OFA Interoperability Working Group

OFA-IWG Interoperability Test Plan
Release 1.28

March 27, 2009
DRAFT

OpenFabrics.org

Revision History

Revision	Release Date	
0.50	Apr 4, 2006	First FrameMaker Draft of the Interop Test Plan which was used in the March 2006 IBTA-OpenFabrics Plugfest.
0.51	Apr 25, 2006	Added DAPL and updated MPI.
0.511	June 1, 2006	Arkady Added iWARP.
0.52	May 30, 2006	Added Intel MPI.
0.53	June 6, 2006	Updated uDAPL section provided by Arkady.
0.54	June 13, 2006	Updated entire Test Spec based on changes made by Arkady to incorporate iWARP into the Test Spec.
0.80	June 14, 2006	Updated for the OFA conference in Paris and for BoD meeting. Added OFA logo and URL.
1.0	June 21, 2006	Released after review and approval at the OFA conference in Paris.
1.01	Aug 17, 2006	Updated the iWARP Equipment requirements in the General System Setup section.
1.02	Oct 31, 2006	Updated Table 4 for iSER, Table 5 for SRP, Table 10 for uDAPL and corresponding info in Tables 17,18 and 22 as per request by Arkady. Added new test section from Bob Jaworski for Fibre Channel Gateway.
1.03	Dec 10, 2006	Updated test procedures based on the October 2006 OFA Interop Event. Updated Fibre Channel Gateway test based on changes submitted by Karun Sharma (QLogic). Added Ethernet Gateway test written by Karun Sharma (QLogic).
1.04	Mar 6, 2007	Updated test procedures in preparation for the April 2007 OFA Interop Event
1.05	Mar 7, 2007	Updated iWARP test procedures based on review by Mikkel Hagen of UNH-IOL. Added missing results tables.
1.06	April 3, 2007	Updated for April 2007 Interop Event based on review from OFA IWG Meeting on 3/27/07.
1.07	April 3, 2007	Updated for April 2007 Interop Event based on review from OFA IWG Meeting on 4/3/07
1.08	April 4, 2007	Added list of Mandatory Tests for April 2007 Interop Event.
1.09	April 9, 2007	Updated Intel MPI based on review by Arlin Davis.
1.10	April 10, 2007	Updated after final review by Arlin Davis and after the OFA IWG meeting on 4/10/2007

Revision	Release Date	
1.11	Sep 7, 2007	Updated with the latest scripts developed by UNH IOL and based on the results from the April 2007 Interop Event
1.12	Sep 12, 2007	Updated the documents to embed the test scripts in the document.
1.13	Jan 22, 2008	Updated the documents for the March 2008 OFA Interop event. IPoIB updated along with Cover Page and the Test Requirements section.
1.14	Feb 11, 2008	Added the following tests: 1. Ethernet Switch Tests 2. IPoIB Connected Mode 3. RDMA Interop 4. RDS
1.15	Feb 18, 2008	Updates to the following tests: 1. Ethernet Switch Tests 2. IPoIB Connected Mode 3. RDMA Interop
1.16	Feb 25, 2008	Removed all reference to Low Latency Ethernet Switches. This is the version for the March 2008 Interop Event
1.17	March 3, 2008	Added HP-MPI
1.18	July 22, 2008	Updated HP-MPI based on results from the March 2008 Interop Event
1.19	July 28, 2008	Updated HP-MPI URL for the tests. Added section for Open MPI Updated MPI based on feedback from UNH IOL
1.20	July 30, 2008	Updated section for Open MPI and added tables Updated IB SM Failover as per Nick Wood
1.21	Aug 1, 2008	Updated SRP call srp_daemon -o -e -n Updated IB SM Failover as Bob Jaworski Updated HP-MPI Updated Intel MPI Updated Open MPI
1.22	Aug 29, 2008	Added a section for MVAPICH 1 under OSU MPI
1.23	Feb 16, 2009	Updated link init, fabric init, srp, sdp, ipoib cm, ipoib dm based on updates received from UNH-IOL
1.24	Feb 23, 2009	Updated Intel MPI and Open MPI to reflect the fact that they are not intended to work in a heterogeneous environment. Updated the RDS test procedure Updated the Test Glossary Updated the Mandatory test table for April 2009

Revision	Release Date	
1.25	Feb 24, 2009	Updated the RDS Test after review by the OFA IWG group.
1.26	Mar 13, 2009	Restructured entire document to accommodate WinOF and OFED Added NFS over RDMA to the test plan. Added WinOF tests Updated HP-MPI Add List of Contributors
1.27	Mar 17, 2009	Updates based on the review from the OFA IWG
1.28	Mar 27, 2009	Added links in Chapter 10 to the InfiniBand Test Scripts Added links to HP-MPI installation Packages

List of Contributors

Name	Company
Mark Alan	HP
Rupert Dance	Lamprey Networks
Sujal Das	Mellanox
Arlin Davis	Intel
Johann George	QLogic
Mike Hagen	UNH-IOL
Allen Hubbe	UNH-IOL
Bob Jaworski	QLogic
Arkady Kanevsky	NetApp
Llolsten Kaonga	Lamprey Networks
Amit Krig	Mellanox
Jon Mason	Open Grid Computing
Bob Noseworthy	UNH-IOL
Yaroslav Pekelis	Mellanox
Hal Rosenstock	Obsidian
Martin Schlining	DataDirect Networks
Karun Sharma	QLogic
Stan Smith	Intel
Dave Sommers	Intel (NetEffect)
Jeff Squyres	Cisco
Dennis Tolstenko	Lamprey Networks
Steve Wise	Open Grid Computing
Robert Woodruff	Intel
Nick Wood	UNH-IOL

Editor: Rupert Dance

Legal Disclaimer

"This version of a proposed OpenFabrics Interop Test Plan is provided "AS IS" and without any warranty of any kind, including, without limitation, any express or implied warranty of non-infringement, merchant-ability or fitness for a particular purpose.

In no event shall OpenFabrics, IBTA or any member of these groups be liable for any direct, indirect, special, exemplary, punitive, or consequential damages, in-cluding, without limitation, lost profits, even if ad-vised of the possibility of such damages."

Conditional text tag *Explanation* is shown in green.

~~Conditional text tag *Deleted* is shown in red with strike through.~~

Conditional text tag *Proposal* is shown in turquoise (r0_g128_b128).

Conditional text tag *Author* is shown as is.

Conditional text tag Comment is shown in red with underline

1 Introduction

Server OEM customers have expressed the need for RDMA hardware and software to interoperate.

Specifically, InfiniBand HCA, OpenFabrics host software to interoperate with InfiniBand Switches, gateways, and bridges with management software provided by OEMs, and IB integrated server OEM vendors. And, iWARP RNIC and OpenFabrics host software to interoperate with Ethernet Switches and management software and hardware provided by Ethernet Switch OEMs and iWARP integrated server OEM vendors.

It is necessary that the interoperability test effort be an industry-wide effort where interoperability testing is conducted under the auspices of the appropriate networking organizations. For InfiniBand it is IBTA, specifically within the charter of the CIWG. And for iWARP it is IETF, and specifically within UNH IOL iWARP Consortium.

1.1 Purpose

This document is intended to describe the production tests step by step explaining each test and its references. The purpose of this test plan is three fold:

1) Define the scope, equipment and software needs, and test procedures for verifying full interoperability of RDMA HW and SW. For Infiniband HW it is InfiniBand HCAs using the latest OpenFabrics IB OFED software with currently available OEM Switches and their management software. The target OEM IB Switch vendors are Flextronics, Mellanox, Obsidian, QLogic and Voltaire. For iWARP HW it is iWARP RNICs using the latest OpenFabrics OFED software with currently available OEM Ethernet Switches, Bridges, Gateways, Edge Devices and so on with their management software.
2) Serve as a basis for evaluating customer acceptance criteria for OFA host software interoperability and OFA Logo.
3) Serve as a basis for extensions to InfiniBand IBTA CIWG test procedures related to interoperability and use of these test procedures in upcoming PlugFest events organized by IBTA.

   Serve as a basis for extensions to iWARP test procedures for OpenFabrics software related to interoperability and use of these test procedures in upcoming PlugFest events organized by UNH IOL iWARP Consortium.

1.2 Intended Audience

The following are the intended audience for this document:

1) Project managers in OEM Switch, Router, Gateway, Bridge Vendor companies to understand the scope of testing and participate in the extension of this test plan and procedures as necessary to meet their requirements.
2) IBTA and CIWG, and iWARP and UNH IOL iWARP testing personnel and companies to evaluate the scope of testing and participate in the extension of this test plan and procedures as necessary to meet their requirements.

3) Test engineering and project leads and managers who will conduct the testing based on this document.
4) Customers and users of OFA host software who rely on OFA Logo for interoperability.
5) Integrators and OEM of RDMA products.

1.3 Test Plan Structure

This test plan is divided into two main sections.

1) Interoperability testing using OFED for Linux.
   a) See Section 10
2) Interoperability testing using WinOF for Windows Platforms.
   a) See Section 11

1.4 INFINIBAND ONLY - TEST OVERVIEW

The tables below list all of the specific test procedures for InfiniBand Devices. See the Transport Independent section for tests that apply to all transports.

Table 1 - IB Link Initialize

Test #	*Test*	*Description Overview*
1	Phy link up all ports	Check that all relevant green LEDs are on for all HCAs and switches.
2	Logical link up all ports switch SM	All vendors should check that the link state is up and the port width and link speed is as advertised by the vendor.
3	Logical link up all ports HCA SM	All vendors should check that the link state is up and the port width and link speed is as advertised by the vendor.

Table 2 - IB Fabric Initialization

Test #	*Test*	*Description Overview*
1	Fabric Initialization	Run SM from each node in cluster and see that all ports are in Armed or Active state.

Table 3 - IB IPoIB - Connect Mode (CM)

Test #	*Test*	*Description Overview*
1	Ping all to all	Run SM from one of the nodes and check all nodes responding. Repeat with all SMs.
2	Connect disconnect host	Run SM from one of the nodes and check all nodes responding.
3	FTP Procedure	Using a 4MB test file, put the file, then get the file and finally compare the file.

Table 4 - IB IPoIB - Datagram Mode (DM)

Test #	*Test*	*Description Overview*
1	Ping all to all	Run SM from one of the nodes and check all nodes responding. Repeat with all SMs.
2	Connect disconnect host	Run SM from one of the nodes and check all nodes responding.
3	FTP Procedure	Using a 4MB test file, put the file, then get the file and finally compare the file.

Table 5 IB SM Tests

Test #	*Test*	*Description Overview*
1	Basic sweep test	verify that all SMs are NOT ACTIVE (after receiving the SMSet of SMInfo to DISABLE) and that the selected SM (SM1) is the master (
2	SM Priority test	Verify Subnet and SMs behavior according to the SMs priority.
3	Failover - Disable SM1	Disable the master SM and verify that standby SM becomes master and configures the cluster.
4	Failover - Disable SM2	Disable the master SM and verify that standby SM becomes master and configures the cluster.

Table 6 - IB SRP Tests

Test #	*Test*	*Description Overview*
1	Basic dd application	Run basic dd application from SRP host connected to target.
2	IB SM kill	Kill the IB master SM while test is running and check that it completes properly.
3	Disconnect Host	Unload SRP Host and check SRP connection properly disconnected.
4	Disconnect Target	Unload SRP Target and check SRP connection properly disconnected.

Table 7 - IB Ethernet Gateway

Test #	*Test*	*Description Overview*
1	Basic Setup	Connect the HCA of the IB host and Ethernet Gateway to the IB fabric. Connect the Ethernet gateway to the Ethernet network or Ethernet device. Start the SM to be used in this test.
2	Start ULP	Determine which ULP your ethernet gateway uses and be sure that ULP is running on the host.
3	Discover Gateway	Restart the ULP or using the tool provided by the ULP, make sure that the host “discovers” the Ethernet Gateway.
4	SM Failover	While the ping is running, kill the master SM. Verify that the ping data transfer is unaffected.
5	Ethernet gateway reboot	Reboot the Ethernet Gateway. After the Ethernet Gateway comes up, verify that the host can discover the Ethernet Gateway as it did before and we are able to configure the interfaces.
6	ULP restart	Restart the ULP used by Ethernet Gateway and verify that after the ULP comes up, the host can discover the Ethernet Gateway and we are able to configure the interfaces.
7	Unload/load ULP	Unload the ULP used by Ethernet Gateway and check that the Ethernet Gateway shows it disconnected. Load the ULP and verify that the Ethernet gateway shows the connection.

Table 8 - IB Fibre Channel Gateway

Test #	*Test*	*Description Overview*
1	Basic Setup	Connect the HCA of the IB host to the IB fabric. Connect the FC Gateway to the IB Fabric. Connect the FC Gateway to the FC network or FC device. Start the SM to be used in this test.
2	Configure Gateway	Configure the FC Gateway appropriately (how to do this is vendor specific).
3	Add Storage Device	Use ibsrpdm tool in order to have the host "see" the FC storage device. Add the storage device as target.
4	Basic dd application	Run basic dd application from SRP host connected to target.
5	IB SM kill	Kill the IB master SM while test is running and check that it completes properly.
6	Disconnect Host/Target	Unload the SRP host / SRP Target (target first/host first) and check that the SRP connection is properly disconnected.
7	Load Host/Target	Load the SRP host / SRP Target. Using ibsrpdm, add the target.
8	dd after SRP Host and Target reloaded	Run basic dd application from the SRP host to the FC storage device.
9	Reboot Gateway	Reboot the FC Gateway. After FC Gateway comes up, verify using ibsrpdm tool that the host see the FC storage device. Add the storage device as target.
10	dd after FC Gateway reboot	Verify basic dd works after rebooting Gateway.

1.5 Ethernet Only - Test Overview

The tables below list all of the specific test procedures for iWARP and Ethernet Devices. See the Transport Independent section for tests that apply to all transports.

Table 9 - Ethernet Link Initialize

Test #	*Test*	*Description Overview*
1	Phy link up all ports	Check that all relevant green LEDs are on for all RNICs and switches.
2	Verify basic IP connectivity	Verify that basic IP connectivity can occur by driving minimum size ICMP echo requests and replies across the link or equivalent traffic.

Table 10 - Ethernet Fabric Initialize

Test #	*Test*	*Description Overview*
1	Fabric Initialization	Verify IP connectivity to all IP attached stations in the Cluster. Source 1000 minimum size ICMP echo requests from all RNICs to all other IP entities to verify cluster connectivity.

Table 11 - Ethernet Fabric Reconvergence

Test #	*Test*	*Description Overview*
1	Fabric Reconvergence	Run SM from each node in cluster and see that all ports are in Armed or Active state.

Table 12 - Ethernet Fabric Failover

Test #	*Test*	*Description Overview*
1	Fabric Failover	Kill root RSTP switch of the primary VLAN, ensure there is a fully redundant path through the fabric and verify recovery occurs

Table 13 iWARP Connections

Test #	*Test*	*Description Overview*
1	UNH iWARP interop tests group 1	Verify that each single iWARP operation over single connection works.
2	UNH iWARP interop tests group 2	Verify that multiple iWARP operations over a single connection work.
3	UNH iWARP interop tests group 3	Verify that multiple iWARP connections work.
4	UNH iWARP interop tests group 4	Verify that disconnect/reconnect physical connections work.
5	UNH iWARP interop tests group 5	Verify that IP Speed negotiation work.
6	UNH iWARP interop tests group 6	Verify that iWARP error ratio work.
7	UNH iWARP interop tests group 7	Verify that stress pattern over iWARP work.
8	UNH iWARP interop tests group 8	Verify that iWARP parameter negotiation work.

1.6 TRANSPORT INDEPENDENT - TEST OVERVIEW

The tables below list the test procedures that apply to devices regardless of the transport.

Table 14 - TI iSER Tests

Test #	*Test*	*Description Overview*
1	Basic dd application	Run basic dd application from iSER host connected to target.
2	IB SM kill	[IB Specific] - Kill the IB master SM while test is running and check that it completes properly.
3	Disconnect Initiator	Unload iSER Host and check iSER connection properly disconnected.
4	Disconnect Target	Unload iSER Target and check iSER connection properly disconnected.
5	Repeat with previous SM Slave	[IB Specific Test] Repeat steps 1-4 now with the previous slave SM (we did not actually stop the target).

Table 15 - TI NFS Over RDMA

Test #	*Test*	*Description Overview*
1	File and directory creation	A total of six files and six directories are created
2	File and directory removal	removes the directory tree that was just created by test1
3	Lookups across mount point	changes directory to the test directory and gets the file status of the working directory
4	Setattr, getattr, and lookup	Permissions are changed (chmod) and the file status is retrieved (stat) for each file
5	Read and write	Creates a file (creat), Gets status of file (fstat) , Checks size of file, Writes 1048576 bytes into the file (write) in 8192 byte buffers, Closes file (close), Gets status of file (stat) , Checks the size of the file
6	Readdir	The program creates 200 files (creat). The current directory is opened (opendir), the beginning is found (rewinddir), and the directory is read (readdir) in a loop until the end is found
7	Link and rename	This program creates ten files. For each of these files, the file is renamed (rename) and file statistics are retrieved (stat) for both the new and old names
8	Symlink and readlink	This program makes 10 symlinks (symlink). It reads (readlink), and gets statistics for (lstat) each, and then removes them (unlink).
9	Statfs	This program changes directory to the test directory (chdir and/or mkdir) and gets the file system status on the current directory (statfs).

Table 16 TI RDS

Test #	*Test*	*Description Overview*
1	rds-ping procedure	Run rds-ping and verify that you can reach all hosts in the cluster

Table 16 TI RDS

Test #	*Test*	*Description Overview*
2	rds-stress procedure	Set up passive receiving instance and an active sender and verify data is exchanged without error

Table 17 - TI SDP Tests

Test #	*Test*	*Description Overview*
1	netperf procedure	Run netperf where message size is 10, 100, 1000, 10000 and local buffer size is 1024, 6000.
2	FTP procedure	Using a 4MB test file, put the file, then get the file and finally compare the file.
3	IB SCP Procedure	Connect via SCP on IPoIB address from all other nodes uploading and downloading a file.
4	iWARP SCP Procedure	Connect via SCP from all other nodes uploading and downloading a file.

Table 18 - TI uDAPL

Test #	*Test*	*Description Overview*
1	Point-to-Point Topology	Connection and simple send receive.
2	Point-to-Point Topology	Verification, polling and scatter gather list.
3	Switched Topology	Verification and private data.
4	Switched Topology	Add multiple endpoints, polling, and scatter gather list.
5	Switched Topology	Add RDMA Write.
6	Switched Topology	Add RDMA Read.
7	Multiple Switches	Multiple threads, RDMA Read, and RDMA Write.
8	Multiple Switches	Pipeline test with RDMA Write and scatter gather list.
9	Multiple Switches	Pipeline with RDMA Read.
10	Multiple Switches	Multiple switches.

Table 19 - Basic RDMA Interop

Test #	*Test*	*Description Overview*
1	Small RDMA READ	Create an RDMA command sequence to send a READ operation of one byte.
2	Large RDMA READ	Create an RDMA command sequence to send a READ operation of 10,000,000 bytes
3	Small RDMA Write	Create an RDMA command sequence to send a Write operation of one byte

Table 19 - Basic RDMA Interop

Test #	*Test*	*Description Overview*
4	Large RDMA Write	Create an RDMA command sequence to send a Write operation of 10,000,000 bytes
5	Small RDMA SEND	Create an RDMA command sequence to send a SEND operation of one byte.
6	Large RDMA SEND	Create an RDMA command sequence to send a SEND operation of one million bytes
7	Small RDMA Verify	Create an RDMA command sequence to send a VERIFY operation of one byte.
8	Large RDMA Verify	Create an RDMA command sequence to send a VERIFY operation of 10,000,000 bytes

Table 20 - RDMA operations over Interconnect Components

Test #	*Test*	*Description Overview*
1	Switch Load	For one pair of endpoints generate a stream of RDMA READ operation in one direction and RDMA write operations in the opposite direction. For all remaining endpoint pairs configure an RDMA WRITE operation of 1 byte and have it sent 10000 times on both streams of the endpoint pair.
2	Switch Fan In	Connect all possible endpoint pairs such that data exchanges between pairs must traverse the pair of ports interconnecting the switch

1.7 HP-MPI - Test Overview

Table 21 - TI - HP-MPI Test Suite Description - (Not part of OFA Stack)

Test #	*HP-MPI TESTs*	*HP-MPI TESTs Suite Description*
1	IMB	This is the Intel MPI Benchmark. If this passes, then the basic interoperability of HP-MPI with the installed OFED is confirmed.
2	rings2	This is a proprietary HP test which has a good history of stressing interconnects to the point of failure. It also includes 1sided operations.
3	fork	New RDMA implementations often have fork issues, As new OS kernels come out the fork problems sometimes re-appear. This test makes a point of stressing that code path.
4	exitpath	The purpose of this test is simply to make sure machines and OFED drivers etc remain stable when applications repeatedly terminate abnormally.
5	alltoone	This test has all the ranks send a flood of messages to rank 0, to make sure the interconnect can handle heavy load in that message pattern.

