OFA Interoperability Working Group

OFA-IWG Interoperability Test Plan
Release 1.16

OPENFABRICS
ALLIANCE
iol TESTED
www.iol.unh.edu

February 25, 2008
DRAFT

OpenFabrics.org

Revision History

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

Revision	Release Date	
1.11	Sep 7, 2007	Updated with the latest scripts developed by UNH IOL and based on the results from the April 2007 Interop Event
1.12	Sep 12, 2007	Updated the douments to embed the test scripts in the document.
1.13	Jan 22, 2008	Updated the douments for the March 2008 OFA Interop event. IPoIB updated along with Cover Page and the Test Requirements section.
1.14	Feb 11, 2008	Added the following tests: 1. Ethernet Switch Tests 2. IPoIB Connected Mode 3. RDMA Interop 4. RDS
1.15	Feb 18, 2008	Updates to the following tests: 1. Ethernet Switch Tests 2. IPoIB Connected Mode 3. RDMA Interop
1.16	Feb 25, 2008	Removed all reference to Low Latency Ethernet Switches. This is the version for the March 2008 Interop Event

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 IBTA, specifically within the charter of the CIWG. And for iWARP it is IETF, and specifically within UNH IOL iWARP Consortium.

1.1 Purpose

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

1) Define the scope, equipment and software needs, and test procedures for verifying full interoperability of RDMA HW and SW. For Infiniband HW it is InfiniBand HCAs using the latest OpenFabrics IB OFED software with currently available OEM Switches and their management software. The target OEM IB Switch vendors are Cisco, Flextronics, 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 OVERVIEW

The tables below list all required tests for the procedures

Table 1 - IB Link Initialize

Test #	Test	Description Overview
1	Phy link up all ports	Check that all relevant green LEDs are on for all HCAs and switches.
2	Logical link up all ports switch SM	All vendors should check that the link state is up and the port width is 4X.
3	Logical link up all ports HCA SM	All vendors should check that the link state is up and the port width is 4X.

Table 2 - IB Fabric Initialization

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

Table 3 - IB IPoIB - Connect Mode (CM)

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

Table 4 - IB IPoIB - Datagram Mode (UD)

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

Table 5 - Ethernet Link Initialize

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

Table 6 - Ethernet Fabric Initialize

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

Table 7 - Ethernet Fabric Reconvergence

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

Table 8 - Ethernet Fabric Failover

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

Table 9 - TI iSER Tests

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

Table 10 - IB SRP Tests

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

Table 11 - TI SDP Tests

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

Table 12 - IB SM Tests

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

Table 13 - TI MPI - OSU

Test #	*Test*	*Description Overview*
1	**Test** 1: PingPong	
2	**Test** 1: PingPing point-to-point	
3	**Test** 2: PingPong	
4	**Test** 2: PingPing	
5	**Test** 2: Sendrecv	

Table 13 - TI MPI - OSU

Test #	Test	Description Overview
6	**Test** 2: Exchange	
7	**Test** 2: Bcast	
8	**Test** 2: Allgather	
9	**Test** 2: Allgatherv	
10	**Test** 2: Alltoall	
11	**Test** 2: Reduce	
12	**Test** 2: Reduce_scatter	
13	**Test** 2: Allreduce	
14	**Test** 2: Barrie	

Table 14a - TI MPI - Intel MPICH2 Suite - (Not part of OFA Stack)

Test #	MPICH2 (16 sections, 290 tests)	Intel - MPICH2 Test Suite Section Description
1	attr	Test programs for attribute routines
2	coll	Test programs for various collective operations
3	comm	Test programs for communicator operations
4	datatype	Test programs for various datatype operations
5	errhan	Test programs for error handling operations
6	group	Test programs for the group operations
7	info	Test programs for varios 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
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 treaded send/recv

Table 14b - TI MPI - Intel MPI Test Suite Description - (Not part of OFA Stack)

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

Table 15 - TI uDAPL

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

Table 16 - iWARP Connections

Test #	*Test*	*Description Overview*
1	UNH iWARP interop tests group 1	Verify that each single iWARP operation over single connection works.
2	UNH iWARP interop tests group 2	Verify that multiple iWARP operations over a single connection work.
3	UNH iWARP interop tests group 3	Verify that multiple iWARP connections work.
4	UNH iWARP interop tests group 4	Verify that disconnect/reconnect physical connections work.

Table 16 - iWARP Connections

Test #	*Test*	*Description Overview*
5	UNH iWARP interop tests group 5	Verify that IP Speed negotiation work.
6	UNH iWARP interop tests group 6	Verify that iWARP error ratio work.
7	UNH iWARP interop tests group 7	Verify that stress pattern over iWARP work.
8	UNH iWARP interop tests group 8	Verify that iWARP parameter negotiation work.

Table 17 - Fibre Channel Gateway - (IB Specific)

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

Table 18 - Ethernet Gateway - (IB Specific)

Test #	*Test*	*Description Overview*
1	Basic Setup	Connect the HCA of the IB host and Ethernet Gateway to the IB fabric. Connect the Ethernet gateway to the Ethernet network or Ethernet device. Start the SM to be used in this test.
2	Start ULP	Determine which ULP your ethernet gateway uses and be sure that ULP is running on the host.

Table 18 - Ethernet Gateway - (IB Specific)

Test #	Test	Description Overview
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 19 - Reliable Datagram Service (RDS)

Test #	Test	Description Overview
1	qperf procedure	Run qperf nodex -t 20 rds_bw and verify non-zero bandwidth number

Table 20 - Basic RDMA Interop

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

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

1.4 SUBJECTS NOT COVERED

Table 22 - Subjects Not Covered

Number	*Subject/ Feature*	*Reason*	*Executor*	*Due Date*
1	NFS-RDMA	Future Testing		March 2008
3	OpenMPI	Future Testing		March 2008

1.5 TEST REQUIREMENTS FOR OFA-UNH-IOL_LOGO_PROGRAM

The following table indicates the mandatory tests to qualify for the OFA-UNH-IOL Logo Program during the March 2008 Interoperability Event. It is anticipated that the Beta tests will be moved to Mandatory status for the following Interop Event.

Table 23 - Test Status for March 2008 Interop Event

Test Procedure	***InfiniBand Devices***	***iWARP Devices***	***Transport Independent***
IB Link Initialize	**Mandatory**	Not Applicable	
IB Fabric Initialization	**Mandatory**	Not Applicable	
IB IPoIB Datagram Mode	**Mandatory**	Not Applicable	
IB IPoIB Connected Mode	**Beta**	Not Applicable	
TI iSER	**Mandatory**	**Beta**	
IB SRP	**Mandatory**	Not Applicable	
TI SDP	**Mandatory**	**Beta**	
IB SM Failover/Handover	**Beta**	Not Applicable	
TI MPI - OSU			**Beta**
TI MPI Intel			**Beta**
TI uDAPL			**Beta**
iWARP Connectivity	Not Applicable	**Mandatory**	
Fibre Channel Gateway (IB)	**Beta**	Not Applicable	
Ethernet Gateway (IB)	**Beta**	Not Applicable	
Ethernet Link Initialize	Not Applicable	**Beta**	
Ethernet Fabric Initialize	Not Applicable	**Beta**	
Ethernet Fabric Reconvergence	Not Applicable	**Beta**	
Ethernet Fabric Failover	Not Applicable	**Beta**	
IB Reliable Datagram Sockets	**Beta**	Not Applicable	
TI Basic RDMA Interop			**Beta**
TI RDMA Operations over Interconnect Components			**Beta**

1.6 Test Glossary

Table 24 Test Glossary

Technical Terms	
HCA	IB Host Channel Adapter.
TD	Test Descriptions.
SM	IB Subnet Manager.
RDF	Readme File.
SA	IB Subnet Administration.
TI	Transport Independent (tests).
RNIC	RDMA NIC (iWARP Network Interface Card).

2 General System Setup

Configuration

The test environment for the user interface contains:

2.1 IB HW Units

Table 25 IB Equipment

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

2.2 IB Software

2.2.1 Linux/Windows platforms

2.2.2 OFED - Most Current Tested Release

2.2.3 IB HCA FW – Version XXX - Vendor Specific

2.2.4 IB Switch FW candidate – Version XXX - Vendor Specific

2.2.5 IB Switch SW – Version XXX - Vendor Specific

2.3 iWARP HW Units

Table 26 iWARP Equipment

Equipment	Amount	Details	Check
Operating System	4 or more	The OS should be supported by OpenFabrics.	
10GbE Cables	10		
10GbE Switch from a 3rd Party Vendor	1 or more		
10GbE Analyzer	1		
RNICs	4 or more		

