

# **OFA Interoperability Working Group**

## **OFA-IWG Interoperability Test Plan Release 1.40**



October 25, 2011  
DRAFT

Copyright © 2011 by OpenFabrics - All rights reserved.

This document contains information proprietary to OpenFabrics. Use or disclosure without written permission from an officer of the OpenFabrics is prohibited.

[OpenFabrics.org](http://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	<ul style="list-style-type: none"> <li>• 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.</li> <li>• Added new test section from Bob Jaworski for Fibre Channel Gateway.</li> </ul>
1.03	Dec 10, 2006	<ul style="list-style-type: none"> <li>• Updated test procedures based on the October 2006 OFA Interop Event.</li> <li>• Updated Fibre Channel Gateway test based on changes submitted by Karun Sharma (QLogic).</li> <li>• Added Ethernet Gateway test written by Karun Sharma (QLogic).</li> </ul>
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	<ul style="list-style-type: none"> <li>Updated with the latest scripts developed by UNH IOL and based on the results from the April 2007 Interop Event</li> </ul>
1.12	Sep 12, 2007	<ul style="list-style-type: none"> <li>Updated the documents to embed the test scripts in the document.</li> </ul>
1.13	Jan 22, 2008	<ul style="list-style-type: none"> <li>Updated the documents for the March 2008 OFA Interop event. IPoIB updated along with Cover Page and the Test Requirements section.</li> </ul>
1.14	Feb 11, 2008	<ul style="list-style-type: none"> <li>Added the following tests: <ul style="list-style-type: none"> <li>1. Ethernet Switch Tests</li> <li>2. IPoIB Connected Mode</li> <li>3. RDMA Interop</li> <li>4. RDS</li> </ul> </li> </ul>
1.15	Feb 18, 2008	<ul style="list-style-type: none"> <li>Updates to the following tests: <ul style="list-style-type: none"> <li>1. Ethernet Switch Tests</li> <li>2. IPoIB Connected Mode</li> <li>3. RDMA Interop</li> </ul> </li> </ul>
1.16	Feb 25, 2008	<ul style="list-style-type: none"> <li>Removed all reference to Low Latency Ethernet Switches. This is the version for the March 2008 Interop Event</li> </ul>
1.17	March 3, 2008	<ul style="list-style-type: none"> <li>Added HP-MPI</li> </ul>
1.18	July 22, 2008	<ul style="list-style-type: none"> <li>Updated HP-MPI based on results from the March 2008 Interop Event</li> </ul>
1.19	July 28, 2008	<ul style="list-style-type: none"> <li>Updated HP-MPI URL for the tests.</li> <li>Added section for Open MPI</li> <li>Updated MPI based on feedback from UNH IOL</li> </ul>
1.20	July 30, 2008	<ul style="list-style-type: none"> <li>Updated section for Open MPI and added tables</li> <li>Updated IB SM Failover as per Nick Wood</li> </ul>
1.21	Aug 1, 2008	<ul style="list-style-type: none"> <li>Updated SRP call <code>srp_daemon -o -e -n</code></li> <li>Updated IB SM Failover as Bob Jaworski</li> <li>Updated HP-MPI</li> <li>Updated Intel MPI</li> <li>Updated Open MPI</li> </ul>
1.22	Aug 29, 2008	<ul style="list-style-type: none"> <li>Added a section for MVAICH 1 under OSU MPI</li> </ul>
1.23	Feb 16, 2009	<ul style="list-style-type: none"> <li>Updated link init, fabric init, <code>srp, sdp, ipoib cm, ipoib dm</code> based on updates received from UNH-IOL</li> </ul>

Revision	Release Date	
1.24	Feb 23, 2009	<ul style="list-style-type: none"> <li>Updated Intel MPI and Open MPI to reflect the fact that they are not intended to work in a heterogeneous environment.</li> <li>Updated the RDS test procedure</li> <li>Updated the Test Glossary</li> <li>Updated the Mandatory test table for April 2009</li> </ul>
1.25	Feb 24, 2009	<ul style="list-style-type: none"> <li>Updated the RDS Test after review by the OFA IWG group.</li> </ul>
1.26	Mar 13, 2009	<ul style="list-style-type: none"> <li>Restructured entire document to accommodate WinOF and OFED</li> <li>Added NFS over RDMA to the test plan.</li> <li>Added WinOF tests</li> <li>Updated HP-MPI</li> <li>Add List of Contributors</li> </ul>
1.27	Mar 17, 2009	<ul style="list-style-type: none"> <li>Updates based on the review from the OFA IWG</li> </ul>
1.28	Mar 27, 2009	<ul style="list-style-type: none"> <li>Added links in Chapter 10 to the InfiniBand Test Scripts</li> <li>Added links to HP-MPI installation Packages</li> </ul>
1.29	Aug 25, 2009	<ul style="list-style-type: none"> <li>Editorial &amp; Technical updates based on April 2009 Interop Event.</li> <li>Updated Mandatory tests for October 2009.</li> <li>Added Topology Check</li> <li>Added new Firmware Policy</li> </ul>
1.30	Sep 4, 2009	<ul style="list-style-type: none"> <li>Updated Mandatory iWARP tests and several comments based on the review from Harry Cropper</li> <li>Added changes suggested by Jess Robel from QLogic to IPoIB DM and CM and Fabric Init.</li> </ul>
1.31	April 6, 2010	<ul style="list-style-type: none"> <li>Added definition of homogenous to Test Glossary</li> <li>Added updates from the November 2009 Interop Event</li> </ul>
1.32	April 20, 2010	<ul style="list-style-type: none"> <li>Updated after the OFA IWG meeting on 4/6/2010</li> <li>Updated MPI and MVAPICH based on changes requested by Jeff Laird and Intel</li> </ul>
1.33	April 23, 2010	<ul style="list-style-type: none"> <li>Major changes to Section 8 which describes the Software and Firmware policies</li> </ul>
1.34	July 20, 2010	<ul style="list-style-type: none"> <li>Changed uDAPL for iWARP to Beta for Aug 2010 GA Event</li> <li>Removed HP MPI which is no longer supported</li> <li>Added -mca mpi_leave_pinned 0 for OpenMPI</li> <li>Add new parameters for MVAPICH2 for iWARP devices.</li> </ul>

Revision	Release Date	
1.35	July 27, 2010	<ul style="list-style-type: none"> <li>Added new parameters for MVAPICH2 for iWARP devices. The parameter is: MV2_USE_RDMA_CM=1</li> </ul>
1.36	Feb 22, 2011	<ul style="list-style-type: none"> <li>Added Link Init section as per changes provided by Chris Hutchins and approved by OFA IWG.</li> <li>Updated Test Plan Status for April 2011 and October 2011</li> <li>Nick Wood from UNH-IOL updated NFSoRDMA</li> <li>Marty requested that we update SRP Results Table 6 and remove the disconnect commands.</li> </ul>
1.37	Oct 4, 2011	<ul style="list-style-type: none"> <li>Updated Test Plan Status for November 2011</li> <li>Added new Test Table for OS and OFED versions</li> <li>Nick Wood updated Link Init for IB</li> <li>Chris Hutchins updated RDMA Interop and RDMA Stress</li> <li>Removed XANSation testing</li> </ul>
1.38	Oct 11, 2011	<ul style="list-style-type: none"> <li>Changed Link Init Section from Recommendation to MOI</li> <li>Updated Section 8 for Firmware, Software and Hardware Policies to bring in line with Logo Program Document</li> <li>Updated InfiniBand Test Table 24</li> </ul>
1.39	Oct 24, 2011	<ul style="list-style-type: none"> <li>Updated Open MPI as per changes submitted by Nick Wood</li> <li>Updated RDMA Interop small test: drop iterations from 100000 to 25000</li> <li>Updated RDMA Interop large test, increase iterations from 100 to 300</li> <li>Updated IPoIB Part A:, drop iterations (number of pings) from 100 to 10.</li> </ul>
1.40	Oct 25, 2011	<ul style="list-style-type: none"> <li>Modified the following sections</li> <li>12.6.9 iwarp client 100000 -&gt; 25000</li> <li>12.6.13</li> <li>olarge read client 65536 -&gt; 1000000</li> <li>olarge write client 65536 -&gt; 1000000</li> <li>Added large send command (section c)</li> </ul>

## List of Contributors

Editor: Rupert Dance

Name	Company
Mark Alan	HP
Harry Cropper	Intel
Rupert Dance	Software Forge
Sujal Das	Mellanox
Arlin Davis	Intel
Johann George	QLogic
Mike Hagen	UNH-IOL
Mitko Haralanov	QLogic
Allen Hubbe	UNH-IOL
Christopher Hutchins	UNH-IOL
Bob Jaworski	QLogic
Arkady Kanevsky	NetApp
Llolsten Kaonga	Software Forge
Amit Krig	Mellanox
Jeff Laird	UNH-IOL
Jon Mason	Open Grid Computing
Bob Noseworthy	UNH-IOL
Yaroslav Pekelis	Mellanox
Jess Robel	Qlogic
Hal Rosenstock	HNR Consulting
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

## **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, merchantability 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, including, without limitation, lost profits, even if advised 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 the IBTA, specifically within the charter of the CIWG and for iWARP it is the IETF.

### 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 Sections 10-12
- 2) Interoperability testing using WinOFED for Windows Platforms.
  - a) See Section 13

Sections 1.4 through 1.10 provide an overview of the tests which are described in detail in sections 10 through 13.

## 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
1	Phy link up all ports	Check that all relevant LEDs are on for all HCAs and switches.

**Table 2 - IB Fabric Initialization**

Test #	Test	Description
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
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
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
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 (

**Table 5 - IB SM Tests**

Test #	Test	Description
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
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
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
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).

**Table 8 - IB Fibre Channel Gateway**

Test #	Test	Description
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
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 IP and RDMA 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
1	Fabric Initialization	Verify IP and RDMA 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
1	Fabric Reconvergence	Verify IP and RDMA connectivity is restored to all nodes after removing a switch or a channel from a link aggregate.

**Table 12 - Ethernet Fabric Failover**

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

## 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**

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

**Table 17 - TI SDP**

Test #	Test	Description
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
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 - RDMA Basic Interop**

Test #	Test	Description
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 - RDMA Basic Interop**

Test #	Test	Description
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 Stress Tests**

Test #	Test	Description
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 INTEL MPI - TEST OVERVIEW

**Table 21a - Intel MPI Benchmark Summary**

Test #	Test	Description
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 21b - TI - Intel MPICH2 Test 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 21b - TI - Intel MPICH2 Test 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 21c - TI - Intel MPI Test Suite - (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

## 1.8 OPEN MPI - TEST OVERVIEW

**Table 22 - TI - Open MPI Test Suite Description**

Test #	Open MPI TESTs	Open MPI TESTs Suite Description
<b>Phase 1: "Short" tests</b>		
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)
<b>Phase 2: "Long" tests</b>		
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 22 - 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)

## 1.9 OSU MPI - TEST OVERVIEW

**Table 23 - TI - OSU MPI**

Test #	Test	Description
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	

## 1.10 REQUIREMENTS FOR OFA INTEROPERABILITY LOGO PROGRAM

The following table indicates the mandatory tests that will be used for Interop Validation during the November 2011 Interop Debug Event and the Interop GA Event which will occur after the release of OFED 1.5.4 GA. It is anticipated that some of the Beta tests will be moved to Mandatory status for the following Interop Event.