1.8 Intel MPI - Test Overview

Table 22a - Intel MPI Benchmark Summary

Test #	*Test*	*Description Overview*
1	Test 1: PingPong	
2	Test 1: PingPing	
3	Test 1: Sendrecv	
4	Test 1: Exchange	
5	Test 1: Allreduce	
6	Test 1: Reduce	
7	Test 1: Reduce_scatter	
8	Test 1: Allgather	
9	Test 1: Allgatherv	
10	Test 1: Alltoall	
11	Test 1: Alltoallv	
12	Test 1: Bcast	
13	Test 1: Barrier	

Table 22b - TI - Intel MPICH2 Suite - (Not part of OFA Stack)

Test #	*MPICH2 (16 sections, 290 tests)*	*Intel - MPICH2 Test Suite Section Description*
1	attr	Test programs for attribute routines
2	coll	Test programs for various collective operations
3	comm	Test programs for communicator operations
4	datatype	Test programs for various datatype operations
5	errhan	Test programs for error handling operations
6	group	Test programs for the group operations
7	info	Test programs for various info operations
8	init	Test programs for init operations
9	pt2pt	Test programs for various point to point routines (send, isend, probe, etc.)
10	rma	Test programs for memory access operations
11	spawn	Test programs for comm_spawn, intercom operations

Table 22b - TI - Intel MPICH2 Suite - (Not part of OFA Stack)

Test #	*MPICH2 (16 sections, 290 tests)*	*Intel - MPICH2 Test Suite Section Description*
12	topo	Test programs for various topology routines
13	io	Test programs for file i/o read/write, sync and async
14	F77	Test programs for f77
15	cxx	Test programs for c++
16	threads	Test programs for threaded send/recv

Table 22c - TI - Intel MPI Test Suite Description - (Not part of OFA Stack)

Test #	*IntelMPITEST (5 sections, 1371 tests)*	*IntelMPITest Suite Description*
1	testlist2l (1085 tests)	c - blocking, coll, datatype, env, group, misc, non-blocking
2	testlist2-2l (23 tests)	c, fortran – datatype create
3	testlist4 (216 tests)	fortran – grp, topo, blocking, coll, datatype, non-blocking, persist, probe, send/recv
4	testlist4lg (1 test)	c - collective overlap
5	testlist6 (46 tests)	c, fortran – topo cart/graph

1.9 Open MPI - Test Overview

Table 23 - TI - Open MPI Test Suite Description

Test #	*Open MPI TESTs*	*Open MPI TESTs Suite Description*
Phase 1: "Short" tests		
1	2	OMPI built with OpenFabrics support
2	3	OMPI basic functionality (hostname)
3	4.1	Simple MPI functionality (hello_c)
4	4.2	Simple MPI functionality (ring_c)
5	5	Point-to-point benchmark (NetPIPE)
6	6.1.1	Point-to-point benchmark (IMB PingPong multi)
7	6.1.2	Point-to-point benchmark (IMB PingPing multi)
Phase 2: "Long" tests		
8	6.2.1	Point-to-point benchmark (IMB PingPong)
9	6.2.2	Point-to-point benchmark (IMB PingPing)
10	6.2.3	Point-to-point benchmark (IMB Sendrecv)
11	6.2.4	Point-to-point benchmark (IMB Exchange)
12	6.2.5	Collective benchmark (IMB Bcast)
13	6.2.6	Collective benchmark (IMB Allgather)
14	6.2.7	Collective benchmark (IMB Allgatherv)
15	6.2.8	Collective benchmark (IMB Alltoall)
16	6.2.9	Collective benchmark (IMB Reduce)
17	6.2.10	Collective benchmark (IMB Reduce_scatter)
18	6.2.11	Collective benchmark (IMB Allreduce)
19	6.2.12	Collective benchmark (IMB Barrier)
20	6.3.1	I/O benchmark (IMB S_Write_Indv)
21	6.3.2	I/O benchmark (IMB S_IWrite_Indv)
22	6.3.3	I/O benchmark (IMB S_Write_Expl)
23	6.3.4	I/O benchmark (IMB S_IWrite_Expl)
24	6.3.5	I/O benchmark (IMB P_Write_Indv)
25	6.3.6	I/O benchmark (IMB P_IWrite_Indv)
26	6.3.7	I/O benchmark (IMB P_Write_Shared)

Table 23 - TI - Open MPI Test Suite Description

Test #	*Open MPI TESTs*	*Open MPI TESTs Suite Description*
27	6.3.8	I/O benchmark (IMB P_IWrite_Shared)
28	6.3.9	I/O benchmark (IMB P_Write_Priv)
29	6.3.10	I/O benchmark (IMB P_IWrite_Priv)
30	6.3.11	I/O benchmark (IMB P_Write_Expl)
31	6.3.12	I/O benchmark (IMB P_IWrite_Expl)
32	6.3.13	I/O benchmark (IMB C_Write_Indv)
33	6.3.14	I/O benchmark (IMB C_IWrite_Indv)
34	6.3.15	I/O benchmark (IMB C_Write_Shared)
35	6.3.16	I/O benchmark (IMB C_IWrite_Shared)
36	6.3.17	I/O benchmark (IMB C_Write_Expl)
37	6.3.18	I/O benchmark (IMB C_IWrite_Expl)
38	6.3.19	I/O benchmark (IMB S_Read_Indv)
39	6.3.20	I/O benchmark (IMB S_IRead_Indv)
40	6.3.21	I/O benchmark (IMB S_Read_Expl)
41	6.3.22	I/O benchmark (IMB S_IRead_Expl)
42	6.3.23	I/O benchmark (IMB P_Read_Indv)
43	6.3.24	I/O benchmark (IMB P_IRead_Indv)
44	6.3.25	I/O benchmark (IMB P_Read_Shared)
45	6.3.26	I/O benchmark (IMB P_IRead_Shared)
46	6.3.27	I/O benchmark (IMB P_Read_Priv)
47	6.3.28	I/O benchmark (IMB P_IRead_Priv)
48	6.3.29	I/O benchmark (IMB P_Read_Expl)
49	6.3.30	I/O benchmark (IMB P_IRead_Expl)
50	6.3.31	I/O benchmark (IMB C_Read_Indv)
51	6.3.32	I/O benchmark (IMB C_IRead_Indv)
52	6.3.33	I/O benchmark (IMB C_Read_Shared)
53	6.3.34	I/O benchmark (IMB C_IRead_Shared)
54	6.3.35	I/O benchmark (IMB C_Read_Expl)
55	6.3.36	I/O benchmark (IMB C_IRead_Expl)
56	6.3.37	I/O benchmark (IMB Open_Close)

1.10 OSU MPI - Test Overview

Table 24 - TI - OSU MPI

Test #	*Test*	*Description Overview*
1	**Test** 1: PingPong	
2	**Test** 1: PingPing point-to-point	
3	**Test** 2: PingPong	
4	**Test** 2: PingPing	
5	**Test** 2: Sendrecv	
6	**Test** 2: Exchange	
7	**Test** 2: Bcast	
8	**Test** 2: Allgather	
9	**Test** 2: Allgatherv	
10	**Test** 2: Alltoall	
11	**Test** 2: Alltoallv	
12	**Test** 2: Reduce	
13	**Test** 2: Reduce_scatter	
14	**Test** 2: Allreduce	
15	**Test** 2: Barrier	

1.11 Requirements for OFA Interoperability Logo Program

The following table indicates the mandatory tests to qualify for the OFA Interoperability Logo Program during the April 2009 Interoperability Event. It is anticipated that some of the Beta tests will be moved to Mandatory status for the following Interop Event.

Table 25 - InfiniBand Transport Test Status for April 2009 Interop Event

Test Procedure	Linux	WinOF
IB Link Initialize	Mandatory	Beta
IB Fabric Initialization	Mandatory	Beta
IB IPoIB Connected Mode	Mandatory	Not Available -1
IB IPoIB Datagram Mode	Mandatory	Beta
IB SM Failover/Handover	Beta	Beta
IB SRP	Mandatory	Beta
IB Ethernet Gateway	Beta	Not Available - 3
IB Fibre Channel Gateway	Beta	Not Available - 3
TI iSER	Mandatory	Beta
TI NFS over RDDMA	Beta	Not Available - 1
TI RDS	Beta	Not Available - 2
TI SDP	Mandatory	Not Available - 1
TI uDAPL	Beta	Beta
TI Basic RDMA Interop	Beta	Not Available - 3
TI RDMA Operations	Beta	Not Available - 3
TI MPI HP	Beta	Not Available - 2
TI MPI Intel	Beta	Beta
TI MPI Open MPI	Beta	Not Available - 2
TI MPI OSU	Beta	Not Available - 2

Not Available means one of three things:

1) The feature is not currenlty supported by the WinOF stack
2) The ULP appliaction has not been ported to the WinOF Stack
3) The test has not updated for WinOF

Table 26 - Ethernet Transport Test Status for April 2009 Interop Event

Test Procedure	Linux
Ethernet Link Initialize	**Beta**
Ethernet Fabric Initialization	**Beta**
Ethernet Fabric Failover	**Beta**
Ethernet Fabric Reconvergence	**Beta**
iWARP Connectivity	**Mandatory**
TI iSER	**Beta**
TI NFS over DRDMA	**Beta**
TI RDS	**Beta**
TI SDP	**Beta**
TI uDAPL	**Beta**
TI Basic RDMA Interop	**Beta**
TI RDMA Operations	**Beta**
TI MPI HP	**Beta**
TI MPI Intel	**Beta**
TI MPI Open MPI	**Beta**
TI MPI OSU	**Beta**

1.12 SUBJECTS NOT COVERED

Table 27 - SUBJECTS NOT COVERED

Number	*Subject/ Feature*	*Reason*	*Executor*	*Due Date*
1	iWARP peer to peer	Future Testing		September 2009
2	IPv6 testing	Future Testing		September 2009

1.13 TEST GLOSSARY

Table 28 Test Glossary

Technical Terms	
HCA	IB Host Channel Adapter.
IPoIB	IP over InfiniBand
iSER	iSCSI Extension for RDMA
MPI	Message Passing Interface
RDF	Readme File.
RDS	Reliable Datagram Socket
RNIC	RDMA NIC (iWARP Network Interface Card).
SA	IB Subnet Administration.
SM	IB Subnet Manager.
SDP	Sockets Direct Protocol
SRP	SCSI RDMA Protocol
TD	Test Descriptions.
TI	Transport Independent (tests).
uDAPL	User Direct Access Programming Library

2 General System Setup

Configuration

The test environment for the user interface contains:

2.1 IB HW Units

Table 29 IB Equipment

Equipment	*Amount*	*Details*	*Check*
Operating System	12 or more	The OS should be supported by OpenFabrics.	
4X IB Cables	30 or more	Between 1M => 10M.	
IB Switch from a 3rd Party Vendor	6	The number and types of switches needed from OEM is dependent on variations in embedded and subnet management and other IBTA defined management software. For example if the software on Switch A is different from the software used in Switch B, both Switches will be needed. Note that it is not dependent on number of ports supported by a switch.	
InfiniBand 4X Analyzer	1		
IB HCAs	12 or more		

2.2 IB Software

2.2.1 Linux/Windows platforms

2.2.2 OFED/WinOF - Most Current Tested Release

2.2.3 IB HCA FW – Version XXX - Vendor Specific

2.2.4 IB Switch FW candidate – Version XXX - Vendor Specific

2.2.5 IB Switch SW – Version XXX - Vendor Specific

2.3 iWARP HW Units

2.4 iWARP Software

2.4.1 Linux/Windows platforms

2.4.2 OFED - Most Current Tested Release

2.4.3 iWARP RNIC FW – Version XXX - Vendor Specific

2.4.4 10GbE Switch FW candidate – Version XXX - Vendor Specific

2.4.5 10GbE Switch SW – Version XXX - Vendor Specific

3 Use of OpenFabrics Software for Pre-Testing

Depending on the schedule of testing and bugs or issues encountered, different snapshots of latest OpenFabrics software will be used during pre-testing prior to the Interoperability Event. Any changes that result in the OpenFabrics software from interoperability testing per this test plan will be deposited back into the OpenFabrics repository so that the OpenFabrics development community will have full access to any bug fixes or feature additions that may result out of this testing effort. The frequency of such deposits will be determined based on completion of adequate testing of the said fixes or feature additions.

4 Use of OpenFabrics Software for IBTA/CIWG Interoperability Plugfest

During the pre-testing phase, UNH-IOL will apply all reasonable effort to ensure that the OpenFabrics source and binary repositories are up-to-date with the results of interoperability testing prior to IBTA/CIWG sponsored interoperability plugfest events. This will enable interoperability testing at plugfests to be conducted using software directly sourced from the OpenFabrics tree.

Should there be any issues with the OpenFabrics community not accepting certain bug fixes or features with the time frames matching with plugfest occurrences, UNH-IOL will inform all participants about the same and offer those bug fixes or features in source code and binary formats directly to the plug fest participants and InfiniBand solution suppliers.

5 Use of OpenFabrics Software for UNH IOL iWARP Interoperability PlugFests

During the pre-testing phase, UNH IOL will apply all reasonable effort to ensure that the OpenFabrics source and binary repositories are up-to-date with the results of interoperability testing prior to UNH IOL iWARP sponsored interoperability plug fest events. This will enable interoperability testing at plug fests to be conducted using software directly sourced from the OpenFabrics tree.

Should there be any issues with the OpenFabrics community not accepting certain bug fixes or features with the time frames matching with plug fest occurrences, UNH IOL will inform all participants about the same and offer those bug fixes or features in source code and binary formats directly to the plug fest participants and iWARP solution suppliers.

6 IB HW description & connectivity

The Test contains 2 major parts - this description is for each of those parts.

6.1 Basic connectivity (P1P1)

6.1.1 HCA 1 should be connected from port 1 to lowest port number in switch

6.1.2 HCA 2 should be connected from port 1 to highest port number in switch

6.1.3 Both with compliant InfiniBand cables

6.2 Switches and Software Needed

6.2.1 Switches provided by OEMs

It is necessary that Switches provided by OEMs cover the full breadth of software versions supported by the Switch OEMs. Port count is not critical for the tests. It is recommended that OEMs provide six switches covering all variations of software supported on the Switches.

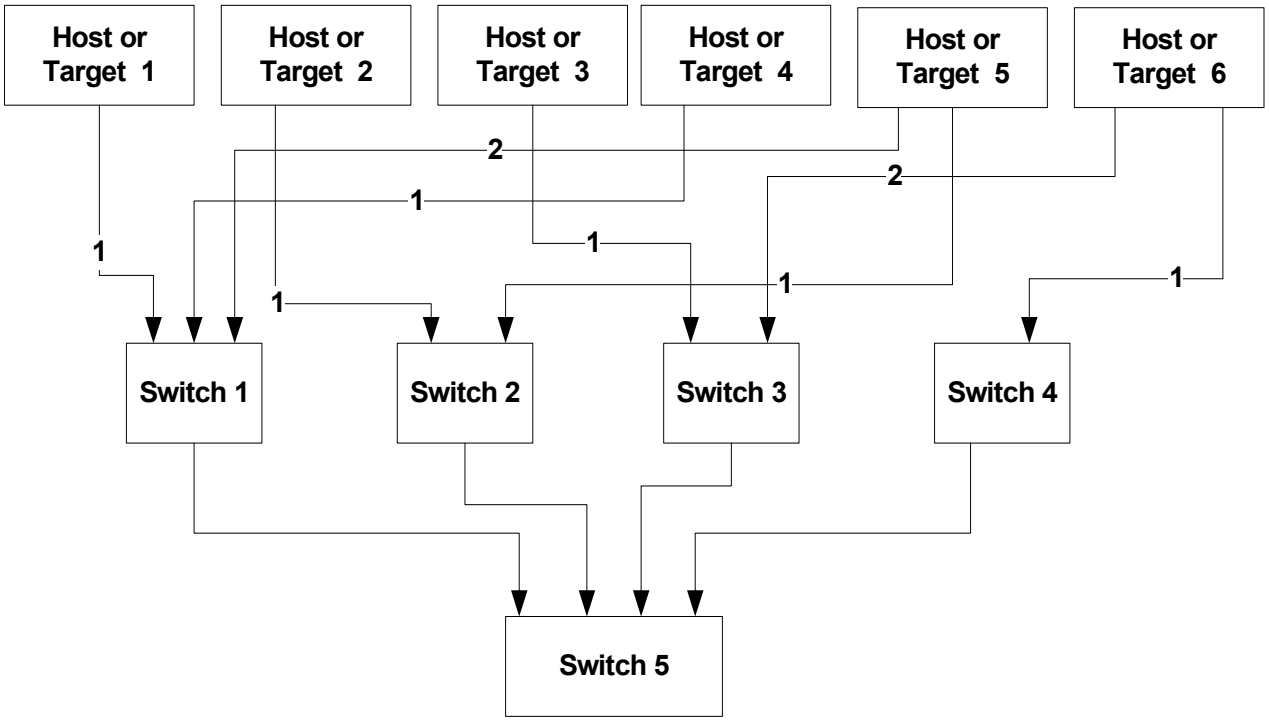
6.2.2 OpenFabrics software running on Hosts

Where there are dependencies of OEM provided and IBTA defined management software (such as subnet managers and agents, performance managers and agents etc.) with OpenFabrics software running on Hosts, such software should be provided to UNH-IOL for interoperability testing. Any known dependencies should be communicated to UNH-IOL.

6.3 Cluster Connectivity

6.3.1 Hosts and Targets 1-6 should be connected from port 1 or 2 to ports X in all switches using compliant InfiniBand cables.

Figure 1 - Template for IB Interop Setup



7 iWARP HW description & connectivity

The Test contains 2 major parts - this description is for each of those parts.

7.1 iWARP Basic connectivity (P1P1)

7.1.1 RNIC 1 on one host should be directly connected to RNIC 2 on another host or to a 10GbE switch.

7.1.2 With 10GbE cables

7.2 Switches and Software Needed

7.2.1 Switches provided by OEMs

It is necessary that Switches provided by OEMs cover the full breadth of software versions supported by the Switch OEMs. Port count is not critical for the tests. It is recommended that OEMs provide a switch per variations of software supported on the Switch.

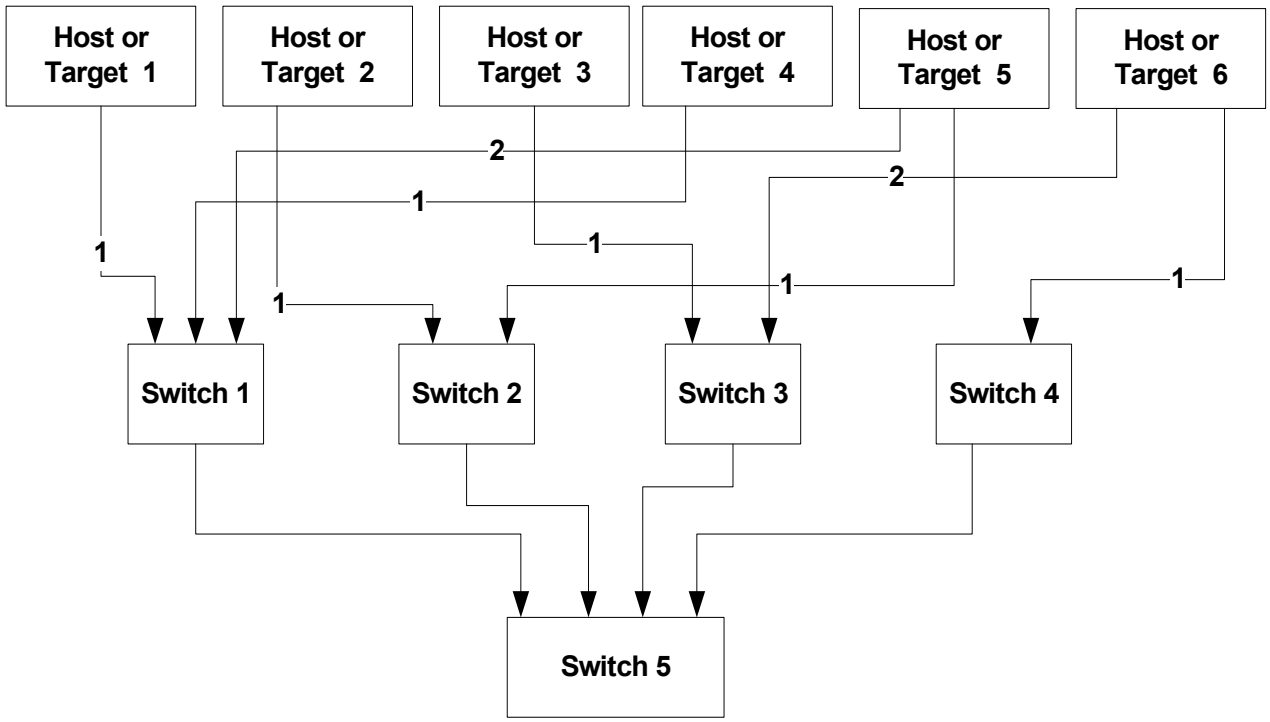
7.2.2 OpenFabrics software running on RNICs

Where there are dependencies of OEM provided with OpenFabrics software running on RNICs, such software should be provided to UNH-IOL for interoperability testing, and any known dependencies should be communicated to UNH-IOL.

7.3 Cluster Connectivity

7.3.1 Hosts and Targets 1-6 should be connected to switches using 10GbE cables.

Figure 2 Template for iWARP Interop Setup



7.4 Gateway, Bridges, Routers Connectivity

TBD

8 SW & HW installation

8.1 Burning the FW

8.1.1 Please refer to firmware burning tools and procedures documentation from HCA IB vendor

8.1.2 No firmware burning required for iWARP vendor

8.2 SW installation

8.2.1 Please refer to software installation manual from HCA IB vendor.

8.2.2 Please refer to software installation manual from RNIC vendor.

9 General Instructions

9.1 First step Instructions

1) Burn the FW release XXX on all HCAs and RNICs using the above procedure as required by vendor.
2) Host and Target Configuration
   a) Install OFED software on host systems (using a 64 bit OS) configured to run OFED.
   b) Install WinOF software on host systems (using a 64 bit OS) configured to run WinOF.
   c) Configure non-OFED systems for use in the cluster as per the vendors instructions.
   d) Configure iSER/SRP targets for use in the cluster as per the vendors instructions.
3) Install the switch or gateway with the candidate SW stack as required by vendor.
4) Burn the switch or gateway with the released FW as required by vendor.
5) Connect the Hosts and Targets to an appropriate switch following the basic connectivity.

10 InfiniBand Specific Interop Procedures using OFED

Note: UNH-IOL has created automated scripts to run many of the OFED based . tests. See http://www.iol.unh.edu/downloads/OFA/scripts.rar

10.1 IB Link Initialize using OFED

10.1.1 Disconnect the full topology and select a cable whose length should be a maximum of 15 meters for SDR and 10 meters for DDR when using copper cables.

1) Verify that no SM is running
2) Connect two devices back to back
3) ssh to one of the two devices
   a) Run "ibdiagnet -lw 4x" to verify portwidth
   b) Run "ibdiagnet -ls 2.5" to check link speed. Interpret output and compare to advertised speed.

   Note: This command will only produce output if the link speed is anything other than SDR. Keep this in mind during your interpretation of the output.
4) Repeat steps 1-3 with a different device pairing.
   a) All device pairs must be tested except target to target: HCA to HCA, HCA to Switch, HCA to Target, Switch to Switch, and Switch to Target.
   b) Each device must link to all other devices in order for the device to pass link init over all.

10.1.2 Recommendations

In order to determine Switch to Target and Switch to Switch link parameters, run commands from an HCA linked to the switch under test. This does require more interpretation of the output to differentiate the reported parameters.

10.2 IB Fabric Initialization using OFED

10.2.1 Architect the Network we want to build.

1) Create a table of IP addresses to assign.
2) Create topology file - this makes sure that the subnet is configured as expected - i.e. SDR and DDR links. This inserts name of devices as well as the GUID.
3) See Figure 3- Sample Network Configuration below.

10.2.2 Connect the HCAs and switches as per the Architected Network and make sure that no SM/SA is running on the Fabric.

10.2.3 Procedure

1) Start an SM on a device and let it initialize (all SM's will need to be tested)
2) Visually verify that all devices are in the active state. Orange led will be on if the port is active.
3) Run "ibdiagnet -pc" to clear all port counters
4) Wait 17 seconds as per the specifications requirements.
5) Run "ibdiagnet -c 1000" to send 1000 node descriptions.
6) Run "ibdiagnet -r" to generate fabric report.
7) Run "ibchecknet" to build guid list.

10.2.4 Verification Procedures

1) Review "PM Counters" section of the fabric report. There should be no illegal PM counters. The Specification says there should be no errors in 17 seconds.
2) Review "Subnet Manager " section of the fabric report. Verify that the running SM is the one you started and verify number of nodes and switches in the fabric.
3) Review the ibchecknet report and verify that there are no duplicate GUIDs in the fabric

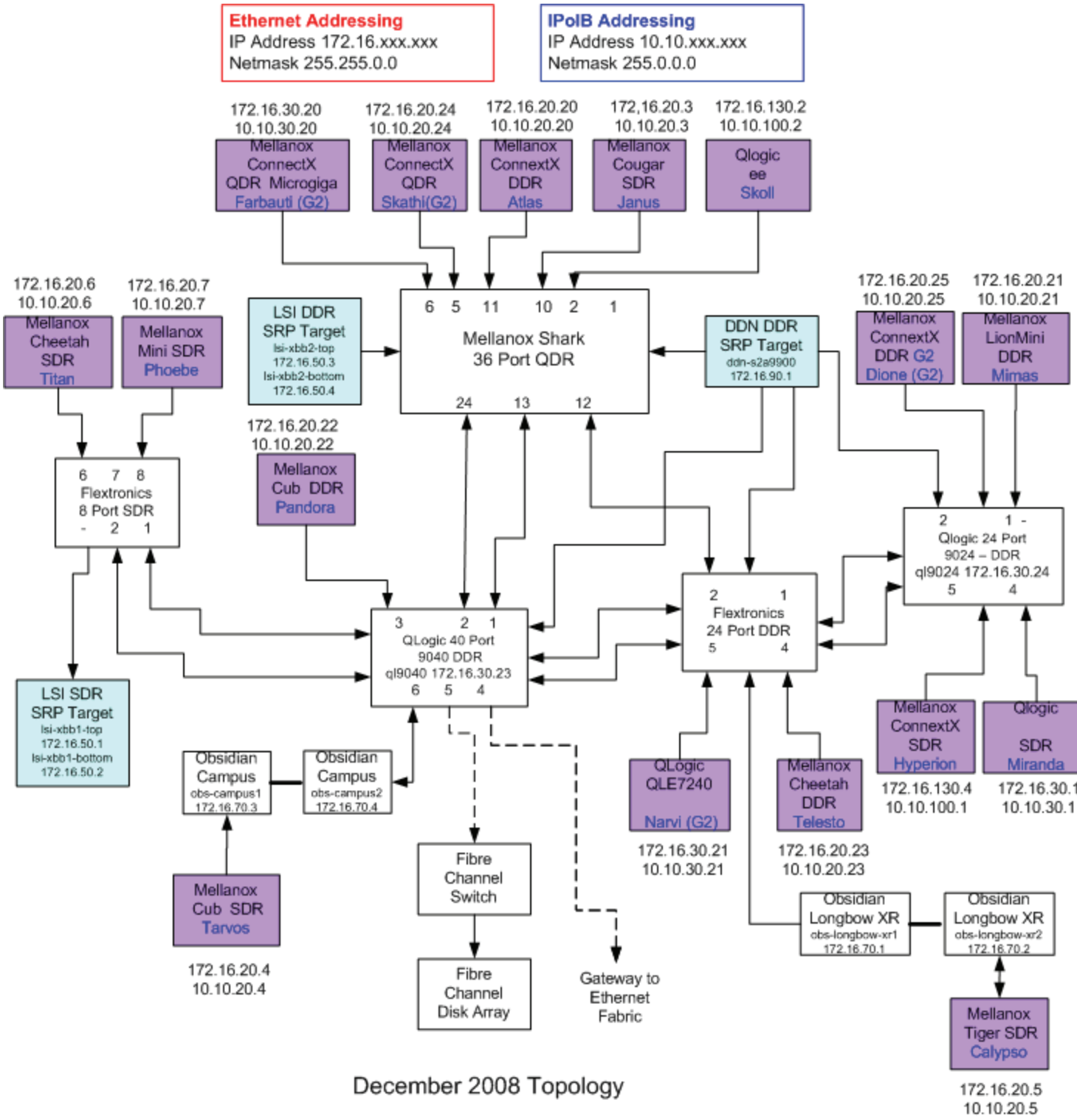
Restart all devices in the fabric and follow Sections 10.2.3 and 10.2.4. Run the SM from a different device in the fabric until all SMs present have been used. All SMs on managed switches and one instance of **opensm** must be used.

