



Cisco OpenFabrics Enterprise Distribution InfiniBand Host Drivers User Guide for Linux

Release 1.1
December 2006

Corporate Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Text Part Number: OL-10778-01



THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

CCVP, the Cisco Logo, and the Cisco Square Bridge logo are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn is a service mark of Cisco Systems, Inc.; and Access Registrar, Aironet, BPX, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, Follow Me Browsing, FormShare, GigaDrive, GigaStack, HomeLink, Internet Quotient, IOS, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, LightStream, Linksys, MeetingPlace, MGX, Networking Academy, Network Registrar, *Packet*, PIX, ProConnect, RateMUX, ScriptShare, SlideCast, SMARTnet, StackWise, The Fastest Way to Increase Your Internet Quotient, and TransPath are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0609R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Cisco OpenFabrics Enterprise Distribution InfiniBand Host Drivers User Guide for Linux
© 2006 Cisco Systems, Inc. All rights reserved.



Preface vii

Audience	vii
Organization	vii
Conventions	viii
Related Documentation	ix
Obtaining Documentation	x
Cisco.com	x
Product Documentation DVD	x
Ordering Documentation	x
Documentation Feedback	x
Cisco Product Security Overview	xi
Reporting Security Problems in Cisco Products	xi
Product Alerts and Field Notices	xii
Obtaining Technical Assistance	xii
Cisco Support Website	xii
Submitting a Service Request	xiii
Definitions of Service Request Severity	xiii
Obtaining Additional Publications and Information	xiv

CHAPTER 1

About Host Drivers 1-1

Introduction	1-1
Architecture	1-2
Supported Protocols	1-3
IPoIB	1-3
SRP	1-3
SDP	1-3
Supported API	1-3
HCA Utilities and Diagnostics	1-4
Root and Non-root Conventions in Examples	1-4

CHAPTER 2

Installing Host Drivers 2-1

Introduction	2-1
Contents of OFED ISO Image	2-2

Install Host Drivers from an ISO Image 2-2
 Uninstall Host Drivers from an ISO Image 2-5

CHAPTER 3

IP over IB Protocol 3-1

Introduction 3-1
 Manually Configure IPoIB for Default IB Partition 3-2
 Subinterfaces 3-2
 Create a Subinterface Associated with a Specific IB Partition 3-3
 Remove a Subinterface Associated with a Specific IB Partition 3-4
 Verify IPoIB Functionality 3-5
 IPoIB Performance 3-6
 Sample Startup Configuration File 3-7

CHAPTER 4

SCSI RDMA Protocol 4-1

Introduction 4-1
 Configure SRP 4-1
 Configure ITLs when Using Fibre Channel Gateway 4-2
 Configure ITLs with Element Manager while No Global Policy Restrictions Apply 4-2
 Configure ITLs with Element Manager while Global Policy Restrictions Apply 4-3
 Configure SRP Host 4-5
 Verify SRP 4-7
 Verify SRP Functionality 4-7
 Verify with Element Manager 4-9

CHAPTER 5

Sockets Direct Protocol 5-1

Introduction 5-1
 Configure IPoIB Interfaces 5-1
 Convert Sockets-Based Application 5-1
 Explicit/Source Code Conversion Type 5-2
 Automatic Conversion Type 5-2
 Log Statement 5-2
 Use Statement 5-3
 SDP Performance 5-4
 Netperf Server with IPoIB and SDP 5-6

CHAPTER 6

MVAPICH MPI and Open MPI 6-1

Introduction 6-1

Initial Setup	6-2
Configure SSH	6-2
Edit Environment Variables	6-5
Set Environment Variables in System-Wide Startup Files	6-5
Set up MVAPICH in System-Wide Startup Files	6-5
Set up Open MPI in System-Wide Startup Files	6-6
Edit Environment Variables In the Users' Shell Startup Files	6-6
Set up MVAPICH in Users' Shell Startup Files	6-7
Set up Open MPI in Users' Shell Startup Files	6-7
Edit Environment Variables Manually	6-7
Set up MVAPICH Manually in a Shell	6-7
Set up Open MPI Manually in a Shell	6-8
Perform MPI Bandwidth Test	6-8
Perform MPI Latency Test	6-10
Perform Intel MPI Benchmarks (IMB) Test	6-11
Compile MPI Programs	6-14

CHAPTER 7**HCA Utilities and Diagnostics** 7-1

Introduction	7-1
hca_self_test Utility	7-1
tvflash Utility	7-3
View Card Type and Firmware Version	7-3
Upgrade Firmware	7-4
Diagnostics	7-4
Performance Tests	7-5
Miscellaneous Utilities	7-6

APPENDIX A**Acronyms and Abbreviations** A-1**INDEX**



Preface

This preface describes who should read the *Cisco Open Fabrics Enterprise Distribution InfiniBand Host Drivers User Guide for Linux*, how it is organized, and its document conventions. It contains the following sections:

- [Audience, page vii](#)
- [Organization, page vii](#)
- [Conventions, page viii](#)
- [Related Documentation, page ix](#)
- [Obtaining Documentation, page x](#)
- [Documentation Feedback, page x](#)
- [Cisco Product Security Overview, page xi](#)
- [Product Alerts and Field Notices, page xii](#)
- [Obtaining Technical Assistance, page xii](#)
- [Obtaining Additional Publications and Information, page xiv](#)

Audience

The intended audience is the administrator responsible for installing, configuring, and managing host drivers and host card adapters. This administrator should have experience administering similar networking or storage equipment.

Organization

This publication is organized as follows:

Chapter	Title	Description
Chapter 1	About Host Drivers	Describes the fundamentals of the OpenFabrics Enterprise Distribution InfiniBand host driver.
Chapter 2	Installing Host Drivers	Describes the installation of host drivers.

Chapter	Title	Description
Chapter 3	IP over IB Protocol	Describes how to configure IPoIB to run IP traffic over an InfiniBand network.
Chapter 4	SCSI RDMA Protocol	Describes how to configure SRP.
Chapter 5	Sockets Direct Protocol	Describes how to configure and run the SDP.
Chapter 6	MVAPICH MPI and Open MPI	Describes the setup and configuration information for MVAPICH MPI and Open MPI.
Chapter 7	HCA Utilities and Diagnostics	Describes the fundamental HCA features for basic usability and the starting points for troubleshooting.
Appendix A	Acronyms and Abbreviations	Defines the acronyms and abbreviations that are used in this publication.

Conventions

This document uses the following conventions:

Convention	Description
boldface font	Commands, command options, and keywords are in boldface . Bold text indicates Chassis Manager elements or text that you must enter as-is.
<i>italic font</i>	Arguments in commands for which you supply values are in <i>italics</i> . Italics not used in commands indicate emphasis.
Menu1 > Menu2 > Item...	Series indicate a pop-up menu sequence to open a form or execute a desired function.
[]	Elements in square brackets are optional.
{ x y z }	Alternative keywords are grouped in braces and separated by vertical bars. Braces can also be used to group keywords and/or arguments; for example, { interface <i>interface</i> type }.
[x y z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
screen font	Terminal sessions and information the system displays are in screen font.
boldface screen font	Information you must enter is in boldface screen font .
<i>italic screen font</i>	Arguments for which you supply values are in <i>italic screen font</i> .

Convention	Description
^	The symbol ^ represents the key labeled Control—for example, the key combination ^D in a screen display means hold down the Control key while you press the D key.
< >	Nonprinting characters, such as passwords are in angle brackets.
[]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Notes use the following conventions:



Note

Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the publication.

Cautions use the following conventions:



Caution

Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Related Documentation

For additional information related to the OFED IB host drivers, see the following documents:

- *Release Notes for Topspin Release 2.7.0 FCS*
- *Cisco SFS 7000 Series Product Family Chassis Manager User Guide*
- *Cisco SFS 7000 Series Product Family Element Manager User Guide*
- *Cisco SFS 7000 Series Product Family Command Reference Guide*
- *Host Channel Adapter Hardware Guide*

Obtaining Documentation

Cisco documentation and additional literature are available on Cisco.com. This section explains the product documentation resources that Cisco offers.

Cisco.com

You can access the most current Cisco documentation at this URL:

<http://www.cisco.com/techsupport>

You can access the Cisco website at this URL:

<http://www.cisco.com>

You can access international Cisco websites at this URL:

http://www.cisco.com/public/countries_languages.shtml

Product Documentation DVD

The Product Documentation DVD is a library of technical product documentation on a portable medium. The DVD enables you to access installation, configuration, and command guides for Cisco hardware and software products. With the DVD, you have access to the HTML documentation and some of the PDF files found on the Cisco website at this URL:

<http://www.cisco.com/univercd/home/home.htm>

The Product Documentation DVD is created and released regularly. DVDs are available singly or by subscription. Registered Cisco.com users can order a Product Documentation DVD (product number DOC-DOCDVD= or DOC-DOCDVD=SUB) from Cisco Marketplace at the Product Documentation Store at this URL:

<http://www.cisco.com/go/marketplace/docstore>

Ordering Documentation

You must be a registered Cisco.com user to access Cisco Marketplace. Registered users may order Cisco documentation at the Product Documentation Store at this URL:

<http://www.cisco.com/go/marketplace/docstore>

If you do not have a user ID or password, you can register at this URL:

<http://tools.cisco.com/RPF/register/register.do>

Documentation Feedback

You can provide feedback about Cisco technical documentation on the Cisco Support site area by entering your comments in the feedback form available in every online document.

Cisco Product Security Overview

Cisco provides a free online Security Vulnerability Policy portal at this URL:

http://www.cisco.com/en/US/products/products_security_vulnerability_policy.html

From this site, you will find information about how to do the following:

- Report security vulnerabilities in Cisco products
- Obtain assistance with security incidents that involve Cisco products
- Register to receive security information from Cisco

A current list of security advisories, security notices, and security responses for Cisco products is available at this URL:

<http://www.cisco.com/go/psirt>

To see security advisories, security notices, and security responses as they are updated in real time, you can subscribe to the Product Security Incident Response Team Really Simple Syndication (PSIRT RSS) feed. Information about how to subscribe to the PSIRT RSS feed is found at this URL:

http://www.cisco.com/en/US/products/products_psirt_rss_feed.html

Reporting Security Problems in Cisco Products

Cisco is committed to delivering secure products. We test our products internally before we release them, and we strive to correct all vulnerabilities quickly. If you think that you have identified a vulnerability in a Cisco product, contact PSIRT:

- For emergencies only—security-alert@cisco.com

An emergency is either a condition in which a system is under active attack or a condition for which a severe and urgent security vulnerability should be reported. All other conditions are considered nonemergencies.

- For nonemergencies—psirt@cisco.com

In an emergency, you can also reach PSIRT by telephone:

- 1 877 228-7302
- 1 408 525-6532



Tip

We encourage you to use Pretty Good Privacy (PGP) or a compatible product (for example, GnuPG) to encrypt any sensitive information that you send to Cisco. PSIRT can work with information that has been encrypted with PGP versions 2.x through 9.x.

Never use a revoked encryption key or an expired encryption key. The correct public key to use in your correspondence with PSIRT is the one linked in the Contact Summary section of the Security Vulnerability Policy page at this URL:

http://www.cisco.com/en/US/products/products_security_vulnerability_policy.html

The link on this page has the current PGP key ID in use.

If you do not have or use PGP, contact PSIRT to find other means of encrypting the data before sending any sensitive material.

Product Alerts and Field Notices

Modifications to or updates about Cisco products are announced in Cisco Product Alerts and Cisco Field Notices. You can receive these announcements by using the Product Alert Tool on Cisco.com. This tool enables you to create a profile and choose those products for which you want to receive information.

To access the Product Alert Tool, you must be a registered Cisco.com user. Registered users can access the tool at this URL:

<http://tools.cisco.com/Support/PAT/do/ViewMyProfiles.do?local=en>

To register as a Cisco.com user, go to this URL:

<http://tools.cisco.com/RPF/register/register.do>

Obtaining Technical Assistance

Cisco Technical Support provides 24-hour-a-day award-winning technical assistance. The Cisco Support website on Cisco.com features extensive online support resources. In addition, if you have a valid Cisco service contract, Cisco Technical Assistance Center (TAC) engineers provide telephone support. If you do not have a valid Cisco service contract, contact your reseller.

Cisco Support Website

The Cisco Support website provides online documents and tools for troubleshooting and resolving technical issues with Cisco products and technologies. The website is available 24 hours a day at this URL:

<http://www.cisco.com/en/US/support/index.html>

Access to all tools on the Cisco Support website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register at this URL:

<http://tools.cisco.com/RPF/register/register.do>

**Note**

Before you submit a request for service online or by phone, use the **Cisco Product Identification Tool** to locate your product serial number. You can access this tool from the Cisco Support website by clicking the **Get Tools & Resources** link, clicking the **All Tools (A-Z)** tab, and then choosing **Cisco Product Identification Tool** from the alphabetical list. This tool offers three search options: by product ID or model name; by tree view; or, for certain products, by copying and pasting **show** command output. Search results show an illustration of your product with the serial number label location highlighted. Locate the serial number label on your product and record the information before placing a service call.



Tip

Displaying and Searching on Cisco.com

If you suspect that the browser is not refreshing a web page, force the browser to update the web page by holding down the Ctrl key while pressing **F5**.

To find technical information, narrow your search to look in technical documentation, not the entire Cisco.com website. After using the Search box on the Cisco.com home page, click the **Advanced Search** link next to the Search box on the resulting page and then click the **Technical Support & Documentation** radio button.

To provide feedback about the Cisco.com website or a particular technical document, click **Contacts & Feedback** at the top of any Cisco.com web page.

Submitting a Service Request

Using the online TAC Service Request Tool is the fastest way to open S3 and S4 service requests. (S3 and S4 service requests are those in which your network is minimally impaired or for which you require product information.) After you describe your situation, the TAC Service Request Tool provides recommended solutions. If your issue is not resolved using the recommended resources, your service request is assigned to a Cisco engineer. The TAC Service Request Tool is located at this URL:

<http://www.cisco.com/techsupport/servicerequest>

For S1 or S2 service requests, or if you do not have Internet access, contact the Cisco TAC by telephone. (S1 or S2 service requests are those in which your production network is down or severely degraded.) Cisco engineers are assigned immediately to S1 and S2 service requests to help keep your business operations running smoothly.

