Considerations for Deploying Ethernet Adapters Provided by Emulex in FreeBSD Systems Using the Emulex Drivers

Emulex provides NIC (Ethernet) driver support for FreeBSD version 8.1 and above. The Ethernet NIC adapters provided by Emulex, support protocol offloads for TCP/IP for maximum bandwidth coupled with low CPU utilization in 10GbE networks. To get the best performance from 10GbE networks, some care should be taken in configuring the cards, drivers and OS parameters.

Introduction

This application note describes some of the important considerations for deploying 10GbE Ethernet adapters provided by Emulex in FreeBSD systems using the Emulex driver. Netperf is referenced as a performance management tool to establish the baseline capabilities of NICs as a foundation for further analysis of performance in a broader network. Analysis beyond the base testing of a simple back-to-back network is beyond the scope of this paper.

Hardware

The Ethernet adapters are single- or dual-port PCI Express Gen2 cards, so it is important to ensure they are physically placed in appropriate PCIe slots in the server. Single-port ards are electrically PCIe x4 with PCIe x8 connectors. They need to be placed in a slot that has at least PCIe x8 as the physical socket with PCIe x4 connections. Dual-port cards are also PCIe x8 physically, but need PCIe x8 electrical capability for optimum performance.

Note: Some systems have PCIe slots which are electrically wired with fewer lines than the slot can allow.

Notes on Emulex FreeBSD driver for NICs provided by Emulex

Driver version used here = 4.2.313.0 on FreeBSD 9.1 AMD64. Firmware = 4.4.249.3

Note: As of 28th May 2013, the public download page for the FreeBSD driver for 9.0 links to the 32-bit version of the driver only. The link is http://www-dl.emulex.com/support/elx/r32/b24/FreeBSD/oce-4.2.313.0-freebsd90-i386.tbz. However, to get the 64-bit version, substitute "i386" in the link with "amd64." Therefore, http://www-dl.emulex.com/support/elx/r32/b24/FreeBSD/oce-4.2.313.0-freebsd90-amd64.tbz should provide the 64-bit driver.

Driver installation is straightforward and well documented in the driver user guide:

1. Download the appropriate driver kit from the Emulex website.
2. Log on as “root” and type
   `pkg_add oce-<VERSION>-<ARCH>.tbz`
   For example:
   `pkg_add oce-4.2.313.0-amd64.tbz`
3. To load the driver on startup, type
   `echo 'oce_load="YES"' >> /boot/loader.conf`
4. Reboot the system.

The adapters provided by Emulex can be identified using the FreeBSD sysctl utility to check driver and firmware versions.

Note: It is important to ensure the latest firmware is loaded on the cards to go with the latest drivers. Verify on the Emulex web site.

Example of using sysctl to identify Ethernet cards with code versions:

```
root@ELXUKBSD91:/root # sysctl -a | grep oce | grep ion
oce0: <Emulex CNA NIC function:///4.2.116.0///> mem 0xfe778000-
0xfe77bfff,0xfe780000-0xfe79ffff,0xfe7a0000-0xfe7bffff irq 16 at device
0.0 on pci1
oce1: <Emulex CNA NIC function:///4.2.116.0///> mem 0xfe77c000-
0xfe77ffff,0xfe7c0000-0xfe7dffff,0xfe7e0000-0xfe7fffff irq 17 at device
0.1 on pci1
net.inet.ip.process_options: 1
dev.oce.0.%desc: Emulex CNA NIC function:///4.2.116.0///
dev.oce.0.%location: slot=0 function=0
dev.oce.0.component_revision: ///4.2.116.0///
dev.oce.0.firmware_version: 4.4.249.3
dev.oce.1.%desc: Emulex CNA NIC function:///4.2.116.0///
dev.oce.1.%location: slot=0 function=1
dev.oce.1.component_revision: ///4.2.116.0///
dev.oce.1.firmware_version: 4.4.249.3
```

To check the oce driver package is installed and verify version, use pkg_info:

```
root@ELXUKBSD91:/root # pkg_info | grep oce
oce-4.2.313.0       oce driver for freebsd
```

The only kernel parameter documented in our driver manual is **max_rsp_handled**, which is said to default to 512. It doesn't - it is 64 with 4.2.313.0 driver:

```
root@ELXUKBSD91:/root # sysctl -a | grep oce | grep max_rsp
dev.oce.0.max_rsp_handled: 64
dev.oce.1.max_rsp_handled: 64
```