Each device must pass all verification procedures with every SM to pass Fabric Initialization test.

Table 30 - ibdiagnet commands

Commands	Description
ibdiagnet -h	Help
Ibdiagnet - pc	Clear Counter
Ibdiagnet -lw 4x - ls 2.5	Specify link width and speed
Ibdiagnet -c 1000	send 1000 Node Descriptions

Figure 3 - Sample Network Configuration



10.3 IB IPoIB Connect Mode (CM) using OFED

10.3.1 Setup

Connect the HCAs and switches as per the Architected Network and make sure that no SM is running on the Fabric.

This procedure, as the previous ones, will be based on the cluster connectivity. An SM/SA which supports IPoIB (sufficient IB multicast support) will be running on the HCAs, or on a switch with an embedded SM/SA or a third HCA which would only run SM/SA for the partner pair (with a switch in the middle). This procedure has been developed for Linux and may be ported to Windows if there is sufficient vendor support.

Optional: In the procedures below, an IB analyzer can be inserted in the appropriate link to obtain traces and validate the aspects of the procedures specifically detailed below in subsequent sections.

10.3.2 IPoIB Interface Creation and IPoIB Subnet Creation

1) Configure IPoIB address. All addresses must reside on the same subnet.
   a) Set interfaces to 10.0.0.x/24 (10.0.0.x/netmask 255.255.255.0) using the command *ifconfig ib0 10.0.0.x netmask 255.255.255.0*

10.3.3 .Bringing the IPoIB in Connected Mode

1) Set "SET_IPOIB_CM=yes" in file /etc/infiniband/openib.conf
2) Restart driver "/etc/init.d/openibd restart"
3) Validate CM mode by checking that "/sys/class/net/<I/F name>/mode" equal to '**connected**'
4) Repeat steps 1-3 in section 10.3.3 on all nodes being tested.

10.3.4 Ping Procedures

Step A

1) Stop all SM's and verify that none are running
2) Power cycle all switches in the fabric (this insures that the new SM will configure all the links and create the multi-cast join).
3) Start an SM (All SM's will need to be tested) and let it initialize
   a) Visually verify that all devices are in the active state. Orange led will be on if the port is active.
   b) Run "ibdiagnet -r" and verify that the SM you started is the one that is running and and that it is the master. You will need to know the GUID of the device since the SM will be reassigned on each reboot.
   c) Verify that all nodes and switches were discovered.

      Note: Ibdiagnet may show more switches than indicated by the physical number of switch platforms present. This is because some switches have multiple switch chips.
4) Examine the arp table (via arp -a) and remove the destination node's ib0 address from the sending node's arp table (via arp -d).

5) Ping every HCA except localhost with packet sizes of 64, 256, 511, 512, 1024, 1025, 2044, 4096, 8192, 16384, 32768, and 65507.
   a) 100 packets of each size will be sent
   b) Repeat step #4 before issuing each ping command. Every packet size is a new ping command.
6) In order to pass Step A, a reply must be received for every ping sent (without losing a single packet) while using each one of the SMs available in the cluster.

Step B

1) Bring up all HCAs but one.
2) Start an SM (all SMs will need to be tested).
3) Check for ping response between all node (All to All).
   a) A response from the disconnected HCA should not be returned.
4) Disconnect one more HCA from the cluster.
5) Ping to the newly disconnected HCA from all nodes (No response should be returned).
6) Connect the first machine (the one that was not connected) and check for ping response from all nodes that are still connected.
7) Connect the disconnected HCA to a different switch on the subnet which will change the topology.
8) Ping again from all nodes (this time we should get a response).
9) Follow Step B, this time bring the interface down and then back up using ifconfig ibX down and ifconfig ibX up commands instead of physically disconnecting the HCAs.

   Note: Each step must exhibit the expected behavior while using each SM in order for the device to pass Step B overall.

Step C

Follow Step A and B using a different SM until all SM's have been used. Only one instance of each available SM is required. Steps A, B, and C must pass in order for the device to pass 10.3.4 overall.

10.3.5 SFTP PROCEDURE

SFTP procedures require an SFTP server to be configured on each machine in the partner pair. An SFTP client needs to be available on each machine as well. The default RHEL install includes both.

A 4 MB file will be SFTP'd to the partner and then SFTP'd back and binary compared to the original file, this will be done in each direction and then bidirectional using every SM available.

10.3.5.1 Setup

1) Make sure vsftpd is installed on each node for SFTP application.
2) A special account for this should be created as follows:
   b) Username: Interop
   c) Password: openfabrics

10.3.5.2 PROCEDURE

1) Run SFTP server on all nodes.
2) Start an SM (all SM's will need to be tested) and let it initialize
   a) Verify that the running SM is the one you started.
3) SFTP:
   a) Connect an HCA pair via SFTP on IPoIB using the specified user name and password.
   b) Put the 4MB file to the /tmp dir on the remote host.
   c) Get the same file to your local dir again.
   d) Compare the file using the command *cmp tfile tfile.orig.*
      i) The two must be identical
4) Repeat the procedure with a different SM.

Note: Every node must SFTP the 4MB file to all others using all SM's and the files must be identical as determined by the binary compare in order for the device to pass 10.3.5 overall.

Note: Sections 10.3.4 and 10.3.5 must pass using the configuration determined by sections 10.3.1, 10.3.2, and 10.3.3 for the device to pass IPoIB Connected mode overall.

10.4 IB IPoIB Datagram Mode (DM) using OFED

10.4.1 Setup

Connect the HCAs and switches as per the Architected Network and make sure that no SM is running on the Fabric.

This procedure, as the previous ones, will be based on the cluster connectivity. An SM/SA which supports IPoIB (sufficient IB multicast support) will be running on the HCAs, or on a switch with an embedded SM/SA or a third HCA which would only run SM/SA for the partner pair (with a switch in the middle). This procedure has been developed for Linux and may be ported to Windows if there is sufficient vendor support.

Optional: In the procedures below, an IB analyzer can be inserted in the appropriate link to obtain traces and validate the aspects of the procedures specifically detailed below in subsequent sections.

10.4.2 IPoIB Interface Creation and IPoIB Subnet Creation

1) Configure IPoIB address. All addresses must reside on the same subnet.
   a) Set interfaces to 10.0.0.x/24 (10.0.0.x/netmask 255.255.255.0) using the command *ifconfig ib0 10.0.0.x netmask 255.255.255.0*

10.4.3 .Bringing the IPoIB in Datagram Mode

1) Set "SET_IPOIB_CM=no" in file /etc/infiniband/openib.conf
2) Restart driver "/etc/init.d/openibd restart"
3) Validate DM mode by checking that "/sys/class/net/<I/F name>/mode" equal to '**datagram**'
4) Repeat steps 1-3 in section 10.4.3 on all nodes being tested.

10.4.4 Ping Procedures

Step A

1) Stop all SM's and verify that none are running
2) Power cycle all switches in the fabric (this insures that the new SM will configure all the links and create the multi-cast join).
3) Start an SM (All SM's will need to be tested) and let it initialize
   a) Visually verify that all devices are in the active state. Orange led will be on if the port is active.
   b) Run "ibdiagnet -r" and verify that the SM you started is the one that is running and and that it is the master. You will need to know the GUID of the device since the SM will be reassigned on each reboot.
   c) Verify that all nodes and switches were discovered.

      Note: Ibdiagnet may show more switches than indicated by the physical number of switch platforms present. This is because some switches have multiple switch chips.
4) Examine the arp table (via arp -a) and remove the destination node's ib0 address from the sending node's arp table (via arp -d).

5) Ping every HCA except localhost with packet sizes of 64, 256, 511, 512, 1024, 1025, 2044, 4096, 8192, 16384, 32768, and 65507.
   a) 100 packets of each size will be sent
   b) Repeat step #4 before issuing each ping command. Every packet size is a new ping command.
6) In order to pass Step A, a reply must be received for every ping sent (without losing a single packet) while using each one of the SMs available in the cluster.

Step B

1) Bring up all HCAs but one.
2) Start an SM (all SMs will need to be tested).
3) Check for ping response between all node (All to All).
   a) A response from the disconnected HCA should not be returned.
4) Disconnect one more HCA from the cluster.
5) Ping to the newly disconnected HCA from all nodes (No response should be returned).
6) Connect the first machine (the one that was not connected) and check for ping response from all nodes that are still connected.
7) Connect the disconnected HCA to a different switch on the subnet which will change the topology.
8) Ping again from all nodes (this time we should get a response).
9) Follow Step B, this time bring the interface down and then back up using ifconfig ibX down and ifconfig ibX up commands instead of physically disconnecting the HCAs.

   Note: Each step must exhibit the expected behavior while using each SM in order for the device to pass Step B overall.

Step C

Follow Step A and B using a different SM until all SM's have been used. Only one instance of each available SM is required. Steps A, B, and C must pass in order for the device to pass 10.4.4 overall.

10.4.5 SFTP PROCEDURE

SFTP procedures require an SFTP server to be configured on each machine in the partner pair. An SFTP client needs to be available on each machine as well. The default RHEL install includes both.

A 4 MB file will be SFTP'd to the partner and then SFTP'd back and binary compared to the original file, this will be done in each direction and then bidirectional using every SM available.

10.4.5.1 SETUP

1) Make sure vsftpd is installed on each node for SFTP application.
2) A special account for this should be created as follows:
   b) Username: Interop
   c) Password: openfabrics

10.4.5.2 PROCEDURE

Run SFTP server on all nodes.

1) Start an SM (all SM's will need to be tested) and let it initialize
   a) Verify that the running SM is the one you started.
2) SFTP:
   a) Connect an HCA pair via SFTP on IPoIB using the specified user name and password.
   b) Put the 4MB file to the /tmp dir on the remote host.
   c) Get the same file to your local dir again.
   d) Compare the file using the command *cmp tfile tfile.orig.*
      i) The two must be identical
3) Repeat the procedure with a different SM.

   Note: Every node must SFTP the 4MB file to all others using all SM's and the files must be identical as determined by the binary compare in order for the device to pass 10.4.5 overall.

   Note: Sections 10.4.4 and 10.4.5 must pass using the configuration determined by sections 10.4.1, 10.4.2, and 10.4.3 for the device to pass IPoIB Datagram mode overall.

10.5 IB SM Failover and Handover Procedure using OFED

10.5.1 Setup

1) Connect HCAs per the selected topology.
2) In this test, all active SMs on the fabric which are going to be tested, must be from the same vendor. They will be tested pairwise; two at a time.

10.5.2 Procedure

1) Disable all SMs in the cluster then start a SM on either machine in a chosen pair.
2) Run "saquery" on a node in the fabric.
   a) Verify that all nodes in the cluster are present in the output
3) Using the ibdiagnet tool with the -r option, verify that the running SM is the master.
4) Start a SM on the second machine in the current pair.
5) Verify that the SMs behave according to the SM priority rules. Use "ibdiagnet -r" again.
   a) SM with highest numerical priority value is master and the other is in standby.
   a) If both SMs have the same priority value then the SM with the smallest guid is master and the other is in standby.
6) Run "saquery" on either machine in the current pair.
   a) Verify that all nodes in the cluster are present in the output.
7) Shutdown the master SM.
8) Verify the other active SM goes into the master state using "ibdiagnet -r" again.
9) Run "saquery" on either machine in the current pair.
   a) Verify that all nodes in the cluster are present in the output.
10) Start the SM you just shutdown.
11) Verify that the newly started SM resumes it's position as master while the other goes into standby again.
12) Run "saquery" on either machine in the current pair.
   a) Verify that all nodes in the cluster are present in the output.
13) Shutdown the standby SM.
14) Verify that the previous master SM is still the master.
15) Run "saquery" on either machine in the current pair.
   a) Verify that all nodes in the cluster are present in the output.
16) Repeat steps 1-15 above 2 more times, ensuring that the below criteria is met (total of 3 tests per pair which can be run in any order):
   a) First SM to be started having highest numerical priority value.
   b) Second SM to be started having highest numerical priority value.

c) Both SMs having equal numerical priority values.

17) Repeat steps 1-16 until all possible SM pairs from identical vendors in the cluster have been tested.

18) All of the "saquery" commands must return the expected list of nodes in order for the SMs in this test to receive a passing grade.

10.6 IB SRP using OFED

10.6.1 Setup

Connect the HCAs and switches as per the Architected Network and make sure that no SM is running on the Fabric.

10.6.2 Procedure

1) Start an SM (all SM's will need to be tested) and let it initialize
   a) Verify that the running SM is the one that you started
2) Choose a node to work with
3) Unload the srp module
4) Load srp module with srp_sg_tablesize=255
   a) **Example**: modprobe ib_srp srp_sg_tablesize=255
   b) Let it initialize
5) Verify that the module loaded correctly
   a) **Example**: lsmod | grep ib_srp
6) Load srp_daemon with -e -o -n options
   a) **Example**: srp_daemon -e -o -n
   b) Let it initialize
7) Find all volumes from all targets
   a) Use lsscsi
8) Perform 6GB read from srp volume to null
   a) **Example**: dd if=$drive of=/dev/null count=600 bs=10M
9) Perform 6GB write from zero to srp volume
   a) **Example**: dd if=/dev/zero of=$drive count=600 bs=10M
10) Repeat step #8 and #9 for all volumes found for all targets as determined by step #7
11) Unload srp module
12) Repeat steps 2 through 9 for all HCAs
13) Reboot all devices in the fabric and repeat the procedure using a different SM.

    Note: An HCA must successfully complete all DD operations to and from all volumes on all targets using all available SM's in order to pass SRP testing. One volume per target is all that is required.

10.7 IB Ethernet Gateway using OFED

10.7.1 Procedure

1) Connect the HCA of the IB host to the IB fabric. Connect the Ethernet Gateway to the IB fabric. Connect the Ethernet gateway to the Ethernet network or Ethernet device. Start the SM to be used in this test.
2) Determine which ULP your ethernet gateway uses and be sure that ULP is running on the host (VNIC or IPoIB).
3) Restart the ULP or using the tool provided by the ULP, make sure that the host “discovers” the Ethernet Gateway. Configure the interfaces and make sure they are up.
4) Run ping from the host to the Ethernet device. While the ping is running, kill the master SM. Verify that the ping data transfer is unaffected.
5) Reboot the Ethernet Gateway. After the Ethernet Gateway comes up, verify that the host can discover the Ethernet Gateway as it did before and we are able to configure the interfaces.
6) Restart the ULP used by Ethernet Gateway and verify that after the ULP comes up, the host can discover the Ethernet Gateway and we are able to configure the interfaces.
7) Unload the ULP used by Ethernet Gateway and check that the Ethernet Gateway shows it disconnected. Load the ULP and verify that the Ethernet gateway shows the connection.
8) Repeat step 4 by using ssh and scp instead of ping.

10.8 IB FIBRECHANNEL GATEWAY USING OFED

10.8.1 Procedure

1) Connect the HCA of the IB host to the IB fabric. Connect the FC Gateway to the IB Fabric (how to do this is determined by the FC Gateway vendor). Connect the FC Gateway to the FC network or FC device. Start the SM to be used in this test.
2) Configure the FC Gateway appropriately (how to do this is vendor specific).
3) Use ibsrpdm tool in order to have the host "see" the FC storage device. Add the storage device as target.
4) Run basic dd application from the SRP host to the FC storage device.
5) Run basic dd application from the SRP host to the FC storage device. While the test is running, kill the master SM. Verify that the test completes properly.
6) Unload the SRP host / SRP Target (target first/host first) and check that the SRP connection is properly disconnected.
7) Load the SRP host / SRP Target. Using ibsrpdm, add the target.
8) Run basic dd application from the SRP host to the FC storage device.
9) Reboot the FC Gateway. After FC Gateway comes up, verify using ibsrpdm tool that the host see the FC storage device. Add the storage device as target.
10) Run basic dd application from the SRP host to the FC storage device.
11) Follow steps 1-10 above with each SM to be tested and with each HCA to be tested, until each HCA and each SM has been tested with the FC Gateway.

11 Ethernet Specific Interop Procedures using OFED

11.1 Ethernet Link Initialize using OFED

11.1.1 Purpose

The Ethernet Link Initialize test is a validation that all Ethernet devices receiving the OFA Logo can link and pass traffic under nominal (unstressed) conditions.

11.1.2 Resource Requirements

1) Gigabit or 10Gigabit Ethernet RNIC,
2) Gigabit or 10Gigabit Ethernet Switch
3) Compliant Cables

11.1.3 Discussion

The validation of the underlying transport infrastructure is essential to the end-users experience of the operation of the OFED software stack. To this end, this test confirms that Ethernet devices receiving the OFA Logo can suitably link and pass traffic in any configuration. Exhaustive compliance testing of BER performance of the channel or electrical signaling of the ports is not performed; however, successful completion of this test provides further evidence of the robustness of the OFA logo bearing device.

11.1.4 Procedure

1) Connect the two link partners together utilizing compliant compliant cables.
2) Check all relevant LEDs on both ends of the link.
3) Verify that basic IP connectivity can occur by driving minimum size ICMP echo requests and replies across the link or equivalent traffic (including RDMA traffic if readily configured, in which case an additional RNIC responder station is required).
4) Repeat steps 1-3 for all combinations of 2 RNICs to switches, switch to switch, and RNIC to RNIC link partner combinations. Previously tested combinations resident in the OFILG cluster may be omitted.

11.1.5 Observable results

1) Link should be established on both ends of the channel.
2) Traffic should pass in both directions. Error rates of 10e-5 or better should be readily confirmed (no lost frames in 10,000).

11.1.6 Possible Problems

1) Traffic directed to a switches IP management address may not be processed at high speed, in such cases, traffic should be passed across the switch to a remote responder.

11.2 Ethernet Fabric Initialize using OFED

11.2.1 Purpose

The Ethernet Fabric Initialization test is a validation that all Ethernet devices receiving the OFA Logo properly interoperate with common OSI Layer 2 protocols including Link Aggregation, RSTP, and MSTP under nominal (unstressed) conditions.

11.2.2 Resource Requirements

1) Gigabit or 10Gigabit Ethernet RNIC,
2) Gigabit or 10Gigabit Ethernet Switch
3) Compliant Cables

11.2.3 Discussion

The validation of the underlying transport infrastructure is essential to the end-users experience of the operation of the OFED software stack. To this end, this test confirms that Ethernet devices receiving the OFA Logo can suitably form link aggregates and establish redundant inter-switch links managed by RSTP and/or MSTP in the selected plugfest or cluster Network Architecture configuration. Neither exhaustive interoperability configuration permutations nor IEEE 802.1 compliance testing is performed as part of this test; however, successful completion of this test provides further evidence of the robustness of the OFA logo bearing device.

Note: IP Connectivity is desired to ensure connectivity is stable and that underlying fabric issues (such as link flapping) are not masked by TCP transport of RDMA traffic. RDMA traffic is desired to observe the effects of topology changes on the iWARP protocol.

11.2.4 Procedure

1) Architect the desired network from available cluster and plugfest participants, similar to that shown in the Cluster Connectivity Section 7.3. All cabling must be compliant cables. Most RNIC-to-RNIC paths should traverse 2 or more switches.
   a) Create a table of IP addresses to assign to RNICs and switch management entities.
   b) When MSTP is supported, create a VLAN topology with at least 2 VLANs (high and normal priority) Create 802.1q VLAN trunk links between supporting switches.
   c) When Link Aggregation is supported by both link partners, create a 2-4 channel link aggregate between the link partners.

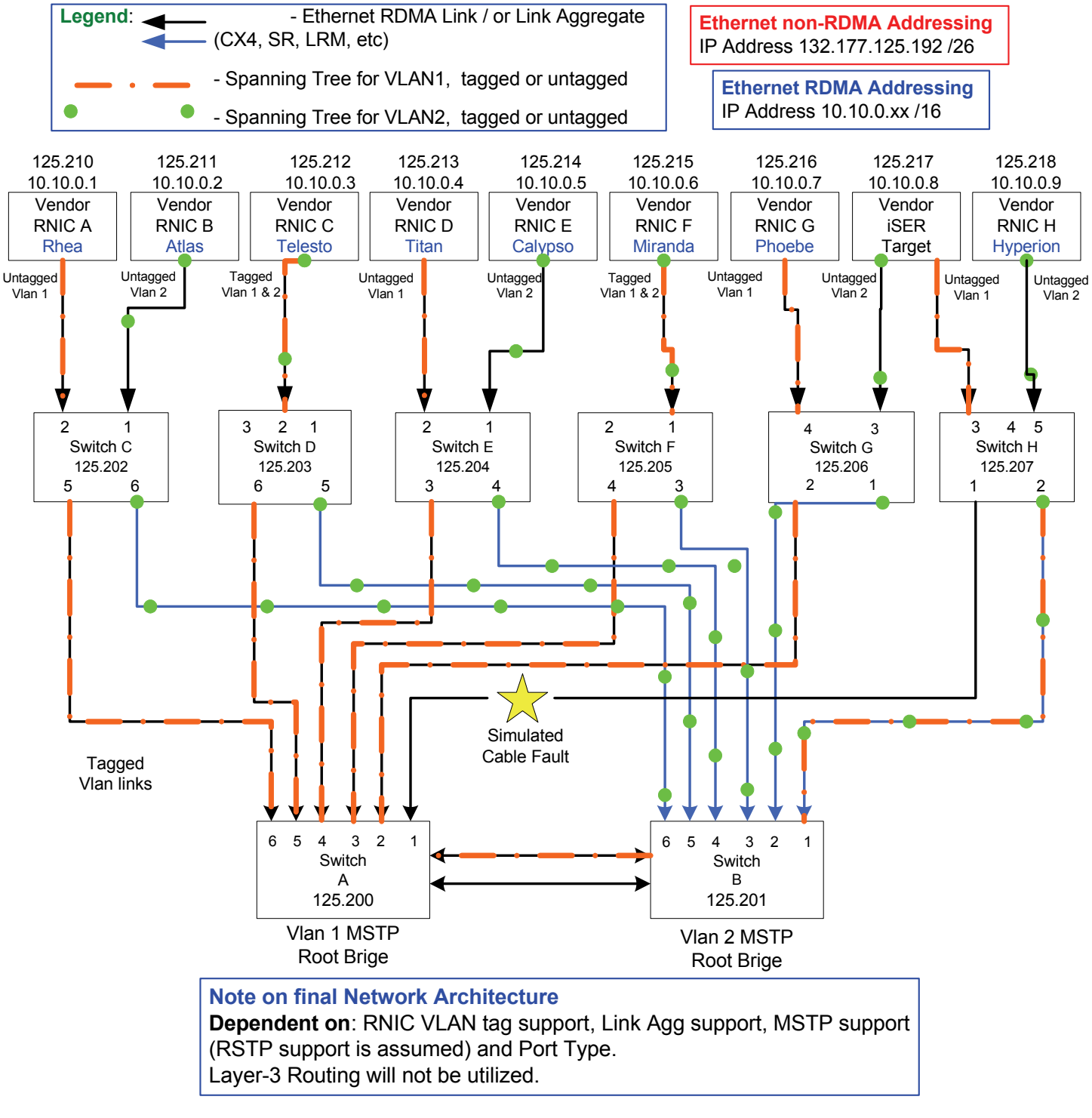
      Note: This includes RNICs supporting Link Aggregation, as well as switch to switch links.
   d) Set spanning tree priorities such that desired bridge(s) becomes root bridge(s).
   e) See Cluster Connectivity.

