NAME

ti - Alteon Networks Tigon I and Tigon II Gigabit Ethernet driver

SYNOPSIS

To compile this driver into the kernel, place the following lines in your kernel configuration file:

```
device ti
options TI_SF_BUF_JUMBO
options TI_JUMBO_HDRSPLIT
```

Alternatively, to load the driver as a module at boot time, place the following line in loader.conf(5):

```
if_ti_load="YES"
```

DESCRIPTION

The **ti** driver provides support for PCI Gigabit Ethernet adapters based on the Alteon Networks Tigon Gigabit Ethernet controller chip. The Tigon contains an embedded R4000 CPU, gigabit MAC, dual DMA channels and a PCI interface unit. The Tigon II contains two R4000 CPUs and other refinements. Either chip can be used in either a 32-bit or 64-bit PCI slot. Communication with the chip is achieved via PCI shared memory and bus master DMA. The Tigon I and II support hardware multicast address filtering, VLAN tag extraction and insertion, and jumbo Ethernet frames sizes up to 9000 bytes. Note that the Tigon I chipset is no longer in active production: all new adapters should come equipped with Tigon II chipsets.

While the Tigon chipset supports 10, 100 and 1000Mbps speeds, support for 10 and 100Mbps speeds is only available on boards with the proper transceivers. Most adapters are only designed to work at 1000Mbps, however the driver should support those NICs that work at lower speeds as well.

Support for jumbo frames is provided via the interface MTU setting. Selecting an MTU larger than 1500 bytes with the ifconfig(8) utility configures the adapter to receive and transmit jumbo frames. Using jumbo frames can greatly improve performance for certain tasks, such as file transfers and data streaming.

Header splitting support for Tigon 2 boards (this option has no effect for the Tigon 1) can be turned on with the TI_JUMBO_HDRSPLIT option. See zero_copy(9) for more discussion on zero copy receive and header splitting.

The **ti** driver uses UMA backed jumbo receive buffers, but can be configured to use sendfile(2) buffer allocator. To turn on sendfile(2) buffer allocator, use the TI_SF_BUF_JUMBO option.

Support for vlans is also available using the vlan(4) mechanism. See the vlan(4) man page for more details.

The **ti** driver supports the following media types:

autoselect	Enable autoselection of the media type and options. The user can manually override the autoselected mode by adding media options to the */etc/rc.conf* file.
10baseT/UTP	Set 10Mbps operation. The *mediaopt* option can also be used to select either *full-duplex* or *half-duplex* modes.
100baseTX	Set 100Mbps (Fast Ethernet) operation. The *mediaopt* option can also be used to select either *full-duplex* or *half-duplex* modes.
1000baseSX	Set 1000Mbps (Gigabit Ethernet) operation. Only *full-duplex* mode is supported at this speed.

The **ti** driver supports the following media options:

full-duplex	Force full-duplex operation.
half-duplex	Force half duplex operation.

For more information on configuring this device, see ifconfig(8).

HARDWARE

The **ti** driver supports Gigabit Ethernet adapters based on the Alteon Tigon I and II chips. The **ti** driver has been tested with the following adapters:

- 3Com 3c985-SX Gigabit Ethernet adapter (Tigon 1)
- 3Com 3c985B-SX Gigabit Ethernet adapter (Tigon 2)
- Alteon AceNIC V Gigabit Ethernet adapter (1000baseSX)
- Alteon AceNIC V Gigabit Ethernet adapter (1000baseT)
- Digital EtherWORKS 1000SX PCI Gigabit adapter
- Netgear GA620 Gigabit Ethernet adapter (1000baseSX)
- Netgear GA620T Gigabit Ethernet adapter (1000baseT)

The following adapters should also be supported but have not yet been tested:

- Asante GigaNIX1000T Gigabit Ethernet adapter
- Asante PCI 1000BASE-SX Gigabit Ethernet adapter
- Farallon PN9000SX Gigabit Ethernet adapter
- NEC Gigabit Ethernet
- Silicon Graphics PCI Gigabit Ethernet adapter

LOADER TUNABLES

Tunables can be set at the loader(8) prompt before booting the kernel or stored in loader.conf(5).

hw.ti.%d.dac

If this tunable is set to 0 it will disable DAC (Dual Address Cycle). The default value is 1 which means driver will use full 64bit DMA addressing.

SYSCTL VARIABLES

The following variables are available as both sysctl(8) variables and loader(8) tunables. The interface has to be brought down and up again before a change takes effect when any of the following tunables are changed. The one microsecond clock tick referenced below is a nominal time and the actual hardware may not provide granularity to this level. For example, on Tigon 2 (revision 6) cards with release 12.0 the clock granularity is 5 microseconds.

dev.ti.%d.rx_coal_ticks

This value, receive coalesced ticks, controls the number of clock ticks (of 1 microseconds each) that must elapse before the NIC DMAs the receive return producer pointer to the Host and generates an interrupt. This parameter works in conjunction with the rx_max_coal_bds, receive max coalesced BDs, tunable parameter. The NIC will return the receive return producer pointer to the Host when either of the thresholds is exceeded. A value of 0 means that this parameter is ignored and receive BDs will only be returned when the receive max coalesced BDs value is reached. The default value is 170.

dev.ti.%d.rx_max_coal_bds

This value, receive max coalesced BDs, controls the number of receive buffer descriptors that will be coalesced before the NIC updates the receive return ring producer index. If this value is set to 0 it will disable receive buffer descriptor coalescing. The default value is 64.

