Preface

The DBL® version 5.5.0 software User Guide (DBLv5.5) document describes the run-time DBLv5.5.0 software package for the ARC Series E family of network adapter products.

Intended Audience:

The document is intended for system and networking architects looking for a tailored solution with a focus on low latency combined with increased throughput. In this context, the DBLv5.5.0 package provides a deployable solution by using a combination of advanced software stacks and 10-Gigabit network adapters. The document assumes that readers are familiar with C programming language, GNU development tools, and general computer maintenance.

A Note on Handling Network Adapters

It is important to follow ESD anti-static procedures when handling network adapters to avoid accidently damaging integrated circuits. For more information on ESD anti-static procedures, go to: https://www.esda.org/about-esd/esd-fundamentals/part-3-basic-esd-control-procedures-and-materials/

Technical Support

DBLv5.5.0 software documentation, technical support, and downloads are available from the ARIA Cybersecurity Solutions website, as follows:

ARIA website at
https://www.ariacybersecurity.com/network-adapters/

Contact Technical Support via the ARIA Customer Portal *
https://www.ariacybersecurity.com/support

ARIA email support
ARIA_support@ariacybersecurity.com

* Follow the instructions on the ARIA Customer Portal website to sign up for an ARIA Customer Support account.
# Table of Contents

**Preface** .................................................................................................................................................. 3

1. **Introduction** ........................................................................................................................................ 9
   1.1. **DBLv5.5.0 at a glance** .................................................................................................................... 9
   1.2. **What’s New in DBLv5.5.0 Software** ............................................................................................... 11
   1.3. **DBLv5.5.0 and the ARC Series E Network Adapter** ...................................................................... 11
   1.4. **DBLv5.5.0 Chapter Summaries** ...................................................................................................... 12

2. **Installing the ARC Series E Adapter Hardware** .............................................................................. 13
   2.1. **Checking Expansion Card Slot Configuration** ............................................................................. 13
   2.2. **Installing the ARC Series E Network Adapter** ............................................................................ 14

3. **Installing DBLv5.5.0 Software in Linux** .......................................................................................... 15
   3.1. **Downloading DBLv5.5.0 RPM or TGZ drivers** ............................................................................. 16
   3.2. **Installing the DBLv5.5.0 RPM software package for Linux** ......................................................... 16
       3.2.1. Uninstalling the DBLv5.5.0 Binary RPM driver ......................................................................... 19
   3.3. **Installing the DBLv5.5.0 TGZ Driver for Linux** .......................................................................... 20
       3.3.1. Uninstalling the DBLv5.5.0 Tarball TGZ driver ......................................................................... 23
       3.3.2. Listing module load-time variables .......................................................................................... 23
   3.4. **Rebuilding after OS/Kernel Update** ............................................................................................. 24
       3.4.1. Updating the kernel .................................................................................................................... 24
       3.4.2. Rebuilding after kernel update .................................................................................................. 24

4. **Installing DBLv5.5.0 with Windows Installer MSI** ............................................................................ 25
   4.1. **Installation Requirements** ............................................................................................................ 26
   4.2. **Downloading DBLv5.5.0 Windows Installer MSI** ......................................................................... 27
   4.3. **Installing DBLv5.5.0 with Windows Installer MSI** ....................................................................... 27
   4.4. **Installing DBLv5.5.0 with Windows msiexec** ............................................................................. 33
   4.5. **Installing DBLv5.5.0 with Windows Imaging** .............................................................................. 33
   4.6. **Uninstalling DBLv5.5.0 with Windows Installer MSI** .................................................................. 34

5. **PHX-TOOLS Network Adapter Toolkit** ............................................................................................. 36
   5.1. **PHX-TOOLS Description** ............................................................................................................. 36
   5.2. **General Information on the ARC Series E Network Adapter** ...................................................... 37
   5.3. **Upgrading FPGA Firmware in Linux** ............................................................................................ 38
   5.4. **Upgrading FPGA Firmware in Windows** ..................................................................................... 39
   5.5. **Tracking Environmental Diagnostics on ARC Series E Network Adapters** .......................... 40
   5.6. **Tracking SFP+ Diagnostics on ARC Series E Network Adapters** ............................................. 41
6. **DBLv5.5.0 Windows Teaming and VLAN** ................................................................. 42
   6.1. **Preamble:** ........................................................................................................... 42
   6.2. **Configuring the DBLv5 VLAN/Team Driver** ..................................................... 43
   6.3. **VLAN/Team Driver Properties** ........................................................................... 47
      6.3.1. Team driver MAC address ............................................................................. 47
      6.3.2. Teaming for fault tolerance ......................................................................... 47
      6.3.3. Duplicate broadcast or multicast packets ..................................................... 47
      6.3.4. Reverse Address Resolution Protocol (RARP) ........................................... 47
      6.3.5. IP Addresses ................................................................................................. 48
   6.4. **DBLv5.5.0 and Teaming for Kernel Bypass Mode** ............................................. 48

7. **Testing DBLv5.5.0 Software** ...................................................................................... 49
   7.1. **Summary of Test Program Commands** .............................................................. 49
   7.2. **DBLv5.5.0 Test Program Requirements** ............................................................ 50
      7.2.1. Setting environment variables ..................................................................... 50
      7.2.2. 0/1 Ports ......................................................................................................... 52
      7.2.3. DBL_IP_VERBOSE=1 ................................................................................. 52
      7.2.4. DBL_CONFIG .............................................................................................. 52
   7.3. **Running DBLv5.5.0 Test Programs** .................................................................. 53
   7.4. **Sample Test Programs** ....................................................................................... 53
      7.4.1. dbl_simple_send ............................................................................................ 54
      7.4.2. dbl_simple_recv ............................................................................................ 55
      7.4.3. dbl_batch_recv ............................................................................................. 56
      7.4.4. dbl_pingpong ............................................................................................... 57
      7.4.5. dbl_perf_test ............................................................................................... 58
      7.4.6. dbl_speedometer ........................................................................................... 59
      7.4.7. dbl_ring_access ............................................................................................. 60
      7.4.8. dbl_eventq ..................................................................................................... 62
      7.4.9. dbltcp_pingpong ............................................................................................ 63
      7.4.10. dbltcp_tick_to_trade .................................................................................... 64
      7.4.11. sock_tick_to_trade ...................................................................................... 66
      7.4.12. tcp_pingpong ............................................................................................... 68

8. **Running DBLv5.5.0 Diagnostic Tool Programs** .......................................................... 70
   8.1. **sbin/phx_bug_report (Linux only)** .................................................................... 71
   8.1.2. **phx_bug_report.ps1 (Windows only)** ........................................................... 71
   8.1.3. **sbin/myri_info** ............................................................................................ 71
   8.1.4. **sbin\team_config_x64.exe (Windows only)** ............................................. 72
   8.1.5. **bin/dbltool** ................................................................................................. 72
   8.1.6. **bin/myri_counters** ...................................................................................... 72
   8.1.7. **bin/myri_endpoint_info** ............................................................................ 73
   8.1.8. **bin/myri_nic_info** ...................................................................................... 74

9. **Transparent Sockets Acceleration (TA) Software** ....................................................... 75
   9.1. **Installing Transparent Acceleration (TA) Modes** ............................................ 75
      9.1.1. Manually Installing and Configuring DBLv5.5.0 TA Mode via LSP .................. 75
9.1.2. Checking the Status of DBLv5.5.0 TA via DLL .............................................. 77
9.1.3. Uninstalling DBLv5.5.0 TA via DLL .......................................................... 77
9.1.4. Enabling Transparent Acceleration (TA) ....................................................... 78

9.2. **DBLv5.5.0 -Specific Socket Extensions** .................................................................. 78
9.2.1. MSG_WARM ...................................................................................................... 78
9.2.2. SIO_GETNICTIME .......................................................................................... 79
9.2.3. SO_TIMESTAMP .............................................................................................. 79
9.2.4. SO_TIMESTAMPING, MSG_ERRQUEUE .......................................................... 79

9.3. **Implementing DBLv5.5.0 TA in Linux** .............................................................. 80

9.4. **Implementing DBLv5.5.0 TA in Windows** ...................................................... 81
9.4.1. TA Acceleration through Layered Service Providers (LSPs). ......................... 81
9.4.2. TA Acceleration by DLL preload ........................................................................ 82

9.5. **TCP and UDP Acceleration with DBLv5.5.0** .................................................. 83
9.5.1. Known Issues .................................................................................................. 84
9.5.2. Special considerations ..................................................................................... 85

9.6. **Running DBLv5.5.0 TA Test Programs** ......................................................... 86
9.6.1. Test programs: ............................................................................................... 86
9.6.2. Special considerations ..................................................................................... 86
9.6.3. Standard IP ping-pong latency test for UDP .................................................. 87
9.6.4. Accelerated ping-pong latency test for UDP .................................................. 87
9.6.5. Standard IP ping-pong latency test for TCP ................................................... 88
9.6.6. Accelerated ping-pong latency test for TCP ................................................... 89

10. **Linux PTP Host Clock Synchronization** ............................................................. 90
10.1. **Synchronizing the System Clock to the ARC Series E Adapter Clock** ............ 90
10.2. **Installing the Linuxptp Package** ...................................................................... 91
10.2.1. Reading the clock by PTP device name: ....................................................... 91
10.2.2. Reading the clock by network name: ............................................................ 92
10.2.3. Setting the clock manually: .......................................................................... 92
10.2.4. Synchronizing the system clock with the network adapter clock (phc2sys): .... 92
10.2.5. Maintaining clock synchronization (ptp4l): .................................................... 92
10.3. **Synchronizing the ARC Series E Clock** ......................................................... 93

11. **DBLv5.5.0 Timestamping Support** .................................................................... 94
11.1. **Timestamping Module Variables** ................................................................... 94
11.1.1. Default setting .............................................................................................. 94
11.1.2. Enabling PPS support ................................................................................... 95
11.1.3. Enabling 10MHz support: .......................................................................... 95
11.2. **Viewing the Time Source Status** .................................................................. 95
11.3. **Arista Timestamping Environment Variables** .................................................. 97
11.3.1. MYRI_ARISTA_ENABLE_TIMESTAMPING = 0, 1 ........................................... 97
11.3.2. MYRI_ARISTA_ENABLE_SKEW = 0, 1 ........................................................... 98
11.3.3. MYRI_ARISTA_PARAM_KF_DST_IP = <IP ADDRESS> .................................... 99
11.3.4. MYRI_ARISTA_PARAM_KF_CHECK_PTP = 0,1 ............................................. 99
11.3.5. MYRI_ARISTA_PARAM_KF_CHECK_RATE = 0,1 ............................................ 99
11.3.6. MYRI_ARISTA_PARAM_KF_KERNDUP = 0, 1 .................................................. 99
12. Tuning .......................................................................................................................... 100

12.1. Tuning Check List ........................................................................................................ 101

12.1.1. PCIe expansion slot seating .................................................................................... 101
12.1.2. Tune for lowest latency ........................................................................................... 101
12.1.3. CPU frequency scaling ............................................................................................ 101
12.1.4. Interrupt balancing .................................................................................................. 102
12.1.5. CPU binding (Linux) ............................................................................................... 102
12.1.6. PCI bridging ............................................................................................................ 102
12.1.7. Hyperthreading ......................................................................................................... 102
12.1.8. Reducing jitter .......................................................................................................... 103

12.2. Customizing Transparent Acceleration (TA) and TCP API Modes using dbl.conf ...................................................................................................................... 103

12.2.1. Restricting TCP and UDP port acceleration ......................................................... 103
12.2.2. Maximizing the number of accelerated UDP sockets ............................................. 104
12.2.3. Enabling delayed ACKs ........................................................................................ 104
12.2.4. Disabling checksum on receive .............................................................................. 104
12.2.5. Assigning CPU affinity masks to service threads* .................................................. 105
12.2.6. Enable timestamping .............................................................................................. 105
12.2.7. Steering SBL mode for DBL TCP API applications ................................................ 106
12.2.8. Enabling a permanent blocking mode ..................................................................... 106
12.2.9. Enabling spin-friendly polling mechanisms .......................................................... 107
12.2.10. User-level TCP congestion management control algorithms ............................. 107
12.2.11. dbl.conf settings .................................................................................................... 108
12.2.12. Special Considerations: ......................................................................................... 108

12.3. Adaptive Interrupt Moderation (AIM) ......................................................................... 108

12.4. Increasing the Size of the Data and Descriptor Rings ................................................ 109

12.4.1. When power cycling the host is not possible .......................................................... 110

12.5. Single-Threaded TCP Applications ............................................................................ 110

12.6. Zero Copy TCP Sends ................................................................................................ 110

12.7. PIO/DMA Threshold ................................................................................................... 110

13. Troubleshooting .............................................................................................................. 112

13.1. Hardware Installation and Performance Issues ......................................................... 112

13.1.1. Operating systems using the lspci command ......................................................... 114

13.2. Software Installation and System Configuration Issues ........................................... 114

13.2.1. Bug report scripts .................................................................................................... 114
13.2.2. Linux RPM-TGZ installation failures .................................................................... 115
13.2.3. Windows MSI installation failure ........................................................................... 115
13.2.4. Windows teaming/VLAN team_config issues ...................................................... 117
13.2.5. Bonding for failover in TA mode ............................................................................ 124
13.2.6. Software Counters .................................................................................................. 125
13.2.7. Environment Variables .......................................................................................... 126
13.2.8. Functional failures ................................................................................................. 127
13.2.9. Performance Issues ............................................................................................... 127
13.2.10. Network Adapter Timestamps ............................................................................. 128
13.2.11. Synchronization .................................................................................................... 129

Appendix 1. DBLv5.5.0 Counters ......................................................................................... i
Appendix 2. Operating Systems and Hardware Support.................................................................xiv
Appendix 3. DBLv5.5.0 Driver Restrictions and Limitations........................................................xvi
Appendix 4. DBL RAW Mode (Beta) ..............................................................................................xviii
Appendix 5. Multicast Whitelist.........................................................................................................xxi
Appendix 6: DBLv5.5.0 Supported 10G Transceivers ....................................................................xxvi
1. Introduction

This chapter contains the following topics:

- DBL® v5.5.0 at a glance
- What’s New in DBLv5.5.0 Software
- DBLv5.5.0 and the ARC Series E Network Adapter
- DBLv5.5.0 Chapter Summaries

1.1. DBLv5.5.0 at a glance

DBL version 5.5.0 software (DBLv5.5.0) is an optional, user-level software interface for accelerating applications whose performance benefits from improved network latency, reduced jitter, and consumed data at higher message rates.

Applications benefitting from DBLv5.5.0 include:

- Voice over internet protocol (VoIP)
- Online gaming engines
- Applications where reduced network latency is essential

DBLv5.5.0 software and the ARC Series E network adapters leverage user-level, kernel-bypass, messaging techniques – originally developed for high performance computing (HPC) applications – and apply these techniques to IP communication over Ethernet in a multicast and/or unicast environment. DBLv5.5.0 is inter-operable and wire-compatible with all standard TCP and UDP implementations.

IP communication offers a full set of IP networking services to hundreds of user application threads, but at the cost of latency in the protocol stack. DBLv5.5.0 on the other hand, takes advantage of kernel-bypass capabilities in ARC Series E network adapters to allow high-priority user threads to send and receive IP frames directly (Figure 1).
DBL is not intended to completely replace IP communication through the host operating system. Nor is it intended to improve performance where overall system performance depends on multiplexing many user application threads over fewer available computing resources (or CPU cores).

Rather, DBL improves throughput, response time, and transaction rates of user application threads with dedicated computing resources. Accordingly, DBL best suits cases where target latencies, response times, and jitter would otherwise be adversely affected by the operating system protocol stack and scheduling policies.

High Frequency Trading (HFT) environments typically have IEEE 802.3 flow control disabled. Similarly, flow control on the ARC Series E adapters is disabled and cannot be enabled. Ingress packets are counted and dropped if there is back pressure from the application. Likewise, ingress XOFF and XON pause frames are counted and ignored.

Software developers interested in using the advanced features of ARIA Cybersecurity Solutions - Myricom products through the DBL interfaces, should refer to the DBLv5.5.0 API Reference Manual.
1.2. What’s New in DBLv5.5.0 Software

- DBLv5.5.0 software supports the new generation of Myricom 10-Gigabit Ethernet ARC Series E Class network adapters (10G-PCIE3-8E-2S).

- DBLv5.5.0 on a 10-Gigabit Ethernet platform provides:
  - Very accurate timestamps.
  - Higher receive packet rates than previous generation hardware platforms.
  - Higher send rates than previous generation hardware platforms.
  - Lower latency than previous generation hardware platforms.

- The DBLv5.5.0 software introduces the following performance enhancements:
  - Reduced ping-pong latency for UDP and TCP.
  - Improved tick-to-trade performance and reduced TCP send cost.
  - Offers applicable program database files (PDB) for meaningful stack traces.

- Support for firmware version 2.0.6 is added to this release for ARC Series E-Class network adapter users.

1.3. DBLv5.5.0 and the ARC Series E Network Adapter

DBLv5.5.0 software introduces DBL support for the ARC Series E network adapters, the latest generation of ARIA Cybersecurity Solutions - Myricom 10-Gigabit Ethernet network adapters. DBLv5.5.0 software is only available for the following:


**NOTE:** The ARC Series E adapters are dual-port Gen3 x8 PCI-Express adapters.
### 1.4. DBLv5.5.0 Chapter Summaries

The DBLv5.5.0 User Guide contains the following chapters:

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 1</td>
<td>Introduction</td>
</tr>
<tr>
<td></td>
<td>Introduction to DBLv5.5.0 software.</td>
</tr>
<tr>
<td>Chapter 2</td>
<td>Installing the ARC Series E Adapter Hardware</td>
</tr>
<tr>
<td></td>
<td>Provides detailed hardware installation instructions.</td>
</tr>
<tr>
<td>Chapter 3</td>
<td>Installing DBLv5.5.0 Software in Linux</td>
</tr>
<tr>
<td></td>
<td>Provides detailed software installation instructions in Linux.</td>
</tr>
<tr>
<td>Chapter 4</td>
<td>Installing DBLv5.5.0 Software for Windows Installer MSI</td>
</tr>
<tr>
<td></td>
<td>Provides detailed software installation for Windows Installer.</td>
</tr>
<tr>
<td>Chapter 5</td>
<td>PHX-TOOLS Network Adapter Toolkit</td>
</tr>
<tr>
<td></td>
<td>Provides FPGA upgrading and diagnostic information for ARC Series E network adapters.</td>
</tr>
<tr>
<td>Chapter 6</td>
<td>Managing Windows Teaming and VLAN</td>
</tr>
<tr>
<td></td>
<td>Describes how to configure and run the Windows Team Adapter GUI for DBLv5.5.0 Windows NIC teaming with VLANs.</td>
</tr>
<tr>
<td>Chapter 7</td>
<td>Testing DBLv5.5.0 Software</td>
</tr>
<tr>
<td></td>
<td>Provides DBL software testing programs and guidelines.</td>
</tr>
<tr>
<td>Chapter 8</td>
<td>Running DBLv5.5.0 Diagnostic Tool Programs</td>
</tr>
<tr>
<td></td>
<td>Describes diagnostic tools and processes for error reporting.</td>
</tr>
<tr>
<td>Chapter 9</td>
<td>Transparent Sockets Acceleration Software</td>
</tr>
<tr>
<td></td>
<td>Provides detailed instructions for running applications in DBL TA mode.</td>
</tr>
<tr>
<td>Chapter 10</td>
<td>Linux PTP Host Clock Synchronization</td>
</tr>
<tr>
<td></td>
<td>Describes linuxptp software installation and ARC Series E network adapter clock synchronization.</td>
</tr>
<tr>
<td>Chapter 11</td>
<td>Timestamping Support</td>
</tr>
<tr>
<td></td>
<td>Describes the options available for TX and RX timestamping.</td>
</tr>
<tr>
<td>Chapter 12</td>
<td>Tuning</td>
</tr>
<tr>
<td></td>
<td>Describes factors contributing to poor packet transfer and their remedies.</td>
</tr>
<tr>
<td>Chapter 13</td>
<td>Troubleshooting</td>
</tr>
<tr>
<td></td>
<td>Describes common DBLv5.5.0 installation, usage, and performance issues.</td>
</tr>
<tr>
<td>Appendix 1</td>
<td>DBLv5.5.0 Counters</td>
</tr>
<tr>
<td>Appendix 2</td>
<td>Operating Systems and Hardware Support</td>
</tr>
<tr>
<td>Appendix 3</td>
<td>DBLv5.5.0 Driver Restrictions and Limitations</td>
</tr>
<tr>
<td>Appendix 4</td>
<td>DBL RAW Mode (Beta)</td>
</tr>
<tr>
<td>Appendix 5</td>
<td>Multicast Whitelist</td>
</tr>
</tbody>
</table>
2. Installing the ARC Series E Adapter Hardware

This chapter describes the following tasks for installing the ARC Series E network adapter hardware. (10G-PCIE3-8E-2S):

- Checking Server Expansion Card Slot Configuration
- Installing the ARC Series E Network Adapter

It is beyond the scope of this manual to address all foreseeable installation issues specific to your networking site. Should you require additional information regarding installing and configuring your Myricom hardware, contact:

- ARIA Technical Support through the ARIA Customer Portal to https://www.ariacybersecurity.com/support or
- Email ARIA Technical Support at ARIA_support@ariacybersecurity.com

2.1. Checking Expansion Card Slot Configuration

The ARC Series E network adapter has been qualified with PCIe server expansion slots with a minimum of x8 lanes. It is recommended that a PCIe Gen3 x8 expansion slot be located closest to the CPU to achieve best performance. See Table 1. to determine which expansion card slots best suit your network adapter hardware requirements.

