



## **High Performance Trading/Algo Speed with Wombat Design and Implementation Guide**

January 24, 2008

**Americas Headquarters**  
Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
USA  
<http://www.cisco.com>  
Tel: 408 526-4000  
800 553-NETS (6387)  
Fax: 408 527-0883

Text Part Number: OL-15617-01

ALL DESIGNS, SPECIFICATIONS, STATEMENTS, INFORMATION, AND RECOMMENDATIONS (COLLECTIVELY, "DESIGNS") IN THIS MANUAL ARE PRESENTED "AS IS," WITH ALL FAULTS. CISCO AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE. IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THE DESIGNS, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

THE DESIGNS ARE SUBJECT TO CHANGE WITHOUT NOTICE. USERS ARE SOLELY RESPONSIBLE FOR THEIR APPLICATION OF THE DESIGNS. THE DESIGNS DO NOT CONSTITUTE THE TECHNICAL OR OTHER PROFESSIONAL ADVICE OF CISCO, ITS SUPPLIERS OR PARTNERS. USERS SHOULD CONSULT THEIR OWN TECHNICAL ADVISORS BEFORE IMPLEMENTING THE DESIGNS. RESULTS MAY VARY DEPENDING ON FACTORS NOT TESTED BY CISCO.

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

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0612R)

*High Performance Trading/Algo Speed with Wombat Design and Implementation Guide*  
© 2007 Cisco Systems, Inc. All rights reserved.



## CONTENTS

Introduction	1-1
Target Audience	1-3
Target Market	1-4
Automated Trading Benefits	1-4
Automated Trading Architecture	1-6
Concept Features	1-10
Tested Components	1-11
Servers	1-11
Networking	1-12
Operating System	1-12
Test Implementation Framework	1-13
Testing Topology	1-13
Testing	1-14
Methodology	1-14
Test Setup	1-14
Procedures	1-15
Data Observations	1-16
Time Synchronization	1-16
Limitations	1-16
Testing Results	1-17
Mean Latency	1-17
Latency Dispersion	1-22
Appendix A—Device Configuration	1-28
Catalyst Switch Configuration	1-28
SFS 7000 Configuration (Core)	1-40
Appendix B—Building and Configuring Switches	1-42
Definitions	1-42
The Basics	1-42
Installation Task and Timing Overview	1-43
The Very First Thing That You Do: Plan	1-43
Install Interface Cards in the Hosts	1-48
Rack and Cable All Hardware	1-48
Write Down Your Cabling Connections	1-48

Configure Ethernet Attributes of Leaf Switches	1-49
Configure Ethernet Attributes of Core Switches	1-49
Validate the Ethernet Management Network	1-50
Set Up SE Tools on a Ethernet-attached Host	1-50
Perform a Switch Chassis Inspection	1-50
Perform a Physical Inspection	1-50
(Optional) Record Leaf Switches and Hosts	1-50
Disable Uplinks on Leaf Switches	1-51
Install Host-Side Drivers and Configure IP Addresses to InfiniBand Ports on Hosts	1-51
Troubleshoot “Bring Up” Pod	1-53
Run Step Troubleshoot “Bring Up” Pod On All Pods	1-55
Connect “bring up” Pod to Core Switches One at a Time	1-55
Connect Pods to Core Switches	1-55
Troubleshooting after Pruning	1-55



# High Performance Trading/Algo Speed with Wombat Design and Implementation Guide

---

## Introduction

Automated trading and new regulatory demands are the two main forces behind the changes in Financial Markets. Firms are trying to maintain their competitive edge by constantly changing their trading strategies and increasing the speed of trading. New financial products, business models, and trading tools require a super fast response.

Automated trading is creating a faster trading cycle, in which milliseconds matter. The faster your trading infrastructure is the better chance you have of hitting your price points in a very dynamic market. This shift from manual to automated trading is creating a huge strain on the information infrastructure of financial firms, because it requires both speed and handling huge volumes of data, with maximum reliability and predictability. The front-office systems of trading departments are the most latency sensitive and the most visible in the value chain. This is where the automated trading happens and where Cisco is offering the High Performance Trading (HPT) Algo Speed solution.

HPT Algo Speed solution enables automated trading applications to communicate fast and handle high volumes of market data in a predictable and reliable manner. For example, to gain a competitive edge in the marketplace, portfolio and risk managers must be able to access real-time financial information and use technical indicators to buy and sell equities and exotic investments. As new investment products are introduced, the need to obtain an accurate view of a fund's value and its related risk is significantly increased. These growing business necessities are creating enormous stress on the current compute infrastructures, where highly computational algorithms in mission-critical applications are unable to scale effectively.

Cisco is uniquely positioned to address this problem with its high-performance computing solutions, which include both Ethernet and InfiniBand technologies. Cisco's broad solution portfolio delivers high speeds, low latencies, open standards, and high system availability and allows financial customers to deploy the right infrastructure for their application.

To solve many of the emerging application requirements, Cisco offers InfiniBand and high-density 10 Gigabit Ethernet solutions, which are optimized for the most-demanding financial applications. The InfiniBand 4X DDR (double data rate) technology in the Server Fabric Switching (SFS) family can provide throughput rates of up to 20 Gbps. This ultra-low-latency computing fabric provides native remote direct memory access (RDMA) capabilities, to share computational power across multiple CPUs and ensure maximum cluster performance. RDMA has the additional benefit of allowing inter-CPU and



---

**Corporate Headquarters:**  
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

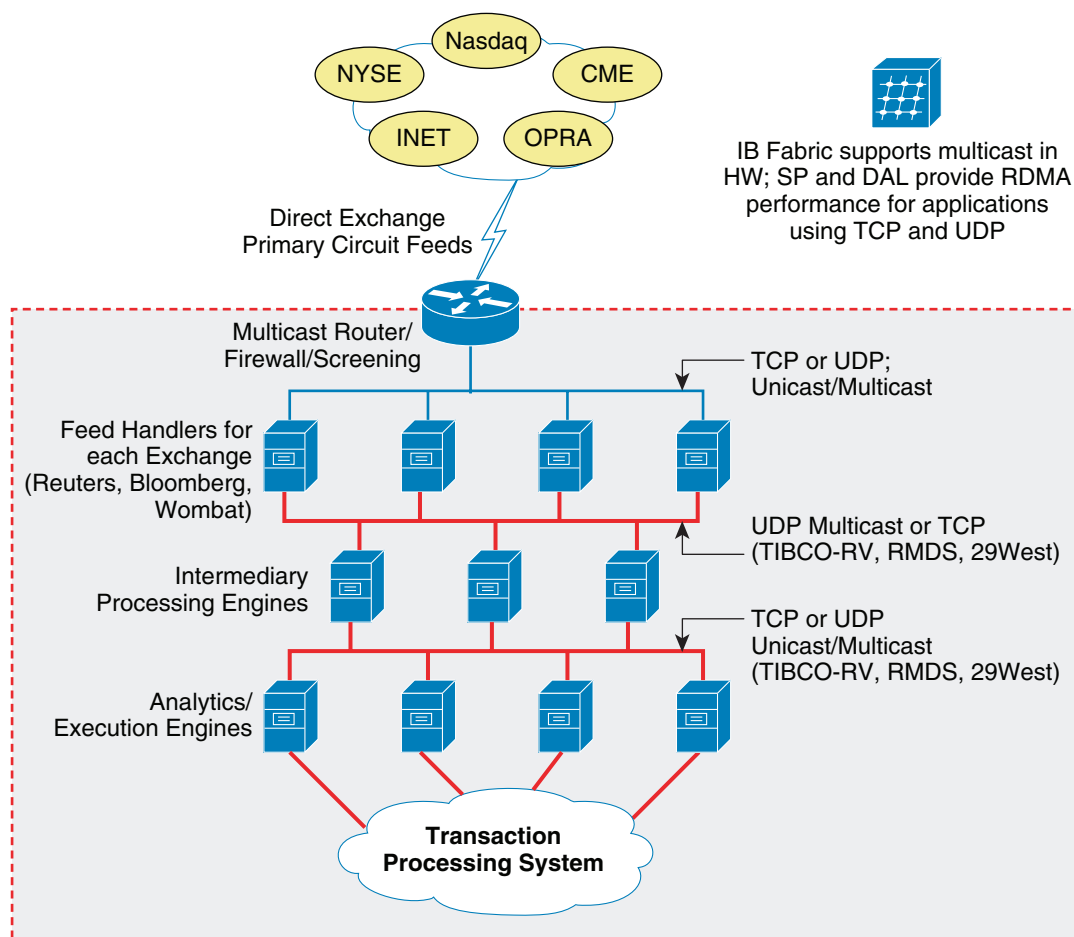
Copyright © 2007 Cisco Systems, Inc. All rights reserved.

memory read/writes, as well as kernel bypass. Applications that specifically support Message Passing Interface (MPI) or Open Fabrics messaging transports can effectively achieve latencies of less than 10 microseconds.

The InfiniBand fabric can also be seamlessly integrated with existing Ethernet networks by using SFS 3000 Series switches-this eliminates any interoperability concerns. The Cisco Catalyst® 6500 Series switches offer a high-density 10 Gigabit Ethernet solution with low latency that is ideal for large Ethernet clusters. The flexibility of multiple-grid computing designs that can adapt to any grid environment allows an organization to scale quickly with growing business demands. Financial market data providers and consumers are aggressively preparing for upcoming changes in the industry driven by the Regulation National Market System (Reg NMS), the Markets in Financial Instruments Directive (MiFID), and FIX Adapted for Streaming (FAST). A utility computing model is necessary to help organizations respond to these changing market conditions and skyrocketing data rates. A utility computing model constructed with Cisco InfiniBand fabric and the VFrame management platform can help manage virtualized computing and network resources. Capacity can be added on demand or as required by business policies. As clusters become grids, Cisco can quickly make use of servers that allow an organization to scale up its computing resources faster and more cost-effectively. New services and extra compute power can be added automatically, on the fly, to maximize the utilization of servers that typically operate below optimum capacity. This intelligent HPC network fabric helps reduce the total cost of ownership because resources are used more efficiently. Further, it provides the architecture for consolidating and virtualizing resources, enabling the evolution to an automated system that is able to dynamically respond to changing business needs.

Figure 1 shows the general automated trading solution environment.

**Figure 1** General Automated Trading Solution Environment



## Target Audience

This document is intended to be used by Cisco Systems Engineers, Advanced Services Engineers, partners, and clients who work on trading floor infrastructure projects. The document was created based on the testing performed by STAC research to quantify the performance gains of low latency computing in an HPT environment. The test environment was limited and not necessarily characteristic of a production environment, but the concepts are the same. The information contained in this document can be used to identify, build and test a production prototype environment.



### Note

This environment was tuned specifically for the testing performed. Note that any change from what was specifically tested here can affect your results. It is important to keep this in mind as you build your test environment and go through the associated test plans.

## Target Market

Financial services firms require vast amounts of computing power to run their business. They use homegrown applications or increasingly ISV applications to do these computations. However, these applications were originally built for SMP machines or hard-coded clusters of a certain size. The net result is that computations takes hours and days, while the business needs it to happen in seconds and minutes.

Further, the average utilization of the server farm is very low ( less than 10 percent). Given a fixed IT budget, this means the firm does a lot less computation than it would like to. Fewer scenarios are run and modeling is done to less accuracy. Some businesses do not use modeling at all due to lack of resources

Solving these problems can provide a competitive advantage to the customer. Specifically, the following businesses are the ones most affected:

- Front office—Pricing and hedging of derivatives, foreign exchange options, and other structured financial products
- Mid office—Risk analysis of portfolio, counter-party credit risk, enterprise risk, proprietary desk analytics
- Back office—Fraud detection, global treasury

## Automated Trading Benefits

Automated trading benefits include the following:

- Delivers low-latency and high speed Ethernet and InfiniBand interconnects for financial applications such as trading floors and market data feed

Automated trading makes reducing latency and increasing performance essential in a market data environment. Feed handlers receive real-time market data feeds from sources such as Options Price Reporting Authority (OPRA), NASDAQ, and electronic communications networks (ECNs), and these feeds need to be “normalized” before being distributed to users. Data normalization allows feeds to be standardized through dedicated feed processors and entered into a uniform database model. The uniform access to multiple normalized market data feeds facilitates data distribution to end-users and ensures data consistency throughout the organization. The high server-to-server traffic that occurs as these computations are performed means that a cluster of servers with the lowest possible latency interconnect is needed to reduce delay in delivering the market data.

Through the use of RDMA technology, an application can offload all communications management to the InfiniBand host channel adapter, which allows more CPU cycles to be spent on processing, rather than communications. Cisco’s innovative HPT solution creates a high-performance server I/O fabric, achieving ultra-low-latency performance to support the growing computing needs of market data feed handlers and other trading floor applications.

- Minimizes latency in each component of delivery platform

It is imperative that latency be minimized when delivering time-sensitive data. As the data traverses the different components of a trading platform—including market data delivery, order routing, and execution—an HPC environment addresses the speed-sensitivity requirements by providing a lowest latency interconnect, so that raw computational power can be used in clusters to deliver the fastest response possible.

- Helps prepare for new regulations that drives high market-data volumes



Regulatory changes such as Reg NMS generate more quote, order, and cancel/replace messages as equity firms adapt to more electronic business processes. The subpenny pricing rule also increases demands on the supporting infrastructure. MiFID, which goes into effect in Europe next year, is expected to lead to higher data volumes as well, since investment banks that internalize trades will be required to publish their pre-trade quotes electronically. An InfiniBand HPC environment provides a secure, scalable solution to meet the growing needs of the financial services industry.

- Increases competitive edge by incorporating FAST protocol for lowest-latency connections

As financial industry experts predict relentless growth in market-data volumes, organizations are preparing for implementation of the FAST Protocol. FAST offers a lower-latency feed that uses a data compression technology. Exchanges are planning to use FAST to deliver new products in areas such as derivatives and equities. However, to support the new, rapid market data message rates and deliver the quickest-possible trade execution, a grid computing model is needed to parse the data from the feeds and then deliver it to consumers. The InfiniBand HPC solution can provide the lowest latency transport while providing the bandwidth to sustain increased market volumes.

- Provides flexibility to support a service-oriented architecture based on industry-standard protocols

The Cisco HPC solution adheres to industry standard protocols such as Open MPI and Open Fabrics. This allows customers to leverage the true high-performance, ultra low-latency characteristics of the InfiniBand fabric. An application environment that supports these industry protocols benefits from the open architecture, as it becomes part of a service-oriented architecture strategy for the adaptive enterprise. This shortens the time to market for new financial products, providing a competitive edge.

- Allows the use of common tools to manage Ethernet and InfiniBand networks

The VFrame management platform enables the delivery of utility computing into the data center environment. This increases the ability to rapidly provision shared-server and I/O resources on demand. By managing and orchestrating diverse Ethernet and InfiniBand resources, a financial organization can become more agile-adapting easily to rapidly changing market conditions. In addition, just intime provisioning reduces operational costs by automating regular tasks. Since InfiniBand creates a high-speed fabric that is shared by all the nodes participating in it, downtime can be quickly averted by reallocating resources to different resource pools.

