Implementing Cisco InfiniBand on IBM BladeCenter

Add high-speed 4X InfiniBand networking to your BladeCenter

Plan and configure the solution for your environment

Learn about the latest InfiniBand DDR products

Khalid M Ansari
Robyn McGlotten
Matt Slavin
David Watts

Redpaper

ibm.com/redbooks

International Technical Support Organization

Implementing Cisco InfiniBand on IBM BladeCenter

October 2007

REDP-3949-01

Note: Before using this information and the product it supports, read the information in "Notices" on page vii.

Second Edition (October 2007)

This edition applies to the following InfiniBand products for IBM BladeCenter H:

Cisco 4X InfiniBand Switch Module, 32R1756
Cisco 4X InfiniBand HCA Expansion Card, 32R1760
QLogic InfiniBand Ethernet Bridge Module, 39Y9207
QLogic InfiniBand Fibre Channel Bridge Module, 39Y9211
InfiniBand 4X DDR Pass-thru Module, 43W4419
4X InfiniBand DDR Expansion Card, 43W4423
Cisco 4X InfiniBand DDR Expansion Card, 43W4421
Voltaire 4X InfiniBand DDR Expansion Card, 43W4420

This document created or updated on December 17, 2007.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

Redbooks (logo) ®
eServer™
BladeCenter®
Chipkill™
DS4000™
IBM®
PowerExecutive™
PowerPC®
Predictive Failure Analysis®
Redbooks®
ServerProven®
System x™
System Storage™

The following terms are trademarks of other companies:

Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.

InfiniBand, and the InfiniBand design marks are trademarks and/or service marks of the InfiniBand Trade Association.

Advanced Micro Devices, AMD, AMD Opteron, the AMD Arrow logo, and combinations thereof, are trademarks of Advanced Micro Devices, Inc.

QLogic, and the QLogic logo are registered trademarks of QLogic Corporation. SANblade is a registered trademark in the United States.

Java, JVM, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows Server, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Xeon, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Preface

InfiniBand® networking offers a high speed, low latency interconnect that is often a requirement for High Performance Computing (HPC) networks. Combined with various Ethernet and Fibre Channel gateways, this technology also offers a simplified multifabric I/O to permit InfiniBand clients to access more traditional interconnects, using InfiniBand as the single unifying infrastructure.

This paper introduces the various IBM®, Cisco, and QLogic InfiniBand components that are available for IBM BladeCenter. These components include the internal 4X InfiniBand switch and pass-thru modules, 4X host channel adapters, and Ethernet and Fibre Channel bridge modules. We also introduce the external Cisco SFS 3012 Multifabric Server switch, which provides scalable and robust Ethernet and Fibre Channel gateway connectivity to InfiniBand clients. Lastly, we introduce general InfiniBand technology concepts.

Beyond product and general InfiniBand introductions, the paper provides a number of step-by-step examples that show how to deploy these products, including how to install the InfiniBand client software and configure the various switches and gateways to demonstrate some of the many connection options that these solutions offer.

This paper is targeted at audiences that need an introduction to the available InfiniBand connectivity options for the IBM BladeCenter H and a basic introduction to InfiniBand, as well as for those who need specific information about how to implement these products.

The team that wrote this paper

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), Raleigh Center.

Khalid M Ansari is a member of the System x™ Advanced Technical Support team and the technical team lead for the IBM Blade Infrastructure Solution Center in Research Triangle Park, N.C. His responsibilities include assisting BladeCenter pre-sale customers with new proof of concepts, solution assurance reviews, solution implementation and pilot testing. He has extensive experience with BladeCenter SAN solutions and was a participant in developing the IBM Storage Networking Solutions V1 Certification. Khalid started with IBM in August 1998 as an ATM Networking Specialist in Level 2 support and later worked in SAN solutions support. He is co-author of the IBM Redbooks® publications *IBM BladeCenter 4Gb SAN Solution* and *IBM BladeCenter iSCSI SAN Solution*.

Robyn McGlotten is a development support engineer in the IBM BladeCenter Development group in RTP, NC, and has six years of experience with IBM. She provides technical support for IBM BladeCenter network products development and implementation. She is co-author of the IBM Redpaper *IBM eServer BladeCenter and TopSpin InfiniBand Switch Technology*. Robyn has a degree in electrical engineering from Florida A&M University.

Matt Slavin is a Consulting Engineer with Cisco Systems Strategic Alliances. He has been in the computer and networking industry for more than 25 years, operating in numerous high-level technical support capacities, and is the co-author of several IBM Redbooks publications on deploying IBM BladeCenter® in Cisco-based infrastructures. His industry certifications include MCSE, MCNE, CCNA, and CCIP. Matt's current professional interests include infrastructure design and support, with a special focus on high density data center networking and security.

David Watts is a Consulting IT Specialist at the IBM ITSO Center in Raleigh. He manages residencies and produces IBM Redbooks publications on hardware and software topics related to IBM System x and BladeCenter servers and associated client platforms. He has authored over 80 books, papers and technotes. He holds a Bachelor of Engineering degree from the University of Queensland (Australia) and has worked for IBM both in the U. S. and Australia since 1989. He is an IBM Certified IT Specialist.

The team (left to right): Khalid, Matt, Robyn, and David

Thanks to the following people for their contributions to this project:

From the International Technical Support Organization:

- Tamikia Barrow
- Carolyn Briscoe
- Todd Kelsey
- Linda Robinson
- Margaret Ticknor
- Erica Wazewski

From IBM Corporation

- Jason Daniel
- Chris Durham
- Ishan Sehgal
- Bill Vetter

From Cisco Systems, Inc:

- Grif Morrel
- Manish Tandon

Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this paper or other IBM Redbooks in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

1

IBM BladeCenter products and technology

Blade servers are thin servers that insert into a single rack-mounted chassis which supplies shared power, cooling, and networking infrastructure. Each server is an independent server with its own processors, memory, storage, network controllers, operating system, and applications. Blade servers came to market around 2000, initially to meet clients' needs for greater ease of administration and increased server density in the data center environment.

When IBM released the IBM BladeCenter in 2002, it quickly changed the industry with its modular design. The IBM BladeCenter provides complete redundancy in a chassis, and enables network and storage integration.

This chapter provides an overview of BladeCenter products and technology. In it, we cover the following topics:

1.1 BladeCenter chassis

There are four chassis in the BladeCenter family:

- *IBM BladeCenter* provides the greatest density and common fabric support and is the lowest entry cost option. To eliminate any confusion, this chassis is also known as *IBM BladeCenter 8677* or *IBM BladeCenter Enterprise*.
- *IBM BladeCenter H* delivers high performance, extreme reliability, and ultimate flexibility for the most demanding IT environments.
- *IBM BladeCenter T* models are designed specifically for telecommunications network infrastructures and other rugged environments.
- *IBM BladeCenter HT* models are a combination of the designs of the H and T chassis. These models are designed specifically for telecommunications and rugged environments.

All four chassis share a common set of blades and switch modules. The exceptions are:

- The oldest HS20 blade server, machine type 8678 is not supported in the BladeCenter H and BladeCenter T chassis.
- The new high-speed switch modules are not supported in the BladeCenter 8677 and BladeCenter T chassis.
- The 4X InfiniBand Switch Module is not supported in the BladeCenter HT.

Note: We will covering the BladeCenter H chassis in detail in this chapter.

1.1.1 BladeCenter H chassis

IBM BladeCenter H delivers high performance, extreme reliability, and ultimate flexibility to even the most demanding IT environments. In 9U of rack space, the BladeCenter H chassis can contain up to 14 blade servers, 10 switch modules, and four power supplies to provide the necessary I/O network switching, power, cooling, and control panel information to support the individual servers.

The chassis supports up to four traditional fabrics using networking switches, storage switches, or pass through devices. The chassis also supports up to four high-speed fabrics for support of protocols such as 4X InfiniBand or 10 Gigabit Ethernet. The built-in media tray includes light path diagnostics, two front USB inputs, and a DVD drive.

Figure 1-1 and Figure 1-2 display the front view of an IBM BladeCenter H.

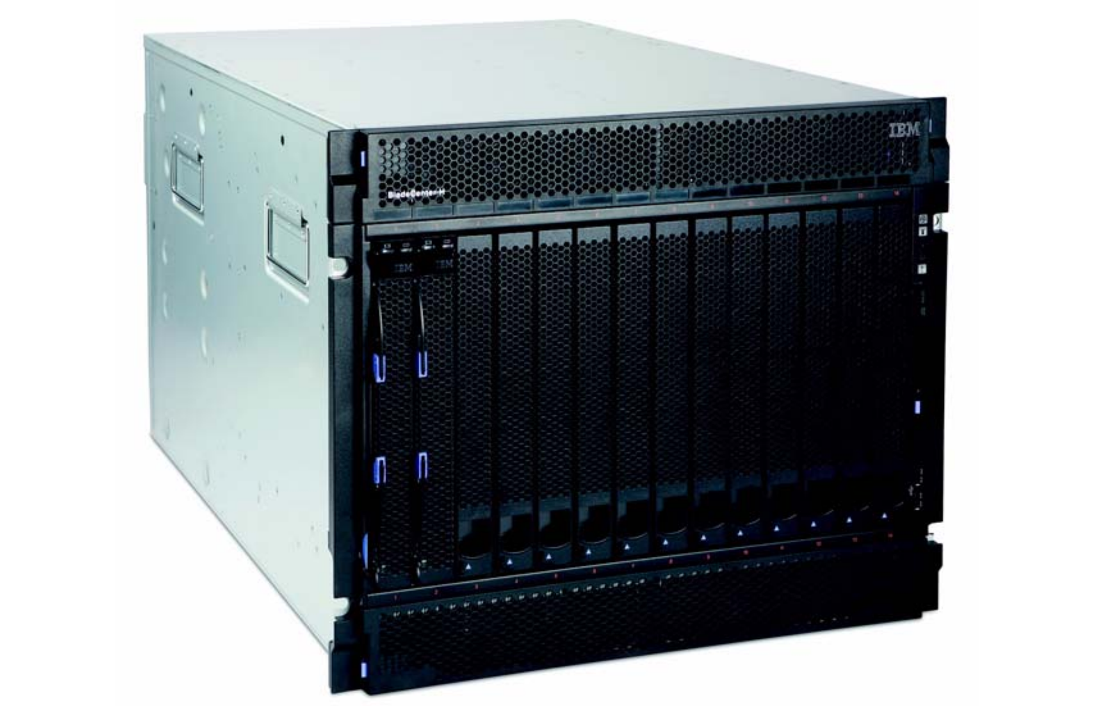


Figure 1-1 BladeCenter H front view

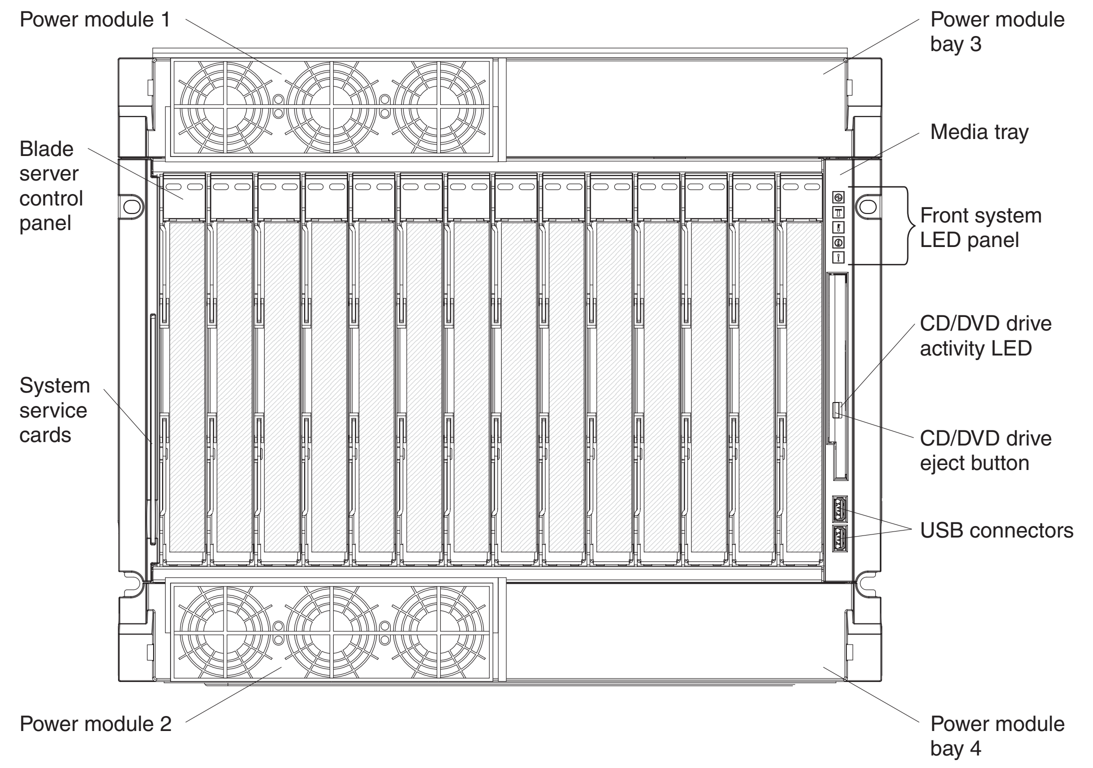


Figure 1-2 Diagram of BladeCenter H front view with the key features of the BladeCenter H chassis

The key features on the front of the BladeCenter H include:

- A media tray at the front right, with a DVD drive, two USB v2.0 ports, and system status LED panel.
- One pair of 2900-watt power modules. An additional power module option (containing two 2900 W power modules) is available.
- Two hot swap fan modules. (Two extra hot swap fan modules are included with the additional power module option.)
- 14 hot swap blade server bays supporting different blade server types.

Figure 1-3 and Figure 1-4 display the back view of an IBM BladeCenter H.

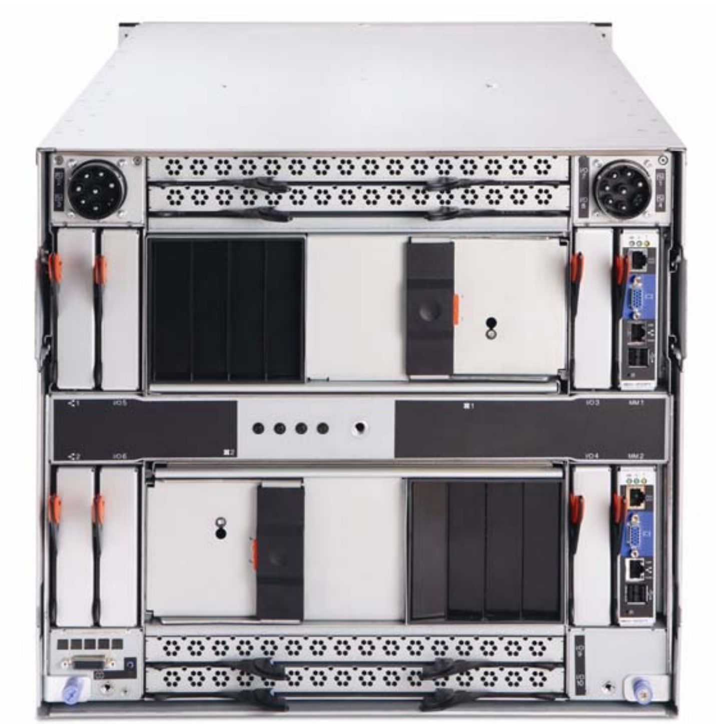


Figure 1-3 BladeCenter H rear view

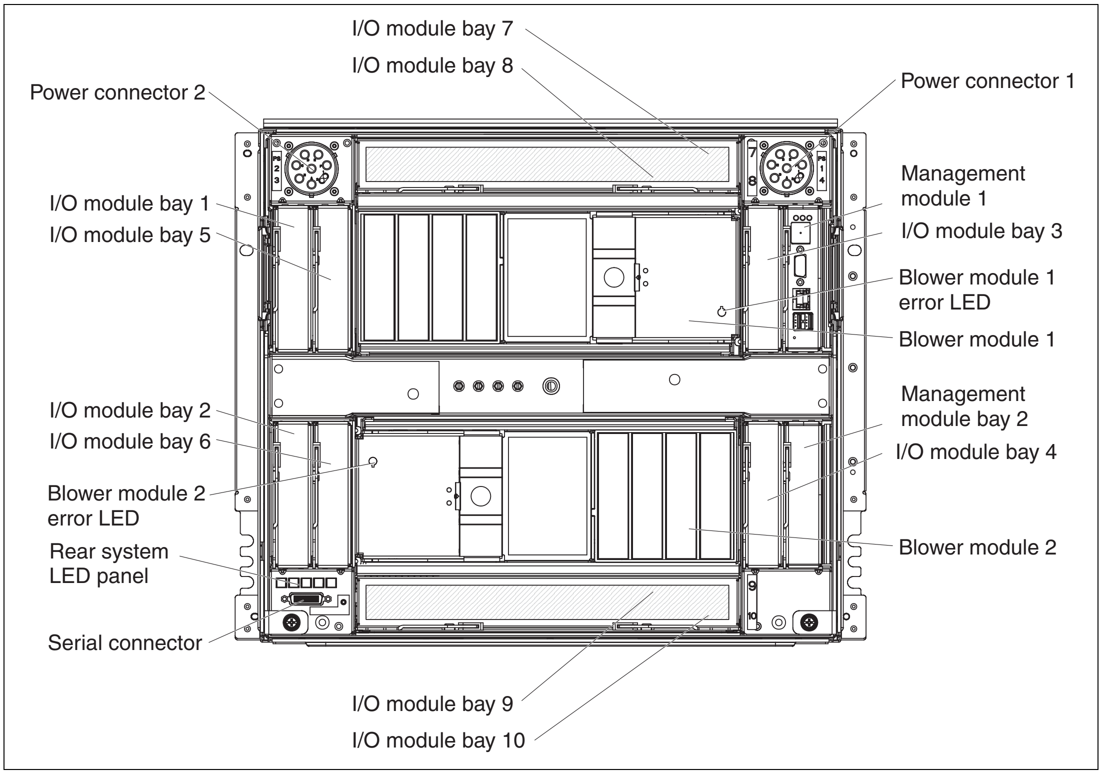


Figure 1-4 Diagram of BladeCenter H rear view showing the key features of the BladeCenter H chassis

The BladeCenter H chassis allows for either 14 single-slot blade servers or seven double-slot blade servers. However, you can mix different blade server models in one chassis to meet your requirements.

The BladeCenter H chassis ships standard with one Advanced Management Module. This module provides the ability to manage the chassis and well as providing the local KVM function. The optional redundant Advanced Management Module provides the IBM BladeCenter H with higher levels of resiliency. This module provides administrators with easy remote management and connectivity to the BladeCenter H chassis.

The BladeCenter has bays for 10 hot-swap I/O modules. You choose these I/O modules based on your connectivity needs. An Ethernet Switch Module (ESM) is required in I/O module bays 1 and 2, to enable the use of both Ethernet ports on a Blade Server. The I/O modules required in I/O module bays 3 and 4 depend on the I/O Expansion Card that is installed in the blade servers. The I/O modules required in the high speed I/O module bays 7, 8, 9, and 10 depend on the I/O Host Channel Adapter cards installed in the blade servers.

1.2 Blade servers

IBM BladeCenter servers supports a wide selection of processor technologies and operating systems to allow clients to run all of their diverse work loads inside a single architecture. The slim, hot-swappable blade servers fit in a single chassis like books in a bookshelf, and each is an independent server, with its own processors, memory, storage, network controllers, operating system and applications. The blade server simply slides into a bay in the chassis and plugs into a midplane or backplane, sharing power, fans, diskette drives, switches, and ports with other blade servers.

The benefits of the blade approach will be obvious to anyone tasked with running down hundreds of cables strung through racks just to add and remove servers. With switches and power units shared, precious space is freed and blade servers enable higher density with far greater ease.

There are two InfiniBand daughter cards available for IBM BladeCenter servers:

- Cisco InfiniBand 1X HCA Expansion Card, part number 32R1896
- Cisco InfiniBand 4X HCA Expansion Card, part number 32R1760

For information about the 4X daughter card, see 2.2, “Cisco 4X InfiniBand HCA Expansion Card” on page 18.

Table 1-1 shows which blade servers support the InfiniBand daughter cards.

Table 1-1 Blade servers and InfiniBand support information

Blade server	Machine type	Processor	1X InfiniBand daughter card	4X InfiniBand daughter card
HS20	8678	Intel® Xeon®	Supported	No
HS20	8832	Intel Xeon	Supported	No
HS20	8843	Intel Xeon	Supported	No
HS20	8678	Intel Xeon	Supported	No
HS21 XM	7995	Intel Xeon	Supported	Supported
HS21	8853	Intel Xeon	Supported	Supported
HS40	8839	Intel Xeon	Supported	No
LS20	8850	AMD™ Opteron™	Supported	No
LS21	7971	AMD Opteron	Supported	Supported
LS41	7972	AMD Opteron	Supported	Supported
JS20	8842	IBM Power	Supported	No
JS21	8844	IBM Power	Supported	Supported

We describe the HS21, LS21, and JS21 in the following sections.

1.2.1 BladeCenter HS21 XM server

The BladeCenter HS21 XM (extended memory) servers are positioned as high-density servers. With one or two dual-core or quad-core Intel processors plus up to 32 GB of DDR2 memory with 8 DIMM sockets, they represent a new approach to the deployment of application servers where performance, capacity, high-availability design, systems management, and easy setup features are combined in an extremely dense package. Reducing an entire server into as little as .5U of rack space does not mean trading away features and capabilities for smaller size.

Integrated dual Gigabit Ethernet controllers are standard, providing high-speed data transfers and offering TCP Offload Engine (TOE) support, load-balancing and failover capabilities. Through optional expansion cards, each blade can also connect to additional Ethernet, Myrinet, Fibre Channel, iSCSI, InfiniBand, and other high-speed communication switches housed in the chassis. Optional 2-port Expansion Cards add additional fabrics to the HS21 XM server as needed. This blade is designed with power management capability to provide the maximum uptime possible for your systems. In extended thermal conditions or power brownouts, rather than shut down completely, or fail, the HS21 XM reduces the processor frequency automatically to maintain acceptable thermal and power levels.

All HS21 XM models also include support one SAS hard disk drive and one USB-based modular flash drive. The optional 30mm Storage and I/O (SIO) Expansion Unit connects to a blade (model-dependent) to provide an additional three 2.5 SAS HDDs with hot-swap support, optional RAID-5 with battery-backed cache, and four additional communication ports.

Figure 1-5 shows the HS21 XM blade server.

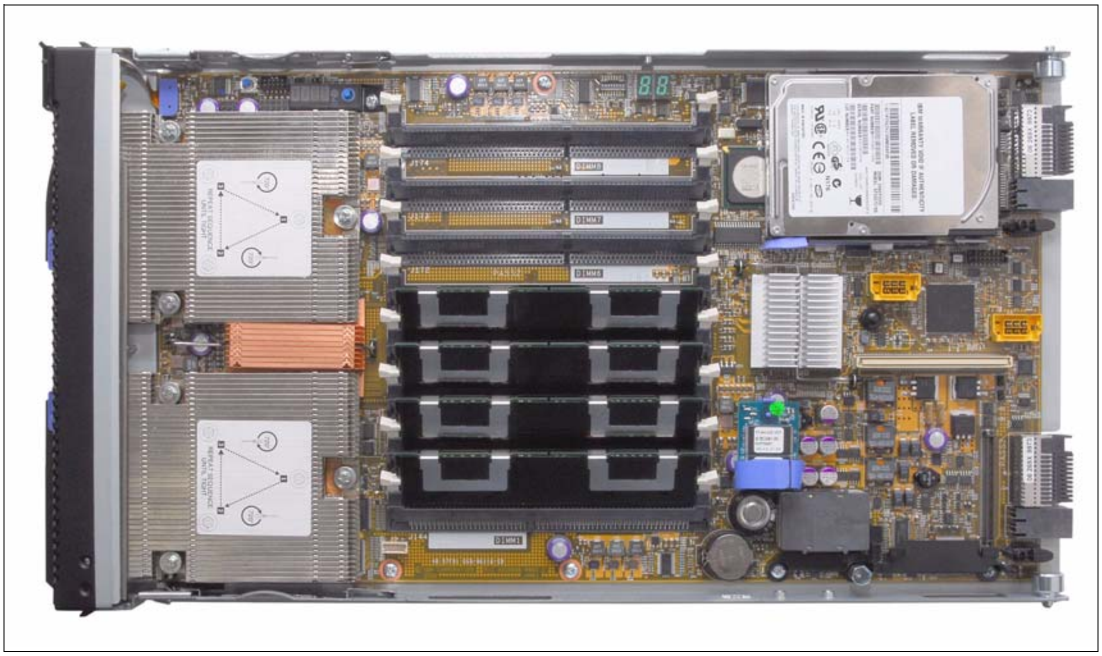


Figure 1-5 The HS21 XM blade server, machine type 7995

Features of the HS21 XM include:

- Blade servers supported in all IBM BladeCenter chassis.
- Two processor sockets supporting either dual-core Intel Xeon 5100 series processors or quad-core Intel Xeon 5300 series processors.
- Up to 32 GB of system memory in 8 DIMM sockets.
- One internal 2.5-inch SAS HDD with support for the Storage and I/O (SIO) expansion unit blade with an additional three 2.5-inch SAS HDD bays and RAID support.
- An optional Modular Flash Drive can be used in place of, or in addition to, the internal SAS HDD, as a boot device.
- Two Gigabit Ethernet ports standard; plus more, using either a 2-port Gigabit Ethernet Expansion Card or a PCI I/O Expansion Unit II.
- Integrated Baseboard Management Controller (BMC) service processor to monitor server availability, perform Predictive Failure Analysis®, etc., and trigger IBM Director alerts.
- Support for IBM PowerExecutive™ 2.0, software designed to take advantage of new system power management features that monitor actual power usage and provide power consumption capping features.
- Concurrent keyboard, video, mouse (KVM) support with addition of the optional IBM BladeCenter Concurrent KVM feature card.

- Three-year, on-site limited warranty.

Table 1-2 provides details on the features of the HS21 XM.

Table 1-2 Features of the HS21 XM

Feature	
Processor	Intel Xeon 5300 Series quad-core or Intel Xeon 5100 Series dual-core processors
Number of processors (std/max)	1 / 2
Front-side bus	1033 MHz or 1333 MHz
Cache	4 MB or 8 MB L2 cache (shared between both cores)
Memory	8 DIMM slots / 32 GB maximum
Memory Type	PC2-5300 (667MHz) Fully Buffered DDR II ECC
Internal hard disk drives (std/max)	0 / 1 (SAS)
Maximum internal storage	On board: One 2.5” Non Hot Swap SAS HDD On board: One 4 GB modular flash drive Optional: SIO Expansion Unit supports 3 additional 2.5” hot-swap SAS drives
Network	Two ports, Integrated Dual Gigabit Ethernet (Broadcom 5708S), TOE
I/O upgrade	One PCI-X expansion connector and 1x8 PCI-Express expansion connector Optional: PCI Expansion Unit II: Two PCI-X slots Optional: SIO Expansion Unit: One or two PCI-X slots

1.2.2 BladeCenter LS21 server

The BladeCenter LS21 blade servers are high-throughput, two-socket, SMP-capable, AMD Opteron-based blade servers. This two-socket AMD Opteron blade server is well suited to HPC and other applications requiring high performance and high-speed fabrics.

The processors used in these blades are standard- and low-power, full-performance Opteron processors. The standard AMD Opteron processors draw a maximum of 95 W. Specially manufactured low-power processors operate at 68 W or less without any performance trade-offs. This savings in power at the processor level combined with the smarter power solution that BladeCenter delivers make these blades very attractive to clients who have limited power and cooling resources.

The BladeCenter LS21 is a single-width blade server that offers these features:

- Up to two high-performance, AMD Dual-Core Opteron processors.
- System board containing eight DIMM connectors, supporting 512 MB, 1 GB, 2 GB, or 4 GB DIMMs. Up to 32 GB of system memory is supported with 4 GB DIMMs.
- SAS controller, supporting one internal SAS drive (36 or 73 GB) and up to three additional SAS drives with optional SIO blade.
- Two TCP/IP Offload Engine-enabled (TOE) Gigabit Ethernet controllers (Broadcom 5706S) standard, with load balancing and failover features.
- Support for concurrent KVM (cKVM) and concurrent USB / DVD (cMedia) through Advanced Management Module and optional daughter card.
- Support for Storage and I/O Expansion (SIO) Unit.
- Three year on-site limited warranty.

Figure 1-6 IBM BladeCenter LS21

Table 1-3 summarizes the features of the LS21.

Table 1-3 Features of the LS21

Feature	Specification
Processor	AMD Opteron Rev F Model 2212, 2212HE, 2216HE and 2218
Number of processors (std/max)	1 / 2
Cache	1 MB L2 per processor core
Memory	8 VLP DIMM slots / DDR2 667 / 32 GB maximum
Internal hard disk drives (standard / maximum)	0 / 1
Maximum internal storage	On board: one 2.5” Non Hot Swap SAS HDD Optional: SIO blade offers support for 3 additional 2.5” hot-swap SAS drives
Network	Two ports, Integrated Dual Gigabit Ethernet (Broadcom 5706S) TOE
I/O upgrade	1 PCI-X expansion connector and 1 PCI Express expansion connector

1.2.3 IBM BladeCenter JS21 server

The BladeCenter JS21 server represents the newest offering in the BladeCenter family of high-density, high-performance PowerPC® based servers, with multi-socket processors and high-speed memory options.

Key features of the JS21 include:

- Two 64-bit PowerPC 970MP single-core or dual-core processors.
- Up to 16 GB ECC Chipkill™ DDR2 memory (1 GB or 2 GB standard).
- Dual Gigabit Ethernet controller (Broadcom 5780).

- One slot for an I/O expansion adapter, either standard form factor (StFF) or small form factor (SFF) design.
- Light Path Diagnostics on system board speeds recovery from individual blade server failures.
- Integrated BMC management processor.
- Integrated SAS controller with support for up to two fixed 2.5inch SFF SAS hard disk with support for RAID-0 and RAID-1.

Table 1-4 provides details of the features of the JS21.

Table 1-4 Features of the JS21

Feature	Specification
Processor	PowerPC 970MP processors, 2.3 GHz - 2.7 GHz (processor speed varies by model and also depends on the BladeCenter chassis the server is installed in)
Number of processors (std/max)	2 / 2
Memory	4 DIMM slots / either 400 MHz (PC2-3200) or 533 MHz (PC2-4200) DIMMs / 16 GB maximum
Internal hard disk drives (standard / maximum)	0 / 2
Network	Two ports, Integrated Dual Gigabit Ethernet (Broadcom 5780)
I/O upgrade	1 PCI-X expansion connector (supports standard and small form factor cards)

1.3 BladeCenter H networking architecture

Each BladeCenter chassis can support different types and quantities of I/O modules, and each blade server in the chassis can be connected to the I/O modules by a number of ways. This section gives you an overview of internal I/O architecture and supported combinations of I/O modules and expansion cards for each chassis.

I/O modules can be grouped into several categories:

- Ethernet switch modules (which include both standard and high-speed Ethernet switch modules)
- InfiniBand switch modules (which include standard and high-speed InfiniBand switch modules as well as InfiniBand bridge modules)
- Fibre Channel switch modules (which include standard FC switch modules)
- Pass-through and interconnect modules (which include copper and optical pass-through modules and the Multi-Switch Interconnect Module (MSIM)

The BladeCenter H chassis has two internal fabrics:

- High-speed fabric which is capable of carrying 10 Gbps Ethernet and 4X InfiniBand
- Standard fabric for Fibre Channel, Ethernet, and 1X InfiniBand (also known as low-speed)

The high-speed fabric is only when used you install a high-speed expansion card into a blade server. This card has its own connectors to the midplane and a PCI Express socket on the blade itself. The PCI Express connection is a 2.5 Gbps x4 connection, meaning it can support 10 Gbps connections.

IBM BladeCenter H chassis has a total of ten I/O bays. Each blade bay has a total of eight dedicated connection paths to the I/O modules. See Figure 1-7.

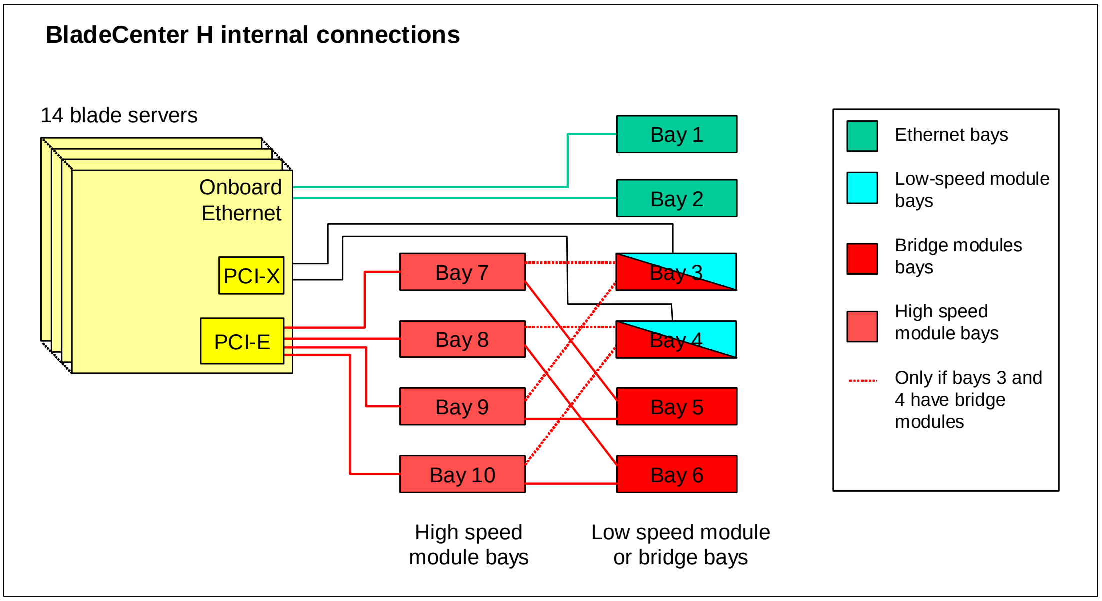


Figure 1-7 BladeCenter H internal connections

The bays are as follows:

- Bays 1 and 2 only support standard Ethernet-compatible I/O modules, which includes Ethernet switch modules as well as optical and fiber pass-thru modules. These are routed internally to the onboard Ethernet controllers on the blades (and slot 1 in the SIO and BSE2 expansion blades).
- Bays 3 and 4 can be used either for standard switch or pass-through modules (such as Fibre Channel connectivity or additional Ethernet ports) or for bridge modules. These are routed internally to the PCI-X connector on the blades.
- Bays 5 and 6 are dedicated for bridge modules only and do not directly connect to the blade bays. Bridge modules provide links to the I/O bays 7 through 10 and can be used as additional outputs for I/O modules in those bays.

  When I/O bays 3 and 4 have bridge modules installed, they are connected the high-speed module bays as shown in Figure 1-7 and are not directly connected to the blades. In this configuration, bay 3 provides redundancy for bay 5, and bay 4 provides redundancy for bay 6.
- I/O bays 7 through 10 are used for high-speed switch modules such as a Cisco 4X InfiniBand Switch Module. this also includes the MSIM, which allows the placement of low speed modules into the high speed slots. In either case, these are routed internally to the PCI Express connector on blades that have it. The Cisco 4X InfiniBand Switch Module is connected to bays 7 and 9 and bays 8 and 10 are not used.

The I/O bays are connected to two separate and redundant midplanes and the blade servers and expansion cards in the blades have ports that connect to both midplanes. The midplanes that connect each of the bays are shown in Table 1-5.

Table 1-5 I/O module bays on each midplane

	Top midplane	Bottom midplane
Standard (low)-speed module bays	Bay 1 Bay 3 Bay 5	Bay 2 Bay 4 Bay 6
High-speed module bays	Bay 7 Bay 8	Bay 9 Bay 10

Be sure that all installed I/O modules are compatible with interface ports present in the blades. For a list of supported combinations of standard I/O modules and I/O expansion cards for BladeCenter H, refer to Table 1-6.

Table 1-6 Supported combinations of standard I/O modules and I/O expansion cards - IBM BladeCenter H

Part	Expansion card	ESM and CPM		IBSM	FCSM	OPM	
	I/O module bay number	1, 2	3, 4	3, 4	3, 4	1, 2	3, 4
None	Integrated Gigabit Ethernet	Yes	•[a]	•	•	Yes	•
39R8624	SFF Gigabit Ethernet Expansion Card	Yes[b]	Yes	No	No	Yes[b]	Yes
39Y9310	Ethernet Expansion Card (CFFv)	Yes[b]	Yes	No	No	Yes[b]	Yes
32R1896	Cisco 1X InfiniBand HCA Expansion Card	•	No	Yes	No	•	No
26K4841	IBM SFF FC Expansion Card	•	No	No	Yes	•	Yes
26R0890	QLogic 4Gb SFF FC Expansion Card	•	No	No	Yes	•	Yes
26R0884	QLogic 4Gb StFF FC Expansion Card	•	No	No	Yes	•	Yes
41Y8527	QLogic 4Gb FC Expansion Card (CFFv)	•	No	No	Yes	•	Yes
39Y9186	Emulex 4Gb SFF FC Expansion Card	•	No	No	Yes	•	Yes
32R1923	QLogic iSCSI Expansion Card	Yes[b]	Yes	No	No	Yes[b]	Yes
73P6000	Myrinet Cluster Expansion Card	•	No	No	No	•	Yes[c]

a. Cells in this table with a • indicate the expansion card that is listed in this row does not impact what can or cannot be used in the I/O module bay in this column.

b. Supported only if the expansion card is installed in slot 1 of either the BladeCenter SCSI Storage Expansion Unit II (part number 39R8625) or BladeCenter Storage and I/O Expansion Unit (part number 39R7563).

c. When used with the Myrinet Cluster Expansion Card, the Optical Pass-thru Module must be installed in I/O bay 4 because the expansion card only has one port.

For list of supported combinations of high-speed modules and high-speed expansion cards for BladeCenter H see Table 1-7.

Table 1-7 Supported combinations of high-speed I/O modules and I/O expansion cards - BladeCenter H

Part	High-speed expansion card	ESM[a,b] CPM[b]	FCSM[b,c]	OPM[b]	HSIBSM	HSESM
	I/O module bay number	**7, 9[d]**	**8, 10[d]**	**7, 8, 9, 10[d]**	**7, 9**	**7, 9**
39Y9271	NetXen 10 Gb Ethernet Expansion Card (CFFh)	No	No	No	No	Yes
39R8624	QLogic Ethernet and 4 Gb FC Exp. Card (CFFh)	Yes	Yes	Yes	No	No
32R1760	Cisco 4X InfiniBand HCA Expansion Card	No	No	No	Yes	No

a. Supported in high-speed I/O bays using MSIM.
b. Nortel Layer 2/7 switch (32R1859) and Nortel 10G uplink switch module (32R1783) are not supported with MSIM.
c. Cisco 4 Gb FC switches (39Y9280 and 39Y9284) are not supported with MSIM.
d. These high-speed I/O bays are converted to the standard I/O bays using MSIM, in which case bay 7 represents the upper left bay of MSIM, bay 8 - upper right, bay 9 - lower left, and bay 10 - lower right.

1.4 High-speed I/O paths in BladeCenter H

As noted, the BM BladeCenter H chassis has two types of fabrics inside:

- Standard fabric (which is almost the same as in the BladeCenter E chassis)
- High-speed fabric which is capable of carrying 10 Gbps Ethernet and 4X InfiniBand

The high-speed fabric is used when you install a high-speed expansion card into a blade server such as the Cisco 4X InfiniBand HCA (see 2.2, “Cisco 4X InfiniBand HCA Expansion Card” on page 18) and the 4X DDR InfiniBand HCAs (see 2.6.2, “InfiniBand DDR HCAs” on page 25).

These cards connect to the blades through a PCI Express x4 socket and provide two x4 (2.5 GBps x4 = 10 GBps) connections to the internal network infrastructure, as shown in Figure 1-8.

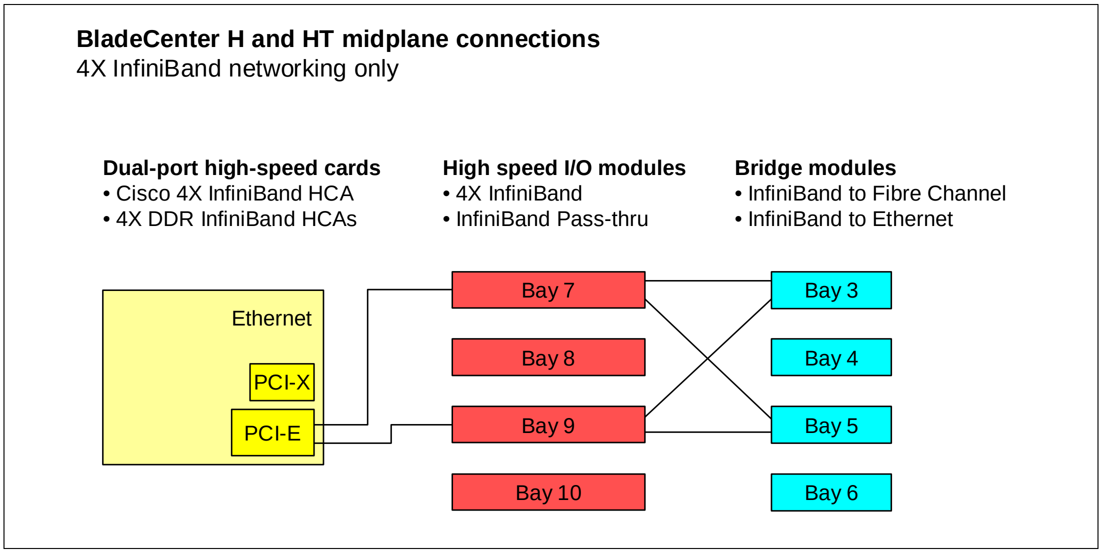


Figure 1-8 4X InfiniBand networking infrastructure

The 4X InfiniBand HCAs installed in the blade servers are PCI Express x4 cards with two output ports that are routed to bays 7 and 9. Two 4X InfiniBand high-speed switch modules are installed in these bays (two modules provide redundancy). Because the 4X InfiniBand HCA is a two-port card and not a four-port card, bays 8 and 10 are not connected.

The bridge module bays used in this configuration are bays 3 and 5, as shown in Figure 1-8. Bridge modules in bays 3 and 5 can be a redundant pair or one can be a InfiniBand-to-Fibre Channel bridge, and the other can be an InfiniBand-to-Ethernet bridge. See Chapter 2, “InfiniBand products for BladeCenter” on page 15 for details about these bridge modules.

Bays 4 and 6 are also suitable for bridge modules, but these bays are not used in this configuration because the 4X InfiniBand HCA does not have a third and fourth port.

2

InfiniBand products for BladeCenter

This chapter discusses the InfiniBand products that are available for the BladeCenter H. From a hardware perspective, this discussion includes the Cisco 4x InfiniBand Switch Module and associated 4x HBA and cables, the QLogic InfiniBand to Ethernet and InfiniBand to Fibre Channel Bridge Modules, and some newly announced DDR InfiniBand products. From a software perspective, this discussion introduces the VFrame for InfiniBand virtualization product.

We also discuss in this chapter the Cisco SFS 3012R Multifabric Server Switch, an external gateway device that offers extensive connectivity to multiple fabrics (Ethernet, Fibre Channel, and InfiniBand).

In this chapter, we discuss the following topics:

- 2.1, “Cisco 4X InfiniBand Switch Module” on page 16
- 2.2, “Cisco 4X InfiniBand HCA Expansion Card” on page 18
- 2.3, “Cisco VFrame for InfiniBand virtualization software” on page 19
- 2.4, “QLogic InfiniBand Ethernet Bridge Module” on page 19
- 2.5, “QLogic InfiniBand Fibre Channel Bridge Module” on page 21
- 2.6, “4X DDR InfiniBand solutions for the IBM BladeCenter H” on page 23
- 2.7, “Cisco SFS 3012R Multifabric Server Switch” on page 26

2.1 Cisco 4X InfiniBand Switch Module

The Cisco 4X InfiniBand Switch Module for IBM BladeCenter, part number 32R1756, adds InfiniBand switching capability to hosts in your IBM BladeCenter H chassis. When you add one or two switch modules to your BladeCenter H chassis and add an HCA expansion card to your blade servers, your servers can communicate to one another over InfiniBand within the chassis. When you connect the switch module to an outside InfiniBand fabric, servers can communicate with all nodes that connect to the InfiniBand network.

This InfiniBand switch for IBM BladeCenter H delivers low-latency, high-bandwidth connectivity (up to 240 Gbps full duplex) between InfiniBand connected internal BladeCenter server blades, additional BladeCenter H chassis, stand-alone servers, and both internal and external gateways for connectivity to Ethernet LANs and Fibre Channel SANs.

Note: Cisco 4X InfiniBand Switch Module is not supported in the BladeCenter HT.

Key features include:

- Up to eight external 4X (10 Gbps) InfiniBand ports (through two 4X connectors and two 12X connectors)
- 16 internal 4X (10 Gbps) InfiniBand ports
- Up to 240 Gbps of aggregate bandwidth available per switch
- Dual switch configurations to provide additional bandwidth, and redundancy

The Cisco 4X InfiniBand Switch Module, shown in Figure 2-1, is installed in either bay 7 or bay 9 of the BladeCenter H or both. The recommendation is to install two modules for redundancy.

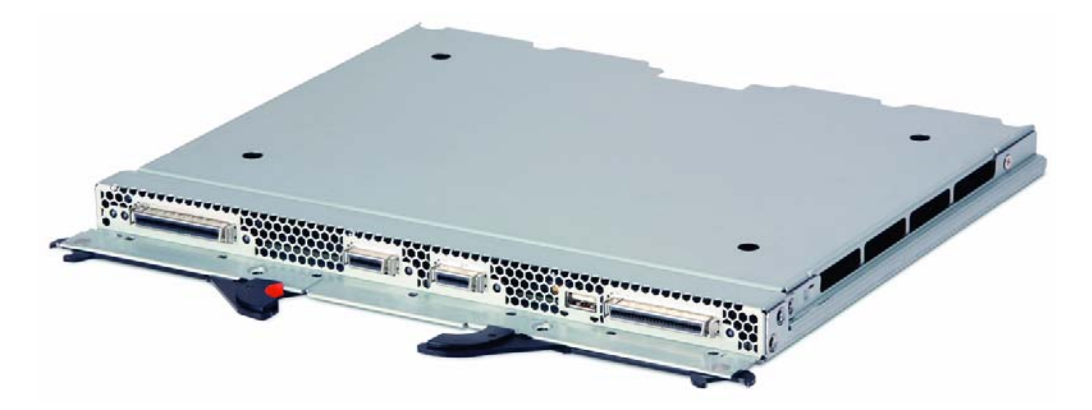


Figure 2-1 Cisco 4X InfiniBand Switch Module

Each switch module includes 16 internal 4x ports to the backplane and eight 4x ports (in the form of two 4x connectors and two 12x connectors) on the front panel. The Cisco InfiniBand Switch Module provides fully non-blocking switching for all 24 ports. On the backplane, 14 of the 16 internal 4x ports provide 10 Gbps connections to the HCA expansion cards on server blades. The two remaining internal 4x ports provide connections to the chassis expansion modules. All external 4x connectors provide 10 Gbps connections to the outside InfiniBand network and can auto negotiate connection speed.

The Cisco 4X InfiniBand Switch Module transmits information to and from BladeCenter management modules over Ethernet (through an internal Ethernet port toward the management modules) to facilitate setup and management. After you set up a switch module and bring it online, the on-board Cisco Subnet Manager brings distributed intelligence to the InfiniBand network.

Within the BladeCenter chassis, Cisco InfiniBand switch modules manage traffic to and from HCA expansion cards on the BladeCenter hosts. Each HCA expansion card adds two InfiniBand ports to a BladeCenter host. Each HCA port connects through the unit backplane to a particular switch module slot. The first InfiniBand port of each HCA card (ib0) connects to the Cisco InfiniBand switch module in I/O slot 7, and the second InfiniBand port of each HCA card (ib1) connects to I/O slot 9.

Figure 2-2 shows the InfiniBand and Management module port connections in the BladeCenter H.

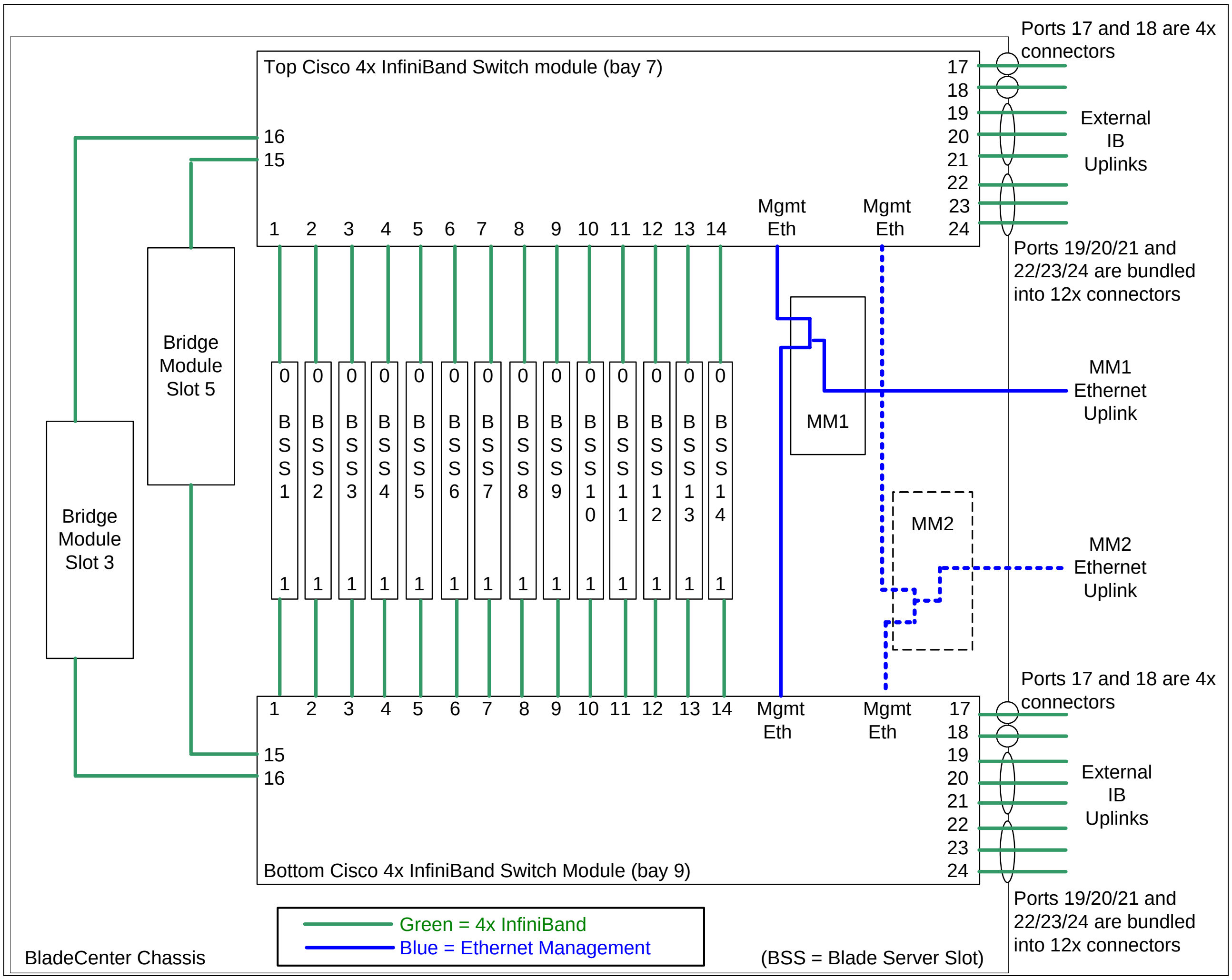


Figure 2-2 Internal BladeCenter InfiniBand connectivity

You can manage your InfiniBand switch module with any of the following interfaces:

- Simple Network Management Protocol (SNMP) versions 1, 2, and 3 with Cisco's Management Information Base (MIBs)
- TopspinOS command line interface (CLI)
- Chassis Manager Web-based GUI
- Element Manager Java™-based GUI
- APIs (through SNMP)

Note: To implement the VFrame solution using the Cisco InfiniBand switch module, you need to purchase Cisco VFrame Server Fabric Virtualization software separately from Cisco resellers.

Table 2-1 shows Cisco 4X InfiniBand switch module-related options:

Table 2-1 Cisco 4X InfiniBand switch module options

Part number	Description
26R0816	InfiniBand 3 meter 12x to (3) 4x Cable for IBM BladeCenter
26R0849	InfiniBand 8 meter 12x to (3) 4x Cable for IBM BladeCenter
26R0813	InfiniBand 3 meter 4x Cable for IBM BladeCenter
26R0847	InfiniBand 8 meter 4x Cable for IBM BladeCenter

For more information, refer to *Cisco 4X InfiniBand Switch Module for IBM BladeCenter User Guide*, which is available at:

http://www.ibm.com/support/docview.wss?uid=psg1MIGR-65966

2.2 Cisco 4X InfiniBand HCA Expansion Card

The Cisco 4X InfiniBand HCA Expansion Card, part number 32R1760, provides connectivity to the Cisco 4X InfiniBand Switch Module (see 2.1, "Cisco 4X InfiniBand Switch Module" on page 16). Together they provide low-latency, high-bandwidth connectivity to other BladeCenter chassis, stand-alone servers and external gateways.

Features include:

- Dual 4x 10 Gbps ports
- Server blade connection to one or two installed InfiniBand Switch Modules
- Ethernet and Fibre Channel connectivity for each InfiniBand connected server blade, through external gateways

The Cisco 4X InfiniBand HCA Expansion Card is a High Speed Form Factor (HSFF) card (see Figure 2-3). It requires that 4X InfiniBand Switch Modules be installed in bay 7 or bay 9 (or both, for redundancy).

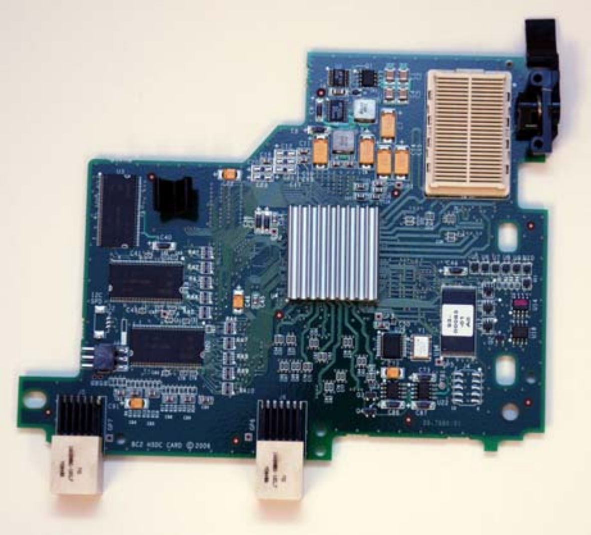


Figure 2-3 Cisco 4X InfiniBand HCA Expansion Card

2.3 Cisco VFrame for InfiniBand virtualization software

Cisco VFrame for InfiniBand virtualization software used in conjunction with Cisco InfiniBand switches and gateways provide server and I/O virtualization capabilities for customers. VFrame programs and coordinates InfiniBand switches to alter dynamically how a server is provisioned, by mapping a physical server to a logical server identity stored in the fabric.

There are two major components of the VFrame virtualization solution:

- Cisco VFrame Server Fabric Virtualization Software 3.1
- Cisco SFS 3000 Series Multifabric Server Switches (3001/3012) and Cisco SFS Server Fabric Switches (including the Cisco 4x InfiniBand Switch Modules for IBM BladeCenter H

You can find more details at:

http://www.cisco.com/en/US/products/ps6429

Contact your Cisco sales representative to purchase the VFrame 3.1 solution.

We do not discuss VFrame in this paper.

2.4 QLogic InfiniBand Ethernet Bridge Module

The QLogic InfiniBand to Ethernet Bridge module, part number 39Y9207, can provide Ethernet connectivity to InfiniBand clients within the BladeCenter H. It can currently be installed in switch bays 3 and 5 (we recommend both bays for redundancy). Its use requires the installation of Cisco 4X InfiniBand Switch Module in bays 7 and 9 (again for redundancy), and the 4X InfiniBand HCA on each blade server that wants to make use of the bridge module.

> **Note:** If the bridge module is installed in bay 4 or 6 there will not be an internal InfiniBand connection because the 4X InfiniBand HCA does not have a third and fourth port routed to bays 8 and 10. See 1.4, “High-speed I/O paths in BladeCenter H” on page 13.

The Ethernet Bridge Module offers six 10/100/1000 Mbps external Ethernet RJ45 ports to connect to an upstream Ethernet network, and two 4x internal InfiniBand ports to connect to the 4x switch modules in slot 7 and 9. See Figure 2-4. It has support for many of the common Ethernet based technologies, such as 802.3ad link aggregation, 802.1Q VLAN trunking and jumbo frames. Management is through either a browser-based GUI interface or a CLI-based interface, with management connectivity coming through Ethernet ports that connect from the bridge module to the BladeCenter H Management Module slots.

This product can help simplify server deployments by allowing the BladeServer to only have a single physical interconnect (InfiniBand) and still have access to other technologies, such as Ethernet with this module or Fibre Channel through the QLogic InfiniBand to FC bridge module.

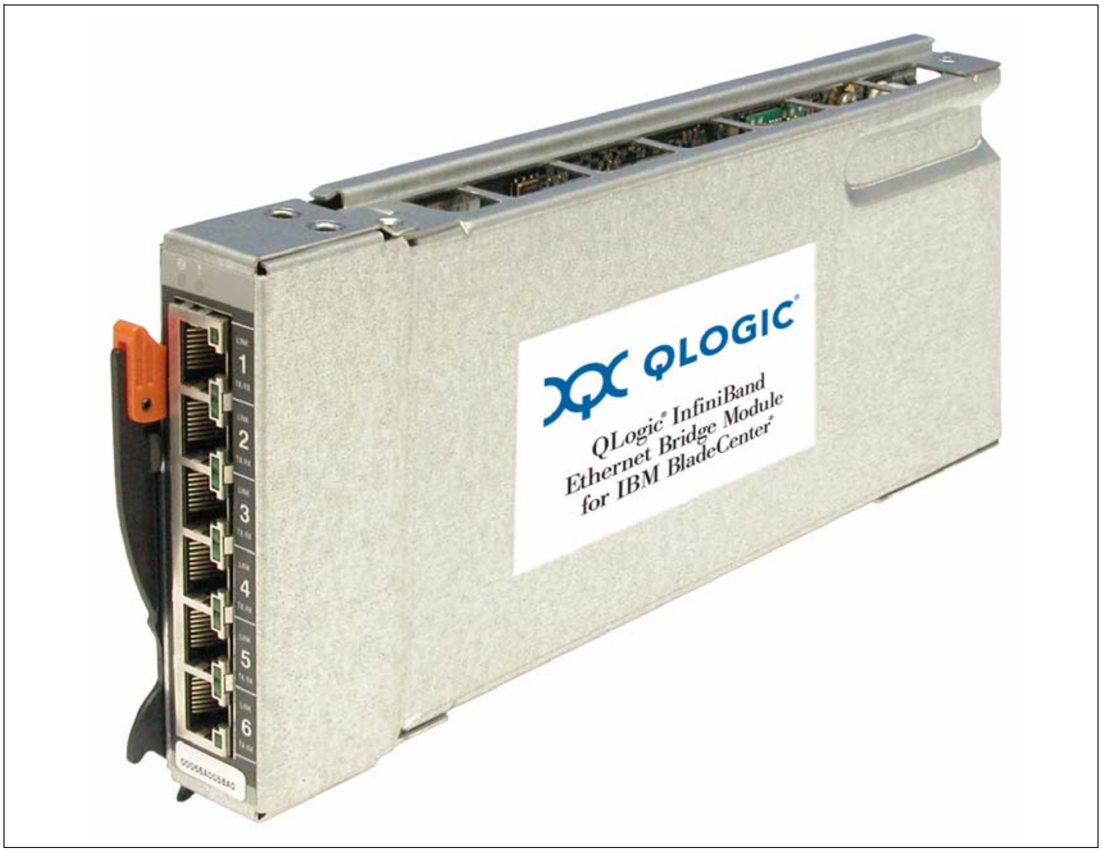


Figure 2-4 QLogic InfiniBand Ethernet Bridge Module

The Ethernet bridge module has the following features and specifications:

- Support for two internal InfiniBand double-data rate (DDR) capable 4X (20 Gbps) links to the high-speed switch modules (HSSMs)
- Support for various types of transport data traffic:
  - Reliable
  - Unreliable
  - Connected

  - Unconnected
- Six external autosensing 10/100/1000 Mbps RJ-45 Ethernet (copper) ports
- 802.3ad link aggregation
- Support for jumbo frames
- Support of IEEE 802.1q VLAN tagging
- Support for up to 1150 Virtual NIC ports per module
- Automatic port and module failover
- TCP/UDP and IP header checksum offload and host checking
- 802.1p Priority Queuing/Scheduling
- Switched internal I2C Interface to the management modules
- Two internal 100 Mbps full-duplex Ethernet links to the management modules
- Power-on diagnostics and status reporting
- Support of simple network management protocol (SNMP) management information bases (MIBs) and traps through the Ethernet management ports

Ethernet Bridge Module supports the following management methods:

- CLI through telnet or SSH
- Web-based access with QLogic's Chassis Viewer
- SNMP/MIB

For more information, refer to the following documentation:

- *QLogic InfiniBand Ethernet Bridge Module and Fibre Channel Bridge Module Installation Guide - IBM BladeCenter H (Type 8852)*

  http://www.ibm.com/support/docview.wss?uid=psg1MIGR-5070108

- *InfiniBand Fibre Channel Bridge Module for IBM BladeCenter & InfiniBand Ethernet Bridge Module for IBM BladeCenter - Support*

  http://support.qlogic.com/support/oem_detail_all.asp?oemid=377

2.5 QLogic InfiniBand Fibre Channel Bridge Module

The QLogic InfiniBand to Fibre Channel Bridge module, part number 39Y9211, can provide Fibre Channel connectivity to InfiniBand clients within the BladeCenter H. It uses a concept of Virtual HBAs to virtualize InfiniBand fabric to provide blade server's access to the resources resided in traditional SANs. Like the Ethernet Bridge Module, it can be installed currently in switch bays 3 and 5 (use two for redundancy) and its use requires the installation of a Cisco InfiniBand 4X switch module in bays 7 and 9 (use two for redundancy and 4X InfiniBand HCA on each blade server that wants to make use of the bridge module.