<table>
<thead>
<tr>
<th>Expansion slot</th>
<th>Support</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>x16 PCIe slot</td>
<td>Supports x8 card</td>
<td>Check motherboard specifications.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Check mechanical fit to guarantee a secure electrical connection.</td>
</tr>
<tr>
<td>x8 PCIe slot</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCIe Gen1 slot</td>
<td>Does not support an 8-lane network adapter. Performance will degrade.</td>
<td></td>
</tr>
<tr>
<td>PCIe Gen2 slot</td>
<td>Does not support an 8-lane network adapter. Performance will degrade.</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Expansion card slot characteristics
2.2. Installing the ARC Series E Network Adapter

Install the ARC Series E network adapter as follows:

1. Close all active applications and shut down the operating system.
2. Turn off the computer and disconnect the power cord.
3. Open the computer case and locate the PCIe expansion card slots on the motherboard. Do not disturb the legacy PCI card slots, which are different in size and electrical specifications.

The ARC Series E adapters are PCIe Gen3 x8 network adapters. For optimal performance, install the adapter in a PCIe Gen3 x8 slot on the server. Minimal testing on PCIe Gen2 has been performed, but is not recommended. Servers with PCIe Gen1 slots are not supported.

4. Carefully remove the ARC Series E network adapter from its clamshell package without touching the gold PCIe connectors (Figure 2).

![ARC Series E network adapter](image)

**Figure 2.** ARC Series E network adapter (10G-PCIE3-8E-2S)

5. Line up the gold PCIe connectors and indexing tab with the empty PCIe 3.0 slot, ensuring that the ports and mounting bracket are facing the back panel of the computer.

   **NOTE** Place the ARC Series E adapter as close as possible to the CPU to achieve best performance.

6. Seat the network adapter firmly into the PCIe expansion slot until the card “clicks” into place.
7. Secure the network adapter to the computer chassis with a screw.
8. Close and secure the computer case, and re-connect the power plug.
9. Insert cabled transceivers in the network adapter ports. Do not kink the cables. The ARC Series E network adapter is now installed.

For more information on DBLv5.5.0 supported 10G transceivers, refer to Appendix 6: DBLv5.5.0 Supported 10G Transceivers

3. Installing DBLv5.5.0 Software in Linux

This chapter describes the following topics:

- Downloading DBLv5.5.0 RPM or TGZ drivers
- Installing the DBLv5.5.0 RPM software package for Linux
- Installing the DBLv5.5.0 TGZ driver for Linux
- Rebuilding after OS/kernel update

NOTE

The IP addresses described in this chapter may differ from your environment. Contact your system administrator if you need assistance.

The downloaded file format may be either .rpm or .tgz for Linux. A different installation process is required for each format and is explained in subsequent sections.

The DBLv5.5.0 software package contains the following:

- A 10-Gigabit Ethernet network adapter driver
- A dynamic library of software modules
- Test programs to demonstrate DBLv5.5.0 functionality
3.1. Downloading DBLv5.5.0 RPM or TGZ drivers

To download a copy of the DBLv5.5.0 RPM or DBLv5.5.0 TGZ driver to your operating system, either download the file from [https://www.ariacybersecurity.com/cybersecurity-products/support/downloads/](https://www.ariacybersecurity.com/cybersecurity-products/support/downloads/) or contact ARIA support (ARIA_support@ariacybersecurity.com). Then save the file to your designated system directory.

3.2. Installing the DBLv5.5.0 RPM software package for Linux

| NOTE | The RPM package can run in Fedora-based distributions including RHEL and CentOS. |

To install the DBLv5.5.0 RPM software package to your Linux operating system, follow these steps:

1. Enter the following command to uninstall any previous versions of ARIA Cybersecurity Solutions - Myricom software:
   
   `$ sudo yum remove myri_dbl`
   
   All previous versions of ARIA Cybersecurity Solutions - Myricom software are deleted.

2. Verify that the operating system detects the presence of the network adapter.
   
   `$ sudo lspci -d 1c09:
   
   The output displays a list of network adapter drivers.

   ```
   01:00.0 Ethernet controller: CSP, Inc. Device 4258 (rev 01)
   01:00.1 Ethernet controller: CSP, Inc. Device 4258 (rev 01)
   ```

3. Enter the following command to install the DBLv5.5.0 (.rpm) driver:

   `$ sudo yum install myri_dbl*rpm`

   The DBL driver is installed.

   ```
   Building PHOENIX dbl driver for 3.10.0-327.13.1.el7.x86_64 in /opt/dbl/src
   DBL driver in /opt/dbl/sbin
   Created symlink from /etc/systemd/system/default.target.wants/myri_start_stop.service to /etc/systemd/system/myri_start_stop.service.
   ```
4. Enter the following command to confirm that the driver is loaded:

   $ sudo lsmod |grep myri

   The output confirms the driver is loaded.

   myri_dbl 177214 0

   OR

   Enter the following command if the driver does not load:

   $ sudo /opt/dbl/sbin/myri_start_stop restart

   The output confirms the driver is loaded.

5. Enter the following command to confirm that the DBL driver detects the network adapter and the Gen3 x8 or the Gen3 x16 expansion slot:

   $ sudo/opt/dbl/sbin/myri_info

   The output displays the network adapter and the expansion slot.

   pci-dev at 01:00.0 vendor:product(rev)=1c09:4258(01)
   behind bridge root-port: 00:01.0 8086:0c01 (x8.3/x8.3)
   Myri-10G-PCIE-8E -- Link x8
   EEPROM String-spec:
   MAC=00:60:dd:43:52:f0
   SN=490333
   PC=10G-PCIE3-8E-2S
   PN=09-04669
   BOM=A
   
   Firmware:
   Version 2.0.6

6. Enter the following command to track the interface names assigned to each port:

   $ ip link show | grep -iB 1 00:60:dd
Example:

In this example, the interface names are enp1s0f0 and enp1s0f1, and the MAC network adapter addresses begin with 00:60:dd:

```
30: enp1s0f0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT qlen 1000
    link/ether 00:60:dd:43:e8:b0 brd ff:ff:ff:ff:ff:ff
31: enp1s0f1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT qlen 1000
    link/ether 00:60:dd:43:e8:b1 brd ff:ff:ff:ff:ff:ff
```

7. Enter the following commands to manually assign an IP address and a subnet to each network adapter port: (You can also create ifcfg files in the /etc/sysconfig/network-scripts directory).

   `$ sudo systemctl stop NetworkManager.service`
   `$ sudo systemctl disable NetworkManager.service`
   `$ sudo ip address add 10.0.0.1/24 dev enp1s0f0`
   `$ sudo ip link set dev enp1s0f0 up`
   `$ sudo ip address add 10.1.0.1/24 dev enp1s0f1`
   `$ sudo ip link set dev enp1s0f1 up`

8. Enter the following command to confirm that each link is functioning:

   `$ ip address show up`

   The output confirms each link is functioning.

```
27: enp1s0f0:
    <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc pfifo_fast state UNKNOWN qlen 1000
    link/ether 00:60:dd:43:52:f0 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.1/24 brd 10.0.0.255 scope global enp1s0f0
       valid_lft forever preferred_lft forever
    inet6 fe80::260:ddff:fe43:52f0/64 scope link
       valid_lft forever preferred_lft forever
```

9. Enter the following commands to verify contact with the remote host:

   `$ ping 10.0.0.2`
   `$ ping 10.1.0.2`

For more information on test programs, go to Testing

DBLv5.5.0 Software
3.2.1. **Uninstalling the DBLv5.5.0 Binary RPM driver**

This section describes how to uninstall the DBLv5.5.0 RPM driver.

To uninstall the DBLv5.5.0 Binary RPM driver, enter the following commands:

$ sudo /opt/dbl/sbin/myri_start_stop stop
$ sudo yum remove myri_dbl

This procedure removes the DBLv5.5.0 RPM driver from the operating system.
### 3.3. Installing the DBLv5.5.0 TGZ Driver for Linux

Non-Fedora-based Linux distributions with Tarball TGZ (.tgz) drivers are provided, offering support up to Linux kernel version 5.5. The TGZ package can also run on Debian-based distributions, including Ubuntu.

Install the DBLv5.5.0 TGZ driver for Linux as follows:

1. Enter the following command to uninstall any previous versions of Myricom software:
   ```
   $ sudo rm -r /opt/dbl
   ```
   All previous versions of Myricom software are deleted.

2. Verify that the operating system detects the presence of the network adapter.
   ```
   $ sudo lspci -d 1c09:
   ```
   The output displays a list of network adapter drivers.

   ```
   01:00.0 Ethernet controller: CSP, Inc. Device 4258 (rev 01)
   01:00.1 Ethernet controller: CSP, Inc. Device 4258 (rev 01)
   ```

3. Enter the following commands to install the DBLv5.5.0 TGZ driver:
   ```
   $ cd /opt
   $ sudo tar xzvf ./myri_dbl-<version_info>*.x86_64.tgz
   $ mv myri_dbl-<version_info>*.x86_64 dbl
   $ cd /opt/dbl
   $ sudo sbin/rebuild.sh
   ```
   The output confirms that the DBLv5.5.0 TGZ driver is installed.

   ```
   Building PHOENIX dbl driver for 3.10.0-327.4.5.el7.x86_64 in /opt/dbl/src
   DBL driver in /opt/dbl/sbin
   ```

4. For Ubuntu servers running **systemd** (16.04 and later), enter the following commands to start the driver automatically:
   ```
   $ sudo tar xzvf ./myri_dbl-<version_info>*.x86_64.tgz
   $ sudo mv myri_dbl-<version_info>*.x86_64 dbl
   $ sudo cp /opt/dbl/sbin/myri_start_stop.service /etc/systemd/system
   $ sudo systemctl enable myri_start_stop.service
   ```
$ sudo systemctl start myri_start_stop.service

NOTE

When changing the default location of the /opt/dbl directory, be sure to edit the sbin/myri_start_stop service unit file to reflect the new location.

5. Enter the following command to confirm that the driver is loaded:
   $ sudo lsmod | grep myri

   The output confirms the driver is loaded.

   myri_db1 177214 0

   OR

   Enter the following command if the driver does not load:
   $ sudo /opt/dbl/sbin/myri_start_stop restart

   The output confirms the driver is loaded.

6. Enter the following command to confirm that the DBLv5.5.0 driver detects the network adapter and the Gen3 x8 or the Gen3 x16 expansion slot:
   $ sudo /opt/dbl/sbin/myri_info

   The output detects the network adapter and the expansion slot.

   pci-dev at 01:00.0 vendor:product(rev)=1c09:4258(01)
       behind bridge root-port: 00:01.0 8086:0c01 (x8.3/x8.3)
   Myri=10G-PCIE=8E -- Link x8
   EEPROM String-spec:
   MAC=00:60:dd:43:52:f0
   SN=490333
   PC=10G-PCIE3-8E-2S
   PN=09-04669
   BOM=A

   Firmware:
       Version 2.0.6

7. Enter the following command to track the interface names assigned to each port:
   $ ip link show | grep -iB 1 00:60:dd
Example:

In this example, the interface names are `enp1s0f0` and `enp1s0f1`. The MAC network adapter addresses begin with `00:60:dd`:

```
30: enp1s0f0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT qlen 1000
   link/ether 00:60:dd:43:e8:b0 brd ff:ff:ff:ff:ff:ff
31: enp1s0f1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT qlen 1000
   link/ether 00:60:dd:43:e8:b1 brd ff:ff:ff:ff:ff:ff
```

8. Enter the following commands to manually assign an IP address and a subnet to each network adapter port. (You can also create `ifcfg` files in the `/etc/sysconfig/network-scripts` directory).

```
$ sudo systemctl stop NetworkManager.service
$ sudo systemctl disable NetworkManager.service
$ sudo ip address add 10.0.0.1/24 dev enp1s0f0
$ sudo ip link set dev enp1s0f0 up
$ sudo ip address add 10.1.0.1/24 dev enp1s0f1
$ sudo ip link set dev enp1s0f1 up
```

9. Enter the following command to confirm that each link is functioning:

```
$ ip address show up
```

The output confirms each link is functioning.

```
27: enp1s0f0:
   <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc pfifo_fast state UNKNOWN qlen 1000
      link/ether 00:60:dd:43:52:f0 brd ff:ff:ff:ff:ff:ff
      inet 10.0.0.1/24 brd 10.0.0.255 scope global enp1s0f0
         valid_lft forever preferred_lft forever
      inet6 fe80::260:ddff:fe43:52f0/64 scope link
         valid_lft forever preferred_lft forever
```

10. Enter the following commands to verify contact with the remote host:

```
$ ping 10.0.0.2
$ ping 10.1.0.2
```

For more information on test programs, go to Testing DBLv5.5.0 Software
3.3.1. **Uninstalling the DBLv5.5.0 Tarball TGZ driver**

This section describes how to uninstall the DBLv5.5.0 Tarball TGZ driver.

To uninstall the DBLv5.5.0 Tarball TGZ driver, enter the following commands:

```
$ sudo /etc/init.d/myri_start_stop stop
$ sudo rm -rf /opt/dbl/
$ sudo rm -f /etc/init.d/myri_start_stop
```

This procedure removes the DBLv5.5.0 TGZ driver from the operating system.

3.3.2. **Listing module load-time variables**

To list module load-time variables, enter the following command:

```
$ sudo modinfo /opt/dbl/sbin/myri_dbl.ko
```
3.4. Rebuilding after OS/Kernel Update

Updating the kernel on Centos, Fedora, or Ubuntu Linux is a straightforward process. You update the kernel first and then rebuild after the kernel update.

3.4.1. Updating the kernel

There are three ways to update the kernel on Centos, Fedora, or Ubuntu.

Option 1:

Command line:
$ sudo dnf upgrade

Option 2:

Command line:
$ sudo yum update

Option 3:

Update your Linux kernel.

3.4.2. Rebuilding after kernel update

Following the kernel update, you must update, or rebuild, the Myricom drivers.

To rebuild after kernel update, enter the following commands:
$ cd /opt/dbl
$ sudo sbin/rebuild.sh
This chapter describes the following topics:

- Downloading DBLv5.5.0 Windows Installer MSI
- Installing DBLv5.5.0 with Windows Installer MSI
- Installing DBLv5.5.0 with Windows msiexec
- Installing DBLv5.5.0 with Windows Imaging
- Uninstalling DBLv5.5.0 with Windows Installer MSI

**NOTE**
The IP addresses described in this chapter may differ from your environment. Contact your system administrator for more assistance.
### 4.1. Installation Requirements

Before you proceed, review the following Windows Installer MSI (.msi) requirements before installing the DBLv5.5.0 driver (Table 3).

<table>
<thead>
<tr>
<th>Installation Platform</th>
<th>Requirements</th>
</tr>
</thead>
</table>
| Windows platforms              | Requires Windows 7 or 10.  
Windows Server versions recommended for best performance. |
| Windows Servers                | Windows Server 2016 (64-bit) is recommended.  
Windows Server 2012 R2 (64-bit) is supported. |
| MSI (.msi) file                | The .msi file resides on a local network drive.  
Remote Desktop-mounted device is not compatible with (.msi) |
| NET 2.0                        | NET 2.0 or later                                                             |

#### Permissions

- Administrator privileges
- User Account Control  
  Installation requires an account with administrator privileges and permissions.  
  User Account Control (UAC) commands should be run from an elevated prompt.

#### Support Programs

- Teaming and VLAN support  
  Included in DBLv5.5.0 (.msi) package
- Adaptive Interrupt Moderation (AIM)  
  Supports AIM but is disabled by default.  
  For more information, go to *Tuning*.

Table 3. Windows Installer requirements for DBLv5.5.0 drivers.
4.2. Downloading DBLv5.5.0 Windows Installer MSI

To download a copy of the DBLv5.5.0 Windows Installer MSI driver to your operating system, either download the file from https://www.ariacybersecurity.com/cybersecurity-products/support/downloads/ or contact ARIA support (ARIA_support@ariacybersecurity.com). Then save the file to your designated system directory.

4.3. Installing DBLv5.5.0 with Windows Installer MSI

Install the DBLv5.5.0 software driver with Windows Installer MSI (.msi) as follows:

| NOTE: | The screenshots depicted in this procedure do not represent the current DBLv5.5.0 software version. |

1. Open Windows Installer.
   The Windows Installer screen appears.

2. Click Next.
The Setup Wizard screen appears.

3. Click **Next**.
   The License Agreement screen appears.

4. Select the **I Agree** radio button to agree to the license.

5. Click **Next**
   The Transparent Acceleration Configuration screen appears.

6. Click the **Manual TA configuration...(Recommended)** radio button to manually configure TA.
The SysWOW64 support screen appears.

7. Click **Manual SysWOW64 configuration (Recommended)** to block socket functionality (Winsock 1.1 or 2.2).

The Select Installation Folder screen appears. The default folder is `C:\DBL_PHX-10G\` (DBLv5.5.0 driver file folder).

8. Click **Next** to confirm the location to install the software.
The Confirm Installation screen appears.

9. Click **Next** to install the software.

The Software Installation Progress screen appears.

10. Click **Next** once the software has fully installed.
The Windows Security screen appears.


OR

Do **not** check the **Always trust software from “CSP Inc.”** check box. The Windows Security screen appears in subsequent installation procedures.

12. Click **Install** to complete the installation process.

The Installation Complete screen appears.

13. Click **Close** to exit Windows Installer.

14. Restart the computer to complete the installation.
15. Go to the Microsoft Support website at https://support.microsoft.com/ to set the IP address and network settings.

16. Open Device Manager to confirm that the DBLv5.5.0 software has been successfully installed.

   The Device Manager screen opens.

![Device Manager Screen](image)

17. Click Network adapters and select the appropriate network adapter.

18. Should you encounter problems installing DBLv5.5.0 with MSI installer, send an MSI install log file to ARIA Technical Support with the /log <filename>.

   ```
c:> dbl-<version_info>.PHX_x64_wlh.msi /log dbl_install_log.txt
   ```

   If the network adapter reference is missing or is inoperable, refer to the Windows MSI Installation Failure section in Troubleshooting.
4.4. Installing DBLv5.5.0 with Windows msiexec

Install DBLv5.5.0 files using the Windows msiexec command as follows:

1. Enter the following command to install DBLv5.5.0 with Windows msiexec:
   
   ```
   $ msiexec /a dbl--<version_info>.PHX_x64_wlh.msi /qb
   TARGETDIR="C:\DBLv5_PHX-10G"
   
   OR
   
   For default settings, enter the following command:
   
   $ msiexec /i dbl--<version_info>.PHX_x64_wlh.msi
   ```

2. Run the Update Driver network adapter option described in the section below.

4.5. Installing DBLv5.5.0 with Windows Imaging

Installing the DBLv5.5.0 software with Windows imaging, using the msiexec command, retains your network configurations.

1. Enter the following command to extract the DBLv5.5.0 files from msiexec:
   
   ```
   $ msiexec /a dbl--<version_info>.PHX_x64_wlh.msi /qb
   TARGETDIR="C:\DBLv5_PHX-10G"
   ```

2. To update the DBLv5.5.0 driver, open Device Manager, select Update Driver, and select .inf in the extracted [Advanced] directory.
4.6. Uninstalling DBLv5.5.0 with Windows Installer MSI

If the DBLv5.5.0 software was installed using Windows Installer (.msi), running it again will include an option to uninstall the software.

NOTE: The screenshots depicted in the is procedure do not represent the current DBLv5.5.0 software version.

To uninstall DBLv5.5.0, follow these steps:

1. Click **Install/Upgrade** in Windows Control Panel.
   The Install/Upgrade screen appears.

2. Select DBLv5.5.0 software from the Applications list.
   The Repair/Remove screen appears.

3. Select **Remove**.

4. Click **Finish**.
The Installation Complete screen appears indicating that the software has been removed.

5. Click **Close** to exit the Windows Installer program.
5. PHX-TOOLS Network Adapter Toolkit

This chapter includes the following topics:

- PHX-TOOLS Description
- General Information on the ARC Series E Network Adapter
- Upgrading FPGA Firmware in Windows
- Upgrading FPGA Firmware in Linux
- Tracking Environmental Diagnostics on ARC Series E Network Adapters
- Tracking SFP+ Diagnostics on ARC Series E Network Adapters

5.1. PHX-TOOLS Description

The PHX-TOOLS Network Adapter Toolkit allows users to run diagnostics on ARC Series E network adapter operation and flash memory FPGA firmware programming.

The PHX-TOOLS toolkit contains the following command-line tools:

<table>
<thead>
<tr>
<th>Linux</th>
<th>Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>myri_info</td>
<td>myri_info.exe</td>
</tr>
<tr>
<td>phx-replacement-eeprom</td>
<td>phx-replacement-eeprom.exe</td>
</tr>
<tr>
<td>phx-scan</td>
<td>phx-scan.exe</td>
</tr>
<tr>
<td>myri_phx_mdio</td>
<td>myri_phx_mdio.exe</td>
</tr>
</tbody>
</table>

The toolkit currently supports the following network card format:

- 10G-PCIE3-8E-2S with two SFP+ cages

Windows (Powershell)

**Command line:**

```
Powershell> Set-ExecutionPolicy RemoteSigned
```

The toolkit instructs the operating system to run the command.
5.2. General Information on the ARC Series E Network Adapter

The myri_info command displays 10G-PCIE3-8E-2S diagnostic information such as product code, serial number, MAC address, FPGA flash image version, and so on.

