SOFTWARE MANUAL

Mellanox WinOF-2 User Manual

Rev 1.60

Software v.1.60.16149

NOTE:
THIS HARDWARE, SOFTWARE OR TEST SUITE PRODUCT ("PRODUCT(S)") AND ITS RELATED DOCUMENTATION ARE PROVIDED BY MELLANOX TECHNOLOGIES "AS-IS" WITH ALL FAULTS OF ANY KIND AND SOLELY FOR THE PURPOSE OF AIDING THE CUSTOMER IN TESTING APPLICATIONS THAT USE THE PRODUCTS IN DESIGNATED SOLUTIONS. THE CUSTOMER'S MANUFACTURING TEST ENVIRONMENT HAS NOT MET THE STANDARDS SET BY MELLANOX TECHNOLOGIES TO FULLY QUALIFY THE PRODUCT(S) AND/OR THE SYSTEM USING IT. THEREFORE, MELLANOX TECHNOLOGIES CANNOT AND DOES NOT GUARANTEE OR WARRANT THAT THE PRODUCTS WILL OPERATE WITH THE HIGHEST QUALITY. ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT ARE DISCLAIMED. IN NO EVENT SHALL MELLANOX BE LIABLE TO CUSTOMER OR ANY THIRD PARTIES FOR ANY DIRECT, INDIRECT, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES OF ANY KIND (INCLUDING, BUT NOT LIMITED TO, PAYMENT FOR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY FROM THE USE OF THE PRODUCT(S) AND RELATED DOCUMENTATION EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Mellanox Technologies
350 Oakmead Parkway Suite
100
Sunnyvale, CA 94085
U.S.A.
www.mellanox.com
Tel: (408) 970-3400
Fax: (408) 970-3403

© Copyright 2017. Mellanox Technologies Ltd. All Rights Reserved.

Mellanox®, Mellanox logo, Accelio®, BridgeX®, CloudX logo, CompustorX®, Connect-IB®, ConnectX®, CoolBox®, CORE-Direct®, EZchip®, EZchip logo, EZappliance®, EZdesign®, EZdriver®, EZsystem®, GPUDirect®, InfiniHost®, InfiniBridge®, InfiniScale®, Kotura®, Kotura logo, Mellanox CloudRack®, Mellanox CloudXMellanox®, Mellanox Federal Systems®, Mellanox HostDirect®, Mellanox Multi-Host®, Mellanox Open Ethernet®, Mellanox OpenCloud®, Mellanox OpenCloud Logo®, Mellanox PeerDirect®, Mellanox ScalableHPC®, Mellanox StorageX®, Mellanox TuneX®, Mellanox Connect Accelerate Outperform logo, Mellanox Virtual Modular Switch®, MetroDX®, MetroX®, MLNX-OS®, NP-1c®, NP-2®, NP-3®, Open Ethernet logo, PhyX®, PlatformX®, PSIPHY®, SiPhy®, StoreX®, SwitchX®, Tilera®, Tilera logo, TestX®, TuneX®, The Generation of Open Ethernet logo, UFM®, Unbreakable Link®, Virtual Protocol Interconnect®, Voltaire® and Voltaire logo are registered trademarks of Mellanox Technologies, Ltd.

All other trademarks are property of their respective owners.

For the most updated list of Mellanox trademarks, visit http://www.mellanox.com/page/trademarks

Table of Contents

Document Revision History 8

About this Manual 14

Scope 14
Intended Audience 14
Documentation Conventions 14
Common Abbreviations and Acronyms 15
Related Documents 16

Chapter 1 Introduction 17

1.1 Supplied Packages 18
1.2 Windows MPI (MS-MPI) 18
1.3 Windows Nano Server 2016 18

Chapter 2 Installation 20

2.1 Hardware and Software Requirements 20
2.2 Downloading Mellanox WinOF-2 Driver 20
2.3 Installing Mellanox WinOF-2 Driver 21
2.3.1 Attended Installation 21
2.3.2 Unattended Installation 28
2.4 Installation Results 28
2.5 Extracting Files Without Running Installation 29
2.6 Uninstalling Mellanox WinOF-2 Driver 31
2.6.1 Attended Uninstallation 31
2.6.2 Unattended Uninstallation 32
2.7 Firmware Upgrade 32
2.8 Deploying the Driver on a Nano Server 32
2.8.1 Offline Installation 32
2.8.2 Online Update 33

Chapter 3 Features Overview and Configuration 34

3.1 Ethernet Network 34
3.1.1 Packet Burst Handling 34
3.1.2 Mode Configuration 34
3.1.3 Assigning Port IP After Installation 37
3.1.4 RDMA over Converged Ethernet (RoCE) 40
3.1.5 RoCEv2 Congestion Management (RCM) 46
3.1.6 Teaming and VLAN 55
3.1.7 Configuring Quality of Service (QoS) 59

3.1.8 Differentiated Services Code Point (DSCP) 63
3.1.9 Configuring the Ethernet Driver. 67
3.1.10 Receive Segment Coalescing (RSC) 67
3.1.11 Receive Side Scaling (RSS) 68
3.1.12 Wake on LAN (WoL) 68
3.1.13 Data Center Bridging Exchange (DCBX) 68
3.1.14 Receive Path Activity Monitoring 71
3.1.15 Head of Queue Lifetime Limit 71
3.2 InfiniBand Network. 71
3.2.1 Feature Limitations 71
3.2.2 Port Configuration. 72
3.2.3 Modifying IPoIB Configuration 72
3.2.4 Displaying Adapter Related Information. 72
3.2.5 Assigning Port IP After Installation 74
3.2.6 Receive Side Scaling (RSS) 74
3.3 Storage Protocols 75
3.3.1 Deploying SMB Direct 75
3.4 Virtualization. 76
3.4.1 Hyper-V with VMQ 76
3.4.2 Network Virtualization using Generic Routing Encapsulation (NVGRE) . . 77
3.4.3 Single Root I/O Virtualization (SR-IOV) 80
3.4.4 Virtual Machine Multiple Queue (VMMQ) 100
3.4.5 Network Direct Kernel Provider Interface. 102
3.4.6 PacketDirect Provider Interface. 105
3.5 Configuration Using Registry Keys 108
3.5.1 Finding the Index Value of the Network Interface 108
3.5.2 Basic Registry Keys 110
3.5.3 Offload Registry Keys 110
3.5.4 Performance Registry Keys. 113
3.5.5 Ethernet Registry Keys 117
3.6 Performance Tuning and Counters 120
3.6.1 General Performance Optimization and Tuning. 120
3.6.2 Application Specific Optimization and Tuning 121
3.6.3 Tunable Performance Parameters. 123
3.6.4 Adapter Proprietary Performance Counters. 125
3.7 Network Direct Interface 138
3.7.1 Test Running 139
Chapter 4 Utilities 141
4.1 Fabric Performance Utilities 141
4.1.1 Win-Linux nd_rping Test. 142
4.2 Management Utilities. 142

4.2.1 mlx5cmd Utilities 142
4.3 Snapshot Utility 145
4.3.1 Snapshot Usage 145
Chapter 5 Troubleshooting 148
5.1 Installation Related Troubleshooting 148
5.1.1 Installation Error Codes and Troubleshooting 148
5.2 InfiniBand Related Troubleshooting 150
5.3 Ethernet Related Troubleshooting 150
5.4 Performance Related Troubleshooting 152
5.4.1 General Diagnostic 152
5.5 Virtualization Related Troubleshooting 153
5.6 Reported Driver Events 154
5.7 State Dumping 159
5.8 Extracting WPP Traces 161
Appendix A NVGRE Configuration Scripts Examples 162
A.1 Adding NVGRE Configuration to Host 14 Example 162
A.2 Adding NVGRE Configuration to Host 15 Example 164
Appendix B Windows MPI (MS-MPI) 166
B.1 Overview 166
B.2 System Requirements 166
B.3 Running MPI 166
B.4 Directing MSMPI Traffic 166
B.5 Running MSMPI on the Desired Priority 167
B.6 Configuring MPI 167
B.7 PFC Example 167
B.8 Running MPI Command Examples 168

List of Tables

Table 1: Document Revision History 8
Table 2: Documentation Conventions 14
Table 3: Abbreviations and Acronyms 15
Table 4: Related Documents 16
Table 5: Hardware and Software Requirements 20
Table 6: Registry Key Parameters 46
Table 7: RCM Parameters 50
Table 8: Default Priority Parameters 52
Table 9: DSCP Registry Keys Settings 65
Table 10: DSCP Default Registry Keys Settings 66
Table 11: Registry Keys Setting 68
Table 12: VF Spoof Protection Registry Keys 95
Table 13: Basic Registry Keys 110
Table 14: Offload Registry Keys 111
Table 15: Performance Registry Keys 113
Table 16: Ethernet Registry Keys 117
Table 17: Flow Control Options 118
Table 18: VMQ Options 119
Table 19: RoCE Options 119
Table 20: SR-IOV Options 120
Table 21: Mellanox WinOF-2 Port Traffic Counters 125
Table 22: Mellanox WinOF-2 Virtual Function (VF) Port Traffic Counters 127
Table 23: Mellanox WinOF-2 Port QoS Counters 129
Table 24: RDMA Activity Counters 131
Table 25: Congestion Control Counters 131
Table 26: WinOF-2 Diagnostics Counters 132
Table 27: Device Diagnostics Counters 133
Table 28: PCI Device Diagnostic Counters 135
Table 29: RSS Diagnostic Counters 137
Table 30: Fabric Performance Utilities 141
Table 31: Installation Related Issues 148
Table 32: Setup Return Codes 148
Table 33: Firmware Burning Warning Codes 148

Table 34: Restore Configuration Warnings 149
Table 35: InfiniBand Related Issues 150
Table 36: Ethernet Related Issues 150
Table 37: Performance Related Issues 152
Table 38: Virtualization Related Issues 153
Table 39: Events Causing Automatic State Dumps 159

Document Revision History

Table 1 - Document Revision History

Document Revision	Date	Changes
Rev 1.60	2017	Added the following sections: • Section 3.1.1, “Packet Burst Handling”, on page 34 • Section 3.1.8, “Differentiated Services Code Point (DSCP)”, on page 63 and its subsections. • Section 3.1.15, “Head of Queue Lifetime Limit”, on page 71 • Section 3.4.3.2.5, “Configuring Host Memory Limit per VF”, on page 88. • Section 3.4.3.5, “VF Spoof Protection”, on page 94 and its subsections. • Section 3.6.4.1.9, “Mellanox WinOF-2 Hardware RSS Diagnostic Counters”, on page 136 and its subsections. • Section 4.2.1.7, “Non-RSS Traffic Capture Utility”, on page 144 Updated the following sections: • Table 4, “Related Documents,” on page 16 • Section 1.1, “Supplied Packages”, on page 18 • Section 2.3.2, “Unattended Installation”, on page 28 • Section 2.4, “Installation Results”, on page 28 • Section 3.1.2, “Mode Configuration”, on page 34 • Section 3.1.8.1, “System Requirements”, on page 63 • Section 3.1.8.2, “Setting the DSCP in the IP Header”, on page 63 • Section 3.1.8.7, “Registry Settings”, on page 65 • Section 3.1.8.8, “DSCP Sanity Testing”, on page 67 • Section 3.4.3, “Single Root I/O Virtualization (SR-IOV)”, on page 80

Table 1 - Document Revision History

Document Revision	Date	Changes
Rev 1.60	2017	• Section 3.4.3.5.1, “Limitations”, on page 96 • Section 3.4.5.3.2, “Validating NDK”, on page 105 • Section 3.5.1, “Finding the Index Value of the Network Interface”, on page 108 • Section 3.5.5.4, “SR-IOV Options”, on page 119 • Section 3.6.2.2, “Ethernet Bandwidth Improvements”, on page 121 • Section 3.6.4.1.5, “Mellanox WinOF-2 Congestion Control Counters”, on page 131 • Section 3.6.4.1.6, “Mellanox WinOF-2 Diagnostics Counters”, on page 132 • Section 3.6.4.1.7, “Mellanox WinOF-2 Device Diagnostic Counters”, on page 133 • Section 3.6.4.1.8, “Mellanox WinOF-2 PCI Device Diagnostic Counters”, on page 135 • Section 4.1.1, “Win-Linux nd_rping Test”, on page 142 • Table 16, “Ethernet Registry Keys,” on page 117 • Section 5.2, “InfiniBand Related Troubleshooting”, on page 150 • Section 5.7, “State Dumping”, on page 159 • Section A, “NVGRE Configuration Scripts Examples”, on page 162 • Section B, “Windows MPI (MS-MPI)”, on page 166 Relocated the following section: • Section 3.7, “Network Direct Interface”, on page 138 (formerly 3.5.6) and its subsection.

Table 1 - Document Revision History

Document Revision	Date	Changes
Rev 1.50	November 2016	Added the following sections: • Section 3.1.14, “Receive Path Activity Monitoring”, on page 71 • Section 3.2, “InfiniBand Network”, on page 71 • Section 3.4.6.3, “Disable Loopback Mode”, on page 108 • Section 3.5.5.4, “SR-IOV Options”, on page 119 • Section 3.6.4.1.5, “Mellanox WinOF-2 Congestion Control Counters”, on page 131 • Section 3.6.4.1.7, “Mellanox WinOF-2 Device Diagnostic Counters”, on page 133 • Section 3.6.2.2, “Ethernet Bandwidth Improvements”, on page 121 Updated the following sections: • Section , “About this Manual”, on page 14 • Section 1, “Introduction”, on page 17 • Section 1.1, “Supplied Packages”, on page 18 • Section 2.1, “Hardware and Software Requirements”, on page 20 • Section 2.8.2, “Online Update”, on page 33 • Section 3.1.2, “Mode Configuration”, on page 34 • Section 3.1.3.1, “Configuring 56GbE Link Speed”, on page 39 • Section 3.1.5, “RoCEv2 Congestion Management (RCM)”, on page 46

Table 1 - Document Revision History

Document Revision	Date	Changes
		• Section 3.1.5.1, “Restrictions and Limitations”, on page 48 • Section 3.1.5.3, “RCM Parameters”, on page 50 • Section 3.1.5.3.1, “RCM Default Parameters”, on page 51 • Section 3.1.5.4.1, “CNP Priority”, on page 52 • Section 3.1.5.4.2, “alpha -”α” = Rate Reduction Factor”, on page 52 • Section 3.1.5.4.3, “Decrease (on the “RP”)”, on page 53 • Section 3.1.5.5, “Mellanox Commands and Examples”, on page 54 • Section 3.1.10, “Receive Segment Coalescing (RSC)”, on page 67 • Section 3.2.1, “Feature Limitations”, on page 71 • Section 3.3.1.2, “Verifying SMB Events that Confirm RDMA Connection”, on page 76 • Section 3.4.2, “Network Virtualization using Generic Routing Encapsulation (NVGRE)”, on page 77 • Section 3.4.4, “Virtual Machine Multiple Queue (VMMQ)”, on page 100 • Section 3.4.4.1, “System Requirements”, on page 100 • Section 3.4.4.2.1, “On the Driver Level”, on page 101 • Section 3.4.6.3, “Disable Loopback Mode”, on page 108 • Section 3.5.3, “Offload Registry Keys”, on page 110 • Section 3.5.5, “Ethernet Registry Keys”, on page 117 • Section 3.6.4.1.1, “Proprietary Mellanox WinOF-2 Port Traffic Counters”, on page 125 • Section 3.6.4.1.5, “Mellanox WinOF-2 Congestion Control Counters”, on page 131 • Section 3.6.4.1.6, “Mellanox WinOF-2 Diagnostics Counters”, on page 132 • Table 28, “PCI Device Diagnostic Counters,” on page 135 • Table 30, “Fabric Performance Utilities,” on page 141 • Section 4.1.1, “Win-Linux nd_rping Test”, on page 142 • Section 4.2.1.4, “QoS Configuration Utility”, on page 143 • Section 5.5, “Virtualization Related Troubleshooting”, on page 153

Table 1 - Document Revision History

Document Revision	Date	Changes
Rev 1.45	September 2016	Added the following sections: • Section 4.2.1.9, “Link Speed Utility”, on page 144 • Section 2.8, “Deploying the Driver on a Nano Server”, on page 32 • Section 2.8.1, “Offline Installation”, on page 32 • Section 2.8.2, “Online Update”, on page 33 • Section 3.4.4, “Virtual Machine Multiple Queue (VMMQ)”, on page 100 • Section 3.4.5, “Network Direct Kernel Provider Interface”, on page 102 • Section 3.4.6, “PacketDirect Provider Interface”, on page 105
Rev 1.40	May 2016	Added the following sections: • Section 4.2.1.6, “Registry Keys Utility”, on page 144 • Section 4.2.1.7, “Non-RSS Traffic Capture Utility”, on page 144 • Section 4.2.1, “mlx5cmd Utilities”, on page 142 • Section 3.1.13, “Data Center Bridging Exchange (DCBX)”, on page 68 • Section 3.1.10, “Receive Segment Coalescing (RSC)”, on page 67 • Section 4.1.1, “Win-Linux nd_rping Test”, on page 142 • Section 5.8, “Extracting WPP Traces”, on page 161 Updated the following sections: • Section 3.1.5.3, “RCM Parameters”, on page 50
Rev 1.35	January 2016	Added the following sections: • Section 3.1.5, “RoCEv2 Congestion Management (RCM)”, on page 46 • Section 3.6.4.1.5, “Mellanox WinOF-2 Congestion Control Counters”, on page 131 • Section , “Mellanox WinOF-2 Congestion Control counters set consists of counters that measure the DCQCN statistics over the network adapter.”, on page 131 • Section 4.2.1.4, “QoS Configuration Utility”, on page 143 • Section 4.2.1.5, “mstdump Utility”, on page 143 • Section 4.3, “Snapshot Utility”, on page 145 Updated the following sections: • Section 3.1.4.6, “Configuring the RoCE Mode”, on page 45 • Section 5.6, “Reported Driver Events”, on page 154 • Section 5.7, “State Dumping”, on page 159

Table 1 - Document Revision History

Document Revision	Date	Changes
Rev 1.30	November 2015	Updated the following sections: • Section 4.2, "Management Utilities", on page 142 • Section 5.6, "Reported Driver Events", on page 154 • Section , "Common Abbreviations and Acronyms", on page 15 Added the following sections: • Section 3.6.4, "Adapter Proprietary Performance Counters", on page 125
Rev 1.21	September 2015	Added the following section: • Section 3.4.3, "Single Root I/O Virtualization (SR-IOV)", on page 80 Updated the version number format - The UM version format was composed of three numbers: major, minor and sub-minor. The sub-minor version was removed from the UM.
Rev 1.20	September, 2015	Added the following sections: • Section 2.2, "Downloading Mellanox WinOF-2 Driver", on page 20 • Section 3.4, "Virtualization", on page 76 • Section 5.5, "Virtualization Related Troubleshooting", on page 153 • Appendix A,"NVGRE Configuration Scripts Examples," on page 162 • Section 3.1.2, "Mode Configuration" • Section 4.2, "Management Utilities"
Rev 1.10	July 8, 2015	Updated the following sections: • Section 1, "Introduction", on page 17 • Section 3.1.4.1, "IP Routable (RoCEv2)", on page 41 • Section 3.1.4.6, "Configuring the RoCE Mode", on page 45
Rev 1.10	June 2015	Beta Release

About this Manual

Scope

Mellanox WinOF-2 is the driver for adapter cards based on the Mellanox ConnectX®-4 family of adapter IC devices. It does not support earlier Mellanox adapter generations.

The document describes WinOF-2 Rev 1.60 features, performance, diagnostic tools, content and configuration. Additionally, this document provides information on various performance tools supplied with this version.

Intended Audience

This manual is intended for system administrators responsible for the installation, configuration, management and maintenance of the software and hardware of Ethernet and InfiniBand adapter cards. It is also intended for application developers.

Documentation Conventions

Table 2 - Documentation Conventions

Description	Convention	Example
File names	file.extension	
Directory names	directory	
Commands and their parameters	command param1	mts3610-1 > show hosts
Required item	< >	
Optional item	[ ]	
Mutually exclusive parameters	{ p1, p2, p3 } or {p1 | p2 | p3}	
Optional mutually exclusive parameters	[p1 | p2 | p3]	
Variables for which users supply specific values	Italic font	*enable*
Emphasized words	Italic font	*These are emphasized words*
Note	<text>	This is a note..
Warning	<text>	May result in system instability.

Common Abbreviations and Acronyms

Table 3 - Abbreviations and Acronyms

Abbreviation / Acronym	Whole Word / Description
B	(Capital) 'B' is used to indicate size in bytes or multiples of bytes (e.g., 1KB = 1024 bytes, and 1MB = 1048576 bytes)
b	(Small) 'b' is used to indicate size in bits or multiples of bits (e.g., 1Kb = 1024 bits)
FW	Firmware
HCA	Host Channel Adapter
HW	Hardware
IB	InfiniBand
LSB	Least significant *byte*
lsb	Least significant *bit*
MSB	Most significant *byte*
msb	Most significant bit
NIC	Network Interface Card
NVGRE	Network Virtualization using Generic Routing Encapsulation
SW	Software
VPI	Virtual Protocol Interconnect
IPoIB	IP over InfiniBand
PFC	Priority Flow Control
PR	Path Record
RDS	Reliable Datagram Sockets
RoCE	RDMA over Converged Ethernet
SL	Service Level
MPI	Message Passing Interface
QoS	Quality of Service
ETW	Event Tracing for Windows
WPP	Windows Software Trace Preprocessor

Related Documents

Table 4 - Related Documents

Document	Description
MFT User Manual	Describes the set of firmware management tools for a single InfiniBand node. MFT can be used for: • Generating a standard or customized Mellanox firmware image Querying for firmware information • Burning a firmware image to a single InfiniBand nodeEnabling changing card configuration to support SRIOV
WinOF-2 Release Notes	For possible software issues, please refer to WinOF-2 Release Notes.
README file	Includes basic installation instructions, summary of main features and requirements.
ConnectX®-4 Firmware Release Notes	For possible firmware issues, please refer to ConnectX®-4 Firmware Release Notes.
InfiniBand™ Architecture Specification, Volume 1, Release 1.2.1	The InfiniBand Specification by IBTA

1 Introduction

This User Manual describes installation, configuration and operation of Mellanox WinOF-2 driver Rev 1.60 package.

Mellanox WinOF-2 is composed of several software modules that contain InfiniBand and Ethernet drivers. It supports 10, 25, 40, 50 or 100 Gb/s Ethernet, and 40, 56 or 100 Gb/s InfiniBand network ports. The port type and speed are determined upon boot based on card capabilities and user settings.

The Mellanox WinOF-2 driver release introduces the following capabilities:

- General applicabilities:
  - Support for Single and Dual port Adapters
  - Receive Side Scaling (RSS)
  - Hardware Tx/Rx checksum offload
  - Large Send Offload (LSO)
  - Adaptive interrupt moderation
  - Support for MSI-X interrupts
  - Network Direct Kernel (NDK) with support for SMBDirect
  - Virtual Machine Queue (VMQ) for Hyper-V
  - Quality of Service (QoS)
    - Support for global flow control and Priority Flow Control (PFC)
    - Enhanced Transmission Selection (ETS)
- Ethernet capabilities:
  - Receive Side Coalescing (RSC)
  - Hardware VLAN filtering
  - RDMA over Converged Ethernet
    - RoCE MAC Based (v1)
    - RoCE IP Based (v1)
    - RoCE over UDP (v2)
  - VXLAN
  - NDKPI v2.0
  - VMMQ
  - PacketDirect Provider Interface (PDPI)
  - NVGRE hardware encapsulation task offload
  - Single Root I/O Virtualization (SR-IOV)
- InfiniBand capabilities:

- Large Receive Offload
- Receive Side Scaling
- Checksum Offloads

For the complete list of Ethernet and InfiniBand Known Issues and Limitations, refer to the latest WinOF-2 Release Notes (www.mellanox.com -> Products -> Software -> InfiniBand/VPI Drivers -> Windows SW/Drivers).

1.1 Supplied Packages

Mellanox WinOF-2 driver Rev 1.60 includes the following package:

- MLNX_WinOF2-1_60_All_x64.exe

1.2 Windows MPI (MS-MPI)

Message Passing Interface (MPI) is meant to provide virtual topology, synchronization, and communication functionality between a set of processes. MPI enables running one process on several hosts. For further details on MPI, please refer to Appendix B,"Windows MPI (MS-MPI)," on page 166.

- Windows MPI runs over the following protocols:
  - Sockets (Ethernet or IPoIB)
  - Network Direct (ND) Ethernet and InfiniBand

1.3 Windows Nano Server 2016

Windows Server 2016 offers a new installation option: Nano Server. Nano Server is a remotely administered server operating system. WinOF-2 currently supports driver-only installation over Nano server, with the following capabilities:

- Support for Single and Dual port Adapters
- Receive Side Scaling (RSS)
- Receive Side Coalescing (RSC)
- Hardware Tx/Rx checksum offload
- Large Send Offload (LSO)
- Adaptive interrupt moderation
- Support for MSI-X interrupts
- Network Direct Kernel (NDK) with support for SMBDirect
- Virtual Machine Queue (VMQ) for Hyper-V
- Hardware VLAN filtering
- RDMA over Converged Ethernet
- RoCE MAC Based (v1)

- RoCE IP Based (v1)
- RoCE over UDP (v2)
- VXLAN
- VMMQ
- PacketDirect Provider Interface (PDPI)
- NVGRE hardware encapsulation task offload
- Quality of Service (QoS)
- Support for global flow control and Priority Flow Control (PFC)
- Enhanced Transmission Selection (ETS)
- Single Root I/O Virtualization (SR-IOV)

2 Installation

2.1 Hardware and Software Requirements

Table 5 - Hardware and Software Requirements

Description[a]	Package
Windows Server 2012 R2	MLNX_WinOF2-1_60_All_x64.exe
Windows Server 2012	MLNX_WinOF2-1_60_All_x64.exe
Windows Server 2016	MLNX_WinOF2-1_60_All_x64.exe
Windows 8.1 Client (64 bit only)	MLNX_WinOF2-1_60_All_x64.exe
Windows 10 Client (64 bit only)	MLNX_WinOF2-1_60_All_x64.exe

a. The Operating System listed above must run with administrator privileges.

2.2 Downloading Mellanox WinOF-2 Driver

To download the .exe according to your Operating System, please follow the steps below:

Step 1. Obtain the machine architecture.

For Windows Server 2012 / 2012 R2 / 8.1 Client

1. To go to the Start menu, position your mouse in the bottom-right corner of the Remote Desktop of your screen.
2. Open a CMD console (Click Task Manager-->File --> Run new task, and enter CMD).
3. Enter the following command.

```
> echo %PROCESSOR_ARCHITECTURE%
```

On an x64 (64-bit) machine, the output will be “AMD64”.

Step 2. Go to the Mellanox WinOF-2 web page at:
http://www.mellanox.com > Products > InfiniBand/VPI Drivers => Windows SW/Drivers.

Step 3. Download the .exe image according to the architecture of your machine (see Step 1). The name of the .exe is in the following format
MLNX_WinOF2-<version>_<arch>.exe.

Installing the incorrect .exe file is prohibited. If you do so, an error message will be displayed. For example, if you try to install a 64-bit .exe on a 32-bit machine, the wizard will display the following (or a similar) error message:

2.3 Installing Mellanox WinOF-2 Driver

WinOF-2 supports adapter cards based on the Mellanox ConnectX®-4 family of adapter IC devices only. If you have ConnectX-3 and ConnectX-3 Pro on your server, you will need to install WinOF driver.
For details on how to install WinOF driver, please refer to WinOF User Manual.

This section provides instructions for two types of installation procedures:

- "Attended Installation"

  An installation procedure that requires frequent user intervention.
- "Unattended Installation"

  An automated installation procedure that requires no user intervention.

Both Attended and Unattended installations require administrator privileges.

2.3.1 Attended Installation

The following is an example of an installation session.