To open a service request by telephone, use one of the following numbers:

Asia-Pacific: +61 2 8446 7411

Australia: 1 800 805 227

EMEA: +32 2 704 55 55

USA: 1 800 553 2447

For a complete list of Cisco TAC contacts, go to this URL:

<http://www.cisco.com/techsupport/contacts>

Definitions of Service Request Severity

To ensure that all service requests are reported in a standard format, Cisco has established severity definitions.

Severity 1 (S1)—An existing network is “down” or there is a critical impact to your business operations. You and Cisco will commit all necessary resources around the clock to resolve the situation.

Severity 2 (S2)—Operation of an existing network is severely degraded, or significant aspects of your business operations are negatively affected by inadequate performance of Cisco products. You and Cisco will commit full-time resources during normal business hours to resolve the situation.

Severity 3 (S3)—Operational performance of the network is impaired while most business operations remain functional. You and Cisco will commit resources during normal business hours to restore service to satisfactory levels.

Severity 4 (S4)—You require information or assistance with Cisco product capabilities, installation, or configuration. There is little or no effect on your business operations.

Obtaining Additional Publications and Information

Information about Cisco products, technologies, and network solutions is available from various online and printed sources.

- The Cisco Online Subscription Center is the website where you can sign up for a variety of Cisco e-mail newsletters and other communications. Create a profile and then select the subscriptions that you would like to receive. To visit the Cisco Online Subscription Center, go to this URL:

<http://www.cisco.com/offer/subscribe>

- The *Cisco Product Quick Reference Guide* is a handy, compact reference tool that includes brief product overviews, key features, sample part numbers, and abbreviated technical specifications for many Cisco products that are sold through channel partners. It is updated twice a year and includes the latest Cisco channel product offerings. To order and find out more about the *Cisco Product Quick Reference Guide*, go to this URL:

<http://www.cisco.com/go/guide>

- Cisco Marketplace provides a variety of Cisco books, reference guides, documentation, and logo merchandise. Visit Cisco Marketplace, the company store, at this URL:

<http://www.cisco.com/go/marketplace/>

- Cisco Press publishes a wide range of general networking, training, and certification titles. Both new and experienced users will benefit from these publications. For current Cisco Press titles and other information, go to Cisco Press at this URL:

<http://www.ciscopress.com>

- *Internet Protocol Journal* is a quarterly journal published by Cisco for engineering professionals involved in designing, developing, and operating public and private internets and intranets. You can access the *Internet Protocol Journal* at this URL:

<http://www.cisco.com/ipj>

- Networking products offered by Cisco, as well as customer support services, can be obtained at this URL:

<http://www.cisco.com/en/US/products/index.html>

- Networking Professionals Connection is an interactive website where networking professionals share questions, suggestions, and information about networking products and technologies with Cisco experts and other networking professionals. Join a discussion at this URL:
<http://www.cisco.com/discuss/networking>
- “What’s New in Cisco Documentation” is an online publication that provides information about the latest documentation releases for Cisco products. Updated monthly, this online publication is organized by product category to direct you quickly to the documentation for your products. You can view the latest release of “What’s New in Cisco Documentation” at this URL:
<http://www.cisco.com/univercd/cc/td/doc/abtnicd/136957.htm>
- World-class networking training is available from Cisco. You can view current offerings at this URL:
<http://www.cisco.com/en/US/learning/index.html>



About Host Drivers

The following sections appear in this chapter:

- [Introduction, page 1-1](#)
- [Architecture, page 1-2](#)
- [Supported Protocols, page 1-3](#)
- [Supported API, page 1-3](#)
- [HCA Utilities and Diagnostics, page 1-4](#)
- [Root and Non-root Conventions in Examples, page 1-4](#)

Introduction

The Cisco IB HCA offers high-performance 10-Gbps InfiniBand connectivity to PCI-X and PCI-Express-based servers. As an integral part of the Cisco server switching solution, the Cisco IB HCA allows you to create a unified fabric for consolidating clustering, networking, and storage communications.

After you physically install the HCA in the server, install the drivers to run IB-capable protocols. HCAs support the following protocols in the Linux environment:

- IPoIB
- SRP
- SDP

HCAs support the following APIs in the Linux environment:

- MVAICH MPI
- Open MPI

Host drivers also provide utilities to help you configure and verify your HCA. These utilities provide upgrade and diagnostic features.



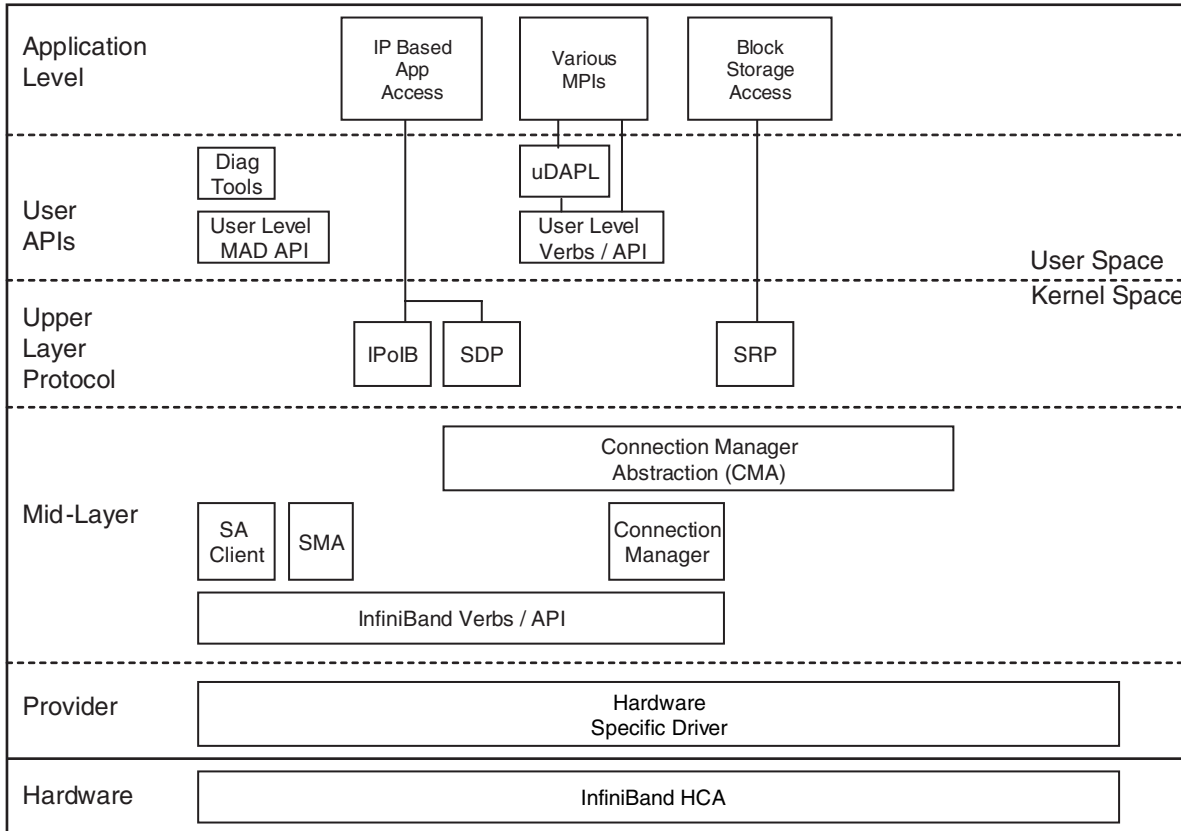
Note

For expansions of acronyms and abbreviations used in this publication, see [Appendix A, “Acronyms and Abbreviations.”](#)

Architecture

Figure 1-1 displays the software architecture of the protocols and APIs that HCAs support. The figure displays ULPs and APIs in relation to other IB software elements.

Figure 1-1 HCA Supported Protocols and API Architecture



IPoIB	IP over InfiniBand	MPI	Message Passing Interface	MAD	Management Datagram
SDP	Sockets Direct Protocol	UDAPL	User Direct Access Programming Lib	SMA	Subnet Manager Agent
SRP	SCSI RDMA Protocol (Initiator)	SA	Subnet Administrator	HCA	Host Channel Adapter

180411

Supported Protocols

The term protocol refers to software in the networking layer in kernel space. The protocols documented in this book are described in this section.

IPoIB

The IPoIB protocol passes IP traffic over the IB network. Configuring IPoIB requires similar steps to configuring IP on an Ethernet network. SDP relies on IPoIB to resolve IP addresses. (See the “SDP” section on page 1-3.)

To configure IPoIB, you assign an IP address and subnet mask to each IB port. IPoIB automatically adds IB interface names to the IP network configuration. To configure IPoIB, refer to [Chapter 3, “IP over IB Protocol.”](#)

SRP

SRP runs SCSI commands across RDMA-capable networks so that IB hosts can communicate with Fibre Channel storage devices and IB-attached storage devices. SRP requires a server switch with a Fibre Channel gateway to connect the host to Fibre Channel storage. In conjunction with a server switch, SRP disguises IB-attached hosts as Fibre Channel-attached hosts. The topology transparency feature lets Fibre Channel storage communicate seamlessly with IB-attached hosts (known as SRP hosts). For configuration instructions, see [Chapter 4, “SCSI RDMA Protocol.”](#)

SDP

SDP is an IB-specific upper layer protocol. It defines a standard wire protocol to support stream sockets networking over IB. SDP enables sockets-based applications to take advantage of the enhanced performance features provided by IB and achieves lower latency and higher bandwidth than IPoIB running sockets-based applications. It provides a high-performance, zero-copy data transfer protocol for stream-socket networking over an IB fabric. You can configure the driver to automatically translate TCP to SDP based on a source IP, a destination, or an application name. For configuration instructions, see [Chapter 5, “Sockets Direct Protocol.”](#)

Supported API

The term API refers to software in the networking layer in user space.

MPI is a standard library functionality in C, C++, and Fortran that can be used to implement a message passing program. MPI allows the coordination of a program running as multiple processes in a distributed memory environment. This book includes setup and configuration information for MVAPICH MPI and Open MPI. For more information, see [Chapter 6, “MVAPICH MPI and Open MPI.”](#)

HCA Utilities and Diagnostics

The HCA utilities provide basic tools to view HCA attributes and run preliminary troubleshooting tasks. For more information, see [Chapter 7, “HCA Utilities and Diagnostics.”](#)

Root and Non-root Conventions in Examples

This document uses the following conventions to signify root and non-root accounts:

Convention	Description
host1# host2#	When this prompt appears in an example, it indicates that you are in a root account.
host1\$ host2\$	When this prompt appears in an example, it indicates that you are in a non-root account.



Installing Host Drivers

The following sections appear in this chapter:

- [Introduction, page 2-1](#)
- [Contents of OFED ISO Image, page 2-2](#)
- [Install Host Drivers from an ISO Image, page 2-2](#)
- [Uninstall Host Drivers from an ISO Image, page 2-5](#)

Introduction

OFED is open source software for RDMA fabric technologies that works independent of the multiple underlying networking layers. OFED provides tools, communications, and resources for vendors and developers to create, refine, and publish standard open source software stacks for RDMA-capable data center and high-performance computing fabrics. OFED is comprised of technology vendors and end-user organizations.

The goal of OFED is to drive industry-standard solutions to the market based on IB interconnect technology and a common software stack. OFED is supported by all major IB vendors, and it provides a common IB stack for customers. For more information about OFED, go to this URL:

<http://www.openfabrics.org/>

The Cisco OFED is delivered as an ISO image. The ISO image contains both source code and binary RPMs for selected Linux distributions. The Cisco OFED distribution contains installation scripts called `ofedinstall`. The install script performs the necessary steps to accomplish the following:

- Discover the currently installed kernel
- Uninstall any IB stacks that are part of the standard operating system distribution or Cisco commercial stack
- Install the Cisco OFED binary RPMs if they are available for the current kernel
- Identify the currently installed IB HCA and perform the required firmware updates



Note

For specific details about which binary RPMs are included and which standard Linux distributions and kernels are currently supported, refer to the Cisco OFED release notes.

**Note**

See the “[Root and Non-root Conventions in Examples](#)” section on page 1-4 for details about the significance of prompts used in the examples in this chapter.

Contents of OFED ISO Image

The OFED ISO image contains the following directories and files:

- docs/
This directory contains the Cisco OFED related documents.
- extras/
This directory contains the components included by Cisco Systems, in addition to the standard OFED offerings.
- firmware/
This directory contains the Cisco IB HCA firmware images.
- ofedinstall
This is the OFED installation script.
- rhel4/
This directory contains the binary RPMs for Red Hat Enterprise Linux 4 (RHEL4).
- sles10/
This directory contains the binary RPMs for SUSE Linux Enterprise Server 10 (SLES10).
- src/
This directory contains the OFED source tarball.

Install Host Drivers from an ISO Image

Follow these steps to install host drivers from an ISO image.

- Step 1** Verify that the system has a viable HCA installed by ensuring that you can see the InfiniHost entries in the display.

This example shows that the installed HCA is viable:

```
host1# lspci -v | grep Mellanox
06:01.0 PCI bridge: Mellanox Technologies MT23108 PCI Bridge (rev a0) (prog-if 00
[Normal decode])
07:00.0 InfiniBand: Mellanox Technologies MT23108 InfiniHost (rev a0)
Subsystem: Mellanox Technologies MT23108 InfiniHost
```

- Step 2** Download an ISO image, and copy it to your network.
You can download an ISO image from <http://www.cisco.com/cgi-bin/tablebuild.pl/sfs-linux>
- Step 3** Use the md5sum utility to confirm the file integrity of your ISO image.
- Step 4** Install drivers from an ISO image on your network.

This example shows how to install host drivers from an ISO image:

```
host1# mount -o ro,loop Cisco_OFED-1.1-fcs.iso /mnt
host1# /mnt/ofedinstall
```

This program will install Cisco OFED IB packages on your machine.

Note that all other Cisco, OFED, or OpenIB IB packages will be removed.