2) Connect the RNICs and switches as per the Architected Network and make sure that desired bridge is the root bridge (in the case of MSTP, the appropriate bridge per VLAN) is running on the Fabric.
3) Verify IP connectivity to all IP attached stations in the Cluster. Source 1000 minimum size ICMP echo requests from all RNICs to all other IP entities to verify cluster connectivity.
4) Verify RDMA connectivity to all RDMA attached stations in the Cluster. Source 100 2k RDMA reads from each RNIC to all other RNICs to verify cluster RDMA connectivity.

11.2.5 Observable results

1) In all cases, the desired root bridge (or in the case of MSTP topologies, the desired root bridges) should always become the root bridge.
2) IP connectivity should occur to all stations without loss of responses.
3) RDMA connectivity should occur to all stations without loss of responses.

Figure 4 - Sample Ethernet Network Configuration



11.3 Ethernet Fabric Reconvergence using OFED

11.3.1 Purpose

The Ethernet Fabric Reconvergence test is a validation that all Ethernet devices receiving the OFA Logo properly converge in the event of a topology change in a timely manner, minimally impacting the fabric.

11.3.2 Resource Requirements

1) Gigabit or 10Gigabit Ethernet RNICs
2) Gigabit or 10Gigabit Ethernet Switches
3) CX4 Cables

11.3.3 Discussion

The validation of the underlying transport infrastructure is essential to the end-users experience of the operation of the OFED software stack. To this end, this test confirms that Ethernet devices receiving the OFA Logo can suitably reconverge in the event of a topology change effecting a link aggregate, and/or an RSTP or MSTP per the selected plugfest or cluster Network Architecture configuration. Neither exhaustive topology change permutations nor IEEE 802.1 compliance testing is performed as part of this test; however, successful completion of this test provides further evidence of the robustness of the OFA logo bearing device.

Note: IP Connectivity is desired to ensure connectivity is stable and that underlying fabric issues (such as link flapping) are not masked by TCP transport of RDMA traffic. RDMA traffic is desired to observe the effects of topology changes on the iWARP protocol.

11.3.4 Procedure

1) Power off all switches. Connectivity remains that of the selected Network Architect configuration.
2) Disconnect one switch, leaving its attached RNICs isolated from the rest of the fabric.
3) Power up all switches. Verify IP connectivity of connected nodes.
4) Reconnect disconnected switch to original location. Verify IP and RDMA connectivity is restored to all nodes.
5) Remove a redundant switch-to-switch interconnect. Verify IP and RDMA connectivity is maintained to all nodes.
6) Create a new redundant switch-to-switch interconnection (potentially forming a loop in the absence of RSTP/MSTP). Verify IP and RDMA connectivity is maintained to all nodes.
7) Remove a channel from a link aggregate. Verify IP and RDMA connectivity is maintained to all nodes.
8) Restore the previously removed channel to the link aggregate. Verify IP and RDMA connectivity is maintained to all nodes.
9) Restart all devices in the fabric and follow Steps 1-8 each time with a different switch in the fabric as the desired root bridge. Repeat, as time allows,

until each switch has been the root of a spanning tree, each switch has been isolated from the fabric at least once, each switch has seen at least one topology change (removal or addition of a link), and all link aggregates have seen a removal and restoration of a link.

Note: In the presence of no hardware/ firmware/ software changes, previously tested combinations resident in the OFILG cluster may be omitted.

11.3.5 Observable results

1) In all cases, the desired root bridge (or in the case of MSTP topologies, the desired root bridges) should always become the root bridge.
2) IP and RDMA connectivity should be restored to all stations rapidly. Editors note: this could be further clarified (some topology changes should not impact traffic, some will), RSTP likely would converge in well under 2sec and thus 2s could form an 'extreme' upper-bound for reconvergence times in the cases of traffic interruption.

11.3.6 Possible Problems

Time limitations of the plugfest may prevent full evaluation of all switches and RNICs. In this case, 'switch to switch' links and 'switch to RNIC' links will be selected in random order to provide as much coverage as time allows.

11.4 Ethernet Fabric Failover using OFED

11.4.1 Purpose

The Ethernet Fabric Failover test is a validation that the Ethernet switch fabric devices receiving the OFA Logo properly recovers in the event of the loss of the root switch and a new topology converges in a timely manner, minimally impacting the fabric.

11.4.2 Resource Requirements

1) Gigabit or 10Gigabit Ethernet RNICs
2) Gigabit or 10Gigabit Ethernet Switches
3) Compliant Cables

11.4.3 Discussion

The validation of the underlying transport infrastructure is essential to the end-users experience of the operation of the OFED software stack. To this end, this test confirms that Ethernet devices receiving the OFA Logo can suitably reconverge in the event of a failure of the root bridge of an RSTP or MSTP topology per the selected plugfest or cluster Network Architecture configuration. A root bridge for a spanning tree topology has full awareness of switch to switch interconnects in its given VLAN domain. Loss of the current root bridge requires rediscovery and re-election of a new root bridge, possibly further delaying network re-convergence.

It is assumed that the Network Architecture will be selected such that the core switches will allow multiple redundant paths between switches, such that the loss of any one switch will not interrupt the flow of cluster traffic. Additionally, it is assumed that the core switches will have no direct connection to any RNIC, and that these core switches will be selected to serve as the root of the spanning trees. It is presumed that every switch under test can serve as one of the cluster's core switches.

Neither exhaustive topology change permutations nor IEEE 802.1 compliance testing is performed as part of this test; however, successful completion of this test provides further evidence of the robustness of the OFA logo bearing device.

Note: IP Connectivity is desired to ensure connectivity is stable and that underlying fabric issues (such as link flapping) are not masked by TCP transport of RDMA traffic. RDMA traffic is desired to observe the effects of topology changes on the iWARP protocol.

11.4.4 Procedure

1) Power off all switches. Connectivity remains that of the selected Network Architect configuration.

   Note: Selected Architecture should allow redundant paths to all RNICs in the event of the loss of the desired root switch.

2) Power up all switches. Verify IP and RDMA connectivity of connected nodes.

3) In a single RSTP environment, remove power from the current root switch. In an MSTP environment, remove power from only one switch serving as the root of a selected VLAN. Verify IP connectivity is eventually restored to all nodes.
4) Restart all devices in the fabric and follow Sections 10.A.1 through 10.A.3, each time with a different switch in the fabric as the desired root bridge. NOTE: This may require the location of the switch-under-test to be moved within the selected Network Architecture to ensure redundant paths exist to all RNICs. Repeat as time allows until each switch has been powered off while serving as the root of a spanning tree. Note: In the presence of no hardware/ firmware/ software changes, previously tested combinations resident in the OFILG cluster may be omitted.

11.4.5 Observable Results

1) In all cases, the desired root bridge (or in the case of MSTP topologies, the desired root bridges) should always become the root bridge.
2) IP and RDMA connectivity should be restored to all stations rapidly. Editors note: the term 'rapidly' could be further clarified, however the loss of the root switch will significantly increase convergence times.

11.4.6 Possible Problems

Available switch ports may restrict a given switch's ability to serve in any location within the Network Architecture. In such cases, if a redundant path to a set of RNICs is not possible in the event the root switch is lost, then reconverged connectivity to the effected RNICs will naturally not be required.

Additionally, time limitations of the plugfest may prevent full evaluation of all switches. In this case, each switch will be selected in random order to provide as much coverage as time allows.

11.5 iWARP Connectivity using OFED

11.5.1 UNH-IOL Interop Suite

See UNH-IOL iWARP Interoperability Test Suite for full details

11.5.2 iWARP Setup

The interoperability tests can be run in point to point mode or switched. Connect 2 iWARP hosts RNICs together or to a 10GbE switch.

11.5.3 Test Procedure

Step A:

Group 1: Single RDMA Operations Over A Single Connection:

- TEST 1.1: RDMA WRITE
- TEST 1.2: RDMA READ
- TEST 1.3: RDMA SEND
- TEST 1.4: RDMA SENDINV
- TEST 1.5: RDMA SENDSE
- TEST 1.6: RDMA SENDSEINV
- TEST 1.7: RDMA TERMINATE
- TEST 1.8: LARGE RDMA WRITE
- TEST 1.9: LARGE RDMA READ

Step B

Group 2: Multiple RDMA Operations Over A Single Connection:

- Test 2.1: Sequence of 10 RDMA Write Commands
- Test 2.2: Sequence of 10 RDMA Read Commands
- Test 2.3: Sequence of 10 RDMA Send Commands
- Test 2.4: Sequence of 10 RDMA Sendinv Commands
- Test 2.5: Sequence of 10 RDMA Sendse Commands
- Test 2.6: Sequence of 10 RDMA Sendseinv Commands
- Test 2.7: Sequence of 10 RDMA Terminate Commands
- Test 2.8: Sequence of Interleaved RDMA Write And Read Commands
- Test 2.9: Sequence of Interleaved RDMA Write And Terminate Commands
- Test 2.10: Sequence of Interleaved RDMA Read And Terminate Commands
- Test 2.11: Sequence of Interleaved RDMA Send And Terminate Commands

- Test 2.12: Sequence of Interleaved RDMA Sendinv And Terminate Commands
- Test 2.13: Sequence of Interleaved RDMA Sendse And Terminate Commands
- Test 2.14: Sequence of Interleaved RDMA Sendseinv And Terminate Commands
- Test 2.15: Sequence of Interleaved RDMA Write With All Other RDMA Commands
- Test 2.16: Sequence of Interleaved RDMA Read With All Other RDMA Commands
- Test 2.17: Sequence of Interleaved RDMA Send With All Other RDMA Commands
- Test 2.18: Sequence of Interleaved RDMA Sendinv With All Other RDMA Commands
- Test 2.19: Sequence of Interleaved RDMA Sendse With All Other RDMA Commands
- Test 2.20: Sequence of Interleaved RDMA Sendseinv With All Other RDMA Commands

Step C

Group 3: Multiple Connections:

- Test 3.1: Single RDMA Operations Over Multiple Connections
- Test 3.2: Multiple RDMA Operations Over Multiple Connections
- Test 3.3: RDMA Operations Over 25 Connections
- Test 3.4: Simultaneous Operations Over 25 Connections

Step D

Group 4: Disconnect/Reconnect Physical Connections:

- Test 4.1: Termination Followed By A WRITE
- Test 4.2: Termination Followed By A READ

Step E

Group 5: Speed Negotiation:

- Test 5.1: RNICs Operating At 10g And 1g Speed

Step F

Group 6: RDMA Error Ratio:

- Test 6.1: Sequence of All Zeros
- Test 6.2: Sequence of All Ones
- Test 6.3: Sequence of Ones Followed By Zeros

- Test 6.4: Sequence of Interleaved Ones And Zeros

Step G

Group 7: Stress Patterns Over RDMA:

- Test 7.1: RDMA Read After Prolonged RDMA Write Operations
- Test 7.2: RDMA Read After Prolonged RDMA Read Operations
- Test 7.3: RDMA Read After Prolonged RDMA Send Operations
- Test 7.4: RDMA Read After Prolonged RDMA Sendinv Operations
- Test 7.5: RDMA Read After Prolonged RDMA Sendse Operations
- Test 7.6: RDMA Read After Prolonged RDMA Sendseinv Operations

Step H

Group 8: Parameters:

- Test 8.1: Markers Support
- Test 8.2: CRC Support

12 Transport Independent Interop Procedures using OFED

12.1 TI iSER using OFED

12.1.1 IB Setup

Connect initiator/target to switch as well as run one or more SMs (embedded in the switch or host based). If more than one SM, let the SMs split into master and slave.

Optional: In the procedures below, an IB analyzer can be inserted in the appropriate link to obtain traces and validate the aspects of the procedures specifically detailed below in subsequent sections.

12.1.2 iWARP Setup

Connect iSER host initiator and target RNICs to an 10GbE switch.

12.1.3 Procedure

1) Load iSER target and iSER initiator to hosts from OpenFabrics tree, check iSER connection.
2) Run basic dd application from iSER initiator host connected to target.
3) [IB Specific Test] Run basic dd application from iSER initiator host connected to target. Kill the master SM while test is running and check that it completes properly.
4) Unload iSER initiator from a Host and check iSER connection properly disconnected on a target host.
5) Unload iSER target from a Host and check iSER connection properly disconnected on an initiator host.
6) [IB Specific Test] Repeat steps 2-5 now with the previous slave SM (we did not actually stop the target).

12.2 TI NFS over RDMA using OFED

Note: This procedure was written by NetApp and Open Grid Computing. For additional help, please use the following links:

1) http://lxr.linux.no/linux+v2.6.28/Documentation/filesystems/nfs-rdma.txt
2) http://www.connectathon.org/nfstests.html
3) nfs-rdma-devel@lists.sourceforge.net

12.2.1 Installation

1) Verify that nfs-utils-1.1.2 or greater is installed on all the clients
   a) /sbin/mount.nfs -V
   b) If the version is less than 1.1.2 or the command does not exist, you should install the latest version of nfs-utils.
      i) http://www.kernel.org/pub/linux/utils/nfs
2) After building the nfs-utils package, there will be a mount.nfs binary in the utils/mount directory. This binary can be used to initiate NFS v2, v3, or v4 mounts. To initiate a v4 mount, the binary must be called mount.nfs4. The standard technique is to create a symlink called mount.nfs4 to mount.nfs. This mount.nfs binary should be installed at /sbin/mount.nfs as follows:
   a) sudo cp utils/mount/mount.nfs /sbin/mount.nfs

      Note: mount.nfs and therefore nfs-utils-1.1.2 or greater is only needed on the NFS client machine. You do not need this specific version of nfs-utils on the server. Furthermore, only the mount.nfs command from nfs-utils-1.1.2 is needed on the client.
3) Verify that you are using a Linux kernel with NFS/RDMA
   a) The NFS/RDMA client and server are both included in the mainline Linux kernel version 2.6.25 and later. This and other versions of the 2.6 Linux kernel can be found at:
      i) ftp://ftp.kernel.org/pub/linux/kernel/v2.6/
   b) Download the sources as needed and place them in an appropriate location
4) Configure the RDMA stack
   a) Make sure your kernel configuration has RDMA support enabled. Under Device Drivers -> InfiniBand support, update the kernel configuration to enable InfiniBand support [**Note**: the option name is misleading. Enabling InfiniBand support is required for all RDMA devices (IB, iWARP, etc.)].
   b) Enable the appropriate IB HCA support (mlx4, mthca, ehca, ipath, etc.) or iWARP adapter support (amso, cxgb3, etc.).
   c) If you are using InfiniBand, be sure to enable IP-over-InfiniBand support.
5) Configure the NFS client and server

a) Your kernel configuration must also have NFS file system support and/or NFS server support enabled. These and other NFS related configuration options can be found under File Systems -> Network File Systems.

6) Build, install, reboot
   a) The NFS/RDMA code will be enabled automatically if NFS and RDMA are turned on. The NFS/RDMA client and server are configured via the hidden SUNRPC_XPRT_RDMA config option that depends on SUNRPC and INFINIBAND. The value of SUNRPC_XPRT_RDMA will be:
      i) - N if either SUNRPC or INFINIBAND are N, in this case the NFS/RDMA client and server will not be built
      ii) - M if both SUNRPC and INFINIBAND are on (M or Y) and at least one is M, in this case the NFS/RDMA client and server will be built as modules
      iii) - Y if both SUNRPC and INFINIBAND are Y, in this case the NFS/RDMA client and server will be built into the kernel
   b) Therefore, if you have followed the steps above and turned on NFS and RDMA, the NFS/RDMA client and server will be built.
   c) Build a new kernel, install it, boot it.
7) Check RDMA Setup
   a) If you are using InfiniBand, make sure there is a Subnet Manager (SM) running on the network.
   b) To further test the InfiniBand software stack, use IPoIB to ping two hosts
8) Check NFS Setup
   a) For the NFS components enabled above (client and/or server), test their functionality over standard Ethernet using TCP/IP or UDP/IP.
9) NFS/RDMA Setup
   a) Use two machines, one to act as the client and one to act as the server.
   b) On the server system, configure the /etc/exports file and start the NFS/RDMA server. Exports entries with the following formats have been tested:
      i) /vol0 192.168.0.47(fsid=0,rw,async,insecure,no_root_squash)
      ii) /vol0 192.168.0.0/255.255.255.0(fsid=0,rw,async,insecure,no_root_squash)
   c) The IP address(es) is (are) the client's IPoIB address for an InfiniBand HCA or the client's iWARP address(es) for an RNIC.

      Note: The "insecure" option must be used because the NFS/RDMA client does not use a reserved port.
   d) Start the NFS server
      i) If the NFS/RDMA server was built as a module (CONFIG_SUNRPC_XPRT_RDMA=m in kernel config), load the RDMA transport module:
         1. $ modprobe svcrdma

ii) Regardless of how the server was built (module or built-in), start the server:
   1. $ /etc/init.d/nfs start **or** $ service nfs start
iii) Instruct the server to listen on the RDMA transport:
   1. $ echo rdma 20049 > /proc/fs/nfsd/portlist

e) On the client system
i) If the NFS/RDMA client was built as a module (CONFIG_SUNRPC_XPRT_RDMA=m in kernel config), load the RDMA client module:
   1. $ modprobe xprtrdma.ko
ii) Regardless of how the client was built (module or built-in), use this command to mount the NFS/RDMA server:
   1. $ mount -o rdma,port=20049 <IPoIB-server-name-or-address>:/<export> /mnt
iii) To verify that the mount is using RDMA, run "cat /proc/mounts" and check the "proto" field for the given mount.

12.2.2 NFS/RDMA Test Procedure

1) Please see Connectathon for instructions on how to run the available tests:
2) Run the following tests
   a) Test 1 - File and directory creation
   b) Test 2 - File and directory removal
   c) Test 3 - Lookups across mount point
   d) Test 4 - Setattr, getattr, and lookup
   e) Test 4a - Getattr and lookup
   f) Test 5 - Read and write
   g) Test 5a - Write
   h) Test 5b - Read
   i) Test 6 - Readdir
   j) Test 7 - Link and rename
   k) Test 7a - Rename
   l) Test 7b - Link
   m) Test 8 - Symlink and readlink
   n) Test 9 - Statfs

12.3 TI Reliable Datagram Service (RDS) using OFED

12.3.1 RDS-Ping Procedure

1) Use the command *modprobe rds* to add RDS support to the kernel
2) Verify that the kernel supports RDS by issuing the *rds-info* command.
   a) The rds-info utility presents various sources of information that the RDS kernel module maintains. When run without any optional arguments rds-info will output all the information it knows of.
3) Start one of the Subnet Managers in the cluster

   Note: RDS is IP based so you need to provide a host address either through an out of band Ethernet connection or through IPoIB. RDS also requires the LIDs to be set in an InfiniBand Fabric and therefore an SM must be run.

   Note: All SMs in the fabric should be tested.
4) Chose a host and use *rds-ping host* to communicate with every other end point in the fabric.
   a) rds-ping is used to test whether a remote node is reachable over RDS. Its interface is designed to operate in a similar way to the standard ping(8) utility, even though the way it works is pretty different.
   b) rds-ping opens several RDS sockets and sends packets to port 0 on the indicated host. This is a special port number to which no socket is bound; instead, the kernel processes incoming packets and responds to them.
5) Verify that all nodes respond without error.

12.3.2 RDS-Stress Procedure

1) Chose a host and start a passive receiving session for the RDS Stress test. It only needs to be told what port to listen on.
   a) $ rds-stress -p 4000
2) Chose a second host and start an active sending instance giving it the address and port at which it will find a listening passive receiver. In addition, it is given configuration options which both instances will use.
   a) $ rds-stress -T 5 -s recvhost -p 4000 -t 1 -d 1
3) Every second, the parent process will display statistics of the ongoing stress test. If the -T option is given, the test will terminate after the specified time and a summary is printed.
4) Verify that the test completes without error.
5) Repeat steps 1-4 until all end points in the cluster have been tested.

12.4 TI SDP using OFED

12.4.1 IB Setup

Connect the HCAs and switches as per the Architected Network and make sure that no SM is running on the Fabric.

This procedure, as the previous ones, will be based on the cluster connectivity. An SM/SA which supports IPoIB (sufficient IB multicast support) will be running on the HCAs, or on a switch with an embedded SM/SA or a third HCA which would only run SM/SA for the partner pair (with a switch in the middle). This procedure has been developed for Linux and maybe ported to Windows if there is sufficient vendor support.

Optional: In the procedures below, an IB analyzer can be inserted in the appropriate link to obtain traces and validate the aspects of the procedures specifically detailed below in subsequent sections.

12.4.2 iWARP Setup

Connect SDP host client and server RNICs to an 10GbE switch.

12.4.3 Installation Requirements

Make sure the following are installed on all nodes:

1) vsftpd - for SFTP application.
2) sshd - for SCP application.

12.4.4 Creating a user name

Special account for this should be created as follows:

1) Username: interop.
2) Password: openfabrics.

12.4.5 Environment Variables

1) Set LD_PRELOAD to:
   a) On 64bit machines - /DEFAULT_INSTALL_LOCATION/lib64/libsdp.so
   b) On 32bit machines - /DEFAULT_INSTALL_LOCATION /lib/libsdp.so
   c) **Example**: export LD_PRELOAD=/usr/local//lib64/libsdp.so
2) Set SIMPLE_LIBSDP to 1 - this says to use SDP
   a) **Example**: export SIMPLE_LIBSDP=1
3) After setting the environment variables restart the xinetd.
   a) **Example**: /etc/init.d/xinetd restart

12.4.6 netperf procedure

1) Start an SM (all SM's will need to be tested) and let it initialize
   a) Verify that the running SM is the one you started.
2) Start a netperf server on one node

a) **Example**: /netperf -p {port number}

3) From all the other nodes run:
   a) [For IB] . /net perf -p {port number} -H {server node's IPoIB} -l 1 -t TCP_STREAM -- -m {message size} -s {local buffer size}
   b) [For iWARP] . /net perf -p {port number} -H {server node's IP} -l 1 -t TCP_STREAM -- -m {message size} -s {local buffer size}
   c) **Example**: /net perf -p 2006 -H 11.4.10.36 -l 1 -t TCP_STREAM -- -m 1000 -s 1024
   d) Where message size is 10, 100, 1000, 10000 and local buffer size is 1024, 6000.
   e) Repeat these steps until all message sizes and all buffer sizes have been used from all nodes
   f) Kill the netperf server
4) Repeat step #2 and #3 with a different node acting as the netperf server until all nodes have done so.
5) Repeat the netperf procedure with a different SM running until all available SMs have been used.

   Note: All nodes are expected to act as a server and as a client using every SM. All operations must finish successfully for the device to pass 10.11.6 overall.

12.4.7 SFTP PROCEDURE

SFTP procedures require an SFTP server to be configured on each machine in the partner pair. An SFTP client needs to be available on each machine as well. The default RHEL install includes both.

A 4 MB file will be SFTP'd to the partner and then SFTP'd back and binary compared to the original file; this will be done in each direction and then bidirectional using every SM available.

12.4.7.1 SETUP

1) Make sure vsftpd is installed on each node for SFTP application.
2) A special account for this should be created as follows:
   a) Username: Interop
   b) Password: openfabrics

12.4.7.2 Procedure

1) Run SFTP server on all nodes.
   a) **Example**: /etc/init.d/vsftpd start
2) Verify SDP is running.
   a) lsmod | grep sdp
   b) ib_sdp should be greater than 0 - reference count should be greater than 0. Each connection opens three reference counts.

c) During these transactions double check that sdp connection has been established, you can see it in /proc/net/sdp/conn_main.

3) Start an SM (all SM's will need to be tested) and let it initialize
   a) Verify that the running SM is the one you started.
4) SFTP:
   a) Connect an HCA pair via SFTP on IPoIB using the specified user name and password.
   b) Put the 4MB file to the /tmp dir on the remote host.
   c) Get the same file to your local dir again.
   d) Compare the file using the command *cmp tfile tfile.orig.*
      i) The two must be identical
   e) Repeat the procedure with a different SM.
5) Repeat the procedure with a different SM.