Step 1. Double click the .exe and follow the GUI instructions to install MLNX_WinOF2.

Step 2. [Optional] Manually configure your setup to contain the logs option.

```
> MLNX_WinOF2-1_60_All_x64.exe /v"/l*vx [LogFile]"
```

Step 3. [Optional] If you do not want to upgrade your firmware version[1].

```
> MLNX_WinOF2-1_60_All_x64.exe /v" MT_SKIPFWUPGRD=1"
```

1. MT_SKIPFWUPGRD default value is False

Step 4. Click Next in the Welcome screen.

Step 5. Read then accept the license agreement and click Next.

Step 6. Select the target folder for the installation.

Step 7. The firmware upgrade screen will be displayed in the following cases:

- If the user has an OEM card. In this case, the firmware will not be displayed.
- If the user has a standard Mellanox card with an older firmware version, the firmware will be updated accordingly. However, if the user has both an OEM card and a Mellanox card, only the Mellanox card will be updated.

Step 8. Select a Complete or Custom installation, follow Step a and on.

a.Select the desired feature to install:

- Performances tools - install the performance tools that are used to measure performance in user environment
- Documentation - contains the User Manual and Release Notes
- Management tools - installation tools used for management, such as mlxstat

- Diagnostic Tools - installation tools used for diagnostics, such as mlx5cmd

b. Click Install to start the installation.

Step 9. In case firmware upgrade option was checked in Step 7, you will be notified if a firmware upgrade is required (see).

Step 10. Click Finish to complete the installation.

2.3.2 Unattended Installation

If no reboot options are specified, the installer restarts the computer whenever necessary without displaying any prompt or warning to the user.
Use the `/norestart` or `/forcerestart` standard command-line options to control reboots.

The following is an example of an unattended installation session.

Step 1. Open a CMD console-> Click Start-> Task Manager File-> Run new task-> and enter CMD.

Step 2. Install the driver. Run:

```
> MLNX_WinOF2-1_60_All_x64.exe /S /v/qn
```

Step 3. [Optional] Manually configure your setup to contain the logs option:

```
> MLNX_WinOF2-1_60_All_x64.exe /S /v/qn /v"/l*vx [LogFile]"
```

Step 4. [Optional] if you want to control whether to install ND provider or not[1].

```
> MLNX_WinOF2-1_60_All_x64.exe /vMT_NDPROPERTY=1
```

Step 5. [Optional] If you do not wish to upgrade your firmware version[2].

```
> MLNX_WinOF2-1_60_All_x64.exe /vMT_SKIPFWUPGRD=1
```

Applications that hold the driver files (such as ND applications) will be closed during the unattended installation.

2.4 Installation Results

Upon installation completion, you can verify the successful addition of the network card(s) through the Device Manager.

Upon installation completion, the inf files can be located at:

- %ProgramFiles%\Mellanox\MLNX_WinOF2\Drivers\<OS>

To see the Mellanox network adapters, display the Device Manager and pull down the “Network adapters” menu.

1. MT_NDPROPERTY default value is True
2. MT_SKIPFWUPGRD default value is False

Figure 1: Installation Results

2.5 Extracting Files Without Running Installation

To extract the files without running installation, perform the following steps.

Step 1. Open a CMD console-> Click Start-> Task Manager-> File-> Run new task-> and enter CMD.

Step 2. Extract the driver and the tools:

```
> MLNX_WinOF2-1_60_All_x64 /a
```

- To extract only the driver files.

```
> MLNX_WinOF2-1_60_All_x64 /a /vMT_DRIVERS_ONLY=1
```

Step 3. Click Next to create a server image.

Step 4. Click Change and specify the location in which the files are extracted to.

Step 5. Click Install to extract this folder, or click Change to install to a different folder.

Step 6. To complete the extraction, click Finish.

2.6 Uninstalling Mellanox WinOF-2 Driver

2.6.1 Attended Uninstallation

➢ ***To uninstall MLNX_WinOF2 on a single node:***

Click Start-> Control Panel-> Programs and Features-> MLNX_WinOF2-> Uninstall.
(NOTE: This requires elevated administrator privileges – see Section 1.1, “Supplied Packages”, on page 18 for details.)

2.6.2 Unattended Uninstallation

If no reboot options are specified, the installer restarts the computer whenever necessary without displaying any prompt or warning to the user.

Use the `/norestart` or `/forcerestart` standard command-line options to control reboots.

➢ ***To uninstall MLNX_WinOF2 in unattended mode:***

Step 1. Open a CMD console-> Click Start-> Task Manager-> File-> Run new task-> and enter CMD.

Step 2. Uninstall the driver. Run:

```
> MLNX_WinOF2-1_60_All_x64.exe /S /x /v"/qn"
```

2.7 Firmware Upgrade

If the machine has a standard Mellanox card with an older firmware version, the firmware will be automatically updated as part of the WinOF-2 package installation.

For information on how to upgrade firmware manually, please refer to MFT User Manual: www.mellanox.com ->Products -> InfiniBand/VPI Drivers -> Firmware Tools

2.8 Deploying the Driver on a Nano Server

2.8.1 Offline Installation

➢ ***To deploy the Driver on a Nano Server:***

Step 1. Go to the Mellanox WinOF web page at http://www.mellanox.com > Products > InfiniBand/VPI Drivers => Windows SW/Drivers.

Step 2. Download the driver (MLNX_WinOF2_MLNX_WinOF2-1_60_mlx5_All_win2016_x64_-fre_1_60_15407.exe).

Step 3. Extract the driver to a local directory (see Section 2.5, “Extracting Files Without Running Installation”, on page 29).

Step 4. Copy the contents of this directory to C:\WS2016TP5_Drivers.

The directory contents should appear as follows:

This location should be specified for DriversPath property when injecting driver into the Nano server image:

```
 New-NanoServerImage -MediaPath \\Path\To\Media\en_us -BasePath .\Base -TargetPath
.\InjectingDrivers.vhdx -DriversPath C:\WS2016TP5_Drivers
```

Step 5. Create the Nano server image.

Follow the instructions in the TechNet article for "Getting Started with Nano Server" https://technet.microsoft.com/en-us/library/mt126167.aspx

2.8.2 Online Update

Step 1. Go to the Mellanox WinOF web page at http://www.mellanox.com > Products > InfiniBand/VPI Drivers => Windows SW/Drivers.

Step 2. Download the driver package.

Step 3. Extract the Mellanox drivers to a local directory (see Section 2.5, “Extracting Files Without Running Installation”, on page 29).

Step 4. Open a remote connection to the Nano server.

Step 5. Copy all the driver files to the Nano server

Step 6. Install the driver:

```
pnputil -I -a <Driver_Folder>\mlx5.inf
```

When upgrading the driver on a server where the remote connection was done over Mellanox device, there will be a loss of connectivity. To upgrade, it is recommended to run a script that will execute all the required upgrade commands.

3 Features Overview and Configuration

Once you have installed Mellanox WinOF-2 package, you can perform various modifications to your driver to make it suitable for your system's needs

> Changes made to the Windows registry happen immediately, and no backup is automatically made.
> Do ***not*** edit the Windows registry unless you are confident regarding the changes.

3.1 Ethernet Network

3.1.1 Packet Burst Handling

This feature allows packet burst handling, while avoiding packet drops that may occur when a large amount of packets is sent in a short period of time. For the feature's registry keys, see 3.5.4 "Performance Registry Keys," on page 113.

1. By default, the feature is disabled, and the AsyncRecieveIndicate registry key is set to 0. To enable the feature, choose one of the following options:
   a. To enable packet burst buffering using threaded DPC (recommended), set the AsyncRecieveIndicate registry key to 1.
   b. To enable packet burst buffering using polling, set the AsyncRecieveIndicate to 2.
2. To control the number of reserved receive packets, set the RfdReservationFactor registry key:

Default	150
Recommended	10,000
Maximum	5,000,000

> The memory consumption will increase in accordance with the "RfdReservationFactor" registry key value.

3.1.2 Mode Configuration

➢ ***For retrieving the port types, perform one of the following:***

1. Run mlx5cmd -stat from the "Command Prompt", and check the link_layer from the output.
2. Accomplish the following steps:

Step 1. Display the Device Manager and expand "Network adapters".

Figure 2: Network Adapters

Step 2. Right-click on the Mellanox “ConnectX-4/ConnectX-5 Adapter”, and left-click to select “Properties”.

Step 3. Open the "Information" tab, and check the Port Type. The figure bellow is an example of the displayed Information window for an InfiniBand port.

Figure 3: IB Port

For configuring the port types to Ethernet/InfiniBand mode on a device, use the mlxconfig.exe utility, which is a part of the MFT package for Windows, and is available at http://www.mellanox.com/page/management_tools.

1. Install the WinMFT package.
2. Retrieve the device name:
   a. In command prompt, run "mst status -v":

```
> mst status -v
MST devices:
------------
  mt4099_pci_cr0          bus:dev.fn=04:00.0
  mt4099_pciconf0         bus:dev.fn=04:00.0
  mt4103_pci_cr0          bus:dev.fn=21:00.0
  mt4103_pciconf0         bus:dev.fn=21:00.0

  mt4115_pciconf0         bus:dev.fn=24:00.0
```

   b. Identify the desired device by its "bus:dev.fn" PCIe address.
3. To configure the port type to:

- Ethernet, execute the following command with the appropriate device name:

```
mlxconfig -d mt4115_pciconf0 set LINK_TYPE_P1=2
```

- InfiniBand, execute the following command with the appropriate device name:

```
mlxconfig -d mt4115_pciconf0 set LINK_TYPE_P1=1
```

In order to set the type of the second port, set the parameter LINK_TYPE_P2.

4. Reboot the system.

For further information, please refer to the MFT User Manual.

Changing the port type will change some of the registry keys to the default values of the new port type.

3.1.3 Assigning Port IP After Installation

By default, your machine is configured to obtain an automatic IP address via a DHCP server. In some cases, the DHCP server may require the MAC address of the network adapter installed in your machine.

➢ ***To obtain the MAC address:***

Step 1. Open a CMD console-> Click Start-> Task Manager-> File-> Run new task-> and enter CMD.

Step 2. Display the MAC address as “Physical Address”

```
> ipconfig /all
```

Configuring a static IP is the same for Ethernet adapters.

➢ ***To assign a static IP address to a network port after installation:***

Step 1. Open the Network Connections window. Locate Local Area Connections with Mellanox devices.

Step 2. Right-click a Mellanox Local Area Connection and left-click Properties.

Step 3. Select Internet Protocol Version 4 (TCP/IPv4) from the scroll list and click Properties.

Step 4. Select the "Use the following IP address:" radio button and enter the desired IP information.

Step 5. Click OK.

Step 6. Close the Local Area Connection dialog.

Step 7. Verify the IP configuration by running ‘ipconfig’ from a CMD console.

```
> ipconfig
...
Ethernet adapter Local Area Connection 4:

   Connection-specific DNS Suffix  . :
   IP Address. . . . . . . . . . . . : 11.4.12.63
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . :
...
```

3.1.3.1 Configuring 56GbE Link Speed

Mellanox offers proprietary speed of 56GbE link speed over FDR systems. To achieve this, only the switch, supporting this speed, must be configured to enable it. The NIC, on the other hand, auto-detects this configuration automatically.

➢ ***To achieve 56GbE link speed over a SwitchX® based switch system***

Make sure your switch supports 56GbE speed rates, and that you have the required switch license installed.

Step 1. Set the system profile to be `eth-single-switch`, and reset the system:

```
switch (config) # system profile eth-single-profile
```

Step 1. Set the speed for the desired interface to 56GbE as follows. For example (for interface 1/1)::

```
profileswitch (config) # interface ethernet 1/1
switch (config interface ethernet 1/1) # speed 56000
switch (config interface ethernet 1/1) #
```

Step 1. Verify that the speed rate is 56GbE.:

```
switch (config) # show interface ethernet 1/1
Eth1/1
Admin state: Enabled
Operational state: Down
Description: N\A
Mac address: 00:02:c9:5d:e0:26
MTU: 1522 bytes
Flow-control: receive off send off
Actual speed: 56 Gbps
Switchport mode: access
Rx
0 frames
0 unicast frames
0 multicast frames
0 broadcast frames
0 octets
0 error frames
0 discard frames
Tx
0 frames
0 unicast frames
0 multicast frames
0 broadcast frames
0 octets
0 discard frames
switch (config) #
```

3.1.4 RDMA over Converged Ethernet (RoCE)

Remote Direct Memory Access (RDMA) is the remote memory management capability that allows server to server data movement directly between application memory without any CPU involvement. RDMA over Converged Ethernet (RoCE) is a mechanism to provide this efficient data transfer with very low latencies on loss-less Ethernet networks. With advances in data center convergence over reliable Ethernet, ConnectX® EN with RoCE uses the proven and efficient RDMA transport to provide the platform for deploying RDMA technology in mainstream data center application at 10GigE and 40GigE link-speed. ConnectX® EN with its hardware offload support takes advantage of this efficient RDMA transport (InfiniBand) services over Ethernet to deliver ultra-low latency for performance-critical and transaction intensive applications such as financial, database, storage, and content delivery networks. RoCE encapsulates IB transport and GRH headers in Ethernet packets bearing a dedicated ether type. While the use of GRH is optional within InfiniBand subnets, it is mandatory when using RoCE. Applications written over IB verbs should work seamlessly, but they require provisioning of GRH information when creating address vectors. The library and driver are modified to provide mapping from GID to MAC addresses required by the hardware.

3.1.4.1 IP Routable (RoCEv2)

RoCE has two addressing modes: MAC based GIDs, and IP address based GIDs. In RoCE IP based, if the IP address changes while the system is running, the GID for the port will automatically be updated with the new IP address, using either IPv4 or IPv6.

RoCE IP based allows RoCE traffic between Windows and Linux systems, which use IP based GIDs by default.

A straightforward extension of the RoCE protocol enables traffic to operate in layer 3 environments. This capability is obtained via a simple modification of the RoCE packet format. Instead of the GRH used in RoCE, routable RoCE packets carry an IP header which allows traversal of IP L3 Routers and a UDP header that serves as a stateless encapsulation layer for the RDMA Transport Protocol Packets over IP.

Figure 4: RoCE and RoCE v2 Frame Format Differences

The proposed RoCEv2 packets use a well-known UDP destination port value that unequivocally distinguishes the datagram. Similar to other protocols that use UDP encapsulation, the UDP source port field is used to carry an opaque flow-identifier that allows network devices to implement packet forwarding optimizations (e.g. ECMP) while staying agnostic to the specifics of the protocol header format.

The UDP source port is calculated as follows: `UDP.SrcPort = (SrcPort XOR DstPort) OR 0xC000`, where SrcPort and DstPort are the ports used to establish the connection.
For example, in a Network Direct application, when connecting to a remote peer, the destination IP address and the destination port must be provided as they are used in the calculation above. The source port provision is optional.

Furthermore, since this change exclusively affects the packet format on the wire, and due to the fact that with RDMA semantics packets are generated and consumed below the AP applications can seamlessly operate over any form of RDMA service (including the routable version of RoCE as shown in Figure 4,"RoCE and RoCE v2 Frame Format Differences"), in a completely transparent way[1].

1. Standard RDMA APIs are IP based already for all existing RDMA technologies

Figure 5: RoCE and RoCEv2 Protocol Stack

The fabric must use the same protocol stack in order for nodes to communicate.

In earlier versions, the default value of RoCE mode was RoCE v1. Starting from v1.30, the default value of RoCE mode will be RoCEv2.

Upgrading from earlier versions to version 1.30 or above will save the old default value (RoCEv1).

3.1.4.2 RoCE Configuration

In order to function reliably, RoCE requires a form of flow control. While it is possible to use global flow control, this is normally undesirable, for performance reasons.

The normal and optimal way to use RoCE is to use Priority Flow Control (PFC). To use PFC, it must be enabled on all endpoints and switches in the flow path.

In the following section we present instructions to configure PFC on Mellanox ConnectX™ cards. There are multiple configuration steps required, all of which may be performed via PowerShell. Therefore, although we present each step individually, you may ultimately choose to write a PowerShell script to do them all in one step. Note that administrator privileges are required for these steps.

3.1.4.2.1 Configuring Windows Host

> Since PFC is responsible for flow controlling at the granularity of traffic priority, it is necessary to assign different priorities to different types of network traffic.
> As per RoCE configuration, all ND/NDK traffic is assigned to one or more chosen priorities, where PFC is enabled on those priorities.

Configuring Windows host requires configuring QoS. To configure QoS, please follow the procedure described in Section 3.1.7, "Configuring Quality of Service (QoS)", on page 59

3.1.4.2.1.1Global Pause (Flow Control)

➢ ***To use Global Pause (Flow Control) mode, disable QoS and Priority:***

```
PS $ Disable-NetQosFlowControl
PS $ Disable-NetAdapterQos <interface name>
```

➢ ***To confirm flow control is enabled in adapter parameters:***

Device manager-> Network adapters-> Mellanox ConnectX-4/ConnectX-5 Ethernet Adapter-> Properties
->Advanced tab

3.1.4.3 Configuring SwitchX® Based Switch System

➢ ***To enable RoCE, the SwitchX should be configured as follows:***

- Ports facing the host should be configured as access ports, and either use global pause or Port Control Protocol (PCP) for priority flow control

- Ports facing the network should be configured as trunk ports, and use Port Control Protocol (PCP) for priority flow control

For further information on how to configure SwitchX, please refer to SwitchX User Manual.

3.1.4.4 Configuring Arista Switch

Step 1. Set the ports that face the hosts as trunk.

```
(config)# interface et10
(config-if-Et10)# switchport mode trunk
```

Step 2. Set VID allowed on trunk port to match the host VID.

```
(config-if-Et10)# switchport trunk allowed vlan 100
```

Step 3. Set the ports that face the network as trunk.

```
(config)# interface et20
(config-if-Et20)# switchport mode trunk
```

Step 4. Assign the relevant ports to LAG.

```
(config)# interface et10
(config-if-Et10)# dcbx mode ieee
(config-if-Et10)# speed forced 40gfull
(config-if-Et10)# channel-group 11 mode active
```

Step 5. Enable PFC on ports that face the network.

```
(config)# interface et20
(config-if-Et20)# load-interval 5
(config-if-Et20)# speed forced 40gfull
(config-if-Et20)# switchport trunk native vlan tag
(config-if-Et20)# switchport trunk allowed vlan 11
(config-if-Et20)# switchport mode trunk
(config-if-Et20)# dcbx mode ieee
(config-if-Et20)# priority-flow-control mode on
(config-if-Et20)# priority-flow-control priority 3 no-drop
```

3.1.4.4.1 Using Global Pause (Flow Control)

➢ ***To enable Global Pause on ports that face the hosts, perform the following:***

```
(config)# interface et10
(config-if-Et10)# flowcontrol receive on
(config-if-Et10)# flowcontrol send on
```

3.1.4.4.2 Using Priority Flow Control (PFC)

➢ ***To enable Global Pause on ports that face the hosts, perform the following:***

```
(config)# interface et10
(config-if-Et10)# dcbx mode ieee
```

```
(config-if-Et10)# priority-flow-control mode on
(config-if-Et10)# priority-flow-control priority 3 no-drop
```

3.1.4.5 Configuring Router (PFC only)

The router uses L3's DSCP value to mark the egress traffic of L2 PCP. The required mapping, maps the three most significant bits of the DSCP into the PCP. This is the default behavior, and no additional configuration is required.

3.1.4.5.1 Copying Port Control Protocol (PCP) between Subnets

The captured PCP option from the Ethernet header of the incoming packet can be used to set the PCP bits on the outgoing Ethernet header.

3.1.4.6 Configuring the RoCE Mode

Configuring the RoCE mode requires the following:

- RoCE mode is configured per adapter or per driver. If RoCE mode key is set for the adapter then it will be used. Otherwise, it will be configured by the per-driver key. The per-driver key is shared between all devices in the system.

The supported RoCE modes depend on the firmware installed. If the firmware does not support the needed mode, the fallback mode would be the maximum supported RoCE mode of the installed NIC.

RoCE is enabled by default. Configuring or disabling the RoCE mode can be done via the registry key.

- To update it for a specific adapter using the registry key, set the roce_mode as follows:

Step 1. Find the registry key index value of the adapter according to Section 3.5.1, “Finding the Index Value of the Network Interface”, on page 108.

Step 2. Set the roce_mode in the following path:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e972-e325-11ce-bfc1-08002be10318}\<IndexValue>
```

- To update it for all the devices using the registry key, set the roce_mode as follows:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\mlx5\Parameters\Roce
```

For changes to take effect, please restart the network adapter after changing this registry key.

3.1.4.6.1 Registry Key Parameters

The following are per-driver and will apply to all available adapters.

Table 6 - Registry Key Parameters

Parameters Name	Parameter type	Description	Allowed Values and Default
roce_mode	DWORD	Sets the RoCE mode. The following are the possible RoCE modes: • RoCE MAC Based • RoCE v2 • No RoCE	• RoCE MAC Based = 0 • RoCE v2 = 2 • No RoCE = 4 • Default: RoCE v2

3.1.5 RoCEv2 Congestion Management (RCM)

Please note that this feature is at beta level.

Network Congestion occurs when the number of packets being transmitted through the network approaches the packet handling the capacity of the network. A congested network will suffer from throughput deterioration manifested by increasing time delays and high latency.

In lossy environments, this leads to a packet loss. In lossless environments, it leads to “victim flows” (streams of data which are affected by the congestion, caused by other data flows that pass through the same network).

Example:

The figure below demonstrates a victim flow scenario. In the absence of congestion control, flow X'Y suffers from reduced bandwidth due to flow F'G, which experiences congestion.

Figure 6: Victim Flow Example

To address this, Congestion Control methods and protocols were defined.

This chapter describes (in High-Level), RoCEv2 Congestion Management (RCM), and provides a guide on how to configure it in Windows environment.

RoCEv2 Congestion Management (RCM) provides the capability to avoid congestion hot spots and optimize the throughput of the fabric.

With RCM, congestion in the fabric is reported back to the “sources” of traffic. The sources, in turn, react by throttling down their injection rates, thus preventing the negative effects of fabric buffer saturation and increased queuing delays.

For signaling of congestion, RCM relies on the mechanism defined in RFC3168, also known as DCQCN.

The source node and destination node can be considered as a “closed-loop control” system. Starting from the trigger, when the destination node reflects the congestion alert to the source node, the source node reacts by decreasing, and later on increasing, the Tx rates according to the feedback provided. The source node keeps increasing the Tx rates until the system reaches a steady state of non-congested flow with traffic as high rate as possible.

The RoCEv2 Congestion Management feature is composed of three points:

- The congestion point (CP) - detects congestion and marks packets using the DCQCN bits
- The notification point (NP) (receiving end node) - reacts to the DCQCN marked packets by sending congestion notification packets (CNPs)
- The reaction point (RP) (transmitting end node) - reduces the transmission rate according to the received CNPs

These three components can be seen in the High-Level sequence diagram below:

Figure 7: High-Level Sequence Diagram

For further details, please refer to the IBTA RoCeV2 Spec, Annex A-17.

3.1.5.1 Restrictions and Limitations

➢ ***General:***

- In order for RCM to function properly, the elements in the communication path must support and be configured for RCM (nodes) and DCQCN marking (Switches, Routers).
- ConnectX®-4 and ConnectX®-4 Lx support congestion control only with RoCEv2.
- RCM does not remove/replace the need for flow control.
  In order for RoCEv2 to work properly, flow control must be configured.
  It is not recommended to configure RCM without PFC or global pauses.

➢ ***Mellanox:***

- SW Versions
  - Minimal firmware version - 2.30
  - Minimal driver version - 1.35
- Mellanox switch support starting from “Spectrum”

3.1.5.2 RCM Configuration

RCM configuration to Mellanox adapter is done via Mlx5Cmd tool.

- In order to view the current status of RCM on the adapter, run the following command:

```
Mlx5Cmd.exe -Qosconfig -Dcqcn -Name <Network Adapter Name> -Get
```

In the output example below, RCM is disabled:

```
PS C:\Users\admin\Desktop>Mlx5Cmd.exe -Qosconfig -Dcqcn -Name "Ethernet" -Get
DCQCN RP attributes for adapter "EthernetcnRPEnablePrio0: 0
        DcqcnRPEnablePrio1: 0
        DcqcnRPEnablePrio2: 0
        DcqcnRPEnablePrio3: 0
        DcqcnRPEnablePrio4: 0
        DcqcnRPEnablePrio5: 0
        DcqcnRPEnablePrio6: 0
        DcqcnRPEnablePrio7: 0
        DcqcnClampTgtRate: 0
        DcqcnClampTgtRateAfterTimeInc: 1
        DcqcnRpgTimeReset: 100
        DcqcnRpgByteReset: 400
        DcqcnRpgThreshold: 5
        DcqcnRpgAiRate: 10
        DcqcnRpgHaiRate: 100
        DcqcnAlphaToRateShift: 11
        DcqcnRpgMinDecFac: 50
        DcqcnRpgMinRate: 1
        DcqcnRateToSetOnFirstCnp: 3000
        DcqcnDceTcpG: 32
        DcqcnDceTcpRtt: 4
        DcqcnRateReduceMonitorPeriod: 32
        DcqcnInitialAlphaValue: 0

DCQCN NP attributes for adapter "Ethernet":
        DcqcnNPEnablePrio0: 0
        DcqcnNPEnablePrio1: 0
        DcqcnNPEnablePrio2: 0
        DcqcnNPEnablePrio3: 0
        DcqcnNPEnablePrio4: 0
        DcqcnNPEnablePrio5: 0
        DcqcnNPEnablePrio6: 0
        DcqcnNPEnablePrio7: 0
        DcqcnCnpDscp: 0
        DcqcnCnp802pPrio: 7
        DcqcnCnpPrioMode: 1
The command was executed successfully
```

- In order to enable/disable DCQCN on the adapter, run the following command:

```
Mlx5Cmd.exe -Qosconfig -Dcqcn -Name <Network Adapter Name> -Enable/Disable
```

This can be used on all priorities or on a specific priority:

```
PS C:\Users\admin\Desktop>Mlx5Cmd.exe -Qosconfig -Dcqcn -Name "Ethernet" -Enable

PS C:\Users\admin\Desktop>Mlx5Cmd.exe -Qosconfig -Dcqcn -Name "Ethernet" -Get
DCQCN RP attributes for adapter "Ethernet":
        DcqcnRPEnablePrio0: 1
        DcqcnRPEnablePrio1: 1
        DcqcnRPEnablePrio2: 1
        DcqcnRPEnablePrio3: 1
        DcqcnRPEnablePrio4: 1
        DcqcnRPEnablePrio5: 1
        DcqcnRPEnablePrio6: 1
        DcqcnRPEnablePrio7: 1
        DcqcnClampTgtRate: 0
        DcqcnClampTgtRateAfterTimeInc: 1
        DcqcnRpgTimeReset: 100
        DcqcnRpgByteReset: 400
        DcqcnRpgThreshold: 5
        DcqcnRpgAiRate: 10
        DcqcnRpgHaiRate: 100
        DcqcnAlphaToRateShift: 11
        DcqcnRpgMinDecFac: 50
        DcqcnRpgMinRate: 1
        DcqcnRateToSetOnFirstCnp: 3000
        DcqcnDceTcpG: 32
        DcqcnDceTcpRtt: 4
        DcqcnRateReduceMonitorPeriod: 32
        DcqcnInitialAlphaValue: 0

DCQCN NP attributes for adapter "Ethernet":
        DcqcnNPEnablePrio0: 1
        DcqcnNPEnablePrio1: 1
        DcqcnNPEnablePrio2: 1
        DcqcnNPEnablePrio3: 1
        DcqcnNPEnablePrio4: 1
        DcqcnNPEnablePrio5: 1
        DcqcnNPEnablePrio6: 1
        DcqcnNPEnablePrio7: 1
        DcqcnCnpDscp: 0
        DcqcnCnp802pPrio: 7
        DcqcnCnpPrioMode: 1
The command was executed successfully
```

3.1.5.3 RCM Parameters

The table below lists the parameters that can be configured, their description and allowed values.