Do you want to continue?[y/N]:y

The following kernels are installed, but do not have drivers available:
2.6.9-34.EL.x86_64

Drivers for the following kernels will be installed.
2.6.9:34.EL:x86_64:smp

```
Installing packages.
Preparing... ##### [100%]
 1:tvflash ##### [ 2%]
 2:kernel-ib ##### [ 5%]
 3:kernel-ib-devel ##### [ 7%]
 4:dapl ##### [ 10%]
 5:dapl-devel ##### [ 12%]
 6:ibutils ##### [ 14%]
 7:ipoibtools ##### [ 17%]
 8:libibcommon ##### [ 19%]
 9:libibcommon-devel ##### [ 21%]
10:libibmad ##### [ 24%]
11:libibmad-devel ##### [ 26%]
12:libibumad ##### [ 29%]
13:libibumad-devel ##### [ 31%]
14:libibverbs ##### [ 33%]
15:libibverbs-devel ##### [ 36%]
16:libibverbs-utils ##### [ 38%]
17:libmthca ##### [ 40%]
18:libmthca-devel ##### [ 43%]
19:libopensm ##### [ 45%]
20:libopensm-devel ##### [ 48%]
21:libosmcomp ##### [ 50%]
22:libosmcomp-devel ##### [ 52%]
23:libosmvendor ##### [ 55%]
24:libosmvendor-devel ##### [ 57%]
25:librdmacm ##### [ 60%]
26:librdmacm-devel ##### [ 62%]
27:librdmacm-utils ##### [ 64%]
28:libsdp ##### [ 67%]
29:mpich_mlx_gcc ##### [ 69%]
30:mpich_mlx_intel ##### [ 71%]
31:mpitests_mpich_mlx_gcc ##### [ 74%]
32:mpitests_mpich_mlx_intel ##### [ 76%]
33:mpitests_openmpi_gcc ##### [ 79%]
34:mpitests_openmpi_intel ##### [ 81%]
35:mstflint ##### [ 83%]
36:ofed-docs ##### [ 86%]
37:ofed-scripts ##### [ 88%]
38:openib-diags ##### [ 90%]
39:openmpi_gcc ##### [ 93%]
40:openmpi_intel ##### [ 95%]
41:perftest ##### [ 98%]
42:srptools ##### [100%]
```

Installing hca_self_test, cisco_srp_add_targets, and MPI examples.

Configuring /etc/security/limits.conf.

Installing hca_self_test, cisco_srp_add_targets, and MPI examples.

Uninstall Host Drivers from an ISO Image

This example shows how to uninstall a host driver from a device:

```
host1# /usr/local/ofed/uninstall.sh
This program will uninstall all IB packages on your machine.
Do you want to continue?[y/N]:y
Removing OFED Software installations
Running /bin/rpm -e mpitests_openmpi_intel-2.0-0
Running /bin/rpm -e mpitests_mpich_mlx_gcc-2.0-0
Running /bin/rpm -e mpitests_openmpi_gcc-2.0-0
Running /bin/rpm -e mpitests_mpich_mlx_intel-2.0-0
Running /bin/rpm -e mpich_mlx_intel-0.9.7_mlx2.2.0-1
Running /bin/rpm -e mpich_mlx_gcc-0.9.7_mlx2.2.0-1
Running /bin/rpm -e openmpi_intel-1.1.2-1
Running /bin/rpm -e openmpi_gcc-1.1.2-1
Running /bin/rpm -e kernel-ib kernel-ib-devel libibverbs libibverbs-devel libibverbs-utils
libmthca libmthca-devel perftest mstflint libsdp srptools ipoibtools tvflash libibcommon
libibcommon-devel libibmad libibmad-devel libibumad libibumad-devel libopensm
libopensm-devel libosmcomp libosmcomp-devel libosmvendor libosmvendor-devel librdmacm
librdmacm-devel librdmacm-utils dapl dapl-devel openib-diags ibutils ofed-docs
ofed-scripts
warning: /etc/infiniband/openib.conf saved as /etc/infiniband/openib.conf.rpmsave
```




IP over IB Protocol

The following sections appear in this chapter:

- [Introduction, page 3-1](#)
- [Manually Configure IPoIB for Default IB Partition, page 3-2](#)
- [Subinterfaces, page 3-2](#)
- [Verify IPoIB Functionality, page 3-5](#)
- [IPoIB Performance, page 3-6](#)
- [Sample Startup Configuration File, page 3-7](#)

Introduction

Configuring IPoIB requires that you follow similar steps to those for configuring IP on an Ethernet network. When you configure IPoIB, you assign an IP address and a subnet mask to each HCA port. The first HCA port on the first HCA in the host is the `ib0` interface, the second port is `ib1`, and so on.



Note

To enable these IPoIB settings across reboots, you must explicitly add these settings to the networking interface startup configuration file. For a sample configuration file, see the [“Sample Startup Configuration File”](#) section on page 3-7.

Refer to your Linux distribution documentation for additional information about configuring IP addresses.



Note

See the [“Root and Non-root Conventions in Examples”](#) section on page 1-4 for details about the significance of prompts used in the examples in this chapter.

Manually Configure IPoIB for Default IB Partition

To configure IPoIB on your Linux host, perform the following steps:

-
- Step 1** Log in to your Linux host.
- Step 2** To configure the interface, enter the **ifconfig** command with the following:
- The appropriate IB interface (**ib0** or **ib1** on a host with one HCA)
 - The IP address that you want to assign to the interface
 - The **netmask** keyword
 - The subnet mask that you want to assign to the interface

This example shows how to configure an IB interface:

```
host1# ifconfig ib0 192.168.0.1 netmask 255.255.252.0
```

- Step 3** (Optional) Enter the **ifconfig** command with the appropriate port identifier *ib#* argument to verify the configuration.

This example shows how to verify the configuration:

```
host1# ifconfig ib0
ib0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
        inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
        inet6 addr: fe80::205:ad00:20:849/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:2044  Metric:1
        RX packets:46  errors:0  dropped:0  overruns:0  frame:0
        TX packets:47  errors:0  dropped:0  overruns:0  carrier:0
        collisions:0 txqueuelen:128
        RX bytes:45056 (44.0 KiB) TX bytes:3011 (2.9 KiB)
```

- Step 4** Repeat [step 2](#) and [step 3](#) on the remaining interface(s).
-

Subinterfaces

Subinterfaces divide primary (parent) interfaces to provide traffic isolation. Partition assignments distinguish subinterfaces from parent interfaces. The default Partition Key (p_key), ff:ff, applies to the primary (parent) interface.

This section includes the following topics:

- [Create a Subinterface Associated with a Specific IB Partition, page 3-3](#)
- [Remove a Subinterface Associated with a Specific IB Partition, page 3-4](#)

Create a Subinterface Associated with a Specific IB Partition

This section describes how to create a subinterface associated with a specific IB partition.

To create a subinterface, perform the following steps:

Step 1 Create a partition on an IB server switch. Alternatively, you can choose to create the partition of the InfiniBand interface first, and then create the partition for the ports on the IB server switch. Refer to the *Cisco SFS 7000 Series Product Family Element Manager User Guide* for information regarding partitions on the IB server switch.

Step 2 Log in to your host.

Step 3 Add the value of the partition key to the file as root user.

This example shows how to add partition 80:02 to the primary interface ib0:

```
host1# echo 0x8002 >> /sys/class/net/ib0/create_child
```

Step 4 Verify that the interface is set up by ensuring that ib0.8002 is displayed.

This example shows how to set up the interface:

```
host1# ls /sys/class/net
eth0 ib0 ib0.8002 ib1 lo sit0
```

Step 5 Enter the `ifconfig -a` command to verify that the interface was created, as shown in this example:

```
host1# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:30:48:20:D5:D1
          inet addr:172.29.237.206  Bcast:172.29.239.255  Mask:255.255.252.0
          inet6 addr: fe80::230:48ff:fe20:d5d1/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:17591 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4831 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8704196 (8.3 MiB)  TX bytes:446771 (436.2 KiB)
          Base address:0x3040  Memory:dd420000-dd440000
ib0       Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::205:ad00:20:849/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:2044  Metric:1
          RX packets:46 errors:0 dropped:0 overruns:0 frame:0
          TX packets:47 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:128
          RX bytes:45056 (44.0 KiB)  TX bytes:3011 (2.9 KiB)
ib0.8002  Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          BROADCAST MULTICAST  MTU:2044  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:128
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
ib1       Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          BROADCAST MULTICAST  MTU:2044  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:128
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:136 errors:0 dropped:0 overruns:0 frame:0
```

```

TX packets:136 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:9152 (8.9 KiB) TX bytes:9152 (8.9 KiB)
sit0 Link encap:IPv6-in-IPv4
NOARP MTU:1480 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)

```

Verify that you see the `ib0.8002` output.

- Step 6** Configure the new interface just as you would the parent interface. (See the “[Manually Configure IPoIB for Default IB Partition](#)” section on page 3-2.)

This example shows how to configure the new interface:

```
host1# ifconfig ib0.8002 192.168.12.1 netmask 255.255.255.0
```

Remove a Subinterface Associated with a Specific IB Partition

This section describes how to remove a subinterface.

To remove a subinterface, perform the following steps:

- Step 1** Take the subinterface offline. You cannot remove a subinterface until you bring it down.

This example shows how to take the subinterface offline:

```
host1# ifconfig ib0.8002 down
```

- Step 2** Remove the value of the partition key to the file as root user.

This example shows how to remove the partition 80:02 from the primary interface `ib0`:

```
host1# echo 0x8002 >> /sys/class/net/ib0/delete_child
```

- Step 3** (Optional) Enter the `ifconfig -a` command to verify that the subinterface no longer appears in the interface list.

This example shows how to verify that the subinterface no longer appears in the interface list:

```

host1# ifconfig -a
eth0 Link encap:Ethernet HWaddr 00:30:48:20:D5:D1
      inet addr:172.29.237.206 Bcast:172.29.239.255 Mask:255.255.252.0
      inet6 addr: fe80::230:48ff:fe20:d5d1/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:21431 errors:0 dropped:0 overruns:0 frame:0
      TX packets:5474 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:9542238 (9.1 MiB) TX bytes:562793 (549.6 KiB)
      Base address:0x3040 Memory:dd420000-dd440000
ib0 Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
-00
      inet addr:192.168.0.1 Bcast:192.168.0.255 Mask:255.255.255.0
      inet6 addr: fe80::205:ad00:20:849/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:2044 Metric:1
      RX packets:46 errors:0 dropped:0 overruns:0 frame:0
      TX packets:47 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:128
      RX bytes:45056 (44.0 KiB) TX bytes:3011 (2.9 KiB)
ib1 Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00

```

```

-00          BROADCAST MULTICAST  MTU:2044  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:128
            RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING  MTU:16436  Metric:1
            RX packets:136 errors:0 dropped:0 overruns:0 frame:0
            TX packets:136 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:9152 (8.9 KiB)  TX bytes:9152 (8.9 KiB)
sit0       Link encap:IPv6-in-IPv4
            NOARP  MTU:1480  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

```

Verify IPoIB Functionality

This section displays how to verify your configuration and your IPoIB functionality.

Step 1 Log in to your hosts.

Step 2 Verify the IPoIB functionality by using the **ifconfig** command.

This example shows how two IB nodes are used to verify IPoIB functionality. In this example, IB node 1 is at 192.168.0.1, and IB node 2 is at 192.168.0.2.

```

host1# ifconfig ib0 192.168.0.1 netmask 255.255.252.0
host2# ifconfig ib0 192.168.0.2 netmask 255.255.252.0

```

Step 3 Enter the **ping** command from 192.168.0.1 to 192.168.0.2.

```

host1# ping -c 5 192.168.0.2
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.
64 bytes from 192.168.0.2: icmp_seq=0 ttl=64 time=0.079 ms
64 bytes from 192.168.0.2: icmp_seq=1 ttl=64 time=0.044 ms
64 bytes from 192.168.0.2: icmp_seq=2 ttl=64 time=0.055 ms
64 bytes from 192.168.0.2: icmp_seq=3 ttl=64 time=0.049 ms
64 bytes from 192.168.0.2: icmp_seq=4 ttl=64 time=0.065 ms

--- 192.168.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 399ms rtt min/avg/max/mdev =
0.044/0.058/0.079/0.014 ms, pipe 2

```

IPoIB Performance

This section describes how to verify IPoIB performance by running the Bandwidth test and the Latency test. These tests are described in detail at this URL:

<http://www.netperf.org/netperf/training/Netperf.html>

Step 1 Download Netperf from this URL:

<http://www.netperf.org/netperf/NetperfPage.html>.

Step 2 Follow the instructions at <http://www.netperf.org/netperf/NetperfPage.html> to compile Netperf.

Step 3 Start the Netperf server.

This example shows how to start the Netperf server:

```
host1% netserver
Starting netserver at port 12865
Starting netserver at hostname 0.0.0.0 port 12865 and family AF_UNSPEC
host1%
```

Step 4 Run the Netperf client. The default test is the Bandwidth test.

This example shows how to run the Netperf client, which starts the Bandwidth test by default:

```
host2% netperf -H 192.168.0.1 -c -C -- -m 65536
TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.0.1 (192.168.0.1)
port 0 AF_INET
Recv  Send  Send           Utilization      Service Demand
Socket Socket Message Elapsed          Send  Recv  Send  Recv
Size  Size  Size    Time    Throughput  local  remote  local  remote
bytes bytes bytes   secs.  10^6bits/s  % S    % S    us/KB  us/KB
      87380 16384 65536   10.00    2903.14  25.29  25.64  2.855  2.894
```



Note You must specify the IPoIB IP address when running the Netperf client.

The following list describes parameters for the **netperf** command:

-H	Where to find the server
192.168.0.1	IPoIB IP address
-c	Client CPU utilization
-C	Server CPU utilization
--	Separates the global and test-specific parameters
-m	Message size, which is 65536 in the example above

The notable performance values in the example above are as follows:

Throughput is 2.90 gigabits per second.

Client CPU utilization is 25.29 percent of client CPU.

Server CPU utilization is 25.64 percent of server CPU.

Step 5 Run the Netperf Latency test.

Run the test once, and stop the server so that it does not repeat the test.