**Table 24 - InfiniBand Transport Test Status for November 2011 Interop Event**

Test Procedure	Linux	WinOF
IB Link Initialize	Mandatory	Mandatory
IB Fabric Initialization	Mandatory	Mandatory
IB IPoIB Connected Mode	Mandatory	Not Available - 1
IB IPoIB Datagram Mode	Mandatory	Beta
IB SM Failover/Handover - OpenSM	Mandatory	Beta
IB SM Failover/Handover - Vendor SM	Optional	Optional
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 RDMA	Beta	Not Available - 1
TI RDS	Mandatory	Not Available - 2
TI SDP	Mandatory	Not Available - 1
TI uDAPL	Mandatory	Beta
TI Basic RDMA Interop	Mandatory	Not Available - 3
TI RDMA Operations	Mandatory	Not Available - 3
TI MPI Intel	Beta	Beta
TI MPI Open MPI	Mandatory	Not Available - 2
TI MVAPICH (1) - OSU	Mandatory	Not Available - 2



**Not Available** means one of three things:

- 1) The feature is not currently supported by the WinOFED stack.
- 2) The ULP application has not been ported to the WinOFED Stack.
- 3) The test has not been updated for WinOFED.

**Optional** means that this test will not be made mandatory because it depends on proprietary vendor capabilities. The test may be run during the OFA Interop Events and reported in the results but it will not affect eligibility for the OFA Logo List.

## InfiniBand Test Plan update for November 2011

IB	Linux: Primary Platform	Linux: Secondary OS	Linux: Tertiary OS	Linux: Secondary OFED	Windows: Primary Platform
	OS: SL 6.1	OS: CentOS 6.1	OS: SLES 11 SP1	OS: SL 6.1	OS: Server 2008 SP2 HPC
<b>Test Procedure</b>	OFED: 1.5.4	OFED: 1.5.4	OFED: 1.5.4	OFED: 1.5.3.2	WinOFED: 2.0
IB Link Initialize	Mandatory	Beta - 5	Beta - 5	Beta - 5	Mandatory: same WinOFED
IB Fabric Initialization	Mandatory	Beta - 5	Beta - 5	Beta - 5	Mandatory: same WinOFED
IB IPoIB Connected Mode	Mandatory	Beta - 5	Beta - 5	Beta - 5	Not Available - 1
IB IPoIB Datagram Mode	Mandatory	Beta - 5	Beta - 5	Beta - 5	Beta
IB SM Failover/Handover - OpenSM	Mandatory	Beta - 5	Beta - 5	Beta - 5	Beta
IB SM Failover/Handover - Vendor SM	Beta	Beta	Beta	Beta	Beta
IB SRP	Mandatory	Beta - 5	Beta - 5	Not Tested - 6	Beta
IB Ethernet Gateway	Beta	Beta	Beta	Beta	Not Available - 3
IB Fibre Channel Gateway	Beta	Beta	Beta	Beta	Not Available - 3
TI iSER	Mandatory	Beta	Beta	Beta	Beta
TI NFS over RDMA	Beta: same OFED - 5	Beta: same OFED - 5	Not Available - 4	Not Tested - 6	Not Available - 1
TI RDS	Mandatory: same OFED	Beta: same OFED	Beta: same OFED	Not Tested - 6	Not Available - 2
TI SDP	Mandatory: same OFED	Beta: same OFED	Beta: same OFED	Not Tested - 6	Not Available - 1
TI uDAPL	Mandatory: same OFED	Beta: same OFED	Beta: same OFED	Not Tested - 6	Beta: same WinOFED, OS
TI Basic RDMA Interop	Mandatory: same OFED	Beta: same OFED	Beta: same OFED	Not Tested - 6	Not Available - 3
TI RDMA Operations	Mandatory: same OFED	Beta: same OFED	Beta: same OFED	Not Tested - 6	Not Available - 3
TI MPI Open MPI	Mandatory: same OFED	Beta: same OFED	Beta: same OFED	Not Tested - 6	Not Available - 2
TI MVAPICH (1) - OSU	Mandatory: same OFED	Beta: same OFED	Beta: same OFED	Not Tested - 6	Not Available - 2

<b>Mandatory</b>	<b>Mandatory for all nodes in all configurations</b>
<b>Mandatory: same OFED</b>	<b>Mandatory for nodes running the same OFED version; OS differences ignored</b>
<b>Mandatory: same WinOFED</b>	<b>Mandatory for nodes running the same WinOFED version; OS differences ignored</b>
<b>Mandatory: same OS</b>	<b>Mandatory for nodes running the same OS version; OFED differences ignored</b>
<b>Mandatory: same OFED, OS</b>	<b>Mandatory for nodes running the same OS and OFED versions</b>
<b>Mandatory: same WinOFED, OS</b>	<b>Mandatory for nodes running the same OS and WinOFED versions</b>
<b>Beta</b>	<b>Beta for all nodes in all configurations</b>
<b>Beta: same OFED</b>	<b>Beta for nodes running the same OFED version; OS differences ignored</b>
<b>Beta: same WinOFED</b>	<b>Beta for nodes running the same WinOFED version; OS differences ignored</b>
<b>Beta: same OS</b>	<b>Beta for nodes running the same OS; OFED differences ignored</b>
<b>Beta: same OFED, OS</b>	<b>Beta for nodes running the same OS and OFED versions</b>
<b>Beta: same WinOFED, OS</b>	<b>Beta for nodes running the same OS and WinOFED versions</b>



**Explanation of table notation:**

- 1) The feature is not currently supported by the WinOF stack.
- 2) The ULP application has not been ported to the WinOF Stack.
- 3) The test has not been updated for WinOF
- 4) The feature is not currently supported by the OFED stack.
- 5) Expected to be promoted to mandatory status for the spring 2012 logo event
- 6) A same test performed in a previous event

**Table 25 - Ethernet Transport Test Status for November 2011 - OFED 1.5.4**

Test Procedure	Linux
Ethernet Link Initialize	Mandatory
Ethernet Fabric Initialization	Mandatory
Ethernet Fabric Failover	Beta
Ethernet Fabric Reconvergence	Beta
iWARP Connectivity	Mandatory
TI iSER	Beta
TI NFS over RDMA	Beta
TI RDS	Beta
TI SDP	Beta
TI uDAPL	Mandatory
TI Basic RDMA Interop	Mandatory
TI RDMA Operations	Beta
TI MPI Intel	Beta
TI MPI Open MPI	Mandatory
TI MVAPICH2 - OSU	Mandatory

## 1.11 SUBJECTS NOT COVERED

**Table 26 - SUBJECTS NOT COVERED**

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

## 1.12 TEST GLOSSARY

**Table 27 - Test Glossary**

Technical Terms	
HCA	IB Host Channel Adapter
IPoIB	IP over InfiniBand
iSER	iSCSI Extensions for RDMA
MPI	Message Passing Interface
RDF	Readme File
RDS	Reliable Datagram Sockets
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

## 1.13 HOMOGENOUS VERSUS HETEROGENEOUS

Heterogeneous & homogeneous clusters are the same with one exception: the end points must be from the same vendor in homogeneous clusters. The table below defines the guidelines for building homogeneous and heterogeneous clusters

Description	Homogenous	Heterogeneous
Mixing switches (both models and vendor products)	Encouraged	Encouraged
The use of any InfiniBand subnet manager	Encouraged	Encouraged
All devices of the same model number shall use the same firmware.	Mandatory	Mandatory
Any mix of products from the same vendor is acceptable - e.g. different model HCAs	Encouraged	Encouraged
A mix of end points (HCA/RNIC) from different OFA vendors	Prohibited	Mandatory
Mixing x86-32 (ix86) and x86_64 Operating System - see notes	Not-Tested	Not-Tested
32 bit architecture and 32 bit OS - see notes	Not-Tested	Not-Tested
Mixing x86-32 and x86-64 user-level application	Optional	Optional
Mixed system architecture - e.g. x86 servers mixed with IA-64 (Itanium) servers	Prohibited	Prohibited
Mixing endianness in system OS	Prohibited	Prohibited
Mixing the quantity of server RAM installed on the hosts	Encouraged	Encouraged
Mixing the server clock speeds	Encouraged	Encouraged
Mixing the number of server cores	Encouraged	Encouraged
Mixing PCIe generations	Encouraged	Encouraged
All servers shall run the same OFED version.	Encouraged	Encouraged
Mixing supported Operating Systems	Encouraged	Encouraged

**Notes:** QLogic drivers do not support 32 bit operating systems

## 2 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.

## 3 USE OF OPENFABRICS SOFTWARE FOR IBTA/CIWG COMPLIANCE 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 latest OFED release. This will enable cable 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 Compliance Events, 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 participants and InfiniBand solution suppliers.

## 4 USE OF OPENFABRICS SOFTWARE FOR OFA IWG INTEROPERABILITY EVENTS

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 latest OFED releases chosen by the OFA IWG for use in the Interoperability Event.

Should there be any issues with the OpenFabrics community not accepting certain bug fixes or features with the time frames matching with Interoperability Events, 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 participants and InfiniBand solution suppliers.

## 5 GENERAL SYSTEM SETUP

### Configuration

The test environment for the user interface contains:

### 5.1 IB HW UNITS

**Table 28 - IB Equipment**

<b>Equipment</b>	<b>Amount</b>	<b>Details</b>	<b>Check</b>
Operating System	12 or more	The OS should be supported by OpenFabrics.	
4X IB Cables	30 or more	Between 1 meter => 10 meters.	
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		

### 5.2 IB SOFTWARE

#### 5.2.1 LINUX/WINDOWS PLATFORMS

#### 5.2.2 OFED - MOST CURRENT TESTED RELEASE

#### 5.2.3 IB HCA FW – VERSION XXX - VENDOR SPECIFIC

#### 5.2.4 IB SWITCH FW CANDIDATE – VERSION XXX - VENDOR SPECIFIC

#### 5.2.5 IB SWITCH SW – VERSION XXX - VENDOR SPECIFIC

### 5.3 IWARP HW UNITS

### 5.4 IWARP SOFTWARE

#### 5.4.1 LINUX/WINDOWS PLATFORMS

#### 5.4.2 OFED - MOST CURRENT TESTED RELEASE

#### 5.4.3 IWARP RNIC FW – VERSION XXX - VENDOR SPECIFIC

#### 5.4.4 10GbE SWITCH FW CANDIDATE – VERSION XXX - VENDOR SPECIFIC

#### 5.4.5 10GbE SWITCH SW – VERSION XXX - VENDOR SPECIFIC

#### 5.4.6 VENDOR SPECIFIC NOTES

**Note:** Currently there is no interoperability between cxgb3 and nes if peer2peer is enabled. Both nes and cxgb3 have their own proprietary ways of doing "client must send the first fpdu". The Chelsio parameter file /sys/module/iw\_cxgb3/parameters/peer2peer should be

modified on all hosts to contain the appropriate value for each test. For example: the value must be set to '0' for the iWARP Connectivity test and set to '1' for the uDAPL test.

