Chelsio Communications

Accelerate

Chelsio T7 ARM Unified Wire Solution for Linux

Installation and User Guide

Chelsio Communications (Headquarters)
735 N Pastoria Avenue,
Sunnyvale, CA 94085
U.S.A

www.chelsio.com

Tel: 408.962.3600
Fax: 408.962.3661

Chelsio (India) Private Limited
Subramanya Arcade, Floor 3, Tower B
No. 12, Bannerghatta Road,
Bangalore-560029
Karnataka,
India

Tel: +91-80-4039-6800

Sales
For all sales inquiries please send email to sales@chelsio.com

Support
For all support related questions please send email to support@chelsio.com

Version History

Version	Revision Date
1.0.0	11/13/2025
1.0.1	06/08/2026
1.0.2	07/15/2026

TABLE OF CONTENTS

I. Chelsio T7 ARM Unified Wire Solution

1. Introduction

Thank you for choosing Chelsio T7 ARM Unified Wire adapters. These high speed, single chip, multi-core firmware cards provide enterprises and datacenters with high performance solutions for various Network and Storage related requirements.

The **Terminator** series of adapters represents Chelsio's next generation of highly integrated, hyper-virtualized 1/10/25/40/50/100/200/400 GbE Ethernet Unified Wire adapters and ASICs, designed for storage networking, virtualized enterprise datacenters, cloud service environments, and embedded or cluster computing applications. The adapters are built around a programmable protocol-processing engine, with full offload of a complete Unified Wire solution comprising NIC, TOE, iWARP RDMA, iSCSI, NVMe-oF, SPDK NVMe/TCP PDU Offload, SPDK iSCSI PDU Offload, and NAT support. It scales to true 400Gb line-rate operation from a single TCP connection to thousands of connections and allows simultaneous low latency and high bandwidth operation thanks to multiple physical channels through the ASIC.

The seventh-generation T7 Data Processing Unit (DPU) and S7 SmartNIC platforms, building on the proven success of the widely deployed T6 family, introduce a powerful new architecture that delivers unmatched acceleration for AI and HPC workloads, breakthrough performance for next-generation storage, and cloud-scale efficiency with integrated security and programmable flexibility.

T7 introduces a powerful new architecture that delivers unmatched acceleration for AI and HPC workloads, breakthrough performance for next-generation storage, and cloud-scale efficiency with integrated security and programmable flexibility.

By combining advanced congestion management, protocol offloads, and a programmable FPGA and RISC pipeline, the T7 provides the networking efficiency, security, and scalability needed to eliminate bottlenecks and accelerate innovation.

This document describes the installation, use, and maintenance of Chelsio T7 ARM Unified Wire solution for Linux and its various components.

1.1. Overview

Chelsio T7 ARM Unified Wire solution for Linux is an easy to use utility developed to provide installation of ARM Linux based drivers and tools for Chelsio's adapters.

1.2. Hardware Requirements

The Chelsio T7 ARM Unified Wire solution supports the Chelsio Terminator series of Unified Wire adapters. The following is the supported hardware:

- T72200-FH-DPU

1.3. Package Contents

The Chelsio T7 ARM Unified Wire solution driver package consists of the following files/directories:

- **docs**: Documentation for this release.
- **firmware**: T7 firmware and T7 bootstrap firmware
- **images**: ITB.bin file, u-boot.bin file
- **tools**: Chelsio tools to configure and debug
- **vpds**: VPD file for ARM adapter
- **EULA**

2. Flashing on the Chelsio T7 adapter

Flashing the Chelsio T7 adapter, updates the software stack required for proper initialization and operation of the card. The process includes updating the bootstrap, init/VPD, U-Boot, and main firmware components. The init/VPD section stores essential configuration data such as MAC addresses and adapter identifiers, ensuring accurate hardware recognition. The main firmware enables core networking, offload, and management features of the T7. Flashing ensures the adapter runs the latest stable release with optimal performance, compatibility, and reliability.

2.1. Connecting the UART cable

Connecting the UART interface provides serial console access to the adapter or target device for monitoring and configuration purposes. It allows users to view boot logs, interact with the U-Boot prompt, and perform firmware flashing or debugging operations.

