Mellanox
TECHNOLOGIES
Connect. Accelerate. Outperform.™

Mellanox WinOF-2 User Manual

Rev 1.10 (Beta)

www.mellanox.com

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

Mellanox Technologies, Ltd.
Hakidma 26
Ofer Industrial Park
Yokneam 2069200
Israel
www.mellanox.com
Tel: +972 (0)74 723 7200
Fax: +972 (0)4 959 3245

Table of Contents

List of Tables

Document Revision History

Table 1 - Document Revision History

Document Revision	Date	Changes
Rev 1.10	July 8, 2015	Updated the following sections: • Section 1, “Introduction”, on page 11 • Section 3.1.2.1, “IP Routable (RoCEv2)”, on page 26 • Section 3.1.2.6, “Configuring the RoCE Mode”, on page 31
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 1.10 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 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

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 node • Enabling changing card configuration to support SRIOV
WinOF-2 Release Notes	For possible software issues, please refer to WinOF-2 Release Notes.

1 Introduction

This User Manual describes installation, configuration and operation of Mellanox WinOF-2 driver Rev 1.10 package.

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

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

- Support for ConnectX®-4 single and dual port adapter cards[1]
- Up to 16 Rx queues per port
- Dedicated PCI function per physical port
- Rx steering mode (RSS)
- Hardware Tx/Rx checksum calculation
- Large Send off-load (i.e., TCP Segmentation Off-load)
- Receive Side Coalescing (RSC, or LRO in Linux)
- Hardware multicast filtering
- Adaptive interrupt moderation
- Support for MSI-X interrupts
- NDK with SMB-Direct
- NDv1 and v2 API support in user space
- VMQ for Hypervisor
- Hardware VLAN filtering
- RDMA over Converged Ethernet
  - RoCE MAC Based (v1)
  - RRoCE over UDP (v2)

1.1 Supplied Packages

Mellanox WinOF-2 driver Rev 1.10 includes the following package:

- MLNX_WinOF2-1_10_All_x64.exe

1.2 WinOF-2 Set of Documentation

Under <installation_directory>\Documentation:

- License file
- User Manual (this document)
- MLNX_WinOF-2 Release Notes

[1] WinOF-2 does not support earlier Mellanox adapters. For earlier adapters, the Windows driver is MLNX_WinOF.

1.3 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.

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

For further details on MPI, please refer to Appendix B,“Windows MPI (MS-MPI),” on page 69.

2 Installation

2.1 Hardware and Software Requirements

Table 5 - Hardware and Software Requirements

Description[a]	Package
Windows Server 2012 R2 (64 bit only)	MLNX_WinOF2-1_10_All_x64.exe
Windows Server 2012 (64 bit only)	MLNX_WinOF2-1_10_All_x64.exe

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

2.2 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.

2.2.1 Attended Installation

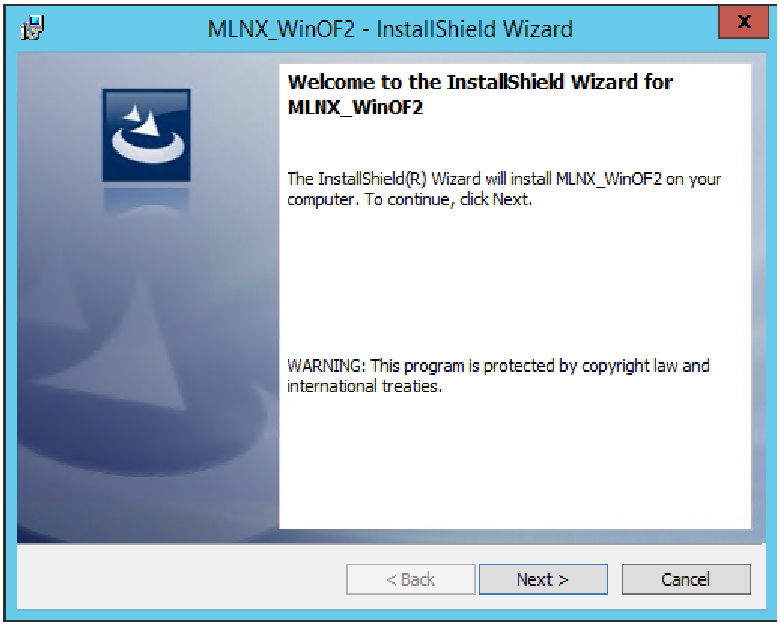
The following is an example of an installation session.

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_10_All_x64.exe /v"/l*vx [LogFile]"
```

Step 3. Click Next in the Welcome screen.



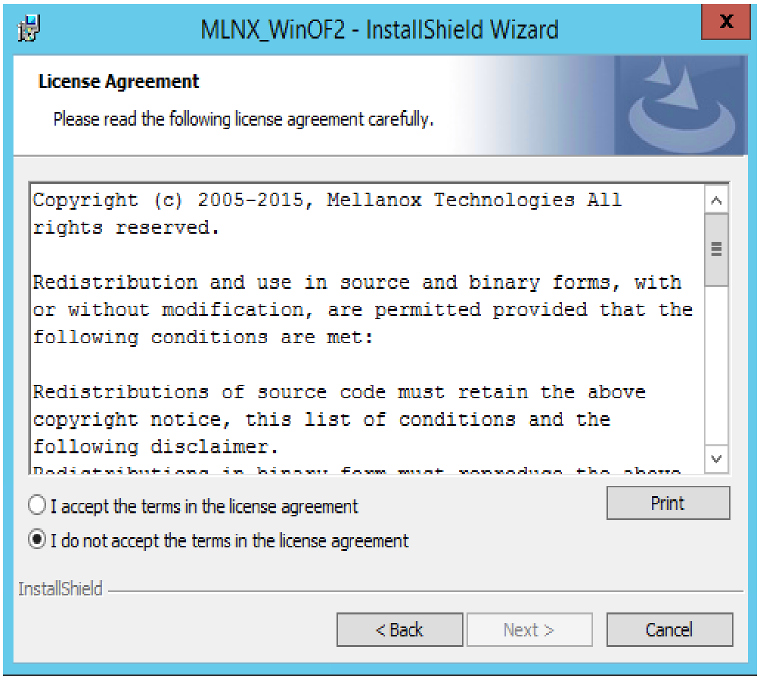
MLNX_WinOF2 - InstallShield Wizard

Welcome to the InstallShield Wizard for MLNX_WinOF2

The InstallShield(R) Wizard will install MLNX_WinOF2 on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

Step 4. Read then accept the license agreement and click Next.



MLNX_WinOF2 - InstallShield Wizard

License Agreement

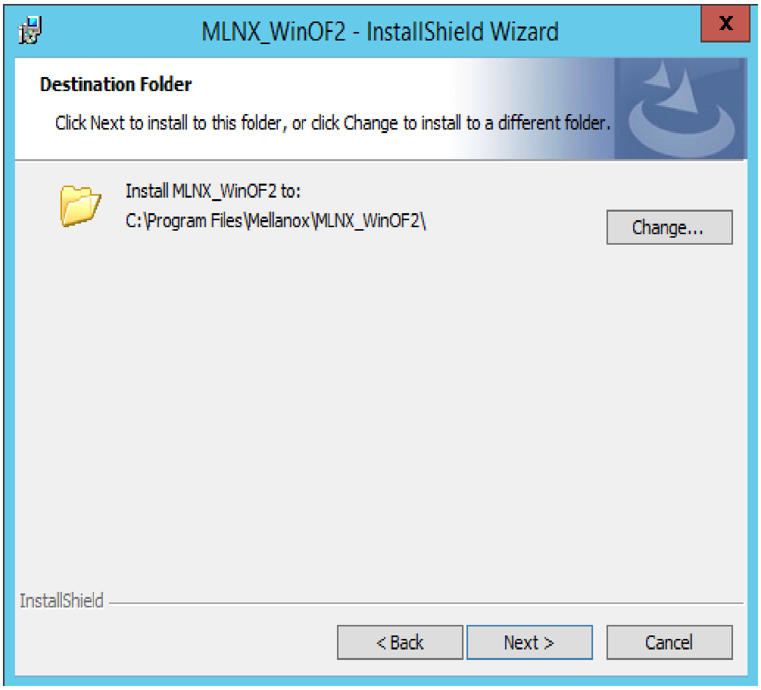
Please read the following license agreement carefully.

```
Copyright (c) 2005-2015, Mellanox Technologies All
rights reserved.