**Linux**

**Command line:**

```
$ sudo ./myri_info
```

**Output:**

```
pci-dev at 01:00.0 vendor:product(rev)=1c09:4258(01)
    behind bridge root-port: 00:01.0 8086:0c01 (x8.3/x16.3)
Myri-10G-PCIE-8E -- Link x8
    EEPROM String-spec:
        MAC=00:60:dd:43:2d:fe
        SN=495855
        PC=10G-PCIE3-8E-2S
        PN=09-04669
        BOM=A

    Firmware:
        Version: 2.0.6
        Type : DBL
        Config : 2 Port x 10 Gb
        SHA1  : c2bb1930b6d4a3b0945719d05b5024b4

    External Inputs:
        PPS: Disabled
        10Mhz Clock: Disabled
        100Mhz Clock Locked: Locked
```

**Windows (Powershell)**

**Command line:**

```
PS C:\Program Files\CSPi\phx-tools> .\myri_info.exe
```
5.3. Upgrading FPGA Firmware in Linux

Before you upgrade the FPGA firmware in Linux, verify that the DBL driver is *not* loaded, otherwise the FPGA firmware will not program.

1. To prevent the DBL software driver from loading, enter the following command:

   ```
   $ sudo /etc/init.d/myri_start_stop stop
   ```

   **NOTE**
   
   Every network adapter format has a different firmware image, which is not interchangeable. For example, the firmware image for the 10G-PCIE3-8E-2S network adapter is `fw-8E-2S-2.0.6.bin`

2. Enter the following command to program the FPGA firmware into flash memory:

   ```
   $ sudo ./bin/phx-replace-eeprom ./fw-8E-2S-2.0.6.bin
   ```

   A progress bar appears displaying the programming and verification pass status.

   **Output:**

   ```
   Preparing to reprogram firmware on unit 0 (129:0.0)
   Programming EEPROM with fpga image contained in ./fw-8E-2S-2.0.6.bin len=7298586
   Please do not turn off power while flash is being programmed.
   Do you want to continue (enter yes)? yes
   Loading.... ##################################################| 100%
   Verifying.. ###################################################################| 100%
   Flash verification succeeded!
   Power cycle the system to enable the new firmware.
   ```

3. Power-cycle the machine to load the new firmware. A reboot/restart is *not* sufficient.

4. In the event there are repeated programming or verification errors:
   - Run the command again.
   - OR
   - Power-cycle the machine then run the command again.

5. Enter the following command to confirm that the new firmware is running:

   ```
   $ sudo ./bin/myri_info
   ```

   **Output:**

   ```
   Firmware version 1.5.0 sha1: 6790572f79f5321888964a56283400ec
   ```
5.4. Upgrading FPGA Firmware in Windows

To upgrade FPGA firmware in Window Powershell, follow these steps.

Windows (Powershell)

1. Click Control Panel>Network and Internet>Network Connections
   The Network Connections window appears.

2. Right-click 10Gbps Ethernet NIC with DBL and select disable to disable the network device.

3. Repeat the previous step for every adapter displayed in the Network Connections window.

   **NOTE** Every network adapter format has a different firmware image and are not interchangeable. For example, the 10G-PCIE3-8E-2S adapter firmware image is fw-8E-2S-2.0.6.bin

4. Enter the following command to program the FPGA firmware into flash memory:
   \[PS C:\Program Files\CSPi\phx-tools> .\phx-replace-eeprom.exe .\fw-8E-2S-2.0.6.bin\]
   
   **OR**
   
   Set \phx-replace-eeprom.exe to the -l flag to load FPGA directly from flash memory after programming:
   \[PS C:\Program Files\CSPi\phx-tools> .\phx-replace-eeprom.exe -l .\fw-8E-2S-2.0.6.bin\]

   **CAUTION** The -l flag method may cause spontaneous reboots on some earlier PCIe Gen3 systems. For that reason, the command is not set as default. Close all applications and ensure other users have logged off.

5. Enter Yes to confirm flash memory reprogramming.
   A progress bar appears displaying the programming and verification pass status.

6. Power-cycle the machine to load the new firmware. A reboot/restart is not sufficient unless you have used the -l flag.

7. In the event there are repeated programming or verification errors:
   - Run the command again.
   **OR**
   - Power-cycle the machine and run the command again.
8. After power-cycling, enter the following command to confirm that the new firmware is running:

```bash
PS C:\Program Files\CSPi\phx-tools> .\myri_info.exe
```

Firmware version 1.5.0 sha1: 6790572f79f5321888964a56283400ec

5.5. Tracking Environmental Diagnostics on ARC Series E Network Adapters:

Enter the following command line to track and record environmental diagnostics on ARC Series E network adapters:

**Command line:**

```bash
$ sudo ./phx-scan
```

**Output: FPGA Revisions**

**K35 FPGA Revisions:**

```
Rev : 0x00350001
Branch: 0x00010010
```

**Output: Sensor Records**

**K35 Sensor Records:**

- Sensor 0: FPGA Core Temperature
- Sensor 1: FPGA Local Temperature
- Sensor 2: Over Temperature Alert
- Sensor 3: Fan Speed Percentage
- Sensor 4: Fan RPMs
- Sensor 5: Power Management Good
- Sensor 6: Power Management Alert

**Output: Sensor Readings:**

**Sensor Readings:**

- FPGA Core Temperature: VALUE=39.31 degrees C, STATE=OK
- FPGA Local Temperature: VALUE=39.50 degrees C, STATE=OK
- Over Temperature Alert: VALUE=1, STATE=OK
- Fan Speed Percentage: VALUE=100%, STATE=OK
- Fan RPMs: VALUE=5892, STATE=OK
5.6. Tracking SFP+ Diagnostics on ARC Series E Network Adapters:

The `myri_phx_mdio` command retrieves SFP+ transceiver modules diagnostics from the 10G-PCIE3-8E-2S adapter.

**Linux**

Command line:

```
$ sudo ./bin/myri_phx_mdio
```

**Windows (Powershell)**

Command Line:

```
PS C:\Program Files\CSP\phx-tools> .\myri_phx_mdio.exe
```

**Commonly used flags:**

- `-m` - includes MAC counters.
- `-v` - includes PHY registers

For more information contact ARIA technical support.
6. DBLv5.5.0 Windows Teaming and VLAN

This chapter describes how to configure the DBLv5.5.0 VLAN/Team driver for Windows teaming with Virtual Local Area Networks (VLANs) on both host and virtual machines.

The chapter topics include:

- Preamble
- Configuring the DBLv5.5.0 VLAN/Team Driver
- DBLv5.5.0 VLAN/Team Driver Properties
- DBLv5.5.0 and Teaming

6.1. Preamble:

Teaming adds fault tolerance, load balancing, and link aggregation features to a group of ports.

**Fault Tolerance**

Fault tolerance uses one or more secondary adapters to take over for the primary adapter should the first adapter, or its cabling, or the link partner, fail. Designed to guarantee server availability to the network.

**Link Aggregation**

Link aggregation combines multiple adapters into a single channel to provide greater bandwidth. Bandwidth increase is only available when connecting to multiple destination addresses. Link aggregation modes require switch support.
6.2. Configuring the DBLv5 VLAN/Team Driver

The DBLv5 VLAN/Team driver enables two DBLv5.5.0 network adaptor ports to team up.

NOTE

For illustrative purposes, these instructions describe how to configure teaming on Windows Server 2016.

The VLAN/Team built within the Myricom Ethernet driver and DBL releases will not appear when installed into a Windows Server 2016 system. Windows Server 2016, however, will allow you to configure teaming/VLAN using the operating system network configuration options as described below.

Team the two DBLv5.5.0 network adapters as follows:

1. Verify Windows Server detects the appropriate network adapters. The adapters should be shown in the Device Manager and Network Connections.

![Device Manager screenshot](image)
2. Open the Server Manager Dashboard.
3. Select **Local Server** and enable teaming on the desired adapter.

4. Create a new team, provide a name for the team, select the appropriate member adapters, and configure the additional properties (Teaming mode, Load balancing mode, and Primary team interface).

5. Click **OK** to create the team.
6. Verify the new VLAN/team is displayed in the Teams list.

7. Verify the VLAN/Team adapter is available in Network Connections.
6.3. VLAN/Team Driver Properties

The following VLAN/Team driver properties are described below:

6.3.1. Team driver MAC address

The team driver MAC address corresponds to one or more DBLv5.5.0 network adapter MAC addresses. There are no restrictions to the number of ports you can use for teaming. You can use as many as you like.

6.3.2. Teaming for fault tolerance

In a two-port network adapter, the teaming driver selects one adapter port as the active adapter. The active adapter sends and receives packets. The remaining adapter port is a backup or non-active adapter. The backup adapter receives, but never sends packets.

When the active adapter’s link goes down, the backup adapter becomes the active adapter. It remains the active adapter until its link goes down.

You can view adapter activity in the teaming driver Properties Dialog box.

6.3.3. Duplicate broadcast or multicast packets

The team adapter can receive duplicate broadcast or multicast packets when both adapters plug into one switch. The switch floods broadcast packets and some multicast packets.

Both active and backup adapters receive packets. The team adapter passes all broadcast and multicast packets to the network stack, followed by a duplicate copy.

NOTE

If the active adapter and the backup adapter cannot send or receive packets, run regedit.exe and select the onActiveAdaptersDropNonUnicast key. Set the registry key to 1. The backup team adapter drops all network broadcast or multicast packets.

6.3.4. Reverse Address Resolution Protocol (RARP)

The active adapter sends a RARP packet every 60 seconds by default. The backup adapter, as active adapter, immediately sends an RARP packet. The RARP packet source and team adapter have the same MAC address.
6.3.5. **IP Addresses**

You can assign an IP address when the team adapter binds to the TCP/IP driver. When the DBLv5.5.0 network adapters bind to the team adapter, however, the action unbinds the TCP/IP driver from the network adapters, removing their IP addresses.

**Program example:**

1. Connect both DBLv5.5.0 network adapters to the switch.
2. Run the PING, NTTTCP, or NETPERF tests between the two machines.
3. Disconnect one network adapter during the test run. The programs may experience slight delays but will continue to run.

6.4. **DBLv5.5.0 and Teaming for Kernel Bypass Mode**

The DBLv5.5.0 driver does not recognize teaming and cannot distinguish between active and backup adapters. By using the `dbl_open` command you can:

**Option1:**

Select the team adapter using the user-specified IP address. The `dbl_open` command selects that particular address.

**Option 2:**

Bind the TCP/IP driver to the underlying network adapter and assign it an IP address.

Select the team adapter with the MAC address matching the user-specified IP address. `dbl_open` selects the underlying adapter that owns the IP address.
7. Testing DBLv5.5.0 Software

After you have successfully installed DBLv5.5.0 to your operating system, best practices recommend that you subject the DBLv5.5.0 software to a testing regimen. DBLv5.5.0 software contains several software test programs that serve in that capacity to verify DBLv5.5.0 operation on your operating system and to familiarize the user with its basic operation.

This chapter contains the following topics:

- Summary of Test Program Commands
- DBLv5.5.0 Test Program Requirements
- Running DBLv5.5.0 Test Programs
- Sample Test Programs

7.1. Summary of Test Program Commands

- `dbl_simple_send`
- `dbl_simple_recv`
- `dbl_batch_recv`
- `dbl_pingpong`
- `dbl_perf_test`
- `dbl_speedometer`
- `dbl_eventq`
- `dbltcp_pingpong`
- `dbltcp_tick_to_trade`
- `sock_tick_to_trade.c`
- `tcp_pingpong.c`
7.2. DBLv5.5.0 Test Program Requirements

Review the following requirements to take full advantage of the DBLv5.5.0 test program features.

7.2.1. Setting environment variables

Set your environment variables according to the following operating systems or languages.

Windows powershell

To see an environment variable, enter the following:

```
get-childitem env:
get-childitem env:var_name
```

To set an environment variable:

```
Set-Content env:var_name
```

To clear an environment variable:

```
remove-item env:\var_name
```

OR

1. Right-click My computer > Properties > Advanced system settings > Environmental Variables.
2. Click New or Edit

Linux bash:

To see an environment variable.

```
$ env
```

To set an environment variable.

```
$ export VAR_NAME=value
```

To clear an environment variable.

```
$ unset VAR_NAME
```
Linux csh:
To see an environment variable.
   $ env
To set an environment variable.
   $ setenv VAR_NAME value
To clear an environment variable.
   $ unsetenv VAR_NAME

C language
To get an environment variable.
   getenv("VAR_NAME");
To set an environment variable.
   putenv("VAR_NAME=value");

C# language
To get an environment variable.
   System.Environment.GetEnvironmentVariable("VAR_NAME")
To set an environment variable.
   System.Environment.SetEnvironmentVariable("VAR_NAME", "value")
### 7.2.2. 0/1 Ports

Many DBLv5.5.0 example test programs require a `-p <port>` as a command line argument. The port number `<port>` refers to the physical PHY connector port on the network adapter. The ports are numbered, starting at 0.

On a network adapter with two physical ports, the port closest to the PCI connector is assigned port number 0 and the port farthest from the connector is assigned port number 1. Port 0 also corresponds to the lower MAC address of a two-port adapter.

For multiple network adapters installed on the server, the adapter port numbers are assigned in the order in which the adapters are detected by the BIOS. Refer to the `myri_nic_info` output to determine the port numbering sequence.

**NOTE**

The ARC Series E network adapter has two ports, labeled PORT 0 and PORT 1 on the PCI faceplate, where PORT 0 is the port closest to the gold PCI connector.

In software tools nomenclature, PORT 0 corresponds to port 0 and PORT 1 corresponds to port 1.

### 7.2.3. DBL_IP_VERBOSE=1

When running UDP and TCP test programs, specify the DBLv5.5.0 environment variable `DBL_IP_VERBOSE=1` as follows:

```bash
$ DBL_IP_VERBOSE=1 ./test/dbl_pingpong -s -l 10.0.0.1
```

The diagnostic output prints to the console.

### 7.2.4. DBL_CONFIG

When running DBLv5.5.0 TCP API, specify the environment variable `DBL_CONFIG` to steer various MTCP variables. These variables include, but are not limited to, congestion modules and delayed ACK.

Run the test program with the command line option `-?` to obtain a usage summary of the command line options.

For more information on `dbl.conf`, refer to the Customizing Transparent Acceleration in TCP API mode using `dbl.conf` section in Transparent Sockets Acceleration Software.
7.3. Running DBLv5.5.0 Test Programs

Running in Linux

To run the DBLv5.5.0 test programs in Linux, go to the:

- `/opt/dbl/bin/tests` directory (binary form)
- `/opt/dbl/share/doc/examples` directory (source form)

Compiling in Linux

To compile the DBLv5.5.0 test programs in Linux, go to the following Makefile:

- `/opt/dbl/share/doc/examples/Makefile` directory

Makefiles are a simple way to organize your code compilations. Make sure that UDP applications link to the `libdbl.so` library. Likewise, ensure that TCP applications link to the `libdbltcp.so` library.

Running in Windows

To run the DBLv5.5.0 test programs in Windows, go to the:

- `C:\DBL_PHX-10G\bin\tests` directory (binary form)
- `C:\DBL_PHX-10G\share\doc\examples` directory (source form)

7.4. Sample Test Programs

The DBLv5.5.0 software package includes sample test programs, found in the `sbin/` and `bin/` directories.

Summary of DBLv5.5.0 test programs:

- `dbl_simple_send`
- `dbl_simple_recv`
- `dbl_batch_recv`
- `dbl_pingpong`
- `dbl_perf_test`
- `dbl_speedometer`
• dbl_ring_access
• dbl_eventq
• dbltcp_pingpong
• dbltcp_tick_to_trade
• sock_tick_to_trade
• tcp_pingpong

7.4.1. **dbl_simple_send**

An test program using the DBLv5.5.0 API function call **dbl_send()**. Client counterpart to **dbl_simple_recv**.

**Definition:**

*dbl_simple_send*

**Example: (Client side)**

```
$ ./dbl_simple_send -h 10.0.0.1 -P 3333 -l 10.0.0.2 -p 3333 -i 10000 -S 100
```

**Usage:**

```
./dbl_simple_send [options]
```

**Command line [options]:**

```
-?       Print help information and usage.
-p <port> Use [remote] port for sending/receiving (1024–65535). Default: 3333
-P <local port> Use local port for sending (1024–65535). Default: 3333
-l <local_address> Specify address of local product interface. Default: DBL_LOCAL
-h <remote_address> Specify address of remote interface. Default: DBL_REMOTE
-S <Size> Specify Size to send. Default: 100
-i <iterations> Set number of send/receive iterations. Default: DBL_ITERATIONS, or 10 if undefined
-k <increment> Increment message length per each send
-v       Allow for packet verification by putting msg size into payload
-g <gap in seconds> Delay in between sends
```
### 7.4.2. **dbl_simple_recv**

A test program running the DBLv5.5.0 API function `dbl_recv()`. Server counterpart to `dbl_simple_send`.

**Definition:**

`dbl_simple_recv`

**Example: (Server side)**

```bash
$ ./dbl_simple_recv -l 10.0.0.1 -p 3333 -i 10000
```

**Usage:**

```
./dbl_simple_recv [options]
```

**Command line [options]:**

- `-?` Print help information and usage.
- `-p <port>` Use [remote] port for sending/receiving (1024-65535i. Default: 3333)
- `-j <mcast_address>` Specify the multicast address to join. Default: DBL_MCAST, or 224.2.50 if undefined
- `-l <local_address>` Specify address of local product interface. Default: DBL_LOCAL
- `-i <iterations>` Set number of send/receive iterations. Must be set equally on client and server. Default: DBL_ITERATIONS, or 10000 if undefined
- `-V` Print verbose output for debugging.
- `-C <req_bytes>` Modify size of receive request.
- `-M <recv_mode>` Select receive mode.
  - 0 - Non-blocking (busy-poll) [default]
  - 1 - DBL-internal blocking
  - 2 - Poll using DBL file descriptor
  - 3 - Non-blocking via EAGAIN
7.4.3.  `dbl_batch_recv`

Demonstrates the use of the API function `dbl_ext_recvfrom()`. This function allows you to receive multiple packets in one call instead of receiving packets sequentially, one at a time, with `dbl_recvfrom()`. Using `dbl_ext_recvfrom()` can help minimize the overhead when receiving packets. By default, `dbl_batch_recv` expects to receive 100 packets.

**Definition:**

`dbl_batch_recv`

**Example: (Receiver)**

```
$ ./dbl_batch_recv -l 10.0.0.1 -n 10
```

**Example: (Sender)**

```
$ ./dbl_simple_send -l 10.0.0.2 -h 10.0.0.1 -i 500
```

**Example: (Output on Receiver)**

```
opened
bound
  testing DBL_EXT_RECV_DEFAULT
  got 10 packets
  testing DBL_EXT_RECV_NONBLOCK
  got 10 packets
  testing DBL_EXT_RECV_COMPLETE
  got 10 packets
  testing DBL_EXT_RECV_COMPLETE and rc for ENOBUFS
  testing DBL_EXT_RECV_DEFAULT
  got 10 packets
  testing DBL_EXT_RECV_NONBLOCK
  got 10 packets
  testing DBL_EXT_RECV_COMPLETE
  got 10 packets
  testing DBL_EXT_RECV_COMPLETE and rc for ENOBUFS
  testing DBL_EXT_RECV_DEFAULT
  got 10 packets
  testing DBL_EXT_RECV_NONBLOCK
  got 10 packets
  testing DBL_EXT_RECV_COMPLETE
  got 10 packets
```

**Usage:**

```
./dbl_batch_recv [options]
```

---

`DBLv5.5.0 User Guide`
Command line [options]:

-?       Print help information and usage.
-p <port> Use [remote] port for sending/receiving (1024-65535). Default: 3333
-l <local_address> Specify address of local product interface.
-n <num_batch> Max number of receive buffers to return Default: 1

7.4.4. **dbl_pingpong**

Measures UDP ping-pong latency with DBLv5.5.0 API. By connecting the client and server machines back-to-back (switchless), the projected half-round trip latency is about 1.3 microseconds.

**Definition:**

dbl_pingpong

**Example: (Server side)**

```bash
$ ./dbl_pingpong -s -l 10.0.0.1 -p 3333 -i 10000
```

**Example: (Client side)**

```bash
$ ./dbl_pingpong -h 10.0.0.1 -l 10.0.0.2 -p 3333 -i 10000
```

**Usage:**

```bash
./dbl_pingpong [options]
```

Command line [options]:

-?, --help
  Print help information and usage.
-s, --server
  Run as server.
-h, --remote <remote_address>
  Specify remote address.
  Default: DBL_REMOTE
-p, --port <port>
  Use [remote] port for sending/receiving (1024-65535).
  Default: 3333
-l, --local <local_address>
  Specify address of local product interface.
  Default: DBL_LOCAL
-i, --iterations <iterations>
  Set number of send/receive iterations.
  Must be set equally on client and server.
  Default: DBL_ITERATIONS, or 10000 if undefined
-t, --timeout <seconds>
  Set time to run before terminating application.
-S, --size <size>
  Set size of packet in bytes to send/receive (min).
  Default: 1
-E, --endsize <size>
  Set size of packet in bytes to send/receive (max).
  Default: 2048
-b, --blocking
  Use blocking receive.
-D, --excelPrint
  Outputs the data in an excel friendly format.
-d, --dup
  Duplicate packets to kernel.
-I, --increment
  Set number of bytes to increment packet size
  Default: power of 2
-X, --send_cost
  Measure client blocking send() cost instead of HRT.
-Z, --turnaround_time
  Time between receiving a packet and sending out the next
  based on timestamps.