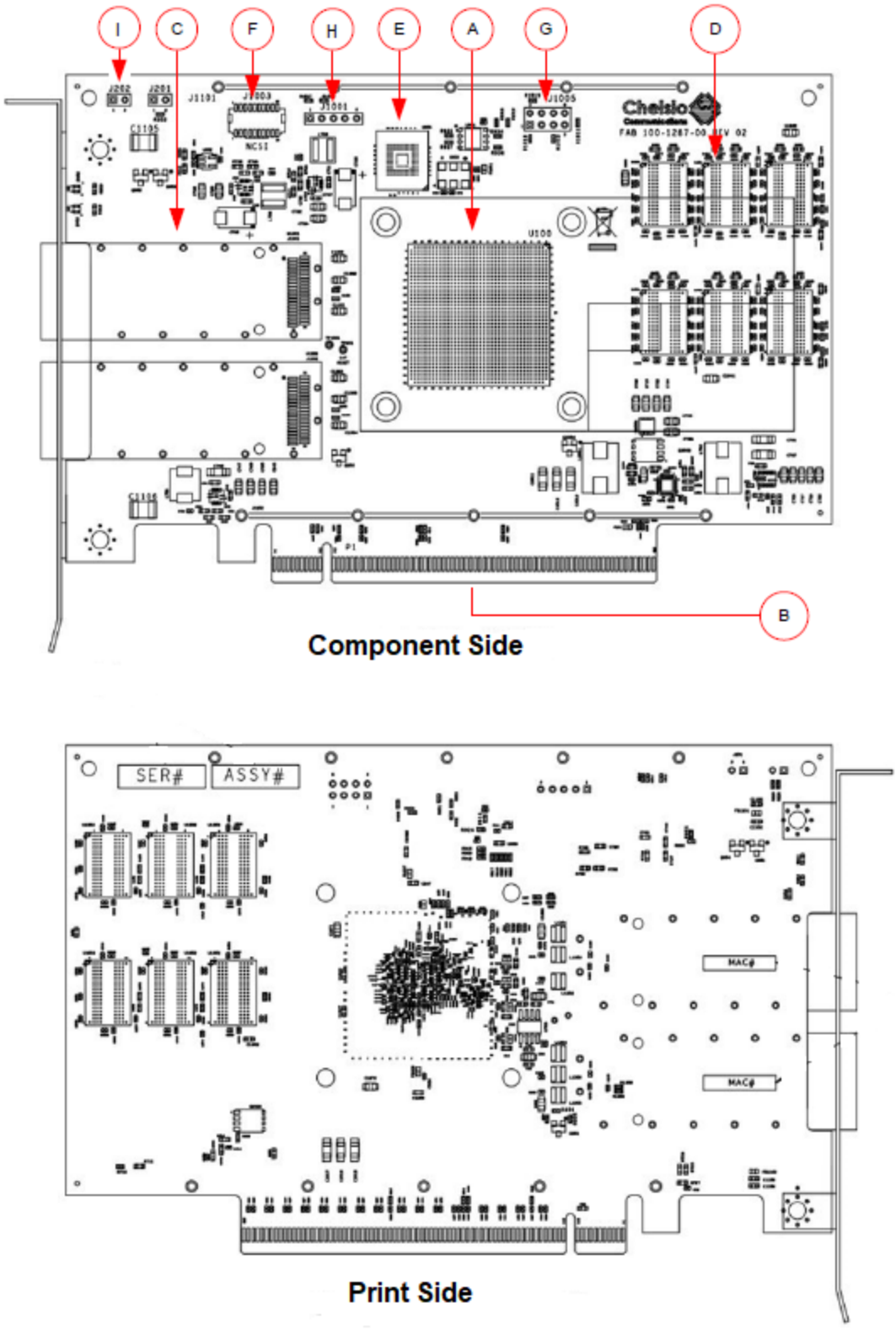


Figure 1: T72200-FH-DPU adapter

Label	Component
A	T7 ASIC 8 Core DPU
B	PCIe Gen 5 x16 Port
C	200Gb QSFP x2
D	DDR5 Memory x12
E	eMMC
F	NCSi
G	JTAG
H	UART x2
I	PCIe mode Jumper header

Perform the following steps to connect UART cable to the host machine:

i. Check the board type and connect the UART pin to the T7 Adapter.
   - **EP Board**
     a. Locate the J1001 jumper header (marked as H in Figure 1) on the T7 adapter.
     b. Identify the following pin assignments on **J1001** (Pin 1 starts from the left):
        - Pin 1: T7UARTRX
        - Pin 2: T7UARTTX
        - Pin 3: T7GROUND
        - Pin 4: T7SLAVEUARTTX
        - Pin 5: T7SLAVEUARTRX
     c. Connect the UART cable as follows to access the ARM:
        - Connect **UART TX** pin to **T7UARTRX**.
        - Connect **UART RX** pin to **T7UARTTX**.
        - Connect **UART GROUND** pin to **T7GROUND**.

ii. Connect the UART cable USB to the host machine.

iii. On the host system, verify the USB device.

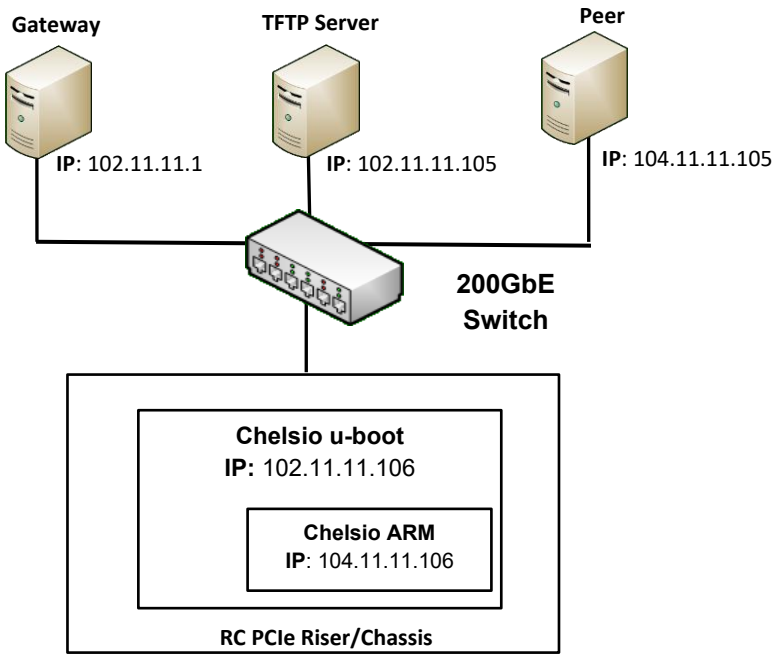
```
[root@host~]# ls /dev | grep USB

  Expected output:
    /dev/ttyUSB0
```

2.1.1. The Test Bed Architecture

Figure 2 illustrates the test bed setup used for the evaluation. The configuration includes a Gateway (IP: 102.11.11.1), a TFTP Server (IP: 102.11.11.105), and a Peer system (IP: 104.11.11.105) interconnected through a 200/400 GbE switch. The Host system comprises the Chelsio U-Boot interface (IP: 102.11.11.106) and the Chelsio T7 ARM Unified Wire solution subsystem (IP: 104.11.11.106). This setup enables high-speed communication and firmware

operations between the Chelsio adapter and the connected systems through distinct network subnets for control and data paths.