Redistribution and use in source and binary forms, with
or without modification, are permitted provided that the
following conditions are met:

Redistributions of source code must retain the above
copyright notice, this list of conditions and the
following disclaimer.
```

○ I accept the terms in the license agreement

● I do not accept the terms in the license agreement

Step 5. Select the target folder for the installation.



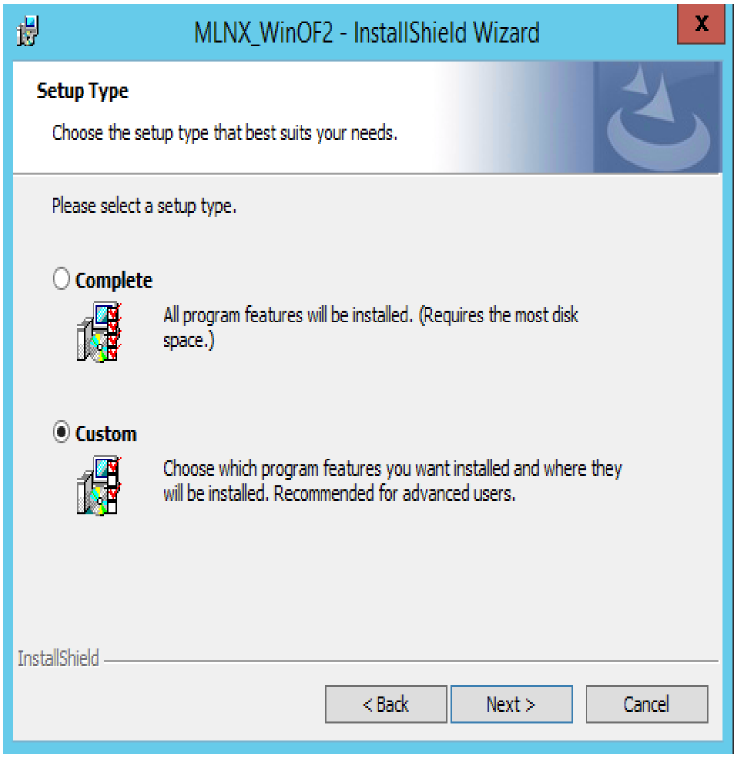
MLNX_WinOF2 - InstallShield Wizard

Destination Folder

Click Next to install to this folder, or click Change to install to a different folder.

Install MLNX_WinOF2 to:
C:\Program Files\Mellanox\MLNX_WinOF2\

Step 6. Select a Complete or Custom installation, follow Step a and on, on page 16.



MLNX_WinOF2 - InstallShield Wizard

Setup Type

Choose the setup type that best suits your needs.

Please select a setup type.

○ **Complete**

All program features will be installed. (Requires the most disk space.)

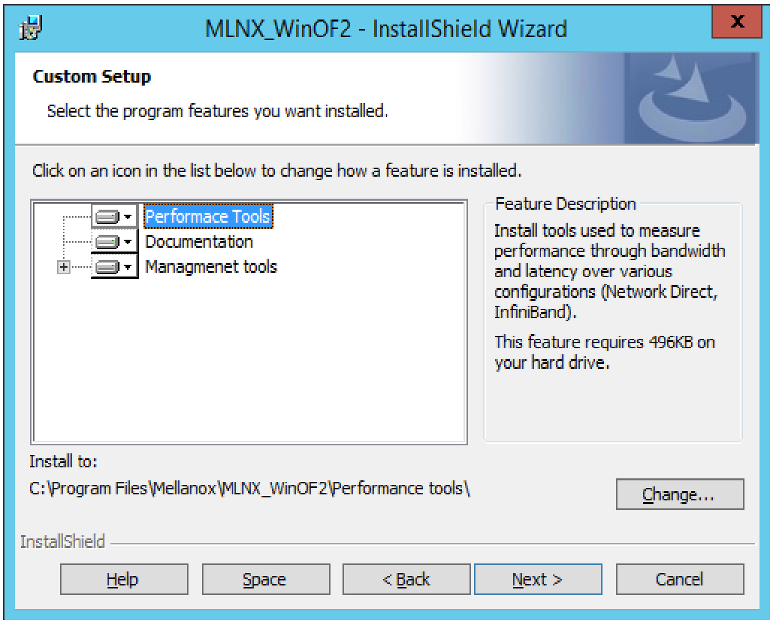
● **Custom**

Choose which program features you want installed and where they will be installed. Recommended for advanced users.

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.



MLNX_WinOF2 - InstallShield Wizard

Custom Setup

Select the program features you want installed.

Click on an icon in the list below to change how a feature is installed.

- Performace Tools
- Documentation
- Managmenet tools

Feature Description

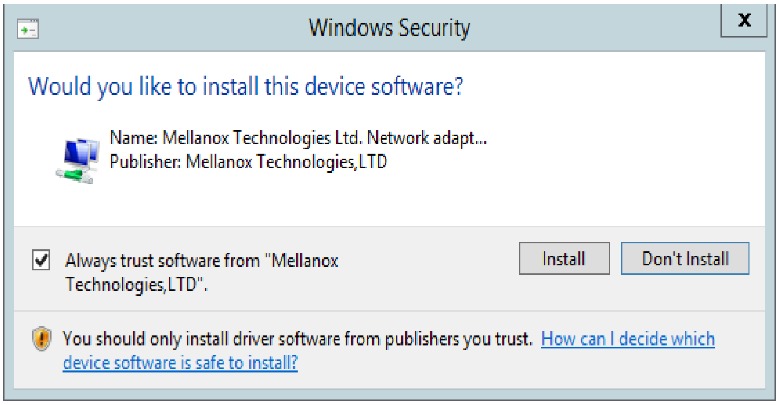
Install tools used to measure performance through bandwidth and latency over various configurations (Network Direct, InfiniBand).

This feature requires 496KB on your hard drive.

Install to:

C:\Program Files\Mellanox\MLNX_WinOF2\Performance tools\

b. Confirm the start of the installation:



Windows Security

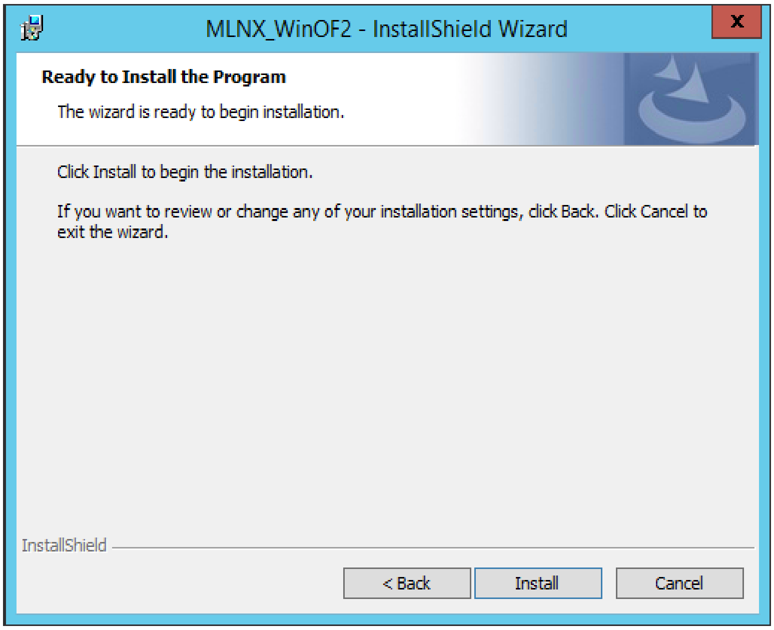
Would you like to install this device software?

Name: Mellanox Technologies Ltd. Network adapt...
Publisher: Mellanox Technologies,LTD

☑ Always trust software from "Mellanox Technologies,LTD".

You should only install driver software from publishers you trust. How can I decide which device software is safe to install?

c. Click Install to start the installation.



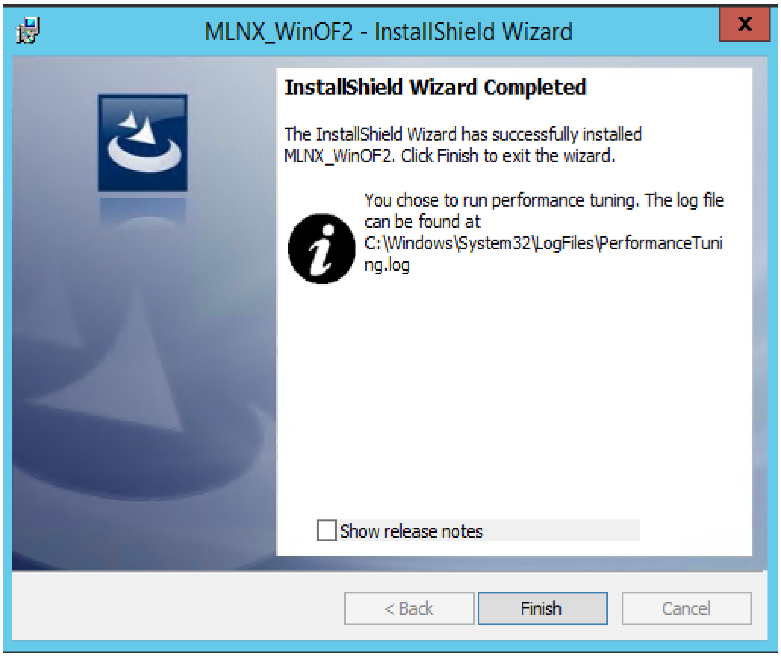
Ready to Install the Program

The wizard is ready to begin installation.

Click Install to begin the installation.

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

Step 7. Click Finish to complete the installation.



InstallShield Wizard Completed

The InstallShield Wizard has successfully installed MLNX_WinOF2. Click Finish to exit the wizard.

You chose to run performance tuning. The log file can be found at C:\Windows\System32\LogFiles\PerformanceTuning.log

□ Show release notes

2.2.2 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 **[Windows Server 2012 R2]** - Click Start --> Task Manager-->File --> Run new task --> and enter CMD.

Step 2. Install the driver. Run:

```
> MLNX_WinOF2-1_10_All_x64.exe /S /v"/qn"
```

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

```
> MLNX_WinOF2-1_10_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_10_All_win2012_x64.exe /vMT_NDPROPERTY=1
```

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

[1] MT_NDPROPERTY default value is True

2.3 Installation Results

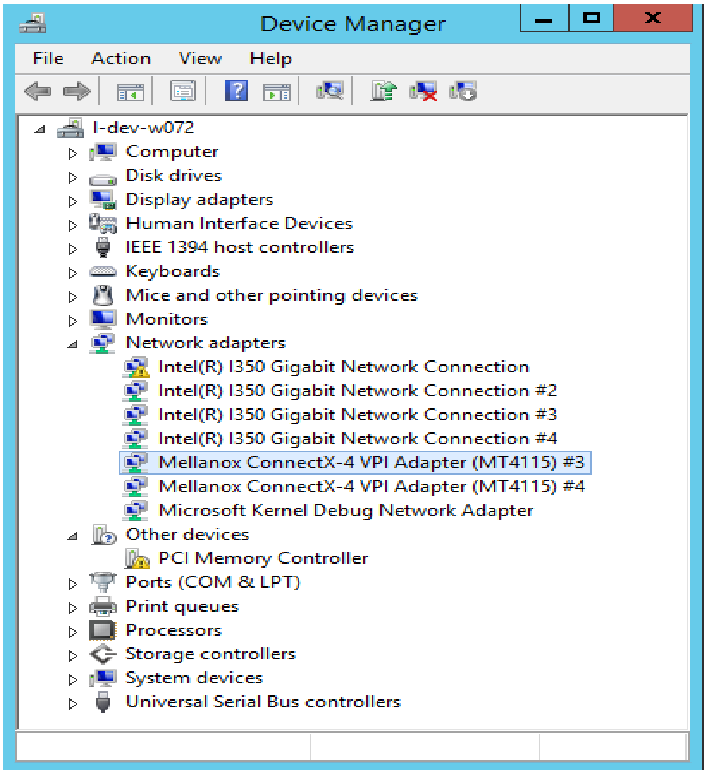
Upon installation completion, you can verify the successful addition of the network card(s) through the Device Manager.

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

- %ProgramFiles%\Mellanox\MLNX_WinOF2\ETH

To see the Mellanox network adapter device, and the Ethernet or IPoIB network device (depending on the used card) for each port, display the Device Manager and expand “Network adapters”.

Figure 1: Installation Results



```
l-dev-w072
  Computer
  Disk drives
  Display adapters
  Human Interface Devices
  IEEE 1394 host controllers
  Keyboards
  Mice and other pointing devices
  Monitors
  Network adapters
    Intel(R) I350 Gigabit Network Connection
    Intel(R) I350 Gigabit Network Connection #2
    Intel(R) I350 Gigabit Network Connection #3
    Intel(R) I350 Gigabit Network Connection #4
    Mellanox ConnectX-4 VPI Adapter (MT4115) #3
    Mellanox ConnectX-4 VPI Adapter (MT4115) #4
    Microsoft Kernel Debug Network Adapter
  Other devices
    PCI Memory Controller
  Ports (COM & LPT)
  Print queues
  Processors
  Storage controllers
  System devices
  Universal Serial Bus controllers
```

2.4 Extracting Files Without Running Installation

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

Step 1. Open a CMD console **[Windows Server 2012 R2]** - Click Start --> Task Manager-->File --> Run new task --> and enter CMD.

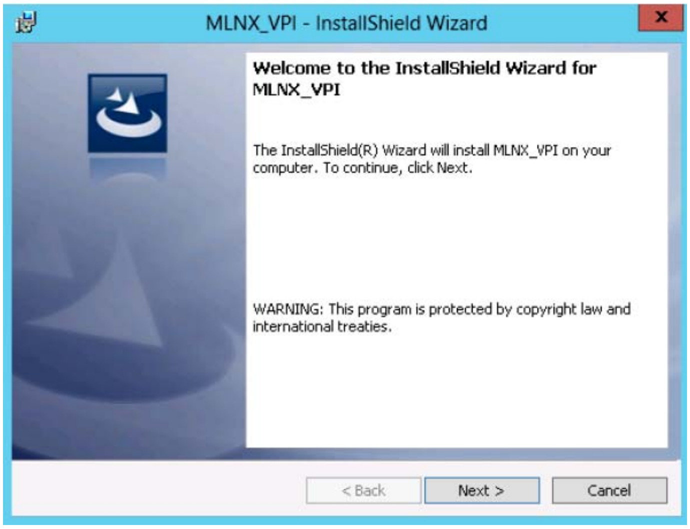
Step 2. Extract the driver and the tools:

```
> MLNX_WinOF2-1_10_All_x64 /a
```

- To extract only the driver files.

```
> MLNX_WinOF2-1_10_All_x64 /a /vMT_DRIVERS_ONLY=1
```

Step 3. Click Next to create a server image.



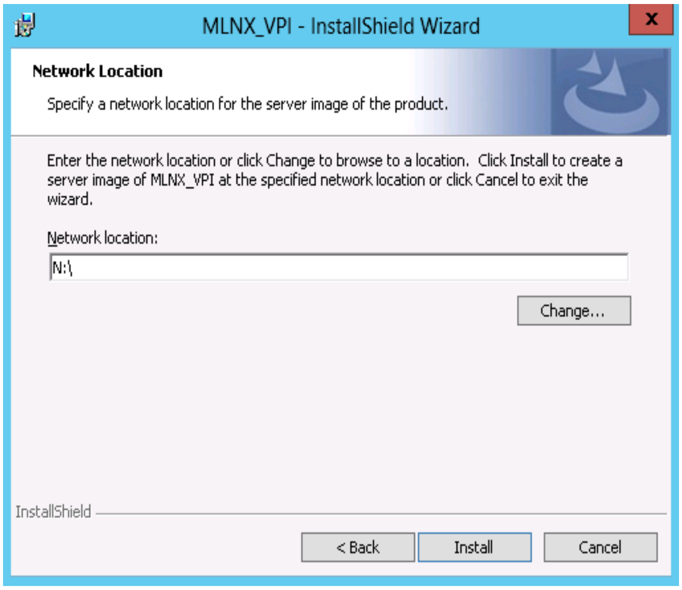
MLNX_VPI - InstallShield Wizard

Welcome to the InstallShield Wizard for MLNX_VPI

The InstallShield(R) Wizard will install MLNX_VPI on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

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



MLNX_VPI - InstallShield Wizard

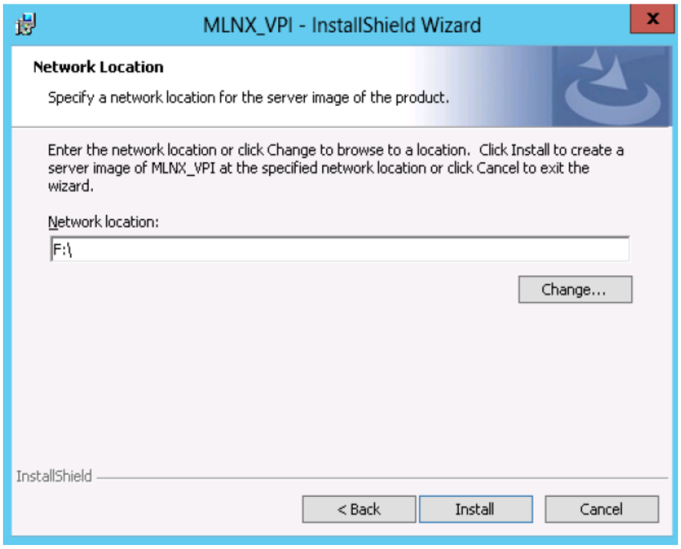
Network Location

Specify a network location for the server image of the product.

Enter the network location or click Change to browse to a location. Click Install to create a server image of MLNX_VPI at the specified network location or click Cancel to exit the wizard.

Network location:
N:\

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



MLNX_VPI - InstallShield Wizard

Network Location

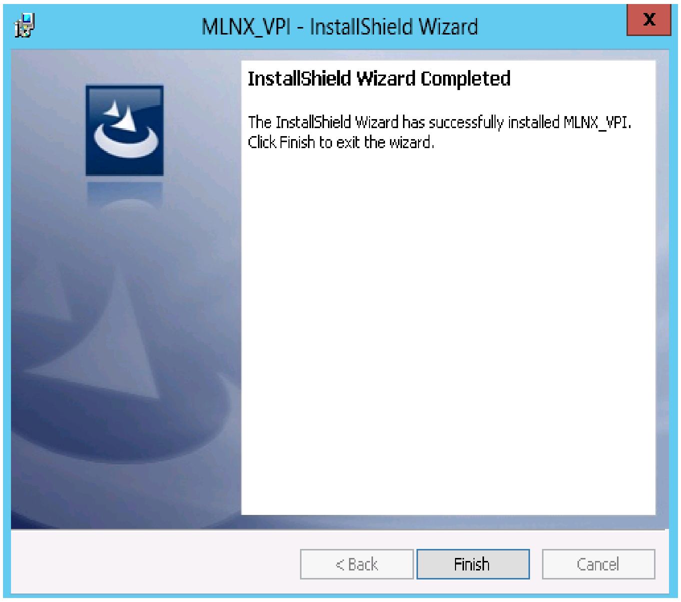
Specify a network location for the server image of the product.

Enter the network location or click Change to browse to a location. Click Install to create a server image of MLNX_VPI at the specified network location or click Cancel to exit the wizard.

Network location:

F:\

Step 6. To complete the extraction, click Finish.



MLNX_VPI - InstallShield Wizard

InstallShield Wizard Completed

The InstallShield Wizard has successfully installed MLNX_VPI. Click Finish to exit the wizard.

2.5 Uninstalling Mellanox WinOF-2 Driver

2.5.1 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 11 for details.)

2.5.2 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_WinOF in unattended mode:***

Step 1. Open a CMD console **[Windows Server 2012 R2]** - Click Start --> Task Manager-->File --> Run new task --> and enter CMD.

Step 2. Uninstall the driver. Run:

```
> MLNX_WinOF2-1_10_All_win2012_x64.exe /S /x /v"/qn"
```

2.6 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

The adapter card may not have been shipped with the latest firmware version. The section below describes how to update firmware.

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 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

[Windows Server 2012 R2] - Click Start --> Task Manager-->File --> Run new task --> and enter CMD.

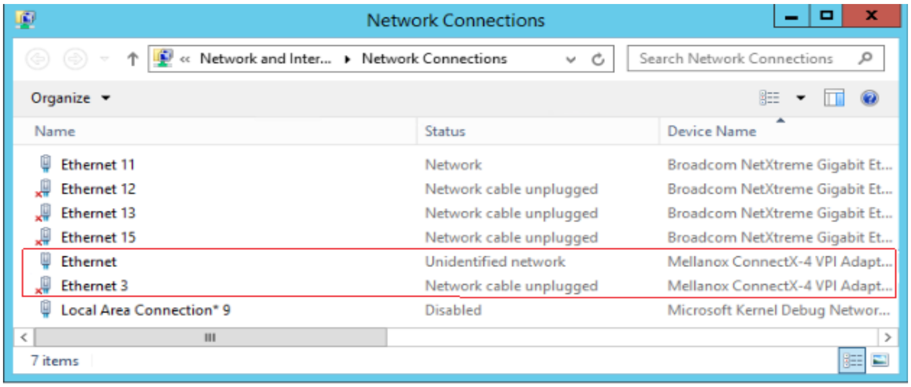
Step 2. Display the MAC address as “Physical Address”

```
> ipconfig /all
```

Configuring a static IP is the same for Ethernet adapters.

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

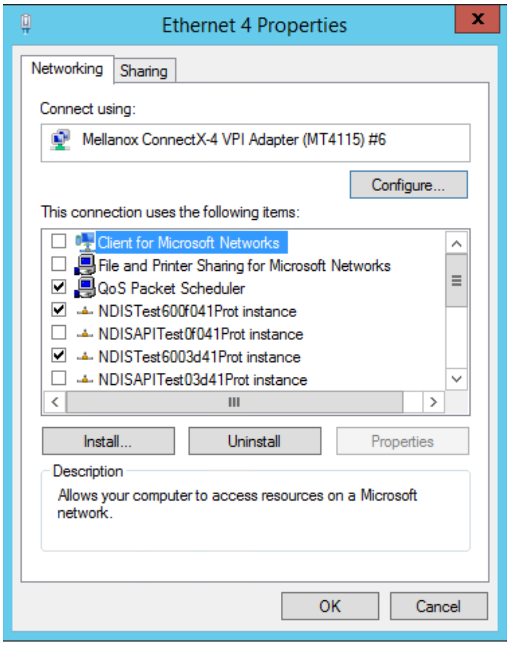
Step 1. Open the Network Connections window. Locate Local Area Connections with Mellanox devices.



Network Connections
Network and Inter... ▸ Network Connections
Search Network Connections
Organize ▾
7 items

Name	Status	Device Name
Ethernet 11	Network	Broadcom NetXtreme Gigabit Et...
Ethernet 12	Network cable unplugged	Broadcom NetXtreme Gigabit Et...
Ethernet 13	Network cable unplugged	Broadcom NetXtreme Gigabit Et...
Ethernet 15	Network cable unplugged	Broadcom NetXtreme Gigabit Et...
Ethernet	Unidentified network	Mellanox ConnectX-4 VPI Adapt...
Ethernet 3	Network cable unplugged	Mellanox ConnectX-4 VPI Adapt...
Local Area Connection* 9	Disabled	Microsoft Kernel Debug Networ...

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



Ethernet 4 Properties

Networking | Sharing

Connect using:

Mellanox ConnectX-4 VPI Adapter (MT4115) #6

Configure...

This connection uses the following items:

- □ Client for Microsoft Networks
- □ File and Printer Sharing for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ NDISTest600f041Prot instance
- □ NDISAPITest0f041Prot instance
- ☑ NDISTest6003d41Prot instance
- □ NDISAPITest03d41Prot instance

Install... | Uninstall | Properties

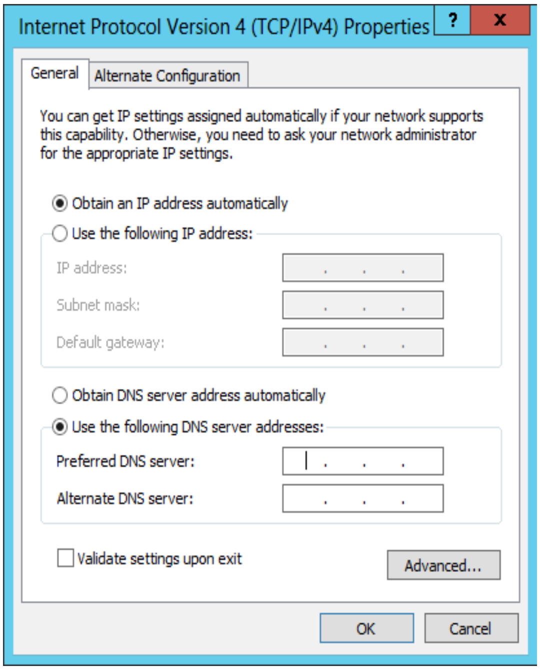
Description

Allows your computer to access resources on a Microsoft network.

OK | Cancel

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.



Internet Protocol Version 4 (TCP/IPv4) Properties

General | Alternate Configuration

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

● Obtain an IP address automatically

○ Use the following IP address:

IP address: . . .

Subnet mask: . . .

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

□ Validate settings upon exit

Advanced...

OK | Cancel

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 . . . . . . . . . :
...
```

3.1.2 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.

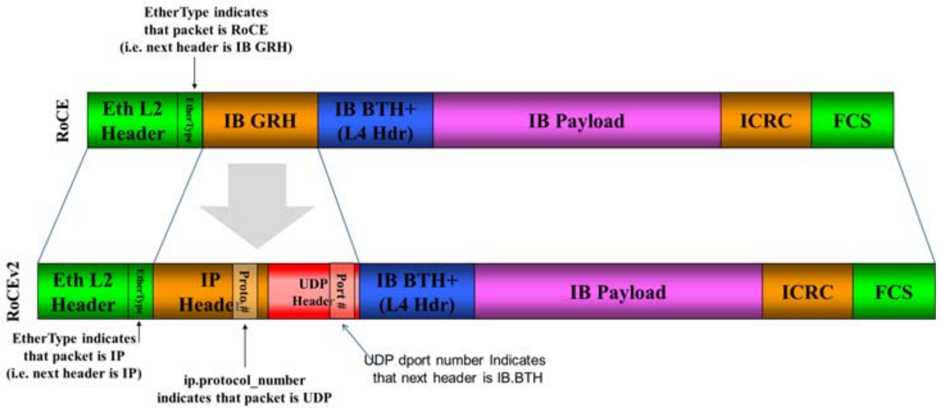
3.1.2.1 IP Routable (RoCEv2)

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 2: RoCE and RoCE v2 Frame Format Differences