7.4.5.  **dbl_perf_test**

Measures DBLv5.5.0 point-to-point, round-trip time (RTT) latency. The default setting runs
tests over multicast UDP by default. The program also runs a short parallel I/O test.

**Definition:**

dbl_perf_test

**Example: (Server side)**

$ ./dbl_perf_test -s -l 10.0.0.1 -P 3333 -i 10000 -V

**Example: (Client side)**

$ ./dbl_perf_test -h 10.0.0.1 -p 3333 -l 10.0.0.2 -i 10000 -V
Usage

./dbl_perf_test [options]

Command line [options]:

[ -?, --help ] Prints help information and usage.
[ -s, --server ] Run as server.
[ -h, --remote <remote_address> ] Specify remote address. Default: DBL_REMOTE.
[ -l, --local <local_address> ] Specify address of local product interface. Default: DBL_LOCAL.
[ -P, --localhost <port> ] Use local <port> (1024-65535). Defaults to remote port or 3333 if not specified.
[ -i, --iterations <iterations> ] Set number of send/receive iterations. Must be set equally on client and server. Default: DBL_ITERATIONS, or 10000 if undefined.
[ -V, --verbose ] Print verbose output for debugging.
[ -Y, --piotest ] Run a short PIO performance test. Note: If the -i (or -V) command line argument is used, it must be specified on both the server and the client.

7.4.6. \textit{dbl\_speedometer}

Measures multicast UDP latency by way of the DBLv5.5.0 API. The test program is particularly useful when you are connected to a multicast feed with only one ARC Series E network adapter.

\begin{tabular}{|c|l|}
\hline
\textbf{NOTE} & Run the dbl\_pingpong test program when there are two ARC Series E adapters. \\
\hline
\end{tabular}

Definition:

\texttt{dbl\_speedometer}

Example: (Client side)

\$ ./dbl_speedometer -l 10.0.0.1 -j 224.0.31.1 -p 14310 -i 100

Usage:

./dbl_speedometer [options]

Command line [options]:

[ -?, --help ]
---

Print help information and usage.

-\p, --port <port>
  Use [remote] port for sending/receiving (1024-65535).
  Default: 3333

-j, --mcast <mcast_address>
  Specify the multicast address to join.
  Default: DBL_MCAST, or 224.1.1.50 if undefined

-l, --local <local_address>
  Specify address of local product interface.
  Default: DBL_LOCAL

-I, --iterations <iterations>
  Set number of send/receive iterations.
  Must be set equally on client and server.
  Default: DBL_ITERATIONS, or 10000 if undefined

-S, --size <size>
  Set size of packet in bytes to send/receive (min).
  Default: 1

-E, --endsize <size>
  Set size of packet in bytes to send/receive (max).
  Default: 2048

-D, --excelPrint
  Outputs the data in an excel friendly format.
---

### 7.4.7. `dbl_ring_access`

Demonstrates the use of the API functions `dbl_eventq_open()`, `dbl_eventq_close()`, `dbl_eventq_peek_head()`, `dbl_eventq_peek_next()`, `dbl_eventq_inspect()` and `dbl_eventq_consume()`. These functions allow an application to operate on ring data without copying the payload. The operations include retrieving pointers to the first and next packets in the queue and to the header and data sections of a given packet. There is also an operation that consumes the packet using two modes: `DBL_CONSUME_SINGLE`, which consumes a single packet at the head of the queue; and `DBL_CONSUME_ALL`, which consumes all outstanding packets.

The `dbl_ring_access` test detects messages in the queue and then operates on the header and payload. It then checks the packets and consume them.

**Definition:**

`dbl_ring_access`

**Example: (Receiver)**

```
$ ./dbl_ring_access -l 10.0.0.1
```
Example: (Sender)

$ ./dbl_simple_send -l 10.0.0.2 -h 10.0.0.1 -k 1 -v

Example: (Output on Receiver)

opened
currently 0 packets
got nothing
currently 0 packets
got nothing
currently 0 packets
got nothing
currently 0 packets
got nothing
currently 10 packets
packet detected (1st)
  pkt holds hdr at 0x7f424e5a9000, payload at 0x7f424e5a902a, paylen 100, timestamp 1553156995354039934
  nxt packet detected (2nd)
  pkt holds hdr at 0x7f424e5a9090, payload at 0x7f424e5a90ba, paylen 101, timestamp 1553156995354040820
  nxt packet detected (3rd)
  pkt holds hdr at 0x7f424e5a9120, payload at 0x7f424e5a914a, paylen 102, timestamp 1553156995354041374

currently 10 packets
currently 10 packets
currently 9 packets
currently 7 packets
got msglen 103
currently 6 packets
got msglen 104
currently 5 packets
got msglen 105
currently 4 packets
got msglen 106
currently 3 packets
got msglen 107
currently 2 packets
got msglen 108
currently 1 packets
got msglen 109
currently 0 packets

Usage:

./dbl_ring_access [options]
7.4.8. **dbl_eventq**

Demonstrates the use of the API function `dbl_eventq_count`. This function allows you to check how many packets are available in the receive queue.

**Definition:**

```
dbl_eventq
```

**Example: (Receiver)**

```
$ ./dbl_eventq -l 10.0.0.1 -D
```

**Example: (Sender)**

```
$ ./dbl_simple_send -l 10.0.0.2 -h 10.0.0.1 -g 1 -k 1 -v
```

**Example: (Output on Receiver)**

```
Detected 0 packets
Detected 0 packets
Detected 0 packets
Detected 0 packets
Detected 0 packets
Detected 0 packets
Detected 1 packets
Detected 2 packets
Detected 3 packets
Detected 4 packets
Detected 5 packets
Detected 6 packets
Detected 7 packets
Detected 8 packets
Detected 9 packets
Detected 10 packets
```

**Usage:**

```
./dbl_eventq [options]
```
7.4.9. **dbltcp_pingpong**

Measures TCP ping-pong latency with DBLv5.5.0 API. By connecting the client and server machines back-to-back (switchless), the projected half-round trip latency is 2 to 3 microseconds.

**Definition:**

dbltcp_pingpong

**Example: (Server side)**

```bash
$ ./dbltcp_pingpong -s -l 10.0.0.1 -i 1000000
```

**Example: (Client side)**

```bash
$ ./dbltcp_pingpong -h 10.0.0.1 -l 10.0.0.2 -i 1000000 -E 4
```

**Usage:**

```
./dbltcp_pingpong [options]
```

---

**Command line [options]:**

- `-?`, `--help`
  
  Print help information and usage.

- `-s`, `--server`
  
  Run as server.

- `-h`, `--remote <remote_address>`
  
  Specify remote address.
  
  Default: DBL_REMOTE

- `-p`, `--port <port>`
  
  Use [remote] port for sending/receiving (1024–65535). 
  
  Default: 3333

- `-l`, `--local <local_address>`
  
  Specify address of local product interface.
  
  Default: DBL_LOCAL
-i, --iterations <iterations>
    Set number of send/receive iterations.
    Must be set equally on client and server.
    Default: DBL_ITERATIONS, or 10000 if undefined
-w, --warmup <warmup_count>
    Set number of warmup sends to do from client before starting main loop.
    Default: 10000
-t, --timeout <seconds>
    Set time to run before terminating application.
-S, --size <size>
    Set size of packet in bytes to send/receive (min).
    Default: 1
-E, --endsize <size>
    Set size of packet in bytes to send/receive (max).
    Default: 2048
-D, --excelPrint
    Outputs the data in an excel frienly format.
-I, --increment
    Set number of bytes to increment packet size
    Default: power of 2

The DBL_ADD_MIX_CHANNELS environment variable, (as in export DBL_ADD_MIX_CHANNELS=5) demonstrates the mix of DBLv5.5.0 and TCP channels (required on both ends) on a device.

### 7.4.10. dbltcp_tick_to_trade

Measures the time interval between a UDP receive and a TCP send by comparing the received UDP packet RX timestamp with the sent TCP packet TX timestamp. The dbltcp_tick_to_trade program runs with DBLv5.5.0 Extension API.

**Definition:**

```
dbltcp_tick_to_trade
```

**Example: (Server side)**

```
$ ./dbltcp_tick_to_trade -l 10.0.0.2 -s
```

**Example: (Client side)**

```
$ ./dbltcp_tick_to_trade -l 10.0.0.1 -h 10.0.0.2 -i 100000
```
Usage:

./dbltcp_tick_to_trade [options]

Command line [options]:

-?, --help
   Print help information and usage.
-s, --server
   Run as server.
-h, --remote <remote_address>
   Specify remote address.
   Default: DBL_REMOTE
-p, --port <port>
   Use [remote] port for sending/receiving (1024-65535).
   Default: 3333
-j, --mcast <mcast_address>
   Specify the multicast address to join.
   Default: DBL_MCAST, or 224.1.1.50 if undefined
-l, --local <local_address>
   Specify address of local product interface.
   Default: DBL_LOCAL
-i, --iterations <iterations>
   Set number of send/receive iterations.
   Must be set equally on client and server.
   Default: DBL_ITERATIONS, or 10000 if undefined
-w, --warmup <warmup_count>
   Set number of warmup sends to do from client before starting main loop.
   Default: 10000
-T, --tcp_loops <num_loops>
   Loop over preset TCP sizes (1-5).
   The program will loop with up to 5 preset TCP sizes.
   TCP sizes are { 5, 38, 64, 128, 400 }.
   Default: 1
-m, --modulus <num_packets>
   Print status after some number of packets has been received.
   Default: 100000
-r, --reuse
   Reuse address.
-D, --excelPrint
   Outputs the data in an excel frienly format.
7.4.11. **sock_tick_to_trade**

**Definition:**

`sock_tick_to_trade`

Measures the time interval between a UDP receive and a TCP send by comparing the received UDP packet RX timestamp with the sent TCP packet TX timestamp. This program runs with Socket API.

**Example: (Server side)**

```bash
$ dblrun sock_tick_to_trade -l 10.0.0.2 -s
```

**Example: (Client side)**

```bash
$ dblrun sock_tick_to_trade -l 10.0.0.1 -h 10.0.0.2 -i 100000
```

**Usage:**

```
./sock_tick_to_trade [options]
```

**Command line [options]:**

- `?`, `--help`
  
  Print help information and usage.

- `s`, `--server`
  
  Run as server.

- `h`, `--remote <remote_address>`
  
  Specify remote address.
  
  Default: SOCK_REMOTE

- `p`, `--port <port>`
  
  Use [remote] port for sending/receiving (1024-65535).
  
  Default: 3333

- `j`, `--mcast <mcast_address>`
  
  Specify the multicast address to join.
  
  Default: SOCK_MCAST, or 224.1.1.50 if undefined
-l, --local <local_address>
    Specify address of local product interface.
    Default: SOCK_LOCAL
-i, --iterations <iterations>
    Set number of send/receive iterations.
    Must be set equally on client and server.
    Default: SOCK_ITERATIONS, or 10000 if undefined
-w, --warmup <warmup_count>
    Set number of warmup sends to do from client before starting main loop.
    Default: 10000
-T, --tcp_loops <num_loops>
    Loop over preset TCP sizes (1-5).
    The program will loop with up to 5 preset TCP sizes.
    TCP sizes are { 5, 38, 64, 128, 400 }.
    Default: 1
-m, --modulus <num_packets>
    Print status after some number of packets has been received.
    Default: 100000
-r, --reuse
    Reuse address.
-U, --udp_loops <num_loops>
    Loop over preset UDP sizes (1-5).
    The program will loop with up to 5 preset UDP sizes
    UDP sizes are { 12, 38, 64, 128, 400 }.
    Default: 1

**NOTE**

The **dblrun** command affects **sock_tick_to_trade** socket program performance. For example, running "**dblrun -b 0 ...**" performs better with the faster non-compliant UDP stack.
7.4.12. *tcp_pingpong*

The test program is a simple point-to-point ping-pong latency test over TCP. If specified, you can run the test over UDP. Latency results may improve if you run the test in IP acceleration mode with the `dblrun` script.

**Definition:**
tcp_pingpong

**Example: (Server side)**

```
$ ./tcp_pingpong -s -l 10.0.0.1 -p 3333 -i 10000
```

**Example: (Client side)**

```
$ ./tcp_pingpong -h 10.0.0.1 -l 10.0.0.2 -p 3333 -i 10000
```

**Usage:**

```
./tcp_pingpong [options]
```

**Command line [options]:**

- `?-`, `--help`
  
  Print help information and usage.

- `s`, `--server`
  
  Run as server.

- `h`, `--remote <remote_address>`
  
  Specify remote address.
  Default: SOCK_REMOTE

- `p`, `--port <port>`
  
  Use [remote] port for sending/receiving (1024-65535).
  Default: 3333

- `l`, `--local <local_address>`
  
  Specify address of local product interface.
  Default: SOCK_LOCAL

- `i`, `--iterations <iterations>`
  
  Set number of send/receive iterations.
  Must be set equally on client and server.
  Default: SOCK_ITERATIONS, or 10000 if undefined

- `w`, `--warmup <warmup_count>`
  
  Set number of warmup sends to do from client before starting main loop.
  Default: 10000

- `-g`, `--delay <usecs>`
Set delay.

-t, --timeout <seconds>
Set time to run before terminating application.

-S, --size <size>
Set size of packet in bytes to send/receive (min).
Default: 1

-E, --endsize <size>
Set size of packet in bytes to send/receive (max).
Default: 2048

-6, --ipv6
Set socket family to IPv6.

-u, --udp
Set socket type to UDP.
Must be set on both client and server.

-D, --excelPrint
Outputs the data in an excel friendly format.

-f, --fork
Fork the process after accept.

-z, --affinity <value>
Set process affinity using bitmask.

-0, --overlap <overlap_mode>
Select overlap mode.
0 - Blocking [default]
1 - Select
2 - Poll
3 - Nonblocking (FIONBIO)

-X, --send_cost
Measure client blocking send() cost instead of HRT.
8. Running DBLv5.5.0 Diagnostic Tool Programs

The DBLv5.5.0 software package includes diagnostic tool programs, found in the sbin/ and bin/ directories.

Summary of diagnostic tool programs:

- sbin/phx_bug_report (Linux only)
- phx_bug_report.ps1 (Windows only)
- sbin/myri_info (both)
- sbin\team_config.exe (Windows only)
- bin/dblttool (both)
- bin/myri_counters (both)
- bin/myri_endpoint_info (both)
- bin/myri_nic_info (both)

The DBLv5.5.0 diagnostic tool programs contain different arguments and outputs than in previous hardware generations. For example, the myri_counters and myri_endpoint_info outputs are noticeably different, which are described in detail below.

The diagnostic tools packages phx-tools-<version>.tar.gz (Linux) and phx-tools-x64-<version>.msi (Windows), are listed as the Toolkit – Phoenix Group in the ARIA Customer Portal. It is similar to the myri-tools.tar.gz package in previous hardware generations (Myri-10G) except under different tool names. Various tool options and output have also changed to reflect the ARC Series E network adapter requirements.

Run /sbin/ tools as root, with the option of adding /opt/dbl/sbin/ to your PATH, or by executing the full path command, such as /opt/dbl/sbin/myri_info.

Most of the diagnostic tools generate diagnostic information for error reporting.
8.1.1. **sbin/phx_bug_report (Linux only)**

Run the `sbin/phx_bug_report` diagnostic bug report script as root.

**Description:**

`sbin/phx_bug_report`

**Example:**

```
$ sudo /opt/dbl/sbin/phx_bug_report
```

For more information, go to *Troubleshooting*

8.1.2. **phx_bug_report.ps1 (Windows only)**

The `phx_bug_report.ps1` diagnostic tool is a Windows diagnostic Powershell script. You must have Administrator privileges to run the Powershell script.

**Description:**

`phx_bug_report.ps1`

**Example:**

```
PS> cd C:\Program Files\CSPi\phx-tools\nPS> phx_bug_report.ps1 > output.txt
```

For more information, go to *Troubleshooting*

8.1.3. **sbin/myri_info**

Provides network adapter information, such as hardware serial number, MAC address, firmware version, and so on.

**Description:**

`sbin/myri_info`

**Usage:**

```
$ sudo /opt/dbl/sbin/myri_info [options]
```

**Command line [options]:**

```
[ -b <board_num> ] Only print info about card instance <board_num>.
[ -v ] Verbose.
```
8.1.4. **sbin\team_config_x64.exe (Windows only)**

The `team_config_x64.exe` command line tool installs the VLAN driver and configures the VLAN network adapters. For a complete listing of all command line arguments, run `team_config_x64.exe -h`.

**Description:**
```
sbin\team_config_x64.exe
```

8.1.5. **bin/dbltool**

Provides `netstat`-style results for DBLv5.5.0 TA mode, by displaying packet counters dropped by the stack. Since DBLv5.5.0 is kernel-bypass, all packets dropped by DBLv5.5.0 will display in the `myri_counters` output.

**Description:**
```
bin/dbltool
```

**Usage:**
```
$ /opt/dbl/bin/dbltool -p <pid> <cmd> [cmd options ...]
```

**Command line [options]:**

```
[ -p <pid> ] pid of the process you want to examine.
<cmd> The command you would like to send. The supported commands are mtcp_config, mtcp_stats, and mtcp_routes.
```

8.1.6. **bin/myri_counters**

Generates output for low-level network adapter counters.

**Description:**
```
bin/myri_counters
```

**Command line [Help]**
```
$ /opt/dbl/bin/myri_counters -h
```

**Usage:**
```
Usage: myri_counters [args]
```
Resetting the counters:
To clear the counters on a specific port of the network adapter, enter the following command line:

**Command line:**

```bash
$ sudo /opt/dbl/bin/myri_counters -p <port_num> -c
```

Resetting the counters requires root privileges.

For more information on `myri_counters`, go to *Appendix 1: DBLv5.5.0 Counters*  

8.1.7. **bin/myri_endpoint_info**

Identifies which processes consume adapter-level resources, such as endpoints. Tracks which process IDs are in use.

**Description:**

`bin/myri_endpoint_info`

**Usage:**

```bash
$ /opt/dbl/bin/myri_endpoint_info [options]
```

**Command line [options]:**

```bash
[ -b <board_num> ] Board number or MAC address. Default: 0.
```
8.1.8.  *bin/myri_nic_info*

Provides diagnostic information on the number of adapters installed, which driver is loaded, and the status of the software license(s) for each network adapter.

**Description:**

*bin/myri_nic_info*

**Usage:**

$ /opt/dbl/bin/myri_nic_info [options]

**Command line [options]:**

- `-h` | Help.
- `-B` | Display board numbers.
- `-m` | Comma separated output (for machine parsing).
- `-a` | Print for all known adapters.
- `--license` | Print software licensing status for all known adapters.
9. Transparent Sockets Acceleration (TA) Software

This chapter contains the following topics:

- Installing Transparent Acceleration (TA) Modes
- DBLv5.5.0 - Specific Socket Extensions
- Implementing DBLv5.5.0 TA in Linux
- Implementing DBLv5.5.0 TA in Windows
- TCP and UDP Acceleration with DBLv5.5.0
- Running DBLv5.5.0 TA Test Programs

9.1. Installing Transparent Acceleration (TA) Modes

This section describes how to manually install the Transparent Acceleration (TA) mode for TCP and UDP sockets (Winsock), as follows:

- Manually installing and configuring DBLv5.5.0 TA mode via LSP
- Checking the status of DBLv5.5.0 TA mode via DLL
- Uninstalling DBLv5.5.0 TA via DLL
- Enabling Transparent Acceleration (TA)

| NOTE | The TA installation process defaults to the LSP method. Specify TA mode DLL to install TA via the DLL method. |

9.1.1. Manually Installing and Configuring DBLv5.5.0 TA Mode via LSP

To install and configure TA mode via LSP, follow these steps:

1. Open an elevated cmd: prompt, select the TA/LSP directory, and enter the following command(s):
   
   ```
   $ instlsp.exe -i -d [TARGETDIR]\lib\ipdb1_lsp11.dll
   ```
   (for Winsock 1.1 support)
OR

$ instlsp.exe -i -d [TARGETDIR]\lib\ipdbl_lsp22.dll

(for Winsock 2.2 support)

The Administrator:cmd screen appears, showing TA via LSP installing.

2. Enter the following command to check that the LSP was installed:

$ instlsp.exe -p

The Administrator:cmd screen appears, confirming TA via LSP has been installed.
9.1.2. **Checking the Status of DBLv5.5.0 TA via DLL**

To check the status of DBLv5.5.0 TA via DLL, follow these steps:

1. Open an elevated cmd: prompt and select the TA/DLL directory.
2. Select the following command from the `[INSTALLDIR]\sbin` directory:
   
   ```
   $ instdllinj.exe -p
   ```

9.1.3. **Uninstalling DBLv5.5.0 TA via DLL**

To uninstall DBLv5.5.0 TA via DLL, follow these steps:

1. Open an elevated cmd: prompt and select the TA/DLL directory.
2. Select the following command from the `[INSTALLDIR]\sbin` directory:
   
   ```
   $ instdllinj.exe -f
   ```
3. Restart the computer if the DLL has to be removed.
4. Enter Run.