Figure 2: Test Bed



2.2. Flash bootstrap firmware, init/vpd, u-boot, and firmware

Note *These steps must be performed only once.*

To operate the ARM device in Root Complex (RC) mode, install the T7 adapter into a Linux host to flash the bootstrap firmware, init/VPD, U-Boot, and firmware. Next, power off the host, remove the T7 adapter, identify Pin J202 and connect a jumper pin to it, and then install the adapter into the target server chassis (or PCIe expansion enclosure) using an RC PCIe riser.

i. Run the following command to flash the bootstrap firmware:

```
[root@host~]# ChelsioARM-1.0.x.x/tools/cudbg_app --loadfw
"/sys/bus/pci/devices/<bus_id>" <path to bootstrap firmware.bin file> --
bootstrap
```

Example:

```
[root@host~]# ChelsioARM-1.0.x.x/tools/cudbg_app --loadfw
"/sys/bus/pci/devices/0000\:c7\:00.4" ChelsioARM-
1.0.x.x/firmware/t7fwbootstrap-2.0.4.0.bin -bootstrap
```

ii. Run the following command to flash the init VPD:

```
[root@host~]# cd ChelsioARM-1.0.x.x/vpds
[root@host~]# ChelsioARM-1.0.x.x/tools/t7seeprom -flash -b <pci-id of
chelsio card>  write -f:<path to seeprom.bin file>
```

Example:

```
[root@host~]# cd ChelsioARM-1.0.x.x/vpds
[root@host~]# ChelsioARM-1.0.x.x/tools/t7seeprom -flash -b c7:00.0 write -
f:t72200_fh_init_rc_gen5x16_dpu_vpd_mfg.bin
```

iii. Run the following command to flash the u-boot.bin:

```
[root@host~]# ChelsioARM-1.0.x.x/tools/cudbg_app -load_uboot
"/sys/bus/pci/devices/<bus_id>" <path to u-boot.bin>
```

Example:

```
[root@host~]# ChelsioARM-1.0.x.x/tools/cudbg_app --load_uboot
"/sys/bus/pci/devices/0000:c7:00.4" ChelsioARM-1.0.x.x/images/u-boot.bin
```

iv. Run the following command to flash firmware:

```
[root@host~]# ChelsioARM-1.0.x.x/tools/cudbg_app --loadfw
"/sys/bus/pci/devices/<bus_id>" <path to firmware.bin file>
```

Example:

```
[root@host~]# ChelsioARM-1.0.x.x/tools/cudbg_app --loadfw
"/sys/bus/pci/devices/0000\:c7\:00.4" ChelsioARM-1.0.x.x/firmware/t7fw-
2.1.25.0.bin
```

v. Run the following command to clear firmware config:

```
[root@host~]# ChelsioARM-1.0.x.x/tools/cudbg_app -loadcfg
"/sys/bus/pci/devices/<bus_id>" --clear
```

Example:

```
[root@host~]# ChelsioARM-1.0.x.x/tools/cudbg_app --loadcfg
"/sys/bus/pci/devices/0000\:c7\:00.4" --clear
```

vi. Run the following command to flash firmware config:

```
[root@host~]# ChelsioARM-1.0.x.x/tools/cudbg_app -loadcfg
"/sys/bus/pci/devices/<bus_id>" <path to firmware_config.txt file>
```

Example:

```
[root@host~]# ChelsioARM-1.0.x.x/tools/cudbg_app --loadcfg
"/sys/bus/pci/devices/0000\:c7\:00.4" ChelsioARM-1.0.x.x/firmware/t7-
config-iep.txt
```

vii. Power off the Linux machine.
viii. Remove the T7 adapter.
ix. Identify the **Pin J202** and connect a jumper pin to it.

Note ***Pin J202** is labeled as "I" in Figure 1.*

x. Install it into a target server chassis using an RC PCIe riser.

3. Set up TFTP server on the peer machine

Setting up a TFTP (Trivial File Transfer Protocol) server on the peer machine is essential for transferring firmware, kernel images, or configuration files to the target system during development or testing. TFTP provides a lightweight and efficient method for file transfer over the network, commonly used in conjunction with U-Boot for system booting and image updates. By configuring a TFTP server on the peer machine, users can seamlessly load images to the target device without requiring local storage or physical media, thereby streamlining system setup, testing, and recovery processes.

Follow these steps to set up a TFTP server on the peer machine:

Note *These steps must be performed only once.*

i. Run the following command to install and configure the TFTP Server:

```
[root@host~]# yum install tftp-server.x86_64
```

ii. Run the following commands to start and enable the service:

```
[root@host~]# systemctl start tftp.service
[root@host~]# systemctl enable tftp.service
[root@host~]# systemctl status tftp.service
```

Ensure that the service is active and running.

iii. Run the following command to set Permissions on TFTP Directory:

```
[root@host~]# chmod 0777 /var/lib/tftpboot
```

iv. Copy the ".itb file" to "/var/lib/tftpboot/" using the following command:

```
[root@host~]# cp ChelsioARM-1.0.x.x/images/chdpu-t7.itb /var/lib/tftpboot/
```

v. Run the following command to assign Peer IP Address:
Assign IP to match the expected 'serverip' from DUT U-Boot environment.

```
[root@host~]# ip addr add <ip/netmask> dev <your_network_interface>
```

Example:

```
[root@host~]# ip addr add 102.11.11.106/24 dev eth0
```

4. Configuring U-Boot and UART access

U-Boot, the primary bootloader, is responsible for hardware initialization, loading the operating system, and providing mechanisms for firmware upgrades and system recovery. UART access offers a low-level serial console interface that facilitates monitoring of boot-time messages, interaction with U-Boot commands, and modification of environment parameters. Configuring U-Boot and UART ensures reliable system bring-up, effective troubleshooting, and recovery from boot-related or configuration failures.

Follow these steps to configure U-Boot and UART access on Chelsio T7 ARM whenever the host is rebooted/powercycled:

i. Access UART.