Table 7 - RCM Parameters

Parameter (Type)	Default Value	Allowed Values
DcqcnEnablePrio0 (BOOLEAN)	True	0/1

Table 7 - RCM Parameters

Parameter (Type)	Default Value	Allowed Values
DcqcnEnablePrio1 (BOOLEAN)	True	0/1
DcqcnEnablePrio2 (BOOLEAN)	True	0/1
DcqcnEnablePrio3 (BOOLEAN)	True	0/1
DcqcnEnablePrio4 (BOOLEAN)	True	0/1
DcqcnEnablePrio5 (BOOLEAN)	True	0/1
DcqcnEnablePrio6 (BOOLEAN)	True	0/1
DcqcnEnablePrio7 (BOOLEAN)	True	0/1
DcqcnClampTgtRate (1 bit)	0	0/1
DcqcnClampTgtRateAfterTimeInc (1 bit)	1	0/1
DcqcnCnpDscp (6 bits)	0	0 - 7
DcqcnCnp802pPrio (3 bits)	7	0 - 7
DcqcnCnpPrioMode(1 bit)	1	0/1
DcqcnRpgTimeReset (uint32)	600	0 - 131071 [uSec]
DcqcnRpgByteReset (uint32)	32767	0 - 32767 [64 bytes]
DcqcnRpgThreshold (uint32)	5	1 - 31
DcqcnRpgAiRate (uint32)	5	1 - line rate [Mbit/sec]
DcqcnRpgHaiRate (uint32)	50	1 - line rate [Mbit/sec]
DcqcnAlphaToRateShift (uint32)	11	0 - 11
DcqcnRpgMinDecFac (uint32)	50	0 - 100
DcqcnRpgMinRate (uint32)	1	0 - line rate
DcqcnRateToSetOnFirstCnp (uint32)	100	0 - line rate [Mbit/sec]
DcqcnDceTcpG (uint32)	4	0 - 1023 (fixed point fraction
DcqcnDceTcpRtt (uint32)	1	0 - 131071 [uSec]
DcqcnRateReduceMonitorPeriod (uint32)	4	0 - UINT32 [uSec]
DcqcnInitialAlphaValue (uint32)	0	0 - 1023 (fixed point fraction

3.1.5.3.1 RCM Default Parameters

Every parameter has a default value assigned to it. The default value was set for optimal congestion control by Mellanox. In order to view the default parameters on the adapter, run the following command:

```
Mlx5Cmd.exe -Qosconfig -Dcqcn -Name <Network Adapter Name> -Defaults
```

3.1.5.3.2 RCM with Untagged Traffic

Congestion control for untagged traffic is configured with the port default priority that is used for untagged frames.

The port default priority configuration is done via Mlx5Cmd tool.

Table 8 - Default Priority Parameters

Parameter (Type)	Default Value	Allowed Values
DefaultUntaggedPriority	0	0 - 7

- In order to view the current default priority on the adapter, run the following command:

```
Mlx5Cmd.exe -QoSConfig -DefaultUntaggedPriority -Name  -Get
```

- In order to set the default priority to a specific priority on the adapter, run the following command:

```
Mlx5Cmd.exe -QoSConfig -DefaultUntaggedPriority -Name  -Set
```

3.1.5.4 How Changing the Parameters Affect Congestion Control Behavior

Changing the values of the parameters may strongly affect the congestion control efficiency. Please make sure you fully understand the parameter usage, value and expected results before changing its default value.

3.1.5.4.1 CNP Priority

- ***fCnpDscp***

  This parameter changes the priority value on IP level that can be set for CNPs.

- ***DcqcnCnpPrioMode***

  If this parameter is set to '0', then use *DcqcnCnp802pPrio* as the priority value (802.1p) on the Ethernet header of generated CNPs. Otherwise, the priority value of CNPs will be taken from received packets that were marked as DCQCN packets.

- ***DcqcnCnp802pPrio***

  This parameter changes the priority value (802.1p) on the Ethernet header of generated CNPs. Set *DcqcnCnpPrioMode* to '0' in order to use this priority value

3.1.5.4.2 alpha -"α" = Rate Reduction Factor

The device maintains an "alpha" value per QP. This alpha value estimates the current congestion severity in the fabric.

- ***DcqcnInitialAlphaValue***

  This parameter sets the initial value of alpha that should be used when receiving the first CNP for a flow (expressed in a fixed point fraction of 2^10).

  The value of alpha is updated once every DcqcnDceTcpRtt, regardless of the reception of a CNP. If a CNP is received during this time frame, alpha value will increase. If no CNP reception happens, alpha value will decrease.

- ***DcqcnDceTcpG and DcqcnDceTcpRtt***

These two parameters maintain alpha.

- If a CNP is received on the RP - alpha is increased:
  `(1-DcqcnDecTcpG)*α + DcqcnDecTcpG`
- If no CNP is received for a duration of `DcqcnDceTcpRtt` microseconds, alpha is decreased:
  `(1- DcqcnDecTcpG) * alpha`

3.1.5.4.3 Decrease (on the "RP")

Changing the `DcqcnRateToSetOnFirstCnp` parameter determines the current rate (cr) that will be set once the first CNP is received.

The rate is updated only once every `DcqcnRateReduceMonitorPeriod` microseconds (multiple CNPs received during this time frame will not affect the rate) by using the following two formulas:

- $Cr1_{(new)} = (1- (\alpha /(2^{\wedge}DcqcnAlphaToRateShift)))*Cr_{(old)}$
- $Cr2_{(new)} = Cr_{(old)}/DcqcnRpgMinDecFac$

The maximal reduced rate will be chosen from these two formulas.

The target rate will be updated to the previous current rate according to Section 3.1.5.4.4, "Increase (on the "RP")" below.

➢ ***DcqcnRpgMinDecFac***

This parameter defines the maximal ratio of decrease in a single step (Denominator: !zero. Please see formula above).

➢ ***DcqcnAlphaToRateShift***

This parameter defines the decrement rate for a given alpha (see formula above)

➢ ***DcqcnRpgMinRate***

In addition to the `DcqcnRpgMinDecFac`, the `DcqcnRpgMinRate` parameter defines the minimal rate value for the entire single flow.

Note: Setting it to a line rate will disable the congestion control.

3.1.5.4.4 Increase (on the "RP")

RP increases its sending rate using a timer and a byte counter. The byte counter increases rate for every `DcqcnRpgByteResetx64` bytes (mark it as B), while the timer increases rate every `DcqcnRpgTimeReset` time units (mark it as T). Every successful increase due to bytes transmitted/time passing is counted in a variable called rpByteStage and rpTimeStage (respectively).

The `DcqcnRpgThreshold` parameter defines the number of successive increase iteration (mark it as Th).

The increase flow is divided into 3 types of phases, which are actually states in the "RP Rate Control State Machine".

The transition between the steps is decided according to `DcqcnRpgThreshold` parameter.

- Fast Recovery
  If MAX (rpByteStage, rpTimeStage) < Th.
  - No change to Tr
- Additive Increase
  If MAX (rpByteStage, rpTimeStage) > Th. && MIN (rpByteStage, rpTimeStage) < Th.
  - `DcqcnRpgAiRate` value is used to increase Tr
- Hyper Additive Increase
  If MAX (rpByteStage, rpTimeStage) > Th. && MIN (rpByteStage, rpTimeStage) > Th.
  - `DcqcnRpgHaiRate` value is used to increase Tr

For further details, please refer to 802.1Qau standard, sections 32.11-32.15.

➢ ***DcqcnClampTgtRateAfterTimeInc***

When receiving a CNP, the target rate should be updated if the transmission rate was increased due to the timer, and not only due to the byte counter.

➢ ***DcqcnClampTgtRate***

If set, whenever a CNP is processed, the target rate is updated to be the current rate.

3.1.5.5 Mellanox Commands and Examples

For a full description of Congestion Control commands please refer to Mellanox WinOF-2 User Manual section MlxCmd Utilities.

1. Set a value for one or more parameters:

   Mlx5Cmd.exe -Qosconfig -Dcqcn -Name <Network Adapter Name> -Set -Arg1 <value> -Arg2 <value>

   Example:

   ```
   PS C:\Users\admin\Desktop>Mlx5Cmd.exe -Qosconfig -Dcqcn -Name "Ethernet" -Set -DcqcnClampTg-
   tRate 1 -DcqcnCnpDscp 3
   ```

2. Enable/Disable DCQCN for a specific priority:

   Mlx5Cmd.exe -Qosconfig -Dcqcn -Name <Network Adapter Name> -Enable <prio>

   Example:

   ```
   PS C:\Users\admin\Desktop>Mlx5Cmd.exe -Qosconfig -Dcqcn -Name "Ethernet" -Enable/Disable 3
   ```

3. Enable/Disable DCQCN for all priorities:

   Mlx5Cmd.exe -Qosconfig -Dcqcn -Name <Network Adapter Name> -Enable

   Example:

   ```
   PS C:\Users\admin\Desktop>Mlx5Cmd.exe -Qosconfig -Dcqcn -Name "Ethernet" -Enable/Disable
   ```

4. Set port default priority for a specific priority:

   Mlx5Cmd.exe -DefaultUntaggedPriority -Name <Network Adapter Name> -Set <prio>

   Example:

   ```
   PS C:\Users\admin\Desktop>Mlx5Cmd.exe -DefaultUntaggedPriority -Name "Ethernet" -Set 3
   ```

5. Restore the default settings of DCQCN the are defined by Mellanox:

   Mlx5Cmd.exe -Dcqcn -Name <Network Adapter Name> -Restore

   Example:

```
PS C:\Users\admin\Desktop>Mlx5Cmd.exe -Dcqcn -Name "Ethernet" -Restore
```

For information on the RCM counters, please refer to Section 3.6.4.1.5, “Mellanox WinOF-2 Congestion Control Counters”, on page 131.

3.1.6 Teaming and VLAN

Windows Server 2012 and above supports Teaming as part of the operating system. Please refer to Microsoft guide “NIC Teaming in Windows Server 2012” following the link below:

http://www.microsoft.com/en-us/download/confirmation.aspx?id=40319

For other earlier operating systems, please refer to the sections below. Note that the Microsoft teaming mechanism is only available on Windows Server distributions.

3.1.6.1 Creating a Port VLAN in Windows Server 2008 R2

You can create a Port VLAN either on a physical Mellanox ConnectX® EN adapter or a virtual team. The following steps describe how to create a port VLAN.

Step 1. Display the Device Manager.

Step 2. Right-click a Mellanox network adapter (under “Network adapters” list) and left-click Properties. Select the VLAN tab from the Properties sheet.

Physical Adapter

Virtual Bundle (Team)

If a physical adapter has been added to a team, the VLAN tab will not be displayed.

Step 3. Click New to open a VLAN dialog window. Enter the desired VLAN Name and VLAN ID, and select the VLAN Priority.

After installing the first virtual adapter (VLAN) on a specific port, the port becomes disabled. This means that it is not possible to bind to this port until all the virtual adapters associated with it are removed.

When using a VLAN, the network address is configured using the VLAN ID. Therefore, the VLAN ID on both ends of the connection must be the same.

Step 4. Verify the new VLAN(s) by opening the Device Manager window or the Network Connections window. The newly created VLAN will be displayed in the following format.

```
Mellanox Virtual Miniport Driver - VLAN <name>
```

3.1.6.2 Removing a Port VLAN in Windows Server 2008 R2

➢ ***To remove a port VLAN, perform the following steps:***

Step 1. In the Device Manager window, right-click the network adapter from which the port VLAN was created.

Step 2. Left-click Properties.

Step 3. Select the VLAN tab from the Properties sheet.

Step 4. Select the VLAN to be removed.

Step 5. Click Remove and confirm the operation.

3.1.6.3 Configuring a Network Interface to Work with VLAN in Windows Server 2012 and Above

In this procedure you DO NOT create a VLAN, rather use an existing VLAN ID.

➢ ***To configure a port to work with VLAN using the Device Manager.***

Step 1. Open the Device Manager.

Step 2. Go to the Network adapters.

Step 3. Go to the properties of Mellanox ConnectX®-4 Ethernet Adapter card.

Step 4. Go to the Advanced tab.

Step 5. Choose the VLAN ID in the Property window.

Step 6. Set its value in the Value window.

3.1.7 Configuring Quality of Service (QoS)

3.1.7.1 QoS Configuration

Prior to configuring Quality of Service, you must install Data Center Bridging using one of the following methods:

➢ ***To Disable Flow Control Configuration***

Device manager->Network adapters->Mellanox ConnectX-4/ConnectX-5 Ethernet Adapter->Properties->Advanced tab

➢ ***To install the Data Center Bridging using the Server Manager:***

Step 1. Open the 'Server Manager'.

Step 2. Select 'Add Roles and Features'.

Step 3. Click Next.

Step 4. Select 'Features' on the left panel.

Step 5. Check the 'Data Center Bridging' checkbox.

Step 6. Click 'Install'.

➢ ***To install the Data Center Bridging using PowerShell:***

Step 1. Enable Data Center Bridging (DCB).

```
PS $ Install-WindowsFeature Data-Center-Bridging
```

➢ ***To configure QoS on the host:***

The procedure below is not saved after you reboot your system. Hence, we recommend you create a script using the steps below and run it on the startup of the local machine.
Please see the procedure below on how to add the script to the local machine startup scripts.

Step 1. Change the Windows PowerShell execution policy:

```
PS $ Set-ExecutionPolicy AllSigned
```

Step 2. Remove the entire previous QoS configuration:

```
PS $ Remove-NetQosTrafficClass
PS $ Remove-NetQosPolicy -Confirm:$False
```

Step 3. Set the DCBX Willing parameter to false as Mellanox drivers do not support this feature.

```
PS $ set-NetQosDcbxSetting -Willing 0
```

Step 4. Create a Quality of Service (QoS) policy and tag each type of traffic with the relevant priority.

In this example, TCP/UDP use priority 1, SMB over TCP use priority 3.

```
PS $ New-NetQosPolicy "DEFAULT" -store Activestore -Default -PriorityValue8021Action 3
PS $ New-NetQosPolicy "TCP" -store Activestore -IPProtocolMatchCondition TCP -Priority-
Value8021Action 1
PS $ New-NetQosPolicy "UDP" -store Activestore -IPProtocolMatchCondition UDP -Priority-
Value8021Action 1
New-NetQosPolicy “SMB” -SMB -PriorityValue8021Action 3
```

Step 5. Create a QoS policy for SMB over SMB Direct traffic on Network Direct port 445.

```
PS $ New-NetQosPolicy "SMBDirect" -store Activestore -NetDirectPortMatchCondition 445 -
PriorityValue8021Action 3
```

Step 6. [Optional] If VLANs are used, mark the egress traffic with the relevant VlanID.
The NIC is referred as "Ethernet 4” in the examples below.

```
PS $ Set-NetAdapterAdvancedProperty -Name "Ethernet 4" -RegistryKeyword "VlanID" -Reg-
istryValue "55"
```

Step 7. [Optional] Configure the IP address for the NIC.

If DHCP is used, the IP address will be assigned automatically.

```
PS $ Set-NetIPInterface -InterfaceAlias “Ethernet 4” -DHCP Disabled
PS $ Remove-NetIPAddress -InterfaceAlias “Ethernet 4” -AddressFamily IPv4 -Con-
firm:$false
PS $ New-NetIPAddress -InterfaceAlias “Ethernet 4” -IPAddress 192.168.1.10 -Prefix-
Length 24 -Type Unicast
```

Step 8. [Optional] Set the DNS server (assuming its IP address is 192.168.1.2).

```
PS $ Set-DnsClientServerAddress -InterfaceAlias “Ethernet 4” -ServerAddresses
192.168.1.2
```

After establishing the priorities of ND/NDK traffic, the priorities must have PFC enabled on them.

Step 9. Disable Priority Flow Control (PFC) for all other priorities except for 3.

```
PS $ Disable-NetQosFlowControl 0,1,2,4,5,6,7
```

Step 10. Enable QoS on the relevant interface.

```
PS $ Enable-NetAdapterQos -InterfaceAlias "Ethernet 4"
```

Step 11. Enable PFC on priority 3.

```
PS $ Enable-NetQosFlowControl -Priority 3
```

➢ ***To add the script to the local machine startup scripts:***

Step 1. From the PowerShell invoke.

```
gpedit.msc
```

Step 2. In the pop-up window, under the 'Computer Configuration' section, perform the following:

1. Select Windows Settings
2. Select Scripts (Startup/Shutdown)
3. Double click Startup to open the Startup Properties
4. Move to “PowerShell Scripts” tab

5. Click Add

The script should include only the following commands:

```
PS $ Remove-NetQosTrafficClass
PS $ Remove-NetQosPolicy -Confirm:$False
PS $ set-NetQosDcbxSetting -Willing 0
PS $ New-NetQosPolicy "SMB" -Policystore Activestore -NetDirectPortMatchCondition
445 -PriorityValue8021Action 3
PS $ New-NetQosPolicy "DEFAULT" -Policystore Activestore -Default -PriorityVal-
ue8021Action 3
PS $ New-NetQosPolicy "TCP" -Policystore Activestore -IPProtocolMatchCondition TCP
-PriorityValue8021Action 1
PS $ New-NetQosPolicy "UDP" -Policystore Activestore -IPProtocolMatchCondition UDP
-PriorityValue8021Action 1
PS $ Disable-NetQosFlowControl 0,1,2,4,5,6,7
PS $ Enable-NetAdapterQos -InterfaceAlias "port1"
PS $ Enable-NetAdapterQos -InterfaceAlias "port2"
PS $ Enable-NetQosFlowControl -Priority 3
```

```
PS $ New-NetQosTrafficClass -name "SMB class" -priority 3 -bandwidthPercentage 50 -
Algorithm ETS
```

6. Browse for the script's location.
7. Click OK
8. To confirm the settings applied after boot run:

```
PS $ get-netqospolicy -policystore activestore
```

3.1.7.2 Enhanced Transmission Selection

Enhanced Transmission Selection (ETS) provides a common management framework for assignment of bandwidth to frame priorities as described in the IEEE 802.1Qaz specification:

http://www.ieee802.org/1/files/public/docs2008/az-wadekar-ets-proposal-0608-v1.01.pdf

For further details on configuring ETS on Windows™ Server, please refer to:

http://technet.microsoft.com/en-us/library/hh967440.aspx

3.1.8 Differentiated Services Code Point (DSCP)

DSCP is a mechanism used for classifying network traffic on IP networks. It uses the 6-bit Differentiated Services Field (DS or DSCP field) in the IP header for packet classification purposes. Using Layer 3 classification enables you to maintain the same classification semantics beyond local network, across routers.

Every transmitted packet holds the information allowing network devices to map the packet to the appropriate 802.1Qbb CoS. For DSCP based PFC or ETS the packet is marked with a DSCP value in the Differentiated Services (DS) field of the IP header.

3.1.8.1 System Requirements

- Operating Systems: Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2 and
- Firmware version: 12/14/16.18.1000 or higher

3.1.8.2 Setting the DSCP in the IP Header

Marking DSCP value in the IP header is done differently for IP packets constructed by the NIC (e.g. RDMA traffic) and for packets constructed by the IP stack (e.g. TCP traffic).

- For IP packets generated by the IP stack, the DSCP value is provided by the IP stack. The NIC does not validate the match between DSCP and Class of Service (CoS) values. CoS and DSCP values are expected to be set through standard tools, such as PowerShell command New-NetQosPolicy using PriorityValue8021Action and DSCPAction flags respectively.
- For IP packets generated by the NIC (RDMA), the DSCP value is generated according to the CoS value programmed for the interface. CoS value is set through standard tools, such as PowerShell command New-NetQosPolicy using PriorityValue8021Action flag.

The NIC uses a mapping table between the CoS value and the DSCP value configured through the RroceDscpMarkPriorityFlow- Control[0-7] Registry keys

3.1.8.3 Configuring Quality of Service for TCP and RDMA Traffic

Step 1. Verify that DCB is installed and enabled (is not installed by default).

```
PS $ Install-WindowsFeature Data-Center-Bridging
```

Step 2. Import the PowerShell modules that are required to configure DCB.

```
PS $ import-module NetQos
PS $ import-module DcbQos
PS $ import-module NetAdapter
```

Step 3. Configure DCB.

```
PS $ Set-NetQosDcbxSetting -Willing 0
```

Step 4. Enable Network Adapter QoS.

```
PS $ Set-NetAdapterQos -Name "CX4_P1" -Enabled 1
```

Step 5. Enable Priority Flow Control (PFC) on the specific priority 3,5.

```
PS $ Enable-NetQosFlowControl 3,5
```

3.1.8.4 Configuring DSCP to Control PFC for TCP Traffic

- Create a QoS policy to tag All TCP/UDP traffic with CoS value 3 and DSCP value 9.

```
PS $ New-NetQosPolicy "DEFAULT" -Default -PriorityValue8021Action 3 -DSCPAction 9
```

DSCP can also be configured per protocol.

```
PS $ New-NetQosPolicy "TCP"  -IPProtocolMatchCondition TCP -PriorityValue8021Action 3 -
DSCPAction 16
PS $ New-NetQosPolicy "UDP"  -IPProtocolMatchCondition UDP -PriorityValue8021Action 3 -
DSCPAction 32
```

3.1.8.5 Configuring DSCP to Control ETS for TCP Traffic

- Create a QoS policy to tag All TCP/UDP traffic with CoS value 0 and DSCP value 8.

```
PS $ New-NetQosPolicy "DEFAULT" -Default -PriorityValue8021Action 0 -DSCPAction 8 -Pol-
icyStore activestore
```

- Configure DSCP with value 16 for TCP/IP connections with a range of ports.

```
PS $ New-NetQosPolicy "TCP1" -DSCPAction 16 -IPDstPortStartMatchCondition 31000 -IPDst-
PortEndMatchCondition 31999 -IPProtocol TCP -PriorityValue8021Action 0 -PolicyStore
activestore
```

- Configure DSCP with value 24 for TCP/IP connections with another range of ports.

```
PS $ New-NetQosPolicy "TCP2" -DSCPAction 24 -IPDstPortStartMatchCondition 21000 -IPDst-
PortEndMatchCondition 31999 -IPProtocol TCP -PriorityValue8021Action 0 -PolicyStore
activestore
```

- Configure two Traffic Classes with bandwidths of 16% and 80%.

```
PS $ New-NetQosTrafficClass -name "TCP1" -priority 3 -bandwidthPercentage 16 -Algorithm
ETS
PS $ New-NetQosTrafficClass -name "TCP2" -priority 5 -bandwidthPercentage 80 -Algorithm
ETS
```

3.1.8.6 Configuring DSCP to Control PFC for RDMA Traffic

- Create a QoS policy to tag the ND traffic for port 10000 with CoS value 3.

```
PS $ New-NetQosPolicy "ND10000" -NetDirectPortMatchCondition 10000 - PriorityVal-
ue8021Action 3
```

Related Commands:

- Get-NetAdapterQos - Gets the QoS properties of the network adapter
- Get-NetQosPolicy - Retrieves network QoS policies
- Get-NetQosFlowControl - Gets QoS status per priority

3.1.8.7 Registry Settings

The following attributes must be set manually and will be added to the miniport registry.

For more information on configuring registry keys, see 3.5 “Configuration Using Registry Keys,” on page 108.

Table 9 - DSCP Registry Keys Settings

Registry Key	Description
TxUntagPriorityTag	If 0x1, do not add 802.1Q tag to transmitted packets which are assigned 802.1p priority, but are not assigned a non-zero VLAN ID (i.e. priority-tagged). Default 0x0, for DSCP based PFC set to 0x1.
RxUntaggedMapToLossless	If 0x1, all untagged traffic is mapped to the lossless receive queue. Default 0x0, for DSCP based PFC set to 0x1.
PriorityToDscpMappingTable_<ID>	A value to mark DSCP for RoCE packets assigned to CoS=ID, when priority flow control is enabled. The valid values range is from 0 to 63, Default is ID value, e.g. PriorityToDscpMapping-Table_3 is 3. ID values range from 0 to 7.
DscpBasedEtsEnabled	If 0x1 - all Dscp based ETS feature is enabled, if 0x0 - disabled. Default 0x0.
DscpForGlobalFlowControl	Default DSCP value for flow control. Default 0x1a.

For changes to take affect, please restart the network adapter after changing any of the above registry keys.

3.1.8.7.1 Default Settings

When DSCP configuration registry keys are missing in the miniport registry, the following defaults are assigned:

Table 10 - DSCP Default Registry Keys Settings

Registry Key	Default Value
TxUntagPriorityTag	0
RxUntaggedMapToLossles	0
PriorityToDscpMappingTable_0	0
PriorityToDscpMappingTable_1	1
PriorityToDscpMappingTable_2	2
PriorityToDscpMappingTable_3	3
PriorityToDscpMappingTable_4	4
PriorityToDscpMappingTable_5	5
PriorityToDscpMappingTable_6	6
PriorityToDscpMappingTable_7	7
DscpBasedEtsEnabled	eth:0
DscpForGlobalFlowControl	26

3.1.8.8 DSCP Sanity Testing

To verify that all QoS and DSCP settings are correct, you can capture the incoming and outgoing traffic by using the mlx5cmd sniffer tool. The tool allows you to see the DSCP value in the captured packets, as displayed in the figure below.

3.1.9 Configuring the Ethernet Driver

The following steps describe how to configure advanced features.

Step 1. Display the Device Manager.

Step 2. Right-click a Mellanox network adapter (under “Network adapters” list) and left-click Properties. Select the Advanced tab from the Properties sheet.

Step 3. Modify configuration parameters to suit your system.

Please note the following:

- For help on a specific parameter/option, check the help button at the bottom of the dialog.
- If you select one of the entries Offload Options, Performance Options, or Flow Control Options, you’ll need to click the Properties button to modify parameters via a pop-up dialog.

3.1.10 Receive Segment Coalescing (RSC)

RSC allows redaction of CPU utilization when dealing with large TCP message size. It allows the drive to indicate to the Operating System once, per-message and not per-MTU that Packet Offload can be disabled for IPv4 or IPv6 traffic in the Advanced tab of the driver proprieties.

RSC provides diagnostic counters documented at Table 21, “Mellanox WinOF-2 Port Traffic Counters,” on page 125: Receive Segment Coalescing (RSC)

3.1.11 Receive Side Scaling (RSS)

RSS settings can be set per individual adapters as well as globally.

➢ ***To do so, set the registry keys listed below:***

For instructions on how to find interface index in registry <nn>, please refer to Section 3.5.1, "Finding the Index Value of the Network Interface", on page 108.

Table 11 - Registry Keys Setting

Sub-key	Description
HKLM\SYSTEM\CurrentControlSet\Control\Class\{4d36e972-e325-11ce-bfc1-08002be10318}\<nn>*MaxRSSProcessors	**Maximum number of CPUs allotted.** Sets the desired maximum number of processors for each interface. The number can be different for each interface. Note: Restart the network adapter after you change this registry key.
HKLM\SYSTEM\CurrentControlSet\Control\Class\{4d36e972-e325-11ce-bfc1-08002be10318}\<nn>*RssBaseProcNumber	**Base CPU number.** Sets the desired base CPU number for each interface. The number can be different for each interface. This allows partitioning of CPUs across network adapters. Note: Restart the network adapter when you change this registry key.
HKLM\SYSTEM\CurrentControlSet\Control\Class\{4d36e972-e325-11ce-bfc1-08002be10318}\<nn>*NumaNodeID	NUMA node affinitization
HKLM\SYSTEM\CurrentControlSet\Control\Class\{4d36e972-e325-11ce-bfc1-08002be10318}\<nn>*RssBaseProcGroup	Sets the RSS base processor group for systems with more than 64 processors.

3.1.12 Wake on LAN (WoL)

Wake on LAN is a technology that allows a network admin to remotely power on a system or to wake it up from sleep mode by a network message. WoL is enabled by default.