   ```
   $ instdllinj.exe -i -d [TARGETDIR]\lib\ipdbl_inj11.dll
   ```

   (for Winsock 1.1 support)

   OR

   ```
   $ instdllinj.exe -i -d [TARGETDIR]\lib\ipdbl_inj22.dll
   ```

   (for Winsock 2.2 support)

**NOTE**

The `dblrun.exe` file must be run to accelerate the application.
Only one registered method (LSP or DLL) can be run at a time.
9.1.4. Enabling Transparent Acceleration (TA)

The `dblrun.exe` command contains several variables to guide TA acceleration. Most notably, it can pass the `DBLv5_IP_ACCELERATION=1` environment variable to a specified application.

| NOTE | A manual setting is *not* recommended. Contact ARIA_support@ariacybersecurity.com if you cannot run the `dblrun.exe` program. |

For more information on TA mode failures, refer to the *TA Failure* section of *Transparent Sockets Acceleration*.

9.2. DBLv5.5.0 -Specific Socket Extensions

Four DBLv5.5.0 -specific sockets extensions are available for Linux and Windows.

- MSG_WARM
- SIO_GETNICTIME
- SO_TIMESTAMP
- SO_TIMESTAMPING, MSG_ERRQUEUE

9.2.1. MSG_WARM

In general, socket applications TCP send performance diminishes from cache misses when timely calls to the code path are not made. The DBLv5.5.0 socket flag `MSG_WARM` deploys `send()` function calls to reduce TCP cold cache symptoms.

The socket flag occasionally allows an application to call `send(s, buf, len, MSG_WARM)` to cache the TCP output code lines (by not injecting any data) and to avoid cache misses.

**Command line:**

```c
dbl.h: #define MSG_WARM 0x20000
```

Where `dbl.h` defines the `MSG_WARM` flag

For more information, go to the DBLv5.5.0 API Reference Manual.
9.2.2. **SIO_GETNICTIME**

The SIO_GETNICTIME sockets extension is available in DBL 3.0.3.52632 and later. On a UDP/TCP socket, an application refers to the `ioctl` (Linux) or `WSAIoctl` (Windows) function call to retrieve the current NIC time.

The argument is a pointer to `dbl_ticks_t` defined in `dbl.h`

**Definition:**

```c
#include <dbl.h>

#define SIO_GETNICTIME 0x5465
```

Where `dbl.h` defines `SIO_GETNICTIME`.

For more information, go to the DBLv5.5.0 API Reference Manual.

9.2.3. **SO_TIMESTAMP**

DBLv5.5.0 TA mode supports the SO_TIMESTAMP[NS] socket extension. Windows does not.

Set the `setsockopt` value to 1 to enable timestamps as follows:

**Definition:**

```c
#include <dbl.h>

#define SO_TIMESTAMP 0x0400
```

Running `SO_TIMESTAMP[NS]` in `setsockopt()` and the application retrieves timestamps (similar to what Linux offers).

For more information on timestamping, refer to the *Network Adapter Timestamps* section in *Troubleshooting*.

9.2.4. **SO_TIMESTAMPING, MSG_ERRQUEUE**

Linux kernels allow for timestamping outgoing TCP packets by setting the `SO_TIMESTAMPING` variable through the `setsockopt()` function call.

This feature is crucial in the trading environment by providing strict oversight and monitoring from the time an order is transmitted via the network adapter. Call `recvmsg()` with `MSG_ERRQUEUE` to retrieve the timestamp.
Windows does not support timestamping; however we can pass the SO_TIMESTAMPING option to the `setsockopt()` function call for the user-level TCP stack. A Windows application retrieves the timing information from the `WSARecvMsg` socket and `MSG_ERRQUEUE` function call.

We define the following socket options in `dbl.h` as follows:

```c
#define SO_TIMESTAMPING 0x0025 /* 37 */
#define MSG_ERRQUEUE    0x30  /* 48 */
```

The command applies to Linux systems not currently supporting TX timestamping.

For more information on timestamping, refer to the `Network Adapter Timestamps` section in `Troubleshooting`.

### 9.3. Implementing DBLv5.5.0 TA in Linux

Library preloads initiate socket acceleration in Linux. The `libdbl_preload.so` library preload intercepts socket function calls and maps DBLv5.5.0 UDP and TCP applications.

The concept relies on dynamically linked libraries (DLLs) that remap socket function calls. The functionality also depends on your `libc` library preload version.

<table>
<thead>
<tr>
<th>NOTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>We strongly recommend running the TA scenarios to check system functionality before proceeding to more complex environments.</td>
</tr>
</tbody>
</table>

Define DBLv5.5.0 TA mode timestamping through the standard socket `SO_TIMESTAMP[NS]` option.

For more information on timestamping and socket application failures refer to the `Network Adapter Timestamps` section in `Troubleshooting`.
9.4. Implementing DBLv5.5.0 TA in Windows

There are two ways to run DBLv5.5.0 TA in the Windows environment:

- Accelerating TA through Layered Service Providers (LSPs)
- Accelerating TA as a DLL Library Preload

**NOTE**
Only one DBLv5.5.0 TA method (LSPs or DLL preload) may be active/registered at one time.

9.4.1. TA Acceleration through Layered Service Providers (LSPs).

You can run TA Acceleration based on Layered Service Providers (LSPs) in two ways:

- With IFS LSPs
- With non-IFS LSPs

Both methods are useful because they can run unmodified applications with accelerated modes.

**Functionality:**

Functionality differentiates one method from the other:

- IFS LSP only offers Winsock 1.1 functionality.
- Non-IFS LSP (ipdbl_lsp22.dll) offers Winsock 2.2 functionality (overlapped calls, IOCP support).

Winsock 2.2 encapsulates Winsock 1.1 functionality. A DBLv5.5.0 TA host installation can only enable one mode at a time: Winsock 1.1 or Winsock 2.2, but not both.

For more information on configuring DBLv5.5.0 TA mode, go to *Installing the DBLv5.5.0 Software*

**Installing/Removing LSP:**

To verify that LSP has been installed, enter the following command:

```
$ instlsp.exe -p
```
Switching Winsock support:

You can switch between DBLv5.5.0 TA Winsock1.1 support and Winsock 2.2 as follows:

1. To query status, run the `instlsp.exe` tool in `[INSTALLDIR]\sbin` and enter the following command:
   
   ```sh
   $ instlsp.exe -p
   ```

2. To remove LSP, run the `instlsp.exe` tool in `[INSTALLDIR]\sbin` and enter the following command:
   
   ```sh
   $ instlsp.exe -f
   ```

3. Reboot to remove LSP.
4. Enter the following command to install LSP for Winsock 1.1 support:
   
   ```sh
   $ instlsp.exe -i -d [TARGETDIR]\lib\ipdbl_lsp11.dll
   ```

5. Enter Run.
   
   OR
   
   Enter the following command to install non-LSP for Winsock 2.2 support:
   
   ```sh
   $ instlsp.exe -i -d [TARGETDIR]\lib\ipdbl_lsp22.dll
   ```

6. Enter Run.

9.4.2. **TA Acceleration by DLL preload**

TA by DLL preload intercepts the application at the Winsock API level, bypassing all LSPs and reducing latency to the 250 to 300 nanosecond range. The program is Winsock 1.1 and Winsock 2.2 compliant.

You can switch between DBLv5.5.0 TA Winsock1.1 support and Winsock 2.2 as follows:

1. To query status, run the `instdllinj.exe` tool in the `[INSTALLDIR]\sbin` directory to determine if a DLL is registered.
   
   ```sh
   $ instdllinj.exe -p
   ```

2. To remove the DLL, run the `instdllinj.exe` tool in the `[INSTALLDIR]\sbin` directory, as follows:
   
   ```sh
   $ instdllinj.exe -f
   ```

3. Reboot to remove the DLL.
4. Enter the following command to install the DLL with Winsock 1.1 support:
   
   ```sh
   $ instdllinj.exe -i -d [TARGETDIR]\lib\ipdbl_inj11.dll
   ```
5. Enter Run.

OR

Enter the following command to install the DLL with Winsock 2.2 support.

$ instdllinj.exe -i -d [TARGETDIR]\lib\ipdbl_inj22.dll

6. Enter Run.

| NOTE | You can only run dblrun.exe with a registered DLL preload to accelerate the application. |
|      | Only one DLL preload can be active/registered at a time. |

For more information on socket application failures, refer to the TA Failure section in Troubleshooting

9.5. TCP and UDP Acceleration with DBLv5.5.0

After you have installed and tested the DBLv5.5.0 software, you can set up TCP and UDP acceleration on an existing program binary (no coding changes or recompilation required) by running the following dblrun program options as follows:

Definition:

dblrun [options] executable arguments

Command line [options]:

```
-i ip_address  DBL IP Address of Accelerating Device for UDP TA to support autobind [myri0]
-l libdir Full path to preload directory [/opt/dbl/lib]
-f logfile Full path to log file [off, e.g. /tmp/log.txt]
-d Bypass acceleration under ld_preload [off]
-b {0,1} use DBL API (0) or BSD stack (1) for all UDP sockets [1]
-v Verbose startup [off]
-u UDP acceleration only [off]
-t TCP acceleration only [off]
-o {0,1} TCP zero copy send optimization [off]
-c cpu_number Process Affinity mask [off]
-C <path_to>/dbl.conf Use non-default dbl.conf for application
-b Multiplexes UDP sockets using the BSD stack [on]
-w (Windows only) Forward cntrl-x events to target application
-? This usage information
```
There are two ways to verify that DBLv5.5.0 TA mode is active.

**Method 1:**
Set the environment variable DBL_IP_VERBOSE to 1:

```
$ DBL_IP_VERBOSE=1 dblrun [options] executable arguments
```

Diagnostic output prints to the console.

**Method 2:**
Enter the following command:

```
dblrun -v <appname>.
```

The following output appears for Windows DBLv5.5.0 TCP:

```
SPSocket: Enabled fd 276 for TCP TA
```

Where `fd` is accelerated.

### 9.5.1. Known Issues

**LSP**

If the notation, “TCP TA”, does not appear in the Windows DBLv5.5.0 TA mode output text, check that LSP is installed by entering the following command:

```
$ instlsp.exe -p
```

If LSP is not installed, refer to the section, *Customizing Transparent Acceleration and TCP API mode using dbl.conf* in this chapter to accelerate applications with the `dblrun.exe <appname>` command line.

For more information, go to Appendix 3: DBLv5.5.0 Driver Restrictions and Limitations.

**Sockets**

If you exceed the number of available DBL endpoints under DBLv5.5.0 TA, enter the following command:

```
$ dblrun -b 1 myapp.exe
```

The `-b 1` option multiplexes UDP sockets on the general TA BSD stack along with TCP sockets, reducing the number of DBL endpoints used, at the expense of increasing latency on the UDP sockets. This also applies to accelerated applications running concurrently and where few endpoints are available.
The `dblrun -i` default command only applies to UDP traffic in the event TA mode runs the DBLv5.5.0 API. It does not affect TCP or UDP traffic when run with `dblrun -b`. It serves only to support binding to INADDR_ANY. In this case a DBLv5.5.0 interface can be specified on which traffic is expected.

If the sockets application fails to run successfully in Transparent Acceleration (TA) mode refer to the TA Failure section in Troubleshooting

### 9.5.2. Special considerations

- DBLv5.5.0 TA mode is restricted to UDP and TCP. Other protocols such as SOCK_RAW are not accelerated.

- DBLv5.5.0 TA mode (`dblrun`) supports loopback communication on the loopback IP address, such as IP 127.0.0.1

- Running loopback with DBLv5.5.0 TA mode (`dblrun`) demotes a socket and redirects the traffic over internal OS-dependent paths - a much faster option than sending packets through DBLv5.5.0.

| NOTE | A network configuration with more than one IP address on the same subnet is not supported. |

Workaround:

Set the `DBL_BYPASS_CHECK=1` environmental variable as your workaround. The host network stack will not know which of the two ports on the same subnet to route an outgoing packet to, but TA applications will be bound to a particular vendor’s card using that same subnet.
9.6. Running DBLv5.5.0 TA Test Programs

The DBLv5.5.0 TA test programs are found in the following directories:

- `/opt/dbl/bin/tests` directory in Linux.
- `C:\DBL_PHX-10G\bin\tests` directory in Windows.

9.6.1. Test programs:

The following test programs evaluate DBLv5.5.0 TA performance and are described in this section:

- Standard IP ping-pong latency test for UDP
- Accelerated IP ping-pong latency test for UDP
- Standard IP ping-pong latency test for TCP
- Accelerated IP ping-pong latency test for TCP

Methodology:

Run the `tcp_pingpong` program for UDP and TCP ping-pong testing between server and client machines. Compare the performance data when the two machines are connected back-to-back (switchless). `tcp_pingpong` is a pure sockets program that bounces TCP (or UDP) packets from one host to another and records latency in microseconds.

Run the tests in this section in standard IP (without `dblrun`) and in IP acceleration mode (with `dblrun`) to evaluate the system configuration and record any improvement in performance.

9.6.2. Special considerations

- DBLv5.5.0 TA mode (`dblrun`) supports loopback communication on the loopback IP address, such as IP 127.0.0.1
- Running loopback with DBLv5.5.0 TA mode (`dblrun`) demotes a socket and redirects the traffic over internal OS-dependent paths; a much faster option than sending packets through DBLv5.5.0.

<table>
<thead>
<tr>
<th>NOTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>A network configuration with more than one IP address on the same subnet is not supported.</td>
</tr>
</tbody>
</table>
Workaround:

Set the DBL_BYPASS_CHECK=1 environmental variable as your workaround. The host network stack will not know which of the two ports on the same subnet to route an outgoing packet to, but TA applications will be bound to a particular vendor’s card using that same subnet.

9.6.3. **Standard IP ping-pong latency test for UDP**

Run the standard IP ping-pong latency test for UDP as follows:

1. Enter the following commands in both machines:
   - `cd /opt/dbl/bin/tests`
   - `$ PATH=$PATH:/opt/dbl/bin:/opt/dbl/bin/tests`
2. Enter the following command in the Server host:
   - `$ tcp_pingpong -u -s -l <local_ip>`
3. Enter the following command in the Client host:
   - `$ tcp_pingpong -u -l <local_ip> -h <server_ip> -i 50000`
   - A successfully executed test reports a sequence of communications between the two machines.
4. Record any system delays and continue to the next step, **Accelerated ping-pong latency test for UDP**.
5. Close all active programs.
   - CPU/ host speed is a crucial factor in host latency performance. For example, when two machines are connected by a 10-gigabit switch, the overall CPU/host latency may vary depending on the make and model of the switch.

9.6.4. **Accelerated ping-pong latency test for UDP**

Run the accelerated ping-pong latency test for UDP as follows:

1. Enter the following commands in both machines:
   - `$ cd /opt/dbl/bin/tests`
   - `$ PATH=$PATH:/opt/dbl/bin:/opt/dbl/bin/tests`
2. Enter the following command in the Server host:
   - `$ dblrun tcp_pingpong -u -s -l <local_ip>`
3. Enter the following command in the Client host:
   - `$ dblrun tcp_pingpong -u -l <local_ip> -h <server_ip> -i 50000`
Under normal conditions, the output displays accelerated TCP traffic with much smaller latency outcomes than conventional Ethernet. Some variability in individual “ping” numbers is normal.

4. Close all active programs.

CPU/host speed is a crucial factor in host latency performance. For example, when two machines are connected by a 10-gigabit switch, the overall CPU/host latency may vary depending on the make and model of the switch.

For more information on latency, go to Tuning.

For more information on TA failures, refer to the TA Failure section in Troubleshooting.

9.6.5. **Standard IP ping-pong latency test for TCP**

Run the standard IP ping-pong latency test for TCP as follows:

1. Enter the following commands in both machines:
   
   ```
   $ cd /opt/dbl/bin/tests
   $ PATH=$PATH:/opt/dbl/bin:/opt/dbl/bin/tests
   ```

2. Enter the following command in the Server host:
   
   ```
   $ tcp_pingpong -s -l <local_ip>
   ```

3. Enter the following command in the Client host:
   
   ```
   $ tcp_pingpong -l <local_ip> -h <server_ip> -i 50000
   ```

   A successfully executed test reports a sequence of communications between the two machines.

4. Record any system delays and continue to the next step, *Accelerated ping-pong latency test for TCP*.

5. Close all active programs.

   CPU/host speed is a crucial factor in host latency performance. For example, when two machines are connected by a 10-gigabit switch, the overall CPU/host latency may vary depending on the make and model of the switch.
9.6.6.  **Accelerated ping-pong latency test for TCP**

Run the accelerated ping-pong latency test for TCP as follows:

1. Enter the following command in the Server host:
   
   ```
   $ dblrun tcp_pingpong -s -l <local_ip>
   ```

2. Enter the following command in the Client host:
   
   ```
   $ dblrun tcp_pingpong -l <local_ip> -h <server_ip> -i 50000
   ```

   Under normal conditions, the output displays accelerated TCP traffic with much smaller latency outcomes than conventional Ethernet. Some variability in individual “ping” numbers is normal.

3. Close all active programs.

   CPU/ host speed is a crucial factor in host latency performance. For example, when two machines are connected by a 10-gigabit switch, the overall CPU/host latency may vary depending on the make and model of the switch.

   For more information on latency, go to *Tuning*

   For more information TA failures, refer to the *TA Failure* section in *Troubleshooting*
10. Linux PTP Host Clock Synchronization

This chapter contains the following topics:

- Synchronizing the System Clock to the ARC Series E Adapter Clock
- Installing the Linuxptp Package
- Synchronizing the ARC Series E Clock

10.1. Synchronizing the System Clock to the ARC Series E Adapter Clock

With DBLv5.5.0, the ARC Series E network adapter clock displays as a PTP Hardware Clock on compatible Linux systems (Linux kernel 3.5 and later).

To synchronize your system clock with the network adapter clock, refer to the following procedure:

1. Install the clock as /dev/ptpN where N is determined by the operating system when the driver is loaded.
2. Enter the ethtool –T command.
   The command generates reports in the PTP Hardware Clock field displaying which PTP device is connected to the network adapter.

Example:
In the following example, the network adapter is installed as enp2s0f0, and the PTP clock is /dev/ptp0:

Command line:
`% ethtool -T enp2s0f0`

Output:

```
Time stamping parameters for enp2s0f0:
Capabilities:
    hardware-transmit    (SOF_TIMESTAMPING_TX_HARDWARE)
    hardware-receive     (SOF_TIMESTAMPING_RX_HARDWARE)
    hardware-raw-clock   (SOF_TIMESTAMPING_RAW_HARDWARE)
PTP Hardware Clock: 0
Hardware Transmit Timestamp Modes:
```
Hardware Receive Filter Modes:

- off (HWTSTAMP_TX_OFF)
- on (HWTSTAMP_TX_ON)

3. You can read, set, and adjust the clock using standard PTP functions included in the `linuxptp` package or an equivalent utility.

10.2. Installing the Linuxptp Package

To install the `Linuxptp` package:


   OR

2. Install through a package manager such as Yum, by entering the following command:

   ```bash
   $ sudo yum install linuxptp
   ```

   Reference the hardware clock by its PTP device name (`/dev/ptpN`) or by its network name, such as `eth3` or `enps20f0`, for example.

   All ports on a single network adapter share a single PTP clock. While there may be two or four network interfaces, there will only be one PTP clock. Access to the clock through different network devices will reference the same physical clock.

   **NOTE**
   Make clock adjustments carefully as all timestamps originate from the interface.

10.2.1. Reading the clock by PTP device name:

Command line:

```bash
$ sudo phc_ctl /dev/ptp0 get
```
10.2.2. **Reading the clock by network name:**

The hardware clock is set to the system time when the driver is loaded.

**Command line:**

```
$ sudo phc_ctl enp2s0f0 get
```

10.2.3. **Setting the clock manually:**

The ARC Series E network adapter clock uses a high-quality crystal that is typically more accurate than a system clock.

**Command line:**

```
$ sudo phc_ctl /dev/ptp0 set
```

10.2.4. **Synchronizing the system clock with the network adapter clock (phc2sys):**

The **Linuxptp** package synchronizes the system clock to the adapter clock on the network.

**Command line:**

```
$ sudo phc2sys -s /dev/ptp0 -O 0 -m
```

10.2.5. **Maintaining clock synchronization (ptp4l):**

The **Linuxptp** command prompts the PTP daemon to send out periodic messages on the interface. To stop messages from affecting adapter speed, switch to a system interface that supports PTP.

**Command line:**

```
$ sudo ptp4l -i eth0 -p /dev/ptp1 -m -s
```

**Command descriptions:**

- The `eth0` interface connected to the `/dev/ptp1` device, supports PTP.
- The `--s` option forces the PTP daemon to run as a slave.
- The `--m` option instructs the application to direct output to the console.
- Omitting the `--m` option prevents `ptp4l` from directing output to the console if running as a daemon.
- Omitting the `--s` option allows the daemon to assume the role of Grand Master if the **Linuxptp** algorithm determines that it is the most accurate clock.
10.3. Synchronizing the ARC Series E Clock

The additional PTP clock synchronizes the ARC Series E adapters to the clock that was setup in the previous step.

1. To synchronize the ARC Series E adapter clock, enter the `phc2sys` command:
   
   ```
   $ sudo phc2sys -s /dev/ptp1 -c /dev/ptp0 -O 0 -m
   ```

2. The `phc2sys` command designates the `/dev/ptp1` device as the non-ARC Series E network adapter clock and designates the `/dev/ptp0` as the ARC Series E adapter clock.