Note *Either picocom or minicom can be used for UART access.*

Using plink

Run the following plink commands from host machine:

```
Using picocom
-------------
picocom -b 115200 -q /dev/ttyUSB0
On success, you should see a U-Boot prompt:
      =>
```

```
Using minicom
-------------
Configure minicom with below settings:
+---------------------------------------------------------------------+
| A -    Serial Device      : /dev/ttyUSB0                            |
|                                                                     |
| C -   Callin Program      :                                         |
| D -  Callout Program      :                                         |
| E -    Bps/Par/Bits       : 115200 8N1                              |
| F - Hardware Flow Control : No                                      |
| G - Software Flow Control : No                                      |

On success, one must see a U-Boot prompt:
       =>
```

ii. Print U-Boot environment variables.

```
=> printenv
Sample output:
    baudrate=115200
 bootargs=console=ttyCHD0,115200n8 earlycon rootwait root=/dev/ram0
 bootcmd=bdinfo; env print; tftp 0x102020000 chdpu-t7.itb; bootm
```

```
 0x10202
 bootdelay=0
 eth1addr=00:07:43:83:b6:38
 ethaddr=00:07:43:83:b6:30
 fdtcontroladdr=1f4941310
 gatewayip= 192.168.1.250
 ipaddr= 192.168.1.125
 loadaddr=0x102020000
 netmask=255.255.255.0
 serverip= 192.168.1.250
 stderr=serial@47130
 stdin=serial@47130
 stdout=serial@47130
Environment size: 427/126972 bytes
```

iii. Change the ip for local and tftpserver.

```
=> setenv ipaddr 102.11.11.105
=> setenv serverip 102.11.11.106
=> setenv gatewayip 102.11.11.1
=> printenv
baudrate=115200
bootargs=console=ttyCHD0,115200n8 earlycon rootwait root=/dev/ram0
bootcmd=bdinfo; env print; tftp 0x102020000 chdpu-t7.itb; bootm
0x102020000
bootdelay=1
fdtcontroladdr=1f4941310
gatewayip=102.11.11.1
ipaddr=102.11.11.105
loadaddr=0x102020000
netmask=255.255.255.0
serverip=102.11.11.106
stderr=serial@47130
stdin=serial@47130
stdout=serial@47130
Environment size: 429/126972 bytes
=>
CTRL-A Z for help | 115200 8N1 | NOR | Minico
```

iv. Boot from TFTP.

```
 => tftp chdpu-t7.itb
 eth_chdpu eth-chdpu-0: link up
 eth_chdpu eth-chdpu-1: link up
 Using eth-chdpu-0 device
 TFTP from server 102.11.11.105; our IP address is 102.11.11.106
 Filename 'chdpu-t7.itb'.
 Load address: 0x102020000
```

```
Loading:
 ##################################################################
 ##################################################################
 ##################################################################
 ##################################################################
 ##################################################################
 ##################################################################
 ##################################################################
 ##################################################################
 ##################################################################
 :
 :
 :
 ##################################################################
 ##################################################################
 ##################################################################
 ##################################################################
 50 MiB/s
 done
 Bytes transferred = 286020627 (110c5413 hex)
 => mw 47178 0x1
 => bootm
```

v. Once boot is successful, login prompt appears as below. Type *root*.

```
  Welcome to Buildroot
  buildroot login: root
  #
  # uname -a
 Linux buildroot 6.12.51 #1 SMP PREEMPT Wed Nov  5 13:35:47 IST 2025
 aarch64 GNU/Linux
```

vi. Load the Chelsio NIC driver.

```
  # modprobe chdpu_pcie
  # modprobe cxgb4
```

vii. Configure ARM interfaces.

```
  # ifconfig -a
dummy0 Link encap:Ethernet HWaddr BA:E1:0E:8B:43:C4
BROADCAST NOARP MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

```
eth0 Link encap:Ethernet HWaddr 00:07:43:83:B6:30
BROADCAST MULTICAST MTU:1500 Metric:1
RX packets:194850 errors:0 dropped:0 overruns:0 frame:0
TX packets:194883 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:295764414 (282.0 MiB) TX bytes:12473081 (11.8 MiB)
eth1 Link encap:Ethernet HWaddr 00:07:43:83:B6:38
BROADCAST MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:27 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:2268 (2.2 KiB)

lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:65536 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

sit0 Link encap:IPv6-in-IPv4
NOARP MTU:1480 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

# ifconfig eth0 104.11.11.106 netmask 255.255.255.0 up
# ifconfig eth1 103.11.11.178 netmask 255.255.255.0 up

# ifconfig -a
dummy0 Link encap:Ethernet HWaddr FA:EE:C4:CD:81:F6
BROADCAST NOARP MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

eth0 Link encap:Ethernet HWaddr 00:07:43:83:B6:30
inet addr:104.11.11.106 Bcast:104.11.11.255 Mask:255.255.255.0
inet6 addr: fe80::207:43ff:fe83:b630/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:198032 errors:0 dropped:0 overruns:0 frame:0
TX packets:198156 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:300578741 (286.6 MiB) TX bytes:12684646 (12.0 MiB)
```

```
eth1 Link encap:Ethernet HWaddr 00:07:43:83:B6:38
inet addr:103.11.11.178 Bcast:103.11.11.255 Mask:255.255.255.0
inet6 addr: fe80::207:43ff:fe83:b638/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:105 errors:0 dropped:0 overruns:0 frame:0
TX packets:236 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:7358 (7.1 KiB) TX bytes:17708 (17.2 KiB)

lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:65536 Metric:1
RX packets:2 errors:0 dropped:0 overruns:0 frame:0
TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:168 (168.0 B) TX bytes:168 (168.0 B)

sit0 Link encap:IPv6-in-IPv4
NOARP MTU:1480 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

viii. Ping peer over ARM interface.

```
 # ping 104.11.11.105 -c 1
PING 104.11.11.105 (104.11.11.105): 56 data bytes
64 bytes from 104.11.11.105: seq=0 ttl=64 time=0.222 ms

--- 104.11.11.105 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.222/0.222/0.222 ms
```