> **Note:** If the bridge module is installed in bay 4 or 6 there will not be an internal InfiniBand connection because the 4X InfiniBand HCA does not have a third and fourth port routed to bays 8 and 10. See 1.4, "High-speed I/O paths in BladeCenter H" on page 13.

The Fibre Channel Bridge Module offers six 1/2/4 Gbps external FC SFP ports to connect to an upstream Fibre Channel network, and two 4x internal InfiniBand ports to connect to the 4x switch modules in slot 7 and 9. See Figure 2-5. Management is through either a browser-based GUI interface or a CLI-based interface, with management connectivity coming through Ethernet ports that connect from the bridge module to the BladeCenter H Management Module slots.

As with the Ethernet Bridge Module, this product can help simplify server deployments by allowing the BladeServer to only have a single physical interconnect (InfiniBand) and still have access to other technologies, such as Fibre Channel through the QLogic InfiniBand to FC bridge module.

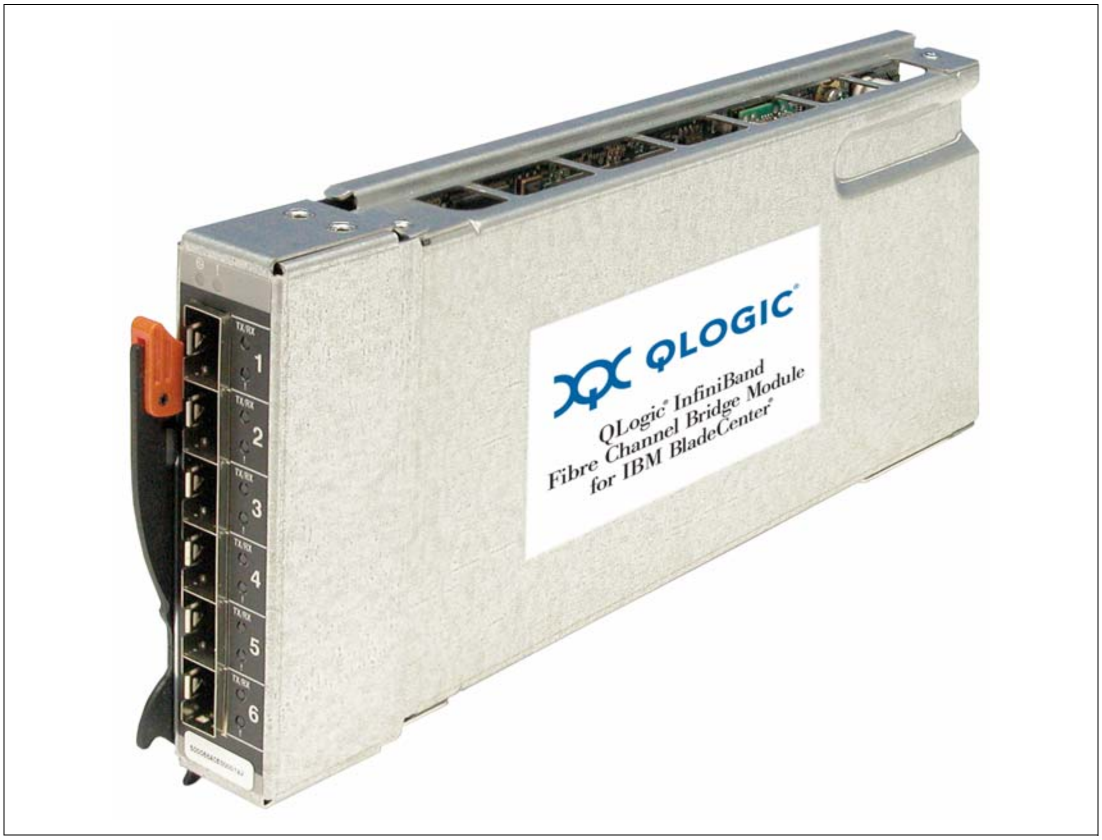


Figure 2-5 QLogic InfiniBand Fibre Channel Bridge Module

The bridge module has the following features and specifications:

- Support for two internal InfiniBand double-data rate (DDR) capable 4X (20 Gbps) links to the high-speed switch modules (HSSMs)
- Support for various types of transport data traffic:
  - Reliable
  - Unreliable
  - Connected
  - Unconnected
- Six external autosensing Fibre Channel 1/2/4 Gbps ports using standard small form-factor pluggable (SFP) connectors (SFPs are not included)
- All common Fibre Channel topologies supported (direct attachment, switch, arbitrated loop)
- Supports up to 128 virtual host bus adapter (HBA) ports per module
- Automatic port and module failover
- Load balancing and port aggregation

- Logical unit number (LUN) mapping and masking
- SCSI-SRP, SCSI-FCP, and FC-PH-3 compliant
- Switched internal I2C Interface to the management modules
- Two internal 100 Mbps full-duplex Ethernet links to the management modules
- Power-on diagnostics and status reporting
- Support of simple network management protocol (SNMP) management information bases (MIBs) and traps through the Ethernet management ports

The bridge module supports the following management methods:

- CLI through telnet or SSH
- Web-based access with QLogic's Chassis Viewer
- SNMP/MIB

This module ships standard without SFPs. You need to order them additionally (see Table 2-2).

Table 2-2 Supported SFPs for QLogic InfiniBand Fibre Channel Bridge Module

Part Number	Description
22R4897	4 Gbps SW SFP Transceiver 4 Pack
22R4902	4 Gbps SW SFP Transceiver
19K1271	2 Gb Fibre Channel Short Wave SFP Module
22R0483	2 Gb Fibre Channel Short Wave SFP 4 pack
19K1272	2 Gb Fibre Channel Long Wave SFP Module
22R0484	2 Gb Fibre Channel Long Wave SFP 4 pack

For more information, refer to the following documentation:

- *QLogic InfiniBand Ethernet Bridge Module and Fibre Channel Bridge Module Installation Guide - IBM BladeCenter H (Type 8852)*

  http://www.ibm.com/support/docview.wss?uid=psg1MIGR-5070108

- *InfiniBand Fibre Channel Bridge Module for IBM BladeCenter & InfiniBand Ethernet Bridge Module for IBM BladeCenter - Support*

  http://support.qlogic.com/support/oem_detail_all.asp?oemid=377

2.6 4X DDR InfiniBand solutions for the IBM BladeCenter H

IBM has recently announced several new 4X Double Data Rate (DDR) InfiniBand solutions. These include a 4X InfiniBand Pass-thru Module and a new dual port 4X DDR HCA that comes in three flavors. When used together they offer 4X DDR performance for blade servers as well as full non-blocking access on each and every HCA port attaching to the InfiniBand Pass-thru Module.

2.6.1 InfiniBand 4X DDR Pass-thru Module

With 14 InfiniBand 4X DDR ports towards the servers and 14 InfiniBand 4X DDR ports toward the upstream network, the 4X InfiniBand DDR Pass-thru Module, part number 43W4419, offers full non-blocking 4X DDR support to all 14 blade servers in a BladeCenter H chassis.

The InfiniBand Pass-thru Module is a double-height module, and up to two can be installed in an IBM BladeCenter H, utilizing either switch bays 7 and 8, or switch bays 9 and 10 in the rear of the chassis.

Figure 2-6 shows the rear of an IBM BladeCenter H with Pass-Thru Modules installed in bays 7 and 8 and 9 and 10.

> **Note:** Although the Pass-Thru Module is a double-high module (to physically fit all the external connectors), it is only connected to the upper module bay. In other words, the module connects to bay 7 and bay 9. It does not connect to bay 8 and bay 10.

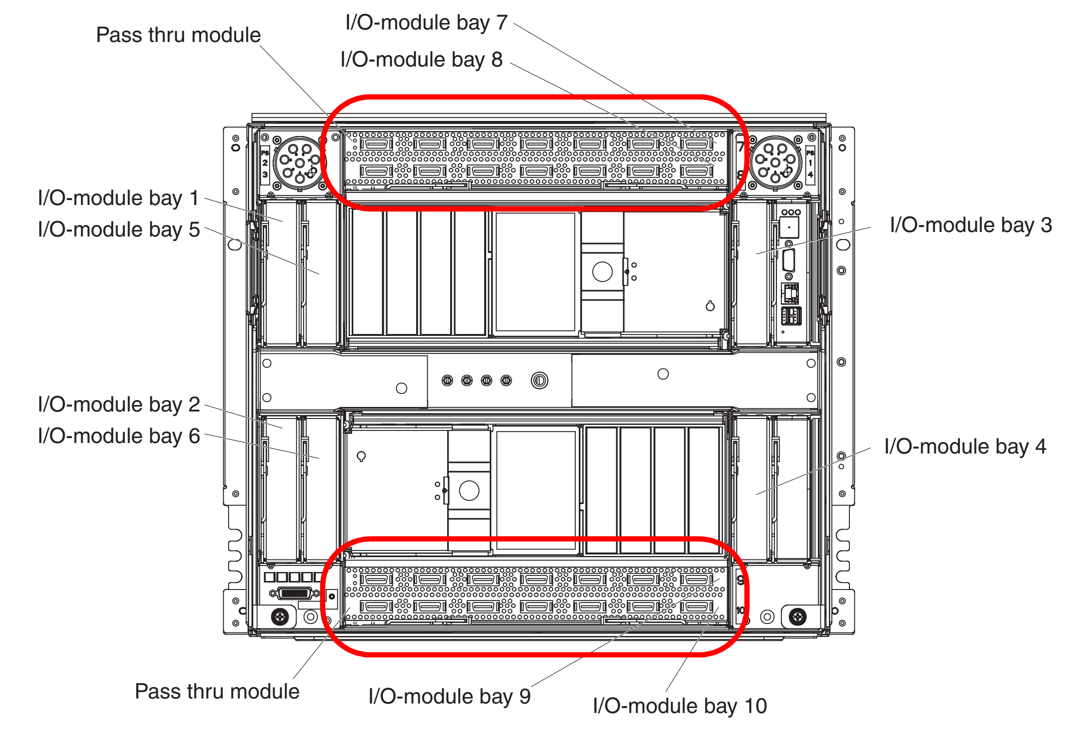


Figure 2-6 Pass-thru Modules in High Speed Switch bays

Some important pass-thru module features:

- Provides a pass-through connection for 14 InfiniBand 4X DDR ports
- External ports: 14 copper InfiniBand 4X connectors
- Internal ports: 14 InfiniBand 4X ports connected to the mid-plane an on to the server slots
- Uplink ports that support hot-pluggable media converter for optical cable
- Two status LED's: *OK* (Green) and *Fault* (Yellow)

Note: The QLogic Bridge modules are not compatible with the pass-thru module, because there is no connectivity from the module to the bridge slots. Also, unlike other InfiniBand switch modules for the BladeCenter, if InfiniBand communications are desired between blade servers in the same chassis, some sort of external InfiniBand connectivity must be provided.

2.6.2 InfiniBand DDR HCAs

Three recently added 4X DDR HCAs are also available, resulting from a partnership between IBM and Mellanox. Software differentiates the three offerings, in that they use a common CFFh base HCA, and then depending on what part number are ordered, will come with support for different software.

These HCAs require that the InfiniBand Pass-thru Module (or other supported InfiniBand switch module) are installed in bay 7 or bay 9 of the BladeCenter H chassis, or both bays 7 and 9 for redundancy.

The details on these options are as follows:

- 4X InfiniBand DDR Expansion Card (43W4423)

  For this HCA the user must independently obtain drivers. This will initially offer limited support from IBM, with OFED support being available from the OFED web site

- Cisco 4X InfiniBand DDR Expansion Card (43W4421)

  This version of the HCA will come with software and support from Cisco, and comes with a license to use the Cisco OFED stack. The software is based on the OFED stack with enhancements and improved support to interface with Cisco switches and gateway products. In the future, Cisco plans to also support their Commercial stack for this HCA

- Voltaire 4X InfiniBand DDR Expansion Card (43W4420)

  This version of the HCA will come with software and support from Voltaire, and comes with a license to use the Voltaire GridStack product. The software is also based on the OFED stack with enhancements and improved support to interface with Voltaire switches

Key features include:

- 1.2µs MPI ping latency
- Two 4x DDR capable (20 Gbps) InfiniBand ports; support auto-negotiation to SDR
- CPU offload of transport operations
- End-to-end QoS and congestion control
- Hardware-based I/O virtualization
- TCP/UDP/IP stateless offload
- Based on Mellanox's new ConnectX InfiniBand to PCI-express bridge chip
- Supports the OpenFabrics Enterprise Distribution (OFED) drivers and protocol support
- Support for Cisco's and Voltaire's drivers stacks

Additional options include:

- InfiniBand 3m DDR Cable (43W6742)
- InfiniBand 8m DDR Cable (43W7243)

2.6.3 Compatibility with other InfiniBand options for BladeCenter

Initially the DDR expansion cards are supported only with the InfiniBand Pass-thru Module. Similarly, initially, the InfiniBand Pass-thru Module is supported only when connected to DDR

HCAs. However, there are plans to support both the existing Cisco 4X InfiniBand HCA, and the Cisco 4X InfiniBand Switch Module. Check IBM ServerProven® for the latest support information:

http://www.ibm.com/servers/eserver/serverproven/compat/us/

2.7 Cisco SFS 3012R Multifabric Server Switch

While technically not part of the IBM BladeCenter H systems, the SFS 3012R is an important component of any InfiniBand deployment, by offering Multifabric I/O (MFIO) to InfiniBand clients.

Many users would like to be able to allow their InfiniBand clients access to Ethernet and Fibre Channel networks, without the need to install Ethernet or Fibre Channel cards in their servers or clients (or Ethernet and Fibre Channel switches into the BladeCenter H). Using a 3012R with various gateway modules, it is possible to deploy servers and clients with only a single connection to the InfiniBand fabric, full access to your Ethernet and Fibre Channel networks. This results in simpler client configurations and cabling which can lead to reduced Total Cost of Ownership (TCO).

Figure 2-7 shows an SFS 3012R fully loaded with 6 Fibre Channel gateways and 6 Ethernet Gateways, two 12 port InfiniBand cards and redundant controllers. Note that you can have any combination of FC and Ethernet gateways, up to and including 12 Ethernet gateway models or 12 FC modules.

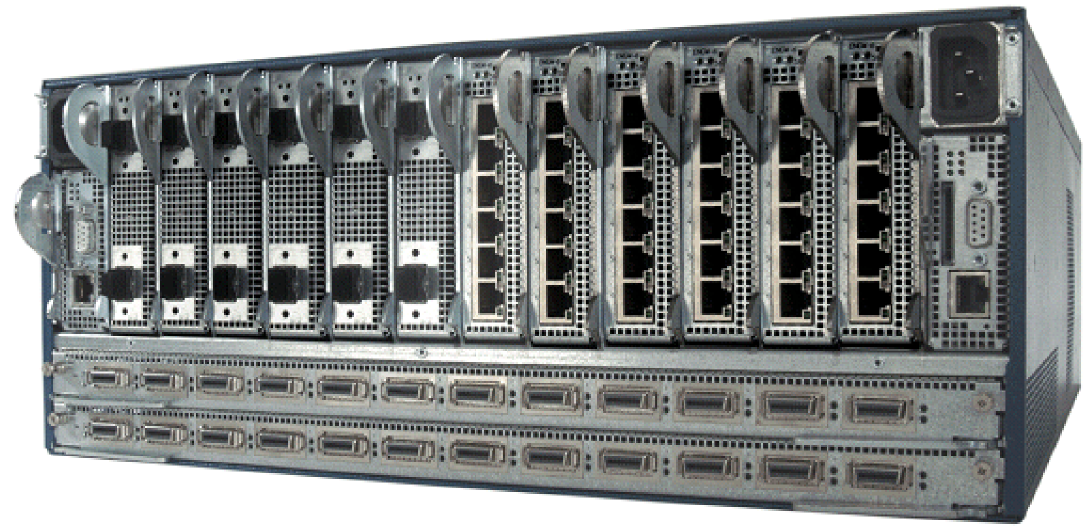


Figure 2-7 Rear view of the SFS 3012R - Fully loaded with gateway modules

Figure 2-8 shows the individual Fibre Channel and Ethernet gateway modules. The Fibre Channel gateway module offers two 1/2G Fibre Channel ports to the SAN network, and two 4X (10 Gbps) InfiniBand ports to the internal InfiniBand fabric. The Ethernet gateway module offers six 10/100/1000G Ethernet ports to the LAN, and two 4X (10 Gbps) InfiniBand ports to the internal InfiniBand fabric.

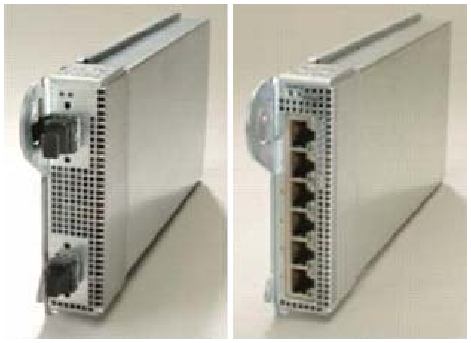


Figure 2-8 Left - 2 port Fibre Channel gateway module, right - 6 port Ethernet gateway module

The Cisco SFS 3012R is designed with enterprise-class redundancy in mind. By combining dynamic load balancing and port aggregation, The Cisco SFS 3012R minimizes failure points in switch blades, controllers, gateways, and ports. All removable components are also hot-swappable, including controllers, gateway modules, power, and cooling.

Configuration and maintenance are simplified with a number of different management tools, including an Element Manager GUI (Java based), Chassis Manager GUI (Web based), CLI using serial console, Telnet, or Secure Shell (SSH) protocols, all of which allow for robust remote monitoring, upgrades, and troubleshooting. The Cisco SFS 3012R is also fully compatible with applications that support Simple Network Management Protocol (SNMP), allowing IT managers to reduce management time and TCO.

Also offered is the ability to fail over to gateway modules located in different 3012R chassis, thus offering maximum redundancy.

Some key features and benefits of the Ethernet gateway module include:

- Virtual I/O for Ethernet
  - Allows a group of servers to share a pool of centralized Ethernet I/O resources. Translates between IP over InfiniBand and IP over Ethernet at the gateway
  - Allows an InfiniBand-attached host to seamlessly join an existing IP subnet
- Full IPv4 Multicast Support
  - Enables multicast-enabled applications across the InfiniBand network
- Loop Protection
  - Choose from a variety of flexible options to prevent broadcast loops
- Jumbo Frame Support
  - Support for up to 9k Ethernet frames and wire-speed IP fragmentation
- VLAN and Partition Support
  - Transparent support for VLANs on the InfiniBand network while maintaining existing business and security rules
- Link Aggregation
  - Combines multiple ports to optimize use of aggregate bandwidth, as well as high availability
  - Supports a variety of metrics, including source/ destination IP, source/destination MAC, and round-robin

- Load Distribution
  - Supports redundancy groups across multiple gateways and multiple chassis
- High-Availability
  - Options Flexible deployment in active-active or active-passive modes to eliminate single points of failure
- DHCP Relay Support
  - Allows DHCP to work across Ethernet and InfiniBand fabrics

The following are some key features and benefits of the Fibre Channel gateway module

- Virtual I/O for Fibre Channel
  - Allows a group of servers to share a pool of centralized Fibre Channel I/O resources. Translates between SCSI over InfiniBand (SRP) and FCP at the gateway
  - Allows an SRP initiator to concurrently talk through multiple shared connections
- Topology Transparency
  - Creates unique World-Wide-Names for every virtual HBA, enabling InfiniBand-attached hosts to seamless connect with existing Fibre Channel storage and management tools
- Failover/Failback
  - Enables sessions to failover and failback
- Multipathing Support
  - Seamless support for existing multipathing tools, including EMC Powerpath and others
- Load Distribution
  - Centralized connection manager dynamically distributes sessions across multiple gateways
- Storage Access Controls
  - Compatible with existing switch-based zoning and LUN-based access controls
  - Multifabric Server Switch includes support for port and LUN access controls through storage management GUI
- Storage Traffic Monitoring
  - Creates graphs and reports on storage performance statistics on individual or aggregated ports

Table 2-3 lists the SFS 3012s product specifications.

Table 2-3 SFS 3012R product specifications

Feature	Description
Cards/Ports/Slots	Two 12-port InfiniBand 4X switch blades Twelve modular expansion slots (any combination Fibre Channel or Ethernet) Redundant (active/standby) control processor modules with One RS-232 serial port and One Ethernet management port

Feature	Description
Reliability and Availability	Redundant (active/standby) control processor modules Redundant, hot-swappable AC power supply modules and cooling Dual AC inputs Hot-plug expansion modules Fully passive backplane Deployable in redundant pairs
Physical Dimensions	Rack-mountable in a standard 19 inch EIA rack 7 inch (4RU) height 24 inch depth 30–95 lb
Approvals and Compliance	FCC: CFR 47 Part 15, Subpart B Class A, UL60950, 3rd ed. ICES-003 Issue 2, CSA 22.2 No. 60950:2000 EN 61000-3-2 (Harmonics), EN 61000-3-3 (Flicker), EN 55022:1998, EN 55024:1998; EN61000-4-1,2,3,4,5,6,8,11, IEC60950, EN60950, EN60825-1 and EN60825-2 VCCI-V3/02.04, IEC60950
Acoustics	Sound Pressure: 38dB at 25C ambient ISO 7779 and section 8.5 of ISO 3744:1994(E) Sound Power: 50dB at 25C ambient ISO 7779, section 8.6 of ISO 3744:1994(E)
Power	Autoranging 90 to 264 VAC, 47 to 63 Hz Power dissipation <650W depending on number of gateway modules

Table 2-4 lists the SFS 3012R InfiniBand switch card specifications.

Table 2-4 SFS 3012R InfiniBand switch card specifications

Feature	Description
Ports	12 4X InfiniBand Ports, 10 Gbps each
Connectors	12 CX4 4X InfiniBand Connectors
Performance	10 Gbps line speed, full duplex
IB Protocols	IPoIB, SDP, SRP, uDAPL, MPI

Table 2-5 lists the SFS 3012R Ethernet gateway module specifications.

Table 2-5 SFS 3012R Ethernet gateway specifications

Feature	Description
Ports	Ports consist of the following ► Six Gigabit Ethernet ports per gateway ► Two internal 10 Gbps InfiniBand ports
Connector	Six external copper RJ-45
Performance	Ethernet performance ► 1 Gbps line speed per port, full duplex ► 12 Gbps aggregate throughput per module (6 Gbps full duplex) ► 144 Gbps aggregate per SFS 3012

Feature	Description
Link Speed Negotiation	Automatic, 10/100/1000 Mbps
IP Protocols	Transparent Topology emulation; IP over InfiniBand (IPoIB)

Table 2-6 lists the SFS 3012R Fibre channel gateway module specifications.

Table 2-6 SFS 3012R Fibre Channel gateway specifications

Feature	Description
Ports	Ports consist of the following ▶ Two 2-Gbps Fibre Channel ports per gateway ▶ Two internal 10 Gbps InfiniBand Ports
Connector	Two external Small Form-Factor Pluggable LC
Performance	Up to 400 MB/s per channel full duplex, Unique memory-optimized architecture
Protocols	▶ Translates between SRP FCP concurrently ▶ Fibre Channel tape support
Fibre Channel Compatibility	▶ Transparent Topology Emulation ▶ NL_Port
Link Speed Negotiation	1 or 2-Gbps Autosensing
Class of Service	Class-3 Fibre Channel service

You can find more information about the SFS 3000 series products at t:

http://www.cisco.com/en/US/products/ps6422/prod_literature.html

Note: Some documentation refers to an *SFS 3012* (and not an *SFS 3012R*). The 3012R is a Restriction of the Use of Certain Hazardous Substances (RoHS) version of the 3012 and has replaced the 3012 as a shipping product. All information and specifications for the 3012 and 3012R remain the same.

Contact your Cisco sales representative to purchase the SFS 3012R Multifabric Server Switch.

3

InfiniBand technology

InfiniBand is an industry standard, switch based serial I/O interconnect architecture that features high speed, low latency interconnects. InfiniBand has the ability to combine networks into a single unified fabric that maximizes bandwidth and is easily scalable. Similar to an Ethernet network, InfiniBand resources can be seperated into function specific subnets. InfiniBand provides Quality of Service (QoS) and Reliability and Servicability (RAS).

This chapter has the following topics:

3.1 InfiniBand Network Layer Model

InfiniBand uses a multi-layer architecture (similar to the seven layer OSI model) to transfer data between nodes, as shown in Figure 3-1. Each layer is responsible for separate tasks in passing messages.

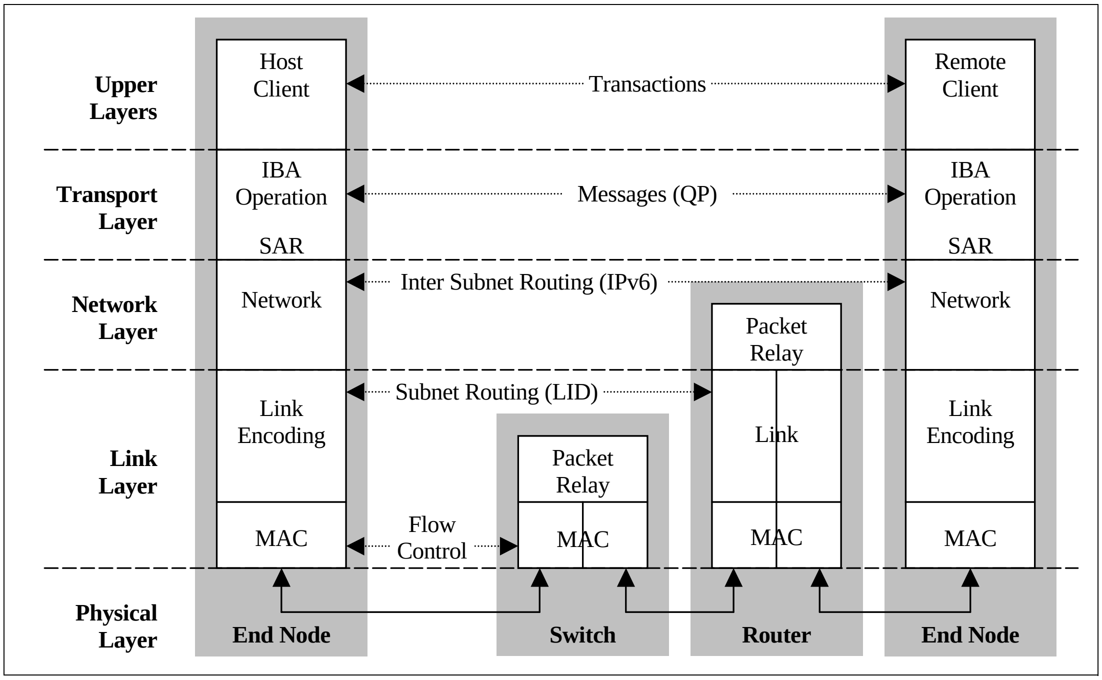


Figure 3-1 InfiniBand Network Layers

3.1.1 Upper layer protocols

The *upper layer protocols* are closest to the operating system and dictate the user applications that can use the InfiniBand host node. They also define how much overhead is needed for data transfer. We define the more common protocols in 3.2, “InfiniBand protocols” on page 35.

3.1.2 Transport layer

The *transport layer* handles transaction data segmentation when sending, and reassembly when receiving. When sending, the transport layer splits messages into packets of up to 4k bytes. It encapsulates the 64-bit Globally Unique ID (GUID) of the destination node with the data of each packet and passes it down to the network layer.

The transport layer is responsible for in-order packet delivery, partitioning, channel multiplexing, and transport services (reliable connection, reliable datagram, unreliable connection, unreliable datagram, raw datagram). Based on the maximum transfer unit (MTU) of the path, the transport layer divides the data into packets of the proper size. The receiver reassembles the packets based on a base transport header (BTH) that contains the destination queue pair and packet sequence number. The receiver acknowledges the packets and the sender receives these acknowledgements and updates the completion queue with the status of the operation. InfiniBand Architecture offers a significant improvement for the transport layer: all functions are implemented in hardware.

InfiniBand specifies multiple transport services for data reliability. Table 3-1 describes each of the supported services. For a given queue pair, one transport level is used.

Table 3-1 Transport services

Class of service	Description
Reliable connection	Acknowledged - connection oriented
Reliable datagram	Acknowledged - multiplexed
Unreliable connection	Unacknowledged - connection oriented
Unreliable datagram	Unacknowledged - connectionless
Raw datagram	Unacknowledged - connectionless

3.1.3 Network layer

The *network layer* handles routing of packets from one subnet to another. (Within a subnet, the network layer is not required.) Packets that are sent between subnets contain a global route header (GRH). The GRH contains the 128-bit IPv6 address for the source and destination of the packet. The packets are forwarded between subnets through a router, based on each device's 64-bit GUID. The router modifies the LRH with the proper local address within each subnet. Therefore the last router in the path replaces the local ID (LID) in the local route header (LRH) with the LID of the destination port. Within the network layer, InfiniBand packets do not require the network layer information and header overhead when used within a single subnet (which is a likely scenario for InfiniBand system area networks).

3.1.4 Data link layer

The *data link layer* encompasses packet layout, point-to-point link operations, and switching within a local subnet. Each device within the subnet has a 16-bit LID assigned to it by the subnet manager which is used for addressing. The data link layer forwards packets to the device specified by the destination LID within an LRH.

Packets

There are two types of packets in the link layer: management packets and data packets. *Management packets* are used for link configuration and maintenance. Device information, such as virtual lane support, is determined with management packets. *Data packets* carry up to 4 KB of a transaction payload.

Switching

Within a subnet, packet forwarding and switching is handled at the link layer. All devices within a subnet have a 16-bit LID assigned by the subnet manager. All packets sent within a subnet use the LID for addressing. Link Level switching forwards packets to the device specified by a destination LID within an LRH in the packet. The LRH is present in all packets.

QoS

QoS is supported by InfiniBand through virtual lanes (VL). These VLs are separate logical communication links that share a single physical link. Each link can support up to 15 standard VLs and one management lane (VL 15). VL15 is the highest priority and VL0 is the lowest. Management packets use VL15 exclusively. Each device must support a minimum of VL0 and VL15 while other VLs are optional.

As a packet traverses the subnet, a service level (SL) is defined to ensure its QoS level. Each link along a path can have a different VL, and the SL provides each link a desired priority of communication. Each switch/router has an SL-to-VL mapping table that is set by the subnet manager to keep the proper priority with the number of VLs supported on each link. Therefore, the InfiniBand Architecture can ensure end-to-end QoS through switches, routers, and over the long haul.

Credit-based flow control

Link-level flow control is used to manage data flow between two point-to-point links. Flow control is handled on a per-VL basis, allowing separate virtual fabrics to maintain communication utilizing the same physical media. Each receiving end of a link supplies credits to the sending device on the link to specify the amount of data that can be received without loss of data. Credit passing between each device is managed by a dedicated link packet to update the number of data packets the receiver can accept. Data is not transmitted unless the receiver advertises credits indicating that receive buffer space is available.

Data integrity

At the link level there are two CRCs per packet, variant CRC (VCRC) and invariant CRC (ICRC), that ensure data integrity. The 16-bit VCRC includes all fields in the packet and is recalculated at each hop. The 32-bit ICRC covers only the fields that do not change from hop to hop. The VCRC provides link-level data integrity between two hops and the ICRC provides end-to-end data integrity. In a protocol such as Ethernet, which defines only a single CRC, an error can be introduced within a device that then recalculates the CRC. The check at the next hop would reveal a valid CRC even though the data has been corrupted. InfiniBand includes the ICRC so that when a bit error is introduced, the error will always be detected.

3.1.5 Physical layer

The *physical layer* transforms the packet into the signal that will be passed over the physical interconnect. The InfiniBand interconnect uses bidirectional serial links between nodes. The base speed of an InfiniBand link is 2.5 Gbps. This base speed is referred to as *1X*. Bandwidth can be increased by increasing the number of channels per link with 4X and 12X connections for link speeds of 10 Gbps and 30 Gbps respectively. The InfiniBand Architecture also supports dual and quad data rates for speeds of up to 120 Gbps (12X link at quad data rate).

The InfiniBand specification also defines electrical and mechanical characteristics for copper and fibre optic cables and connectors.

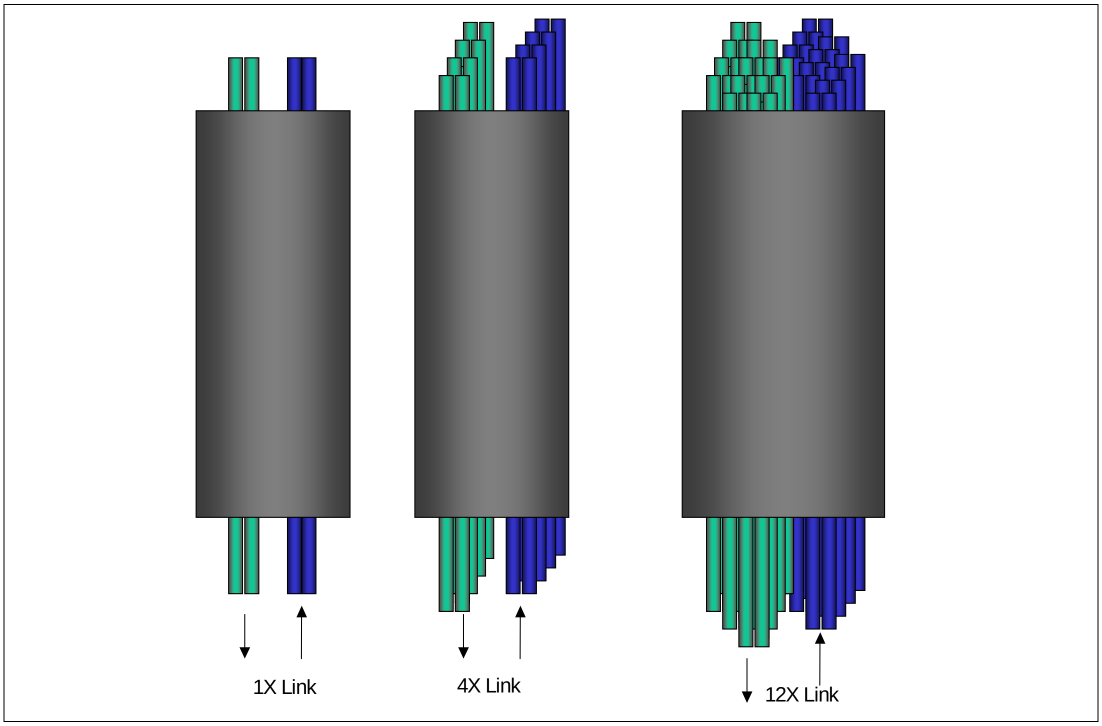


Figure 3-2 InfiniBand physical link

3.2 InfiniBand protocols

InfiniBand supports a number of upper-layer protocols (ULPs) that enable InfiniBand to be exploited by different types of software with different requirements and objectives.

3.2.1 Internet Protocol over InfiniBand

Internet Protocol over InfiniBand (IPoIB) is the lowest-level existing network interface that is provided on InfiniBand. IPoIB supports a wide variety of industry-standard applications, middleware, and operating systems. Any communications protocol that is layered on top of IP (in other words, anything that communicates using IP-based protocols such as TCP/IP or UDP/IP), can be transported over InfiniBand through the IPoIB interface.

3.2.2 Sockets Direct Protocol

Sockets Direct Protocol (SDP) provides a legacy interface somewhat higher up the protocol stack. For applications written to a TCP sockets interface, there is no need to implement all of the TCP and IP protocol layers in order to communicate over InfiniBand. The SDP protocol provides an *asynchronous sockets* interface through which applications and middleware can communicate without needing to understand that there might be no TCP or IP protocol layers underneath.

The difference between synchronous sockets and asynchronous sockets is small, but it is critically important to SDP. It involves the details of how application software is programmed to request communications and how it issues the request (a call to the sockets interface). Synchronous sockets calls from an application require that upon return from the call, the requested communications have completed, and any data buffers involved are no longer needed and can be reused immediately. Asynchronous sockets can (and generally do) enqueue the requested work, and return immediately with the request not completed (often, not even started) and the data buffers still in use.

The use of asynchronous sockets is standard under the Windows® operating systems. Sockets-based communications for Windows programs is always asynchronous, and therefore requires no modification to operate over SDP. Historically, UNIX® and Linux® applications have used synchronous sockets and, therefore, do not map transparently into the SDP protocol. To accommodate this mismatch, a synchronous-to-asynchronous conversion process can be provided between the sockets interface to the application (synchronous) and the SDP interface (asynchronous). This conversion can involve delaying the return from the call, copying data buffers, or other methods that trade some performance and efficiency to achieve full compatibility with existing applications.

In addition, synchronous sockets are being developed and exploited in UNIX and Linux environments today. As they become more standard, software will exploit this interface and will map more easily onto higher-efficiency transports through the use of SDP or similar protocols. With many applications already being cross-compiled onto alternate operating systems and platforms, it is very possible that compile options might already exist to use asynchronous sockets.

3.2.3 Storage RDMA Protocol

Storage RDMA Protocol (SRP) is the industry standard storage protocol for InfiniBand. SRP provides a standard SCSI protocol transport over InfiniBand. The SCSI communications generated by either an application or the file system (often part of the operating system), are issued directly to the SRP interface, which provides full legacy support. The SCSI protocol and the data structures used in SCSI are transported without modification by SRP over InfiniBand, just as SCSI is transported over Fibre Channel (FCP protocol), and over TCP/IP (iSCSI protocol).

The areas in which FC, iSCSI, and SRP differ are related to the network aware functions, such as device discovery and enumeration, multi-path support, and path failover. There are two technical solutions for mapping the network aspects on top of SRP: emulation of Fibre Channel and its network and naming constructs; or emulation of iSCSI and its network and naming constructs.

The current InfiniBand storage implementations, specifically the Topspin InfiniBand-Fibre Channel bridges, implement Fibre Channel emulation. This provides the most seamless interoperability between existing Fibre Channel SANs and new InfiniBand-connected host systems. The host applications see a normal SCSI connection and through the Topspin drivers can treat the connection as they would a normal Fibre Channel connection. Storage vendor specific load balancing and failover drivers operate over SRP and mixed SRP-FCP SANs without any awareness of it communicating over different fabrics.

When the mixed SAN environment gets more complicated, and both hosts and storage are spread on both the InfiniBand and non-InfiniBand (Fibre Channel or iSCSI) fabrics, bridging them all together in a Fibre Channel emulation mode will become more difficult.

Voltaire has been leading an effort, with the support of IBM and others, to standardize around a more iSCSI-like view of the network. When iSCSI was developed, a very rich set of functions was architected to provide a well-defined and interoperable mechanism for discovery, naming, authorization, and most of the fabric-aware functions. By utilizing iSCSI and its services such as iSCSI names services (iSNS), SRP picks up a standard method of naming so that devices on the InfiniBand fabric can recognize devices on Fibre Channel or iSCSI fabrics through a common namespace that understands how to map names from fabric to fabric. Several such functions are enabled or made simpler through the use of an underlying iSCSI-like mechanism. This does not preclude emulating a Fibre Channel connection, while enabling a much more functional mode, where applications can have visibility to mixed SAN fabric environments without having to understand and account for the differences between those fabrics.

3.2.4 User-level Device Access Programming Layer

The Direct Access Programming Layer (DAPL) protocol for user-space applications (uDAPL), and for kernel mode use (kDAPL) are relatively new industry standards for high-efficiency, low-latency server-to-server communications. User-level Device Access Programming Layer (uDAPL) is one of the lowest-latency, highest-bandwidth, and highest-efficiency standard protocols available on InfiniBand. MPI is currently slightly better then uDAPL, but additional performance tuning can put uDAPL in first place. uDAPL is the preferred interface for new development in low-latency commercial applications. Database vendors (such as Oracle® with RAC 10i) will be using uDAPL for scale-out database clustering.

uDAPL has also been used as the underlying support for other protocol implementations. Specifically, Scali has delivered MPI support for InfiniBand by layering their MPI stack on top of uDAPL on InfiniBand. Scali is counting on uDAPL being a small, simple, common layer on top of which MPI can still provide significant performance and efficiency benefits.

kDAPL is a similar interface for use from kernel mode applications or from the operating system itself. Primary use is aimed at the file system interface. NFS/RDMA, iSER, and other storage interfaces could go directly through kDAPL.

IT API is an industry-standard initiative (also involving IBM) working to standardize a common API for commercial applications. At this time, uDAPL is providing this function. This effort might, at some point, come up with another interface. However, with the momentum behind uDAPL, it is likely that the *new* interface will be based on what we have now.

3.2.5 Message Passing Interface

Message Passing Interface (MPI) is the standard protocol for scientific and technical high-performance clustering (HPC). MPI can be used over many different cluster fabrics, including Ethernet, Myrinet, Quadrics, and InfiniBand. MPI provides the lowest latency, highest bandwidth, and highest efficiency of all of the standard protocols available on InfiniBand.

Most HPC Linux environments utilize MPI protocols. Much of this market is led by work done at national labs and research universities around the world. The MPI stack (MPICH) developed at Ohio State University is the most commonly recommended MPI implementation with the Topspin InfiniBand solution.

3.3 Hardware

This section describes the hardware used in an InfiniBand configuration:

- Host Channel Adapters

  A Host Channel Adapter (HCA) provides a processor node, such as a server or workstation, with a connection port to other InfiniBand devices. An HCA can have one or multiple ports and can be an add in card on a standard interconnect bus or onboard the system main board. The adapter can be connected to an InfiniBand switch, a target device or another HCA.

  Verbs are an abstract representation that defines the required interface between the client software and the functions of the HCA. Verbs do not specify the application programming interface (API) for the operating system, but define the operation for OS vendors to develop a usable API.

- Target Channel Adapters

  A Target Channel Adapter (TCA) provides I/O nodes, such as storage devices, with an InfiniBand connection port. It can be connected to an InfiniBand switch or an HCA.

- Switches

  An InfiniBand switch forwards packets from one device to another based on the Local ID contained within the Local Route Header of a packet (see 3.1.4, “Data link layer” on page 33). The only type of packets generated on a switch are management packets.

- Routers

  InfiniBand routers forward packets between subnets. The router reads the Global Route Header within a packet. (See 3.1.3, “Network layer” on page 33.)

- Gateways

  InfiniBand to Ethernet or InfiniBand to Fibre Channel gateways allow nodes on InfiniBand subnets to be connected to Ethernet networks or to Fibre Channel SANs. The use of a gateway merges other fabric networks while eliminating the need for additional HCAs in every processor node.

3.4 Management

The InfiniBand Architecture defines two methods of system management for handling all subnet bringup, maintenance, and general service functions associated with the devices in the subnet. Each method has a dedicated queue pair (QP) that is supported by all devices on the subnet to distinguish management traffic from all other traffic

3.4.1 Subnet management

The first method, *subnet management*, is handled by a subnet manager. All subnet management uses QP0 and is handled exclusively on a high-priority virtual lane (VL15) to ensure the highest priority within the subnet. Subnet Management Packets (SMPs) are the only packets allowed on QP0 and VL15. This VL uses the Unreliable Datagram transport service and does not follow the same flow control restriction as other VLs on the links. Subnet management information is passed through the subnet ahead of all other traffic on a link

- Subnet Manager (SM)

  The SM monitors the health and performance of the InfiniBand subnet, maintains topology information, and provides routing information to all of the switches in the network. These responsibilities include LID assignment, SL-to-VL mapping, link bringup and teardown, and link failover.

  There must be at least one SM for a subnet and the SM must reside within the fabric subnet, either on a switch or host node. Backup SMs can be defined in the network in the event that the primary SM should fail.

- Subnet Management Agents (SMAs)

  The nodes within the subnet will maintain SMAs that process data from the SM.

- Subnet Management Packets (SMPs)

  Managers and agents communicate through Management Datagrams (MADs). SMPs are used for communication between SMs and SMAs.

3.4.2 General services interface

The second method defined by InfiniBand is the *general services interface* (GSI). The GSI handles functions such as chassis management, out-of-band I/O operations, and other functions not associated with the subnet manager. These functions do not have the same high-priority needs as subnet management, so the GSI management packets (GMPs) do not use the high-priority virtual lane, VL15. All GSI commands use QP1 and must follow the flow control requirements of other data links.

- General Service Manager (GSM)

  The GSM provides a variety of management functions not maintained by the SM including performance, communication, and chassis management for physical devices.

- General Service Agents (GSAs)

  GSAs run on the nodes of the subnet and maintain management data and parameters managed by GSMs.

- General Service Management Packets (GMPs)

  GMPs are used for communication general service managers and general service agents.

3.5 Common markets

Important markets, such as application clustering, storage area networks, inter-tier communication and interprocessor communication, require high bandwidth, QoS, and RAS features. Also, many embedded systems (including routers, storage systems, and intelligent switches) utilize the PCI bus, often in the Compact PCI format, for their internal I/O architecture. Such systems are unable to keep up with high-speed networking interconnects such as Gigabit Ethernet and ATM. Therefore, many companies are developing proprietary I/O interconnect architectures.

Building on the experience of developing Ethernet local area networks (LAN), Fibre Channel storage area networks, and numerous wide area network (WAN) interconnects, InfiniBand has been networked to exceed the needs of today’s markets and provide a cohesive interconnect for a wide range of systems. This is accomplished with direct support for highly important items such as RAS, QoS, and scalability.

3.5.1 Application clustering

The Internet today has evolved into a global infrastructure supporting applications such as streaming media, business-to-business solutions, e-commerce, and interactive portal sites. Each of these applications must support an ever-increasing volume of data and demand for reliability. Service providers are in turn experiencing tremendous pressure to support these applications. They must route traffic efficiently through increasingly congested communication lines while offering the opportunity to charge for differing QoS and security levels. Application Service Providers (ASP) have arisen to support the outsourcing of e-commerce, e-marketing, and other e-business activities to companies specializing in Web-based applications.

These ASPs must be able to offer highly reliable services that offer the ability to dramatically scale in a short period of time to accommodate the explosive growth of the Internet. The cluster has evolved as the preferred mechanism to support these requirements. A cluster is simply a group of servers connected by load-balancing switches working in parallel to serve a particular application.

InfiniBand simplifies application cluster connections by unifying the network interconnect with a feature-rich managed architecture. InfiniBand's switched architecture provides native cluster connectivity, thus supporting scalability and reliability inside and out of the box. Devices can be added and multiple paths can be utilized with the addition of switches to the fabric. High-priority transactions between devices can be processed ahead of lower-priority items through QoS mechanisms built into InfiniBand.

3.5.2 Interprocessor communication

Interprocessor communication enables multiple servers to work together on a single application. A high-bandwidth, low-latency reliable connection is required between servers to ensure reliable processing. Scalability is critical as applications require more processor bandwidth. The switched nature of InfiniBand provides connection reliability for interprocessor communication systems by allowing multiple paths between systems. Scalability is supported with fully hot-swappable connections managed by a single unit (subnet manager). With multicast support, single transactions can be made to multiple destinations. This includes sending to all systems on the subnet, or to only a subset of these systems. The higher-bandwidth connections (4X, 12X) defined by InfiniBand provide backbone capabilities for interprocessor communication clusters without the need for a secondary I/O interconnect.

3.5.3 Storage area networks

Storage area networks are groups of complex storage systems connected through managed switches to allow very large amounts of data to be accessed from multiple servers. Today, storage area networks are built using Fibre Channel switches, hubs, and servers that are attached through Fibre Channel host bus adapters (HBA). Storage area networks are used to provide reliable connections to large databases of information that the Internet Data Center requires. A storage area network can restrict the data that individual servers can access, thereby providing an important partitioning mechanism (sometimes called zoning or fencing).

The fabric topology of InfiniBand enables communication between storage and server to be simplified. Removal of the Fibre Channel network enables servers to directly connect to a storage area network without a costly HBA. With features such as Remote DMA (RDMA) support, simultaneous peer-to-peer communication, and end-to-end flow control, InfiniBand overcomes the deficiencies of Fibre Channel without the need for an expensive, complex HBA.

4

Configuring InfiniBand with QLogic bridges

In this chapter, we discuss the steps to connect and configure the hardware for the following basics tasks:

1. Install the QLogic InfiniBand Bridge Module drivers for Windows and Linux.
2. Configure the Virtual NIC driver for the InfiniBand Ethernet Bridge Module.
3. Configure IPoIB.
4. Configure SRP for the InfiniBand Fibre Channel Bridge Module.

For these tasks, we used a basic configuration as shown in Figure 4-1.

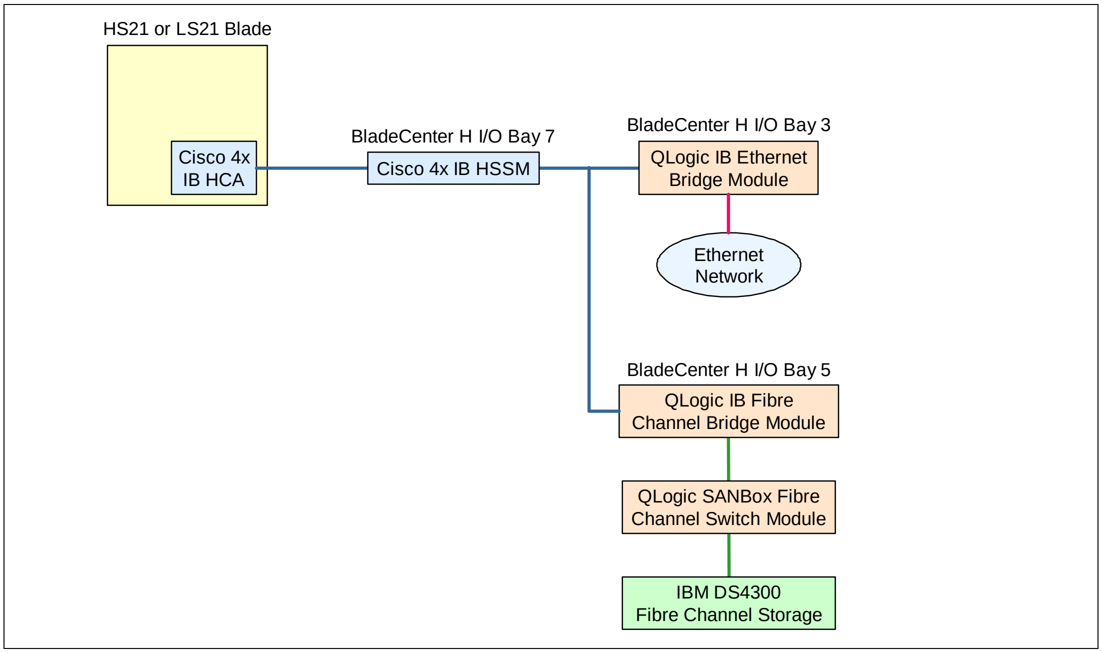

HS21 or LS21 Blade
Cisco 4x IB HCA
BladeCenter H I/O Bay 7
Cisco 4x IB HSSM
BladeCenter H I/O Bay 3
QLogic IB Ethernet Bridge Module
Ethernet Network
BladeCenter H I/O Bay 5
QLogic IB Fibre Channel Bridge Module
QLogic SANBox Fibre Channel Switch Module
IBM DS4300 Fibre Channel Storage

Figure 4-1 Basic configuration

4.1 Configuring QLogic InfiniBand Bridge Modules with Linux

This section shows how to install the QLogic bridge module drivers for Linux operating systems, how to configure the InfiniBand Ethernet Bridge Module, and how to configure the InfiniBand Fibre Channel Bridge Module.

The steps are:

- 4.1.1, “Installing the drivers”
- 4.1.2, “QLogic InfiniBand Ethernet Bridge Module: Configuring VNIC” on page 45
- 4.1.3, “QLogic InfiniServ Driver: Configuring IPoIB” on page 48
- 4.1.4, “QLogic Fibre Channel Bridge: Configuring SRP” on page 50
- 4.1.5, “Hints and tips” on page 60

Before you begin your configuration, make sure that the following components are up to date:

- Advanced Management Module firmware
- Cisco 4X HCA firmware
- Cisco 4X High Speed Switch Module firmware
- QLogic InfiniBand Bridge module firmware

4.1.1 Installing the drivers

Here, we install the Bridge Module drivers for Linux. Follow these steps:

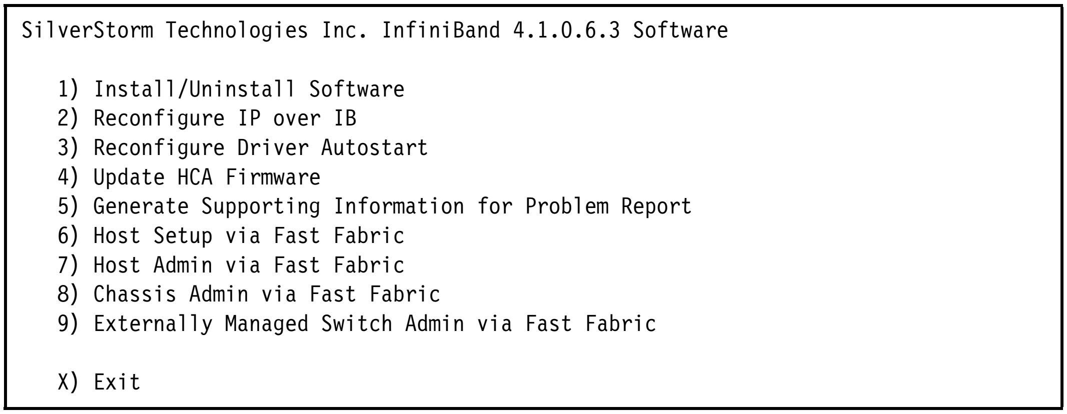
1. Download the latest QLogic InfiniServ drivers from the IBM support web site:

   http://www.ibm.com/support/docview.wss?uid=psg1MIGR-5069861

2. Untar the driver file. You then have a new folder called InfiniServBasic*x.x.x.x*, where *x.x.x.x* is the driver version.
3. Use the Linux **cd** command to enter the folder.
4. Run the installation script to install the QLogic drivers:

   ```
   ./INSTALL
   ```

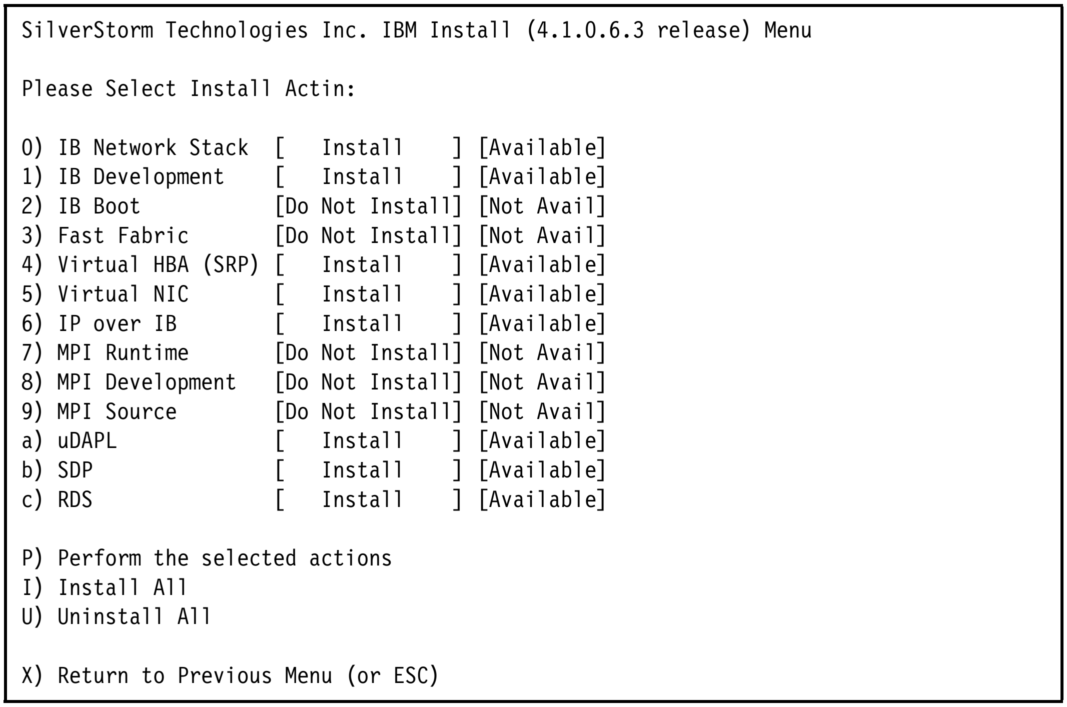
   The QLogic driver menu then displays as shown in Figure 4-2.



```
SilverStorm Technologies Inc. InfiniBand 4.1.0.6.3 Software

   1) Install/Uninstall Software
   2) Reconfigure IP over IB
   3) Reconfigure Driver Autostart
   4) Update HCA Firmware
   5) Generate Supporting Information for Problem Report
   6) Host Setup via Fast Fabric
   7) Host Admin via Fast Fabric
   8) Chassis Admin via Fast Fabric
   9) Externally Managed Switch Admin via Fast Fabric

   X) Exit
```

Figure 4-2 QLogic driver menu

5. Select 1 to install the necessary drivers. For the examples that we cover in this section, install the driver components as shown in Figure 4-3.



```
SilverStorm Technologies Inc. IBM Install (4.1.0.6.3 release) Menu

Please Select Install Actin:

0) IB Network Stack  [   Install    ] [Available]
1) IB Development    [   Install    ] [Available]
2) IB Boot           [Do Not Install] [Not Avail]
3) Fast Fabric       [Do Not Install] [Not Avail]
4) Virtual HBA (SRP) [   Install    ] [Available]
5) Virtual NIC       [   Install    ] [Available]
6) IP over IB        [   Install    ] [Available]
7) MPI Runtime       [Do Not Install] [Not Avail]
8) MPI Development   [Do Not Install] [Not Avail]
9) MPI Source        [Do Not Install] [Not Avail]
a) uDAPL             [   Install    ] [Available]
b) SDP               [   Install    ] [Available]
c) RDS               [   Install    ] [Available]

P) Perform the selected actions
I) Install All
U) Uninstall All

X) Return to Previous Menu (or ESC)
```

Figure 4-3   InfiniServ driver install menu

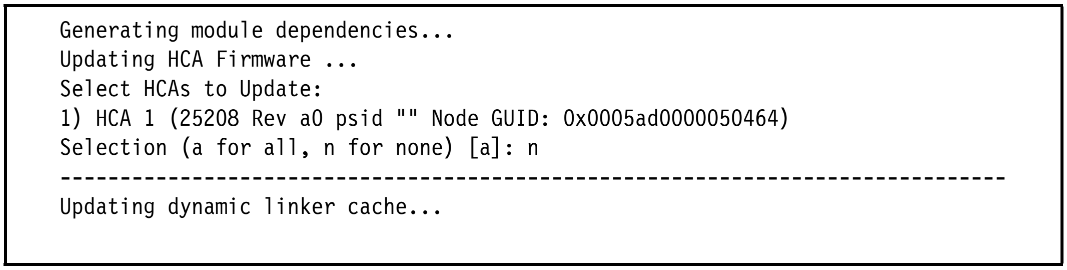
6. Follow the driver installation prompts and accept the defaults as shown in Figure 4-4 on page 44.

> **Note:** Do not put the SilverStorm firmware on the Cisco HCA. If you install the SilverStorm firmware, the HCA will no longer respond to the operating system.

```
Installing IB Network Stack...
Adding module dependencies...
Adding memory locking limits...
Copying ibt.ko...
Copying ics_dsc.ko...
Copying 82808XA.ko...
Copying mt23108vpd.ko...
Copying mt25218vpd.ko...
Creating IB Network Stack (iba) system startup files...
Creating IB Port Monitor (iba_mon) system startup files...
-------------------------------------------------------------------------------
Installing IB Development...
-------------------------------------------------------------------------------
Installing Virtual HBA (SRP) Driver...
Copying ics_srp.ko...
Creating Virtual HBA (SRP) (ics_srp) system startup files...
-------------------------------------------------------------------------------
Installing Virtual NIC Driver...
Copying ics_inic.ko...
Creating Virtual NIC (ics_inic) system startup files...
-------------------------------------------------------------------------------
Installing IP over IB Driver...
Copying ipoib.ko...
Assign IP over IB static IPV4 addresses now? [n]:n

IP over IB requires an ifcfg file for each IP over IB device instance.
Manually create files such as '//etc/sysconfig/network-scripts/ifcfg-ib1'

IP over IB also requires /etc/sysconfig/ipoib.cfg specify parameters
for each IP over IB device.
The default configuration file provides for a 2 port redundant configuration.
If you desire a different configuration for IP over IB, Manually edit the file.

Hit any key to continue...
Creating IP over IB (ipoib) system startup files...
-------------------------------------------------------------------------------
Installing uDAPL...
You have a uDAPL configuration file from an earlier install
Do you want to keep //etc/dat.conf? [y]:

Leaving //etc/dat.conf unchanged...
Copying udapl_module.ko...
Creating uDAPL (udapl) system startup files...
-------------------------------------------------------------------------------
Installing RDS Driver...
Copying ics_offload.ko...
Copying rds.ko...
Creating RDS (rds) system startup files...

Enable IB Network Stack (iba) to autostart? [y]:
Enable IB Port Monitor (iba_mon) to autostart? [y]:
Enable Virtual HBA (SRP) (ics_srp) to autostart? [y]:
Enable Virtual NIC (ics_inic) to autostart? [y]:
Enable IP over IB (ipoib) to autostart? [y]:
Enable uDAPL (udapl) to autostart? [y]:
Enable SDP (ics_sdp) to autostart? [y]:
Enable RDS (rds) to autostart? [y]:
Hit any key to continue...
```

Figure 4-4 Bridge Module Linux Driver installation prompts

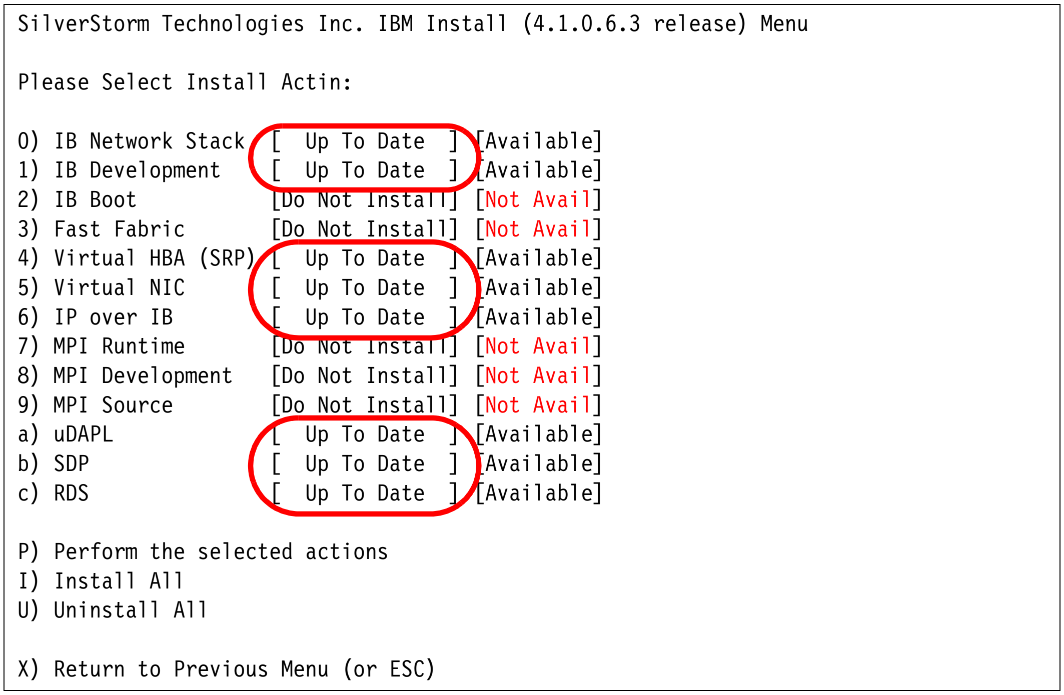
7. The installation continues. As shown in Figure 4-5, when prompted to install the SilverStorm firmware onto the HCA, select **n** to select no.



```
Generating module dependencies...
Updating HCA Firmware ...
Select HCAs to Update:
1) HCA 1 (25208 Rev a0 psid "" Node GUID: 0x0005ad0000050464)
Selection (a for all, n for none) [a]: n
--------------------------------------------------------------------------------
Updating dynamic linker cache...
```

Figure 4-5 Do not install SilverStorm firmware

8. Your drivers should now be up to date as shown in Figure 4-6.



```
SilverStorm Technologies Inc. IBM Install (4.1.0.6.3 release) Menu

Please Select Install Actin:

0) IB Network Stack  [  Up To Date  ] [Available]
1) IB Development    [  Up To Date  ] [Available]
2) IB Boot           [Do Not Install] [Not Avail]
3) Fast Fabric       [Do Not Install] [Not Avail]
4) Virtual HBA (SRP) [  Up To Date  ] [Available]
5) Virtual NIC       [  Up To Date  ] [Available]
6) IP over IB        [  Up To Date  ] [Available]
7) MPI Runtime       [Do Not Install] [Not Avail]
8) MPI Development   [Do Not Install] [Not Avail]
9) MPI Source        [Do Not Install] [Not Avail]
a) uDAPL             [  Up To Date  ] [Available]
b) SDP               [  Up To Date  ] [Available]
c) RDS               [  Up To Date  ] [Available]

P) Perform the selected actions
I) Install All
U) Uninstall All

X) Return to Previous Menu (or ESC)
```

Figure 4-6 Driver installation complete

4.1.2 QLogic InfiniBand Ethernet Bridge Module: Configuring VNIC

In this section, we describe how to configure a virtual network interface (VNIC) on a Linux blade using the QLogic InfiniServ drivers and the QLogic InfiniBand Ethernet Bridge. The VNIC driver allows you to create up to 84 virtual interfaces per bridge module.