   The `-O 0` option (capital “oh” and zero) synchronizes the clock with a zero second offset.
11. DBLv5.5.0 Timestamping Support

This chapter contains the following topics:

- Timestamping Module Variables
- Viewing the Time Source Status
- Arista Timestamping Environment Variables

DBLv5.5.0 in API mode supports TX and RX timestamping by default. Timestamping support is also available for DBLv5.5.0 in TA mode.

For more information on implementing timestamping, refer to the Network Adapter Timestamps section in Troubleshooting.

To run timestamping in DBLv5.5.0 TA mode, refer to the Customizing TA with dbl.conf section in Transparent Sockets Acceleration Software.

11.1. Timestamping Module Variables

The DBLv5.5.0 driver has three timestamping module variables to specify the time source.

- myri_clk_enable_pps
- myri_clk_enable_10mhz
- myri_clk_invert_pps

11.1.1. Default setting

By default, we assume that there is nothing connected to the PPS input connector on the front of the card. The driver sets both MYRI_CLK_ENABLE_PPS and MYRI_CLK_ENABLE_10MHZ to zero, and PPS is disabled.

Command lines (Disabled):

```plaintext
myri_clk_enable_pps=0
myri_clk_enable_10mhz=0
```
11.1.2. **Enabling PPS support**

To enable PPS support, set the MYRI_CLK_ENABLE_PPS module variable to 1 in `/opt/dbl/sbin/myri_start_stop` and restart the driver.

**Command line (Enabled):**

```bash
myri_clk_enable_pps=1
```

11.1.3. **Enabling 10MHz support:**

To enable 10 MHz support, set the MYRI_CLK_ENABLE_10MHZ module variable to 1 in `/opt/dbl/sbin/myri_start_stop` and restart the driver.

**Command line (Enabled):**

```bash
myri_clk_enable_10mhz=1
```

11.2. **Viewing the Time Source Status**

To view the time source status, do one of the following:

1. Enter the `dmesg` command in Linux.
   OR
   Select the EventViewer utility in Windows.

   The `myri_info` command displays one set of external time source connections per network adapter. Likewise, two network adapters create a duplicate output for each function.

   **NOTE**
   Do not choose the MYRI_CLK_ENABLE_10MHZ variable if it is not connected to an external 10MHz time source. Disable the variable first before disconnecting the 10MHz time source.

   **Example (Linux):**
   The command for one function and its output is as follows:

   **Command line:**

   ```bash
   $ sudo ./myri_info -b 0
   ```
Output:

pci-dev at 01:00.0 vendor:product(rev)=1c09:4258(01)
  behind bridge root-port: 00:01.0 8086:0c01 (x8.3/x16.3)
Myri-10G-PCIE-8E -- Link x8
  EEPROM String-spec:
    MAC=00:60:dd:43:cd:40
    SN=491940
    PC=10G-PCIE3-8E-2S
    PN=09-04669
    BOM-A

Firmware:
  Version 2.0.6

External Inputs:
  PPS: Disabled
  10Mhz Clock: Disabled
  100Mhz Clock Locked: Locked
11.3. Arista Timestamping Environment Variables

Arista 7150 series 10-gigabit switches offer timestamping that is supported by DBL software versions 3.0 to 5.5.0.

The DBLv5.5.0 driver supports the Arista series before-FCS timestamp mode, which adds four bytes to the end of each packet (and recalculates the FCS). In addition, the Arista periodically sends keyframe packets describing the time references.

For more information on the Arista 7150 series, go to: https://eos.arista.com/timestamping-on-the-7150-series/

For more information on Configuring Arista Timestamping, go to: http://www.aristanetworks.com/docs/Manuals/ConfigGuide.pdf

This section describes the various timestamping environment variables:

- MYRI_ARISTA_ENABLE_TIMESTAMPING = 0,1
- MYRI_ARISTA_ENABLE_SKEW = 0,1
- MYRI_ARISTA_PARAM_KF_DST_IP = <IP ADDRESS>
- MYRI_ARISTA_PARAM_KF_CHECK_PTP = 0,1
- MYRI_ARISTA_PARAM_KF_CHECK_RATE = 0,1
- MYRI_ARISTA_PARAM_KF_KERNDUP = 0,1

11.3.1. MYRI_ARISTA_ENABLE_TIMESTAMPING = 0, 1

1. Start Arista timestamping support by entering the following environment variable:

   MYRI_ARISTA_ENABLE_TIMESTAMPING = 1

The MYRI_ARISTA_ENABLE_TIMESTAMPING environment variable responds when it receives its first keyframe and then initiates timestamping.

When the Arista switch is under heavy load, it may delay timely keyframe delivery and timestamping. Under these conditions, the keyframe arrival interval may surpass the timestamp wrap time of 6.1 seconds. DBLv5.5.0 responds by approximating timestamps before the next keyframe arrives by adding a time lapse.
**Linux**

In Linux, the kernel log displays the following output:

**Output (Linux):**

```
myri_dbl: INFO: ** Arista timestamping [on] supported modes: append (before-fcs) *
features: keyframe check min.rate [on] kf check ptp [on] kf kernel dup [off]
```

**OR**

**Windows**

In Windows, open the Event Viewer to verify that Arista timestamping has been enabled.

**Output (Windows):**

```
myri_dbl: INFO: ** Arista timestamping [on] supported modes: append (before-fcs) *
features: keyframe check min.rate [on] kf check ptp [on] kf kernel dup [off]
```

### 11.3.2. **MYRI_ARISTA_ENABLE_SKEW = 0, 1**

Set the MYRI_ARISTA_ENABLE_SKEW environment variable to 1 to enable support for the skew fields. The default setting is 0.

**Command line (Default):**

```
MYRI_ARISTA_ENABLE_SKEW = 0
```

By invoking the environment variable D BL_IP_VERBOSE, the call to `dbl_open()` prints the string to `stderr`.

**Output:**

```
** Arista timestamping [on], mode: append (before-fcs), source: 1.1.1.0
```

The string describes the various features that are enabled or disabled.

**Output:**

```
* features: keyframe check min.rate [on] kf check ptp [on] kf kernel dup [off] kf skew [off]
```
11.3.3. **MYRI_ARISTA_PARAM_KF_DST_IP = <IP ADDRESS>**

Set the MYRI_ARISTA_PARAM_KF_DST_IP environment variable to the Arista switch keyframe generator IP address. The default setting is 1.1.1.0.

```
MYRI_ARISTA_PARAM_KF_DST_IP = 1.1.1.0
```

11.3.4. **MYRI_ARISTA_PARAM_KF_CHECK_PTP = 0,1**

The MYRI_ARISTA_PARAM_KF_CHECK_PTP environment variable monitors Arista switch PTP synchronization. The default setting is 1.

```
MYRI_ARISTA_PARAM_KF_CHECK_PTP=1
```

11.3.5. **MYRI_ARISTA_PARAM_KF_CHECK_RATE = 0,1**

The MYRI_ARISTA_PARAM_KF_CHECK_RATE environment variable checks keyframe accuracy. The default setting is 1.

```
MYRI_ARISTA_PARAM_KF_CHECK_RATE=1
```

11.3.6. **MYRI_ARISTA_PARAM_KF_KERNDUP = 0, 1**

The MYRI_ARISTA_PARAM_KF_KERNDUP environment variable duplicates keyframe packets to the host stack. The default setting is zero.

```
MYRI_ARISTA_PARAM_KF_KERNDUP = 0
```
12. Tuning

ARIA Cybersecurity Solutions - Myricom high-speed, low-latency networking solutions in real
customer environments vary depending upon the particular details of the network
configuration, end-user applications, and transaction workloads.

The key to maximum performance improvements is latency, which may be accelerated as
much as four times for UDP and two-and-a-half times for TCP transactions in optimal
situations. At present there is no direct way to measure these effects to determine real
acceleration in a particular environment. Also, it must be appreciated that bulk data transfers
and other application-specific delays reduces the effective user-application transaction rate in
real-world situations.

Critical applications often have internal measurements of effective transaction rates. To that
end, we recommend running performance tests before and after installing the ARC Series E
network adapter hardware to compare performance results.

Ideally one should use reliable and recurring workload data. If that is not possible, sample
enough daily traffic to establish typical performance metrics. If there is no noticeable
improvement after installing the hardware, verify that the network adapters and software are
properly installed.

For information on DBLv5.5.0 API mode and
DBLv5.5.0 TA mode performance metrics, go to
Testing DBLv5.5.0 Software

This chapter contains the following topics:

- Tuning Check List
- Customizing Transparent Acceleration (TA) and TCP API Modes using dbl.conf
- Adaptive Interrupt Moderation (AIM)
- Increasing the Size of the Data and Descriptor Rings
- Single-Threaded TCP applications
- Zero Copy TCP Sends
- PIO/DMA Threshold
12.1. Tuning Check List

DBLv5.5.0 is a kernel-bypass driver. As such, it requires little or no tuning. Nonetheless it helps to be aware of the following issues that may affect latency and packet rate.

12.1.1. PCIe expansion slot seating

Verify that the ARC Series E network adapter is seated properly into a PCI-Express Gen3 expansion slot that is compatible with the correct PCI-Express link width (x8).

Run the `myri_info` command from the DBLv5.5.0 `/opt/dbl/sbin/` directory (also available from the `phx-tools` package) to confirm the correct x8 width of the PCI-Express link.

For more information, refer to the Hardware Installation/Performance Issues section in Troubleshooting.

12.1.2. Tune for lowest latency

The `tuned` tuning daemon synchronizes tuning profiles for optimum operating system performance. The `tuned-adm` command line tool allows users to switch between different tuning profiles.

Command line:

```
$ sudo yum install tuned
$ sudo systemctl enable tuned.service
$ sudo systemctl start tuned.service
$ sudo tuned-adm profile latency-performance
```

12.1.3. CPU frequency scaling

Disable CPU frequency scaling on the processor.
12.1.4. **Interrupt balancing**

In some situations, interrupt balancing adversely affects performance. Minimize latency and jitter by:

- Disabling IRQ balancing
- Manually assigning each IRQ to a specific CPU
- Binding processes to a specific CPU (via `taskset` or `numactl`)

12.1.5. **CPU binding (Linux)**

DBLv5.5.0 performance may improve with CPU binding. CPU binding has two components:

- The `taskset` command, which sets and checks affinity information for a given process.
- The `numactl` utility, which binds processes to processors on Linux-based supercomputers.

Hosts with sockets and CPUs in close proximity to the network adapter and memory perform better.

Some systems have multiple PCI-Express root ports. For example, AMD machines and PCI Express slots can connect to one root port, making them more accessible from one CPU socket.

12.1.6. **PCI bridging**

Some machine PCIe slots may have deeper PCI bridging. Extra bridge chips between the CPU and the network adapter may result in higher latency.

Try moving the network adapter to a different PCI-Express slot to improve DBLv5.5.0 latency if there are too many bridge chips. Run the `myri_info` tool to detect PCI bridges between the CPU and network adapter.

12.1.7. **Hyperthreading**

Do not have more than one DBLv5.5.0 receive thread (or process) on a physical core. Allow hyperthreading only when the threads are mapped correctly.
12.1.8. Reducing jitter

Reduce jitter by assigning the DBLv5.5.0 application to a specific core. Exclude all other processes from the core.

12.2. Customizing Transparent Acceleration (TA) and TCP API Modes using dbl.conf

The `dbl.conf` file, found in the `/opt/dbl/` directory, customizes the DBLv5.5.0 Transparent Acceleration (TA) and DBLv5.5.0 TCP API configuration.

By default, running DBLv5.5.0 in TA mode accelerates all TCP and UDP ports, and disables delayed ACKs, including network adapter timestamping.

12.2.1. Restricting TCP and UDP port acceleration

You can modify `dbl.conf` configuration file behavior by restricting which sockets are accelerated by adding the `include` or `exclude` notation, as follows:

**Command lines:**

- `dbl.tcp.ports_include=1-65535`
- `dbl.udp.ports_exclude=1-1023`

**Example:**

An example notation is: `1-79,81-1023,447,993`

**TCP ports**

On TCP, all excluded ports are non-autobound ports:

- Destination port in a client `connect()`
- Source port in a server `listen()`

**UDP ports**

On UDP, excluded ports are only honored for `server binds()`
Include/Exclude rules
The following include/exclude rules apply:

- If only include is provided, only those ports are accelerated.
- If only exclude is provided, only those ports are not accelerated.
- If both are provided, include is applied first, followed by exclude.

12.2.2. Maximizing the number of accelerated UDP sockets
Setting the default value to zero offers a maximum of 32 UDP accelerated sockets per network adapter, system-wide.

Command line: Default
dbl.udp.many_sock=0
Setting the value to 1 removes this limitation at the expense of a little latency.

Command line: Disabled
dbl.udp.many_sock=1

12.2.3. Enabling delayed ACKs
By default, set the delayed ACK to ten milliseconds.

Command lines:
net.inet.tcp.delayed_ack=1
net.inet.tcp.delacktime=10

12.2.4. Disabling checksum on receive
We’ve introduced a new option, net.inet.tcp.rx_chksum, to the /opt/dbl/dbl.conf directory. By setting the default value to 1, the TCP stack computes the receive checksum. By setting the value to zero, TCP receive checksums are not calculated, allowing for faster receives.

Command line: Default
net.inet.tcp.rx_chksum=1

Command Line: Disabled
net.inet.tcp.rx_chksum=0
12.2.5. Assigning CPU affinity masks to service threads*

*using hexa notation, 0x3 for core[0-1]

The CPU affinity mask prevents the service thread from polluting an execution, due to the kernel-scheduling effect on threads. This option allows the user to pin the service thread to cores. The affinity mask does not cause starvation due to low load demands (TCP time-out checks). The user can selectively assign service threads to specific cores or run the OS scheduler.

Command line:
mtcp.thread_softclock_cpu_affinity_mask=0x1
mtcp.thread_monitor_cpu_affinity_mask=0x1

Pinning service threads to cores

Command Line:
mtcp.thread_softclock_cpu_affinity_mask

Running an inactive monitor thread

You can run an inactive “monitor thread” for managing the connection with the OS internal NETLINK.

Command Line:
mtcp.thread_monitor_cpu_affinity_mask

A more active “clock thread” computes timeouts and arms for connections' timeouts timer.

12.2.6. Enable timestamping

Enabling DBLv5.5.0 TA timestamping in dbl.conf affects whether or not you receive a timestamp from the SO_TIMESTAMP[NS] feature.

Windows does not support SO_TIMESTAMP[NS] at the NDIS level; however, you can redefine the command to retrieve timestamps.

NDIS level

Command line: Default
mtcp.hw_timestamping=0

By setting the default to zero, DBLv5.5.0 TA mode timestamping is disabled to minimize latency.
**Hardware timestamping**

Set the value to 1 to enable hardware timestamping for UDP packets. Note that users are still expected to use the Linux style of setting SO_TIMESTAMP on the UDP socket, as well as running the `recvmsg()` function call to receive data and control socket information where timestamps are available as a `struct timeval`.

**Command line: Enabled**

```
mtcp.hw_timestamping=1
```

For more information on retrieving timestamps in Windows, refer to DBL-specific Socket Extension section in *Transparent Sockets Acceleration (TA) Software*.

### 12.2.7. Steering SBL mode for DBL TCP API applications

By default, DBL TCP API applications enable the TCP fastpath features (SBL), found in the `/opt/dbl/dbl.conf` directory. To disable the fastpath feature, set the option to zero.

**NOTE:** SBL mode can run with a VLAN-tagged packet.

**Command line: Default**

```
mtcp.tcp_fastpath=1
```

**Command line: Disabled**

```
mtcp.tcp_fastpath=0
```

### 12.2.8. Enabling a permanent blocking mode

To enable a permanent blocking mode, set the value to 1. Setting the default value to zero disables the permanent blocking mode.

**Command line: Default**

```
mtcp.ip.block=0
```

**Command line: Disabled**

```
mtcp.ip.block=1
```
12.2.9. **Enabling spin-friendly polling mechanisms**

By default, the `mtcp.spin_sleep=0` option polls continuously for the next packet to achieve the lowest latency. By setting `mtcp.spin_sleep` to 1, the receiving thread initially polls aggressively and then will start an approach where it yields the CPU if no packets are received after a certain number of polling iterations. The algorithm will yield longer each time and sleep up to one millisecond. Once a packet is received it will start aggressively polling until the next interval. In previous releases, the poll/yield algorithm was the default.

The spin-friendly polling option impacts users running TCP in TA mode (`dblrn -b 0`, `dblrn -b 1`) or DBLv5.5.0 API modes or UDP in TA mode (`dblrn -b 1`).

**Command line: Default**

```
mtcp.spin_sleep=0
```

Polls continuously for the next packet

**Command line: Disabled**

```
mtcp.spin_sleep=1
```

Polls aggressively at first, then gradually yields the CPU if it receives no packets.

12.2.10. **User-level TCP congestion management control algorithms**

DBLv5.5.0 is compatible with the following congestion-control, loss-based family of algorithms:

- NewReno
- HTCP
- Cubic (recommended for Linux-based latency-sensitive applications)

**Command line:**

```
# net.inet.tcp.cc.algorithm=newreno
```
12.2.11. dbl.conf settings

To specify a `dbl.conf` for a specific application, run one of the following command line arguments to `dblrun`:

**Command line:**

- `C /opt/dbl/dbl.conf`

**OR**

**Command line:**

`DBL_CONFIGFILE=/opt/dbl/dbl.conf`

12.2.12. Special Considerations:

`dblrun -f logfile`

The `dblrun -f logfile` command governs what actions to take when DBLv5.5.0 examines the `dbl.conf` file.

**If you cannot find the `dbl.conf` file**

If the `dbl.conf` file cannot be found:

- Run the `-C` command line option. The target does not start.
- Do **not** run the `-C` command line option. The default `dbl.conf` file cannot be found in the `/opt/dbl/` root directory. DBLv5.5.0 TA mode starts after a warning is logged.

12.3. Adaptive Interrupt Moderation (AIM)

DBLv5.5.0 TA mode runs an Adaptive Interrupt Moderation (AIM) program in Windows that decreases socket latency while bypassing TA altogether. AIM enhances socket applications that rely on overlapping functionality and IOCP support (Winsock 2).

To achieve the optimal low latency, run `poll/select` instead of `IOCP`, as the latter requires a context switch. Compare the results in default mode versus AIM mode without running TA / `dblrun.exe`.

The default setting for AIM is ‘disabled’.
12.4. Increasing the Size of the Data and Descriptor Rings

If the application's traffic pattern is bursty and the application is unable to consume the input data stream at its presented rate, dropped packets may occur. To absorb bursty traffic patterns and prevent dropped packets, increase the size of the descriptor and data rings in DBLv5.5.0. The application data and descriptor ring memory is allocated for each DBLv5.5.0 endpoint. Increasing the descriptor ring size increases DBLv5.5.0 endpoint memory usage.

The `MYRI_RECVQ_VPAGE_CNT` environment variable controls data and descriptor ring size. Its value determines how many pages of memory are allocated to the descriptor rings. The operating system determines page size (4k or 8k). The number of available pages is limited by the amount of physical memory on the server (up to 4GB). The size of the descriptor ring is usually half the size of the data ring. The default value of `MYRI_RECVQ_VPAGE_CNT` is 32768 pages (128MB).

**Linux:**

In Linux, we need to modify/uncomment the `MYRI_MODULE_PARAMS` variable line in the `sbin/myri_start_stop` script to increase the size of the data and descriptor rings, as follows:

```
MYRI_MODULE_PARAMS=" myri_recvq_vpage_cnt=131072 $MYRI_MODULE_PARAMS"
```

After power cycling the host and reloading the DBLv5.5.0 driver, verify the setting change by checking the **EP Data Pages** and **EP Desc Pages** counter values in the `myri_counters` output. The counters are only non-zero when an endpoint is in use.

Generate traffic before checking the **EP Data Pages** and **EP Desc Pages** counter values in the `myri_counters` output, as described in the following example.

**Output:**

```
$ dbl_pingpong -s -l <IP_address>

$ bin/myri_counters -e 1 -v
(snip)  
Endpoint counters for endpoint 1  
(snip)  
EP Data Pages: 131072  
EP Desc Pages: 65536
```

The **EP Desc Pages** counter value is always half the size of the data ring.
Windows:

In the following examples, ring size is set at four times the default setting (131072 pages or 512MB).

In Windows, we need to modify a registry entry followed by power cycling the system to increase the size of the data and descriptor rings as follows:

```
REG ADD HKLM\SYSTEM\CurrentControlSet\services\dbl /v myri_recvq_vpage_cnt /t REG_DWORD /d 131072
```

12.4.1. When power cycling the host is not possible

Linux:

Enter the following command to reload the driver:

```
$ sudo ./myri_start_stop restart
```

Windows:

Enter the following commands to unload and reload the driver respectively:

```
wmic path win32_networkadapter where manufacturer="Myricom" call disable
wmic path win32_networkadapter where manufacturer="Myricom" call enable
```

12.5. Single-Threaded TCP Applications

For single-threaded applications, set the environment variable `DBL_ST_MODE` to 1 for a faster TCP stack.

12.6. Zero Copy TCP Sends

DBL version 4.3 and later offers faster TCP send paths, drastically reducing tick-to-trade latencies. The mode is enabled by default. Disable the mode via `dblrun -o 0` or setting `DBL_TCP_SBL=0` to DBL TCP applications.