5. Network (NIC)

5.1. Driver Loading

Perform the following step to load the driver:

i. Load NIC driver (cxgb4),

```
[root@host~]# modprobe chdpu_pcie
[root@host~]# modprobe cxgb4
```

5.2. Driver Unloading

Perform the following step to unload the driver:

i. Load NIC driver (cxgb4),

```
[root@host~]# rmmod cxgb4
```

6. LIO iSCSI PDU Offload (Target)

6.1. Driver Loading

Perform the following step to load the driver:

i. Load the following drivers:

```
[root@host~]# modprobe chdpu_pcie
[root@host~]# modprobe cxgb4
[root@host~]# modprobe cxgbit
```

ii. Bring up the Chelsio interface(s).

```
[root@host~]# ifconfig ethX x.x.x.x up
```

Example:

```
[root@host~]# ifconfig ethX 104.11.11.106 up
```

6.2. Configuration

6.2.1. Target

1. The following commands configure target using *nvmetcli* with a LUN:

```
[root@host~]# targetcli /backstores/block/ create name=LUN0 dev=<lun_path>
[root@host~]# targetcli /iscsi create wwn=iqn.2017-01.org.lun0.target
[root@host~]# targetcli /iscsi/iqn.2017-01.org.lun0.target/tpg1/portals/
delete 0.0.0.0 3260
[root@host~]# targetcli /iscsi/iqn.2017-01.org.lun0.target/tpg1/portals/
create 104.11.11.106 ip_port=3260
[root@host~]# targetcli /iscsi/iqn.2017-01.org.lun0.target/tpg1/luns create
/backstores/block/LUN0
[root@host~]# targetcli /iscsi/iqn.2017-
01.org.lun0.target/tpg1/portals/104.11.11.106:3260 enable_offload
boolean=True
[root@host~]# targetcli /iscsi/iqn.2017-01.org.lun0.target/tpg1 set
attribute authentication=0 demo_mode_write_protect=0 generate_node_acls=1
cache_dynamic_acls=1
```

2. Save the target configuration to a file.

```
[root@host~]# targetcli saveconfig /root/iscsi_pdu_offload_target.json
```

3. To clear the targets,

```
[root@host~]# targetcli clearconfig confirm=TRUE
```

6.2.2. Peer

1. Discover the target.

```
[root@host~]# iscsiadm -m discovery -t st -p <target ip address>:<target
port no>
```

Example:

```
[root@host~]# iscsiadm -m discovery -t st -p 104.11.11.106:3260
```

2. Connect to target.

```
[root@host~]# iscsiadm -m node -T <iqn name of target> -p <target ip
address>:<target port no> -l
```

Example:

```
[root@host~]# iscsiadm -m node -T iqn.2017-01.org.lun0.target -p
104.11.11.106:3260 -l
```

3. List the connected targets.

```
[root@host~]# iscsiadm -m session -P 3
```

4. Format and mount the iSCSI disks shown with the above command.

5. Logout from the target.

```
[root@host~]# iscsiadm -m node -T <iqn name of target> -p <target ip
address>:<target port no> -u
```

Example:

```
[root@host~]# iscsiadm -m node -T iqn.2017-01.org.lun0.target -p
104.11.11.106:3260 -u
```

6.3. Driver Unloading

Follow the steps mentioned below to unload the drivers:

1. Log out from the initiator.

2. Run the following command:

```
[root@host~]# targetcli /iscsi/<target_iqn>/tpg1/portals/<target_ip>\:3260
enable_offload boolean=False
```

Example:

```
[root@host~]# targetcli /iscsi/iqn.2017-
01.org.lun0.target/tpg1/portals/104.11.11.106:3260 enable_offload
boolean=False
```

Note *Execute the above command for every portal address listening on the Chelsio interface*

3. Unload the driver.

```
[root@host~]# rmmod cxgbit
[root@host~]# rmmod cxgb4
```

7. NVMe-oF iWARP (Target)

7.1. Driver Loading

Perform the following step to load the driver:

i. Load the following drivers:

```
[root@host~]# modprobe chdpu_pcie
[root@host~]# modprobe cxgb4
[root@host~]# modprobe iw_cxgb4
```

ii. Bring up the Chelsio interface(s).

```
[root@host~]# ifconfig ethX x.x.x.x up
```

Example:

```
[root@host~]# ifconfig ethX 104.11.11.106 up
```

iii. Mount configfs.

```
[root@host~]# mount -t configfs none /sys/kernel/config
```

iv. On target, load the following drivers:

```
[root@host~]# modprobe nvmet
[root@host~]# modprobe nvmet-rdma
```

Note *On peer, load nvme and nvme-rdma drivers*

```
[root@host~]# modprobe nvme
[root@host~]# modprobe nvme-rdma
```

7.2. Target Configuration

1. The following commands configure target using *nvmetcli* with a LUN:

```
[root@host~]# nvmetcli
/> cd subsystems
/subsystems> create nvme-ram0
/subsystems> cd nvme-ram0/namespaces
/subsystems/n...m0/namespaces> create nsid=1
/subsystems/n...m0/namespaces> cd 1
/subsystems/n.../namespaces/1> set device path=<lun_path>
```

```
/subsystems/n.../namespaces/1> cd ../..
/subsystems/nvme-ram0> set attr allow_any_host=1
/subsystems/nvme-ram0> cd namespaces/1
/subsystems/n.../namespaces/1> enable
/subsystems/n.../namespaces/1> cd ../../../..
/> cd ports
/ports> create 1
/ports> cd 1/
/ports/1> set addr adrfam=ipv4
/ports/1> set addr trtype=rdma
/ports/1> set addr trsvcid=4420
/ports/1> set addr traddr=104.11.11.106
/ports/1> cd subsystems
/ports/1/subsystems> create nvme-ram0
```

2. Save the target configuration to a file.

```
/ports/1/subsystems> saveconfig /root/nvme-target_setup
/ports/1/subsystems> exit
```