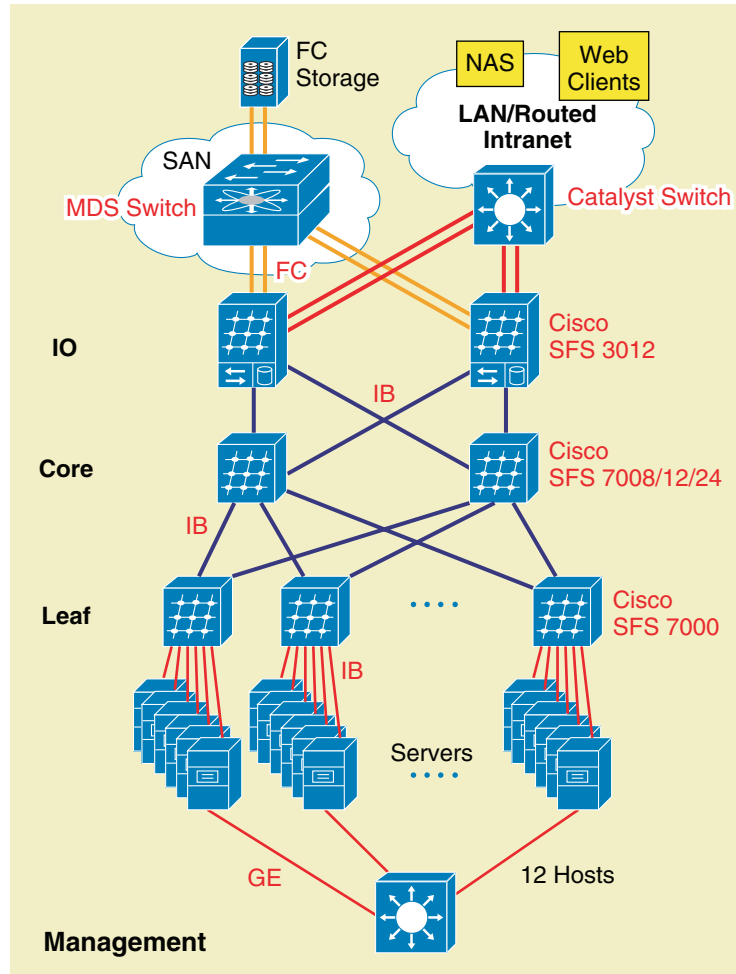
- Supports the increasing trend toward a utility computing model to support heavy computations

As market-data levels continue to rise and financial organizations look to expand their product portfolios, increasing raw computational power is required to support the algorithms needed for portfolio performance analytics, Monte Carlo simulations, value calculations, and risk profiling of trades. Grid services virtualize computing silos that underperform or are underutilized and makes them well-balanced, fully optimized enterprise backbones. The Server Fabric HPC infrastructure can be scaled quickly, because additional computing power can be added to the grid dynamically to support business processes with increasing demands. The combination of Server Fabric Switching and VFrame allows for effective consolidation, virtualization, and automation of resources that deliver instantaneous return on investment.

# Automated Trading Architecture

Figure 2 shows the general automated trading network architecture.

Figure 2 General Automated Trading Network Architecture



223252

Figure 3 shows a high-level view of the automated trading architecture.

**Figure 3** *Automated Trading—High Level Architecture*

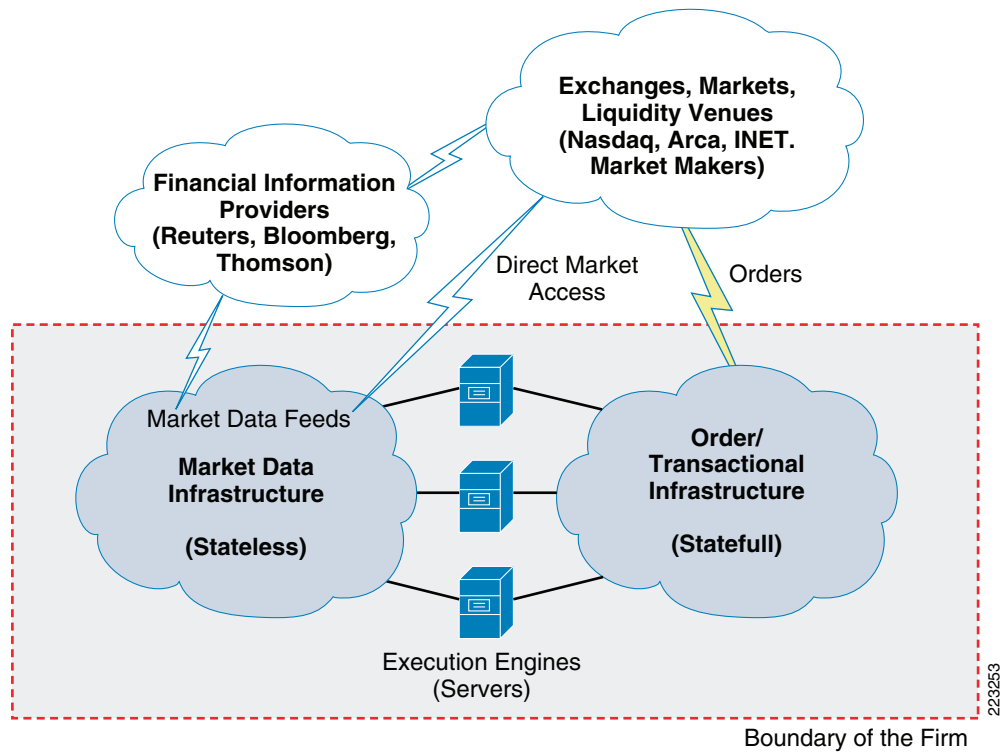


Figure 4 shows the “buy” side of the transactional system architecture.

**Figure 4 Transactional System Architecture—Buy Side**

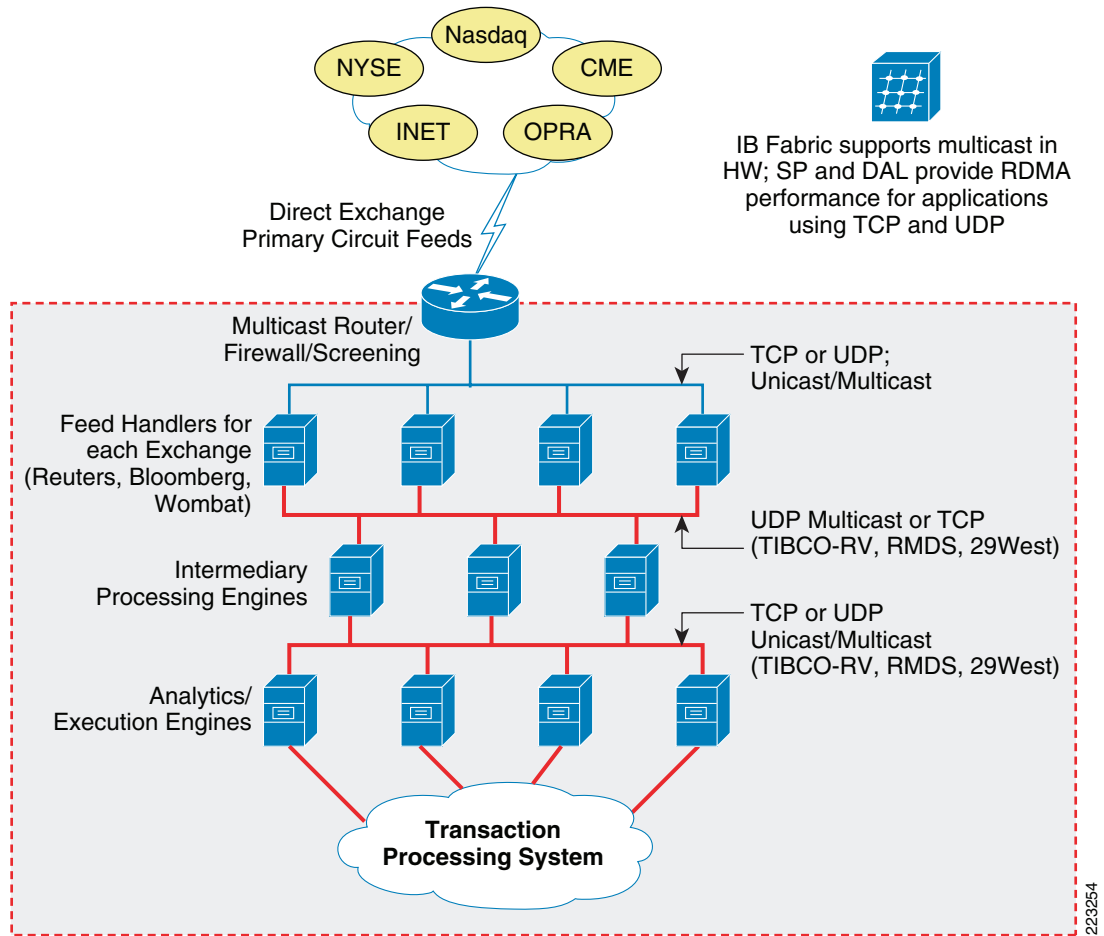
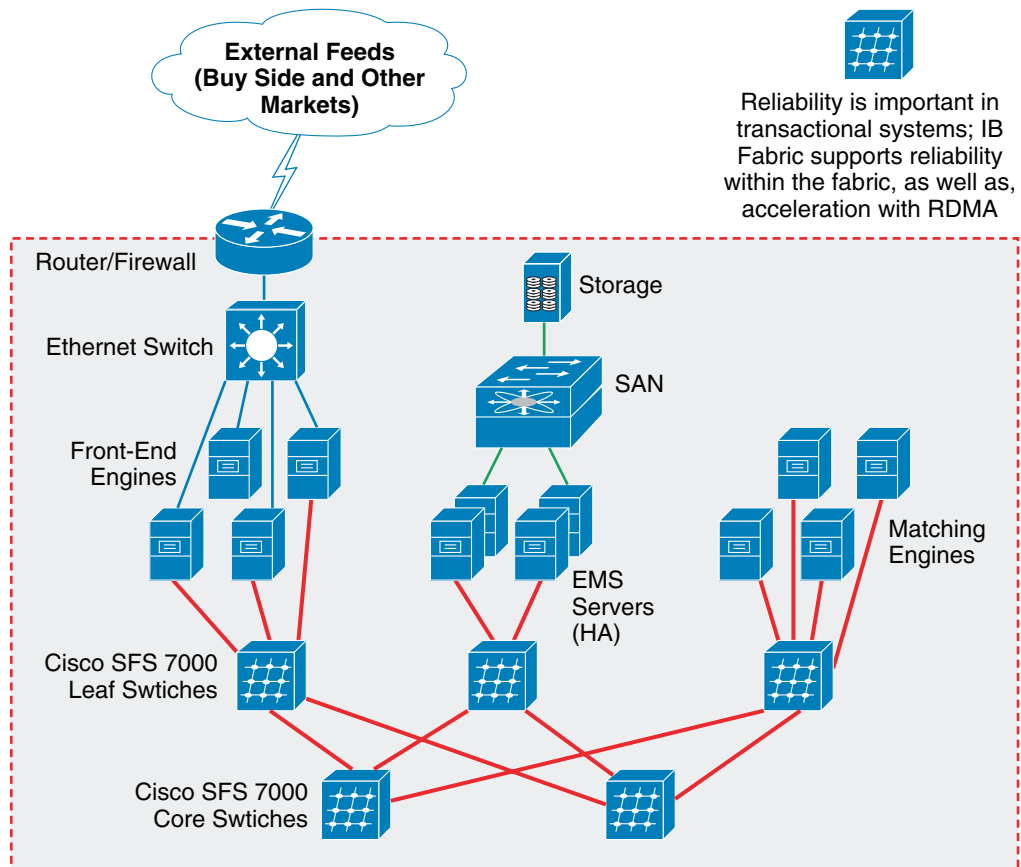


Figure 5 shows the “sell” side of the transactional system architecture.

**Figure 5** Transactional System Architecture—Sell Side



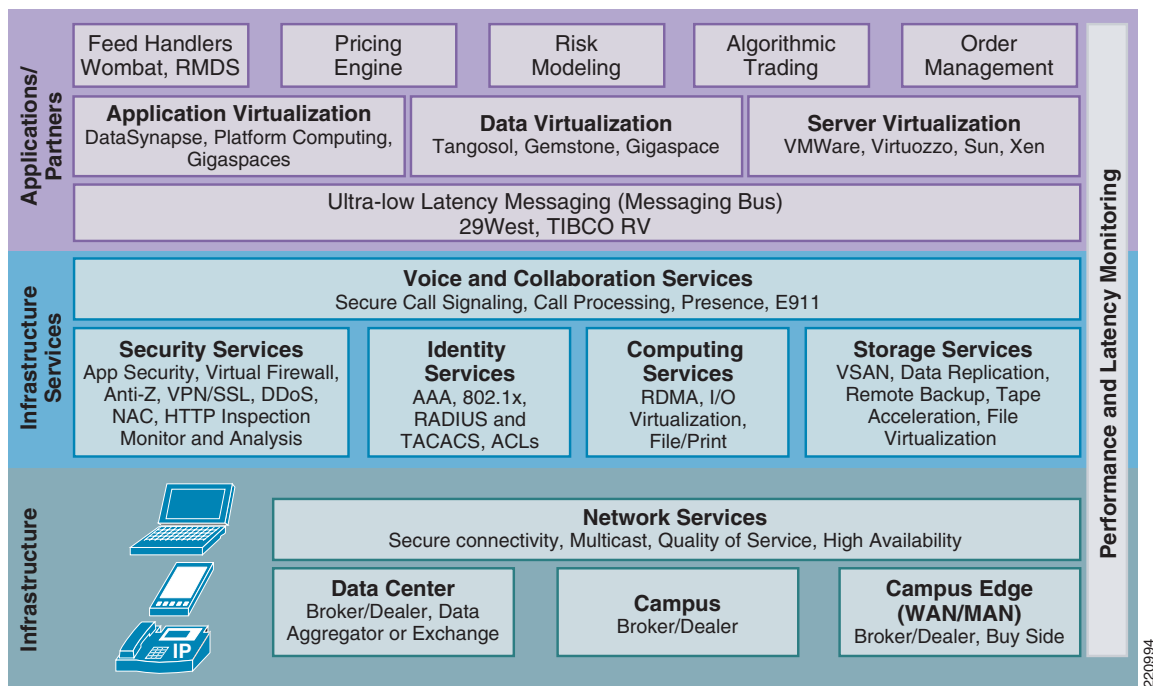
# Concept Features

We are proposing a services-oriented framework for building the next-generation trading architecture. This approach provides a conceptual framework and an implementation path based on modularization and minimization of inter-dependencies. This framework provides firms with a methodology to:

- Evaluate their current state in terms of services
- Prioritize services based on their value to the business
- Evolve the trading platform to the desired state using a modular approach

The high performance trading architecture relies on the following services, as defined by the services architecture framework represented in Figure 6.