2.4 iWARP SOFTWARE

2.4.1 LINUX/WINDOWS PLATFORMS

2.4.2 OFED - MOST CURRENT TESTED RELEASE

2.4.3 iWARP RNIC FW – VERSION XXX - VENDOR SPECIFIC

2.4.4 10GBE SWITCH FW CANDIDATE – VERSION XXX - VENDOR SPECIFIC

2.4.5 10GBE SWITCH SW – VERSION XXX - VENDOR SPECIFIC

3 Use of OpenFabrics Software for Pre-Testing

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

4 Use of OpenFabrics Software for IBTA/CIWG Interoperability Plugfest

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

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

5 Use of OpenFabrics Software for UNH IOL iWARP Interoperability PlugFests

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

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

6 IB HW DESCRIPTION & CONNECTIVITY

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

6.1 BASIC CONNECTIVITY (P1P1)

6.1.1 HCA 1 SHOULD BE CONNECTED FROM PORT 1 TO LOWEST PORT NUMBER IN SWITCH

6.1.2 HCA 2 SHOULD BE CONNECTED FROM PORT 1 TO HIGHEST PORT NUMBER IN SWITCH

6.1.3 BOTH WITH 4X 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.

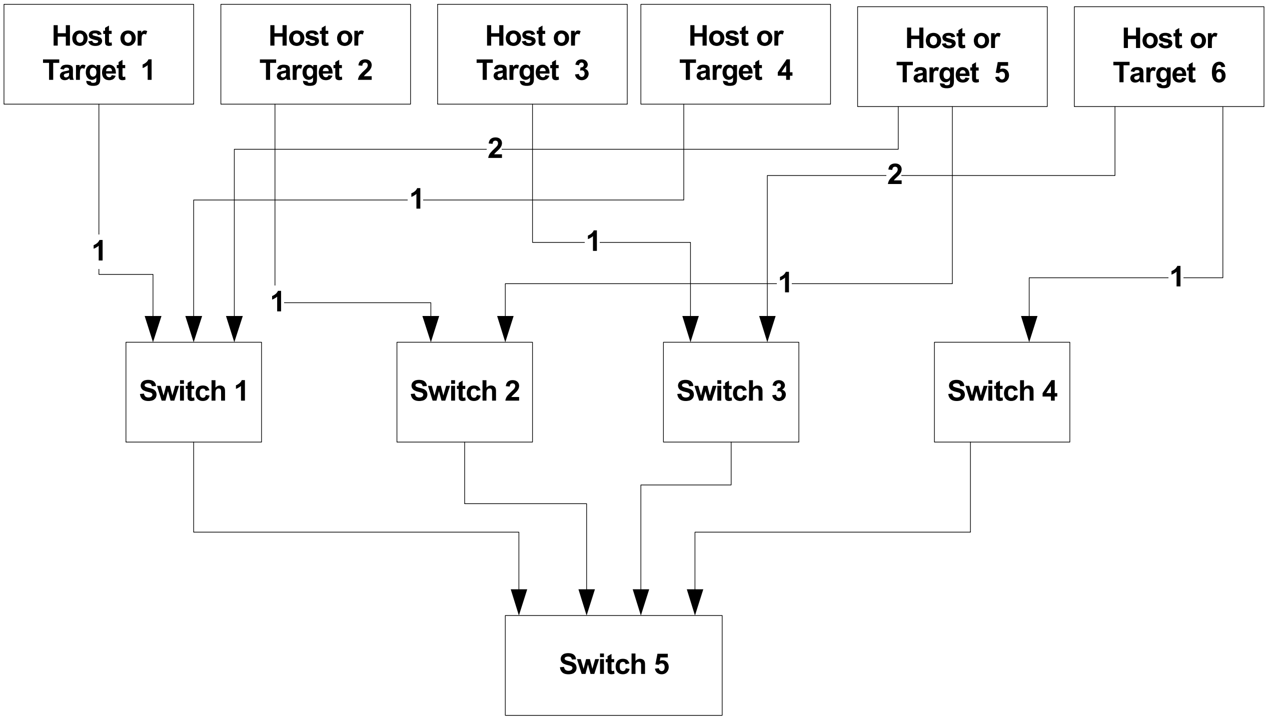
6.2.2 OPENFABRICS SOFTWARE RUNNING ON HOSTS

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 4X INFINIBAND CABLES.

Figure 1 Example IB Interop Setup



7 IWARP HW DESCRIPTION & CONNECTIVITY

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

7.1 IWARP BASIC CONNECTIVITY (P1P1)

7.1.1 RNIC 1 ON ONE HOST SHOULD BE DIRECTLY CONNECTED TO RNIC 2 ON ANOTHER HOST

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.

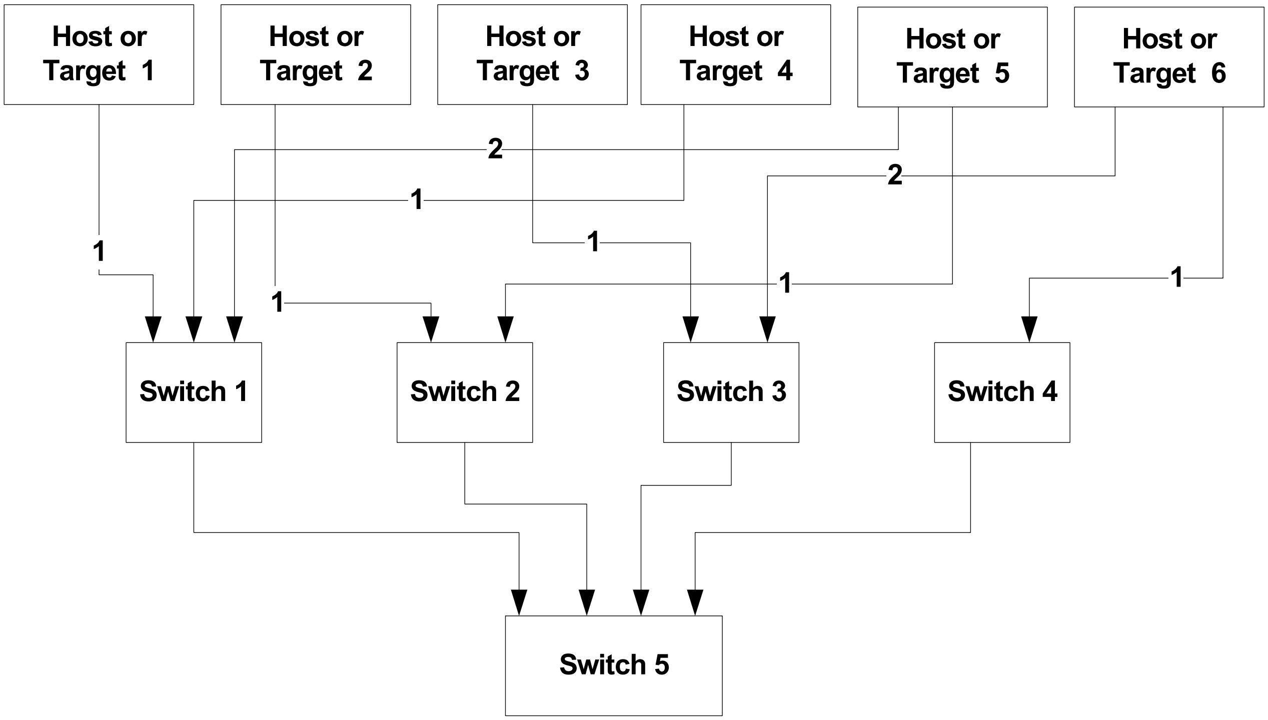
7.2.2 OPENFABRICS SOFTWARE RUNNING ON RNICS

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 1-4 SHOULD BE CONNECTED TO SWITCHES USING 10GBE CABLES.

Figure 2 Example iWARP Interop Setup



7.4 GATEWAY, BRIDGES, ROUTERS CONNECTIVITY

TBD

8 SW & HW installation

8.1 Burning the FW

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

8.1.2 No firmware burning required for iWARP vendor

8.2 SW installation

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

8.2.2 Please refer to software installation manual from RNIC vendor.

9 General Instructions

9.1 First step Instructions

1) Burn the FW release XXX on all HCAs and RNICs using the above procedure as required by vendor.
2) Host and Target Configuration
   a) Install OFED software on host systems (using a 64 bit OS) configured to run OFED.
   b) Configure non-OFED systems for use in the cluster as per the vendors instructions.
   c) 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 Interop Procedures

10.1 IB Link Initialize

10.1.1 Connect the 6 HCAs (Port 1) to the switches as shown in the Cluster Connectivity Section. Cable length should be a maximum of 17 meter for SDR and 10 meters for DDR.

1) It is suggested that all switches be connected to one power strip to make rebooting easier.

2) Switches should also be located in between the servers.

10.1.2 Turn off the SM on all devices.

10.1.3 Check that all relevant green LEDs are on (Not blinking) for all HCAs and switches. All vendors should check that the link state is up and the port width is 4X.

10.1.4 Repeat Section 10.1.3 and verify that each HCA is able to link to the other HCAs in the fabric and also to all switches.

10.2 IB FABRIC INITIALIZATION

10.2.1 Architect the Network we want to build.

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

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

10.2.3 Run the SM/SA on one of the devices to perform device discovery, then drive all the ports through Armed and Active states.

1) **Optional**: Use a protocol analyzer to verify SMP transaction between ports as well as to verify final port states:
   a) For Channel Adapters, check that PortInfo:PortState=Active.
   b) For Switches check that either PortInfo:PortState=Armed or PortInfo:PortState=Active.
2) ibdiagnet can be used when running openSM on an HCA.
   a) Clear counters - ibdiagnet -pc.
   b) Send 100 Node Descriptions - ibdiagnet -c 1000.

10.2.4 Verification Procedures

1) Port error counters (in PMA PortCounters) are validated to ensure that there are no ongoing link errors. The Specification says there should be no errors in 17 seconds.
2) There should be no bad port counters - must be zero.
3) No duplicate GUIDs.
4) SM verification
   a) Verify that the SM running is the one you specified. Check /tmp/ibdiagnet.sm.
   b) Verify number of nodes and switches in the network.

Restart all devices in the fabric and follow Sections 10.2.3 through 10.2.4 and each time run the SM/SA from a different component in the system switch/HCA.