3. To clear the targets,

```
[root@host~]# nvmetcli clear
```

7.3. Peer

1. Discover the target.

```
[root@host~]# nvme discover -t rdma -a <target_ip> -s 4420
```

Example:

```
[root@host~]# nvme discover -t rdma -a 104.11.11.106 -s 4420
```

2. Connect to the target.

   - Connecting to a specific target.

```
[root@host~]# nvme connect -t rdma -a <target_ip> -s 4420 -n <target_name>
```

Example:

```
[root@host~]# nvme connect -t rdma -a 104.11.11.106 -s 4420 -n nvme-ram0
```

   - Connecting to all targets configured on a portal.

```
[root@host~]# nvme connect-all -t rdma -a <target_ip> -s 4420
```

Example:

```
[root@host~]# nvme connect-all -t rdma -a 104.11.11.106 -s 4420
```

3. List the connected targets.

```
[root@host~]# nvme list
```

4. Format and mount the NVMe disks shown with the above command.

5. Disconnect from the target and unmount the disk.

```
[root@host~]# nvme disconnect -d <nvme_disk_name>
```

Note *nvme_disk_name is the name of the device (e.g., nvme0n1) and not the device path.*

7.4. Driver Unloading

Follow the steps mentioned below to unload the drivers:

On target, run the following commands:

```
[root@host~]# rmmod nvmet-rdma
[root@host~]# rmmod nvmet
[root@host~]# rmmod iw_cxgb4
```

8. NVMe-oF RoCEv2 (Target)

8.1. Driver Loading

Perform the following step to load the driver:

i. Load the following drivers:

```
[root@host~]# modprobe chdpu_pcie
[root@host~]# modprobe cxgb4
[root@host~]# modprobe iw_cxgb4 roce_mode=1
```

ii. Bring up the Chelsio interface(s).

```
[root@host~]# ifconfig ethX x.x.x.x up
```

Example:

```
[root@host~]# ifconfig ethX 104.11.11.106 up
```

iii. Mount configfs.

```
[root@host~]# mount -t configfs none /sys/kernel/config
```

iv. On target, load the following drivers:

```
[root@host~]# modprobe nvmet
[root@host~]# modprobe nvmet-rdma
```

Note *The peer must be either an NVIDIA adapter or a SoftRoCE instance. On peer, load nvme and nvme-rdma drivers*

```
[root@host~]# modprobe nvme
[root@host~]# modprobe nvme-rdma
```

8.2. Target Configuration

1. The following commands configure target using *nvmetcli* with a LUN:

```
[root@host~]# nvmetcli
/> cd subsystems
/subsystems> create nvme-ram0
```

```
/subsystems> cd nvme-ram0/namespaces
/subsystems/n...m0/namespaces> create nsid=1
/subsystems/n...m0/namespaces> cd 1
/subsystems/n.../namespaces/1> set device path=<lun_path>
/subsystems/n.../namespaces/1> cd ../..
/subsystems/nvme-ram0> set attr allow_any_host=1
/subsystems/nvme-ram0> cd namespaces/1
/subsystems/n.../namespaces/1> enable
/subsystems/n.../namespaces/1> cd ../../../..
/> cd ports
/ports> create 1
/ports> cd 1/
/ports/1> set addr adrfam=ipv4
/ports/1> set addr trtype=rdma
/ports/1> set addr trsvcid=4420
/ports/1> set addr traddr=104.11.11.106
/ports/1> cd subsystems
/ports/1/subsystems> create nvme-ram0
```

2. Save the target configuration to a file.

```
/ports/1/subsystems> saveconfig /root/nvme-target_setup
/ports/1/subsystems> exit
```

3. To clear the targets,

```
[root@host~]# nvmetcli clear
```

8.3. Peer

1. Discover the target.

```
[root@host~]# nvme discover -t rdma -a <target_ip> -s 4420
```

Example:

```
[root@host~]# nvme discover -t rdma -a 104.11.11.106 -s 4420
```

2. Connect to the target.

   - Connecting to a specific target.

```
[root@host~]# nvme connect -t rdma -a <target_ip> -s 4420 -n <target_name>
```

Example:

```
[root@host~]# nvme connect -t rdma -a 104.11.11.106 -s 4420 -n nvme-ram0
```

   - Connecting to all targets configured on a portal.

```
[root@host~]# nvme connect-all -t rdma -a <target_ip> -s 4420
```

Example:

```
[root@host~]# nvme connect-all -t rdma -a 104.11.11.106 -s 4420
```

3. List the connected targets.

```
[root@host~]# nvme list
```

4. Format and mount the NVMe disks shown with the above command.

5. Disconnect from the target and unmount the disk.

```
[root@host~]# nvme disconnect -d <nvme_disk_name>
```

Note *nvme_disk_name is the name of the device (e.g., nvme0n1) and not the device path.*

8.4. Driver Unloading

Follow the steps mentioned below to unload the drivers:

On target, run the following commands:

```
[root@host~]# rmmod nvmet-rdma
[root@host~]# rmmod nvmet
[root@host~]# rmmod iw_cxgb4
```

9. NVMe-oF/TCP PDU Offload (Target)

9.1. Driver Loading

Perform the following step to load the driver:

i. Unload the existing drivers.

```
[root@host~]# rmmod csiostor cxgb4i cxgbit iw_cxgb4 chcr cxgb4vf cxgb4
libcxgbi libcxgb
```

ii. To load NIC driver (cxgb4):

```
[root@host~]# modprobe chdpu_pcie
[root@host~]# modprobe cxgb4
```

iii. To load NVMe-oF/TCP PDU Offload Target driver (cnvmet-tcp):

```
[root@host~]# rmmod nvmet-tcp nvmet
[root@host~]# modprobe cnvmet-tcp
```

iv. On the peer load NVMe-oF/TCP kernel driver :

```
 [root@host~]# modprobe nvme-tcp nvme-fabrics nvme-core
```