12.7. PIO/DMA Threshold

By default, packets less than or equal to 2048 bytes transfer through PIO for best performance. Packets larger than 2048 bytes transfer through DMA. You can adjust the threshold to a lower value. The firmware does not support a threshold greater than 2048 bytes.
Follow these steps to set the PIO/DMA threshold to a lower value through a module variable or registry key:

**Linux**

1. Edit the `myri_start_stop` script
   
   ```bash
   $ sudo vi /opt/dbl/sbin/myri_start_stop
   ```

2. Add the following lines:
   ```bash
   MYRI_MODULE_PARAMS=" myri_tx_threshold=7 $MYRI_MODULE_PARAMS"
   MYRI_MODULE_PARAMS=" myri_dbl_dmasend_threshold=512 $MYRI_MODULE_PARAMS"
   ```

3. Power-cycle your server (a reboot is not sufficient). This allows the FPGA to start using the new values.

4. Start the driver
   ```bash
   $ sudo /opt/dbl/sbin/myri_start_stop restart
   ```

**Windows**

1. Open a Powershell window as Administrator

2. Add the following keys to the registry:
   ```bash
   PS> REG ADD HKLM\SYSTEM\CurrentControlSet\services\dbl /v myri_tx_threshold /t REG_DWORD /d 7
   PS> REG ADD HKLM\SYSTEM\CurrentControlSet\services\dbl /v myri_dbl_dmasend_threshold /t REG_DWORD /d 512
   ```

3. Power-cycle your server (a restart is not sufficient). This allows the FPGA to start using the new values.
13. Troubleshooting

The cause of a defective network adapter can be easily assessed and resolved. When this situation arises, contact ARIA Technical Support to initiate a Return Merchandise Authorization (RMA) to obtain a replacement.

If the network appears to be functioning correctly but application transaction rates have not significantly improved after installing the ARC Series E network adapter and the DBLv5.5.0 software, it is possible that the proprietary software features are not properly enabled. Only those hosts with valid DBLv5.5.0 licenses will demonstrate accelerated performance. Repeat the software license activation process on all suspect host machines to certify proper licensing requirements for all network adapters.

This chapter contains the following topics:

- Hardware Installation and Performance Issues
- Software Installation and System Configuration

13.1. Hardware Installation and Performance Issues

The ARC Series E network adapter is a PCIe Gen3 x8 10-Gigabit Ethernet network adapter. For optimal performance, properly seat the adapter in an x8 PCIe Gen3 expansion slot on the server. The ARC Series E adapter auto-negotiates operation in the widest available mode (x8, x4, x2, or x1) supported by the expansion slot into which it is installed, and at the highest data rate (8, 5, or 2.5 gigatransfers per second GT/s).

For optimal performance, verify that the adapter reports Gen3 x8 (8 GT/s) PCIe link speed, once it is seated in the PCIe expansion slot on the server.

Two ways to determine if the network adapter is properly seated in a Gen3 PCI Express slot:
Linux

Check the output of `lspci -vvv`.

**Example:**

Sample `myri_info` output from an ARC Series E network adapter:

**Output:**

```
$ sudo ./myri_info -b 0
pci-dev at 01:00.0 vendor:product (rev)=1c09:4258(01)
    behind bridge root-port: 00:01.0 8086:0c01 (x8.3/x16.3)
Myri-10G-PCIE-8E -- Link x8
    EEPROM String-spec:
        MAC=00:60:dd:43:cd:40
        SN=491940
        PC=10G-PCIE3-8E-2S
        PN=09-04669
        BOM=A

Firmware:
    Version 2.0.6

External Inputs:
    PPS: Disabled
    10Mhz Clock: Disabled
    100Mhz Clock Locked: Locked
```

**pci-dev output results:**

- The ".3" notation refers to a PCIe 3.0 Gen3 slot.

- “behind the bridge root-port: 00:01.0 8086:0c01 (x8.3/x16.3)” indicates that the adapter is running at Gen3 x8 speed (maximum capability).

- The motherboard PCIe slot is x16-able.

- “Myri-10G-PCIE-8E -- Link x8” indicates that the ARC Series E network adapter is running optimally at x8 speed.

Windows

Check the `/opt/dbl/sbin/myri_info` directory file output.
13.1.1. Operating systems using the lspci command

For operating systems with the `lspci` command, examine the `lspci -vvv` output to check link speed (Lnk Sta).

Output:

```
LnkSta: Speed SGT/s, Width x8, TrErr- Train= SlotClk+ DLActive- BWMgmt- AABWMgmt-
```

13.2. Software Installation and System Configuration Issues

If you encounter problems with DBLv5.5.0 software installation, usage, or performance, send the bug report script to ARIA Technical Support. The script output contains the vital information we need to quickly resolve your software issues.

**NOTE**

It is very important that you retrieve the bug report script from the `/sbin/` directory, otherwise important diagnostic information will not be collected.

13.2.1. Bug report scripts

**Linux**

The `/opt/dbl/sbin/phx_bug_report` is a diagnostic script included in the Linux DBLv5.5.0 software distribution. The script collects diagnostic information about a customer’s system configuration, such as `uname` output, processor files such as `cpuinfo` and interrupts, `lspci`, kernel messages, `ethtool`, `myri_counters`, and so on. The script must be run as root.

**Windows**

The `phx_bug_report.ps1` is a Windows Powershell script in the Windows `phx-tools` distribution (Toolkit – Phoenix Group on the ARIA Customer Portal). The script extracts logging information from Windows logs and prints them to `stdout`. The script must be run as Administrator.

To execute the script, enter the following command:

```
Command line:
set-executionpolicy remotesigned
```

The output determines if the DBLv5.5.0 license is valid, which driver is loaded, and records any error information.
13.2.2. Linux RPM-TGZ installation failures

**Linux RPM**
If you encounter errors during the Linux RPM installation process, send the complete output from the RPM command, the kernel log output, and the `/tmp/myri_dbl.log` log to ARIA Technical Support.

**Linux TGZ**
If you encounter errors during the Linux TGZ installation, send the complete output from the `sbin/rebuild.sh` log to ARIA Technical Support.

| NOTE | To build a DBLv5.5.0 kernel module, configure the source kernel tree to match the running kernel.  
| RedHat example: You must install the `kernel-devel` package and the `kernel-headers` package from the RedHat distribution to build the DBLv5.5.0 kernel module. |

13.2.3. Windows MSI installation failure

To confirm that the MSI software is installed correctly, follow these steps:

1. Open the **Device Manager (Start->Run->devmgmt.msc)**.
   The text Myricom 10Gbps Ethernet NIC with DBL appears under **Device Manager > Network adapters**.
   If you encounter an install error, the **Device Manager > Network adapters** output reveals no DBLv5.5.0 devices or a yellow exclamation mark (!) next to the device. The presence of an exclamation mark indicates an error message.

2. Check the **EventViewer** for error messages, such as:
   "Insufficient number of MSI-X vectors..."
   The message indicates that the system does not have enough resources to install the driver, due to other NICs claiming additional resources.
3. To free up resources, decrease the number of endpoints by creating a MYRI_MAX_ENDPOINTS registry key and set it to 4, as follows:

   **Command line:**
   
   `REG ADD HKLM\SYSTEM\CurrentControlSet\services\dbl /myri_max_endpoints /t REG_DWORD /d 4`

   The keyword "dbl" indicates that you are running the DBLv5.5.0 driver.

4. If the Windows MSI installation fails while running the installer, send the following log by adding `/log <filename>` to the MSI install, as follows:

   **Command line:**
   
   `c:\> dbl-<version_info>.PHX_x64_wlh.msi /log dbl_install_log.txt`

   OR

   If a Windows MSI installation fails with following message:

   "There is a problem with this Windows Installer Package. A program run as part of setup did not finish as expected."

   Verify that an earlier removal has unlocked all processes from DBLv5.5.0-related files. Run the following command from a command prompt to determine if DBLv5.5.0 is still in use:

   **Command line:**
   
   `c:\> tasklist /m dbl.dll`

   The output may show processes still using DBLv5.5.0-related software.

   **Output:**

<table>
<thead>
<tr>
<th>Image Name</th>
<th>PID</th>
<th>Modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>svchost.exe</td>
<td>8948</td>
<td>dbl.dll</td>
</tr>
</tbody>
</table>

5. Reboot Windows.

   Alternatively, you can install Windows DBLv5.5.0 software using Windows msiexec as described in *Installing DBLv5.5.0 Software.*
13.2.4. Windows teaming/VLAN team_config issues

You may need to uninstall and reinstall the VLAN/Team driver if you are experiencing the following team_config.exe utility issues:

- The utility reports that the ARC Series E network adapters are “not bound”, OR
- The utility does not detect the network adapters.

NOTE
The steps and screenshots described in this section may differ from your Windows environment. Contact your system administrator if you need assistance.

Uninstalling and reinstalling the teaming driver

To uninstall and reinstall the teaming driver, follow these steps:

**Manual Uninstall:**

1. Click Control Panel > Network and Internet > Network Connections to verify the network status of the ARC Series E adapter.

   The Network Connections window appears.

2. Select the network adapter from the Network Connections list. In this example the network adapter is called “Ethernet 3”.

   The steps and screenshots described in this section may differ from your Windows environment. Contact your system administrator if you need assistance.
The Ethernet 3 Status prompt appears.
3. Click **Properties** on the Ethernet 3 Status prompt.

   ![Image of Ethernet 3 Properties prompt]

   The Ethernet 3 Properties prompt appears.

4. Click **Uninstall** on the Ethernet 3 Properties prompt.
A confirmation prompt to uninstall the team driver appears.

5. Click Yes.
The team driver uninstalls.

The Select Network Protocol prompt appears. You are now ready to reinstall the team driver (protocol).
Manual Reinstall:

1. Click **Teaming/VLAN Driver...** from the Select Network Protocol prompt.
2. Click **OK**.

The Select Network Feature Type prompt appears.
3. Select **Protocol** and click **Add…**

   ![Windows Network Connections](image)

   The **Install From Disk** prompt appears.

   You are now ready to reinstall the DBLv5.5.0 team driver from the **C:\** drive.

4. Select, copy, or enter the team driver directory file. In this example, the directory file is **C:\DBL_PHX-10G\team**.

5. Click **OK**.

   The VLAN/Team driver is now reinstalled.
### 13.2.5. Bonding for failover in TA mode

#### Linux

DBLv5.5.0 provides limited bonding support. In TA mode, it is only used for failover and redundancy. Under Linux, this feature requires that you configure and run the network bonding driver with the Ethernet devices exposed by DBLv5.5.0. For more details, go to:


https://www.kernel.org/doc/Documentation/networking/bonding.txt

#### Windows

Under Windows, this feature requires that you configure the supplied teaming driver for a failover team.

When it detects that the Ethernet device is part of a failover bond, TA mode executes a `dbl_open` on each underlying device, at which point the native bonding driver detects that a link is down and selects a new active slave. Once TA mode detects that the active slave has changed, it switches to the new active slave for its operation.

#### TA failure

If the sockets application fails to run in DBLv5.5.0 Transparent Acceleration (TA) mode, follow the troubleshooting steps described below to identify the source of the TA failure.

Report the error to ARIA Technical Support using the `-f` option in `dblrun` to generate the error log, as follows:

**Command line;**

```bash
$ dblrun -f logfile myapp.exe
```

**NOTE**

Some error logfiles are very large and cumbersome. Reduce the size of the logfile by reproducing the error with the smallest number of connections (feeds) and a minimum number of variables that still show the error. Send the smaller error logfile

There are two ways to verify that DBLv5.5.0 TA mode is being activated when running the application:
Option 1:

Set the environment variable DBL_IP_VERBOSE to 1 when executing the `dblrun` command as follows:

Command line:

```bash
$ DBL_IP_VERBOSE=1 dblrun [options] executable arguments
```

A diagnostic output prints to the console.

Option 2:

Examine the output of `dblrun -v <appname>`.

13.2.6. Software Counters

The `myri_counters` tool provides low-level DBLv5.0 hardware and software counters for traffic that passes through the network adapter in DBLv5.0 mode.

Linux

Command line:

```bash
$ /opt/dbl/bin/myri_counters
```

Windows

Command line:

```bash
$ [INSTALL_DIR]\bin\myri_counters.exe
```

By default, the `myri_counters` output only displays on port/board 0. Dual-port adapters appear to `myri_counters` as different ports/boards. If you have a dual-port network adapter installed in the host, you must specify the command line argument `-p <port_num>` to obtain the counters output for each port.

The environment variable `port_num` has an integer value from 0 to `n-1`, where “n” represents the number of network adapters installed in the host and running the DBLv5.0 driver, as follows:

Command lines:

```bash
$ /opt/dbl/bin/myri_counters -p 0
$ /opt/dbl/bin/myri_counters -p 1
```

The space between the “p” and the number is optional.

If a host has two dual-port adapters, assign `-p0` and `-p1` to the ports of one network adapter and `-p2` and `-p3` to the ports of the second adapter. To clear and reset the counters requires root privileges.
To clear the counters on a specific port of a network adapter, enter the following:

**Command line:**

```
$ sudo /opt/dbl/bin/myri_counters -p <port_num> -c
```

For a detailed list of command line arguments to `myri_counters`, go to the DBLv5.5.0 Diagnostic Tool Programs section in Testing DBLv5.5.0 Software

### 13.2.7. Environment Variables

DBLv5.5.0 software offers a variety of environment variables for debugging, timestamping, and for customizing the software configuration to meet application requirements.

**DBL_IP_VERBOSE**

Setting `DBL_IP_VERBOSE` to 1 prints additional diagnostic output to the console, as follows:

**Command line:**

```
$ DBL_IP_VERBOSE=1 ./test/dbl_pingpong -s -l 10.0.0.1
```

**OR**

**Command line:**

```
$ DBL_IP_VERBOSE=1 dblrun [options] executable arguments
```

**DBL_CONFIG**

The environment variable `DBL_CONFIG`, steers various MTCP variables in DBLv5.5.0 TCP API. These variables include, but are not limited to, congestion modules and delayed ACK.

For more information, refer to the Customizing Transparent Acceleration (TA) and TCP API mode using dbl.conf section in Transparent Sockets Acceleration Software

**DBL_ST_MODE**

For single-threaded applications, set the environment variable `DBL_ST_MODE` to 1 for a faster TCP stack for DBLv5.5.0 TCP extensions.

For more information on timestamping environment variables, go to Installing DBLv5.5.0 software
13.2.8. Functional failures

Some sockets-based applications may experience a functional failure because DBLv5.5.0 behavior differs from the regular OS stack.

Linux assigns a substantial portion of the transparent sockets service to BSD-style Linux sockets emulation. In the event a functional failure occurs, enter the following command to save tracing output:

**Command line:**

```bash
$ dblrun -f output <application_name>
```

Send the output file to ARIA Technical Support.

13.2.9. Performance Issues

Refer to the *Tuning* chapter first to familiarize yourself with improving DBLv5.5.0 performance. If these suggestions do not address the issue, follow these steps:

1. Verify that the Transparent Socket Acceleration (DBLv5.5.0 TA mode) is effectively loaded.
   
   Check for the following information after running DBLv5.5.0:
   ```
   <process_name>[<pid>] DBL 5.5.0.0.<build_id> Copyright 2020 CSPi
   ```

2. Verify that the network adapter is installed in a PCIe expansion slot that can sustain 10-Gigabits per second.

3. Monitor *myri counters* for accelerated traffic. Check that the network is not experiencing “receive data buffering” overflow conditions.

4. Check the network adapter run-time counter values as described below:
   a. Examine **Packets Received (all ports)** and **Packets RX (this port)** counter values to verify that traffic is passing directly to a userspace stack.
   b. Examine **Packets Drop Filter, Packets Drop Filter (HW), Packets Rx Filter, Packets Drop Incoming**, and **PCIE FIFO * counter values for traffic that the network adapter and stack cannot sustain. The counter values are zero in normal operation.**

   For a non-zero counter value, go to the *Increasing the Size of the Data and Descriptor Rings* section of *Tuning*
By default, the myri_counters output only displays on port/board 0. Dual-port adapters appear to myri_counters as different ports/boards. If you have a dual-port network adapter installed in the host, you must specify the command line argument \(-p <port_num>\) to obtain the counters output for each port. The environment variable PORT_NUM has an integer value from 0 to \(n-1\), where \(n\) represents the number of network adapters installed in the host and running the DBLv5.5.0 drive.

For more information on DBLv5.5.0 API mode and DBLv5.5.0 TA mode performance, go to Testing DBLv5.5.0 Software

### 13.2.10. Network Adapter Timestamps

DBLv5.5.0 supports socket timestamps on ARC Series E network adapters.

The timestamp is made available through the socket interface. The socket interface, running the SO_TIMESTAMP[NS] socket option, is a high-precision clock synchronized at startup to system host time (returned by the gettimeofday() function call). The timestamp automatically attaches to the packet when it arrives at the network adapter.

Linux users can access the clock as a POSIX clock or through the PTP protocol.

| CAUTION | Use caution when running NTP services (that rely on network time protocols) on the POSIX clock. NTP services cause variations in system host times that may exceed inter-packet arrival times. |
| NOTE | While you can inject timestamps into Windows packets, there is no interface to adjust the clock after it is initially set by loading the driver. |
13.2.11. Synchronization

There are different levels of timestamp synchronization.

- Network Adapter-to-host synchronization (local sync)
- Host-to-host synchronization (global sync)

Network Adapter-to-host synchronization (local sync)

Use the `phc2sys` tool in Linux to run Network Adapter-to-host synchronization. Set the ARC Series E network adapter to “Master” clock, for higher resolution and accuracy (recommended).

Host-to-host synchronization (global sync)

Most systems use `NTPD` to run host-to-host synchronization (global sync). More precise options to `NTPD` exist; however, it only raises the accuracy level from milliseconds to tens of microseconds. Since both protocols are not typically run on dedicated networks and because there is typically a lot of host overhead in processing the protocol, the time can only be so accurate.

The ARC Series E network adapters connect externally to a PPS or a 10MHz input to improve adapter clock accuracy. When loading the driver, set the module variable inputs as follows:

```
"myri_clk_enable_pps=1"
"myri_clk_enable_10mhz=1"
```
Appendix 1. DBLv5.5.0 Counters

To download DBLv5.5.0 hardware and software counter values from the time the driver was loaded (or since the counters were reset through `myri_counters -c`), use the `myri_counters` tool.

**Linux**

```bash
$ /opt/dbl/bin/myri_counters
```

**Windows**

```bash
$ [INSTALL_DIR]\bin\myri_counters.exe
```

By default, the `myri_counters` output is only displayed for port 0. Dual-port adapters appear to `myri_counters` as different ports. If you have a two-port network adapter installed in the host, you must specify the command line argument `-p <port_num>` to obtain the counters output for each port.

**Example:**

```bash
$ /opt/dbl/bin/myri_counters -p 0
$ /opt/dbl/bin/myri_counters -p 1
```

The space between the "p" and the number is optional. If a host contains two dual-port adapters, assign `-p0` and `-p1` to the ports of the first adapter and `-p2` and `-p3` to the ports of the second adapter.

**Command line option [Help]:**

```bash
$ /opt/dbl/bin/myri_counters -h
```
**Usage:**

```
Usage: myri_counters [args]
-p N  - Port number or Ethernet MAC
-b N  - Port number or Ethernet MAC (by default all ports are shown)
-c    - clear the counters
-q    - quiet: show only nonzero counters
-i    - show host interrupt counters
-x    - expert: show all counters
-o    - show register offset
-r    - raw: show register contents
-e N  - show counters for specified endpoint [0]
-a    - show counters for all endpoints
-v    - show all counters
-F    - show filter state including all registered filters
-M    - show MAC filters, if available
-h    - help
```

To clear the counters on a specific port of a network adapter, enter the following command:

```
sudo $ /opt/dbl/bin/myri_counters -p <port_num> -c
```

**NOTE:** All myri_counters are 64-bit counters. The majority of the counters are for developer use only and are not described in detail.

**Linux Help**

If there are additional questions with Linux DBLv5.5.0, run the `/opt/dbl/bin/phx_bug_report` script and send the output to ARIA Technical Support.

**Windows Help**

If there are additional questions with Windows DBLv5.5.0, run the Windows `phx_bug_report.ps1` Powershell script (available in `phx-tools`) and send the output to ARIA Technical Support.
**DBLv5.5.0 Counters**

**Driver Uptime**
Displays the time and date that the DBL driver was started and the number of elapsed seconds.

**Counters Uptime**
Displays the time and date that the counters were last cleared and the number of elapsed seconds.

**PIO CP Read Dword**
Developer use only.

**PIO CP Read Header**
Developer use only.

**PIO CP Write Dword**
Developer use only.

**PIO CP Write Header**
Developer use only.

**LTSSM Not L0 state**
PCIe link training and status register. It shows the number of cycles that the PCI Express link is attempting to retrain. The default value is zero. A positive value for LTSSM Not L0 state may indicate that there are signal integrity issues across the PCIe interface causing link training to be re-attempted.

**EP Packets Received**
The number of packets received by this endpoint.

**EP Bytes Received**
The number of bytes received by this endpoint.

**EP TX PIO Packets**
Packets sent by PIO to the adapter for transmit by this endpoint.
EP TX PIO Bytes
Bytes sent by PIO to the adapter for transmit by this endpoint.

