Linux* Base Driver for the Intel® PRO/1000 Family of Adapters

Overview

This file describes the Linux* Base Driver for the Intel® PRO/1000 Family of Adapters, version 6.0.x. This driver supports the 2.4.x and 2.6.x kernels. This driver includes support for Itanium®2-based systems.

This driver is only supported as a loadable module at this time. Intel is not supplying patches against the kernel source to allow for static linking of the driver. For questions related to hardware requirements, refer to the documentation supplied with your Intel PRO/1000 adapter. All hardware requirements listed apply to use with Linux.

Native VLANs are now available with supported kernels.

The driver information previously displayed in the /proc filesystem is not supported in this release. Alternatively, you can use ethtool (version 1.6 or later), lspci, and ifconfig to obtain the same information. Instructions on updating ethtool can be found in the section Additional Configurations later in this document.

Identifying Your Adapter

For more information on how to identify your adapter, go to the Adapter & Driver ID Guide at:

> http://support.intel.com/support/network/adapter/pro100/21397.htm

For the latest Intel network drivers for Linux, refer to the following website. In the search field, enter your adapter name or type, or use the networking link on the left to search for your adapter:

> http://downloadfinder.intel.com/scripts-df/support_intel.asp

Building and Installation

To build a binary RPM* package of this driver, run 'rpmbuild -tb <filename.tar.gz>'. Replace <filename.tar.gz> with the specific file name of the driver.

***NOTES**:*

- For the build to work properly, the currently running kernel MUST match the version and configuration of the installed kernel sources. If you have just recompiled the kernel reboot the system now.
- RPM functionality has only been tested in Red Hat distributions.

1. Move the base driver tar file to the directory of your choice. For example, use '/home/username/e1000' or '/usr/local/src/e1000'.
2. Untar/unzip the archive, where <x.x.x> is the version number for the driver tar file:

   ```
   tar zxf e1000-<x.x.x>.tar.gz
   ```

3. Change to the driver src directory, where <x.x.x> is the version number for the driver tar:

   ```
   cd e1000-<x.x.x>/src/
   ```

4. Compile the driver module:

   ```
   make install
   ```

   The binary will be installed as:

   ```
   /lib/modules/<KERNEL VERSION>/kernel/drivers/net/e1000/e1000.[k]o
   ```

   The install locations listed above are the default locations. They might not be correct for certain Linux distributions. For more information, go to ldistrib.htm.

5. Install the module:

   ```
   insmod e1000 <parameter>=<value>
   ```

6. Assign an IP address to the interface by entering the following, where <x> is the interface number:

   ```
   ifconfig eth<x> <IP_address>
   ```

7. Verify that the interface works. Enter the following, where <IP_address> is the IP address for another machine on the same subnet as the interface that is being tested:

   ```
   ping <IP_address>
   ```

Command Line Parameters

If the driver is built as a module, the following optional parameters are used by entering them on the command line with the modprobe or insmod command using this syntax:

```
modprobe e1000 [<option>=<VAL1>,<VAL2>,...]

insmod e1000 [<option>=<VAL1>,<VAL2>,...]
```

For example, with two PRO/1000 PCI adapters, entering:

```
insmod e1000 TxDescriptors=80,128
```

loads the e1000 driver with 80 TX descriptors for the first adapter and 128 TX descriptors for the second adapter.

The default value for each parameter is generally the recommended setting, unless otherwise noted.

NOTES:

- For more information about the AutoNeg, Duplex, and Speed parameters, see the Speed and Duplex Configuration section in this document.
- For more information about the InterruptThrottleRate, RxIntDelay, TxIntDelay, RxAbsIntDelay, and TxAbsIntDelay parameters, see the application note at: http://www.intel.com/design/network/applnots/ap450.htm.
- A descriptor describes a data buffer and attributes related to the data buffer. This information is accessed by the hardware.

Parameter Name	Valid Range/Settings	Default	Description
AutoNeg	0x01-0x0F, 0x20-0x2F	0x2F	This parameter is a bit mask that specifies which speed and duplex settings the board advertises. When this parameter is used, the Speed and Duplex parameters must not be specified. (adapters using copper connections only) ***NOTE:*** Refer to the Speed and Duplex section of this readme for more information on the AutoNeg parameter.
Duplex	0-2 (0=auto-negotiate, 1=half, 2=full)	0	Defines the direction in which data is allowed to flow. Can be either one or two-directional. If both Duplex and the link partner are set to auto-negotiate, the board auto-detects the correct duplex. If the link partner is forced (either full or half), Duplex defaults to half-duplex. (adapters using copper connections only)