Arlin Davis suggests the following given the current situation:

- 1)The daplttest -T P (performance tests) will always send data from server side first. This test will NOT work reliably with iWARP vendors.
- 2)The daplttest -T T (transaction tests) should work fine with both IB and iWARP vendors given that it always sends from client side first.
- 3)I recommend using only daplttest transaction mode (-T T) in your test plan and removing -T P mode tests.

## 5.5 MPI TESTING

- 1)HCA/RNIC vendors must provide a minimum of five adapters. The adapters need not be all the same model, but they can be.

## 6 IB HW DESCRIPTION & CONNECTIVITY

The test contains two 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.

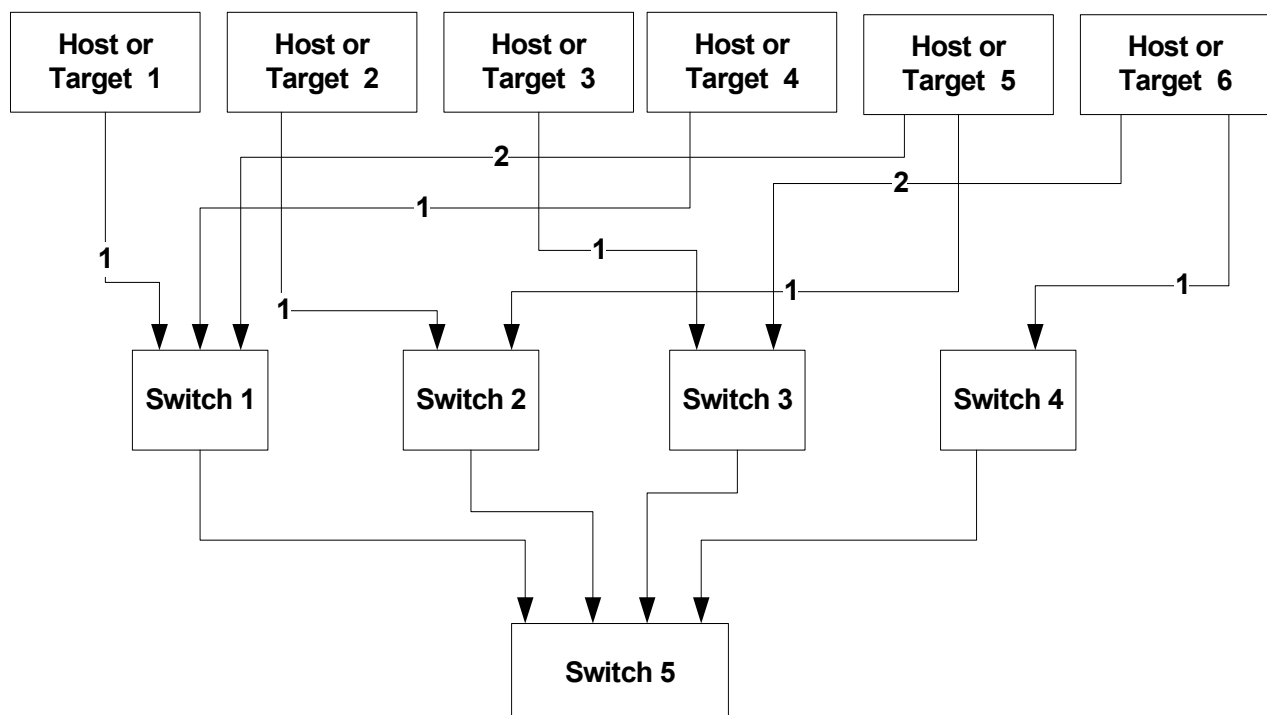
#### 6.2.2 OPENFABRICS SOFTWARE RUNNING ON HOSTS

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

### 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.

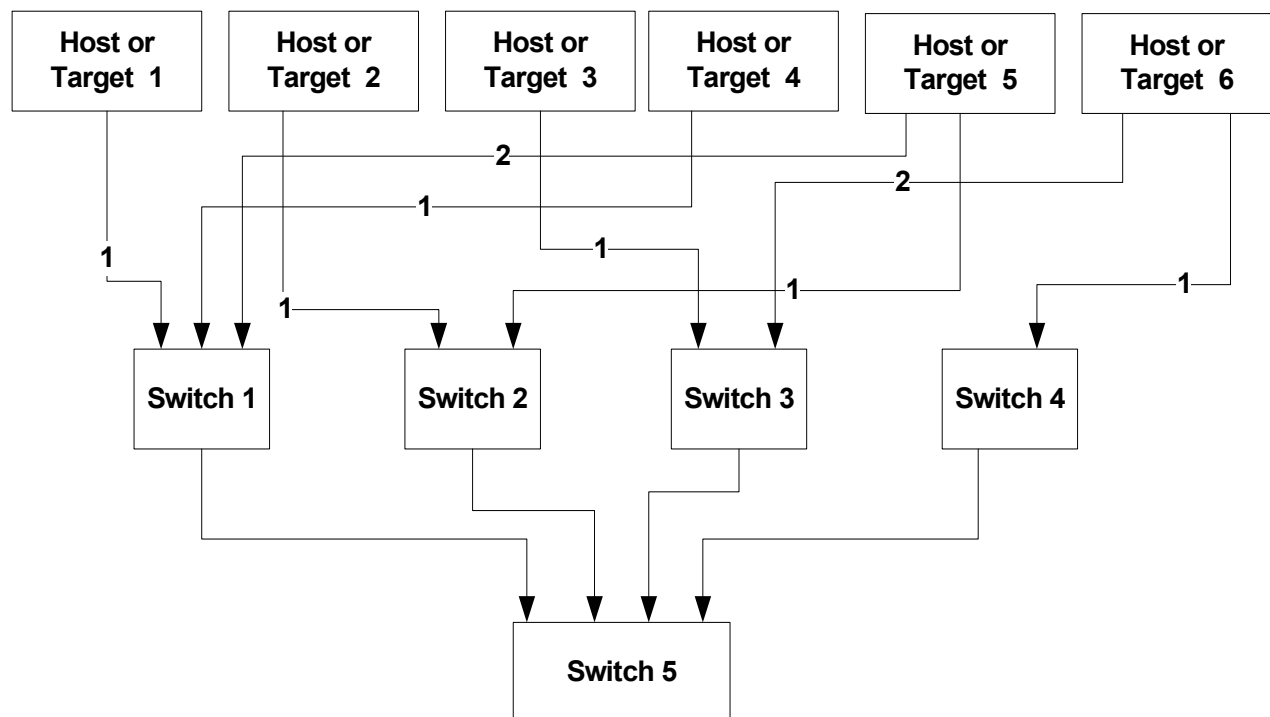
#### 7.2.2 OPENFABRICS SOFTWARE RUNNING ON RNICs

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

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42

## 8 FW & SW INSTALLATION

### 8.1 BURNING THE FW

#### 8.1.1 FIRMWARE POLICY

##### **Firmware Policy during the Interop Debug Event**

The firmware used during the Interop Debug Event is at the discretion of the device vendor. Vendors will be allowed to make changes to the firmware during the Interop Debug Event. However changes should be made as early in the event period as possible to reduce the amount of retesting which will result from these changes.

##### **Firmware Policy during the Interop GA Event**

The firmware image used during the Interop GA Event must be provided to the UNH-IOL at least one week prior to the event. No firmware changes of any kind are allowed during the Interop GA Event. If the vendor does not provide updated firmware by the deadline, then the UNH-IOL will use the firmware from the Interop Debug Event or from the vendor's website, whichever is more current.

##### **Firmware Policy after the Interop GA Event**

The firmware used to obtain the OFA Logo (or a child of this firmware with the same base functionality) must be the default publicly available firmware on the vendor's website and must be the default firmware that is shipped with the product. This must be completed within six months of the Interop GA Event.

#### 8.1.2 PLEASE REFER TO FIRMWARE BURNING TOOLS AND PROCEDURES DOCUMENTATION FROM HCA IB VENDOR

### 8.2 SW INSTALLATION

#### 8.2.1 SOFTWARE POLICY

##### **Software Policy during an Interop Debug Event**

The software used during an Interop Debug Event will be an agreed-upon RC release of the subsequent OFED version. During the Interop Debug Event vendors will be allowed to make changes to the software, provided that the changes are based on the same RC release. Vendors are not allowed to extensively modify the software or completely replace it.

##### **Software Policy during the Interop GA event**

The software used during an Interop GA Event will be the GA release of the same OFED version as was used during the Interop Debug Event. No software changes of any kind are allowed during the Interop GA Event. It is the vendor's responsibility to ensure that any changes made during the Interop Debug Event are present in the OFED GA release. Vendors whose products do not use firmware may request that patches be applied to an OFED GA release if that release has known defects that prevent the vendor product from being interoperable. The Arbitration Committee will be responsible for approving the requested patches.

##### **Software Policy after the Interop GA event**

All products that are granted the OFA Logo must be distributed by default with the OFED GA version (or a later revision of OFED with the same base functionality).

#### 8.2.2 PLEASE REFER TO SOFTWARE INSTALLATION MANUAL FROM HCA IB VENDOR.

### 8.2.3 PLEASE REFER TO SOFTWARE INSTALLATION MANUAL FROM RNIC VENDOR.

## 8.3 SUMMARY

- For the Interop GA Event the vendor cannot update or change any part of the device under test - this includes hardware, firmware and software. The only exception is for an outright hardware failure in which case the hardware may be replaced with an identical piece of hardware with the same SW and FW.
- If an end user requests customized firmware or a modified version of OFED, then the vendor must disclose that this is not an OFA certified configuration.
- The OFA reserves the right to revoke the OFA Logo for products that do not follow these policies.
- These policies will be in effect for the April 2011 Interop Events and all events thereafter.

## 8.4 HARDWARE POLICY

For MPI testing, HCA/RNIC vendors must provide at least five adapters. The adapters need not be all the same model, but they can be.

## 8.5 OFED USAGE

- OFED Release Candidates (RC) should be used during the Interop Debug Event. This allows vendors to resolve bugs and issues and commit them to the OFED tree before the OFED General Availability (GA) is released.
- OFED GA versions shall be used for the Interop GA Events.