Table 27 - ibdiagnet commands

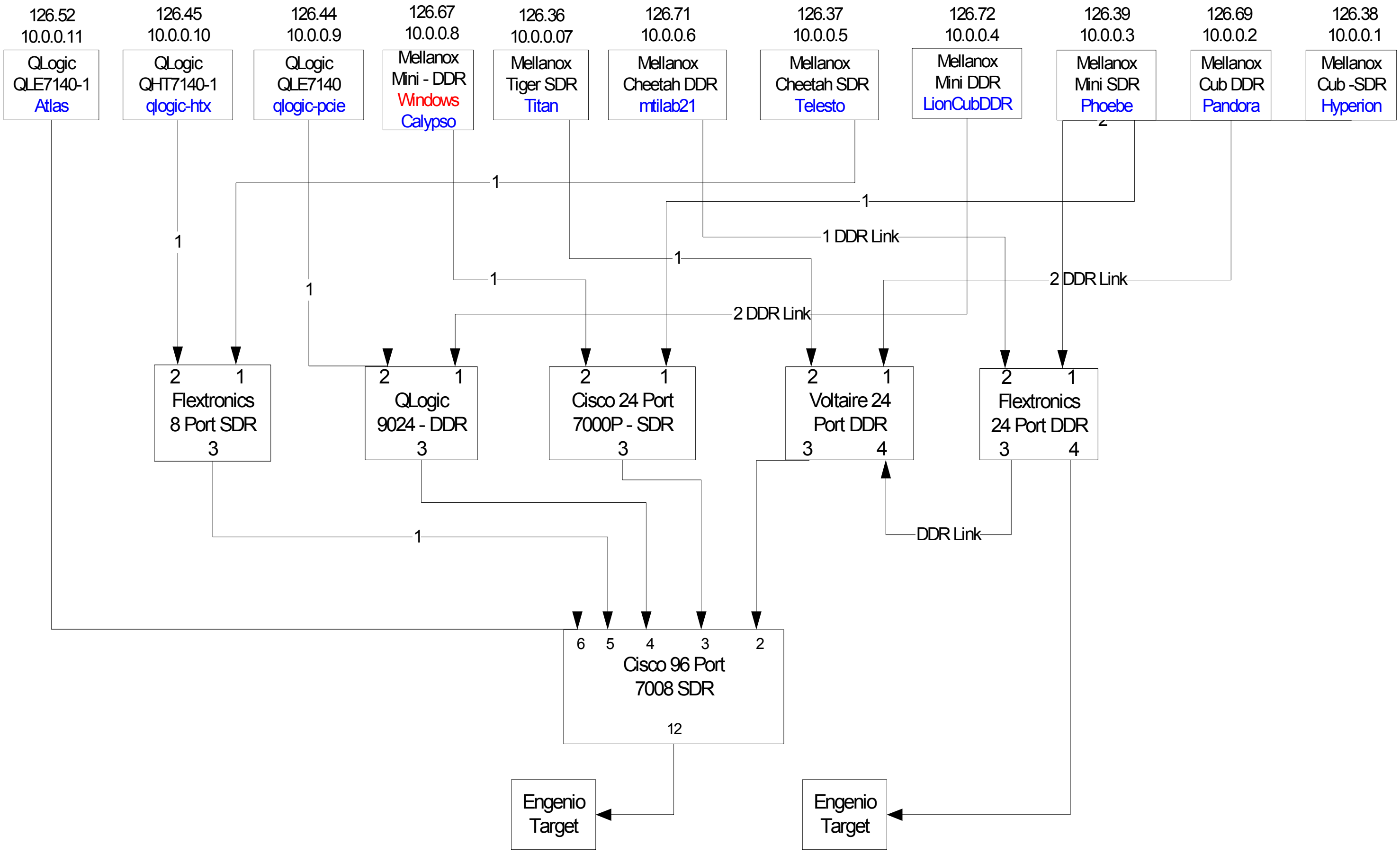
Commands	Description
ibdiagnet -h	Help
Ibdiagnet - pc	Clear Counter
Ibdiagnet -lw 4x - ls 2.5	Specify link width and speed
Ibdiagnet -c 1000	send 1000 Node Descriptions

Figure 3 - Sample Network Configuration

Legend:
132.177.126.xx - Ethernet Address
10.0.0.xx IPoIB Address

Ethernet Addressing
IP Address 132.177.126.xxx
Netmask 255.255.255.0

IPoIB Addressing
IP Address 10.0.0.xx
Netmask 255.255.0.0



Devices
8 Mellanox HCAs
3 QLogic HCAs
1 Cisco 24 Port switch - managed
1 Cisco 96 port switch - managed
1 Flextronics 24 port switch - un-managed
1 Flextronics 8 port switch - un-managed
1 QLogic 24 port switch - managed
1 Voltaire 24 port switch - managed
2 Engenio Targets

Device Rotation
Cisco - switch 24
Cisco - switch 96
QLogic - switch 24
Voltaire - switch 24
Mellanox HCA
QLogic HCA

InfiniBand Test Cluster

Configure IPoIB Addressing
ifconfig ib0 10.0.0.x netmask 255.255.0.0

10.3 IB IPoIB Connect Mode (CM)

10.3.1 Setup

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 if a corresponding connected HCA does not support this capability, it can run 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).

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

A single IPoIB subnet is reserved for Plugfest IPoIB testing. This subnet to be setup on the full default partition (0xFFFF). Its IPoIB address is 10.0.0.x/24 (10.0.0.x/netmask 255.255.255.0).

Once the IPoIB interfaces are configured on all partner HCA ports, the following procedures will be performed. The default IPoIB MTU of 2048 will be used.

The ability for each partner to create the all-IPoIB nodes IB multicast group, if observable, as well as to join that multicast group is tested.

In some configurations, when the SM/SA is local to the IPoIB implementation, not all operations will be observable with the IB analyzer (when the side with the SM/SA creates the IPoIB broadcast group). Additionally, with some SM/SAs, the creation of the IPoIB broadcast group may be previously administered and hence would not be observable by an IB analyzer.

mo In addition, the procedure will test the SM/SA ability to support the following functions:

1) SA in terms of performing the multicast group creation and joining.
2) SM in terms of programming the multicast topology (MulticastForwardingTable) in any switches.

The various parameters of the MCMemberRecord will be validated. In general, it will be checked that the group creator characteristics (Q_Key, etc.) are returned to the subsequent group joiners.

10.3.3 Bringing the IPoIB in Connected Mode

1) Set "SET_IPOIB_CM=yes" in file /etc/infiniband/openib.conf on all nodes being tested.
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**'
   a) For IPoIB CM: /sys/class/net/ib0/mode = "connected"
   a) For IPoIB UD: /sys/class/net/ib0/mode = "datagram"

10.3.4 Ping Procedures

Step A

1) Assign IP Addresses using the command *ifconfig ib0 10.0.0.x netmask 255.255.255.0*
2) Turn off SMs. Use ibdiagnet to verify that the master SM is missing.
3) Power cycle all switches.
   a) This insures that the new SM will configure all the links and create the multi-cast join.
   b) Run ibdiagnet to verify that all nodes have come up. Ibdiagnet does not require the SM to discover the node.
4) Use ibdiagnet to determine 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.
5) Run SM/SA from one of the nodes in the cluster.
   a) Verify that the new SM is the master. You will need to know the GUID of the device since the SM will be reassigned on each reboot.
6) Pings (ICMP requests) of the following lengths will be performed from each node (All to all): first in one direction, then the other, and finally bidirectional: 64, 256, 511, 512, 1024, 1025, 2044, 4096, 8192, 16384, 32768 and 65507. The count is 100.

   Note: In the above, the lengths of the IP (20 bytes for IPoIB Encapsulation) and IB headers are included although they will need to be subtracted out on the actual invocation of the ping command. It is also unknown whether the standard ping application without modification will allow all the lengths specified above.
7) At the completion of each different ping invocation, the arp table should be locally examined (via arp -a) and then the partner should be removed from the arp table (via arp -d) prior to starting the next ping invocation.

An IB trace of this should be examined to make sure that:

1) ARP is resolved properly (both ARP request and response are properly formatted).
2) Proper fragmentation (at the IB level) is occurring.

   Note: the case of length of 65536 ("ping of death" or long ICMP) exceeds the maximum IP length and no response is expected for this case.

   Note: At the completion of each different ping invocation, the arp table should be locally examined (via arp -a) and then the partner should be removed from the arp table (via arp -d) prior to starting the next ping invocation.

Step B

1) Bring up all HCAs but one.
2) Check for ping response between all players.
3) Disconnect one more HCA from the cluster (you should see that the ping stopped).

4) Ping to the newly disconnected HCA from all nodes (No response should be returned).
5) Connect the first machine (the one that was not connected) and check for ping response.
6) Connect the disconnected HCA to a different switch on the subnet which will change the topology. Check for ping response.
7) Ping again from all nodes (this time we should get a response).
8) Follow steps 1 to 7, this time bring the interface down and then back up using ifconfig ibX down and ifconfig ibX up commands.

Step C

Follow Step A and B running the SM/SA from each device in the cluster (If all HCAs have the same SW no need to test more than one HCA/node).

10.3.5 FTP 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.

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.

Step A

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

Step B

Run SFTP server on all nodes.

1) For each node:
   a) Connect via SFTP on IPoIB using the specified user name and passwd.
   b) Put the 4MB file to the /tmp dir on the remote host * 4 times.
   c) Get the same file to your local dir again 4 * times.
   d) Compare the file using the command *cmp tfile tfile.orig.*

10.3.6 AUTOMATED TEST SCRIPT

```
#!/bin/sh
# -----------------------------------------------------------------------------
#
# ipoibsftp.sh
#
# This script automates the sftp procedure of the OFA Test Plan.
# This script takes two arguments, the IP Address of the remote
# machine (in dot notataion) and the number of times to loop the
# sftp procedure.
#
# Copyright (c) 2008 The University of New Hampshire InterOperability Laboratory
#   All Rights Reserved
#
# Author: Dustin M. Schoenbrun
#
# -----------------------------------------------------------------------------


# Variable declarations
USAGE="Usage: ./ipoibsftp.sh <IP Address> <Loop Count>"
FAIL=" ipoibsftp.sh FAIL"
CP="/bin/cp"
ADDR=$1
COUNT=$2

# Check the number of arguments to make sure that there are 2
if [ $# -ne 2 ]; then
    echo $USAGE
    exit 1
fi

# Copy tfile
cd /opt/ofa/scripts
$CP -pr tfile.orig tfile
EXIT_STATUS=$?

# Check to make sure that the copy completed successfully
if [ $EXIT_STATUS -eq 1 ]; then
    echo "$CP -pr tfile.orig tfile FAIL"
    echo $FAIL
    exit $EXIT_STATUS
fi

# Create a temporary script called tmp.ipoibftp.sh
echo "#!/bin/bash" >> tmp_ipoibsftp.sh
echo "echo \"sftp $ADDR for $COUNT loops\"" >> tmp_ipoibsftp.sh
echo "date" >> tmp_ipoibsftp.sh
echo "sftp  root@$ADDR <<-ENDIPOIBFTP" >> tmp_ipoibsftp.sh

# We've got to loop here, as sftp does not support looping
```

```
i=0
while [ $i -lt $COUNT ]
do
   i=`expr $i + 1`
   echo "put tfile" >> tmp_ipoibsftp.sh
   echo "!rm tfile" >> tmp_ipoibsftp.sh
   echo "get tfile" >> tmp_ipoibsftp.sh
   #echo "del tfile" >> tmp_ipoibisftp.sh # Not supported by sftp
done

# After the loop, close the ftp session

echo "quit" >> tmp_ipoibsftp.sh
echo "ENDIPOIBFTP" >> tmp_ipoibsftp.sh

# Compare the two files
echo "ls -lasg tfile" >> tmp_ipoibsftp.sh
echo "ls -lasg tfile.orig" >> tmp_ipoibsftp.sh
echo "cmp tfile tfile.orig" >> tmp_ipoibsftp.sh
echo "x=\$?" >> tmp_ipoibsftp.sh
echo "echo \"Status of binary compare is \$x\"" >> tmp_ipoibsftp.sh

echo "if [ \$x -eq 0 ]" >> tmp_ipoibsftp.sh
echo "then echo -e \"\\033[01;32mFiles are identical!\\033[00m\"" >> tmp_ipoibsftp.sh
echo "else echo -e \"\\033[01;31mFiles are NOT identical or there was an error during the compare\\033[00m\"" >> tmp_ipoibsftp.sh
echo "fi" >> tmp_ipoibsftp.sh
echo "date" >> tmp_ipoibsftp.sh
echo "echo \"sftp $ADDR test completed\"" >> tmp_ipoibsftp.sh

# Change the permissions of the temp script
chmod 744 tmp_ipoibsftp.sh

# Now we can run the temp script
./tmp_ipoibsftp.sh

# Clean up by removing the script
rm -f tmp_ipoibsftp.sh
```

10.4 IB IPoIB DATAGRAM (UD)

10.4.1 Setup

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 if a corresponding connected HCA does not support this capability, it can run 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).