To create virtual network interfaces on a Linux blade server, complete the following steps.

1. Get the IO Controller (IOC) numbers for the Ethernet bridge ports by running the command shown in Figure 4-7.

```
[root@localhost ~]# cat /proc/driver/ics_inic/iocs
0x66a01e1000147 BC2GE in Chassis 0x0000000000000000, Slot 5, Ioc 1
0x66a02e1000147 BC2GE in Chassis 0x0000000000000000, Slot 5, Ioc 2
0x66a03e1000147 BC2GE in Chassis 0x0000000000000000, Slot 5, Ioc 3
0x66a04e1000147 BC2GE in Chassis 0x0000000000000000, Slot 5, Ioc 4
0x66a05e1000147 BC2GE in Chassis 0x0000000000000000, Slot 5, Ioc 5
0x66a06e1000147 BC2GE in Chassis 0x0000000000000000, Slot 5, Ioc 6
[root@localhost ~]#
```

Figure 4-7 InfiniBand Ethernet Bridge IOCs

   Each IOC represents a physical port on the Ethernet bridge module.

2. Open the VNIC config file /etc/sysconfig/ics_inic.cfg in a text editor.
3. Uncomment the lines in the ics_inic.cfg file as shown in Figure 4-8 (at the bottom) and update the information based on your desired configuration.

```
{CREATE; NAME="eioc2";
   PRIMARY={IOCGUID=0x66A02E1000104; INSTANCE=0; PORT=1; }
   SECONDARY={IOCGUID=0x66A013000010C; INSTANCE=0; PORT=2;}
}
```

Figure 4-8 The ics_inic.cfg file

   This creates a virtual NIC on your system that communicates through the specified port on your Ethernet bridge module based on the port IOC that you designated. The parameters have the following meanings:

   – NAME: The name of the virtual interface.
   – PRIMARY: The primary settings for the virtual interface.
   – SECONDARY: The backup settings for the virtual interface.
   – IOCGUID: The GUID of the Ethernet port on the bridge that you want to use for this interface.
   – INSTANCE: The HCA adapter that you want to use. 0 represents the 1st HCA on the PCI bus.
   – PORT: The HCA port that you want to use. **1** represents the 1st port on the HCA. In a BladeCenter environment each port will only communicate with one switch module. For example, in a BCH chassis InfiniBand HCA port 1 will communicate with an InfiniBand switch module in bay 7.

4. Create an interface profile by creating a file ifcfg-eioc*X*, where *X* is the virtual interface number that you want to use. The easiest way to do create this file is to copy and modify and existing interface profile for eth0 or eth1. See Figure 4-9.

```
[root@localhost ~]# cd /etc/sysconfig/network-scripts/
[root@localhost network-scripts]# cp ifcfg-eth0 ifcfg-eioc1
```

Figure 4-9 Copy eth0 config file

For this setup, we used the settings (config file) shown in Figure 4-10.

```
DEVICE=eioc1
BOOTPROTO=static
IPADDR=192.168.199.4
NETMASK=255.255.255.0
ONBOOT=yes
TYPE=Ethernet
```

Figure 4-10 The eioc config file

5. Restart VNIC using the command **`ics_inic restart`** so that the new VNIC configuration settings (ics_inic.cfg) are applied, as shown in Figure 4-11.

```
[root@localhost ~]# /etc/init.d/ics_inic restart
Stopping Virtual NIC                                       [  OK  ]
Starting Virtual NIC                                       [  OK  ]
[root@localhost ~]#
```

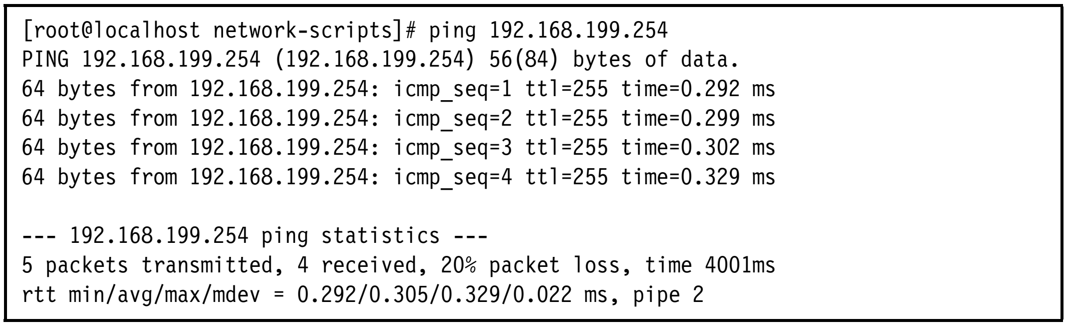
Figure 4-11 Restart the virtual NIC driver

6. Check the new IP settings with the **`ifconfig`** command as shown in Figure 4-12.

```
[root@localhost network-scripts]# ifconfig
eioc1     Link encap:Ethernet  HWaddr 00:06:6A:00:61:49
          inet addr:192.168.199.5  Bcast:192.168.199.255  Mask:255.255.255.0
          inet6 addr: fe80::206:6aff:fe00:6149/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:908 (908.0 b)

eth0      Link encap:Ethernet  HWaddr 00:14:5E:D6:17:02
          inet addr:9.42.166.99  Bcast:9.42.166.255  Mask:255.255.255.0
          inet6 addr: fe80::214:5eff:fed6:1702/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:42 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3648 (3.5 KiB)  TX bytes:3356 (3.2 KiB)
          Interrupt:209 Memory:da000000-da011100

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:24 errors:0 dropped:0 overruns:0 frame:0
          TX packets:24 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1916 (1.8 KiB)  TX bytes:1916 (1.8 KiB)

[root@localhost network-scripts]#
```

Figure 4-12 Network interfaces

7. Verify your connection works by pinging another device on the same subnet as shown in Figure 4-13.



```
[root@localhost network-scripts]# ping 192.168.199.254
PING 192.168.199.254 (192.168.199.254) 56(84) bytes of data.
64 bytes from 192.168.199.254: icmp_seq=1 ttl=255 time=0.292 ms
64 bytes from 192.168.199.254: icmp_seq=2 ttl=255 time=0.299 ms
64 bytes from 192.168.199.254: icmp_seq=3 ttl=255 time=0.302 ms
64 bytes from 192.168.199.254: icmp_seq=4 ttl=255 time=0.329 ms

--- 192.168.199.254 ping statistics ---
5 packets transmitted, 4 received, 20% packet loss, time 4001ms
rtt min/avg/max/mdev = 0.292/0.305/0.329/0.022 ms, pipe 2
```

Figure 4-13 Successful ping

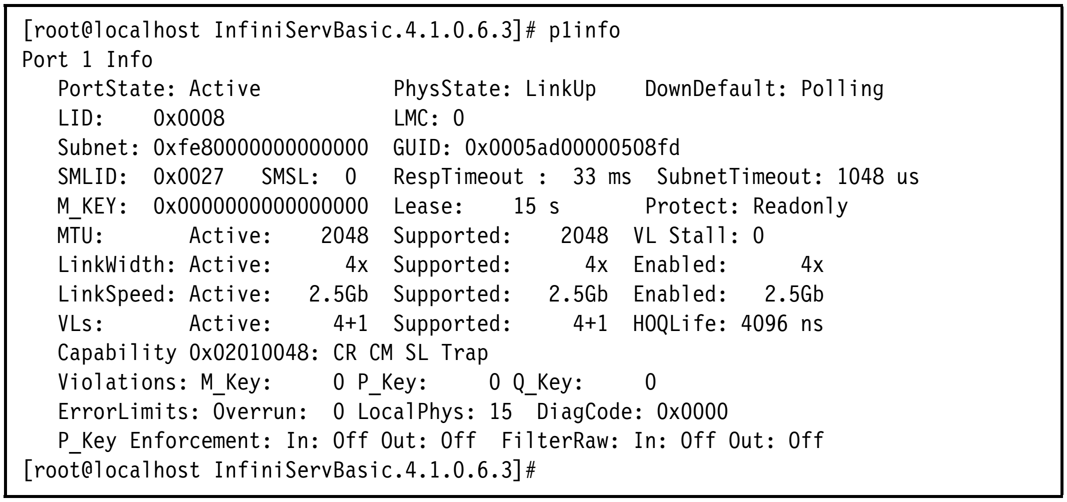
We have now created a virtual network interface on the Linux host that now will send Ethernet traffic over our InfiniBand network.

4.1.3 QLogic InfiniServ Driver: Configuring IPoIB

In this section, we describe how to configure a network interface on a Linux blade using the QLogic InfiniServ drivers that will send IP traffic of the InfiniBand interfaces of the HCA. Because this method uses the InfiniBand ports on the HCA, you are limited by the number of HCA ports on the expansion card (two in this case). Because the IP traffic is over the InfiniBand interfaces and not translated into true Ethernet traffic, the Ethernet bridge is not needed. However without the bridge module, you cannot ping a true Ethernet device—just another InfiniBand interface running IP over InfiniBand.

To create IPoIB interfaces on a Linux blade server, complete these steps:

1. Get the InfiniBand Globally Unique ID (GUID) of the HCA port that you want to use with the **p1info** command. This command shows you the unique identifier and status information for port 1 of the HCA. See Figure 4-14.



```
[root@localhost InfiniServBasic.4.1.0.6.3]# p1info
Port 1 Info
   PortState: Active          PhysState: LinkUp    DownDefault: Polling
   LID:    0x0008             LMC: 0
   Subnet: 0xfe80000000000000  GUID: 0x0005ad00000508fd
   SMLID:  0x0027   SMSL:  0   RespTimeout :  33 ms  SubnetTimeout: 1048 us
   M_KEY:  0x0000000000000000  Lease:    15 s       Protect: Readonly
   MTU:       Active:    2048  Supported:    2048  VL Stall: 0
   LinkWidth: Active:      4x  Supported:      4x  Enabled:      4x
   LinkSpeed: Active:   2.5Gb  Supported:   2.5Gb  Enabled:   2.5Gb
   VLs:       Active:     4+1  Supported:     4+1  HOQLife: 4096 ns
   Capability 0x02010048: CR CM SL Trap
   Violations: M_Key:     0 P_Key:     0 Q_Key:     0
   ErrorLimits: Overrun:  0 LocalPhys: 15  DiagCode: 0x0000
   P_Key Enforcement: In: Off Out: Off  FilterRaw: In: Off Out: Off
[root@localhost InfiniServBasic.4.1.0.6.3]#
```

Figure 4-14 InfiniBand HCA port 1 information

> **Tip:** If no information is returned on the port, verify that the system can see the HCA by running the **lspci** command. Also, check to verify that the HCA driver is loaded using the **lsmod** command. See 4.1.5, “Hints and tips” on page 60 for more details.

2. Open the IPoIB config file /etc/sysconfig/ipoib.cfg in a text editor.
3. Uncomment the lines in the ipoib.cfg file as shown in Figure 4-15 (at the bottom) and update the information based on your desired configuration.

```
{CREATE; NAME="ib1";
     PRIMARY={PORTGUID=0x0005ad00000508fd; }
}
```

Figure 4-15 The ipoib.cfg file

PORTGUID is the GUID of the HCA port that you want to use for this interface.

4. Create an interface profile by creating a file with the name ifcfg-ib*X* where *X* is the IPoIB interface number that you want to use. The easiest way to do create this file is to copy and modify and existing interface profile for eth0 or eth1.

   For this setup, we used the settings (config file) shown in Figure 4-16.

```
DEVICE=ib1
BOOTPROTO=static
IPADDR=192.168.199.4
NETMASK=255.255.255.0
ONBOOT=yes
TYPE=Ethernet
```

Figure 4-16 The ib1 config file

5. Restart IPoIB using the command **iba_ipoib restart** so that the new IPoIB configuration settings (ipoib.cfg) are applied. See Figure 4-17.

```
[root@localhost network-scripts]# iba_restart ipoib
Starting IP over IB                                        [  OK  ]
```

Figure 4-17 Restart IPoIB driver

6. Check the new IP settings with the **ifconfig** command as shown in Figure 4-18.

```
[root@localhost network-scripts]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:14:5E:D6:16:A8
          inet addr:9.42.166.100  Bcast:9.42.166.255  Mask:255.255.255.0
          inet6 addr: fe80::214:5eff:fed6:16a8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:603 errors:0 dropped:0 overruns:0 frame:0
          TX packets:25 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:44940 (43.8 KiB)  TX bytes:4274 (4.1 KiB)
          Interrupt:209 Memory:da000000-da011100

ib1       Link encap:Ethernet  HWaddr 02:00:00:00:00:01
          inet addr:192.168.199.4  Bcast:192.168.199.255  Mask:255.255.255.0
          inet6 addr: fe80::ff:fe00:1/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:2044  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:10 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          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:3460 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3460 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3775486 (3.6 MiB)  TX bytes:3775486 (3.6 MiB)

[root@localhost network-scripts]#
```

Figure 4-18 Network interfaces

7. Verify that your connection works by pinging another device on the same subnet.

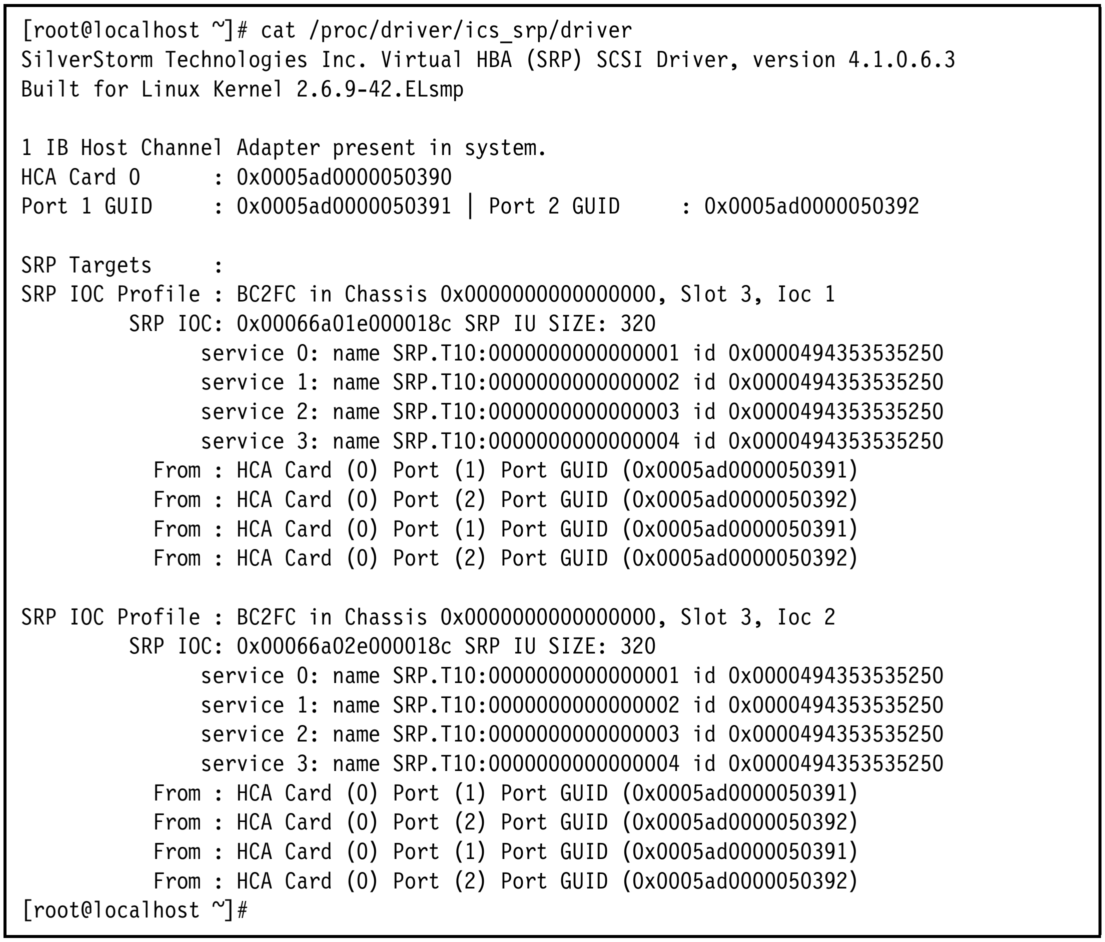
We have now configured the IPoIB protocol so that we can send IP traffic over our two InfiniBand HCA ports.

4.1.4 QLogic Fibre Channel Bridge: Configuring SRP

In this section, we describe how to configure a connection on a Linux blade to a remote Fibre Channel storage device using the QLogic InfiniServ drivers and the QLogic InfiniBand Fibre Channel Bridge Module. The InfiniBand HCA uses GUIDs and Fibre Channel storage devices use World Wide Names (WWNs). The mapping of GUID to WWN is done on the bridge module.

To create virtual SRP interfaces on a Linux blade server, complete these steps:

1. Each bridge module has two IO Controllers (IOC). Find the SRP IOCs of your bridge module by displaying the SRP driver information using the **cat** command as shown in Figure 4-19. The driver file is located in the folder /proc/driver/ics_srp.



```
[root@localhost ~]# cat /proc/driver/ics_srp/driver
SilverStorm Technologies Inc. Virtual HBA (SRP) SCSI Driver, version 4.1.0.6.3
Built for Linux Kernel 2.6.9-42.ELsmp

1 IB Host Channel Adapter present in system.
HCA Card 0      : 0x0005ad0000050390
Port 1 GUID     : 0x0005ad0000050391 | Port 2 GUID     : 0x0005ad0000050392

SRP Targets     :
SRP IOC Profile : BC2FC in Chassis 0x0000000000000000, Slot 3, Ioc 1
         SRP IOC: 0x00066a01e000018c SRP IU SIZE: 320
               service 0: name SRP.T10:0000000000000001 id 0x0000494353535250
               service 1: name SRP.T10:0000000000000002 id 0x0000494353535250
               service 2: name SRP.T10:0000000000000003 id 0x0000494353535250
               service 3: name SRP.T10:0000000000000004 id 0x0000494353535250
           From : HCA Card (0) Port (1) Port GUID (0x0005ad0000050391)
           From : HCA Card (0) Port (2) Port GUID (0x0005ad0000050392)
           From : HCA Card (0) Port (1) Port GUID (0x0005ad0000050391)
           From : HCA Card (0) Port (2) Port GUID (0x0005ad0000050392)

SRP IOC Profile : BC2FC in Chassis 0x0000000000000000, Slot 3, Ioc 2
         SRP IOC: 0x00066a02e000018c SRP IU SIZE: 320
               service 0: name SRP.T10:0000000000000001 id 0x0000494353535250
               service 1: name SRP.T10:0000000000000002 id 0x0000494353535250
               service 2: name SRP.T10:0000000000000003 id 0x0000494353535250
               service 3: name SRP.T10:0000000000000004 id 0x0000494353535250
           From : HCA Card (0) Port (1) Port GUID (0x0005ad0000050391)
           From : HCA Card (0) Port (2) Port GUID (0x0005ad0000050392)
           From : HCA Card (0) Port (1) Port GUID (0x0005ad0000050391)
           From : HCA Card (0) Port (2) Port GUID (0x0005ad0000050392)
[root@localhost ~]#
```

Figure 4-19 Displaying the SRP driver information

In Figure 4-19, the IOC numbers are `0x00066a01e000018c` and `0x00066a02e000018c`.

2. Open the SRP config file /etc/sysconfig/ics_srp.cfg in a text editor.

3. Uncomment the lines in the ics_srp.cfg file as shown in Figure 4-20 (at the bottom) and update the information based on your desired configuration.

```
session
 begin
   card: 0
   port: 1
   targetIOCGuid: 0x00066a01e000018c
   initiatorExtension: 1
 end

 adapter
 begin
   description: "ITSO HBA 1"
 end
```

Figure 4-20 The ics_srp.cfg file

This creates a virtual Fibre Channel HBA (SRP initiator) on your system that communicates through the specified SRP IO Controller on your Fibre Channel bridge module. The following components are used in your config file:

– card: The HCA adapter that you want to use. **0** represents the first HCA on the PCI bus.

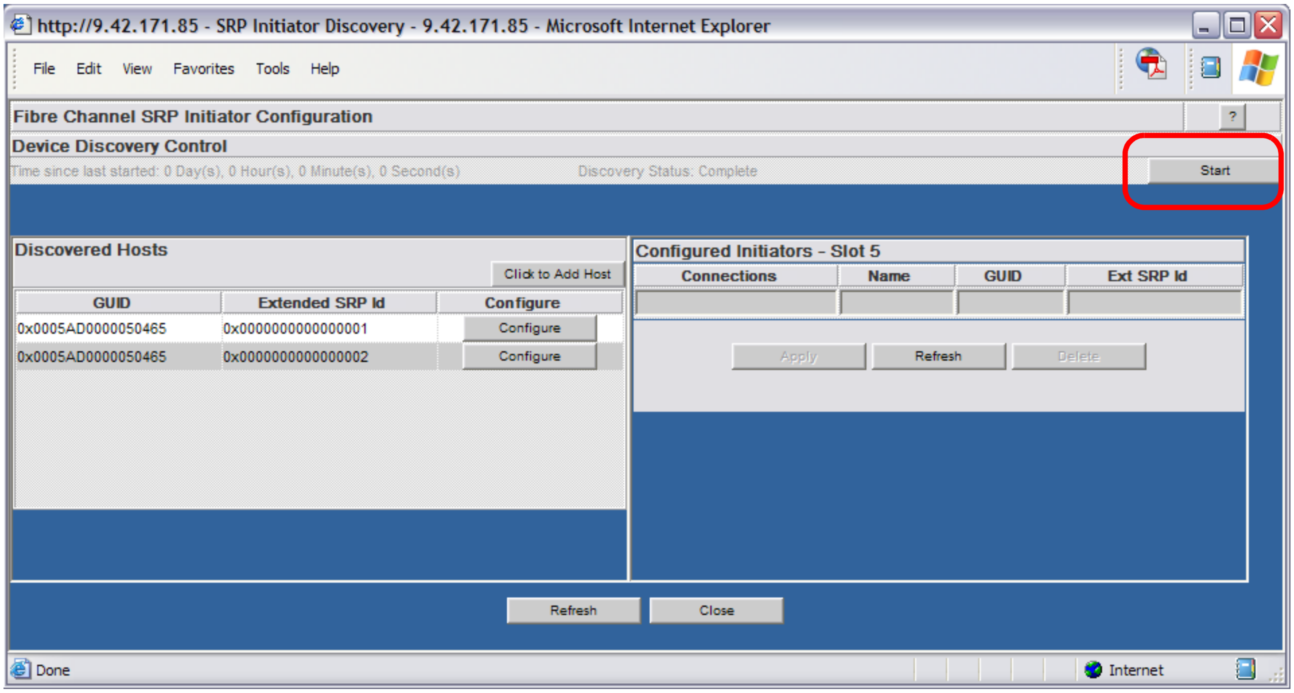
– port: The HCA port that you want to use. **1** represents the first port on the HCA. In a BladeCenter environment each port will only communicate with one switch module. For example, in a BCH chassis InfiniBand HCA port 1 communicates with an InfiniBand switch module in bay 7.

– targetIOCGUID: The SRP IOC of the IO Controller that you want to use

– initiatorExtension: A designation that will allow you to create multiple Virtual Fibre Channel HBAs on the same InfiniBand HCA port.

4. Restart SRP on your system using the command **ics_srp restart** so that the new virtual HBA configuration settings (ics_srp.cfg) are applied. This creates SRP initiators that you can discover from your bridge module.

5. Open the Web interface of your bridge module and click **SRP Initiator Discovery**, then click **Start** on the upper, right side of the page as shown in Figure 4-21. The SRP initiators that you created should be discovered and displayed under **Discovered Hosts**.



Fibre Channel SRP Initiator Configuration

Device Discovery Control

Time since last started: 0 Day(s), 0 Hour(s), 0 Minute(s), 0 Second(s)   Discovery Status: Complete

Discovered Hosts

GUID	Extended SRP Id	Configure
0x0005AD0000050465	0x0000000000000001	Configure
0x0005AD0000050465	0x0000000000000002	Configure

Configured Initiators - Slot 5

Connections	Name	GUID	Ext SRP Id

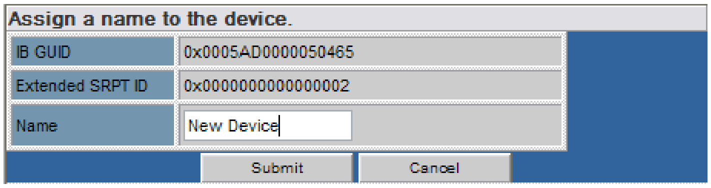
Figure 4-21 SRP Initiator Discovery screen

The following components are used to identify your virtual SRP initiator:

– GUID: InfiniBand HCA Port GUID that is being used for the virtual HBA.

– Extended SRP ID: The initiatorExtension that was set in the ics_srp.cfg file.

6. Click **Configure** to designate a name for the initiator (Figure 4-22). You will use this name later when mapping to a storage target.

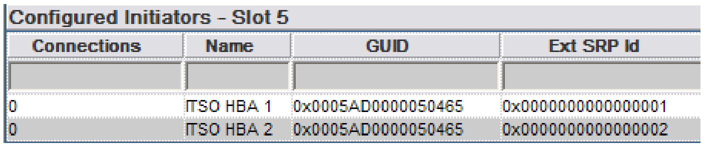


Assign a name to the device.

IB GUID	0x0005AD0000050465
Extended SRPT ID	0x0000000000000002
Name	New Device

Figure 4-22 Assign a name to the initiator.

7. After a name has been designated, click **Submit**. The initiator moves from Discovered Hosts to Configured Initiators (Figure 4-23).

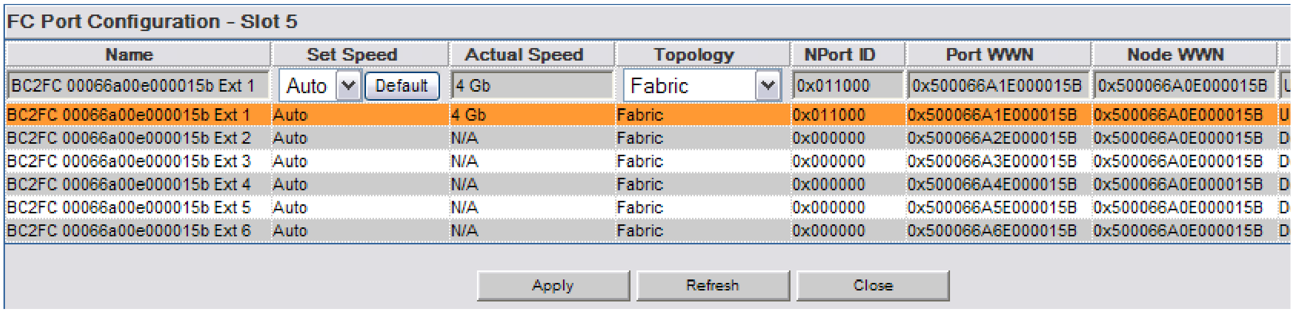


Configured Initiators - Slot 5

Connections	Name	GUID	Ext SRP Id
0	ITSO HBA 1	0x0005AD0000050465	0x0000000000000001
0	ITSO HBA 2	0x0005AD0000050465	0x0000000000000002

Figure 4-23 Configured Initiators

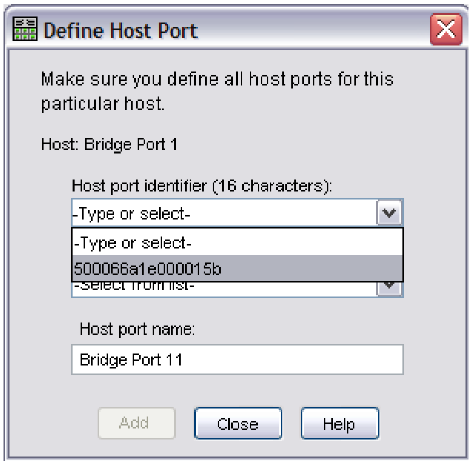
8. Each Fibre Channel port on the bridge module has a port world wide name (WWN). On the storage device, the bridge port WWN needs to be mapped to the drives. A Fibre Channel connection has to be established between the bridge port and the storage device for the storage device to see the bridge port WWN.



FC Port Configuration - Slot 5

Name	Set Speed	Actual Speed	Topology	NPort ID	Port WWN	Node WWN	
BC2FC 00066a00e000015b Ext 1	Auto Default	4 Gb	Fabric	0x011000	0x500066A1E000015B	0x500066A0E000015B	U
BC2FC 00066a00e000015b Ext 1	Auto	4 Gb	Fabric	0x011000	0x500066A1E000015B	0x500066A0E000015B	U
BC2FC 00066a00e000015b Ext 2	Auto	N/A	Fabric	0x000000	0x500066A2E000015B	0x500066A0E000015B	D
BC2FC 00066a00e000015b Ext 3	Auto	N/A	Fabric	0x000000	0x500066A3E000015B	0x500066A0E000015B	D
BC2FC 00066a00e000015b Ext 4	Auto	N/A	Fabric	0x000000	0x500066A4E000015B	0x500066A0E000015B	D
BC2FC 00066a00e000015b Ext 5	Auto	N/A	Fabric	0x000000	0x500066A5E000015B	0x500066A0E000015B	D
BC2FC 00066a00e000015b Ext 6	Auto	N/A	Fabric	0x000000	0x500066A6E000015B	0x500066A0E000015B	D

Figure 4-24 Bridge Port information



Define Host Port

Make sure you define all host ports for this particular host.

Host: Bridge Port 1

Host port identifier (16 characters):

-Type or select-

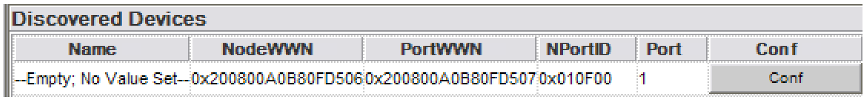
-Type or select-
500066a1e000015b
-Select from list-

Host port name:

Bridge Port 11

Figure 4-25 Bridge Port WWN on IBM FAStT DS4300

9. When the bridge port to remote drive mapping has been established, another mapping from SRP initiator to remote drive needs to be established. On the bridge Web GUI main menu, click **FCP Device Discovery**, then **Start**. The remote drives that are mapped to the bridge ports display in **Discovered Devices**, as shown in Figure 4-26.

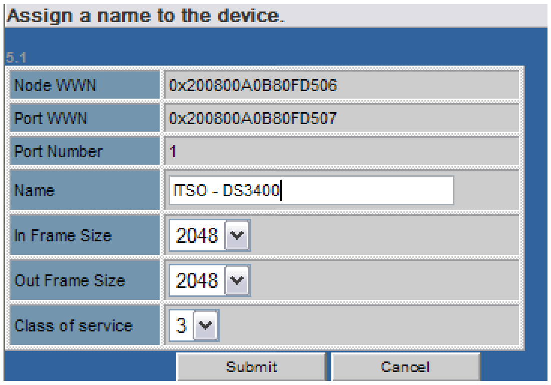


Discovered Devices

Name	NodeWWN	PortWWN	NPortID	Port	Conf
--Empty; No Value Set--	0x200800A0B80FD506	0x200800A0B80FD507	0x010F00	1	Conf

Figure 4-26 Discovered Fibre Channel Storage

10. Click **Configure** to designate a name for the device as shown in Figure 4-27. This name will be used later when mapping to an SRP Initiator.



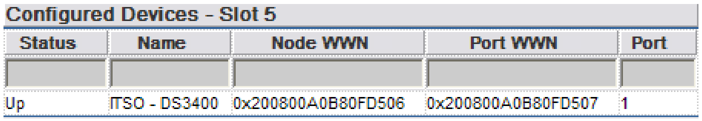
Assign a name to the device.

5.1

Node WWN	0x200800A0B80FD506
Port WWN	0x200800A0B80FD507
Port Number	1
Name	ITSO - DS3400
In Frame Size	2048
Out Frame Size	2048
Class of service	3

Figure 4-27 Assign a name to the storage device

11. After you designate a name, click **Submit**. The remote drive moves from Discovered Devices to Configured Devices (Figure 4-28).



Configured Devices - Slot 5

Status	Name	Node WWN	Port WWN	Port
Up	ITSO - DS3400	0x200800A0B80FD506	0x200800A0B80FD507	1

Figure 4-28 Configured Storage Device

12. Close the FCP Device Discovery window and click **SRP map config** on the GUI main menu page. Here, you see the virtual HBAs listed:

    – If your virtual HBA was created on IOC 1, click the **Click To Add** link under the IOC 1 heading.

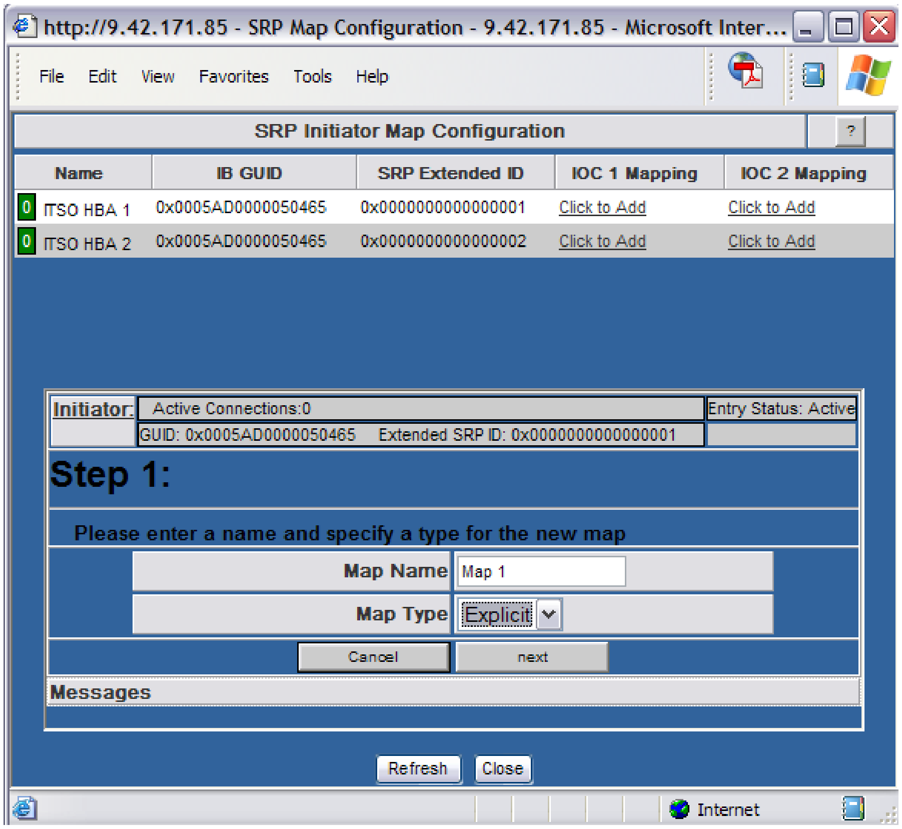
    – If your virtual HBA was created on IOC 2, click the **Click To Add** link under the IOC 2 heading.

13. Give the map a name and select a map type, then click **Next** as shown in Figure 4-29 on page 56. There are two mapping types:

    – *Explicit*: In an explicit mapping, the host and target LUN numbers must be listed in the mapping.

    – *Direct*: In a direct mapping, the host sees all remote drives that are visible to the port WWNs without a specific LUN being identified.

    You address explicit mapping first, and then direct mapping.


http://9.42.171.85 - SRP Map Configuration - 9.42.171.85 - Microsoft Inter...
File Edit View Favorites Tools Help
?
Refresh
Close
Internet

SRP Initiator Map Configuration

	Name	IB GUID	SRP Extended ID	IOC 1 Mapping	IOC 2 Mapping
0	ITSO HBA 1	0x0005AD0000050465	0x0000000000000001	Click to Add	Click to Add
0	ITSO HBA 2	0x0005AD0000050465	0x0000000000000002	Click to Add	Click to Add

Initiator: Active Connections:0 Entry Status: Active

GUID: 0x0005AD0000050465 Extended SRP ID: 0x0000000000000001

Step 1:

Please enter a name and specify a type for the new map

Map Name: Map 1

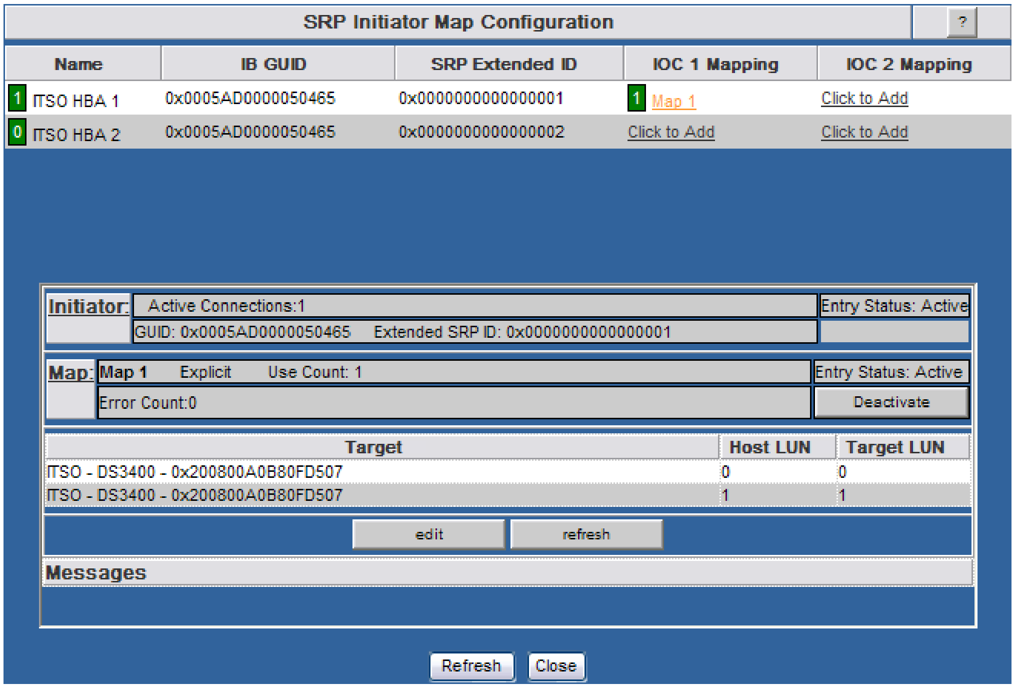
Map Type: Explicit

Cancel next

Messages

Figure 4-29 Add explicit map

14. Select a storage target and identify the host LUN and target LUN. Then click **Add Row**. The host LUN is the LUN number that displays on the Linux host. The target LUN is LUN number that was determined for the logical drive on the storage device. When you are done adding all logical drives that you want mapped to this virtual HBA, click **Finish**.



SRP Initiator Map Configuration

Name	IB GUID	SRP Extended ID	IOC 1 Mapping	IOC 2 Mapping
1 ITSO HBA 1	0x0005AD0000050465	0x0000000000000001	1 Map 1	Click to Add
0 ITSO HBA 2	0x0005AD0000050465	0x0000000000000002	Click to Add	Click to Add

Initiator: Active Connections:1 — Entry Status: Active
GUID: 0x0005AD0000050465 Extended SRP ID: 0x0000000000000001

Map: Map 1 Explicit Use Count: 1 — Entry Status: Active
Error Count:0

Target	Host LUN	Target LUN
ITSO - DS3400 - 0x200800A0B80FD507	0	0
ITSO - DS3400 - 0x200800A0B80FD507	1	1

Figure 4-30 Explicit map configuration

15. Restart SRP on the host so that it picks up the new mapping. Check for the remote drives using the command **`cat /proc/scsi/scsi`**. Figure 4-31 shows sample output.

```
Attached devices:
Host: scsi0 Channel: 00 Id: 00 Lun: 00
  Vendor: IBM-ESXS Model: ST973401SS       Rev: B51C
  Type:   Direct-Access                    ANSI SCSI revision: 05
Host: scsi3 Channel: 00 Id: 00 Lun: 00
  Vendor: IBM      Model: 1722-600         Rev: 0520
  Type:   Direct-Access                    ANSI SCSI revision: 03
Host: scsi3 Channel: 00 Id: 00 Lun: 01
  Vendor: IBM      Model: 1722-600         Rev: 0520
  Type:   Direct-Access                    ANSI SCSI revision: 03
```

Figure 4-31 Remote drives on Linux server

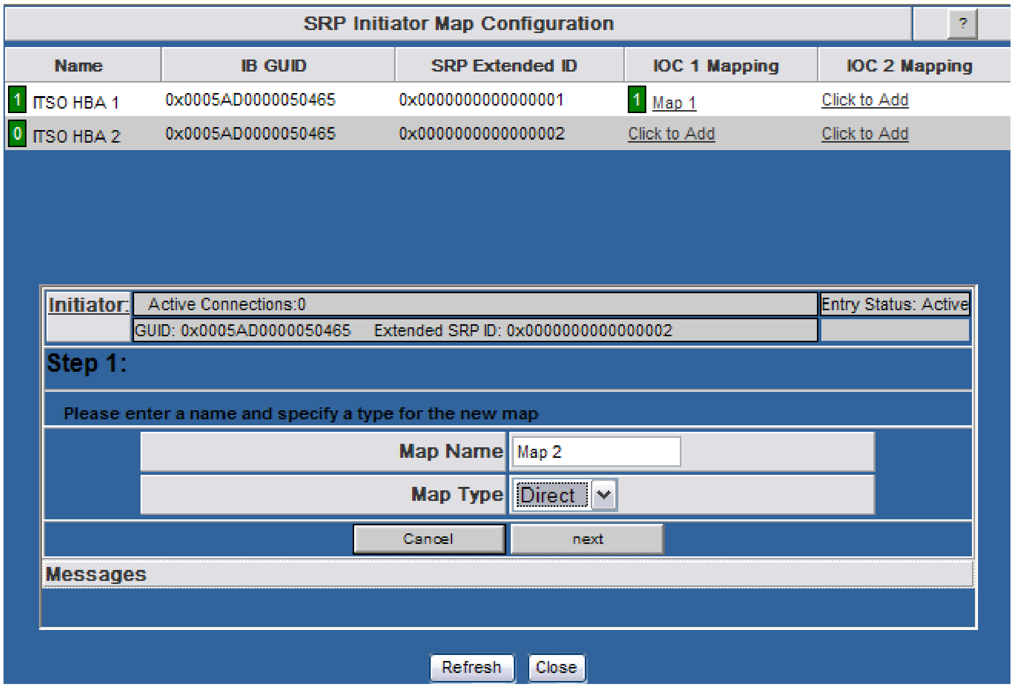
16. The boxes beside the SRP initiator name (shown in Figure 4-32 on page 58) and the IOC Map name indicate the active connections to the virtual HBA. It should go from 0 to 1. You might need to click **Refresh** on the SRP map page.

17. Now, you create a direct mapping. Click **SRP map config** on the GUI main menu page. Here, you see the virtual HBAs listed:

    – If your virtual HBA was created on IOC 1, click the **Click To Add** link under the IOC 1 heading.

    – If your virtual HBA was created on IOC 2, click the **Click To Add** link under the IOC 2 heading.

18. Give the map a name and select **Direct** type, then click **Next**.



SRP Initiator Map Configuration

Name	IB GUID	SRP Extended ID	IOC 1 Mapping	IOC 2 Mapping
1 ITSO HBA 1	0x0005AD0000050465	0x0000000000000001	1 Map 1	Click to Add
0 ITSO HBA 2	0x0005AD0000050465	0x0000000000000002	Click to Add	Click to Add

Initiator: Active Connections:0 Entry Status: Active
GUID: 0x0005AD0000050465 Extended SRP ID: 0x0000000000000002

Step 1:

Please enter a name and specify a type for the new map

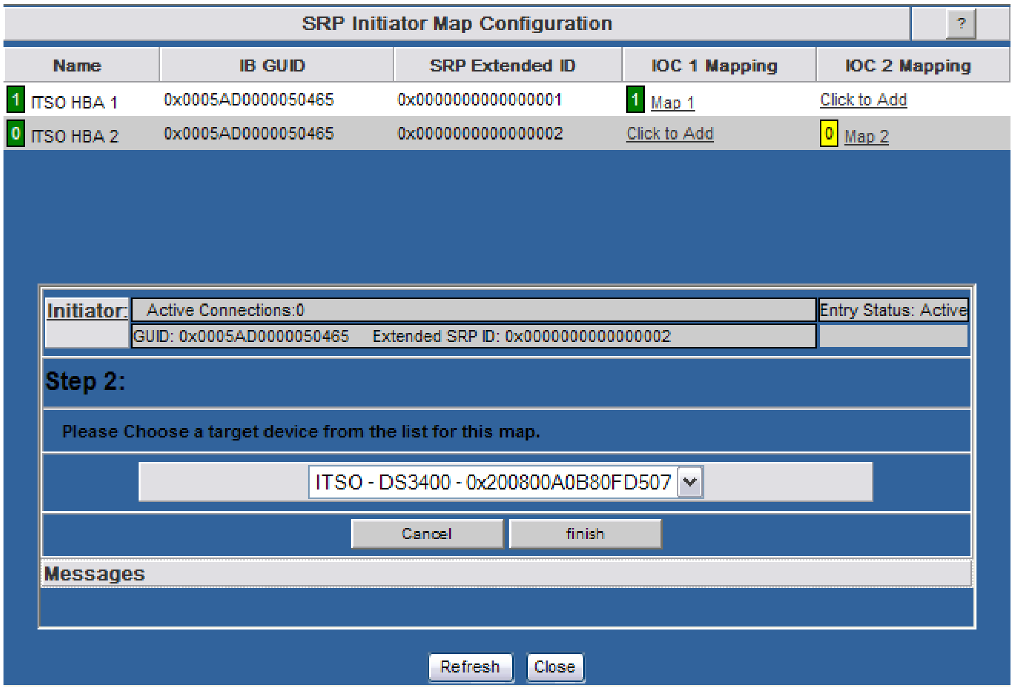
Map Name: Map 2

Map Type: Direct

Messages

Figure 4-32 Choose Direct Map

19. Select a storage target and click **Finish** (Figure 4-33).



SRP Initiator Map Configuration

Name	IB GUID	SRP Extended ID	IOC 1 Mapping	IOC 2 Mapping
1 ITSO HBA 1	0x0005AD0000050465	0x0000000000000001	1 Map 1	Click to Add
0 ITSO HBA 2	0x0005AD0000050465	0x0000000000000002	Click to Add	0 Map 2

Initiator: Active Connections:0 Entry Status: Active
GUID: 0x0005AD0000050465 Extended SRP ID: 0x0000000000000002

Step 2:

Please Choose a target device from the list for this map.

ITSO - DS3400 - 0x200800A0B80FD507

Messages

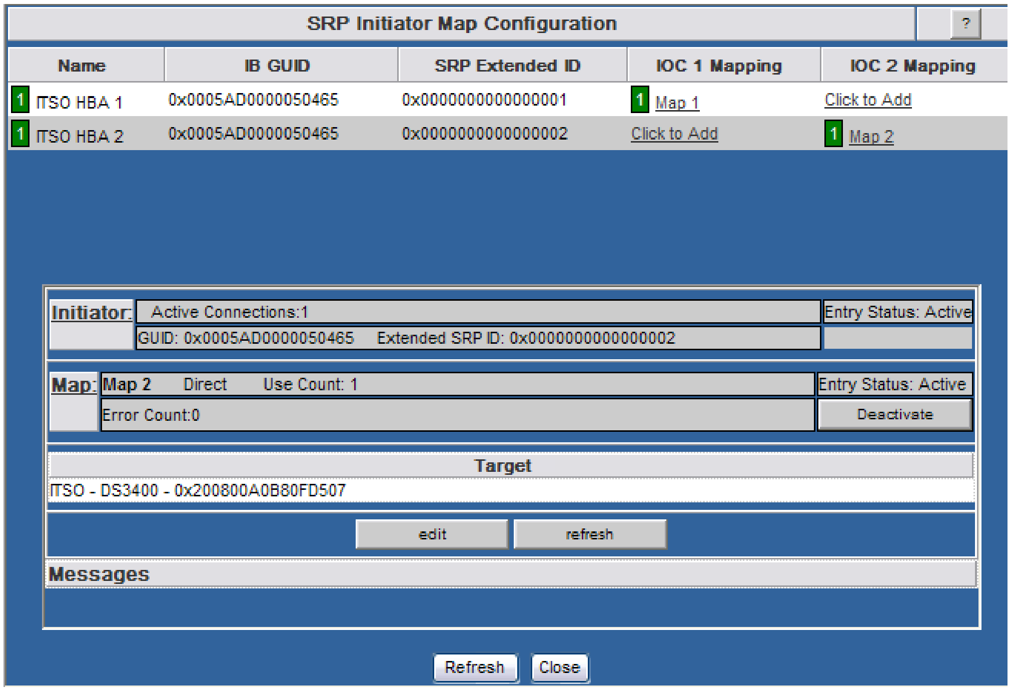
Figure 4-33 Choose storage device

20. Restart SRP on the host so that it picks up the new mapping. Check for the remote drives using the command **cat /proc/scsi/scsi**. Sample output is shown in Figure 4-34.

```
Attached devices:
Host: scsi0 Channel: 00 Id: 00 Lun: 00
  Vendor: IBM-ESXS Model: ST973401SS       Rev: B51C
  Type:   Direct-Access                    ANSI SCSI revision: 05
Host: scsi4 Channel: 00 Id: 00 Lun: 00
  Vendor: IBM      Model: 1722-600         Rev: 0520
  Type:   Direct-Access                    ANSI SCSI revision: 03
Host: scsi4 Channel: 00 Id: 00 Lun: 01
  Vendor: IBM      Model: 1722-600         Rev: 0520
  Type:   Direct-Access                    ANSI SCSI revision: 03
Host: scsi5 Channel: 00 Id: 00 Lun: 00
  Vendor: IBM      Model: 1722-600         Rev: 0520
  Type:   Direct-Access                    ANSI SCSI revision: 03
Host: scsi5 Channel: 00 Id: 00 Lun: 01
  Vendor: IBM      Model: 1722-600         Rev: 0520
  Type:   Direct-Access                    ANSI SCSI revision: 03
```

Figure 4-34 Remote drives on Linux server

21. On the bridge module web GUI, The green boxes beside the SRP initiator name and the IOC Map should go from 0 to 1. You might need to click **Refresh** on the SRP map page.



SRP Initiator Map Configuration

Name	IB GUID	SRP Extended ID	IOC 1 Mapping	IOC 2 Mapping
1 ITSO HBA 1	0x0005AD0000050465	0x0000000000000001	1 Map 1	Click to Add
1 ITSO HBA 2	0x0005AD0000050465	0x0000000000000002	Click to Add	1 Map 2

Initiator: Active Connections:1 Entry Status: Active
GUID: 0x0005AD0000050465 Extended SRP ID: 0x0000000000000002

Map: Map 2 Direct Use Count: 1 Entry Status: Active
Error Count:0

Target
ITSO - DS3400 - 0x200800A0B80FD507

Messages

Figure 4-35 Active direct mapping

We have now created an SRP initiator on the host which is mapped through the bridge module to the storage device.

4.1.5 Hints and tips

The following information might help you resolve connectivity issues with your Linux installation:

- Verify that the HCA can be seen by the operating system.

  If the HCA is not detected, use the linux command **lspci** as shown in Figure 4-36 to ensure that the operating system can see the card. If the card is not detected, shut down the blade and ensure that it is properly seated in the blade.

```
0c:00.0 InfiniBand: Mellanox Technologies MT25208 InfiniHost III Ex (Tavor
compatibility mode) (rev 20)
```

Figure 4-36 Portion of the output of the lspci command showing the InfiniBand HCA

- Verify that the HCA ports are active.

  Use the command **p1info** or **p2info** as shown in Figure 4-37 to check the status of the HCA ports to ensure that they are active.

```
[root@localhost InfiniServBasic.4.1.0.6.3]# p1info
Port 1 Info
   PortState: Active          PhysState: LinkUp    DownDefault: Polling
   LID:    0x0008             LMC: 0
   Subnet: 0xfe80000000000000  GUID: 0x0005ad00000508fd
   SMLID:  0x0027   SMSL:  0   RespTimeout :  33 ms  SubnetTimeout: 1048 us
   M_KEY:  0x0000000000000000  Lease:    15 s       Protect: Readonly
   MTU:       Active:    2048  Supported:    2048  VL Stall: 0
   LinkWidth: Active:      4x  Supported:      4x  Enabled:      4x
   LinkSpeed: Active:   2.5Gb  Supported:   2.5Gb  Enabled:   2.5Gb
   VLs:       Active:     4+1  Supported:     4+1  HOQLife: 4096 ns
   Capability 0x02010048: CR CM SL Trap
   Violations: M_Key:     0 P_Key:     0 Q_Key:     0
   ErrorLimits: Overrun:  0 LocalPhys: 15  DiagCode: 0x0000
   P_Key Enforcement: In: Off Out: Off  FilterRaw: In: Off Out: Off
[root@localhost InfiniServBasic.4.1.0.6.3]#
```

Figure 4-37 HCA port 1 information

- Ensure that the driver is loaded.

  Run the **lsmod** command to ensure that the driver is loaded as shown in Figure 4-38.

```
ipoib                 141792  1 ics_sdp
ics_srp                86356  0
ics_inic              123492  0
ics_dsc                84256  3 ipoib,ics_srp,ics_inic
mt23108vpd            213296  0
```

Figure 4-38 Portion of lsmod output showing Linux driver modules

- Restart network services.

  If the virtual NIC does not come up, restart the linux network service using the command **service network restart** as shown in Figure 4-39 so that the eioc interface profile (ifcfg-eiocX) is re-initiated.

```
[root@localhost network-scripts]# service network restart
Shutting down interface eth0 :                             [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Setting network parameters:                                [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface eioc1:                               [  OK  ]
Bringing up interface eth0:                                [  OK  ]
[root@localhost network-scripts]#
```

Figure 4-39 Restart network service

- Re-initialize IPoIB driver.

  If the IPoIB interface does not come up, restart the virtual NIC driver using the command **iba_restart ipoib** so that the IPoIB configuration file (/etc/sysconfig/ipoib.cfg) is re-initiated.

```
[root@localhost network-scripts]# iba_restart ipoib
Stopping IP over IB                                        [  OK  ]
Starting IP over IB                                        [  OK  ]
[root@localhost network-scripts]#
```

Figure 4-40 Restart IPoIB Driver

- Re-initialize the VNIC driver.

  If the virtual NIC interface does not come up, restart the virtual NIC driver using the command **/etc/init.d/ics_inic restart** so that the VNIC configuration file (/etc/sysconfig/ics_inic.cfg) is re-initiated.

```
[root@localhost ~]# /etc/init.d/ics_inic restart
Stopping Virtual NIC                                       [  OK  ]
Starting Virtual NIC                                       [  OK  ]
[root@localhost ~]#
```

Figure 4-41 Restart Virtual NIC Driver

- Re-initialize the SRP driver.

  If the SRP interface does not come up, restart the SRP driver using the command **/etc/init.d/ics_srp restart** as shown in Figure 4-42 so that the SRP configuration file (/etc/sysconfig/ics_inic.cfg) is re-initiated.

```
[root@localhost ~]# /etc/init.d/ics_srp restart
Stopping Virtual HBA (SRP):                                [  OK  ]
Starting Virtual HBA (SRP):                                [  OK  ]
[root@localhost ~]#
```

Figure 4-42 Restart SRP Driver

- Check the InfiniBand switch Web interface to verify that you have an active connection to the bridge module.

4.2 Configuring QLogic InfiniBand Bridge Modules with Windows

This section shows how to install the QLogic bridge module drivers for Windows operating systems, how to configure the InfiniBand Ethernet Bridge Module, and how to configure the InfiniBand Fibre Channel Bridge Module.

The steps are:

- 4.2.1, “Installing the HCA Drivers” on page 62
- 4.2.2, “QLogic InfiniBand Ethernet Bridge Module: Configuring VNIC” on page 63
- 4.2.3, “QLogic InfiniServ Driver: Configuring IPoIB” on page 65
- 4.2.4, “QLogic Fibre Channel Bridge: Configuring SRP” on page 67

Before you begin your configuration, make sure that the following components are up to date:

- Advanced Management Module firmware
- Cisco 4X HCA firmware
- Cisco 4X High Speed Switch Module firmware
- QLogic InfiniBand Bridge module firmware

4.2.1 Installing the HCA Drivers

Here we will install the Bridge Module drivers for Windows. Follow these steps:

1. Download the latest QLogic InfiniServ drivers from the IBM support Web site:

   `http://www.ibm.com/support/docview.wss?uid=psg1MIGR-5069862`

2. On your blade server, Windows will detect your new hardware automatically. Move this window aside for now.
3. Decompress the driver file. You will have an MSI file called SilverStormHCA.msi. Double-click the MSI file to start driver extraction. The driver files defaults to the following location:

   C:\Program Files\SilverStorm\SilverStorm HCA

4. Open the plug-and-play window again to install HCA driver.
5. Select **Install from a specific location**.
6. Click **Browse** under **Include this location in the search**. Select the following location and then click **OK**.

   C:\Program Files\SilverStorm\SilverStorm HCA\HCA

7. Click **Next** to continue the driver installation. The Wizard finds and installs InfiniHost Mellanox InfiniBand HCA for PCI Express. When the driver instillation is complete, you should see an entry for the HCA in device manager under InfiniBand Host Channel Adapters as shown in Figure 4-43.

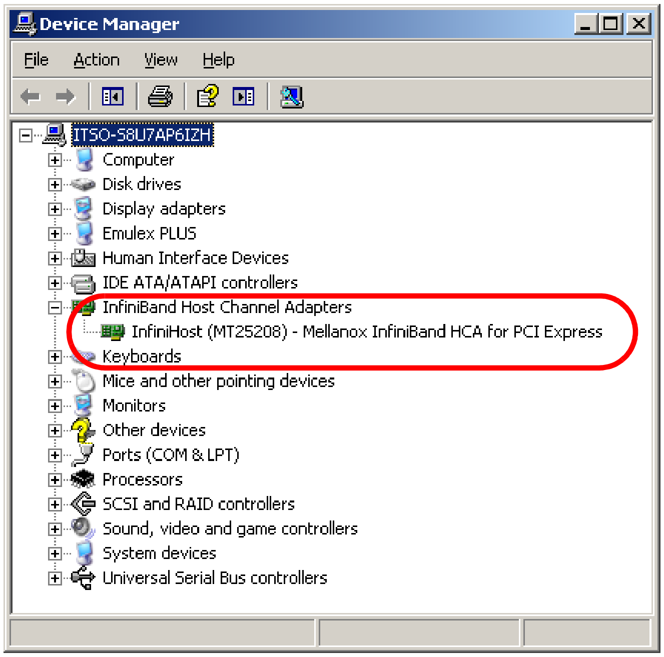

Device Manager
File
Action
View
Help
ITSO-S8U7AP6IZH
Computer
Disk drives
Display adapters
Emulex PLUS
Human Interface Devices
IDE ATA/ATAPI controllers
InfiniBand Host Channel Adapters
InfiniHost (MT25208) - Mellanox InfiniBand HCA for PCI Express
Keyboards
Mice and other pointing devices
Monitors
Other devices
Ports (COM & LPT)
Processors
SCSI and RAID controllers
Sound, video and game controllers
System devices
Universal Serial Bus controllers

Figure 4-43   InfiniBand HCA in Windows Device Manager

4.2.2 QLogic InfiniBand Ethernet Bridge Module: Configuring VNIC

In this section we describe how to configure a virtual network interface (VNIC) on a Windows blade server using the QLogic InfiniServ drivers and the QLogic InfiniBand Ethernet Bridge. The VNIC driver allows you to create six virtual interfaces (one per port) per bridge module.