9.2. Target Configuration

1. The following commands configure target using *nvmetcli* with a LUN:

```
[root@host~]# nvmetcli
/> cd subsystems
/subsystems> create nvme-ram0
/subsystems> cd nvme-ram0/namespaces
/subsystems/n...m0/namespaces> create nsid=1
/subsystems/n...m0/namespaces> cd 1
/subsystems/n.../namespaces/1> set device path=<lun_path>
/subsystems/n.../namespaces/1> cd ../..
/subsystems/nvme-ram0> set attr allow_any_host=1
/subsystems/nvme-ram0> cd namespaces/1
/subsystems/n.../namespaces/1> enable
/subsystems/n.../namespaces/1> cd ../../../..
/> cd ports
/ports> create 1
/ports> cd 1/
```

```
/ports/1> set addr adrfam=ipv4
/ports/1> set addr trtype=tcp
/ports/1> set addr trsvcid=4420
/ports/1> set addr traddr=104.11.11.106
/ports/1> cd subsystems
/ports/1/subsystems> create nvme-ram0
```

2. Save the target configuration to a file.

```
/ports/1/subsystems> saveconfig /root/nvme-target_setup
/ports/1/subsystems> exit
```

3. To clear the targets,

```
[root@host~]# nvmetcli clear
```

9.3. Peer

1. Discover the target.

```
[root@host~]# nvme discover -t tcp -a <target_ip> -s 4420
```

Example:

```
[root@host~]# nvme discover -t tcp -a 104.11.11.106 -s 4420
```

2. Connect to the target.

- Connecting to a specific target.

```
[root@host~]# nvme connect -t tcp -a <target_ip> -s 4420 -n <target_name>
```

Example:

```
[root@host~]# nvme connect -t tcp -a 104.11.11.106 -s 4420 -n nvme-ram0
```

- Connecting to all targets configured on a portal.

```
[root@host~]# nvme connect-all -t tcp -a <target_ip> -s 4420
```

Example:

```
[root@host~]# nvme connect-all -t tcp -a 104.11.11.106 -s 4420
```

3. List the connected targets.

```
[root@host~]# nvme list
```

4. Format and mount the NVMe disks shown with the above command.

5. Disconnect from the target and unmount the disk.

```
[root@host~]# nvme disconnect -d <nvme_disk_name>
```

Note *nvme_disk_name is the name of the device (e.g., nvme0n1) and not the device path.*

9.4. Driver Unloading

Follow the steps mentioned below to unload the drivers:

On target, run the following commands:

```
[root@host~]# rmmod cnvmet-tcp
```

10. Chelsio End-User License Agreement (EULA)

Installation and use of the driver/software implies acceptance of the terms in the Chelsio End-User License Agreement (EULA).

IMPORTANT: PLEASE READ THIS SOFTWARE LICENSE CAREFULLY BEFORE DOWNLOADING OR OTHERWISE USING THE SOFTWARE OR ANY ASSOCIATED DOCUMENTATION OR OTHER MATERIALS (COLLECTIVELY, THE "SOFTWARE"). BY CLICKING ON THE "OK" OR "ACCEPT" BUTTON YOU AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, CLICK THE "DO NOT ACCEPT" BUTTON TO TERMINATE THE INSTALLATION PROCESS.

IMPORTANT: PLEASE READ THIS SOFTWARE LICENSE CAREFULLY BEFORE DOWNLOADING OR OTHERWISE USING THE SOFTWARE OR ANY ASSOCIATED DOCUMENTATION OR OTHER MATERIALS (COLLECTIVELY, THE "SOFTWARE"). BY CLICKING ON THE "OK" OR "ACCEPT" BUTTON YOU AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, CLICK THE "DO NOT ACCEPT" BUTTON TO TERMINATE THE INSTALLATION PROCESS.

IMPORTANT: PLEASE READ THIS SOFTWARE LICENSE CAREFULLY BEFORE DOWNLOADING OR OTHERWISE USING THE SOFTWARE OR ANY ASSOCIATED DOCUMENTATION OR OTHER MATERIALS (COLLECTIVELY, THE "SOFTWARE"). BY CLICKING ON THE "OK" OR "ACCEPT" BUTTON YOU AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, CLICK THE "DO NOT ACCEPT" BUTTON TO TERMINATE THE INSTALLATION PROCESS.

1. License. Chelsio Communications, Inc. ("Chelsio") hereby grants you, the Licensee, and you hereby accept, a limited, non-exclusive, non-transferable license to install and use the Software with one or more Chelsio network adapters on a single server computer for use in communicating with one or more other computers over a network. You may also make one copy of the Software in machine readable form solely for back-up purposes, provided you reproduce Chelsio's copyright notice and any proprietary legends included with the Software or as otherwise required by Chelsio.

2. Restrictions. The license granted hereunder does not constitute a sale of the Software or any copy thereof. The Software contains trade secrets and, in order to protect them, you may not decompile, reverse engineer, disassemble, or otherwise reduce the Software to a human-perceivable form, except to the extent that such activity is expressly permitted in Section 5 (Authorized Distribution) by applicable law notwithstanding this limitation. You assume full responsibility for the use of the Software and agree to use the Software legally and responsibly. You may not distribute, lease, or lend the Software to third parties except as expressly provided in a separate mutual agreement or under the authorized distribution terms defined in this Agreement.

3. Ownership of Software. As Licensee, you own only the media upon which the Software is recorded or fixed (if applicable), but Chelsio retains all right, title, and interest in and to the Software and all copies of the Software, whether original or subsequent, and regardless of who possesses them. This ownership remains with Chelsio regardless of the form or media in or on which the Software may be embedded.

4. Confidentiality. You agree to maintain the non-public aspects of the Software (including source code and internal logic) in confidence. You shall not disclose the Software to any third party without the express written consent of Chelsio, except where such disclosure is a necessary part of a distribution authorized by Chelsio via mutual agreement. You further agree to take all reasonable precautions to limit access of the Software to those of your employees who reasonably require such access to perform their employment obligations and who arebound by confidentiality agreements with you. In the

event of an authorized distribution, you agree to pass on these confidentiality obligations to the recipient.