3.1.13 Data Center Bridging Exchange (DCBX)

Data Center Bridging Exchange (DCBX) protocol is an LLDP based protocol which manages and negotiates host and switch configuration. The WinOF-2 driver supports the following:

- PFC - Priority Flow Control
- ETS - Enhance Transmission Selection
- Application priority

The protocol is widely used to assure lossless path when running multiple protocols at the same time. DCBX is functional as part of configuring QoS mentioned in section Section 3.1.7, "Con-

figuring Quality of Service (QoS)", on page 59. Users should make sure the willing bit on the host is enabled, using PowerShell if needed.:

```
set-NetQosDcbxSetting -Willing 1
```

This is required to allow negotiating and accepting peer configurations. Willing bit is set to 1 by default by the operating system.

The new settings can be queried by calling the following command in PowerShell

```
Get-NetAdapterQos
```

Note: The below configuration was received from the switch in the below example.

The output would look like the following:

```
PS C:\Users\Administrator> get-netadapterqos


Name    : Ethernet 9
Enabled : True

Name    : Ethernet 10
Enabled : True

Name                        : Ethernet 7
Enabled                     : True
Capabilities                :                        Hardware     Current
                                                     --------     -------
                              MacSecBypass         : NotSupported NotSupported
                              DcbxSupport          : IEEE         IEEE
                              NumTCs(Max/ETS/PFC)  : 8/8/8        8/8/8

OperationalTrafficClasses   : TC TSA    Bandwidth Priorities
                              -- ---    --------- ----------
                               0 ETS    25%       0-1
                               1 ETS    25%       2-3
                               2 ETS    25%       4-5
                               3 ETS    25%       6-7

OperationalFlowControl      : Priorities 0-4 Enabled
OperationalClassifications  : Not Available
RemoteTrafficClasses        : TC TSA    Bandwidth Priorities
                              -- ---    --------- ----------
                               0 ETS    25%       0-1
                               1 ETS    25%       2-3
                               2 ETS    25%       4-5
                               3 ETS    25%       6-7

RemoteFlowControl           : Priorities 0-4 Enabled
RemoteClassifications       : Not Available
```

In a scenario where both peers are set to Willing, the adapter with a lower MAC address takes the settings of the peer.

DCBX is disabled in the driver by default and in the some firmware versions as well.

➢ ***To use DCBX:***

1. Query and enable DCBX in the firmware.
   a. Install WinMFT package and go to `\Program Files\Mellanox\WinMFT`
   b. Get the list of devices, run "mst status".

   c. Verify is the DCBX is enabled or disabled, run "`mlxconfig.exe -d mt4117_pciconf0 query`".

   d. If disabled, run the following commands for a dual-port card.

```
mlxconfig -d mt4117_pciconf0 set LLDP_NB_RX_MODE_P1=2
mlxconfig -d mt4117_pciconf0 set LLDP_NB_TX_MODE_P1=2
mlxconfig -d mt4117_pciconf0 set LLDP_NB_DCBX_P1=1
mlxconfig -d mt4117_pciconf0 set LLDP_NB_RX_MODE_P2=2
mlxconfig -d mt4117_pciconf0 set LLDP_NB_TX_MODE_P2=2
mlxconfig -d mt4117_pciconf0 set LLDP_NB_DCBX_P2=1
```

2. Add the "`DcbxMode`" registry key, set the value to "2" and reload the adapter.
   The registry key should be added to `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControl-Set\Control\Class\{4d36e972-e325-11ce-bfc1-08002be10318}\<IndexValue>`

To find the IndexValue, refer to Section 3.5.1, “Finding the Index Value of the Network Interface”, on page 108

3.1.14 Receive Path Activity Monitoring

In the event where the device or the Operating System unexpectedly becomes unresponsive for a long period of time, the Flow Control mechanism may send pause frames, which will cause congestion spreading to the entire network.

To prevent this scenario, the device monitors its status continuously, attempting to detect when the receive pipeline is stalled. When the device detects a stall for a period longer than a pre-configured timeout, the Flow Control mechanisms (Global Pause and PFC) are automatically disabled.

If the PFC is in use, and one or more priorities are stalled, the PFC will be disabled on all priorities. When the device detects that the stall has ceased, the flow control mechanism will resume with its previously configured behavior.

Two registry parameters control the mechanism’s behavior: the DeviceRxStallTime-out key controls the time threshold for disabling the flow control, and the DeviceRxStallWatermark key controls a diagnostics counter that can be used for early detection of stalled receive. For more information, see Table 16 - “Ethernet Registry Keys,” on page 117.

3.1.15 Head of Queue Lifetime Limit

This feature enables the system to drop the packets that have been awaiting transmission for a long period of time, preventing the system from hanging. The implementation of the feature complies with the Head of Queue Lifetime Limit (HLL) definition in the InfiniBand™ Architecture Specification (see Table 4 - “Related Documents,” on page 16).

The HLL has three registry keys for configuration:

TCHeadOfQueueLifeTimeLimit, TCStallCount and TCHeadOfQueueLifeTimeLimitEnable (see Table 16 - “Ethernet Registry Keys,” on page 117).

3.2 InfiniBand Network

3.2.1 Feature Limitations

The following features are not supported

- VXLAN

- NVGRE
- Receive Side Coalescing (RSC)
- VLAN
- SRIOV
- PKeys

3.2.2 Port Configuration

For more information on port configuration, please refer to 3.1.2 “Mode Configuration,” on page 34.

3.2.3 Modifying IPoIB Configuration

➢ ***To modify the IPoIB configuration after installation, perform the following steps:***

Step 1. Open Device Manager and expand Network Adapters in the device display pane.

Step 2. Right-click the Mellanox ConnectX Adapter entry and left-click Properties.

Step 3. Click the Advanced tab and modify the desired properties.

The IPoIB network interface is automatically restarted once you finish modifying IPoIB parameters. Consequently, it might affect any running traffic.

3.2.4 Displaying Adapter Related Information

To display a summary of network adapter software, firmware- and hardware-related information such as driver version, firmware version, adapter identity, and network port link information, perform the following steps:

Step 1. Display the Device Manager.

Step 2. Select the Information tab from the Properties sheet.

To save this information for debug purposes, click **Save to File** and provide the output file name.

3.2.5 Assigning Port IP After Installation

For more information on port configuration, please refer to Section 3.1.3, "Assigning Port IP After Installation", on page 37 under the Ethernet Network.

3.2.6 Receive Side Scaling (RSS)

For more information on port configuration, please refer to Section 3.1.11, "Receive Side Scaling (RSS)", on page 68 under the Ethernet Network.

3.3 Storage Protocols

3.3.1 Deploying SMB Direct

The Server Message Block (SMB) protocol is a network file sharing protocol implemented in Microsoft Windows. The set of message packets that defines a particular version of the protocol is called a dialect.

The Microsoft SMB protocol is a client-server implementation and consists of a set of data packets, each containing a request sent by the client or a response sent by the server.

SMB protocol is used on top of the TCP/IP protocol or other network protocols. Using the SMB protocol allows applications to access files or other resources on a remote server, to read, create, and update them. In addition, it enables communication with any server program that is set up to receive an SMB client request.

3.3.1.1 SMB Configuration Verification

3.3.1.1.1 Verifying Network Adapter Configuration

Use the following PowerShell cmdlets to verify Network Direct is globally enabled and that you have NICs with the RDMA capability.

- Run on both the SMB server and the SMB client.

```
PS $ Get-NetOffloadGlobalSetting | Select NetworkDirect
PS $ Get-NetAdapterRDMA
PS $ Get-NetAdapterHardwareInfo
```

3.3.1.1.2 Verifying SMB Configuration

Use the following PowerShell cmdlets to verify SMB Multichannel is enabled, confirm the adapters are recognized by SMB and that their RDMA capability is properly identified.

- On the SMB client, run the following PowerShell cmdlets:

```
PS $ Get-SmbClientConfiguration | Select EnableMultichannel
PS $ Get-SmbClientNetworkInterface
```

- On the SMB server, run the following PowerShell cmdlets[1]:

```
PS $ Get-SmbServerConfiguration | Select EnableMultichannel
PS $ Get-SmbServerNetworkInterface
PS $ netstat.exe -xan | ? {$_ -match "445"}
```

3.3.1.1.3 Verifying SMB Connection

➢ ***To verify the SMB connection on the SMB client:***

Step 1. Copy the large file to create a new session with the SMB Server.

1. The NETSTAT command confirms if the File Server is listening on the RDMA interfaces.

Step 2. Open a PowerShell window while the copy is ongoing.

Step 3. Verify the SMB Direct is working properly and that the correct SMB dialect is used.

```
PS $ Get-SmbConnection
PS $ Get-SmbMultichannelConnection
PS $ netstat.exe -xan | ? {$_ -match "445"}
```

If you have no activity while you run the commands above, you might get an empty list due to session expiration and absence current connections.

3.3.1.2 Verifying SMB Events that Confirm RDMA Connection

➢ ***To confirm RDMA connection, verify the SMB events:***

Step 1. Open a PowerShell window on the SMB client.

Step 2. Run the following cmdlets.
NOTE: Any RDMA-related connection errors will be displayed as well.

```
PS $ Get-WinEvent -LogName Microsoft-Windows-SMBClient/Operational | ? Message -match
"RDMA"
```

For further details on how to configure the switches to be lossless, please refer to https://community.mellanox.com

3.4 Virtualization

3.4.1 Hyper-V with VMQ

3.4.1.1 System Requirements

Operating Systems: Windows Server 2012 and Windows Server 2012 R2

3.4.1.2 Using Hyper-V with VMQ

Mellanox WinOF-2 Rev 1.60 includes a Virtual Machine Queue (VMQ) interface to support Microsoft Hyper-V network performance improvements and security enhancement.

VMQ interface supports:

- Classification of received packets by using the destination MAC address to route the packets to different receive queues
- NIC ability to use DMA to transfer packets directly to a Hyper-V child-partition's shared memory

- Scaling to multiple processors, by processing packets for different virtual machines on different processors.

➢ ***To enable Hyper-V with VMQ using UI:***

Step 1. Open Hyper-V Manager.

Step 2. Right-click the desired Virtual Machine (VM), and left-click Settings in the pop-up menu.

Step 3. In the Settings window, under the relevant network adapter, select “Hardware Acceleration”.

Step 4. Check/uncheck the box “Enable virtual machine queue” to enable/disable VMQ on that specific network adapter.

➢ ***To enable Hyper-V with VMQ using PowerShell:***

Step 1. Enable VMQ on a specific VM: Set-VMNetworkAdapter <VM Name> -VmqWeight 100

Step 2. Disable VMQ on a specific VM: Set-VMNetworkAdapter <VM Name> -VmqWeight 0

3.4.2 Network Virtualization using Generic Routing Encapsulation (NVGRE)

Network Virtualization using Generic Routing Encapsulation (NVGRE) offload is currently supported in Windows Server 2012 R2 with the latest updates for Microsoft.

3.4.2.1 System Requirements

Operating Systems: Windows Server 2012 R2

3.4.2.2 Using NVGRE

Network Virtualization using Generic Routing Encapsulation (NVGRE) is a network virtualization technology that attempts to alleviate the scalability problems associated with large cloud computing deployments. It uses Generic Routing Encapsulation (GRE) to tunnel layer 2 packets across an IP fabric, and uses 24 bits of the GRE key as a logical network discriminator (which is called a tenant network ID).

Configuring the Hyper-V Network Virtualization, requires two types of IP addresses:

- **Provider Addresses (PA)** - unique IP addresses assigned to each Hyper-V host that are routable across the physical network infrastructure. Each Hyper-V host requires at least one PA to be assigned.
- **Customer Addresses (CA)** - unique IP addresses assigned to each Virtual Machine that participate on a virtualized network. Using NVGRE, multiple CAs for VMs running on a Hyper-V host can be tunneled using a single PA on that Hyper-V host. CAs must be unique across all VMs on the same virtual network, but they do not need to be unique across virtual networks with different Virtual Subnet ID.

The VM generates a packet with the addresses of the sender and the recipient within the CA space. Then Hyper-V host encapsulates the packet with the addresses of the sender and the recipient in PA space.

PA addresses are determined by using Virtualization table. Hyper-V host retrieves the received packet, identifies recipient and forwards the original packet with the CA addresses to the desired VM.

NVGRE can be implemented across an existing physical IP network without requiring changes to physical network switch architecture. Since NVGRE tunnels terminate at each Hyper-V host, the hosts handle all encapsulation and de-encapsulation of the network traffic. Firewalls that block GRE tunnels between sites have to be configured to support forwarding GRE (IP Protocol 47) tunnel traffic.

For further details on configuring NVGRE, please refer to Appendix A,"NVGRE Configuration Scripts Examples," on page 162

Figure 8: NVGRE Packet Structure

3.4.2.3 Enabling/Disabling NVGRE Offloading

To leverage NVGRE to virtualize heavy network IO workloads, the Mellanox ConnectX®-4 network NIC provides hardware support for GRE offload within the network NICs by default.

➢ ***To enable/disable NVGRE offloading:***

Step 1. Open the Device Manager.

Step 2. Go to the Network adapters.

Step 3. Right click 'Properties' on Mellanox ConnectX®-4 Ethernet Adapter card.

Step 4. Go to Advanced tab.

Step 5. Choose the `'Encapsulate Task Offload'` option.

Step 6. Set one of the following values:

- Enable - GRE offloading is Enabled by default
- Disabled - When disabled the Hyper-V host will still be able to transfer NVGRE traffic, but TCP and inner IP checksums will be calculated by software that significant reduces performance.

3.4.2.3.1 Configuring the NVGRE using PowerShell

Hyper-V Network Virtualization policies can be centrally configured using PowerShell 3.0 and PowerShell Remoting.

Step 1. **[Windows Server 2012 Only]** Enable the Windows Network Virtualization binding on the physical NIC of each Hyper-V Host (Host 1 and Host 2)

```
PS $ Enable-NetAdapterBinding <EthInterfaceName>(a)-ComponentID ms_netwnv
```

<EthInterfaceName> - Physical NIC name

Step 2. Create a vSwitch.

```
PS $ New-VMSwitch <vSwitchName> -NetAdapterName <EthInterfaceName>-AllowManagementOS
$true
```

Step 3. Shut down the VMs.

```
PS $ Stop-VM -Name <VM Name> -Force -Confirm
```

Step 4. Configure the Virtual Subnet ID on the Hyper-V Network Switch Ports for each Virtual Machine on each Hyper-V Host (Host 1 and Host 2).

```
PS $ Add-VMNetworkAdapter -VMName <VMName> -SwitchName <vSwitchName> -StaticMacAddress
<StaticMAC Address>
```

Step 5. Configure a Subnet Locator and Route records on all Hyper-V Hosts (same command on all Hyper-V hosts)

```
PS $ New-NetVirtualizationLookupRecord -CustomerAddress <VMInterfaceIPAddress 1/n> -
ProviderAddress <HypervisorInterfaceIPAddress1> -VirtualSubnetID <virtualsubnetID> -
MACAddress <VMmacaddress1>[a] -Rule "TranslationMethodEncap"

PS $ New-NetVirtualizationLookupRecord -CustomerAddress <VMInterfaceIPAddress 2/n> -
ProviderAddress <HypervisorInterfaceIPAddress2> -VirtualSubnetID <virtualsubnetID> -
MACAddress <VMmacaddress2>[a] -Rule "TranslationMethodEncap"
```

a. This is the VM's MAC address associated with the vSwitch connected to the Mellanox device.

Step 6. Add customer route on all Hyper-V hosts (same command on all Hyper-V hosts).

```
PS $ New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-
000000005001}" -VirtualSubnetID <virtualsubnetID> -DestinationPrefix <VMInterfaceIPAd-
dress/Mask> -NextHop "0.0.0.0" -Metric 255
```

Step 7. Configure the Provider Address and Route records on each Hyper-V Host using an appropriate interface name and IP address.

```
PS $ $NIC = Get-NetAdapter <EthInterfaceName>
PS $ New-NetVirtualizationProviderAddress -InterfaceIndex $NIC.InterfaceIndex -Provid-
erAddress <HypervisorInterfaceIPAddress> -PrefixLength 24
```

```
PS $ New-NetVirtualizationProviderRoute -InterfaceIndex $NIC.InterfaceIndex -Destina-
tionPrefix "0.0.0.0/0" -NextHop <HypervisorInterfaceIPAddress>
```

Step 8. Configure the Virtual Subnet ID on the Hyper-V Network Switch Ports for each Virtual Machine on each Hyper-V Host (Host 1 and Host 2).

```
PS $ Get-VMNetworkAdapter -VMName <VMName> | where {$_.MacAddress -eq <VMmacaddress1>}
| Set-VMNetworkAdapter -VirtualSubnetID <virtualsubnetID>
```

Please repeat steps 5 to 8 on each Hyper-V after rebooting the Hypervisor.

3.4.2.4 Verifying the Encapsulation of the Traffic

Once the configuration using PowerShell is completed, verifying that packets are indeed encapsulated as configured is possible through any packet capturing utility. If configured correctly, an encapsulated packet should appear as a packet consisting of the following headers:

Outer ETH Header, Outer IP, GRE Header, Inner ETH Header, Original Ethernet Payload.

3.4.2.5 Removing NVGRE configuration

Step 1. Set VSID back to 0 (on each Hyper-V for each Virtual Machine where VSID was set)

```
PS $ Get-VMNetworkAdapter <VMName>(a) | where {$_.MacAddress -eq <VMMacAddress>(b)} |
Set-VMNetworkAdapter -VirtualSubnetID 0
```

- VMName - the name of Virtual machine
- VMMacAddress - the MAC address of VM's network interface associated with vSwitch that was connected to Mellanox device.

Step 2. Remove all lookup records (same command on all Hyper-V hosts).

```
PS $ Remove-NetVirtualizationLookupRecord
```

Step 3. Remove customer route (same command on all Hyper-V hosts).

```
PS $ Remove-NetVirtualizationCustomerRoute
```

Step 4. Remove Provider address (same command on all Hyper-V hosts).

```
PS $ Remove-NetVirtualizationProviderAddress
```

Step 5. Remove provider routed for a Hyper-V host.

```
PS $ Remove-NetVirtualizationProviderRoute
```

Step 6. For HyperV running Windows Server 2012 only disable network adapter binding to ms_-netwnv service

```
PS $ Disable-NetAdapterBinding <EthInterfaceName>(a) -ComponentID ms_netwnv
<EthInterfaceName> - Physical NIC name
```

3.4.3 Single Root I/O Virtualization (SR-IOV)

Single Root I/O Virtualization (SR-IOV) is a technology that allows a physical PCIe device to present itself multiple times through the PCIe bus. This technology enables multiple virtual instances of the device with separate resources. Mellanox adapters are capable of exposing up to 96 virtual instances called Virtual Functions (VFs) per port. These virtual functions can then be provisioned separately. Each VF can be seen as an addition device connected to the Physical Function. It also shares resources with the Physical Function.

SR-IOV is commonly used in conjunction with an SR-IOV enabled hypervisor to provide virtual machines direct hardware access to network resources hence increasing its performance.

This guide demonstrates the setup and configuration of SR-IOV, using Mellanox adapter cards family. SR-IOV VF is a single port device.

3.4.3.1 SR-IOV Ethernet over Hyper-V

3.4.3.1.1 System Requirements

- A server and BIOS with SR-IOV support. BIOS settings might need to be updated to enable virtualization support and SR-IOV support.
- Hypervisor OS: Windows Server 2012 R2
- Virtual Machine (VM) OS:
  - The VM OS can be either Windows Server 2012 and above
- Mellanox ConnectX®-4 VPI Adapter Card family
- Mellanox WinOF-2 1.20 or higher

3.4.3.1.2 Feature Limitations

- RDMA (i.e RoCE) capability is not available in SR-IOV mode
- SR-IOV is supported only in Ethernet ports

3.4.3.2 Configuring SR-IOV Host Machines

The following are the necessary steps for configuring host machines:

3.4.3.2.1 Enabling SR-IOV in BIOS

Depending on your system, perform the steps below to set up your BIOS. The figures used in this section are for illustration purposes only.

For further information, please refer to the appropriate BIOS User Manual.

➢ ***To enable SR-IOV in BIOS:***

Step 1. Make sure the machine's BIOS supports SR-IOV.

Please, consult BIOS vendor website for SR-IOV supported BIOS versions list. Update the BIOS version if necessary.

Step 2. Follow BIOS vendor guidelines to enable SR-IOV according to BIOS User Manual. For example:

a. Enable SR-IOV.

b. Enable “Intel Virtualization Technologhy” Support

For further details, please refer to the vendor's website.

3.4.3.2.2 Installing Hypervisor Operating System (SR-IOV Ethernet Only)

➢ ***To install Hypervisor Operating System:***

Step 1. Install Windows Server 2012 R2

Step 2. Install Hyper-V role:

- Go to: Server Manager -> Manage -> Add Roles and Features and set the following:
  - Installation Type -> Role-based or Feature-based Installation
  - Server Selection -> Select a server from the server pool
  - Server Roles -> Hyper-V (see figures below)

Step 3. Install Hyper-V Management Tools.

Features - > Remote Server Administration Tools -> Role Administration Tools -> Hyper-V Administration Tool.

Step 4. Confirm the installation

Step 5. Click Install

Step 6. Reboot the system.

3.4.3.2.3 Verifying SR-IOV Support within the Host Operating System (SR-IOV Ethernet Only)

➢ ***To verify that the system is properly configured for SR-IOV:***

Step 1. Go to: Start-> Windows Powershell.

Step 2. Run the following PowerShell commands.

```
PS $ (Get-VmHost).IovSupport
PS $ (Get-VmHost).IovSupportReasons
```

In case that SR-IOV is supported by the OS, the output in the PowerShell is as in the figure below.

Figure 9: Operating System Supports SR-IOV

Note: If BIOS was updated according to BIOS vendor instructions and you see the message displayed in the figure below, update the registry configuration as described in the (Get-VmHost).IovSupportReasons message.

Figure 10: SR-IOV Support

Step 3. Reboot

Step 4. Verify the system is configured correctly for SR-IOV as described in Steps 1/2.

3.4.3.2.4 Creating a Virtual Machine (SR-IOV Ethernet Only)

➢ ***To create a virtual machine***

Step 1. Go to: Server Manager -> Tools -> Hyper-V Manager.

Step 2. Go to: New->Virtual Machine and set the following:

- Name: <name>
- Startup memory: 4096 MB
- Connection: Not Connected

Figure 11: Hyper-V Manager

Step 3. Connect the virtual hard disk in the New Virtual Machine Wizard.

Step 4. Go to: Connect Virtual Hard Disk -> Use an existing virtual hard disk.

Step 5. Select the location of the vhd file.

Figure 12: Connect Virtual Hard Disk

3.4.3.2.5 Configuring Host Memory Limit per VF

In SR-IOV mode, the host allocates memory resources per the adapter`s needs for each VF. It is important to limit the amount of memory that the VF can receive from the host, in order to ensure the host's stability. To prevent excessive allocation, the MaxFWPagesUsagePerVF registry key must be configured to the maximum number of 4KB pages that the host could allocate for VFs resources. See 3.5.5.4 "SR-IOV Options," on page 119.

3.4.3.3 Configuring Mellanox Network Adapter for SR-IOV

The following are the steps for configuring Mellanox Network Adapter for SR-IOV:

3.4.3.3.1 Enabling SR-IOV in Firmware

For non-Mellanox (OEM) branded cards you may need to download and install the new firmware. For the latest OEM firmware, please go to:
http://www.mellanox.com/page/oem_firmware_download

➢ ***To enable SR-IOV using mlxconfig:***

mlxconfig is part of MFT tools used to simplify firmware configuration. The tool is available with MFT tools 3.6.0 or higher.

Step 1. Download MFT for Windows.

www.mellanox.com > Products > Software > Firmware Tools

Step 2. Get the device ID (look for the "`_pciconf`" string in the output).

```
> mst status
```

Example:

```
MST devices:
------------
mt4115_pciconf0
```

Step 3. Check the current SR-IOV configuration.

```
> mlxconfig -d mt4115_pciconf0 q
```

Example:

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

Device type:    ConnectX4
PCI device:     mt4115_pciconf0

Configurations:          Current
         SRIOV_EN        N/A
         NUM_OF_VFS      N/A
         WOL_MAGIC_EN_P2 N/A
         LINK_TYPE_P1    N/A
         LINK_TYPE_P2    N/A
```

Step 4. Enable SR-IOV with 16 VFs.