**Figure 6 Service Architecture Framework for High Performance Trading**



**Table 1 Service Descriptions and Technologies**

Service Description	Technology
Ultra-low latency messaging	Middleware
Latency monitoring	Instrumentation—appliances, software agents, and router modules
Computing services	OS and I/O virtualization, Remote Direct Memory Access (RDMA), TCP Offload Engines (TOE)
Application virtualization	Middleware which parallelizes application processing
Data virtualization	Middleware which speeds-up data access for applications, e.g., in-memory caching
Multicast service	Hardware-assisted multicast replication through-out the network; multicast Layer 2 and Layer 3 optimizations

**Table 1 Service Descriptions and Technologies (continued)**

Service Description	Technology
Storage services	Virtualization of storage hardware (VSANs), data replication, remote backup, and file virtualization
Trading resilience and mobility	Local and site load balancing and high availability campus networks
Wide area application services	Acceleration of applications over a WAN connection for traders residing off-campus
Thin client service	De-coupling of the computing resources from the end-user facing terminals

The main Algo Speed features are as follows (the implementation of the test case offers these features):

- Acceleration of trading applications.
- More efficient use of server capacity—Servers have more CPU "headroom" to process applications instead of network traffic.
- Predictability—The variation in application messaging delay is reduced, ensuring consistent data input into algorithmic trading engines.
- Reliability—High availability features in Cisco switches enable the design of robust network, proven by large deployments in critical environments.
- Manageability—End-to-end management across multiple switching fabrics (not to be confused with network management).
- Visibility—Microbursts detection in market data traffic; FIX protocol monitoring; identification of the source of delay

## Tested Components

### Servers

Each of the servers in the test harness had the following specifications:

**Table 2 Server Components**

Vendor Model	Dell PowerEdge 1950
Processors	2 x Quad Core Intel Xeon 5355 2.66 GHz
Cache	2*4086KB L2 Cache
Bus Speed	1.33 GHz
Memory	8 GB (4x2048MB) Fully Buffered DIMMs @ 667 MHz
BIOS	Dell Inc. Version 1.2.0 dated 10/18/2006
Disk Controller	LSI SAS 1068 Controller
Disks	1x66GB SAS
Fault Tolerance	None

## Networking

**Table 3**      **Networking Components**

Ethernet Switch	Cisco Catalyst 6509 1 GbE, 720 supervisor, 6748 line cards
Ethernet NIC	Embedded Broadcom 5708 GbE NIC
Network Interface Configuration	None
InfiniBand Switch	Cisco InfiniBand SFS7000 (single data rate)
InfiniBand HCA	Cisco Cheetah DDR HCA PCIe 8x slot (running at SDR)

## Operating System

**Table 4**      **Operating System**

Version	RHEL 4.4 64bit Kernel 2.6.9-42.ELsmp #1 SMP
TCP and UDP Buffers	The following parameters were set in the operating system (/etc/sysctl.conf).  net/core/rmem_max=8388608 net/core/wmem_max=8388608 net/core/wmem_default=262144 net/core/rmem_default=262144 net/core/netdev_max_backlog=10000 net/ipv4/tcp_rmem=4096 262144 8388608 net/ipv4/tcp_wmem=4096 262144 8388608 net/ipv4/tcp_window_scaling=0
Operating System Services	rhnsd isdn kudzu cups anacron cpuspeed ip6tables pcmcia xfs iptables arptables_jf cups-config-daemon apmd mdmonitor hpoj nfslock netfs sendmail openibd ntpd chkconfig crond

## Application Software

**Table 5**      **Application Software**

Server binaries	Wombat Feeds 2.17.15a (opra) with 29 West LBM 2.3.4
Server configuration parameters	MamaTimeFieldFormat: double, MamaPublishSendTime: true, ActivityTimeStamps: false, LbtTransportLbtrmMulticastAddressHigh,LbtTransportLbtrmMulticastAddressLow, LbtTransportLbtrmDestinationPort: unique per channel. For Gigabit: LbtImplicitBatchingInterval: 5, LbtImplicitBatchingMinimumLength: 500. For DAL/IB: LbtImplicitBatchingInterval: unset, LbtImplicitBatchingMinimumLength: 1
Server affinities and process priorities	Not set
Client binaries	mamaperf 4.0.0a
Client configuration parameters	Default LBM Parameters with separate MAMA transports defined for data dictionary and data. Symbol file for each channel containing all Symbols available on channel

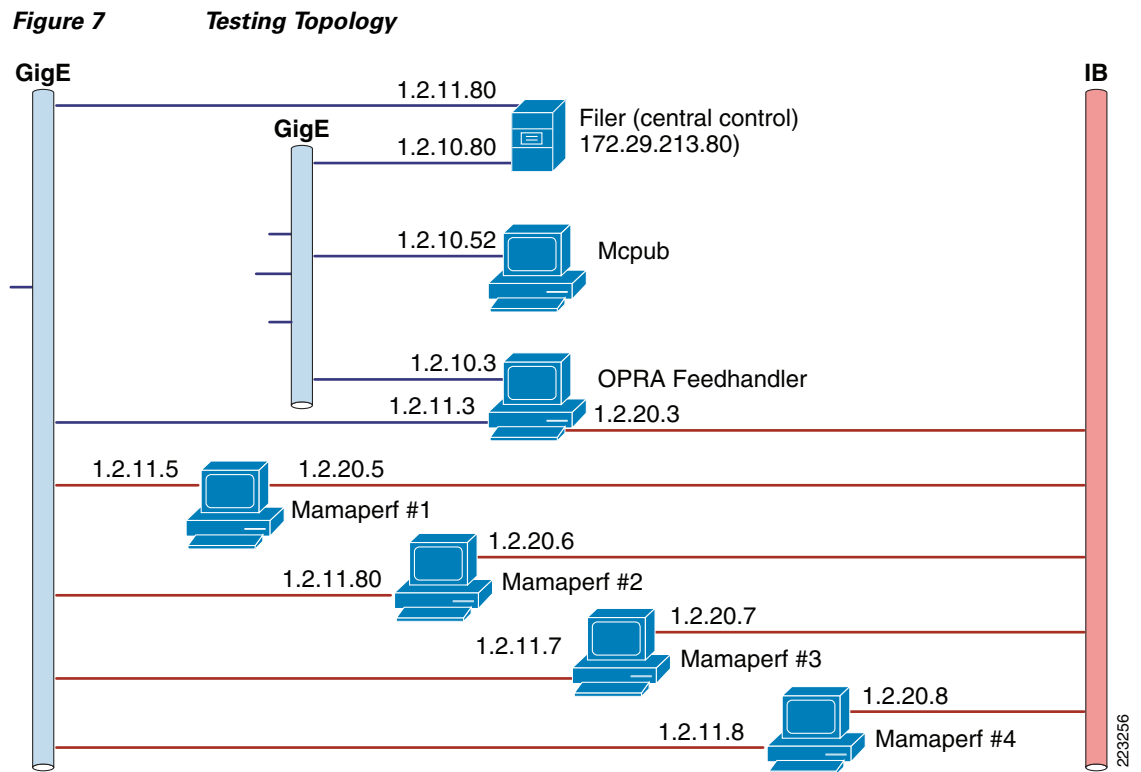


**Table 5 Application Software (continued)**

Client affinities and process priorities	Not set
Playback Data	OPRA data recorded on 2 April 07

# Test Implementation Framework

## Testing Topology



# Testing

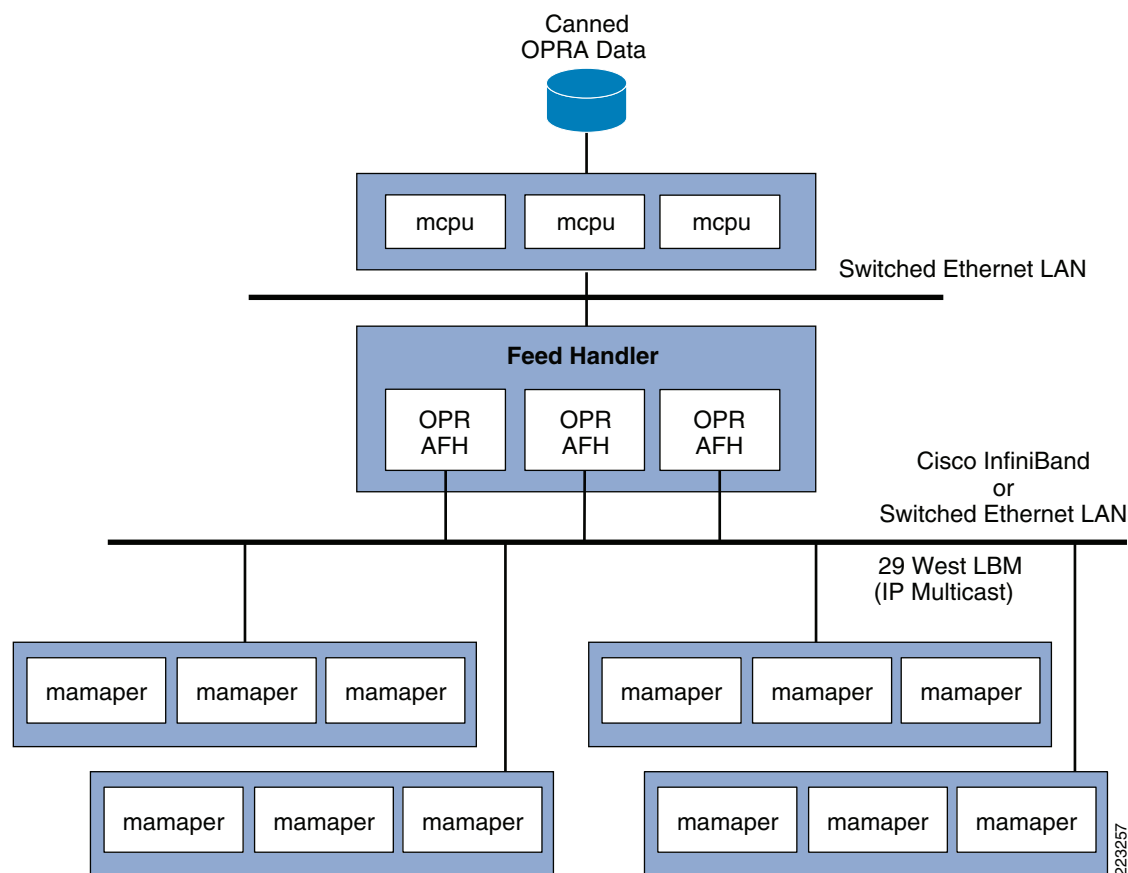
## Methodology

### Test Setup

Six servers are arranged as in [Figure 8](#). The components consist of recorded OPRA (options price reporting authority) data (from April 2, 2007), the Wombat playback mechanism *mcpub* using the *papareplay* library, the Wombat OPRA feed handler (OPRA FH), and Wombat's preferred performance measurement client, *mamaperf*. *Mcpub* replays data from previously captured OPRA files. The goal of the *papareplay* library is to mimic the original timing of the market data. This is important, since market data is notoriously spikey, which can affect performance. The job of *papareplay* is to preserve the relative timing of updates even as the playback rate is increased. The OPRA feed handler normalizes and redistributes content, keeps a current cache, and otherwise manages the feed on behalf of consuming clients. *mamaperf* subscribes to content and calculates and logs latency statistics (see [Data Observations](#), [page 16](#) for more detail).

Three *mcpub* instances were run on one server, each listening to a single OPRA line (multicast channel) plus its backup (i.e., performing line arbitration) and publishing on a two multicast channels. These transmitted exchange data over a GigE switch. Downstream was a single server running three instances of the Wombat OPRA feed handler. Each instance was dedicated to a single OPRA FH channel. Twelve *mamaperf* clients were distributed across four machines, three to a machine. Each *mamaperf* instance subscribed to all of the items from one of the three channels. Across the system, this meant that there were four consuming applications for each of the three OPRA FH channels.

Figure 8 Test Setup



## Procedures

A remote shell script was used to ensure that the timing of each run was as consistent with other runs as possible. The script first started the OPRA FH instances. After they had loaded their symbol caches, the *mcpub* instances were started, and data was played back from the start of the file, which contained approximately 5 minutes of play time before market open. During this period (immediately after the *mcpub* instances were started), the *mamaperf* clients were started, and each *mamaperf* requested and received its initial images from the OPRA FH instances. Thus, each client was able to instantiate its watchlist during the period before market open, as is standard practice for real trading applications. The *mcpub* instances were configured to pause at a synchronization point in their playback files just before 09:30 (market open) in the data, allowing them to begin market-rate playback simultaneously. The *mamaperf* clients were configured to record statistics for fifteen minutes from when they started, yielding at least eight minutes of data from market open.

In the initial set of test runs, *mcpub* was configured to play back at the same rate at which the data were originally recorded, or 1x recorded rate. Three runs were performed over UDP/GigE and three over DAL/InfiniBand-SDR.

Next, we determined the highest integer multiple of the recorded rate that the system could sustain in this configuration. That turned out to be 4. So in the second round of tests, *mcpub* was configured to play back at 4x recorded rate while preserving the relative timing of updates to whatever degree it could manage (we were unable to verify the timing fidelity of playback; see [Limitations, page 16](#)). In this case,

the *mcpub* instances were instructed to begin playback at 1x recorded rate, and increase to 4x recorded rate at around 20 seconds after market-open. This resulted in four times as much data being played back over the remainder of the run.

## Data Observations

The Wombat software provides for in-line time-stamping at several points through the data path. In this case, we focused on transport latency—roughly, the time it takes from the moment the feed handler publishes data to the network stack to the time the application receives parsed data.

In terms of Wombat-defined timestamps, this meant the *mamaperf* timestamp minus *mamaSendTime*. The *mamaSendTime* is a conservative approximation of the publication time, since it occurs just before the feed handler hands an update off to LBM, not when LBM hands off to the network stack.

The *mamaperf* calculates time deltas for each update it receives. At the end of each ten-second interval, *mamaperf* calculates latency statistics (mean, standard deviation, minimum, maximum) for the last 10 seconds, as well as the number of messages received in the interval and the CPU and memory at the end of that interval. It writes these 10-second statistics to a file.

## Time Synchronization

Timestamp accuracy was managed according to standard Wombat procedures. A server that does not participate in data traffic acts as an NTP server. Each machine runs *ntpdate* once per second, which resets its clock to that of the NTP server. NTP daemons are not run. *Ntpdate* communicates with the master clock over a quiet network that is not carrying OPRA data.

## Limitations

A few aspects of the test procedure have known limitations:

- Time synchronization—NTP using CPU clocks has limited accuracy in the sub-millisecond range, and using *ntpdate* rather than NTP daemons does not take advantage of the corrective algorithms in NTP. An explicit assumption in this report is that while some of the sub-millisecond jitter may be due to clocks, the error is unbiased.
- Data granularity—The Wombat capture tool (*mamaperf*) does not preserve underlying data points after it calculates latency statistics for an interval. This limits our ability to obtain statistics over multiple intervals such as standard deviation, percentiles, etc. It also limits our ability to understand behavior within an interval, such as 1-second or 1-millisecond spikes.
- Market timing replication—As described above, the intent of *papareplay* is to play data back with its original spikiness. However, our observations of the 10-second-interval update rate data from *mamaperf* showed that at 4x recorded rate, much of the spikiness seemed to get smoothed out. We could not ascertain the spikiness of the data within the 10-second intervals (see previous bullet). Some of the smoothness is no doubt due to the fact that the update rate is reported by the *mamaperf* consumer, which means that updates may have been buffered at one or more points before arrival. Nevertheless, the overall update rate was indeed four times the original update rate, and *mcpub* output was considerably spikier than it is when *papareplay* is not used.
- Source data—The recorded OPRA data was from April 2, 2007. This has two drawbacks:
  - a. It is now well past April, so data rates have increased substantially since then.
  - b. This day may not have been as busy as other days. Therefore, a 4x recorded rate playback does not necessarily mean four times April's maximum rates.

## Testing Results

### Mean Latency

The sections below tabulate the mean latencies for test runs at varying rates of playback data. Multiple runs were performed to ensure the consistency of results.

#### Playback at 1x Recorded Rate

When *mcpub/papareply* was programmed to play back at the same rate at which the data had been recorded, the average update rate across all runs was 10.5 Kups in aggregate for the three channels with a 10-second peak of approximately 154 Kups. This would correspond to a 10-second peak of 1.23 Mups for the full OPRA feed.

Table 6 show the latency results for twelve clients consuming data from the feed handler over three multicast channels, each corresponding to an OPRA line. The tests were run three times.