This example shows how to run the Latency test and then stop the Netperf server:

```
host2% netperf -H 192.168.0.1 -c -C -t TCP_RR -- -r 1,1
```

```

TCP REQUEST/RESPONSE TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.0.1
(192.168.0.1) port 0 AF_INET
Local /Remote
Socket Size   Request Resp.   Elapsed Trans.   CPU    CPU    S.dem  S.dem
Send  Recv  Size   Size   Time    Rate    local  remote local  remote
bytes bytes bytes  bytes  secs.   per sec % S    % S    us/Tr  us/Tr

16384 87380 1       1       10.00  30927.36 13.06  13.82 16.896 17.878
16384 87380

Stop netperf server.
host1% pskill netserver

```

The following list describes parameters for the **netperf** command:

-H	Where to find the server
192.168.0.1	IPoIB IP address
-c	Client CPU utilization
-C	Server CPU utilization
-t	Test type
TCP_RR	TCP required response test
--	Separates the global and test-specific parameters
-r 1,1	The request size sent and how many bytes requested back

The notable performance values in the example above are as follows:

Client CPU utilization is 13.06 percent of client CPU.

Server CPU utilization is 13.82 percent of server CPU.

Latency is 16.896 microseconds and 17.878 microseconds.

Sample Startup Configuration File

IP addresses that are configured manually are not persistent across reboots. You must use a configuration file to configure IPoIB when the host boots. The sample configuration below shows an example file named `ifcfg-ib0` that resides on a Linux host in `/etc/sysconfig/networks-scripts/` on RHEL4 and in `/etc/sysconfig/network/` on SLES10. The configuration file configures an IP address at boot time.

```

host1# cat > /etc/sysconfig/network-scripts/ifcfg-ib0 << EOF
> DEVICE=ib0
> BOOTPROTO=static
> IPADDR=192.168.0.1
> NETMASK=255.255.255.0
> ONBOOT=yes
> EOF

```




SCSI RDMA Protocol

The following sections appear in this chapter:

- [Introduction, page 4-1](#)
- [Configure SRP, page 4-1](#)
- [Verify SRP, page 4-7](#)

Introduction

SRP runs SCSI commands across RDMA-capable networks so that IB hosts can communicate with Fibre Channel storage devices and IB-attached storage devices. SRP requires a server switch with a Fibre Channel Gateway to connect the host to Fibre Channel storage. In conjunction with a server switch, SRP disguises IB-attached hosts as Fibre Channel-attached hosts. The topology transparency feature enables Fibre Channel storage to communicate seamlessly with IB-attached hosts, called SRP hosts.

To connect an IB-attached SRP host to a SAN, cable your SRP host to an IB fabric that includes a server switch with an Fibre Channel Gateway or IB-attached storage. Log in to the server switch to configure the Fibre Channel connection between the SAN and the SRP host, and then log in to the host and configure the SRP host.



Note

See the [“Root and Non-root Conventions in Examples”](#) section on page 1-4 for details about the significance of prompts used in the examples in this chapter.

Configure SRP

We provide a number of ways to configure the connection between the SAN and the SRP host. The method that you choose depends on the interfaces available to you and the global access settings on your server switch. The instructions in this section provide one example of how to configure the connection. For detailed instructions, refer to the *Cisco SFS 3012 InfiniBand Multifabric Server Fibre Channel Gateway User Guide*.



Note

If you have a Fibre Channel Gateway, you must configure ITLs. If you have IB-attached storage, refer to the relevant storage documentation.

To configure your IB fabric to connect an SRP host to a SAN, follow the instructions in the following sections:

- [Configure ITLs when Using Fibre Channel Gateway, page 4-2](#)
- [Configure SRP Host, page 4-5](#)

**Note**

If you intend to manage your environment with Cisco VFrame software, do not configure ITL.

Configure ITLs when Using Fibre Channel Gateway

When you configure initiators, you assign Fibre Channel WWNNs to SRP hosts so that the SAN can recognize the hosts. Steps to configure initiators are provided in this section below.

To configure initiators that you have not yet connected to your fabric, enter the GloballyUnique Identifier GUID of the initiator into the CLI or Element Manager so that the configuration works when you connect the SRP host.

You must configure ITLs for your initiators to communicate with your storage. You can configure ITLs with the CLI or the Element Manager GUI.

- If you restricted port and LUN access when you configured global attributes, proceed to the [“Configure ITLs with Element Manager while Global Policy Restrictions Apply”](#) section on page 4-3.
- If you have not configured access, perform the steps as appropriate in [“Configure ITLs with Element Manager while No Global Policy Restrictions Apply”](#) section on page 4-2 or in [“Configure ITLs with Element Manager while Global Policy Restrictions Apply”](#) section on page 4-3.

**Note**

If you enter an Fibre Channel command and receive an error message that reads `Operation temporarily failed - try again`, give your Fibre Channel gateway time to finish initializing, and then retry the command.

Configure ITLs with Element Manager while No Global Policy Restrictions Apply

To configure ITLs with a Linux SRP host while your port masking and LUN masking policies are unrestricted, perform the following steps:

-
- Step 1** Log in to your host.
- Step 2** Enter the `hca_self_test | grep -i guid` command to display the host GUID.

**Note**

Record the GUID value. You are required to enter it repeatedly.

- Step 3** To bring up the Fibre Channel Gateways on your server switch, perform the following steps:
- Launch Element Manager.
 - Double-click the Fibre Channel Gateway card that you want to bring up. The Fibre Channel Card window opens.
 - Click the **Up** radio button in the **Enable/Disable Card** field, and then click **Apply**.
 - (Optional) Repeat this process for additional gateways.

The Fibre Channel Gateway automatically discovers all attached storage.



Note Discovered LUs remain gray (inactive) until an SRP host connects to them. Once a host connects to an LU, its icon becomes blue (active). Hosts do not stay continually connected to LUs, so the color of the icon may change.

- Step 4** From the Fibre Channel menu of the Element Manager, select **Storage Manager**. The Cisco Storage Manager window opens.
- Step 5** Click the **SRP Hosts** folder in the **Storage** navigation tree in the left-hand frame of the interface. The SRP Hosts display appears in the right-hand frame of the interface.
- Step 6** Click **Define New** in the SRP Hosts display. The Define New SRP Host window opens.



Note If your host includes multiple HCAs, you must configure each individual HCA as an initiator. When you configure one HCA in a host, other HCAs in the host are not automatically configured.

- Step 7** Select a GUID from the Host GUID pulldown menu in the Define New SRP Host window. The menu displays the GUIDs of all connected hosts that you have not yet configured as initiators.
- Step 8** (Optional) Type a description in the Description field in the Define New SRP Host window.
- Step 9** Click the **Next >** button. The Define New SRP Host window displays a recommended WWNN for the host and recommended WWPNNs that represent the host on all existing and potential Fibre Channel Gateway ports.



Note Although you can manually configure the WWNN or WWPNNs, use the default values to avoid conflicts.

- Step 10** Click the **Finish** button. The new host appears in the SRP Hosts display.
- Step 11** Expand the **SRP Hosts** folder in the **Storage** navigation tree, and then click the host that you created. The host display appears in the right-hand frame of the interface.
- Step 12** (Optional) Click the **LUN Access** tab in the host display, and then click **Discover LUNs**. The targets and associated LUNs that your Fibre Channel Gateway sees appear in the Accessible LUNs field.
- Step 13** Click **Refresh** in the Cisco Storage Manager window.

Configure ITLs with Element Manager while Global Policy Restrictions Apply

These instructions apply to environments where the portmask policy and LUN masking policy are both restricted. To verify that you have restricted your policies, enter the **show fc srp-global** command at the CLI prompt. View the **default-gateway-portmask-policy** and **default-lun-policy** fields. If restrictions apply to either field, **restricted** appears in the field output.

To configure ITLs with a Linux SRP host while your port masking and LUN masking policies are restricted, perform the following steps:

Step 1 Log in to your host.

Step 2 Enter the `hca_self_test | grep -i guid` command at the host CLI to display the host GUID.



Note Record the GUID value. You are required to enter it repeatedly.

Step 3 Bring up the Fibre Channel Gateways on your Server Switch with the following steps:

- a. Launch Element Manager.
- b. Double-click the Fibre Channel Gateway card that you want to bring up. The Fibre Channel Card window opens.
- c. Click the **Up** radio button in the **Enable/Disable Card** field, and then click **Apply**.
- d. (Optional) Repeat this process for additional gateways.

The Fibre Channel Gateway automatically discovers all attached storage.



Note Discovered LUs remain gray (inactive) until an SRP host connects to them. Once a host connects to an LU, its icon becomes blue (active).

Step 4 From the Fibre Channel menu, select **Storage Manager**.

Step 5 Click the **SRP Hosts** folder in the **Storage** navigation tree in the left-hand frame of the interface. The SRP Hosts display appears in the right-hand frame of the interface.

Step 6 Click Define New in the SRP Hosts display. The Define New SRP Host window opens.



Note If your host includes multiple HCAs, you must configure each individual HCA as an initiator. When you configure one HCA in a host, any other HCAs in the host are not automatically configured.

Step 7 Select a GUID from the Host GUID pulldown menu in the Define New SRP Host window. The menu displays the GUIDs of all available hosts that you have not yet configured as initiators.

Step 8 (Optional) Type a description in the Description field in the Define New SRP Host window. If you do not enter a description, your device will assign a description.

Step 9 Click the **Next >** button. The Define New SRP Host window displays a recommended WWNN for the host and recommended WWPNNs that will represent the host on all existing and potential Fibre Channel Gateway ports.



Note Although you can manually configure the WWNN or WWPNNs, we recommend that you use the default values to avoid conflicts.

Step 10 Click **Finish**. The new host appears in the SRP Hosts display.

Step 11 Expand the **SRP Hosts** folder in the **Storage** navigation tree, then click the host that you created. The host display appears in the right-hand frame of the interface.

- Step 12** Click the **Targets** tab in the host display. Double-click the WWPN of the target that you want your host to access. The IT Properties window opens.
 - Step 13** Click the ... button next to the Port Mask field. The Select Port(s) window opens and displays two port numbers for each slot in the chassis. The *raised* port numbers represent restricted ports. The *pressed* port numbers represent accessible ports.
 - Step 14** Click the port(s) to which the SAN connects to grant the initiator access to the target through those ports, and then click **OK**.
 - Step 15** Click the **Apply** button in the IT Properties window, and then close the window.
 - Step 16** Click the **LUN Access** tab in the host display, and then click **Discover LUNs**. The targets and associated LUNs that your Fibre Channel Gateway sees appear in the Available LUNs field.
 - Step 17** Click the **LUN Access** tab, click the target that you configured in [step 16](#), and then click **Add >**. The target and its LUN(s) appear in the Accessible LUNs field in an **Inactive ITLs** folder.
 - Step 18** Click the LUN that you want your host to reach, and then click **Edit ITL Properties**. The ITL Properties window opens.
 - Step 19** Click the ... button next to the **Port Mask** field. The Select Port(s) window opens and displays two port numbers for each slot in the chassis. The *raised* port numbers represent restricted ports. The *pressed* port numbers represent accessible ports.
 - Step 20** Click the port(s) to which the SAN connects to grant the initiator access to the target through those ports, and then click the **OK** button.
 - Step 21** Click the **Refresh** button in the Cisco Storage Manager window.
-

Configure SRP Host

Configuring SRP requires associating IB ports with Fibre Channel targets. You configure SRP by discovering SRP targets and connecting IB ports to access them. If there are multiple Fibre Channel paths or IB paths to a target, then third-party multipathing software can be used to provide high availability and load balancing.

The SRP driver is not loaded at boot time by default. To enable SRP driver, change `srp_load=no` to `srp_load=yes` in `/etc/infiniband/openib.conf`. To load the SRP driver manually, run `modprobe ib_srp`.

Step 1 The program `ibsrpdm` displays the available SRP targets.

This example shows a display of SRP targets when the host is connected to a RAID array gateway.

```
host1# ibsrpdm
IO Unit Info:
port LID: 000b
port GUID: fe800000000000000005ad00000013e9
change ID: 80b0
max controllers: 0x01
controller[ 1]
GUID: 0005ad00000013e7
vendor ID: 0005ad
device ID: 0005ad
IO class : 0100
ID: Topspin SRP/FC TCA
service entries: 2
service[ 0]: 0000000000000066 / SRP.T10:20030003BA27CC7A
service[ 1]: 0000000000000066 / SRP.T10:20030003BA27CF53
```

If the Fibre Channel Gateway is not configured correctly or the SRP driver is not loaded, zero service entries display.

This is an example that shows zero service entries:

```
host1# ibsrpdm
IO Unit Info:
port LID:      000b
port GUID:    fe800000000000000005ad00000013e9
change ID:    6d20
max controllers: 0x01

controller[ 1]
GUID:        0005ad00000013e7
vendor ID:   0005ad
device ID:   0005ad
IO class :   0100
ID:          Topspin SRP/FC TCA
service entries: 0
```

Step 2 Check for SCSI disks before configuring SRP.

This example shows how to check for SCSI disk:

```
host1# cat /proc/scsi/scsi
Attached devices:
Host: scsi0 Channel: 00 Id: 01 Lun: 00
Vendor: SEAGATE Model: ST373307LC Rev: 0006
Type: Direct-Access ANSI SCSI revision: 03
Host: scsi0 Channel: 00 Id: 06 Lun: 00
Vendor: SDR Model: GEM318P Rev: 1
Type: Processor ANSI SCSI revision: 02
```

The above example shows one local Seagate Model ST373307LC SCSI disk.

Step 3 Configure an IB port to access all SRP targets.

A script is provided to configure an IB port to access all SRP targets. It requires one optional command-line parameter, the IB port name for SRP.

```
host1# ls /sys/class/infiniband_srp
srp-mthca0-1 srp-mthca0-2
host1# cisco_srp_add_targets srp-mthca0-1
Attaching SRP target id_ext=20030003BA27CC7A to IB interface srp-mthca0-1
Attaching SRP target id_ext=20030003BA27CF53 to IB interface srp-mthca0-1
```

Step 4 Check for SCSI disks after configuring SRP.

This example checks for SCSI disks after configuring SRP:

```
host1# cat /proc/scsi/scsi
Attached devices:
Host: scsi0 Channel: 00 Id: 01 Lun: 00
  Vendor: SEAGATE Model: ST373307LC Rev: 0006
  Type: Direct-Access ANSI SCSI revision: 03
Host: scsi0 Channel: 00 Id: 06 Lun: 00
  Vendor: SDR Model: GEM318P Rev: 1
  Type: Processor ANSI SCSI revision: 02
Host: scsi1 Channel: 00 Id: 00 Lun: 31
  Vendor: SUN Model: T4 Rev: 0300
  Type: Direct-Access ANSI SCSI revision: 03
Host: scsi1 Channel: 00 Id: 00 Lun: 32
  Vendor: SUN Model: T4 Rev: 0300
  Type: Direct-Access ANSI SCSI revision: 03
```

Two additional Sun Model T4 SRP LUNs are available after the configuration is complete.