```
> mlxconfig -d mt4115_pciconf0 s SRIOV_EN=1 NUM_OF_VFS=16
```

All servers are guaranteed to support 16 VFs. Increasing the number of VFs can lead to exceeding the BIOS limit of MMIO available address space.

OS limits the maximum number of VFs to 32 per Network Adapter.
To increase the number of VFs, the following PowerShell command should be used:
`Set-NetAdapterSRIOV - name <AdapterName> -NumVFs <Required number of VFs>`

Example:

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

Device type:    ConnectX4
PCI device:     mt4115_pciconf0

Configurations:          Current New
         SRIOV_EN        N/A     1
         NUM_OF_VFS      N/A     16
         WOL_MAGIC_EN_P2 N/A     N/A
         LINK_TYPE_P1    N/A     N/A
         LINK_TYPE_P2    N/A     N/A

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

3.4.3.4 Configuring Operating Systems

3.4.3.4.1 Configuring Virtual Machine Networking (Ethernet SR-IOV Only)

➢ ***To configure Virtual Machine networking:***

Step 1. Create an SR-IOV-enabled Virtual Switch over Mellanox Ethernet Adapter.

Go to: Start -> Server Manager -> Tools -> Hyper-V Manager

In the Hyper-V Manager: Actions -> Virtual SwitchManager -> External-> Create Virtual Switch

Step 2. Set the following:

- Name:
- External network:
- Enable single-root I/O virtualization (SR-IOV)

Figure 13: Virtual Switch with SR-IOV

Step 3. Click **Apply**.

Step 4. Click OK.

Step 5. Add a VMNIC connected to a Mellanox vSwitch in the VM hardware settings:

- Under Actions, go to Settings -> Add New Hardware-> Network Adapter-> OK.
- In “Virtual Switch” dropdown box, choose Mellanox SR-IOV Virtual Switch.

Figure 14: Adding a VMNIC to a Mellanox v-Switch

Step 6. Enable the SR-IOV for Mellanox VMNIC:

1. Open VM settings Wizard.
2. Open the Network Adapter and choose Hardware Acceleration.
3. Tick the “Enable SR-IOV” option.
4. Click OK.

Figure 15: Enable SR-IOV on VMNIC

Step 7. Start and connect to the Virtual Machine:

Select the newly created Virtual Machine and go to: Actions panel-> Connect.
In the virtual machine window go to: Actions-> Start

Step 8. Copy the WinOF-2 driver package to the VM using Mellanox VMNIC IP address.

Step 9. Install WinOF-2 driver package on the VM.

Step 10. Reboot the VM at the end of installation.

Step 11. Verify that Mellanox Virtual Function appears in the device manager.

Figure 16: Virtual Function in the VM

To achieve best performance on SR-IOV VF, please run the following powershell commands on the host:

- For 10Gbe:
  - `PS $ Set-VMNetworkAdapter -Name "Network Adapter" -VMName vm1 -Iov-QueuePairsRequested 4`
- For 40Gbe and above:
  - `PS $ Set-VMNetworkAdapter -Name "Network Adapter" -VMName vm1 -Iov-QueuePairsRe-quested 8`

3.4.3.5 VF Spoof Protection

WinOF-2 supports two levels of spoof protection:

- Hypervisor sets VF's MAC address and only packets with that MAC can be transmitted by the VF
- Hypervisor can control allowed Ethertypes that the VF can transmit

If a VF attempts to transmit packets with undesired source MAC or Ethertype, the packets will be dropped by an internal e-Switch.

By default, the anti-spoof filter is enabled with the following Ethertypes:

- Internet Protocol version 4 (IPv4) (0x0800)
- Internet Protocol Version 6 (IPv6) (0x86DD)
- Address Resolution Protocol (ARP) (0x0806)

The hypervisor can configure an Ethertype table for VFs, which includes a set of allowed Ethertypes values for transmission via the NIC registry. The registry keys are as follows:

Table 12 - VF Spoof Protection Registry Keys

Key Name	Key Type	Values	Description
VFAllowedTxEther-TypeListEnable	REG_SZ	0 = Disabled 1 = Enabled (default)	Enables/disables the feature
VFAllowedTxEther-Type0	REG_D-WORD	Ethertype value	The first Ethertype to allow VF to transmit
VFAllowedTxEther-Type1	REG_D-WORD	Ethertype value	The second Ethertype to allow VF to transmit
VFAllowedTxEther-Type2	REG_D-WORD	Ethertype value	The third Ethertype to allow VF to transmit
VFAllowedTxEther-Type3	REG_D-WORD	Ethertype value	The fourth Ethertype to allow VF to transmit
VFAllowedTxEther-Type4	REG_D-WORD	Ethertype value	The fifth Ethertype to allow VF to transmit
VFAllowedTxEther-Type5	REG_D-WORD	Ethertype value	The sixth Ethertype to allow VF to transmit
VFAllowedTxEther-Type6	REG_D-WORD	Ethertype value	The seventh Ethertype to allow VF to transmit
VFAllowedTxEther-Type7	REG_D-WORD	Ethertype value	The eighth Ethertype to allow VF to transmit

- By default, the feature is enabled and uses the default Ethertype table.
- The Source MAC protection cannot be disabled, and the Ethertype protection can be disabled by setting the VFAllowedTxEtherTypeListEnable key to 0.
- When the feature is disabled, only the Ethernet flow control protocol (0x8808) is restricted to be transmitted by the VF.
- Configuring at least one Ethertype in the registry will override the default table of the Ethertypes mentioned above.

3.4.3.5.1 Limitations

When one of the following Ethertypes is enabled, the other is automatically disabled, and vice versa:

- 0x8906 - Fibre Channel over Ethernet (FCoE)
- 0x8914 - FCoE Initialization Protocol

3.4.3.6 Configuring Operating Systems

3.4.3.6.1 Configuring Virtual Machine Networking (InfiniBand SR-IOV Only)

For further details on enabling/configuring SR-IOV on KVM, please refer to the section titled "Single Root IO Virtualization (SR-IOV)" in *Mellanox OFED for Linux User Manual.*

3.4.3.6.2 Configuring Virtual Machine Networking (Ethernet SR-IOV Only)

➢ ***To configure Virtual Machine networking:***

Step 1. Create an SR-IOV-enabled Virtual Switch over Mellanox Ethernet Adapter.

Go to: Start -> Server Manager -> Tools -> Hyper-V Manager

In the Hyper-V Manager: Actions -> Virtual SwitchManager -> External-> Create Virtual Switch

Step 2. Set the following:

- Name:
- External network:
- Enable single-root I/O virtualization (SR-IOV)

Figure 17: Virtual Switch with SR-IOV

Step 3. Click **Apply**.

Step 4. Click **OK**.

Step 5. Add a VMNIC connected to a Mellanox vSwitch in the VM hardware settings:

- Under Actions, go to Settings -> Add New Hardware-> Network Adapter-> OK.
- In “Virtual Switch” dropdown box, choose Mellanox SR-IOV Virtual Switch.

Figure 18: Adding a VMNIC to a Mellanox V-switch

Step 6. Enable the SR-IOV for Mellanox VMNIC:

1. Open VM settings Wizard.
2. Open the Network Adapter and choose Hardware Acceleration.
3. Tick the “Enable SR-IOV” option.
4. Click OK.

Figure 19: Enable SR-IOV on VMNIC

Step 7. Start and connect to the Virtual Machine:

Select the newly created Virtual Machine and go to: Actions panel-> Connect.
In the virtual machine window go to: Actions-> Start

Step 8. Copy the WinOF driver package to the VM using Mellanox VMNIC IP address.

Step 9. Install WinOF driver package on the VM.

Step 10. Reboot the VM at the end of installation.

Step 11. Verify that Mellanox Virtual Function appears in the device manager.

Figure 20: Virtual Function in the VM

To achieve best performance on SR-IOV VF, please run the following powershell commands on the host:

For 10Gbe:

PS $ Set-VMNetworkAdapter -Name "Network Adapter" -VMName vm1 -IovQueuePairsRequested 4

For 40Gbe and 56Gbe:

PS $ Set-VMNetworkAdapter -Name "Network Adapter" -VMName vm1 -IovQueuePairsRequested 8

3.4.4 Virtual Machine Multiple Queue (VMMQ)

Virtual Machine Multiple Queues (VMMQ), formerly known as Hardware vRSS, is a NIC offload technology that provides scalability for processing network traffic of a VPort in the host (root partition) of a virtualized node. In essence, VMMQ extends the native RSS feature to the VPorts that are associated with the physical function (PF) of a NIC including the default VPort.

VMMQ is available for the VPorts exposed in the host (root partition) regardless of whether the NIC is operating in SR-IOV or VMQ mode. VMMQ is a feature available in Windows Server 2016.

3.4.4.1 System Requirements

- Operating System(s): Windows Server 2016
- Available only for Ethernet (no IPOIB)
- Mellanox ConnectX-4/ConnectX-4 LxVPI/ConnectX-5 adapter card family

3.4.4.2 Enabling/Disabling VMMQ

3.4.4.2.1 On the Driver Level

➢ ***To enable/disable VMMQ:***

Step 1. Go to: Display Manager-> Network adapters->Mellanox ConnectX-4/ConnectX-5 Ethernet Adapter->Properties-> advanced tab->Virtual Switch Rss

Step 2. Select Enabled or Disabled

➢ ***To enable/disable VMMQ using a Registry Key:***

Set the `RssOnHostVPorts` registry key in the following path to either `1` (enabled) or `0` (disabled)

```
HKLM\SYSTEM\CurrentControlSet\Control\Class\{4d36e972-e325-11ce-bfc1-08002be10318}\<nn>\* RssOnHostVPorts
```

3.4.4.2.2 On a VPort

➢ ***To enable VMMQ on a VPort:***

```
PS $ Set-VMNetworkAdapter -Name "Virtual Adapter Name" -VmmqEnabled $true
```

➢ ***To disable VMMQ on a VPort:***

```
PS $ Set-VMNetworkAdapter -Name "Virtual Adapter Name" -VmmqEnabled $false
```

Since the VMMQ is an offload feature for vRss, vRss must be enabled prior to enabling VMMQ.

3.4.4.3 Controlling the Number of Queues Allocated for a vPort

The requested number of queues for a virtual network adapter (vPort) can be set by invoking this PS cmdlet:

```
PS $ Set-VMNetworkAdapter -Name "VM Name" -name "Virtual Adapter Name" -VmmqQueuePairs <number>
```

The number provided to this cmdlet is the requested number of queues per vPort. However, the OS might decide to not fulfill the request due to some resources and other factors considerations.

3.4.5 Network Direct Kernel Provider Interface

As of v1.45, WinOF-2 supports NDIS Network Direct Kernel Provider Interface version 2. The Network Direct Kernel Provider Interface (NDKPI) is an extension to NDIS that allows IHVs to provide kernel-mode Remote Direct Memory Access (RDMA) support in a network adapter.

3.4.5.1 System Requirement

- Operating System: Windows Server 2012 R2 (Without NDK from/to a VM) and Windows 2016

3.4.5.2 Configuring NDK

3.4.5.2.1 General Configurations

Step 1. Make sure the port is configured as Ethernet.

Step 2. Make sure the RoCE mode is configured the same on both ends, run "`mlx5cmd -stat`" from the "Command Prompt". ROCE v2 is the default mode.

3.4.5.2.2 Configuring NDK for Virtual NICs

Step 1. Create a VMSwitch.

```
PS $ New-VMSwitch -Name <vSwitchName> -NetAdapterName <EthInterfaceName> -AllowManagementOS $False
```

Step 2. Create the virtual network adapters.

```
PS $ Add-VMNetworkAdapter -SwitchName <vSwitchName> -Name <EthInterfaceName> -Manage-
mentOS
```

Step 3. Enable the "Network Direct(RDMA)" on the new virtual network adapters.

```
PS $ Enable-NetAdapterRdma <EthInterfaceName>
```

3.4.5.2.3 Configuring the VM

Step 1. Make sure your machine supports SR-IOV.

Step 2. Create a VM (make sure the VM is running the same OS as host)

Step 3. Create an SR-IOV enabled VMSwitch.

```
PS $ New-VMSwitch -Name <vSwitchName> -NetAdapterName <EthInterfaceName> -EnableIov
$True -AllowManagementOS $True
```

Step 4. Add a Network Adapter to the VM in the Hyper-V Manager, and choose the VMSwitch just created.

Step 5. Check the "Enable SR-IOV" option on the "Hardware Acceleration" under the Network Adapter.

If you turn ON the VM at this time in the VM Device Manager, you should see Mellanox Virtual Adapter under the Network adapters.

Step 6. Install the Mellanox Driver in the VM.
Use the same package you installed on the host

Step 7. Enable RDMA on the corresponding network adapter in the VM (Run the command in the VM)

```
PS $ Enable-NetAdapterRdma <EthInterfaceName>
```

3.4.5.3 Utility to Run and Monitor NDK

3.4.5.3.1 Running NDK

Since SMB is NDK's client, it should be used to generate traffic. To generate traffic, do a big copy from one machine to the other.

For instance, use "xcopy" to recursively copy the entire c:\Windows directory or from a "Command Prompt" window, run:

```
xcopy /s c:\Windows \\<remote machine ip>\<remote machine directory for receiving>
```

For example:

```
xcopy /s c:\Windows \\11.0.0.5\c$\tmp
```

3.4.5.3.2 Validating NDK

During the run time of NDK test (xcopy), with "RDMA Activity" in the perfmon.

Use the mlx5cmd sniffer to see the protocol information of the traffic packet.

3.4.6 PacketDirect Provider Interface

As of v1.45, WinOF-2 supports NDIS PacketDirect Provider Interface. PacketDirect extends NDIS with an accelerated I/O model, which can increase the number of packets processed per second by an order of magnitude and significantly decrease jitter when compared to the traditional NDIS I/O path.

PacketDirect is supported only on Ethernet ports.

3.4.6.1 System Requirements

- Hypervisor OS: Windows Server 2016
- Available only for Ethernet (no IPOIB)
- Virtual Machine (VM) OS: Windows Server 2012 and above
- Mellanox ConnectX-4/ConnectX-4 Lx/ConnectX-5/ConnectX-5 Ex
- Mellanox WinOF-2 1.45 or higher
- Firmware version: 12.16.1020/14.16.1020 or higher

3.4.6.2 Using PacketDirect for VM

➢ ***To allow a VM to send/receive traffic in PacketDirect mode:***

Step 1. Enable PacketDirect:

- On the Ethernet adapter.

```
PS $ Enable-NetAdapterPacketDirect -Name <EthInterfaceName>
```

- In the Device Manager.

Step 2. Create a vSwitch with PacketDirect enabled.

```
 PS $ New-VMSwitch <vSwitchName> -NetAdapterName <EthInterfaceName> -EnablePacketDirect
$true -AllowManagementOS $true
```

Step 3. Enable VFP extension:

- On the vSwitch.

```
 PS $ Enable-VMSwitchExtension -VmSwitchName <vSwitchName> -Name "Windows Azure VFP
Switch Extension"
```

- In the Hyper-V Manager: Action->Virtual Switch Manager...

Step 4. Shut down the VM.

```
PS $ Stop-VM -Name <VMName> -Force -Confirm
```

Step 5. Add a virtual network adapter for the VM.

```
PS $ Add-VMNetworkAdapter -VMName <VMName> -SwitchName <vSwitchName> -StaticMacAddress
<StaticMAC Address>
```

Step 6. Start the VM.

```
PS $ Start-VM -Name <VMName>
```

Since VFP is enabled, without any forwarding rules, it will block all traffic going through the VM.

Follow the following steps to unblock the traffic

Step a. Find the port name for the VM.

```
CMD > vfpctrl /list-vmswitch-port
……..
Port name              : E431C413-D31F-40EB-AD96-0B2D45FE34AA
Port Friendly name     :
Switch name            : 8B288106-9DB6-4720-B144-6CC32D53E0EC
Switch Friendly name   : MlnxSwitch
PortId                 : 3
VMQ Usage              : 0
SR-IOV Usage           : 0
Port type              : Synthetic
 Port is Initialized.
 MAC Learning is Disabled.
NIC name             : bd65960d-4215-4a4f-bddc-962a5d0e2fa0--e7199a49-6cca-
4d3c-a4cd-22907592527e
NIC Friendly name    : testnic
MTU                  : 1500
MAC address          : 00-15-5D-4C-46-00
VM name              : vm
......
Command list-vmswitch-port succeeded!
```

Step 7. Disable the port to allow traffic.

```
CMD > vfpctrl /disable-port /port <PortName>
Command disable-port succeeded!
```

The port should be disabled after each reboot of the VM to allow traffic.

3.4.6.3 Disable Loopback Mode

In this mode, the NIC switch does not forward traffic from one function to another. All transmitted traffic is sent to the Up-link port, and the packet classification on the transmit side is performed by an external switch. Filtering logic only affects packets arriving from the Up-link port.
The user can configure loopback disable mode separately for multi-cast and uni-cast traffic, by setting eSWUCLoopback and eSWMCLoopback registry values accordingly. For more information, see Table 20 - “SR-IOV Options,” on page 120.

3.5 Configuration Using Registry Keys

3.5.1 Finding the Index Value of the Network Interface

To find the index value of your Network Interface from the Device Manager please perform the following steps:

Step 1. Open Device Manager, and go to Network Adapters.

Step 2. Right click ->Properties on Mellanox Connect-X® Ethernet Adapter.

Step 3. Go to Details tab.

Step 4. Select the Driver key, and obtain the nn number.

In the below example, the index equals 0010

All registry keys added for driver configuration should be of string type (REG_SZ).

After setting a registry key and re-loading the driver, you may use the mlx5cmd -regkeys command to assure that the value was read by the driver.

3.5.2 Basic Registry Keys

This group contains the registry keys that control the basic operations of the NIC

Table 13 - Basic Registry Keys

Value Name	Default Value	Description
*JumboPacket	ETH: 1514 IPoIB: 4092	The maximum size of a frame (or a packet) that can be sent over the wire. This is also known as the maximum transmission unit (MTU). The MTU may have a significant impact on the network's performance as a large packet can cause high latency. However, it can also reduce the CPU utilization and improve the wire efficiency. The standard Ethernet frame size is 1514 bytes, but Mellanox drivers support wide range of packet sizes. The valid values are: • Ethernet: 600 up to 9600 • IPoIB: 256 up to 4092 **Note**: All the devices across the network (switches and routers) should support the same frame size. Be aware that different network devices calculate the frame size differently. Some devices include the header, i.e. information in the frame size, while others do not. Mellanox adapters do not include Ethernet header information in the frame size. (i.e when setting *JumboPacket to 1500, the actual frame size is 1514).
*ReceiveBuffers	512	The number of packets each ring receives. This parameter affects the memory consumption and the performance. Increasing this value can enhance receive performance, but also consumes more system memory. In case of lack of received buffers (dropped packets or out of order received packets), you can increase the number of received buffers. The valid values are 256 up to 4096.
*TransmitBuffers	2048	The number of packets each ring sends. Increasing this value can enhance transmission performance, but also consumes system memory. The valid values are 256 up to 4096.

3.5.3 Offload Registry Keys

This group of registry keys allows the administrator to specify which TCP/IP offload settings are handled by the adapter rather than by the operating system.

Enabling offloading services increases transmission performance. Due to offload tasks (such as checksum calculations) performed by adapter hardware rather than by the operating system (and, therefore, with lower latency). In addition, CPU resources become more available for other tasks.

Table 14 - Offload Registry Keys

Value Name	Default Value	Description
*LsoV2IPv4	1	Large Send Offload Version 2 (IPv4). The valid values are: • 0: disable • 1: enable
*LsoV2IPv6	1	Large Send Offload Version 2 (IPv6). The valid values are: • 0: disable • 1: enable
LSOSize	64000	The maximum number of bytes that the TCP/IP stack can pass to an adapter in a single packet. This value affects the memory consumption and the NIC performance. The valid values are MTU+1024 up to 64000. **Note:** This registry key is not exposed to the user via the UI. If LSOSize is smaller than MTU+1024, LSO will be disabled.
LSOMinSegment	2	The minimum number of segments that a large TCP packet must be divisible by, before the transport can offload it to a NIC for segmentation. The valid values are 2 up to 32. **Note**: This registry key is not exposed to the user via the UI.
LSOTcpOptions	1	Enables that the miniport driver to segment a large TCP packet whose TCP header contains TCP options. The valid values are: • 0: disable • 1: enable **Note**: This registry key is not exposed to the user via the UI.
LSOIpOptions	1	Enables its NIC to segment a large TCP packet whose IP header contains IP options. The valid values are: • 0: disable • 1: enable **Note**: This registry key is not exposed to the user via the UI.

Table 14 - Offload Registry Keys

Value Name	Default Value	Description
*IPChecksumOff-loadIPv4	3	Specifies whether the device performs the calculation of IPv4 checksums. The valid values are: • 0: (disable) • 1: (Tx Enable) • 2: (Rx Enable) • 3: (Tx and Rx enable)
*TCPUDPChecksu-mOffloadIPv4	3	Specifies whether the device performs the calculation of TCP or UDP checksum over IPv4. The valid values are: • 0: (disable) • 1: (Tx Enable) • 2: (Rx Enable) • 3: (Tx and Rx enable)
*TCPUDPChecksu-mOffloadIPv6	3	Specifies whether the device performs the calculation of TCP or UDP checksum over IPv6. The valid values are: • 0: (disable) • 1: (Tx Enable) • 2: (Rx Enable) • 3: (Tx and Rx enable)
*RssOnHostVPorts	1	Virtual Machine Multiple Queue (VMMQ) HW Offload The valid values are: • 0: disable • 1: enable
*SwParsing	N/A	Specifies whether the device performs the calculation of TCP checksum over IP-in-IP encapsulated IPv4/6 sent packets. The valid values are: •0: (disable) •1: (Tx Enable)

3.5.4 Performance Registry Keys

This group of registry keys configures parameters that can improve adapter performance.

Table 15 - Performance Registry Keys

Value Name	Default Value	Description
TxIntModerationProfile	1	Enables the assignment of different interrupt moderation profiles for send completions. Interrupt moderation can have great effect on optimizing network throughput and CPU utilization. The valid values are: • 0: Low Latency Implies higher rate of interrupts to achieve better latency, or to handle scenarios where only a small number of streams are used. • 1: Moderate Interrupt moderation is set to midrange defaults to allow maximum throughput at minimum CPU utilization for common scenarios. • 2: Aggressive Interrupt moderation is set to maximal values to allow maximum throughput at minimum CPU utilization for more intensive, multi-stream scenarios.
RxIntModerationProfile	1	Enables the assignment of different interrupt moderation profiles for receive completions. Interrupt moderation can have a great effect on optimizing network throughput and CPU utilization. The valid values are: • 0: Low Latency Implies higher rate of interrupts to achieve better latency, or to handle scenarios where only a small number of streams are used. • 1: Moderate Interrupt moderation is set to midrange defaults to allow maximum throughput at minimum CPU utilization for common scenarios. • 2: Aggressive Interrupt moderation is set to maximal values to allow maximum throughput at minimum CPU utilization, for more intensive, multi-stream scenarios.

Table 15 - Performance Registry Keys

Value Name	Default Value	Description
RecvCompletionMethod	1	Sets the completion methods of the receive packets, and it affects network throughput and CPU utilization. The supported methods are: • Polling - increases the CPU utilization, because the system polls the received rings for incoming packets; however, it may increase the network bandwidth since the incoming packet is handled faster. • Adaptive - combines the interrupt and polling methods dynamically, depending on traffic type and network usage. The valid values are: • 0: polling • 1: adaptive
*InterruptModeration	1	Sets the rate at which the controller moderates or delays the generation of interrupts, making it possible to optimize network throughput and CPU utilization. When disabled, the interrupt moderation of the system generates an interrupt when the packet is received. In this mode, the CPU utilization is increased at higher data rates, because the system must handle a larger number of interrupts. However, the latency is decreased, since that packet is processed more quickly. When interrupt moderation is enabled, the system accumulates interrupts and sends a single interrupt rather than a series of interrupts. The valid values are: • 0: disable • 1: enable
RxIntModeration	2	Sets the rate at which the controller moderates or delays the generation of interrupts, making it possible to optimize network throughput and CPU utilization. The default setting (Adaptive) adjusts the interrupt rates dynamically, depending on traffic type and network usage. Choosing a different setting may improve network and system performance in certain configurations. The valid values are: • 1: static • 2: adaptive The interrupt moderation count and time are configured dynamically, based on traffic types and rate.

Table 15 - Performance Registry Keys

Value Name	Default Value	Description
TxIntModeration	2	Sets the rate at which the controller moderates or delays the generation of interrupts, making it possible to optimize network throughput and CPU utilization. The default setting (Adaptive) adjusts the interrupt rates dynamically, depending on traffic type and network usage. Choosing a different setting may improve network and system performance in certain configurations. The valid values are: • 1: static • 2: adaptive The interrupt moderation count and time are configured dynamically, based on traffic types and rate.
*RSS	1	Sets the driver to use Receive Side Scaling (RSS) mode to improve the performance of handling incoming packets. This mode allows the adapter port to utilize the multiple CPUs in a multi-core system for receiving incoming packets and steering them to their destination. RSS can significantly improve the number of transactions per second, the number of connections per second, and the network throughput. This parameter can be set to one of two values: • 1: enable (default) Sets RSS Mode. • 0: disable The hardware is configured once to use the Toeplitz hash function and the indirection table is never changed.
ThreadPoll	3000	The number of cycles that should be passed without receiving any packet before the polling mechanism stops when using polling completion method for receiving. Afterwards, receiving new packets will generate an interrupt that reschedules the polling mechanism. The valid values are 0 up to 200000. **Note**: This registry value is not exposed via the UI.

Table 15 - Performance Registry Keys

Value Name	Default Value	Description
VlanId	ETH: 0	Enables packets with VlanId. It is used when no team intermediate driver is used. The valid values are: • 0: disable No Vlan Id is passed. • 1-4095 Valid Vlan Id that will be passed. **Note**: This registry value is only valid for Ethernet.
*NumRSSQueues	8	The maximum number of the RSS queues that the device should use. **Note**: This registry key is only in Windows Server 2012 and above.
BlueFlame	1	The latency-critical Send WQEs to the device. When a BlueFlame is used, the WQEs are written directly to the PCI BAR of the device (in addition to memory), so that the device may handle them without having to access memory, thus shortening the execution latency. For best performance, it is recommended to use the BlueFlame when the HCA is lightly loaded. For high-bandwidth scenarios, it is recommended to use regular posting (without BlueFlame). The valid values are: • 0: disable • 1: enable **Note**: This registry value is not exposed via the UI.
*MaxRSSProcessors	8	The maximum number of RSS processors. **Note**: This registry key is only in Windows Server 2012 and above.
AsyncRecieveIndicate	0	Disabled default
	1	Enables packet burst buffering using threaded DPC
	2	Enables packet burst buffering using polling
RfdReservationFactor	150	Controls the number of reserved receive packets,

3.5.5 Ethernet Registry Keys

The following section describes the registry keys that are only relevant to Ethernet driver.

Table 16 - Ethernet Registry Keys

Value Name	Default Value	Description
RoceMaxFrameSize	1024	The maximum size of a frame (or a packet) that can be sent by the RoCE protocol (a.k.a Maximum Transmission Unit (MTU). Using larger RoCE MTU will improve the performance; however, one must ensure that the entire system, including switches, supports the defined MTU. Ethernet packet uses the general MTU value, whereas the RoCE packet uses the RoCE MTU The valid values are: • 256 • 512 • 1024 • 2048 **Note**: This registry key is supported only in Ethernet drivers.
*PriorityVLANTag	3 (Packet Priority & VLAN Enabled)	Enables sending and receiving IEEE 802.3ac tagged frames, which include: • 802.1p QoS (Quality of Service) tags for priority-tagged packets. • 802.1Q tags for VLANs. When this feature is enabled, the Mellanox driver supports sending and receiving a packet with VLAN and QoS tag.
DeviceRxStallTimeout	1000	The maximum period for a single received packet processing. If the packet was not processed during this time, the device will be declared as stalled and will increase the "Critical Stall Watermark Reached" counter. The value is given in mSec. The maximum period is 8000 mSec. the special value of 0, indicates that the DeviceRxStallTimeout is inactive.
DeviceRxStallWatermark	0	The maximum period for a single received packet processing. If the packet was not processed during this time, the device will increase a diagnostic counter called "Minor Stall Watermark Reached”. The value is given in mSec. The maximum period is 8000 mSec. The special value of 0, indicates that the DeviceRxStallWatermark is inactive
TCHeadOfQueueLifeTimeLimit	0-20 Default: 19	The time a packet can live at the head of a TC queue before it is discarded. The timeout value is defined by 4,096us multiplied by 2^TCHeadOfQueueLifetimeLimit.

Table 16 - Ethernet Registry Keys

Value Name	Default Value	Description
TCStallCount	0-7 0: Disabled	The number of sequential packets dropped due to Head Of Queue Lifetime Limit, in order for the port to enter the TCStalled state. All packets for the TC are discarded in this state for a period of 8 times the timeout defined by TCHead-OfQueueLifetimeLimit.
TCHead-OfQueueLifeTime-LimitEnable	0	The TCs for which Head Of Queue Lifetime Limit is enabled. Bit 0 represents TC0, bit 1 represents TC1 and so on. The valid values are: • 0-255 • 0: disabled

3.5.5.1 Flow Control Options

This group of registry keys allows the administrator to control the TCP/IP traffic by pausing frame transmitting and/or receiving operations. By enabling the Flow Control mechanism, the adapters can overcome any TCP/IP issues and eliminate the risk of data loss.

Table 17 - Flow Control Options

Value Name	Default Value	Description
*FlowControl	3	When Rx Pause is enabled, the receiving adapter generates a flow control frame when its received queue reaches a pre-defined limit. The flow control frame is sent to the sending adapter. When TX Pause is enabled, the sending adapter pauses the transmission if it receives a flow control frame from a link partner. The valid values are: • 0: Flow control is disabled • 1: Tx Flow control is Enabled • 2: Rx Flow control is enabled • 3: Rx & Tx Flow control is enabled

3.5.5.2 VMQ Options

This section describes the registry keys that are used to control the NDIS Virtual Machine Queue (VMQ). VMQ is supported by WinOF-2 and allows a performance boost for Hyper-V VMs.

For more details about VMQ please refer to Microsoft web site,
http://msdn.microsoft.com/en-us/library/windows/hardware/ff571034(v=vs.85).aspx

Table 18 - VMQ Options

Value Name	Default Value	Description
*VMQ	1	The support for the virtual machine queue (VMQ) features of the network adapter. The valid values are: • 1: enable • 0: disable
*RssOrVmqPreference	0	Specifies whether VMQ capabilities should be enabled instead of receive-side scaling (RSS) capabilities. The valid values are: • 0: Report RSS capabilities • 1: Report VMQ capabilities **Note**: This registry value is not exposed via the UI.
*VMQVlanFiltering	1	Specifies whether the device enables or disables the ability to filter network packets by using the VLAN identifier in the media access control (MAC) header. The valid values are: • 0: disable • 1: enable

3.5.5.3 RoCE Options

This section describes the registry keys that are used to control RoCE mode.

Table 19 - RoCE Options

Value Name	Default Value	Description
roce_mode	0 - RoCE	The RoCE mode. The valid values are: • 0 - RoCE • 4 - No RoCE **Note**: The default value depends on the WinOF package used.

3.5.5.4 SR-IOV Options

This section describes the registry keys that are used to control the NDIS Single Root I/O Virtualization (SR-IOV). The SR-IOV is supported by WinOF-2, and allows a performance boost for Hyper-V VMs.

For more details about the VMQ, please see Single Root I/O Virtualization (SR-IOV) on Microsoft website.

Table 20 - SR-IOV Options

Value Name	Default Value	Description
SRIOV	1	The support for the SR-IOV features of the network adapter. The valid values are: • 1: enable • 0: disable
SriovPreferred	N/A (hidden)	A value that defines whether SR-IOV capabilities should be enabled instead of the virtual machine queue (VMQ), or receive side scaling (RSS) capabilities.
eSWUCLoopback	N/A	When set to 1, the Unicast traffic Loopback is disabled.
eSWMCLoopback	N/A	When set to 1, the Multicast traffic Loopback is disabled.
MaxFWPagesUsage-PerVF	250000	This key sets the limitation for the maximum number of 4KB pages that the host could allocate for VFs resources. When set to 0, limitation is disabled.

3.6 Performance Tuning and Counters

For further information on WinOF-2 performance, please refer to the Performance Tuning Guide for Mellanox Network Adapters.

This section describes how to modify Windows registry parameters in order to improve performance.

Please note that modifying the registry incorrectly might lead to serious problems, including the loss of data, system hang, and you may need to reinstall Windows. As such it is recommended to back up the registry on your system before implementing recommendations included in this section. If the modifications you apply lead to serious problems, you will be able to restore the original registry state. For more details about backing up and restoring the registry, please visit www.microsoft.com.

3.6.1 General Performance Optimization and Tuning

To achieve the best performance for Windows, you may need to modify some of the Windows registries.

3.6.1.1 Registry Tuning

The registry entries that may be added/changed by this “General Tuning” procedure are:

Under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters:

- Disable TCP selective acks option for better cpu utilization:

```
SackOpts, type REG_DWORD, value set to 0.
```

Under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AFD\Parameters:

- Enable fast datagram sending for UDP traffic:

```
FastSendDatagramThreshold, type REG_DWORD, value set to 64K.
```

Under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Ndis\Parameters:

- Set RSS parameters:

```
RssBaseCpu, type REG_DWORD, value set to 1.
```

3.6.1.2 Enable RSS

Enabling Receive Side Scaling (RSS) is performed by means of the following command:

```
"netsh int tcp set global rss = enabled"
```

3.6.1.3 Improving Live Migration

In order to improve live migration over SMB direct performance, please set the following registry key to 0 and reboot the machine:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\LanmanServer\Parameters\RequireSecuritySig-
nature
```

3.6.2 Application Specific Optimization and Tuning

3.6.2.1 Ethernet Performance Tuning

The user can configure the Ethernet adapter by setting some registry keys. The registry keys may affect Ethernet performance.