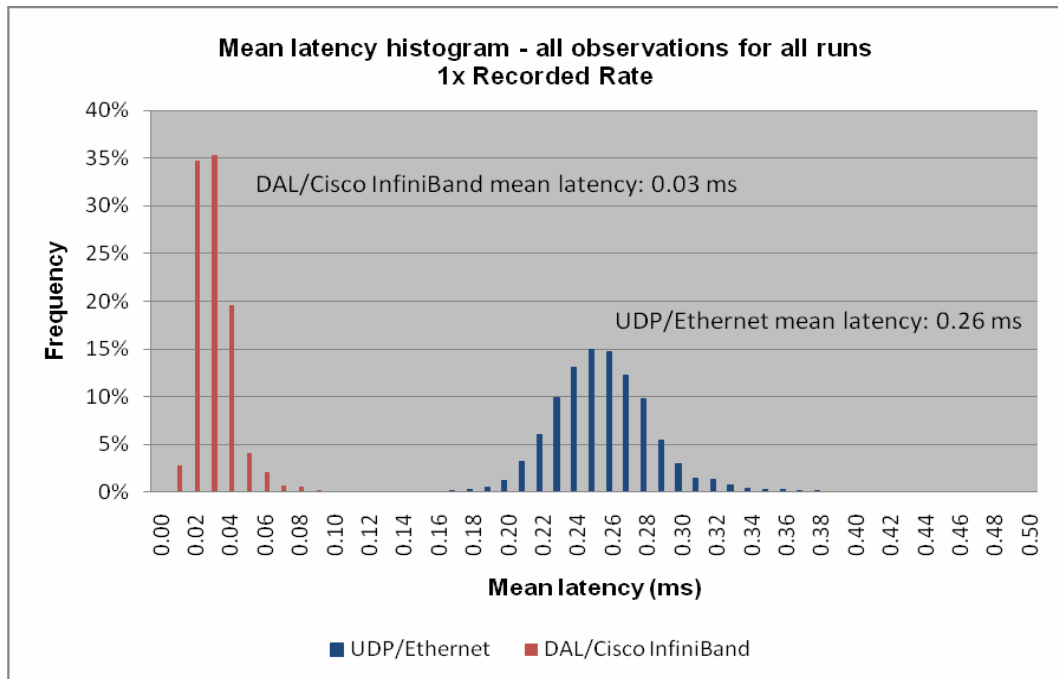
**Table 6** Latency Test Results

1x Recorded Rate		Mean Latency (milliseconds)					
		InfiniBand			Ethernet		
		Run 1	Run 2	Run 3	Run 1	Run 2	Run 3
Line 13	1	0.028	0.042	0.030	0.259	0.256	0.255
	2	0.038	0.039	0.038	0.262	0.264	0.262
	3	0.027	0.028	0.036	0.256	0.255	0.256
	4	0.021	0.022	0.023	0.249	0.242	0.242
Line 15	1	0.027	0.053	0.034	0.267	0.262	0.254
	2	0.039	0.038	0.037	0.264	0.272	0.265
	3	0.028	0.028	0.030	0.256	0.256	0.261
	4	0.020	0.020	0.024	0.249	0.244	0.247
Line 17	1	0.027	0.031	0.029	0.259	0.262	0.259
	2	0.038	0.039	0.037	0.274	0.270	0.266
	3	0.029	0.028	0.026	0.260	0.260	0.261
	4	0.021	0.022	0.024	0.248	0.243	0.252
<b>Mean</b>		0.028	0.033	0.031	0.259	0.257	0.257
<b>Mean of all clients, all runs</b>		0.031			0.258		

The overall mean latency for the Cisco InfiniBand was 30  $\mu$ sec versus 260  $\mu$ sec for Gigabit Ethernet. This is an 88 percent reduction in overall mean latency.

Figure 9 shows a histogram of the mean latency observations for both Cisco IB and Ethernet.

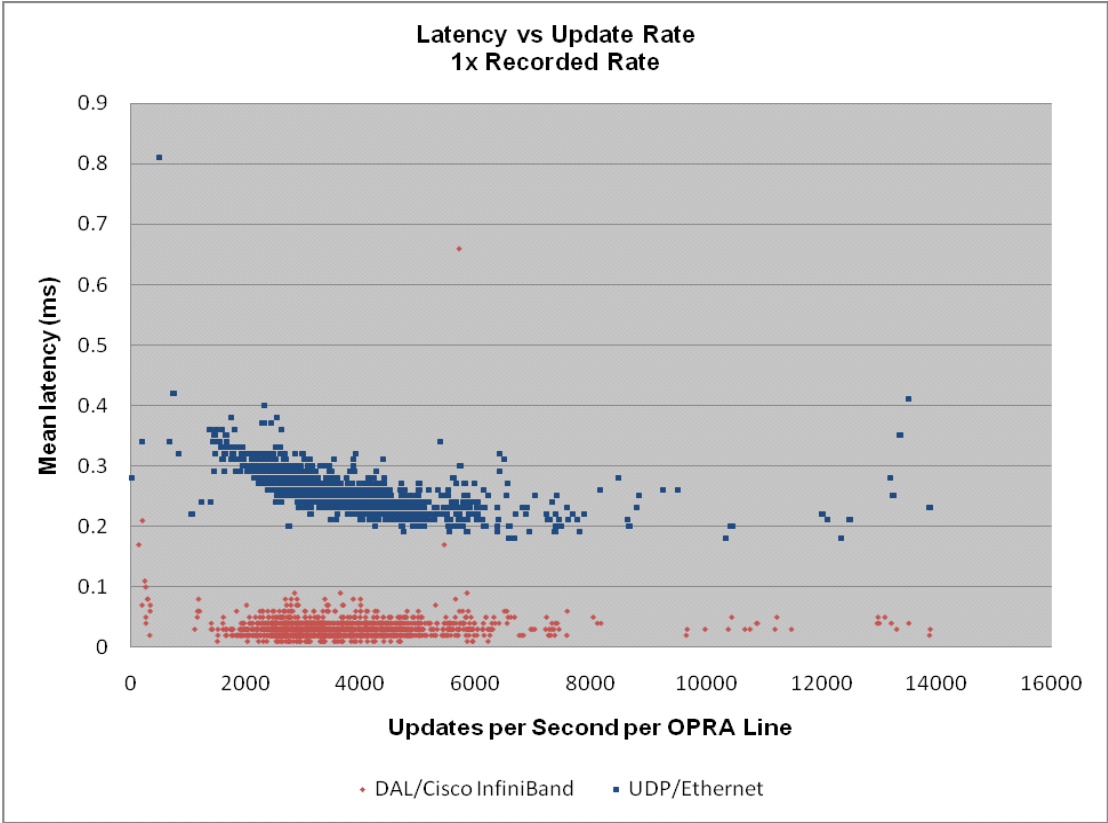
**Figure 9** *Transport Mean Latency at 1x Recorded Rate*



Because the update rate varied over time, it is important to check whether there is a correlation between update rate and latency. Live market data does not flow at a steady state. It has peaks and troughs and bursts of updates over small and large intervals. Yet many latency-sensitive applications value predictability of latency. The less variable a system's latency is as traffic ebbs and flows, the more beneficial it is.

Figure 10 plots updates per second (per line/client) against mean latency. Ethernet exhibited decreasing mean latency with increasing update rate. Note that this is not the ingress update rate (i.e., the rate at which arrived at the OPRA FH machine and the Wombat tools do not measure). It is the rate at which updates arrived at the mamaperf consumer. This makes it more difficult to draw conclusions about cause and effect, but it is likely due to batching and flushing behavior that becomes more efficient with higher update rates that InfiniBand does not exhibit the same behavior is consistent with this explanation, since there is much less buffering.

Figure 10 Latency vs Update Rate at 1x Recorded Rate



Consistent latency in the face of differences in 10-second update rates is one illustration of the ability of the DAL/IB solution to improve predictability.

## 4x Recorded Rates

When mcpub/papareply was programmed to play back four times faster than the recording, the average update rate across all runs was 41 Kups in aggregate for the three channels with a 10-second peak of approximately 75 Kups, which corresponds to a 10-second peak of 603 Kups for a full OPRA feed.

The [Table 7](#) show the latency results. The mean latency across all runs was 50  $\mu$ sec for DAL/IB and 240  $\mu$ sec for UDP/Ethernet. Ethernet latency was lower. This is consistent with the effect previously noted and suggests that this is due to batching or other efficiencies that kick in as the update rates increase.

**Table 7**      **4X Recorded Rates**

4x Recorded Rate		Mean latency (milliseconds)					
		InfiniBand			Ethernet		
		Run 1	Run 2	Run 3	Run 1	Run 2	Run 3
<b>Line 13</b>	<b>1</b>	0.045	0.048	0.052	0.245	0.241	0.244
	<b>2</b>	0.056	0.054	0.054	0.245	0.245	0.249
	<b>3</b>	0.045	0.051	0.047	0.238	0.241	0.241
	<b>4</b>	0.035	0.038	0.033	0.219	0.224	0.221
<b>Line 15</b>	<b>1</b>	0.045	0.049	0.048	0.259	0.258	0.255
	<b>2</b>	0.055	0.058	0.056	0.261	0.258	0.260
	<b>3</b>	0.044	0.052	0.054	0.249	0.253	0.250
	<b>4</b>	0.033	0.043	0.033	0.230	0.234	0.236
<b>Line 17</b>	<b>1</b>	0.048	0.053	0.052	0.239	0.240	0.236
	<b>2</b>	0.051	0.062	0.069	0.243	0.245	0.248
	<b>3</b>	0.046	0.049	0.043	0.239	0.237	0.239
	<b>4</b>	0.038	0.038	0.032	0.216	0.219	0.219
<b>Mean</b>		0.045	0.050	0.048	0.240	0.241	0.242
<b>Mean of all clients &amp; runs</b>		0.047			0.241		



Figure 11 plots a histogram of the mean latency observations for both Cisco IB and Ethernet.

Figure 11 Mean latency histogram at 4x Recorded Rate

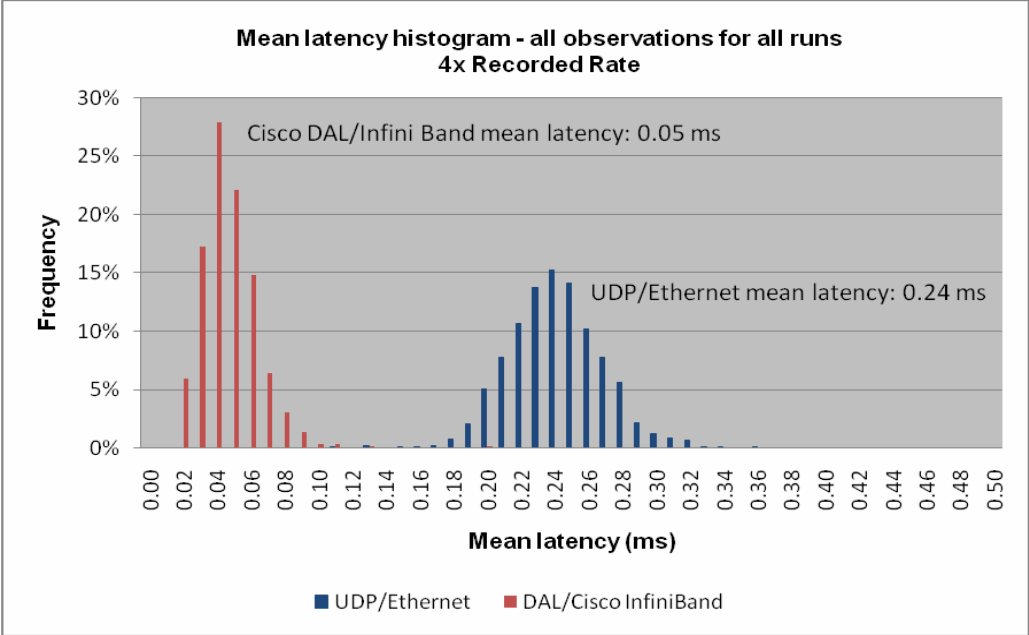
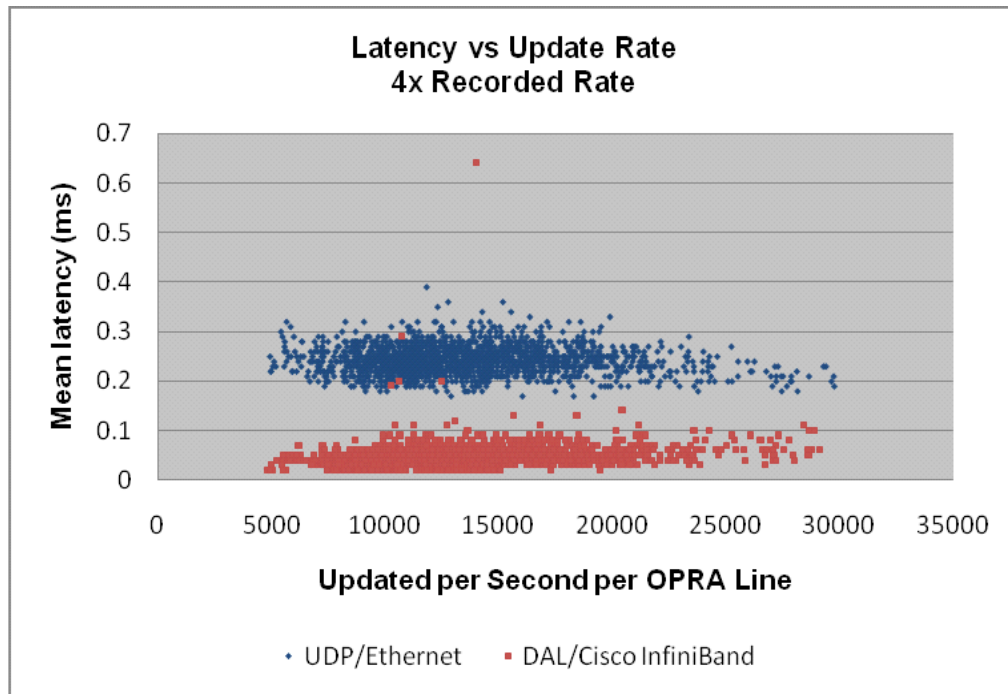


Figure 12 plots the updates per second (per OPRA line/client) against mean latency. Unlike at 1x recorded rate, the Ethernet latencies do not fall appreciably with update rate. This is consistent with the previous hypothesis (that the inverse relationship was due to batching and flushing). At the rates experienced in the 4x scenario, the buffering and flushing may have been maxed out.

**Figure 12** Latency vs Update Rate at 4x Recorded Rate



## Latency Dispersion

Latency dispersion—or roughly, how spread out the latency values are—is just as important as the average latency. Like the man who drowned in a river that was six inches deep on average, trader's do not care a lot about mean latency if the quote or order they cared the most about was delayed far beyond the mean. Moreover, if mean latency is low enough, reducing dispersion can actually be more important than reducing mean latency, since dispersion befuddles trading algorithms with unpredictability.

Dispersion is an abstract concept that is captured in a variety of statistics. Below, we look at standard deviation and maximum latencies.

## Standard Deviation

Table 8 and Table 9 show the results for standard deviation for each client during the runs, for the 1x playback rate and the 4x playback rate, respectively.