dev.ti.%d.ti_tx_coal_ticks

This value, send coalesced ticks, controls the number of clock ticks (of 1 microseconds each) that must elapse before the NIC DMAs the send consumer pointer to the Host and generates an interrupt. This parameter works in conjunction with the tx_max_coal_bds, send max coalesced BDs, tunable parameter. The NIC will return the send consumer pointer to the Host when either of the thresholds is exceeded. A value of 0 means that this parameter is ignored and send

BDs will only be returned when the send max coalesced BDs value is reached. The default value is 2000.

dev.ti.%d.tx_max_coal_bds
This value, send max coalesced BDs, controls the number of send buffer descriptors that will be coalesced before the NIC updates the send consumer index. If this value is set to 0 it will disable send buffer descriptor coalescing. The default value is 32.

dev.ti.%d.tx_buf_ratio
This value controls the ratio of the remaining memory in the NIC that should be devoted to transmit buffer vs. receive buffer. The lower 7 bits are used to indicate the ratio in 1/64th increments. For example, setting this value to 16 will set the transmit buffer to 1/4 of the remaining buffer space. In no cases will the transmit or receive buffer be reduced below 68 KB. For a 1 MB NIC the approximate total space for data buffers is 800 KB. For a 512 KB NIC that number is 300 KB. The default value is 21.

dev.ti.%d.stat_ticks
The value, stat ticks, controls the number of clock ticks (of 1 microseconds each) that must elapse before the NIC DMAs the statistics block to the Host and generates a STATS_UPDATED event. If set to zero then statistics are never DMAed to the Host. It is recommended that this value be set to a high enough frequency to not mislead someone reading statistics refreshes. Several times a second is enough. The default value is 2000000 (2 seconds).

IOCTLS

In addition to the standard socket(2) ioctl(2) calls implemented by most network drivers, the **ti** driver also includes a character device interface that can be used for additional diagnostics, configuration and debugging. With this character device interface, and a specially patched version of gdb(1) (*ports/devel/gdb*), the user can debug firmware running on the Tigon board.

These ioctls and their arguments are defined in the <*sys/tiio.h*> header file.

TIIOCGETSTATS	Return card statistics DMAed from the card into kernel memory approximately every 2 seconds. (That time interval can be changed via the TIIOCSETPARAMS ioctl.) The argument is *struct ti_stats*.
TIIOCGETPARAMS	Get various performance-related firmware parameters that largely affect how interrupts are coalesced. The argument is *struct ti_params*.
TIIOCSETPARAMS	Set various performance-related firmware parameters that largely affect how

interrupts are coalesced. The argument is *struct ti_params*.

TIIOCSETTRACE Tell the NIC to trace the requested types of information. The argument is *ti_trace_type*.

TIIOCGETTRACE Dump the trace buffer from the card. The argument is *struct ti_trace_buf*.

ALT_ATTACH This ioctl is used for compatibility with Alteon's Solaris driver. They apparently only have one character interface for debugging, so they have to tell it which Tigon instance they want to debug. This ioctl is a noop for FreeBSD.

ALT_READ_TG_MEM Read the requested memory region from the Tigon board. The argument is *struct tg_mem*.

ALT_WRITE_TG_MEM Write to the requested memory region on the Tigon board. The argument is *struct tg_mem*.

ALT_READ_TG_REG Read the requested register from the Tigon board. The argument is *struct tg_reg*.

ALT_WRITE_TG_REG Write to the requested register on the Tigon board. The argument is *struct tg_reg*.

FILES

/dev/ti[0-255] Tigon driver character interface.

DIAGNOSTICS

ti%d: couldn't map memory A fatal initialization error has occurred.

ti%d: couldn't map interrupt A fatal initialization error has occurred.

ti%d: no memory for softc struct! The driver failed to allocate memory for per-device instance information during initialization.

ti%d: failed to enable memory mapping! The driver failed to initialize PCI shared memory mapping. This might happen if the card is not in a bus-master slot.

ti%d: no memory for jumbo buffers! The driver failed to allocate memory for jumbo frames during initialization.

ti%d: bios thinks we're in a 64 bit slot, but we aren't The BIOS has programmed the NIC as though it had been installed in a 64-bit PCI slot, but in fact the NIC is in a 32-bit slot. This happens as a result of a bug in some BIOSes. This can be worked around on the Tigon II, but on the Tigon I initialization will fail.

ti%d: board self-diagnostics failed! The ROMFAIL bit in the CPU state register was set after system startup, indicating that the on-board NIC diagnostics failed.

ti%d: unknown hwrev The driver detected a board with an unsupported hardware revision. The **ti** driver supports revision 4 (Tigon 1) and revision 6 (Tigon 2) chips and has firmware only for those devices.

ti%d: watchdog timeout The device has stopped responding to the network, or there is a problem with the network connection (cable).

SEE ALSO

sendfile(2), altq(4), arp(4), netintro(4), ng_ether(4), vlan(4), ifconfig(8), zero_copy(9)

HISTORY

The **ti** device driver first appeared in FreeBSD 3.0.

AUTHORS

The **ti** driver was written by Bill Paul <*wpaul@bsdi.com*>. The header splitting firmware modifications, character ioctl(2) interface and debugging support were written by Kenneth Merry <*ken@FreeBSD.org*>. Initial zero copy support was written by Andrew Gallatin <*gallatin@FreeBSD.org*>.