➢ ***To improve performance, activate the performance tuning tool as follows:***

Step 1. Start the "Device Manager" (open a command line window and enter: devmgmt.msc).

Step 2. Open "Network Adapters".

Step 3. Right click the relevant Ethernet adapter and select Properties.

Step 4. Select the "Advanced" tab

Step 5. Modify performance parameters (properties) as desired.

3.6.2.1.1 Performance Known Issues

- On Intel I/OAT supported systems, it is highly recommended to install and enable the latest I/OAT driver (download from www.intel.com).
- With I/OAT enabled, sending 256-byte messages or larger will activate I/OAT. This will cause a significant latency increase due to I/OAT algorithms. On the other hand, throughput will increase significantly when using I/OAT.

3.6.2.2 Ethernet Bandwidth Improvements

➢ ***To improve Ethernet Bandwidth:***

Step 1. Check you are running on the closest NUMA.

Step a. In the PowerShell run: `Get-NetAdapterRss -Name "adapter name"`

```
PS C:\Users\herod> Get-NetAdapterrss -Name "ethernet 6"


Name                                           : Ethernet 6
InterfaceDescription                           : Mellanox ConnectX-4 Lx Ethernet Adapter
Enabled                                        : True
NumberOfReceiveQueues                          : 8
Profile                                        : Closest
BaseProcessor: [Group:Number]                  : 0:0
MaxProcessor: [Group:Number]                   : 0:27
MaxProcessors                                  : 8
RssProcessorArray: [Group:Number/NUMA Distance] : 0:0/0  0:1/0  0:2/0  0:3/0  0:4/0  0:5/0  0:6/0  0:7/0
                                                  0:8/0  0:9/0  0:10/0  0:11/0  0:12/0  0:13/0  0:14/32767  0:15/32767
                                                  0:16/32767  0:17/32767  0:18/32767  0:19/32767  0:20/32767
                                                  0:21/32767  0:22/32767  0:23/32767
                                                  0:24/32767  0:25/32767  0:26/32767  0:27/32767
IndirectionTable: [Group:Number]               : 0:0    0:1    0:2    0:3    0:4    0:5    0:6    0:7
                                                  0:0    0:1    0:2    0:3    0:4    0:5    0:6    0:7
                                                  0:0    0:1    0:2    0:3    0:4    0:5    0:6    0:7
                                                  0:0    0:1    0:2    0:3    0:4    0:5    0:6    0:7
                                                  0:0    0:1    0:2    0:3    0:4    0:5    0:6    0:7
                                                  0:0    0:1    0:2    0:3    0:4    0:5    0:6    0:7
                                                  0:0    0:1    0:2    0:3    0:4    0:5    0:6    0:7
                                                  0:0    0:1    0:2    0:3    0:4    0:5    0:6    0:7
                                                  0:0    0:1    0:2    0:3    0:4    0:5    0:6    0:7
                                                  0:0    0:1    0:2    0:3    0:4    0:5    0:6    0:7
                                                  0:0    0:1    0:2    0:3    0:4    0:5    0:6    0:7
                                                  0:0    0:1    0:2    0:3    0:4    0:5    0:6    0:7
                                                  0:0    0:1    0:2    0:3    0:4    0:5    0:6    0:7
                                                  0:0    0:1    0:2    0:3    0:4    0:5    0:6    0:7
                                                  0:0    0:1    0:2    0:3    0:4    0:5    0:6    0:7
                                                  0:0    0:1    0:2    0:3    0:4    0:5    0:6    0:7
```

Step b. Validate that the `IndirectionTable` CPUs are located at the closest NUMA.
As illustrated in the figure above, the CPUs are 0:0 - 0:7, CPU 0 -7 and the distance from the NUMA is 0, 0:0/0 - 0:7/0, unlike CPU 14-27/32767.

Step c. If the CPUs are not close to the NUMEA, change the `"RSS Base Processor Number"` and `"RSS Max Processor Number"` settings under the Advance tab to point to the closest CPUs.

For high performance, it is recommended to work with at least 8 processors.

Step 2. Check the Ethernet bandwidth, run ntttcp.exe.

- **Server side:** ntttcp -r -m 32,*,server_ip
- **Client side:** ntttcp -s -m 32,*,server_ip

3.6.2.3 IPoIB Performance Tuning

The user can configure the IPoIB adapter by setting some registry keys. The registry keys may affect IPoIB performance.

For the complete list of registry entries that may be added/changed by the performance tuning procedure, see MLNX_VPI_WinOF Registry Keys following the path below:

http://www.mellanox.com/page/products_dyn?product_family=32&mtag=windows_sw_drivers

➢ ***To improve performance, activate the performance tuning tool as follows:***

Step 1. Start the "Device Manager" (open a command line window and enter: devmgmt.msc).

Step 2. Open "Network Adapters".

Step 3. Right click the relevant IPoIB adapter and select Properties.

Step 4. Select the "Advanced" tab

Step 5. Modify performance parameters (properties) as desired.

3.6.3 Tunable Performance Parameters

The following is a list of key parameters for performance tuning.

- **Jumbo Packet**

  The maximum available size of the transfer unit, also known as the Maximum Transmission Unit (MTU). The MTU of a network can have a substantial impact on performance. A 4K MTU size improves performance for short messages, since it allows the OS to coalesce many small messages into a large one.

  - Valid MTU values range for an Ethernet driver is between 614 and 9614.

> All devices on the same physical network, or on the same logical network, must have the same MTU.

- **Receive Buffers**

  The number of receive buffers (default 512).

- **Send Buffers**

  The number of sent buffers (default 2048).

- **Performance Options**

  Configures parameters that can improve adapter performance.

  - Interrupt Moderation

    Moderates or delays the interrupts' generation. Hence, optimizes network throughput and CPU utilization (default Enabled).

    - When the interrupt moderation is enabled, the system accumulates interrupts and sends a single interrupt rather than a series of interrupts. An interrupt is generated after receiving 5 packets or after 10ms from the first packet received. It improves performance and reduces CPU load however, it increases latency.
    - When the interrupt moderation is disabled, the system generates an interrupt each time a packet is received or sent. In this mode, the CPU utilization data rates increase, as the system handles a larger number of interrupts. However, the latency decreases as the packet is handled faster.

  - Receive Side Scaling (RSS Mode)

    Improves incoming packet processing performance. RSS enables the adapter port to utilize the multiple CPUs in a multi-core system for receiving incoming packets and steering them to the designated destination. RSS can significantly improve the number of transactions, the number of connections per second, and the network throughput.

    This parameter can be set to one of the following values:

    - Enabled (default): Set RSS Mode

- Disabled: The hardware is configured once to use the Toeplitz hash function, and the indirection table is never changed.

> IOAT is not used while in RSS mode.

- Receive Completion Method

  Sets the completion methods of the received packets, and can affect network throughput and CPU utilization.

  - **Polling Method**

    Increases the CPU utilization as the system polls the received rings for the incoming packets. However, it may increase the network performance as the incoming packet is handled faster.

  - **Adaptive (Default Settings)**

    A combination of the interrupt and polling methods dynamically, depending on traffic type and network usage. Choosing a different setting may improve network and/or system performance in certain configurations.

- Rx Interrupt Moderation Type

  Sets the rate at which the controller moderates or delays the generation of interrupts making it possible to optimize network throughput and CPU utilization. The default setting (Adaptive) adjusts the interrupt rates dynamically depending on the traffic type and network usage. Choosing a different setting may improve network and system performance in certain configurations.

- Send Completion Method

  Sets the completion methods of the Send packets and it may affect network throughput and CPU utilization.

- **Offload Options**

  Allows you to specify which TCP/IP offload settings are handled by the adapter rather than the operating system.

  Enabling offloading services increases transmission performance as the offload tasks are performed by the adapter hardware rather than the operating system. Thus, freeing CPU resources to work on other tasks.

  - IPv4 Checksums Offload

    Enables the adapter to compute IPv4 checksum upon transmit and/or receive instead of the CPU (default Enabled).

  - TCP/UDP Checksum Offload for IPv4 packets

    Enables the adapter to compute TCP/UDP checksum over IPv4 packets upon transmit and/or receive instead of the CPU (default Enabled).

  - TCP/UDP Checksum Offload for IPv6 packets

    Enables the adapter to compute TCP/UDP checksum over IPv6 packets upon transmit and/or receive instead of the CPU (default Enabled).

  - Large Send Offload (LSO)

Allows the TCP stack to build a TCP message up to 64KB long and sends it in one call down the stack. The adapter then re-segments the message into multiple TCP packets for transmission on the wire with each pack sized according to the MTU. This option offloads a large amount of kernel processing time from the host CPU to the adapter.

3.6.4 Adapter Proprietary Performance Counters

Proprietary Performance Counters are used to provide information on Operating System, application, service or the drivers' performance. Counters can be used for different system debugging purposes, help to determine system bottlenecks and fine-tune system and application performance. The Operating System, network, and devices provide counter data that the application can consume to provide users with a graphical view of the system's performance quality. WinOF-2 counters hold the standard Windows CounterSet API that includes:

- Network Interface
- RDMA activity
- SMB Direct Connection

3.6.4.1 Supported Standard Performance Counters

3.6.4.1.1 Proprietary Mellanox WinOF-2 Port Traffic Counters

Proprietary Mellanox WinOF-2 port traffic counters set consists of global traffic statistics which gather information from ConnectX®-4 and ConnectX®-4 Lx network adapters, and includes traffic statistics, and various types of error and indications from both the Physical Function and Virtual Function.

Table 21 - Mellanox WinOF-2 Port Traffic Counters

Mellanox Adapter Traffic Counters	Description
Bytes IN	
Bytes Received	Shows the number of bytes received by the adapter. The counted bytes include framing characters.
Bytes Received/Sec	Shows the rate at which bytes are received by the adapter. The counted bytes include framing characters.
Packets Received	Shows the number of packets received by the network interface.
Packets Received/Sec	Shows the rate at which packets are received by the network interface.
Bytes/ Packets OUT	
Bytes Sent	Shows the number of bytes sent by the adapter. The counted bytes include framing characters.
Bytes Sent/Sec	Shows the rate at which bytes are sent by the adapter. The counted bytes include framing characters.
Packets Sent	Shows the number of packets sent by the network interface.

Table 21 - Mellanox WinOF-2 Port Traffic Counters

Mellanox Adapter Traffic Counters	Description
Packets Sent/Sec	Shows the rate at which packets are sent by the network interface.
Bytes' TOTAL	
Bytes Total	Shows the total of bytes handled by the adapter. The counted bytes include framing characters.
Bytes Total/Sec	Shows the total rate of bytes that are sent and received by the adapter. The counted bytes include framing characters.
Packets Total	Shows the total of packets handled by the network interface.
Packets Total/Sec	Shows the rate at which packets are sent and received by the network interface.
Control Packets	The total number of successfully received control frames.[a]
ERRORS, DROP, AND MISC. INDICATIONS	
Packets Outbound Errors[b]	Shows the number of outbound packets that could not be transmitted because of errors found in the physical layer.[a]
Packets Outbound Discarded[a]	Shows the number of outbound packets to be discarded in the physical layer, even though no errors had been detected to prevent transmission. One possible reason for discarding packets could be to free up some buffer space.
Packets Received Errors[a]	Shows the number of inbound packets that contained errors in the physical layer, preventing them from being deliverable.
Packets Received with Frame Length Error	Shows the number of inbound packets that contained error where the frame has length error. Packets received with frame length error are a subset of packets received errors.[a]
Packets Received with Symbol Error	Shows the number of inbound packets that contained symbol error or an invalid block. Packets received with symbol error are a subset of packets received errors.
Packets Received with Bad CRC Error	Shows the number of inbound packets that failed the CRC check. Packets received with bad CRC error are a subset of packets received errors.
Packets Received Discarded[a]	Shows the number of inbound packets that were chosen to be discarded in the physical layer, even though no errors had been detected to prevent their being deliverable. One possible reason for discarding such a packet could be a buffer overflow.
Receive Segment Coalescing (RSC)	
RSC Aborts	Number of RSC abort events. That is, the number of exceptions other than the IP datagram length being exceeded. This includes the cases where a packet is not coalesced because of insufficient hardware resources.[a]

Table 21 - Mellanox WinOF-2 Port Traffic Counters

Mellanox Adapter Traffic Counters	Description
RSC Coalesced Events	Number of RSC coalesced events. That is, the total number of packets that were formed from coalescing packets.[a]
RSC Coalesced Octets	Number of RSC coalesced bytes.[a]
RSC Coalesced Packets	Number of RSC coalesced packets.[a]
RSC Average Packet Size	RSC Average Packet Size is the average size in bytes of received packets across all TCP connections.[a]

a. This counter is relevant only for ETH ports.
b. Those error/discard counters are related to layer-2 issues, such as CRC, length, and type errors. There is a possibility of an error/discard in the higher interface level. For example, a packet can be discarded for the lack of a receive buffer. To see the sum of all error/discard packets, read the Windows Network-Interface Counters. Note that for IPoIB, the Mellanox counters are for IB layer-2 issues only, and Windows Network-Interface counters are for interface level issues.

3.6.4.1.2 Mellanox WinOF-2 Virtual Function (VF) Port Traffic Counters

Mellanox WinOF-2 VF Port Traffic set consists of counters that measure the rates at which bytes and packets are sent and received over a virtual port network connection that is bound to a virtual PCI function. It includes counters that monitor connection errors.

This set is available only on hypervisors and not on virtual network adapters.

This counters set is relevant only for ETH ports.

Table 22 - Mellanox WinOF-2 Virtual Function (VF) Port Traffic Counters

Mellanox WinOF-2 VF Port Traffic Counters	Description
Bytes/Packets IN	
Bytes Received/Sec	Shows the rate at which bytes are received over each network VPort. The counted bytes include framing characters.
Bytes Received Unicast/Sec	Shows the rate at which subnet-unicast bytes are delivered to a higher-layer protocol.
Bytes Received Broadcast/Sec	Shows the rate at which subnet-broadcast bytes are delivered to a higher-layer protocol.
Bytes Received Multicast/Sec	Shows the rate at which subnet-multicast bytes are delivered to a higher-layer protocol.

Table 22 - Mellanox WinOF-2 Virtual Function (VF) Port Traffic Counters

Mellanox WinOF-2 VF Port Traffic Counters	Description
Packets Received Unicast/Sec	Shows the rate at which subnet-unicast packets are delivered to a higher-layer protocol.
Packets Received Broadcast/Sec	Shows the rate at which subnet-broadcast packets are delivered to a higher-layer protocol.
Packets Received Multicast/Sec	Shows the rate at which subnet-multicast packets are delivered to a higher-layer protocol.
Bytes/Packets IN	
Bytes Sent/Sec	Shows the rate at which bytes are sent over each network VPort. The counted bytes include framing characters.
Bytes Sent Unicast/Sec	Shows the rate at which bytes are requested to be transmitted to subnet-unicast addresses by higher-level protocols. The rate includes the bytes that were discarded or not sent.
Bytes Sent Broadcast/Sec	Shows the rate at which bytes are requested to be transmitted to subnet-broadcast addresses by higher-level protocols. The rate includes the bytes that were discarded or not sent.
Bytes Sent Multicast/Sec	Shows the rate at which bytes are requested to be transmitted to subnet-multicast addresses by higher-level protocols. The rate includes the bytes that were discarded or not sent.
Packets Sent Unicast/Sec	Shows the rate at which packets are requested to be transmitted to subnet-unicast addresses by higher-level protocols. The rate includes the packets that were discarded or not sent.
Packets Sent Broadcast/Sec	Shows the rate at which packets are requested to be transmitted to subnet-broadcast addresses by higher-level protocols. The rate includes the packets that were discarded or not sent.
Packets Sent Multicast/Sec	Shows the rate at which packets are requested to be transmitted to subnet-multicast addresses by higher-level protocols. The rate includes the packets that were discarded or not sent.
ERRORS, DISCARDED	
Packets Outbound Discarded	Shows the number of outbound packets to be discarded even though no errors had been detected to prevent transmission. One possible reason for discarding a packet could be to free up buffer space.

Table 22 - Mellanox WinOF-2 Virtual Function (VF) Port Traffic Counters

Mellanox WinOF-2 VF Port Traffic Counters	Description
Packets Outbound Errors	Shows the number of outbound packets that could not be transmitted because of errors.
Packets Received Discarded	Shows the number of inbound packets that were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space.
Packets Received Errors	Shows the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.

3.6.4.1.3 Proprietary Mellanox WinOF-2 Port QoS Counters

Proprietary Mellanox WinOF-2 Port QoS counters set consists of flow statistics per (VLAN) priority. Each QoS policy is associated with a priority. The counter presents the priority's traffic, pause statistic.

This counters set is relevant only for ETH ports.

Table 23 - Mellanox WinOF-2 Port QoS Counters

Mellanox Qos Counters	Description
Bytes/Packets IN	
Bytes Received	The number of bytes received that are covered by this priority. The counted bytes include framing characters (modulo 2^64).
Bytes Received/Sec	The number of bytes received per second that are covered by this priority. The counted bytes include framing characters.
Packets Received	The number of packets received that are covered by this priority (modulo 2^64).
Packets Received/Sec	The number of packets received per second that are covered by this priority.
Bytes/Packets OUT	
Bytes Sent	The number of bytes sent that are covered by this priority. The counted bytes include framing characters (modulo 2^64).

Table 23 - Mellanox WinOF-2 Port QoS Counters

Mellanox Qos Counters	Description
Bytes Sent/Sec	The number of bytes sent per second that are covered by this priority. The counted bytes include framing characters.
Packets Sent	The number of packets sent that are covered by this priority (modulo 2^64).
Packets Sent/Sec	The number of packets sent per second that are covered by this priority.
Bytes and Packets Total	
Bytes Total	The total number of bytes that are covered by this priority. The counted bytes include framing characters (modulo 2^64).
Bytes Total/Sec	The total number of bytes per second that are covered by this priority. The counted bytes include framing characters.
Packets Total	The total number of packets that are covered by this priority (modulo 2^64).
Packets Total/Sec	The total number of packets per second that are covered by this priority.
PAUSE INDICATION	
Sent Pause Frames	The total number of pause frames sent from this priority to the far-end port. The untagged instance indicates the number of global pause frames that were sent.
Sent Pause Duration	The total duration of packets transmission being paused on this priority in microseconds.
Received Pause Frames	The number of pause frames that were received to this priority from the far-end port. The untagged instance indicates the number of global pause frames that were received.
Received Pause Duration	The total duration that far-end port was requested to pause for the transmission of packets in microseconds.
Sent Discard Frames	The number of packets discarded by the transmitter. **Note**: this counter is per TC and not per priority.

3.6.4.1.4 RDMA Activity Counters

RDMA Activity counter set consists of NDK performance counters. These performance counters allow you to track Network Direct Kernel (RDMA) activity, including traffic rates, errors, and control plane activity.

Table 24 - RDMA Activity Counters

RDMA Activity Counters	Description
RDMA Accepted Connections	The number of inbound RDMA connections established.
RDMA Active Connections	The number of active RDMA connections.
RDMA Completion Queue Errors	This counter is not supported, and always is set to zero.
RDMA Connection Errors	The number of established connections with an error before a consumer disconnected the connection.
RDMA Failed Connection Attempts	The number of inbound and outbound RDMA connection attempts that failed.
RDMA Inbound Bytes/sec	The number of bytes for all incoming RDMA traffic. This includes additional layer two protocol overhead.
RDMA Inbound Frames/sec	The number, in frames, of layer two frames that carry incoming RDMA traffic.
RDMA Initiated Connections	The number of outbound connections established.
RDMA Outbound Bytes/sec	The number of bytes for all outgoing RDMA traffic. This includes additional layer two protocol overhead.
RDMA Outbound Frames/sec	The number, in frames, of layer two frames that carry outgoing RDMA traffic.

3.6.4.1.5 Mellanox WinOF-2 Congestion Control Counters

Mellanox WinOF-2 Congestion Control counters set consists of counters that measure the DCQCN statistics over the network adapter.

Table 25 - Congestion Control Counters

Congestion Control Counters	Description
Notification Point	
Notification Point – CNPs Sent Successfully	Number of congestion notification packets (CNPs) successfully sent by the notification point.
Notification Point – RoCEv2 DCQCN Marked Packets	Number of RoCEv2 packets that were marked as congestion encountered.
Reaction Point	
Reaction Point – Current Number of Flows	Current number of Rate Limited Flows due to RoCEv2 Congestion Control.

Table 25 - Congestion Control Counters

Reaction Point – Ignored CNP Packets	Number of ignored congestion notification packets (CNPs).
Reaction Point – Successfully Handled CNP Packets	Number of congestion notification packets (CNPs) received and handled successfully.

3.6.4.1.6 Mellanox WinOF-2 Diagnostics Counters

Mellanox WinOF-2 diagnostics counters set consists of the following counters:

Table 26 - WinOF-2 Diagnostics Counters

Mellanox WinOF-2 Diagnostics Counters	Description
Reset Requests	Number of resets requested by NDIS.
Link State Change Events	Number of link status updates received from the hardware.
Queued Send Packets	Number of send packets pending transmission due to hardware queues overflow.
Send Completions in Passive/Sec	Number of send completion events handled in passive mode per second.
Receive Completions in Passive/Sec	Number of receive completion events handled in passive mode per second.
Copied Send Packets	Number of send packets that were copied in slow path.
Correct Checksum Packets In Slow Path	Number of receive packets that required the driver to perform the checksum calculation and resulted in success.
Bad Checksum Packets In Slow Path	Number of receive packets that required the driver to perform checksum calculation and resulted in failure.
Undetermined Checksum Packets In Slow Path	Number of receive packets with undetermined checksum result.
Watch Dog Expired/Sec	Number of watch dogs expired per second.
Requester time out received	Number of time out received when the local machine generates outbound traffic.
Requester out of order sequence NAK	Number of Out of Sequence NAK received when the local machine generates outbound traffic, i.e. the number of times the local machine received NAKs indicating OOS on the receiving side.
Requester RNR NAK	Number of RNR (Receiver Not Ready) NAKs received when the local machine generates outbound traffic.
Responder RNR NAK	Number of RNR (Receiver Not Ready) NAKs sent when the local machine receives inbound traffic.

Table 26 - WinOF-2 Diagnostics Counters

Mellanox WinOF-2 Diagnostics Counters	Description
Responder out of order sequence received	Number of Out of Sequence packets received when the local machine receives inbound traffic, i.e. the number of times the local machine received messages that are not consecutive.
Responder duplicate request received	Number of duplicate requests received when the local machine receives inbound traffic.
Requester RNR NAK retries exceeded errors	Number of RNR (Receiver Not Ready) NAKs retries exceeded errors when the local machine generates out-bound traffic.

3.6.4.1.7 Mellanox WinOF-2 Device Diagnostic Counters

Mellanox WinOF-2 device diagnostic counters set consists of the following counters:.