# 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. Please contact them at [ofalab@iol.unh.edu](mailto:ofalab@iol.unh.edu) if you wish to obtain copies of the latest scripts

### 10.1 IB LINK INITIALIZE USING OFED FOR LINUX

#### 10.1.1 Procedure

- 1) Select a pair of devices to test from the created topology
- 2) Determine the maximum port width and lane speed supported by both devices
- 3) Select a cable to use which has been certified for the link parameters determined by step 2 of section 10.1.1 during an IBTA Plugfest held within the last 6 months
- 4) Disconnect all IB cables from the selected devices
- 5) Shutdown all SMs running on the selected devices
- 6) Connect the selected devices back to back using the cable selected during step 3 of section 10.1.1
- 7) Wait for a physical indication that a link has been established
- 8) Verify that the link created in step 4 of section 10.1.1 has come up with the parameters determined in step 2 of section 10.1.1
- 9) Repeat steps 1-8 with a different device pairing
  - a) All unique device pairs present in the created topology must be tested; except SRP target to SRP target
  - b) Each device must link at the maximum port width and lane speed supported by both devices in all pairings for said device to pass link initialization testing

#### 10.1.2 Method of Implementation for all Linux OSs

- 1) To perform step 7 of section 10.1.1:
  - a) Look for link LEDs on the ports you are using
- 2) To perform step 8 of section 10.1.1:
  - a) ssh into a device supporting such remote connections and is running the OFED stack; usually a compute node with an HCA
  - b) Run "ibdiagnet -wt <desired-topology-file-name>"
  - c) Check the topology file created by the previous command:
    - i) Match the GUIDs to the devices in the selected pair
    - ii) Verify link width is the highest common denominator of pair capabilities (1x, 4x, 12x)
    - iii) Verify link speed is the highest common denominator of pair capabilities (2.5G, 5G, 10G, 14G)
- 3) To determine switch to SRP target and switch to switch link parameters
  - a) Run the commands outlined by step 2 of section 10.1.2 from a third device

- i) Should be a compute node with an HCA that is linked to a switch that is part of the desired pairing
- ii) Carefully match the GUIDS as you now have more than just two in the topology file

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42

## 10.2 IB FABRIC INITIALIZATION USING OFED

### 10.2.1 Architect the Network we want to build.

- 1) Develop a cluster diagram based on the devices that have been submitted for Interop Testing and assign IP addresses to the IPoB interfaces and the ethernet management interfaces.
- 2) See [Figure 3- Sample Network Configuration](#) below.

### 10.2.2 Procedure

- 1) Connect the HCAs and switches as per the Architected Network and make sure that no SM/SA is running on the Fabric.
- 2) Start an SM on a device and let it initialize (all SM's will need to be tested)
- 3) Visually verify that all devices are in the active state. Verify that the LED is on when the port is active.
- 4) Run "ibdiagnet -wt <file>" to generate a topology file
- 5) Run "ibdiagnet -pc" to clear all port counters
- 6) Wait 17 seconds as per the specifications requirements.
- 7) Run "ibdiagnet -c 1000" to send 1000 node descriptions.
- 8) Run "ibdiagnet" to generate fabric report.
  - a) Use /tmp/ibdiagnet.sm file to determine running sm
  - b) sminfo can also be used to determine the master SM or saquery -s to find all SMs.

**Note:** "ibdiagnet -r" seg faulted but was fixed in OFED 1.5 according to Bug 1618
- 9) Run "ibchecknet" to build guid list.
- 10) Run "ibdiagnet -t <file>" to compare current topology to the previously generated topology file

### 10.2.3 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
- 4) Verify that step 10 above indicates that the topology before the test and the topology after the test are the same.

Restart all devices in the fabric and follow Sections 10.2.2 and 10.2.3. Run the SM from a different device in the fabric until all SMs present have been used. All SMs on managed switches (including those switches running **opensm**) should be tested and at least one instance of **opensm** on an HCA must be tested. If there are HCAs from more than one vendor, then **opensm** should be run from each vendor's HCA.

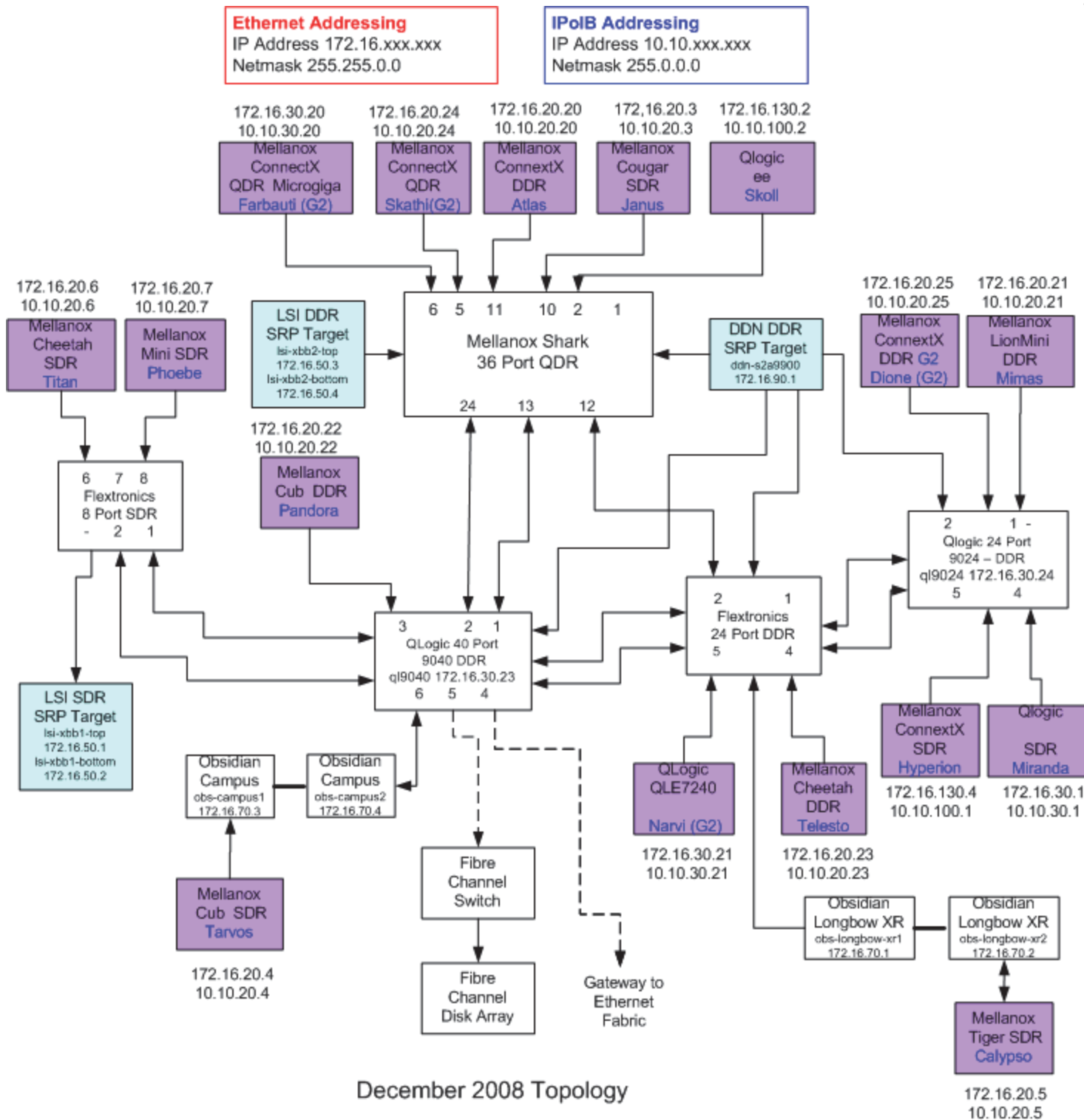
Each device must pass all verification procedures with every SM to pass Fabric Initialization test.

**Table 29 - ibdiagnet commands**

Commands	Description
ibdiagnet -c 1000	Send 1000 node descriptions
ibdiagnet -h	Help
ibdiagnet -lw 4x - ls 2.5	Specify link width and speed
ibdiagnet - pc	Clear counters
ibdiagnet -t <file>	Compare current topology to saved topology
ibdiagnet -wt	Writes the topology to a file

**Note:** The topology file is being generated after the SM starts but before any testing has started. The topology comparison is being performed after testing has been completed but before the systems get rebooted. A topology check is performed during every part of every test section that does not specifically state "change the topology". For example Fabric Init only has 1 part so there is only 1 check but RDS has 2 parts so 2 checks are performed. However, IPoIB has 3 parts for each of 2 modes but 1 of those parts specifically says to change the topology so only 4 checks occur.

**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. Verify that the LED is on when the port is active.
  - b) Run `"ibdiagnet -r"` and verify that the SM you started is the one that is running 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) ping -i 0.01 -t 3 -c 10 -s <ping size> <destination>
      - i) "-i" - interval 0.01 seconds
      - ii) "-t" - IP Time to Live equals 3 seconds
      - iii) "-c" - count equals 100
      - iv) "-s" - size of the ping
      - v) "destination" - the IP address of the IPoIB interface being pinged.
    - 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. Verify that the LED is on when the port is active.
  - b) Run "ibdiagnet -r" and verify that the SM you started is the one that is running 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.
- 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*).

**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.

- 5) Issue the command: `sysctl net.ipv4.neigh.ib0.unres_qlen=33`
  - a) This sets the qlen variable to 33 which increases the buffer size so that you do not get an initial dropped packet when using ping sizes 8192 and greater.
- 6) Ping every HCA except localhost with packet sizes of 64, 256, 511, 512, 1024, 1025, 2044, 4096, 8192, 16384, 32768, and 65507.
  - a) `ping -i 0.01 -t 3 -c 10 -s <ping size> <destination>`
    - i) "-i" - interval 0.01 seconds
    - ii) "-t" - IP Time to Live equals 3 seconds
    - iii) "-c" - count equals 100
    - iv) "-s" - size of the ping
    - v) "destination" - the IP address of the IPoIB interface being pinged.
  - b) Repeat step #4 before issuing each ping command. Every packet size is a new ping command.
- 7) 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

- 1) 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.
- 2) Issue the command: `sysctl net.ipv4.neigh.ib0.unres_qlen=3`
  - a) This sets the qlen variable back to the default.

## 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.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42



## 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 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). To verify that an RDMA link has been initialized between Host A and Host B run the following commands:
  - a) Start a server in verbose mode on Host A:
    - i) `rping -sv`
  - b) Start a client on Host B to ping Host A.
    - i) `rping -cv -a Host A RNIC_IP_Address`
  - c) Optional Command for the client
    - i) `rping -cv -a Host A RNIC_IP_Address -C 4 -S 50`

**Note:** This sends a count of 4 pings and character strings of size 50
- 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.