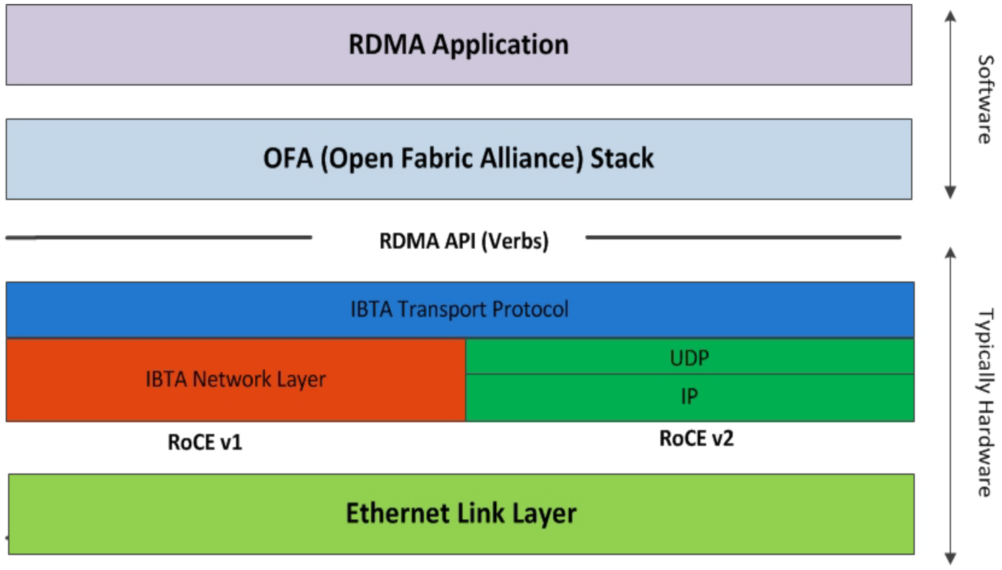
The proposed RoCEv2 packets use a well-known UDP destination port value that unequivocally distinguishes the datagram. Similar to other protocols that use UDP encapsulation, the UDP source port field is used to carry an opaque flow-identifier that allows network devices to implement packet forwarding 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 2,“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 3: RoCE and RoCEv2 Protocol Stack



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

The default RoCE mode in Windows is MAC based.
The default RoCE mode in Linux is IP based.
In order to communicate between Windows and Linux over RoCE, please use RoCE v2 (the default mode for Windows).

3.1.2.2 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.

For further information about RoCE configuration, please refer to:
https://community.mellanox.com/docs/DOC-1844

3.1.2.2.1 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 3.1.4, “Configuring Quality of Service (QoS)”, on page 32

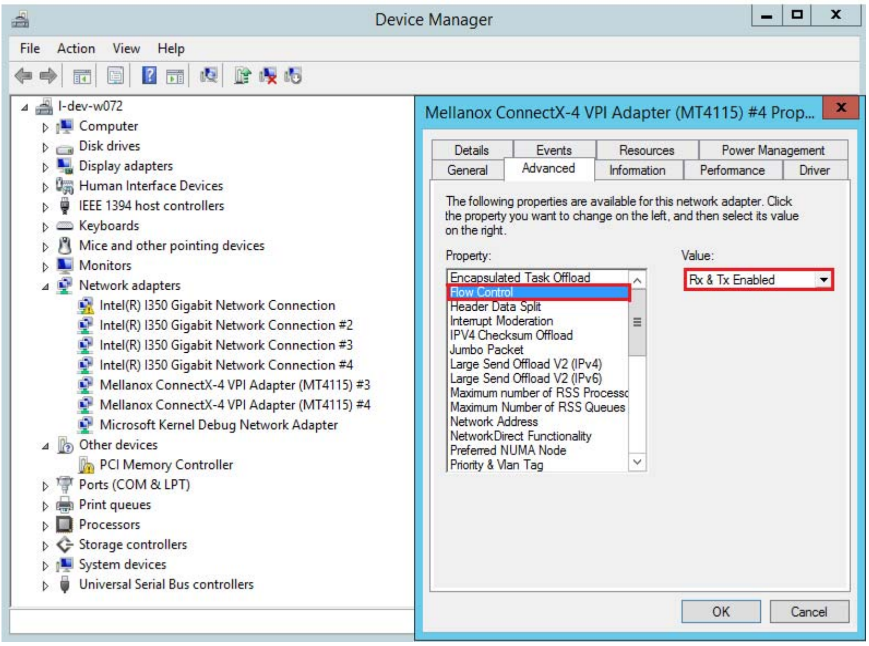
3.1.2.2.1.1 Global Pause (Flow Control)

➢ ***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 Ethernet Adapter-> Properties ->Advanced tab



Device Manager
File Action View Help
l-dev-w072
Computer
Disk drives
Display adapters
Human Interface Devices
IEEE 1394 host controllers
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Intel(R) I350 Gigabit Network Connection
Intel(R) I350 Gigabit Network Connection #2
Intel(R) I350 Gigabit Network Connection #3
Intel(R) I350 Gigabit Network Connection #4
Mellanox ConnectX-4 VPI Adapter (MT4115) #3
Mellanox ConnectX-4 VPI Adapter (MT4115) #4
Microsoft Kernel Debug Network Adapter
Other devices
PCI Memory Controller
Ports (COM & LPT)
Print queues
Processors
Storage controllers
System devices
Universal Serial Bus controllers
Mellanox ConnectX-4 VPI Adapter (MT4115) #4 Prop...
Details
Events
Resources
Power Management
General
Advanced
Information
Performance
Driver
Encapsulated Task Offload
Flow Control
Header Data Split
Interrupt Moderation
IPV4 Checksum Offload
Jumbo Packet
Large Send Offload V2 (IPv4)
Large Send Offload V2 (IPv6)
Maximum number of RSS Processo
Maximum Number of RSS Queues
Network Address
NetworkDirect Functionality
Preferred NUMA Node
Priority & Vlan Tag
OK
Cancel

The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property: Flow Control

Value: Rx & Tx Enabled

3.1.2.3 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.

3.1.2.4 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
```

3.1.2.4.1 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
```