**Table 8** *Standard Deviation for 1X Playback Rate*

1x Recorded Rate		Standard Deviation latency (milliseconds)					
		InfiniBand			Ethernet		
		Run 1	Run 2	Run 3	Run 1	Run 2	Run 3
<b>Line 13</b>	<b>1</b>	0.085	0.114	0.088	0.647	0.651	0.653
	<b>2</b>	0.081	0.057	0.087	0.652	0.651	0.647
	<b>3</b>	0.045	0.066	0.081	0.648	0.651	0.648
	<b>4</b>	0.073	0.086	0.072	0.652	0.650	0.648
<b>Line 15</b>	<b>1</b>	0.081	0.221	0.101	0.658	0.654	0.655
	<b>2</b>	0.077	0.061	0.082	0.644	0.664	0.661
	<b>3</b>	0.053	0.070	0.058	0.644	0.671	0.656
	<b>4</b>	0.059	0.082	0.078	0.648	0.645	0.643
<b>Line 17</b>	<b>1</b>	0.077	0.059	0.090	0.659	0.672	0.663
	<b>2</b>	0.074	0.064	0.083	0.671	0.663	0.666
	<b>3</b>	0.049	0.072	0.042	0.669	0.668	0.670
	<b>4</b>	0.063	0.093	0.081	0.654	0.649	0.662
<b>Mean</b>		0.068	0.087	0.079	0.654	0.658	0.656
<b>Mean of all clients, all runs</b>		0.078			0.656		

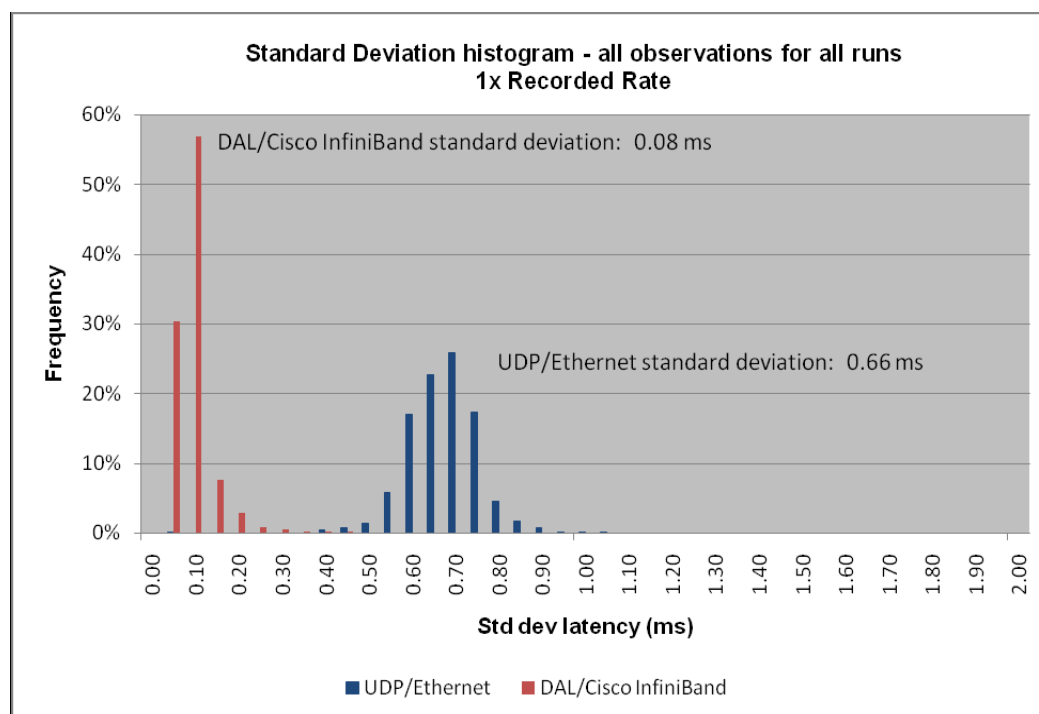
**Table 9**      **4x Playback Recorded Rate**

4x Recorded Rate		Mean latency (milliseconds)					
		InfiniBand			Ethernet		
		Run 1	Run 2	Run 3	Run 1	Run 2	Run 3
<b>Line 13</b>	1	0.123	0.124	0.140	0.428	0.429	0.426
	2	0.131	0.097	0.109	0.416	0.413	0.415
	3	0.075	0.116	0.088	0.415	0.414	0.424
	4	0.075	0.088	0.082	0.406	0.412	0.411
<b>Line 15</b>	1	0.116	0.124	0.140	0.467	0.457	0.458
	2	0.151	0.112	0.109	0.453	0.441	0.440
	3	0.083	0.096	0.088	0.448	0.451	0.447
	4	0.067	0.086	0.082	0.431	0.434	0.448
<b>Line 17</b>	1	0.127	0.168	0.127	0.418	0.421	0.423
	2	0.107	0.164	0.237	0.420	0.415	0.420
	3	0.079	0.084	0.077	0.420	0.411	0.418
	4	0.128	0.072	0.080	0.393	0.397	0.405
<b>Mean</b>		0.105	0.111	0.113	0.426	0.425	0.428
<b>Mean of all clients, all runs</b>		0.110			0.426		

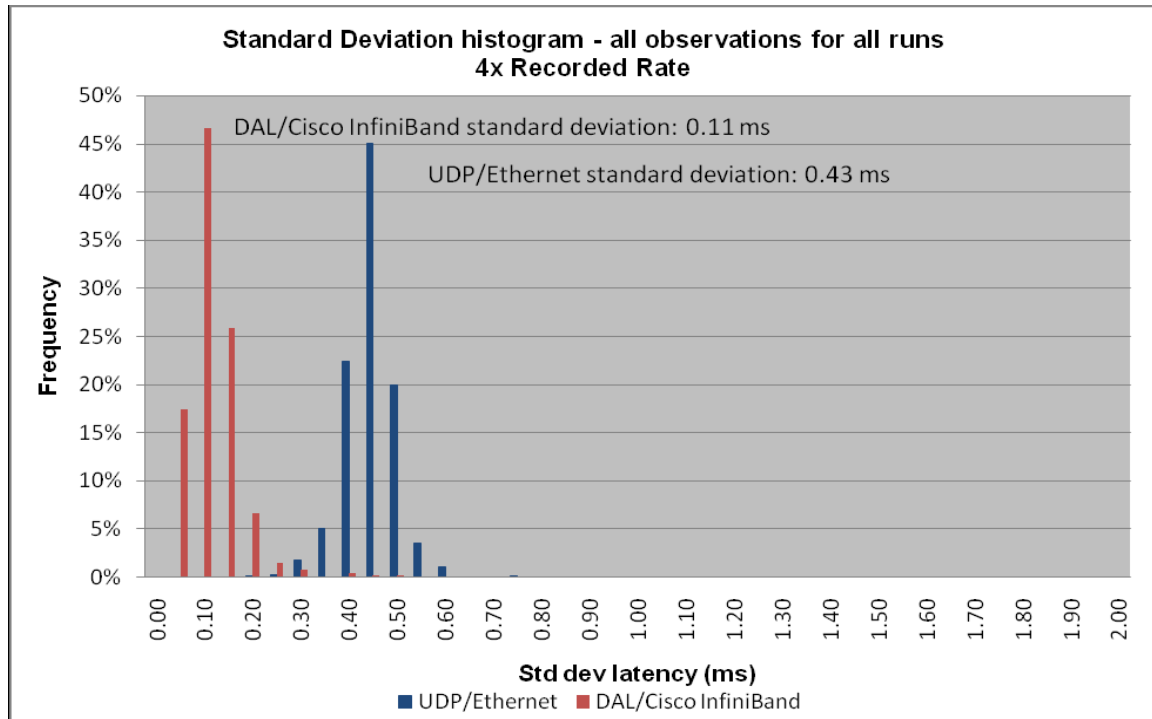
The standard deviation for the Cisco InfiniBand is 110  $\mu$ sec versus 430  $\mu$ sec for ethernet. This is a 74 percent reduction in the standard deviation. During analysis of the data, several outliers were observed in the ethernet data. These values occurred in almost every run, on every client, and very close to the same spot during each run. It is believed that these outliers can be attributed to the clock adjustment or something else on the network that was extraneous to the test. It was decided to eliminate the outliers. This makes the analysis of DAL's benefits more conservative. A few outliers were noted in the InfiniBand data, however because they did exhibit the same patten as the outliers on Ethernet, those points were left into the analysis.

Figure 13 and Figure 14 show a histogram of standard deviations for the 1x playback rate and at the 4x playback rate. In both charts, the InfiniBand data shows a very narrow distribution of standard deviations, or low latency dispersion. The very low standard deviations indicate that the InfiniBand solution exhibits good predictability and low jitter.

**Figure 13** Standard deviation of latency histogram at 1x Recorded Rate



**Figure 14** Standard Deviation of Latency Histogram at 4x Recorded Rate



## Max Latencies

Figure 15 and Figure 16 show the distribution of latencies at 1x recorded rate and at 4x recorded rate. At 1x recorded rate the Infiniband solution the mean of the maximum latencies for DAL/Cisco InfiniBand is 1.4 milliseconds and with UDP/Ethernet it is 5.7 milliseconds. At 4x recorded rate, the means of the two networks move closer. At the higher rate, the mean of the maximum latencies is 4.9 milliseconds on DAL/Cisco InfiniBand and 6.9 milliseconds with UDP/Ethernet.

**Figure 15** *Max Latency Histogram at 1x Recorded Rate*

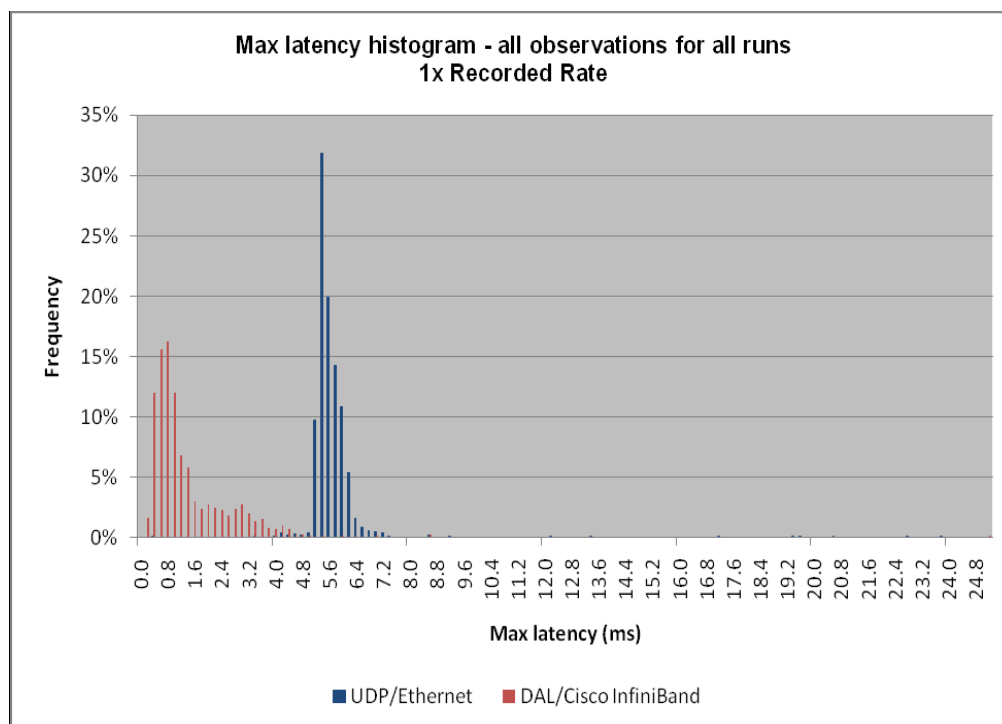
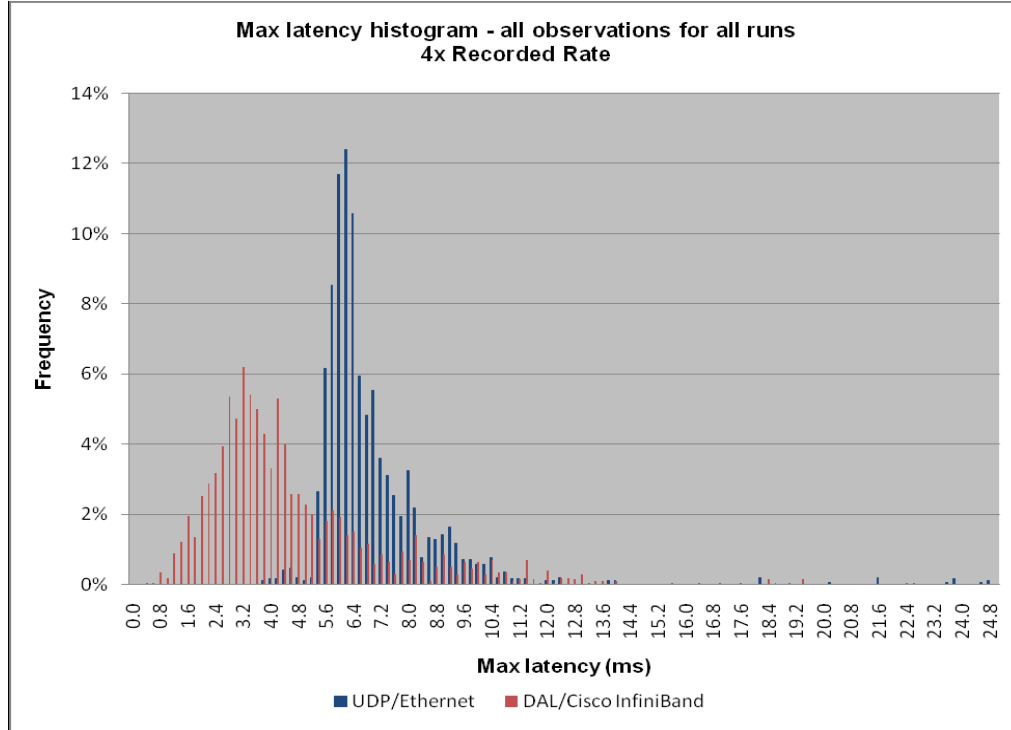


Figure 16 Max Latency Histogram at 4x Recorded Rate



## Appendix A—Device Configuration

This section provides sample configurations for the two devices used in the solution; the Cisco 6500 Catalyst Switch and the SFS 7000.

### Catalyst Switch Configuration

```
en-6509-1#show version
Cisco Internetwork Operating System Software
IOS (tm) s72033_rp Software (s72033_rp-ADVENTERPRISEK9_WAN-VM), Version 12.2(18)
SXF10, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2007 by cisco Systems, Inc.
Compiled Fri 13-Jul-07 08:58 by kellythw
Image text-base: 0x01020150, data-base: 0x01021000

ROM: System Bootstrap, Version 12.2(17r)S2, RELEASE SOFTWARE (fc1)
BOOTLDR:
en-6509-1 uptime is 13 weeks, 5 days, 16 hours, 1 minute
Time since en-6509-1 switched to active is 13 weeks, 5 days, 16 hours, 1 minute
System returned to ROM by reload at 02:17:40 UTC Sat Aug 25 2007 (SP by reload)
System image file is "disk0:s72033-adventerprisek9_wan-vz.122-18.SXF10.bin"
```

This product contains cryptographic features and is subject to United States and local country laws governing import, export, transfer and use. Delivery of Cisco cryptographic products does not imply



third-party authority to import, export, distribute or use encryption. Importers, exporters, distributors and users are responsible for compliance with U.S. and local country laws. By using this product you agree to comply with applicable laws and regulations. If you are unable to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at: <http://www.cisco.com/wwl/export/crypto/tool/stqrg.html>

If you require further assistance please contact us by sending email to [export@cisco.com](mailto:export@cisco.com).

cisco WS-C6509-E (R7000) processor (revision 1.2) with 1015808K/32768K bytes of memory.

Processor board ID SMG0934NEVE

SR71000 CPU at 600Mhz, Implementation 1284, Rev 1.2, 512KB L2 Cache

Last reset from power-on

Bridging software.

X.25 software, Version 3.0.0.

SuperLAT software (copyright 1990 by Meridian Technology Corp).

TN3270 Emulation software.

11 Virtual Ethernet/IEEE 802.3 interfaces

108 Gigabit Ethernet/IEEE 802.3 interfaces

12 Ten Gigabit Ethernet/IEEE 802.3 interfaces

1917K bytes of non-volatile configuration memory.

65536K bytes of Flash internal SIMM (Sector size 512K).