Network interface configuration tool: ifconfig

As well as being able to set IP configuration data, ifconfig can also display and set driver parameters related to offloads. The Emulex driver manual documents LSO [*sic*] and TSO but the driver capabilities show more. Run ifconfig -m oceX to see allowed options and current settings. The data is displayed as a bit map and as a text label which does not immediately map to the ifconfig command to set/clear the flags:

```
root@ELXUKBSD91:/root # ifconfig -m oce0
oce0: flags=8002<BROADCAST,MULTICAST> metric 0 mtu 1500

options=507bb<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,JUMBO_MTU,VLAN_HWCS
UM,TSO4,TSO6,LRO,VLAN_HWFILTER,VLAN_HWTSO>

capabilities=507bb<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,JUMBO_MTU,VLAN
_HWCSUM,TSO4,TSO6,LRO,VLAN_HWFILTER,VLAN_HWTSO>
        ether 00:00:c9:e6:67:a8
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
        media: Ethernet autoselect
        status: no carrier
        supported media:
                media autoselect
```

Here the capabilities and enabled flags are the same: 0x507bb.

Note: **lso** is not listed here and attempting to enable it results in an error. It is a misprint and should be **lro**.

To set the hardware parameters on a "per port" basis, ifconfig can be used:

```
ifconfig [port] [parameter1] [parameter2] ...
```

To disable a parameter, use a minus "-" in front of the parameter(s)

```
ifconfig [port] [-parameter1] [-parameter2] ...
```

Example to disable RXCSUM:

```
ifconfig oce0 -txcsum
```

Table 1. List of hardware offload parameters and settings

Flag bit	ifconfig label	ifconfig set value	Notes
0	RXCSUM	rxcsum	
1	TXCSUM	txcsum	also disables tso4/6
3	VLAN_MTU		can't change
4	VLAN_HWTAGGING	vlanhwtag	
5	JUMBO_MTU		can't change
7	VLAN_HWCSUM	vlanhwcsum	can't disable
8	TSO4	tso4 or tso	tso or tso4 act on tso4 only
9	TSO6	tso6	can't disable tso6
10	LRO	lro	
16	VLAN_HWFILTER	vlanhwfilter	
18	VLAN_HWTSO	vlanhwtso	can't disable

Performance Tuning

Due to the complexities of network configurations, there is not likely to be a single set of parameters to give the best performance in every circumstance. Using a performance test tool such as Netperf http://www.netperf.org/ with two servers connected directly (back-to-back) with 10GbE cables is a good way to check the raw capabilities of a particular system. As supplied, the Emulex driver and firmware combination have all available hardware acceleration parameters enabled for optimum performance. Assuming the NIC is installed in the appropriate PCIe slot on a reasonably capable modern server, data transfers close to line rate should be observed for simple applications in a back-to-back connection.

Netperf allows several types of network traffic to be simulated, such as TCP Streaming, TCP SendFile and UDP Streaming to verify the data transfer capabilities of a particular system. In a “real” application, these results can be used as a foundation for analysis; however, there could be many network variables that affect performance as well as application specifics.

- The FreeBSD wiki has some information on 10GbE tuning at https://wiki.freebsd.org/NetworkPerformanceTuning
- Notes based on Nginx developer's work http://serverfault.com/questions/64356/freebsd-performance-tuning-sysctls-loader-conf-kernel

Netperf performance tests

Netperf 2.6.0 was used to establish the raw performance capabilities of the NICs in a simple back-to-back test.

Driver version used = 4.2.313.0 on FreeBSD 9.1 AMD64. Firmware = 4.4.249.3

Netperf server (NetServer) set up on HP DL360p (Gen8) server with a OCe11102 (Emulex BE3-based). OS=Centos 6.2.

Netperf client run on FreeBSD 9.1 on Supermicro X7DBU, Intel Xeon 2.83GHz, 16 GB RAM also with an Emulex OCe11102 (Emulex BE3-based).

