one port is set to 10 Gb/s the other port can not be set to 25 Gb/s. If a user attempts to set incompatible port speeds, the second port to be brought up does not link. Refer to “Auto-Negotiation Configuration” on page 24 for more details.
- Can I use 10 Gb/s for PXE connectivity on a 25 Gb/s port?

  Currently only 25 Gb/s PXE speed is supported. It is not recommended to use 10 Gb/s PXE connectivity on a 25 Gb/s adapter. This is due to the lack of support for auto negotiation on existing 10 Gb/s switches and possible complications caused by incompatible port link speed settings.

Revision History

NetXtreme-UG101; April 2, 2019

Updated:

- Updated links.

NetXtreme-UG100; August 23, 2018

Initial Release.


BROADCOM®