



Open-FCoE Software Initiator

Chelsio Communications, Inc.
<http://www.chelsio.com>

Copyright 2009 Chelsio Communications, Inc. All rights reserved.

370 San Aleso Ave, Suite 100, Sunnyvale, CA 94085

This document and related products are distributed under licenses restricting their use, copying, distribution, and reverse-engineering. No part of this document may be reproduced in any form or by any means without prior written permission by Chelsio Communications.

N210 and N110 are trademarks of Chelsio Communications, Inc.

Microsoft, Windows, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation.

THIS DOCUMENTATION IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

THE USE OF THE SOFTWARE AND ANY ASSOCIATED MATERIALS (COLLECTIVELY THE "SOFTWARE") IS SUBJECT TO THE SOFTWARE LICENSE TERMS OF CHELSIO COMMUNICATIONS, INC.

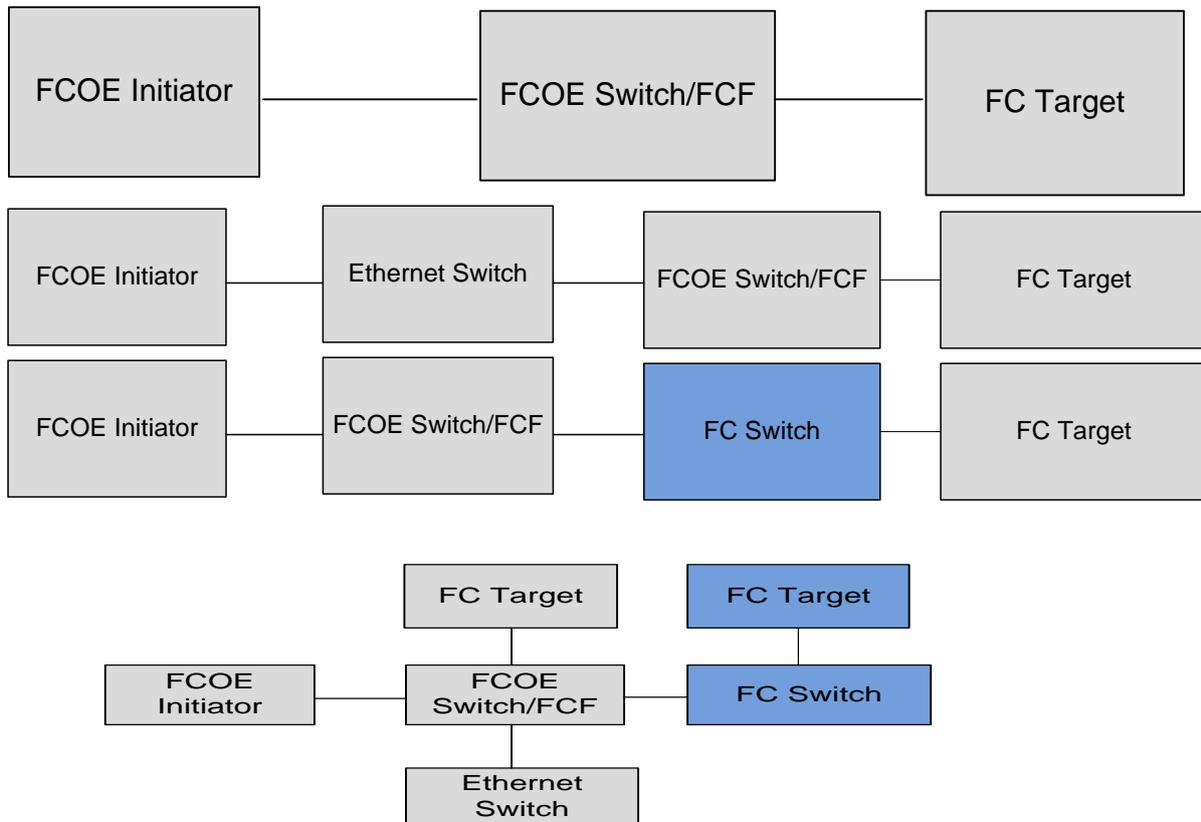
Table of Contents

Table of Contents	3
TOPOLOGY Support Tested.....	4
FABRIC Topology	4
Protocol validation	5
FCoE Initialization Protocol (FIP) Validation.....	5
Feature Tested.....	5
Multiple LUNS / TARGETS	5
ZONING	5
Bus / Target / Host RESET.....	5
Stress Test	5
Link / Port Up/Down	5
Setup Details	6
System Details	6
I/O Configurations	6
FCoE Initiator Setup	6

TOPOLOGY Support Tested

All the experiments using Open-FCoE have been performed on Chelsio T3 hardware as the underlying Network Interface Card.

FABRIC Topology



Note: Different combinations of READ/WRITE tests were run on all the above topologies.

Protocol validation

FCoE Initialization Protocol (FIP) Validation

1. FIP Discovery Advertisement
2. FIP Discovery Solicitation
3. FIP Flogi Request
4. FIP Flogi Response
5. FIP Keep Alive
6. FC Discovery Process (PLOGI,PRLI,SCSI Command Sets)

Feature Tested

Multiple LUNS / TARGETS

Connected multiple targets, with multiple LUNs, to the Chelsio T3 Open FCoE HBA port and ran 24 hour stress test.

ZONING

Created Zones on the switch and verified that only Zone members were discovered by Chelsio T3 Open FCoE Initiator.

Bus / Target / Host RESET

Using SG3 Utils performed Bus, Target, Host Reset tests.

Stress Test

Performed 48 hours of stress test.

Link / Port Up/Down

Verified the adapter can access all the devices after re-inserting the cable to the Initiator and data integrity program could continue to run without failing.

Setup Details

System Details

CPU: Intel Xeon 3.2GHz
RAM: 1GB
Chelsio driver version: 1.1.2-ko (in-box)
OS: RHEL 5.3
Kernel: 2.6.30.9

I/O Configurations

- 1) 50% Sequential/50% Random Write Test - Block size 512K to 1MB.
- 2) 50% Sequential/50% Random Read Test - Block size 512K to 1MB.
- 3) 50% Sequential/50% Random 70% Read/30% Write Test - Block size 512K to 1MB.
- 4) 50% Sequential / 50% Random 50%Read 50%write Test –Block size 16K

FCoE Initiator Setup

- a) Kernel Configuration and Compilation: Get a kernel that supports Open-FCoE. You can use the latest kernels, anything above 2.6.30 from <http://kernel.org/>

1. Untar the kernel tar ball.
2. Run 'make menuconfig' and enable the following option

```
Symbol: FCOE [=m] Prompt: FCoE module
Depends on: SCSI_LOWLEVEL [=y] && SCSI [=y] && PCI [=y]
Location:
-> Device Drivers
-> SCSI device support
-> SCSI low-level drivers (SCSI_LOWLEVEL [=y])
Selects: LIBFCOE [=m],
```

Note: Enabling this option will also enable the libfc, libfc and the FC transport.

3. Save the file and exit .
4. In the terminal window, type: <make && make modules>
5. Install the new kernel and modules: <make modules_install && make install>
6. Reboot and boot to the modified kernel

- b) Loading FCOE Modules:
 - 1) modprobe fcoe
- c) Making NIC Interface to generate FCOE traffic

- 1) Creating a FCOE Interface Port

```
echo "NIC Interface" >/sys/modules/fcoe/create
```

- 2) Deleting FCOE Interface Port

```
echo "NIC Interface" >/sys/modules/fcoe/destroy
```

where, "NIC Interface" is ethX