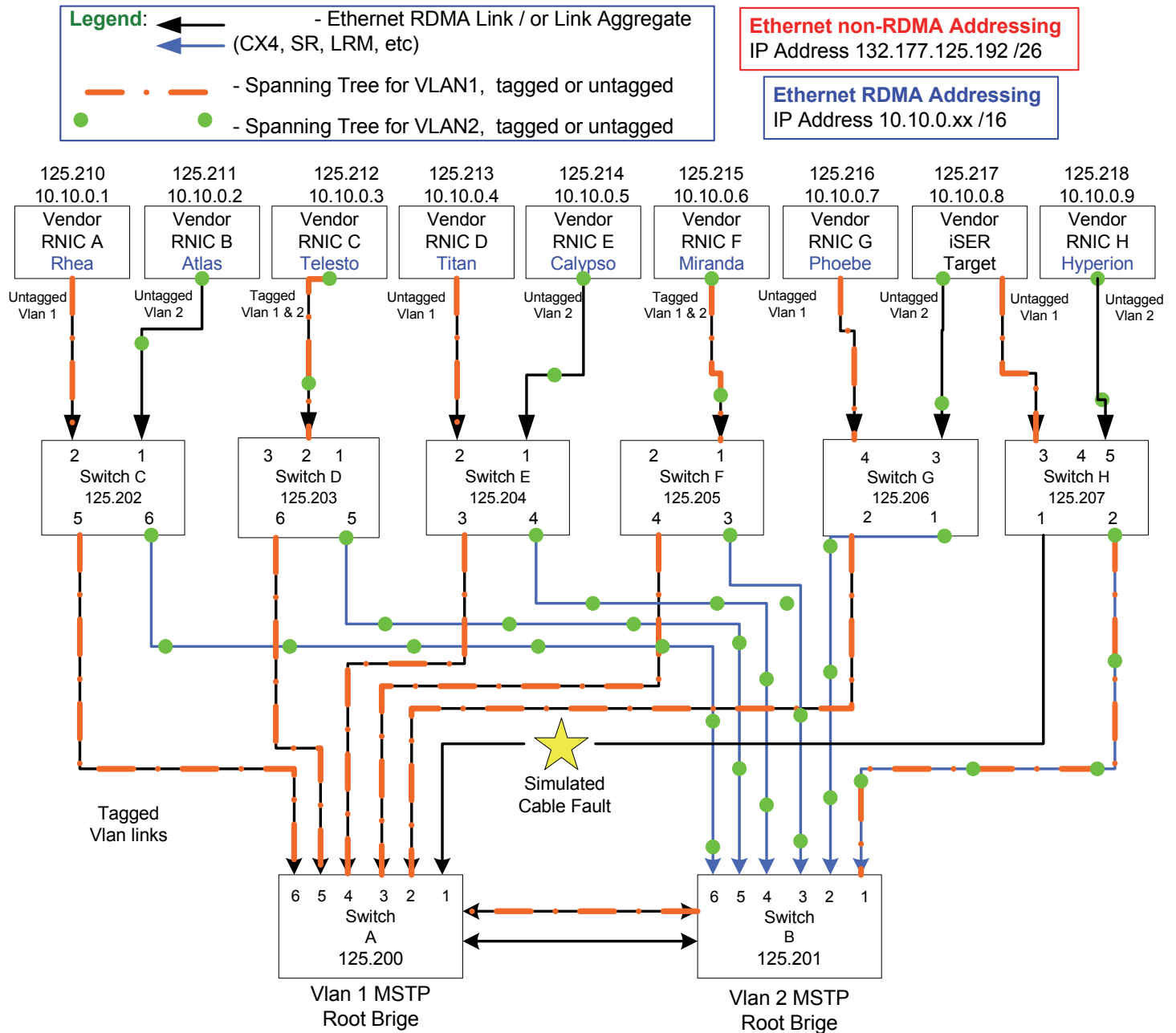
**Note:** This includes RNICs supporting Link Aggregation, as well as switch to switch links.
  - d) Set spanning tree priorities such that desired bridge(s) becomes root bridge(s).
  - e) See [Cluster Connectivity](#).

- 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**



**Note on final Network Architecture**

**Dependent on:** RNIC VLAN tag support, Link Agg support, MSTP support (RSTP support is assumed) and Port Type.  
Layer-3 Routing will not be utilized.



## 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) Compliant 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 re-discovery 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

- 1) The interoperability tests can be run in point to point mode or switched. Connect 2 iWARP hosts RNICs together or to a 10GbE switch.
- 2) Ensure that /sys/module/iw\_cxgb3/parameters/peer2peer contains '0' on all hosts.

### 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	1 2
	• Test 2.13: Sequence of Interleaved RDMA Sendse And Terminate Commands	3 4
	• Test 2.14: Sequence of Interleaved RDMA Sendseinv And Terminate Commands	5 6
	• Test 2.15: Sequence of Interleaved RDMA Write With All Other RDMA Commands	7 8
	• Test 2.16: Sequence of Interleaved RDMA Read With All Other RDMA Commands	9 10
	• Test 2.17: Sequence of Interleaved RDMA Send With All Other RDMA Commands	11 12
	• Test 2.18: Sequence of Interleaved RDMA Sendinv With All Other RDMA Commands	13 14
	• Test 2.19: Sequence of Interleaved RDMA Sendse With All Other RDMA Commands	15 16
	• Test 2.20: Sequence of Interleaved RDMA Sendseinv With All Other RDMA Commands	17 18 19
<b>Step C</b>	<b>Group 3: Multiple Connections:</b>	20 21
	• Test 3.1: Single RDMA Operations Over Multiple Connections	22
	• Test 3.2: Multiple RDMA Operations Over Multiple Connections	23
	• Test 3.3: RDMA Operations Over 25 Connections	24 25
	• Test 3.4: Simultaneous Operations Over 25 Connections	26 27
<b>Step D</b>	<b>Group 4: Disconnect/Reconnect Physical Connections:</b>	28 29
	• Test 4.1: Termination Followed By A WRITE	30
	• Test 4.2: Termination Followed By A READ	31 32
<b>Step E</b>	<b>Group 5: Speed Negotiation:</b>	33 34
	• Test 5.1: RNICs Operating At 10g And 1g Speed	35 36
<b>Step F</b>	<b>Group 6: RDMA Error Ratio:</b>	37 38
	• Test 6.1: Sequence of All Zeros	39
	• Test 6.2: Sequence of All Ones	40
	• Test 6.3: Sequence of Ones Followed By Zeros	41 42

- 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

### 12.2.1 Installation

**Note:** Steps 2-4 are unneeded if an OFED supported OS is used along with an official OFED release downloaded from <http://www.openfabrics.org>

- 1) Verify that you are using a Linux kernel with NFS/RDMA on every system used

- 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: <ftp://ftp.kernel.org/pub/linux/kernel/v2.6/>

**Note:** OFED supported OS releases of lower kernel revision than mentioned above have been updated by their respected maintainers to allow NFS RDMA to function. Check the `nfs-rdma.release-notes.txt` provided with the OFED release you are using for supported OS releases.

**Note:** As of OFED 1.5.3 rc2 NFSoRDMA is not installed by default. To do so you must have built OFED from src with `nfsrdma=y` and `rnfs-utils=y` directives contained within the `ofed.conf` file used by the OFED installer.

- i) To generate an `ofed.conf` file run the following from within the downloaded OFED src.

1. `./install.pl -p`

- ii) Add the following directives to the generated `ofed-all.conf` file

1. `nfsrdma=y`

2. `rnfs-utils=y`

- iii) Install OFED

1. `./install.pl -c ofed-all.conf`

- 2) Configure the RDMA stack on every system used

- 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`, `mtchca`, `ehca`, `ipath`, `qib`, etc.) or iWARP adapter support (`amso`, `cxgb3`, etc.).

- c) If you are using InfiniBand, be sure to enable IP-over-InfiniBand (IPoIB) support.

- 3) Configure the NFS client

- 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.

- 4) 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) 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 and boot it
- 5) Check RDMA Setup
  - a) If you are using InfiniBand, make sure there is a Subnet Manager (SM) running on the network.
  - b) Use IPoIB to ping two hosts.
- 6) Configure NFS exports, start NFS server
  - 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. Export 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. This does not interfere with normal NFS over TCP/IP operations.

  - d) The remainder of this section will assume an export of /server
  - e) 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
- 7) 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.
- b) On the client system:
  - i) Use this command to mount the NFS server export:
    1. `$ mount <server name or TCP/IP address>:<export> /<mount path>`
  - ii) To verify that the mount is using TCP, run "`cat /proc/mounts`" and check the "proto" field for the given mount.
- 8) Check NFS/RDMA Setup
  - a) For the NFS components enabled above (client and/or server), test their functionality over RDMA.
  - b) 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`
    - ii) Regardless of how the client was built (module or built-in), use this command to mount the NFS server export:
      1. `$ /sbin/mount.nfs <IPoIB server name or address>:<export> /<mount path> -o \ rdma,port=20049`

**Note:** OFED will build and install the mount utility needed. The binary is called `mount.nfs`. Either this binary or the mount binary provided with `nfs-utils` revision greater than version 1.1 can be used. The remainder of this section will assume `mount.nfs` built by OFED is used.
    - iii) To verify that the mount is using RDMA, run "`cat /proc/mounts`" and check the "proto" field for the given mount.
- 9) Connectathon
  - a) Download the Connectathon test suite from <http://www.connectathon.org/nfstests.html>
  - b) Install Connectathon on every client to be used
    - i) Modify `tests.init` within the connectathon tarball to suit your environment.
      1. The `MOUNTCMD`, `UMOUNTCMD` and `MNTOPTIONS` directives are unimportant as we will be calling the runtests connectathon binary directly.
      2. Be sure to remove the `-fwritable-strings` option from the `CFLAGS` variable. Your build will fail if this is used.
    - ii) Run `make` to build the connectathon binaries.
- 10) Test the connectathon runtests binary
  - a) Run `sudo ./runtests -a -t /mnt/` to test the binary against the local filesystem.
  - b) All tests should pass but you will see 1 warning. This is ok.

### 12.2.2 NFSoRDMA Test Procedure

- 1) Start an SM
- 2) Server setup
  - a) Add nfs rdma server support to the running kernel if not already present.
    - i) `$ modprobe svcrdma`
  - b) Start the server
    - i) `$ /etc/init.d/nfs start`
  - c) Tell the server to listen for rdma connection requests on port 20049
    - i) `$ echo rdma 20049 > /proc/fs/nfsd/portlist`
- 3) Client setup
  - a) Add nfs rdma client support to the running kernel if not already present.
    - i) `$ modprobe xprtrdma`
  - b) Mount the servers export using rdma
    - i) `$ /sbin/mount.nfs <server IPoIB address>:/server /<mount path> -o \ rdma,port=20049 -i`  
**Note:** <mount path> is assumed to be /mnt/<servername> for the remainder of this section
  - c) Verify that the mount is using the rdma protocol
    - i) Verify that the mount is using RDMA, run "cat /proc/mounts" and check the "proto" field for the given mount.
- 4) Run Connectathon's runtests binary
  - a) `$ ./runtests -a -t /mnt/<servername>/<hostname>`
- 5) Repeat steps 2-4 using a new client-server pair until all nodes have acted as both a server and a client.
- 6) Repeat steps 2-5 using a new SM until all registered SM's have been used.
- 7) All tests run by the connectathon runtests binary must pass on all client nodes rdma mount points from all server nodes using all SM's in order for the device to pass 12.2.2 overall.

## 12.3 TI RELIABLE DATAGRAM SERVICE (RDS) USING OFED

### 12.3.1 RDS-PING PROCEDURE

- 1) Use the command `modprobe rds_rdma` 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) **[For IB]** 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) **[For iWARP hosts with Chelsio RNICs]** Set the `peer2peer` bit to one with the following command:
  - a) **For T3s:** `echo -n 1 > /sys/module/iw_cxgb3/parameters/peer2peer`
  - a) **For T4s:** `echo -n 1 > /sys/module/iw_cxgb4/parameters/peer2peer`
- 5) Choose a host and use `rds-ping host` to communicate with every other end point in the fabric.