Configuration register is 0x2102

Patching is not available since the system is not running from an installed image. To install please use the "install file" command

en-6509-1#**show configuration**

Using 11493 out of 1964024 bytes

```

!
upgrade fpd auto
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service counters max age 10
!
hostname en-6509-1
!
boot system flash disk0:s72033-adventerprisek9_wan-vz.122-18.SXF10.bin
logging snmp-authfail
enable secret 5 $1$XCzj$D2M9m8lEtkKzxSwa4wLz0
!
no aaa new-model
ip subnet-zero
!
!
no ip igmp snooping
!
no mls flow ip
no mls acl tcam share-global
mls ip multicast flow-stat-timer 9
mls cef error action freeze
!
!
!
!
```

```

!
!
fabric buffer-reserve queue
port-channel load-balance src-dst-mac
diagnostic cns publish cisco.cns.device.diag_results
diagnostic cns subscribe cisco.cns.device.diag_commands
!
redundancy
mode sso
main-cpu
auto-sync running-config
spanning-tree mode pvst
no spanning-tree optimize bpdu transmission
!
vlan internal allocation policy ascending
vlan access-log ratelimit 2000
!
!
no crypto ipsec nat-transparency udp-encaps
!
!
interface Port-channel1
switchport
switchport access vlan 302
switchport trunk encapsulation dot1q
switchport trunk native vlan 302
switchport mode access
no ip address
!
interface Port-channel10
switchport
no ip address
!
interface GigabitEthernet1/1
switchport
switchport access vlan 228
switchport mode access
no ip address
!
interface GigabitEthernet1/2
no ip address
shutdown
!
interface GigabitEthernet1/3
no ip address
shutdown
!
interface GigabitEthernet1/4
no ip address
shutdown
!
interface GigabitEthernet1/5
no ip address
shutdown
!
interface GigabitEthernet1/6
no ip address
shutdown
!
interface GigabitEthernet1/7
no ip address
shutdown
!
interface GigabitEthernet1/8

```

```
no ip address
shutdown
!
interface GigabitEthernet1/9
no ip address
shutdown
!
interface GigabitEthernet1/10
no ip address
shutdown
!
interface GigabitEthernet1/11
no ip address
shutdown
!
interface GigabitEthernet1/12
no ip address
shutdown
!
interface GigabitEthernet1/13
no ip address
shutdown
!
interface GigabitEthernet1/14
no ip address
shutdown
!
interface GigabitEthernet1/15
no ip address
shutdown
!
interface GigabitEthernet1/16
no ip address
shutdown
!
interface GigabitEthernet1/17
no ip address
shutdown
!
interface GigabitEthernet1/18
no ip address
shutdown
!
interface GigabitEthernet1/19
no ip address
shutdown
!
interface GigabitEthernet1/20
no ip address
shutdown
!
interface GigabitEthernet1/21
no ip address
shutdown
!
interface GigabitEthernet1/22
no ip address
shutdown
!
interface GigabitEthernet1/23
no ip address
shutdown
!
interface GigabitEthernet1/24
```

```
no ip address
shutdown
!
interface GigabitEthernet1/25
switchport
switchport access vlan 10
switchport mode access
no ip address
!
interface GigabitEthernet1/26
switchport
switchport access vlan 10
switchport mode access
no ip address
!
interface GigabitEthernet1/27
switchport
switchport access vlan 10
switchport mode access
no ip address
!
interface GigabitEthernet1/28
switchport
switchport access vlan 10
switchport mode access
no ip address
!
interface GigabitEthernet1/29
switchport
switchport access vlan 11
switchport mode access
no ip address
!
interface GigabitEthernet1/30
switchport
switchport access vlan 11
switchport mode access
no ip address
!
interface GigabitEthernet1/31
switchport
switchport access vlan 11
switchport mode access
no ip address
!
interface GigabitEthernet1/32
switchport
switchport access vlan 11
switchport mode access
no ip address
!
interface GigabitEthernet1/33
switchport
switchport access vlan 14
switchport mode access
no ip address
!
interface GigabitEthernet1/34
switchport
switchport access vlan 14
switchport mode access
no ip address
!
interface GigabitEthernet1/35
```

```

switchport
switchport access vlan 14
switchport mode access
no ip address
!
interface GigabitEthernet1/36
switchport
switchport access vlan 14
switchport mode access
no ip address
!
interface GigabitEthernet1/37
switchport
switchport access vlan 14
switchport mode access
no ip address
!
interface GigabitEthernet1/38
switchport
switchport access vlan 14
switchport mode access
no ip address
interface GigabitEthernet1/39
switchport
switchport access vlan 302
switchport mode access
no ip address
wrr-queue bandwidth 200 0 0
wrr-queue queue-limit 100 0 0
wrr-queue threshold 1 100 100 100 100 100 100 100 100
!
interface GigabitEthernet1/40
switchport
switchport access vlan 14
switchport mode access
no ip address
!
interface GigabitEthernet1/41
switchport
switchport access vlan 16
switchport mode access
no ip address
!
interface GigabitEthernet1/42
switchport
switchport access vlan 16
switchport mode access
no ip address
!
interface GigabitEthernet1/43
switchport
switchport access vlan 16
switchport mode access
no ip address
!
interface GigabitEthernet1/44
switchport
switchport access vlan 16
switchport mode access
no ip address
!
interface GigabitEthernet1/45
switchport
switchport access vlan 16

```

```
switchport mode access
no ip address
!
interface GigabitEthernet1/46
switchport
switchport access vlan 16
switchport mode access
no ip address
!
interface GigabitEthernet1/47
switchport
switchport access vlan 16
switchport mode access
no ip address
!
interface GigabitEthernet1/48
switchport
switchport access vlan 16
switchport mode access
no ip address
!
interface GigabitEthernet2/1
no ip address
shutdown
!
interface GigabitEthernet2/2
no ip address
shutdown
!
interface GigabitEthernet2/3
no ip address
shutdown
!
interface GigabitEthernet2/4
no ip address
shutdown
!
interface GigabitEthernet2/5
no ip address
shutdown
!
interface GigabitEthernet2/6
no ip address
shutdown
!
interface GigabitEthernet2/7
no ip address
shutdown
!
interface GigabitEthernet2/8
no ip address
shutdown
!
interface GigabitEthernet2/9
no ip address
shutdown
!
interface GigabitEthernet2/10
no ip address
shutdown
!
interface GigabitEthernet2/11
no ip address
shutdown
```

```
!  
interface GigabitEthernet2/12  
  no ip address  
  shutdown  
!  
interface GigabitEthernet2/13  
  no ip address  
  shutdown  
!  
interface GigabitEthernet2/14  
  no ip address  
  shutdown  
!  
interface GigabitEthernet2/15  
  no ip address  
  shutdown  
!  
interface GigabitEthernet2/16  
  no ip address  
  shutdown  
!  
interface GigabitEthernet2/17  
  no ip address  
  shutdown  
!  
interface GigabitEthernet2/18  
  no ip address  
  shutdown  
!  
interface GigabitEthernet2/19  
  no ip address  
  shutdown  
!  
interface GigabitEthernet2/20  
  no ip address  
  shutdown  
!  
interface GigabitEthernet2/21  
  no ip address  
  shutdown  
!  
interface GigabitEthernet2/22  
  no ip address  
  shutdown  
!  
interface GigabitEthernet2/23  
  no ip address  
  shutdown  
!  
interface GigabitEthernet2/24  
  no ip address  
  shutdown  
!  
interface GigabitEthernet2/25  
  switchport  
  switchport access vlan 11  
  switchport mode access  
  no ip address  
!  
interface GigabitEthernet2/26  
  switchport  
  switchport access vlan 11  
  switchport mode access  
  no ip address
```

```
!  
interface GigabitEthernet2/27  
  switchport  
  switchport access vlan 11  
  switchport mode access  
  no ip address  
!  
interface GigabitEthernet2/28  
  switchport  
  switchport access vlan 11  
  switchport mode access  
  no ip address  
!  
interface GigabitEthernet2/29  
  switchport  
  switchport access vlan 14  
  switchport mode access  
  no ip address  
!  
interface GigabitEthernet2/30  
  switchport  
  switchport access vlan 14  
  switchport mode access  
  no ip address  
!  
interface GigabitEthernet2/31  
  switchport  
  switchport access vlan 14  
  switchport mode access  
  no ip address  
!  
interface GigabitEthernet2/32  
  switchport  
  switchport access vlan 14  
  switchport mode access  
  no ip address  
!  
interface GigabitEthernet2/33  
  switchport  
  switchport access vlan 10  
  switchport mode access  
  no ip address  
!  
interface GigabitEthernet2/34  
  switchport  
  switchport access vlan 10  
  switchport mode access  
  no ip address  
!  
interface GigabitEthernet2/35  
  switchport  
  switchport access vlan 10  
  switchport mode access  
  no ip address  
!  
interface GigabitEthernet2/36  
  switchport  
  switchport access vlan 10  
  switchport mode access  
  no ip address  
!  
interface GigabitEthernet2/37  
  no ip address  
  shutdown
```



```
!  
interface GigabitEthernet2/38  
  switchport  
  switchport access vlan 10  
  switchport mode access  
  no ip address  
!  
interface GigabitEthernet2/39  
  switchport  
  switchport access vlan 302  
  switchport trunk encapsulation dot1q  
  switchport trunk native vlan 302  
  switchport mode access  
  no ip address  
  channel-group 1 mode on  
!  
interface GigabitEthernet2/40  
  switchport  
  switchport access vlan 302  
  switchport trunk encapsulation dot1q  
  switchport trunk native vlan 302  
  switchport mode access  
  no ip address  
  channel-group 1 mode on  
!  
interface GigabitEthernet2/41  
  switchport  
  switchport access vlan 302  
  switchport trunk encapsulation dot1q  
  switchport trunk native vlan 302  
  switchport mode access  
  no ip address  
  channel-group 1 mode on  
!  
interface GigabitEthernet2/42  
  no ip address  
  shutdown  
!  
interface GigabitEthernet2/43  
  switchport  
  switchport access vlan 10  
  switchport mode access  
  no ip address  
!  
interface GigabitEthernet2/44  
  switchport  
  switchport access vlan 10  
  switchport mode access  
  no ip address  
!  
interface GigabitEthernet2/45  
  switchport  
  switchport access vlan 10  
  switchport mode access  
  no ip address  
!  
interface GigabitEthernet2/46  
  switchport  
  switchport access vlan 10  
  switchport mode access  
  no ip address  
!  
interface GigabitEthernet2/47  
  switchport
```

```
switchport access vlan 11
switchport mode access
no ip address
!
interface GigabitEthernet2/48
switchport
switchport access vlan 14
switchport mode access
no ip address
!
interface GigabitEthernet5/1
no ip address
shutdown
!
interface GigabitEthernet5/2
no ip address
shutdown
!
interface TenGigabitEthernet7/1
switchport
switchport access vlan 50
switchport mode access
no ip address
!
interface TenGigabitEthernet7/2
switchport
switchport access vlan 50
switchport mode access
no ip address
!
interface TenGigabitEthernet7/3
switchport
switchport access vlan 50
switchport mode access
no ip address
!
interface TenGigabitEthernet7/4
switchport
switchport access vlan 50
switchport mode access
no ip address
!
interface TenGigabitEthernet8/1
no ip address
!
interface TenGigabitEthernet8/2
no ip address
!
interface TenGigabitEthernet8/3
no ip address
!
interface TenGigabitEthernet8/4
no ip address
!
interface TenGigabitEthernet8/5
no ip address
!
interface TenGigabitEthernet8/6
no ip address
!
interface TenGigabitEthernet8/7
no ip address
!
interface TenGigabitEthernet8/8
```

```

no ip address
!
interface Vlan1
no ip address
!
interface Vlan10
ip address 1.2.10.1 255.255.255.0
!
interface Vlan11
ip address 1.2.11.1 255.255.255.0
!
interface Vlan12
no ip address
shutdown
!
interface Vlan14
ip address 1.2.12.1 255.255.255.0
!
interface Vlan15
no ip address
shutdown
!
interface Vlan16
ip address 1.2.16.1 255.255.255.0
!
interface Vlan20
ip address 1.2.20.1 255.255.255.0
!
interface Vlan30
ip address 1.2.30.1 255.255.255.0
!
interface Vlan228
ip address 172.29.228.51 255.255.0.0
!
interface Vlan302
ip address 172.19.8.124 255.255.254.0
ip pim sparse-mode
ip igmp query-interval 10
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.29.228.1
!
no ip http server
!
!
!
control-plane
!
!
!
dial-peer cor custom
!
!
!
!
line con 0
line vty 0 4
exec-timeout 0 0
privilege level 15
password topspin
login
transport input lat pad udptn telnet rlogin mop ssh nasi acercon
line vty 5 15
login

```

```

transport input lat pad udptn telnet rlogin mop ssh nasi acercon
!
exception core-file
!
no cns aaa enable
end

```

## SFS 7000 Configuration (Core)

```
svbu-hs-ts120-8> show version
```

```

=====
                        System Version Information
=====
system-version : SFS-7000D TopspinOS 2.10.0-ALPHA releng #323 04/16/2007
23:28:29
        contact : tac@cisco.com
           name : svbu-hs-ts120-8
        location : 170 West Tasman Drive, San Jose, CA 95134
           up-time : 116(d):21(h):5(m):22(s)
        last-change : Tue Oct 30 15:20:25 2007
last-config-save : none
           action : none
           result : none
        oper-mode : normal

```

```
svbu-hs-ts120-8> show config
```

```

! TopspinOS-2.10.0/build323
! Thu Nov 29 10:42:03 2007
enable
config terminal
!
boot-config primary-image-source TopspinOS-2.10.0/build323
!
interface mgmt-ethernet
  addr-option static
  ip address 172.29.213.5 255.255.255.0
  gateway 172.29.213.1
  no shutdown
!
!
interface ib 1
  speed 4x-sdr
!
interface ib 2
  speed 4x-sdr
!
interface ib 3
  speed 4x-sdr
!
interface ib 4
  speed 4x-sdr
!
interface ib 5
  speed 4x-sdr
!
interface ib 6
  speed 4x-sdr
!
interface ib 7

```

```
    speed 4x-sdr
!
interface ib 8
    speed 4x-sdr
!
interface ib 9
    speed 4x-sdr
!
interface ib 10
    speed 4x-sdr
!
interface ib 13
    speed 4x-sdr
!
interface ib 14
    speed 4x-sdr
!
interface ib 15
    speed 4x-sdr
!
interface ib 16
    speed 4x-sdr
!
interface ib 17
    speed 4x-sdr
!
interface ib 18
    speed 4x-sdr
!
    speed 4x-sdr
!
interface ib 20
    speed 4x-sdr
!
interface ib 21
    speed 4x-sdr
!
interface ib 22
    speed 4x-sdr
!
interface ib 23
    speed 4x-sdr
!
interface ib 24
    speed 4x-sdr
!
!
hostname "svbu-hs-ts120-8"
!
```

# Appendix B—Building and Configuring Switches

## Definitions

**Table 10**      *Definition of Key Terms*

Term	Description
Blocking	Blocking topologies do not provide a 1:1 ratio of paths in and paths out. In a blocking topology, traffic may potentially contend for paths.
Non-blocking	Non-blocking topologies provide, for each path into a switch or network, an equal path out. Non-blocking topologies avoid oversubscription.
Cluster Node Number	A number that uniquely identifies every node in a cluster. For example, a 500-node cluster would have Cluster Node Numbers from 1 to 500.
Core Switch	Core switches form the second tier of an InfiniBand fabric and are used to interconnect Leaf Switches and Edge Switches. They form the backbone of the InfiniBand fabric. A typical Core Switch would be a 96-port Cisco SFS 7008.
Edge Switch	Edge switches provide support for InfiniBand I/O Gateways and are used to connect an InfiniBand fabric to external network and storage subsystems. A typical Edge Switch would be a 24-port Cisco SFS 3504 with support for up to 12 Gigabit Ethernet or Fibre Channel Gateways.
Host Channel Adapter (HCA)	An InfiniBand Host Channel Adapter (HCA) connects a server to the InfiniBand fabric. HCAs are either PCI-X or PCI-Express based. They are typically dual ported to allow for redundant InfiniBand connections, though single-port HCAs are available. Each port runs at 10 Gb/s, 20 Gb/s for double data rate (DDR).
Leaf Switch	Leaf switches form the first tier of an InfiniBand Fabric and are used to connect hosts to the Fabric. A typical Leaf Switch would be a 24-port Cisco SFS 7000.
Management Network	Also referred to as “Administration Network”. A 100 Base T or Gigabit Ethernet network used for out-of-band administration and management of the nodes and switches.
Pod	A leaf switch and all attached hosts.
Host	A single compute element of the cluster, for example a 1U server with one or more CPUs.
Rack Node Number	A number that uniquely identifies a node within a rack or frame (the terms “rack” and “frame” are frequently used interchangeably, but both refer to the structure that supports the nodes and switches in the cluster). Typically a Rack Node Number will range from 1 (lowest slot in a rack) to 42 (for a 42 RU Rack).