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

A single IPoIB subnet is reserved for Plugfest IPoIB testing. This subnet to be setup on the full default partition (0xFFFF). Its IPoIB address is 10.0.0.x/24 (10.0.0.x/netmask 255.255.255.0).

Once the IPoIB interfaces are configured on all partner HCA ports, the following procedures will be performed. The default IPoIB MTU of 2048 will be used.

The ability for each partner to create the all-IPoIB nodes IB multicast group, if observable, as well as to join that multicast group is tested.

In some configurations, when the SM/SA is local to the IPoIB implementation, not all operations will be observable with the IB analyzer (when the side with the SM/SA creates the IPoIB broadcast group). Additionally, with some SM/SAs, the creation of the IPoIB broadcast group may be previously administered and hence would not be observable by an IB analyzer.

In addition, the procedure will test the SM/SA ability to support the following functions:

1) SA in terms of performing the multicast group creation and joining.
2) SM in terms of programming the multicast topology (MulticastForwardingTable) in any switches.

The various parameters of the MCMemberRecord will be validated. In general, it will be checked that the group creator characteristics (Q_Key, etc.) are returned to the subsequent group joiners.

10.4.3 Bringing the IPoIB in Datagram Mode (UD)

1) Set "SET_IPOIB_CM=no" in file /etc/infiniband/openib.conf on all nodes being tested.
2) Restart driver "/etc/init.d/openibd restart"
3) Validate Datagram mode by checking that "/sys/class/net/<I/F name>/mode" equal to '**datagram**'
    a) For IPoIB CM: /sys/class/net/ib0/mode = "connected"
    b) For IPoIB UD: /sys/class/net/ib0/mode = "datagram"

10.4.4 Ping Procedures

Step A

1) Assign IP Addresses using the command *ifconfig ib0 10.0.0.x netmask 255.255.255.0*
2) Turn off SMs. Use ibdiagnet to verify that the master SM is missing.
3) Power cycle all switches.
   a) This insures that the new SM will configure all the links and create the multi-cast join.
   b) Run ibdiagnet to verify that all nodes have come up. Ibdiagnet does not require the SM to discover the node.
4) Use ibdiagnet to determine 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.
5) Run SM/SA from one of the nodes in the cluster.
   a) Verify that the new SM is the master. You will need to know the GUID of the device since the SM will be reassigned on each reboot.
6) Pings (ICMP requests) of the following lengths will be performed from each node (All to all): first in one direction, then the other, and finally bidirectional: 64, 256, 511, 512, 1024, 1025, 2044, 4096, 8192, 16384, 32768 and 65507. The count is 100.

   Note: In the above, the lengths of the IP (20 bytes for IPoIB Encapsulation) and IB headers are included although they will need to be subtracted out on the actual invocation of the ping command. It is also unknown whether the standard ping application without modification will allow all the lengths specified above.
7) At the completion of each different ping invocation, the arp table should be locally examined (via arp -a) and then the partner should be removed from the arp table (via arp -d) prior to starting the next ping invocation.

An IB trace of this should be examined to make sure that:

1) ARP is resolved properly (both ARP request and response are properly formatted).
2) Proper fragmentation (at the IB level) is occurring.

   Note: the case of length of 65536 ("ping of death" or long ICMP) exceeds the maximum IP length and no response is expected for this case.

   Note: At the completion of each different ping invocation, the arp table should be locally examined (via arp -a) and then the partner should be removed from the arp table (via arp -d) prior to starting the next ping invocation.

Step B

1) Bring up all HCAs but one.
2) Check for ping response between all players.
3) Disconnect one more HCA from the cluster (you should see that the ping stopped).

4) Ping to the newly disconnected HCA from all nodes (No response should be returned).
5) Connect the first machine (the one that was not connected) and check for ping response.
6) Connect the disconnected HCA to a different switch on the subnet which will change the topology. Check for ping response.
7) Ping again from all nodes (this time we should get a response).
8) Follow steps 1 to 7, this time bring the interface down and then back up using ifconfig ibX down and ifconfig ibX up commands.

Step C

Follow Step A and B running the SM/SA from each device in the cluster (If all HCAs have the same SW no need to test more than one HCA/node).

10.4.5 FTP 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.

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.

Step A

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

Step B

Run SFTP server on all nodes.

1) For each node:
   a) Connect via SFTP on IPoIB using the specified user name and passwd.
   b) Put the 4MB file to the /tmp dir on the remote host * 4 times.
   c) Get the same file to your local dir again 4 * times.
   d) Compare the file using the command *cmp tfile tfile.orig.*

10.4.6 Automated Test Script

```
#!/bin/sh
# -----------------------------------------------------------------------------
#
# ipoibsftp.sh
#
# This script automates the sftp procedure of the OFA Test Plan.
# This script takes two arguments, the IP Address of the remote
# machine (in dot notataion) and the number of times to loop the
# sftp procedure.
#
# Copyright (c) 2008 The University of New Hampshire InterOperability Laboratory
#   All Rights Reserved
#
# Author: Dustin M. Schoenbrun
#
# -----------------------------------------------------------------------------


# Variable declarations
USAGE="Usage: ./ipoibsftp.sh <IP Address> <Loop Count>"
FAIL=" ipoibsftp.sh FAIL"
CP="/bin/cp"
ADDR=$1
COUNT=$2

# Check the number of arguments to make sure that there are 2
if [ $# -ne 2 ]; then
    echo $USAGE
    exit 1
fi

# Copy tfile
cd /opt/ofa/scripts
$CP -pr tfile.orig tfile
EXIT_STATUS=$?

# Check to make sure that the copy completed successfully
if [ $EXIT_STATUS -eq 1 ]; then
    echo "$CP -pr tfile.orig tfile FAIL"
    echo $FAIL
    exit $EXIT_STATUS
fi

# Create a temporary script called tmp.ipoibftp.sh
echo "#!/bin/bash" >> tmp_ipoibsftp.sh
echo "echo \"sftp $ADDR for $COUNT loops\"" >> tmp_ipoibsftp.sh
echo "date" >> tmp_ipoibsftp.sh
echo "sftp  root@$ADDR <<-ENDIPOIBFTP" >> tmp_ipoibsftp.sh

# We've got to loop here, as sftp does not support looping
```

```
i=0
while [ $i -lt $COUNT ]
do
   i=`expr $i + 1`
   echo "put tfile" >> tmp_ipoibsftp.sh
   echo "!rm tfile" >> tmp_ipoibsftp.sh
   echo "get tfile" >> tmp_ipoibsftp.sh
   #echo "del tfile" >> tmp_ipoibisftp.sh # Not supported by sftp
done

# After the loop, close the ftp session

echo "quit" >> tmp_ipoibsftp.sh
echo "ENDIPOIBFTP" >> tmp_ipoibsftp.sh

# Compare the two files
echo "ls -lasg tfile" >> tmp_ipoibsftp.sh
echo "ls -lasg tfile.orig" >> tmp_ipoibsftp.sh
echo "cmp tfile tfile.orig" >> tmp_ipoibsftp.sh
echo "x=\$?" >> tmp_ipoibsftp.sh
echo "echo \"Status of binary compare is \$x\"" >> tmp_ipoibsftp.sh

echo "if [ \$x -eq 0 ]" >> tmp_ipoibsftp.sh
echo "then echo -e \"\\033[01;32mFiles are identical!\\033[00m\"" >> tmp_ipoibsftp.sh
echo "else echo -e \"\\033[01;31mFiles are NOT identical or there was an error during the compare\\033[00m\"" >> tmp_ipoibsftp.sh
echo "fi" >> tmp_ipoibsftp.sh
echo "date" >> tmp_ipoibsftp.sh
echo "echo \"sftp $ADDR test completed\"" >> tmp_ipoibsftp.sh

# Change the permissions of the temp script
chmod 744 tmp_ipoibsftp.sh

# Now we can run the temp script
./tmp_ipoibsftp.sh

# Clean up by removing the script
rm -f tmp_ipoibsftp.sh
```

10.5 Ethernet Link Initialize

10.5.1 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.

10.5.2 Resource Requirements

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

10.5.3 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.

10.5.4 Procedure

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

10.5.5 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).

10.5.6 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.

10.6 Ethernet Fabric Initialize

10.6.1 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.

10.6.2 Resource Requirements

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

10.6.3 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.

10.6.4 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.