Verify SRP

This section describes how to verify SRP functionality and verify SRP host-to-storage connections with the Element Manager GUI.

Verify SRP Functionality

To verify SRP functionality, perform the following steps:

Step 1 Log in to your SRP host.

Step 2 Create a disk partition.

This example shows how to partition a disk by using approximately half of the first SRP disk:

```
host1# fdisk /dev/sdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel. Changes will remain in memory only,
until you decide to write them. After that, of course, the previous
content won't be recoverable.
```

```
The number of cylinders for this disk is set to 8200.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
```

```

2) booting and partitioning software from other OSs
(e.g., DOS FDISK, OS/2 FDISK)
Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)
Command (m for help): p
Disk /dev/sdb: 8598 MB, 8598847488 bytes
64 heads, 32 sectors/track, 8200 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes
Device Boot Start End Blocks Id System
Command (m for help): n
Command action
e extended
p primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-8200, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-8200, default 8200): 4000
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.

```

Step 3 Create a filesystem on the partition.

This example shows how to create a filesystem on the partition:

```

host1 # mke2fs -j /dev/sdb1
mke2fs 1.35 (28-Feb-2004)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
512000 inodes, 1023996 blocks
51199 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=1048576000
32 block groups
32768 blocks per group, 32768 fragments per group
16000 inodes per group
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 38 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
host1# mount /dev/sdb1 /mnt
host1# df -k

```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/sda3	68437272	7811640	57149168	13%	/
/dev/sda1	101086	13159	82708	14%	/boot
none	3695248	0	3695248	0%	/dev/shm
sjc-filer25a.cisco.com:/data/home	1310720000	1217139840	93580160	93%	/data/home
sjc-filer25a.cisco.com:/software	943718400	839030128	104688272	89%	/data/software
sjc-filer25b.cisco.com:/qadata	1353442040	996454024	356988016	74%	/qadata
/dev/sdb1	4031664	40800	3786068	2%	/mnt

Step 4 Write some data to the filesystem.

This example shows how to write some data to the filesystem:

```
host1# dd if=/dev/zero of=/mnt/dd.test count=1000
1000+0 records in
1000+0 records out
host1# ls -l /mnt/dd.test
-rw-r--r-- 1 root root 512000 Jul 25 13:25 /mnt/dd.test
```

Verify with Element Manager

To verify that your host connects successfully to Fibre Channel storage, perform the following steps:

-
- Step 1** Launch Element Manager, and log in to the server switch that connects your SRP host to Fibre Channel storage.
- Step 2** Click the **FibreChannel** menu, and select **Storage Manager**. The Storage Manager window opens.
- Step 3** Expand the SRP hosts folder in the **Storage** navigation tree. A list of SRP hosts appears. Those SRP hosts that are successfully connected to storage appear as blue icons.
- Step 4** (Optional) Verify LUN access with the following steps:
- Click an SRP host in the **Storage** navigation tree.
 - Click the **LUN Access** tab in the right-hand frame of the display.
 - Expand all icons in the Accessible LUNs field. Those SRP hosts that are successfully connected to LUNs appear as blue LUN icons.
-



Sockets Direct Protocol

The following sections appear in this chapter:

- [Introduction, page 5-1](#)
- [Configure IPoIB Interfaces, page 5-1.](#)
- [Convert Sockets-Based Application, page 5-1](#)
- [SDP Performance, page 5-4](#)
- [Netperf Server with IPoIB and SDP, page 5-6](#)

Introduction

SDP is an IB-specific upper layer protocol. It defines a standard wire protocol to support stream sockets networking over IB. SDP enables sockets-based applications to take advantage of the enhanced performance features provided by IB and achieves lower latency and higher bandwidth than IPoIB running sockets-based applications. It provides a high-performance, zero-copy data transfer protocol for stream-socket networking over an IB fabric. You can configure the driver to automatically translate TCP to SDP based on source IP, destination, or application name.



Note

See the [“Root and Non-root Conventions in Examples”](#) section on page 1-4 for details about the significance of prompts used in the examples in this chapter.

Configure IPoIB Interfaces

SDP uses the same IP addresses and interface names as IPoIB. Configure the IPoIB IP interfaces if you have not already done so. (See [Chapter 3, “IP over IB Protocol.”](#))

Convert Sockets-Based Application

You can convert your socket-based applications to use SDP instead of TCP by using one of two conversion types. These conversion types are as follows:

- [Explicit/Source Code Conversion Type, page 5-2](#)
- [Automatic Conversion Type, page 5-2](#)

Explicit/Source Code Conversion Type

The explicit or source code conversion type method converts sockets to use SDP based on application source code. This method is useful when you want full control from your application when using SDP.

To employ this method, change your source code to use `AF_INET_SDP` instead of `AF_INET` when calling the `socket()` system call.

`AF_INET_SDP` is defined as 27. Add the following line of code to the beginning of your program:

```
#define AF_INET_SDP 27
```

Automatic Conversion Type

Use a text editor to open the `libsdp` configuration file (located in `/usr/local/ofed/etc/libsdp.conf`). This file defines when to automatically use SDP instead of TCP. You may edit this file to specify connection overrides. Use the environment variable `libsdp` configuration file to specify an alternate configuration file.

The automatic conversion type method converts socket streams based upon a destination port, listening port, or program name.

Load the installed `libsdp.so` library using either of these two methods:

- Set the `LD_PRELOAD` environment variable to `libsdp.so` before running the executable.
- Add the full path of the library into `/etc/ld.so.preload`. This action causes the library to preload for every executable that is linked with `libc`.

This configuration file supports two main types of statements:

- **log**
The **log** keyword sets logging related configurations. The log settings take immediate effect, so they are defined at the beginning of the file.
- **use**
The **use** keyword defines the address family used for the sockets that specifies a rule for when to use SDP.

Log Statement

The log directive allows the user to specify which debug and error messages are sent and where they are sent. The log statement format is as follows:

```
log [destination stderr | syslog | file filename] [min-level 1-9]
```

Command	Description
destination	Defines the destination of the log messages.
stderr	Forwards messages to the <code>STDERR</code> .
syslog	Sends messages to the <code>syslog</code> service.

Command	Description
<i>file filename</i>	Writes messages to the file /tmp/filename.
<i>min-level</i>	Defines the verbosity of the log as follows: 9—Errors are printed. 8—Warnings. 7—Connect and listen summary. Track SDP usage 2—Function calls and return values. 1—Debug messages.

1. This example shows how to get the full verbosity printed into the /tmp/libsdp.log file:

```
log min-level 1 destination file libsdp.log
```

2. This example shows how to get the full verbosity printed into the STDERR:

```
log min-level 1 destination stderr
```

The default behavior is set at a min-level of 9.

Use Statement

The socket control statements allow users to specify when libsdp replaces AF_INET/SOCK_STREAM sockets with AF_INET_SDP/SOCK_STREAM sockets. Each control statement specifies a rule for when to use SDP.

The command statements that control what type of sockets to open contain the following elements:

use *address-family* *role* *program-name* *address:port range*

The syntax description for the socket control command statement is as follows:

<i>address-family</i>	This argument can be one of the following values: Type <i>sdp</i> to specify when to use an SDP. Type <i>tcp</i> to specify when not to match the SDP socket. Type <i>both</i> to specify when to use both SDP and AF_INET sockets. Note The semantics for <i>both</i> is different between the <i>server</i> and <i>client</i> roles. In the case of a server, it means that the server is listening on both SDP and TCP. In the case of a client, the connect prefers using SDP but silently falls back to TCP if the SDP connection fails.
<i>role</i>	This keyword defines the listening port address family that is one of server or listener. Alternatively, this keyword defines the connected port address family that is one of client or connect.
<i>program-name</i>	This argument defines the program name, not including the path, to which the rule applies. Note Wildcards with same semantics as <i>ls</i> are supported (* and ?). For example, db2* matches any program with a name starting with db2 and t?cp matches on tcp. If no default is set, the statement matches all programs.

<i>address</i>	<p>This argument is the local address to which the server is bound or the remote server address to which the client connects. The syntax for address matching is as follows:</p> <p style="text-align: center;"><i>IPv4 address</i> [<i>prefix_length</i>] *</p> <p><i>IPv4 address</i> = [0-9]+\.[0-9]+\.[0-9]+\.[0-9]+ each sub number < 255 <i>prefix_length</i> = [0-9]+ and with value <= 32.</p> <p>Note A <i>prefix_length</i> of 24 matches the subnet mask 255.255.255.0. A <i>prefix_length</i> of 32 requires an exact IP match.</p>
<i>port range</i>	<p><i>start-port</i> [-<i>end-port</i>]</p> <p>Valid values for port numbers are >0 and < 65536.</p> <p>Note Rules are evaluated in order of definition, and the first match wins.</p>

1. This example shows how SDP is used by clients connecting to devices that belong to the subnet 192.168.1.* :

```
# family role program address:port [-range]
use sdp connect * 192.168.1.0/24:*
```

2. This example shows how SDP is used by `ttcp` when it connects to port 5001 of any device:

```
# family role program address:port [-range]
use sdp listen ttcp *:5001
```

3. This example shows how to use TCP for any program starting with `ttcp*` serving ports 22 to 25:

```
# family role program address:port [-range]
use tcp server ttcp* *:22-25
```

4. This example shows how both TCP and SDP are used by any server to listen on port 8080:

```
# family role program address:port [-range]
use both server * *:8080
```

5. This example shows how to connect SSH through SDP and revert to TCP to hosts on 11.4.8.* port 22:

```
# family role program address:port [-range]
use both connect * 11.4.8.0/24:22
```

**Note**

If all *use* rules are commented, SDP takes the simple SDP mode and uses SDP for all connections.

SDP Performance

This section describes how to verify SDP performance by running the Netperf Bandwidth test and the Latency test. These tests are described in detail at this URL:
<http://www.netperf.org/netperf/training/Netperf.html>

-
- Step 1** Download Netperf from this URL:

<http://www.netperf.org/netperf/NetperfPage.html>.

- Step 2** Follow the instructions at <http://www.netperf.org/netperf/NetperfPage.html> to compile Netperf.

Step 3 Run the Netperf server, forcing SDP to be used instead of TCP.

This example shows how to run the Netperf server with SDP:

```
host1% LD_PRELOAD=libsdp.so netserver
Starting netserver at port 12865
Starting netserver at hostname 0.0.0.0 port 12865 and family AF_UNSPEC
host1%
```

Step 4 Run the Netperf Bandwidth test, forcing SDP to be used instead of TCP.

This example shows how to run the Netperf Bandwidth test with SDP:

```
host2% LD_PRELOAD=libsdp.so netperf -H 192.168.0.1 -c -C -- -m 65536
TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.0.206 (192.168.
0.206) port 0 AF_INET
Recv  Send  Send  Utilization  Service Demand
Socket Socket Message Elapsed  Send  Recv  Send  Recv
Size  Size  Size  Time  Throughput  local  remote  local  remote
bytes bytes bytes secs.  10^6bits/s  % S  % S  us/KB  us/KB

135168 135168 65536 10.00 7124.42 26.37 27.32 1.213 1.256
```

The following list describes parameters for the **netperf** command:

-H	Where to find the server
192.168.0.1	IPoIB IP address
-c	Client CPU utilization
-C	Server CPU utilization
--	Separates the global and test-specific parameters
-m	The message size, which is 65536 in the example above

The notable performance values in the example above are as follows:

Throughput is 7.12 gigabits per second.

Client CPU utilization is 26.37 percent of the client CPU.

Server CPU utilization is 27.32 percent of the server CPU.

Step 5 Run the Netperf Latency test, forcing SDP to be used instead of TCP.

After the test runs once, stop the server so that it does not repeat the test.

This example shows how to run the Netperf Latency test with SDP:

```
host2% LD_PRELOAD=libsdp.so netperf -H 192.168.0.1 -c -C -t TCP_RR -- -r 1,1
TCP REQUEST/RESPONSE TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.0.206
(192.168.0.206) port 0 AF_INET
Local /Remote
Socket Size Request Resp. Elapsed Trans. CPU CPU S.dem S.dem
Send Recv Size Size Time Rate local remote local remote
bytes bytes bytes bytes secs. per sec % S % S us/Tr us/Tr

135168 135168 1 1 10.00 55506.40 26.25 29.00 18.914 20.896
135168 135168
Stop netperf server.
host1% pkill netserver
```

The following list describes parameters for the **netperf** command:

-H	Where to find the server
192.168.0.1	IPoIB IP address
-c	Client CPU utilization
-C	Server CPU utilization
-t	Test type
TCP_RR	TCP request response test
--	Separates the global and test-specific parameters
-r 1,1	Request size sent and how many bytes requested back

The notable performance values in the example above are as follows:

Client CPU utilization is 26.25 percent of client CPU.

Server CPU utilization is 29.00 percent of server CPU.

Latency is 18.914 microseconds and 20.896 microseconds.

Netperf Server with IPoIB and SDP

When using libsdp, it is possible for the Netperf server to work with both IPoIB and SDP. This section describes how to use the Netperf server with IPoIB as well as SDP

Step 1 Create the libsdp configuration file.

This example shows how to create the libsdp configuration file:

```
host1% echo "use both server netserver *.*" > $HOME/both.conf
```

Step 2 Ensure that netserver is not running already, and then start netserver.

This example stops the Netperf server if it is already running and then starts the server:

```
host1% pkill netserver
host1% LD_PRELOAD=libsdp.so LIBSDP_CONFIG_FILE=$HOME/both.conf netserver
Starting netserver at port 12865
Starting netserver at hostname 0.0.0.0 port 12865 and family AF_UNSPEC
```

Step 3 Run the Netperf Bandwidth test, forcing SDP to be used instead of TCP.