## The Basics

When you begin to design and implement a Cisco high-performance computing cluster over InfiniBand with Cisco Server Fabric Switches, it helps to understand the conventional wisdom in this section.

- Contact Cisco’s partner, Infrastructure Development Corporation (IDC), to prepare an installation plan. IDC has experience with this process and it will make your job much easier. This plan will include physical factors, such as air flow and cooling. Contact information is as follows:
  - [Dominick.Rappa@infrastructuredev.com](mailto:Dominick.Rappa@infrastructuredev.com)
  - [Tim.LaFazia@infrastructuredev.com](mailto:Tim.LaFazia@infrastructuredev.com)

- Focus on your out-of-band (Ethernet) network first. Verify that all of your hosts and switches are available on the out-of-band network before you bring up the InfiniBand network.



**Note** Do not try to bring up the cluster using the in-band IPoIB management interfaces.

- Break any given cluster into segments or “pods.” Bringing up a “pod” means bringing up all hosts connected to a leaf switch that is not logically connected to any core switches. This document describes the bring-up process in more detail below.
- Keep things in perspective: this process will probably take longer than you anticipate. This cluster involves numerous devices and two overlapping networks (in-band and out-of-band). Remember Murphy’s Law: *if anything can go wrong, it will*. Break the process up into smaller milestones and approach the network one piece at a time.

## Installation Task and Timing Overview

The amount of time and man-power required for installation will vary directly with the size of the cluster. As an example, a 4500+-node cluster took approximately 8 to 10 man-weeks to bring up the InfiniBand fabric. However, this example was an unusually challenging scenario because of the following factors:

- All racks were densely populated, with most using 41 of 42 U of space.
- Installation of the leaf switches were done after the racks had been populated with nodes and internal management cabling, leaving very little free working space.
- Leaf switches were forced to be installed in a manner that greatly limited accessibility, both for switch racking within the rack and for connecting cables.

As a general guideline, 85 percent of the installation time is spent performing tasks associated with cable management, including the following:

- Cable labeling
- Connecting all InfiniBand cables to nodes and switches
- Debugging and replacing cables throughout the bring-up process



**Note** Unexpected issues are certain to arise, and installation complexity is certain to grow with the size of a cluster, regardless of previous experience or expectations.

## The Very First Thing That You Do: Plan

To plan for your cluster bring-up, know *everything* in [Table 11](#) before you take *any* action:

**Table 11** *Planning Requirements*

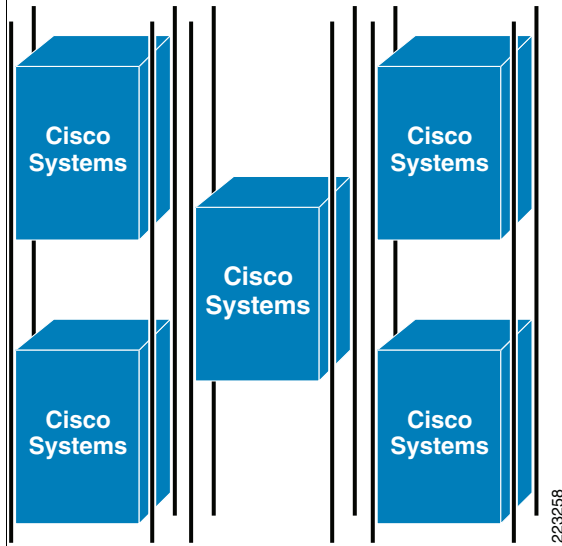
Issue	Requirement
Will the fabric be oversubscribed (“blocking) or not (non-blocking)?	The physical layout of the cluster depends on the subscription attribute of the fabric, so you must answer this question before you begin <i>any</i> physical installation.

**Table 11** *Planning Requirements (continued)*

Where do I put my core switches?

Core switches should reside in racks that contain no hosts and at most one additional core switch. Racks for core switches should have side panels removed. Rack space immediately to the left and immediately to the right of all core switches should be vacant because cables will feed out from the core switches into this space. If you must rack core switches next to one another, stagger them to provide the required cable space (see [Figure 17](#)).

**Figure 17** *Staggered Core Switches in Racks*



Avoid having hosts, switches, or other devices reside in rack space immediately to the left or right of any core switch.



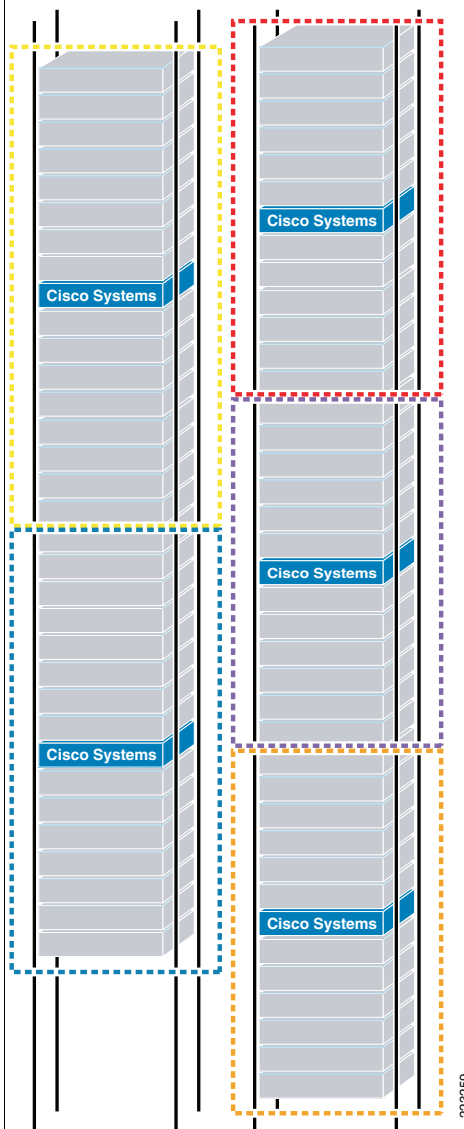
Table 11 Planning Requirements (continued)

Where do I put my leaf switches?

Leaf switches typically reside in the same racks as hosts. The number of leaf switches per rack and the placement of each switch in the rack depends on the blocking factor of the fabric. Refer to the [Definitions, page 42](#) for details on blocking/subscription.

Cisco provides two common rack configuration models. The first model, for oversubscribed fabrics, uses two leaf switches with up to 32 1U servers per rack (50 percent blocking). The second model, for non-blocking fabrics, uses three leaf switches with up to 36 1U servers per rack (see [Figure 18](#)).

**Figure 18** Leaf Switches in the 2 Common Rack Configurations



223259

**Table 11** Planning Requirements (continued)




	 <p><b>Note</b> Dashed borders in <a href="#">Figure 18</a> delineate pods.</p> <hr/> <p>Remember, do not rack anything yet! Just keep in mind where it is all going. This section is about planning, not executing.</p>
How many HCAs do I need?	You need at least one HCA per host. You can install one-port HCAs or two-port HCAs.
Where do I put my hosts?	<p>Ideally, 32 to 36 hosts reside in each non-core rack, along with 2 or 3 leaf switches. At this time, note the number of hosts that you plan to bring up and the number of racks that they will require. When it becomes time to assign IP addresses, the best course of action is to match certain addresses to certain racks. Ideally, each node then receives an address with the following format:</p> <p><b>10.0.rackNumber.nodeNumber</b> (e.g. 10.0.2.1 applies to the first host in the second rack).</p>
What cable lengths do I need, and how many of each cable do I need?	<p>After you plan where your switches and hosts will reside in your data center, measure your cabling requirements. All hosts connect to leaf switches. All leaf switches connect to all core switches.</p>  <p><b>Note</b> All cables from leaf switches to core switches must run overhead or under the floor, so allow for room for cables to reach up to and down from overhead tracks.</p> <hr/> <p>Failure rates for InfiniBand cables average at 1%. Be sure to include an additional 1% of cables for <i>each cable size</i>.</p>
How many Velcro straps do I need to organize cables, and what size straps do I need?	Cisco recommends that you buy bulk double-sided Velcro straps so you can 1) cut as many straps as you need and 2) cut all straps to size.
How do I configure my Ethernet management network?	<p>You must develop, independently, a plan for Ethernet installation, management, and networking. This document deals with InfiniBand installation and management only, but requires an Ethernet management network as part of the installation.</p>  <p><b>Note</b> You must successfully bring up your Ethernet network after you rack all hardware but before you begin to bring up pods in the InfiniBand network.</p>
How many IP addresses will I need?	<p>You will need IP addresses for all of the following:</p> <ul style="list-style-type: none"> <li>• Each host (to run the IPoIB protocol)</li> <li>• Each SFS chassis (for connectivity via an Ethernet management network)</li> </ul>

Table 11 Planning Requirements (continued)

How will I identify my switches and hosts?	<p>You should create naming conventions that address the following components:</p> <ul style="list-style-type: none"> <li>• Rack names</li> <li>• Host names</li> <li>• Switch names</li> <li>• Rack in which a given host resides</li> <li>• Rack in which a given switch resides</li> </ul> <p>In the event that the organization for which you are installing the cluster already has established naming conventions, defer to the existing rules. If you must invent conventions, keep in mind the following options:</p> <ol style="list-style-type: none"> <li>1. Name racks (e.g., Rack2, coreRack1, leafRack2).</li> <li>2. Name core switches according to function and rack location (e.g., R2C2, R5L1).</li> <li>3. Name hosts according to rack location (R2slot3, R3slot4)</li> </ol> <p>Alternatively, name devices based on rack and IP addresses (e.g. Rack2-125, where the IP address is 172.168.0.125). In these instances, apply IP addresses to the hosts in ascending order as you go down the rack. Ideally, match IP addresses to the rack (e.g., 172.168.4.x for Rack 4 and 172.168.5.x for rack 5).</p> <p>Create a topology map of your planned cluster. Create an inventory of your planned cluster in a text file.</p>
What devices will I connect with IB cables?	<p>Each host in the fabric connects to <i>one</i> leaf switch (if each host is using one port). The switch and the hosts that connect to the switch comprise a pod. Most commonly, 2 or 3 pods reside in each non-core rack.</p> <p>Each leaf switch connects to <i>every</i> core switch. Cable the same port on every leaf switch to the same core switch. (For instance, every “port 24” on the leaf switches connect to the same core switch.) Design symmetry into the network. If possible, each leaf switch should have the same number of connections to each core switch.</p>
Do I run the embedded Subnet Manager or the Cisco High-Performance Subnet Manager? How many backup SMs do I run?	<p>For any given fabric, you should configure a master subnet manager with one backup subnet manager. The embedded subnet manager that automatically arrives on all Cisco SFS chassis cannot synchronize its database with the Cisco High-Performance Subnet Manager. You must choose your subnet management method before you begin to build your fabric. Generally speaking, use the host-based SM for large fabrics or fabrics using only large (144 – 288 port) switches and the embedded SM for small fabrics (containing 7000 and/or 7008 switches). For details, refer to the High-Performance Subnet Manager documentation.</p>

## Install Interface Cards in the Hosts

Install your HCA(s) in your hosts. For detailed instructions, refer to the installation guides that arrive with your HCA. Install your NIC(s) in your hosts (if necessary).

## Rack and Cable All Hardware

Rack and Cable all hardware as follows:

- 
- Step 1** Mount your switches, hosts, and any other chassis in your racks according to the plan that you developed in [The Very First Thing That You Do: Plan, page 43](#).
  - Step 2** Connect all Ethernet and InfiniBand cables and label each end of each cable with the two ports that the cable connects.
  - Step 3** Connect power cables.
- 



**Note**

All cables destined for ports 1 to 6 or 13 to 18 of a Cisco SFS 7000 switch should be routed through the left side of the rack. All cables destined for ports 7 to 12 or 19 to 24 of a Cisco SFS 7000 switch should be routed through the right side of the rack, if possible.

---



**Note**

All cables destined for ports 1 to 6 for all LIMs of a Cisco SFS 7008 switch should be routed through the left side of the rack. All cables destined for ports 7 to 12 of a Cisco SFS 7008 should be routed through the right side of the rack, if possible.

---

For any racks that use SFS 7008/7012/7024 chassis, you should feed cables from the free space to the left and the right through the free space in adjacent frames. (This presumes that you followed the recommendation in [Figure 17](#).) This is the golden rule: 288 ports in a single rack is not a problem as long as the cables can be fed horizontally without stress. Free space and slack management is the biggest concern. To manage slack, Cisco recommends that you plug cables into core switches first, then plug them into leaf switches. This gives you the opportunity to manage slack at the leaf switch instead of focusing all cabling management at the core.

## Write Down Your Cabling Connections

By now, you have named each of your switches and each of your hosts, and you have connected your switches and hosts with Ethernet cables and InfiniBand cables. You have labeled your cables so that you can select any port in your fabric and instantly identify the port to which it connects. Now, compile this information in a text file that indicates how all of your devices are interconnected.

For the Ethernet ports on your hosts, label the first Ethernet port with the convention *hostname-eth0* (e.g. R2H01-eth0). Label the second Ethernet port on your host (if applicable) *hostname-eth1* (e.g. R2H01-eth1). For the InfiniBand ports on your hosts, label the first IB port with the convention *hostname-ib0* (e.g. R2H01-ib0). Label the second Ethernet port on your host (if applicable) *hostname-ib1* (e.g. R2H01-ib1).

## Configure Ethernet Attributes of Leaf Switches

Enter the following series of commands on each leaf switch:

<b>Step 1</b>	Login: <b>super</b> Password: <b>xxxxx</b>	Log in to the switch.
<b>Step 2</b>	switch> <b>enable</b>	Enter Privileged Exec mode.
<b>Step 3</b>	switch# <b>configure terminal</b>	Enter Global Configuration mode.
<b>Step 4</b>	switch(config)# <b>hostname R2S101</b>	Configure a device name from your naming conventions. (The CLI prompt will not immediately reflect the name change.)
<b>Step 5</b>	switch(config)# <b>interface mgmt-ethernet</b>	Enter Management Ethernet Configuration submode.
<b>Step 6</b>	R2S101(config-if-mgmt-ethernet)# <b>ip address 10.0.2.101 255.255.0.0</b>	Apply an IP address to the switch.
<b>Step 7</b>	R2S101(config-if-mgmt-ethernet)# <b>gateway 10.0.0.1</b>	Apply a default gateway to the switch.
<b>Step 8</b>	R2S101(config-if-mgmt-ethernet)# <b>no shutdown</b>	Enable the Ethernet management port.
<b>Step 9</b>	R2S101(config-if-mgmt-ethernet)# <b>exit</b>	Return to Global Configuration mode.
<b>Step 10</b>	R2S101# <b>copy running-config startup-config</b>	Save the running configuration as the startup configuration.
<b>Step 11</b>	R2S101# <b>ping 10.0.0.1</b>	Ping the default gateway to verify Ethernet connectivity from the switch side.