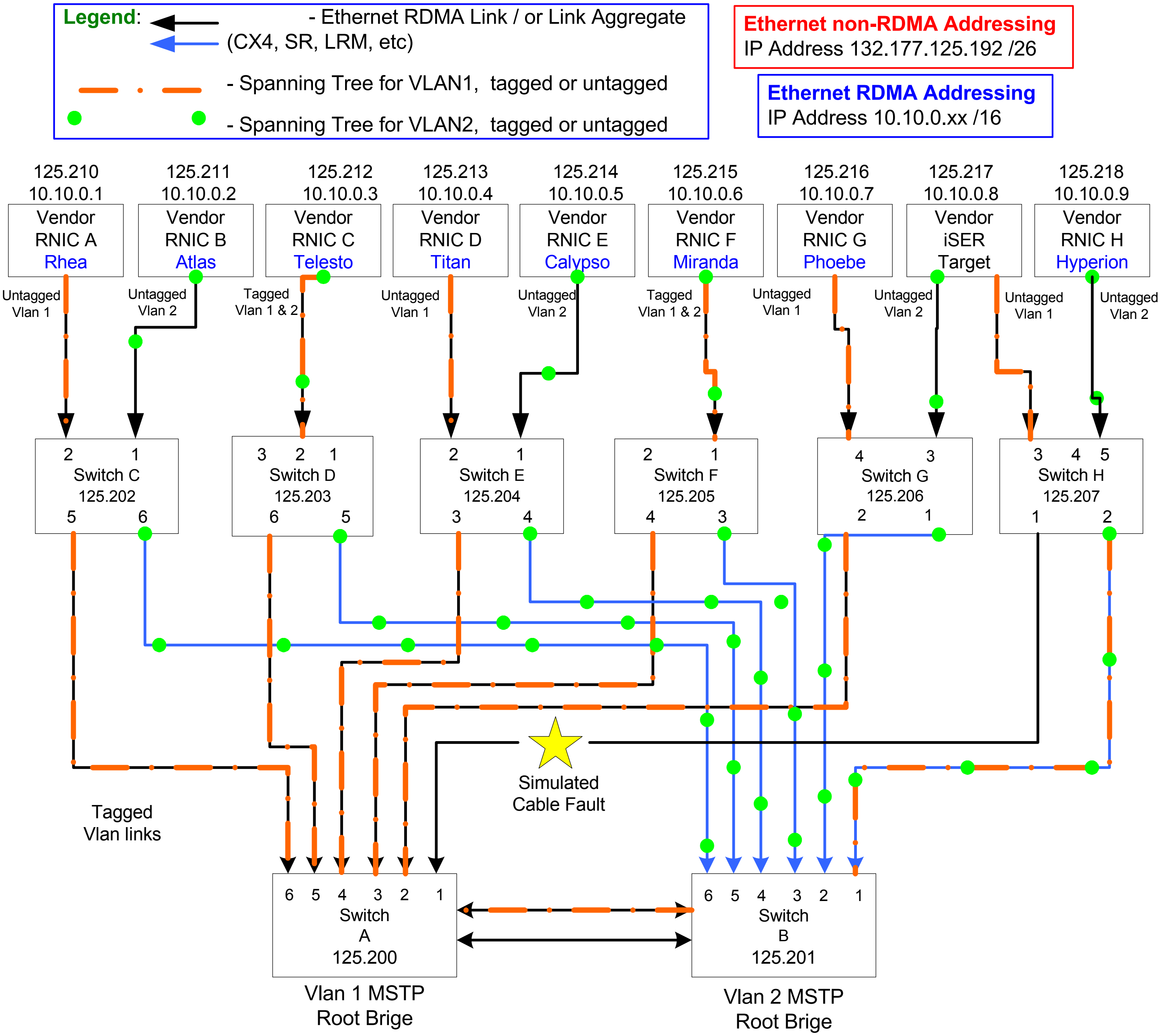
10.6.5 OBSERVABLE RESULTS

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.

10.7 ETHERNET FABRIC RECONVERGENCE

10.7.1 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.

10.7.2 Resource Requirements

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

10.7.3 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.

10.7.4 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.

10.7.5 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.

10.7.6 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.

10.8 Ethernet Fabric Failover

10.8.1 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.

10.8.2 Resource Requirements

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

10.8.3 Discussion

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

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

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

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

10.8.4 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.

10.8.5 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.

10.8.6 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.

10.9 TI iSER

10.9.1 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.

10.9.2 iWARP Setup

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

10.9.3 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).

10.9.4 TEST SCRIPT

Suggested Test Procedures

Here is the script I have been using for iSER testing on all nodes.
This is the simple version of the test without dealing with the SMs.

```
  * Start iSCSI
    On SuSE, type
      /etc/init.d/open-iscsi start
    On RedHat, type
      /etc/init.d/iscsi start
    See also: iscsi, start/stop
  * Discover new targets
    iscsi_discovery 10.0.0.12
  * See what we have discovered
    Type
      iscsiadm -m node
    Note
      * Seems to sometime give a usage message if in a bad mood.
  * Login to a specified node record
    * The previous comand should print a line for each device one may log
```

```
      into containing Address and Name which might look like
        192.168.10.35:3260,0 iqn.2000-03.com.falconstor-ipstor.sage
    * Pick the device we want to log onto and type
        iscsiadm -m node -p Address -T Name --login
      so using the previous example, it might look like
        iscsiadm -m node -p 192.168.10.35:3260,0 -T \
            iqn.2000-03.com.falconstor-ipstor.sage --login
* Look at scsi mapping to determine where new SCSI device is
    sg_map -i -x
  It will likely appear as /dev/sg2
* Attempt to read the disk.  One should use the block device which is the
  device that /dev/sg2 (or whatever) maps onto.  It will probably be
  /dev/sdc.
    time dd if=Device of=/dev/null bs=1K count=1M
```

10.10 IB SRP

Step A

Connect 2 HCAs to one of the switches and if possible, run SM/SA from the switch. If not, then run the SM/SA from one of the HCAs.

1) Initial Setup
   a) Run *ibnetdiscover* - this will show the devices that are connected on the network.
   b) Verify that you have an SM running.
   c) Run *modprobe ib-srp* - this will insert the module for SRP.
   d) Run *lsmod* | *grep ib_srp* - this will verify that the module has loaded.
2) Load SRP target and then Host, check SRP connection.
3) Load SRP host then target, and check the rescan utility.
4) Run basic dd application from SRP host connected to target.
5) Run basic dd application from SRP host connected to target. Kill the master SM while test is running and check that it completes properly.
6) Unload SRP Host / SRP target (target first / host first) and check SRP connection properly disconnected.

Follow those steps with all switches available.

1) Run SM/SA from every node/switch.
2) SM/SA can be running from all nodes.

Step B

Disconnect one of the Hosts from the switch and reconnect, then run basic dd application both from host and target.

10.10.1 AUTOMATED TEST SCRIPT

Automated Test Script

```
##############################################################
#
#       This is a script designed to automate the SRP
#       testing section of the OFA test plan. The script
#       will login to each cluster machine in turn, detect
#       all availible SRP targets, and begin to transfer
#       data to and from the targets.
#       Author: Len Mazzone
#       Last Updated: 7/17/07
#
##############################################################
```

# This is the second version of this script, incorportating for loops as opposed to using a case statement. Much more elegant, thanks to Allen Hubbe for the suggestion and help.

# Several important variables are defined here. $Hosts corresponds to the hostnames of the devices under test. $Drives corresponds to the drives on the target(s) we wish to test.

```
Hosts="telesto atlas hyperion phoebe"

#             IMPORTANT
# MAKE SURE TO UPDATE THIS TO REFLECT THE ACTUAL DISKS TO BE USED.
# Otherwise you could destroy a filesystem.
#             IMPORTANT

Drives="/dev/sdb /dev/sdc /dev/sdd"

# The purpose of the for loops is to change the order in which the hosts
# are accessed.

for HostA in $Hosts; do

RET= echo ===Starting tests on $HostA===
# Unload the srp module and reload it to assure a good state for testing.

ssh $HostA  modprobe -r ib_srp
if(($?)); then exit 1; fi

# The sleeps are included to keep the script from stepping on it's own toes.

sleep 5s

echo ===Done with cleanup===
ssh $HostA modprobe ib_srp
if(($?)); then exit 1; fi
echo ===Done loading ib_srp module===
sleep 5s

# Use the srp daemon included with the ofa package to load the targets
# as SCSI disks on the initiators.

ssh $HostA srp_daemon -e -o
if(($?)); then exit 1; fi

echo ===Done with srp setup===
echo ===Beginning dd op===
sleep 5s

# The purpose of this for loop is to allow for flexibility in  multiple drive
# /target testing.

for DriveA in $Drives; do

#             IMPORTANT
# Here we begin reading and writing to the SCSI devices we created. MAKE SURE THIS
# IS CORRECT. Otherwise you could destroy a filesystem.
#             IMPORTANT
echo ===Beginning read from drive $DriveA===
ssh $HostA dd if=$DriveA of=/dev/null count=600 bs=10M
```

```
if(($?)); then exit 1; fi
echo ===Beginning write to drive $DriveA===
ssh $HostA dd if=/dev/zero of=$DriveA count=600 bs=10M
if(($?)); then exit 1; fi

done

# The process for HostA repeats for the rest of the hosts.

for HostB in $Hosts; do

if test $HostB != $HostA; then

echo ===Starting tests on" $HostB"===

ssh $HostB  modprobe -r ib_srp
if(($?)); then exit 1; fi
echo ===Done loading ib_srp module===
sleep 5s
echo ===Done with cleanup===
ssh $HostB modprobe ib_srp
if(($?)); then exit 1; fi
sleep 5s
ssh $HostB srp_daemon -e -o
if(($?)); then exit 1; fi
echo ===Done with srp setup===
echo ===Beginning dd op===
sleep 5s

for DriveA in $Drives; do

echo ===Beginning read from drive $DriveA===
ssh $HostB dd if=$DriveA of=/dev/null count=600 bs=10M
if(($?)); then exit 1; fi
echo ===Beginning write to drive $DriveA===
ssh $HostB dd if=/dev/zero of=$DriveA count=600 bs=10M
if(($?)); then exit 1; fi

done

fi

done

echo

done
```

10.11 TI SDP

10.11.1 IB SETUP

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.

10.11.2 IWARP SETUP

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

10.11.3 INSTALLATION REQUIREMENTS

Make sure the following are installed on all nodes:

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

10.11.4 CREATING A USER NAME

Special account for this should be created as follows:

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

10.11.5 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

10.11.6 NETPERF PROCEDURE

Step A Each node will act as server.

1) For each node:
    a) Run. /netserver -p {port number}
2) From all the other nodes run:

a) [For IB] . /net perf -p {port number} -H {server nod's IPoIB} -l 1 -t TCP_STREAM -- -m {message size} -s {local buffer size}

a) [For iWARP] . /net perf -p {port number} -H {server nod's IP} -l 1 -t TCP_STREAM -- -m {message size} -s {local buffer size}

b) i.e. /net perf -p 2006 -H 11.4.10.36 -l 1 -t TCP_STREAM -- -m 1000 -s 1024

c) Where message size is 10, 100, 1000, 10000 and local buffer size is 1024, 6000.

3) Tests are expected to end on all nodes.