EP TX DMA Packets
Packets sent by DMA to the adapter for transmit by this endpoint.

EP TX DMA Bytes
Bytes sent by DMA to the adapter for transmit by this endpoint.

EP Host RX Packets
Software counter of received packets by this endpoint.

EP Host RX Packets with MAC Errors
Software counter of received packets with the MAC error flag set.

EP DBL Drop Bad Protocol
Developer use only.

EP DBL Drop Bad Filter
Developer use only.

EP DBL Drop No Socket
Developer use only.

EP DBL Drop Runt
Developer use only.

EP DBL Drop Broadcast Filtered
Developer use only.

EP TX Completions
Developer use only.
EP Data Pages
DMA pages allocated to the receive data ring. The endpoint zero default is 1024. Other endpoints are zero when not in use. The default is 32768 if other endpoints are in use by an application. For more information refer to the *Increasing the Size of the Data and Descriptor Rings* section in the *Tuning* chapter.

EP Desc Pages
DMA pages allocated to a receive descriptor ring. The endpoint zero default is 512. Other endpoints are zero when not in use. The default is 16384 if other endpoints are in use by an application. For more information refer to the *Increasing the Size of the Data and Descriptor Rings* section in the *Tuning* chapter.

EP Data Consumer Position
Flow control counter for receive data ring.

EP Data Producer Position
Flow control counter for receive data ring.

EP Desc Consumer Position
Flow control counter for receive descriptor ring.

EP Desc Producer Position
Flow control counter for receive descriptor ring.

Cycle Count
The FPGA cycle count from the time the DBLv5.5.0 driver was loaded running at 167 MHz (6.0ns/cycle).

Packets Received
The number of packets received at DBL Rx and pushed to an endpoint.

Bytes Received
The number of bytes received in packets at DBL Rx and pushed to an endpoint.

Filter Packets Rx
The number of packets received and pushed beyond the filter core.
Filter Bytes Rx
The number of bytes in packets received and pushed beyond the filter core.

Packets Drop Filter
The number of packets dropped by the filter core due to congestion.

MAC Receive Errors
The number of packets received but discarded due to MAC errors. The errors may be due to receiving a runt packet (less than 60 bytes, not including FCS), receiving a packet greater than Jumbo size (9018 Bytes), or receiving an Ethernet frame with an FCS that does not match the FCS calculated by the MAC.

To view detailed MAC statistics, run the following command from the phx-tools package:

Command line:
$ sudo ./myri_phx_mdio -m

Packets Filtered
The number of packets processed by the filter core.

Packets Unfiltered
The number of packets not filtered since they are not IP packets. A non-zero number is not an error.

Max Filter Latency
The maximum latency for any packet, running at 167 MHz (6.0ns/cycle).

Packets Skip Filter (HW)
The number of packets skipped by the filter core due to congestion. These packets are sent to the kernel endpoint.

Packets Drop Filter (HW)
The number of packets dropped by the filter core due to complete filter backup.
Packets Rx Filter
The sum of all packets dropped between the filter core and the endpoint receiving cores.

Packets Drop Incoming
The number of packets dropped on the incoming side of the filter core.

Intentional Discard Pkts
The number of packets dropped because they do not match a filter.

Multicast Packets Rx
The number of multicast packets received.

Multicast Pkts Discarded
The number of multicast packets dropped because they do not match a filter.

FF Packets Received
The number of packets finalized and pushed to the host. (32-bit)

FF Bytes Received
The count of bytes pushed to the host (32-bit)

FD Packets Discarded
The number of packets discarded because of host descriptors full.

FD Bytes Discarded
The number of bytes discarded because of host descriptors full.

PCIE FIFO empty
The number of clock cycles where Rx PCIe FIFO was empty.

PCIE FIFO < ¼
The number of clock cycles where Rx PCIe FIFO was less than one quarter full.
PCIE FIFO $\frac{1}{4} - \frac{1}{2}$
The number of clock cycles where Rx PCIe FIFO was between one quarter and one-half full.

PCIE FIFO $\frac{1}{2} - \frac{3}{4}$
The number of clock cycles for which Rx PCIe FIFO was between one-half and three-quarters full.

PCIE FIFO $> \frac{3}{4}$
The number of clock cycles for which Rx PCIe FIFO was between three-quarters and full.

PCIE FIFO full
The number of clock cycles for which Rx PCIe FIFO was full. These are cycles for which the PCIe is blocking Rx. Packets may have been dropped if the value is non-zero.

Filter Kernel Endpoint
Developer use only.

Filter Lookup Count
Developer use only.

Filter Lookup Fail
Developer use only.

Filter Lookup Search
Developer use only.

Filter Add Error
Developer use only.

Filter Remove Error
Developer use only.
Replay Buffer Packets Received
Number of replay buffer packets received.

Replay Buffer Packets Sent
Number of replay buffer packets sent.

Replay Buffer Bytes Sent
Number of replay buffer bytes sent.

Replay Buffer Packets Dropped
Number of replay buffer packets dropped.

Replay Buffer Sent Duplicate
Number of replay buffer packets sent multiple times due to filter configuration.

Replay Buffer Sent Abort
Number of replay buffer packets sent with the error flag set. These are usually dropped but may pass through due to cut-through.

Replay Buffer Dropped Abort
Number of replay buffer packets dropped due to abort flag set. Included in “Replay Buffer Packets Dropped”.

Replay Buffer Dropped Control
Number of replay buffer packets dropped due to control data full. Included in “Replay Buffer Packets Dropped”.

Replay Buffer Dropped Meta Data
Number of replay buffer packets dropped due to metadata full. Included in “Replay Buffer Packets Dropped”.

Replay Buffer Control Full
Indicates replay buffer control packets dropped due to full condition.
Replay Buffer Receive Abort
Indicates replay buffer packets received with abort flag.

Replay Buffer Meta Error
Indicates replay buffer meta receive error.

Filter Add Slots
Slot handles available for adding filters. Should always be non-zero.

Filter Used Slots
Number of filter slots used.

Filter Free Slots
Number of filter slots available.

RX Packet Count Limit
Number of received packets to coalesce before posting a descriptor.

RX Length Limit
Number of received bytes to coalesce before posting a descriptor.

RX Time Limit
Number of cycle counts to wait after receiving the first packet before posting a descriptor.

Packet Overflow
When an endpoint is not consuming packets as fast as the adapter is sending them. In normal operation, this should not occur. It may occur if the host is heavily loaded.

Descriptor Overflow
When an endpoint is not consuming descriptors as fast as the adapter is sending them. In normal operation, this should not occur. It may occur if the host is heavily loaded.

TX Pause MAC CTRL frames
Flow control frames sent by the MAC.
RX Pause MAC CTRL frames
Flow control frames received by the MAC.

APU version
Version number of the Address Prefetch Unit in the FPGA.

APU configuration
Developer use only.

Rx Sel Packets Rx
Developer use only.

Rx Sel Packets Dropped
Developer use only.

Rx Sel Packets Aborted
Developer use only.

Rx Sel Packets Ok
Developer use only.

Rx Sel Packets No Meta
Developer use only.

Rx Sel Packets Meta Dropped
Developer use only.

Rx Sel Packets Extra Meta
Developer use only.

Rx Sel Packets Ctrl Dropped
Developer use only.
**Rx Sel Bytes OK**
Developer use only.

**Rx Sel Packets Bypassed**
Developer use only.

**Rx Sel Bytes Bypassed**
Developer use only.

**Rx Sel Bypass Disable**
Developer use only.

**MAC Filter Packets Dropped**
Number of packets dropped by the MAC filter.

**MAC Filter Bytes Dropped**
Number of bytes dropped by the MAC filter.

**MAC Filter Packets FWD**
Developer use only.

**MAC Filter Bytes FWD**
Developer use only.

**IRQ Endpoints**
Number of endpoint interrupts.

**IRQ Vectors Allocated**
Number of interrupt vectors used by the card total.

**IRQ Operation Mode**
Interrupt mode.
**IRQ Global Mask**
Developer use only.

**IRQ Enable Lo**
Developer use only.

**IRQ Enable Mid**
Developer use only.

**IRQ Enable Hi**
Developer use only.

**EP Filter Packets Sent**
The number of packets sent to this endpoint by the shared filter core.

**EP Filter Bytes Sent**
The number of bytes sent to any endpoint by the shared filter core.
Appendix 2. Operating Systems and Hardware Support

Software Support

Linux support

- CentOS 8.0 is recommended. Centos 7.7 is supported.

- Some testing has also been performed with RHEL 6.

- For non-RPM-based Linux distributions, a TGZ driver is provided and supported up to Linux kernel version 5.5.

Windows support

- Windows Server 2012R2, 2016, and 2019 in 64-bit mode are supported. Windows Server 2019 is recommended.

- Windows 10 is supported. For best performance, Windows Server versions are recommended.

Hardware Support

Network Adapters:

ARC Series E (10G-PCIE3-8E-2S) network adapter

Part number: 09-04669

- SYNC capabilities with hardware timestamping are not yet available. The SMA and MMCX connectors should not be attached to cables. This feature will be supported in an upcoming release.
Processors:

- Intel Haswell-E processors are recommended.
- Testing has been performed on i7-4790K, i7-5960X, and E5-3699 processors.

Motherboards

- The following motherboards are compatible:
  - Asus Z97-Deluxe
  - Asus X99-WS/IPMI
  - Dell PowerEdgeR630
  - HP 2440 motherboards.
- Other HP Servers that are are not recommended:
  - HP Z840 Workstation
  - HP G& ProLiant servers
  - HP ProLiant Gen9 servers
Appendix 3. DBLv5.5.0 Driver Restrictions and Limitations

- There is no Makefile available to build test programs in Windows.
- We recommend connecting both Rx and Tx to the 10G-PCIE3-8E-2S adapter port. Disconnecting Tx may cause the adapter to hang.
- `dbltcp_tick_to_trade` will seg fault in `device_under_test_mode` if fewer than 10,000 iterations are specified using the `-i` option.
- When running `dbltcp_tick_to_trade` and `sock_tick_to_trade` benchmarks together you may observe a large amount of jitter in the results. This has only been observed in device-under-test mode when started via `dblrn -b 0`.
- Running `sock_tick_to_trade` may display false results when started via `dblrn -b 0`.
- When switching transceiver or cable types, the link may not be detected without first reloading the driver.
- The maximum packet size for the `dbl_pingpong` program is 65535. Specifying a larger size with the `-E` option will cause the adapter to hang.
- Heavy traffic keyframe drought may occur. In a situation where keyframes are gapping and the reference clock is not updated, UTC timestamps have been observed going backwards.
**Example:**

<table>
<thead>
<tr>
<th>Phone Number</th>
<th>1476441837373474457</th>
</tr>
</thead>
<tbody>
<tr>
<td>2119428777</td>
<td>1476441837373474614</td>
</tr>
<tr>
<td>2119428873</td>
<td>1476441837373474731</td>
</tr>
<tr>
<td>2119428929</td>
<td>1476441837373474891</td>
</tr>
<tr>
<td>2119428978</td>
<td>1476441837373475031</td>
</tr>
<tr>
<td><strong>2119429033</strong></td>
<td><strong>1476441837373475188</strong></td>
</tr>
<tr>
<td>1770277926</td>
<td>1476441836375900300</td>
</tr>
<tr>
<td>2119429084</td>
<td>1476441837373473503</td>
</tr>
<tr>
<td>2119429133</td>
<td>1476441837373473643</td>
</tr>
<tr>
<td>2119429184</td>
<td>1476441837373473789</td>
</tr>
<tr>
<td>2119429234</td>
<td>1476441837373473931</td>
</tr>
<tr>
<td>2119429283</td>
<td>1476441837373474071</td>
</tr>
<tr>
<td>2119429338</td>
<td>1476441837373474229</td>
</tr>
<tr>
<td>2119429382</td>
<td>1476441837373474354</td>
</tr>
<tr>
<td>2119429441</td>
<td>1476441837373474523</td>
</tr>
<tr>
<td>2119429489</td>
<td>1476441837373474660</td>
</tr>
<tr>
<td>2119429538</td>
<td>1476441837373474800</td>
</tr>
<tr>
<td>2119429590</td>
<td>1476441837373474949</td>
</tr>
<tr>
<td>2119429637</td>
<td>1476441837373475083</td>
</tr>
<tr>
<td>2119429693</td>
<td>1476441837373475243</td>
</tr>
</tbody>
</table>
Appendix 4. DBL RAW Mode (Beta)

The DBL RAW mode is an extension of DBL that allows for receiving and sending RAW IP frames.

NOTE: Raw Mode does not support VLAN

Initialization

From a programming perspective, you need to run `dbl_open` by opening a DBL endpoint to access the device in a kernel-bypass mode. To allow for RAW mode functionality, set the flag as follows:

```
flags |= DBL_OPEN_RAW_MODE;
```

User space memory is registered with the device, where hardware automatically transfers the data.

```
rc = dbl_open (struct in_addr *, int flags, &test_dev);
```

Configuration

Prior to receiving, the `dbl_set_filter()` needs to be called after having opened an endpoint in RAW mode.

```
rc = dbl_set_filter(dbl_device_t *dev, enum dbl_filtermode, int ip_proto, struct sockaddr_in *, int flags);
```

After placing the filter, the traffic is placed directly into user memory.
**Receiving**

For receiving RAW packets, you can choose between `dbl_raw_recv()` or `dbl_recvfrom()` function calls.

**Command lines:**

```c
rc = dbl_raw_recv(dbl_device_t, enum recv_mode, char *buf, int *len, int *timestamp);

rc = dbl_recvfrom(dbl_device_t, enum rmode, char *buf, int len, dbl_struct_info *);
```

The buffer message IP headers is followed by the payload. The message length returned contains the “sum of the headers and the payload”.

**Sending**

When sending on a RAW endpoint, use `dbl_send_raw` on a DBL endpoint opened for RAW mode. The function then expects the passed buffer to contain a fully qualified frame to be put out on the wire.

**Command line:**

```c
int dbl_raw_send(dbl_device_t dep, uint8_t *hdr, int hdrlen,
                 uint32_t chksum_hdr,   /* header checksum so far */
                 int chksum_offset,    /* offset of where checksum field resides */
                 const uint8_t *buffer, /* payload */
                 int paylen,            /* payload len */
                 int flags);            /* flags, e.g request for TX timestamp */
```
Special Considerations

- The device maintains regular Ethernet functionality. There are no known restrictions.

- Designate a filter for each connection you want to receive on.

- The total number of filters per port is 512.

- Run `dbl_set_filter` and `dbl_delete_filter` for routine filter cleanup and release.

- Some lightweight DBL functions are available, such as the “create IP/UDP headers” function.
Appendix 5. Multicast Whitelist

DBLv5.5.0 provides a Multicast Whitelist (IP Class D Discard) capability to limit multicast traffic to the host side of the adapter through the kernel endpoint.

**NOTE:** Users should enable the Whitelist feature if significant latency jitter is experienced, as it may indicate that the CPU is being over-utilized by handling unwanted multicast traffic through the kernel.

**Theory of Operation**

The ARC series E network adapter receives all packets into the kernel that do not match DBL filters set up by the user application. Packets that match the DBL filters bypass the kernel and are stored in user space. The kernel network stack addresses all other multicast packets. Without IGMP snooping switches, a DBL system may receive lots of unwanted multicast packets, which may result in significant DBL application jitter. This jitter is caused by the system’s need to consume CPU cycles as the kernel network stack processes the sockets listening for those multicast packets and subsequently drop them.

The ARC series E network adapters have 'class D discard' or 'Whitelist' feature to remedy this problem and remove the jitter in the system. When the feature is enabled, IPv4 multicast packets are discarded unless they match DBL filters (belong to the DBL application). This breaks non-DBL sockets applications and the kernel that also use IPv4 multicast, because they will no longer receive any packets. Hence, we currently use a whitelisting technique that discards unwanted multicast traffic before the kernel needs to process it. To that end, CPU utilization and jitter are reduced.

**Using the Whitelist Feature**

To use the whitelist feature, you create a list of IPv4 multicast addresses (whitelist) that should be accepted into the kernel. From a programming perspective, you need to invoke `dbl_config` to install the whitelist into the driver. The driver adds DBL filters for the specified multicast addresses so corresponding packets are received into the kernel.

Standard kernel based network adapters, do not have this issue as they implement industry standard multicasting filtering at L2. L2 filtering is not yet implemented in ARC series network adapters.
NOTE: By default, whitelisting is disabled on all ports.

Linux

Definition:
dbl_config

Command line:
$ sudo .\dbl_config.exe

Usage:
dbl_config [options]

Usage:
(dbl_config has the same usage on both Linux and Windows)

- c  Clear the whitelist of multicast addresses
- e mask  Enable discard feature on the specified ports. 0 means off.
  Bit n in the mask represents port n. Set bit n enables discard
  feature on port n. **Discard must be enabled before adding white
  list**
- q  Show configured whitelisted mcast addresses
- v  Verbose mode
- f file  Add whitelisted mcast addresses, erases old addresses
  This does not alter the multicast discard bitmask setting
  File format: <adapter IP address> <IPv4 multicast address>
  Example:
  10.0.0.1 238.3.7.10
  10.0.0.1 238.3.7.11

Windows

Definition:
dbl_config.exe

Command line:
PS C:\DBL_PHX-10G\sbin> .\dbl_config.exe
Usage:

dbl_config.exe [options]

Usage:
(dbl_config has the same usage on both Linux and Windows)

- `c` Clear the whitelist of multicast addresses
- `e mask` Enable discard feature on the specified ports. 0 means off.
  
  Bit n in the mask represents port n. Set bit n enables discard feature on port n. **Discard must be enabled before adding white list**
- `q` Show configured whitelisted mcast addresses
- `v` Verbose mode
- `f file` Add whitelisted mcast addresses, erases old addresses
  
  This does not alter the multicast discard bitmask setting
  
  File format: <adapter IP address> <IPv4 multicast address>
  
  Example:
  
  10.0.0.1 238.3.7.10  
  10.0.0.1 238.3.7.11

Using the whitelist feature:

To set up `dbl_config` for white listing, follow these steps:

1. Load the driver with or without
   `myri_enable_ipv4_multicast_discard_bitmask`

2. Configure the DBLv5.5.0 interfaces to assign individual IP addresses

3. Enter the following command:
   
   `dbl_config -e 0xf`
   
   This enables discard on the ports specified in `-e`. So, bit N of the `<mask>` value represents port N. A set bit of 1 enables discard on that port. When bit N is set, the driver enables class D discard on board (port) N.
   
   For example, 0x3 enables class D discard on board 0 and 1. `0xf` enables discard on ports 0, 1, 2, and 3.

4. Enter the following command:
   
   `dbl_config -f whitelist.txt`
This adds the whitelist to the driver and adds filters to ports that have 
discard enabled. If a port has been discard disabled, the whitelist has no 
effect.

The whitelist is global (one list per driver). Each line in the whitelist file has
"<interface IPv4 address> <IPv4 multicast address>". The driver adds 
a filter for the given multicast address on the port that has the given 
interface address. A line may be empty or start with '#' (comment line).

Example:
16.0.0.1 238.3.7.10
16.0.0.1 238.3.7.11
17.0.0.1 238.3.7.12
17.0.0.1 238.3.7.13

Port IP: 16.0.0.1 - the driver removes the previous whitelist filters and then 
adds two filters for 238.3.7.10, 238.3.7.11. Their destination endpoint is the 
port’s kernel receive endpoint.

Port IP: 17.0.0.1 - the driver again removes the previous whitelist filters 
and then adds two filters for 238.3.7.12 and 238.3.7.13.

NOTE: You should always check dmesg or Windows event logs to verify the 
whitelist is correctly applied to the intended ports.

The whitelist may contain at most 128 entries.
Whitelist Guidance

At a minimum, each port should whitelist 224.0.0.1 (all systems on the subnet). Typical multicast groups include the following:

- 224.0.0.2 (all routers on the subnet)
- 224.0.1.1 (NTP)
- 224.0.1.107,129-132 (PTP)

Compiling

Compile your list for whitelisting as follows:

1. Run the applications with class D discard disabled. All applications should function correctly.
2. Get the list of joined group addresses:
   On Linux, `netstat -gn`
   On Windows, `netsh interface ip show joins`.
3. From the list, remove only those used by DBL applications.
   This list becomes the whitelist.
Appendix 6: DBLv5.5.0 Supported 10G Transceivers

The following 10G transceivers have been tested and are supported to run with ARC Series E (10G-PCIE3-8E-2S) network adapters.

Supported 10G transceiver modules

<table>
<thead>
<tr>
<th>CSPi Model Number</th>
<th>Description</th>
<th>DBL Software Release verified in:</th>
</tr>
</thead>
<tbody>
<tr>
<td>10G-SFP-LR</td>
<td>Optical-Fiber SFP+ transceiver for 10GBase-LR (1310nm wavelength)</td>
<td>DBL Release Versions 2, 3, and 5.</td>
</tr>
<tr>
<td>10G-XFP-SR</td>
<td>Optical-Fiber XFP transceiver for 10GBase-SR (850nm wavelength)</td>
<td>DBL Release Versions 2, 3, and 5.</td>
</tr>
<tr>
<td>10G-XFP-LR</td>
<td>Optical-Fiber XFP transceiver for 10GBase-LR (1310nm wavelength)</td>
<td>DBL Release Versions 2, 3, and 5.</td>
</tr>
</tbody>
</table>

Table 5: Supported 10G transceiver modules.