3.1.2.4.2 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
```

3.1.2.5 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.

3.1.2.5.1 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.

3.1.2.6 Configuring the RoCE Mode

Configuring the RoCE mode requires the following:

- RoCE mode is configured per-driver and is enforced on all the 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 mode can be enabled and disabled either via the registry key or the PowerShell.

➢ ***RoCE is enabled by default. To enable it 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.

3.1.2.6.1 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: No RoCE

3.1.3 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

Note that the Microsoft teaming mechanism is only available on Windows Server distributions.

3.1.3.1 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.

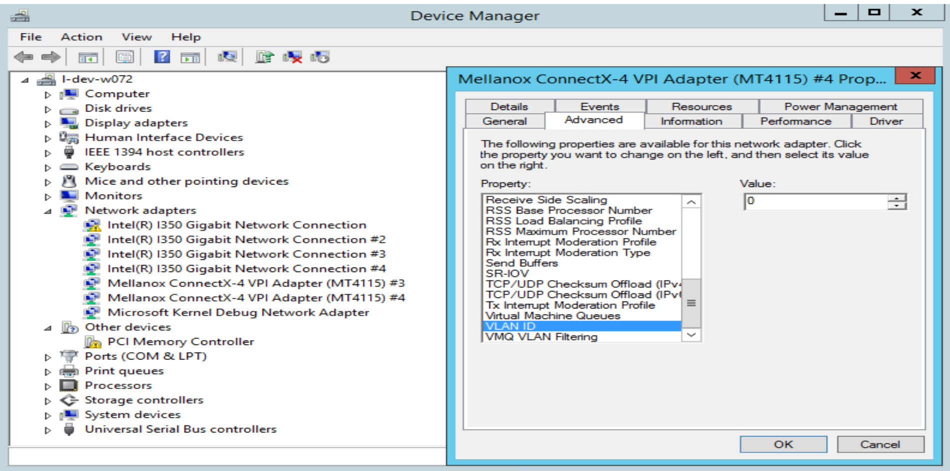
Step 2. Go to the Network adapters.

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.


Device Manager
File
Action
View
Help
l-dev-w072
Computer
Disk drives
Display adapters
Human Interface Devices
IEEE 1394 host controllers
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Intel(R) I350 Gigabit Network Connection
Intel(R) I350 Gigabit Network Connection #2
Intel(R) I350 Gigabit Network Connection #3
Intel(R) I350 Gigabit Network Connection #4
Mellanox ConnectX-4 VPI Adapter (MT4115) #3
Mellanox ConnectX-4 VPI Adapter (MT4115) #4
Microsoft Kernel Debug Network Adapter
Other devices
PCI Memory Controller
Ports (COM & LPT)
Print queues
Processors
Storage controllers
System devices
Universal Serial Bus controllers
Mellanox ConnectX-4 VPI Adapter (MT4115) #4 Prop...
Details
Events
Resources
Power Management
General
Advanced
Information
Performance
Driver
Receive Side Scaling
RSS Base Processor Number
RSS Load Balancing Profile
RSS Maximum Processor Number
Rx Interrupt Moderation Profile
Rx Interrupt Moderation Type
Send Buffers
SR-IOV
TCP/UDP Checksum Offload (IPv
TCP/UDP Checksum Offload (IPv
Tx Interrupt Moderation Profile
Virtual Machine Queues
VMQ VLAN Filtering
OK
Cancel

The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property: VLAN ID

Value: 0

3.1.4 Configuring Quality of Service (QoS)

3.1.4.1 System Requirements

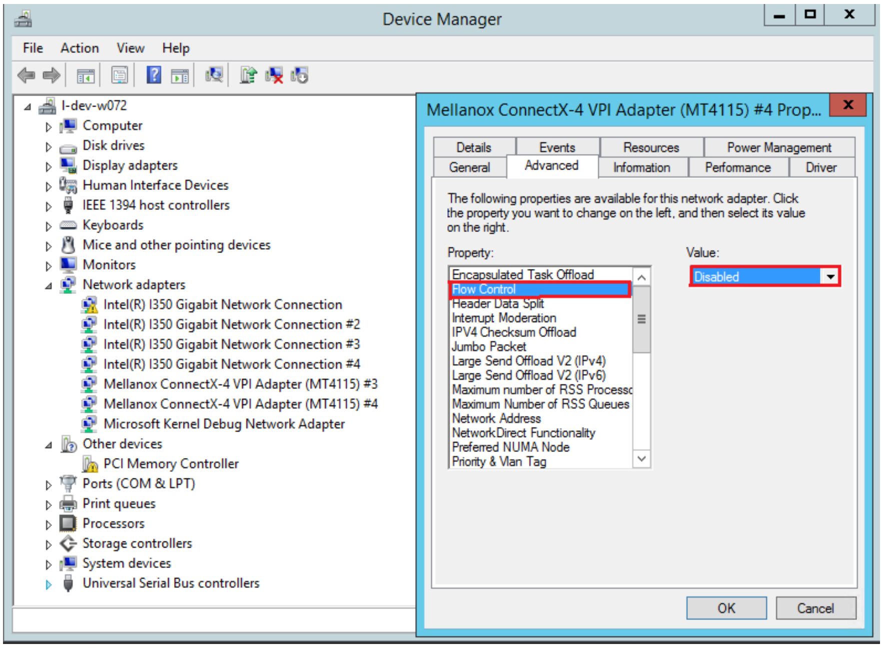
Operating Systems: Windows Server 2012, and Windows Server 2012 R2

3.1.4.2 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 Ethernet Adapter->Properties->Advanced tab


Device Manager
File
Action
View
Help
I-dev-w072
Computer
Disk drives
Display adapters
Human Interface Devices
IEEE 1394 host controllers
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Intel(R) I350 Gigabit Network Connection
Intel(R) I350 Gigabit Network Connection #2
Intel(R) I350 Gigabit Network Connection #3
Intel(R) I350 Gigabit Network Connection #4
Mellanox ConnectX-4 VPI Adapter (MT4115) #3
Mellanox ConnectX-4 VPI Adapter (MT4115) #4
Microsoft Kernel Debug Network Adapter
Other devices
PCI Memory Controller
Ports (COM & LPT)
Print queues
Processors
Storage controllers
System devices
Universal Serial Bus controllers
Mellanox ConnectX-4 VPI Adapter (MT4115) #4 Prop...
Details
Events
Resources
Power Management
General
Advanced
Information
Performance
Driver
OK
Cancel

The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property:

- Encapsulated Task Offload
- Flow Control
- Header Data Split
- Interrupt Moderation
- IPV4 Checksum Offload
- Jumbo Packet
- Large Send Offload V2 (IPv4)
- Large Send Offload V2 (IPv6)
- Maximum number of RSS Processc
- Maximum Number of RSS Queues
- Network Address
- NetworkDirect Functionality
- Preferred NUMA Node
- Priority & Vlan Tag

Value: Disabled

➢ ***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 -Confirm:$false
PS $ New-NetIPAddress -InterfaceAlias “Ethernet 4” -IPAddress 192.168.1.10 -PrefixLength 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
```

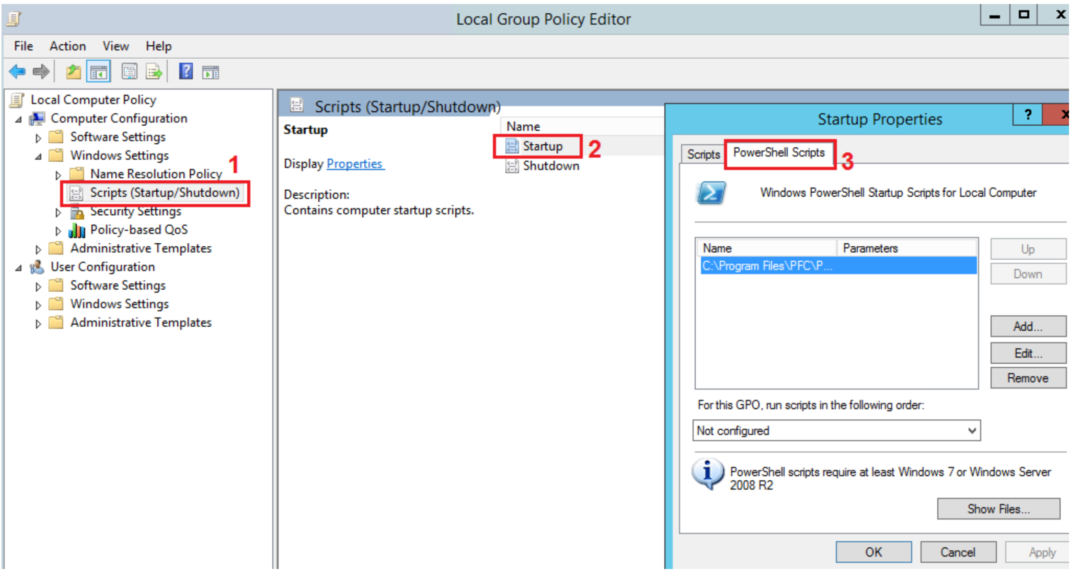
Step 2. In the pop-up window, under the 'Computer Configuration' section, perform the following:

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
```

3.1.5 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 Off-load Options, Performance Options, or Flow Control Options, you’ll need to click the Properties button to modify parameters via a pop-up dialog.

3.1.6 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 3.3.1, “Finding the Index Value of the Network Interface”, on page 39.

Table 7 - 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.

3.2 Storage Protocols

3.2.1 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.

3.2.1.1 SMB Configuration Verification

3.2.1.1.1 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
```