Netperf scripts were run to measure bandwidth at various block sizes and MTUs in 60 second runs as follows. For MTU sizes of 576, 1500 and 9000:

- TCP_STREAM
  netperf -H $HOSTIP -t TCP_STREAM -C -c -l 60 -- -m $BSIZE
  Where BSIZE = 64, 128, 256, 512, 1K, 2K, 4K, 8K, 16K, 32K, 64K, 128K, 256K
- TCP_SENDFILE
  netperf -H $HOSTIP -t TCP_SENDFILE -C -c -l 60 -F $SENDFILE -- -m $BSIZE
  Where BSIZE = 64, 128, 256, 512, 1K, 2K, 4K, 8K, 16K, 32K, 64K, 128K, 256K
- UDP_STREAM
  netperf -H $HOSTIP -t UDP_STREAM -C -c -l 60 -- -m $BSIZE -s 128K -S 128K
  Where BSIZE = 64, 128, 256, 512, 1K, 2K, 4K, 8K, 16K, 32K, 65500

Chart 1: TCP_STREAM bandwidth

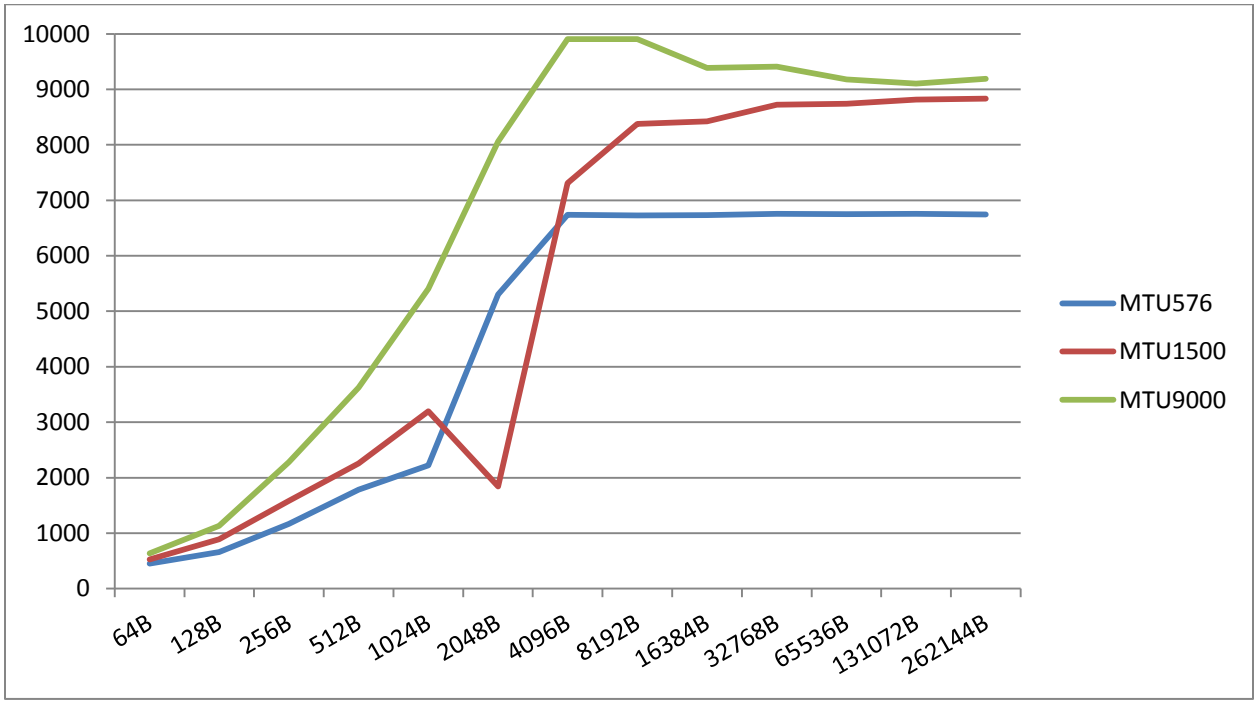


Chart 2: TCP_SENDFILE bandwidth

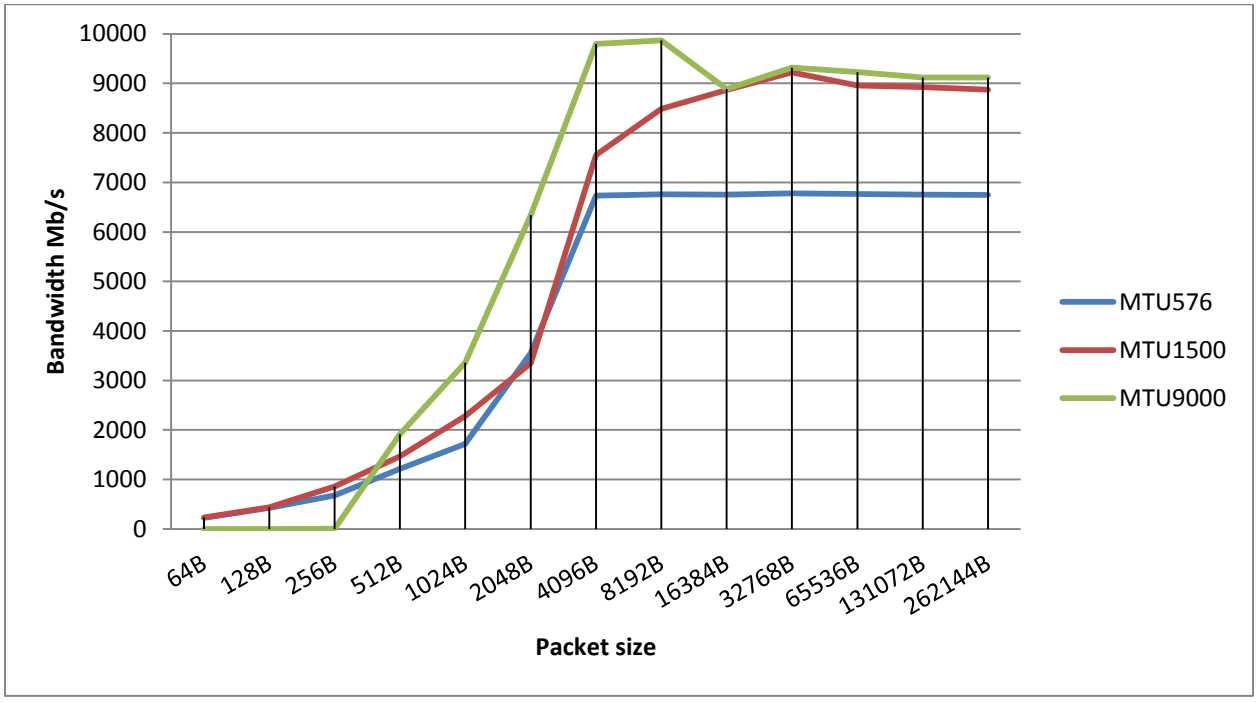
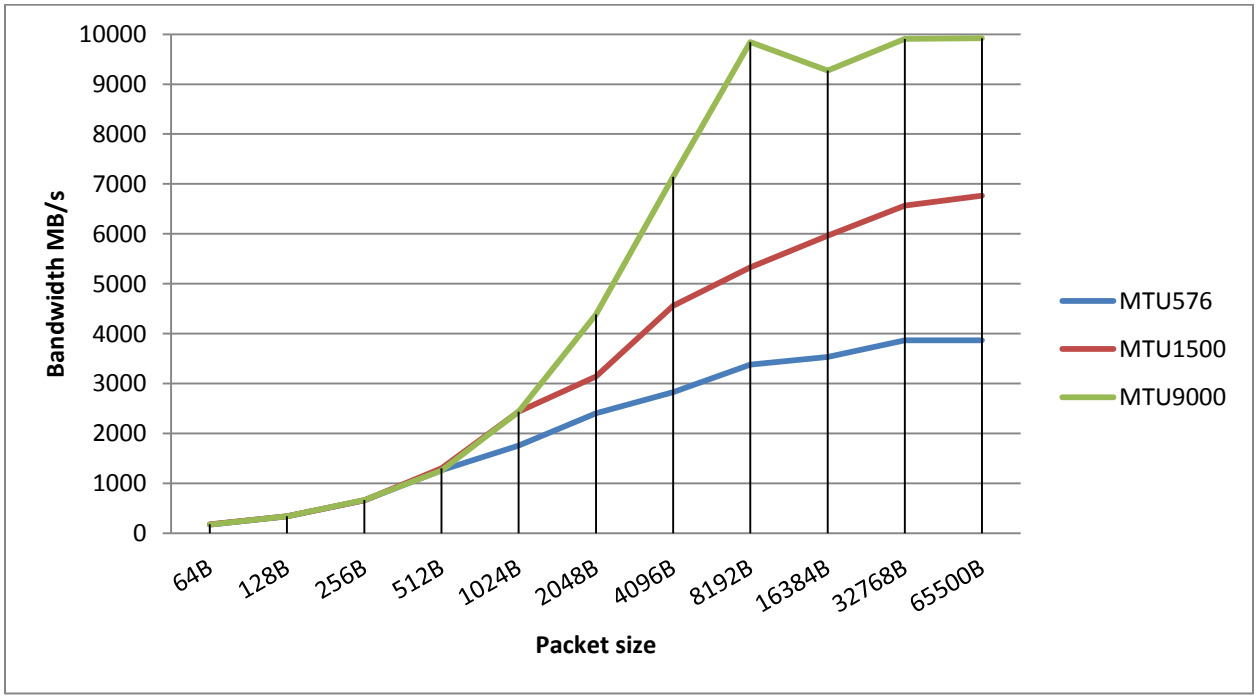


