

Linux Base Driver for the Intel(R) PRO/100 Family of Adapters

June 1, 2018

Contents

- In This Release
- Identifying Your Adapter
- Building and Installation
- Driver Configuration Parameters
- Additional Configurations
- Known Issues
- Support

In This Release

This file describes the Linux Base Driver for the Intel(R) PRO/100 Family of Adapters. This driver includes support for Itanium(R)2-based systems.

For questions related to hardware requirements, refer to the documentation supplied with your Intel PRO/100 adapter.

The following features are now available in supported kernels:

- Native VLANs
- Channel Bonding (teaming)
- SNMP

Channel Bonding documentation can be found in the Linux kernel source: /Documentation/networking/bonding.rst

Identifying Your Adapter

For information on how to identify your adapter, and for the latest Intel network drivers, refer to the Intel Support website: <https://www.intel.com/support>

Driver Configuration Parameters

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

Rx Descriptors:

Number of receive descriptors. A receive descriptor is a data structure that describes a receive buffer and its attributes to the network controller. The data in the descriptor is used by the controller to write data from the controller to host memory. In the 3.x.x driver the valid range for this parameter is 64-256. The default value is 256. This parameter can be changed using the command:

```
ethtool -G eth? rx n
```

Where n is the number of desired Rx descriptors.

Tx Descriptors:

Number of transmit descriptors. A transmit descriptor is a data structure that describes a transmit buffer and its attributes to the network controller. The data in the descriptor is used by the controller to read data from the host memory to the controller. In the 3.x.x driver the valid range for this parameter is 64-256. The default value is 128. This parameter can be changed using the command:

```
ethtool -G eth? tx n
```

Where n is the number of desired Tx descriptors.

Speed/Duplex:

The driver auto-negotiates the link speed and duplex settings by default. The ethtool utility can be used as follows to force speed/duplex.:

```
ethtool -s eth? autoneg off speed {10|100} duplex {full|half}
```

NOTE: setting the speed/duplex to incorrect values will cause the link to fail.

Event Log Message Level:

The driver uses the message level flag to log events to syslog. The message level can be set at driver load time. It can also be set using the command:

```
ethtool -s eth? msglvl n
```

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/modprobe.d/*.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/100 Family of Adapters is `e100`.

As an example, if you install the `e100` driver for two PRO/100 adapters (`eth0` and `eth1`), add the following to a configuration file in `/etc/modprobe.d/`:

```
alias eth0 e100
alias eth1 e100
```

Viewing Link Messages

In order to see link messages and other Intel driver information on your console, you must set the `dmesg` level up to six. This can be done by entering the following on the command line before loading the `e100` driver:

```
dmesg -n 6
```

If you wish to see all messages issued by the driver, including debug messages, set the `dmesg` level to eight.

NOTE: This setting is not saved across reboots.

ethtool

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

The latest release of `ethtool` can be found from <https://www.kernel.org/pub/software/network/ethtool/>

Enabling Wake on LAN (WoL)

WoL is provided through the `ethtool` utility. For instructions on enabling WoL with `ethtool`, refer to the `ethtool` man page. WoL will be enabled on the system during the next shut down or reboot. For this driver version, in order to enable WoL, the `e100` driver must be loaded when shutting down or rebooting the system.

NAPI

NAPI (Rx polling mode) is supported in the `e100` driver.

See <https://wiki.linuxfoundation.org/networking/napi> for more information on NAPI.

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

1. 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), or

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

Support

For general information, go to the Intel support website at: <https://www.intel.com/support/>

or the Intel Wired Networking project hosted by Sourceforge at: <http://sourceforge.net/projects/e1000> If an issue is identified with the released source code on a supported kernel with a supported adapter, email the specific information related to the issue to e1000-devel@lists.sf.net.