## Configure Ethernet Attributes of Core Switches

Enter the following series of commands on each core switch:

<b>Step 1</b>	Login: <b>super</b> Password: <b>xxxxx</b>	Log in to the switch.
<b>Step 2</b>	switch> <b>enable</b>	Enter Privileged Exec mode.
<b>Step 3</b>	switch# <b>configure terminal</b>	Enter Global Configuration mode.
<b>Step 4</b>	switch(config)# <b>hostname R1S101</b>	Configure a device name from your naming conventions. (The CLI prompt will not immediately reflect the name change.)
<b>Step 5</b>	switch(config)# <b>interface mgmt-ethernet</b>	Enter Management Ethernet Configuration submode.
<b>Step 6</b>	R2S101(config-if-mgmt-ethernet)# <b>ip address 10.0.1.101 255.255.0.0</b>	Apply an IP address to the switch.
<b>Step 7</b>	R2S101(config-if-mgmt-ethernet)# <b>gateway 10.0.0.1</b>	Apply a default gateway to the switch.
<b>Step 8</b>	R2S101(config-if-mgmt-ethernet)# <b>no shutdown</b>	Enable the Ethernet management port.
<b>Step 9</b>	R2S101(config-if-mgmt-ethernet)# <b>exit</b>	Return to Global Configuration mode.
<b>Step 10</b>	R2S101# <b>copy running-config startup-config</b>	Save the running configuration as the startup configuration.
<b>Step 11</b>	R2S101# <b>ping 10.0.0.1</b>	Ping the default gateway to verify Ethernet connectivity from the switch side.

## Validate the Ethernet Management Network

Bring up the Ethernet management network according to the plan that you developed in [The Very First Thing That You Do: Plan, page 43](#).

- Set up Ethernet IP addresses on all switches and hosts.
- Verify logical connectivity to all switches and hosts.

## Set Up SE Tools on a Ethernet-attached Host

- Install expect software.
- Perl, TCL, Python.
- Collect tools from SVBU.

## Perform a Switch Chassis Inspection

Scan the physical hosts and switches for the following indicators:

Look for box problems. Check management LEDs on each SFS chassis using the following commands:

- **show diag fru error**
- **show diag post**

## Perform a Physical Inspection

Amber or red LEDs	Consult your hardware documentation for potential causes of amber and red LEDs and troubleshoot as specified.
Blinking green LEDs	Blinking green LEDs typically indicate a bad physical link. Perform the following steps: Firmly secure cable on both ends? then what?
Blinking green HCA LED	If a blinking green LED appears on an HCA, the HCA is probably bad and should be replaced.
Blinking green Cisco 7008 or 7008p LED	If a blinking green LED appears on a Cisco 7008(p), the Line Interface Module (LIM) is probably bad and should be replaced.

## (Optional) Record Leaf Switches and Hosts

- Create a NFS directory called **leaves**.
- For each leaf switch in your fabric, create a text file and assign the text file the same name as the leaf switch.
- In the text file, list each host that connects to the leaf switch, listing one host per line.



**Note**

Do not skip this task. The file that you create can be used during the troubleshooting process in conjunction with the SE Tools (discussed in “Appendix B: SE Tools”).

The text that follows is an example file named R2S101:

```
R2H01
R2H02
R2H03
R2H04
R2H05
R2H06
R2H07
R2H08
R2H09
R2H10
R2H11
R2H12
R2H13
R2H14
R2H15
R2H16
```

## Disable Uplinks on Leaf Switches

Access each leaf switch through the Ethernet management network and disable the uplinks to the core switches.

<b>Step 1</b>	Login: <b>super</b> Password: <i>xxxxx</i>	Log in to the switch.
<b>Step 2</b>	R2S101> <b>enable</b>	Enter Privileged Exec mode.
<b>Step 3</b>	R2S101# <b>configure terminal</b>	Enter Global Configuration mode.
<b>Step 4</b>	R2S101(config)# <b>interface ib 23-24</b>	Enter Interface Configuration submode for uplinks to core switches.
<b>Step 5</b>	R2S101(config-int-ib-23-24)# <b>shutdown</b>	Disable uplinks.
<b>Step 6</b>	R2S101(config)# <b>exit</b>	Return to Privileged Exec mode.

## Install Host-Side Drivers and Configure IP Addresses to InfiniBand Ports on Hosts

Log on to each host over the Ethernet management network and install Cisco InfiniBand drivers (from CD-ROM or NFS). Apply an IP address to the ib0 port and, if applicable, to the ib1 port.

### Install Drivers from a CD

<b>Step 1</b>	host login: <i>user-id</i> Password: <i>password</i>	Log in to your host.
<b>Step 2</b>	Host~ # <b>mount /media/cdrom</b>	Mount the CD-ROM.
<b>Step 3</b>	Host~ # <b>cd /media/cdrom</b>	Navigate to the CD-ROM..
<b>Step 4</b>	Host~ # <b>./tsinstall</b>	Enter the <b>tsinstall</b> command.

### Install Drivers from an ISO on NFS

<b>Step 1</b>	host login: <i>user-id</i> Password: <i>password</i>	Log in to your host.
<b>Step 2</b>	Host~ # <b>cd path/image</b>	Navigate to the ISO on your file system.
<b>Step 3</b>	Host:/path/image # <b>mount -o loop cisco.iso /mnt</b>	Mount the ISO.
<b>Step 4</b>	Host:/path/image # <b>cd /mnt</b>	Navigate to the ISO.
<b>Step 5</b>	Host: /mnt #./ <b>tsinstall</b>	Enter the <b>tsinstall</b> command.
<b>Step 6</b>	host:~ # <b>reboot</b>	Reboot your host.

### Apply IP Addresses to InfiniBand Ports (IPoIB Users Only)

<b>Step 1</b>	host login: <i>user-id</i> Password: <i>password</i>	Log in to your host.
<b>Step 2</b>	Host:~ # <b>ifconfig ib0 10.0.2.1 netmask 255.255.0.0</b>	Enter the <b>ifconfig</b> command with <ul style="list-style-type: none"> <li>the appropriate IB interface (ib0 or ib1 on a host with one HCA)</li> <li>the IP address that you want to assign to the interface</li> <li>the <b>netmask</b> keyword</li> <li>the subnet mask that you want to assign to the interface</li> <li>to configure the IB interface.</li> </ul>
<b>Step 3</b>	host:~ # <b>ifconfig ib0</b>  ib0 Link encap:Ethernet HWaddr 93:C1:2A:29:33:3E inet addr:10.0.2.1 Bcast:10.0.2.255 Mask:255.255.0.0 UP BROADCAST RUNNING MULTICAST MTU:2044 Metric:1 RX packets:2695034 errors:0 dropped:0 overruns:0 frame:0 TX packets:1195933 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:128 RX bytes:343087140 (327.1 Mb) TX bytes:67417660 (64.2 Mb)	(Optional) Enter the <b>ifconfig</b> command with the appropriate port identifier (ib0 or ib1) to verify the configuration.

### Generate a /etc/hosts File

Generate a /etc/hosts file for each host. Include all device hostnames (switches and servers). Add all of your IPoIB and/or Management Ethernet addresses to your DNS server or your /etc/hosts file.

### Persistence

Edit your network startup scripts appropriately to make your interface settings persistent.



## Troubleshoot “Bring Up” Pod

### Embedded SM

<b>Step 1</b>	Login: <b>super</b> Password: <i>xxxxx</i>	Log in to the switch.
<b>Step 2</b>	R2S101> <b>enable</b>	Enter Privileged Exec mode.
<b>Step 3</b>	R2S101# <b>terminal length 0</b>	Configure unlimited output from <b>show</b> commands.
<b>Step 4</b>	R2S101# <b>configure terminal</b>	Enter Global Configuration mode.
<b>Step 5</b>	R2S101(config)# <b>trace app 26 mod 10 level terse flow 0x1000</b>	Configure trace level for log tracking.
<b>Step 6</b>	R2S101(config)# <b>exit</b>	Returns to Privileged Exec mode.
<b>Step 7</b>	R2S101# <b>show logging end</b>	Displays the Subnet Manager log.



### High-Performance SM

- **config trace 2 0x1000**
- **tail -f var syslog messages** or **tail -f <syslog-server>**

### Isolate Problems and Prune Ports

It is important to constantly monitor the `ts_log` on the switch where the SM is running or on the High-Performance Subnet Manager. As every piece of equipment is added to the subnet, you *must* make sure the SM is able to complete its sweep of the subnet, and that there are no errors.

As you add each component to the subnet, watch the `ts_log` with the **show logging end** command (embedded) or **tail -f var syslog messages** command (HSM). When a new component is added, there should be a number of messages about “in service traps” and changes to the topology. However, they should quickly settle down, and after a few sweeps, the SM should stop logging. If new log entries continue to appear, there is a problem with the subnet. If you see continuous messages starting with “Configuration caused by,” then the SM has not been able to completely sweep the subnet, and is constantly retrying. You should address these errors before you add any additional components to the fabric.

Error	Course of Action
<p>Configuration caused by some ports in INIT state</p>	<ol style="list-style-type: none"> <li>1. Look for <b>Failed discover node test, node 00:05:ad:00:00:02:22:d0, port_num= 14, error code 1</b> in the log.                      Note that the message provides you with the device GUID (in this case, 00:05:ad:00:00:02:22:d0) and port number (14).</li> <li>2. Match the GUID to its SFS chassis and identify the chassis type.</li> </ol> <p> <b>Note</b> If the chassis type is a Cisco 7000, the external port number is the same as the port number reported in the ts_log. If the chassis is a Cisco 7008, refer to “Appendix C: Cisco SFS 7008 Port Mapping” to map the port listed in the log to the external port number of the chassis.</p> <ol style="list-style-type: none"> <li>3. Disable the offending port(s) and check the error log to verify that the log entries have stopped.</li> </ol>
<p>SM OUT_OF_SERVICE trap for                      GID=0xfe8000000000000005ad00000348e1</p>	<p> <b>Note</b> This message is not a problem when changes have occurred in the network. It is just a problem if there are ports which are bouncing between in-service and out-of-service, causing the SM to continuously sweep.</p> <ol style="list-style-type: none"> <li>1. Identify the subnet prefix in the log entry. As of the release of this document, the subnet prefix is always fe:80:00:00:00:00:00.</li> <li>2. Identify the GUID. The GUID immediately follows the subnet prefix. In our example, the GUID is 00:05:ad:00:00:03:48:e1.</li> <li>3. Find the port. For these log entries, the GUID usually represents a channel adapter port. If the GUID is odd the Node GUID will be the Port GUID - 1. In the case above, the Node GUID is: 00:05:ad:00:00:03:48:e0. You use the GUID in the message (the port GUID) to find the node guid (minus 1, possibly 2), and then find the switch port to which the node connects and take down the port.</li> <li>4. Run neighbor_to_cal.</li> <li>5. Shut down the port.</li> </ol>

**Look at Port Errors**

Monitor the port counters in the network. Expect to find numerous problems during bring-up.

SM	The SM can monitor thresholds on the error counters and the SM can notify you. Refer to the relevant SM documentation for details (Cisco SFS CLI guide or Cisco High-Performance SM User Guide).
Manual	We have get_counters script and reset_counters script. Reset counters clears the port counters throughout the network. Get counters gets the counters throughout the network. Counters_to_errs pulls the data from the get_counters script and identifies problems from the data. Error counters will accumulate on the fabric even when jobs don't run. Some errors, however, will only appear when jobs run.

In the event that the error counters indicate a problem, shut down the port(s).

### Port Shut-down

<b>Step 1</b>	Login: <b>super</b> Password: xxxxxx	Log in to the switch.
<b>Step 2</b>	R2S101> <b>enable</b>	Enter Privileged Exec mode.
<b>Step 3</b>	R2S101# <b>configure terminal</b>	Enter Global Configuration mode.
<b>Step 4</b>	R2S101(config)# <b>interface 1/5</b>	Enter Interface Configuration submode.
<b>Step 5</b>	R2S101(config-int-1/5)# <b>shutdown</b>	Shut down the problematic port.

### Run Step Troubleshoot “Bring Up” Pod On All Pods

Refer to the material under step [Troubleshoot “Bring Up” Pod, page 53](#). Create a checklist and cross off each pod as you complete the step.

### Connect “bring up” Pod to Core Switches One at a Time

For each core switch, repeat the steps under Step [Troubleshoot “Bring Up” Pod, page 53](#).

### Connect Pods to Core Switches

Connect each pod to all core switches. Connect 1 to 4 pods at a time to the core switches. Be sure to connect each pod to all of the core switches at once. Debug as per steps in [Troubleshoot “Bring Up” Pod, page 53](#).

### Troubleshooting after Pruning

Move components around to determine the source of the problem:


- Cable
- Port
- Chassis (interior port)
- HCA
- TCA/gateway

- Fabric module
- Return material authorization (RMA) the offending device

This means that there are bad links and the port is turned off.

- Q.** In the case of a 120: is it the port on the switch, the port on the HCA, or the cable?
- Q.** How do I tell what's bad?
- A.** Swap the switch port. Swap the HCA port.
- A.** Review the problem symptoms (from Step 15).
- A.** Re-enable the port that you shut down in step 15.
- A.** Verify that the problem recurs.
- A.** Start swapping components and track the component that the error follows.

If it was an internal link inside a 270g, follow these troubleshooting steps:

Troubleshooting Task	Details
1. Re-seat the cable on both ends.	If the error disappears,, you have fixed a faulty physical cable connection, and the problem is solved.
2. Move cable to a different switch port	If the error disappears, you have identified a faulty switch port.
3. Move cable to a different HCA port.	This step assumes that you have two HCA ports on your host or that you have an additional host with an HCA available. If the problem disappears, you have identified a faulty HCA port.
4. Remove cable	<p>If the problem persisted through the previous 2 steps, you have identified a faulty cable. Replace the cable. (If possible, try a different cable between the same two ports.)</p> <p> <b>Note</b> Most cables that go through the RMA process come back with no errors. Be sure to take the time to thoroughly test the cable before you send it back to Cisco. Make sure another cable works between the same 2 ports before you confirm that the original cable is faulty.</p>

<p>5. Internal link problem (Cisco 7008; similar steps apply to the Cisco 7012 and 7024)</p>	<p>Try to identify the bad fru:</p> <ul style="list-style-type: none"> <li>• node card</li> <li>• core card</li> <li>• backplane (almost never)</li> </ul> <p>Figure out failed ports. (Use the show diag fru error command at the switch CLI.) Identify the cards that create the failed connection. Begin by swapping out the relevant node card with another node card (do not introduce an outside card). Check and see if the location of the error follows the swapped card. If so, you have identified a faulty node card. If the error has not moved, the problem is with the core card or the backplane.</p> <p>Leave the swapped node cards in their new state and swap the relevant core card with another fabric module in the chassis. (Choose the module that will cause you the least amount of effort or overhead.) Check and see if the problem follows the module. If it does, you have identified a faulty SFM.</p> <p>If the problem has not moved at all, you have identified a faulty backplane. If the problem disappeared altogether during any of these steps, you probably had a seating problem.</p>
<p>6. Internal link problem (Cisco 3504)</p>	<p>These steps are similar to the steps for the Cisco 7008 but the blanking panels in the 360 cause links between the switch cards. In addition to looking for connections between switch cards and gateway cards and between fabric controller cards and switch cards, look for issues between switch cards and blanking panels.</p> <p>The switch cards on the Cisco 3504 are symmetric, so you cannot swap the cards with one another to determine which has a problem, because the problem will appear in the same location. Instead, you must swap one of the existing cards with a third card to diagnose problems.</p>