1. Windows finds the Open IPoIB Adapter.

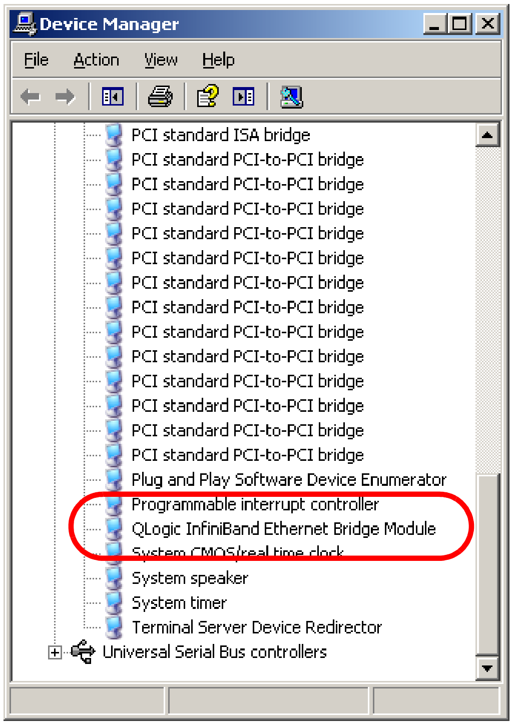
   We cover IPoIB in 4.2.3, “QLogic InfiniServ Driver: Configuring IPoIB” on page 65. If you would like to install the IPoIB drivers now, you can skip ahead to that section and return to the next step 2 when you are done. To skip IPoIB and to continue installing the VNIC drivers, click **Cancel** now to exit and continue to step 2.

2. Windows finds the QLogic InfiniBand Ethernet Bridge Module. Select **Install from a specific location** and click **Next**.

3. Click **Browse** under **Include this location in the search**. Select the following location and then click **OK**.

   C:\Program Files\SilverStorm\SilverStorm HCA\net

4. Click **Next** to continue driver installation. When the driver instillation is complete, you see the bridge module under System Devices in device manager as shown in Figure 4-44.



PCI standard ISA bridge
PCI standard PCI-to-PCI bridge
PCI standard PCI-to-PCI bridge
PCI standard PCI-to-PCI bridge
PCI standard PCI-to-PCI bridge
PCI standard PCI-to-PCI bridge
PCI standard PCI-to-PCI bridge
PCI standard PCI-to-PCI bridge
PCI standard PCI-to-PCI bridge
PCI standard PCI-to-PCI bridge
PCI standard PCI-to-PCI bridge
PCI standard PCI-to-PCI bridge
PCI standard PCI-to-PCI bridge
PCI standard PCI-to-PCI bridge
Plug and Play Software Device Enumerator
Programmable interrupt controller
QLogic InfiniBand Ethernet Bridge Module
System CMOS/real time clock
System speaker
System timer
Terminal Server Device Redirector
Universal Serial Bus controllers

Figure 4-44 Bridge in System Devices

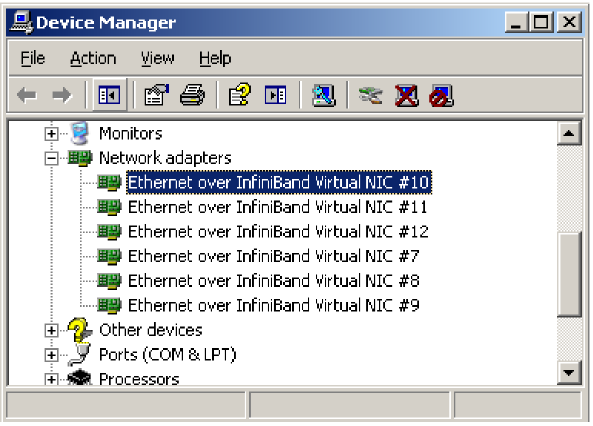
5. Windows now finds the SilverStorm Technologies VEx I/O Controller. Select **Install from a specific location** and click **Next**.

6. Click **Browse** under **Include this location in the search**. Select the following location and click **OK**.

   C:\Program Files\SilverStorm\SilverStorm HCA\net

7. Click **Next** to continue driver installation. The wizard finds and installs the Ethernet over InfiniBand Virtual NIC.

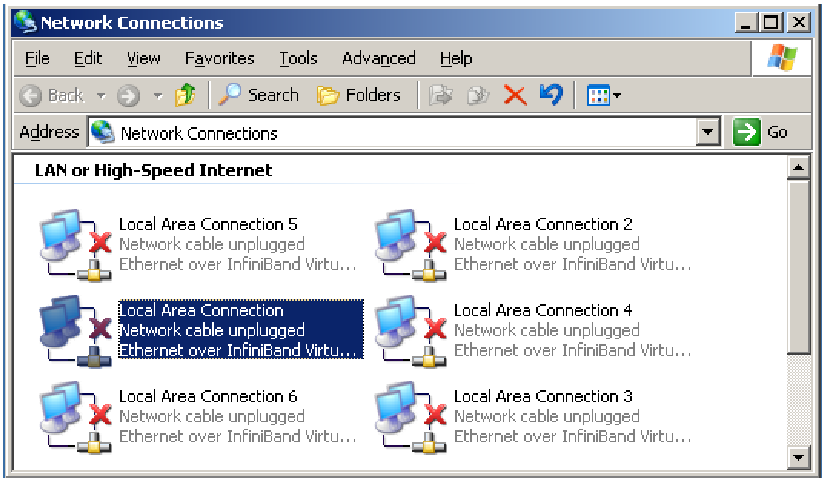
8. Repeat steps 5 through 7 for the other VNIC adapters. There is a total of six adapters. When the driver instillation is complete, you see six Virtual NICs under Network Adapters in the Device Manager as shown in Figure 4-45.



Monitors
Network adapters
Ethernet over InfiniBand Virtual NIC #10
Ethernet over InfiniBand Virtual NIC #11
Ethernet over InfiniBand Virtual NIC #12
Ethernet over InfiniBand Virtual NIC #7
Ethernet over InfiniBand Virtual NIC #8
Ethernet over InfiniBand Virtual NIC #9
Other devices
Ports (COM & LPT)
Processors

Figure 4-45 Virtual NICs in Device Manager

9. Make an Ethernet connection from the bridge to a test or other network. Inside Network Connections, you see six Local Area Connections, one for each virtual NIC, as shown in Figure 4-46.


Network Connections
File Edit View Favorites Tools Advanced Help
Back Search Folders
Address Network Connections Go

LAN or High-Speed Internet

Local Area Connection 5
Network cable unplugged
Ethernet over InfiniBand Virtu...

Local Area Connection 2
Network cable unplugged
Ethernet over InfiniBand Virtu...

Local Area Connection
Network cable unplugged
Ethernet over InfiniBand Virtu...

Local Area Connection 4
Network cable unplugged
Ethernet over InfiniBand Virtu...

Local Area Connection 6
Network cable unplugged
Ethernet over InfiniBand Virtu...

Local Area Connection 3
Network cable unplugged
Ethernet over InfiniBand Virtu...

Figure 4-46 Virtual NICs in Windows Network Connections

10. Set an IP address as you would with any Ethernet adapter and save your settings. Ping another device on the test network to test your connection as shown in Figure 4-47.

```
C:\>ping 192.168.70.111

Pinging 192.168.1.100 with 32 bytes of data:

Reply from 192.168.70.111: bytes=32 time<1ms TTL=128
Reply from 192.168.70.111: bytes=32 time<1ms TTL=128
Reply from 192.168.70.111: bytes=32 time<1ms TTL=128
Reply from 192.168.70.111: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.70.111:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>arp -a

Interface: 192.168.70.95 --- 0x10005
  Internet Address      Physical Address      Type
  192.168.70.111        00-16-41-12-7d-62     dynamic
```

Figure 4-47 A successful PING

4.2.3 QLogic InfiniServ Driver: Configuring IPoIB

In this section, we describe how to configure a network interface on a Windows blade using the QLogic InfiniServ drivers that will send IP traffic of the InfiniBand interfaces of the HCA. Because this method uses the InfiniBand ports on the HCA, you are limited by the number of HCA ports on the expansion card (two in this case). Because the IP traffic is over the InfiniBand interfaces and not translated into true Ethernet traffic, the Ethernet bridge is not needed. However without the bridge module, you cannot ping a true Ethernet device—just another InfiniBand interface running IP over InfiniBand. Follow these steps:

1. Windows finds the Open IPoIB Adapter. Select **Install from a specific location** and click **next**.
2. Click the **Browse** button under **Include this location in the search**. Select the following location and click **OK**:

   C:\Program Files\SilverStorm\SilverStorm HCA\net

3. Click **Next** to continue the driver installation. The Wizard finds and installs the Open IPoIB adapter.
4. Repeat steps 1 through 3 for the second IPoIB adapter. When the driver installation is complete, you should see two entries for the IPoIB in device manager under Network Adapters in Device Manager as shown in Figure 4-48.

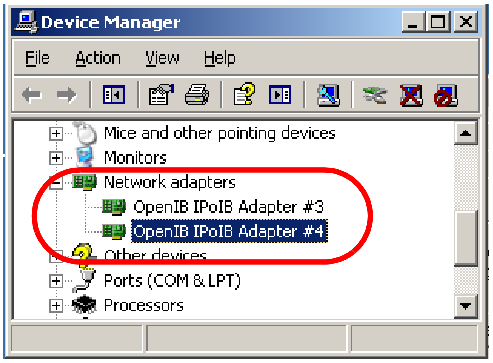
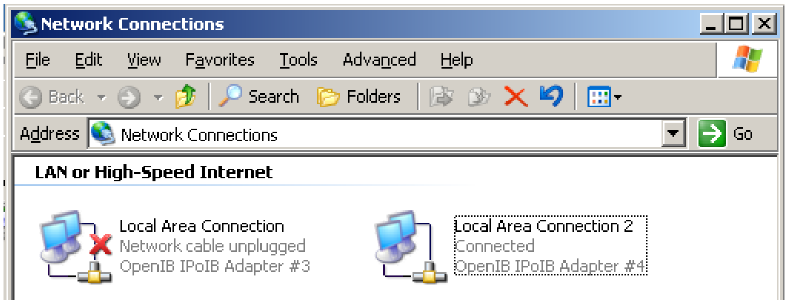


Figure 4-48 IP over InfiniBand adapters in Device Manager

5. Inside windows Network Connections, you see two Local Area Connections, one for each IPoIB interface, as shown in Figure 4-49.

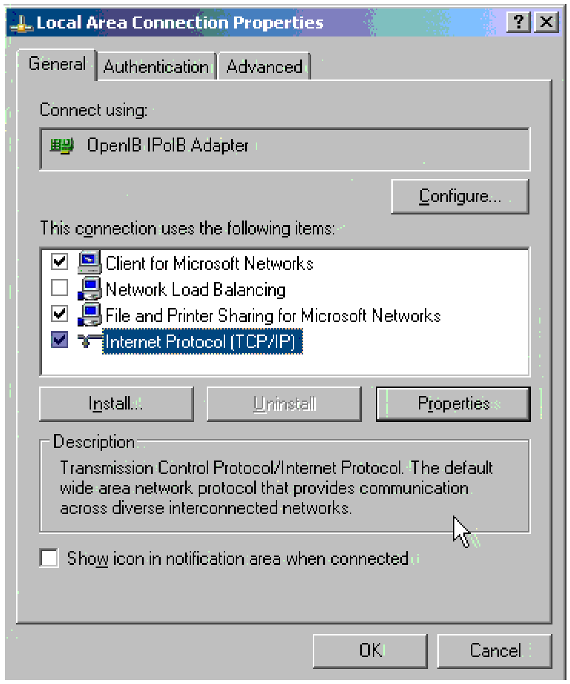


Local Area Connection
Network cable unplugged
OpenIB IPoIB Adapter #3

Local Area Connection 2
Connected
OpenIB IPoIB Adapter #4

Figure 4-49 IP over InfiniBand Adapters in Network Connections

6. Set an IP address as you would with any Ethernet adapter and save your settings. See Figure 4-50.



Connect using:

OpenIB IPoIB Adapter

This connection uses the following items:

- ☑ Client for Microsoft Networks
- □ Network Load Balancing
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ Internet Protocol (TCP/IP)

Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

□ Show icon in notification area when connected

Figure 4-50 Local Area Connection Properties

7. Ping another device on the test network to test your connection. If both blades are in the same chassis, you do not need any external cables as long as a functioning InfiniBand switch is in one of the supported I/O bays. If the devices are in two separate chassis, you need an InfiniBand cable to connect the InfiniBand switch modules in the respective chassis.

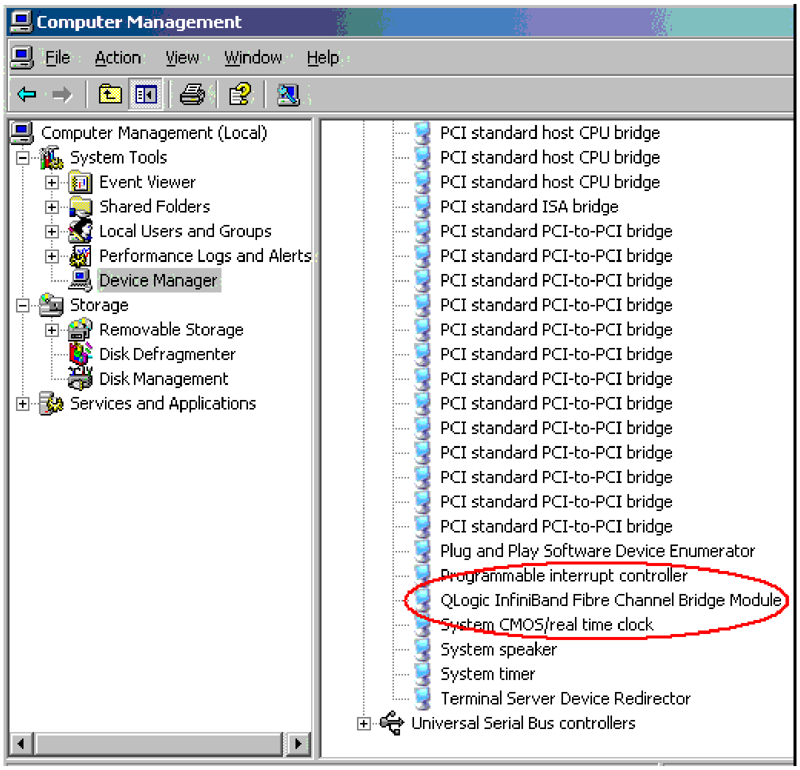
4.2.4 QLogic Fibre Channel Bridge: Configuring SRP

In this section, we describe how to configure a connection on a Windows blade to a remote Fibre Channel storage device using the QLogic InfiniServ drivers and the QLogic InfiniBand Fibre Channel Bridge Module. The InfiniBand HCA uses Globally Unique IDs (GUIDs) and Fibre Channel storage devices use World Wide Names (WWNs). The mapping of GUID to WWN is done on the bridge module. Follow these steps:

1. Windows finds the QLogic InfiniBand Fibre Channel Bridge Module. Select **Install from a specific location** and click **Next**.
2. Click **Browse** under **Include this location in the search**. Select the following location and then click **OK**.

   C:\Program Files\SilverStorm\SilverStorm HCA\storage

3. Click **Next** to continue driver installation. When the driver instillation is complete, you see the bridge module under **System Devices** in Device Manager, as shown in Figure 4-51.



Computer Management (Local)
System Tools
Event Viewer
Shared Folders
Local Users and Groups
Performance Logs and Alerts
Device Manager
Storage
Removable Storage
Disk Defragmenter
Disk Management
Services and Applications

PCI standard host CPU bridge
PCI standard host CPU bridge
PCI standard host CPU bridge
PCI standard ISA bridge
PCI standard PCI-to-PCI bridge
PCI standard PCI-to-PCI bridge
PCI standard PCI-to-PCI bridge
PCI standard PCI-to-PCI bridge
PCI standard PCI-to-PCI bridge
PCI standard PCI-to-PCI bridge
PCI standard PCI-to-PCI bridge
PCI standard PCI-to-PCI bridge
PCI standard PCI-to-PCI bridge
PCI standard PCI-to-PCI bridge
PCI standard PCI-to-PCI bridge
PCI standard PCI-to-PCI bridge
PCI standard PCI-to-PCI bridge
Plug and Play Software Device Enumerator
Programmable interrupt controller
QLogic InfiniBand Fibre Channel Bridge Module
System CMOS/real time clock
System speaker
System timer
Terminal Server Device Redirector
Universal Serial Bus controllers

Figure 4-51 Fibre Channel Bridge in Device Manager

4. Windows now finds the SilverStorm Technologies VFx I/O Controller. Select **Install from a specific location** and click **Next**.

5. Click **Browse** under **Include this location in the search**. Select the following location and then click **OK**.

   C:\Program Files\SilverStorm\SilverStorm HCA\storage

6. Click **Next** to continue driver installation. The wizard finds and installs the SilverStorm VFx I/O Controller.

7. Repeat steps 4 through 6 for the other VFx Controller. There are a total of two controllers. When the driver instillation is complete, you see two SilverStorm VFx I/O Controllers under SCSI and RAID Controllers in Device Manager as shown in Figure 4-52.

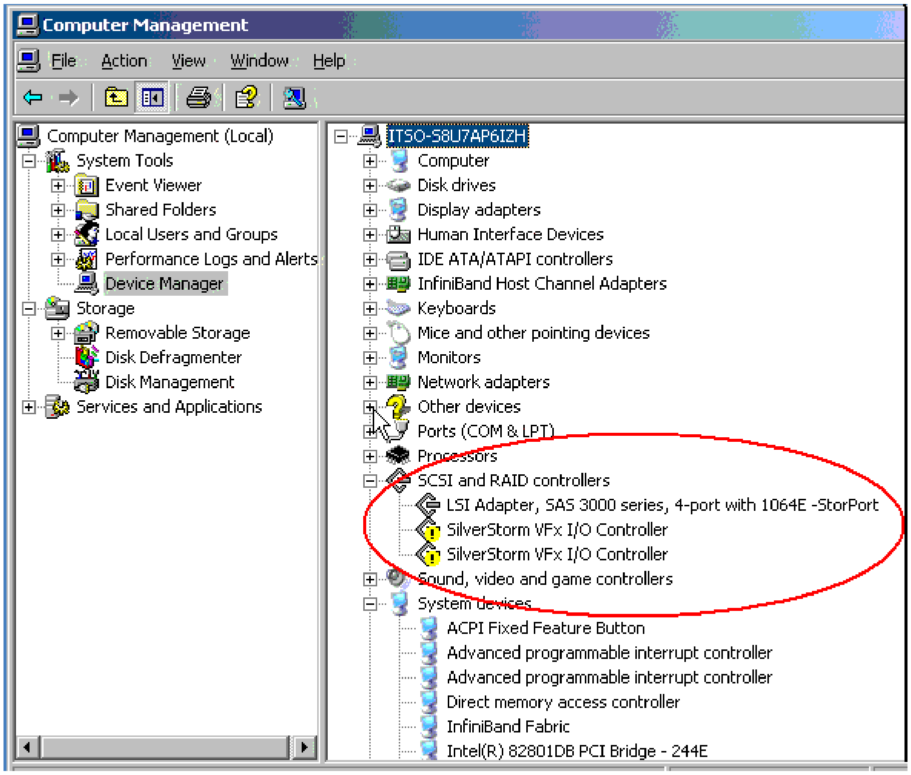
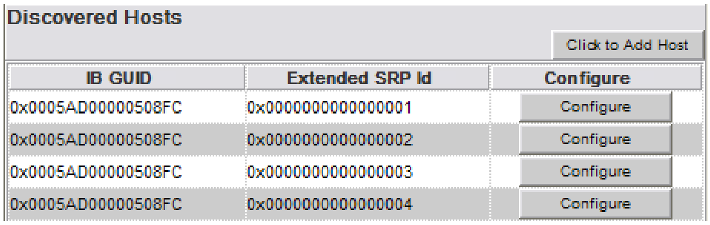

Computer Management
File Action View Window Help
Computer Management (Local)
System Tools
Event Viewer
Shared Folders
Local Users and Groups
Performance Logs and Alerts
Device Manager
Storage
Removable Storage
Disk Defragmenter
Disk Management
Services and Applications
ITSO-S8U7AP6IZH
Computer
Disk drives
Display adapters
Human Interface Devices
IDE ATA/ATAPI controllers
InfiniBand Host Channel Adapters
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Other devices
Ports (COM & LPT)
Processors
SCSI and RAID controllers
LSI Adapter, SAS 3000 series, 4-port with 1064E -StorPort
SilverStorm VFx I/O Controller
SilverStorm VFx I/O Controller
Sound, video and game controllers
System devices
ACPI Fixed Feature Button
Advanced programmable interrupt controller
Advanced programmable interrupt controller
Direct memory access controller
InfiniBand Fabric
Intel(R) 82801DB PCI Bridge - 244E

Figure 4-52 VFx Controllers in Device Manager

The VFx Controllers will have exclamation points (!) on them until an SRP mapping is created. Each bridge module has two IO Controllers (IOC), thus the two VFx controllers in Device Manager.

8. Open the Web interface of your bridge module. Then, click **SRP Initiator Discovery** and **Start**. The bridge module discovers your InfiniBand HCA and creates four **Extended SRP IDs** per HCA port as shown in Figure 4-53. These IDs are later mapped to the remote drives on the storage target.

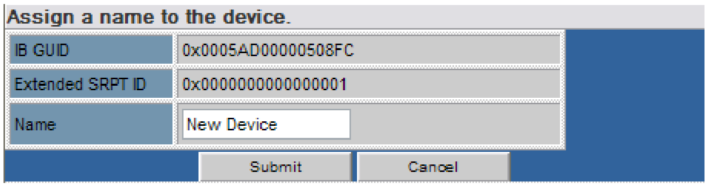


IB GUID	Extended SRP Id	Configure
0x0005AD00000508FC	0x0000000000000001	Configure
0x0005AD00000508FC	0x0000000000000002	Configure
0x0005AD00000508FC	0x0000000000000003	Configure
0x0005AD00000508FC	0x0000000000000004	Configure

Figure 4-53 Discovered SRP Initiators

In this example, only one InfiniBand switch was present in the BladeCenter chassis, so only one InfiniBand HCA port was available to the bridge module.

9. Click **Configure** to designate a name for the initiator (Figure 4-54). This name will be used later when mapping to a storage target.

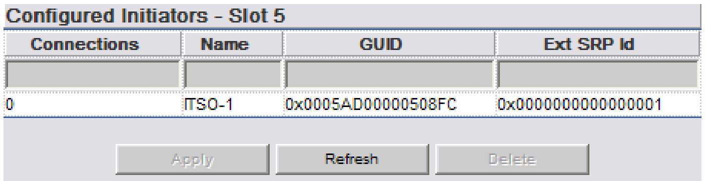


Assign a name to the device.

IB GUID	0x0005AD00000508FC
Extended SRPT ID	0x0000000000000001
Name	New Device

Figure 4-54 Assign name to SRP Initiator

10. After you designate a name, click **Submit**. The initiator moves from Discovered Hosts to Configured Initiators (Figure 4-55).

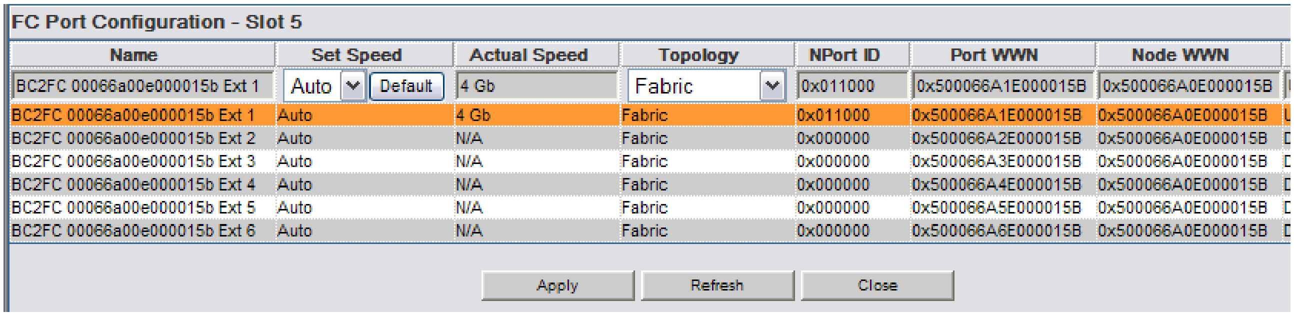


Configured Initiators - Slot 5

Connections	Name	GUID	Ext SRP Id
0	ITSO-1	0x0005AD00000508FC	0x0000000000000001

Figure 4-55 Configures SRP Initiator

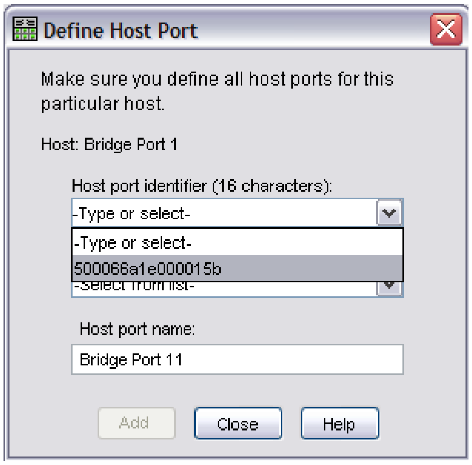
11. Each Fibre Channel port on the bridge module has a port WWN. On the storage device, the bridge port WWN needs to be mapped to the logical drives. A Fibre Channel connection has to be established between the bridge port and the storage device for the storage device to see the bridge port WWN.



FC Port Configuration - Slot 5

Name	Set Speed	Actual Speed	Topology	NPort ID	Port WWN	Node WWN	
BC2FC 00066a00e000015b Ext 1	Auto Default	4 Gb	Fabric	0x011000	0x500066A1E000015B	0x500066A0E000015B	
BC2FC 00066a00e000015b Ext 1	Auto	4 Gb	Fabric	0x011000	0x500066A1E000015B	0x500066A0E000015B	U
BC2FC 00066a00e000015b Ext 2	Auto	N/A	Fabric	0x000000	0x500066A2E000015B	0x500066A0E000015B	D
BC2FC 00066a00e000015b Ext 3	Auto	N/A	Fabric	0x000000	0x500066A3E000015B	0x500066A0E000015B	D
BC2FC 00066a00e000015b Ext 4	Auto	N/A	Fabric	0x000000	0x500066A4E000015B	0x500066A0E000015B	D
BC2FC 00066a00e000015b Ext 5	Auto	N/A	Fabric	0x000000	0x500066A5E000015B	0x500066A0E000015B	D
BC2FC 00066a00e000015b Ext 6	Auto	N/A	Fabric	0x000000	0x500066A6E000015B	0x500066A0E000015B	D

Figure 4-56 Bridge Port Link Information



Define Host Port

Make sure you define all host ports for this particular host.

Host: Bridge Port 1

Host port identifier (16 characters):

-Type or select-

-Type or select-

500066a1e000015b

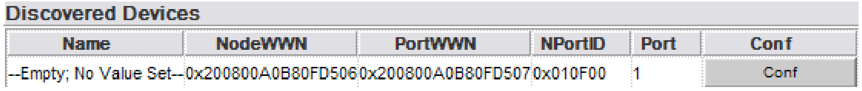
-Select from list-

Host port name:

Bridge Port 11

Figure 4-57 Bridge Port WWN in IBM System Storage™ DS4300

12. After you establish the bridge port to remote drive mapping, you need to establish another mapping from SRP initiator to remote drive. On the bridge Web GUI main menu, click **FCP Device Discovery**, and then **Start**. The remote drives that are mapped to the bridge ports will show up in Discovered Devices (Figure 4-58).

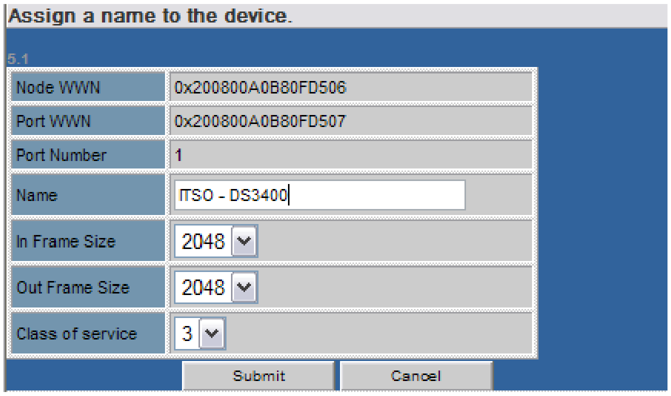


Discovered Devices

Name	NodeWWN	PortWWN	NPortID	Port	Conf
--Empty; No Value Set--	0x200800A0B80FD506	0x200800A0B80FD507	0x010F00	1	Conf

Figure 4-58 Discovered Fibre Channel Target

13. Click **Configure** to designate a name for the device (Figure 4-59). This name will be used later when mapping to an SRP Initiator.



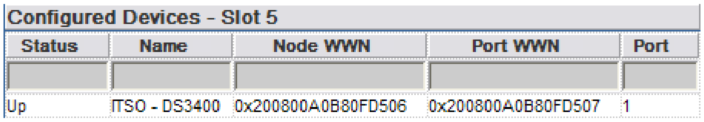
Assign a name to the device.

5.1

Node WWN	0x200800A0B80FD506
Port WWN	0x200800A0B80FD507
Port Number	1
Name	ITSO - DS3400
In Frame Size	2048
Out Frame Size	2048
Class of service	3

Figure 4-59 Assign name to Fibre Channel Storage Device

14. After you designate a name, click **Submit**. The remote logical drive moves from Discovered Devices to Configured Devices (Figure 4-60).



Configured Devices - Slot 5

Status	Name	Node WWN	Port WWN	Port
Up	ITSO - DS3400	0x200800A0B80FD506	0x200800A0B80FD507	1

Figure 4-60 Configured Fibre Channel Target

15. Close the FCP Device Discovery window and click **SRP map config** on the GUI main menu page. Here, you see your virtual HBAs listed. There will be two IOC headings, one for each I/O Controller on the bridge module. These correspond to the two VFx Controllers that you saw in device manager back in step 7:

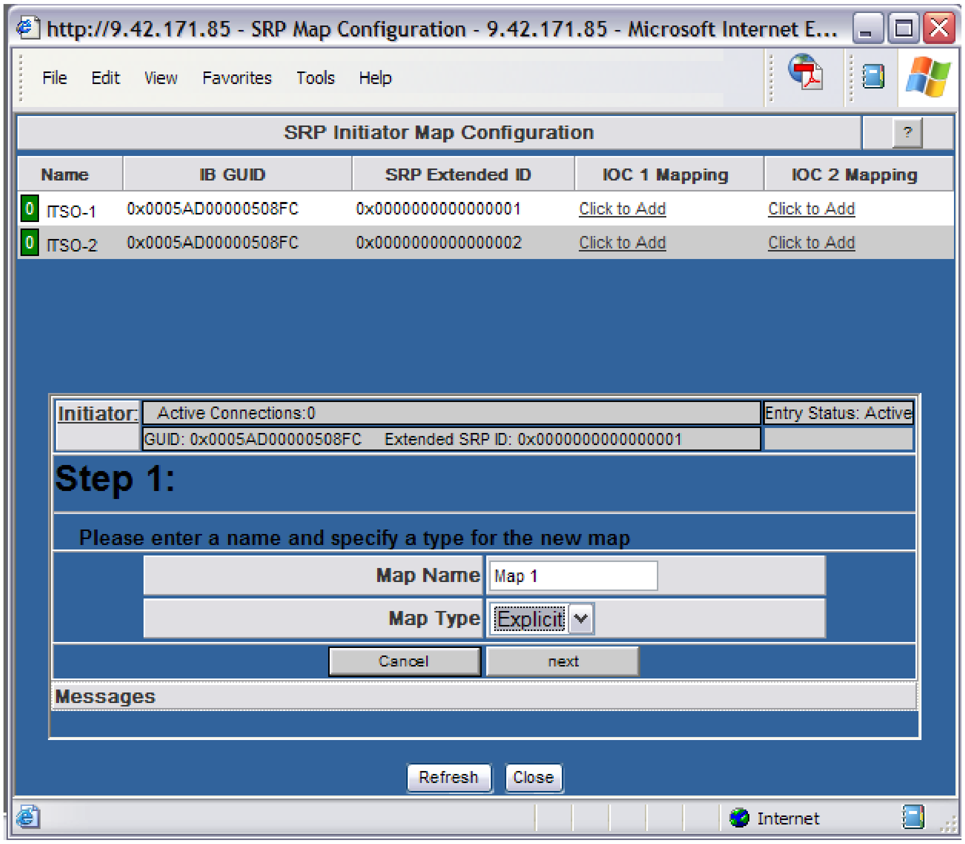
   – If you want to map your target through IOC 1, click the **Click To Add** link under the IOC 1 heading.
   – If you want to map your target through IOC 2, click the **Click To Add** link under the IOC 2 heading.

16. Give the map a name and select a map type, then click **Next**.

   There are two mapping types:

   – *Explicit*: In an explicit mapping, the host and target LUN numbers must be listed in the mapping.
   – *Direct*: In a direct mapping, the host will see all remote drives that are visible to the port WWNs without a specific LUN being identified.

   You address explicit mapping first, and then direct mapping.


http://9.42.171.85 - SRP Map Configuration - 9.42.171.85 - Microsoft Internet E...
File Edit View Favorites Tools Help
?
Cancel
next
Messages
Refresh
Close
Internet

SRP Initiator Map Configuration

Name	IB GUID	SRP Extended ID	IOC 1 Mapping	IOC 2 Mapping
0 ITSO-1	0x0005AD00000508FC	0x0000000000000001	Click to Add	Click to Add
0 ITSO-2	0x0005AD00000508FC	0x0000000000000002	Click to Add	Click to Add

Initiator: Active Connections:0 Entry Status: Active

GUID: 0x0005AD00000508FC Extended SRP ID: 0x0000000000000001

Step 1:

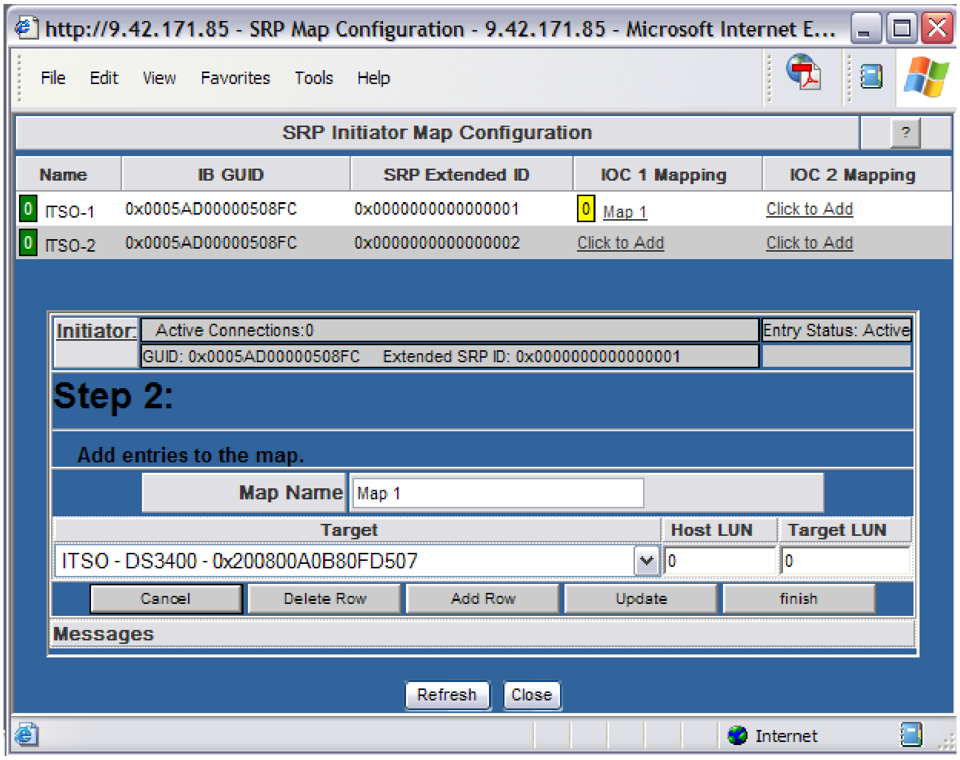
Please enter a name and specify a type for the new map

Map Name: Map 1

Map Type: Explicit

Figure 4-61 Add Explicit Map

17. Select a storage target and identify the host LUN and target LUN, then click **Add Row**. The host LUN is the LUN number that displays on the Linux host. The target LUN is LUN number that was determined for the logical drive on the storage device. When you are done adding all logical drives that you want mapped to this virtual HBA, click **Finish**.



SRP Initiator Map Configuration

Name	IB GUID	SRP Extended ID	IOC 1 Mapping	IOC 2 Mapping
0 ITSO-1	0x0005AD00000508FC	0x0000000000000001	0 Map 1	Click to Add
0 ITSO-2	0x0005AD00000508FC	0x0000000000000002	Click to Add	Click to Add

Initiator: Active Connections:0 Entry Status: Active
GUID: 0x0005AD00000508FC Extended SRP ID: 0x0000000000000001

Step 2:

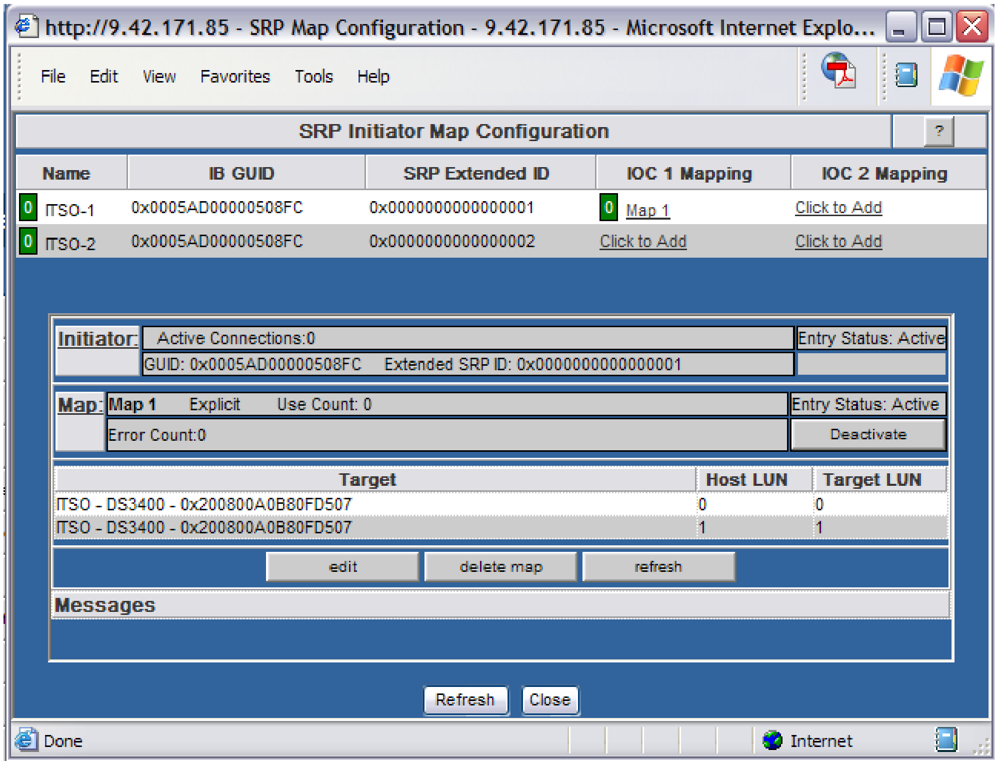
Add entries to the map.

Map Name: Map 1

Target	Host LUN	Target LUN
ITSO - DS3400 - 0x200800A0B80FD507	0	0

Messages

Figure 4-62 Add Remote Drive


http://9.42.171.85 - SRP Map Configuration - 9.42.171.85 - Microsoft Internet Explo...
File
Edit
View
Favorites
Tools
Help
?
Deactivate
edit
delete map
refresh
Refresh
Close
Done
Internet

SRP Initiator Map Configuration

Name	IB GUID	SRP Extended ID	IOC 1 Mapping	IOC 2 Mapping
0 ITSO-1	0x0005AD00000508FC	0x0000000000000001	0 Map 1	Click to Add
0 ITSO-2	0x0005AD00000508FC	0x0000000000000002	Click to Add	Click to Add

Initiator: Active Connections:0 Entry Status: Active
GUID: 0x0005AD00000508FC Extended SRP ID: 0x0000000000000001

Map: Map 1 Explicit Use Count: 0 Entry Status: Active
Error Count:0

Target	Host LUN	Target LUN
ITSO - DS3400 - 0x200800A0B80FD507	0	0
ITSO - DS3400 - 0x200800A0B80FD507	1	1

Messages

Figure 4-63 Completed SRP Map

18. Back on the Windows blade, open Device Manager. Find the VFx (I/O) Controller that you used to create your SRP mapping. Right-click the controller to disable and then re-enable

it. The exclamation point (!) should go away, and your mapping is now active. I/O Controller 1 was used for the explicit mapping.

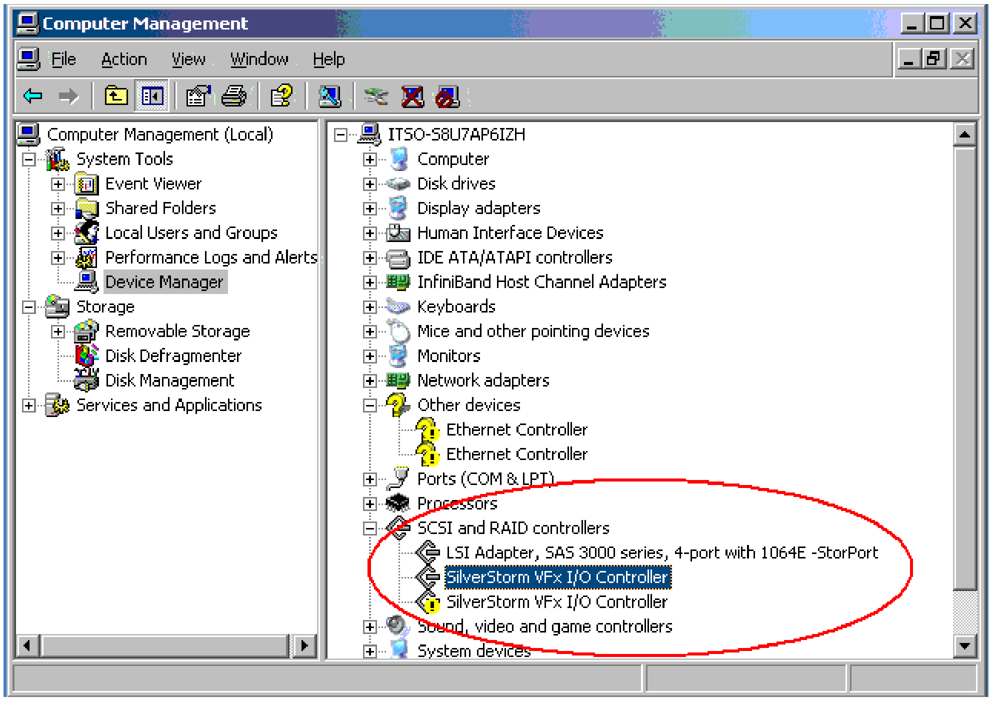
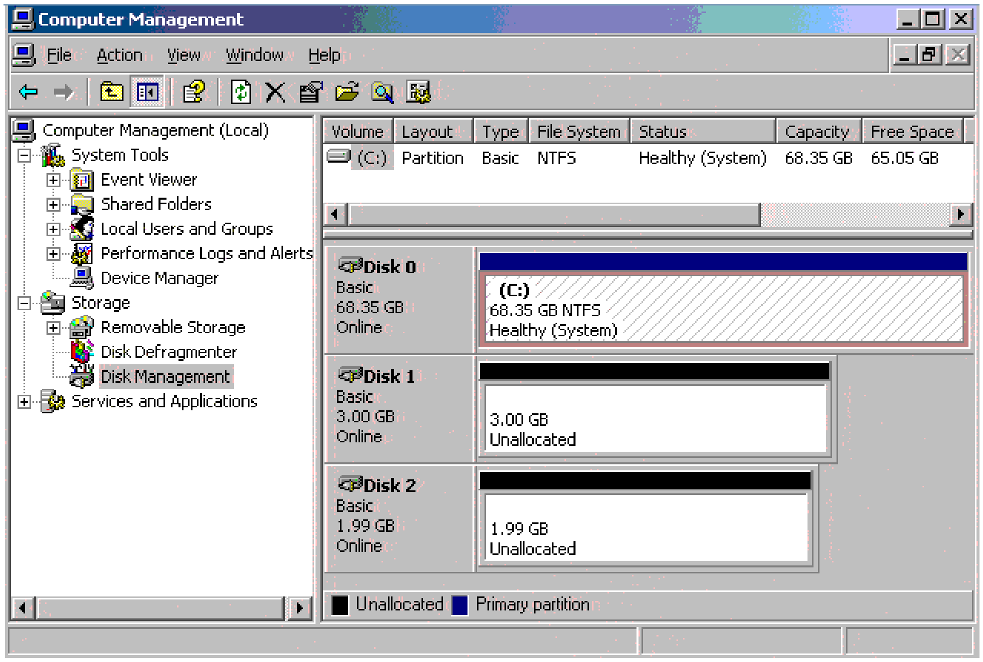

Computer Management
File Action View Window Help
Computer Management (Local)
System Tools
Event Viewer
Shared Folders
Local Users and Groups
Performance Logs and Alerts
Device Manager
Storage
Removable Storage
Disk Defragmenter
Disk Management
Services and Applications
ITSO-S8U7AP6IZH
Computer
Disk drives
Display adapters
Human Interface Devices
IDE ATA/ATAPI controllers
InfiniBand Host Channel Adapters
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Other devices
Ethernet Controller
Ethernet Controller
Ports (COM & LPT)
Processors
SCSI and RAID controllers
LSI Adapter, SAS 3000 series, 4-port with 1064E -StorPort
SilverStorm VFx I/O Controller
SilverStorm VFx I/O Controller
Sound, video and game controllers
System devices

Figure 4-64 VFx Controller in Device Manager

19. You should now see your remote drives in Disk Management, as shown in Figure 4-65.



Volume	Layout	Type	File System	Status	Capacity	Free Space
(C:)	Partition	Basic	NTFS	Healthy (System)	68.35 GB	65.05 GB

Disk 0
Basic
68.35 GB
Online

(C:)
68.35 GB NTFS
Healthy (System)

Disk 1
Basic
3.00 GB
Online

3.00 GB
Unallocated

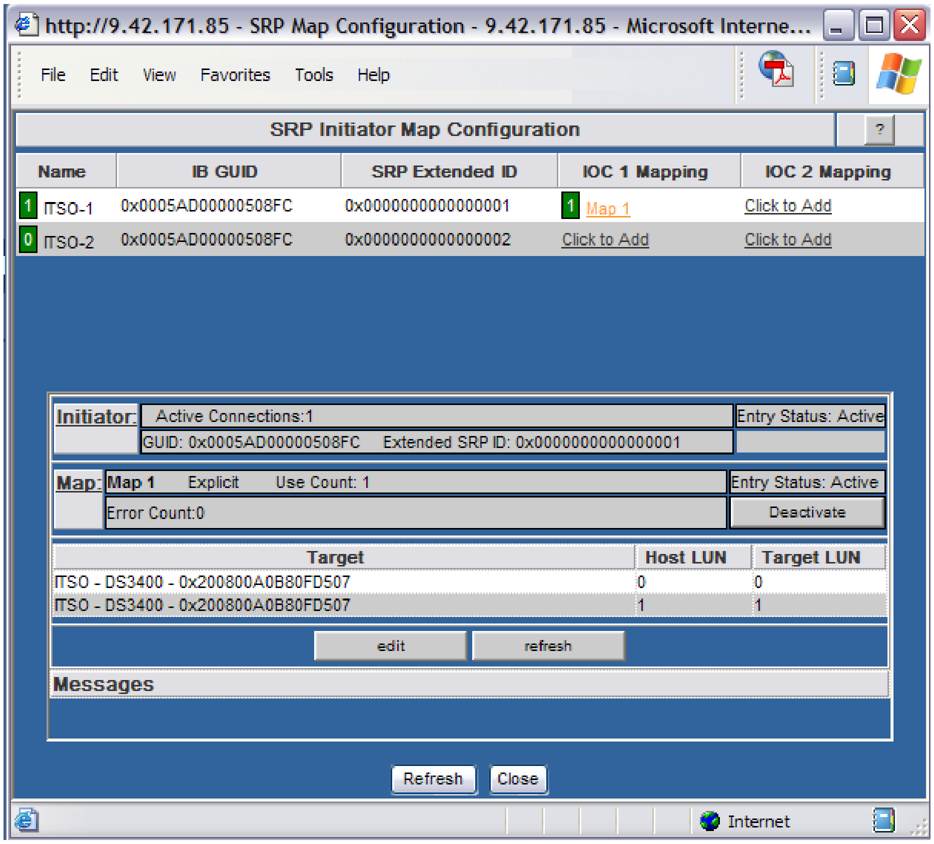
Disk 2
Basic
1.99 GB
Online

1.99 GB
Unallocated

■ Unallocated ■ Primary partition

Figure 4-65 Remote Drives in Disk Management

20. On the bridge web GUI, the boxes beside the SRP initiator name and the IOC Map name indicate the active connections to the virtual HBA, as shown in Figure 4-66. It should go from 0 to 1. You might need to click **Refresh** on the SRP map page.


http://9.42.171.85 - SRP Map Configuration - 9.42.171.85 - Microsoft Interne...
File
Edit
View
Favorites
Tools
Help
?
edit
refresh
Messages
Refresh
Close
Internet

SRP Initiator Map Configuration

Name	IB GUID	SRP Extended ID	IOC 1 Mapping	IOC 2 Mapping
1 ITSO-1	0x0005AD00000508FC	0x0000000000000001	1 Map 1	Click to Add
0 ITSO-2	0x0005AD00000508FC	0x0000000000000002	Click to Add	Click to Add

Initiator: Active Connections:1 — Entry Status: Active
GUID: 0x0005AD00000508FC Extended SRP ID: 0x0000000000000001

Map: Map 1 Explicit Use Count: 1 — Entry Status: Active
Error Count:0 — Deactivate

Target	Host LUN	Target LUN
ITSO - DS3400 - 0x200800A0B80FD507	0	0
ITSO - DS3400 - 0x200800A0B80FD507	1	1

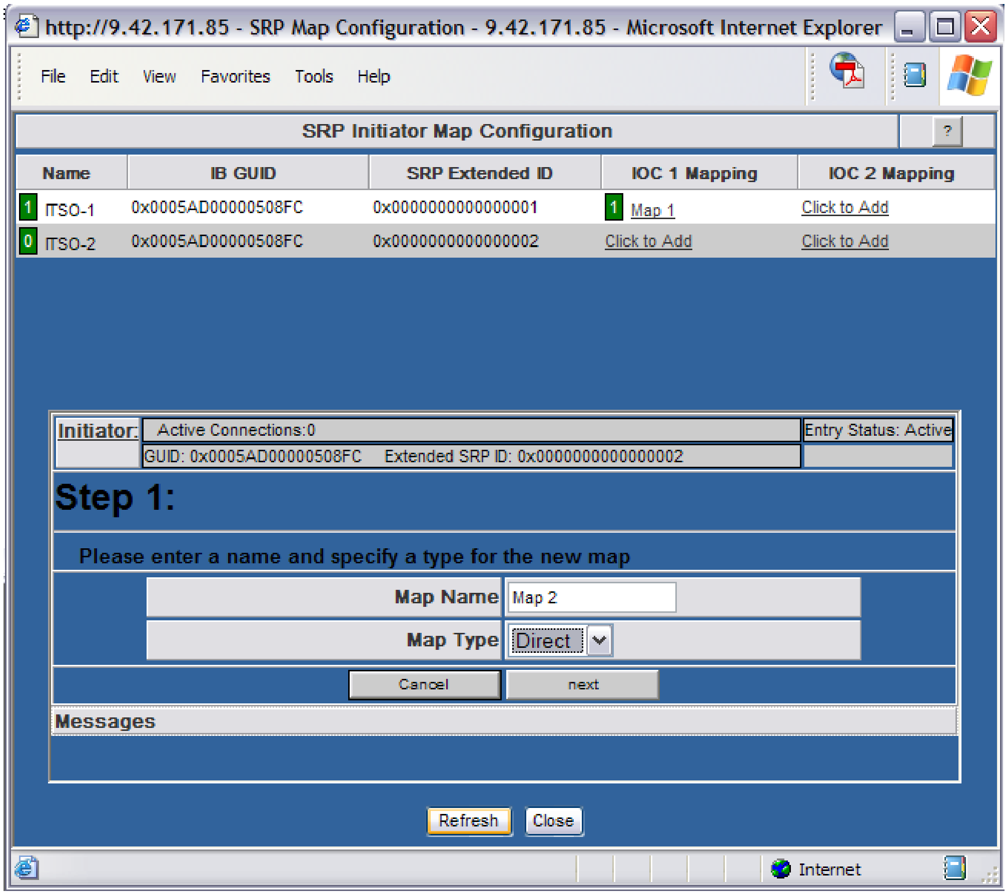
Figure 4-66 Active SRP Map

21. Now you create a direct mapping. Click **SRP map config** on the GUI main menu page. Here you see the virtual HBAs listed:

– If you want to map your target through IOC 1, click the **Click To Add** link under the IOC 1 heading.

– If you want to map your target through IOC 2, click the **Click To Add** link under the IOC 2 heading.

22. Give the map a name and select **Direct** type, and then click **Next**.



http://9.42.171.85 - SRP Map Configuration - 9.42.171.85 - Microsoft Internet Explorer
File Edit View Favorites Tools Help

SRP Initiator Map Configuration

Name	IB GUID	SRP Extended ID	IOC 1 Mapping	IOC 2 Mapping
1 ITSO-1	0x0005AD00000508FC	0x0000000000000001	1 Map 1	Click to Add
0 ITSO-2	0x0005AD00000508FC	0x0000000000000002	Click to Add	Click to Add

Initiator: Active Connections:0 Entry Status: Active
GUID: 0x0005AD00000508FC Extended SRP ID: 0x0000000000000002

Step 1:

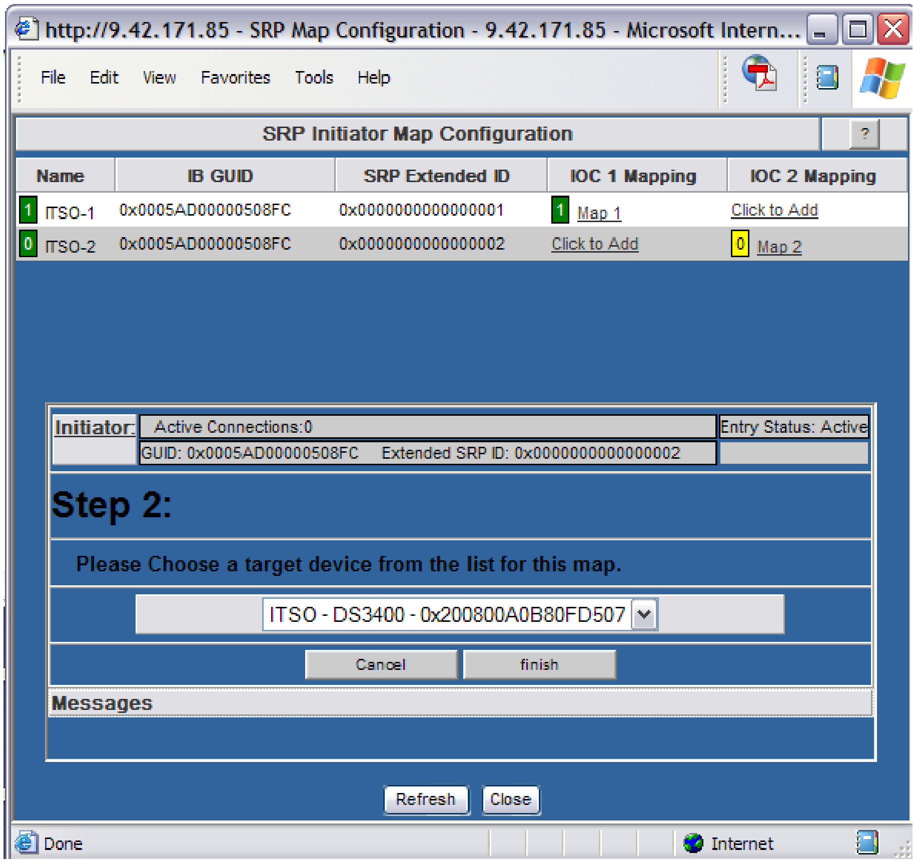
Please enter a name and specify a type for the new map

Map Name: Map 2

Map Type: Direct

Figure 4-67 Add Direct Map

22. Select a storage target and click **Finish**.


http://9.42.171.85 - SRP Map Configuration - 9.42.171.85 - Microsoft Intern...
File Edit View Favorites Tools Help
?
Cancel
finish
Refresh
Close
Done
Internet

SRP Initiator Map Configuration

Name	IB GUID	SRP Extended ID	IOC 1 Mapping	IOC 2 Mapping
1 ITSO-1	0x0005AD00000508FC	0x0000000000000001	1 Map 1	Click to Add
0 ITSO-2	0x0005AD00000508FC	0x0000000000000002	Click to Add	0 Map 2

Initiator: Active Connections:0 Entry Status: Active

GUID: 0x0005AD00000508FC Extended SRP ID: 0x0000000000000002

Step 2:

Please Choose a target device from the list for this map.

ITSO - DS3400 - 0x200800A0B80FD507

Messages

Figure 4-68 Choose Direct Map

23.Back on the windows blade, open device manager. Find the VFx (I/O) Controller that you used to create your direct SRP mapping (Figure 4-69). Right-click the controller to disable then re-enable it. The exclamation point (!) should go away and your mapping is now active. I/O Controller 2 was used for the direct mapping.