Table 27 - Device Diagnostics Counters

Mellanox WinOF-2 Device Diagnostic Counters	Description
L0 MTT miss	The number of access to L0 MTT that were missed
L0 MTT miss/Sec	The rate of access to L0 MTT that were missed
L0 MTT hit	The number of access to L0 MTT that were hit
L0 MTT hit/Sec	The rate of access to L0 MTT that were hit
L1 MTT miss	The number of access to L1 MTT that were missed
L1 MTT miss/Sec	The rate of access to L1 MTT that were missed
L1 MTT hit	The number of access to L1 MTT that were hit
L1 MTT hit/Sec	The rate of access to L1 MTT that were hit
L0 MPT miss	The number of access to L0 MKey that were missed
L0 MPT miss/Sec	The rate of access to L0 MKey that were missed
L0 MPT hit	The number of access to L0 MKey that were hit
L0 MPT hit/Sec	The rate of access to L0 MKey that were hit
L1 MPT miss	The number of access to L1 MKey that were missed
L1 MPT miss/Sec	The rate of access to L1 MKey that were missed
L1 MPT hit	The number of access to L1 MKey that were hit
L1 MPT hit/Sec	The rate of access to L1 MKey that were hit
RXS no slow path credis	No room in RXS for slow path packets
RXS no fast path credis	No room in RXS for fast path packets
RXT no slow path credis	No room in RXT for slow path packets
RXT no fast path credis	No room in RXT for fast path packets

Table 27 - Device Diagnostics Counters

Mellanox WinOF-2 Device Diagnostic Counters	Description
Slow path packets slice load	Number of slow path packets loaded to HCA as slices from the network
Fast path packets slice load	Number of fast path packets loaded to HCA as slices from the network
Steering pipe 0 processing time	Number of clocks that steering pipe 0 worked
Steering pipe 1 processing time	Number of clocks that steering pipe 1 worked
WQE address translation back-pressure	No credits between RXW and TPT
Receive WQE cache miss	Number of packets that got miss in RWqe buffer L0 cache
Receive WQE cache hit	Number of packets that got hit in RWqe buffer L0 cache
Slow packets miss in LDB L1 cache	Number of slow packet that got miss in LDB L1 cache
Slow packets hit in LDB L1 cache	Number of slow packet that got hit in LDB L1 cache
Fast packets miss in LDB L1 cache	Number of fast packet that got miss in LDB L1 cache
Fast packets hit in LDB L1 cache	Number of fast packet that got hit in LDB L1 cache
Packets miss in LDB L2 cache	Number of packet that got miss in LDB L2 cache
Packets hit in LDB L2 cache	Number of packet that got hit in LDB L2 cache
Slow packets miss in REQSL L1	Number of slow packet that got miss in REQSL L1 fast cache
Slow packets hit in REQSL L1	Number of slow packet that got hit in REQSL L1 fast cache
Fast packets miss in REQSL L1	Number of fast packet that got miss in REQSL L1 fast cache
Fast packets hit in REQSL L1	Number of fast packet that got hit in REQSL L1 fast cache
Packets miss in REQSL L2	Number of packet that got miss in REQSL L2 fast cache
Packets hit in REQSL L2	Number of packet that got hit in REQSL L2 fast cache
No PXT credits time	Number of clocks in which there were no PXT credits
EQ slices busy time	Number of clocks where all EQ slices were busy
CQ slices busy time	Number of clocks where all CQ slices were busy
MSIX slices busy time	Number of clocks where all MSIX slices were busy
QP done due to VL limited	Number of QP done scheduling due to VL limited (e.g. lack of VL credits)

Table 27 - Device Diagnostics Counters

Mellanox WinOF-2 Device Diagnostic Counters	Description
QP done due to desched	Number of QP done scheduling due to desched (Tx full burst size)
QP done due to work done	Number of QP done scheduling due to work done (Tx all QP data)
QP done due to limited	Number of QP done scheduling due to limited (e.g. max read)
QP done due to E2E creadits	Number of QP done scheduling due to e2e credits (other peer credits)
Packets sent by SXW to SXP	Number of packets that were authorized to send by SXW (to SXP)
Steering hit	Number of steering lookups that were hit
Steering miss	Number of steering lookups that were miss
Steering processing time	Number of clocks that steering pipe worked
No send credits for scheduling time	The number of clocks that were no credits for scheduling (Tx)
No slow path send credits for scheduling time	The number of clocks that were no credits for scheduling (Tx) for slow path
TPT indirect memory key access	The number of indirect mkey accesses

3.6.4.1.8 Mellanox WinOF-2 PCI Device Diagnostic Counters

Mellanox WinOF-2 PCI device diagnostic counters set consists of the following counters:

Table 28 - PCI Device Diagnostic Counters

Mellanox WinOF-2 PCI Device Diagnostic Counters	Description
PCI back-pressure cycles	The number of clocks where BP was received from the PCI, while trying to send a packet to the host.
PCI back-pressure cycles/Sec	The rate of clocks where BP was received from the PCI, while trying to send a packet to the host.
PCI write back-pressure cycles	The number of clocks where there was lack of posted outbound credits from the PCI, while trying to send a packet to the host.
PCI write back-pressure cycles/Sec	The rate of clocks where there was lack of posted outbound credits from the PCI, while trying to send a packet to the host.
PCI read back-pressure cycles	The number of clocks where there was lack of non-posted outbound credits from the PCI, while trying to send a packet to the host.

Table 28 - PCI Device Diagnostic Counters

Mellanox WinOF-2 PCI Device Diagnostic Counters	Description
PCI read back-pressure cycles/Sec	The rate of clocks where there was lack of non-posted outbound credits from the PCI, while trying to send a packet to the host.
PCI read stuck no receive buffer	The number of clocks where there was lack in global byte credits for non-posted outbound from the PCI, while trying to send a packet to the host.
Available PCI BW	The number of 128 bytes that are available by the host.
Used PCI BW	The number of 128 bytes that were received from the host.
RX PCI errors	The number of physical layer PCIe signal integrity errors. The number of transitions to recovery due to Framing errors and CRC (dlp and tlp). If the counter is advancing, try to change the PCIe slot in use. **Note:** Only a continues increment of the counter value is considered an error.
TX PCI errors	The number of physical layer PCIe signal integrity errors. The number of transition to recovery initiated by the other side (moving to Recovery due to getting TS/EIEOS). If the counter is advancing, try to change the PCIe slot in use. **Note:** transitions to recovery can happen during initial machine boot. The counter should not increment after boot. **Note:** Only a continues increment of the counter value is considered an error.
TX PCI non-fatal errors	The number of PCI transport layer Non-Fatal error msg sent. If the counter is advancing, try to change the PCIe slot in use.
TX PCI fatal errors	The number of PCIe transport layer fatal error msg sent. If the counter is advancing, try to change the PCIe slot in use.

3.6.4.1.9 Mellanox WinOF-2 Hardware RSS Diagnostic Counters

Mellanox WinOF-2 hardware RSS diagnostic counters set provides monitoring for hardware RSS behavior. These counters are accumulative and collect packets per type (IPv4 or IPv6 only, IPv4/6 TCP or UDP), for tunneled and non-tunneled traffic separately, and when the hardware RSS is functional or dysfunctional.

The counters are activated upon first addition into perfmon, and are stopped upon removal.

Setting "RssCountersActivatedAtStartup" registry key to 1 in the NIC properties will cause the RSS counters to collect data from the startup of the device.

All RSS counters are provided under the counter set “Mellanox Adapter RSS Counters”

Each Ethernet adapter provides multiple instances:

- Instance per vPort per CPU in HwRSS mode is formatted: <NetworkAdapter> + vPort_<id> CPU_<cpu>
- Instance per network adapter per CPU in native RSS per CPU is formatted: <Network-Adapter> CPU_<cpu> .

Table 29 - RSS Diagnostic Counters

Mellanox WinOF-2 RSS Diagnostic Counters	Description
Rss IPv4 Only	Shows the number of received packets that have RSS hash calculated on IPv4 header only
Rss IPv4/TCP	Shows the number of received packets that have RSS hash calculated on IPv4 and TCP headers
Rss IPv4/UDP	Shows the number of received packets that have RSS hash calculated on IPv4 and UDP headers
Rss IPv6 Only	Shows the number of received packets that have RSS hash calculated on IPv6 header only
Rss IPv6/TCP	Shows the number of received packets that have RSS hash calculated on IPv6 and TCP headers
Rss IPv6/UDP	Shows the number of received packets that have RSS hash calculated on IPv6 and UDP headers
Encapsulated Rss IPv4 Only	Shows the number of received encapsulated packets that have RSS hash calculated on IPv4 header only
Encapsulated Rss IPv4/TCP	Shows the number of received encapsulated packets that have RSS hash calculated on IPv4 and TCP headers
Encapsulated Rss IPv4/UDP	Shows the number of received encapsulated packets that have RSS hash calculated on IPv4 and UDP headers
Encapsulated Rss IPv6 Only	Shows the number of received encapsulated packets that have RSS hash calculated on IPv6 header only
Encapsulated Rss IPv6/TCP	Shows the number of received encapsulated packets that have RSS hash calculated on IPv6 and TCP headers
Encapsulated Rss IPv6/UDP	Shows the number of received encapsulated packets that have RSS hash calculated on IPv6 and UDP headers
NonRss IPv4 Only	Shows the number of IPv4 packets that have no RSS hash calculated by the hardware
NonRss IPv4/TCP	Shows the number of IPv4 TCP packets that have no RSS hash calculated by the hardware

Table 29 - RSS Diagnostic Counters

Mellanox WinOF-2 RSS Diagnostic Counters	Description
NonRss IPv4/UDP	Shows the number of IPv4 UDP packets that have no RSS hash calculated by the hardware
NonRss IPv6 Only	Shows the number of IPv6 packets that have no RSS hash calculated by the hardware
NonRss IPv6/TCP	Shows the number of IPv6 TCP packets that have no RSS hash calculated by the hardware
NonRss IPv6/UDP	Shows the number of IPv6 UDP packets that have no RSS hash calculated by the hardware
Encapsulated NonRss IPv4 Only	Shows the number of encapsulated IPv4 packets that have no RSS hash calculated by the hardware
Encapsulated NonRss IPv4/TCP	Shows the number of encapsulated IPv4 TCP packets that have no RSS hash calculated by the hardware
Encapsulated NonRss IPv4/UDP	Shows the number of encapsulated IPv4 UDP packets that have no RSS hash calculated by the hardware
Encapsulated NonRss IPv6 Only	Shows the number of encapsulated IPv6 packets that have no RSS hash calculated by the hardware
Encapsulated NonRss IPv6/TCP	Shows the number of encapsulated IPv6 TCP packets that have no RSS hash calculated by the hardware
Encapsulated NonRss IPv6/UDP	Shows the number of encapsulated IPv6 UDP packets that have no RSS hash calculated by the hardware
Rss Misc	Shows the number of received packets that have RSS hash calculated with unknown RSS hash type
Encapsulated Rss Misc	Shows the number of received encapsulated packets that have RSS hash calculated with unknown RSS hash type
NonRss Misc	Shows the number of packets that have no RSS hash calculated by the hardware for no apparent reason
Encapsulated NonRss Misc	Shows the number of encapsulated packets that have no RSS hash calculated by the hardware for no apparent reason

3.7 Network Direct Interface

The Network Direct Interface (NDI) architecture provides application developers with a networking interface that enables zero-copy data transfers between applications, kernel-bypass I/O generation and completion processing, and one-sided data transfer operations.

NDI is supported by Microsoft and is the recommended method to write an RDMA application. NDI exposes the advanced capabilities of the Mellanox networking devices and allows applications to leverage advances of RDMA.

Both RoCE and InfiniBand (IB) can implement NDI.

For further information please refer to:

http://msdn.microsoft.com/en-us/library/cc904397(v=vs.85).aspx

3.7.1 Test Running

In order to run the test, follow the steps below:

1. Connect two servers to Mellanox adapters.
2. Verify ping between the two servers.
3. Configure the RoCE version to be:
   - RoCE V2:
     - i. Linux side - V2
     - ii. Win side - V2
     - iii.Verify that ROCE udp_port is the same on the two servers. For the registry key, refer to 3.5.5.3 "RoCE Options," on page 119.
4. Select the server side and the client side, and run accordingly:
   a. Server:
   ```
   nd_rping/rping -s [-v -V -d] [-S size] [-C count] [-a addr] [-p port]
   ```
   b. Client:
   ```
   nd_rping/rping -c [-v -V -d] [-S size] [-C count] -a addr  [-p port]
   ```

Executable Options:

Letter	Usage
-s	Server side
-P	Persistent server mode allowing multiple connections
-c	Client side
-a	Address
-p	Port

Debug Extensions:

Letter	Usage
-v	Displays ping data to stdout every test cycle
-V	Validates ping data every test cycle
-d	Shows debug prints to stdout
-S	Indicates ping data size - must be < (64*1024)
-C	Indicates the number of ping cycles to perform

Example:

- ***Linux server:***

```
rping -v -s -a <IP address> -C 10
```

- ***Windows client:***

```
nd_rping -v -c -a <same IP as above> -C 10
```

The Network Direct Interface (NDI) architecture provides application developers with a networking interface that enables zero-copy data transfers between applications, kernel-bypass I/O generation and completion processing, and one-sided data transfer operations.

NDI is supported by Microsoft and is the recommended method to write an RDMA application. NDI exposes the advanced capabilities of the Mellanox networking devices and allows applications to leverage advances of RDMA.

Both RoCE and InfiniBand (IB) can implement NDI.

For further information please refer to:

http://msdn.microsoft.com/en-us/library/cc904397(v=vs.85).aspx

4 Utilities

4.1 Fabric Performance Utilities

The performance utilities described in this chapter are intended to be used as a performance micro-benchmark. They support both InfiniBand and RoCE.

For further information on the following tools, please refer to the help text of the tool by running the --help command line parameter.

Table 30 - Fabric Performance Utilities

Utility	Description
nd_write_bw	This test is used for performance measuring of RDMA-Write requests in Microsoft Windows Operating Systems. nd_write_bw is performance oriented for RDMA-Write with maximum throughput, and runs over Microsoft's NetworkDirect standard. The level of customizing for the user is relatively high. User may choose to run with a customized message size, customized number of iterations, or alternatively, customized test duration time. nd_write_bw runs with all message sizes from 1B to 4MB (powers of 2), message inlining, CQ moderation.
nd_write_lat	This test is used for performance measuring of RDMA-Write requests in Microsoft Windows Operating Systems. nd_write_lat is performance oriented for RDMA-Write with minimum latency, and runs over Microsoft's NetworkDirect standard. The level of customizing for the user is relatively high. User may choose to run with a customized message size, customized number of iterations, or alternatively, customized test duration time. nd_write_lat runs with all message sizes from 1B to 4MB (powers of 2), message inlining, CQ moderation.
nd_read_bw	This test is used for performance measuring of RDMA-Read requests in Microsoft Windows Operating Systems. nd_read_bw is performance oriented for RDMA-Read with maximum throughput, and runs over Microsoft's NetworkDirect standard. The level of customizing for the user is relatively high. User may choose to run with a customized message size, customized number of iterations, or alternatively, customized test duration time. nd_read_bw runs with all message sizes from 1B to 4MB (powers of 2), message inlining, CQ moderation.
nd_read_lat	This test is used for performance measuring of RDMA-Read requests in Microsoft Windows Operating Systems. nd_read_lat is performance oriented for RDMA-Read with minimum latency, and runs over Microsoft's NetworkDirect standard. The level of customizing for the user is relatively high. User may choose to run with a customized message size, customized number of iterations, or alternatively, customized test duration time. nd_read_lat runs with all message sizes from 1B to 4MB (powers of 2), message inlining, CQ moderation.

Table 30 - Fabric Performance Utilities

Utility	Description
nd_send_bw	This test is used for performance measuring of Send requests in Microsoft Windows Operating Systems. nd_send_bw is performance oriented for Send with maximum throughput, and runs over Microsoft's NetworkDirect standard. The level of customizing for the user is relatively high. User may choose to run with a customized message size, customized number of iterations, or alternatively, customized test duration time. nd_send_bw runs with all message sizes from 1B to 4MB (powers of 2), message inlining, CQ moderation.
nd_send_lat	This test is used for performance measuring of Send requests in Microsoft Windows Operating Systems. nd_send_lat is performance oriented for Send with minimum latency, and runs over Microsoft's NetworkDirect standard. The level of customizing for the user is relatively high. User may choose to run with a customized message size, customized number of iterations, or alternatively, customized test duration time. nd_send_lat runs with all message sizes from 1B to 4MB (powers of 2), message inlining, CQ moderation.

4.1.1 Win-Linux nd_rping Test

The purpose of this test is to check interoperability between Linux and Windows via an RDMA ping. The Windows `nd_rping` was ported from Linux's RDMACM example: `rping.c`

- Windows
  - To use the built-in nd_rping.exe tool, go to: `Program Files\Mellanox\MLNX_VPI\IB\Tools`
  - To build the `nd_rping.exe` from scratch, use the SDK example: choose the machine's OS in the configuration manager of the solution, and build the `nd_rping.exe`.
- Linux

Installing the MLNX_OFED on a Linux server will also provide the "rping" application.

4.2 Management Utilities

The management utilities described in this chapter are used to manage device's performance, NIC attributes information and traceability.

4.2.1 mlx5cmd Utilities

mlx5cmd is a general management utility used for configuring the adapter, retrieving its information and collecting its WPP trace.

➢ ***Usage***

```
Mlx5cmd.exe <tool-name> <tool-arguments>
```

4.2.1.1 Performance Tuning Utility

This utility is used mostly for IP forwarding tests to optimize the driver's configuration to achieve maximum performance when running in IP router mode.

➢ ***Usage***

```
Mlx5cmd.exe -PerfTuning <tool-arguments>
```

4.2.1.2 Information Utility

This utility displays information of Mellanox NIC attributes. It is the equivalent utility to ibstat and vstat utilities in WinOF.

➢ ***Usage***

```
Mlx5cmd.exe -Stat <tool-arguments>
```

4.2.1.3 Trace Utility

The utility saves the ETW WPP tracing of the driver.

➢ ***Usage***

```
Mlx5cmd.exe -Trace <tool-arguments>
```

4.2.1.4 QoS Configuration Utility

The utility configures Quality of Service (QoS) settings.

➢ ***Usage***

```
Mlx5cmd.exe -QoSConfig -Name <Network Adapter Name> <-DefaultUntaggedPriority | -Dcqcn>
```

For further information about the parameters, you may refer to Section 3.1.5.2, "RCM Configuration", on page 48.

4.2.1.5 mstdump Utility

This utility creates 3 mstdump file upon user request. For further information on the files created, you may refer to Table 39, "Events Causing Automatic State Dumps," on page 159.

➢ ***Usage***

```
Mlx5cmd.exe -Mstdump [-bdf <pci-bus#> <pci-device#> <pci-function#>]
```

- The PCI information can be queried from the "General" properties tab under "Location".

  Example:

  If the "Location" is "PCI Slot 3 (PCI bus 8, device 0, function 0)", run the following command:

  ```
  Mlx5cmd.exe -Mstdump -bdf 8.0.0
  ```

- The output will indicate the files location.

  Example:

"Mstdump succeeded. Dump files for device at location 8.0.0 were created in c:\windows\temp directory."

4.2.1.6 Registry Keys Utility

This utility shows the registry keys that were set in the registry and are read by the driver.

➢ ***Usage***

```
Mlx5Cmd.exe -RegKeys [-bdf <pci-bus#> <pci-device#> <pci-function#>]
```

The PCI information can be queried from the "General" properties tab under "Location".

Example:

If the "Location" is "PCI Slot 3 (PCI bus 8, device 0, function 0)", run the following command:

```
Mlx5Cmd.exe -RegKeys -bdf 8.0.0
```

4.2.1.7 Non-RSS Traffic Capture Utility

The RssSniffer utility provides sampling of packets that did not pass through the RSS engine, whether it is non-RSS traffic, or in any other case that the hardware determines to avoid RSS hashing.

The tool generates a packet dump file in a .pcap format. The RSS sampling is performed globally in native RSS mode, or per vPort in virtualization mode, when the hardware vRSS mode is active.

Note that the tool can be configured to capture only a part of the packet, as well as specific packets in a sequence (N-th).

For detailed usage, run `mlx5cmd.exe -RssSniffer -hh`

4.2.1.8 Sniffer Utility

Sniffer utility provides the user with the ability to capture Ethernet and RoCE traffic that flows to and from the Mellanox NIC's ports The tool generates a packet dump file in `.pcap` format. This file can be read using the Wireshark tool (www.wireshark.org) for graphical traffic analysis.

For detailed usage, run `mlx5cmd.exe -sniffer -help`

4.2.1.9 Link Speed Utility

This utility provides the ability to query supported link speeds by the adapter. Additionally, it enables the user to force set a particular link speed that the adapter can support.

➢ ***Usage***

```
Mlx5Cmd -LinkSpeed -Name <Network Adapter Name> -Query
```

Example:

```
Mlx5Cmd -LinkSpeed -Name <Network Adapter Name> -Set 1
```

For detailed usage, run `mlx5cmd.exe -LinkSpeed -hh`

4.3 Snapshot Utility

The snapshot tool scans the machine and provides information on the current settings of the operating system, networking and hardware.

> It is highly recommended to add this report when you contact the support team.

4.3.1 Snapshot Usage

The snapshot tool can be found at:

```
<installation_directory>\Management Tools\MLNX_System_Snapshot.exe
```

The user can set the report location.

➢ ***To generate the snapshot report:***

Step 1. [**Optional**] Change the location of the generated file by setting the full path of the file to be generated, or by pressing "Set target file" and choosing the directory that will hold the generated file and its name.

Step 2. Click on Generate HTML button

Once the report is ready, the folder which contains the report will open automatically.

5 Troubleshooting

You may be able to easily resolve the issues described in this section. If a problem persists and you are unable to resolve it, please contact your Mellanox representative or Mellanox Support at support@mellanox.com.

5.1 Installation Related Troubleshooting

Table 31 - Installation Related Issues

Issue	Cause	Solution
The installation of WinOF-2 fails with the following error message: `"This installation package is not supported by this processor type. Contact your product vendor".`	An incorrect driver version might have been installed, e.g., you are trying to install a 64-bit driver on a 32-bit machine (or vice versa).	Use the correct driver package according to the CPU architecture.

5.1.1 Installation Error Codes and Troubleshooting

5.1.1.1 Setup Return Codes

Table 32 - Setup Return Codes

Error Code	Description	Troubleshooting
1603	Fatal error during installation	Contact support
1633	The installation package is not supported on this platform.	Make sure you are installing the right package for your platform

For additional details on Windows installer return codes, please refer to:
http://support.microsoft.com/kb/229683

5.1.1.2 Firmware Burning Warning Codes

Table 33 - Firmware Burning Warning Codes

Error Code	Description	Troubleshooting
1004	Failed to open the device	Contact support
1005	Could not find an image for at least one device	The firmware for your device was not found. Please try to manually burn the firmware.
1006	Found one device that has multiple images	Burn the firmware manually and select the image you want to burn.

Table 33 - Firmware Burning Warning Codes

Error Code	Description	Troubleshooting
1007	Found one device for which force update is required	Burn the firmware manually with the force flag.
1008	Found one device that has mixed versions	The firmware version or the expansion rom version does not match.

For additional details, please refer to the MFT User Manual:

http://www.mellanox.com > Products > Firmware Tools

5.1.1.3 Restore Configuration Warnings

Table 34 - Restore Configuration Warnings

Error Code	Description	Troubleshooting
3	Failed to restore the configuration	Please see log for more details and contact the support team

5.2 InfiniBand Related Troubleshooting

Table 35 - InfiniBand Related Issues

Issue	Cause	Solution
The InfiniBand interfaces are not up after the first reboot after the installation process is completed.	Port status might be `PORT_DOWN`: Switch port state might be “disabled” or cable is disconnected.	Enable switch admin or connect cable.
	Port status might be `PORT_INITIALIZED`: SM might not be running on the fabric.	Run the SM on the fabric.
	Port status might be `PORT_ARMED`: Firmware issue.	Please contact Mellanox Support.
	SR-IOV might be enabled with firmware that does not support SR-IOV and IPoIB simultaneously. in this case, the driver will report an error message stating that IPoIB is not supported by the firmware.	Use the mlxconfig tool to disable SR-IOV. Consult the MFT User Manual for further details.

5.3 Ethernet Related Troubleshooting

For further performance related information, please refer to the *Performance Tuning Guide* and to Section 3.6, “Performance Tuning and Counters”, on page 120

Table 36 - Ethernet Related Issues

Issue	Cause	Solution
Low performance	Non-optimal system configuration might have occurred.	See section “Performance Tuning and Counters” on page 120. to take advantage of Mellanox 10/40/56 GBit NIC performance.
The driver fails to start.	There might have been an RSS configuration mismatch between the TCP stack and the Mellanox adapter.	1. Open the event log and look under `"System"` for the `"mlx5"` source. 2. If found, enable RSS, run: `"netsh int tcp set global rss = enabled"`. or a less recommended suggestion (as it will cause low performance): • Disable RSS on the adapter, run: `"netsh int tcp set global rss = no dynamic balancing"`.

Table 36 - Ethernet Related Issues

Issue	Cause	Solution
The driver fails to start and a yellow sign appears near the "`Mellanox ConnectX-4/ ConnectX-5 Adapter <X>`" in the Device Manager display. (Code 10)	Look into the Event Viewer to view the error.	• If the failure occurred due to unsupported mode type, refer to Section 3.1.2, "Mode Configuration" for the solution. • If the solution isn't mentioned in event viewer, disable and re-enable "`Mellanox ConnectX-4/ConnectX-5 Adapter <X>`" from the Device Manager display. If the failure resumes, please refer to Mellanox support at support@mellanox.com.
No connectivity to a Fault Tolerance team while using network capture tools (e.g., Wireshark).	The network capture tool might have captured the network traffic of the non-active adapter in the team. This is not allowed since the tool sets the packet filter to "`promiscuous`", thus causing traffic to be transferred on multiple interfaces.	Close the network capture tool on the physical adapter card, and set it on the team interface instead.
No Ethernet connectivity on 10Gb adapters after activating Performance Tuning (part of the installation).	A TcpWindowSize registry value might have been added.	• Remove the value key under `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\TcpWindowSize` Or • Set its value to `0xFFFF`.
Packets are being lost.	The port MTU might have been set to a value higher than the maximum MTU supported by the switch.	Change the MTU according to the maximum MTU supported by the switch.
NVGRE changes done on a running VM, are not propagated to the VM.	The configuration changes might not have taken effect until the OS is restarted.	Stop the VM and afterwards perform any NVGRE configuration changes on the VM connected to the virtual switch.

5.4 Performance Related Troubleshooting

Table 37 - Performance Related Issues

Issue	Cause	Solution
Low performance issues	The OS profile might not be configured for maximun performace.	1. Go to "Power Options" in the "Control Panel". Make sure "Maximum Performance" is set as the power scheme 2. Reboot the machine.

5.4.1 General Diagnostic

Issue 1. Go to “Device Manager”, locate the Mellanox adapter that you are debugging, right-click and choose “Properties” and go to the “Information” tab:

- PCI Gen 1: should appear as "PCI-E 2.5 GT/s"
- PCI Gen 2: should appear as "PCI-E 5.0 GT/s"
- PCI Gen 3: should appear as "PCI-E 8.0 GT/s"
- Link Speed: 56.0 Gbps / 40.0Gbps / 10.0Gbps / 100 Gbps

Issue 2. To determine if the Mellanox NIC and PCI bus can achieve their maximum speed, it's best to run nd_send_bw in a loopback. On the same machine:

1. Run "start /b /affinity 0x1 nd_send_bw -S <IP_host>" where <IP_host> is the local IP.
2. Run "start /b /affinity 0x2 nd_send_bw -C <IP_host>"
3. Repeat for port 2 with the appropriate IP.
4. On PCI Gen3 the expected result is around 5700MB/s

On PCI Gen2 the expected result is around 3300MB/s

Any number lower than that points to bad configuration or installation on the wrong PCI slot. Malfunctioning QoS settings and Flow Control can be the cause as well.

Issue 3. To determine the maximum speed between the two sides with the most basic test:

1. Run "nd_send_bw -S <IP_host1>" on machine 1 where <IP_host1> is the local IP.
2. Run "nd_send_bw -C <IP_host1>" on machine 2.
3. Results appear in Gb/s (Gigabits 2^30), and reflect the actual data that was transferred, excluding headers.
4. If these results are not as expected, the problem is most probably with one or more of the following:

- Old Firmware version.
- Misconfigured Flow-control: Global pause or PFC is configured wrong on the hosts, routers and-switches. See Section 3.1.3.1,“Configuring 56GbE Link Speed,” on page 39
- CPU/power options are not set to "Maximum Performance".

5.5 Virtualization Related Troubleshooting

Table 38 - Virtualization Related Issues

Issue	Cause	Solution
When enabling the VMQ, in case NVGRE offload is enabled, and a teaming of two virtual ports is performed, no ping is detected between the VMs and/or ping is detected but no establishing of TCP connection is possible.	Might be missing critical Microsoft updates.	Please refer to: http://support.microsoft.com/kb/2975719 "August 2014 update rollup for Windows server RT 8.1, Windows server 8.1, and Windows server 2012 R2" – specifically, fixes.
When running the system from an SR-IOV, The operation of opening RDMA resources might fail.	Low resources for VF	1. Run the mlxconfig tool, according to the instructions in the "MFT User Manual" that is available on www.mellanox.com ->Products -> InfiniBand/VPI Drivers -> Firmware Tools". 2. Extract the device name from "mst status", select the appropriate size (> 0, 2,4,8), and run the following command: `mlxconfig -[device name] set VF_LOG_BAR_SIZE=size`

5.6 Reported Driver Events

The driver records events in the system log of the Windows server event system which can be used to identify, diagnose, and predict sources of system problems.

To see the log of events, open System Event Viewer as follows:

- Right click on My Computer, click Manage, and then click Event Viewer.