Note: Every node must SFTP the 4MB file to all others using all SM's and the files must be identical as determined by the binary compare in order for the device to pass 10.11.7 overall.

12.4.8 SCP PROCEDURE

A 4MB file will be SCP'd to the partner and then SCP'd back and binary compared to the original file, this will be done in each direction and then bidirectional using every SM available.

12.4.8.1 SETUP

1) A special account for this should be created as follows:
   a) Username: Interop
   b) Password: openfabrics

12.4.8.2 Procedure

1) Start an SM (all SM's will need to be tested) and let it initialize
   a) Verify that the running SM is the one you started.
2) SCP:
   a) Put the 4MB file to the /tmp dir on the remote host via SCP.
   b) Get the same file to your local dir again via SCP.
   c) Compare the file using the command *cmp tfile tfile.orig.*
      i) The two must be identical
   d) Repeat step #2 with a different HCA pair until all HCAs have been tested with all others (All to All).
3) Repeat the procedure with a different SM.

Note: Every node must SCP the 4MB file to all others using all SM's and the files must be identical as determined by the binary compare in order for the device to pass 10.11.8 overall.

Note: Sections 10.11.6, 10.11.7 and 10.11.8 must pass using the configuration determined by sections 10.11.1/10.11.2, 10.11.3, 10.11.4 and 10.11.5 for the device to pass SDP overall.

12.5 TI uDAPLTEST Commands using OFED

Server Command: dapltest -T S -D <ia_name>

12.5.1 IB Setup

- The /etc/dat.conf needs to be verified to be sure that the correct interface is used. By default the dapl interface for IB is ib0 and for iWARP is eth2. If these are not correct for the current cluster then errors will occur.
- It is also important to verify that the desired dapl library is being used.
- To run dapltest on IB, an SM needs to be running.

12.5.2 Group 1: Point-to-Point Topology

[1.1] 1 connection and simple send/recv:

- dapltest -T T -s <server_name> -D <ia_name> -i 100 -t 1 -w 1 -R BE
- client SR 256 1 server SR 256 1

[1.2] Verification, polling, and scatter gather list:

- dapltest -T T -s <sever_name> -D <ia_name> -i 100 -t 1 -w 1 -V -P -R BE
- client SR 1024 3 -f \
- server SR 1536 2 -f

12.5.3 Group 2: Switched Topology

InfiniBand Switch: Any InfiniBand switch

iWARP Switch: 10 GbE Switch

[2.1] Verification and private data:

- dapltest -T T -s <server_name> -D <ia_name> -i 100 -t 1 -w 1 -V -P -R BE
- client SR 1024 1 \
- server SR 1024 1

[2.2] Add multiple endpoints, polling, and scatter gather list:

- dapltest -T T -s <server_name> -D <ia_name> -i 100 -t 1 -w 10 -V -P -R
- BE client SR 1024 3 \
- server SR 1536 2

[2.3] Add RDMA Write :

- dapltest -T T -s <server_name> -D <ia_name> -i 100 -t 1 -w 1 -V -P -R BE
- client SR 256 1 \
- server RW 4096 1 server SR 256 1

[2.4] Add RDMA Read:

- dapltest -T T -s <server_name> -D <ia_name> -i 100 -t 1 -w 1 -V -P -R BE
- client SR 256 1 \
- server RR 4096 1 server SR 256 1

12.5.4 Group 3: Switched Topology with Multiple Switches

[3.1] Multiple threads, RDMA Read, and RDMA Write:

- dapltest -T T -s <server_name> -D <ia_name> -i 100 -t 4 -w 8 -V -P -R BE
- client SR 256 1 \
- server RR 4096 1 server SR 256 1 client SR 256 1 server RR 4096 1 \
- server SR 256 1

[3.2] Pipeline test with RDMA Write and scatter gather list:

- dapltest -T P -s <server_name> -D <ia_name> -i 1024 -p 64 -m p RW 8192 2

[3.3] Pipeline with RDMA Read:

- **InfiniBand**: dapltest -T P -s <server_name> -D <ia_name> -i 1024 -p 64 -m p RR 4096 2
- **iWARP**: dapltest -T P -s <server_name> -D <ia_name> -i 1024 -p 64 -m p RR 4096 1

[3.4] Multiple switches:

- dapltest -T T -s <server_name> -D <ia_name> -i 100 -t 1 -w 10 -V -P -R
- BE client SR 1024 3 \
- server SR 1536 2

12.6 TI BASIC RDMA Interop using OFED - Command Line

12.6.1 Purpose

To demonstrate the ability of endpoints to exchange core RDMA operations across a simple network path. This test procedure validates the operation of endpoints at the RDMA level, in a simple network configuration.

The Basic RDMA interop test identifies interoperability issues in one of four ways:

- The inability to establish connections between endpoints
- The failure of RDMA operations to complete
- Incorrect data after the completion of RDMA exchanges
- Inconsistent performance levels.

12.6.2 General Setup

The RDMA interop procedure can be carried out using the OFA Verbs API to create RDMA Connections and send RDMA operation or by using a 3rd party traffic generation tool such as XANStorm.

12.6.3 Topology

The topology of the network that interconnects the switches can be changed to validate operation of the endpoints over different networks paths. It is recommended that this procedure first be executed between endpoints connected by a single switch, and then the process repeated for more complex network configurations.

12.6.4 IB Setup

Connect endpoints to switch and run one or more SMs (embedded in the switch or host based).

Optional: Insert analyzer in the appropriate link to obtain traces as needed.

12.6.5 iWARP Setup

Connect iWARP RDMA endpoints to an 10GbE switch.

Optional: insert analyzer to capture traces as needed.

12.6.6 RDMA Connectivity Setup

Create two IP connections between each unique pair of RDMA interfaces involved in the testing process. The creation of two IP connections per interface pair ensures that the different semantics associated with active and passive sides of the connection are exercised.

1) For each unique pair of RDMA interfaces create two RDMA streams where each RDMA interface is sending RDMA data (Requestor) on a connection and receiving RDMA data (Target) on the other connection.

12.6.7 Small RDMA READ Procedure

1) Select the two devices that will be tested:
2) On the server device issue the following command on command line:

a) ib_read_bw -d <dev_name> -i <port>

3) On the client device issue the following command on command line:

a) ib_read_bw -d <dev_name> -i <port> -s 1 -n 100000

4) Verify that the operation completed without error and the level of performance achieved is reasonable and as expected.

12.6.8 Large RDMA READ Procedure

1) Select the two devices that will be tested:

2) On the server device issue the following command on command line:

a) ib_read_bw -d <dev_name> -i <port>

3) On the client device issue the following command on command line:

a) ib_read_bw -d <dev_name> -i <port>-s 1000000 -n 100

4) Verify that the operation completed without error and the level of performance achieved is reasonable and as expected.

12.6.9 Small RDMA Write Procedure

1) Select the two devices that will be tested:

2) On the server device issue the following command on command line:

a) ib_write_bw -d <dev_name> -i <port>

3) On the client device issue the following command on command line:

a) ib_write_bw -d <dev_name> -i <port> -s 1 -n 100000

4) Verify that the operation completed without error and the level of performance achieved is reasonable and as expected.

12.6.10 Large RDMA Write Procedure

1) Select the two devices that will be tested:

2) On the server device issue the following command on command line:

a) ib_write_bw -d <dev_name> -i <port>

3) On the client device issue the following command on command line:

a) ib_ write _bw -d <dev_name> -i <port>-s 1000000 -n 100

4) Verify that the operation completed without error and the level of performance achieved is reasonable and as expected.

12.6.11 Small RDMA SEND Procedure

This procedure may fail due to the inability of a endpoint to repost the consumed buffers.

1) Select the two devices that will be tested:

2) On the server device issue the following command on command line:

a) ib_ send _bw -d <dev_name> -i <port>

3) On the client device issue the following command on command line:

a) ib_writesend_bw -d <dev_name> -i <port> -s 1 -n 100000

4) Verify that the operation completed without error and the level of performance achieved is reasonable and as expected.

12.6.12 Large RDMA SEND Procedure

This procedure may fail due to the inability of a endpoint to repost the consumed buffers.

1) Select the two devices that will be tested:
2) On the server device issue the following command on command line:
   a) ib_ send _bw -d <dev_name> -i <port>
3) On the client device issue the following command on command line:
   a) ib_ send _bw -d <dev_name> -i <port>-s 1000000 -n 100
4) Verify that the operation completed without error and the level of performance achieved is reasonable and as expected.

Example:
DevInfo - Server

```
hca_id: mthca0
        fw_ver:                 1.2.0
        node_guid:              0002:c902:0020:b4dc
        sys_image_guid:         0002:c902:0020:b4df
        vendor_id:              0x02c9
        vendor_part_id:         25204
        hw_ver:                 0xA0
        board_id:               MT_0230000001
        phys_port_cnt:          1
                port:   1
                        state:          PORT_ACTIVE (4)
                        max_mtu:        2048 (4)
                        active_mtu:     2048 (4)
                        sm_lid:         1
                        port_lid:       2
                        port_lmc:       0x00
```

Command Line: ib_read_bw -d mthca0 -i 1

DevInfo - Client

```
hca_id: mlx4_0
        fw_ver:                 2.2.238
        node_guid:              0002:c903:0000:1894
        sys_image_guid:         0002:c903:0000:1897
        vendor_id:              0x02c9
        vendor_part_id:         25418
        hw_ver:                 0xA0
        board_id:               MT_04A0110002
        phys_port_cnt:          2
                port:   1
```

```
state:              PORT_ACTIVE (4)
max_mtu:            2048 (4)
active_mtu:         2048 (4)
sm_lid:             1
port_lid:           1
port_lmc:           0x00
```

Command Line: ib_send_bw -d mlx4_0 -i 1 10.0.0.1 -s 1 -n 100

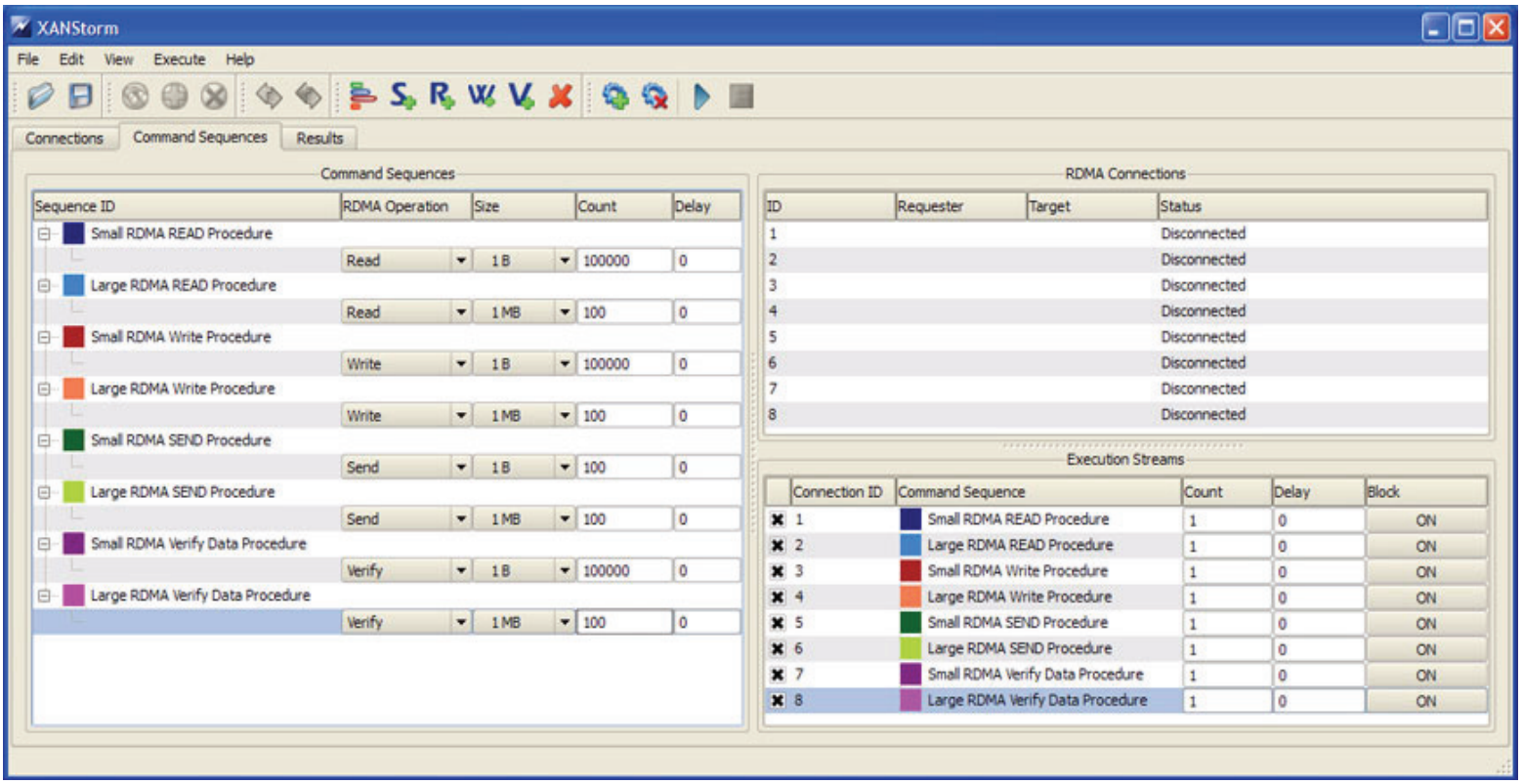
12.7 TI BASIC RDMA INTEROP USING OFED - USING XANSTORM

12.7.1 Load XANStorm Test Configuration file

```
<?xml version="2.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE xan:iWITTConfiguration>
<xan:iWITTConfiguration>
 <iWARPAgentList/>
 <RDMAStreamList>
  <RDMAStream ID="1" >
   <Requester></Requester>
   <Target></Target>
  </RDMAStream>
  <RDMAStream ID="2" >
   <Requester></Requester>
   <Target></Target>
  </RDMAStream>
  <RDMAStream ID="3" >
   <Requester></Requester>
   <Target></Target>
  </RDMAStream>
  <RDMAStream ID="4" >
   <Requester></Requester>
   <Target></Target>
  </RDMAStream>
  <RDMAStream ID="5" >
   <Requester></Requester>
   <Target></Target>
  </RDMAStream>
  <RDMAStream ID="6" >
   <Requester></Requester>
   <Target></Target>
  </RDMAStream>
  <RDMAStream ID="7" >
   <Requester></Requester>
   <Target></Target>
  </RDMAStream>
  <RDMAStream ID="8" >
   <Requester></Requester>
   <Target></Target>
  </RDMAStream>
 </RDMAStreamList>
 <CommandSequenceList>
  <CommandSequence ID="Small RDMA READ Procedure" >
   <RDMAOperation size="   1 B" count="1000000" type="Read" delay="0" />
  </CommandSequence>
  <CommandSequence ID="Large RDMA READ Procedure" >
   <RDMAOperation size="  64 MB" count="10000" type="Read" delay="0" />
  </CommandSequence>
  <CommandSequence ID="Small RDMA Write Procedure" >
   <RDMAOperation size="   1 B" count="1000000" type="Write" delay="0" />
  </CommandSequence>
```

```
 <CommandSequence ID="Large RDMA Write Procedure" >
  <RDMAOperation size="  64 MB" count="10000" type="Write" delay="0" />
 </CommandSequence>
 <CommandSequence ID="Small RDMA SEND Procedure" >
  <RDMAOperation size="   1 B" count="1000000" type="Send" delay="0" />
 </CommandSequence>
 <CommandSequence ID="Large RDMA SEND Procedure" >
  <RDMAOperation size="  64 MB" count="10000" type="Send" delay="0" />
 </CommandSequence>
 <CommandSequence ID="Small RDMA Verify Data Procedure" >
  <RDMAOperation size="   1 B" count="1000000" type="Verify" delay="0" />
 </CommandSequence>
 <CommandSequence ID="Large RDMA Verify Data Procedure" >
  <RDMAOperation size="  64 MB" count="10000" type="Verify" delay="0" />
 </CommandSequence>
</CommandSequenceList>
<ExecutionStreamList>
 <ExecutionStream block="ON" count="1" checked="true" delay="0" >
  <RDMAStream>1</RDMAStream>
  <CommandSequence>Small RDMA READ Procedure</CommandSequence>
 </ExecutionStream>
 <ExecutionStream block="ON" count="1" checked="true" delay="0" >
  <RDMAStream>2</RDMAStream>
  <CommandSequence>Large RDMA READ Procedure</CommandSequence>
 </ExecutionStream>
 <ExecutionStream block="ON" count="1" checked="true" delay="0" >
  <RDMAStream>3</RDMAStream>
  <CommandSequence>Small RDMA Write Procedure</CommandSequence>
 </ExecutionStream>
 <ExecutionStream block="ON" count="1" checked="true" delay="0" >
  <RDMAStream>4</RDMAStream>
  <CommandSequence>Large RDMA Write Procedure</CommandSequence>
 </ExecutionStream>
 <ExecutionStream block="ON" count="1" checked="true" delay="0" >
  <RDMAStream>5</RDMAStream>
  <CommandSequence>Small RDMA SEND Procedure</CommandSequence>
 </ExecutionStream>
 <ExecutionStream block="ON" count="1" checked="true" delay="0" >
  <RDMAStream>6</RDMAStream>
  <CommandSequence>Large RDMA SEND Procedure</CommandSequence>
 </ExecutionStream>
 <ExecutionStream block="ON" count="1" checked="true" delay="0" >
  <RDMAStream>7</RDMAStream>
  <CommandSequence>Small RDMA Verify Data Procedure</CommandSequence>
 </ExecutionStream>
 <ExecutionStream block="ON" count="1" checked="true" delay="0" >
  <RDMAStream>8</RDMAStream>
  <CommandSequence>Large RDMA Verify Data Procedure</CommandSequence>
 </ExecutionStream>
</ExecutionStreamList>
</xan:iWITTConfiguration>
```

12.7.2 Run XANStorm Application



12.8 TI RDMA Operations using OFED - Command Line

12.8.1 Purpose

This test is designed to identify problems that arise when RDMA operations are performed over interconnection devices in the fabric. The test is not designed to measure the forwarding rate or switching capacity of a device, but does use performance measures to identify failures.

Test failures are identified by the following events:

- The inability to establish connections between endpoints
- The failure of RDMA operations to complete
- Incorrect data after the completion of RDMA exchanges
- Inconsistent performance levels.

12.8.2 General Setup

The RDMA interop procedure can be carried out using the OFA Verbs API to create RDMA Connections and send RDMA operation or by using a 3rd party traffic generation tool such as XANStorm.

12.8.3 Topology

This test does not define a detailed topology and can be used either on a single switch or across a RDMA fabric that may include gateways to and from other technologies. The test configuration depends on the number of endpoints available to perform the testing.

12.8.4 Switch Load

The switch load test validates proper operation of a switch when processing a large number of small RDMA frames. This test is analogous to normal switch testing.

1) Attach a device to each port on the switch.
2) Select two ports on the switch to test (This will be your control stream)
3) Generate RDMA WRITE Operations of size 1024 bytes 100, 000 times on each device by issuing the following commands
   a) On the server device issue the following command on command line:
      i) ib_read_bw -d <dev_name> -i <port>
   b) On the client device issue the following command on command line:
      i) ib_read_bw -d <dev_name> -i <port> -s 1024 -n 100000
4) This must be done on both devices at the same time.
5) On all other pairs generate RDMA WRITE Operations of size 1 byte continuously until the control stream completes.
6) Repeat above steps until all port pairs are tested.
7) Repeat the above steps with all endpoint pairs, except the control stream changed such that the size of the RDMA WRITE operation is 1,000,000 bytes (~1 MB)

12.8.5 Switch FAN in

The switch fan in test attempts to validate proper operation of RDMA exchanges in the presence of traffic loads that exceed the forwarding capacity of the switch. The test requires a minimum of two switches that are interconnected by one port pair.

1) Connect all possible endpoint pairs such that data exchanges between pairs must traverse the pair of ports interconnecting the switch. The control connections must be across the interconnect network.
2) Select two ports such that it has to cross both switches. (This will be your control stream)
3) Generate RDMA WRITE Operations of size 1024 bytes 100, 000 times on each device by issuing the following commands
   a) On the server device issue the following command on command line:
      i) ib_read_bw -d <dev_name> -i <port>
   b) On the client device issue the following command on command line:
      i) ib_read_bw -d <dev_name> -i <port> -s 1024 -n 100000
4) This must be done on both devices at the same time.
5) On all other pairs generate RDMA WRITE Operations of size 1 byte continuously until the control stream completes.
6) Repeat above steps until all port pairs are tested.
7) Repeat the above steps with all endpoint pairs, except the control stream changed such that the size of the RDMA WRITE operation is 1,000,000 bytes (~1 MB)

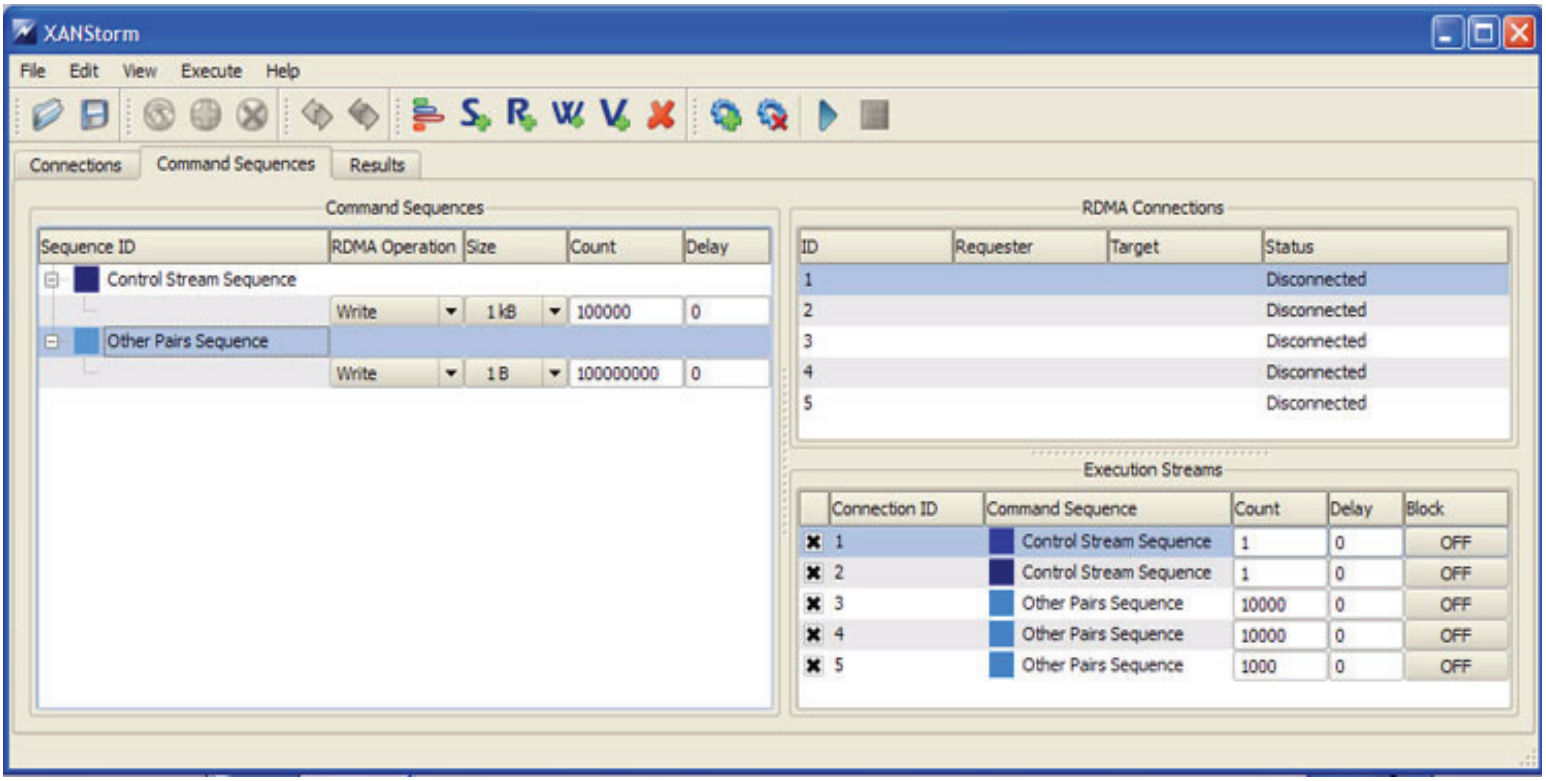
12.9 TI RDMA OPERATIONS USING OFED - USING XANSTORM

12.9.1 Load XANStorm Test Configuration file

```
<?xml version="2.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE xan:iWITTConfiguration>
<xan:iWITTConfiguration>
 <iWARPAgentList/>
 <RDMAStreamList>
  <RDMAStream ID="1" >
   <Requester></Requester>
   <Target></Target>
  </RDMAStream>
  <RDMAStream ID="2" >
   <Requester></Requester>
   <Target></Target>
  </RDMAStream>
  <RDMAStream ID="3" >
   <Requester></Requester>
   <Target></Target>
  </RDMAStream>
  <RDMAStream ID="4" >
   <Requester></Requester>
   <Target></Target>
  </RDMAStream>
  <RDMAStream ID="5" >
   <Requester></Requester>
   <Target></Target>
  </RDMAStream>
 </RDMAStreamList>
 <CommandSequenceList>
  <CommandSequence ID="Control Stream Sequence" >
   <RDMAOperation size="   1 kB" count="100000" type="Write" delay="0" />
  </CommandSequence>
  <CommandSequence ID="Other Pairs Sequence" >
   <RDMAOperation size="   1 B" count="100000000" type="Write" delay="0" />
  </CommandSequence>
 </CommandSequenceList>
 <ExecutionStreamList>
  <ExecutionStream block="OFF" count="1" checked="true" delay="0" >
   <RDMAStream>1</RDMAStream>
   <CommandSequence>Control Stream Sequence</CommandSequence>
  </ExecutionStream>
  <ExecutionStream block="OFF" count="1" checked="true" delay="0" >
   <RDMAStream>2</RDMAStream>
   <CommandSequence>Control Stream Sequence</CommandSequence>
  </ExecutionStream>
  <ExecutionStream block="OFF" count="10000" checked="true" delay="0" >
   <RDMAStream>3</RDMAStream>
   <CommandSequence>Other Pairs Sequence</CommandSequence>
  </ExecutionStream>
  <ExecutionStream block="OFF" count="10000" checked="true" delay="0" >
```

```
  <RDMAStream>4</RDMAStream>
  <CommandSequence>Other Pairs Sequence</CommandSequence>
 </ExecutionStream>
 <ExecutionStream block="OFF" count="1000" checked="true" delay="0" >
  <RDMAStream>5</RDMAStream>
  <CommandSequence>Other Pairs Sequence</CommandSequence>
 </ExecutionStream>
</ExecutionStreamList>
</xan:iWITTConfiguration>
```

12.9.2 Run XANStorm Application



12.10 TI MPI - HP-MPI using OFED

12.10.1 Cluster setup

Note: The tests referenced below are in the following location and contain both 32 and 64 bit versions:

http://www.iol.unh.edu/downloads/OFA/HP/ofatests_v3.tar.gz .