3.2.1.1.2 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"}
```

3.2.1.1.3 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.

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.

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

3.2.1.2 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

3.3 Configuration Using Registry Keys

3.3.1 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:

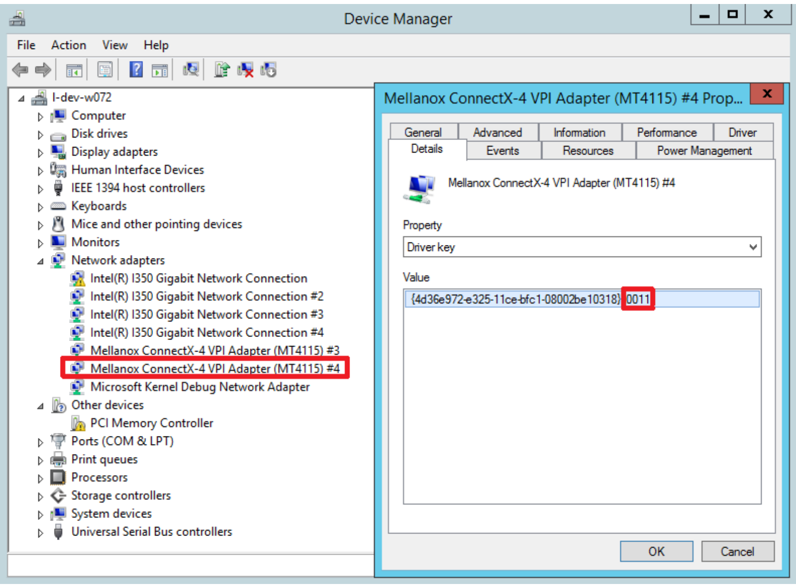
Step 1. Open Device Manager, and go to Network Adapters.

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



Device Manager
File
Action
View
Help
l-dev-w072
Computer
Disk drives
Display adapters
Human Interface Devices
IEEE 1394 host controllers
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Intel(R) I350 Gigabit Network Connection
Intel(R) I350 Gigabit Network Connection #2
Intel(R) I350 Gigabit Network Connection #3
Intel(R) I350 Gigabit Network Connection #4
Mellanox ConnectX-4 VPI Adapter (MT4115) #3
Mellanox ConnectX-4 VPI Adapter (MT4115) #4
Microsoft Kernel Debug Network Adapter
Other devices
PCI Memory Controller
Ports (COM & LPT)
Print queues
Processors
Storage controllers
System devices
Universal Serial Bus controllers
Mellanox ConnectX-4 VPI Adapter (MT4115) #4 Prop...
General
Advanced
Information
Performance
Driver
Details
Events
Resources
Power Management
OK
Cancel

Mellanox ConnectX-4 VPI Adapter (MT4115) #4

Property: Driver key

Value: {4d36e972-e325-11ce-bfc1-08002be10318}0011

3.3.2 Basic Registry Keys

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

Value Name	Default Value	Description
*JumboPacket	1514	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 **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.
*SpeedDuplex	7	The Speed and Duplex settings that a device supports. This registry key should not be changed and it can be used to query the device capability. Mellanox ConnectX device is set to 7 meaning10Gbps and Full Duplex. **Note**: Default value should not be modified.

Value Name	Default Value	Description
RxIntModerationProfile	2	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.
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.

3.3.3 Off-load 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.

Value Name	Default Value	Description
*LsoV1IPv4	1	Large Send Offload Version 1 (IPv4). The valid values are: • 0: disable • 1: enable
*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.

Value Name	Default Value	Description
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.
*IPChecksumOffloadIPv4	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)
*TCPUDPChecksumOffloadIPv4	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)
*TCPUDPChecksumOffloadIPv6	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)

3.3.4 Performance Registry Keys

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

Value Name	Default Value	Description
RecvCompletion-Method	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. An interrupt is generated after receiving 5 packets or after the passing of 10 micro seconds from receiving the first packet. 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.

Value Name	Default Value	Description
*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. **Note**: the I/O Acceleration Technology (IOAT) is not functional in this mode.
ReturnPacketThreshold	341	The allowed number of free received packets on the rings. Any number above it will cause the driver to return the packet to the hardware immediately. When the value is set to 0, the adapter uses 2/3 of the received ring size. The valid values are: 0 to 4096. **Note**: This registry value is not exposed via the UI.
NumTcb	16	The number of send buffers that the driver allocates for sending purposes. Each buffer is in LSO size, if LSO is enabled, or in MTU size, otherwise. The valid values are 1 up to 64. **Note**: This registry value is not exposed via the UI.
ThreadPoll	10000	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.

Value Name	Default Value	Description
AverageFactor	16	The weight of the last polling in the decision whether to continue the polling or give up when using polling completion method for receiving. The valid values are 0 up to 256. **Note**: This registry value is not exposed via the UI.
AveragePollThreshold	10	The average threshold polling number when using polling completion method for receiving. If the average number is higher than this value, the adapter continues to poll. The valid values are 0 up to 1000. **Note**: This registry value is not exposed via the UI.
ThisPollThreshold	100	The threshold number of the last polling cycle when using polling completion method for receiving. If the number of packets received in the last polling cycle is higher than this value, the adapter continues to poll The valid values are 0 up to 1000. **Note**: This registry value is not exposed via the UI.
VlanId	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.

Value Name	Default Value	Description
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.

3.3.5 Ethernet Registry Keys

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

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.

Value Name	Default Value	Description
PromiscuousVlan	0	Specifies whether a promiscuous VLAN is enabled or not. When this parameter is set, all the packets with VLAN tags are passed to an upper level without executing any filtering. The valid values are: • 0: disable • 1: enable **Note**: This registry value is not exposed via the UI.

3.3.5.1 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.

Value Name	Default Value	Description
*FlowControl	0	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

3.3.5.2 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

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

Value Name	Default Value	Description
*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

3.3.6 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 InfiniBand application. NDI exposes the advanced capabilities of the Mellanox networking devices and allows applications to leverage advances of InfiniBand.

For further information please refer to:

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

3.4 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.

3.4.1 General Performance Optimization and Tuning

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

3.4.1.1 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.
```

3.4.1.2 Enable RSS

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

```
“netsh int tcp set global rss = enabled”
```

3.4.1.3 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\RequireSecuritySignature
```

3.4.2 Application Specific Optimization and Tuning

3.4.2.1 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.

3.4.2.1.1 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.

3.4.3 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 1024).

- **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.

  - **Interrupt Method**

    Optimizes the CPU as it uses interrupts for handling incoming messages. However, in certain scenarios it can decrease the network throughput.

  - **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.

- Interrupt Moderation RX Packet Count

  Number of packets that need to be received before an interrupt is generated on the receive side (default 5).

- Interrupt Moderation RX Packet Time

  Maximum elapsed time (in usec) between the receiving of a packet and the generation of an interrupt, even if the moderation count has not been reached (default 10).

- 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.

- Interrupt Moderation TX Packet Count

  Number of packets that need to be sent before an interrupt is generated on the send side (default 0).

- Interrupt Moderation TX Packet Time

  Maximum elapsed time (in usec) between the sending of a packet and the generation of an interrupt even if the moderation count has not been reached (default 0).

- **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.

3.4.4 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 counters hold the standard Windows CounterSet API that includes:

- Network Interface
- RDMA activity
- SMB Direct Connection

3.4.4.0.1 RDMA Activity

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 8 - RDMA Activity

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.

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 9 - 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.

Utility	Description
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.
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.
NTttcp	NTttcp is a Windows base testing application that sends and receives TCP data between two or more endpoints. It is a Winsock-based port of the ttcp tool that measures networking performance bytes/second. To download the latest version of NTttcp (5.28), please refer to Microsoft website following the link below: http://gallery.technet.microsoft.com/NTttcp-Version-528-Now-f8b12769 **NOTE**: This tool should be run from cmd only.

The following InfiniBand performance tests are deprecated and might be removed in future releases.

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 10 - 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.
The installation of WinOF-2 fails and reads as follows: `"The installation cannot be done while the RDSH service is enabled, please disable it. You may re-enable it after the installation is complete".`	A known issue in windows installer when using the chain MSI feature, as described in the following link: http://rcmtech.wordpress.com/2013/08/27/server-2012-remote-desktop-session-host-installation-hangs-at-windows-installer-coordinator/	Follow the recommendation in the article.

5.1.1 Installation Error Codes and Troubleshooting

5.1.1.1 Setup Return Codes

Table 11 - 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 12 - 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.
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 13 - 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 Ethernet Related Troubleshooting

For further performance related information, please refer to the *Performance Tuning Guide* and to Section 3.4, “Performance Tuning and Counters”, on page 50

Table 14 - Ethernet Related Issues

Issue	Cause	Solution
Low performance.	Non-optimal system configuration might have occurred.	See section “Performance Tuning and Counters” on page 50. to take advantage of Mellanox 10/40/56 GBit NIC performance.

Table 14 - Ethernet Related Issues

Issue	Cause	Solution
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 `"mlx4ethX"` 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"`.
The driver fails to start and a yellow sign appears near the `"Mellanox ConnectX 10Gb Ethernet Adapter"` in the Device Manager display. (Code 10)	A hardware error might have occurred.	Disable and re-enable `"Mellanox ConnectX Adapter"` from the Device Manager display. In case it does not work, refer to support.
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 SR-IOV-enabled virtual switch.

5.3 Performance Related Troubleshooting

Table 15 - 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.

5.3.1 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 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

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 127.0.0.1"
2. Run "start /b /affinity 0x2 nd_send_bw -C 127.0.0.1"
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 -C <IP_host>" on machine 1 where <IP_host1> is the local IP.
2. Run "nd_send_bw -C <IP_host>" on machine 2.
3. Results appear in MB/s (Mega Bytes 2^20), 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 3.1.2,“RDMA over Converged Ethernet (RoCE),” on page 26
- CPU/power options are not set to "Maximum Performance".

5.4 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 Ethernet Adapter <X> has been successfully initialized and enabled.
   - Failed to initialize Mellanox ConnectX Ethernet Adapter.
   - Mellanox ConnectX Ethernet Adapter <X> has been successfully initialized and enabled. The port's network address is <MAC Address>
   - The Mellanox ConnectX Ethernet was reset.
   - Failed to reset the Mellanox ConnectX Ethernet NIC. Try disabling then re-enabling the "Mellanox Ethernet Bus Driver" device via the Windows device manager.
   - Mellanox ConnectX Ethernet Adapter <X> has been successfully stopped.
   - Failed to initialize the Mellanox ConnectX Ethernet Adapter <X> because it uses old firmware version (<old firmware version>). You need to burn firmware version <new firmware version> or higher, and to restart your computer.
   - Mellanox ConnectX Ethernet Adapter <X> device detected that the link connected to port <Y> is up, and has initiated normal operation.
   - Mellanox ConnectX Ethernet Adapter <X> device detected that the link connected to port <Y> is down. This can occur if the physical link is disconnected or damaged, or if the other end-port is down.
   - Mismatch in the configurations between the two ports may affect the performance. When Using MSI-X, both ports should use the same RSS mode. To fix the problem, configure the RSS mode of both ports to be the same in the driver GUI.
   - Mellanox ConnectX Ethernet Adapter <X> device failed to create enough MSI-X vectors. The Network interface will not use MSI-X interrupts. This may affects the performance. To fix the problem, configure the number of MSI-X vectors in the registry to be at least <Y>

Appendix A: Performance Tools

A.1 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_bw Synopsis