This example shows how to run the Netperf Bandwidth test with SDP:

```
host2% LD_PRELOAD=libsdp.so netperf -H 192.168.0.1 -c -C -- -m 65536
TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.0.206 (192.168.0.206) port 0 AF_INET
Recv  Send  Send
Socket Socket Message Elapsed Utilization Service Demand
Size Size Size Time Throughput Send Recv Send Recv
bytes bytes bytes secs. 10^6bits/s % S % S us/KB us/KB

135168 135168 65536 10.00 7124.42 26.37 27.32 1.213 1.256
```

The following list describes parameters for the **netperf** command:

-H	Where to find the server
192.168.0.1	IPoIB IP address
-c	Client CPU utilization
-C	Server CPU utilization
--	Separates the global and test-specific parameters
-m	The message size, which is 65536 in the example above

The notable performance values in the example above are as follows:

Throughput is 7.12 bits per second.

Client CPU utilization is 26.37 percent of client CPU.

Server CPU utilization is 27.32 percent of server CPU.

Step 4 Run the Netperf client.

The default test is the Bandwidth test.

This example shows how to run the Netperf client, which starts the Bandwidth test by default:

```
host2% netperf -H 192.168.0.1 -c -C -- -m 65536
TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.0.1 (192.168.0.1)
port 0 AF_INET
Recv  Send  Send  Utilization  Service Demand
Socket Socket Message  Elapsed      Send  Recv  Send  Recv
Size  Size  Size  Time  Throughput  local  remote  local  remote
bytes bytes bytes  secs.  10^6bits/s  % S   % S   us/KB  us/KB

      87380 16384 65536   10.00    2903.14  25.29  25.64   2.855  2.894
```



Note You must specify the IPoIB IP address when running the Netperf client.

The following list describes parameters for the **netperf** command:

-H	Where to find the server
192.168.0.1	IPoIB IP address
-c	Client CPU utilization
-C	Server CPU utilization
--	Separates the global and test-specific parameters
-m	Message size, which is 65536 in the example above

The notable performance values in the example above are as follows:

Throughput is 2.90 gigabits per second.

Client CPU utilization is 25.29 percent of client CPU.

Server CPU utilization is 25.64 percent of server CPU.



MVAPICH MPI and Open MPI

The following sections appear in this chapter:

- [Introduction, page 6-1](#)
- [Initial Setup, page 6-2](#)
- [Configure SSH, page 6-2](#)
- [Edit Environment Variables, page 6-5](#)
- [Perform MPI Bandwidth Test, page 6-8](#)
- [Perform MPI Latency Test, page 6-10](#)
- [Perform Intel MPI Benchmarks \(IMB\) Test, page 6-11](#)
- [Compile MPI Programs, page 6-14](#)

Introduction

The MPI is a standard library functionality in C, C++, and Fortran that can be used to implement a message passing program. MPI allows the coordination of a program running as multiple processes in a distributed memory environment.

This chapter includes setup and configuration information for MVAPICH MPI and Open MPI. MVAPICH MPI and Open MPI support both the GNU and Intel compiler suites. Each of these compiler suites, in turn, support the C, C++, Fortran77, and Fortran90 programming languages.

For additional details about MPI, go to these URLs:

<http://webct.ncsa.uiuc.edu:8900/public/MPI/>

and

<http://www.mpi-forum.org>

For additional details about MVAPICH MPI, go to this URL:

<http://nowlab.cse.ohio-state.edu/projects/mpi-iba/>

For additional details about Open MPI, go to this URL:

<http://www.open-mpi.org/>



Note

See the “[Root and Non-root Conventions in Examples](#)” section on page 1-4 for details about the significance of prompts used in the examples in this chapter.

Initial Setup

MPI requires that you be able to launch executables on remote hosts without manually entering a login name, password or passphrase. This procedure typically involves a one-time setup on one or more of the hosts that you wish to use.

Although many technologies are available to meet this requirement, this chapter describes one method: how to set up SSH for password-less logins.

Configure SSH

There are many ways to configure SSH to allow password-less logins. This section describes one way; your local policies or system administrators may advocate different ways. Any of them are sufficient as long as you can log in to remote nodes without manually entering a login name, password, or passphrase during the MPI run.

The example in this section distinguishes between passwords and passphrases. Passwords are associated with usernames and are normally used to login and/or authenticate a user on a node. SSH can be configured to login to remote nodes by using public key encryption to establish credentials on those nodes, making the use of passwords unnecessary. SSH keys can optionally be encrypted with passphrases, meaning that the keys cannot be accessed (and automated logins cannot be performed) without providing the proper passphrase, either by typing them in or caching them in a secure mechanism.

Since MPI requires fully automatic logins on remote nodes, typing of passphrases during the MPI run is disallowed. For simplicity, the text below describes how to setup SSH with a public key that uses no passphrase. Setting up SSH to use a cached passphrase is also permitted, but is not described in this document.



Note The instructions in this section assume that you have never set up SSH before and have no existing public or private keys. Additionally, the instructions assume that you always launch MPI jobs from a single host (host1 in the following example). If you have already used SSH with key-based authentication, you should not use this procedure because it overwrites your existing keys.

To configure SSH, perform the following steps:

Step 1 Log in to the host that you want to configure as the local host, host1.

This example shows how to log in to the host:

```
login: username
Password: password
host1$
```



Note Your exact login output is slightly different and could display information such as the day and the last time you logged in.

Step 2 Enter the `ssh-keygen -t dsa` command to generate a public/private DSA key pair. You are prompted for a folder in which to store the key.

This example shows how to generate a public/private DSA key pair:

```
host1$ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/username/.ssh/id_dsa):
```



Note In the above example, replace /home/username/ with the location of your home directory.

Step 3 Press the **Enter** key to store the key in the default directory.

This example shows how to store the key in the default directory:

```
Enter file in which to save the key (/home/username/.ssh/id_rsa):
Created directory '/home/username/.ssh'.
Enter passphrase (empty for no passphrase):
```



Note If you have used SSH before, you may not see the created directory message as displayed in the example above.

Step 4 Press the **Return** key to create an empty passphrase. You will be prompted to re-enter the passphrase. Press the **Return** key again.



Note Do not enter a passphrase!

This example shows how to create an empty passphrase:

```
Enter passphrase (empty for no passphrase): <hit Return>
Enter same passphrase again: <hit Return>
```

Upon success, a fingerprint of the generated key displays.

This example shows the display of the fingerprint of the host:

```
Your identification has been saved in /home/username/.ssh/id_dsa.
Your public key has been saved in /home/username/.ssh/id_dsa.pub.
The key fingerprint is:
0b:3e:27:86:0d:17:a6:cb:45:94:fb:f6:ff:ca:a2:00
host1$
```

Step 5 Change into the .ssh directory that you created.

This example shows how to change into the .ssh directory:

```
host1$ cd .ssh
```

Step 6 Copy the public key that was just generated to the authorized keys file.

This example shows how to copy the public key to authorized keys file:

```
host1$ cp id_dsa.pub authorized_keys
host1$ chmod 0600 authorized_keys
```

Step 7 Test your SSH connection to host1. You should be able to establish a SSH session to host1 without being prompted for a username, password or passphrase.

This example shows how to verify that you can establish a SSH session to host1 without being prompted for a password or passphrase:

```
host1$ ssh host1 hostname
host1
host1$
```



Note If this is the first time you have used SSH to log in to host1, you may see a message similar to the one below.

```
The authenticity of host 'host1 (10.0.0.1)' can't be established.
RSA key fingerprint is 6b:47:70:fb:6c:c1:a1:90:b9:30:93:75:c3:ee:a9:53.
Are you sure you want to continue connecting (yes/no)?
```

If you see this prompt, type **yes**, and press **Enter**. You may then see a message similar to this:

```
Warning: Permanently added 'host1' (RSA) to the list of known hosts.
```

Next, you see the host1 output and are returned to a shell prompt. You should see this authentication message only the first time you use SSH to connect to a particular host. For example, if you run **ssh host1 hostname** again, you do not see the authentication message again.



Note If your home directory is shared between all nodes through a network filesystem, skip ahead to [step 10](#).

Step 8 Log in to another host that you want to use with MPI, host2. Create a `.ssh` directory in your home directory on host2, and set its permissions to 0700.

This example shows how to create a `.ssh` directory in the root directory and set its permissions to 0700:

```
host2$ mkdir .ssh
host2$ chmod 0700 .ssh
```

Step 9 Return to host1 and copy the authorized keys file from [step 6](#) to the directory that you created in [step 8](#).

This example shows how to return to host1 and copy the authorized keys file to the directory created:

```
host1$ scp authorized_keys host2:~/.ssh
```



Note If this is the first time you have logged in to host2 using SSH or SCP, you see an authenticity message for host2. Type **yes** to continue connecting. You do not see the message when connecting from host1 to host2 again.

Upon success, you see output similar to the following:

```
host1$ scp authorized_keys host2:~/.ssh
username@host1's password:
authorized_keys                               100% 2465      2.4KB/s   00:00
The user will need to enter their password at the "username@host1's password:"
prompt.authorized_keys                         100% 2465      2.4KB/s   00:00
```

Step 10 Test your SSH connection. You should be able to log in to the remote node without being prompted for a username, password or passphrase.

This example shows how to test your SSH connection:

```
host1$ ssh host2 hostname
host2
host1$
```

Step 11 Repeat [step 8](#) through [step 10](#) for each host that you want to use with MPI.



Note Clear all the authenticity messages before continuing to repeat the steps.

Edit Environment Variables

It is easiest to use MPI if you edit some environment variables based on the MPI implementation that you are using. This procedure enables you to run commands without typing long executable filenames. There are three main methods:

- [Set Environment Variables in System-Wide Startup Files](#)
- [Edit Environment Variables In the Users' Shell Startup Files](#)
- [Edit Environment Variables Manually](#)

The following sections describe each of these methods.



Note You should set up only one MPI implementation in the environment. Setting multiple MPI implementations simultaneously in the environment can cause unexpected results.

Set Environment Variables in System-Wide Startup Files

This method is used to set a system-wide default for which MPI implementation is used. This method is the easiest for end users; users who log in automatically have MPI implementations set up for them without executing any special commands to find MPI executables, such as mpirun or mpicc.



Note You can have either MVAPICH or Open MPI as the default in the system-wide startup files. You cannot set up both MPI implementations as the default.

Set up MVAPICH in System-Wide Startup Files

This example shows how to make two system-wide shell startup files (one for Bourne shell variants and one for C shell variants) that set up all users to use MVAPICH. These commands must be run by the superuser on all nodes where MPI is used:

```
host1# echo 'export PATH=MPI_PATH:$PATH' > /etc/profile.d/mpi.sh
host1# echo 'set path = (MPI_PATH $path)' > /etc/profile.d/mpi.csh
host1# chmod 755 /etc/profile.d/mpi.sh /etc/profile.d/mpi.csh
```

Replace *MPI_PATH* with the directory name of the MVAICH that you wish to use. For example, to use MVAICH with the GNU compiler, use `/usr/local/ofed/mpl/gcc/mvapich-0.9.7-mlx2.2.0/bin` for *MPI_PATH*. If you want to use the Intel compiler, use `/usr/local/ofed/mpl/intel/mvapich-0.9.7-mlx2.2.0/bin` for *MPI_PATH*.

Set up Open MPI in System-Wide Startup Files

This example shows how to make two system-wide shell startup files (one for Bourne shell variants and one for C shell variants) that sets up all users to use Open MPI. These commands must be run by the superuser on all nodes where MPI is used:

```
host1# cat > /etc/profile.d/mpl.sh <<EOF
> export PATH=MPL_PATH/bin:\$PATH
> export LD_LIBRARY_PATH=MPL_PATH/lib:\$LD_LIBRARY_PATH
> EOF
host1# cat > /etc/profile.d/mpl.csh <<EOF
> set path = (MPL_PATH/bin $path)
> if ($?LD_LIBRARY_PATH == 0) then
>   setenv LD_LIBRARY_PATH MPL_PATH/lib
> else
>   setenv LD_LIBRARY_PATH MPL_PATH/lib:\$LD_LIBRARY_PATH
> endif
> EOF
host1# chmod 755 /etc/profile.d/mpl.sh /etc/profile.d/mpl.csh
```

Replace *MPL_PATH* with the directory name of the Open MPI that you wish to use. For example, to use Open MPI with the GNU compiler, use `/usr/local/ofed/mpl/gcc/openmpi-1.1.2-1` for *MPL_PATH*. If you want to use the Intel compiler, use `/usr/local/ofed/mpl/intel/openmpi-1.1.2-1` for *MPL_PATH*.

Edit Environment Variables In the Users' Shell Startup Files

This method allows users to have their own preference of which MPI to use, but it requires that users manually modify their own shell startup files. Individual users can use this method to override the system default MPI implementation selection.

All shells have some type of script file that is executed at login time to set environment variables (such as `PATH` and `LD_LIBRARY_PATH`) and perform other environmental setup tasks. While your system may be different, [Table 6-1](#) lists some common shells and the startup files that might require edits to set up MPI upon login.

Table 6-1 Common Shells and Startup Files

Shell	Startup file to edit
sh (Bourne shell, or bash named sh)	<code>\$HOME/.profile</code>
csh	<code>\$HOME/.cshrc</code>
tcsh	<code>\$HOME/.tcshrc</code> if it exists, or <code>\$HOME/.cshrc</code> if it does not
bash	<code>\$HOME/.bashrc</code> if it exists, or <code>\$HOME/.bash_profile</code> if it exists, or <code>\$HOME/.profile</code> if it exists (in that order)

Set up MVAPICH in Users' Shell Startup Files

This example shows how to edit a user's shell startup files to use MVAPICH. If the user uses the Bourne or Bash shell, edit the startup file after referring to [Table 6-1](#) on all nodes where the user uses MPI, and add the following line:

```
export PATH=MPI_PATH:$PATH
```

If the user uses the C or T shell, edit the startup file after referring to [Table 6-1](#), and add the following line:

```
set path = (MPI_PATH $path)
```

Replace *MPI_PATH* with the directory name of the MVAPICH that you wish to use. For example, to use MVAPICH with the GNU compiler, use `/usr/local/ofed/mpi/gcc/mvapich-0.9.7-mlx2.2.0/bin` for *MPI_PATH*. If you want to use the Intel compiler, use `/usr/local/ofed/mpi/intel/mvapich-0.9.7-mlx2.2.0/bin` for *MPI_PATH*.