Note: HP-MPI is not part of the OFA Stack

1) Ethernet or some form of TCP/IP must be installed and configured on all systems.
2) OFED library path must be configured on all systems (ldconfig should be executed after OFED installation).
3) OFED uDAPL /etc/dat.conf must match /sbin/ifconfig setup. (Modify /etc/dat.conf and change the netdev reference to the appropriate interface (ib0 or ib1) being used).
4) OSs supported by HP-MPI are Red Hat Enterprise Linux AS 4 and 5, or SuSE Linux Enterprise Server 9 and 10.
5) All systems must be setup with identical user accounts (SSH access with no password prompts (key's setup) or rsh with .rhosts setup).
6) HP-MPI must be installed (in the same location) on all the machines in the cluster (or copying the HP-MPI tree to a shared directory also works).
7) Increase the max lockable memory limits on all the machines in the cluster:
   a) edit /etc/security/limits.conf and add the following:
      i) * hard memlock 500000
      ii) * soft memlock 500000
8) A shared directory is very much recommended for ease of use in running the below tests.
9) Perl should be installed and located at /usr/bin/perl (or else the exitpath/kill.pl script needs to be edited to point at an appropriate installation of perl.

12.10.2 Required Files

1) HP-MPI is packaged as a binary .rpm. Version 2.3.1 has OFED 1.4.1 support:
   a) hpmpi-2.03.01.00-20090402r.x86_64.rpm .
   b) hpmpi-2.03.01.00-20090402r.i386.rpm .
2) The version of an installed HP-MPI can be checked using mpirun -version

12.10.3 Test Suite Instructions

1) Although not absolutely required, these tests are easier to run from a shared directory, and the below instructions assume the use of one.
2) Download and unpack tests:
   a) ofatests_v2.tar.gz

3) Unpack this preferably into a shared directory and cd into the directory it unpacks into:
   b) tar zxvf ofatests_v2.tar.gz
   c) cd ofatests/
4) Construct a file "hosts" on the machine you'll be running "mpirun" from. The format should be:
   a) first_machine_name 2
   b) second_machine_name 2
   c) etc
5) Later when this file is given with the "-hostfile" option to "mpirun", HP-MPI will launch two ranks on the first machine, two on the second, etc.

12.10.4 BUILDING THE TESTS

1) All the HP-MPI tests are shipped as a single binary hpmpitest.x which has a permanent unrestricted license built in. It will run any of the following five tests based on the first command line argument:
   a) IMB (command line "IMB")
   b) rings2 (command line "rings2")
   c) hello world (command line "hw")
   d) fork (command line "fork")
   e) ping pong ring (command line "ppr")
   f) alltoone (command line "alltoone")

   Note: For reference, these tests are included individually in the ofatests/src directory (except IMB which is available from Intel).

12.10.5 RUNNING THE TESTS

1) The test directory contains two scripts: "runit.ib" and "runit.iwarp", which runs the test in several different modes:
   a) for runit.ib:
      i) IBV in RDMA mode with IBV intra-host
      ii) IBV in SRQ mode with IBV intra-host
      iii) UDAPL in RDMA mode with UDAPL intra-host
   b) for runit.iwarp:
      i) UDAPL in RDMA mode with UDAPL intra-host
2) Use runit.ib on a cluster with InfiniBand cards, or runit.iwarp on a cluster with iWARP cards.

12.10.6 CHECKING THE TEST STATUSES

1) The "runit.*" scripts run all the tests one after the other, reporting "passed" or "FAILED" to stdout for each. If failures occur, they are logged in

LOG.ib.failures or LOG.iwarp.failures along with the full stdout/stderr for the failed job.

12.10.7 TEST DESCRIPTIONS

1) The HP-MPI test suite includes five tests:
   a) detection
      i) This is a simple ping pong application that is used to test HP-MPI's default interconnect selection. The other test cases use explicit interconnect selection.
   b) IMB
      i) This is the Intel MPI Benchmark. If this passes, then the basic interoperability of HP-MPI with the installed OFED is confirmed.
   c) rings2
      i) This is a proprietary HP test which has a good history of stressing interconnects to the point of failure. It also includes 1sided operations.
   d) fork
      i) New RDMA implementations often have fork issues, and as new OS kernels come out the fork problems sometimes re-appear. This test makes a point of stressing that code path.
   e) exitpath
      i) The purpose of this test is simply to make sure machines and OFED drivers etc remain stable when applications repeatedly terminate abnormally.
   f) alltoone
      i) This test has all the ranks send a flood of messages to rank 0, to make sure the interconnect can handle heavy load in that message pattern.

12.11 TI MPI - Intel MPI using OFED

12.11.1 General Issues

Note: Intel MPI is not part of the OFA Stack

1) Network configuration requirements
   a) Ethernet must be installed and configured on all systems.
   b) DNS names must match hostnames.
   c) /etc/hosts should be setup with static IB hostnames and addresses.
2) OFED Installation requirements
   a) OFED library path must be configured on all systems (ldconfig should be executed after OFED installation).
   b) OFED uDAPL /etc/dat.conf must match /sbin/ifconfig setup.
3) Setup Requirements
   a) All systems must be setup with identical user accounts on all nodes (SSH access with no password prompts (key's setup) or rsh with .rhosts setup).
   b) Requires NFS setup from headnode and mount points (/home/test/export) on user accounts.

      Note: any node on the cluster can be setup as the headnode.
   c) MPI testing requires a reliable IB fabric without other fabric interop testing occurring.
4) Here is the location for the free Intel MPI runtime environment kit
   a) http://www.intel.com/cd/software/products/asmo-na/eng/222346.htm
5) Here is the location for the Intel MPI Benchmarks
   a) http://www.intel.com/cd/software/products/asmo-na/eng/cluster/mpi/219848.htm

12.11.2 Setup for the Cluster

1) Install same O/S version on homogenous x86_64 systems. (Recommend RHEL 5.2, EM64T). See the Figure 5- Intel Requirements for Homogeneous Environment and following link for details:
   a) http://www3.intel.com/cd/software/products/asmo-na/eng/308295.htm
2) Install Ethernet interface with dynamic addresses from DHCP and hostnames registered with DNS.
3) Verify "hostname" on each system returns the hostname that DNS reports.

Hardware	
Minimum Requirements	IA-32, Intel® 64 or IA-64 (formerly Itanium) architecture-based system. Examples of such Intel processors are: Intel® Pentium® 4 processor, or Intel® Xeon® processor, or Intel® Itanium® processor, or Intel® Core™2 Duo processor (example of Intel® 64 architecture) Note that it is assumed that the processors listed above are configured into homogeneous clusters 4 GB of RAM (8 GB of RAM recommended) 1 GB of hard disk space (10 GB of space recommended)
Operating System Support	
All three architectures	Red Hat* Enterprise Linux* 4.0, 5.0 SUSE* Linux Enterprise Server* (SLES) 9, 10
IA-32 and Intel 64 architectures	Microsoft* Windows Vista*
Intel® 64 and IA-64 architectures	SGI ProPack* 5
IA-32 architecture only	Microsoft Windows* XP
Intel® 64 architecture only	Red Hat Fedora 7 through 8 cAos* 2 CentOS* 4.6, 5.1 openSuSE* Linux* 10.3 Microsoft* Windows Compute Cluster Server 2003* Microsoft* Windows Server 2003* Microsoft* Windows XP Professional x64 Edition* Microsoft* Windows HPC Server 2008* Microsoft* Windows Server 2008*
Other Supported Software	
Intel® MPI Benchmarks	
Intel® Math Kernel Library	
Intel® Trace Analyzer and Collector	
Intel® C++ Compiler	
Intel® Fortran Compiler	
Microsoft* Visual Studio and Visual C++ Compilers	
GNU C, C++, and FORTRAN Compilers	
OpenFabrics* Enterprise Distribution (OFED*)	

Figure 5 - Intel Requirements for Homogeneous Environment

12.11.3 Setup information for OFED

1) Install the current version of OFED on all systems.
2) Bump up the max locked memory limits on the system.

   edit /etc/security/limits.conf and add the following:

   ```
   *       hard    memlock         500000
   *       soft    memlock         500000
   ```

3) Run /sbin/ldconfig to pick up new OFED library path
4) Modify /etc/hosts and add IB hostnames and addresses for the IB network interfaces
5) Modify /etc/dat.conf and change the netdev reference to the appropriate interface (ib0 or ib1) being used
6) Run OpenSM either on the headnode OR from one of the switches. Verify by pinging IB addresses on all systems.

12.11.4 Setup information for Intel MPI

1) Install Intel MPI in /opt/intel/mpi/3.1 on every system.
2) Add identical user account (/home/test) on every system. For example "useradd –m test –u 555 –g users
3) Update the .bashrc for /home/test on every system:

   ```
   export PATH=$PATH:./
   source /opt/intel/mpi/3.1/bin64/mpivars.sh
   # for IB, (mpi will default to rdssm if nothing defined)
   export I_MPI_DEVICE=rdssm
   # for ethernet
   export I_MPI_DEVICE=sock
   export MPIEXEC_TIMEOUT=180
   ulimit -c unlimited
   ```

4) Add .mpd.conf file in /home/test on every system.

   add single line "MPD_SECRETWORD=testing" to .mpd.conf

   chmod 600 /home/test/.mpd.conf
5) Add 2 mpd.hosts files in /home/test on the headnode, one for ethernet and one for IB

   Create mpd.hosts.ethernet and add a line for every system on the cluster using ethernet addresses or hostnames

   Create mpd.hosts.ib and add a line for every system on the cluster using IPoIB addresses
6) Add nfs export /home/test/export on headnode and change /etc/fstab for mount points:

   edit /etc/exports and add "/home/test/exports *(rw)" on headnode

   edit /etc/fstab and add "hostname:/home/test/exports /home/test/exports nfs" on all other nodes

7) Untar the Intel Test Suites on the headnode in /home/test/exports
8) run mpdboot on the head node. For example: if you have 6 nodes on the cluster and want to run over ethernet:

   From the /home/test directory run: “mpdboot –n 6 –r ssh –f ./mpd.host.ethernet”
9) Run test suite over Ethernet to validate your installation:

   “export I_MPI_DEVICE = sock”

   run tests…(refer to test plan)

   “mpdallexit”
10) Run test suite over IB

    export I_MPI_DEVICE = rdssm

    mpdboot –n 6 –r ssh –f ./mpd.host.ib

    run tests…. (refer to test plan)

    “mpdallexit”

12.11.5 Additional Information

1) Go to the individual test directories and follow the steps in the respective README-*.txt files. The recommended order for running the test suites in the order of increasing execution time:
   a) mpich2-test: see README-mpich2-test.txt file.
2) For Intel MPI Support Services go to:

   http://www.intel.com/support/performancetools/cluster/index.htm

   See the Intel MPI Reference Manual for Additional information

12.11.6 Intel MPI Benchmark setup

The IMB tests must be compiled with the -DCHECK compiler flag set, to enable automatic self-checking of the results. Modify the appropriate make_arch file as follow:

```
MPI_HOME      =
MPI_INCLUDE = .
LIB_PATH      =
LIBS          =
CC             = mpicc
OPTFLAGS      = -O
CLINKER       = ${CC}
LDFLAGS       =
CPPFLAGS      =
```

12.11.7 Intel IHV test suite setup

All test suites are configured, built, and run in a uniform way.

- Configure for mpich2-test: ./configure –with-mpich2=/opt/intel/mpi/3.1 –cc=mpicc –f77=mpif77 –cxx=mpicxx
- Configure for IntelMPITEST: ./configure –with-mpich2=/opt/intel/mpi/3.1

1) If you installed the library to another location, then replace the default Intel(R) MPI Library installation path "/opt/intel/mpi/2.0".

   A detailed description of the extra configuration options is contained in the respective README-*.txt file.

2) Run the tests:

   If you use a Bourne-compatible shell (sh, bash, ksh, etc.), do:

   export MPIEXEC_TIMEOUT=180

   nohup make testing > xlog 2>&1 &

   If you use a Csh-compatible shell (csh, tcsh, etc.), do:

   setenv MPIEXEC_TIMEOUT 180

   nohup make testing >&! xlog &

The expected duration of the test run is detailed in the respective README-*.txt file.

3) Check the results:

   grep ">pass" summary.xml | wc -l

   grep ">fail" summary.xml | wc -l

The exact number of passed and failed tests is specified in the respective README-*.txt file.

12.11.8 TEST PROCEDURE

These sets of tests should be run for both Intel mpich2-test and the IntelMPITEST suite:

Note: “Set ulimit –c unlimited” to capture core files in case of abnormal terminations.

Test suite mpich2-test: use default settings with no environment variables.

Test suite IntelMPITEST: use default settings with no environment variables.

12.11.9 INTERPRETING THE RESULTS

1) For mpich2-test test suites: See Table - TI - Intel MPICH2 Suite - (Not part of OFA Stack)

   The **summary.xml** file produced by the test suites has the following uniform format:

   - The file header contains information on the test suite and testing environment.
   - The rest of the file represents the results of the test suite run.

2) For IntelMPITEST test suite: See Table - TI - Intel MPI Test Suite Description - (Not part of OFA Stack)

The **Tests/summary.xml** file produced by the test suites has the following uniform format:

- The file header contains information on the test suite and testing environment
- The rest of the file represents the results of the test suite run.

12.12 TI MPI - Open MPI using OFED

12.12.1 Cluster setup

1) Network configuration requirements
   a) All systems must be reachable by each other a common network that supports TCP (Ethernet, IPoIB, etc.).
   b) All nodes must agree on the IP addresses for all TCP networks on all systems (e.g., via /etc/hosts, DNS, or some other mechanism).
   c) If multiple, physically separate OpenFabrics networks are used in the testing, then all the devices on each network must report a subnet ID through the verbs stack that is both the same as all other ports on the same physical network and unique among all other ports on other physical networks.

      Note: this is a new requirement among all the MPI's. This likely means that IB vendors will need to change the default subnet ID reported by their systems. It is only necessary for testing scenarios when multiple physically separate OpenFabrics networks are available, such as (but not limited to):

      i) so-called "multi-rail" scenarios, where one or more systems in the test have multiple OpenFabrics devices, each connected to physically separate networks
      ii) when some systems are connected to IB network A, and other systems are connected to IB network B
2) The same version of OFED must be installed in the same filesystem location on all systems under test.
3) The same version of Open MPI must be available in the same filesystem location on all systems under test.
   a) Open MPI can be installed once on a shared network filesystem that is available on all nodes, or can be individually installed on all systems. The main requirement is that Open MPI's filesystem location is the same on all systems under test.
   b) If Open MPI is built from source, the --prefix value given to configure should be the filesystem location that is common on all systems under test. For example, if installing to a network filesystem on the filesystem server, be sure to specify the filesystem location under the common mount point, not the "native" disk location that is only valid on the file server.
   c) Note that Open MPI only started supporting iWARP as of the v1.3 series. As of this writing, Open MPI v1.3 has not yet been released, but nightly "snapshot" tarballs are available on the Open MPI web site.
   d) The version of Open MPI can be obtained by running "ompi_info | head".
4) All systems must be setup with at least one identical user account. This user must be able to SSH or RSH to all systems under test from the system that will launch the Open MPI tests with no additional output to stdout or stderr (e.g., all SSH host keys should already be cached, no password/passphrase prompts should be emitted, etc.).

5) The lockable memory limits on each machine should be set to allow unlimited locked memory per process.
6) The underlying OpenFabrics network(s) used in the test should be stable and reliable.
7) No other fabric interoperability tests should be running during the Open MPI tests.
8) Note that Open MPI is included in some Linux distributions and other operating systems. Multiple versions of Open MPI can peacefully co-exist on a system as long as they are installed into separate filesystem locations (i.e., configured with a different --prefix argument). All MPI tests must be built and run with a single installation of Open MPI.
9) MPI tests should be run across at least 2 separate systems to force the use of the OpenFabrics networks (vs. using just shared memory for in-system communication).
10) Open MPI currently assumes a "mostly homogeneous" scenario, where the software and hardware of all systems under test are more-or-less the same. They should be the same general server vendor model with similar hardware accessories, running the same operating operating system with the same features enabled, etc. Some "drift" between the systems under test is acceptable (in both software and hardware), but for the scope of this interoperability testing, it is best to test with systems that are as homogeneous as possible. Specifically: this interoperability testing focuses on homogeneous and heterogeneous network hardware, not heterogeneous server hardware or software. Open MPI v1.3.x does not support heterogeneous HCA/ RNICs in a single MPI job. Such heterogeneous network hardware scenarios may be supported via manual parameter tweaking, or "out of the box" in a future version.

12.12.2 TEST SETUP

1) The following values are used in examples below:
   a) $OMPI_SOURCE_TREE: The directory where the Open MPI source code resides.
   b) $MPIHOME: The absolute directory location of the Open MPI installation that is common to all systems under test.
2) Open MPI can be used from the OFED installation, or, if a later version is required, can be downloaded and installed from the main Open MPI web site:

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

   a) If building Open MPI from source, and if the OpenFabrics libraries and headers are installed in a non-default location, be sure to use the --with-openib=<dir> option to configure to specify the OpenFabrics filesystem location.
3) Create a hostfile listing the hostname of each system that will be used in the test. If a system under test can run more than one MPI process (such as multiprocessor or multicore systems), either add a "slots" parameter after each hostname indicating how many processes to run on that system, or list the hostname as many times as MPI processes are desired. For example,

for two 4 processor systems named node1.example.com and node2.example.com:

```
shell$ cat hostfile.txt
node1.example.com slots=4
node2.example.com slots=4
shell$ cat equivalent-hostfile.txt
node1.example.com
node1.example.com
node1.example.com
node1.example.com
node2.example.com
node2.example.com
node2.example.com
node2.example.com
shell$
```

4) Open MPI defaults to probing all available networks at run-time to determine which to use. OpenFabrics testing should specifically force Open MPI to ***only*** use its OpenFabrics stack for testing purposes (e.g., do not fail over to TCP if the OpenFabrics stack is unavailable). There are three ways to force Open MPI to use the OpenFabrics stack by default:
   a) Set a per-user file that is visible on all nodes (either if the $HOME is a networked filesystem that is common to all systems under test, or this process is invoked on all systems):

```
shell$ mkdir $HOME/.openmpi
shell$ cat > $HOME/.openmpi/mca-params.conf <<EOF
btl = openib,self,sm
EOF
```

   b) Set an environment variable on the node/shell where mpirun is invoked:

```
# sh-flavored shells
shell$ export OMPI_MCA_btl=openib,self,sm
# csh-flavored shells
shell% setenv OMPI_MCA_btl openib,self,sm
```

   c) Add an extra command line parameter to mpirun (not shown in all the examples below):

```
shell$ mpirun --mca btl openib,self,sm ...rest of command line...
```

5) Open MPI includes several trivial test programs to verify basic MPI functionality. Assuming the Open MPI source tree is available, the tests can be built with:

```
shell$ cd $OMPI_SOURCE_TREE/examples
shell$ make
```

6) NetPIPE should be obtained from its main web site:

   http://www.scl.ameslab.gov/netpipe/

   a) Open MPI should be in the $PATH so that "mpicc" can be found.The test suite can then be built with:

      shell$ cd NetPIPE-3.7.1

      shell$ make mpi

7) The Intel MPI Benchmarks should be obtained from the same URL provided in the Intel MPI test section of this document.

   a) The test suite can be built with:

      shell$ cd IMB_3.1/src

      shell$ make -f make_mpich MPI_HOME=$MPIHOME

8) It may be desirable to set the shell to unlimit the size of corefiles for analysis of aborted tests. This limit should be set in the shell startup files of the test user on every node.

12.12.3 TEST PROCEDURE

1) The following values are used in examples below:

   a) $OMPI_SOURCE_TREE: The directory where the Open MPI source code resides.

   b) $MPIHOME: The absolute directory location of the Open MPI installation that is common to all systems under test.

   c) $NP: The number of MPI processes to use in the test. Unless otherwise specified, it is usually the sum of the number of processors on all systems under test.

   d) $HOSTFILE: The absolute filename location of the hostfile.

2) Ensure that the Open MPI installation includes OpenFabrics support:

   shell$ $MPIHOME/bin/ompi_info | grep openib

   MCA btl: openib (MCA v1.0, API v1.0.1, Component v1.4)

   The exact version numbers displayed will vary depending on your version of Open MPI. The important part is that a single "btl" line appears showing the openib component.

3) Basic Open MPI run-time functionality can first be verified by running simple non-MPI applications. This ensures that the test user's rsh and/or ssh settings are correct, etc.

   shell$ $MPIHOME/bin/mpirun -np $NP --hostfile $HOSTFILE hostname

   The output should show the hostname of each host listed in the hostfile. If a host was listed with "slots=X", the hostname should appear X times. The list of hostnames may appear in random order; this is normal. Note that any serial application can be run; "hostname" is a good, short test that clearly identifies that specific hosts were used, etc.

4) Basic Open MPI functionality can be verified with several trivial test programs that are included in Open MPI. Run them with:

   shell$ cd $OMPI_SOURCE_TREE/examples

shell$ $MPIHOME/bin/mpirun -np $NP --hostfile $HOSTFILE hello_c

shell$ $MPIHOME/bin/mpirun -np $NP --hostfile $HOSTFILE ring_c

The first program prints a simple "hello world" message from each MPI process; there should be one line of output from each. This test simply verifies that trivial MPI applications are able to start, properly initialize, properly finalize, and exit successfully. The lines may output out of order; this is normal.