**Note:** Be sure that you identify the correct host when using the command `rds-ping host`. iWARP hosts are usually configured with `eth0` as the on-board NIC and `eth2` as the RNIC

- 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.
- 6) Verify that all nodes respond without error.

**Note:** To avoid losing packets, do not run this while RDS-Stress is running.

### 12.3.2 RDS-STRESS PROCEDURE

- 1) Choose 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`

**Note:** If you repeat the test in less than one minute you may get the error message "Cannot assign requested address" since the port numbers are not immediately reusable. Either wait or change the port number using the `-p` option

**Note:** The `-t` option is for the number of tasks (child processes), which defaults to 1 so `"-t 1"` is optional. The `-d` option is for the message queue depth, which also defaults to 1 so `"-d 1"` is optional.

- 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

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

**Note:** Currently SDP is not available to iWARP vendors due to a licensing issue.

### 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) [For IB] 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] netperf -p {port number} -H {server node's IPoIB} -l 1 -t TCP\_STREAM -- -m {message size} -s {local buffer size}
    - b) [For iWARP] netperf -p {port number} -H {server node's IP} -l 1 -t TCP\_STREAM -- -m {message size} -s {local buffer size}
    - c) **Example:** netperf -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) [For IB] 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 12.4.6 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 must be done in each direction and then bidirectional and for IB using every SM available. If needed, you can create a 4 MB file 'tfile.orig' filled with random data using the following command:

```
dd if=/dev/urandom of=tfile.orig bs=1000 count=4000
```

### 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` 1
- b) `ib_sdp` should be greater than 0 - reference count should be greater than 0. Each connection opens three reference counts. 2
- c) During these transactions double check that `sdp` connection has been established, you can see it in `/proc/net/sdp/conn_main`. 3
- 3) [For IB] Start an SM (all SM's will need to be tested) and let it initialize 4
- a) Verify that the running SM is the one you started. 5
- 4) SFTP: 6
- a) [For IB] Connect an HCA pair via SFTP on IPoIB using the specified user name and password. 7
- b) Put the 4MB file to the `/tmp` dir on the remote host. 8
- c) Get the same file to your local dir again. 9
- d) Compare the file using the command `cmp tfile tfile.orig`. 10
- i) The two must be identical 11
- e) [For IB] Repeat the procedure with a different SM. 12
- 5) [For IB] Repeat the procedure with a different SM. 13
- 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 12.4.7 overall. 14

## 12.4.8 SCP PROCEDURE 15

A 4MB file will be SCP'd to the partner and then SCP'd back and binary compared to the original file. This must be done in each direction and then bidirectional and for IB, using every SM available. If needed, you can create a 4 MB file 'tfile.orig' filled with random data using the following command: 16

```
dd if=/dev/urandom of=tfile.orig bs=1000 count=4000 17
```

### 12.4.8.1 SETUP 18

- 1) A special account for this should be created as follows: 19
- a) Username: Interop 20
- b) Password: openfabrics 21

### 12.4.8.2 Procedure 22

- 1) [For IB] Start an SM (all SM's will need to be tested) and let it initialize 23
- a) Verify that the running SM is the one you started. 24
- 2) SCP: 25
- a) Put the 4MB file to the `/tmp` dir on the remote host via SCP. 26
- b) Get the same file to your local dir again via SCP. 27
- c) Compare the file using the command `cmp tfile tfile.orig`. 28
- i) The two must be identical 29

- d) Repeat step #2 with a different HCA pair until all HCAs have been tested with all others (All to All). 1  
2  
3) [For IB] Repeat the procedure with a different SM. 3

**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 12.4.8 overall. 4  
5  
6

**Note:** Sections 12.4.6, 12.4.7 and 12.4.8 must pass using the configuration determined by sections 12.4.1, 12.4.2, 12.14.3, 12.4.4 and 12.4.5 for the device to pass SDP overall. 7  
8  
9

## 12.5 TI uDAPLTEST COMMANDS USING OFED

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

### 12.5.1 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.
- [For IB] an SM needs to be running.
- [For iWARP hosts with Chelsio RNICs] Ensure that `/sys/module/iw_cxgb3/parameters/peer2peer` contains '1' on all hosts.

### 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 RDMA BASIC INTEROP

### 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.

### 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).

### 12.6.5 iWARP Setup

Connect iWARP RDMA endpoints to an 10GbE switch.

### 12.6.6 RDMA Connectivity Setup

Each of the tests described below must be run twice with Host A being the server and then Host B being the server. This ensures that the different semantics associated with active and passive sides of the connection are exercised. This way each RDMA interface tested will be sending RDMA data (Requestor) in one test and receiving RDMA data (Target) in the next.

### 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) **[For IB]** `ib_read_bw -d <dev_name> -i <port> -m 2048`
  - b) **[For iWARP]** - Not applicable - see 12.6.9
- 3) On the client device issue the following command on command line:
  - a) **[For IB]** `ib_read_bw -d <dev_name> -i <port> -s 1 -n 25000 -m 2048`
  - b) **[For iWARP]** - Not applicable - see 12.6.9

- 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) **[For IB]** `ib_read_bw -d <dev_name> -i <port> -m 2048`
  - b) **[For iWARP]** - Not applicable - see 12.6.10
- 3) On the client device issue the following command on command line:
  - a) **[For IB]** `ib_read_bw -d <dev_name> -i <port> -s 1000000 -n 300 -m 2048`
  - b) **[For iWARP]** - Not applicable - see 12.6.10
- 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) **[For IB]** `ib_write_bw -d <dev_name> -i <port> -m 2048`
  - b) **[For iWARP]** `rdma_bw -c -s 1 -n 25000`
- 3) On the client device issue the following command on command line:
  - a) **[For IB]** `ib_write_bw -d <dev_name> -i <port> -s 1 -n 25000 -m 2048`
  - b) **[For iWARP]** `rdma_bw -c -s 1 -n 25000 RNIC_IP_Address`
- 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) **[For IB]** `ib_write_bw -d <dev_name> -i <port> -m 2048`
  - b) **[For iWARP]** `rdma_bw -c -s 1000000 -n 300`
- 3) On the client device issue the following command on command line:
  - a) **[For IB]** `ib_write_bw -d <dev_name> -i <port> -s 1000000 -n 300 -m 2048`
  - b) **[For iWARP]** `rdma_bw -c -s 1000000 -n 300 RNIC_IP_Address`
- 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) **[For IB]** `ib_send_bw -d <dev_name> -i <port> -m 2048`
  - b) **[For iWARP]** - Not applicable - see 12.6.9
- 3) On the client device issue the following command on command line:
  - a) **[For IB]** `ib_writesend_bw -d <dev_name> -i <port> -s 1 -n 25000 -m 2048`
  - b) **[For iWARP]** - Not applicable - see 12.6.9
- 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) **[For IB]** `ib_send_bw -d <dev_name> -i <port> -m 2048`
  - b) **[For iWARP]** - Not applicable - see 12.6.10
- 3) On the client device issue the following command on command line:
  - a) **[For IB]** `ib_send_bw -d <dev_name> -i <port> -s 1000000 -n 300 -m 2048`
  - b) **[For iWARP]** - Not applicable - see 12.6.10
- 4) Verify that the operation completed without error and the level of performance achieved is reasonable and as expected.

### 12.6.13 Additional IB Notes

- 1) Alternate read commands available
  - a) Server command: `ib_read_bw -m 2048`
  - b) Client command (small): `ib_read_bw -s 1 -n 25000 IPoB Address for server -m 2048`
  - c) Client command (large): `ib_read_bw -s 1000000 -n 300 IPoB Address for server -m 2048`
- 2) Alternate write commands available
  - a) Server command: `ib_write_bw -m 2048`
  - b) Client command (small): `ib_write_bw -s 1 -n 25000 IPoB Address for server`
  - c) Client command (large): `ib_write_bw -s 1000000 -n 300 IPoB Address for server -m 2048`
- 3) Alternate send commands available
  - a) Server command: `ib_send_bw -m 2048`

- b) Client command: `ib_send_bw -s 1 -n 25000 IPoB Address for server -m 2048`
- c) Client command (large): `ib_send_bw -s 1000000 -n 300 IPoB Address for server -m 2048`
- 4) Explanation of parameters
  - a) "-d" allows you to specify the device name which may be obtained from the command `lane: ibv_devinfo`
  - b) "-i" allows you to specify the port number. This may be useful if you are running the tests consecutively because a port number is not immediately released and this will allow you to specify another port number to run the test.
  - c) "-s" - this is the size of the operation you wish to complete
  - d) "-n" - this is the number of operations you wish to complete.
  - e) "-m" - this specifies the IB PMTU size. AS of 10/3/2011 some devices did not support greater than 2048

#### 12.6.14 Additional iWARP Notes

- 1) The "-c" option specifies to use the `rdma_cm` for connection

#### IB 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 300

## 12.7 TI RDMA STRESS TEST

### 12.7.1 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.

### 12.7.2 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](#).

### 12.7.3 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.

### 12.7.4 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) **[For IB]** `ib_write_bw -d <dev_name> -i <port> -m 2048`
    - ii) **[For iWARP]** `rdma_bw -c -s 1024 -n 25000`
  - b) On the client device issue the following command on command line:
    - i) **[For IB]** `ib_write_bw -d <dev_name> -i <port> -s 1024 -n 25000 -m 2048`
    - ii) **[For iWARP]** `rdma_bw -c -s 1024 -n 25000 RNIC_IP_Address`
- 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)

### 12.7.5 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) **[For IB]** `ib_write_bw -d <dev_name> -i <port> -m 2048`
    - ii) **[For iWARP]** `rdma_bw -c -s 1024 -n 25000`
  - b) On the client device issue the following command on command line:
    - i) **[For IB]** `ib_write_bw -d <dev_name> -i <port> -s 1024 -n 25000 -m 2048`
    - ii) **[For iWARP]** `rdma_bw -c -s 1024 -n 25000 RNIC_IP_Address`
- 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 TI MPI - INTEL MPI USING OFED

### 12.8.1 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) **[For IB]** 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/mapi/219848.htm>

### 12.8.2 SETUP FOR THE CLUSTER

- 1) Install same O/S version on homogenous x86\_64 systems. (Recommend RHEL 5.2, EM64T or newer). 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	<p>IA-32, Intel® 64 or IA-64 (formerly Itanium) architecture-based system. Examples of such Intel processors are:</p> <p>Intel® Pentium® 4 processor, or Intel® Xeon® processor, or Intel® Itanium® processor, or Intel® Core™2 Duo processor (example of Intel® 64 architecture)</p> <p>Note that it is assumed that the processors listed above are configured into homogeneous clusters</p> <p>4 GB of RAM (8 GB of RAM recommended) 1 GB of hard disk space (10 GB of space recommended)</p>
Operating System Support	
All three architectures	<p>Red Hat* Enterprise Linux* 4.0, 5.0 SUSE* Linux Enterprise Server* (SLES) 9, 10</p>
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	<p>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*</p>
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**

### 12.8.3 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/RNIC hostnames and addresses for the IB/RNIC network interfaces
- 5) Modify /etc/dat.conf and change the netdev reference to the appropriate interface being used.
  - a) **[For IB]** - ib0, ib1, *ibx*
  - b) **[For iWARP]** - eth2, eth3, *ethx*
- 6) [For IB] Run OpenSM either on the headnode OR from one of the switches.
- 7) Verify connectivity by pinging the IB/RNIC addresses on all systems.