FlowControl	0-3 (0=none, 1=Rx only, 2=Tx only, 3=Rx&Tx)	Read flow control settings from the EEPROM	This parameter controls the automatic generation(Tx) and response(Rx) to Ethernet PAUSE frames.
InterruptThrottleRate	100-100000 (0=off, 1=dynamic)	8000	This value represents the maximum number of interrupts per second the controller generates. InterruptThrottleRate is another setting used in interrupt moderation. Dynamic mode uses a heuristic algorithm to adjust InterruptThrottleRate based on the current traffic load. Un-supported Adapters: InterruptThrottleRate is *not* supported by 82542, 82543, or 82544-based adapters. ***NOTE**:* InterruptThrottleRate takes precedence over the TxAbsIntDelay and RxAbsIntDelay parameters. In other words, minimizing the receive and/or transmit absolute delays does not force the controller to generate more interrupts than what the Interrupt Throttle Rate allows. ***CAUTION**:* If you are using the Intel PRO/1000 CT Network Connection (controller 82547), setting InterruptThrottleRate to a value greater than 75,000, may hang (stop transmitting) adapters under certain network conditions. If this occurs a NETDEV WATCHDOG message is logged in the system event log. In addition, the controller is automatically reset, restoring the network connection. To eliminate the potential for the hang, ensure that InterruptThrottleRate is set no greater than 75,000 and is not set to 0. ***NOTE**:* When e1000 is loaded with default settings and multiple adapters are in use simultaneously, the CPU utilization may increase non-linearly. In order to limit the CPU utilization without impacting the overall throughput, we recommend that you load the driver as follows:

			insmod e1000.o InterruptThrottleRate=3000,3000,3000 This sets the InterruptThrottleRate to 3000 interrupts/sec for the first, second, and third instances of the driver. The range of 2000 to 3000 interrupts per second works on a majority of systems and is a good starting point, but the optimal value will be platform-specific. If CPU utilization is not a concern, use RX_POLLING (NAPI) and default driver settings.
RxDescriptors	80-256 for 82542 and 82543-based adapters 80-4096 for all other supported adapters	256	This value is the number of receive descriptors allocated by the driver. Increasing this value allows the driver to buffer more incoming packets. Each descriptor is 16 bytes. A receive buffer is also allocated for each descriptor and can be either 2048, 4096, 8192, or 16384 bytes, depending on the MTU setting. The maximum MTU size is 16110. ***NOTE**:* MTU designates the frame size. It only needs to be set for Jumbo Frames. ***NOTE**:* Depending on the available system resources, the request for a higher number of receive descriptors may be denied. In this case, use a lower number.
RxIntDelay	0-65535 (0=off)	0	This value delays the generation of receive interrupts in units of 1.024 microseconds. Receive interrupt reduction can improve CPU efficiency if properly tuned for specific network traffic. Increasing this value adds extra latency to frame reception and can end up decreasing the throughput of TCP traffic. If the system is reporting dropped receives, this value may be set too high, causing the driver to run out of available receive descriptors. ***CAUTION**:* When setting RxIntDelay to a value other than 0, adapters may hang (stop transmitting) under certain network conditions. If this occurs a NETDEV WATCHDOG message is logged in the system event log. In addition, the controller is automatically reset, restoring the network

			connection. To eliminate the potential for the hang ensure that RxIntDelay is set to zero.
RxAbsIntDelay	0-65535 (0=off)	128	This value, in units of 1.024 microseconds, limits the delay in which a receive interrupt is generated. Useful only if RxIntDelay is non-zero, this value ensures that an interrupt is generated after the initial packet is received within the set amount of time. Proper tuning, along with RxIntDelay, may improve traffic throughput in specific network conditions. (82540, 82545, and later adapters only)
Speed	0, 10, 100, 1000	0	Speed forces the line speed to the specified value in megabits per second (Mbps). If this parameter is not specified or is set to 0 and the link partner is set to auto-negotiate, the board will auto-detect the correct speed. Duplex must also be set when Speed is set to either 10 or 100. (adapters using copper connections only)
TxDescriptors	80-256 for 82542 and 82543-based adapters 80-4096 for all other supported adapters	256	This value is the number of transmit descriptors allocated by the driver. Increasing this value allows the driver to queue more transmits. Each descriptor is 16 bytes.
TxIntDelay	0-65535 (0=off)	64	This value delays the generation of transmit interrupts in units of 1.024 microseconds. Transmit interrupt reduction can improve CPU efficiency if properly tuned for specific network traffic. If the system is reporting dropped transmits, this value may be set too high causing the driver to run out of available transmit descriptors.
TxAbsIntDelay	0-65535 (0=off)	64	This value, in units of 1.024 microseconds, limits the delay in which a transmit interrupt is generated. Useful only if TxIntDelay is non-zero, this value ensures that an interrupt is generated after the initial packet is sent on the wire within the set amount of time. Proper tuning, along with TxIntDelay, may improve traffic throughput in specific network conditions. (82540, 82545, and later adapters only)