4) A zip file with all src files will be added.

Step B Kill the server running on each node.

10.11.7 FTP PROCEDURE

FTP procedures require 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.

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

Step A **Setup**

1) Open one window to each of the partners being tested.

2) Export the environment variable on each partner.

3) Create user name and password as specified in 10.6.4.

4) Start the FTP Daemon on both partners.

a) **Example**: /etc/init.d/ftpd start

5) Verify SDP is running.

a) lsmod | grep sdp

a) ib_sdp should be greater than 0 - reference count should be greater than 0. Each connection opens three reference counts.

Procedure

1) For each node:

a) Connect via FTP on IPoIB using the specified user name and passwd.

b) Put the 4MB file to the /tmp dir on the remote host * 4 times.

c) Get the same file to your local dir again 4 * times.

d) Compare the file.

2) During this transaction double check that sdp connection has been established, you can see it in /proc/net/sdp/conn_main.

10.11.8 SCP PROCEDURE

1) For each node:

    a) [For IB] Connect via SCP on IPoIB address from all other nodes uploading and downloading a file.

    a) [For iWARP] Connect via SCP from all other nodes uploading and downloading a file.

10.12 IB SM Failover and Handover Procedure

10.12.1 Setup

1) Connect 2 HCAs to one of the switches.
2) In this test, all active SMs on the fabric which are going to be tested, must be from the same vendor.

10.12.2 Procedure

Make sure the following are installed on all nodes:

1) Disable all SMs on the cluster until only one SM is still active.
2) Using the Agilent Exerciser, verify that all SMs are NOT ACTIVE (after receiving the SMSet of SMInfo to DISABLE) and that the selected SM (SM1) is the master (query PortInfo:SMLid should show the selected SM as active).
3) Start another SM (SM2) on the Subnet.
4) Verify Subnet and SMs behavior according to the SMs priority.
5) If SM1 priority is higher then the new SM2 priority then:
   a) Verify new SM2 goes into STANDBY and the MASTER SM1 is still the same one.
6) Disable MASTER SM1.
7) Verify the new active SM (SM2) goes into MASTER SM state and cluster nodes are configured accordingly.
8) Re-enable the original SM (SM1).
9) Next, verify SM1 goes into MASTER SM state and cluster nodes are configured accordingly while SM2 goes into STANDBY state.
10) Disable SM1.
11) Verify SM2 goes into MASTER SM state and cluster nodes are configured accordingly.
12) The utility osmtest should be used to validate the SA after failover/handover
13) Repeat steps 3 through 12 till all SMs, which are from the same vendor and are active on the subnet, have participated in the test.

Follow these steps with all switches available.

10.13 TI MPI - OHIO STATE UNIVERSITY

10.13.1 SETUP

1) Download and install MPI from:

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

2) Download and install Intel® MPI Benchmarks from:

   http://www.intel.com/cd/software/products/asmo-na/eng/308295.htm

3) Software package should be installed on all cluster nodes with typical configuration. The IMB tests must be compiled with the -DCHECK compiler flag set, to enable automatic self-checking of the results.

4) All cluster nodes should be connected and SM should be running from one management node.

10.13.2 TEST PROCEDURE

Step A: Enter the management node and define the following params:

1) $MPIHOME - path to mpi home directory.

2) $NP - number of jobs that you want run in the system (usual it is equal to [number of CPUs per node] X [number of nodes]).

3) $HOSTFILE - path to host file with list of all nodes in the system.

4) $PMB_HOME - path to Intel® MPI Benchmarks location.

Step B Run Intel® MPI Benchmarks:

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

   a) $MPIHOME/bin/mpirun_rsh -np $NP -hostfile $HOSTFILE $PMB_HOME/PMB-MPI1 -multi 0 PingPong PingPing

   a) $MPIHOME/bin/mpirun_rsh -np $NP -hostfile $HOSTFILE $PMB_HOME/PMB-MPI1

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

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

2) If the test passes move to the next SM in the cluster, and run the test again.

10.14 MPI - INTEL MPI - (NOT PART OF THE OFA STACK)

10.14.1 GENERAL ISSUES

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

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

10.14.2 SETUP FOR THE CLUSTER

1) Install same O/S version on homogenous x86_64 systems. (Recommend RH EL5 U4, EM64T)
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.

10.14.3 Setup information for OFED

1) Install OFED 1.3 on all systems.
2) Bump up the max locked memory limits on the system.

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

   ```
   *          hard    memlock        500000
   *          soft    memlock        500000
   ```
3) Run /sbin/ldconfig to pick up new OFED library path
4) Modify /etc/hosts and add IB hostnames and addresses for the IB network interfaces

5) Modify /etc/dat.conf and change the netdev reference to the appropriate interface (ib0 or ib1) being used

6) Run OpenSM either on the headnode OR from one of the switches. Verify by pinging IB addresses on all systems.

10.14.4 Setup information for Intel MPI

1) Install Intel MPI in /opt/intel/mpi/3.0 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/3.0/bin64/mpivars.sh

   # for IB, (mpi will default to rdssm if nothing defined)

   export I_MPI_DEVICE=rdssm

   # for ethernet

   export I_MPI_DEVICE=sock

   export MPIEXEC_TIMEOUT=180

   ulimit -c unlimited

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

   add single line “MPD_SECRETWORD=testing” to .mpd.conf

   chmod 600 /home/test/.mpd.conf

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

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

   Create mpd.hosts.ib and add a line for every system on the cluster using IPoIB addresses

6) Add nfs export /home/test/export on headnode and change /etc/fstab for mount points:

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

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

7) Untar the Intel Test Suites on the headnode in /home/test/exports

8) run mpdboot on the head node. For example: if you have 6 nodes on the cluster and want to run over ethernet:

   From the /home/test directory run: “mpdboot –n 6 –r ssh –f ./mpd.host.ethernet”

9) Run test suite over Ethernet to validate your installation:

```
“export I_MPI_DEVICE = sock”
run tests…(refer to test plan)
“mpdallexit”
```

10) Run test suite over IB

```
export I_MPI_DEVICE = rdssm
mpdboot –n 6 –r ssh –f ./mpd.host.ib
run tests…. (refer to test plan)
“mpdallexit”
```

10.14.5 ADDITIONAL INFORMATION

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

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

    See the Intel MPI Reference Manual for Additional information

10.14.6 INTEL MPI BENCHMARK SETUP

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

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

10.14.7 INTEL IHV TEST SUITE SETUP

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

- Configure for mpich-test ./configure –with-mpich2=/opt/intel/mpi/3.0
- Configure for mpich2-test: ./configure –with-mpich2=/opt/intel/mpi/3.0 –cc=mpicc –f77=mpif77 –cxx=mpicxx
- Configure for IntelMPITEST: ./configure –with-mpich2=/opt/intel/mpi/3.0

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.

10.14.8 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.

