cxgb.txt
Chelsio N210 10Gb Ethernet Network Controller

Driver Release Notes for Linux

Version 2.1.1

June 20, 2005

CONTENTS
========

INTRODUCTION
FEATURES
PERFORMANCE
DRIVER MESSAGES
KNOWN ISSUES
SUPPORT

INTRODUCTION
============

This document describes the Linux driver for Chelsio 10Gb Ethernet Network Controller. This driver supports the Chelsio N210 NIC and is backward compatible with the Chelsio N110 model 10Gb NICs.

FEATURES
========

Adaptive Interrupts (adaptive-rx)

This feature provides an adaptive algorithm that adjusts the interrupt coalescing parameters, allowing the driver to dynamically adapt the latency settings to achieve the highest performance during various types of network load.

The interface used to control this feature is ethtool. Please see the ethtool manpage for additional usage information.

By default, adaptive-rx is disabled.
To enable adaptive-rx:

```
ethtool -C <interface> adaptive-rx on
```

To disable adaptive-rx, use ethtool:

```
ethtool -C <interface> adaptive-rx off
```

After disabling adaptive-rx, the timer latency value will be set to 50us. You may set the timer latency after disabling adaptive-rx:

```
ethtool -C <interface> rx-usecs <microseconds>
```

An example to set the timer latency value to 100us on eth0:

```
    ethtool -C eth0 rx-usecs 100
```

You may also provide a timer latency value while disabling adaptive-rx:

```
    ethtool -C <interface> adaptive-rx off rx-usecs <microseconds>
```

If adaptive-rx is disabled and a timer latency value is specified, the timer will be set to the specified value until changed by the user or until adaptive-rx is enabled.

To view the status of the adaptive-rx and timer latency values:

```
    ethtool -c <interface>
```

TCP Segmentation Offloading (TSO) Support

This feature, also known as "large send", enables a system's protocol stack to offload portions of outbound TCP processing to a network interface card thereby reducing system CPU utilization and enhancing performance.

The interface used to control this feature is ethtool version 1.8 or higher. Please see the ethtool manpage for additional usage information.

By default, TSO is enabled.
To disable TSO:

```
    ethtool -K <interface> tso off
```

To enable TSO:

```
    ethtool -K <interface> tso on
```

To view the status of TSO:

```
    ethtool -k <interface>
```

PERFORMANCE

The following information is provided as an example of how to change system parameters for "performance tuning" an what value to use. You may or may not want to change these system parameters, depending on your server/workstation application. Doing so is not warranted in any way by Chelsio Communications, and is done at "YOUR OWN RISK". Chelsio will not be held responsible for loss of data or damage to equipment.

Your distribution may have a different way of doing things, or you may prefer a different method. These commands are shown only to provide an example of what to do and are by no means definitive.

Making any of the following system changes will only last until you reboot your system. You may want to write a script that runs at boot-up which includes the optimal settings for your system.

```
Setting PCI Latency Timer:
    setpci -d 1425:* 0x0c.l=0x0000F800

Disabling TCP timestamp:
    sysctl -w net.ipv4.tcp_timestamps=0

Disabling SACK:
    sysctl -w net.ipv4.tcp_sack=0

Setting large number of incoming connection requests:
    sysctl -w net.ipv4.tcp_max_syn_backlog=3000

Setting maximum receive socket buffer size:
    sysctl -w net.core.rmem_max=1024000

Setting maximum send socket buffer size:
    sysctl -w net.core.wmem_max=1024000

Set smp_affinity (on a multiprocessor system) to a single CPU:
    echo 1 > /proc/irq/<interrupt_number>/smp_affinity

Setting default receive socket buffer size:
    sysctl -w net.core.rmem_default=524287

Setting default send socket buffer size:
    sysctl -w net.core.wmem_default=524287

Setting maximum option memory buffers:
    sysctl -w net.core.optmem_max=524287

Setting maximum backlog (# of unprocessed packets before kernel drops):
    sysctl -w net.core.netdev_max_backlog=300000

Setting TCP read buffers (min/default/max):
    sysctl -w net.ipv4.tcp_rmem="10000000 10000000 10000000"

Setting TCP write buffers (min/pressure/max):
    sysctl -w net.ipv4.tcp_wmem="10000000 10000000 10000000"

Setting TCP buffer space (min/pressure/max):
    sysctl -w net.ipv4.tcp_mem="10000000 10000000 10000000"

TCP window size for single connections:
 The receive buffer (RX_WINDOW) size must be at least as large as the
 Bandwidth-Delay Product of the communication link between the sender and
 receiver. Due to the variations of RTT, you may want to increase the buffer
 size up to 2 times the Bandwidth-Delay Product. Reference page 289 of
 "TCP/IP Illustrated, Volume 1, The Protocols" by W. Richard Stevens.
 At 10Gb speeds, use the following formula:
     RX_WINDOW >= 1.25MBytes * RTT(in milliseconds)
     Example for RTT with 100us: RX_WINDOW = (1,250,000 * 0.1) = 125,000
 RX_WINDOW sizes of 256KB - 512KB should be sufficient.
 Setting the min, max, and default receive buffer (RX_WINDOW) size:
     sysctl -w net.ipv4.tcp_rmem="<min> <default> <max>"
```

TCP window size for multiple connections:
The receive buffer (RX_WINDOW) size may be calculated the same as single connections, but should be divided by the number of connections. The smaller window prevents congestion and facilitates better pacing, especially if/when MAC level flow control does not work well or when it is not supported on the machine. Experimentation may be necessary to attain the correct value. This method is provided as a starting point for the correct receive buffer size.
Setting the min, max, and default receive buffer (RX_WINDOW) size is performed in the same manner as single connection.