5. Authorized Distribution. You are granted a limited, non-exclusive right to distribute the Software to your end-users solely for the purpose of operating [Specific Hardware/System Name] products, provided that:

(a) Flow-Down: You ensure that your end-users are subject to a license agreement with terms no less protective of Chelsio’s rights (including Confidentiality and Reverse Engineering restrictions) than those contained herein.

(b) Indemnification: You agree to indemnify and hold Chelsio harmless from any claims arising out of your distribution or your end-users' use of the Software.

6. Term. This license is effective in perpetuity, unless terminated earlier. You may terminate the license at anytime by destroying the Software (including the related documentation), together with all copies or modifications in any form. Chelsio may terminate this license, and this license shall be deemed to have automatically terminated, if you fail to comply with any term or condition of this Agreement. Upon any termination, including termination by you, you must destroy the Software (including the related documentation), together with all copies or modifications in any form.

7. Limited Warranty. If Chelsio furnishes the Software to you on media, Chelsio warrants only that the media upon which the Software is furnished will be free from defects in material or workmanship under normal use and service for a period of thirty (30) days from the date of delivery to you.

CHELSIO DOES NOT AND CANNOT WARRANT THE PERFORMANCE OR RESULTS YOU MAY OBTAIN BY USING THE SOFTWARE OR ANY PART THEREOF. EXCEPT FOR THE FOREGOING LIMITED WARRANTY, CHELSIO MAKES NO OTHER WARRANTIES, EXPRESS OR IMPLIED, AND HEREBY DISCLAIMS ALL OTHER WARRANTIES, INCLUDING, BUT NOT LIMITED TO, NON-INFRINGEMENT OF THIRD PARTY RIGHTS, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow the exclusion of implied warranties or limitations on how long an implied warranty may last, so the above limitations may not apply to you. This warranty gives you specific legal rights and you may also have other rights which vary from state to state.

8. Remedy for Breach of Warranty. The sole and exclusive liability of Chelsio and its distributors, and your sole and exclusive remedy, for a breach of the above warranty, shall be the replacement of any media furnished by Chelsio not meeting the above limited warranty and which is returned to Chelsio. If Chelsio or its distributor is unable to deliver replacement media which is free from defects in materials or workmanship, you may terminate this Agreement by returning the Software.

9. Limitation of Liability. IN NO EVENT SHALL CHELSIO HAVE ANY LIABILITY TO YOU OR ANY THIRD PARTY FOR ANY INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR PUNITIVE DAMAGES, HOWEVER CAUSED, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR RELATED TO THE LICENSE OR USE OF THE SOFTWARE, INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR LOSS OF ANTICIPATED PROFITS, EVEN IF CHELSIO HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL CHELSIO'S LIABILITY ARISING OUT OF OR RELATED TO THE LICENSE OR USE OF THE SOFTWARE EXCEED THE AMOUNTS PAID BY YOU FOR THE LICENSE GRANTED HEREUNDER. THESE LIMITATIONS SHALL APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY.

10. High Risk Activities. The Software is not fault- tolerant and is not designed, manufactured or intended for use or resale as online equipment control equipment in hazardous environments requiring fail-safe performance, such as in the operation of

nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the failure of the Software could lead directly to death, personal injury, or severe physical or environmental damage. Chelsio specifically disclaims any express or implied warranty of fitness for any high risk uses listed above.

11. Export. You acknowledge that the Software is of U.S. origin and subject to U.S. export jurisdiction. You acknowledge that the laws and regulations of the United States and other countries may restrict the export and re-export of the Software. You agree that you will not export or re-export the Software or documentation in any form in violation of applicable United States and foreign law. You agree to comply with all applicable international and national laws that apply to the Software, including the U.S. Export Administration Regulations, as well as end-user, end-use, and destination restrictions issued by U.S. and other governments.

12. Government Restricted Rights. The Software is subject to restricted rights as follows. If the Software is acquired under the terms of a GSA contract: use, reproduction or disclosure is subject to the restrictions set forth in the applicable ADP Schedule contract. If the Software is acquired under the terms of a DoD or civilian agency contract, use, duplication or disclosure by the Government is subject to the restrictions of this Agreement in accordance with 48 C.F.R. 12.212 of the Federal Acquisition Regulations and its successors and 49 C.F.R. 227.7202-1 of the DoD FAR Supplement and its successors.

13. General. You acknowledge that you have read this Agreement, understand it, and that by using the Software you agree to be bound by its terms and conditions. You further agree that it is the complete and exclusive statement of the agreement between Chelsio and you, and supersedes any proposal or prior agreement, oral or written, and any other communication between Chelsio and you relating to the subject matter of this Agreement. No additional or any different terms will be enforceable against Chelsio unless Chelsio gives its express consent, including an express waiver of the terms of this Agreement, in writing signed by an officer of Chelsio. This Agreement shall be governed by California law, except as to copyright matters, which are covered by Federal law. You hereby irrevocably submit to the personal jurisdiction of, and irrevocably waive objection to the laying of venue (including a waiver of any argument of forum non conveniens or other principles of like effect) in, the state and federal courts located in Santa Clara County, California, for the purposes of any litigation undertaken in connection with this Agreement. Should any provision of this Agreement be declared unenforceable in any jurisdiction, then such provision shall be deemed severable from this Agreement and shall not affect the remainder hereof. All rights in the Software not specifically granted in this Agreement are reserved by Chelsio. You may not assign or transfer this Agreement (by merger, operation of law or in any other manner) without the prior written consent of Chelsio and any attempt to do so without such consent shall be void and shall constitute a material breach of this Agreement.

Should you have any questions concerning this Agreement, you may contact Chelsio by writing to:

Chelsio Communications, Inc.
735 N Pastoria Avenue,
Sunnyvale, CA 94085