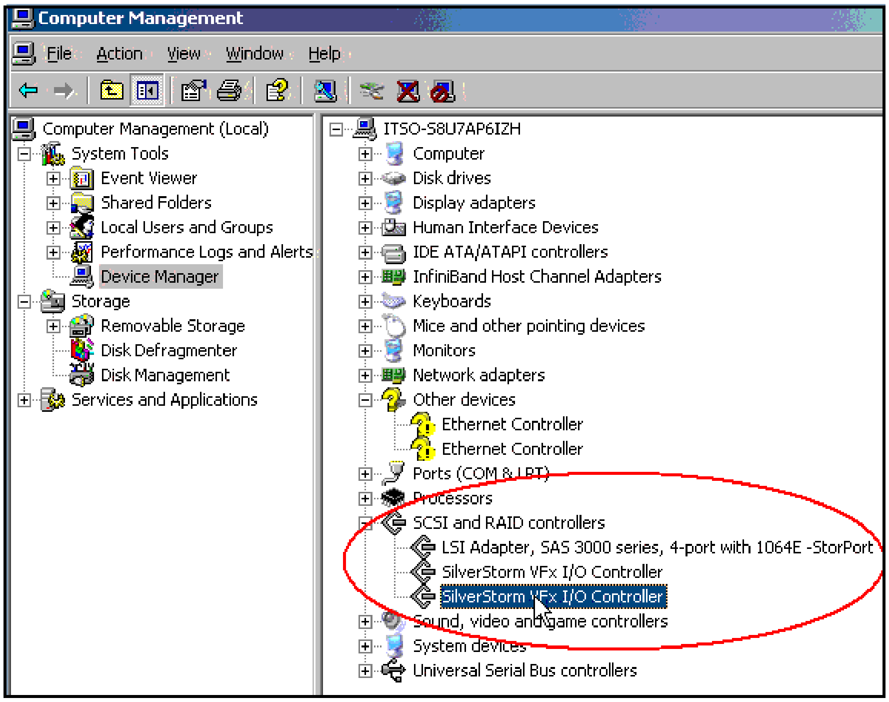
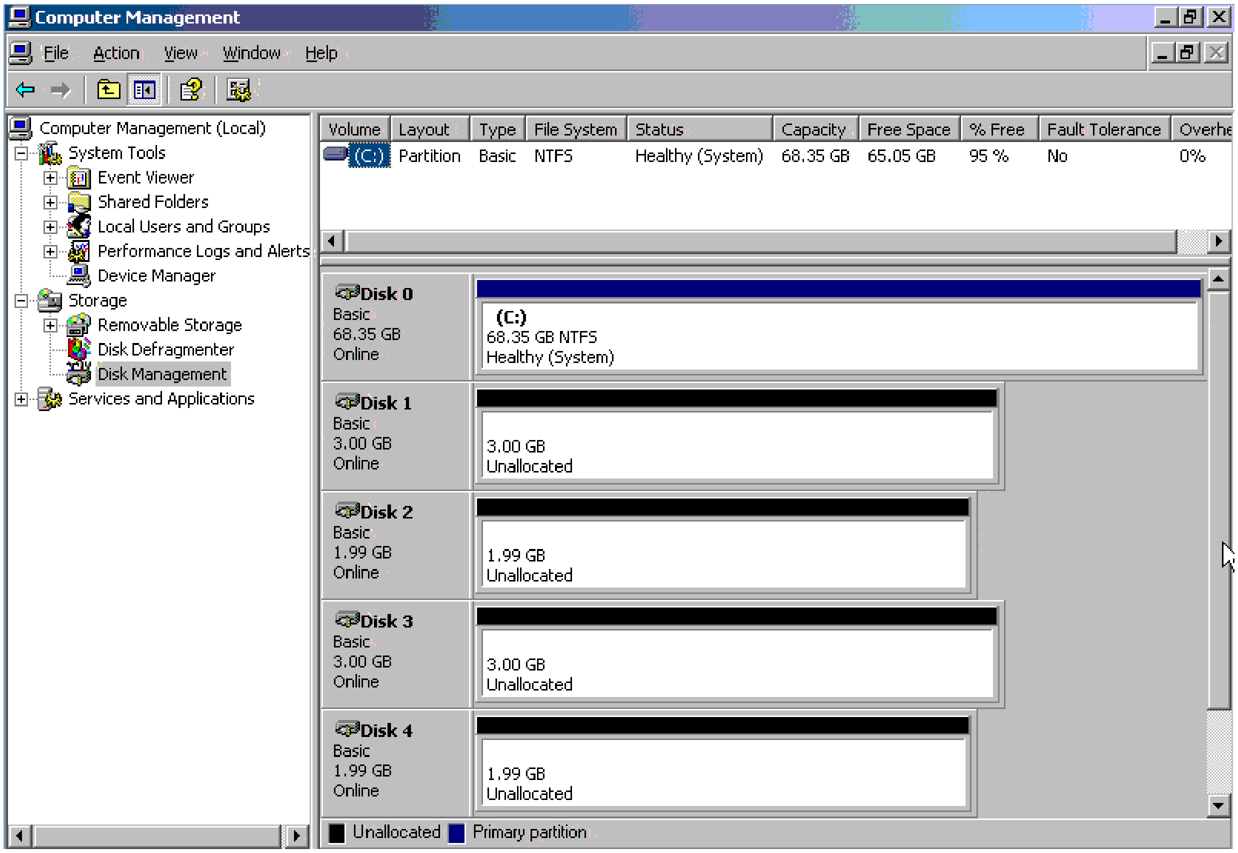

Computer Management
File
Action
View
Window
Help
Computer Management (Local)
System Tools
Event Viewer
Shared Folders
Local Users and Groups
Performance Logs and Alerts
Device Manager
Storage
Removable Storage
Disk Defragmenter
Disk Management
Services and Applications
ITSO-S8U7AP6IZH
Computer
Disk drives
Display adapters
Human Interface Devices
IDE ATA/ATAPI controllers
InfiniBand Host Channel Adapters
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Other devices
Ethernet Controller
Ethernet Controller
Ports (COM & LPT)
Processors
SCSI and RAID controllers
LSI Adapter, SAS 3000 series, 4-port with 1064E -StorPort
SilverStorm VFx I/O Controller
SilverStorm VFx I/O Controller
Sound, video and game controllers
System devices
Universal Serial Bus controllers

Figure 4-69 VFx I/O Controller now enabled in Device Manager

24. You should now see your additional remote drives in disk management as shown in Figure 4-70.



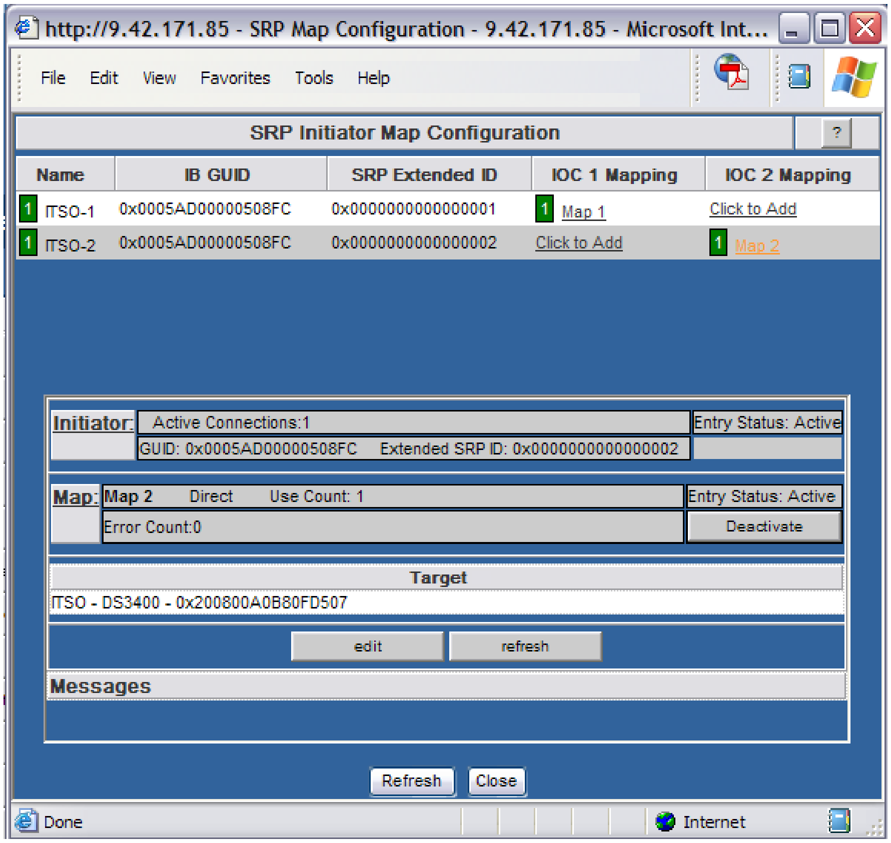
Volume	Layout	Type	File System	Status	Capacity	Free Space	% Free	Fault Tolerance	Overhe
(C:)	Partition	Basic	NTFS	Healthy (System)	68.35 GB	65.05 GB	95 %	No	0%

Disk 0 Basic 68.35 GB Online	**(C:)** 68.35 GB NTFS Healthy (System)
Disk 1 Basic 3.00 GB Online	3.00 GB Unallocated
Disk 2 Basic 1.99 GB Online	1.99 GB Unallocated
Disk 3 Basic 3.00 GB Online	3.00 GB Unallocated
Disk 4 Basic 1.99 GB Online	1.99 GB Unallocated

■ Unallocated ■ Primary partition

Figure 4-70 Remote Drives in Disk Management

25. On the bridge web GUI, the boxes beside the SRP initiator name and the IOC Map name indicate the active connections to the virtual HBA, as shown in Figure 4-71. It should go from 0 to 1. You might need to click **Refresh** on the SRP map page.


http://9.42.171.85 - SRP Map Configuration - 9.42.171.85 - Microsoft Int...
File
Edit
View
Favorites
Tools
Help
?
Done
Internet

SRP Initiator Map Configuration

Name	IB GUID	SRP Extended ID	IOC 1 Mapping	IOC 2 Mapping
1 ITSO-1	0x0005AD00000508FC	0x0000000000000001	1 Map 1	Click to Add
1 ITSO-2	0x0005AD00000508FC	0x0000000000000002	Click to Add	1 Map 2

Initiator:	Active Connections:1	Entry Status: Active
	GUID: 0x0005AD00000508FC Extended SRP ID: 0x0000000000000002	
Map:	Map 2 Direct Use Count: 1	Entry Status: Active
	Error Count:0	Deactivate

Target
ITSO - DS3400 - 0x200800A0B80FD507

edit refresh

Messages

Refresh Close

Figure 4-71 Active Direct SRP Map

5

Configuring IP over InfiniBand for use with Cisco switches

This chapter covers the details of implementing IP over InfiniBand (IPoIB) on BladeCenter HS21 servers with the Cisco InfiniBand SDR dual port 4X Host Channel Adapter (HCA) installed. It includes procedures for both Red Hat Enterprise Linux 4 U4 and Windows Server® 2003 x64.

For the two servers (IPoIB nodes) to communicate using the IPoIB protocol, they first need to be connected to the InfiniBand fabric through at least one of the ports. In the BladeCenter, this is done by installing a Cisco 4X InfiniBand switch module in bay 7 or 9 (or both), and then installing a blade server containing an InfiniBand HCA in the blade slots in the front of the BladeCenter. This connects the servers to the InfiniBand fabric through the BladeCenter midplane automatically.

Note: This chapter specifically deals with an installation using the internal Cisco 4X InfiniBand switch modules, which interconnect all HCAs in a given BladeCenter. If you are using the InfiniBand Pass-thru Module, you will need to have some sort of external InfiniBand switch to allow HCAs in the same BladeCenter to talk with each other.

In the examples in this chapter, the two IPoIB nodes are installed in the same BladeCenter H chassis in blade bays 10 and 11. In Figure 5-1 on page 82, we use the Cisco SFS Element Manager GUI to show the LED status is solid green for each host, which confirms that the two blades are connected to the Cisco 4x SDR InfiniBand switch module in module bay 9 of the chassis.

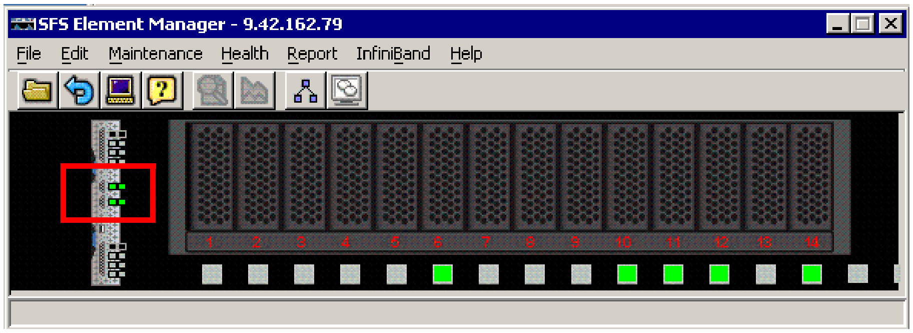


Figure 5-1 BCH high speed InfiniBand Switch Module GUI

The topics that we discuss in this chapter are:

- 5.1, "Configuring IPoIB on a Linux host" on page 82
- 5.2, "Configuring the IPoIB on a Windows host" on page 88

5.1 Configuring IPoIB on a Linux host

This section covers the step-by-step procedure of installing the InfiniBand host drivers and configuring IPoIB on the blade servers running Linux.

5.1.1 Installing the OFED drivers

OpenFabrics Enterprise Distribution (OFED) is open source software, from the OpenFabrics Alliance, for RDMA fabric technologies that works independent of the multiple underlying networking layers. The OpenFabrics Alliance 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.

The OpenFabrics Alliance is comprised of technology vendors and user organizations.

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

http://www.openfabrics.org/

5.1.2 Cisco InfiniBand host drivers

This section describes the procedure to install the Cisco OFED InfiniBand Host Drivers on the blade with RHEL 5 operating system. Cisco provides two versions of InfiniBand host drivers:

- Cisco OFED Driver
- SRP Host Drivers

The choice of which driver to use will be based on the customers specific requirements, which is usually driven by what upper layer applications will be making use of the drivers. In this section we will be focusing on the Cisco OFED drivers.

For more information about the Cisco InfiniBand host drivers, refer to the release notes and user guide available at the Cisco software download site. You can obtain the Cisco OFED and SRP host drivers for Linux download site from:

http://www.cisco.com/pcgi-bin/tablebuild.pl?topic=280511613

Installing the InfiniBand host drivers

Install the drivers as follows:

1. Download the InfiniBand Host Driver ISO image from the URL above
2. Copy the driver ISO file to the blade server
3. Mount the ISO image as shown in Example 5-1.

Example 5-1 Mount the Cisco OFED driver ISO on the blade server

```
[root@localhost ~]# mount -o loop /temp/Cisco_OFED-1.2-fcs.iso /mnt
[root@localhost ~]# cd /mnt
[root@localhost mnt]# ls
docs  examples  extras  firmware  ofedinstall  rhel4  rhel5  sles10  sles9  src
```

4. From the /mnt directory, run the installation script as shown in Example 5-2.

> **Note:** The driver installation script also checks for the firmware currently installed on the HCA and if it is down level, then it automatically updates the firmware on the HCA as shown in the following example.

Example 5-2 Example of installation progress for OFED drivers

```
[root@localhost 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:
  2.6.9-42.EL.x86_64

Drivers for the following kernels will be installed.
  2.6.9:42.EL:x86_64:smp

The following installed OpenIB packages conflict with the packages to be
installed and will be removed:
  libsdp-0.9.0-1
  libibcommon-1.0-1
  perftest-1.0-1
  libibverbs-devel-1.0.3-1
  openib-diags-1.0-1
  libibumad-1.0-1
  librdmacm-0.9.0-1
  libibmad-1.0-1
  libmthca-1.0.2-1
  dapl-devel-1.2.0-1
  libibcommon-devel-1.0-1
  libipathverbs-devel-1.0-1
  opensm-devel-1.2.0-1
  libibmad-devel-1.0-1
  opensm-libs-1.2.0-1
  dapl-1.2.0-1
  libibcm-0.9.0-1
  libibverbs-utils-1.0.3-1
  mstflint-1.0-1
  libipathverbs-1.0-1
  libibverbs-1.0.3-1
  libibumad-devel-1.0-1
  kernel-ib-1.0-1
  libmthca-devel-1.0.2-1
  opensm-1.2.0-1
  srptools-0.0.4-1
  tvflash-0.9.0-1

Installing packages.
Preparing...                ########################################### [100%]
   1:kernel-ib              ########################################### [100%]
Preparing...                ########################################### [100%]
   1:kernel-ib-devel        ########################################### [100%]
Preparing...                ########################################### [100%]
   1:ib-bonding             ########################################### [100%]
Preparing...                ########################################### [100%]
   1:tvflash                ########################################### [  2%]
   2:dapl                   ########################################### [  4%]
   3:dapl-devel             ########################################### [  6%]
   4:dapl-utils             ########################################### [  8%]
   5:ibutils                ########################################### [  9%]
   6:libibcommon            ########################################### [ 11%]
   7:libibcommon-devel      ########################################### [ 13%]
   8:libibmad               ########################################### [ 15%]
   9:libibmad-devel         ########################################### [ 17%]
  10:libibumad              ########################################### [ 19%]
  11:libibumad-devel        ########################################### [ 21%]
  12:libibverbs             ########################################### [ 23%]
  13:libibverbs-devel       ########################################### [ 25%]
  14:libibverbs-utils       ########################################### [ 26%]
  15:libmthca               ########################################### [ 28%]
  16:libmthca-devel         ########################################### [ 30%]
  17:libopensm              ########################################### [ 32%]
  18:libopensm-devel        ########################################### [ 34%]
  19:libosmcomp             ########################################### [ 36%]
  20:libosmcomp-devel       ########################################### [ 38%]
  21:libosmvendor           ########################################### [ 40%]
  22:libosmvendor-devel     ########################################### [ 42%]
  23:librdmacm              ########################################### [ 43%]
  24:librdmacm-devel        ########################################### [ 45%]
  25:librdmacm-utils        ########################################### [ 47%]
  26:libsdp                 ########################################### [ 49%]
  27:mpi-selector           ########################################### [ 51%]
  28:mpitests_mvapich2_gcc  ########################################### [ 53%]
  29:mpitests_mvapich2_intel########################################### [ 55%]
  30:mpitests_mvapich2_pgi  ########################################### [ 57%]
  31:mpitests_mvapich_gcc   ########################################### [ 58%]
  32:mpitests_mvapich_intel ########################################### [ 60%]
  33:mpitests_mvapich_pgi   ########################################### [ 62%]
  34:mpitests_openmpi_gcc   ########################################### [ 64%]
  35:mpitests_openmpi_intel ########################################### [ 66%]
  36:mpitests_openmpi_pgi   ########################################### [ 68%]
  37:mstflint               ########################################### [ 70%]
  38:mvapich2_gcc           ########################################### [ 72%]
  39:mvapich2_intel         ########################################### [ 74%]
  40:mvapich2_pgi           ########################################### [ 75%]
  41:mvapich_gcc            ########################################### [ 77%]
  42:mvapich_intel          ########################################### [ 79%]
  43:mvapich_pgi            ########################################### [ 81%]
  44:ofed-docs              ########################################### [ 83%]
  45:ofed-scripts           ########################################### [ 85%]
  46:openib-diags           ########################################### [ 87%]
  47:openmpi_gcc            ########################################### [ 89%]
  48:openmpi_intel          ########################################### [ 91%]
  49:openmpi_pgi            ########################################### [ 92%]
  50:perftest               ########################################### [ 94%]
  51:rds-tools              ########################################### [ 96%]
  52:sdpnetstat             ########################################### [ 98%]
  53:srptools               ########################################### [100%]

Installing hca_self_test and MPI examples.

Configuring /etc/InfiniBand/openib.conf and /etc/security/limits.conf.
Upgrading HCA 0 HCA.HSDC.A0.Boot to firmware build 3.2.0.149
New Node  GUID = 0005ad00000ce994
New Port1 GUID = 0005ad00000ce995
New Port2 GUID = 0005ad00000ce996
WARNING: Out-of-date Invariant sector found. Flash reprogramming won't be
failsafe, continuing in 10 seconds
Programming HCA firmware... Flash Image Size = 395040
Flashing - EIIVVVVEFFFEPPPEWWWEWWWEWWWEWWWEWWWEWWWEWVVVVVVVVVVVVVVVVVVVVVVVVVV
Flash verify passed!

Please reboot your system for the changes to take effect.

[root@localhost mnt]# reboot

Broadcast message from root (pts/2) (Fri Aug 24 10:03:35 2007):

The system is going down for reboot NOW!
[root@localhost mnt]# Connection to 9.42.166.54 closed.
[root@localhost mnt]#
```

5. Notice at the end of the installation in Example 5-2 that we reboot the server to allow the new drivers to load.

5.1.3 Configuring IP over InfiniBand

The IPoIB protocol passes IP traffic over the InfiniBand network. Configuring IPoIB is very similar to configuring IP on the Ethernet interfaces. The procedure for configuring IPoIB using the Cisco SRP drivers is also the same, but not specifically shown in this chapter.

To configure IPoIB, you assign an IP address and subnet mask to each InfiniBand port on the host. IPoIB automatically adds InfiniBand interface names to the IP network configuration. To configure IPoIB, perform the following steps:

1. From the command prompt, assign the IP address and subnet mask as shown in Example 5-3.

Example 5-3 Command to directly place an IP address on the IB interface

```
host1# ifconfig ib0 172.16.240.10 netmask 255.255.255.0
```

2. The IPoIB configuration is not persistent across the reboots if performed using the procedure specified above. To keep the IPoIB configuration persistent across reboots, and ensure the interface initializes at boot, create a new script file for the InfiniBand interface (ib0 in this example) and enter the device name, IP address and subnet mask information as shown in Example 5-4.

Example 5-4 Configuring the IPoIB interfaces on the blade server

```
[root@localhost ~]# vi /etc/sysconfig/network-scripts/ifcfg-ib0
DEVICE=ib0
BOOTPROTO=static
IPADDR=172.16.240.10
NETMASK=255.255.255.0
ONBOOT=yes
```

3. After the configuration parameters listed in Example 5-4 are entered in /etc/sysconfig/network-scripts/ifcfg-ib0 file, save and close the file.
4. Follow the same procedure to configure additional InfiniBand interfaces on the same or different blade servers.
5. If you used the procedure listed in Example 5-3, you still need to initialize the IPoIB interfaces, which is done by restarting the network services as shown in Example 5-5:

Example 5-5 Initialize the IPoIB interface

```
[root@localhost ~]# service network restart
Shutting down interface eth0:                              [  OK  ]
Shutting down interface eth1:                              [  OK  ]
Shutting down interface ib0:                               [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface eth0:
Determining IP information for eth0... done.
                                                           [  OK  ]
Bringing up interface ib0:                                 [  OK  ]
[root@localhost ~]#
```

6. Issue the command shown in Example 5-6 to verify the ib0 interface status.

Example 5-6 Verify the ib0 interface

```
[root@localhost ~]# ifconfig ib0
```

```
ib0       Link encap:InfiniBand  HWaddr
80:00:04:04:FE:80:00:00:00:00:00:00:00:00:00:00:00:00:00:00
          inet addr:172.16.240.10  Bcast:172.16.240.255  Mask:255.255.255.0
          inet6 addr: fe80::205:ad00:5:391/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:65520  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:79 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:128
          RX bytes:0 (0.0 b)  TX bytes:17380 (16.9 KiB)
```

5.1.4 Verifying the IPoIB communication

To verify IPoIB functionality, follow these steps:

1. Log in to the hosts.
2. Ensure that the IPoIB interfaces on the source and destination nodes are up by using the **ifconfig** command as shown in Example 5-6 on page 86.
3. Issue a **ping** command from source to a destination node as shown in Example 5-7.

Example 5-7 Verify IPoIB communication between the two InfiniBand hosts

```
[root@localhost ~]# ping 172.16.240.11
PING 172.16.240.11 (172.16.240.11) 56(84) bytes of data.
64 bytes from 172.16.240.11: icmp_seq=1 ttl=64 time=0.043 ms
64 bytes from 172.16.240.11: icmp_seq=2 ttl=64 time=0.042 ms
64 bytes from 172.16.240.11: icmp_seq=3 ttl=64 time=0.037 ms
64 bytes from 172.16.240.11: icmp_seq=4 ttl=64 time=0.047 ms
64 bytes from 172.16.240.11: icmp_seq=5 ttl=64 time=0.040 ms
64 bytes from 172.16.240.11: icmp_seq=6 ttl=64 time=0.039 ms
64 bytes from 172.16.240.11: icmp_seq=7 ttl=64 time=0.040 ms
--- 172.16.240.11 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6000ms
rtt min/avg/max/mdev = 0.037/0.041/0.047/0.004 ms
[root@localhost ~]#
```

This completes the IPoIB configuration a Linux host using the Cisco OFED drivers.

5.1.5 Configuring SRP storage on Linux

For information about how to configure SRP attached storage using the Cisco SFS 3012 Multifabric switch, refer to Chapter 6, “Boot from InfiniBand using the Cisco 3012 InfiniBand to FC gateway” on page 95.

5.2 Configuring the IPoIB on a Windows host

The follow sections cover the procedure to install the InfiniBand host drivers on an HS21 blade with Windows Server 2003 x64.

5.2.1 Installing Windows host drivers

Prior to beginning this process, you might need to download the Cisco InfiniBand host drivers for Windows, which is available at:

```
http://www.cisco.com/pcgi-bin/tablebuild.pl?topic=280511613
```

To install drivers after you install the HCA, perform the following steps:

1. Log in to your Windows Server 2003 host. Ensure that the InfiniBand HCA (Host Channel Adapter) is installed on the blade from the Windows Device Manager as shown in Figure 5-2. Notice that Windows could not recognize a PCI device because the InfiniBand host driver is not yet installed on the system.

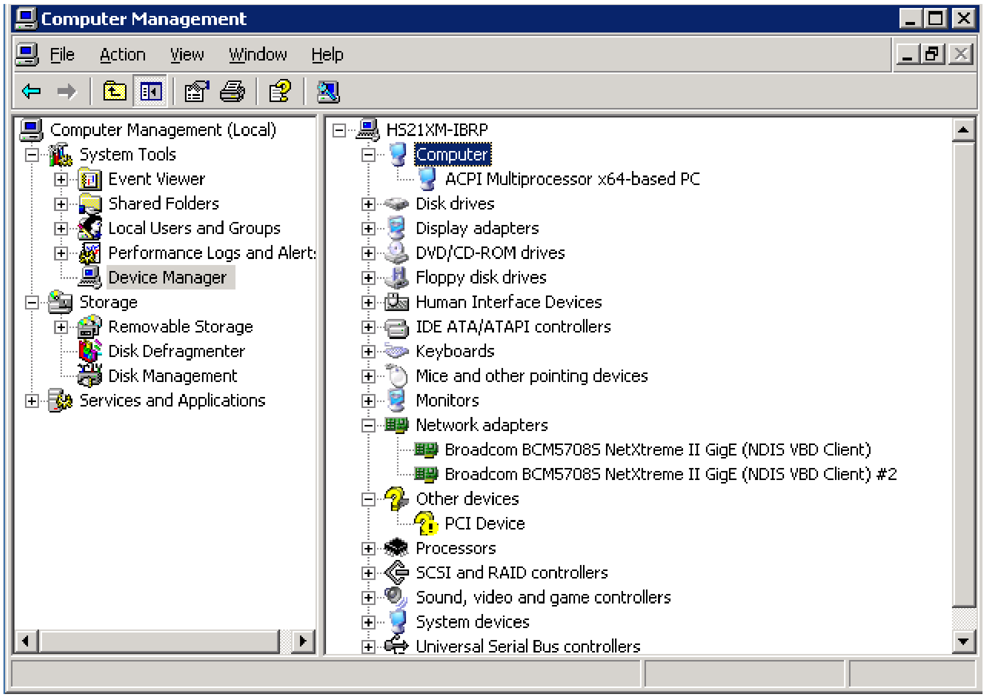
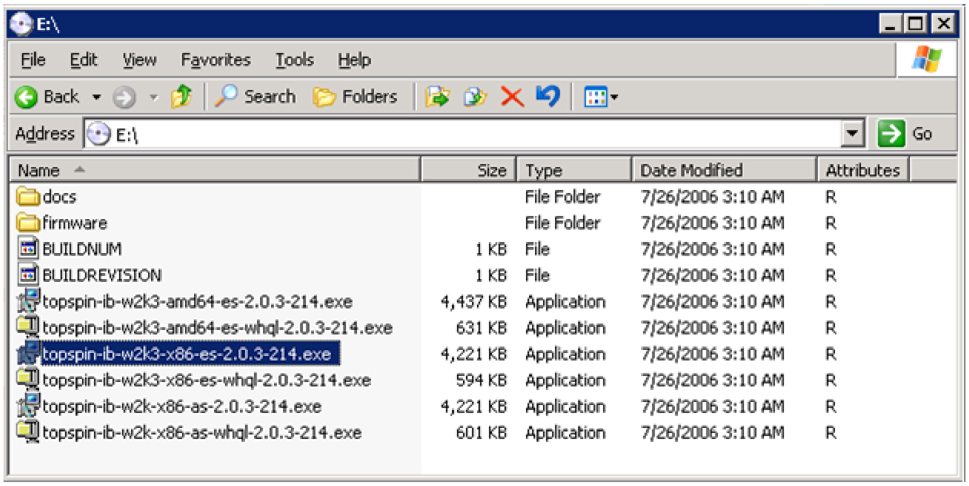

Computer Management
File
Action
View
Window
Help
Computer Management (Local)
System Tools
Event Viewer
Shared Folders
Local Users and Groups
Performance Logs and Alert:
Device Manager
Storage
Removable Storage
Disk Defragmenter
Disk Management
Services and Applications
HS21XM-IBRP
Computer
ACPI Multiprocessor x64-based PC
Disk drives
Display adapters
DVD/CD-ROM drives
Floppy disk drives
Human Interface Devices
IDE ATA/ATAPI controllers
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Broadcom BCM5708S NetXtreme II GigE (NDIS VBD Client)
Broadcom BCM5708S NetXtreme II GigE (NDIS VBD Client) #2
Other devices
PCI Device
Processors
SCSI and RAID controllers
Sound, video and game controllers
System devices
Universal Serial Bus controllers

Figure 5-2 Verify InfiniBand HCA installation on Windows Host

2. Insert the InfiniBand host driver CD or download the Windows Server 2003 drivers on the blade and verify the CD contents as shown in Figure 5-3.



Name	Size	Type	Date Modified	Attributes
docs		File Folder	7/26/2006 3:10 AM	R
firmware		File Folder	7/26/2006 3:10 AM	R
BUILDNUM	1 KB	File	7/26/2006 3:10 AM	R
BUILDREVISION	1 KB	File	7/26/2006 3:10 AM	R
topspin-ib-w2k3-amd64-es-2.0.3-214.exe	4,437 KB	Application	7/26/2006 3:10 AM	R
topspin-ib-w2k3-amd64-es-whql-2.0.3-214.exe	631 KB	Application	7/26/2006 3:10 AM	R
topspin-ib-w2k3-x86-es-2.0.3-214.exe	4,221 KB	Application	7/26/2006 3:10 AM	R
topspin-ib-w2k3-x86-es-whql-2.0.3-214.exe	594 KB	Application	7/26/2006 3:10 AM	R
topspin-ib-w2k-x86-as-2.0.3-214.exe	4,221 KB	Application	7/26/2006 3:10 AM	R
topspin-ib-w2k-x86-as-whql-2.0.3-214.exe	601 KB	Application	7/26/2006 3:10 AM	R

Figure 5-3 Verify InfiniBand Host driver CD contents

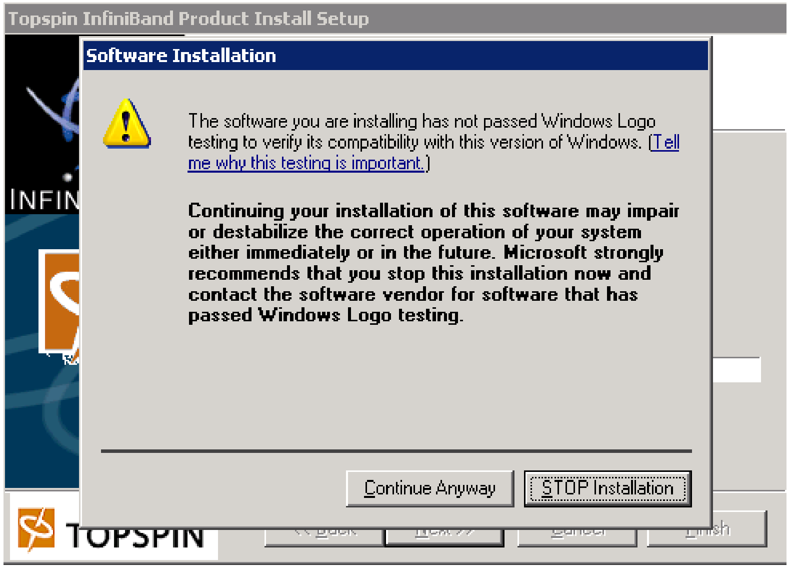
3. Run the topspin-ib-W2k3-x86-es-2.0.3-214.exe file as shown in Figure 5-3.

4. Begin the installation. The Product Install window displays. Click **Next** to proceed with installation.

5. Read the license and select **I Agree to the terms listed above**. Then click **Next** to continue.

6. Review the subsequent windows and click **Next** to continue each time.

7. During the installation, you will see an alert indicating the driver has not passed Windows Logo testing as shown in Figure 5-4. Click **Continue Anyway**.



The software you are installing has not passed Windows Logo testing to verify its compatibility with this version of Windows. (Tell me why this testing is important.)

Continuing your installation of this software may impair or destabilize the correct operation of your system either immediately or in the future. Microsoft strongly recommends that you stop this installation now and contact the software vendor for software that has passed Windows Logo testing.

Figure 5-4 InfiniBand Windows Host Driver Installation process

8. Click **Finish** when the installation has completed.

5.2.2 Verifying the installation

Follow these steps to verify the installation:

1. To verify the driver installation, log in to your host and check whether the Topspin InfiniBand SDK menu is listed under All Programs as shown in Figure 5-5.

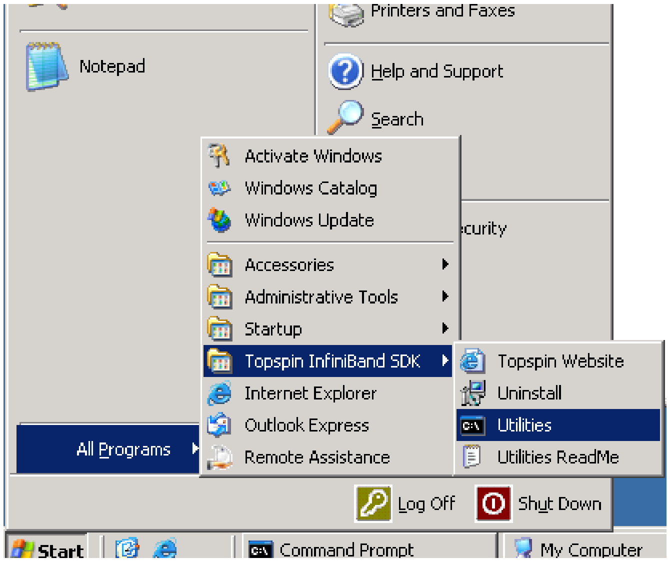


Figure 5-5 Verify InfiniBand Windows Host Driver menu under Program Files

2. Open Device Manager to verify that the InfiniBand HCA is listed as shown in Figure 5-6.

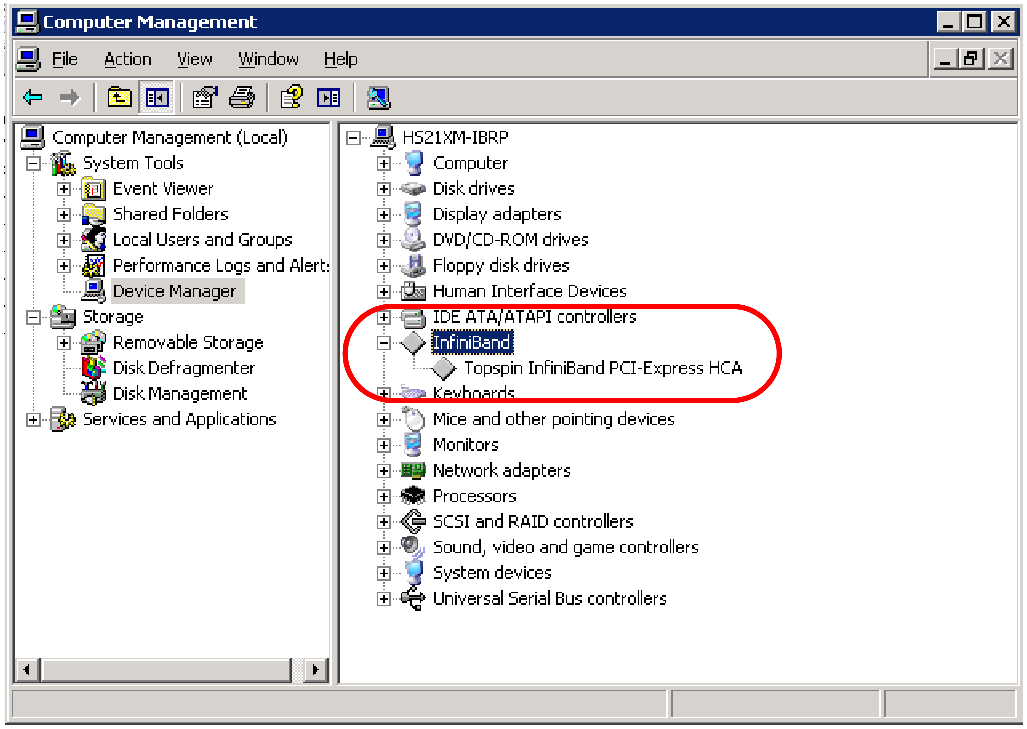


Figure 5-6 Verify InfiniBand Host Channel Adapter from the Device Manager menu

3. Verify that a Topspin IPoIB Virtual Channel Adapters for each port on the blade server are listed in the Network Connections menu as shown in Figure 5-7.

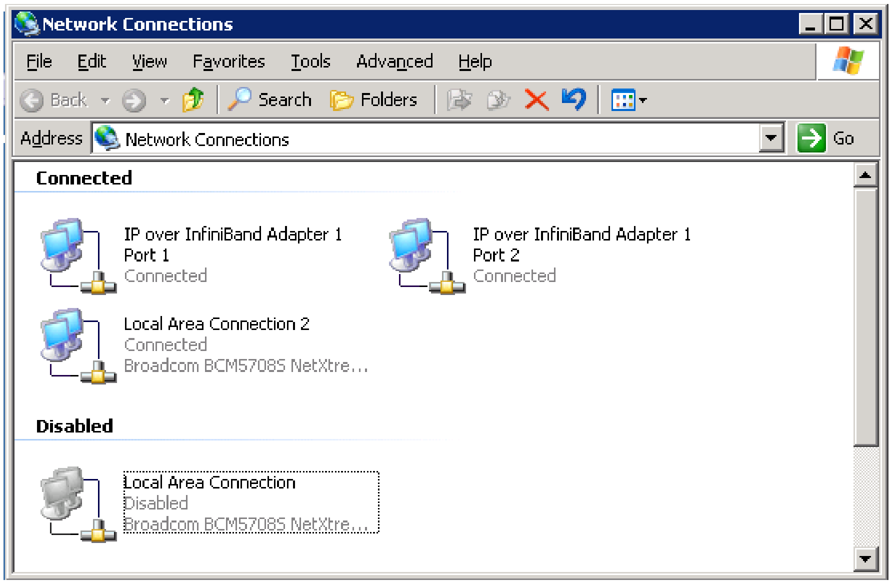

Network Connections
File Edit View Favorites Tools Advanced Help
Back Search Folders
Address Network Connections
Go
Connected
IP over InfiniBand Adapter 1
Port 1
Connected
IP over InfiniBand Adapter 1
Port 2
Connected
Local Area Connection 2
Connected
Broadcom BCM5708S NetXtre...
Disabled
Local Area Connection
Disabled
Broadcom BCM5708S NetXtre...

Figure 5-7 Verify IP over InfiniBand virtual interfaces from Network Connections menu

5.2.3 Configuring IPoIB

Now that the required drivers are installed and configured, the next step is to configure IPoIB. The IPoIB driver is automatically initialized when the installation is successfully completed.

The IPoIB configuration process on a Windows host is very much similar to configuring IP on an Ethernet interface network. You will assign an IP address, subnet mask and default gateway on desired IPoIB interfaces (in most cases, only one interface will have a default gateway assigned), using the Network connections menu. The first port on the HCA becomes interface ib0 and the second port on the same HCA is interface ib1 on the blade server.

Complete the following steps:

1. From the Windows Network Connections menu, right-click one of the IPoIB interfaces and select **Properties** as shown in Figure 5-8.

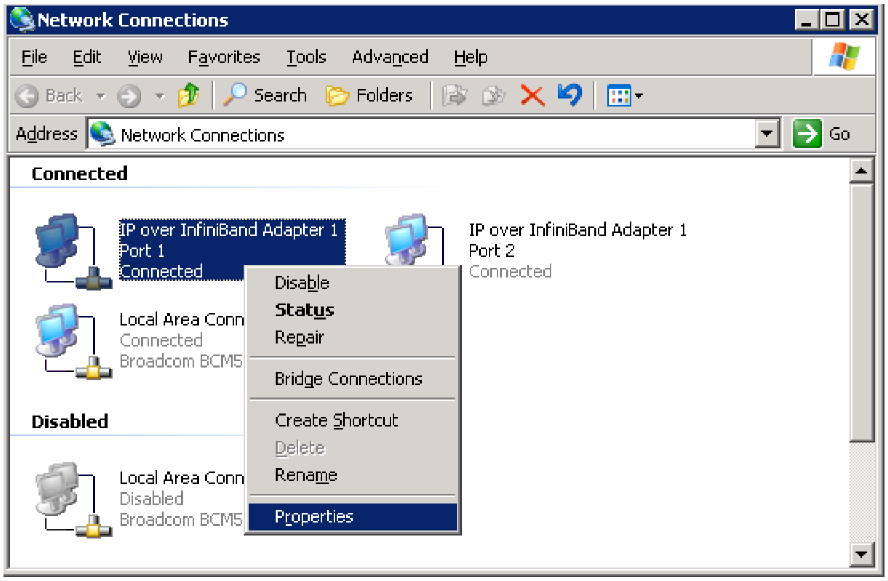
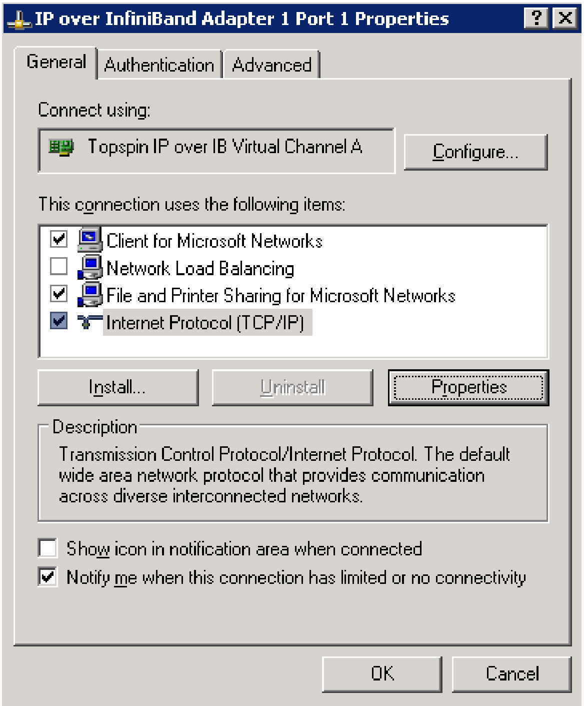

Network Connections
File Edit View Favorites Tools Advanced Help
Back Search Folders
Address Network Connections
Go
Connected
IP over InfiniBand Adapter 1 Port 1 Connected
IP over InfiniBand Adapter 1 Port 2 Connected
Local Area Conn Connected Broadcom BCM5
Disabled
Local Area Conn Disabled Broadcom BCM5
Disable
Status
Repair
Bridge Connections
Create Shortcut
Delete
Rename
Properties

Figure 5-8 configure IPoIB Interface, select Properties

2. Select **Internet Protocol (TCP/IP)** and then click **Properties** as shown in Figure 5-9.


IP over InfiniBand Adapter 1 Port 1 Properties
General Authentication Advanced
Configure...
Install... Uninstall Properties
OK Cancel

Connect using:

Topspin IP over IB Virtual Channel A

This connection uses the following items:

☑ Client for Microsoft Networks
□ Network Load Balancing
☑ File and Printer Sharing for Microsoft Networks
☑ Internet Protocol (TCP/IP)

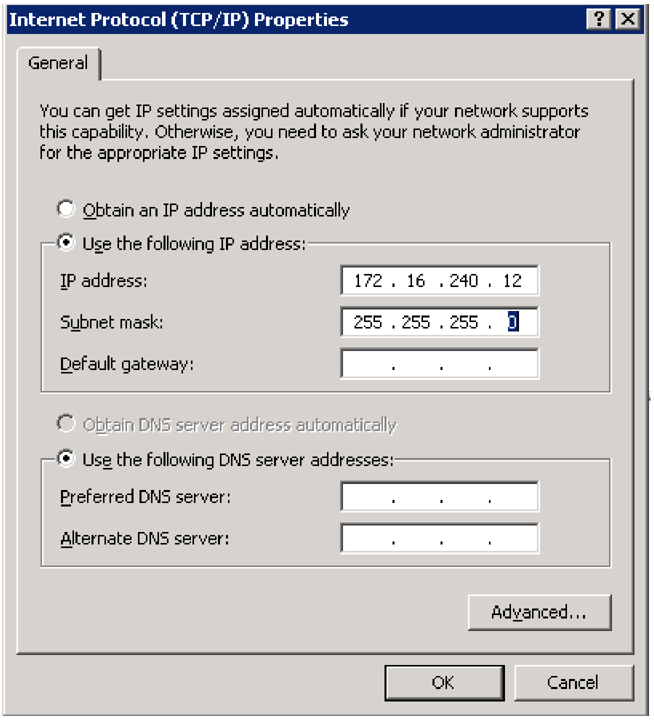
Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

□ Show icon in notification area when connected
☑ Notify me when this connection has limited or no connectivity

Figure 5-9 configure IPoIB Interface, select Internet Protocol (TCP/IP) properties option

3. Enter an IP address, subnet mask, and optionally a default gateway. Then, click **OK** as shown in Figure 5-10.



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

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 172 . 16 . 240 . 12

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

Figure 5-10 Internet Protocol TCP/IP Properties menu

4. Click **OK** to apply the TCP/IP configuration.
5. Upon successful binding of IP address and subnet mask information to the interface ib0, verify successful initialization of the ib0 interface from the command prompt using the command **`ipconfig`**, as shown in Figure 5-11.

```
Windows IP Configuration

Ethernet adapter Local Area Connection 2:

        Connection-specific DNS Suffix  . : raleigh.ibm.com
        IP Address. . . . . . . . . . . . : 9.42.166.84
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 9.42.166.1

Ethernet adapter IP over InfiniBand Adapter 1 Port 2:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 169.254.31.5
        Subnet Mask . . . . . . . . . . . : 255.255.0.0
        Default Gateway . . . . . . . . . :

Ethernet adapter IP over InfiniBand Adapter 1 Port 1:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 172.16.240.12
        Subnet Mask . . . . . . . . . . . : 255.255.0.0
        Default Gateway . . . . . . . . . :
```

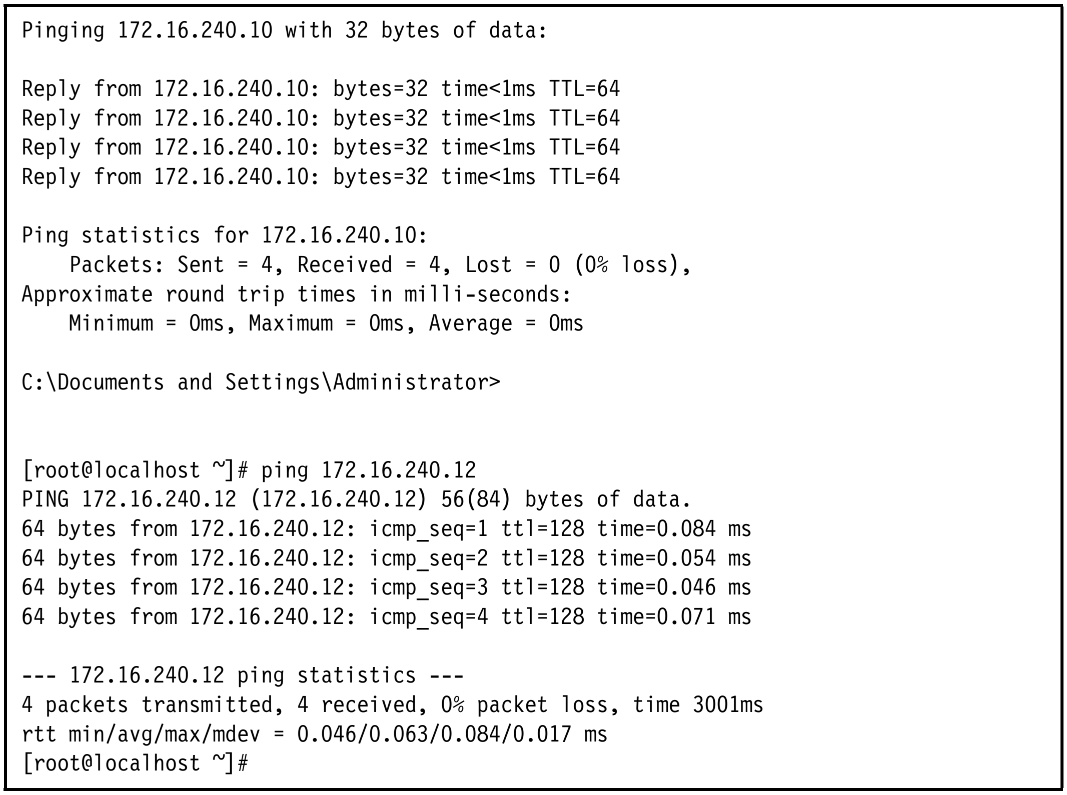
Figure 5-11 TCP/IP configuration for IPoIB Interface

This completes the IPoIB configuration process.

5.2.4 Verifying the IPoIB communication

To verify IPoIB functionality on Linux hosts:

1. Login in to the hosts and ensure that the IPoIB interfaces on the source and destination nodes are up by using the **`ifconfig`** command specified in Example 5-6 on page 86.
2. Issue a **`ping`** command between a source node to a destination node to ensure successful IPoIB communication as shown in Figure 5-12.



```
Pinging 172.16.240.10 with 32 bytes of data:

Reply from 172.16.240.10: bytes=32 time<1ms TTL=64
Reply from 172.16.240.10: bytes=32 time<1ms TTL=64
Reply from 172.16.240.10: bytes=32 time<1ms TTL=64
Reply from 172.16.240.10: bytes=32 time<1ms TTL=64

Ping statistics for 172.16.240.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Documents and Settings\Administrator>


[root@localhost ~]# ping 172.16.240.12
PING 172.16.240.12 (172.16.240.12) 56(84) bytes of data.
64 bytes from 172.16.240.12: icmp_seq=1 ttl=128 time=0.084 ms
64 bytes from 172.16.240.12: icmp_seq=2 ttl=128 time=0.054 ms
64 bytes from 172.16.240.12: icmp_seq=3 ttl=128 time=0.046 ms
64 bytes from 172.16.240.12: icmp_seq=4 ttl=128 time=0.071 ms

--- 172.16.240.12 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 0.046/0.063/0.084/0.017 ms
[root@localhost ~]#
```

Figure 5-12 Verify IPoIB communication across the blades

To test a Windows host, use the same steps, except instead of using **`ifconfig`** to verify the ib0/1 interfaces are present and configured, use the **`ipconfig`** command.

6

Boot from InfiniBand using the Cisco 3012 InfiniBand to FC gateway

This chapter describes how to configure booting from external storage attached through a high-speed InfiniBand fabric. Boot from InfiniBand (BoIB) allows diskless servers to mount the boot device located on an external RAID-capable storage subsystem attached to Fibre Channel SAN.

When a blade server with 4x InfiniBand HCA is initialized, it executes the BoIB firmware. The firmware on the HCA then communicates with the switch server to determine which boot LUN to request. Upon receiving the information from the switch server, the HCA performs the functions of a standard SCSI initiator and reads the boot sector from remote Fibre Channel attached storage. The remainder of the operating system loads as though the storage were locally attached storage.

The topics that we discuss in this chapter are:

- 6.1, “Benefits of booting from high-speed InfiniBand” on page 96
- 6.2, “Blade booting from a SAN” on page 96
- 6.3, “Installing the OS on the Fibre Channel storage” on page 110

6.1 Benefits of booting from high-speed InfiniBand

The BoIB feature on the IBM BladeCenter blade servers provides high availability, reduces cabling complexity, minimizes the downtime windows and helps consolidate the IT infrastructure. The BoIB feature allows for deployment of diskless servers and exploit the10 Gbps bandwidth available with BC-H solution.

The benefits include:

- Cost reduction

  The BladeCenter architecture helps cut the equipment costs by reducing the number of moving parts such as fans, power supply, and internal disk drives when compared with the standalone servers. BladeCenter provides the capability to share the resources such as power, cooling, cabling, IO paths among all the servers with in the chassis and facilitates centralized management.

- High availability

  The system downtime is greatly minimized in situations where a critical component such as a processor, memory or the system planar fails and needs to be replaced. The system administrator only needs to swap the hardware with similar hardware and install the same HCA (InfiniBand host channel adapter) on the blade server thus making the system quickly available for production.

- Centralized storage management

  The solution eliminates any need for local storage. The boot LUN is located on a RAID capable storage subsystem thus providing high availability in case of a failure of one of the physical disks.

6.2 Blade booting from a SAN

The following sections illustrates the step-by-step procedure for implementing boot from SAN using the HS21 blade server attached DS4700 storage over high-speed InfiniBand fabric using the SFS3012 FC Gateway device.

The sections that we cover here are:

- 6.2.1, “Pre-configuration checklist” on page 97
- 6.2.2, “Verifying and installing HCA firmware” on page 97
- 6.2.3, “Configuring the SRP host on the SFS 3012 FC gateway” on page 98
- 6.2.4, “Configuring Fibre Channel SAN” on page 102
- 6.2.5, “Configuring storage” on page 104
- 6.2.6, “Discovering storage” on page 105
- 6.2.7, “Configuring the gateway port and LUN access” on page 105
- 6.2.8, “Port masking” on page 106
- 6.2.9, “Discovering the boot LUN” on page 109

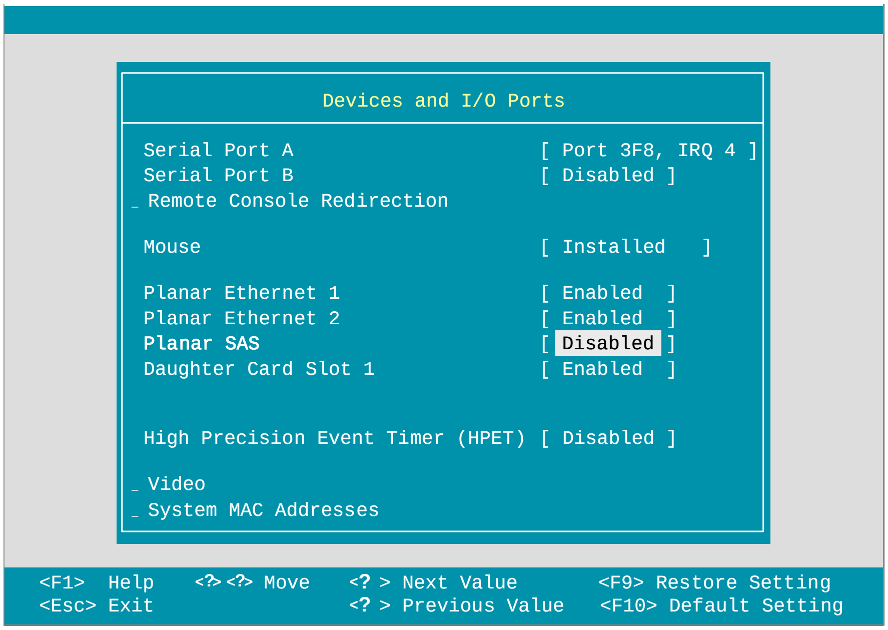
6.2.1 Pre-configuration checklist

Before proceeding to implement the function, ensure that the following tasks are completed.

Installing hardware

To set up the physical environment for your SAN boot, perform the following high-level steps:

1. Install the HCA expansion card in the blade server.
2. Install the high-speed InfiniBand switch into bay 7 of the BladeCenter H chassis.
3. Connect the high-speed InfiniBand switch in bay 7 to the 4X InfiniBand port on the SFS3012 server switch.
4. Install the Fibre Channel gateway into the SFS3012 Server Switch. For details, refer to the Fibre Channel Gateway User Guide.
5. Connect the Fibre Channel gateway to a SAN fabric with a FC storage device.
6. Connect the Fibre Channel storage subsystem to the SAN.
7. Verify and confirm from the SAN switch CLI or GUI interface that the SFS3012 FC gateway ports and DS4700 port are online.
8. Disable the onboard SAS planar in BIOS of the HS21 by going to the Devices and I/O Ports menu and set Planar SAS to *Disable* as shown in Figure 6-1.



```
                    Devices and I/O Ports

 Serial Port A                         [ Port 3F8, IRQ 4 ]
 Serial Port B                         [ Disabled ]
_ Remote Console Redirection

 Mouse                                 [ Installed   ]

 Planar Ethernet 1                     [ Enabled  ]
 Planar Ethernet 2                     [ Enabled  ]
 Planar SAS                            [ Disabled ]
 Daughter Card Slot 1                  [ Enabled  ]


 High Precision Event Timer (HPET)     [ Disabled ]

_ Video
_ System MAC Addresses

<F1>  Help    <?><?> Move    <? > Next Value      <F9> Restore Setting
<Esc> Exit                   <? > Previous Value  <F10> Default Setting
```

Figure 6-1 Disable Planar SAS

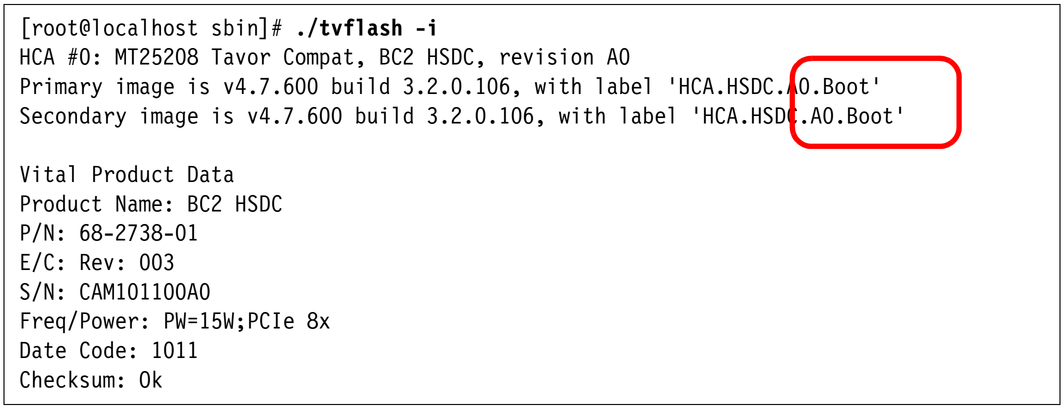
6.2.2 Verifying and installing HCA firmware

Verify whether Boot over InfiniBand firmware is installed on the HCA in either of the following ways:

- From the CLI prompt on the operating system, enter the **`/usr/local/topspin/sbin tvflash -i`** as shown in Figure 6-2 on page 98.

> **Note:** This method can only be performed if the Linux OS and InfiniBand host drivers are installed on the local SAS disk on the HS21 or LS21 blade.

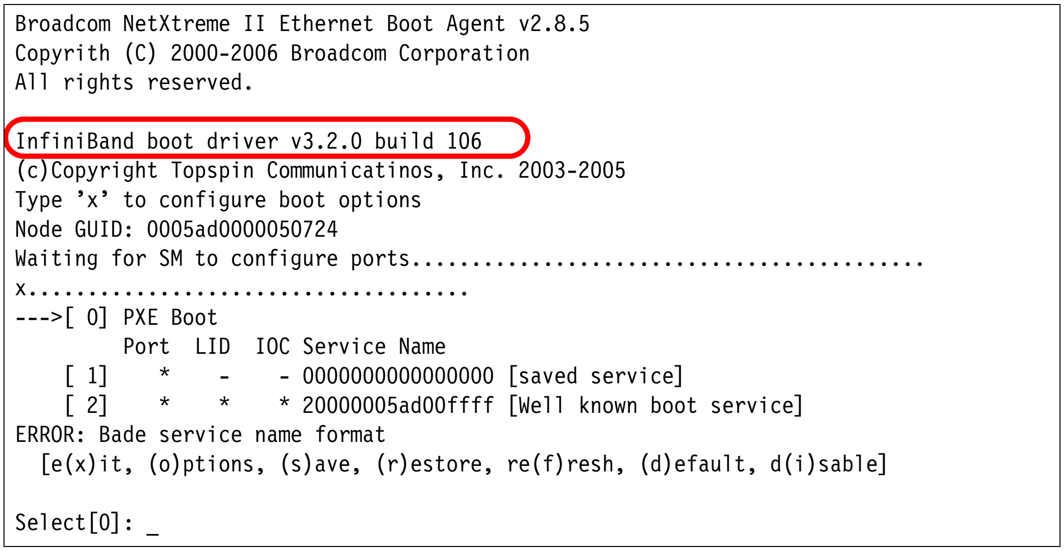
Look for the *.Boot* extension in the description as shown in Figure 6-2 to confirm that Boot over InfiniBand is enabled.



```
[root@localhost sbin]# ./tvflash -i
HCA #0: MT25208 Tavor Compat, BC2 HSDC, revision A0
Primary image is v4.7.600 build 3.2.0.106, with label 'HCA.HSDC.A0.Boot'
Secondary image is v4.7.600 build 3.2.0.106, with label 'HCA.HSDC.A0.Boot'

Vital Product Data
Product Name: BC2 HSDC
P/N: 68-2738-01
E/C: Rev: 003
S/N: CAM101100A0
Freq/Power: PW=15W;PCIe 8x
Date Code: 1011
Checksum: Ok
```

Figure 6-2 Verify Boot over InfiniBand Firmware installed on the HCA

- Verify that the firmware is installed by booting the blade into the InfiniBand firmware menu as shown in Figure 6-3.



```
Broadcom NetXtreme II Ethernet Boot Agent v2.8.5
Copyrith (C) 2000-2006 Broadcom Corporation
All rights reserved.

InfiniBand boot driver v3.2.0 build 106
(c)Copyright Topspin Communicatinos, Inc. 2003-2005
Type ’x’ to configure boot options
Node GUID: 0005ad0000050724
Waiting for SM to configure ports.......................................
x...................................
--->[ 0] PXE Boot
         Port  LID  IOC Service Name
    [ 1]    *    -    - 0000000000000000 [saved service]
    [ 2]    *    *    * 20000005ad00ffff [Well known boot service]
ERROR: Bade service name format
  [e(x)it, (o)ptions, (s)ave, (r)estore, re(f)resh, (d)efault, d(i)sable]

Select[0]: _
```

Figure 6-3 Record the GUID from the InfiniBand Firmware menu

6.2.3 Configuring the SRP host on the SFS 3012 FC gateway

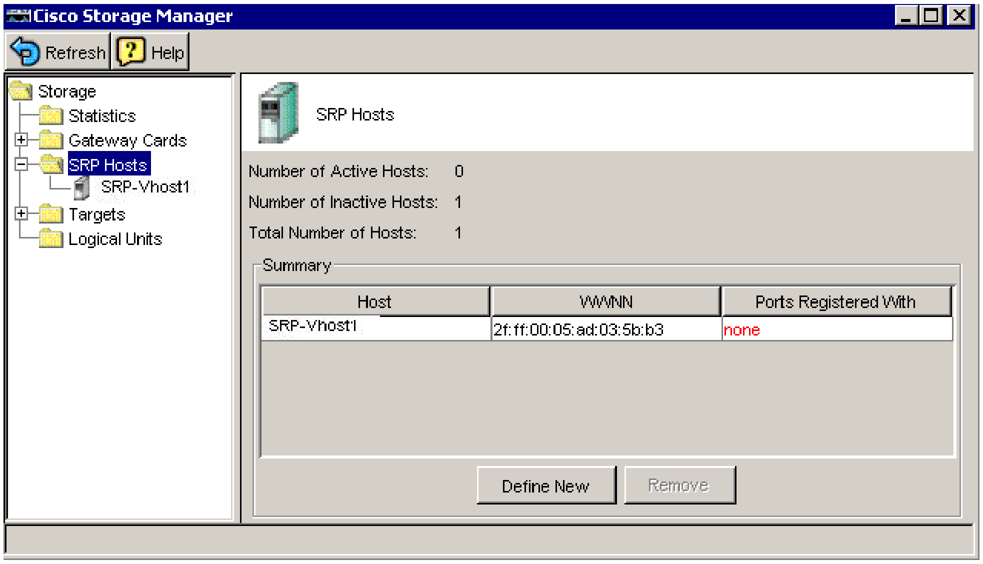
This section illustrates the procedure to configure an SRP host from the SFS3012 FC gateway device. The SRP host configuration enables it to login to the fabric and zoned with the corresponding target devices.

1. Initialize the InfiniBand port of the blade by booting the blade into the InfiniBand firmware menu and pressing the **x** key when the following prompt is displayed during the boot process (see Figure 6-3):

   ```
   Type ‘x’ to configure boot options
   ```

   The GUID of the host shows in the host display. Record the GUID. Keep the InfiniBand firmware menu active to configure SRP host in the following steps. When the SRP host is configured, port masking is completed and Boot LUN is mapped return to this menu and refresh the screen.

2. From the Element Manager UI, select **Fibre Channel** → **Storage Manager**. Select the **SRP Hosts** folder in the left-hand pane as shown in Figure 6-4.