Chart 3: UDP_STREAM bandwidth



FreeBSD notes

Supported Versions

Emulex drivers and utilities are currently noted as supporting FreeBSD versions 8.1, 8.2 and 9.0. The Emulex NIC drivers for adapters provided by Emulex work fine under FreeBSD 9.1 and install correctly; however, it is recommended you verify the support status before using in production.

Offline Utilities

Emulex provides management tools to configure the NICs, including firmware loading. The Offline utilities are available for download from the FreeBSD part of the Downloads on www.emulex.com

If used on an unsupported version, the install script fails. In the install.sh script, modify the **determine_os()** function to check for the required version. For example, to install on FreeBSD 9.1, add the following check in **determine_os()**:

```
9.1)
    DRIVER_OS_FULL="$CLU_OS-$freebsd_version"
    CLU_OS_FULL="$CLU_OS-$freebsd_version"
    ;;
```

You also need to copy the source files to match the required OS version to include support for FreeBSD 9.1 under amd64 systems. Copy the entire directory: **elxflashOffline-FreeBSD-<version>/amd64/fbsd-9.0** to **elxflashOffline-FreeBSD-<version>/amd64/fbsd-9.1**. d

Note: Be aware that this is considered ***unsupported***.

Some FreeBSD installs do not include the bash shell as a default. The Emulex installation scripts expect bash to be present - note the first line "shebang" entry of:

#!/usr/local/bin/bash

To install bash, cd to **/usr/ports/shells/bash** and run **make install**. This will take a few minutes to complete if all goes well. The scripts may not work in the default shell (csh ?).

Foibles

When changing MTU size on FreeBSD NIC ports, cycle the port down then up again using ifconfig, or performance will suffer with TCP tests. Observed netperf impact is to drop near line rate (9000 Mb/s) to around 6000 Mb/s, and it is reasonable to assume this effect may be observed in production applications.

```
ifconfig oce0 down
ifconfig oce0 up
```

Appendix A: Script to run Netperf tests

```
#!/usr/local/bin/bash
#
# Script to run selection of Netperf tests
# SKP - 30 May, 2013
#

# Time for each test cycle.
RUNTIME=60
LOGFILE=npresults.txt
HOSTIP=10.1.1.110
LOCALNIC=oce0
REMIP=192.168.1.121
REMNIC=eth2
SENDFILE=/boot/kernel/kernel
UDPs=128K
UDPS=128K

date > $LOGFILE

for MTU in 576 1500 9000
do
# Set MTU on both sides
      ifconfig $LOCALNIC mtu $MTU
      ssh root@$REMIP ifconfig $REMNIC mtu $MTU

# Need to bounce local NIC on FreeBSD after MTU change or performance drops
# May as well do remote one while we're at it.

        ifconfig $LOCALNIC down
        ifconfig $LOCALNIC up

      ssh root@$REMIP ifconfig $REMNIC down
      ssh root@$REMIP ifconfig $REMNIC up

# Do TCP_STREAM tests
      echo TCP_STREAM tests >> $LOGFILE

      for BSIZE in 64 128 256 512 1K 2K 4K 8K 16K 32K 64K 128K 256K
      do
            echo "`date` TCP_STREAM mtu=$MTU Block size = $BSIZE" | tee -a
$LOGFILE
            netperf -H $HOSTIP -t TCP_STREAM -C -c -l $RUNTIME -- -m $BSIZE>>
$LOGFILE
      done
# Do TCP_SENDFILE tests
        echo TCP_SENDFILE tests >> $LOGFILE

        for BSIZE in 64 128 256 512 1K 2K 4K 8K 16K 32K 64K 128K 256K
        do
                echo "`date` TCP_SENDFILE mtu=$MTU Block size = $BSIZE" | tee
-a $LOGFILE
```