Set up Open MPI in Users' Shell Startup Files

This example shows how to edit a user's shell startup files to use Open MPI. If the user uses the Bourne or Bash shell, edit the startup file after referring to [Table 6-1](#) on all nodes where the user uses MPI, and add the following lines:

```
export PATH=MPI_PATH/bin:$PATH
export LD_LIBRARY_PATH=MPI_PATH/lib:$LD_LIBRARY_PATH
```

If the user uses the C or T shell, edit the startup file after referring to [Table 6-1](#), and add the following lines:

```
set path = (MPI_PATH/bin $path)
if ($?LD_LIBRARY_PATH == 0) then
    setenv LD_LIBRARY_PATH MPI_PATH/lib
else
    setenv LD_LIBRARY_PATH MPI_PATH/lib:$LD_LIBRARY_PATH
endif
```

Replace *MPI_PATH* with the directory name of the Open MPI that you wish to use. For example, if you want to use Open MPI with the GNU compiler, use `/usr/local/ofed/mpi/gcc/openmpi-1.1.2-1` for *MPI_PATH*. If you want to use the Intel compiler, use `/usr/local/ofed/mpi/intel/openmpi-1.1.2-1` for *MPI_PATH*.

Edit Environment Variables Manually

Typically, you edit environment variables manually when it is necessary to run temporarily with a given MPI implementation. For example, when it is not desirable to change the default MPI implementation, editing environment variables manually can set which MPI is used for the shell where the variables are set.

Set up MVAPICH Manually in a Shell

This example shows how to create a setup that uses MVAPICH in a single shell. If the user uses the Bourne or Bash shell, enter the following command:

```
host1$ export PATH=MPI_PATH:$PATH
```

If the user uses the C or T shell, enter the following command:

```
host1% set path = (MPI_PATH $path)
```

Replace *MPI_PATH* with the directory name of the MVAPICH that you wish to use. For example, to use MVAPICH with the GNU compiler, use `/usr/local/ofed/mpi/gcc/mvapich-0.9.7-mlx2.2.0/bin` for *MPI_PATH*. If you want to use the Intel compiler, use `/usr/local/ofed/mpi/intel/mvapich-0.9.7-mlx2.2.0/bin` for *MPI_PATH*.

Set up Open MPI Manually in a Shell

This example shows how to create a setup that uses Open MPI in a single shell. If the user uses the Bourne or Bash shell, enter the following command:

```
host1$ export PATH=MPI_PATH/bin:$PATH
host1$ export LD_LIBRARY_PATH=MPI_PATH/lib:$LD_LIBRARY_PATH
```

If the user uses the C or T shell and if the `LD_LIBRARY_PATH` environment variable already has a value, enter the following command:

```
host1% setenv LD_LIBRARY_PATH MPI_PATH/lib:$LD_LIBRARY_PATH
```



Note

If you receive an error, then use the command for the `LD_LIBRARY_PATH` environment variable not defined as shown in the example below.

If the user uses the C or T shell and if the `LD_LIBRARY_PATH` environment is not defined, enter the following command:

```
host1% setenv LD_LIBRARY_PATH MPI_PATH/lib
```

Replace *MPI_PATH* with the directory name of the Open MPI that you wish to use. For example, if you want to use Open MPI with the GNU compiler, use `/usr/local/ofed/mpi/gcc/openmpi-1.1.2-1` for *MPI_PATH*. If you want to use the Intel compiler, use `/usr/local/ofed/mpi/intel/openmpi-1.1.2-1` for *MPI_PATH*.

Perform MPI Bandwidth Test

The MPI bandwidth test is a good test to ensure that MPI and your OFED installation is functioning properly. This procedure requires that you can log in to remote nodes without a login name and password and that the MPI bin directory is in your PATH.

-
- Step 1** Log in to your local host.
 - Step 2** Create a text file containing the names of two hosts on which to run the test. These hostnames are likely to be unique to your cluster. The first name should be the name of the host into which you are currently logged.

This example shows one method to create a hostfile named *hostfile* that contains the hostnames `host1` and `host2`.

```
host1$ cat > /tmp/hostfile <<EOF
> host1
> host2
> EOF
host1$
```

Step 3 Run the MPI bandwidth test across multiple hosts. Use the **mpirun** command to launch MPI jobs; it takes several command line parameters:

- The **-np** keyword to specify the number of processes
- The number of processes (an integer; use 2 for this test)
- The **-hostfile** keyword to specify a file containing the hosts on which to run
- The name of the hostfile
- The *bw* executable name

a. This example shows how to run the MVAPICH MPI bandwidth test:

```
host1$ mpirun_rsh -np 2 -hostfile /tmp/hostfile \
/usr/local/ofed/mpi/compiler/mvapich-0.9.7-mlx2.2.0/tests/osu-benchmarks-2.2/osu_bw
```

b. This example shows how to run the Open MPI bandwidth test:

```
host1$ mpirun --prefix /usr/local/ofed/mpi/compiler/openmpi-1.1.2-1 -np 2 \
-hostfile tmp/hostfile \
/usr/local/ofed/mpi/compiler/openmpi-1.1.2-1/tests/osu-benchmarks-2.2/osu_bw
```



Note If Open MPI was set up in system-wide or user-shell startup files, the “`--prefix /usr/local/ofed/mpi/compiler/openmpi-1.1.2-1`” options are unnecessary.

When the test completes successfully, you see output that is similar to the following:

```
# OSU MPI Bandwidth Test (Version 2.2)
# Size      Bandwidth (MB/s)
1           3.352541
2           6.701571
4           10.738255
8           20.703599
16          39.875389
32          75.128393
64          165.294592
128         307.507508
256         475.587808
512         672.716075
1024        829.044908
2048        932.896797
4096        1021.088303
8192        1089.791931
16384       1223.756784
32768       1305.416744
65536       1344.005127
131072      1360.208200
262144      1373.802207
524288      1372.083206
1048576     1375.068929
2097152     1377.907100
4194304     1379.956345
```

Perform MPI Latency Test

The MPI latency test is another good test to ensure that MPI and your OFED installation are functioning properly. This procedure requires your ability to log in to remote nodes without a login name and password, and it requires that the MPI directory is in your PATH.

Step 1 Log in to your local host.

Step 2 Create a text file containing the names of two hosts on which to run the test. These hostnames are likely to be unique to your cluster. The first name should be the name of the host into which you are currently logged.

This example shows one way to create a hostfile named *hostfile* that contains the hostnames *host1* and *host2*

```
host1$ cat > /tmp/hostfile <<EOF
> host1
> host2
> EOF
host1$
```

Step 3 Run the MPI latency test across multiple hosts. Use the **mpirun** command to launch MPI jobs; it takes several command line parameters:

- The **-np** keyword to specify the number of processes
- The number of processes (an integer; use 2 for this test)
- The **-hostfile** keyword to specify a file containing the hosts on which to run
- The name of the hostfile
- The *latency* executable name

a. This example shows how to run the MVAICH MPI latency test:

```
host1$ mpirun_rsh -np 2 -hostfile /tmp/hostfile \
/usr/local/ofed/mpi/compiler/mvapich-0.9.7-mlx2.2.0/tests/osu-benchmarks-2.2\
/osu_latency
```

b. This example shows how to run the Open MPI latency test:

```
host1$ mpirun --prefix /usr/local/ofed/mpi/compiler/openmpi-1.1.2-1 -np 2 \
-hostfile /tmp/hostfile \
/usr/local/ofed/mpi/compiler/openmpi-1.1.2-1/tests/osu-benchmarks-2.2/osu_latency
```



Note If Open MPI was set up in system-wide or user-shell startup files, the “**--prefix /usr/local/ofed/mpi/compiler/openmpi-1.1.2-1**” options are unnecessary.

When the test completes successfully, you see output that is similar to the following:

```
# OSU MPI Latency Test (Version 2.2)
# Size      Latency (us)
0           2.83
1           2.85
2           2.86
4           2.94
8           2.97
16          2.97
32          3.08
64          3.11
128         3.90
256         4.26
512         4.95
1024        6.07
2048        7.31
4096        9.88
8192        23.35
16384       29.03
32768       41.23
65536       65.07
131072      113.01
262144      209.19
524288      400.72
1048576     780.69
2097152     1540.19
4194304     3072.65
```

Perform Intel MPI Benchmarks (IMB) Test

The IMB test executes a variety of communication patterns across multiple nodes as a simple stress test of your MPI and OFED installations. The tested patterns are as follows:

- PingPong and PingPing: tested across pairs of nodes
- Sendrecv, Exchange, Allreduce, Reduce, Reduce_scatter, Allgather, Allgatherv, Alltoall, Bcast, Barrier: tested across multiple nodes, always using a power of two such as 2, 4, 8, 16.

When your OFED installation is not working properly, the IMB test might lead to VAPI_RETRY_EXEC errors. You should check the output of the PingPong, PingPing, and Sendrecv bandwidth measurements against known good results on similar architectures and devices. Low-bandwidth values, especially at high numbers of nodes, might indicate either severe congestion or functionality problems within the InfiniBand (IB) fabric. Congestion can occur when the IMB test is run across a large number of nodes on fabrics with a high-blocking factor.

Step 1 Log in to your local host.

Create a text file containing the names of all hosts on which to run the test. You should include at least two hosts. These hostnames are likely to be unique to your cluster. The first name should be the name of the host into which you are currently logged.

This example shows one way to create a hostfile named *hostfile* that contains the hostnames *host1* through *host4*:

```
host1$ cat > /tmp/hostfile <<EOF
> host1
> host2
> host3
> host4
> EOF
host1$
```

Step 2 Run the IMB tests across multiple hosts. Use the **mpirun** command to launch MPI jobs; it takes several command line parameters:

- The **-np** keyword to specify the number of processes
 - The number of processes (an integer; use the number of hosts in the hostfile for this test)
 - The **-hostfile** keyword to specify a file containing the on which hosts to run
 - The name of the hostfile
 - The IMB-MPI1 executable name
- a. This example shows how to perform the MVAPICH MPI IMB test by compiling and running *IMB-MPI1* (vary the value of the *-np* parameter to reflect the number of hosts that you want to run):

```
host1$ mpirun_rsh -np 2 -hostfile /tmp/hostfile \
/usr/local/ofed/mpi/compiler/mvapich-0.9.7-mlx2.2.0/tests/IMB-2.3/IMB-MPI1
```

- b. This example shows how to perform the Open MPI IMB test by compiling and running *IMB-MPI1* (vary the value of the *-np* parameter to reflect the number of hosts that you want to run):

```
host1$ mpirun --prefix /usr/local/ofed/mpi/compiler/openmpi-1.1.2-1 -np 2 \
-hostfile /tmp/hostfile \
/usr/local/ofed/mpi/compiler/openmpi-1.1.2-1/tests/IMB-2.3/IMB-MPI1
```



Note If Open MPI was set up in system-wide or user-shell startup files, the “`--prefix /usr/local/ofed/mpi/compiler/openmpi-1.1.2-1`” options are unnecessary.

Compile MPI Programs

Compiling MPI applications from source code requires adding several compiler and linker flags. Both MVAPICH and Open MPI provide *wrapper* compilers that add all appropriate compiler and linker flags to the command line and then invoke the appropriate underlying compiler, such as the GNU or Intel compilers, to actually perform the compile and/or link. This section provides examples of how to use the wrapper compilers.

Step 1 Log in to your local host.

Step 2 Copy the example files to your \$HOME directory.

The example files can be copied as follows:

```
host1$ cp -r /usr/local/ofed/mpi/examples $HOME/mpi-examples
```

The files in the /usr/local/ofed/mpi/examples directory are sample MPI applications that are provided both as a trivial primer to MPI as well as simple tests to ensure that your MPI installation works properly. There are two MPI examples in the directory, each in four programming languages.

This example shows Hello world:

C	hello_c.c
C++	hello_cxx.cc
F77	hello_f77.f
F90	hello_f90.f90

This example sends a trivial message around in a ring:

C	ring_c.c
C++	ring_cxx.cc
F77	ring_f77.f
F90	ring_f90.f90



Note A comprehensive MPI tutorial is available at the following URL:
<http://webct.ncsa.uiuc.edu:8900/public/MPI/>

Step 3 Compile the examples as shown below:

```
host1$ cd $HOME/mpi-examples
host1$ mpicc -o hello_c hello_c.c
host1$ mpiCC -o hello_cxx hello_cxx.cc
host1$ mpif77 -o hello_f77 hello_f77.f
host1$ mpif90 -o hello_f90 hello_f90.f90
```

Step 4 If the \$HOME/mpi-examples directory is not shared across all hosts in the cluster, copy the executables to a directory that is shared across all hosts, such as to a directory on a network filesystem.

Step 5 Run the MPI program.

- a. This example shows how to run an MVAPICH MPI C program Hello World:

```
host1$ mpirun_rsh -np 2 -hostfile /tmp/hostfile $HOME/mpi-examples/hello_c
Hello, world, I am 0 of 2
Hello, world, I am 1 of 2
```

- b. This example shows how to run an Open MPI C program Hello World:

```
host1$ mpirun --prefix /usr/local/ofed/mpi/gcc/openmpi-1.1.2-1/ -np 2 \
-hostfile /tmp/hostfile $HOME/mpi-examples/hello_c
Hello, world, I am 0 of 2
Hello, world, I am 1 of 2
```



Note If Open MPI was set up in system-wide or user-shell startup files, the “`--prefix /usr/local/ofed/mpi/compiler/openmpi-1.1.2-1`” options are unnecessary.



HCA Utilities and Diagnostics

The following sections appear in this chapter:

- [Introduction](#), page 7-1
- [hca_self_test Utility](#), page 7-1
- [tvflash Utility](#), page 7-3
- [Diagnostics](#), page 7-4
- [Performance Tests](#), page 7-5
- [Miscellaneous Utilities](#), page 7-6

Introduction

The sections in this chapter discuss HCA utilities and diagnostics. These features address basic usability and provide starting points for troubleshooting.



Note

See the [“Root and Non-root Conventions in Examples”](#) section on page 1-4 for details about the significance of prompts used in the examples in this chapter.

hca_self_test Utility

The `hca_self_test` utility displays basic HCA attributes and provides introductory troubleshooting information. To run this utility, perform the following steps:

-
- Step 1** Log in to your host.
 - Step 2** Run the `hca_self_test` command.