Number of Active Hosts: 0

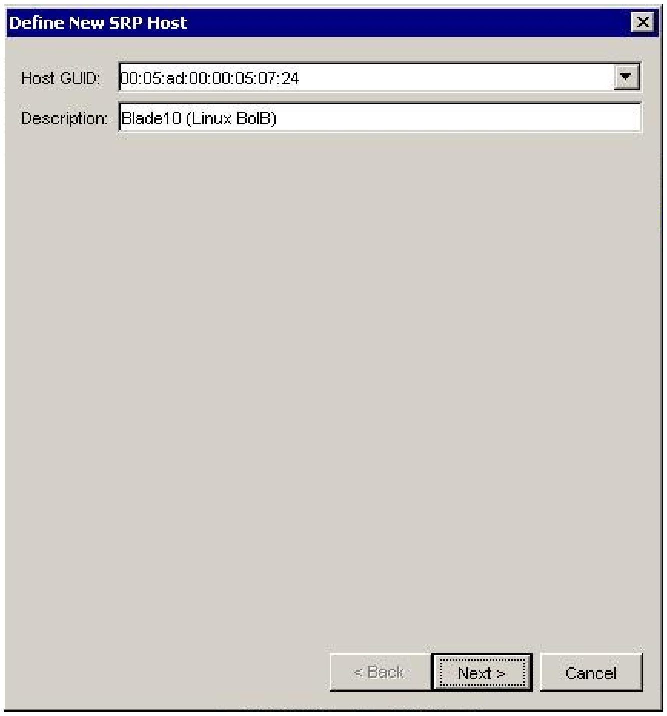
Number of Inactive Hosts: 1

Total Number of Hosts: 1

Host	WWNN	Ports Registered With
SRP-Vhost1	2f:ff:00:05:ad:03:5b:b3	none

Figure 6-4 Define a New SRP Host from the TS3012 Storage Manager menu

3. Click **Define New**. The define New SRP Host Windows displays as shown in Figure 6-5.



Host GUID: 00:05:ad:00:00:05:07:24

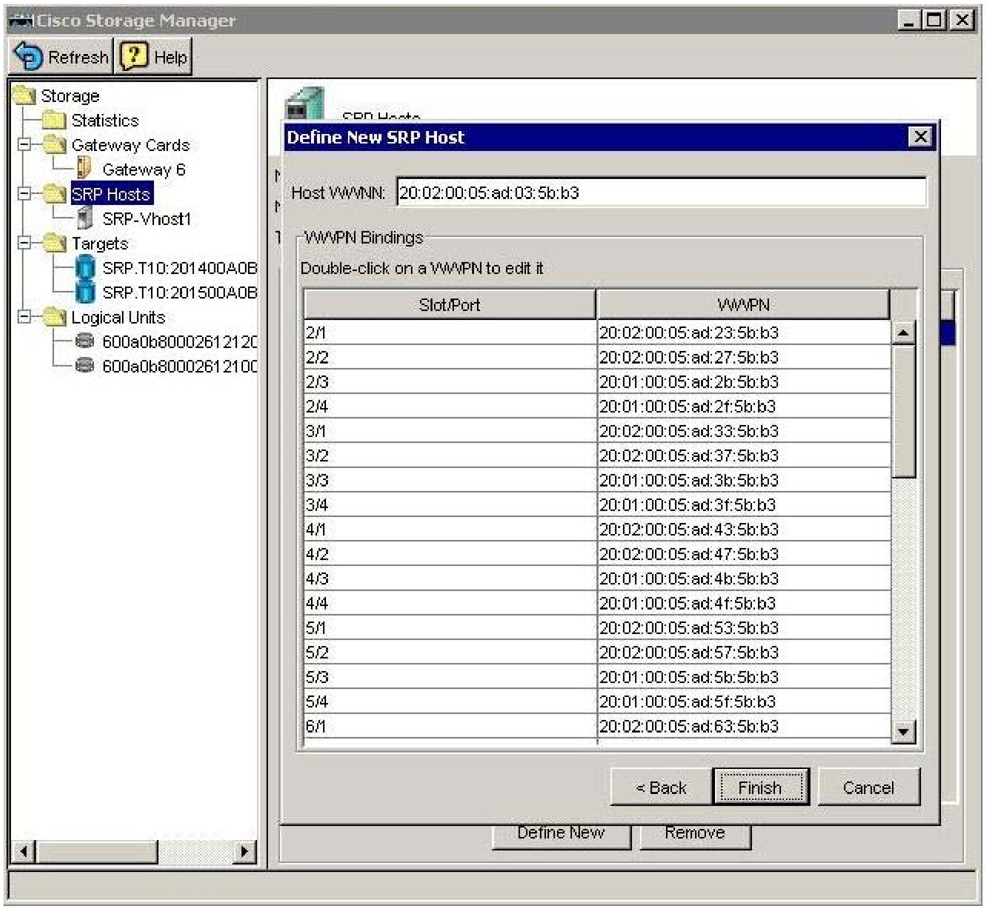
Description: Blade10 (Linux BoIB)

Figure 6-5 Select the host GUID from the drop-down list

4. From the drop-down menu, select the GUID of the blade that you recorded in step 1 on page 98 as shown in Figure 6-5.

5. In the description field, you can specify some text that identifies the SRP host. In Figure 6-5, the description *Blade10 (Linux BoIB)* means that this blade is in slot 10 in the BladeCenter H chassis and boot over InfiniBand with Linux.

6. Click **Next**. The WWNN is assigned to the SRP host shown in Figure 6-6.



Host WWNN: 20:02:00:05:ad:03:5b:b3

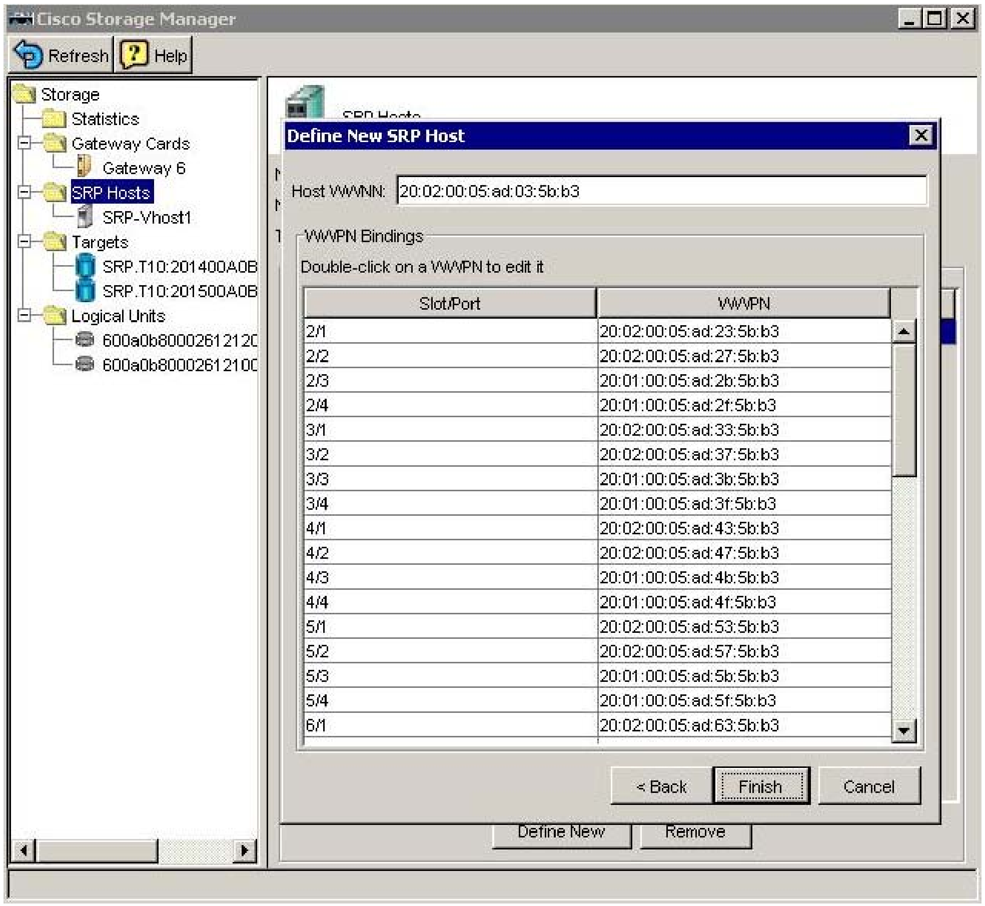
WWPN Bindings

Double-click on a WWPN to edit it

Slot/Port	WWPN
2/1	20:02:00:05:ad:23:5b:b3
2/2	20:02:00:05:ad:27:5b:b3
2/3	20:01:00:05:ad:2b:5b:b3
2/4	20:01:00:05:ad:2f:5b:b3
3/1	20:02:00:05:ad:33:5b:b3
3/2	20:02:00:05:ad:37:5b:b3
3/3	20:01:00:05:ad:3b:5b:b3
3/4	20:01:00:05:ad:3f:5b:b3
4/1	20:02:00:05:ad:43:5b:b3
4/2	20:02:00:05:ad:47:5b:b3
4/3	20:01:00:05:ad:4b:5b:b3
4/4	20:01:00:05:ad:4f:5b:b3
5/1	20:02:00:05:ad:53:5b:b3
5/2	20:02:00:05:ad:57:5b:b3
5/3	20:01:00:05:ad:5b:5b:b3
5/4	20:01:00:05:ad:5f:5b:b3
6/1	20:02:00:05:ad:63:5b:b3

Figure 6-6 Define SRP host

7. Select **Finish** to complete the SRP host creation process as shown in Figure 6-7.



Define New SRP Host

Host WWNN: 20:02:00:05:ad:03:5b:b3

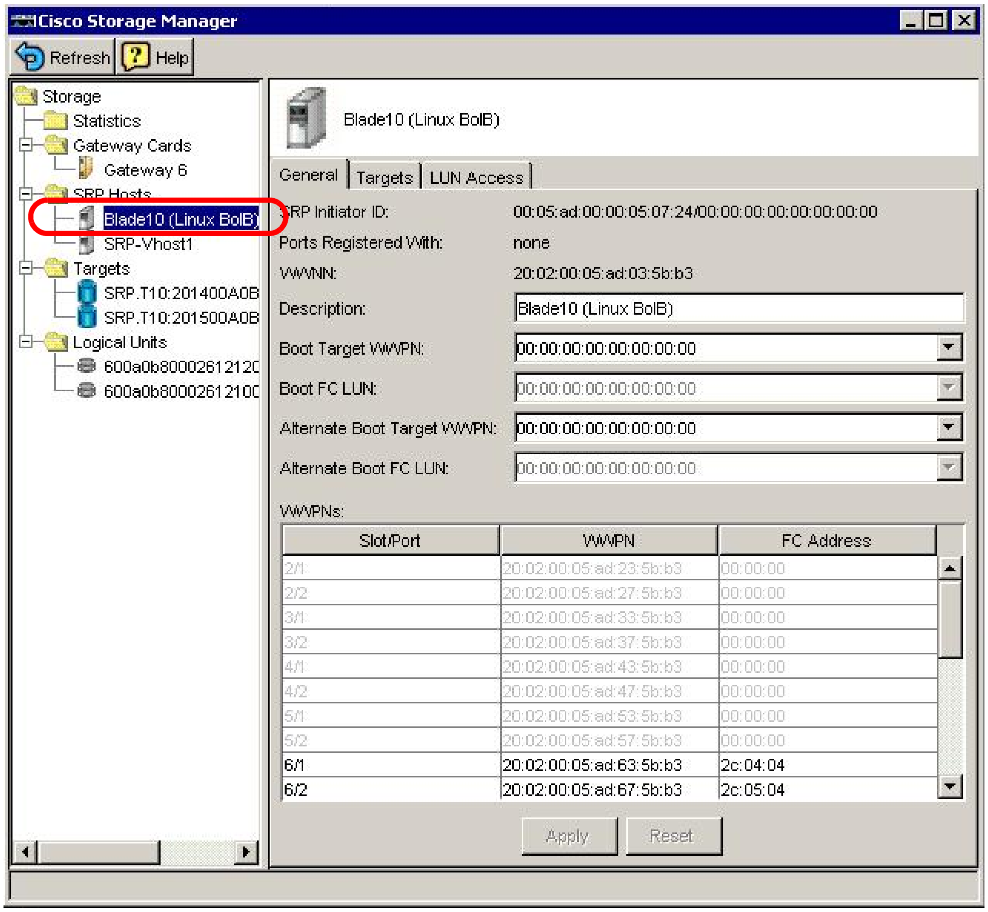
WWPN Bindings

Double-click on a WWPN to edit it

Slot/Port	WWPN
2/1	20:02:00:05:ad:23:5b:b3
2/2	20:02:00:05:ad:27:5b:b3
2/3	20:01:00:05:ad:2b:5b:b3
2/4	20:01:00:05:ad:2f:5b:b3
3/1	20:02:00:05:ad:33:5b:b3
3/2	20:02:00:05:ad:37:5b:b3
3/3	20:01:00:05:ad:3b:5b:b3
3/4	20:01:00:05:ad:3f:5b:b3
4/1	20:02:00:05:ad:43:5b:b3
4/2	20:02:00:05:ad:47:5b:b3
4/3	20:01:00:05:ad:4b:5b:b3
4/4	20:01:00:05:ad:4f:5b:b3
5/1	20:02:00:05:ad:53:5b:b3
5/2	20:02:00:05:ad:57:5b:b3
5/3	20:01:00:05:ad:5b:5b:b3
5/4	20:01:00:05:ad:5f:5b:b3
6/1	20:02:00:05:ad:63:5b:b3

Figure 6-7 Define New SRP Host menu

The new SRP host *Blade 10 (Linux BoIB)* that you just created is now listed under the SRP Hosts folder as shown in Figure 6-8.



SRP Initiator ID: 00:05:ad:00:00:05:07:24/00:00:00:00:00:00:00:00

Ports Registered With: none

WWNN: 20:02:00:05:ad:03:5b:b3

Description: Blade10 (Linux BoIB)

Boot Target WWPN: 00:00:00:00:00:00:00:00

Boot FC LUN: 00:00:00:00:00:00:00:00

Alternate Boot Target WWPN: 00:00:00:00:00:00:00:00

Alternate Boot FC LUN: 00:00:00:00:00:00:00:00

WWPNs:

Slot/Port	WWPN	FC Address
2/1	20:02:00:05:ad:23:5b:b3	00:00:00
2/2	20:02:00:05:ad:27:5b:b3	00:00:00
3/1	20:02:00:05:ad:33:5b:b3	00:00:00
3/2	20:02:00:05:ad:37:5b:b3	00:00:00
4/1	20:02:00:05:ad:43:5b:b3	00:00:00
4/2	20:02:00:05:ad:47:5b:b3	00:00:00
5/1	20:02:00:05:ad:53:5b:b3	00:00:00
5/2	20:02:00:05:ad:57:5b:b3	00:00:00
6/1	20:02:00:05:ad:63:5b:b3	2c:04:04
6/2	20:02:00:05:ad:67:5b:b3	2c:05:04

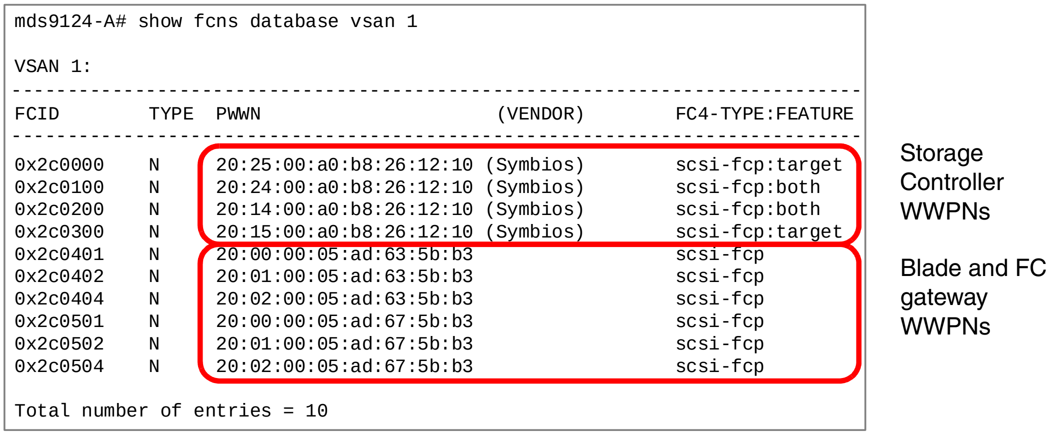
Figure 6-8 Define New SRP Host menu

6.2.4 Configuring Fibre Channel SAN

This section discusses the Fibre Channel SAN configuration process. The SFS3012 Gateway and the IBM System Storage DS4000™ storage subsystem is connected to a Cisco MDS9124 FC switch. For detailed procedure on implementing zoning refer to the *MDS9000 Switch User Guide*.

To configure Fibre Channel SAN, follow these steps:

1. Configure Fibre Channel SAN implies for end-to-end connectivity, the WWPNs of blade, storage, and Fibre Channel gateway are listed in the Name Server as shown in Figure 6-9.



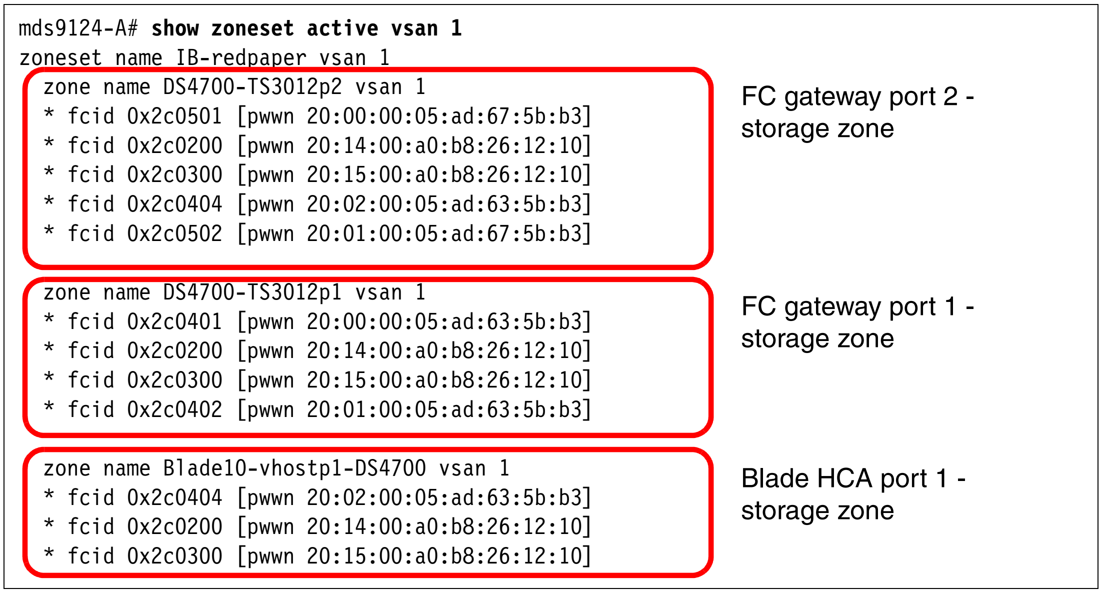
```
mds9124-A# show fcns database vsan 1

VSAN 1:
--------------------------------------------------------------------------
FCID        TYPE  PWWN                     (VENDOR)        FC4-TYPE:FEATURE
--------------------------------------------------------------------------
0x2c0000    N     20:25:00:a0:b8:26:12:10 (Symbios)       scsi-fcp:target
0x2c0100    N     20:24:00:a0:b8:26:12:10 (Symbios)       scsi-fcp:both
0x2c0200    N     20:14:00:a0:b8:26:12:10 (Symbios)       scsi-fcp:both
0x2c0300    N     20:15:00:a0:b8:26:12:10 (Symbios)       scsi-fcp:target
0x2c0401    N     20:00:00:05:ad:63:5b:b3                 scsi-fcp
0x2c0402    N     20:01:00:05:ad:63:5b:b3                 scsi-fcp
0x2c0404    N     20:02:00:05:ad:63:5b:b3                 scsi-fcp
0x2c0501    N     20:00:00:05:ad:67:5b:b3                 scsi-fcp
0x2c0502    N     20:01:00:05:ad:67:5b:b3                 scsi-fcp
0x2c0504    N     20:02:00:05:ad:67:5b:b3                 scsi-fcp

Total number of entries = 10
```

Figure 6-9 FC Zoning: Active Zoneset view

2. When all the WWPNs are found in the Name Server database as shown in Figure 6-9, configure the zones using the WWPNs (World Wide Port Names) that consist of the following members
   - Zone A: Fibre Channel Gateway port1 & Storage Controller Port(s)
   - Zone B: Fibre Channel Gateway port2 & Storage Controller Port(s)
   - Zone C: Blade HCA-port1 & Storage Controller Port(s)

> **Note:** For instructions about how to implement FC Zoning, refer to the FC switch User Guide installed in your environment.

Figure 6-10 lists the zoning implementation for this testing.



```
mds9124-A# show zoneset active vsan 1
zoneset name IB-redpaper vsan 1
  zone name DS4700-TS3012p2 vsan 1
  * fcid 0x2c0501 [pwwn 20:00:00:05:ad:67:5b:b3]
  * fcid 0x2c0200 [pwwn 20:14:00:a0:b8:26:12:10]
  * fcid 0x2c0300 [pwwn 20:15:00:a0:b8:26:12:10]
  * fcid 0x2c0404 [pwwn 20:02:00:05:ad:63:5b:b3]
  * fcid 0x2c0502 [pwwn 20:01:00:05:ad:67:5b:b3]

  zone name DS4700-TS3012p1 vsan 1
  * fcid 0x2c0401 [pwwn 20:00:00:05:ad:63:5b:b3]
  * fcid 0x2c0200 [pwwn 20:14:00:a0:b8:26:12:10]
  * fcid 0x2c0300 [pwwn 20:15:00:a0:b8:26:12:10]
  * fcid 0x2c0402 [pwwn 20:01:00:05:ad:63:5b:b3]

  zone name Blade10-vhostp1-DS4700 vsan 1
  * fcid 0x2c0404 [pwwn 20:02:00:05:ad:63:5b:b3]
  * fcid 0x2c0200 [pwwn 20:14:00:a0:b8:26:12:10]
  * fcid 0x2c0300 [pwwn 20:15:00:a0:b8:26:12:10]
```

FC gateway port 2 - storage zone
FC gateway port 1 - storage zone
Blade HCA port 1 - storage zone

Figure 6-10 Active Zoneset View from the MDS9000 switch CLI

6.2.5 Configuring storage

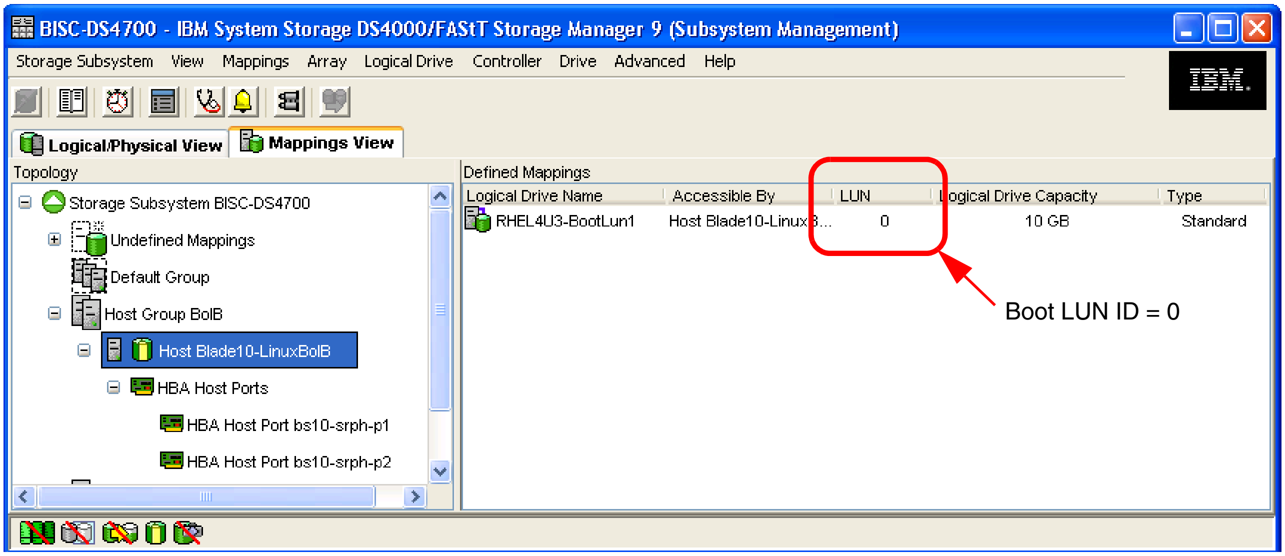
> **Tip:** The storage configuration procedure is different for different vendor products, but the concept of host definition and LUN masking is generally the same. For detail instructions about storage configuration, refer to the corresponding storage subsystem user guide.

To configure storage on the DS4000 storage subsystems, perform the following steps:

1. Define a Host Group.
2. Define a Host.
3. Define a Host Port.
4. Create a Logical Drive.
5. Map the Logical Drive.

> **Important:** The boot LUN should always be mapped with the `LUN ID = 0` because both Windows and Linux operating systems configures the first logical drive it sees as the root partition for Linux or C drive for Windows.

Figure 6-11 displays the boot LUN mapped with `LUN ID = 0` to the HS21 server defined as the host under Host Group BoIB.



Logical Drive Name	Accessible By	LUN	Logical Drive Capacity	Type
RHEL4U3-BootLun1	Host Blade10-LinuxB...	0	10 GB	Standard

Boot LUN ID = 0

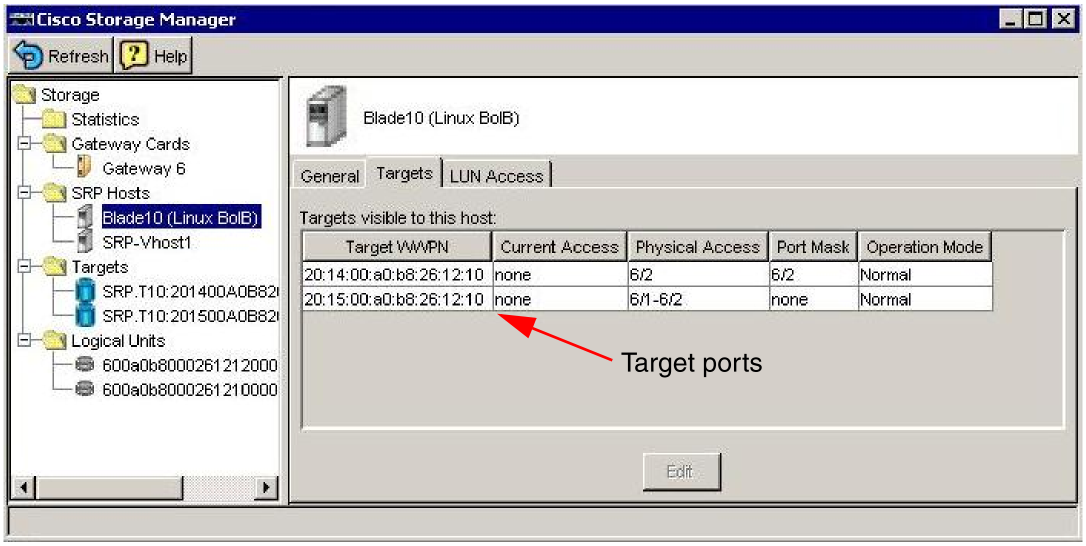
Figure 6-11 DS4000 Storage Manager: Mappings View list the boot LUN ID=0

This completes the Storage Configuration process for the Boot over InfiniBand.

6.2.6 Discovering storage

If the FC zoning, host definition and LUN masking are configured correctly, then the target ports and the boot LUN is discovered from the SFS3012 storage configuration menu.

From the Storage configuration, select the new initiator under the SRP Hosts folder (Blade10 in our example) and then go to the Targets tab. Verify that the WWPN of the target onto which you want to install the image displays in the list as shown in Figure 6-12.



Targets visible to this host:

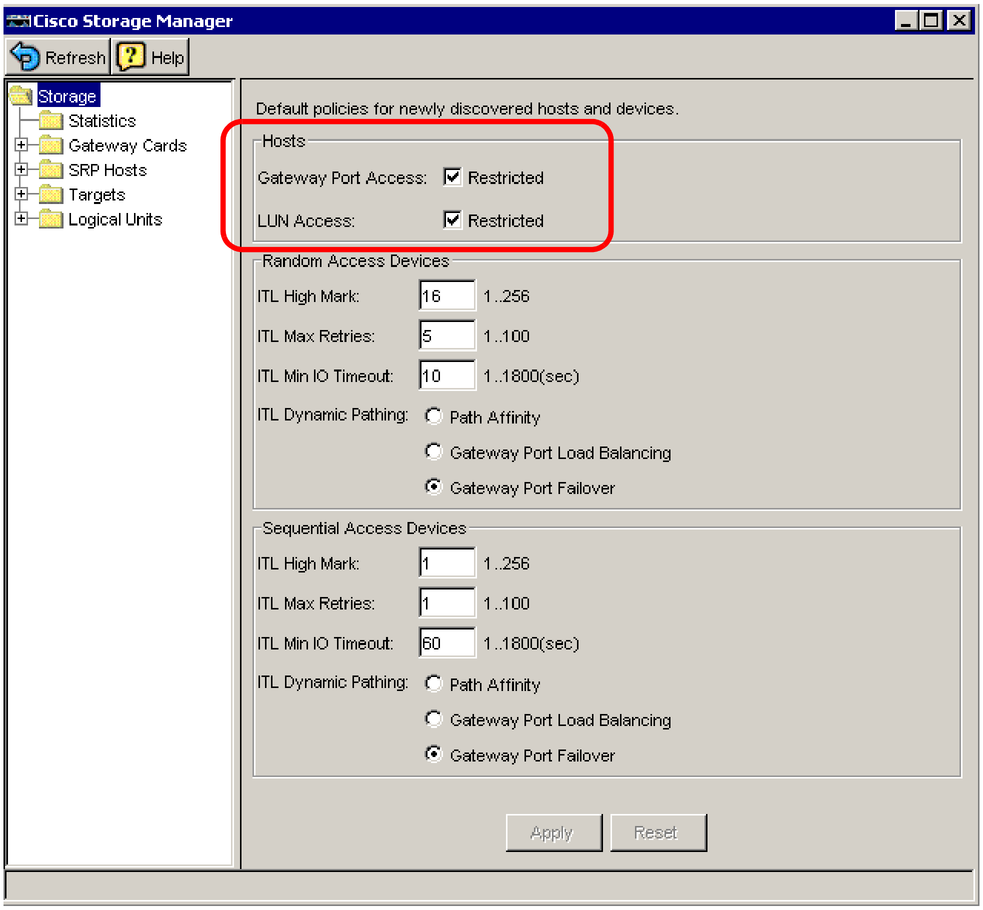
Target WWPN	Current Access	Physical Access	Port Mask	Operation Mode
20:14:00:a0:b8:26:12:10	none	6/2	6/2	Normal
20:15:00:a0:b8:26:12:10	none	6/1-6/2	none	Normal

Figure 6-12 View available targets for the SRP host

6.2.7 Configuring the gateway port and LUN access

From the SFS3012 Element Manager GUI, click **Fibre Channel** then click **Storage**. Select **Gateway Port Access** and **LUN Access**, as shown in Figure 6-13 on page 106 and click **Apply** to restrict access to all newly discovered hosts and devices. If the access is not restricted globally, only the first seven LUNs are available in the boot menu.

The Boot over InfiniBand host should have access only to its boot LUN. The initial operating system installation fails if there are multiple paths to the boot LUN and if the host sees more than one LUN during the operating system installation process. After the operating system is installed successfully, additional data LUNs and paths can be added to the host.



Default policies for newly discovered hosts and devices.

Hosts

Gateway Port Access: ☑ Restricted

LUN Access: ☑ Restricted

Random Access Devices

ITL High Mark: 16 1..256

ITL Max Retries: 5 1..100

ITL Min IO Timeout: 10 1..1800(sec)

ITL Dynamic Pathing: ○ Path Affinity ○ Gateway Port Load Balancing ● Gateway Port Failover

Sequential Access Devices

ITL High Mark: 1 1..256

ITL Max Retries: 1 1..100

ITL Min IO Timeout: 60 1..1800(sec)

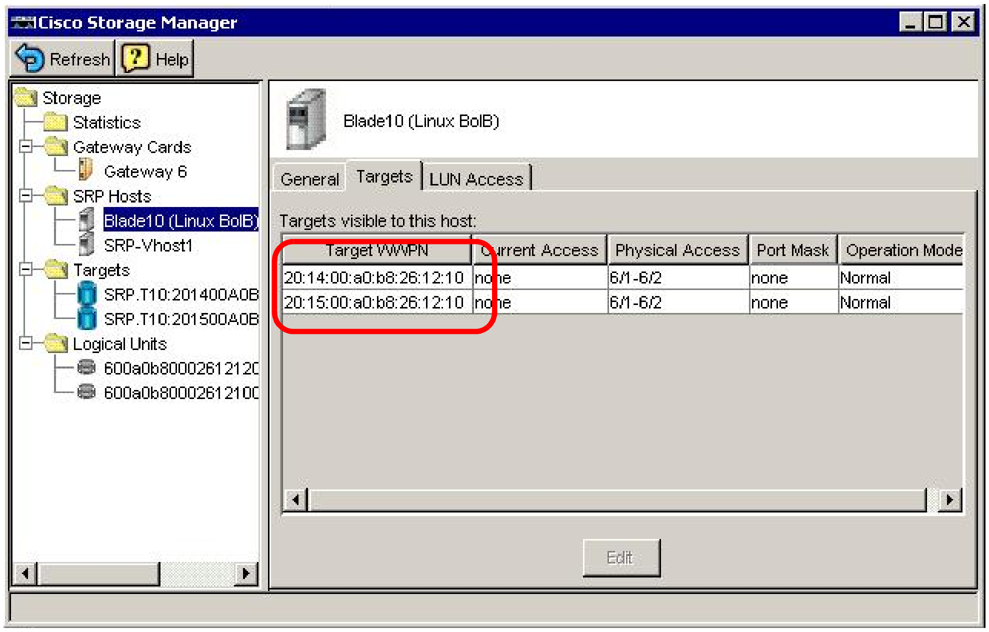
ITL Dynamic Pathing: ○ Path Affinity ○ Gateway Port Load Balancing ● Gateway Port Failover

Figure 6-13 Default host access policies

6.2.8 Port masking

This section covers port masking on the SFS3012 gateway. The port masking feature is used to permit or deny host access through the port. Follow these steps:

1. Select the SRP initiator listed in the SRP Hosts folder, then go to the Targets tab (Figure 6-14 on page 107).
2. Verify that the WWPN of the target controller or controllers are listed.



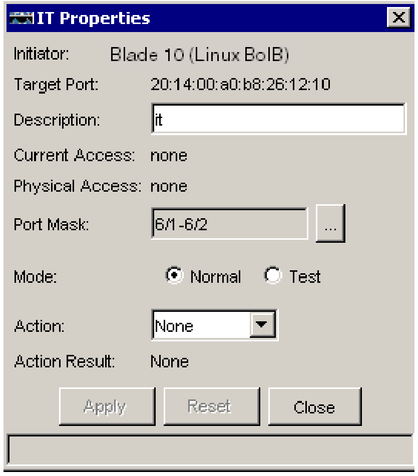
Blade10 (Linux BoIB)

Targets visible to this host:

Target WWPN	Current Access	Physical Access	Port Mask	Operation Mode
20:14:00:a0:b8:26:12:10	none	6/1-6/2	none	Normal
20:15:00:a0:b8:26:12:10	none	6/1-6/2	none	Normal

Figure 6-14   SRP Host: List target devices accessible

3. Double-click any of the target controller WWPN ports through which the BoIB image will be installed. The ITL Properties window opens as shown in Figure 6-15. Notice that the default value is to permit access through ports 6/1 and 6/2.



Initiator: Blade 10 (Linux BoIB)

Target Port: 20:14:00:a0:b8:26:12:10

Description: it

Current Access: none

Physical Access: none

Port Mask: 6/1-6/2

Mode: ● Normal ○ Test

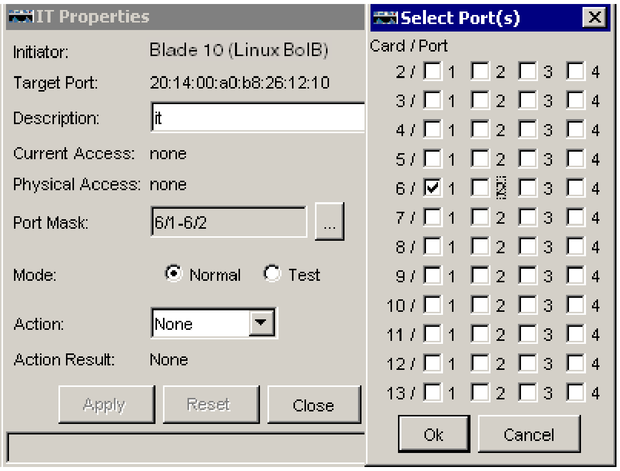
Action: None

Action Result: None

Figure 6-15   Configure port masking for the SRP host

4. Click the ... button shown in Figure 6-15 to open the Select Ports menu that gives the option to allow or to deny access to the SRP host through one or more ports. Add check marks to only one of the two ports as shown in Figure 6-16 and click **OK** to save the changes.

> **Note:** It is critical that a single and unique path is enabled from the host to the boot LUN for the initial operating system installation to complete successfully. If there is more than one path from the host to the boot LUN, then the operating system installation will fail.



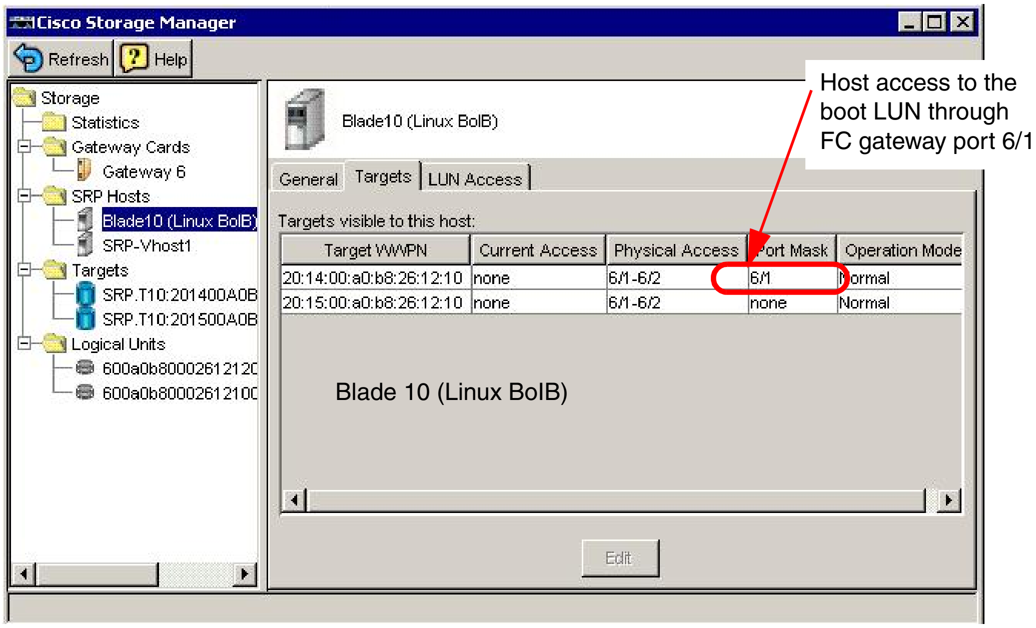
Initiator:	Blade 10 (Linux BoIB)
Target Port:	20:14:00:a0:b8:26:12:10
Description:	it
Current Access:	none
Physical Access:	none
Port Mask:	6/1-6/2 ...
Mode:	● Normal ○ Test
Action:	None
Action Result:	None

Card / Port

2 /	□ 1	□ 2	□ 3	□ 4
3 /	□ 1	□ 2	□ 3	□ 4
4 /	□ 1	□ 2	□ 3	□ 4
5 /	□ 1	□ 2	□ 3	□ 4
6 /	☑ 1	□ 2	□ 3	□ 4
7 /	□ 1	□ 2	□ 3	□ 4
8 /	□ 1	□ 2	□ 3	□ 4
9 /	□ 1	□ 2	□ 3	□ 4
10 /	□ 1	□ 2	□ 3	□ 4
11 /	□ 1	□ 2	□ 3	□ 4
12 /	□ 1	□ 2	□ 3	□ 4
13 /	□ 1	□ 2	□ 3	□ 4

Figure 6-16 Configure port masking for the SRP host: Select Ports menu

5. Figure 6-17 shows that the SRP host has access only to the boot LUN through the SFS3012-1 port 6/1.



Host access to the boot LUN through FC gateway port 6/1

Targets visible to this host:

Target WWPN	Current Access	Physical Access	Port Mask	Operation Mode
20:14:00:a0:b8:26:12:10	none	6/1-6/2	6/1	Normal
20:15:00:a0:b8:26:12:10	none	6/1-6/2	none	Normal

Figure 6-17 Port masking configuration view

This completes the port masking configuration process for the Boot over InfiniBand host.

6.2.9 Discovering the boot LUN

This section covers the procedure to discover the boot LUN from the Storage Manager menu of the SFS3012 FC Gateway. Follow these steps:

1. Select the SRP initiator listed in the SRP Hosts folder, then go to the Targets tab.
2. Click **Discover LUNs** (Figure 6-18).
3. Select the target port and the LUN on which the boot image will be installed and click **Add**. Then, click **Apply** to save the change.

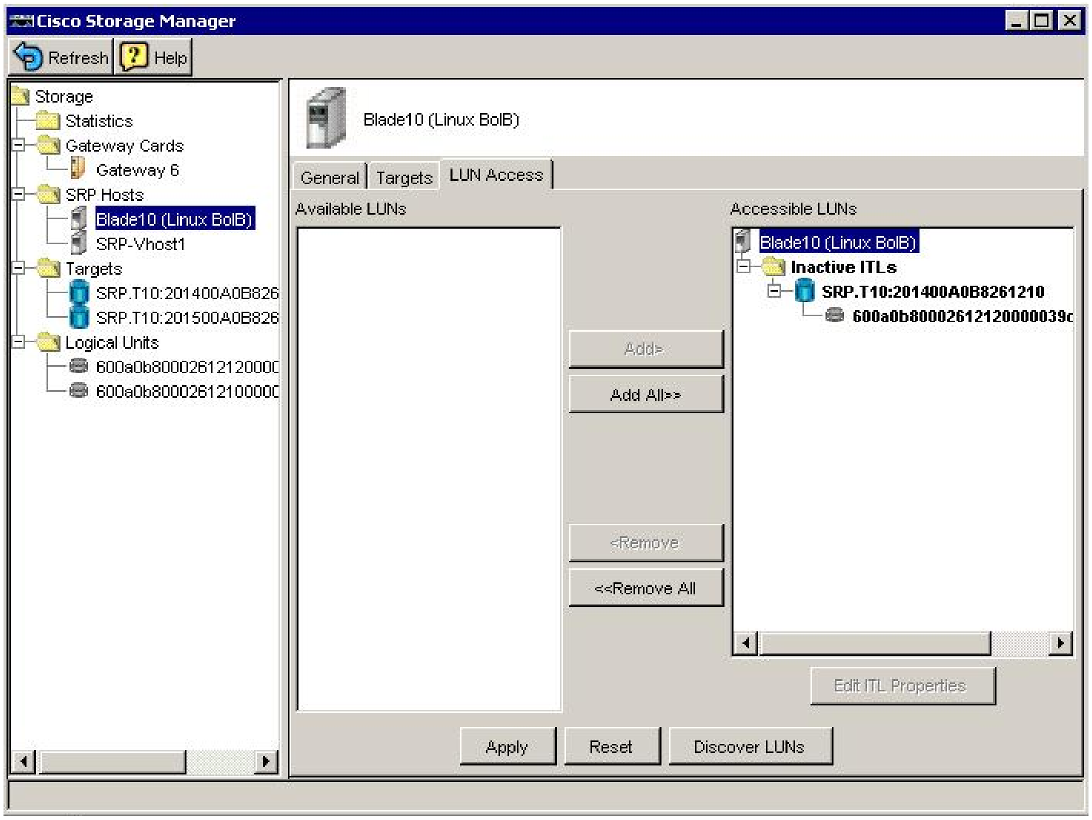
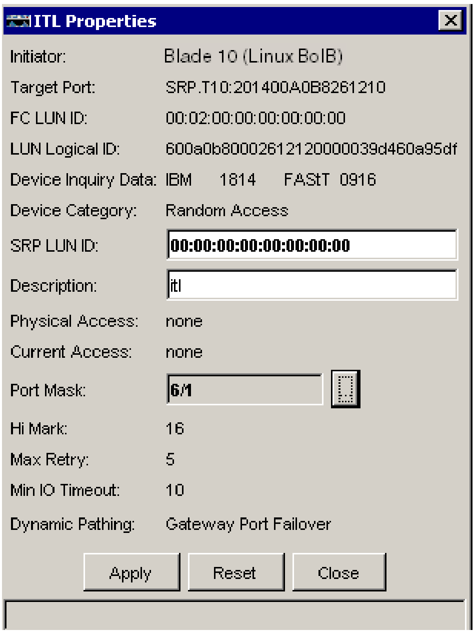


Figure 6-18 Accessible LUNs menu

4. Verify that the Boot LUN ID is 0 by selecting the boot LUN that was added in the preceding step and selecting ITL properties. Figure 6-19 opens with an SRP LUN ID=0.



Initiator: Blade 10 (Linux BoIB)
Target Port: SRP.T10:201400A0B8261210
FC LUN ID: 00:02:00:00:00:00:00:00
LUN Logical ID: 600a0b800026121200000039d460a95df
Device Inquiry Data: IBM 1814 FAStT 0916
Device Category: Random Access
SRP LUN ID: 00:00:00:00:00:00:00:00
Description: itl
Physical Access: none
Current Access: none
Port Mask: 6/1
Hi Mark: 16
Max Retry: 5
Min IO Timeout: 10
Dynamic Pathing: Gateway Port Failover

Figure 6-19 Verify Boot LUN ID = 0

6.3 Installing the OS on the Fibre Channel storage

This section covers some key configuration steps for installing Red Hat Enterprise Linux 4 Update 3 x64 on the HS21 blade booting over InfiniBand through the NFS Server. For the host to find the NFS server successfully through PXE, it must be configured to boot from the Network in the blade boot sequence.

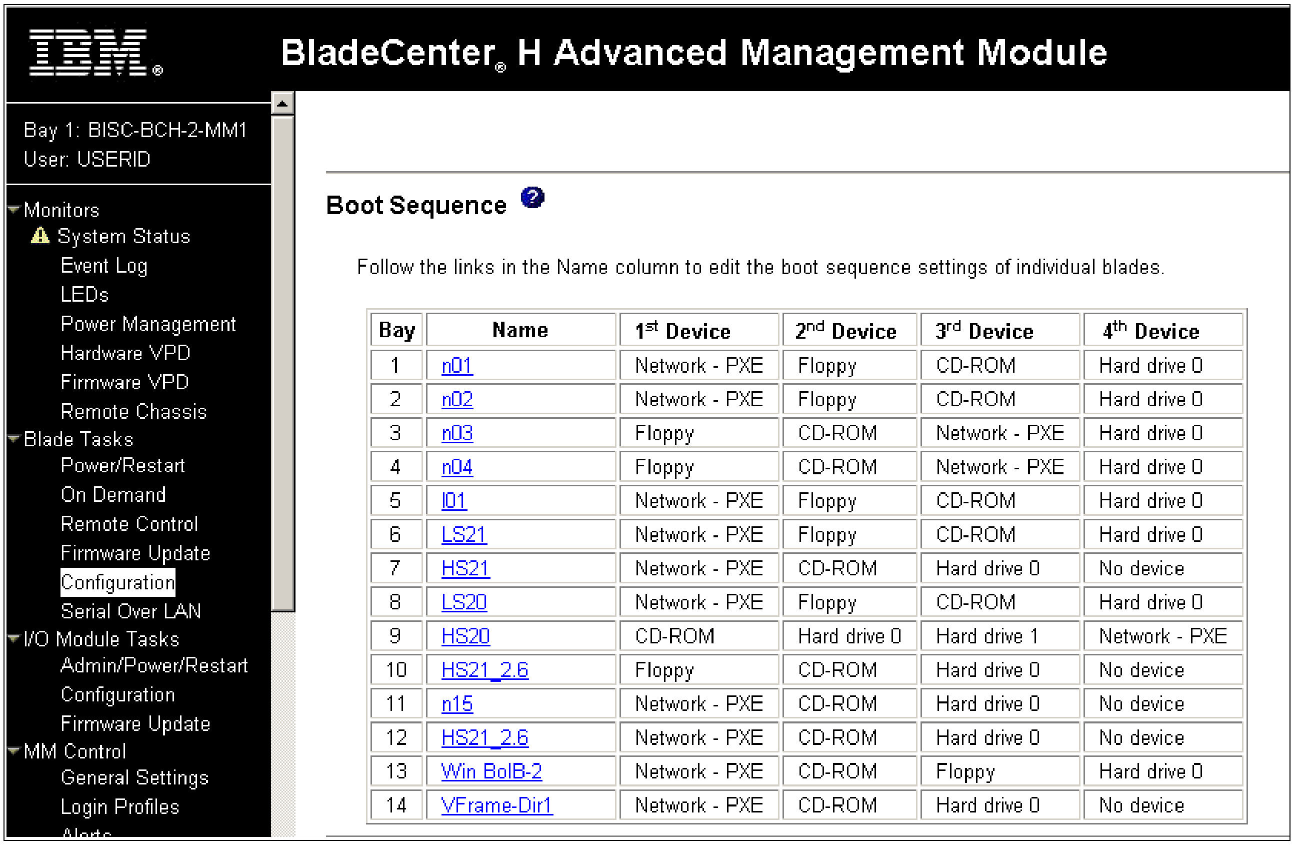
Figure 6-20 lists the procedure to load the driver over the network by using the kickstart configuration file with the driver disk parameter in it to access the InfiniBand host driver CD.

```
#use NFS installation media
driverdisk --source=nfs:9.42.167.200:/install/iso/drivers/topspin/cisco-BladeCenter-linux-3.2.0-118.iso
nfs --server 9.42.167.200 --dir /install/iso/rhel4/u3/x86_64
```

Figure 6-20 Command to load the host drivers through the network by using the kickstart configuration file

The blade boot sequence can be configured from the Advanced Management Module Web interface. Follow these steps:

1. Configure the blade server to boot over the network using the BladeCenter Advanced Management Module by clicking **Blade Tasks** → **Configuration** as shown in Figure 6-21.



Boot Sequence

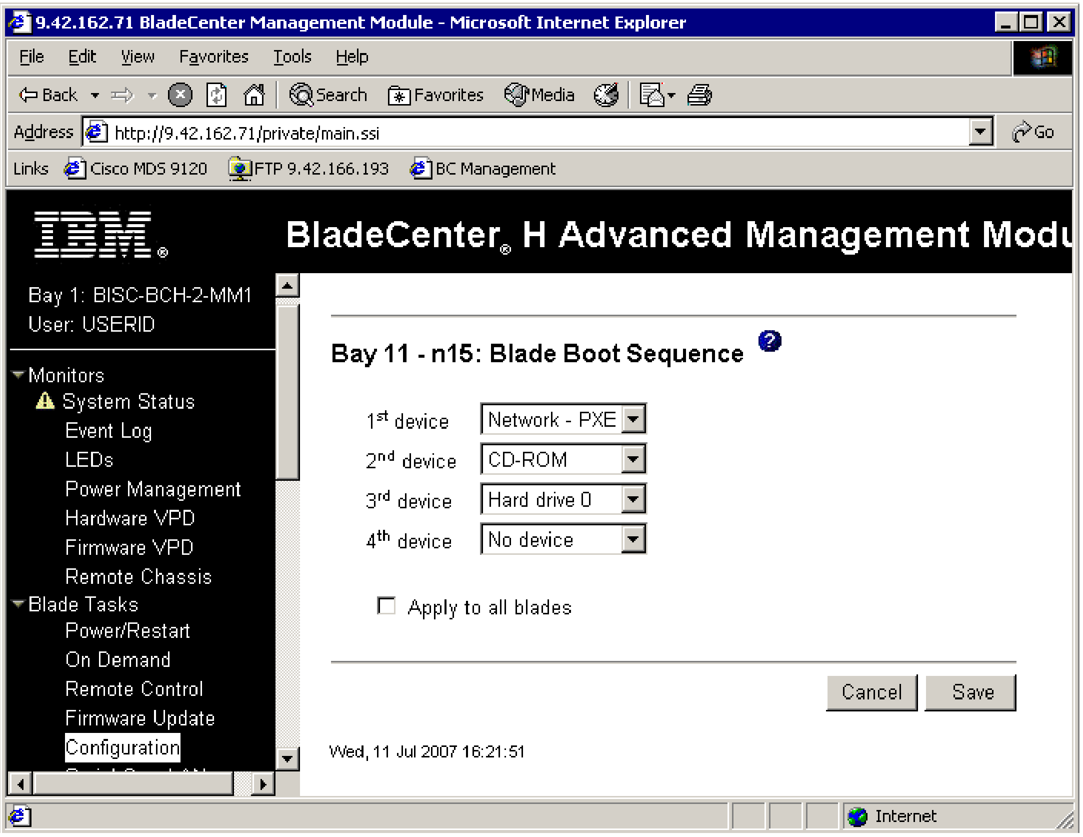
Follow the links in the Name column to edit the boot sequence settings of individual blades.

Bay	Name	1st Device	2nd Device	3rd Device	4th Device
1	n01	Network - PXE	Floppy	CD-ROM	Hard drive 0
2	n02	Network - PXE	Floppy	CD-ROM	Hard drive 0
3	n03	Floppy	CD-ROM	Network - PXE	Hard drive 0
4	n04	Floppy	CD-ROM	Network - PXE	Hard drive 0
5	l01	Network - PXE	Floppy	CD-ROM	Hard drive 0
6	LS21	Network - PXE	Floppy	CD-ROM	Hard drive 0
7	HS21	Network - PXE	CD-ROM	Hard drive 0	No device
8	LS20	Network - PXE	Floppy	CD-ROM	Hard drive 0
9	HS20	CD-ROM	Hard drive 0	Hard drive 1	Network - PXE
10	HS21_2.6	Floppy	CD-ROM	Hard drive 0	No device
11	n15	Network - PXE	CD-ROM	Hard drive 0	No device
12	HS21_2.6	Network - PXE	CD-ROM	Hard drive 0	No device
13	Win BoIB-2	Network - PXE	CD-ROM	Floppy	Hard drive 0
14	VFrame-Dir1	Network - PXE	CD-ROM	Hard drive 0	No device

Figure 6-21 Verify boot device order from the Advanced Management Module Web interface

2. Change specific servers by clicking the server name. Figure 6-22 opens, and you can select the boot order.

   Verify that the first boot device is selected as Network PXE.



9.42.162.71 BladeCenter Management Module - Microsoft Internet Explorer

Address http://9.42.162.71/private/main.ssi

BladeCenter® H Advanced Management Modu

Bay 1: BISC-BCH-2-MM1
User: USERID

Bay 11 - n15: Blade Boot Sequence

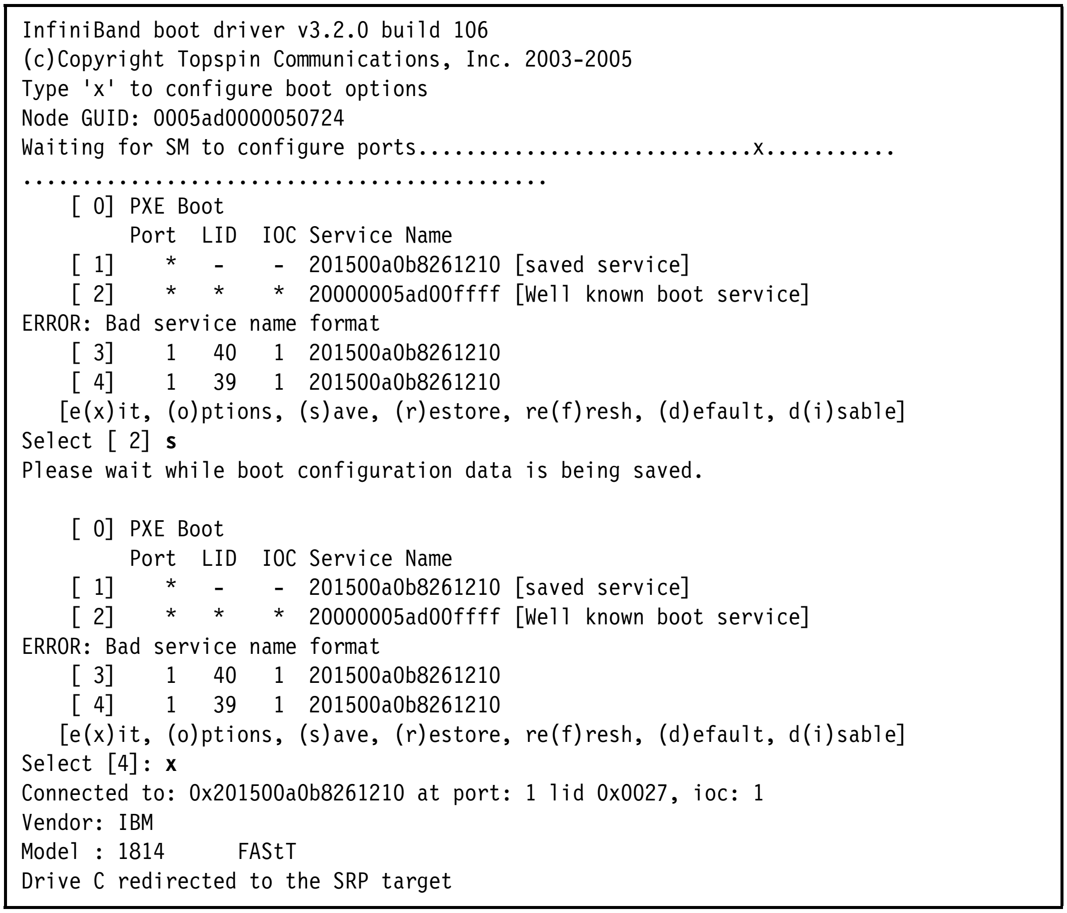
1st device	Network - PXE
2nd device	CD-ROM
3rd device	Hard drive 0
4th device	No device

☐ Apply to all blades

Wed, 11 Jul 2007 16:21:51

Figure 6-22 Configure Boot Sequence from the Advanced Management Module Web interface

3. Upon power on or restart, the service initializes into the Boot over InfiniBand firmware menu shown in Figure 6-23. Ensure that option *2 “Well Known boot service”* (the default value) is selected. Then select option **s** to save the configuration and select **x** to exit the InfiniBand firmware menu.

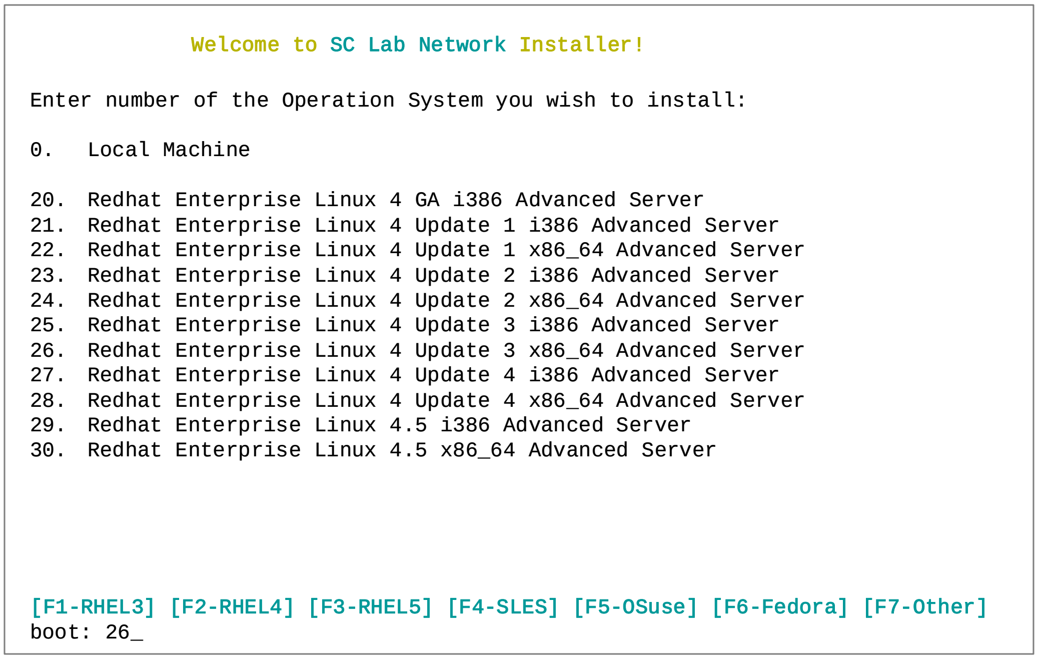


```
InfiniBand boot driver v3.2.0 build 106
(c)Copyright Topspin Communications, Inc. 2003-2005
Type 'x' to configure boot options
Node GUID: 0005ad0000050724
Waiting for SM to configure ports............................x...........
............................................
    [ 0] PXE Boot
         Port  LID  IOC Service Name
    [ 1]    *   -    -  201500a0b8261210 [saved service]
    [ 2]    *   *    *  20000005ad00ffff [Well known boot service]
ERROR: Bad service name format
    [ 3]    1   40   1  201500a0b8261210
    [ 4]    1   39   1  201500a0b8261210
   [e(x)it, (o)ptions, (s)ave, (r)estore, re(f)resh, (d)efault, d(i)sable]
Select [ 2] s
Please wait while boot configuration data is being saved.

    [ 0] PXE Boot
         Port  LID  IOC Service Name
    [ 1]    *   -    -  201500a0b8261210 [saved service]
    [ 2]    *   *    *  20000005ad00ffff [Well known boot service]
ERROR: Bad service name format
    [ 3]    1   40   1  201500a0b8261210
    [ 4]    1   39   1  201500a0b8261210
   [e(x)it, (o)ptions, (s)ave, (r)estore, re(f)resh, (d)efault, d(i)sable]
Select [4]: x
Connected to: 0x201500a0b8261210 at port: 1 lid 0x0027, ioc: 1
Vendor: IBM
Model : 1814      FAStT
Drive C redirected to the SRP target
```

Figure 6-23 InfiniBand HCA Firmware menu

4. The blade power-on initialization process proceeds to look for a DHCP server using PXE. Upon successful discovery of DHCP server, it allows you to select the appropriate operating system to install. The installation menu in Figure 6-24 is a custom menu and can also be implemented differently in your environment.