XsumRX	0-1	1	A value of '1' indicates that the driver should enable IP checksum offload for received packets (both UDP and TCP) to the adapter hardware. (not available on the 82542-based adapter)

Speed and Duplex Configuration

Three keywords are used to control the speed and duplex configuration. These keywords are Speed, Duplex, and AutoNeg.

If the board uses a fiber interface, these keywords are ignored, and the fiber interface board only links at 1000 Mbps full-duplex.

For copper-based boards, the keywords interact as follows:

> The default operation is auto-negotiate. The board advertises all supported speed and duplex combinations, and it links at the highest common speed and duplex mode IF the link partner is set to auto-negotiate.
>
> If Speed = 1000, limited auto-negotiation is enabled and only 1000 Mbps is advertised (The 1000BaseT spec requires auto-negotiation.)
>
> If Speed = 10 or 100, then both Speed and Duplex should be set. Auto-negotiation is disabled, and the AutoNeg parameter is ignored. Partner SHOULD also be forced.

The AutoNeg parameter is used when more control is required over the auto-negotiation process. It should be used when you wish to control which speed
and duplex combinations are advertised during the auto-negotiation process.

The parameter may be specified as either a decimal or hexidecimal value as determined by the bitmap below.

Bit position	7	6	5	4	3	2	1	0
Decimal Value	128	64	32	16	8	4	2	1
Hex Value	80	40	20	10	8	4	2	1
Speed (Mbps):	N/A	N/A	1000	N/A	100	100	10	10
Duplex:				Full	Full	Half	Full	Half

Some examples of using AutoNeg:

modprobe e1000 AutoNeg=0x01 (Restricts autonegotiation to 10 Half)
modprobe e1000 AutoNeg=1 (Same as above)
modprobe e1000 AutoNeg=0x02 (Restricts autonegotiation to 10 Full)
modprobe e1000 AutoNeg=0x03 (Restricts autonegotiation to 10 Half or 10 Full)

```
modprobe e1000 AutoNeg=0x04 (Restricts autonegotiation to 100 Half)
modprobe e1000 AutoNeg=0x05 (Restricts autonegotiation to 10 Half or 100 Half)
modprobe e1000 AutoNeg=0x020 (Restricts autonegotiation to 1000 Full)
modprobe e1000 AutoNeg=32 (Same as above)
```

Note that when this parameter is used, Speed and Duplex must not be specified.

If the link partner is forced to a specific speed and duplex, then this parameter should not be used. Instead, use the Speed and Duplex parameters previously mentioned to force the adapter to the same speed and duplex.

Additional Configurations

Configuring the Driver on Different Distributions

Configuring a network driver to load properly when the system is started is distribution dependent. Typically, the configuration process involves adding an alias line to /etc/modules.conf as well as editing other system startup scripts and/or configuration files. Many popular Linux distributions ship with tools to make these changes for you. To learn the proper way to configure a network device for your system, refer to your distribution documentation. If during this process you are asked for the driver or module name, the name for the Linux Base Driver for the Intel PRO/1000 family of adapters is e1000.

As an example, if you install the e1000 driver for two PRO/1000 adapters (eth0 and eth1) and set the speed and duplex to 10full and 100half, add the following to modules.conf:

```
alias eth0 e1000
alias eth1 e1000
options e1000 Speed=10,100 Duplex=2,1
```

Viewing Link Messages

Link messages will not be displayed to the console if the distribution is restricting system messages. In order to see network driver link messages on your console, set dmesg to eight by entering the following:

```
dmesg -n 8
```

***NOTE**:* This setting is not saved across reboots.

Jumbo Frames

The driver supports Jumbo Frames for all adapters except 82542 and 82573-based adapters. Jumbo Frames support is enabled by changing the MTU to a value larger than the default of 1500. Use the ifconfig command to increase the MTU size. For example:

```
ifconfig eth<x> mtu 9000 up
```

NOTE: This setting is not saved across reboots. The setting change can be made permanent by adding:

MTU=9000
to the file /etc/sysconfig/network-scripts/ifcfg-eth<x> (RedHat distributions)

The maximum MTU setting for Jumbo Frames is 16110. This value coincides with the maximum Jumbo Frames size of 16128.