### 12.8.4 Setup information for Intel MPI

- 1) Install Intel MPI in /opt/intel/mpi/3.x.x 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 1
- edit /etc/fstab and add "hostname:/home/test/exports /home/test/exports nfs" 2
- on all other nodes 3
- 7) Untar the Intel Test Suites on the headnode in /home/test/exports 4
- 8) run mpdboot on the head node. For example: if you have 6 nodes on the 5
- cluster and want to run over ethernet: 6
- From the /home/test directory run: "mpdboot -n 6 -r ssh -f ./mpd.host.eth- 7
- ernet" 8
- 9) Run test suite over Ethernet to validate your installation: 9
- "export I\_MPI\_DEVICE = sock" 10
- run tests...(refer to test plan) 11
- "mpdallexit" 12
- 10) Run test suite over IB 13
- export I\_MPI\_DEVICE = rdssm 14
- mpdboot -n 6 -r ssh -f ./mpd.host.ib 15
- run tests.... (refer to test plan) 16
- "mpdallexit" 17

## 12.8.5 ADDITIONAL INFORMATION 19

- 1) Go to the individual test directories and follow the steps in the respective 20
- README-\*.txt files. The recommended order for running the test suites in 21
- the order of increasing execution time: 22
- a) mpich2-test: see README-mpich2-test.txt file. 23
- 2) For Intel MPI Support Services go to: 24
- <http://www.intel.com/support/performance/tools/cluster/index.htm> 25
- See the [Intel MPI Reference Manual](#) for Additional information 26

## 12.8.6 INTEL MPI BENCHMARK SETUP 28

The IMB tests must be compiled with the -DCHECK compiler flag set, to enable 29

automatic self-checking of the results. Modify the appropriate make\_arch file as 30

follow: 31

```
MPI_HOME      = 32
MPI_INCLUDE = . 33
LIB_PATH      = 34
LIBS          = 35
CC            = mpicc 36
OPTFLAGS      = -O 37
CLINKER       = ${CC} 38
LDFLAGS       = 39
CPPFLAGS      = 40
```

### 12.8.7 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.

### 12.8.8 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.

### 12.8.9 INTERPRETING THE RESULTS

- 1) For mpich2-test test suites: See Table 22b - [TI - Intel MPICH2 Test 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. 1
- 2) For IntelMPITEST test suite: See Table 22c - [TI - Intel MPI Test Suite - \(Not 2](#)  
[part of OFA Stack\)](#) 3

The **Tests/summary.xml** file produced by the test suites has the following 4  
uniform format: 5

- The file header contains information on the test suite and testing envi- 6  
ronment 7
- The rest of the file represents the results of the test suite run. 8

## 12.9 TI MPI - OPEN MPI USING OFED

The following values are used in examples below:

- \$MPIHOME: The absolute directory location of the Open MPI installation that is common to all systems under test.
- \$NP: The number of MPI processes to use in the test.
- \$HOSTFILE: The absolute filename location of the hostfile
- \$IMBHOME: The absolute directory location of the Intel MPI Benchmark (IMB) tools installation that is common to all systems under test.

### 12.9.1 CLUSTER SETUP

- 1) Network configuration requirements
  - a) All systems must be reachable by each other over IPoB.
  - b) All nodes must agree on the IPoB IP addresses of 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) The same version of the Intel MPI Benchmark (IMB) tools must be installed in the same filesystem location on all systems under test.
  - a) IMB can be used from the OFED installation or, if a later version of Open MPI is to be used, IMB can be downloaded from Intel's web site:  
<http://software.intel.com/en-us/articles/intel-mpi-benchmarks/?wapkw=intel%20mpi%20benchmarks>
- 4) The same version of Open MPI must be available in the same filesystem location on all systems under test.
  - a) 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/>
    - i) 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.
    - ii) 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.
    - iii) 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.

- iv) **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 file-system locations (i.e., configured with a different `--prefix` argument). All MPI tests must be built and run with a single installation of Open MPI.
- v) Ensure that the Open MPI installation includes OpenFabrics support:  

```
shell$ $MPIHOME/bin/mpi_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.
- b) 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 -ssh -np $NP --hostfile $HOSTFILE host-  
name
```

  - i) The output should show the hostname of each host listed in the hostfile; the hostname should appear as many times as there are lines in the hostfile. The list of hostnames may appear in random order; this is normal
  - ii) Note that any serial application can be run; "hostname" is a good, short test that clearly identifies that specific hosts were used
- 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 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.).
- 6) The lockable memory limits on each machine should be set to allow unlimited locked memory per process.
- 7) The underlying OpenFabrics network used in the test should be stable and reliable.
- 8) No other fabric interoperability tests should be running during the Open MPI tests.
- 9) MPI tests should be run across at least 5 separate systems to force the use of the OpenFabrics network (vs. using just shared memory for in-system communication).