10.14.9 INTERPRETING THE RESULTS

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

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

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

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

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

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

10.15 TI uDAPLTEST COMMANDS

Automated test script

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

10.15.1 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

10.15.2 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

10.15.3 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

10.15.4 Automated Test Script

```
#!/bin/sh
#
# Copyright (c) 2007 The University Of New Hampshire InterOperability Laboratory
#     All Rights Reserved.
#
# This script performs tests according to the IWG Interoperability Test
# Plan.  Tests are performed on all possible sequences of two hosts from
# a list of hosts.

# List of hosts that should be used in the test run
# Example: this would run tests between hosts titan and telesto

HOSTS="titan telesto"

# For each host listed in HOSTS, there must be three entries:
# HOST_publicip="the address where the host is accessible"
# HOST_privateip="the address of the host's rdma interface"
# HOST_ianame="the rdma interface adapter name"

#  ex. information for host telesto
telesto_publicip="132.177.126.190"
telesto_privateip="10.0.0.3"
telesto_ianame="OpenIB-cma"

#  ex. information for host titan
titan_publicip="132.177.126.184"
titan_privateip="10.0.0.4"
titan_ianame="OpenIB-cma"

#  ex. information for host janus (note: janus is not included in the test)
janus_publicip="132.177.126.166"
janus_privateip="10.0.0.8"
janus_ianame="OpenIB-cma"
```

```
########################################################################

for CLIENT in $HOSTS; do
   for SERVER in $HOSTS; do

if test $CLIENT != $SERVER; then

   echo
   echo "****************************************************"
   echo "    Running tests from $CLIENT to $SERVER"
   echo "****************************************************"
   echo

   eval echo "~~~~\> Starting server on \$${SERVER}_publicip ... backgrounding"
   ## dapltest -T S -D <ia_name>

   COMMAND=$(eval echo "ssh root@\$${SERVER}_publicip killall dapltest")
   echo \# $COMMAND
   $COMMAND

   sleep 3  # let the server die completely

   COMMAND=$(eval echo "ssh -f root@\$${SERVER}_publicip dapltest -T S -D \$${SERVER}_ianame")
   echo \# $COMMAND
   $COMMAND

   sleep 3  # let the server initialize completely

   echo
   eval echo "~~~~\> Starting test 1.1 from \$${CLIENT}_publicip to \$${SERVER}_privateip"
   echo
   ## 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

  COMMAND=$(eval echo "ssh root@\$${CLIENT}_publicip dapltest -T T -s \$${SERVER}_privateip -D
  \$${CLIENT}_ianame -i 100 -t 1 -w 1 -R BE client SR 256 1 server SR 256 1")
   echo \# $COMMAND
   $COMMAND

   sleep 3

   echo
   eval echo "~~~~\> Starting test 1.2 from \$${CLIENT}_publicip to \$${SERVER}_privateip"
   echo
  ## dapltest -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

  COMMAND=$(eval echo "ssh root@\$${CLIENT}_publicip dapltest -T T -s \$${SERVER}_privateip -D
  \$${CLIENT}_ianame -i 100 -t 1 -w 1 -V -P -R BE client SR 1024 3 -f server SR 1536 2 -f")
   echo \# $COMMAND
   $COMMAND

   sleep 3
```

```
   echo
   eval echo "~~~~\> Starting test 2.1 from \$${CLIENT}_publicip to \$${SERVER}_privateip"
   echo
   ## 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

  COMMAND=$(eval echo "ssh root@\$${CLIENT}_publicip dapltest -T T -s \$${SERVER}_privateip -D
  \$${CLIENT}_ianame -i 100 -t 1 -w 1 -V -P -R BE client SR 1024 1 server SR 1024 1")
   echo \# $COMMAND
   $COMMAND

   sleep 3

   echo
   eval echo "~~~~\> Starting test 2.2 from \$${CLIENT}_publicip to \$${SERVER}_privateip"
   echo
   ## 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

  COMMAND=$(eval echo "ssh root@\$${CLIENT}_publicip dapltest -T T -s \$${SERVER}_privateip -D
  \$${CLIENT}_ianame -i 100 -t 1 -w 10 -V -P -R BE client SR 1024 3 server SR 1536 2")
   echo \# $COMMAND
   $COMMAND

   sleep 3

   echo
   eval echo "~~~~\> Starting test 2.3 from \$${CLIENT}_publicip to \$${SERVER}_privateip"
   echo
   ## 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

  COMMAND=$(eval echo "ssh root@\$${CLIENT}_publicip dapltest -T T -s \$${SERVER}_privateip -D
  \$${CLIENT}_ianame -i 100 -t 1 -w 1 -V -P -R BE client SR 256 1 server RW 4096 1 server SR 256 1")
   echo \# $COMMAND
   $COMMAND

   sleep 3

   echo
   eval echo "~~~~\> Starting test 2.4 from \$${CLIENT}_publicip to \$${SERVER}_privateip"
   echo
   ## 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

  COMMAND=$(eval echo "ssh root@\$${CLIENT}_publicip dapltest -T T -s \$${SERVER}_privateip -D
  \$${CLIENT}_ianame -i 100 -t 1 -w 1 -V -P -R BE client SR 256 1 server RR 4096 1 server SR 256 1")
   echo \# $COMMAND
   $COMMAND

   sleep 3

   echo
   eval echo "~~~~\> Starting test 3.1 from \$${CLIENT}_publicip to \$${SERVER}_privateip"
```

```
  echo
  ## 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

 COMMAND=$(eval echo "ssh root@\$${CLIENT}_publicip dapltest -T T -s \$${SERVER}_privateip -D
 \$${CLIENT}_ianame -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")
  echo \# $COMMAND
  $COMMAND

  sleep 3

  echo
  eval echo "~~~~\> Starting test 3.2 from \$${CLIENT}_publicip to \$${SERVER}_privateip"
  echo
  ## dapltest -T P -s <server_name> -D <ia_name> -i 1024 -p 64 -m p RW 8192 2

 COMMAND=$(eval echo "ssh root@\$${CLIENT}_publicip dapltest -T P -s \$${SERVER}_privateip -D
 \$${CLIENT}_ianame -i 1024 -p 64 -m p RW 8192 2")
  echo \# $COMMAND
  $COMMAND

  sleep 3

  echo
  eval echo "~~~~\> Starting test 3.3 from \$${CLIENT}_publicip to \$${SERVER}_privateip"
  echo
  ## for infiniband, use the following
  ##     dapltest -T P -s <server_name> -D <ia_name> -i 1024 -p 64 -m p RW 4096 2
  ## for iwarp, use the following
  ##     dapltest -T P -s <server_name> -D <ia_name> -i 1024 -p 64 -m p RW 4096 1

  ## infiniband
  COMMAND=$(eval echo "ssh root@\$${CLIENT}_publicip dapltest -T P -s \$${SERVER}_privateip -D
 \$${CLIENT}_ianame -i 1024 -p 64 -m p RW 4096 2")

  ## iwarp
 #COMMAND=$(eval echo "ssh root@\$${CLIENT}_publicip dapltest -T P -s \$${SERVER}_privateip -D
 \$${CLIENT}_ianame -i 1024 -p 64 -m p RW 4096 1")

  echo \# $COMMAND
  $COMMAND

  sleep 3

  echo
  eval echo "~~~~\> Starting test 3.4 from \$${CLIENT}_publicip to \$${SERVER}_privateip"
  echo
  ## 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

 COMMAND=$(eval echo "ssh root@\$${CLIENT}_publicip dapltest -T T -s \$${SERVER}_privateip -D
 \$${CLIENT}_ianame -i 100 -t 1 -w 10 -V -P -R BE client SR 1024 3 server SR 1536 2")
```

```
    echo \# $COMMAND
    $COMMAND

    sleep 3

    echo
    eval echo "~~~~\> Killing server on \$${SERVER}_publicip"
    echo

    COMMAND=$(eval echo "ssh root@\$${SERVER}_publicip killall dapltest")
    echo \# $COMMAND
    $COMMAND

    sleep 3  # let the server die completely

    fi
    done
done
```

10.16 iWARP Connectivity

10.16.1 UNH-IOL Interop Suite

See UNH-IOL iWARP Interoperability Test Suite for full details

10.16.2 iWARP Setup

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

10.16.3 Test Procedure

Step A:

Group 1: Single RDMA Operations Over A Single Connection:

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

Step B

Group 2: Multiple RDMA Operations Over A Single Connection:

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

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

Step C

Group 3: Multiple Connections:

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

Step D

Group 4: Disconnect/Reconnect Physical Connections:

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

Step E

Group 5: Speed Negotiation:

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

Step F

Group 6: RDMA Error Ratio:

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

- Test 6.4: Sequence of Interleaved Ones And Zeros

Step G

Group 7: Stress Patterns Over RDMA:

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

Step H

Group 8: Parameters:

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

10.17 IB FibreChannel Gateway

10.17.1 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.

10.18 IB Ethernet Gateway

10.18.1 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.

10.19 IB RELIABLE DATAGRAM SERVICE (RDS)

10.19.1 Procedure

1) Designate one of the cluster nodes as the master and run **qperf** on it without any arguments. Assume it is called nodex.
2) On each of the other nodes, type qperf nodex -t 20 rds_bw
3) Repeat (1) and (2) using a different node as the master.
4) If the test is successful, it will print out a non-zero bandwidth number. If it fails, it will print out an error.

   Note: This will verify that RDS packets are sent and received between all the nodes in the cluster. This particular test is also a simple stress test as it inundates the nodes with packets. The "-t 20" causes the test to be run for exactly 20 seconds.

10.20 TI BASIC RDMA Interop - Command Line

10.20.1 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.

10.20.2 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.

10.20.3 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.

10.20.4 IB Setup

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

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

10.20.5 iWARP Setup

Connect iWARP RDMA endpoints to an 10GbE switch.

Optional: insert analyzer to capture traces as needed.

10.20.6 RDMA Connectivity Setup

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

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

10.20.7 Small RDMA READ Procedure

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

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

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

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

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

10.20.8 Large RDMA READ Procedure

1) Select the two devices that will be tested:

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

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

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

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

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

10.20.9 Small RDMA Write Procedure

1) Select the two devices that will be tested:

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

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

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

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

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

10.20.10 Large RDMA Write Procedure

1) Select the two devices that will be tested:

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

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

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

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

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

10.20.11 Small RDMA SEND Procedure

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

1) Select the two devices that will be tested:

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

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

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

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

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

10.20.12 Large RDMA SEND Procedure

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

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

Example:
DevInfo - Server

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

Command Line: ib_read_bw -d mthca0 -i 1

DevInfo - Client

```
hca_id: mlx4_0
        fw_ver:                         2.2.238
        node_guid:                      0002:c903:0000:1894
        sys_image_guid:                 0002:c903:0000:1897
        vendor_id:                      0x02c9
        vendor_part_id:                 25418
        hw_ver:                         0xA0
        board_id:                       MT_04A0110002
        phys_port_cnt:                  2
                port:   1
                state:          PORT_ACTIVE (4)
                max_mtu:             2048 (4)
                active_mtu:          2048 (4)
                sm_lid:              1
                port_lid:            1
                port_lmc:            0x00
```

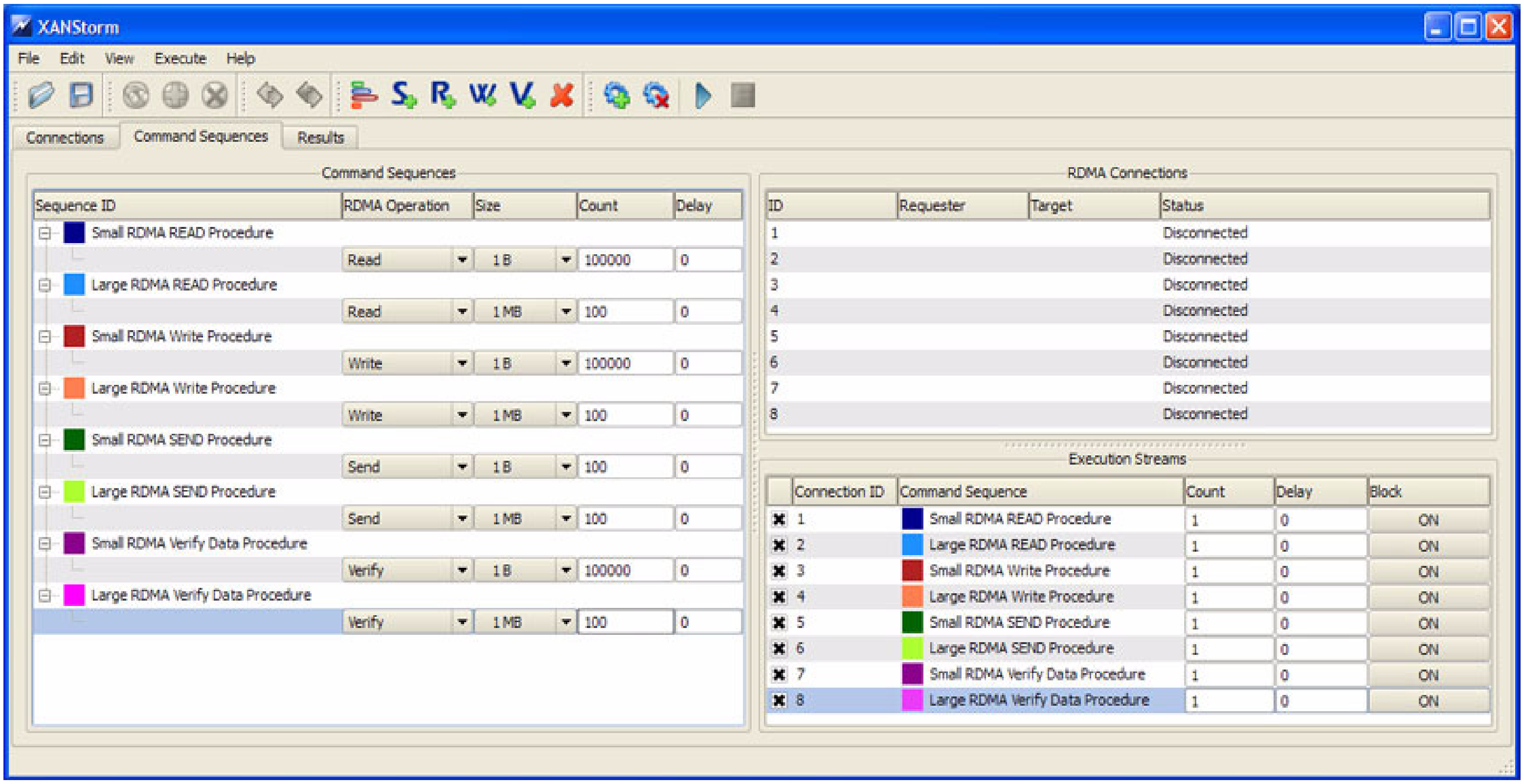
Command Line: ib_send_bw -d mlx4_0 -i 1 10.0.0.1 -s 1 -n 100