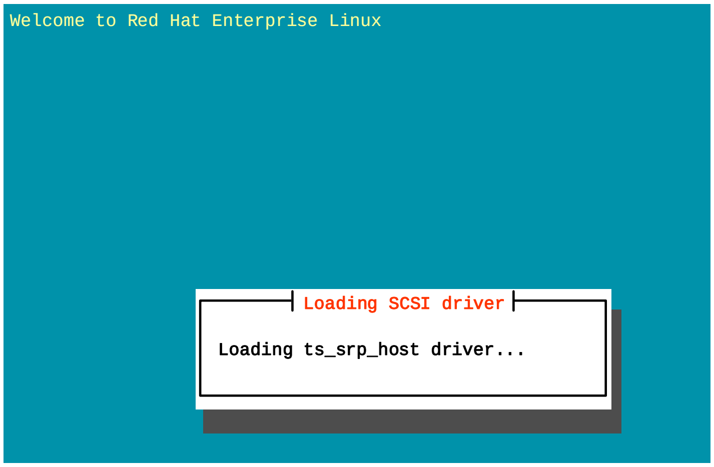
   At the boot prompt, we selected RHEL 4 U3 x86_64.



```
             Welcome to SC Lab Network Installer!

Enter number of the Operation System you wish to install:

0.   Local Machine

20.  Redhat Enterprise Linux 4 GA i386 Advanced Server
21.  Redhat Enterprise Linux 4 Update 1 i386 Advanced Server
22.  Redhat Enterprise Linux 4 Update 1 x86_64 Advanced Server
23.  Redhat Enterprise Linux 4 Update 2 i386 Advanced Server
24.  Redhat Enterprise Linux 4 Update 2 x86_64 Advanced Server
25.  Redhat Enterprise Linux 4 Update 3 i386 Advanced Server
26.  Redhat Enterprise Linux 4 Update 3 x86_64 Advanced Server
27.  Redhat Enterprise Linux 4 Update 4 i386 Advanced Server
28.  Redhat Enterprise Linux 4 Update 4 x86_64 Advanced Server
29.  Redhat Enterprise Linux 4.5 i386 Advanced Server
30.  Redhat Enterprise Linux 4.5 x86_64 Advanced Server




[F1-RHEL3] [F2-RHEL4] [F3-RHEL5] [F4-SLES] [F5-OSuse] [F6-Fedora] [F7-Other]
boot: 26_
```

Figure 6-24 Select the operating system version and release to install over the network

5. The installation proceeds. Figure 6-25 shows the installer is loading the ts_srp_host driver correctly.



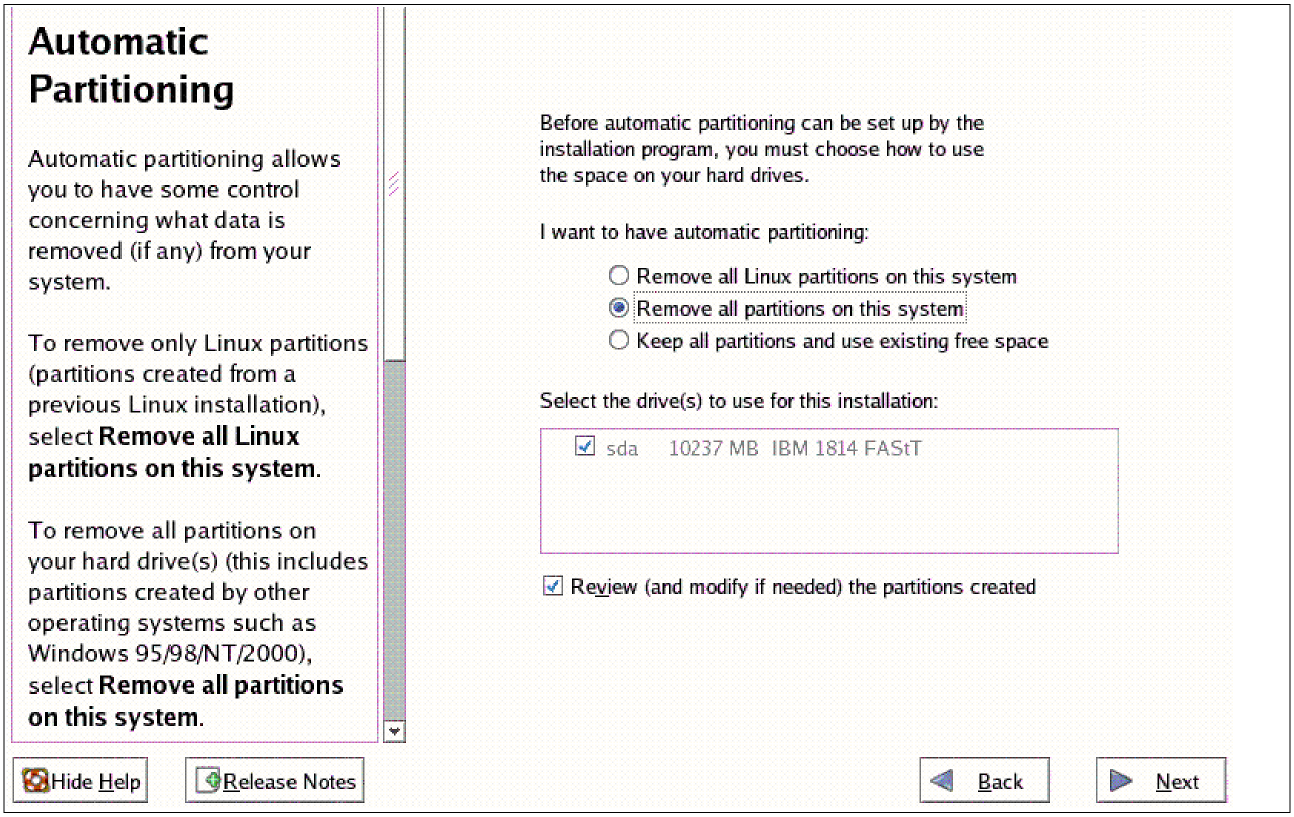
```
Welcome to Red Hat Enterprise Linux

              Loading SCSI driver

  Loading ts_srp_host driver...
```

Figure 6-25 Loading SRP driver on the blade

6. Next, select the language type and then the keyboard type.

7. Configure storage partitioning by selecting the Fibre Channel-attached boot LUN. You can select auto partitioning or manual. Then, click **Next** to proceed with storage partitioning.

8. If there was a previous image installed on the boot disk, then it is highly recommended that you remove all the partitions as shown in Figure 6-26. Click **Next**.



Automatic Partitioning

Automatic partitioning allows you to have some control concerning what data is removed (if any) from your system.

To remove only Linux partitions (partitions created from a previous Linux installation), select **Remove all Linux partitions on this system**.

To remove all partitions on your hard drive(s) (this includes partitions created by other operating systems such as Windows 95/98/NT/2000), select **Remove all partitions on this system**.

Before automatic partitioning can be set up by the installation program, you must choose how to use the space on your hard drives.

I want to have automatic partitioning:

○ Remove all Linux partitions on this system
● Remove all partitions on this system
○ Keep all partitions and use existing free space

Select the drive(s) to use for this installation:

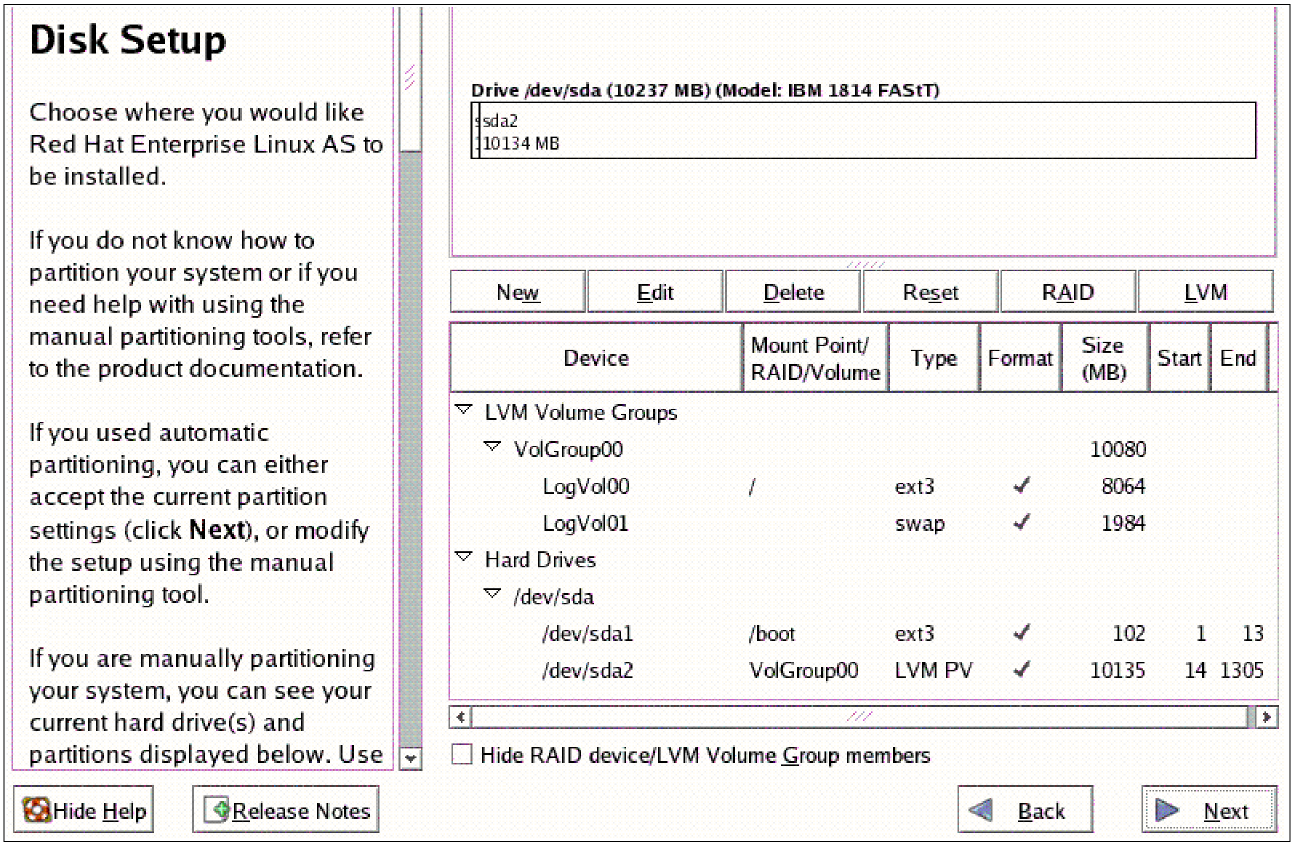
☑ sda 10237 MB IBM 1814 FAStT

☑ Review (and modify if needed) the partitions created

Figure 6-26 Remove all partitions on the system

9. Select Yes to confirm deleting all the partitions.

10. Figure 6-27 shows the storage configuration and confirms that the Fibre Channel-attached boot LUN is accessible by the host for read and write operations.



Disk Setup

Choose where you would like Red Hat Enterprise Linux AS to be installed.

If you do not know how to partition your system or if you need help with using the manual partitioning tools, refer to the product documentation.

If you used automatic partitioning, you can either accept the current partition settings (click **Next**), or modify the setup using the manual partitioning tool.

If you are manually partitioning your system, you can see your current hard drive(s) and partitions displayed below. Use

Drive /dev/sda (10237 MB) (Model: IBM 1814 FAStT)

sda2
10134 MB

Device	Mount Point/ RAID/Volume	Type	Format	Size (MB)	Start	End
▽ LVM Volume Groups						
▽ VolGroup00				10080		
LogVol00	/	ext3	✓	8064		
LogVol01		swap	✓	1984		
▽ Hard Drives						
▽ /dev/sda						
/dev/sda1	/boot	ext3	✓	102	1	13
/dev/sda2	VolGroup00	LVM PV	✓	10135	14	1305

□ Hide RAID device/LVM Volume Group members

Figure 6-27 Disk Partitions on the boot LUN through automatic storage partitioning utility

11. Continue with the remainder of the installation process.

This completes the configuration and installation procedure for the successful installation of RHEL 4 U3 on the HS21 blade booting from InfiniBand with and a FC gateway and the DS4700 connected to the SFS3012.

7

Configuring the Cisco 3012 InfiniBand to Ethernet gateway

This chapter discusses the Cisco 3012 InfiniBand to Ethernet Gateway Module that is available for use in the Cisco SFS 3000 platforms. We discuss the basics of how the product works and then cover basic usage and more advanced topics, such as using VLANs, EtherChannel, and gateway redundancy.

The topics that we discuss in this chapter are:

7.1 Introduction to the Ethernet gateway module

The optional InfiniBand to Ethernet gateway module provides a way for InfiniBand clients to talk to devices on an Ethernet LAN, and vise versa. This module is considered a transparent L2 device, in that the clients do not need to be aware they are using it. Physically it offers two inward facing 4X (10G) InfiniBand ports, and six outward facing 10/100/1000 Ethernet ports. It can be installed in any of the expansion slots of the SFS 3012, or in the single expansion slot of a 3001 (the 3001 is not covered here). The gateway modules and most other components of the 3012 are hot swappable.

Figure 7-1 shows a Cisco 3012 installed with an equal number of Ethernet and Fibre Channel gateway modules.

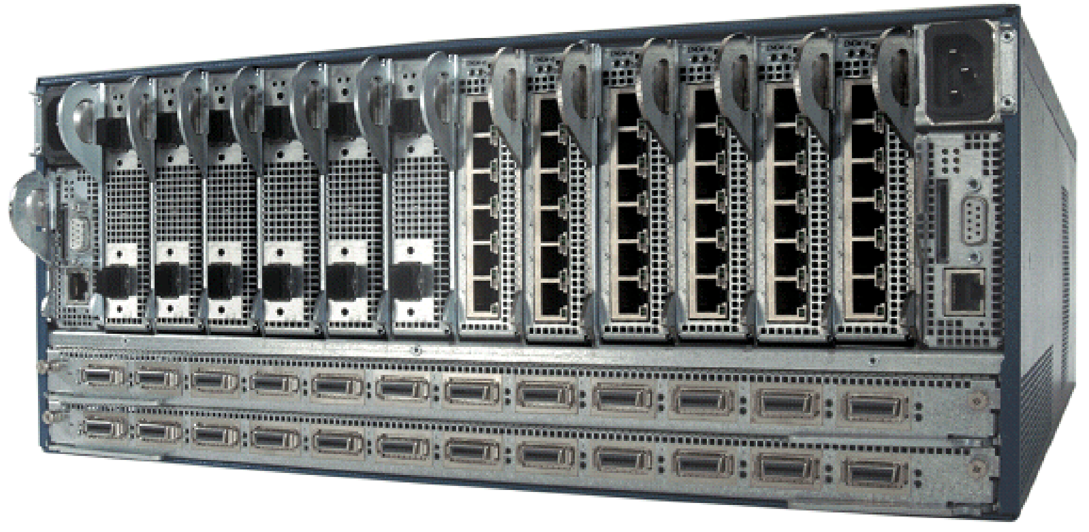


Figure 7-1 Cisco 3012 - Fully loaded

Figure 7-2 shows the six port Ethernet gateway module

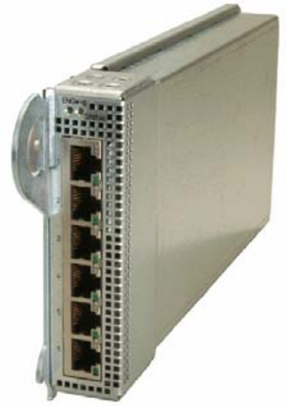


Figure 7-2 The 6 port InfiniBand to Ethernet gateway module

Features and characteristics of the Ethernet gateway module include:

- Two inward facing 4X (10G) InfiniBand ports
- Six outward facing RJ45 10/100/1000 Ethernet ports with MDI/MDI-X support
- Can be installed in any of the expansion slots of the SFS 3012
- Hot-swappable
- Jumbo frame support built in
  - 9 K, Ethernet to InfiniBand
  - 2 K, InfiniBand to Ethernet
- Supports up to 32 VLANs, both static VLANs and 802.1Q trunking
- Up to a six port static EtherChannel (also referred to as trunking or aggregation)
- Some items not supported in this module
  - Does not support Spanning-tree (uses proprietary methods to prevent loops)
  - Does not support CDP (CDP is supported on the Ethernet management port of the 3012 itself, but not on the Ethernet gateway modules Ethernet ports)

Figure 7-3 shows a basic block flow of the Ethernet gateway module within the 3012, along with how a bridge group is used to move packets from InfiniBand to Ethernet and vise versa.

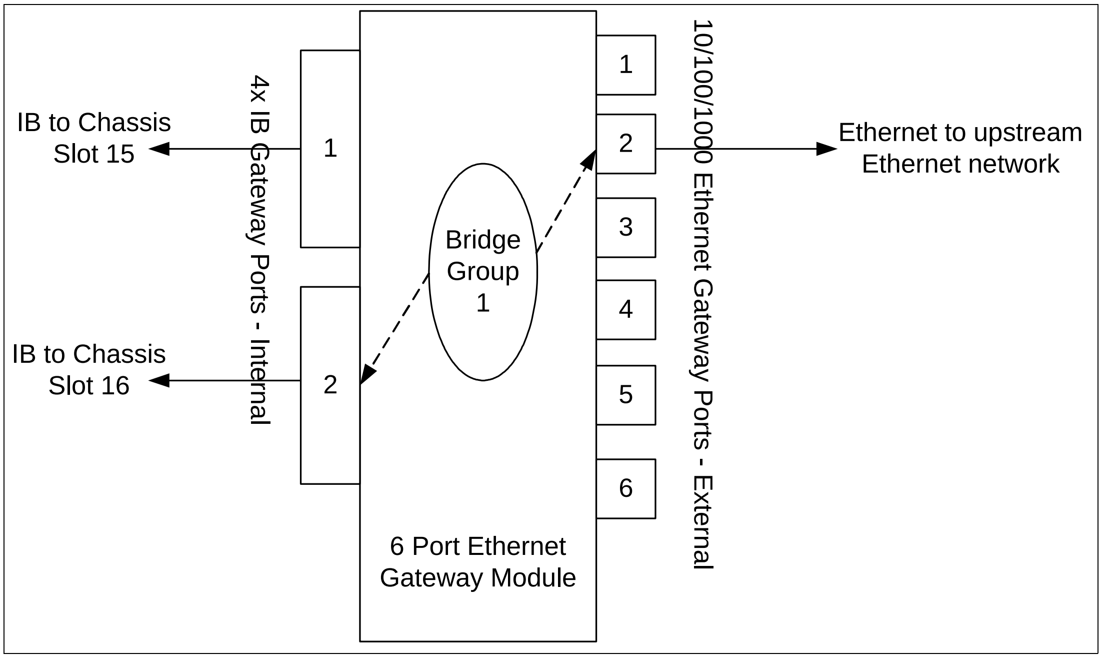

IB to Chassis Slot 15
IB to Chassis Slot 16
4x IB Gateway Ports - Internal
1
2
Bridge Group 1
6 Port Ethernet Gateway Module
1
2
3
4
5
6
10/100/1000 Ethernet Gateway Ports - External
Ethernet to upstream Ethernet network

Figure 7-3 Block flow of Ethernet gateway module

Some facts about the gateway module:

- This module is considered a Target Channel Adapter (TCA) on the InfiniBand side. It terminates an InfiniBand link and executes transport level functions.
- This module operates in what is known as *cut-through mode*, which provides for maximum throughput of data.

- The 3012 (and thus the Ethernet gateway module) supports four different options for configuring:
  - CLI
  - GUI - http
  - GUI - Element Manager
  - SNMP
- LED meanings at the gateway module level:
  - Green – InfiniBand Module Status Indicator
    - On – Module active
    - Off – Module inactive
  - Yellow – InfiniBand Attention Indicator
    - On – Attention required (Something is wrong)
    - Off – All OK
- Port Level LED meanings
  - Dual color LED
    - Off – no signal
    - Solid Yellow – Port Disabled
    - Flashing Yellow – Port Fault
    - Solid Green – Port online
  - Green LED:
    - Off – no traffic
    - Blinking – traffic exists on port

7.1.1 Some general assumptions

All examples in this chapter make the following assumptions:

- An SFS3012 is in use for gateway support to Ethernet.
- The basic operation of the 3012 (for example, out-of-band management) have already been configured, and you can communicate with the 3012 by either telnet or the Element Manager (depending on if you want to use a CLI or a GUI).
- All GUI examples are from the Element Manager. (There is similar functionality available from the built-in http access to the 3012, but we do not discuss that functionality here.)
- All necessary cabling is already in place.
- Examples are provided on how to configure the various gateway options through CLI and the Element Manager GUI.
- The Element Manager used for GUI configuration examples in this section was version 2.9.0.10, running on JVM™ 1.4.2_05.
- The TopSpin operating system used on the 3012 was Topspin-360 TopspinOS 2.8.0 releng #127 03/15/2007 22:37:23.
- InfiniBand clients used for testing gateway access were IBM BladeCenter HS21 servers (machine type 8853), running Red Hat Enterprise Linux 5, Kernel 2.6.18-8.el5, and OFED (OpenFabrics Enterprise Distribution) host InfiniBand driver version OFED-1.2 1.2-2.6.18_8.el5.
- All clients discussed are using the default p_key (ff:ff) for IP over InfiniBand traffic. Configuring for non-default p_keys is not necessarily a difficult task, but it is beyond the scope of this document.

7.1.2 Some general items to consider

Some general topics that are important to those working with this product include:

- Use of the GUI or CLI interface

  Either the GUI or the CLI or both can be used to configure the gateway module. In general, the GUI (either the Element Manager or the built-in http support) is good for one-off configurations or for use by those that are not familiar with the CLI. The CLI is usually used by those deploying (scripting) a number of installs, and that are more comfortable with the greater flexibility (and thus increased complexity) that it offers.

  For those that are unfamiliar with the CLI but want to know more about how it operates, one of the best ways to become familiar with the CLI is to use the GUI to configure a desired option, and then go in to the CLI of the 3012 and do a **`show config`** to see what changes were made by the GUI.

  On that note, it should be kept in mind that making changes through the GUI will result in those changes becoming part of the CLI configuration. The reverse also applies: making changes in the CLI are reflected in what is displayed in the GUI.

  In both cases, changes made take effect immediately (with CLI, when you press Enter on a command, and with GUI, when you click **OK** or the option to **Apply**).

  Any changes made are only stored in the running configuration. You should also save configuration changes to NVRAM to make them safe in case of a reload or power cycle. With CLI, saving the configuration is done with the command **`copy running startup`**. With the Element Manager GUI, this is done by clicking **Maintenance** → **Save Config**.

- What is a trunk?

  The term *trunk* is used differently by different groups. In most Cisco products (other than the SFS products) a trunk is an IEEE 802.1Q term for a connection that carries multiple VLANs.

  For the SFS platform (and some other companies) a trunk is created by bundling multiple physical Ethernet ports into a larger logical port, for increased performance and high availability. Most other Cisco products refer to this bundling as either aggregation or EtherChannel.

  When referenced in this section, trunk will represent the SFS version of this term (aggregation/EtherChannel).

- What are VLANs?

  VLANs are *Virtual LANs*. In most cases, each VLAN represents a separate and distinct IP subnet. VLANs offer a way in the Ethernet world to isolate traffic flows within a switch and on a shared wire, which helps to reduce the size of broadcast domains. It can also add some security, in that traffic in one VLAN will not be seen on a different VLAN, unless it can be carried over an L3 (router) device.

  When carrying VLANs between Ethernet switches, some sort of method needs to be used to let each side of the connection know what VLAN a given packet is to be placed in upon arrival. This is done by inserting a tag in the header, that among other things, carries a field defining the VLAN it is part of. The most common protocol used for this tagging is IEEE 802.1Q, which is the tagging protocol supported by the 3012 gateway.

  Whether or not to use VLANs depends on your own network requirements. In general, using a single VLAN (no tagging) is simplest to configure. Multiple VLANs should be used when you're network requires this functionality.

You can find more information about understanding and configuring the Ethernet gateway module in documentation that is available at the following links:

- Cisco SFS 3000 series documentation home

  http://www.cisco.com/en/US/products/ps6422/index.html

- *Cisco SFS InfiniBand Ethernet Gateway User Guide*

  http://www.cisco.com/en/US/products/ps6422/products_user_guide_list.html

- *Cisco SFS InfiniBand Redundancy Configuration Guide*

  http://www.cisco.com/en/US/products/ps6422/products_installation_and_configuration_guides_list.html

- *SFS 7000 Series Command Reference* (covers 3000 series also)

  http://www.cisco.com/en/US/products/ps6421/products_command_reference_book09186a008070c2eb.html

- *SFS 3000 Specific Command Reference*

  http://www.cisco.com/application/pdf/en/us/guest/products/ps6421/c2001/ccmigration_09186a00808669e2.pdf

7.2 Client information

Before configuring the Ethernet gateway, it is usually best to configure a couple of clients for IP over InfiniBand access. By configuring two IP over InfiniBand clients, you can use these to test IPoIB on the InfiniBand fabric, irrespective of the gateway (making sure they can ping each other). This ensures the clients are configured correctly and will help to minimize troubleshooting if you have issues configuring the gateway.

It is also assumed the client will have a default gateway configured, that points to the IP address on the Ethernet device that is acting as the default route for non-local traffic. If a default gateway is not configured (and no static routes exist), then the client will usually be unable to connect to IP devices in different IP subnets.

Tip: To permit an InfiniBand client to talk with IP subnets other than their own on the Ethernet network, you must have two items:

- Either a default gateway or static routes added directly to the client
- Either a default gateway or static routes assigned within the bridge-group on the Ethernet gateway module in the 3012

It is very easy to forget to have one or the other, and thus be unable to ping outside the clients own subnet.

For this section, we tested only Red Hat Linux clients. It is also possible to use other versions of Linux or Windows, but we do not cover other versions in this section.

This section uses clients with the OpenFabrics Enterprise Distribution (OFED) drivers. For information about installing and configuring the OFED drivers, see 5.1, “Configuring IPoIB on a Linux host” on page 82.

You can find more information and download the OFED driver from the following link (requires a CCO ID to access):

http://www.cisco.com/cgi-bin/tablebuild.pl/sfs-linux

After you have configured these two clients and they can ping each other, you can proceed to configuring the Ethernet gateway module, examples of which can be found in the following sections.

7.3 Implementing a simple bridge group connection to an upstream Ethernet network

In this section, we demonstrate the ability to ping from an InfiniBand server to an IP address on the upstream Ethernet switch, using a single Ethernet connection out of the gateway module. This is the simplest design but offers no redundancy and no specific VLAN support.

7.3.1 The design

Figure 7-4 shows the basic layout of our test environment for this connection.

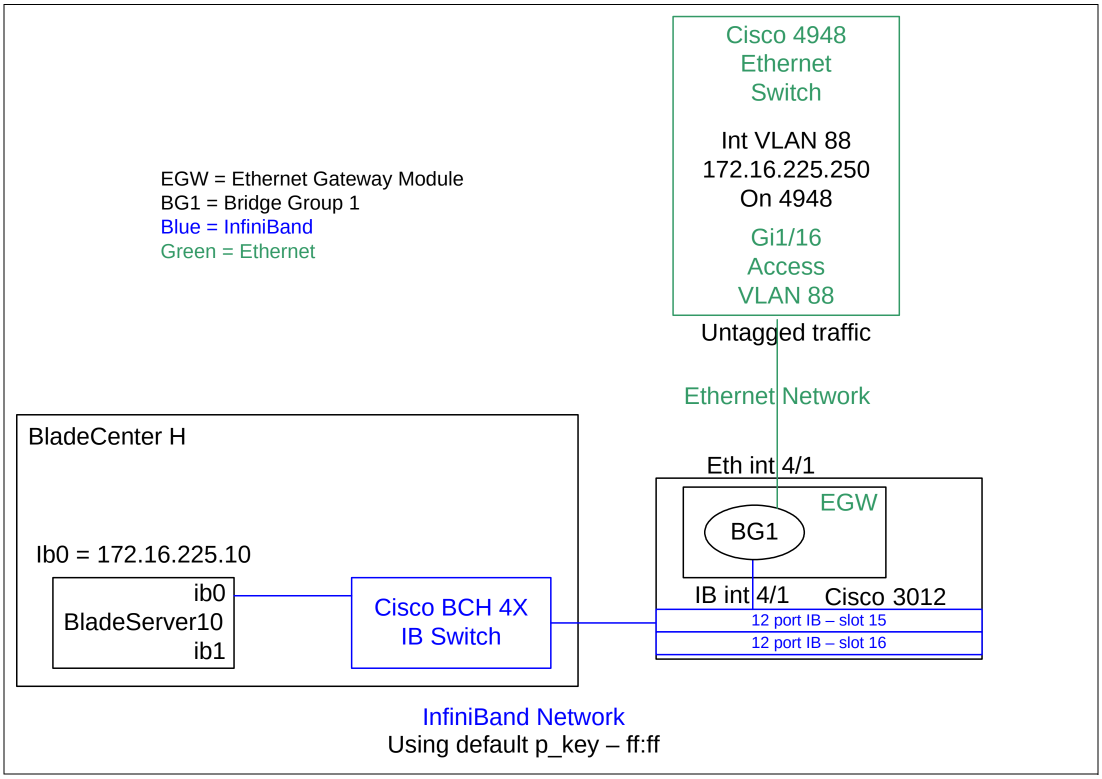


Figure 7-4 Simple connectivity from InfiniBand client to Ethernet network

The IP address of server 10, *172.16.225.10*, is on the ib0 interface for this server. It is assumed that this IP address is already configured and tested at this point.

On the gateway module, we use Ethernet port 4/1 and InfiniBand port 4/1 in this example and create and configure bridge group 1 (BG1).

This example uses VLAN 88 on the Ethernet switch port connecting to interface Ethernet port 4/1 on the Ethernet gateway. Since this port will be configured as Access mode on the upstream switch, only untagged packets will be sent and received. This means the Ethernet gateway port should be configured as *untagged* (when the untagged packet reaches the

Cisco 4948 Ethernet Switch, the 4948 will put any untagged packet on to VLAN 88 inside the 4948).

Assuming the client and other components are already configured, all we have to do is create a bridge group with the desired options in the gateway module, to permit packets to flow. Note that a bridge group is a logical entity that allows the gateway to pass InfiniBand frames to the Ethernet network.

7.3.2 Some comments on bridge groups

When creating a bridge group, remember:

- A bridge group is an entity that runs on the Ethernet gateway module and enables the bridging of one IPoIB partition to one VLAN.
- The Ethernet gateway acts like a Layer 2 bridge between InfiniBand and Ethernet. Each Ethernet gateway must be configured for Layer 2 bridging (with or without link aggregation and redundancy groups).
- Configuring a bridge group enables the system to learn everything it needs to know about the location of nodes on the network with very little input from the administrator.
- The bridge group only bridges the IP protocol and drops all other traffic.
- Attributes of the bridge port must be created or deleted with the bridge group. For example, the physical port attribute is assigned at the time the bridge group is created, and cannot be changed without deleting the bridge group.
- The parameters that you define for the bridge group determine the way the traffic is handled between the InfiniBand network and the Ethernet network. The bridge port has certain attributes that are always attached to the bridge port entity, and certain attributes that are optional to the bridge port.
- Some mandatory items:
  - You must select a p_key (default is ff:ff). If you use a non default p_key on the gateway module, you will need to configure the InfiniBand clients and InfiniBand fabric to use this p_key.
  - You must select a physical Ethernet port number(s) or aggregation/trunk interface.
- Some optional items:
  - Non-default p_key (not covered in this chapter)
  - VLAN tagging
  - Aggregation/trunking

For our example, we demonstrate creating the bridge group using the Element Manager. Then we show what the CLI looks like after this configuration is applied.

7.3.3 Summary of steps to implement a simple bridge group design

Here, we provide a summary of the steps that we describe in the remainder of this section to implement the simple bridge group design:

1. In the 3012 Element Manager, open the Bridging window (step 1 on page 125).
2. Open the Add Bridge Group window and complete the information in the Groups tab (step 2 on page 126).
3. Complete the Forwarding and Subnet tabs in the Add Bridge Group window (step 3 on page 128).

4. Commit the new Bridge Group information and review (step 4 on page 129).
5. Save the 3012 config to NVRAM (step 5 on page 130).
6. Configure the upstream 4948 to match the Ethernet Gateway configuration (step 6 on page 130).
7. Test the configurations by pinging from the InfiniBand client to the default gateway on the 4948 (step 7 on page 131).

7.3.4 Detailed steps to implement a simple bridge group design

To implement a simple bridge group design, follow these detailed steps:

1. To create our first bridge group, from the Element Manager GUI, click **Ethernet** → **Bridging** (see Figure 7-5).

> **Tip:** The default user ID and password for the 3012 is `super/super`.

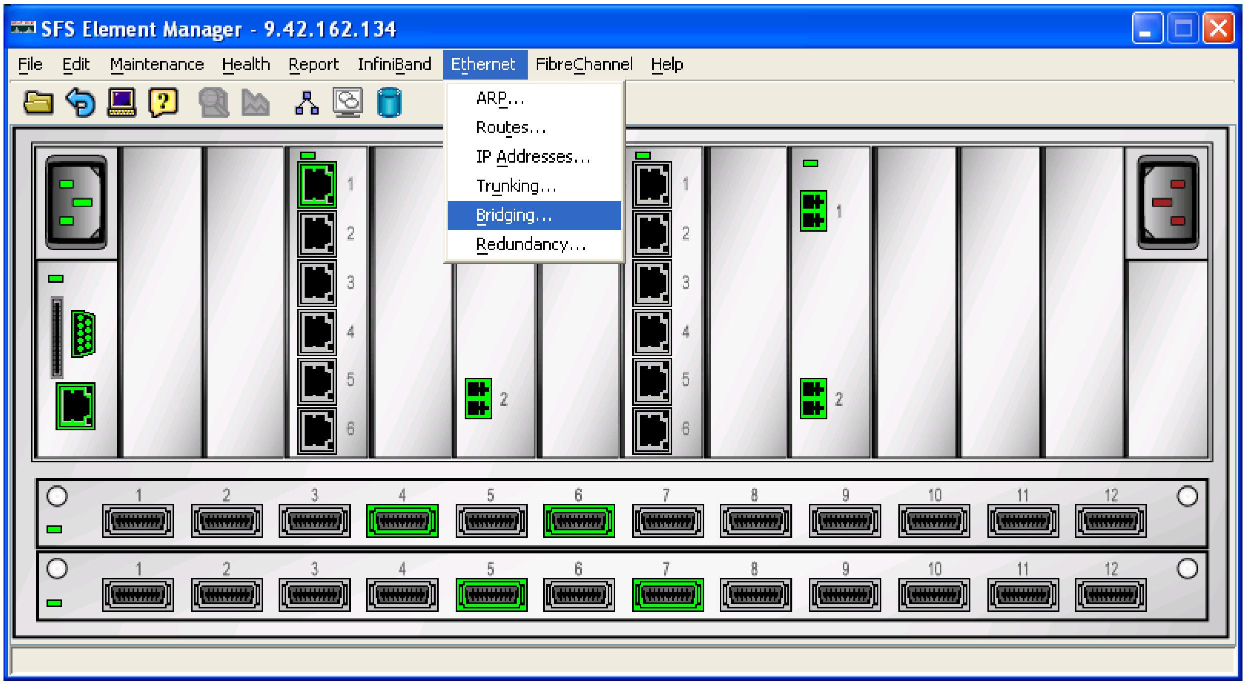
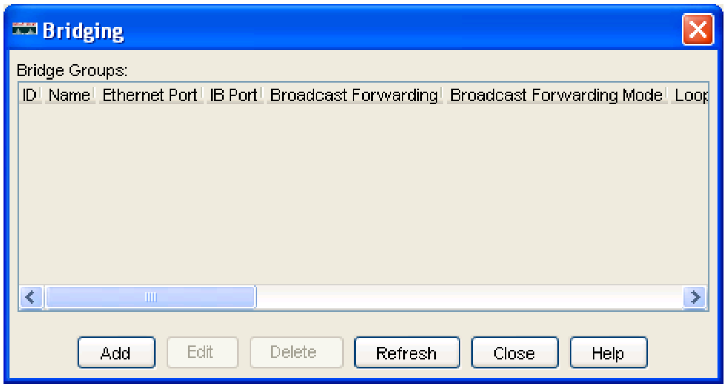


Figure 7-5 First step in creating a bridge group through the Element Manager GUI

Assuming a fresh installation, Figure 7-6 opens.



Bridge Groups:

ID	Name	Ethernet Port	IB Port	Broadcast Forwarding	Broadcast Forwarding Mode	Loop

Figure 7-6 Bridge group page with no bridge groups

2. Click **Add** to create your first bridge group with the following characteristics:

   – The Ethernet port that we use is 4/1. This port is untagged because we want to send untagged packets to our upstream switch.

   – The InfiniBand port is 4/1 in this example and uses the default p_key, ff:ff.

   > **Tip:** If only using a single Ethernet Gateway, you should use InfiniBand port 2 for connection to the InfiniBand fabric. This is because the design of the 3012 is such that the module in slot 15 (InfiniBand port 1 on the Ethernet gateway) will not work if the InfiniBand module in slot 16 is not present or otherwise in a failed state. To state this a different way, an InfiniBand module in slot 16 will work by itself, but an InfiniBand module in slot 15 will only work if there is an InfiniBand module in slot 16.

   – We do not enable Broadcast forwarding on this bridge group.

   > **Tip:** A note about Broadcast forwarding. In most cases, this is not necessary or even desired to be enabled. The gateway will still correctly pass ARP packets without enabling this feature. With that said, if you have an application that requires Broadcast forwarding enabled, you will need to enable this.
   >
   > One example of when you will want to enable this feature is if you have InfiniBand clients using DHCP to obtain an IP address. Do not enable this feature if you plan on having two gateway modules with bridge groups carrying the same p_key and VLAN, and hooked to the same InfiniBand and Ethernet fabrics, as this would result in a network loop.
   >
   > If you do want to have two gateways configured the same with the same connectivity, and require Broadcast forwarding, then place the modules into a Redundancy Group. The Redundancy Group will prevent this design from looping the network.

   – We set Loop Protection Method to one (the default). Regardless of this setting, the following two loop protection methods are always enabled and cannot be turned off:

     - Self-cancelling ARP requests: By default, the self-cancelling ARP requests feature is always active. It prevents duplicate ARP (requests that have the same target protocol address) from creating loops. The duplicate ARP is seen by multiple gateways and discarded.

- Delayed proxy ARP transaction: By default, the delayed proxy ARP transaction feature is always active. This feature is an extension of the self-cancelling ARP request and comes into play when a duplicate ARP request is delayed.

– Setting Loop Protection to one allows for ARP packet painting, an extra loop protection mechanism.

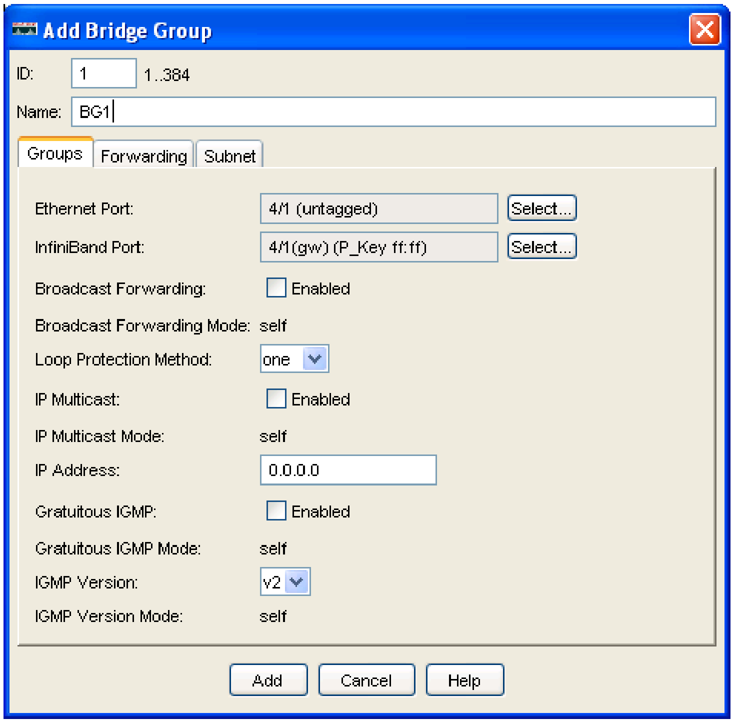
  Though the loop protection mechanisms described above are very effective on most LANs, it is possible for a duplicated ARP request to be received on the other side of the bridge before the original has arrived. In this event, the duplicated packet is considered to be the original, and the original packet is dropped.

  To prevent this rare occurrence, the Ethernet gateway provides the option of inserting a signature at the end of every proxy ARP request. This signature allows the Ethernet gateway to filter the duplicate requests, and break the loop. This feature is enabled by default and can be disabled if your Ethernet switch does not support it

– We do not configure an IP address for this bridge group for our example.

– We leave the multicast settings default.

Figure 7-7 shows the settings selected on the Groups tab of the Add Bridge Group dialog box.



ID: 1 1..384

Name: BG1

Ethernet Port: 4/1 (untagged)

InfiniBand Port: 4/1(gw) (P_Key ff:ff)

Broadcast Forwarding: □ Enabled

Broadcast Forwarding Mode: self

Loop Protection Method: one

IP Multicast: □ Enabled

IP Multicast Mode: self

IP Address: 0.0.0.0

Gratuitous IGMP: □ Enabled

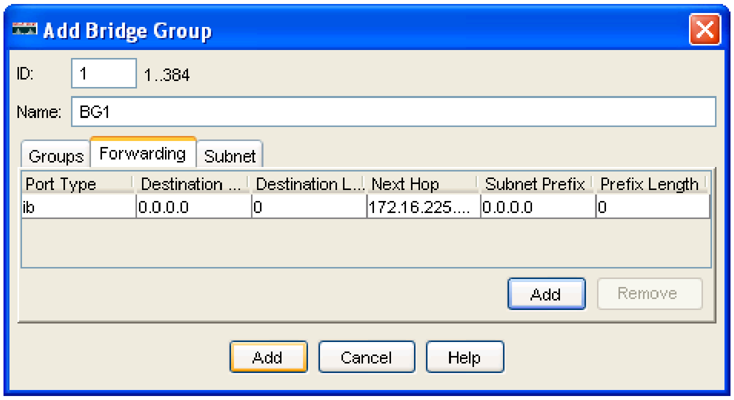
Gratuitous IGMP Mode: self

IGMP Version: v2

IGMP Version Mode: self

Figure 7-7 Groups tab settings for BG1

3. After making changes to the Add Bridge Group Groups tab, you need to configure the Forwarding tab and the Subnet Tab as shown in Figure 7-8 and Figure 7-9 on page 129



ID: 1 1..384

Name: BG1

Port Type	Destination ...	Destination L...	Next Hop	Subnet Prefix	Prefix Length
ib	0.0.0.0	0	172.16.225....	0.0.0.0	0

Figure 7-8 Forwarding tab on Add Bridge Group window

In the example in Figure 7-8, you set up a default gateway for this bridge group to use on the upstream Ethernet network.

Some comments about the settings in the Forwarding tab:

– Created by clicking the **Add** button in this tab (*not* the **Add** button at the bottom of the page).

– For an InfiniBand client to talk to devices on a different IP subnet, a default gateway needs to be set in two places:

  - On the InfiniBand client itself
  - In this tab for the bridge group

– Port Type is set to *ib*. This indicates packets coming in from the InfiniBand side will use this address on the Ethernet side as their next hop.

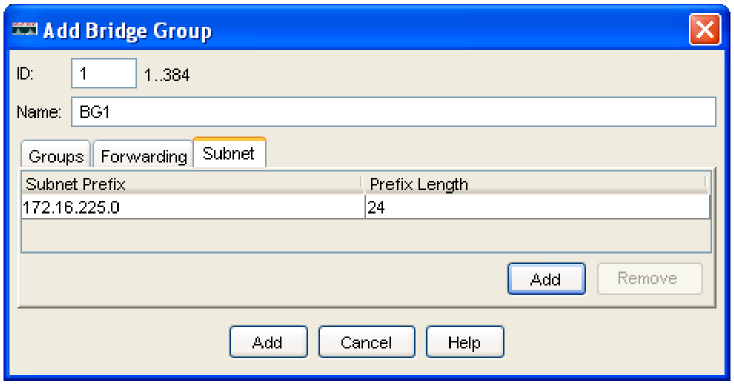
> **Tip:** Make sure the Port Type is set to *ib* or this will not work as a default gateway for the InfiniBand clients.

> **Tip:** It is important not to confuse the default gateway of the bridge group with the default gateway for the Ethernet management of the 3012. In almost all cases, these will be completely different and not interact. On that note, you normally do not want to put the management IP subnet of the 3012 itself on the same subnet as what might be used by the InfiniBand clients and this bridge group.

– Destination address of 0.0.0.0 = all routes, This makes it a default gateway rather than a simple static route.

– Next Hop is the default gateway on the Ethernet network, and in this example represents a routed interface on the 4948 that is acting as the default gateway for this IP subnet.

– While in our example we are setting up a default gateway, you can also use the Forwarding tab to create static routes to specific subnets. Usually a default gateway is sufficient, but some users might need to do static routes to networks not known by the default gateway (again, this is not normally the case, but it is an option that can be entered in this tab if desired).



ID: 1 1..384

Name: BG1

Subnet Prefix	Prefix Length
172.16.225.0	24

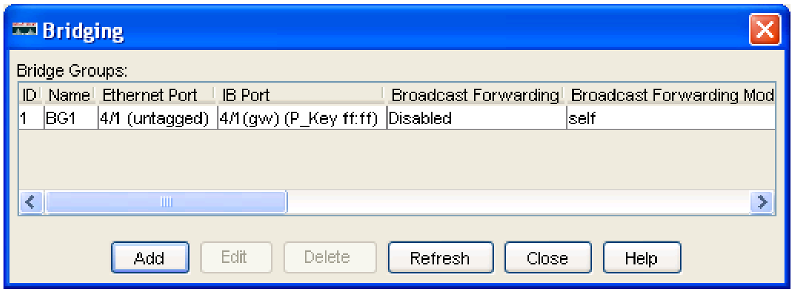
Figure 7-9 Subnet tab on Add Bridge Group window

In the example in Figure 7-9, we define the Subnet for this bridge group. Some comments on these settings:

- 172.16.225.0 is the subnet being used for this bridge group (based on us using a 24-bit mask of 255.255.255.0.
- Because we are using 24-bit mask, this is the prefix length for this subnet.
- In our example, we only have a single IP subnet using this bridge group, and that subnet is what we entered into the Subnet tab. There might be situations where there are more than a single IP subnet on a single p_key/VLAN. This is not common, but it is seen from time to time. If you do want to have more than a single IP subnet in this situation, you can enter the information for that subnet in the Subnet tab. Up to eight subnets are supported in a bridge group (but in most cases, there will only be a single entry in this tab).

> **Tip:** It is important not to confuse the entries in the Subnet tab with having InfiniBand clients using different p_keys, VLANs and IP subnets. If you have InfiniBand clients using different p_key's, VLANs and IP subnets, you would accommodate this by creating another bridge group for these clients.

4. After you have completed the three tabs in the Add Bridge Group window, click **Add** at the bottom of the window. The commands are executed on the 3012 to provision this bridge group with the defined characteristics and the bridge group information as shown in Figure 7-10.



Bridge Groups:

ID	Name	Ethernet Port	IB Port	Broadcast Forwarding	Broadcast Forwarding Mod
1	BG1	4/1 (untagged)	4/1 (gw) (P_Key ff:ff)	Disabled	self

Figure 7-10 Bridge group created

5. The final step on the 3012 is to save this configuration to NVRAM.

   Note: As mentioned previously, all changes done through the GUI take effect immediately and are placed in the running config, but the changes in the running config are not explicitly saved to NVRAM until you perform this step.

   From the main page of the Element Manager, click **Maintenance** → **Save Config** to save the configuration to NVRAM.

6. After saving the configuration of the 3012 to NVRAM per these instructions, we must now configure the 4948. In this example, we connect the Ethernet gateway module port 4/1 to the 4948 on port gi1/16.

   Note that all of these steps might or might not be necessary in your environment. Contact your network engineer if you are not sure about what you need to do for this to work.

   Tip: Although we selected VLAN 88 for use for the port on the 4948 facing the gateway module, this could have been any VLAN, including VLAN 1. VLAN 88 was just used as an example for this section. Since the gateway is sending untagged packets, and we will configure port gi1/16 for access mode, then only untagged packets will be sent between the 4948 and the gateway module, and when the untagged packet is received on port gi1/16, it will be placed on to VLAN 88 inside the 4948. What ever VLAN you choose, it should be the correct VLAN for the IP subnet that is being used by the InfiniBand clients.

   Note that it is assumed that the VLAN interface has already been created and is configured as desired for this lab. Just in case, Example 7-1 shows how to create and configure the management interface for VLAN 88 for this chapter.

Example 7-1 Creating the management interface if not already done

```
! Assumes starting from enable mode, enter config mode
conf t
! Create the VLAN to be used
vlan 88
! Create and configure the management interface for this VLAN
int vlan 88
ip address 172.16.225.250 255.255.255.0
no shut
end
write
```

After logging into the 4948 with sufficient privileges, execute the commands as shown in Example 7-2 on the 4948. Items starting with an exclamation mark (!) are for reference or comment only and are not executed.

Example 7-2 Configure the 4948 to accept untagged packets from the gateway

```
! Assumes starting from enable mode, enter config mode
conf t
! Configure the gateway facing interface for mode access, vlan 88
int gi1/16
description Connection to 3012 Ethernet Switch Module port 4/1
switchport mode access
switchport access vlan 88
! Enable portfast to bring the port up quickly
spanning-tree portfast
no shut
! Exit config mode
end
! Save config to NVRAM
write
```

> **Tip:** Example 7-2 is for a switch running IOS. If the upstream switch is running CatOS, the configuration would be different, CatOS configuration examples are not included in this document.

7. After the changes have been made to both the Ethernet gateway module and the 4948, the InfiniBand client at 172.16.225.10 can successfully ping the VLAN 88 default gateway IP address (172.16.225.250) on the 4948 as demonstrated in Example 7-3.

Example 7-3 InfiniBand host pinging the default gateway

```
[root@localhost ~]# ping -c 5 172.16.225.250
PING 172.16.225.250 (172.16.225.250) 56(84) bytes of data.
64 bytes from 172.16.225.250: icmp_seq=1 ttl=255 time=0.298 ms
64 bytes from 172.16.225.250: icmp_seq=2 ttl=255 time=0.279 ms
64 bytes from 172.16.225.250: icmp_seq=3 ttl=255 time=0.274 ms
64 bytes from 172.16.225.250: icmp_seq=4 ttl=255 time=0.266 ms
64 bytes from 172.16.225.250: icmp_seq=5 ttl=255 time=0.282 ms

--- 172.16.225.250 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4000ms
rtt min/avg/max/mdev = 0.266/0.279/0.298/0.023 ms
[root@localhost ~]#
```

7.3.5 CLI reference for this section

This section includes what the appropriate CLI looks like after the steps were executed in the GUI to create this bridge group, as well as after the changes made to the 4948. It is a useful reference for those wanting to understand the CLI better, or for those that simply want to use the CLI and not the GUI to achieve this task.

Example 7-4 is the CLI of the 3012 after bridge group BG1 was created and fully operational.

Example 7-4 3012 - after BG1 added

```
3012-1# show config
...
bridge-group 1 subnet-prefix 172.16.225.0 24
bridge-group 1 ib-next-hop 172.16.225.250
bridge-group 1 name "BG1"
!
interface gateway 4/1
 bridge-group 1 pkey ff:ff
!
interface Ethernet 4/1
 bridge-group 1
...
```

Example 7-5 is the CLI of the 4948 after changes to support this bridge group.

Example 7-5 4948 - after support for BG1 added

```
4848-1#show run
...
interface GigabitEthernet1/16
 description Connection to 3012 Ethernet Switch Module port 4/1
 switchport access vlan 88
 switchport mode access
 spanning-tree portfast
...
interface Vlan88
 ip address 172.16.225.250 255.255.255.0
 no ip route-cache
...
```

7.4 Using a VLAN tagged bridge group design

This section demonstrates using VLAN tagged packets on the Ethernet uplink of the gateway module. It is assumed we are starting from scratch (configurations from the last example have been deleted).

As discussed previously, VLANs are used on some networks to aid in reducing the size of the L2 broadcast domain and to provide isolation of data flows between different VLAN packets.

Note that the design example shown in this section is not necessarily realistic, in that you would usually implement tagged packets if you wanted to carry more than a single VLAN, and we are only showing a single VLAN, albeit a tagged one. Setting up multiple VLANs is considered beyond the scope of this document.

7.4.1 Some comments on VLANs and the Ethernet gateway module

When discussing VLANs and the Ethernet gateway module, remember:

- Each Ethernet gateway module supports up to 32 VLANs
- Ethernet bridge module ports can be tagged or untagged
- Standard 802.1Q VLANs are supported
- Static port based VLANs are supported
- A full range of VLAN IDs are supported
- One VLAN is mapped to one InfiniBand partition (unique p_key)
- Separate multicast groups are created per InfiniBand partition (for multiple VLANs)

Figure 7-11 represents the configuration that we use to demonstrate using VLAN tagging on the uplink. Note that by creating multiple bridge groups, each using a different p_key on the InfiniBand side, and different VLAN tagging on the Ethernet side, and then associating them to the same uplink or uplinks, it is possible to carry multiple VLANs on the same link or links between the gateway module and the upstream switch. (We do not demonstrate this in this section.)

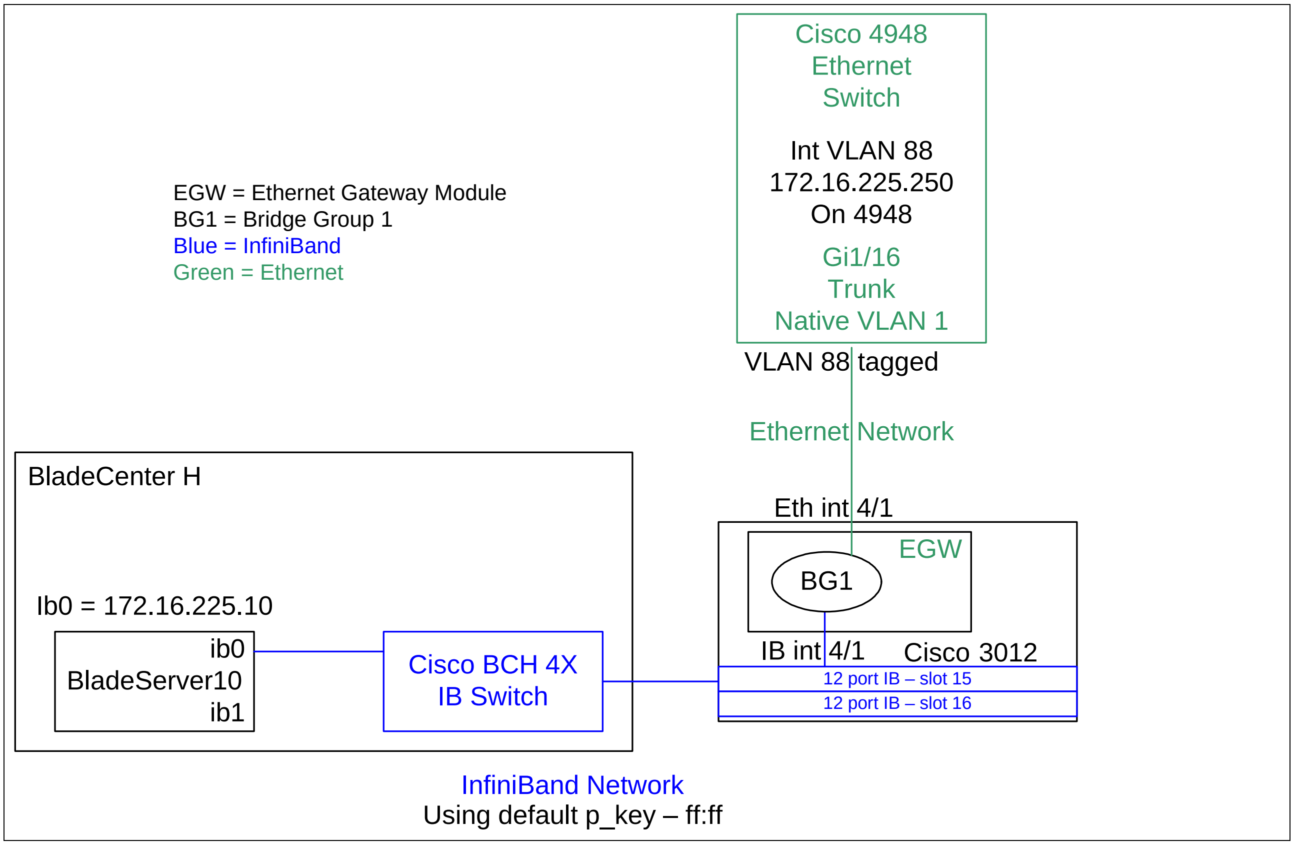


Figure 7-11 Example for using tagged VLAN88

Assuming previous configurations have been removed, we can begin the configuration for this section.

7.4.2 Summary of steps to create a tagged VLAN bridge group design

Here, we provide a summary of the steps that we describe in the remainder of this section to implement a tagged VLAN bridge group design:

1. In the 3012 Element Manager, bring up the Bridging window and fill in the Groups, Forwarding and subnet tabs (step 1 on page 134).
2. Commit the new Bridge Group information and review (step 2 on page 135).
3. Save the 3012 config to NVRAM (step 3 on page 135).
4. Configure the upstream 4948 to match the Ethernet Gateway configuration (step 4 on page 135).
5. Test the configurations by pinging from the InfiniBand client to the default gateway on the 4948 (step 5 on page 136).

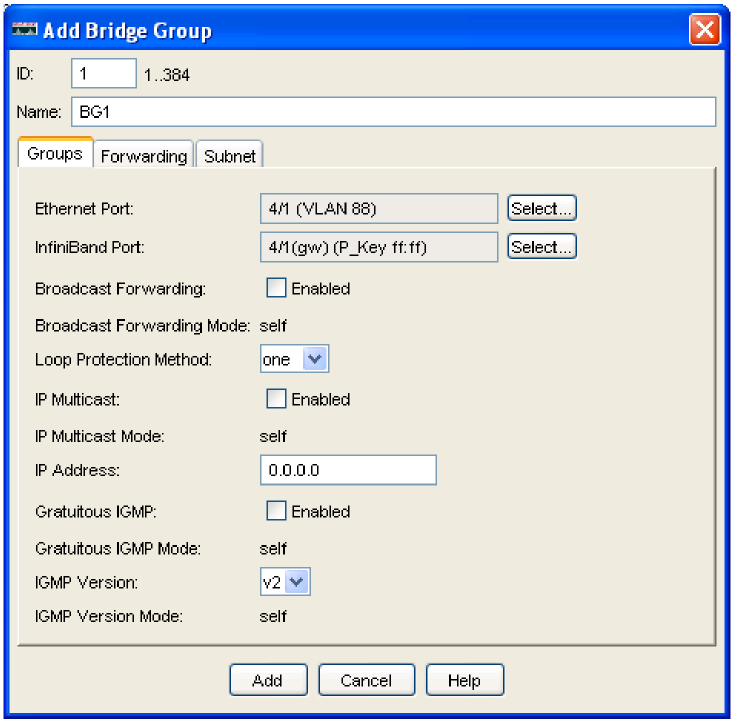
7.4.3 Detailed steps to implement a tagged VLAN bridge group design

To implement a tagged VLAN bridge group design, follow these detailed steps:

1. On the Element Manager window, click **Ethernet** → **Bridging**. Follow the steps in the previous example (Figure 7-5 on page 125), but when you select the Ethernet members, enter 88 into the VLAN field and then click **OK**.

   Make sure to also configure the Forwarding tab and the Subnet tab as per the previous examples in Figure 7-8 on page 128 and Figure 7-9 on page 129.

   Figure 7-12 shows the new BG1 bridge group, tagged VLAN 88, ready for saving.

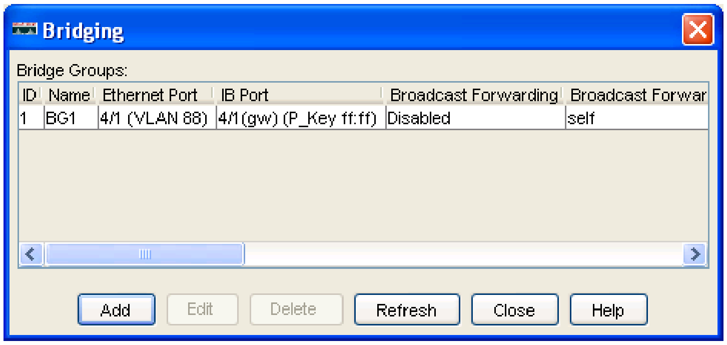


ID: 1 1..384
Name: BG1

Ethernet Port: 4/1 (VLAN 88)
InfiniBand Port: 4/1(gw) (P_Key ff:ff)
Broadcast Forwarding: □ Enabled
Broadcast Forwarding Mode: self
Loop Protection Method: one
IP Multicast: □ Enabled
IP Multicast Mode: self
IP Address: 0.0.0.0
Gratuitous IGMP: □ Enabled
Gratuitous IGMP Mode: self
IGMP Version: v2
IGMP Version Mode: self

Figure 7-12 Bridge group BG1 configured to use VLAN 88

2. After you have configured all of the tabs of the Add Bridge Group window, click **Add** to create the new bridge group BG1 and make it active. Figure 7-13 shows the bridge group with tagged VLAN 88 selected and ready for use.



Bridge Groups:

ID	Name	Ethernet Port	IB Port	Broadcast Forwarding	Broadcast Forwar
1	BG1	4/1 (VLAN 88)	4/1(gw) (P_Key ff:ff)	Disabled	self

Figure 7-13 BG1 ready to use tagged packets on VLAN 88

3. The final step on the 3012 is to save this configuration to NVRAM

> **Note:** As mentioned previously, all changes done through the GUI take effect immediately and are placed in the running config, but the changes in the running config are not explicitly saved to NVRAM until you perform this step.

From the main page of the Element Manager, click **Maintenance** → **Save Config** to save the configuration to NVRAM.

4. After saving the configuration of the 3012, you need to configure the 4948. In this example, we use the same connection that we used in the previous section, connecting the Ethernet gateway module to the 4948 on port gi1/16.

   As noted in the previous section, VLAN 88 was an arbitrary choice. The VLAN that you choose should be the same IP subnet that is desired by the InfiniBand clients.

   On the 4948 side, the only way to have a port support tagged frames is to set the port or ports into trunked mode. Furthermore, Cisco switches have the concept of a single untagged VLAN, called the Native VLAN. Any untagged packet coming into a trunk port will be placed on this Native VLAN (default VLAN 1, but can be changed).

   Because the bridge group is now using tagged packets for VLAN 88, we must make sure VLAN 88 is not the native VLAN. To restate this, this means that what ever VLAN tag we set on the Ethernet gateway module, cannot be the native VLAN on the upstream Cisco switch. To address this on the 4948, we define the connecting ports as a trunk, leave the native VLAN at 1, and permit VLAN 88 on this trunk (default is all VLANs permitted, but best practices usually say to limit the allowed VLANs to only those that are needed). In doing this, since the packets coming up from the gateway module will be tagged with VLAN 88, the 4948 port will place them on VLAN 88 inside the 4948, and packets can then get to the desired subnet.

   Contact your network engineer if you are not sure about what you need to do for this to work.

   Note that it is assumed that VLAN interface already has been created and is configured as desired for this lab. Just in case, Example 7-1 on page 130 shows how to configure the management interface for VLAN 88.

After logging into the 4948 with sufficient privileges, execute the commands as shown in Example 7-6 on the 4948 (items starting with an exclamation mark (!) are for reference/comments only and will not be executed).