  OR

1. Click start-->Run and enter "eventvwr.exe".
2. In Event Viewer, select the system log.

   The following events are recorded:

   - Mellanox ConnectX-4 VPI Adapter <X> device has been requested for <Y> Virtual Functions (VFs), while it only supports <Z> VFs. Therefore, only <q> VFs will be allowed.
   - Mellanox ConnectX-4 VPI Adapter <X> device has been configured to use RSS while Windows' TCP RSS is disabled. This configuration prevents the initialization and enabling of the port. You need to either enable Windows' TCP RSS, or configure the adapter's port to disable RSS. For further details, see the README file under the documentation folder.
   - Mellanox ConnectX-4 VPI Adapter <X>: Jumbo packet value read from registry (<Y>) is greater than the value supported by FW (<Z>). Therefore use the maximum value supported by FW(<q>).
   - Mellanox ConnectX-4 VPI Adapter <X>: Maximum MTU supported by FW <L>.<Y>.<Z>(<q>) is smaller than the minimum value <K>.
   - Mellanox ConnectX-4 VPI Adapter <X> device is successfully stopped.
   - Mellanox ConnectX-4 VPI Adapter <X> device startup fails due to less than minimum MSI-X vectors available.
   - Mellanox ConnectX-4 VPI Adapter <X> device detects that the link is up, and has initiated a normal operation.
   - Mellanox ConnectX-4 VPI Adapter <X> device detects that the link is down. This may occur if the physical link is disconnected or damaged, or if the other end-port is down.
   - Mellanox ConnectX-4 VPI Adapter <X>Mellanox ConnectX-4 VPI Adapter <X> device configures not to use RSS. This configuration may significantly affect the network performance.
   - Mellanox ConnectX-4 VPI Adapter <X> device reports an "Error event" on CQn #<Y>. Since the event type is:<Z>, the NIC will be reset. (The issue is reported in Function <K>).
   - Mellanox ConnectX-4 VPI Adapter <X> device reports a send=<Y> "CQE error" on cqn #<Z> qpn #<q> cqe_error->syndrome <K>, cqe_error->vendor_error_syndrome <L>, Opcode <F>. Therefore, the NIC will be reset. (The issue is reported in Function <M>). For more information refer to details.

- Mellanox ConnectX-4 VPI Adapter <X> device reports an "EQ stuck" on EQn <Y>. Attempting recovery.
- Mellanox ConnectX-4 VPI Adapter <X> device reports a send completion handling time-out on TxQueue 0x<Y>. Attempting recovery.
- Mellanox ConnectX-4 VPI Adapter <X> device reports a receive completion handling timeout on RxQueue 0x<Y>. Attempting recovery.
- Flow control on the device Mellanox ConnectX-4 VPI Adapter <X> wasn't enabled. Therefore, RoCE cannot function properly. To resolve this issue, please make sure that flow control is configured on both the hosts and switches in your network. For more details, please refer to the user manual.
- Mellanox ConnectX-4 VPI Adapter <X> device is configured with a MAC address designated as a multicast address: <Y>.
  Please configure the registry value NetworkAddress with another address, then restart the driver.
- The miniport driver initiates reset on device Mellanox ConnectX-4 VPI Adapter <X>.
- NDIS initiates reset on device Mellanox ConnectX-4 VPI Adapter <X>.
- Reset on device Mellanox ConnectX-4 VPI Adapter <X> has finished.
- Mellanox ConnectX-4 VPI Adapter <X> has got:
  - vendor_id <Y>
  - device_id <Z>
  - subvendor_id <F>
  - subsystem_id <L>
  - HW revision <M>
  - FW version <R>.<G>.<Q>
  - port type <N>
- Mellanox ConnectX-4 VPI Adapter <X>: QUERY_HCA_CAP command fails with error <Y>.
  The adapter card is dysfunctional.
  Most likely a FW problem.
  Please burn the last FW and restart the Mellanox ConnectX device.
- Mellanox ConnectX-4 VPI Adapter <X>: QUERY_ADAPTER command fails with error <Y>.
  The adapter card is dysfunctional.
  Most likely a FW problem.
  Please burn the last FW and restart the Mellanox ConnectX device.

- Mellanox ConnectX-4 VPI Adapter <X>: The number of allocated MSI-X vectors is less than recommended. This may decrease the network performance.
  The number of requested MSI-X vectors is: <Y> while the number of allocated MSI-X vectors is: <Z>.
- Mellanox ConnectX-4 VPI Adapter <X>: FW command fails. op 0x<Y>, status 0x<Z>, errno <F>, syndrome 0x<L>.
- Too many IPs in-use for RRoCE.
  Mellanox ConnectX-4 VPI Adapter <X>: RRoCE supports only <Y> IPs per port.
  Please reduce the number of IPs to use the new IPs.
- Mellanox ConnectX-4 VPI Adapter <X>: Driver startup fails because an insufficient number of Event Queues (EQs) is available.
  (<Y> are required, <Z> are recommended, <M> are available)
- Mellanox ConnectX-4 VPI Adapter <X>: Execution of FW command fails. op 0x<Y>, errno <Z>.
- Mellanox ConnectX-4 VPI Adapter <X>: Driver startup has failed due to unsupported port type=<Y> configured on the device.
  The driver supports Ethernet mode only, please refer to the Mellanox WinOF-2 User Manual for instructions on how to configure the correct mode.
- Mellanox ConnectX-4 VPI Adapter <X>: Driver startup fails because minimal driver requirements are not supported by FW <Y>.<Z>.<L>.

  FW reported:
  - rss_ind_tbl_cap <Q>
  - vlan_cap <M>
  - max_rqs <F>
  - max_sqs <N>
  - max_tirs <O>

  Please burn a firmware that supports the requirements and restart the Mellanox ConnectX device. For additional information, please refer to Support information on http://mellanox.com.
- Mellanox ConnectX-4 VPI Adapter <X>: Driver startup fails because maximum flow table size that is supported by FW <Y>.<Z>.<L> is too small (<K> entries).
  Please burn a firmware that supports a greater flow table size and restart the Mellanox ConnectX device. For additional information, please refer to Support information on http://mellanox.com.
- Mellanox ConnectX-4 VPI Adapter <X>: Driver startup fails because required receive WQE size is greater than the maximum WQEs size supported by FW <Y>.<Z>.<M>.
  (<F> are required, <O> are supported)

- Mellanox ConnectX-4 VPI Adapter <X>: Driver startup fails because maximum WQE size that is supported by FW <Y>.<L>.<M> is too small (<K>).
  Please burn a firmware that supports a greater WQE size and restart the Mellanox ConnectX device. For additional information, please refer to Support information on http://mellanox.com
- Mellanox ConnectX-4 VPI Adapter <X>: CQ moderation is not supported by FW <Y>.<Z>.<L>.
- Mellanox ConnectX-4 VPI Adapter <X>: CQ to EQ remap is not supported by FW <Y>.<Z>.<L>.
- Mellanox ConnectX-4 VPI Adapter <X>: VPort counters are not supported by FW <Y>.<Z>.<L>.
- Mellanox ConnectX-4 VPI Adapter <X>: LSO is not supported by FW <Y>.<Z>.<L>.
- Mellanox ConnectX-4 VPI Adapter <X>: Checksum offload is not supported by FW <Y>.<Z>.<L>.
- NDIS initiated reset on device Mellanox ConnectX-4 VPI Adapter <X> has failed.
- Mellanox ConnectX-4 VPI Adapter <X>: mstdump %SystemRoot%\Temp\<Y>_<Z><L>_<M>_<F>_<O>.log was created after fatal error.
- Mellanox ConnectX-4 VPI Adapter <X>: mstdump %SystemRoot%\Temp\<Y>_<Z><L>_<M>_<F>_<O>.log was created after OID request.
- Mellanox ConnectX-4 VPI Adapter <X> Physical/Virtual function drivers compatibility issue <Y>.
- Mellanox ConnectX-4 VPI Adapter <X> (module <Y>) detects that the link is down. Cable is unplugged. Please connect the cable to continue working.
- Mellanox ConnectX-4 VPI Adapter <X> (module <Y>) detects that the link is down. Cable is unplugged. Please connect the cable to continue working.
- Mellanox ConnectX-4 VPI Adapter <X> Setting QoS port default priority is not allowed on a virtual device. This adapter will use the default priority <Y>.
- Mellanox ConnectX-4 VPI Adapter <X>: FW health report - ver <Y>, hw <Z>, callra <A>, var[1] <B> synd <C>.
- Mellanox ConnectX-4 VPI Adapter <X>: Adapter failed to initialize due to FW initialization timeout.
- Mellanox ConnectX-4 VPI Adapter <X>: Setting QoS port default priority is not allowed on a virtual device. This adapter will use the default priority <Y>.
- Mellanox ConnectX-4 VPI Adapter <X> failed to set port default priority to <Y>. This adapter will use the default priority <Z>.
- Mellanox ConnectX-4 VPI Adapter <X>: Dcqcn is not allowed on a virtual device.

- Dcqcn was enabled for adapter Mellanox ConnectX-4 VPI Adapter <X> but FW <Y>.<Z>.<W> does not support it. Dcqcn congestion control will not be enabled for this adapter. Please burn a newer firmware. For more details, please refer to the user manual document.
- Mellanox ConnectX-4 VPI Adapter <X> failed to set Dcqcn RP/NP congestion control parameters. This adapter will use default Dcqcn RP/NP congestion control values. Please verify the Dcqcn configuration and then restart the adapter.
- Mellanox ConnectX-4 VPI Adapter <X> failed to enable Dcqcn RP/NP congestion control for priority <Y>. This adapter will continue without Dcqcn <Y> congestion control for this priority. Please verify the Dcqcn configuration and then restart the adapter.
- Mellanox ConnectX-4 VPI Adapter <X>: mstdump SystemRoot\Temp\<Y>.log was created after a timeout on TxQueue.
- Mellanox ConnectX-4 VPI Adapter <X>: mstdump SystemRoot\Temp\<Y>.log was created after a timeout on RxQueue.
- Mellanox ConnectX-4 VPI Adapter <X>: Dcqcn RP attributes:
  - DcqcnClampTgtRate = <Y>
  - DcqcnClampTgtRateAfterTimeInc = <Z>
  - DcqcnRpgTimeReset = <E>
  - DcqcnRpgByteReset = <L>
  - DcqcnRpgThreshold = <M>
  - DcqcnRpgAiRate = <N>
  - DcqcnRpgHaiRate = <R>
  - DcqcnAlphaToRateShift = <W>
  - DcqcnRpgMinDecFac = <G>
  - DcqcnRpgMinRate = <Q>
  - DcqcnRateToSetOnFirstCnp = <F>
  - DcqcnDceTcpG = <V>
  - DcqcnDceTcpRtt = <O>
  - DcqcnRateReduceMonitorPeriod = <K>
  - DcqcnInitialAlphaValue = <J>
- Mellanox ConnectX-4 VPI Adapter <X>: Dcqcn NP attributes:
  - DcqcnMinTimeBetweenCnps = <Y>
  - DcqcnCnpDscp = <Z>
  - DcqcnCnpPrioMode = <V>
  - DcqcnCnp802pPrio = <W>

- Mellanox ConnectX-4 VPI Adapter <X>: FW health report - ver <Y>, hw <Z>, callra <W>, var[1] <K>, synd <K>.
- Mellanox ConnectX-4 VPI Adapter <X>: RDMA device initialization failure <Y>. This adapter will continue running in Ethernet only mode.
- Mellanox ConnectX-4 VPI Adapter <X>: mstdump %SystemRoot%\Temp\<A>_<B>_<C>_<D>_<E>_<F>.log was created after changed of link state.
- Mellanox ConnectX-4 VPI Adapter <X> adapter detected that the port type was changed. Therefore, the following registry keys were set to the default values of the new port type(<Y>).
- *JumboPacket = <Z>

5.7 State Dumping

Upon several types of events, the drivers can produce a set of files reflecting the current state of the adapter.

Automatic state dumps are done upon the following events:

Table 39 - Events Causing Automatic State Dumps

Event Type	Description	Provider	Default	Tag
CMD_ERR	Command failure or timeout on a command	Mlx5	On	c
EQ_STUCK	Driver decided that an event queue is stuck	Mlx5	On	e
TXCQ_STUCK	Driver decided that a transmit completion queue is stuck	Mlx5	On	t
RXCQ_STUCK	Driver decided that a receive completion queue is stuck	Mlx5	On	r
PORT_STATE	Adapter passed to “port up” state, “port down” state or “port unknown” state.	Mlx5	On	p
ON_OID	User application asked to generate dump files	Mlx5	N/A	o

where

Provider	The driver creating the set of files.
Default	Whether or not the state dumps are created by default upon this event.
Tag	Part of the file name, used to identify the event that has triggered the state dump.

Dump events can be enabled/disabled by adding DWORD32 parameters into HKLM\System\CurrnetControlSet\Services\mlx5\Parameters\Diag as follows:

- ***Dump events can be disabled by adding MstDumpMode parameter as follows:***

```
MstDumpMode                                    0
```

- ***PORT_STATE events can be disabled by adding EnableDumpOnUnknownLink and EnableDumpOnPortDown parameters as follows:***

```
EnableDumpOnUnknownLink                        0
EnableDumpOnPortDown                           0
EnableDumpOnPortUp                             0
```

- ***EQ_STUCK, TXCQ_STUCK and RXCQ_STUCK events can be disabled by adding DisableDumpOnEqStuck, DisableDumpOnTxCqStuck and DisableDumpOnRxCqStuck parameters as follows:***

```
DisableDumpOnTxCqStuck                         1
DisableDumpOnTxCqStuck                         1
DisableDumpOnRxCqStuck                         1
```

The set consists of the following files:

- 3 consecutive mstdump files

These files are created in the %SystemRoot%\temp directory, and should be sent to Mellanox Support for analysis when debugging WinOF2 driver problems. Their names have the following format:<Driver_mode_of_work>_<card_location>_<event_tag_name>_<event_number>_<event_name>_<file_type>_<file_index>.log

where:

Driver_mode_of_work	The mode of driver work. For example: ‘SingleFunc’
card_location	In form bus_device_function, For example: 4_0_0
event_tag_name	One-symbol tag. See in Table 39 - “Events Causing Automatic State Dumps,” on page 159
event_number	The index of dump files set and created for this event. This number is restricted by the hidden Registry parameter DumpEventsNum
event_name	A short string naming the event. For example: ‘eth-down-1’ = “Ethernet port1 passed to DOWN state”
file_type	Type of file in the set. For example: “crspace”, “fwtrace”, “eq_dump” and “eq_print”
file_index	The file number of this type in the set

Example:

Name: SingleFunc_4_0_0_p000_eth-down-1_eq_dump_0.log

The default number of sets of files for each event is 20. It can be changed by adding DumpEventsNum DWORD32 parameter under HKLM\System\CurrnetControlSet\Services\mlx5\Parameters and setting it to another value.

5.8 Extracting WPP Traces

WinOF-2 Mellanox driver automatically dumps trace messages that can be used by the driver developers for debugging issues that have recently occurred on the machine.

The default location for the trace file is:

```
%SystemRoot%\system32\LogFiles\Mlnx\Mellanox-WinOF2-System.etl
```

The automatic trace session is called Mellanox-WinOF2-Kernel.

➢ ***To view the session:***

```
logman query Melloanox-WinOF2-Kernel -ets
```

➢ ***To stop the session:***

```
logman stop Mellanox-WinOF2-Kernel -ets
```

When opening a support ticket, it is advised to attach the file to the ticket.

Appendix A: NVGRE Configuration Scripts Examples

The setup is as follow for both examples below:

```
 Hypervisor host14 = "Port1", 192.168.20.114/24
      VM on host14 = host14-005, 172.16.14.5/16, Mac 00155D720100
      VM on host14 = host14-006, 172.16.14.6/16, Mac 00155D720101
 Hypervisor host15 = "Port1", 192.168.20.115/24
      VM on host15 = host15-005, 172.16.15.5/16, Mac 00155D730100
      VM on host15 = host15-006, 172.16.15.6/16, Mac 00155D730101
```

A.1 Adding NVGRE Configuration to Host 14 Example

The following is an example of adding NVGRE to Host 14.

```
 # On both sides
 # vSwitch create command

 # Note, that vSwitch configuration is persistent, no need to configure it after each
reboot

 New-VMSwitch "VSwMLNX" -NetAdapterName "Port1" -AllowManagementOS $true

 # Shut down VMs
 Stop-VM -Name "host14-005" -Force -Confirm
 Stop-VM -Name "host14-006" -Force -Confirm
 # Connect VM to vSwitch (maybe you have to switch off VM before), doing manual does also
work
 # Connect-VMNetworkAdapter -VMName " host14-005" -SwitchName "VSwMLNX"
 Add-VMNetworkAdapter -VMName "host14-005" -SwitchName "VSwMLNX" -StaticMacAddress
"00155D720100"
 Add-VMNetworkAdapter -VMName "host14-006" -SwitchName "VSwMLNX" -StaticMacAddress
"00155D720101"
```

```
# -------  The commands from Step 2 - 4 are not persistent, Its suggested to create
script is running after each OS reboot

# Step 2. Configure a Subnet Locator and Route records on each Hyper-V Host (Host 1 and
Host 2) host14 & host15
New-NetVirtualizationLookupRecord -CustomerAddress 172.16.14.5 -ProviderAddress
192.168.20.114 -VirtualSubnetID 5001 -MACAddress "00155D720100" -Rule "TranslationMetho-
dEncap"
New-NetVirtualizationLookupRecord -CustomerAddress 172.16.14.6 -ProviderAddress
192.168.20.114 -VirtualSubnetID 5001 -MACAddress "00155D720101" -Rule "TranslationMetho-
dEncap"
New-NetVirtualizationLookupRecord -CustomerAddress 172.16.15.5 -ProviderAddress
192.168.20.115 -VirtualSubnetID 5001 -MACAddress "00155D730100" -Rule "TranslationMetho-
dEncap"
New-NetVirtualizationLookupRecord -CustomerAddress 172.16.15.6 -ProviderAddress
192.168.20.115 -VirtualSubnetID 5001 -MACAddress "00155D730101" -Rule "TranslationMetho-
dEncap"
# Add customer route
New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-
000000005001}" -VirtualSubnetID "5001" -DestinationPrefix "172.16.0.0/16" -NextHop
"0.0.0.0" -Metric 255
# Step 3. Configure the Provider Address and Route records on Hyper-V Host 1 (Host 1
Only) host14
$NIC = Get-NetAdapter "Port1"
New-NetVirtualizationProviderAddress -InterfaceIndex $NIC.InterfaceIndex -ProviderAd-
dress 192.168.20.114 -PrefixLength 24
New-NetVirtualizationProviderRoute -InterfaceIndex $NIC.InterfaceIndex -Destination-
Prefix "0.0.0.0/0" -NextHop 192.168.20.1
# Step 5. Configure the Virtual Subnet ID on the Hyper-V Network Switch Ports for each
Virtual Machine on each Hyper-V Host (Host 1 and Host 2)
# Run the command below for each VM on the host the VM is running on it, i.e. the for
host14-005, host14-006 on
# host 192.168.20.114 and for VMs host15-005, host15-006 on host 192.168.20.115
# host14 only
Get-VMNetworkAdapter -VMName host14-005 | where {$_.MacAddress -eq "00155D720100"} |
Set-VMNetworkAdapter -VirtualSubnetID 5001
Get-VMNetworkAdapter -VMName host14-006 | where {$_.MacAddress -eq "00155D720101"} |
Set-VMNetworkAdapter -VirtualSubnetID 5001
```

A.2 Adding NVGRE Configuration to Host 15 Example

The following is an example of adding NVGRE to Host 15.

```
  # On both sides
  # vSwitch create command

  # Note, that vSwitch configuration is persistent, no need to configure it after each
reboot

  New-VMSwitch "VSwMLNX" -NetAdapterName "Port1" -AllowManagementOS $true

  # Shut down VMs
  Stop-VM -Name "host15-005" -Force -Confirm
  Stop-VM -Name "host15-006" -Force -Confirm
  # Connect VM to vSwitch (maybe you have to switch off VM before), doing manual does also
work
  # Connect-VMNetworkAdapter -VMName " host14-005" -SwitchName "VSwMLNX"
  Add-VMNetworkAdapter -VMName "host15-005" -SwitchName "VSwMLNX" -StaticMacAddress
"00155D730100"
  Add-VMNetworkAdapter -VMName "host15-006" -SwitchName "VSwMLNX" -StaticMacAddress
"00155D730101"
  # -------  The commands from Step 2 - 4 are not persistent, Its suggested to create
script is running after each OS reboot

  # Step 2. Configure a Subnet Locator and Route records on each Hyper-V Host (Host 1 and
Host 2) host14 & host15
  New-NetVirtualizationLookupRecord -CustomerAddress 172.16.14.5 -ProviderAddress
192.168.20.114 -VirtualSubnetID 5001 -MACAddress "00155D720100" -Rule "TranslationMetho-
dEncap"
  New-NetVirtualizationLookupRecord -CustomerAddress 172.16.14.6 -ProviderAddress
192.168.20.114 -VirtualSubnetID 5001 -MACAddress "00155D720101" -Rule "TranslationMetho-
dEncap"
  New-NetVirtualizationLookupRecord -CustomerAddress 172.16.15.5 -ProviderAddress
192.168.20.115 -VirtualSubnetID 5001 -MACAddress "00155D730100" -Rule "TranslationMetho-
dEncap"
  New-NetVirtualizationLookupRecord -CustomerAddress 172.16.15.6 -ProviderAddress
192.168.20.115 -VirtualSubnetID 5001 -MACAddress "00155D730101" -Rule "TranslationMetho-
dEncap"
  # Add customer route
  New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-
000000005001}" -VirtualSubnetID "5001" -DestinationPrefix "172.16.0.0/16" -NextHop
"0.0.0.0" -Metric 255
  # Step 4. Configure the Provider Address and Route records on Hyper-V Host 2 (Host 2
Only) host15
  $NIC = Get-NetAdapter "Port1"
  New-NetVirtualizationProviderAddress -InterfaceIndex $NIC.InterfaceIndex -ProviderAd-
dress 192.168.20.115 -PrefixLength 24
  New-NetVirtualizationProviderRoute -InterfaceIndex $NIC.InterfaceIndex -Destination-
Prefix "0.0.0.0/0" -NextHop 192.168.20.1
```

```
# Step 5. Configure the Virtual Subnet ID on the Hyper-V Network Switch Ports for each
Virtual Machine on each Hyper-V Host (Host 1 and Host 2)
# Run the command below for each VM on the host the VM is running on it, i.e. the for
host14-005, host14-006 on
# host 192.168.20.114 and for VMs host15-005, host15-006 on host 192.168.20.115
# host15 only
Get-VMNetworkAdapter -VMName host15-005 | where {$_.MacAddress -eq "00155D730100"} |
Set-VMNetworkAdapter -VirtualSubnetID 5001
Get-VMNetworkAdapter -VMName host15-006 | where {$_.MacAddress -eq "00155D730101"} |
Set-VMNetworkAdapter -VirtualSubnetID 5001
```

Appendix B: Windows MPI (MS-MPI)

B.1 Overview

Message Passing Interface (MPI) is meant to provide virtual topology, synchronization, and communication functionality between a set of processes.

With MPI you can run one process on several hosts.

- Windows MPI run over the following protocols:
  - Sockets (Ethernet)
  - Network Direct (ND)

B.1.1 System Requirements

- Install HPC (Build: 4.0.3906.0).
- Validate traffic (ping) between the whole MPI Hosts.
- Every MPI client need to run smpd process which open the mpi channel.
- MPI Initiator Server need to run: mpiexec. If the initiator is also client it should also run smpd.

B.2 Running MPI

Step 1. Run the following command on each mpi client.

```
start smpd -d -p <port>
```

Step 2. Install ND provider on each MPI client in MPI ND.

Step 3. Run the following command on MPI server.

```
mpiexec.exe -p <smpd_port> -hosts <num_of_hosts>
<hosts_ip_list> -env MPICH_NETMASK <network_ip/subnet> -
env MPICH_ND_ZCOPY_THRESHOLD -1 -env MPICH_DISABLE_ND <0/
1> -env MPICH_DISABLE_SOCK <0/1> -affinity <process>
```

B.3 Directing MSMPI Traffic

Directing MPI traffic to a specific QoS priority may delayed due to:

- Except for NetDirectPortMatchCondition, the QoS powershell CmdLet for NetworkDirect traffic does not support port range. Therefore, NetwrokDirect traffic cannot be directed to ports 1-65536.
- The MSMPI directive to control the port range (namely: MPICH_PORT_RANGE 3000,3030) is not working for ND, and MSMPI chose a random port.

B.4 Running MSMPI on the Desired Priority

Step 1. Set the default QoS policy to be the desired priority (Note: this prio should be lossless all the way in the switches*)

Step 2. Set SMB policy to a desired priority only if SMD Traffic running.

Step 3. **[Recommended]** Direct ALL TCP/UDP traffic to a lossy priority by using the "IPProtocol-MatchCondition".

> TCP is being used for MPI control channel (smpd), while UDP is being used for other services such as remote-desktop.

Arista switches forwards the pcp bits (e.g. 802.1p priority within the vlan tag) from ingress to egress to enable any two End-Nodes in the fabric as to maintain the priority along the route.

In this case the packet from the sender goes out with priority X and reaches the far end-node with the same priority X.

> The priority should be losslessin the switches

➢ ***To force MSMPI to work over ND and not over sockets, add the following in mpiexec command:***

```
-env MPICH_DISABLE_ND 0 -env MPICH_DISABLE_SOCK 1
```

B.5 Configuring MPI

Step 1. Configure all the hosts in the cluster with identical PFC (see the PFC example below).

Step 2. Run the WHCK ND based traffic tests to Check PFC (ndrping, ndping, ndrpingpong, ndpingpong).

Step 3. Validate PFC counters, during the run-time of ND tests, with "Mellanox Adapter QoS Counters" in the perfmon.

Step 4. Install the same version of HPC Pack in the entire cluster.
NOTE: Version mismatch in HPC Pack 2012 can cause MPI to hung.

Step 5. Validate the MPI base infrastructure with simple commands, such as `"hostname"`.

B.5.1 PFC Example

In the example below, ND and NDK go to priority 3 that configures no-drop in the switches. The TCP/UDP traffic directs ALL traffic to priority 1.

- Install dcbx.

```
Install-WindowsFeature Data-Center-Bridging
```

- Remove the entire previous settings.

```
Remove-NetQosTrafficClass
Remove-NetQosPolicy -Confirm:$False
```

- Set the DCBX Willing parameter to false as Mellanox drivers do not support this feature

```
Set-NetQosDcbxSetting -Willing 0
```

- Create a Quality of Service (QoS) policy and tag each type of traffic with the relevant priority.
  In this example we used TCP/UDP priority 1, ND/NDK priority 3.

```
New-NetQosPolicy "SMB" -NetDirectPortMatchCondition 445 -PriorityValue8021Action 3
New-NetQosPolicy "DEFAULT" -Default -PriorityValue8021Action 3
New-NetQosPolicy "TCP" -IPProtocolMatchCondition TCP -PriorityValue8021Action1
New-NetQosPolicy "UDP" -IPProtocolMatchCondition UDP -PriorityValue8021Action 1
```

- Enable PFC on priority 3.

```
Enable-NetQosFlowControl 3
```

- Disable Priority Flow Control (PFC) for all other priorities except for 3.

```
Disable-NetQosFlowControl 0,1,2,4,5,6,7
```

- Enable QoS on the relevant interface.

```
Enable-netadapterqos -Name
```

B.5.2 Running MPI Command Examples

- Running MPI pallas test over ND.

```
> mpiexec.exe -p 19020 -hosts 4 11.11.146.101 11.21.147.101
11.21.147.51
11.11.145.101 -env MPICH_NETMASK 11.0.0.0/
255.0.0.0 -env MPICH_ND_ZCOPY_THRESHOLD -1 -env MPICH_DISABLE_ND 0
-env
MPICH_DISABLE_SOCK 1 -affinity c:\\test1.exe
```

- Running MPI pallas test over ETH.

```
> exempiexec.exe -p 19020 -hosts 4 11.11.146.101 11.21.147.101
11.21.147.51
11.11.145.101 -env MPICH_NETMASK 11.0.0.0/
255.0.0.0 -env MPICH_ND_ZCOPY_THRESHOLD -1 -env MPICH_DISABLE_ND 1
-env
MPICH_DISABLE_SOCK 0 -affinity c:\\test1.exe
```