NOTES:

- Jumbo Frames are supported at 1000 Mbps only. Using Jumbo Frames at 10 or 100 Mbps may result in poor performance or loss of link.
- MTU designates the frame size. To enable Jumbo Frames increase the MTU size on the interface beyond 1500.
- The Intel PRO/1000 PM Network Connection does not support jumbo frames.

Ethtool

The driver utilizes the ethtool interface for driver configuration and diagnostics, as well as displaying statistical information. Ethtool version 1.6 or later is required for this functionality.

The latest release of ethtool can be found at: http://sourceforge.net/projects/gkernel.

NOTE: Ethtool 1.6 only supports a limited set of ethtool options. Support for a more complete ethtool feature set can be enabled by upgrading ethtool to ethtool-1.8.1.

Enabling Wake on LAN* (WoL)

WoL is configured through the Ethtool* utility. Ethtool is included with all versions of Red Hat after Red Hat 7.2. For other Linux distributions, download and install Ethtool from the following website: http://sourceforge.net/projects/gkernel.

For instructions on enabling WoL with Ethtool, refer to the website listed above.

WoL will be enabled on the system during the next shut down or reboot. For this driver version, in order to enable WoL, the e1000 driver must be loaded prior to shutting down or suspending the system.

NAPI

NAPI (Rx polling mode) is supported in the e1000 driver. NAPI is enabled or disabled based on the configuration of the kernel. To override the default, use the following compile-time flags.

To enable NAPI, compile the driver module, passing in a configuration option:

```
make CFLAGS_EXTRA=-DE1000_NAPI install
```

To disable NAPI, compile the driver module, passing in a configuration option:

```
make CFLAGS_EXTRA=-DE1000_NO_NAPI install
```

See http://www.cyberus.ca/~hadi/usenix-paper.tgz for more information on NAPI.

Known Issues

NOTE*:* For distribution-specific information, see ldistrib.htm.

Compiling the Driver

When trying to compile the driver by running make install, the following error may occur: "Linux kernel source not configured - missing version.h"

To solve this issue, create the version.h file by going to the Linux source tree and entering:

```
make include/linux/version.h
```

Jumbo Frames System Requirement

Memory allocation failures have been observed on Linux systems with 64 MB of RAM or less that are running Jumbo Frames. If you are using Jumbo Frames, your system may require more than the advertised minimum requirement of 64 MB of system memory.

Performance Degradation with Jumbo Frames

Degradation in throughput performance may be observed in some Jumbo frames environments. If this is observed, increasing the application's socket buffer size and/or increasing the /proc/sys/net/ipv4/tcp_*mem entry values may help. See the specific application manual and /usr/src/linux*/Documentation/networking /ip-sysctl.txt for more details.

Jumbo frames on Foundry BigIron 8000 switch

There is a known issue using Jumbo frames when connected to a Foundry BigIron 8000 switch. This is a 3rd party limitation. If you experience loss of packets,
lower the MTU size.

Multiple Interfaces on Same Ethernet Broadcast Network

Due to the default ARP behavior on Linux, it is not possible to have one system on two IP networks in the same Ethernet broadcast domain (non-partitioned switch) behave as expected. All Ethernet interfaces will respond to IP traffic for any IP address assigned to the system. This results in unbalanced receive traffic.

If you have multiple interfaces in a server, either turn on ARP filtering by entering:

```
echo 1 > /proc/sys/net/ipv4/conf/all/arp_filter
(this only works if your kernel's version is higher than 2.4.5)
```

NOTE: This setting is not saved across reboots. The configuration change can be made permanent by adding the line:

net.ipv4.conf.all.arp_filter = 1
to the file /etc/sysctl.conf

or,

install the interfaces in separate broadcast domains (either in different switches or in a switch partitioned to VLANs).

82541/82547 can't link or is slow to link with some link partners

There is a known compatibility issue with 82541/82547 and some low-end switches where the link will not be established, or will be slow to establish. In particular, these switches are known to be incompatible with 82541/82547:

Planex FXG-08TE
I-O Data ETG-SH8

To workaround this issue, the driver can be compiled with an override of the PHY's master/slave setting. Forcing master or forcing slave mode will improve time-to-link.

```
# make CFLAGS_EXTRA=-DE1000_MASTER_SLAVE=<n>
```

Where <n> is:

0 = Hardware default
1 = Master mode
2 = Slave mode
3 = Auto master/slave

Disable rx flow control with ethtool

In order to disable receive flow control using ethtool on the following adapters, you must turn off auto-negotiation on the same command line.

For example:

```
ethtool -A eth? autoneg off rx off
```

Last modified on 3/09/05 10:30a Revision 6