The second program sends a message around in a ring. In addition to testing the same functionality as "hello world", it exercises basic message passing (using the OpenFabrics verbs stack, in this case). The output should indicate that a message was sent around a ring 10 times, and then that each process exited successfully. Some lines may be output out of order; this is normal.

The same two test programs are also available in C++, Fortran 77, and Fortran 90, but they are not relevant to this test.

5) NetPIPE can only be run with 2 MPI processes. It can be invoked:

shell$ cd NetPIPE-3.7.1

shell$ $MPIHOME/bin/mpirun -np $NP --bynode --hostfile $HOSTFILE \
NPmpi

The "--bynode" option forces Open MPI to place MPI processes on separate nodes (to force testing of the network, as opposed to shared memory).

NetPIPE will run through ping-pong benchmarks of a variety of message sizes. It is fairly obvious if NetPIPE hangs or fails to complete successfully.

6) The Intel MPI benchmarks can be invoked with the following:

shell$ cd IMB_3.1/src

shell$ $MPIHOME/bin/mpirun -np $NP --bynode --hostfile $HOSTFILE \
IMB-MPI1 -multi 0 PingPong PingPing

shell$ $MPIHOME/bin/mpirun -np $NP --hostfile $HOSTFILE IMB-MPI1

shell$ $MPIHOME/bin/mpirun -np $NP --hostfile $HOSTFILE IMB-IO

The first command runs just the PingPong and PingPing point-to-point benchmarks, but makes all the MPI processes active in a pairwise fashion. The "--bynode" option forces Open MPI to place successive MPI processes on separate nodes (to force testing of the network, as opposed to shared memory).

The second command runs all the benchmarks in the suite. Depending on the number of processes in the test, it may take a while to run.

The third command runs a variety of MPI file tests, each of which involve MPI message passing. You may see warnings about ADIO failing to delete files; these warnings are a known issue and are safe to ignore. Depending on the number of processes in the test and the back-end filesystem used, it may take a long time to run. Periodic "hang"-like behavior is also not uncommon (largely caused by filesystem issues). For small node/process counts, hangs shouldn't last for more than 1-2 minutes each. For larger node/process counts, the hangs may be longer.

12.13 TI MPI - Ohio State University using OFED

12.13.1 MVAPICH 1 - Setup

1) Network configuration requirements
   a) All systems must be reachable by each other a common network that supports TCP (Ethernet, IPoIB, etc.)
   b) All nodes must agree on the IP addresses for all TCP networks on all systems (e.g., via /etc/hosts, DNS, or some other mechanism).
2) The same version of OFED must be installed in the same filesystem location on all systems under test.
3) MVAPICH1 is included in OFED distributions. The updated versions of MVAPICH1 can be obtained from OpenFabrics website.
4) The same version of MVAPICH must be available in the same filesystem location on all systems under test.
   a) MVAPICH1 can be installed once on a shared network filesystem that is available on all nodes, or can be individually installed on all systems. The main requirement is that MVAPICH1 filesystem location is the same on all systems under test.
5) All systems must be setup with at least one identical user account. This user must be able to SSH or RSH to all systems under test from the system that will launch the MVAPICH1 tests with no additional output to stdout or stderr (e.g., all SSH host keys should already be cached, no password/passphrase prompts should be emitted, etc.).
6) The lockable memory limits on each machine should be set to allow unlimited locked memory per process. This can be achieved by using ulimit command.
7) The underlying IB network(s) used in the test should be stable and reliable. No other fabric interoperability tests should be running during the MVAPICH1 tests.
8) Multiple versions of MVAPICH can peacefully co-exist on a system as long as they are installed into separate filesystem locations (i.e., configured with a different --prefix argument). All tests must be built and run with a single installation of MVAPICH.
9) MVAPICH tests should be run across at least 2 separate systems to force the use of the IB networks (vs. using just shared memory for in-system communication).

12.13.2 MVAPICH 1 - Test Setup and Procedure

1) Test Setup
   a) Create a hostfile listing the hostname of each system that will be used in the test. If a system under test can run more than one MPI process (such as multiprocessor or multicore systems) list the hostname as many times as MPI processes are desired. For example, for two 2 processor systems named host1 and host2

```
$ cat hostfile.txt
host1
host1
```

host2
host2

b) Download and install Intel® MPI Benchmarks on all nodes from:

http://www.intel.com/cd/software/products/asmo-na/eng/cluster/mpi/219848.htm

Follow the instructions below to install:

i) untar downloaded archive

ii) open <natured directory>/src/make_mpich and fill in the following variables:

- MPI_HOME=<path to mvapich1 directory> #mine was /usr/mpi/gcc/mvapich-1.0.1
- CPPFLAGS= -DCHECK

iii) gmake -f make_mpich

This will install the benchmarks inside the MPI_HOME/tests directory

Note: Intel® MPI Benchmarks are installed with OFED installation by default

c) Enter all nodes and run the following commands:

i) echo "PATH=\$PATH:<path to mvapich1 directory>/bin:<path to mvapich1 directory>/tests/IMB-3.0" >> /<username>/.bashrc # or .cshrc

ii) echo "ulimit -l unlimited" >> /<username>/.bashrc # or .cshrc

iii) source /<username>/.bashrc # or .cshrc

Note: these commands may fail or produce unexpected results with a shared $HOME

2) Testing Procedure

a) The following values are used in the examples below

i) $MPIHOME - The absolute directory location of the MVAPICH installation that is common to all systems under test

ii) $NP - The number of MPI processes to use in the tests. Unless otherwise specified, it is usually the sum of the number of cores on all systems under test

iii) $HOSTFILE - The absolute location of the hostfile

b) Run Intel® MPI Benchmarks:

i) Run the PingPong and PingPing point-to-point tests

$MPIHOME/bin/mpirun -np $NP IMB-MPI1 -multi 0 PingPong PingPing -hostfile $HOSTFILE

ii) Run all the tests (PingPong, PingPing, Sendrecv, Exchange, Bcast, Allgather, Allgatherv, Alltoall, Reduce, Reduce_scatter, Allreduce, Barrier), in non-multi mode.

$MPIHOME/bin/mpirun -np $NP IMB-MPI1 -multi 0 -hostfile $HOSTFILE

12.13.3 MVAPICH 2 - SETUP

1) Download and install OFED on all nodes from:

   http://www.openfabrics.org/downloads/OFED

2) Download and install Intel® MPI Benchmarks on all nodes from:

   http://www.intel.com/cd/software/products/asmo-na/eng/cluster/mpi/219848.htm

   You will have to accept a license. Follow the instructions below to install.

   a) untar downloaded archive

   b) open <untarred directory>/src/make_mpich and fill in the following variables:

      i) MPI_HOME=<path to mvapich2 directory> #mine was /usr/mpi/gcc/mvapich2-1.0.3

      ii) CPPFLAGS= -DCHECK

   c) gmake -f make_mpich

   This will install the benchmarks inside the MPI_HOME/tests directory

3) All nodes should be physically connected.

4) Enter all nodes and run the following cmds:

   a) echo "PATH=\$PATH:<path to mvapich2 directory>/bin:<path to mvapich2 directory>/tests/IMB-3.0" >> /<username>/.bashrc # or .cshrc

   b) echo "ulimit -l unlimited" >> /<username>/.bashrc;

   c) source /<username>/.bashrc # or .cshrc

5) Create an mpi ring:

   a) Construct a file called hosts that has the following format. Include as many lines as you have hosts. Be sure to leave a blank line at the end of the file:

      i) <host>ifhn=<infiniband ip address>

   b) Run the following commands

      i) mpdboot -n `cat hosts|wc -l` -f hosts --ifhn=<localhost infiniband ip address>

      ii) mpdtrace -l #OPTIONAL, shows current ring members.

12.13.4 MVAPICH 2 - TEST PROCEDURE

Step A: Run a subnet manager from one node only.

Step B Run Intel® MPI Benchmarks:

1) Two sets of tests should be run, with these command lines

   a) mpirun -np <number of nodes X number of processors/node> IMB-MPI1 -multi 0 PingPong PingPing

   a) mpirun -np <number of nodes X number of processors/node> IMB-MPI1

   The first command runs just the PingPong and PingPing point-to-point tests, but makes all tasks active (pairwise).

The second command runs all the tests (PingPong, PingPing, Sendrecv, Exchange, Bcast, Allgather, Allgatherv, Alltoall, Reduce, Reduce_scatter, Allreduce, Barrier), in non-multi mode.

2) If the test passes shutdown current subnet manager and start another one on a different node; run both tests again.

3) Repeat until all nodes have run a subnet manager and passed all tests.

13 InfiniBand Specific Interop Procedures using WinOF

13.1 IB Link Initialize using WinOF

13.1.1 Disconnect the full topology and select a cable whose length should be a maximum of 15 meters for SDR and 10 meters for DDR when using copper cables.

Note: The WinOF Subnet Manager and diagnostics are still evolving as compared to OFED. Therefore, you must include an OFED Linux node along with the WinOF node to run diagnostics for this test.

1) Verify that no SM is running
2) Connect two devices back to back
3) ssh to the OFED node.
   a) Run "ibdiagnet -lw 4x" to verify portwidth
   b) Run "ibdiagnet -ls 2.5" to check link speed. Interpret output and compare to advertised speed.

   Note: This command will only produce output if the link speed is anything other than SDR. Keep this in mind during your interpretation of the output.
4) Repeat steps 1-3 with a different device pairing.
   a) All device pairs must be tested except target to target: HCA to HCA, HCA to Switch, HCA to Target, Switch to Switch, and Switch to Target.
   b) Each device must link to all other devices in order for the device to pass link init over all.

13.1.2 Recommendations

In order to determine Switch to Target and Switch to Switch link parameters, run commands from an HCA linked to the switch under test. This does require more interpretation of the output to differentiate the reported parameters.

13.2 IB Fabric Initialization using WinOF

13.2.1 Architect the Network we want to build.

Note: The WinOF Subnet Manager and diagnostics are still evolving as compared to OFED. Therefore, you must include an OFED Linux node along with the WinOF node to run diagnostics for this test. This will limit WinOF link testing to HCA to HCA pairs.

1) Create a table of IP addresses to assign.
2) Create topology file - this makes sure that the subnet is configured as expected - i.e. SDR and DDR links. This inserts the name of devices as well as the GUID.
3) See Figure 6- Sample Network Configuration below.

13.2.2 Connect the HCAs and switches as per the Architected Network and make sure that no SM/SA is running on the Fabric.

13.2.3 Procedure

1) Start an SM on a device and let it initialize (all SMs will need to be tested)
2) Visually verify that all devices are in the active state. Orange led will be on if the port is active.
3) Run "ibdiagnet -pc" to clear all port counters
4) Wait 17 seconds as per the specifications requirements.
5) Run "ibdiagnet -c 1000" to send 1000 node descriptions.
6) Run "ibdiagnet -r" to generate fabric report.
7) Run "ibchecknet" to build guid list.

13.2.4 Verification Procedures

1) Review "PM Counters" section of the fabric report. There should be no illegal PM counters. The Specification says there should be no errors in 17 seconds.
2) Review "Subnet Manager " section of the fabric report. Verify that the running SM is the one you started and verify number of nodes and switches in the fabric.
3) Review the ibchecknet report and verify that there are no duplicate GUIDs in the fabric

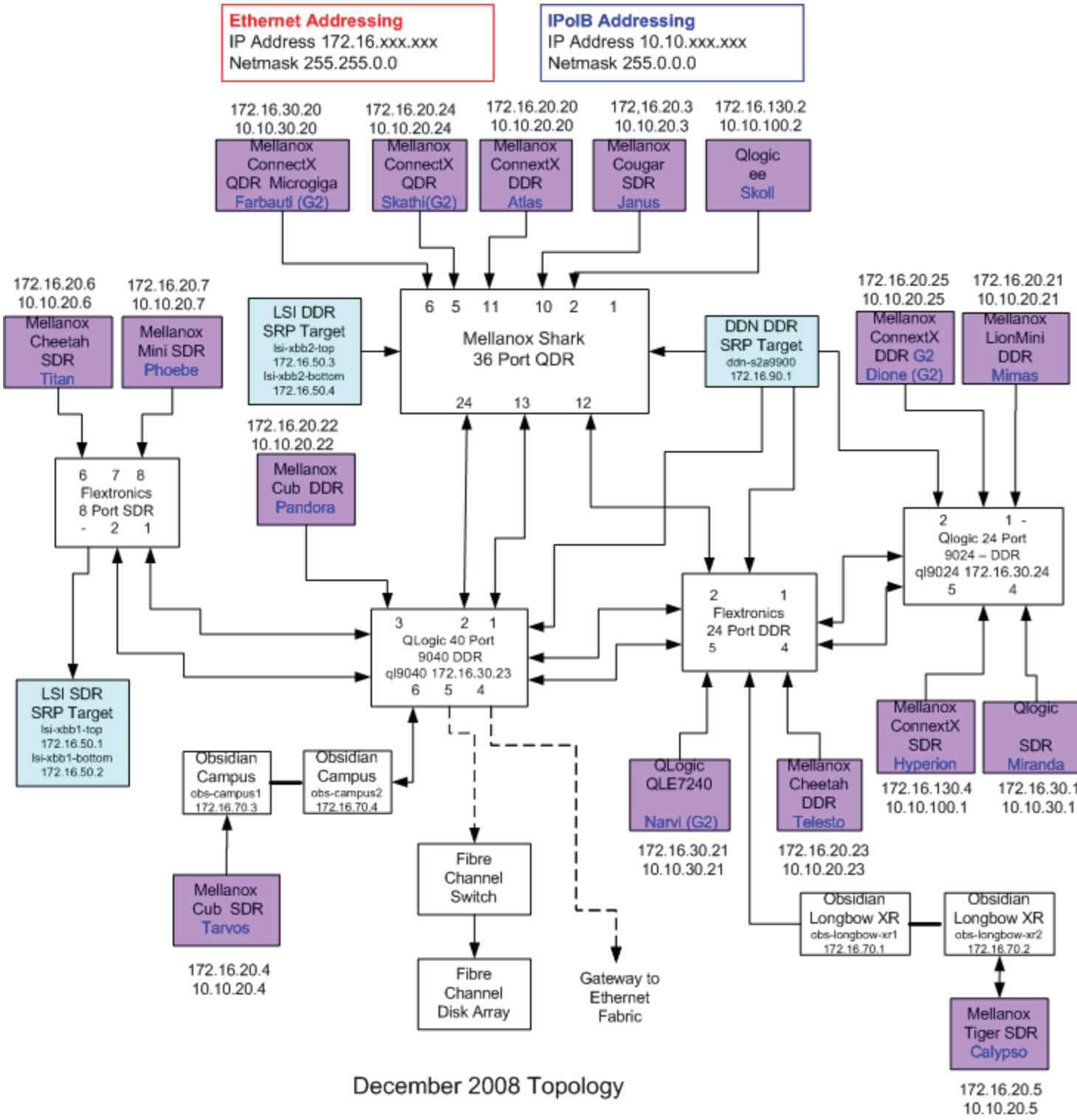
Restart all devices in the fabric and follow Sections 13.2.3 and 13.2.4. Run the SM from a different device in the fabric until all SMs present have been used. All SMs on managed switches and one instance of **opensm** must be used.

Each device must pass all verification procedures with every SM to pass Fabric Initialization test.

Table 31 - ibdiagnet commands

Commands	Description
ibdiagnet -h	Help
Ibdiagnet - pc	Clear Counter
Ibdiagnet -lw 4x - ls 2.5	Specify link width and speed
Ibdiagnet -c 1000	send 1000 Node Descriptions

Figure 6 - Sample Network Configuration



13.3 IB IPoIB Datagram Mode (DM) using WinOF

13.3.1 Setup

Note: WinOF 2.0.2 only supports IPoIB Datagram Mode. Future WinOF releases will support IPoIB Connected-Mode.

Connect the HCAs and switches as per the Architected Network and make sure that no SM is running on the Fabric.

This procedure, as the previous ones, will be based on the cluster connectivity. An SM/SA which supports IPoIB (sufficient IB multicast support) will be running on the HCAs, or on a switch with an embedded SM/SA or a third HCA which would only run SM/SA for the partner pair (with a switch in the middle). This procedure has been developed for the Windows environment.

Optional: In the procedures below, an IB analyzer can be inserted in the appropriate link to obtain traces and validate the aspects of the procedures specifically detailed below in subsequent sections.

13.3.2 IPoIB Interface Creation and IPoIB Subnet Creation

1) Configure IPoIB address. All addresses must reside on the same subnet.
2) Verify which 'Local Area Connection' the IPoIB interfaces are bound to:
   a) Start | Server Manager | View Network Connections.
   b) Find the OpenFabrics IPoIB interfaces (one per HCA port). If your platform has two Ethernet ports, then IPoIB interfaces likely will be assigned '**Local Area Connection 3**' & '**Local Area Connection 4**' as the Ethernet ports are assigned '**Local Area Connection**' and '**Local Area Connection 2**' .
3) Set interfaces to 10.0.0.x/24 (10.0.0.x/netmask 255.255.255.0) using the following commands:
   a) netsh interface ip set address "Local Area Connection 3" static 10.10.4.x 255.255.255.0
   b) netsh interface ip set address "Local Area Connection 4" static 10.10.4.y 255.255.255.0
4) View the IPoIB IP address using the following command
   a) netsh interface ip show address "Local Area Connection 3"

13.3.3 Ping Procedures

Step A

1) Stop all SM's and verify that none are running
2) Power cycle all switches in the fabric (this insures that the new SM will configure all the links and create the multi-cast join).
3) Start an SM (All SM's will need to be tested) and let it initialize

   Note: For link testing it is recommended to use an OFED Linux OpenSM as the Windows version of OpenSM does not support all SA queries and functionality of the OFED 1.4 OpenSM.

Note: All WinOF installed systems contain a disabled OpenSM windows service. A WinOF installation option/feature is to automatically 'start/enable' the OpenSM service on the local node.

- Start | Server Manager | Configuration | Services | InfiniBand Subnet Manager | Automatic | apply
- Start | Apply will enable the local OpenSM to start and be started upon system boot.

a) Visually verify that all devices are in the active state. Orange led will be on if the port is active.

b) From a Linux system, Run "ibdiagnet -r" and verify that the SM you started is the one that is running and and that it is the master. You will need to know the GUID of the device since the SM will be reassigned on each reboot; the Windows 'vstat' command displays HCA info.

c) Verify that all nodes and switches were discovered.

d) WinOF 2.0.2 does not provide a ibdiagnet utility.

Note: Ibdiagnet may show more switches than indicated by the physical number of switch platforms present. This is because some switches have multiple switch chips.

4) Examine the arp table (via arp -a) and remove the destination node's ib0 address from the sending node's arp table (via arp -d).

5) Ping every IPoIB interface IPv4 address except localhost with packet sizes of 64, 256, 511, 512, 1024, 1025, 2044, 4096, 8192, 16384, 32768, and 65507. 'ping /?' displays ping help.

a) 100 packets of each size will be sent

b) Every packet size is a new ping command.

6) In order to pass Step A, a reply must be received for every ping sent (without losing a single packet) while using each one of the SMs available in the cluster.

Step B

1) Bring up all HCAs but one.

2) Start an SM (all SMs will need to be tested).

3) Check for ping response between all node (All to All).

a) A response from the disconnected HCA should not be returned.

4) Disconnect one more HCA from the cluster.

5) Ping to the newly disconnected HCA from all nodes (No response should be returned).

6) Connect the first machine (the one that was not connected) and check for ping response from all nodes that are still connected.

7) Connect the disconnected HCA to a different switch on the subnet which will change the topology.

8) Ping again from all nodes (this time we should get a response).

9) Follow Step B, this time bring the interface down and then back up: Start | Server Manager | View Network Connections | IPoIB(Local Area connection) disable and enable commands instead of physically disconnecting the HCAs.

   Note: Each step must exhibit the expected behavior while using each SM in order for the device to pass Step B overall.

Step C

Follow Step A and B using a different SM until all SM's have been used. Only one instance of each available SM is required. Steps A, B, and C must pass in order for the device to pass 13.3.3 overall.

13.3.4 FTP PROCEDURE

FTP procedures requires an FTP server to be configured on each machine in the partner pair. An FTP client needs to be available on each machine as well; an FTP client is a standard Windows component.

An FTP server is a component of the IIS '**Internet Information Services**' manger which **not** a part of a standard Windows installation:

See Start | Server Manager | Roles | Add IIS. Configure FTP server via IIS manager.

13.3.4.1 SETUP

1) Make sure ftpd is installed on each node for the FTP application.
2) A special account for this should be created as follows:
   b) Username: Interop
   c) Password: openfabrics

13.3.4.2 PROCEDURE

Run FTP server on all nodes.

1) Start an SM (all SMs will need to be tested) and let it initialize (ref MS Network utilities docs)
   a) Verify that the running SM is the one you started.
2) FTP:
   a) Connect an HCA pair via FTP on IPoIB using the specified user name and password.
   b) Put the 4MB file to the %windir%\temp folder (generally C:\Windows\Temp) on the remote host.
   c) Get the same file to your local dir again.
   d) Binary compare the file using the Windows command 'fc /B tfile tfile.orig'.
      i) The two must be identical

3) Repeat the procedure with a different SM.

Note: Every node must FTP the 4MB file to all others using all SMs and the files must be identical as determined by the binary compare in order for the device to pass 13.3.4 overall.

Note: Sections 13.3.3 and 13.3.4 must pass using the configuration determined by sections 13.3.1 and 13.3.2 for the device to pass IPoIB Datagram mode overall.

13.4 IB SM Failover and Handover Procedure using WinOF

13.4.1 Setup

1) Connect HCAs per the selected topology.
2) In this test, all active SMs on the fabric which are going to be tested, must be from the same vendor. They will be tested pairwise: two at a time.

13.4.2 Procedure

1) Disable all SMs in the cluster.
2) Start a SM on either machine in a chosen pair.
   a) Start | Server Manager | Configuration | Services | InfiniBand Subnet Manager | start | apply
3) Run "vstat" on all Windows nodes in the fabric.
   a) Verify HCA link active in vstat output.
4) Verify IPoIB is active on each node
   a) Verify Local Area Connection assigned to IPoIB interface:
      i) Start | Control Panel | Network and Sharing Center | Manage Network Connections.
   b) Show IPv4 address assigned to IPoIB Interface(s):
      i) netsh interface ip show address "Local Area Connection 3"
      ii) netsh interface ip show address "Local Area Connection 4"
   c) Verify the IPoIB devices (one per cabled connected HCA port) are visible & operational from a device driver perspective using Device Manager
      i) Start | Run | devmgmt.msc
   d) Ping the IPoIB interface IPv4 address local and remote, verify traffic is actually going in/out over IPoIB 'local area connection x'.
5) Start an Open SM on the second machine in the current pair.
6) Verify that the SMs behave according to the SM priority rules.
   a) The Windows OpenSM log file is located at '%windir%\temp\osm.log'.

      Note: The SM with highest numerical priority value is master and the other is in standby.If both SMs have the same priority value then the SM with the smallest guid is master and the other is in standby.
7) Verify that all nodes in the cluster are present - ping all IPoIB interfaces
8) Shutdown the master SM.
9) Verify the other active SM goes into the master state: see osm.log file.
10) Verify that all nodes in the cluster are present - ping all IPoIB interfaces
11) Start the SM you just shutdown.
12) Verify that the newly started SM resumes it's position as master while the other goes into standby again; see '%windir%\temp\osm.log'.
13) Verify that all nodes in the cluster are present - ping all IPoIB interfaces

14) Shutdown the standby SM.
15) Verify that the previous master SM is still the master; view '%windir%\temp\osm.log'.
16) Verify that all nodes in the cluster are present - ping all IPoIB interfaces
17) Repeat proceeding steps [1-16] 2 more times with the same node pair, ensuring that the below criteria is met (total of 3 tests per pair which can be run in any order):
    a) First SM to be started having highest numerical priority value.
    b) Second SM to be started having highest numerical priority value.
    c) Both SMs having equal numerical priority values.
18) Repeat steps 1-17 until all possible SM pairs from identical vendors in the cluster have been tested.