DRIVER MESSAGES

The following messages are the most common messages logged by syslog. These may be found in /var/log/messages.

Driver up:
```
Chelsio Network Driver - version 2.1.1
```

NIC detected:
```
eth#: Chelsio N210 1x10GBaseX NIC (rev #), PCIX 133MHz/64-bit
```

Link up:
```
eth#: link is up at 10 Gbps, full duplex
```

Link down:
```
eth#: link is down
```

KNOWN ISSUES

These issues have been identified during testing. The following information is provided as a workaround to the problem. In some cases, this problem is inherent to Linux or to a particular Linux Distribution and/or hardware platform.

1. Large number of TCP retransmits on a multiprocessor (SMP) system.

   On a system with multiple CPUs, the interrupt (IRQ) for the network controller may be bound to more than one CPU. This will cause TCP retransmits if the packet data were to be split across different CPUs and re-assembled in a different order than expected.

   To eliminate the TCP retransmits, set smp_affinity on the particular interrupt to a single CPU. You can locate the interrupt (IRQ) used on the N110/N210 by using ifconfig:
   ```
   ifconfig <dev_name> | grep Interrupt
   ```
   Set the smp_affinity to a single CPU:
   ```
   echo 1 > /proc/irq/<interrupt_number>/smp_affinity
   ```

   It is highly suggested that you do not run the irqbalance daemon on your system, as this will change any smp_affinity setting you have applied. The irqbalance daemon runs on a 10 second interval and binds interrupts

to the least loaded CPU determined by the daemon. To disable this daemon:
chkconfig --level 2345 irqbalance off

By default, some Linux distributions enable the kernel feature, irqbalance, which performs the same function as the daemon. To disable this feature, add the following line to your bootloader:
noirqbalance

Example using the Grub bootloader:
```
title Red Hat Enterprise Linux AS (2.4.21-27.ELsmp)
root (hd0,0)
kernel /vmlinuz-2.4.21-27.ELsmp ro root=/dev/hda3 noirqbalance
initrd /initrd-2.4.21-27.ELsmp.img
```

2. After running insmod, the driver is loaded and the incorrect network interface is brought up without running ifup.

When using 2.4.x kernels, including RHEL kernels, the Linux kernel invokes a script named "hotplug". This script is primarily used to automatically bring up USB devices when they are plugged in, however, the script also attempts to automatically bring up a network interface after loading the kernel module. The hotplug script does this by scanning the ifcfg-eth# config files in /etc/sysconfig/network-scripts, looking for HWADDR=<mac_address>.

If the hotplug script does not find the HWADDRR within any of the ifcfg-eth# files, it will bring up the device with the next available interface name. If this interface is already configured for a different network card, your new interface will have incorrect IP address and network settings.

To solve this issue, you can add the HWADDR=<mac_address> key to the interface config file of your network controller.

To disable this "hotplug" feature, you may add the driver (module name) to the "blacklist" file located in /etc/hotplug. It has been noted that this does not work for network devices because the net.agent script does not use the blacklist file. Simply remove, or rename, the net.agent script located in /etc/hotplug to disable this feature.

3. Transport Protocol (TP) hangs when running heavy multi-connection traffic on an AMD Opteron system with HyperTransport PCI-X Tunnel chipset.

If your AMD Opteron system uses the AMD-8131 HyperTransport PCI-X Tunnel chipset, you may experience the "133-Mhz Mode Split Completion Data Corruption" bug identified by AMD while using a 133Mhz PCI-X card on the bus PCI-X bus.

AMD states, "Under highly specific conditions, the AMD-8131 PCI-X Tunnel can provide stale data via split completion cycles to a PCI-X card that is operating at 133 Mhz", causing data corruption.

AMD's provides three workarounds for this problem, however, Chelsio recommends the first option for best performance with this bug:

For 133Mhz secondary bus operation, limit the transaction length and

the number of outstanding transactions, via BIOS configuration programming of the PCI-X card, to the following:

Data Length (bytes): 1k
Total allowed outstanding transactions: 2

Please refer to AMD 8131-HT/PCI-X Errata 26310 Rev 3.08 August 2004, section 56, "133-MHz Mode Split Completion Data Corruption" for more details with this bug and workarounds suggested by AMD.

It may be possible to work outside AMD's recommended PCI-X settings, try increasing the Data Length to 2k bytes for increased performance. If you have issues with these settings, please revert to the "safe" settings and duplicate the problem before submitting a bug or asking for support.

NOTE: The default setting on most systems is 8 outstanding transactions and 2k bytes data length.

4. On multiprocessor systems, it has been noted that an application which is handling 10Gb networking can switch between CPUs causing degraded and/or unstable performance.

   If running on an SMP system and taking performance measurements, it is suggested you either run the latest netperf-2.4.0+ or use a binding tool such as Tim Hockin's procstate utilities (runon) <http://www.hockin.org/~thockin/procstate/>.

   Binding netserver and netperf (or other applications) to particular CPUs will have a significant difference in performance measurements. You may need to experiment which CPU to bind the application to in order to achieve the best performance for your system.

   If you are developing an application designed for 10Gb networking, please keep in mind you may want to look at kernel functions sched_setaffinity & sched_getaffinity to bind your application.

   If you are just running user-space applications such as ftp, telnet, etc., you may want to try the runon tool provided by Tim Hockin's procstate utility. You could also try binding the interface to a particular CPU: runon 0 ifup eth0

SUPPORT
=======

If you have problems with the software or hardware, please contact our customer support team via email at support@chelsio.com or check our website at http://www.chelsio.com

==========

Chelsio Communications
370 San Aleso Ave.
Suite 100
Sunnyvale, CA 94085
http://www.chelsio.com

==========