## 12.9.2 TEST SETUP

- 1) Create a hostfile (\$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 systems named node1.example.com and node2.example.com that are each able to run 4 processes:

```
shell$ cat 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

- 2) Determine the number of Open MPI processes (\$NP) that are to be run determined by the number of host entries in the created hostfile.
- 3) Open MPI defaults to probing all available networks at run-time to determine which to use. OpenFabrics testing must 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). To do this add an extra command line parameter; both iWarp and InfiniBand:  
`--mca btl openib,self`
- 4) It has been discovered that the following Open MPI command line parameter is required to facilitate multi RDMA adaptor vendor MPI rings; both iWarp and InfiniBand:  
`--mca pml ob1 --mca btl_openib_flags 306`
- 5) It has been discovered that the following Open MPI command line parameter is required to facilitate multi RNIC adaptor vendors MPI rings; iWarp specific:  
`--mca btl_openib_receive_queues P,65536,256,192,128`

### 12.9.3 TEST PROCEDURE

- 1) Create a hostfile listing the MPI ring nodes, process distribution, and total number of processes to use as indicated in steps 1 and 2 of section 12.11.2. The filesystem location of this hostfile is irrelevant.
- 2) Locate the "mpirun" binary that will be used. This determines the version of Open MPI that will be used.
- 3) Locate the "IMB-MPI1" IMB binary. This must have been built against the version of Open MPI selected above. If using an OFED distribution this build process has already been performed.
- 4) Verify that a subnet manager has configured the fabric. If not, start one.
- 5) Verify that all hosts present within the hostfile are online and accessible.
- 6) Run the IMB-MPI1 benchmarks
- 7) Repeat steps 4-6 using a different subnet manager until all subnet managers under test have been used.
- 8) All IMB benchmarks must pass successfully using all subnet managers under test in order for the devices under test defined within the hostfile pass.

#### 12.9.4 METHOD OF IMPLEMENTATION FOR ALL LINUX OS'S

- 1) To perform step 4 of section 12.9.3 use "ibdiagnet -r" from a host defined in the mpi hostfile and look for an "SM - Master" entry in the output
- 2) To perform step 5 of section 12.9.3 ping the IPoIB address of all hosts defined in the mpi hostfile from a host defined in said hostfile.
- 3) To perform step 6 of section 12.9.3 use the following command from a host that can access all hosts defined within the hostfile; this host can be part of the hostfile
  - a) For **InfiniBand**:  
\$MPIHOME/bin/mpirun --mca btl openib,self,sm --mca pml ob1 -mca btl\_openib\_flags \ 306 -np \$NP -hostfile \$HOSTFILE \$IMBHOME/IMB-MPI1
  - a) For **iWarp**:  
\$MPIHOME/bin/mpirun --mca btl openib,self,sm --mca pml ob1 --mca \ btl\_openib\_flags 306 --mca btl\_openib\_receive\_queues P,65536,256,192,128 -np \ \$NP -hostfile \$HOSTFILE \$IMBHOME/IMB-MPI1

## 12.10 TI MPI - OHIO STATE UNIVERSITY USING OFED

### 12.10.1 MVAPICH - 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) MVAPICH is included in OFED distributions. The updated versions of MVAPICH 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) MVAPICH 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 MVAPICH 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 MVAPICH 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 MVAPICH 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 5 separate systems to force the use of the IB networks (vs. using just shared memory for in-system communication).
- Note:** MVAPICH is commonly referred to as MVAPICH1 to distinguish it from the new and updated MVAPICH2

### 12.10.2 MVAPICH - 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_rsh -ssh -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\_rsh -ssh -np \$NP IMB-MPI1 -multi 0 -hostfile  
\$HOSTFILE

### 12.10.3 MVAPICH2 - 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.
- 6) MVAPICH tests should be run across at least 5 separate systems to force the use of the IB networks (vs. using just shared memory for in-system communication).

### 12.10.4 MVAPICH2 - TEST PROCEDURE

**Step A:** [For IB] 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  
**[For IB]**
  - a) `mpirun_rsh -ssh -np <number of nodes X number of processors/node>  
IMB-MPI1 -multi 0 PingPong PingPing`
  - b) `mpirun_rsh -ssh -np <number of nodes X number of processors/node>  
IMB-MPI1`**[For iWARP]**
  - a) `mpirun_rsh -ssh -np <number of nodes X number of processors/node>  
MV2_USE_IWARP_MODE=1 MV2_USE_RDMA_CM=1 IMB-MPI1 -  
multi 0 PingPong PingPing`
  - b) `mpirun_rsh -ssh -np <number of nodes X number of processors/node>  
MV2_USE_IWARP_MODE=1 MV2_USE_RDMA_CM=1 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) **[For IB]** If the test passes shutdown current subnet manager and start another one on a different node; run both tests again.
- 3) **[For IB]** 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 Setup

**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 Win

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. OF node to run diagnostics for this test.

2) Verify that no SM is running

3) Connect two devices back to back

4) 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.

5) Repeat steps 1-3 with a different device pairing.

a) All device pairs must be tested except SRP target to SRP target.

i) HCA to HCA

ii) HCA to Switch

iii) HCA to Target

iv) Switch to Switch

v) Switch to Target

**Note:** HCA to Target and HCA to HCA cannot be tested under WinOF 2.0.2 because there are no utilities available. Switches can be tested by using a Linux Host and the OFED Utilities.

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.

- 1) Design and implement a Cluster Topology.
- 2) End to end IPoIB connectivity is required between all end points. Therefore you must create and assign IP addresses to each IB end point.
- 3) See [Figure 6- Sample Network Configuration](#) below.

### 13.2.2 Procedure

- 1) Connect the HCAs and switches as per the Architected Network and make sure that no SM/SA is running on the Fabric.
- 2) Start an SM on a device and let it initialize (all SMs will need to be tested)
- 3) Visually verify that all devices are in the active state using LEDs (however the vendor decided to implement it).
- 4) The following steps must be done using a Linux OFED end point.
  - a) Run "ibdiagnet -pc" to clear all port counters
  - b) Wait 17 seconds as per the specifications requirements.
  - c) Run "ibdiagnet -c 1000" to send 1000 node descriptions.
  - d) Run "ibdiagnet" to generate fabric report and open report to see results. /tmp/ibdiagnet.sm
  - e) Run "ibchecknet" to build guid list.

### 13.2.3 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

**Note:** the reports are located in the /tmp directory

Restart all devices in the fabric and follow Sections 13.2.2 and 13.2.3. 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.

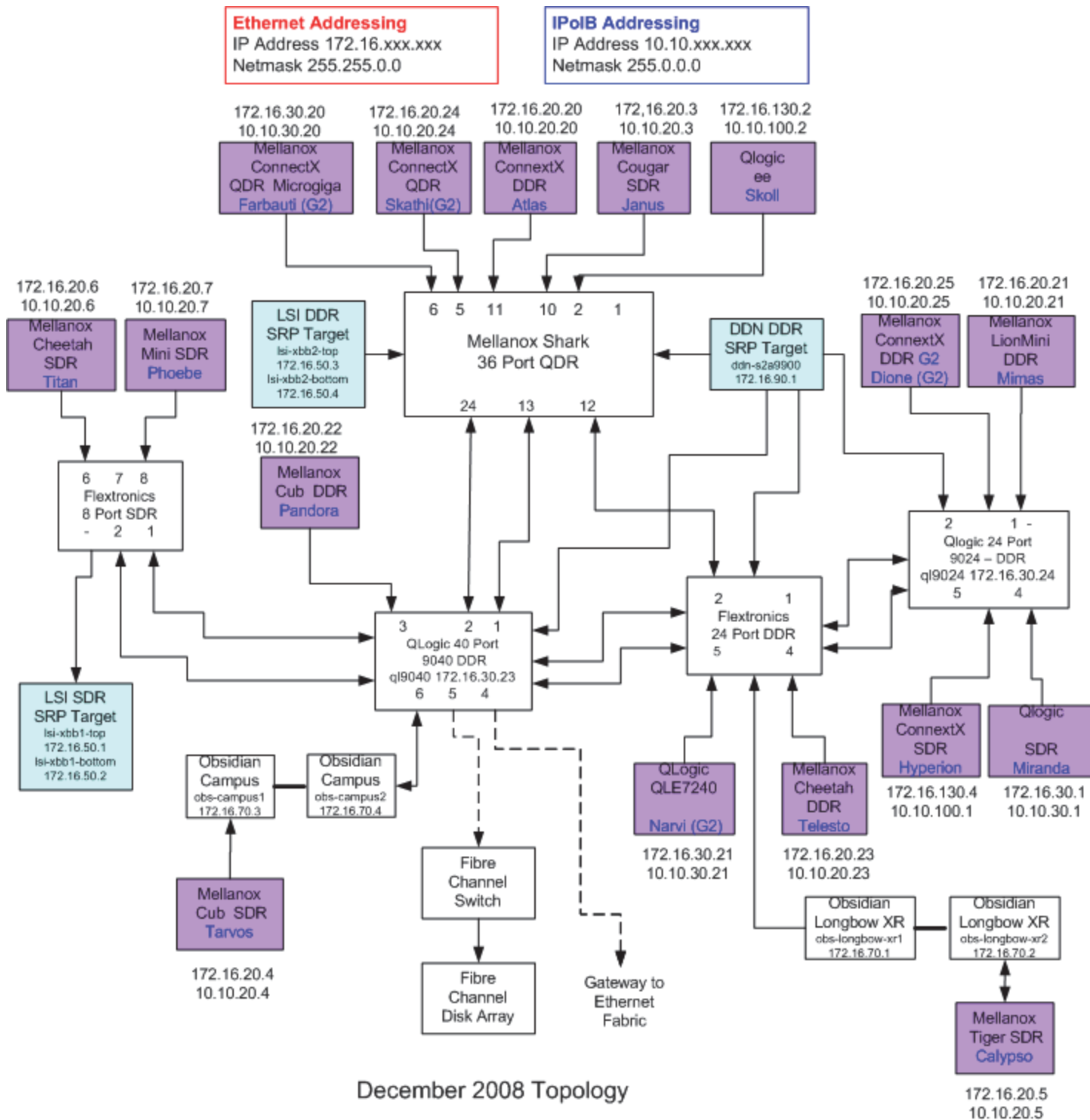
Each device must pass all verification procedures with every SM to pass Fabric Initialization test.

**Table 30 - ibdiagnet commands**

Commands	Description
ibdiagnet -c 1000	send 1000 Node Descriptions
ibdiagnet -h	Help
ibdiagnet -lw 4x - ls 2.5	Specify link width and speed
ibdiagnet - pc	Clear Counter
ibdiagnet -t <file>	Compare current topology to saved topology
ibdiagnet -wt	Writes the topology to a file

**Note:** The topology file is being generated after the SM starts but before any testing has started. The topology comparison is being performed after testing has been completed but before the systems get rebooted. A topology check is performed during every part of every test section that does not specifically state "change the topology". For example Fabric Init only has 1 part so there is only 1 check but RDS has 2 parts so 2 checks are performed. However, IPoIB has 3 parts for each of 2 modes but 1 of those parts specifically says to change the topology so only 4 checks occur.

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" 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) Issue the command: sysctl net.ipv4.neigh.ib0.unres\_qlen=18
  - a) This sets the qlen variable to 18 which increases the buffer size so that you do not get an initial dropped packet when using ping sizes 8192 and greater.
- 6) Ping every IPoIB interface IPv4 address except localhost with packet sizes of 64, 256, 511, 512, 1024, 1025, 2044, 4096, 8192, 16384, 32768, and 65500. 'ping /?' displays ping help.
  - a) 10 packets of each size will be sent
  - b) Every packet size is a new ping command.

**Note:** Windows does not support 65507 so we used 65500.

**Note:** This is done from the Head Node utility "Run a Command" using the following command:

```
for %i in  
(64,256,511,512,1024,1025,2044,4096,8192,16384,32768,65500) DO  
%d arp -d %d & ping -i 1 -n 10 -l %i %d & arp -d %d
```

- 7) 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. 1
  - 5) Ping to the newly disconnected HCA from all nodes (No response should be returned). 2
  - 6) Connect the first machine (the one that was not connected) and check for ping response from all nodes that are still connected. 3
  - 7) Connect the disconnected HCA to a different switch on the subnet which will change the topology. 4
  - 8) Ping again from all nodes (this time we should get a response). 5
  - 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. 6
- Note:** Each step must exhibit the expected behavior while using each SM in order for the device to pass Step B overall. 7

### Step C 8

- 1) 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. 9
- 2) Issue the command: sysctl net.ipv4.neigh.ib0.unres\_qlen=3 10
- a) This sets the qlen variable back to the default. 11

### 13.3.4 FTP PROCEDURE 12

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. 13

An FTP server is a component of the IIS '**Internet Information Services**' manger which **not** a part of a standard Windows installation: 14

See Start | Server Manager | Roles | Add IIS. Configure FTP server via IIS manager. 15

#### 13.3.4.1 SETUP 16

- 1) Make sure ftpd is installed on each node for the FTP application. 17
- 2) A special account for this should be created as follows: 18
- b) Username: Interop 19
- c) Password: openfabrics 20

#### 13.3.4.2 PROCEDURE 21

Run FTP server on all nodes. 22

- 1) Start an SM (all SMs will need to be tested) and let it initialize (ref MS Network utilities docs) 23
- a) Verify that the running SM is the one you started. 24
- 2) FTP: 25



- a) Connect an HCA pair via FTP on IPoIB using the specified user name and password. 1
- b) Put the 4MB file to the %windir%\temp folder (generally C:\Windows\Temp) on the remote host. 2
- c) Get the same file to your local dir again. 3
- d) Binary compare the file using the Windows command 'fc /B tfile tfile.orig'. 4
- i) The two must be identical. 5
- 3) Repeat the procedure with a different SM. 6
- 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. 7
- 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. 8

## 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. 1
  - 15) Verify that the previous master SM is still the master; view 2  
'%windir%\temp\osm.log'. 3
  - 16) Verify that all nodes in the cluster are present - ping all IPoIB interfaces 4
  - 17) Repeat proceeding steps [1-16] 2 more times with the same node pair, en- 5  
suring that the below criteria is met (total of 3 tests per pair which can be run 6  
in any order): 7
    - a) First SM to be started having highest numerical priority value. 8
    - b) Second SM to be started having highest numerical priority value. 9
    - c) Both SMs having equal numerical priority values. 10
  - 18) Repeat steps 1-17 until all possible SM pairs from identical vendors in the 11  
cluster have been tested. 12
- 13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42

## 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 1
- g) fc /B T:\tmp\WinOF\_wlh\_x64.msi T:\tmp\WOF2.msi 2
- h) fc /B WinOF\_wlh\_x64.msi T:\tmp\WOF2.msi 3
- i) copy /B T:\tmp\WOF2.msi WOF3.msi 4
- j) fc /B WinOF\_wlh\_x64.msi WOF3.msi 5
- k) del T:\tmp\WOF2.msi 6
- l) del T:\tmp\WinOF\_wlh\_x64.msi 7
- m) dir T:\tmp (should be empty) 8
- n) rmdir T:\tmp 9
- o) dir T:\ (should be empty) 10
- p) del WOF3.msi 11
- 10) For each SRP target located in Procedure #4 12
- a) Repeat step 9 for all volumes found for all targets as determined by Win- 13
- dows Procedure step #4 - see [Discover + Enable \(bring online\) the SRP](#) 14
- [drive\(s\)](#) 15
- 11) Take SRP drive offline 16
- a) Start | Server Manager | Storage | Disk Management 17
- b) Right-click the online disk and select 'offline' 18
- c) dir T:\ (should fail). 19
- 12) Reboot all devices in the fabric and repeat the procedure using a different 20
- SM. 21
- Note:** An HCA must successfully complete all operations to and from all volumes 22
- on all targets using all available SM's in order to pass SRP testing. One volume 23
- per target is all that is required. 24

## 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 <server\_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 verify 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 255.255.255.0This 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
    - `mpiexec -np <number of nodes X number of processors/node> IMB-MPI1 -multi 0 PingPong PingPing`
    - `mpiexec -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, 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 email by UNH-IOL to the focal point assigned by the 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 are 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			

**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			

**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 RDMA Basic 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 RDMA Basic 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 Stress Tests**

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

## 15.4 INTEL MPI TEST RESULTS

Results Table 21a - **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 21b - 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 21b - 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 21c - 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)			

## 15.5 OPEN MPI TEST RESULTS

**Results Table 22 - TI MPI - Open MPI**

Test #	Test Suite	Pass	Fail	Comment
<b>Phase 1: "Short" tests</b>				
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)			
<b>Phase 2: "Long" tests</b>				
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 22 - 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 22 - TI MPI - Open MPI**

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

## 15.6 OSU MPI TEST RESULTS

Results Table 23 - 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 24 Remarks

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