Example 7-6 Configure the 4948 to accept packets tagged with VLAN 88

```
! Assumes starting from enable mode, enter config mode
conf t
! Change int gi1/16 to support tagging for VLAN 88
int gi1/16
description Connection to 3012 Ethernet Switch Module port 4/1 - VLAN 88
! Tell the port to use 802.1Q tagging encapsalation
switchport trunk encapsulation dot1q
! Force the port to trunk mode to support multiple VLANs and tagging
switchport mode trunk
! Restrict VLANs to this single VLAN (can carry more if desired)
switchport trunk allowed vlan 88
! Change portfast to work in trunk mode
spanning-tree portfast trunk
! Exit config mode
end
! Save config to NVRAM
write
```

5. When the changes have been made to both the Ethernet gateway module and the 4948, the InfiniBand client at 172.16.225.10 can successfully ping the VLAN 88 default gateway IP address (172.16.225.250) on the 4948 as shown in Example 7-7.

Example 7-7 InfiniBand host pinging the default gateway

```
[root@localhost ~]# ping -c 5 172.16.225.250
PING 172.16.225.250 (172.16.225.250) 56(84) bytes of data.
64 bytes from 172.16.225.250: icmp_seq=1 ttl=255 time=0.303 ms
64 bytes from 172.16.225.250: icmp_seq=2 ttl=255 time=0.267 ms
64 bytes from 172.16.225.250: icmp_seq=3 ttl=255 time=0.289 ms
64 bytes from 172.16.225.250: icmp_seq=4 ttl=255 time=0.335 ms
64 bytes from 172.16.225.250: icmp_seq=5 ttl=255 time=0.291 ms

--- 172.16.225.250 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4000ms
rtt min/avg/max/mdev = 0.267/0.297/0.335/0.022 ms
[root@localhost ~]#
```

7.4.4 CLI reference for this section

This section includes what the appropriate CLI looks like after the steps were executed in the GUI to create this bridge group, as well as after the changes made to the 4948. It is a useful reference for those wanting to understand the CLI better, or for those that simply want to use the CLI and not the GUI to achieve this task.

Note that we include only the commands that effect the area of the device that we configured. We do not include the entire device configuration.

Example 7-8 is the CLI of the 3012 creating a new BG1 to use VLAN tagging on VLAN 88.

Example 7-8 3012 - after VLAN 88 tagging added

```
3012-1# show config
...
bridge-group 1 subnet-prefix 172.16.225.0 24
bridge-group 1 ib-next-hop 172.16.225.250
bridge-group 1 name "BG1"
!
interface gateway 4/1
 bridge-group 1 pkey ff:ff
!
interface Ethernet 4/1
 bridge-group 1 vlan-tag 88
!
```

Example 7-9 is the CLI of the 4948 after changes to support tagged VLAN 88 packets.

Example 7-9 4948 - after support for tagged VLAN 88 configured

```
4848-1#show run
...
interface GigabitEthernet1/16
 description Connection to 3012 Ethernet Switch Module port 4/1 - VLAN 88
 switchport access vlan 88
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 88
 switchport mode trunk
 spanning-tree portfast trunk
!...
interface Vlan88
 ip address 172.16.225.250 255.255.255.0
 no ip route-cache
...
```

7.5 Using Ethernet aggregation on the uplinks of the Ethernet gateway module

As noted briefly in 7.1, “Introduction to the Ethernet gateway module” on page 118, the Ethernet gateway module supports the ability to aggregate (also known as EtherChannel or Trunking) multiple links on the same gateway module, to provide for added bandwidth and increased high availability.

7.5.1 Some facts about this feature with this gateway module

When using Ethernet aggregation with this gateway module, remember:

- Up to six ports can be aggregated together. They must be on the same gateway module (you cannot aggregate between different modules in the same or other chassis).
- A link aggregation group can be assigned one bridge group or multiple bridge groups.
- VLAN tagging is supported with this feature.

- Seven different types of frame distribution algorithms are supported with this product, as shown in Table 7-1.

Table 7-1 Load balancing options for link aggregation

Distribution	Description
dst-ip	Load distribution is based on the destination IP address. Packets to the same destination are sent on the same port, but packets to different destinations are sent on different ports in the channel
dst-mac	InfiniBand hosts do not have a MAC address, so load distribution is based on the LID address of the InfiniBand node and MAC address of Ethernet node. Packets to the same destination are sent on the same port, but packets to different destinations are sent on different ports in the channel
src-dst-ip	Load distribution is based on the source logic gate (XOR) destination IP address.
src-dst-mac	InfiniBand hosts do not have a MAC address, so load distribution is based on the source logic gate (XOR) LID and MAC address. This is the default method for the gateway module if none is specified
src-ip	Load distribution is based on the source IP address. Packets to the same destination are sent on the same port, but packets to different destinations are sent on different ports in the channel
src-mac	InfiniBand hosts do not have a MAC address, so load distribution is based on the source-LID address of the incoming packet. Packets from different hosts use different ports in the channel, but packets from the same host use the same port in the channel
round-robin	Round-robin is a load balancing algorithm that distributes load in a circular fashion, thereby creating an evenly distributed load. When using redundancy groups and load balancing, selecting the round-robin distribution can increase performance in many cases. Even with a topology that contains as few as one Ethernet host, the performance could benefit from using this distribution type.

- These algorithms only impact outbound traffic from the gateway module on the Ethernet aggregation. Packets inbound from the upstream switch will be load balanced across the ports according to whatever algorithm is in use on the upstream device.
- Which method to choose is almost a trial and error routine, in that there is not one mode that is perfect for all environments. With that said, the default method (src-dst-mac in Table 7-1 on page 138) is usually a good choice to try first.
- To check the efficiency of load balancing, look a the port counters on the ports in the aggregation to see how evenly they get utilized. Again, perfect load balancing is not really possible, especially since traffic patterns have a habit of changing.

Tip: We recommend leaving the load balance method as default, as in most cases, it will be as likely to produce good load balance as any other.

Tip: During testing of this feature it was found that only static aggregation would work with the Ethernet gateway module. This means on an upstream IOS Cisco switch that is attempting to aggregate to this device, you will need to use *channel-group X mode on*. Some of the stand alone documentation for this product indicated that it supported IEEE 802.3ad, but attempting to use *mode active* always resulted in the EtherChannel ports going in to stand-alone mode.

Configuring aggregation must be done on both sides of the links (in the Ethernet gateway module and the 4948). If you only configure one side, it might or might not work, and unexpected or undesired results will more than likely occur.

Figure 7-14 shows the configuration we will be using to demonstrate aggregation.

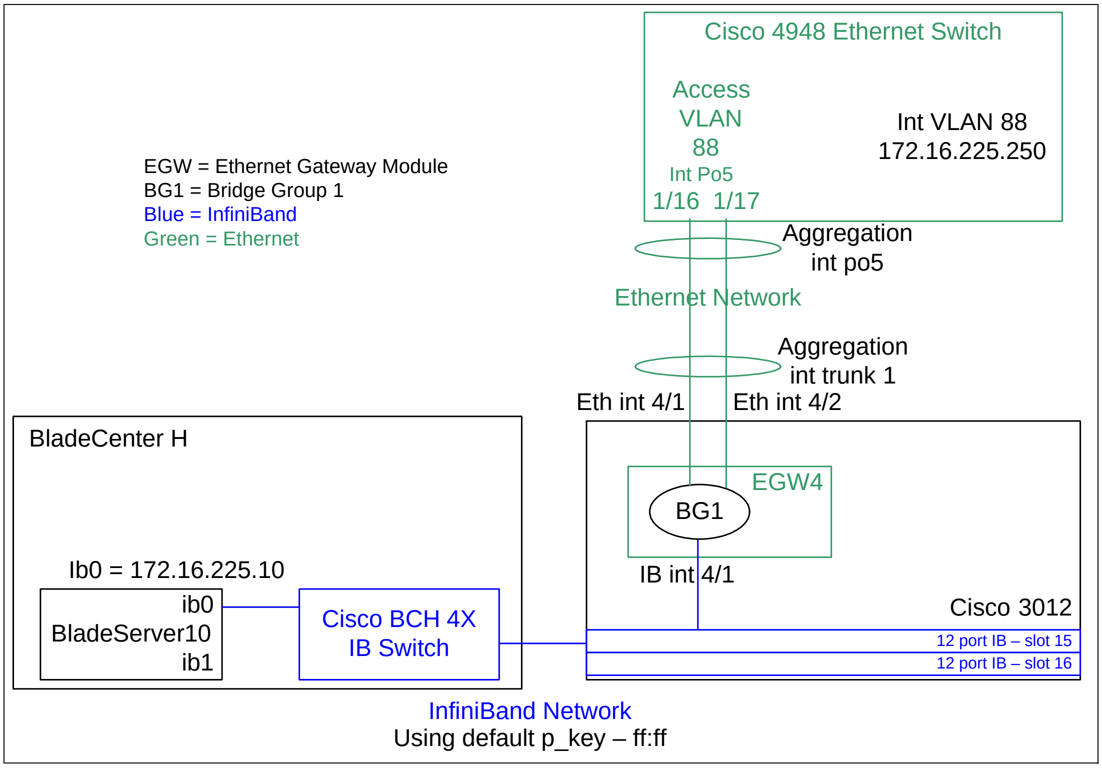


Figure 7-14 Network design used to demonstrate link aggregation

7.5.2 Summary of steps to create a bridge group using a trunked connection

Tip: It is always a good idea to shut down the ports or disconnect the cables between the Ethernet gateway module and the upstream switch while making changes to aggregation. Failure to do this can result in temporary loops in the network, which can cause a disruption of the network traffic.

Here, we provide a summary of the steps that we describe in the remainder of this section to implement a bridge group using a trunked connection:

1. Make sure connections between the 3012 and the 4948 are down (step 1 on page 140).
2. Remove any old configurations from the last example (step 2 on page 140).
3. Use Element Manager to open up the Trunking window (step 3 on page 140).
4. Create the trunk group and add desired ports to the trunk (step 4 on page 141).
5. Create the new bridge group and add the new trunk group as the Ethernet connection (step 5 on page 142).
6. Save the 3012 config to NVRAM (step 6 on page 142).
7. Configure the upstream 4948 to match the Ethernet Gateway configuration (step 7 on page 143).

8. Re-attach or bring up the links between the 3012 and the 4948 (step 8 on page 143).
9. Test the configurations by pinging from the InfiniBand client to the default gateway on the 4948 and remove links from the trunk one at a time to ensure trunk is working (step 9 on page 143).

7.5.3 Detailed steps to implement a trunked bridge group design

To implement a trunked bridge group design, follow these detailed steps:

1. Make sure the cables between the module and the upstream switch are either unplugged or logically shut down. This is to insure we do not introduce any temporary spanning tree loops which could disrupt network communications.
2. It is assumed that you are starting from scratch, so delete any bridge groups that might exist from the previous examples.
3. Using the Element Manager, click **Ethernet** → **Trunking** as shown in Figure 7-15.

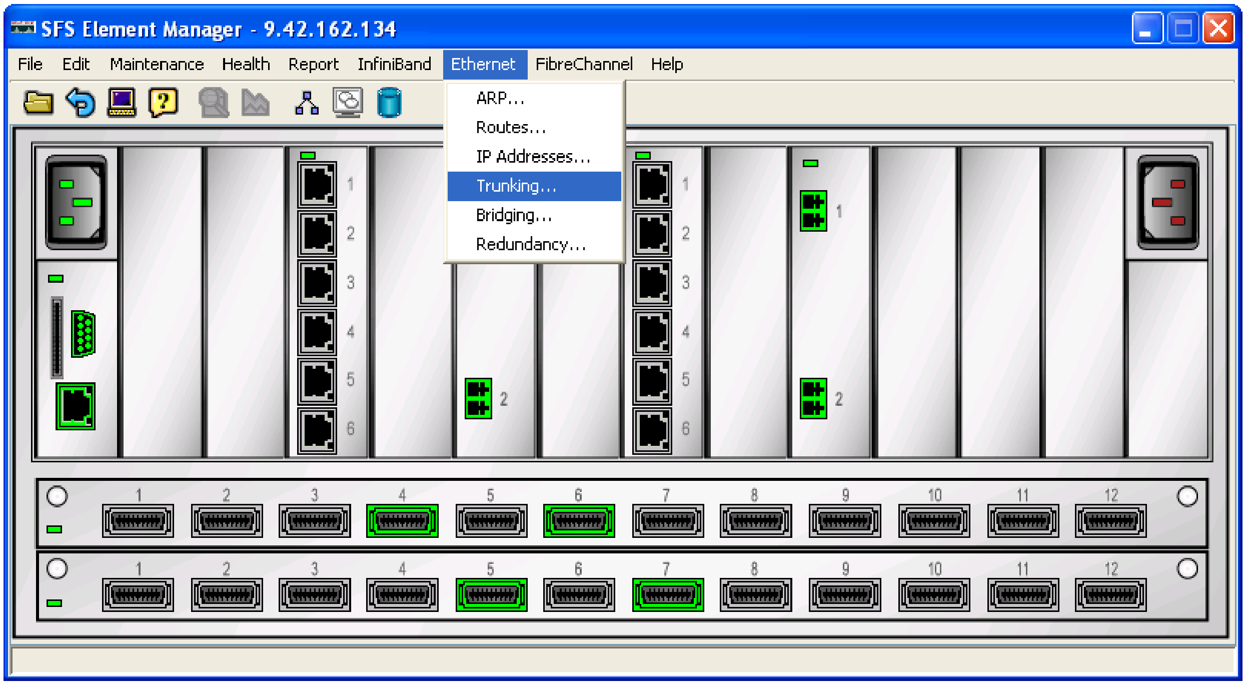

SFS Element Manager - 9.42.162.134
File Edit Maintenance Health Report InfiniBand Ethernet FibreChannel Help
ARP...
Routes...
IP Addresses...
Trunking...
Bridging...
Redundancy...

Figure 7-15 Begin the process of creating a trunk group

Figure 7-16 opens.

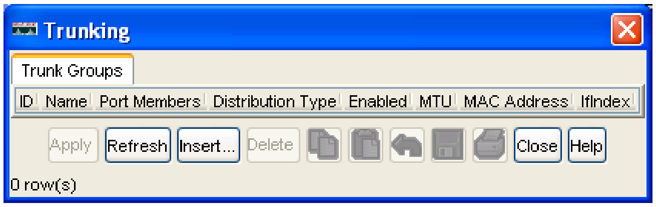
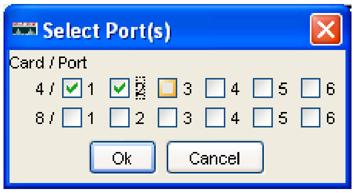

Trunking
Trunk Groups
ID Name Port Members Distribution Type Enabled MTU MAC Address IfIndex
Apply Refresh Insert... Delete Close Help
0 row(s)

Figure 7-16 Ethernet Trunking (aggregation) with no entries

4. Click **Insert**, select a trunk port number to assign (trunk group 1 in this example) and give it a name (TG1 in this example), then click the option to select port members and select the port member numbers (4/1 and 4/2 in this example). See Figure 7-17.

> **Tip:** Port channel numbers and trunk group numbers are locally significant only. In other words, when creating aggregations, the port channel number on the upstream switch and the trunk number on the gateway module can be different on each side of the link. So for example, the Ethernet gateway module can use Trunk group 1 and the upstream switch can use aggregation channel-group 5. This is how we demonstrate aggregation in this example.



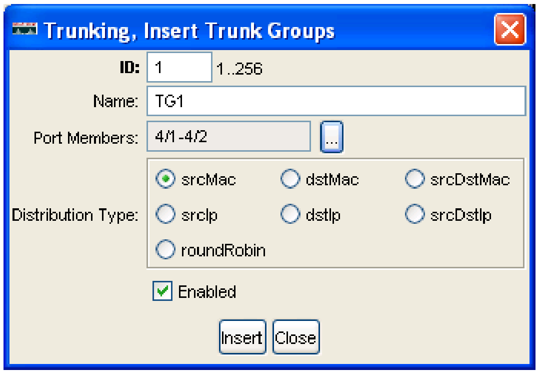
Card / Port

4 / ☑ 1 ☑ 2 □ 3 □ 4 □ 5 □ 6

8 / □ 1 □ 2 □ 3 □ 4 □ 5 □ 6

Figure 7-17 Selecting members in a trunk group

Figure 7-18 shows the settings that we selected for this example.



ID: 1 1..256

Name: TG1

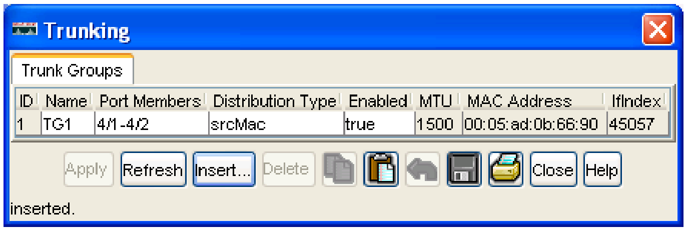
Port Members: 4/1-4/2

Distribution Type: ● srcMac ○ dstMac ○ srcDstMac ○ srcIp ○ dstIp ○ srcDstIp ○ roundRobin

☑ Enabled

Figure 7-18 Selecting trunk group options for this example

After you have configured options as desired in the Trunking, Insert Trunk Groups window, click **Insert** to create the new Trunk group. A window similar to Figure 7-19 shows the trunk group ready to use.



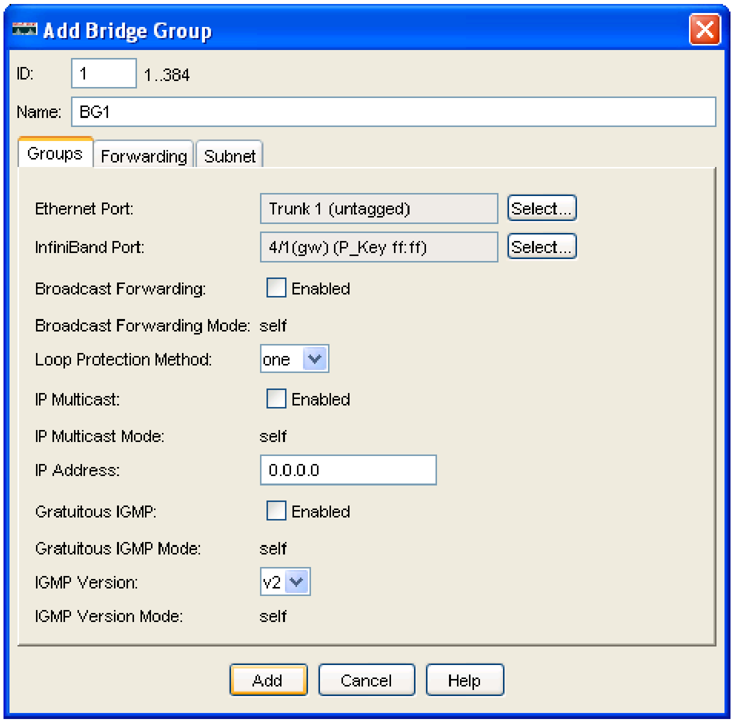
ID	Name	Port Members	Distribution Type	Enabled	MTU	MAC Address	IfIndex
1	TG1	4/1-4/2	srcMac	true	1500	00:05:ad:0b:66:90	45057

inserted.

Figure 7-19 Trunk group created and operational

5. After configuring the trunk group (TG1) to use the gateway module Ethernet ports 4/1 and 4/2, recreate the bridge group BG1 and assign trunk group TG1 as its Ethernet member. Follow the steps in starting at step 1 on page 125, but instead of selecting a physical port for the Ethernet connection, select *trunk 1* as the Ethernet port, as seen in Figure 7-20.

   Remember to also configure the Forwarding and Subnet tabs per the instructions starting at step 1 on page 125.



ID: 1 1..384

Name: BG1

Ethernet Port: Trunk 1 (untagged)

InfiniBand Port: 4/1(gw) (P_Key ff:ff)

Broadcast Forwarding: □ Enabled

Broadcast Forwarding Mode: self

Loop Protection Method: one

IP Multicast: □ Enabled

IP Multicast Mode: self

IP Address: 0.0.0.0

Gratuitous IGMP: □ Enabled

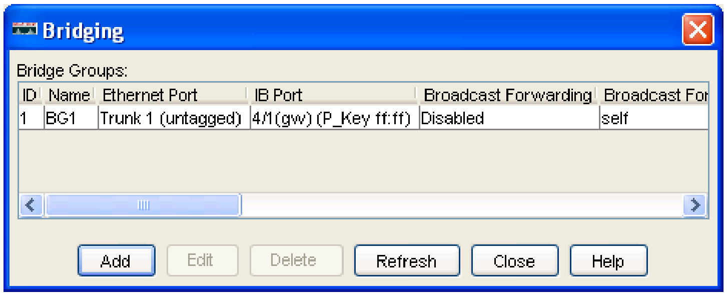
Gratuitous IGMP Mode: self

IGMP Version: v2

IGMP Version Mode: self

Figure 7-20 BG1 set to use the trunk group ID 1 (TG1)

Clicking **Add** at this point results in a new bridge group being created, using the new trunk group on its upstream connections to the Ethernet network (as seen in Figure 7-21).



Bridge Groups:

ID	Name	Ethernet Port	IB Port	Broadcast Forwarding	Broadcast For
1	BG1	Trunk 1 (untagged)	4/1(gw) (P_Key ff:ff)	Disabled	self

Figure 7-21 Bridge group BG1 using trunk group TG1 operational

6. The final step on the 3012 is to save this configuration to NVRAM.

> **Note:** As mentioned previously, all changes done through the GUI take effect immediately and are placed in the running config, but the changes in the running config are not explicitly saved to NVRAM until you perform this step.

From the main page of the Element Manager, click **Maintenance** → **Save Config** to save the configuration to NVRAM.

7. After the 3012 configuration is saved, you need to configure the 4948. In this example, we use ports gi1/16 and gi1/17 on the 4948 to connect to TG1 on the Ethernet gateway module.

   On the 4948 side, we use static aggregation, because this is the only mode that works correctly with the Ethernet gateway module.

   Contact your network engineer if you are not sure what you need to do for this to work.

   After logging into the 4948 with sufficient privileges, execute the commands as shown in Example 7-10 on the 4948. Items that start with an exclamation mark (!) are for reference/comments only and will not be executed.

   Example 7-10 Configure the 4948 to use aggregation

```
! Assumes starting from enable mode, enter config mode
conf t
! Change int gi1/16 and 17 to the desired config
int range gi1/16 -17
description Connection to 3012 ESM - 4/1 - 4/2 trunk group TG1
! Force the ports back to access and remove any tagging from last section
switchport mode access
no switchport trunk encapsulation dot1q
no switchport trunk allowed vlan 88
! make sure both ports are set to VLAN 88 and have portfast enabled
switchport access vlan 88
spanning-tree portfast
! Create the new aggragation and make it static (mode on)
channel-group 5 mode on
no shut
! 4948 will create a new interface called int portchannel5 (po5 for short))
! Confoigure po5 to match ports 16 and 17
int po5
 switchport access vlan 88
 switchport mode access

! Exit config mode
end
! Save config to NVRAM
write
```

   Tip: While it is possible to directly create the port channel interface with the command **`int po5`**, doing this could confuse the port channel and result in undesired operation. The best way to create a port channel is to run the channel-group command on the physical interface as shown in Example 7-10, and let that create the port channel interface automatically.

8. After the changes have been made to both the Ethernet gateway module and the 4948, reattach the cables or re-enable the ports.

9. When the ports are up and ready, the InfiniBand client at 172.16.225.10 should be able to ping the VLAN 88 default gateway IP address (172.16.225.250) on the 4948. It is possible to verify the trunk group is working by removing first one, and then the other cable (one at

a time) and make sure pings continue to work, regardless of if we have either cable plugged in, or both cables plugged in.

In Example 7-11, we lost a total of three pings while pulling and adding cables. Ping packet 9 was lost when we pulled the first cable, ping packets 21 and 22 were lost when we reinserted the first cable and pulled the second cable.

Example 7-11 InfiniBand host pinging the default gateway while cables being pulled and re-added

```
[root@localhost ~]# ping  172.16.225.250
PING 172.16.225.250 (172.16.225.250) 56(84) bytes of data.
64 bytes from 172.16.225.250: icmp_seq=1 ttl=255 time=0.290 ms
64 bytes from 172.16.225.250: icmp_seq=2 ttl=255 time=0.270 ms
64 bytes from 172.16.225.250: icmp_seq=3 ttl=255 time=0.277 ms
64 bytes from 172.16.225.250: icmp_seq=4 ttl=255 time=0.301 ms
64 bytes from 172.16.225.250: icmp_seq=5 ttl=255 time=0.269 ms
64 bytes from 172.16.225.250: icmp_seq=6 ttl=255 time=0.331 ms
64 bytes from 172.16.225.250: icmp_seq=7 ttl=255 time=0.271 ms
64 bytes from 172.16.225.250: icmp_seq=8 ttl=255 time=0.291 ms
64 bytes from 172.16.225.250: icmp_seq=10 ttl=255 time=0.344 ms
64 bytes from 172.16.225.250: icmp_seq=11 ttl=255 time=0.359 ms
64 bytes from 172.16.225.250: icmp_seq=12 ttl=255 time=0.253 ms
64 bytes from 172.16.225.250: icmp_seq=13 ttl=255 time=0.263 ms
64 bytes from 172.16.225.250: icmp_seq=14 ttl=255 time=0.273 ms
64 bytes from 172.16.225.250: icmp_seq=15 ttl=255 time=0.338 ms
64 bytes from 172.16.225.250: icmp_seq=16 ttl=255 time=0.326 ms
64 bytes from 172.16.225.250: icmp_seq=17 ttl=255 time=0.340 ms
64 bytes from 172.16.225.250: icmp_seq=18 ttl=255 time=0.337 ms
64 bytes from 172.16.225.250: icmp_seq=19 ttl=255 time=0.726 ms
64 bytes from 172.16.225.250: icmp_seq=20 ttl=255 time=0.322 ms
64 bytes from 172.16.225.250: icmp_seq=23 ttl=255 time=0.292 ms
64 bytes from 172.16.225.250: icmp_seq=24 ttl=255 time=0.323 ms
64 bytes from 172.16.225.250: icmp_seq=25 ttl=255 time=0.339 ms
64 bytes from 172.16.225.250: icmp_seq=26 ttl=255 time=0.333 ms
64 bytes from 172.16.225.250: icmp_seq=27 ttl=255 time=0.302 ms
64 bytes from 172.16.225.250: icmp_seq=28 ttl=255 time=0.329 ms
64 bytes from 172.16.225.250: icmp_seq=29 ttl=255 time=0.264 ms

--- 172.16.225.250 ping statistics ---
29 packets transmitted, 26 received, 10% packet loss, time 28004ms
rtt min/avg/max/mdev = 0.253/0.321/0.726/0.088 ms
[root@localhost ~]#
```

7.5.4 CLI reference for this section

This section includes what the appropriate CLI looks like after the steps were executed in the GUI to create TG1, as well as after the changes made to the 4948. It is a useful reference for those wanting to understand the CLI better, or for those that simply want to use the CLI and not the GUI to achieve this task.

Note that we only include the commands that effect the area of the device we configured. We do not include the entire device configuration.

Example 7-12 is the CLI of the 3012 after creating trunk group TG1 and adding it to bridge group BG1.

Example 7-12 3012 - after changing to untagged and adding trunk group

```
3012-1# show config
...
bridge-group 1 subnet-prefix 172.16.225.0 24
bridge-group 1 ib-next-hop 172.16.225.250
bridge-group 1 name "BG1"
!
interface trunk 1
 enable
 name "TG1"
 distribution-type src-mac
!
interface gateway 4/1
 bridge-group 1 pkey ff:ff
!
interface Ethernet 4/1
 trunk-group 1
!
interface Ethernet 4/2
 trunk-group 1
!
interface trunk 1
 bridge-group 1
!
```

Example 7-13 is the CLI of the 4948 after changes to support the aggregation po5.

Example 7-13 4948 - after changes to support aggregation

```
4848-1#show run
...
interface Port-channel5
 switchport
 switchport access vlan 88
 switchport mode access
...
interface GigabitEthernet1/16
 description Connection to 3012 ESM - 4/1 - 4/2 trunk group TG1
 switchport access vlan 88
 switchport mode access
 channel-group 5 mode on
 spanning-tree portfast
!
interface GigabitEthernet1/17
 description Connection to 3012 ESM - 4/1 - 4/2 trunk group TG1
 switchport access vlan 88
 switchport mode access
 channel-group 5 mode on
 spanning-tree portfast
...
interface Vlan88
 ip address 172.16.225.250 255.255.255.0
 no ip route-cache
...
```

7.6 Implementing a design using gateway redundancy

In many cases, customers want to make sure the InfiniBand hosts have High Availability (HA) connectivity to the outside world. One element of this HA environment is redundant Ethernet gateway modules to the Ethernet network.

7.6.1 Some considerations for redundancy

When implementing a design using gateway redundancy, remember the following considerations:

- In this example, both gateway modules are in the same 3012. For maximum HA, it would be desirable to have each of the gateway modules in the redundancy group in two separate 3012s. Cross-3012 Ethernet Gateway redundancy is supported, but we are not going to demonstrate it in this redundancy example.
- If placing both gateway modules in the same 3012, it is best to have one gateway module use its internal port to the top InfiniBand module (internal port 1) and the other gateway module use the port to the bottom InfiniBand module (internal port 2). This will permit maximum redundancy

> **Tip:** At a minimum, you should use the InfiniBand port 2 for connection to the InfiniBand fabric. This is because the design of the 3012 is such that the module in slot 15 (InfiniBand port 1 on the Ethernet gateway) will not work if the InfiniBand module in slot 16 is not present or otherwise in a failed state. To state this a different way, an InfiniBand module in slot 16 will work by itself, but an InfiniBand module in slot 15 will only work if there is an InfiniBand module in slot 16.

- The InfiniBand management interface (int mgmt-ib) must be configured and up (the **`no shutdown`** command must be present in the config for the mgmt-ib interface) for redundancy to work. This interface is used by the redundant gateways to keep track of each others operational state
- When using gateway load-balancing (active-active mode), each bridge group must have an IP assigned to it. This is not necessary if using active-standby or non redundant operation

Figure 7-22 represents the test environment for demonstrating gateway redundancy. In this example bridge group 1 (BG1) and bridge group 2 (BG2) will be put into Redundancy Group 1 (RG1).

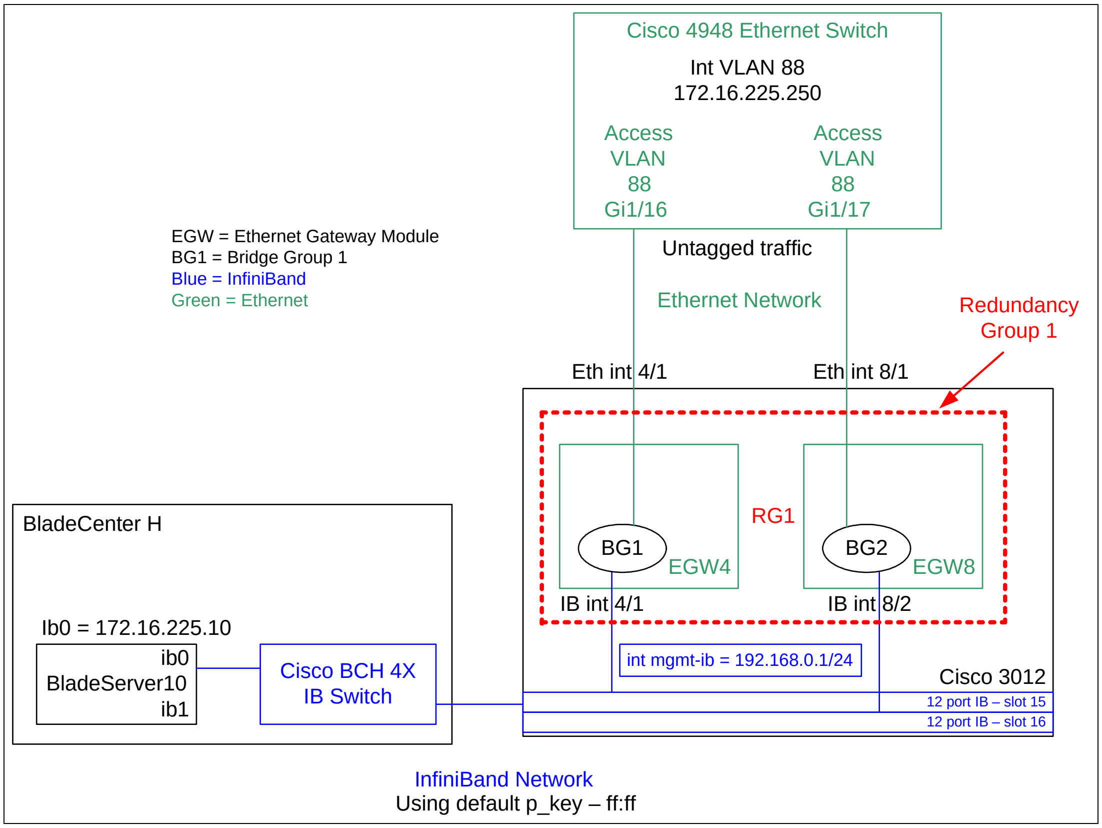


Figure 7-22 Design showing Ethernet gateway redundancy in a single 3012

Note that for full redundancy, there would need to be two BCH InfiniBand switches, both ib0 and ib1 would be used on the servers, the Ethernet gateways would be in two separate 3012s, and there would be two upstream Ethernet switches. Since we are only trying to demonstrate gateway redundancy, these extra elements are missing from our test environment.

In this example, we are going to start from scratch, create two bridge groups and then add them both to a new Redundancy group.

7.6.2 Summary of steps to create a redundant gateway design

Here, we provide a summary of the steps that we describe in the remainder of this section to implement a redundant gateway design:

1. Make sure connections between the 3012 and the 4948 are down (step 1 on page 148).
2. Remove any old configurations from the last example (step 2 on page 148).
3. Add and enable the InfiniBand management interface (step 3 on page 148).
4. Create the bridge groups that are used for this redundant configuration (step 4 on page 149).
5. Open up the Redundancy Group tabs in preparation for creating the Redundancy group (step 5 on page 151).

6. Create the redundancy group using the two previously created bridge groups (step 6 on page 151).
7. Save the 3012 config to NVRAM (step 7 on page 152).
8. Configure the upstream 4948 to match the Ethernet Gateway configuration (step 8 on page 152).
9. Re-attach or bring up the links between the 3012 and the 4948 (step 9 on page 153).
10. Test the configurations by pinging from the InfiniBand client to the default gateway on the 4948 and remove links from the gateway modules one at a time to ensure trunk is working (step 10 on page 153).

7.6.3 Detailed steps to implement a redundant gateway design

To implement a redundant gateway design, follow these detailed steps:

1. Make sure the cables between the module and the upstream switch are either unplugged or logically shut down. This is to ensure that we do not introduce any temporary spanning tree loops that could disrupt network communications.
2. Prior to configuring, delete any trunk group and/or bridge group that might have existed from previous examples (note that you can instead choose to just modify existing groups, but this example does not do this).
3. After you have a clean configuration, you can start configuring the 3012. The first step for configuring the 3012 is to add a management InfiniBand interface and give it an IP address.

> **Tip:** If you do not have an IP address on the mgmt-ib interface, and do not bring it up, redundancy will not pass packets. If individual bridge groups pass traffic just fine, but fail to pass traffic when placed in a redundancy group, check to make sure the mgmt-ip interface is configured with an IP address and in a **`no shutdown`** state.

During the creation of this document, the only way found to set the IP address on the mgmt-ib interface and bring it up was through the CLI. To perform this step, telnet to the management IP address of the 3012, log in (a user ID and password of `super/super` is the default) and then run the commands as show in Example 7-14 on page 148.

Example 7-14 Adding an ib management interface for redundancy

```
conf t
interface mgmt-ib
ip address 192.168.0.1 255.255.255.0
no shutdown
exit
exit
copy running startup
```

The IP address that we used in this example is on a subnet not in use elsewhere on the network, and if you want to use separate 3012 chassis for the gateway redundancy, these addresses need to be on the same IP subnet on each 3012.

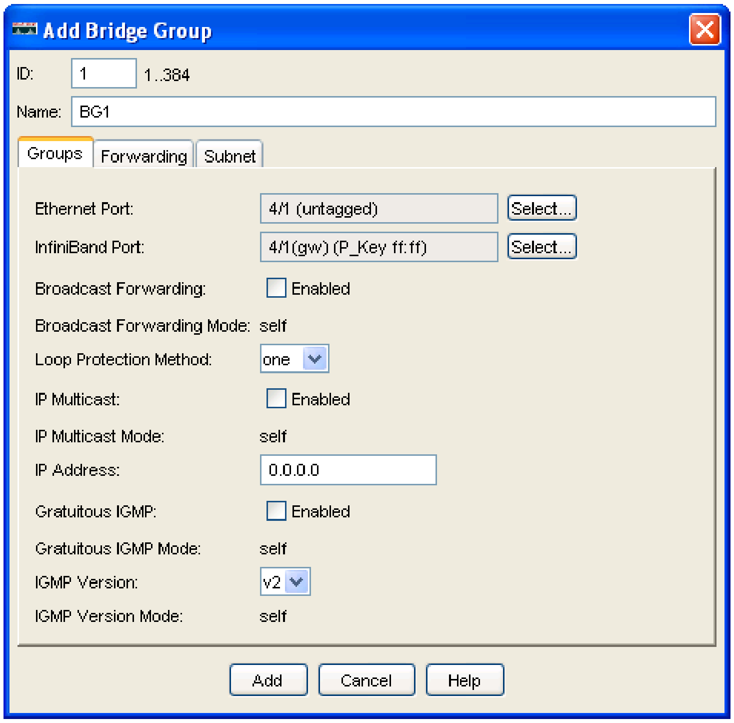
> **Tip:** Do *not* put a default gateway on the mgmt-ib interface. There might be rare cases when this is desired, but in those cases, make sure there is not a default gateway also on the mgmt-ethernet interface. Having more than one default gateway can confuse management traffic flows and lead to unexpected operation.

4. When the mgmt-ib interface is configured, proceed to create BG1 and BG2.

   Note that for redundancy to work, the Bridge groups should be configured similarly. Same p_key on the InfiniBand side and same VLAN (or untagged) on upstream side.

   To begin the process of creating bridge group BG1 and BG2, use the example starting at step 1 on page 125 and create BG1, then repeat the process for creating bridge group BG2, using ID2, name BG2, and for the Ethernet port 8/1 and for the InfiniBand port 8/2.

   Figure 7-23 and Figure 7-24 on page 150 show the two groups, just prior to clicking **Add** (as always, remember to configure the Forwarding tab and Subnet tab for each group per the example starting with step 1 on page 125).



ID: 1 1..384

Name: BG1

Ethernet Port: 4/1 (untagged)

InfiniBand Port: 4/1(gw) (P_Key ff:ff)

Broadcast Forwarding: □ Enabled

Broadcast Forwarding Mode: self

Loop Protection Method: one

IP Multicast: □ Enabled

IP Multicast Mode: self

IP Address: 0.0.0.0

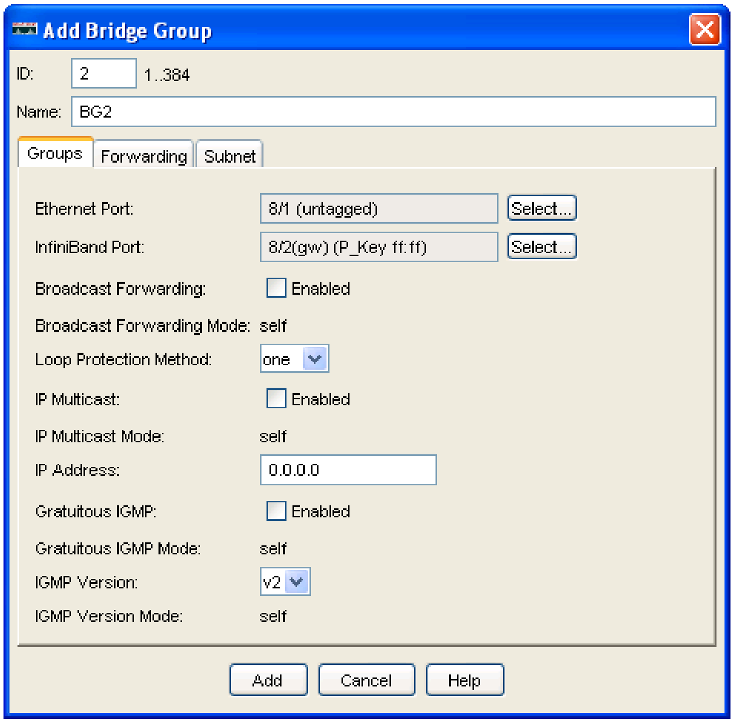
Gratuitous IGMP: □ Enabled

Gratuitous IGMP Mode: self

IGMP Version: v2

IGMP Version Mode: self

Figure 7-23 Bridge group BG1 ready to be added



ID: 2 1..384

Name: BG2

Ethernet Port: 8/1 (untagged)

InfiniBand Port: 8/2(gw) (P_Key ff:ff)

Broadcast Forwarding: □ Enabled

Broadcast Forwarding Mode: self

Loop Protection Method: one

IP Multicast: □ Enabled

IP Multicast Mode: self

IP Address: 0.0.0.0

Gratuitous IGMP: □ Enabled

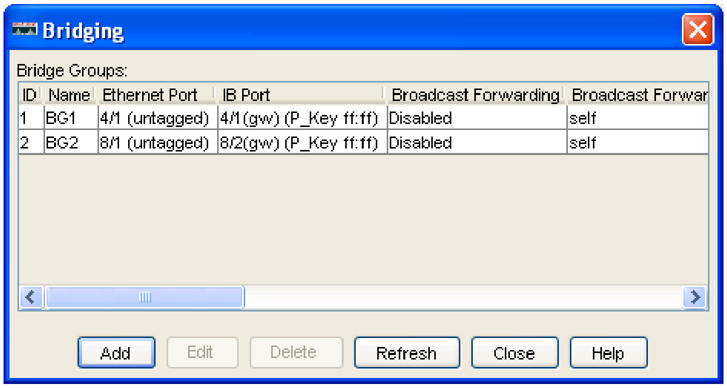
Gratuitous IGMP Mode: self

IGMP Version: v2

IGMP Version Mode: self

Figure 7-24 Bridge group BG2 ready to be added

When BG1 and BG 2 are added, the Bridging window should look as it does in Figure 7-25.



Bridge Groups:

ID	Name	Ethernet Port	IB Port	Broadcast Forwarding	Broadcast Forwar
1	BG1	4/1 (untagged)	4/1(gw) (P_Key ff:ff)	Disabled	self
2	BG2	8/1 (untagged)	8/2(gw) (P_Key ff:ff)	Disabled	self

Figure 7-25 BG1 and BG2 created and ready for use by the redundancy group

5. After BG1 and BG 2 have been created, close the Bridging window and from the main window of the Element Manager, click **Ethernet** → **Redundancy**. This opens a window similar to Figure 7-26.

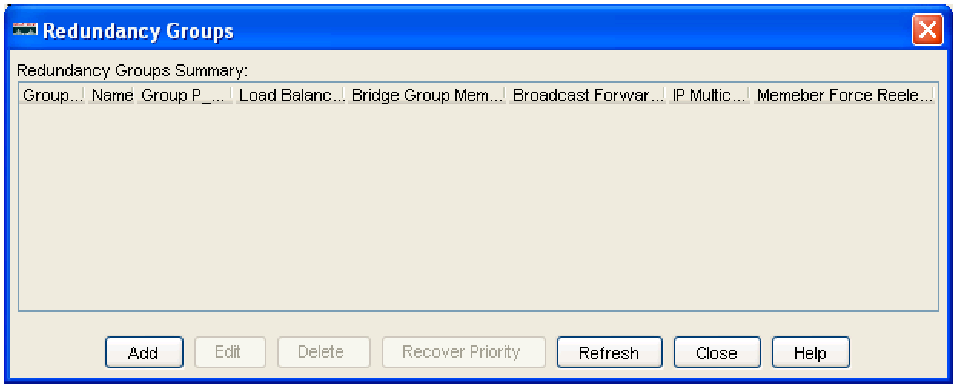
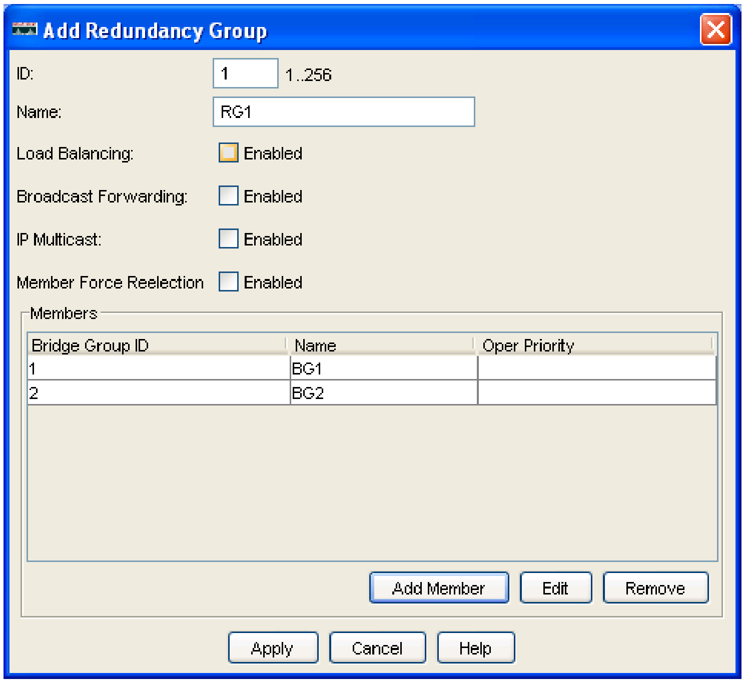


Figure 7-26 Redundancy Groups window before a Redundancy groups has been created

6. Click **Add** in the Redundancy Groups window to begin creating the Redundancy Group. This will open the Add Redundancy Group window, Figure 7-27. Select the ID (default is 1). Give the group a name (RG1 in this example). Click **Add Members**, and select the two bridge groups that had previously been created for this example. The end result should like Figure 7-27.



ID: 1 1..256

Name: RG1

Load Balancing: ☐ Enabled

Broadcast Forwarding: ☐ Enabled

IP Multicast: ☐ Enabled

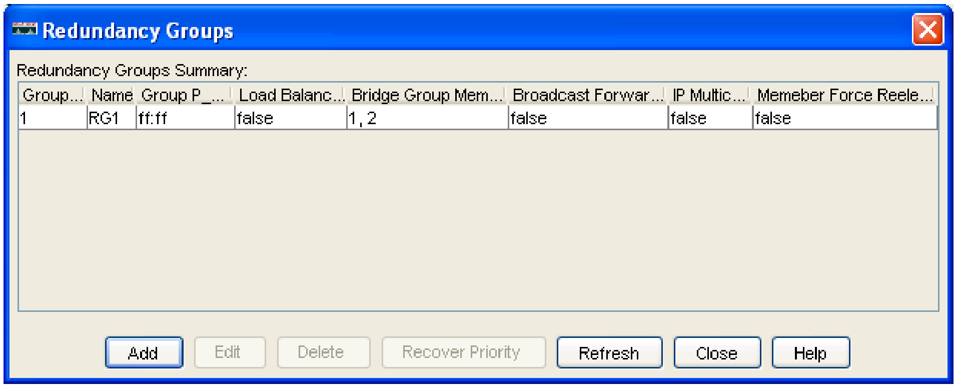
Member Force Reelection ☐ Enabled

Members

Bridge Group ID	Name	Oper Priority
1	BG1	
2	BG2	

Figure 7-27 Redundancy Group filled out, ready to be created

Click **Apply** to create the Redundancy group as shown in Figure 7-28.



Redundancy Groups Summary:

Group...	Name	Group P_...	Load Balanc...	Bridge Group Mem...	Broadcast Forwar...	IP Multic...	Memeber Force Reele...
1	RG1	ff:ff	false	1, 2	false	false	false

Figure 7-28 Redundancy Group RG1 created and ready for operation

7. The final step on the 3012 is to save this config to NVRAM.

> **Note:** As mentioned previously, all changes done through the GUI take effect immediately and are placed in the running config, but the changes in the running config are not explicitly saved to NVRAM until you perform this step.

From the main page of the Element Manager, click **Maintenance** → **Save Config** to save the configuration to NVRAM.

8. After saving the configuration of the 3012 to NVRAM, you need to configure the 4948. In this example, we connect the Ethernet gateway module to the 4948 on ports gi1/16 and 17, with the gateway module 4/1 going to gi0/16 and gateway module 8/2 going to gi0/17.

Note that for reference purposes, we show removing the port channel added in the last example and then setting up all other elements needed to support his redundancy. All of these steps might or might not be necessary in your environment. Contact your network engineer if you are not sure what you need to do for this to work.

After logging into the 4948 with sufficient privileges, execute the commands in Example 7-15 on the 4948. Items that start with an exclamation mark (!) are for reference/comments only and will not be executed.

Example 7-15 Configure 4948 for redundant connection to the gateway

```
! Assumes starting from enable mode, enter config mode
conf t
! Remove port channel 5 from last example
no int po5
! Create VLAN interface and assign an IP address (may already be done)
int vlan 88
ip address 172.16.225.250 255.255.255.0
no shut
! Configure the gateway facing interfaces for mode access, vlan 88
int range gi1/16 -17
description Connection to 3012 Ethernet Switch Module port 4/1 and 8/1
switchport mode access
switchport access vlan 88
! Enable portfast to bring the port up quickly
spanning-tree portfast
! Bring the ports up
no shut
```

```
! Exit config mode
end
! Save config to NVRAM
write
```

> **Tip:** Example 7-15 is for a switch running IOS. If the upstream switch is running CatOS, the configuration would be different, CatOS configuration examples are not included in this document.

9. After the changes have been made to both the Ethernet gateway module and the 4948, re-attach the cables or re-enable the ports.
10. To verify redundancy is working, start a looping ping from the InfiniBand client to the default gateway IP address (172.16.225.250) located on the 4948, and then remove each cable, one at a time. During our testing we found that we lost around 10 ping packets during failover. In the test in Example 7-16, we lost packets 6-14 when we pulled the cable currently being used by the redundancy (4/1), we then re-inserted 4/1 and removed 8/1, and lost packets 24-34 during fail-back.

Example 7-16 InfiniBand host pinging the default gateway

```
[root@localhost ~]# ping  172.16.225.250
PING 172.16.225.250 (172.16.225.250) 56(84) bytes of data.
64 bytes from 172.16.225.250: icmp_seq=1 ttl=255 time=0.307 ms
64 bytes from 172.16.225.250: icmp_seq=2 ttl=255 time=0.331 ms
64 bytes from 172.16.225.250: icmp_seq=3 ttl=255 time=0.289 ms
64 bytes from 172.16.225.250: icmp_seq=4 ttl=255 time=0.301 ms
64 bytes from 172.16.225.250: icmp_seq=5 ttl=255 time=0.339 ms
64 bytes from 172.16.225.250: icmp_seq=15 ttl=255 time=0.256 ms
64 bytes from 172.16.225.250: icmp_seq=16 ttl=255 time=0.275 ms
64 bytes from 172.16.225.250: icmp_seq=17 ttl=255 time=0.345 ms
64 bytes from 172.16.225.250: icmp_seq=18 ttl=255 time=0.275 ms
64 bytes from 172.16.225.250: icmp_seq=19 ttl=255 time=0.292 ms
64 bytes from 172.16.225.250: icmp_seq=20 ttl=255 time=0.279 ms
64 bytes from 172.16.225.250: icmp_seq=21 ttl=255 time=0.287 ms
64 bytes from 172.16.225.250: icmp_seq=22 ttl=255 time=0.295 ms
64 bytes from 172.16.225.250: icmp_seq=23 ttl=255 time=0.331 ms
64 bytes from 172.16.225.250: icmp_seq=35 ttl=255 time=0.344 ms
64 bytes from 172.16.225.250: icmp_seq=36 ttl=255 time=0.286 ms
64 bytes from 172.16.225.250: icmp_seq=37 ttl=255 time=0.294 ms
64 bytes from 172.16.225.250: icmp_seq=38 ttl=255 time=0.342 ms
64 bytes from 172.16.225.250: icmp_seq=39 ttl=255 time=0.280 ms
64 bytes from 172.16.225.250: icmp_seq=40 ttl=255 time=0.277 ms

--- 172.16.225.250 ping statistics ---
40 packets transmitted, 20 received, 50% packet loss, time 38999ms
rtt min/avg/max/mdev = 0.256/0.301/0.345/0.029 ms
```

7.6.4 CLI reference for this section

This section includes what the appropriate CLI looks after the steps were executed in the GUI to create this Redundancy Group, as well as after the changes made to the 4948. It is a useful reference for those wanting to understand the CLI better, or for those that simply want to use the CLI and not the GUI to achieve this task.

Note that we include only the commands that effect the area of the device that we configured. We do not include the entire device configuration.

Example 7-17 is the CLI of the 3012 after the changes made in this section.

Example 7-17 3012 - after redundancy added

```
3012-1# show config
...
interface mgmt-ib
ip address 192.168.0.1 255.255.255.0
no shutdown

bridge-group 1 subnet-prefix 172.16.225.0 24
bridge-group 1 ib-next-hop 172.16.225.250
bridge-group 1 name "BG1"
bridge-group 2 subnet-prefix 172.16.225.0 24
bridge-group 2 ib-next-hop 172.16.225.250
bridge-group 2 name "BG2"
!
interface gateway 4/1
 bridge-group 1 pkey ff:ff
!
interface gateway 8/2
 bridge-group 2 pkey ff:ff
!
interface Ethernet 4/1
 bridge-group 1
!
interface Ethernet 8/1
 bridge-group 2
!
redundancy-group 1
redundancy-group 1 name "RG1"
!
bridge-group 1 redundancy-group 1
bridge-group 2 redundancy-group 1
...
```

Example 7-18 is the CLI of the 4948 prior to changes to support the bridge group (includes left over port channel from last config).

Example 7-18 4948 - before support for redundancy added (assumes po5 already removed)

```
4848-1#show run
...
interface GigabitEthernet1/16
!
interface GigabitEthernet1/17
!
...
interface Vlan88
 ip address 172.16.225.250 255.255.255.0
 no ip route-cache
...
```

Example 7-19 is the CLI of the 4948 after changes to support this bridge group.

Example 7-19 4948 - after support for redundancy added

```
4848-1#show run
...
!
interface GigabitEthernet1/16
 description Connection to 3012 Ethernet Switch Module port 4/1 and 8/1
 switchport access vlan 88
 switchport mode access
 spanning-tree portfast
!
interface GigabitEthernet1/17
 description Connection to 3012 Ethernet Switch Module port 4/1 and 8/1
 switchport access vlan 88
 switchport mode access
 spanning-tree portfast
...
interface Vlan88
 ip address 172.16.225.250 255.255.255.0
 no ip route-cache
...
```

7.7 Useful CLI commands

This section includes some useful CLI commands for supporting this environment.

7.7.1 For the 3012

Table 7-2 shows some useful commands for supporting the 3012 and the Ethernet gateway module.

Table 7-2 3012 commands

Command	Description
`show config`	Shows the current running configuration of the 3012
`show bridge-group`	Shows the configuration and status of any currently configured bridge groups
`show trunk`	Shows the configuration and status of any currently configured trunk groups
`show redundancy-group`	Shows the configuration and status of any currently configured redundancy groups

7.7.2 For the 4948

Table 7-3 shows some useful commands for supporting an upstream Cisco IOS device (a 4948 in our example).

Table 7-3 IOS commands

Command	Description
`show run`	Shows the current running configuration of the 4948
`show int status`	Shows a snap-shot of all of the ports on the system and their status (connected/not connected, VLAN in use, speed, and so forth)
`show int trunk`	Shows information and status on any ports configured as 802.1Q trunks
`show etherchannel summary`	Shows information and status on any configured aggregation groups

Abbreviations and acronyms

AC	alternating current
AMD	Advanced Micro Devices™
API	application programming interface
ARP	Address Resolution Protocol
ASP	active server page
ATM	asynchronous transfer mode
BG	bridge group
BIOS	basic input output system
BMC	baseboard management controller
BTH	base transport header
CCIP	Cisco Certified Internetwork Professional
CCNA	Cisco Certified Network Associate
CCO	Cisco Connection Online
CD	compact disk
CDP	Cisco Discovery Protocol
CLI	command-line interface
CPU	central processing unit
CRC	cyclic redundancy check
DAPL	Direct Access Programming Layer
DC	domain controller
DDR	Double Data Rate
DHCP	Dynamic Host Configuration Protoco
DIMM	dual inline memory module
DMA	direct memory access
ECC	error checking and correcting
ESM	Ethernet switch modules
ETSI	European Telecommunications Standard Industry
FC	Fibre Channel
FCP	Fibre Channel Protocol
GB	gigabyte
GMP	General Service Management Packets
GRH	global route header
GSA	General Service Agents
GSI	General Services Interface
GSM	General Service Manager
GUI	graphical user interface
GUID	Globally Unique ID
HA	high availability
HBA	host bus adapter
HCA	host channel adapter
HDD	hard disk drive
HPC	high performance computing
HSFF	high-speed form factor
I/O	input/output
IB	InfiniBand
IBM	International Business Machines Corporation
ICRC	invariant cyclic redundancy check
ID	identifier
IEEE	Institute of Electrical and Electronics Engineers
IOC	IO controllers
IOS	Internetwork Operating System
IP	Internet Protocol
IPTV	Internet Protocol Television
ISO	International Organization for Standardization
IT	information technology
ITL	initiator, target, LUN
ITSO	International Technical Support Organization
JVM	Java Virtual Machine
KB	kilobyte
KVM	keyboard video mouse
LAN	local area network
LED	light emitting diode
LID	Local ID
LRH	Local Route Header
LUN	logical unit number
MAC	media access control
MB	megabyte
MCNE	Master Certified Novell Engineer
MCSE	Microsoft® Certified Systems Engineer
MDI	medium dependent interface
MFIO	Multifabric I/O)
MPI	Message Passing Interface
MPICH	Message Passing Interface Chameleon

MSI	Microsoft Installer
MTU	maximum transfer unit
NEBS	network equipment building system
NFS	network file system
NGN	next-generation networks
NIC	network interface card
NVRAM	non-volatile random access memory
OFED	OpenFabrics Enterprise Distribution
OS	operating system
OSI	Open Systems Interconnect
PCI	Peripheral Component Interconnect
PXE	Pre-boot-execution
QP	queue pair
RAC	Real Application Clusters
RAID	redundant array of independent disks
RAS	remote access services; row address strobe
RDMA	Remote Direct Memory Access
RDS	Reliable Datagram Sockets
RHEL	Red Hat Enterprise Linux
RSA	Remote Supervisor Adapter
SAN	storage area network
SAS	Serial Attached SCSI
SCSI	small computer system interface
SDK	Software Developers' Kit
SDP	Sockets Direct Protocol
SDR	Single Data Rate
SFF	Small Form Factor
SFP	small form-factor pluggable
SFS	Server Fabric Switch
SIO	Storage and I/O
SL	SlimLine
SM	Subnet Manager Switch Module
SNMP	Simple Network Management Protocol
SRP	Storage RDMA Protocol
SSH	Secure Shell
TCA	Target Channel Adapter
TCO	total cost of ownership
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TOE	TCP offload engine
TX	transmit
UI	user interface
URL	Uniform Resource Locator
USB	universal serial bus
VCRC	variant cyclic redundancy check
VL	virtual lane
VLAN	virtual LAN
VNIC	virtual network interface card
VOIP	Voice over Internet Protcol
WAN	wide area network
WWN	World Wide Name
WWNN	World Wide Node Name
WWPN	World Wide Port Name

Related publications

We consider the publications that we list in this section particularly suitable for a more detailed discussion of the topics that we cover in this paper.

IBM Redbooks

You can search for, view, or download books, papers, Technotes, draft publications and additional materials, as well as order hardcopy Redbooks, at the IBM Redbooks Web site:

ibm.com/redbooks

Related publications from IBM Redbooks publications include the following:

- *IBM BladeCenter Products and Technology*, SG24-7523

Other publications

These publications are also relevant as further information sources:

- Cisco 4x InfiniBand Switch Module User’s Guide

  http://www.ibm.com/support/docview.wss?uid=psg1MIGR-65966

- QLogic InfiniBand Ethernet Bridge Module and Fibre Channel Bridge Module Installation Guide

  http://www.ibm.com/support/docview.wss?uid=psg1MIGR-5070108

- Cisco SFS 3000 Series Multifabric Server Switches product publications

  http://www.cisco.com/en/US/products/ps6422/products_user_guide_list.html

- Cisco VFrame Server Fabric Virtualization Software documentation

  http://www.cisco.com/en/US/products/ps6429/index.html

- Cisco SFS InfiniBand Redundancy Configuration Guide

  http://www.cisco.com/en/US/products/ps6422/products_installation_and_configuration_guides_list.html

- Cisco SFS 7000 Series Product Family Command Reference

  http://www.cisco.com/en/US/products/ps6421/products_command_reference_book09186a008070c2eb.html

- Cisco SFS Product Family Command Reference

  http://www.cisco.com/application/pdf/en/us/guest/products/ps6421/c2001/ccmigration_09186a00808669e2.pdf

Online resources

These Web sites are also relevant as further information sources:

IBM Web sites

- IBM ServerProven

  `http://www.ibm.com/servers/eserver/serverproven/compat/us/`

- QLogic InfiniBand Fibre Channel Bridge Module firmware update 4.1.0.2.2

  `http://www.ibm.com/support/docview.wss?uid=psg1MIGR-5069861`

- QLogic InfiniBand Fibre Channel Bridge Module firmware update 3.3.0050.0

  `http://www.ibm.com/support/docview.wss?uid=psg1MIGR-5069862`

Cisco Web sites

- Cisco SFS 3000 Series Multifabric Server Switches Product Literature

  `http://www.cisco.com/en/US/products/ps6422/prod_literature.html`

- Cisco OFED and SRP host drivers for Linux

  `http://www.cisco.com/pcgi-bin/tablebuild.pl?topic=280511613`

- Connecting InfiniBand with Ethernet and Fibre Channel

  `http://www.cisco.com/en/US/products/ps6422/index.html`

- OFED driver download

  `http://www.cisco.com/cgi-bin/tablebuild.pl/sfs-linux`

QLogic Web sites

- InfiniBand Fibre Channel Bridge Module firmware

  `http://support.qlogic.com/support/oem_detail_all.asp?oemid=377`

Other Web sites

- OpenFabrics Alliance

  `http://www.openfabrics.org/`

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services