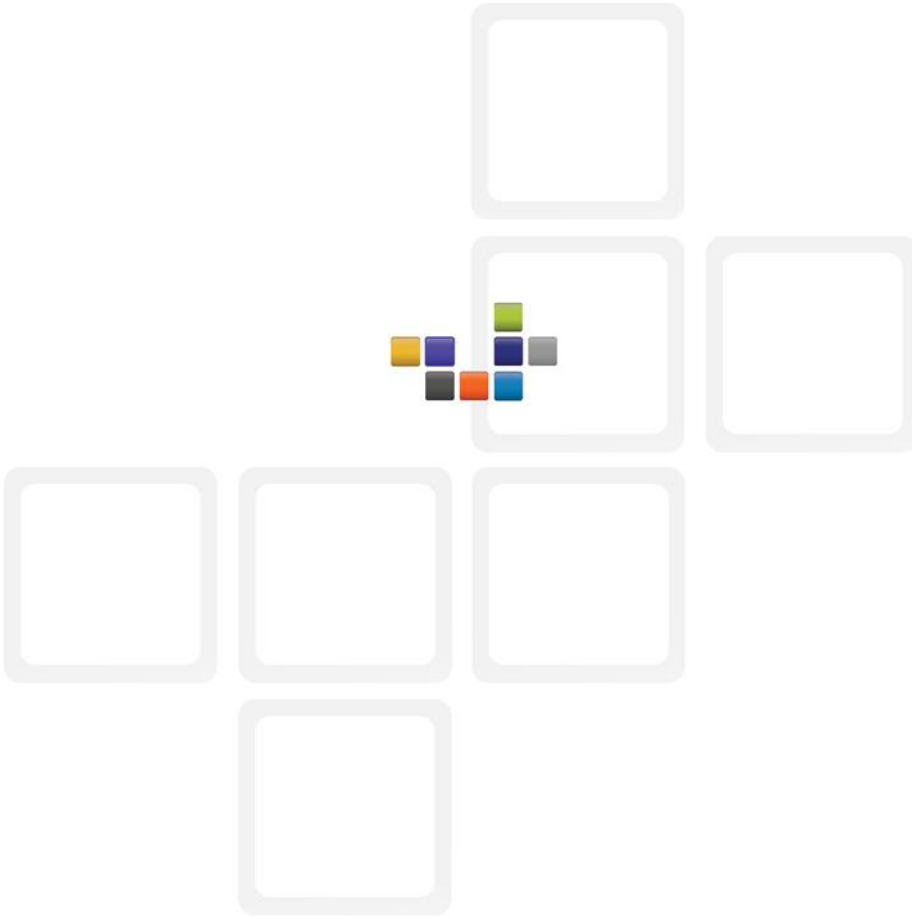
```
                netperf -H $HOSTIP -t TCP_SENDFILE -C -c -l $RUNTIME -F
$SENDFILE -- -m $BSIZE>> $LOGFILE
        done
# Do UDP_STREAM tests
        echo UDP_STREAM tests >> $LOGFILE

        for BSIZE in 64 128 256 512 1K 2K 4K 8K 16K 32K 65500
        do
                echo "`date` UDP_STREAM mtu=$MTU Block size = $BSIZE" | tee -
a $LOGFILE
                netperf -H $HOSTIP -t UDP_STREAM -C -c -l $RUNTIME -- -m
$BSIZE -s $UDPs -S $UDPS >> $LOGFILE
        done

done
```

More information

The Implementer's Lab website www.implementerslab.com

To help us improve our documents, please provide feedback at implementerslab@emulex.com.

Some of these products may not be available in the U.S. Please contact your supplier for more information.



www.emulex.com

World Headquarters 3333 Susan Street, Costa Mesa, California 92626 +1 714 662 5600
Bangalore, India +91 80 40156789 | **Beijing, China** +86 10 68499547
Dublin, Ireland+35 3 (0)1 652 1700 | **Munich, Germany** +49 (0) 89 97007 177
Paris, France +33 (0) 158 580 022 | **Tokyo, Japan** +81 3 5322 1348
Wokingham, United Kingdom +44 (0) 118 977 2929