10.21 TI BASIC RDMA INTEROP - USING XANSTORM

10.21.1 Load XANStorm Test Configuration file

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

10.21.2 Run XANStorm Application



Command Sequences

Sequence ID	RDMA Operation	Size	Count	Delay
🟦 Small RDMA READ Procedure				
	Read	1 B	100000	0
🟦 Large RDMA READ Procedure				
	Read	1 MB	100	0
🟥 Small RDMA Write Procedure				
	Write	1 B	100000	0
🟧 Large RDMA Write Procedure				
	Write	1 MB	100	0
🟩 Small RDMA SEND Procedure				
	Send	1 B	100	0
🟩 Large RDMA SEND Procedure				
	Send	1 MB	100	0
🟪 Small RDMA Verify Data Procedure				
	Verify	1 B	100000	0
🟪 Large RDMA Verify Data Procedure				
	Verify	1 MB	100	0

RDMA Connections

ID	Requester	Target	Status
1			Disconnected
2			Disconnected
3			Disconnected
4			Disconnected
5			Disconnected
6			Disconnected
7			Disconnected
8			Disconnected

Execution Streams

	Connection ID	Command Sequence	Count	Delay	Block
☒	1	🟦 Small RDMA READ Procedure	1	0	ON
☒	2	🟦 Large RDMA READ Procedure	1	0	ON
☒	3	🟥 Small RDMA Write Procedure	1	0	ON
☒	4	🟧 Large RDMA Write Procedure	1	0	ON
☒	5	🟩 Small RDMA SEND Procedure	1	0	ON
☒	6	🟩 Large RDMA SEND Procedure	1	0	ON
☒	7	🟪 Small RDMA Verify Data Procedure	1	0	ON
☒	8	🟪 Large RDMA Verify Data Procedure	1	0	ON

10.22 TI RDMA OPERATIONS OVER INTERCONNECT COMPONENTS - COMMAND LINE

10.22.1 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.

10.22.2 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.

10.22.3 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.

10.22.4 Switch Load

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

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

10.22.5 Switch FAN in

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

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

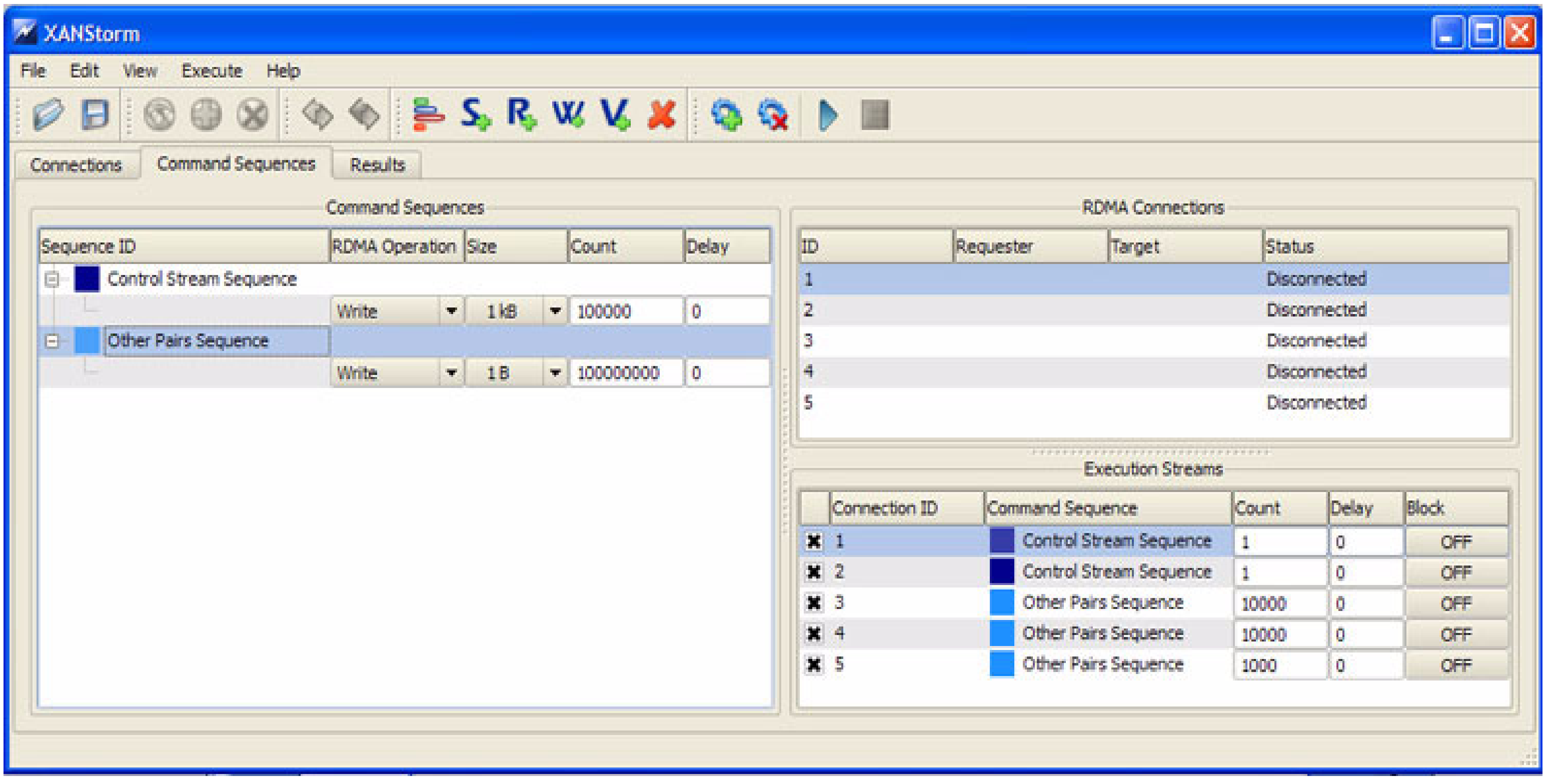
10.23 TI RDMA OPERATIONS OVER INTERCONNECT COMPONENTS - USING XANSTORM

10.23.1 Load XANStorm Test Configuration file

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

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

10.23.2 Run XANStorm Application



Command Sequences

Sequence ID	RDMA Operation	Size	Count	Delay
Control Stream Sequence				
	Write	1 kB	100000	0
Other Pairs Sequence				
	Write	1 B	100000000	0

RDMA Connections

ID	Requester	Target	Status
1			Disconnected
2			Disconnected
3			Disconnected
4			Disconnected
5			Disconnected

Execution Streams

	Connection ID	Command Sequence	Count	Delay	Block
✖	1	Control Stream Sequence	1	0	OFF
✖	2	Control Stream Sequence	1	0	OFF
✖	3	Other Pairs Sequence	10000	0	OFF
✖	4	Other Pairs Sequence	10000	0	OFF
✖	5	Other Pairs Sequence	1000	0	OFF

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

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

12 TEST SUMMARY

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 Up

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

Results Table 2 - IB Fabric Initialization

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

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

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

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

Test #	Test	Pass	Fail	Comment
1	Ping all to all - Ping using SM 1			
2	Ping all to all - Ping using SM 2			

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

Test #	Test	Pass	Fail	Comment
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 - Ethernet Link Initialize

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

Table 6 - Ethernet Fabric Initialize

Test #	Test	Pass	Fail	Comment
1	Fabric Initialization			

Table 7 - Ethernet Fabric Reconvergence

Test #	Test	Pass	Fail	Comment
1	Fabric Reconvergence			

Table 8 - Ethernet Fabric Failover

Test #	Test	Pass	Fail	Comment
1	Fabric Failover			

Results Table 9 - 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 10 - IB SRP

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

Results Table 11 - TI SDP

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

Results Table 12 - IB SM

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 13 - 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: Reduce			
12	**Test** 2: Reduce_scatter			
13	**Test** 2: Allreduce			
14	**Test** 2: Barrie			

Results Table 14a - 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 14a - 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 14b 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)			

Results Table 15 - 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			

Results Table 15 - TI uDAPL

Test #	Test	Pass	Fail	Comment
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 16 - 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			
7	Group 7 - Verify that stress pattern over iWARP work			
8	Group 8 - Verify that iWARP parameter negotiation work			

Results Table 17 - Fibre Channel Gateway - (IB Specific)

Test #	Test	Pass	Fail	Comment
1	Basic Setup			
2	Configure Gateway			

Results Table 17 - Fibre Channel Gateway - (IB Specific)

Test #	Test	Pass	Fail	Comment
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 18 - 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			

Results Table 19 - Reliable Datagram Service (RDS)

Test #	Test	Pass	Fail	Comment
1	qperf procedure			

Results Table 20 - Basic RDMA Interop

Test #	Test	Pass	Fail	Comment
1	Small RDMA READ			

Results Table 20 - Basic RDMA Interop

Test #	Test	Pass	Fail	Comment
2	Large RDMA READ			
3	Small RDMA Write			
4	Large RDMA Write			
5	Small RDMA SEND			
6	Large RDMA SEND			
7	Small RDMA Verify			
8	Large RDMA Verify			

Results Table 21 - RDMA operations over Interconnect Components

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

Results Table 22 Remarks

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