This example shows how to run the `hca_self_test` command:

```
host1# hca_self_test

---- Performing InfiniBand HCA Self Test ----
Number of HCAs Detected ..... 1
PCI Device Check ..... PASS
Kernel Arch ..... x86_64
Host Driver Version ..... OFED-1.1 1.1-2.6.9_34.ELsmp
Host Driver RPM Check ..... PASS
```

```

HCA Type of HCA #0 ..... LionMini
HCA Firmware on HCA #0 ..... v5.1.400 build 3.2.0.102 HCA.LionMini.A
0
HCA Firmware Check on HCA #0 ..... PASS
Host Driver Initialization ..... PASS
Number of HCA Ports Active ..... 2
Port State of Port #0 on HCA #0 ..... UP 4X
Port State of Port #1 on HCA #0 ..... UP 4X
Error Counter Check on HCA #0 ..... PASS
Kernel Syslog Check ..... PASS
Node GUID ..... 00:05:ad:00:00:20:08:48
----- DONE -----

```

Table 7-1 lists and describes the fields in the `hca_self_test` output.

Table 7-1 Fields in `hca_self_test` Output

Field	Description
Number of HCAs Detected	Number of HCAs on the host that the test recognizes.
PCI Device Check	Confirms that HCA shows up correctly as a PCI device.
Kernel Architecture	Kernel architecture on the host.
Host Driver Version	Version of the drivers on the host.
Host Driver RPM Check	Confirms that the RPMS that are installed are compatible with the host operating system.
HCA Type of HCA #0	Displays the HCA card type.
HCA Firmware on HCA #0	Firmware version that runs on the HCA.
HCA Firmware Check on HCA #0	Displays PASS or FAIL.
Host Driver Initialization	Confirms that the IPoIB driver is installed correctly.
Number of HCA Ports Active	Number of enabled ports on the HCA.
Port State of Port #0 on HCA #0	Displays up or down to reflect the status of the port.
Port State of Port #1 on HCA #0	Displays up or down to reflect the status of the port.
Error Counter Check	Displays PASS or FAIL.
Kernel Syslog Check	Displays PASS or FAIL.
Node GUID	IB node GUID.

tvflash Utility

The tvflash utility performs the following tasks:

- View card type and firmware version. The steps to view the card type and firmware version are described in a section below.
- Upgrades the firmware on the HCA. The steps to upgrade your firmware on the HCA are described in a section below.

**Note**

The firmware upgrade is handled automatically by the OFED installation script. You should not have to upgrade the firmware manually. For more information about OFED installation script, see [Chapter 2, “Installing Host Drivers.”](#)

View Card Type and Firmware Version

To display the type of HCA in your host and the firmware that it runs, perform the following steps:

Step 1 Log in to your host.

Step 2 Enter the **tvflash** command with the **-i** flag.

This example shows how to enter the **tvflash** command with the **-i** flag:

```
host1# tvflash -i
HCA #0: MT25208 Tavor Compat, Lion Cub, revision A0
  Primary image is v4.7.600 build 3.2.0.82, with label 'HCA.LionCub.A0'
  Secondary image is v4.7.400 build 3.2.0.67, with label 'HCA.LionCub.A0'

Vital Product Data
Product Name: Lion cub
P/N: 99-00026-01
E/C: Rev: B03
S/N: TS0520X01634
Freq/Power: PW=10W;PCIe 8X
Date Code: 0520
Checksum: Ok
```

The firmware that runs on the HCA appears in the **Primary image** line displayed in [Step 2](#). The card type also appears in this line as one of the following:

- PCI-X Cougar
- PCI-X Cougar Cub
- PCI-e Lion Cub
- PCI-e Lion Mini
- PCI-e Cheetah

The ASIC version appears as A1 or A0.

- `ibaddr`
- `ibcheckerrors`
- `ibcheckerrs`
- `ibchecknet`
- `ibchecknode`
- `ibcheckport`
- `ibcheckportstate`
- `ibcheckportwidth`
- `ibcheckstate`
- `ibcheckwidth`
- `ibclearcounters`
- `ibclearerrors`
- `ibhosts`
- `ibnetdiscover`
- `ibnodes`
- `ibping`
- `ibportstate`
- `ibroute`
- `ibstat`
- `ibstatus`
- `ibswitches`
- `ibsysstat`
- `ibtracert`
- `perfquery`
- `sminfo`
- `smpdump`
- `smpquery`

For more information about these diagnostics, see the Linux man page for that specific diagnostic (for example, `ibstatus`).

Performance Tests

OFED includes several IB benchmark programs. These benchmarks do not require the higher-level protocols such as IPoIB, and can be used to ensure that IB is functioning properly.

- `ib_clock_test`
- `ib_rdma_bw`
- `ib_rdma_lat`
- `ib_read_bw`

- `ib_read_lat`
- `ib_send_bw`
- `ib_send_lat`
- `ib_write_bw`
- `ib_write_bw_postlist`
- `ib_write_lat`

For more information about these diagnostics, see `/usr/local/ofed/docs/PERF_TEST_README.txt`.


Note

These IB benchmark programs should be run on pairs of hosts.

This example shows how to run an `ib_rdma_lat` performance test on a pair of hosts:

```
host1$ ib_rdma_lat
  local address: LID 0x05 QPN 0xca0442 PSN 0xdbc06e RKey 0x5e04ae20 VAddr 0x000
0000508001
<waiting for client to connect>
host2$ ib_rdma_lat host1
  local address: LID 0x06 QPN 0x90442 PSN 0xa6a389 RKey 0x5804ae00 VAddr 0x0000
0000508001
  remote address: LID 0x05 QPN 0xc90442 PSN 0xe7c97f RKey 0x5804ae20 VAddr 0x000
0000508001
Latency typical: 2.96723 usec
Latency best   : 2.93973 usec
Latency worst  : 18.9458 usec
```

Miscellaneous Utilities

OFED includes several miscellaneous utilities that are simpler and offer less functionality than those listed in the “[Diagnostics](#)” section on page 7-4. Some of these utilities are listed below:

- `ibv_asyncwatch`
- `ibv_devices`
- `ibv_devinfo`
- `ibv_rc_pingpong`
- `ibv_srq_pingpong`
- `ibv_uc_pingpong`
- `ibv_ud_pingpong`

For more information about these utilities, see the Linux man page for that specific utility (for example, `man ibv_devices`).



Acronyms and Abbreviations

Table A-1 defines the acronyms and abbreviations that are used in this publication.

Table A-1 *List of Acronyms and Abbreviations*

Acronym	Expansion
API	Application Program Interface
CLI	command-line interface
GUI	graphical user interface
GUID	globally unique identifier
HCA	Host Channel Adapter
IB	InfiniBand
IPoIB	Internet Protocol over InfiniBand
ITL	Initiator/Target/LUN
LU	logical unit
LUN	logical unit number
MPI	Message Passing Interface
MVAPICH MPI	MVAPICH Message Passing Interface
OFED	OpenFabrics Enterprise Distribution
Open MPI	Open Message Passing Interface
PCU	protocol control information
SAN	Storage Area Network
SCSI	Small Computer System Interface
SDP	Sockets Direct Protocol
SRP	SCSI RDMA Protocol
SSH	Secure Shell Protocol
RAID	Redundant Array of Independent Disks
RDMA	Remote Direct Memory Access
RPM	Red Hat Package Manager
SCP	Secure Copy

Table A-1 *List of Acronyms and Abbreviations (continued)*

Acronym	Expansion
TCP	Transmission Control Protocol
uDAPL	User Direct Access Programming Library
ULP	upper-level protocol
WWNN	worldwide node name
WWPN	worldwide port name



A

architecture, HCA supported [1-2](#)
audience [vii](#)
authenticity messages [6-5](#)

B

Bandwidth test [3-6, 5-6, 5-7, 6-8](#)

C

card type, view [7-3](#)
CLI [4-2](#)
command-line interface. See CLI.
compile MPI programs [6-14](#)
compiler
 GNU [6-1](#)
 Intel [6-1](#)
configure
 IPoIB [3-2, 5-1](#)
 ITL [4-2](#)
 SRP [4-1, 4-5](#)
 SSH [6-2](#)
connections, host-to-storage [4-7](#)
conventions, document [viii](#)
conversion type
 automatic [5-2](#)
 explicit/source code [5-2](#)
create subinterface [3-3](#)

D

diagnostic programs, list [7-4](#)
distributed memory environment [6-1](#)
document
 audience [vii](#)
 conventions [viii](#)
 organization [vii](#)
 related [ix](#)

E

Element Manager [4-2](#)
environment variables
 edit manually [6-7](#)
 set system-wide [6-5](#)
 users' shell [6-6](#)

F

Fibre Channel
 Gateway [4-1](#)
 storage [4-1](#)
 storage devices [1-3, 4-1](#)
fingerprint, key [6-3](#)
firmware version [2-4, 7-3](#)

G

gateway [4-1](#)
Gateway, Fibre Channel [4-1](#)
Globally Unique Identifier. See GUID.
global policy restrictions [4-2, 4-3](#)

GNU compiler [6-1](#)
 graphical user interface. See GUI.
 GUI [4-2](#)
 GUID [4-2](#)

H

HCA

description [1-1](#)
 diagnostics [1-4, 7-1](#)
 firmware version [2-4](#)
 ports [2-4, 3-1](#)
 supported APIs [1-1](#)
 supported protocols [1-1](#)
 utilities [1-4, 7-1](#)

hca_self_test

output [7-2](#)
 utility [7-1](#)

high availability [4-5](#)

Host Channel Adapter. See HCA.

host drivers

install [2-2](#)
 uninstall [2-5](#)

host operating system log files [2-4](#)

host-to-storage connections [4-7](#)

I

IB [1-1, 1-3, 4-1, 5-1](#)

IB partition [3-2, 3-3, 3-4](#)

ifconfig command [3-2](#)

IMB [6-11](#)

InfiniBand. See IB.

InfiniHost [2-2](#)

Initiator/Target/LUNs. See ITLs.

install, host drivers [2-2](#)

Intel compiler [6-1](#)

IPoIB

configure [3-2, 5-1](#)

description [1-3](#)

functionality [3-5](#)

IP over InfiniBand. See IPoIB.

ISO image

contents [2-2](#)

install [2-2](#)

OFED [2-2](#)

uninstall [2-5](#)

ITLs [4-1](#)

K

kernel modules [2-4](#)

key pair [6-2](#)

L

Latency test [3-6, 6-10](#)

load balancing [4-5](#)

log files, host operating system [2-4](#)

logical unit number. See LUN.

login, password-less [6-2](#)

log statement [5-2](#)

LUN [4-2](#)

LUN masking policy [4-3](#)

M

management tools, TopspinOS [7-4](#)

md5sum utility [2-2](#)

Message Passing Interface. See MPI.

message passing program [6-1](#)

MPI

Bandwidth test [6-8](#)

compile programs [6-14](#)

description [1-3, 6-1](#)

Intel Benchmarks test [6-11](#)

Latency test [6-10](#)
 MVAPICH [1-3, 6-1](#)
 Open [1-3, 6-1](#)
 tutorial [6-14](#)
 MPI implementation
 multiple [6-5](#)
 single [6-5](#)
 MVAPICH MPI [1-3, 6-1](#)

N

netmask [3-2](#)
 Netperf [3-6, 5-6](#)
 Netperf server [3-6, 5-6](#)

O

OFED
 ISO image [2-2](#)
 release notes [2-1](#)
 OpenFabrics Enterprise Distribution. See OFED.
 Open MPI [1-3, 6-1](#)
 organization, document [vii](#)

P

password-less login [6-2](#)
 PCI-e
 Cheetah [7-3](#)
 Lion Cub [7-3](#)
 Lion Mini [7-3](#)
 PCI-Express server [1-1](#)
 PCI-X
 Cougar [7-3](#)
 Cougar Cub [7-3](#)
 PCI-X server [1-1](#)
 performance tests [7-5](#)
 policy

LUN masking [4-3](#)
 portmask [4-3](#)
 portmask policy [4-3](#)
 programming languages [6-1](#)
 public/private key pair [6-2](#)

R

RAID [4-6](#)
 RDMA [4-1](#)
 Red Hat Package Manager. See RPM.
 Redundant Array of Independent Disk. See RAID.
 related documentation [ix](#)
 remote direct memory access. See RDMA.
 remote node [6-5](#)
 remove subinterface [3-4](#)
 RPM [2-1](#)

S

SAN [4-1](#)
 SCP [6-4](#)
 SCSI [1-3, 4-1](#)
 SCSI RDMA Protocol. See SRP.
 SDP [1-1, 1-3, 5-1](#)
 secure copy. See SCP.
 Secure Shell Protocol. See SSH.
 server switch [4-1](#)
 Small Computer System Interface. See SCSI.
 sockets-based application [5-1](#)
 Sockets Direct Protocol. See SDP.
 SRP [1-1, 1-3, 4-1](#)
 SRP, configure [4-1, 4-5](#)
 SSH [6-2](#)
 SSH, configure [6-2](#)
 standard wire protocol [5-1](#)
 startup configuration file [3-7](#)
 statement

log [5-2](#)
 use [5-3](#)
 storage area network. See SAN.
 stream sockets networking [5-1](#)
 subinterface
 create [3-3](#)
 description [3-2](#)
 remove [3-4](#)

T

test
 Bandwidth [3-6](#)
 Bandwidth, default [3-6, 5-7](#)
 Bandwidth, MPI [6-8](#)
 Bandwidth, with SDP [5-6](#)
 IMB [6-11](#)
 Intel MPI Benchmarks. See IMB.
 Latency [3-6](#)
 Latency, MPI [6-10](#)
 TopspinOS management tools [7-4](#)
 tvflash utility [7-3](#)

U

uninstall
 host drivers [2-5](#)
 upgrade, firmware [7-4](#)
 upper layer protocol [5-1](#)
 use rules [5-4](#)
 use statement [5-3](#)
 utility
 hca_self_test [7-1](#)
 tvflash [7-3](#)

V

verify

 with Element Manager [4-9](#)
 view
 card type [7-3](#)
 firmware version [7-3](#)

W

worldwide node names. See WWNNs.
 worldwide port names. See WWPN.
 WWNN [4-2, 4-3, 4-4](#)
 WWPN [4-3, 4-4](#)