```
<running on specific single core>
Server side: start /b /affinity 0X1 nd_write_bw -s1048576 -D10 -S 11.137.53.1
Client side: start /b /wait /affinity 0X1 nd_write_bw -s1048576 -D10 -C 11.137.53.1
```

nd_write_bw Options

The table below lists the various flags of the command.

Table 16 - nd_write_bw Flags and Options

Flag	Description
-h	Shows the Help screen.
-v	Shows the version number.
-p	Connects to the port <port> <default 6830>.
-s <msg size>	Exchanges the message size with <default 65536B>, and it must not be combined with -a flag.
-a	Runs all the messages’ sizes from 1B to 8MB, and it must not be combined with -s flag.
-n <num of iterations>	The number of exchanges (at least 2, the default is 100000)
-I <max inline size>	The maximum size of message to send inline. The default number is 128B.
-D <test duration in seconds>	Tests duration in seconds.
-f <margin time in seconds>	The margin time to avoid calculation, and it must be less than half of the duration time.
-Q	CQ-Moderation <value>. The default number is 100.
-S <server interface IP>	<server side only, must be last parameter>
-C <server interface IP>	<client side only, must be last parameter>

A.2 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_write_lat Synopsis

```
<running on specific single core>
Server side: start /b /affinity 0X1 nd_write_lat -s1048576 -D10 -S 11.137.53.1
Client side: start /b /wait /affinity 0X1 nd_write_lat -s1048576 -D10 -C 11.137.53.1
```

nd_write_lat Options

The table below lists the various flags of the command.

Table 17 - nd_write_lat Options

Flag	Description
-h	Shows the Help screen.
-v	Shows the version number.
-p	Connects to the port <port> <default 6830>.
-s <msg size>	Exchanges the message size with <default 65536B>, and it must not be combined with -a flag.
-a	Runs all the messages’ sizes from 1B to 8MB, and it must not be combined with -s flag.
-n <num of iterations>	The number of exchanges (at least 2, the default is 100000)
-I <max inline size>	The maximum size of message to send inline. The default number is 128B.
-D <test duration in seconds>	Tests duration in seconds.
-f <margin time in seconds>	The margin time to avoid calculation, and it must be less than half of the duration time.
-S <server interface IP>	<server side only, must be last parameter>
-C <server interface IP>	<client side only, must be last parameter>
-h	Shows the Help screen.

A.3 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_bw Synopsis

```
<running on specific single core>
Server side: start /b /affinity 0X1 nd_read_bw -s1048576 -D10 -S 11.137.53.1
Client side: start /b /wait /affinity 0X1 nd_read_bw -s1048576 -D10 -C 11.137.53.1
```

nd_read_bw Options

The table below lists the various flags of the command.

Table 18 - nd_read_bw Options

Flags	Description
-h	Shows the Help screen.
-v	Shows the version number.
-p	Connects to the port <port> <default 6830>.
-s <msg size>	Exchanges the message size with <default 65536B>, and it must not be combined with -a flag.
-a	Runs all the messages’ sizes from 1B to 8MB, and it must not be combined with -s flag.
-n <num of iterations>	The number of exchanges (at least 2, the default is 100000)
-I <max inline size>	The maximum size of message to send inline. The default number is 128B.
-D <test duration in seconds>	Tests duration in seconds.
-f <margin time in seconds>	The margin time to avoid calculation, and it must be less than half of the duration time.
-Q	CQ-Moderation <value>. The default number is 100.
-S <server interface IP>	<server side only, must be last parameter>
-C <server interface IP>	<client side only, must be last parameter>
-h	Shows the Help screen.

A.4 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.

nd_read_lat SynopsisSynopsis

```
<running on specific single core>
Server side: start /b /affinity 0X1 nd_read_lat -s1048576 -D10 -S
11.137.53.1
Client side: start /b /wait /affinity 0X1 nd_read_lat -s1048576 -D10 -C
11.137.53.1
```

nd_read_lat Options

The table below lists the various flags of the command.

Table 19 - nd_read_lat Options

Flags	Description
-h	Shows the Help screen.
-v	Shows the version number.
-p	Connects to the port <port> <default 6830>.
-s <msg size>	Exchanges the message size with <default 65536B>, and it must not be combined with -a flag.
-a	Runs all the messages’ sizes from 1B to 8MB, and it must not be combined with -s flag.
-n <num of iterations>	The number of exchanges (at least 2, the default is 100000)
-I <max inline size>	The maximum size of message to send inline. The default number is 128B.
-D <test duration in seconds>	Tests duration in seconds.
-f <margin time in seconds>	The margin time to avoid calculation, and it must be less than half of the duration time.
-S <server interface IP>	<server side only, must be last parameter>
-C <server interface IP>	<client side only, must be last parameter>
-h	Shows the Help screen.

A.5 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_bw Synopsis

```
<running on specific single core>
Server side: start /b /affinity 0X1 nd_send_bw -s1048576 -D10 -S 11.137.53.1
Client side: start /b /wait /affinity 0X1 nd_send_bw -s1048576 -D10 -C
11.137.53.1
```

nd_send_bw Options

The table below lists the various flags of the command.

Table 20 - nd_send_bw Flags and Options

Flag	Description
-h	Shows the Help screen.
-v	Shows the version number.
-p	Connects to the port <port> <default 6830>.
-s <msg size>	Exchanges the message size with <default 65536B>, and it must not be combined with -a flag.
-a	Runs all the messages' sizes from 1B to 8MB, and it must not be combined with -s flag.
-n <num of iterations>	The number of exchanges (at least 2, the default is 100000)
-I <max inline size>	The maximum size of message to send inline. The default number is 128B.
-D <test duration in seconds>	Tests duration in seconds.
-f <margin time in seconds>	The margin time to avoid calculation, and it must be less than half of the duration time.
-Q	CQ-Moderation <value>. The default number is 100.
-S <server interface IP>	<server side only, must be last parameter>
-C <server interface IP>	<client side only, must be last parameter>

A.6 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.

nd_send_lat Synopsis

```
<running on specific single core>
Server side: start /b /affinity 0X1 nd_send_lat -s1048576 -D10 -S
11.137.53.1
Client side: start /b /wait /affinity 0X1 nd_send_lat -s1048576 -D10 -C
11.137.53.1
```

nd_send_lat Options

The table below lists the various flags of the command.

Table 21 - nd_send_lat Options

Flag	Description
-h	Shows the Help screen.
-v	Shows the version number.
-p	Connects to the port <port> <default 6830>.
-s <msg size>	Exchanges the message size with <default 65536B>, and it must not be combined with -a flag.
-a	Runs all the messages' sizes from 1B to 8MB, and it must not be combined with -s flag.
-n <num of iterations>	The number of exchanges (at least 2, the default is 100000)
-I <max inline size>	The maximum size of message to send inline. The default number is 128B.
-D <test duration in seconds>	Tests duration in seconds.
-f <margin time in seconds>	The margin time to avoid calculation, and it must be less than half of the duration time.
-S <server interface IP>	<server side only, must be last parameter>
-C <server interface IP>	<client side only, must be last parameter>
-h	Shows the Help screen.

A.7 NTttcp

NTttcp is a Windows base testing application that sends and receives TCP data between two or more endpoints. It is a Winsock-based port of the ttcp tool that measures networking performance bytes/second.

To download the latest version of NTttcp (5.28), please refer to Microsoft website following the link below:

http://gallery.technet.microsoft.com/NTttcp-Version-528-Now-f8b12769

This tool should be run from cmd only.

NTttcp Synopsis

```
Server: ntttcp_x64.exe -r -t 15 -m 16,*,<interface IP>
Client: ntttcp_x64.exe -s -t 15 -m 16,*,<same address as above>
```

NTttcp Options

The table below lists the various flags of the command.

Table 22 - NTttcp Options

Flags	Description
-s	Works as a sender
-r	Works as a receiver
-l	<Length of buffer> [default TCP: 64K, UDP: 128]
-n	<Number of buffers> [default: 20K]
-p	<port base> [default: 5001]
-sp	Synchronizes data ports, if used -p should be same on every instance
-a	<outstanding I/O> [default: 2]
-x	<PacketArray size> [default: 1]
-rb	<Receive buffer size> [default: 64K]
-sb	<Send buffer size>[default: 8K]
-u	UDP send/recv
-w	WSARecv/WSASend
-d	Verifies Flag
-t	<Runtime> in seconds.
-cd	<Cool-down> in seconds
-wu	<Warm-up> in seconds
-nic	<NIC IP> Use NIC with for sending data (sender only).
-m	<mapping> [mapping]

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 “IPProtocolMatchCondition”.

> 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
```