13.5 IB SRP using WinOF

13.5.1 Setup

1) Connect the HCAs and switches as per the Architected Network and make sure that no SM is running on the Fabric.
2) Configure and Start a Linux OFED SRP target - VDISK BLOCKIO mode; (some assembly required) - https://wiki.openfabrics.org/tiki-index.php?page=SRPT+Installation
   a) assume /dev/sdb1 & /dev/sdc1 are formatted with /sbin/mkfs.msdos
   b) Setting SRPT_LOAD=yes in /etc/infiniband/openib.conf is not good enough. It only loads ib_srpt module and does not load scst and its dev_handlers.
   c) modprobe scst
   d) modprobe scst_vdisk
   e) echo "open vdisk0 /dev/sdb BLOCKIO" > /proc/scsi_tgt/vdisk/vdisk
   f) echo "open vdisk1 /dev/sdc BLOCKIO" > /proc/scsi_tgt/vdisk/vdisk
   g) echo "add vdisk0 0" >/proc/scsi_tgt/groups/Default/devices
   h) echo "add vdisk1 1" >/proc/scsi_tgt/groups/Default/devices

13.5.2 Windows Procedure

1) Start an SM (all SM's will need to be tested) and let it initialize
   a) Verify that the running SM is the one that you started
2) Choose a node to work with
3) Verify the SRP driver loaded correctly; locate the SRP Miniport.
   a) Start |Control Panel | Device Manager | Storage Controllers [InfiniBand SRP Miniport]
4) Discover + Enable (bring online) the SRP drive(s)
   a) Start | Server Manager | Storage | Disk Management
5) You will find a basic ‘unknown’ and ‘offline’ disk; this one of your SRP volume(s).
6) Right-click the offline disk and select ‘online’.
7) Right-click the volume space, assign the drive letter ‘T’.
8) Right-click the volume space, format the volume.
9) Access the SRP drive via assigned drive letter. From a Windows/DOS command prompt window, execute the following commands.
   a) vol T:
   b) dir T:\ (should be empty)
   c) mkdir T:\tmp
   d) copy /B WinOF_wlh_x64.msi T:\tmp
   e) fc /B WinOF_wlh_x64.msi T:\tmp\WinOF_wlh_x64.msi

f) copy /B T:\tmp\WinOF_wlh_x64.msi T:\tmp\WOF2.msi
g) fc /B T:\tmp\WinOF_wlh_x64.msi T:\tmp\WOF2.msi
h) fc /B WinOF_wlh_x64.msi T:\tmp\WOF2.msi
i) copy /B T:\tmp\WOF2.msi WOF3.msi
j) fc /B WinOF_wlh_x64.msi WOF3.msi
k) del T:\tmp\WOF2.msi
l) del T:\tmp\WinOF_wlh_x64.msi
m) dir T:\tmp (should be empty)
n) rmdir T:\tmp
o) dir T:\ (should be empty)
p) del WOF3.msi

10) For each SRP target located in Procedure #4
    a) Repeat step 9 for all volumes found for all targets as determined by Windows Procedure step #4 - see Discover + Enable (bring online) the SRP drive(s)
11) Take SRP drive offline
    a) Start | Server Manager | Storage | Disk Management
    b) Right-click the online disk and select ‘offline’
    c) dir T:\ (should fail).
12) Reboot all devices in the fabric and repeat the procedure using a different SM.

Note: An HCA must successfully complete all operations to and from all volumes on all targets using all available SM's in order to pass SRP testing. One volume per target is all that is required.

13.6 IB uDAPLTEST Commands using WinOF

Server Command: dapl2test -T S -D <ia_name>

13.6.1 IB Setup

- The %SystemDrive%\DAT\dat.conf needs to be verified to be sure that the correct interface is used. The DAPL interface for IB is ibnic0v2.
- It is also important to verify that the desired dat/dapl libraries are available
  - %windir%\dat2.dll
  - %windir%\dapl2.dll
- To run dapl2test on IB, an SM needs to be running.

13.6.2 Group 1: Point-to-Point Topology

[1.3] 1 connection and simple send/recv:

- dapl2test -T T -s <server_name> -D <ia_name> -i 100 -t 1 -w 1 -R BE
- client SR 256 1 server SR 256 1

[1.4] Verification, polling, and scatter gather list:

- dapl2test -T T -s <sever_name> -D <ia_name> -i 100 -t 1 -w 1 -V -P -R BE
- client SR 1024 3 -f \
- server SR 1536 2 -f

13.6.3 Group 2: Switched Topology

InfiniBand Switch: Any InfiniBand switch

[2.5] Verification and private data:

- dapl2test -T T -s <server_name> -D <ia_name> -i 100 -t 1 -w 1 -V -P -R BE
- client SR 1024 1 \
- server SR 1024 1

[2.6] Add multiple endpoints, polling, and scatter gather list:

- dapl2test -T T -s <server_name> -D <ia_name> -i 100 -t 1 -w 10 -V -P -R
- BE client SR 1024 3 \
- server SR 1536 2

[2.7] Add RDMA Write :

- dapl2test -T T -s <server_name> -D <ia_name> -i 100 -t 1 -w 1 -V -P -R BE
- client SR 256 1 \
- server RW 4096 1 server SR 256 1

[2.8] Add RDMA Read:

- dapl2test -T T -s <server_name> -D <ia_name> -i 100 -t 1 -w 1 -V -P -R BE
- client SR 256 1 \

- server RR 4096 1 server SR 256 1

13.6.4 GROUP 3: SWITCHED TOPOLOGY WITH MULTIPLE SWITCHES

[3.5] Multiple threads, RDMA Read, and RDMA Write:

- dapl2test -T T -s <server_name> -D <ia_name> -i 100 -t 4 -w 8 -V -P -R BE
- client SR 256 1 \
- server RR 4096 1 server SR 256 1 client SR 256 1 server RR 4096 1 \
- server SR 256 1

[3.6] Pipeline test with RDMA Write and scatter gather list:

- dapl2test -T P -s <server_name> -D <ia_name> -i 1024 -p 64 -m p RW 8192 2

[3.7] Pipeline with RDMA Read:

- dapl2test -T P -s <server_name> -D <ia_name> -i 1024 -p 64 -m p RR 4096 2

[3.8] Multiple switches:

- dapl2test -T T -s <server_name> -D <ia_name> -i 100 -t 1 -w 10 -V -P -R
- BE client SR 1024 3 \
- server SR 1536 2

13.6.5 WINOF DAPL2TEST WRAPPER SCRIPTS

All the specified DAPL tests are conveniently located in the WinOF distributed DAPL test server & client scripts.

- %ProgramFiles(x86)%\WinOF\dt-svr.bat
  - To run the dapl2test Server, to a Windows cmd-prompt window type ‘dt-svr’. Only one server is necessary – multiple clients can communicate with a single dapl2test server; multiple servers on different nodes can exist. A single dapl2test client communicates with only one dapl2test server at a time.
  - No further server action is required as the dapl2test server is persistent; looping waiting for dapltest client requests.
- %ProgramFiles(x86)%\WinOF\dt-cli.bat
  - ‘dt-cli’ no arguments, will display dt-cli command args & options.
  - Dapl2test client invocation: ‘dt-cli IPoIB_IPv4_server_address cmd’
  - If the dt-svr command was executed on a system where the IPoIB interface address is 10.10.4.200 then
  - ‘dt-cli 10.10.4.200 interop’ would run the above dap2tests between the client and server.
  - ‘dt-cli 10.10.4.200 conn’ is a simple, quick test to verity dapl2test client | server connection is operational.

13.7 IB MPI - Intel MPI using WinOF

13.7.1 Requirements

1) Intel MPI is not part of the WinOF installation; acquire Intel MPI installer file from Intel.
2) Install same O/S version (Windows Server 2008-HPC) on homogenous x86_64 systems.
3) MPI testing requires a reliable IB fabric without other fabric interop testing occurring.
4) Private Ethernet Network configuration
   a) DNS names must match hostnames in hosts file.
5) WinOF Installation requirements
   a) Install the latest version of WinOF on all systems (double-click WinOF_wlh_x64.msi); see
      i) http://www.openfabrics.org/downloads/WinOF/README
      ii) Select the 'default' set of install features; includes uDAPL.
      iii) Run OpenSM either on the headnode OR from one of the IB switches.
      iv) If OpenSM on the headnode, select WinOF install feature 'OpenSM Started'.
   b) Once WinOF installation on all nodes has completed, configure IPoIB interfaces.
      i) %windir%\system32\Drivers\etc\hosts should be setup with IB hostnames and static IP addresses.
      ii) Assign IPv4 address, from hosts file, to each IPoIB interface; Example: Local Area Connection 3 is the 1st IPoIB interface.
         - netsh interface ip set address "Local Area Connection 4" static 10.10.4.y 255.255.255.0

           This allows you to **set** the IPoIB IP address.
         - netsh interface ip show address "Local Area Connection 3"

           This allows you to **view** the IPoIB IP address.
      iii) Verify by pinging IPoIB interface addresses on all nodes.

13.7.2 Setup information for Intel MPI

Install Intel MPI on every cluster node:

1) Intel MPI runtime environment kit
   a) http://www.intel.com/cd/software/products/asmo-na/eng/308295.htm
2) Intel MPI Benchmarks ,
   a) http://www.intel.com/cd/software/products/asmo-na/eng/cluster/mpi/219848.htm
3) Add identical user account (%SystemDrive%\users\test) on every node.

4) Headnode mount points (%SystemDrive%\test\export) on user accounts.

13.7.3 Additional Information

1) Go to the individual test directories and follow the steps in the respective README-*.txt files.
2) For Intel MPI Support Services go to:
   a) http://software.intel.com/en-us/articles/intel-mpi-library-for-windows/all/1/
   b) See Intel MPI Reference Manual for Additional information

13.7.4 Intel MPI (MVAPICH 2) - Test Procedure

1) Run a subnet manager from one node only.
2) Run Intel® MPI Benchmarks from the HPC head-node:
   a) Two sets of tests should be run, with these command lines
      - mpirun -np <number of nodes X number of processors/node> IMB-MPI1 -multi 0 PingPong PingPing
      - mpirun -np <number of nodes X number of processors/node> IMB-MPI1

      The first command runs just the PingPong and PingPing point-to-point tests, but makes all tasks active (pairwise).

      The second command runs all the tests (PingPong, PingPing, Sendrecv, Exchange, Bcast, Allgather, Allgatherv, Alltoall, Reduce, Reduce_scatter, Allreduce, Barrier), in non-multi mode.
   b) If the test passes shutdown current subnet manager and start another one on a different node; run both tests again.
3) Repeat until all nodes have run a subnet manager and passed all tests.

13.7.5 Interpreting the results

1) TBA

14 Bug reporting methodology during pre-testing

The following bug reporting methodology will be followed during the execution of interoperability pre-testing at UNH-IOL.

1) UNH-IOL and the OEMs (e.g. Chelsio, Data Direct, Flextronics, Intel, LSI Logic, Mellanox, NetEffect, Obsidian, QLogic and Voltaire) will assign a focal point of contact to enable fast resolution of problems.
2) Bug reports will include:
   a) Detailed fail report with all relevant detail (Test/Application, Topology.).
   b) [For IB] IB trace if needed.
   c) [For iWARP] iWARP, TCP and SCTP traces if needed.
3) Bug reports will be sent via mail by UNH-IOL to the focal point assigned by the switch OEM
4) Bug reports and suggested fixes will be sent to the OpenFabrics development community - OFA Bugzilla. When such reports are communicated, UNH-IOL will ensure that confidentiality between UNH-IOL and the OEM will be maintained. Bug reports will be generalized and not include any company specific proprietary information such as product name, software name, version etc.
5) All bug fixes/issues that we will found during testing will be uploaded to the OpenFabrics repository. Documentation related to fixes will not mention any company specific proprietary information.

Note: This test plan does not cover how bugs will be reported by IBTA/CIWG or IETF iWARP during or after interoperability testing at plugfests.

15 Results Summary

15.1 InfiniBand Specific Test Results

Please add a check mark whenever a test case passes and when the system is behaving according to the criteria mentioned below. Otherwise indicate a failure along with a comment explaining the nature of the failure.

Results Table 1 - IB Link Initialize

Test #	Test	Pass	Fail	Comment
1	Phy link up all ports			
2	Logical link up all ports switch SM			
3	Logical link up all ports HCA SM			

Results Table 2 - IB Fabric Initialization

Test #	Test	Pass	Fail	Comment
1	Verify that all ports are in Armed or Active state			

Results Table 3 - IB IPoIB - Connected Mode (CM)

Test #	Test	Pass	Fail	Comment
1	Ping all to all - Ping using SM 1			
2	Ping all to all - Ping using SM 2			
3	Ping all to all - Ping using SM 3			
4	Ping all to all - Ping using SM 4			
5	Ping all to all - Ping using SM 5			
6	Ping all to all - Ping using SM 6			
7	Ping all to all - Ping using SM x			
8	Connect/Disconnect Host			
9	FTP Procedure			

Results Table 4 - IB IPoIB - Datagram Mode (DM)

Test #	Test	Pass	Fail	Comment
1	Ping all to all - Ping using SM 1			
2	Ping all to all - Ping using SM 2			
3	Ping all to all - Ping using SM 3			
4	Ping all to all - Ping using SM 4			
5	Ping all to all - Ping using SM 5			
6	Ping all to all - Ping using SM 6			
7	Ping all to all - Ping using SM x			
8	Connect/Disconnect Host			
9	FTP Procedure			

Table 5 - IB SM Failover/Handover

Test #	Test	Pass	Fail	Comment
1	Basic sweep test			
2	SM Priority test			
3	Failover test - Disable SM1			
4	Failover test - Disable SM2			

Results Table 6 - IB SRP

Test #	Test	Pass	Fail	Comment
1	Basic dd application			
2	IB SM kill			
3	Disconnect Initiator			
4	Disconnect Target			

Results Table 7 - Fibre Channel Gateway - (IB Specific)

Test #	Test	Pass	Fail	Comment
1	Basic Setup			
2	Configure Gateway			
3	Add Storage Device			
4	Basic dd application			
5	IB SM kill			
6	Disconnect Host/Target			
7	Load Host/Target			
8	dd after SRP Host and Target reloaded			
9	Reboot Gateway			
10	dd after FC Gateway reboot			

Results Table 8 - Ethernet Gateway - (IB Specific)

Test #	Test	Pass	Fail	Comment
1	Basic Setup			
2	Start ULP			
3	Discover Gateway			
4	SM Failover			
5	Ethernet gateway reboot			
6	ULP restart			
7	Unload/load ULP			

15.2 ETHERNET SPECIFIC TEST RESULTS

Results Table 9 - Ethernet Link Initialize

Test #	Test	Pass	Fail	Comment
1	Phy link up all ports			
2	Verify basic IP connectivity			

Results Table 10 - Ethernet Fabric Initialize

Test #	Test	Pass	Fail	Comment
1	Fabric Initialization			

Results Table 11 - Ethernet Fabric Reconvergence

Test #	Test	Pass	Fail	Comment
1	Fabric Reconvergence			

Results Table 12 - Ethernet Fabric Failover

Test #	Test	Pass	Fail	Comment
1	Fabric Failover			

Results Table 13 - iWARP Connectivity

Test #	Test	Pass	Fail	Comment
1	Group 1 - Verify that each single iWARP operation over single connection works			
2	Group 2 - Verify that multiple iWARP operations over a single connection work			
3	Group 3 - Verify that multiple iWARP connections work			
4	Group 4 - Verify that disconnect/reconnect physical connections work			
5	Group 5 - Verify that IP Speed negotiation work			
6	Group 6 - Verify that iWARP error ratio work			

Results Table 13 - iWARP Connectivity

Test #	Test	Pass	Fail	Comment
7	Group 7 - Verify that stress pattern over iWARP work			
8	Group 8 - Verify that iWARP parameter negotiation work			

15.3 TRANSPORT INDEPENDENT TEST RESULTS

Results Table 14 - TI iSER

Test #	Test	Pass	Fail	Comment
1	Basic dd application			
2	IB SM kill			
3	Disconnect Initiator			
4	Disconnect Target			
5	Repeat with previous SM Slave			

Results Table 15 - TI NFS Over RDMA

Test #	Test	Pass	Fail	Comment
1	File and directory creation			
2	File and directory removal			
3	Lookups across mount point			
4	Setattr, getattr, and lookup			
5	Read and write			
6	Readdir			
7	Link and rename			
8	Symlink and readlink			
9	Statfs			

Results Table 16 - TI RDS

Test #	Test	Pass	Fail	Comment
1	rds-ping procedure			
2	rds-stress procedure			

Results Table 17 - TI SDP

Test #	Test	Pass	Fail	Comment
1	netperf procedure			
2	FTP Procedure			
3	IB SCP Procedure			
4	iWARP SCP Procedure			

Results Table 18 - TI uDAPL

Test #	Test	Pass	Fail	Comment
1	P2P - Connection & simple send receive			
2	P2P - Verification, polling & scatter gather list			
3	Switched Topology -Verification and private data			
4	Switched Topology - Add multiple endpoints, polling, & scatter gather list			
5	Switched Topology - Add RDMA Write			
6	Switched Topology - Add RDMA Read			
7	Multiple Switches - Multiple threads, RDMA Read, & RDMA Write			
8	Multiple Switches - Pipeline test with RDMA Write & scatter gather list			
9	Multiple Switches - Pipeline with RDMA Read			
10	Multiple Switches - Multiple switches			

Results Table 19 - TI Basic RDMA Interop

Test #	Test	Pass	Fail	Comment
1	Small RDMA READ			
2	Large RDMA READ			
3	Small RDMA Write			
4	Large RDMA Write			

Results Table 19 - TI Basic RDMA Interop

Test #	Test	Pass	Fail	Comment
5	Small RDMA SEND			
6	Large RDMA SEND			
7	Small RDMA Verify			
8	Large RDMA Verify			

Results Table 20 - TI RDMA operations over Interconnect Components

Test #	Test	Pass	Fail	Comment
1	Switch Load			
2	Switch Fan In			

15.4 HP-MPI TEST RESULTS

Results Table 21 - TI MPI - HP-MPI - (Not part of OFA Stack)

Test #	Test Suite	Pass		Comment
1	IMB			
2	rings2			
3	fork			
4	exitpath			
5	alltoone			

15.5 Intel MPI Test Results

Results Table 22a - Intel MPI Benchmark Summary

Test #	Test Suite	Pass	Fail	Comment
1	Test 1: PingPong			
2	Test 1: PingPing			
3	Test 1: Sendrecv			
4	Test 1: Exchange			
5	Test 1: Allreduce			
6	Test 1: Reduce			
7	Test 1: Reduce_scatter			
8	Test 1: Allgather			
9	Test 1: Allgatherv			
10	Test 1: Alltoall			
11	Test 1: Alltoallv			
12	Test 1: Bcast			
13	Test 1: Barrier			

Results Table 22b - TI MPI - Intel MPICH2 (Not part of OFA stack) Pass/Fail Summary

Test #	Test Suite	Pass	Fail	Comment
1	attr			
2	coll			
3	comm			
4	datatype			
5	errhan			
6	group			
7	info			
8	init			
9	pt2pt			
10	rma			
11	spawn			

Results Table 22b - TI MPI - Intel MPICH2 (Not part of OFA stack) Pass/Fail Summary

Test #	Test Suite	Pass	Fail	Comment
12	topo			
13	io			
14	F77			
15	cxx			
16	threads			

Results Table 22c TI MPI - Intel MPI (Not part of OFA stack) Test Failure Details

Test #	Test Suite	Pass		Comment
1	testlist2l (1085 tests)			
2	testlist2-2l (23 tests)			
3	testlist4 (216 tests)			
4	testlist4lg (1 test)			
5	testlist6 (46 tests)			

15.6 OPEN MPI TEST RESULTS

Results Table 23 - TI MPI - Open MPI

Test #	Test Suite	Pass	Fail	Comment
Phase 1: "Short" tests				
2	OMPI built with OpenFabrics support			
3	OMPI basic functionality (hostname)			
4.1	Simple MPI functionality (hello_c)			
4.2	Simple MPI functionality (ring_c)			
5	Point-to-point benchmark (NetPIPE)			
6.1.1	Point-to-point benchmark (IMB PingPong multi)			
6.1.2	Point-to-point benchmark (IMB PingPing multi)			
Phase 2: "Long" tests				
6.2.1	Point-to-point benchmark (IMB PingPong)			
6.2.2	Point-to-point benchmark (IMB PingPing)			
6.2.3	Point-to-point benchmark (IMB Sendrecv)			
6.2.4	Point-to-point benchmark (IMB Exchange)			
6.2.5	Collective benchmark (IMB Bcast)			
6.2.6	Collective benchmark (IMB Allgather)			
6.2.7	Collective benchmark (IMB Allgatherv)			
6.2.8	Collective benchmark (IMB Alltoall)			
6.2.9	Collective benchmark (IMB Reduce)			
6.2.10	Collective benchmark (IMB Reduce_scatter)			
6.2.11	Collective benchmark (IMB Allreduce)			
6.2.12	Collective benchmark (IMB Barrier)			
6.3.1	I/O benchmark (IMB S_Write_Indv)			
6.3.2	I/O benchmark (IMB S_IWrite_Indv)			
6.3.3	I/O benchmark (IMB S_Write_Expl)			
6.3.4	I/O benchmark (IMB S_IWrite_Expl)			
6.3.5	I/O benchmark (IMB P_Write_Indv)			
6.3.6	I/O benchmark (IMB P_IWrite_Indv)			

Results Table 23 - TI MPI - Open MPI

Test #	Test Suite	Pass	Fail	Comment
6.3.7	I/O benchmark (IMB P_Write_Shared)			
6.3.8	I/O benchmark (IMB P_IWrite_Shared)			
6.3.9	I/O benchmark (IMB P_Write_Priv)			
6.3.10	I/O benchmark (IMB P_IWrite_Priv)			
6.3.11	I/O benchmark (IMB P_Write_Expl)			
6.3.12	I/O benchmark (IMB P_IWrite_Expl)			
6.3.13	I/O benchmark (IMB C_Write_Indv)			
6.3.14	I/O benchmark (IMB C_IWrite_Indv)			
6.3.15	I/O benchmark (IMB C_Write_Shared)			
6.3.16	I/O benchmark (IMB C_IWrite_Shared)			
6.3.17	I/O benchmark (IMB C_Write_Expl)			
6.3.18	I/O benchmark (IMB C_IWrite_Expl)			
6.3.19	I/O benchmark (IMB S_Read_Indv)			
6.3.20	I/O benchmark (IMB S_IRead_Indv)			
6.3.21	I/O benchmark (IMB S_Read_Expl)			
6.3.22	I/O benchmark (IMB S_IRead_Expl)			
6.3.23	I/O benchmark (IMB P_Read_Indv)			
6.3.24	I/O benchmark (IMB P_IRead_Indv)			
6.3.25	I/O benchmark (IMB P_Read_Shared)			
6.3.26	I/O benchmark (IMB P_IRead_Shared)			
6.3.27	I/O benchmark (IMB P_Read_Priv)			
6.3.28	I/O benchmark (IMB P_IRead_Priv)			
6.3.29	I/O benchmark (IMB P_Read_Expl)			
6.3.30	I/O benchmark (IMB P_IRead_Expl)			
6.3.31	I/O benchmark (IMB C_Read_Indv)			
6.3.32	I/O benchmark (IMB C_IRead_Indv)			
6.3.33	I/O benchmark (IMB C_Read_Shared)			
6.3.34	I/O benchmark (IMB C_IRead_Shared)			
6.3.35	I/O benchmark (IMB C_Read_Expl)			
6.3.36	I/O benchmark (IMB C_IRead_Expl)			

Results Table 23 - TI MPI - Open MPI

Test #	Test Suite	Pass	Fail	Comment
6.3.37	I/O benchmark (IMB Open_Close)			

15.7 OSU MPI Test Results

Results Table 24 - TI MPI - OSU

Test #	Test	Pass	Fail	Comment
1	**Test** 1: PingPong			
2	**Test** 1: PingPing point-to-point			
3	**Test** 2: PingPong			
4	**Test** 2: PingPing			
5	**Test** 2: Sendrecv			
6	**Test** 2: Exchange			
7	**Test** 2: Bcast			
8	**Test** 2: Allgather			
9	**Test** 2: Allgatherv			
10	**Test** 2: Alltoall			
11	**Test** 2: Alltoallv			
12	**Test** 2: Reduce			
13	**Test** 2: Reduce_scatter			
14	**Test** 2: Allreduce			
15	**Test** 2: Barrier			

Results Table 25 Remarks

General Remarks: Comments about the set-up, required updates to the TD, and any other issues that came up during the testing.