

# Intel e1000e Device Driver Version 3.8.4 Installation on RedHawk OS

## Release Notes

November 2, 2021



# 1. Introduction:

本書は、RedHawk に移植済の Intel e1000e version 3.8.4 デバイスドライバ・リリースノートである。 .

## 2. Requirements:

このデバイスドライバをインストールする OS は、RedHawk 64 bits を想定している。

## 3. Installation:

本パッケージは、rpm バイナリで提供される。

以下の手順で、rpm ファイルをインストールする。

ドライバパッケージは RedHawk 各カーネルフレーバー用の、ドライバディレクトリ下にインストールされ、必要に応じて TRACE,DEBUG,STANDARD,KDUMP の initramfs カーネルモジュールが自動生成される。

このディレクトリは、RedHawk7.x では、

“/lib/modules`uname -r`/updates/drivers/net/ethernet/intel/e1000e/” である。

以下にインストール例を示す。

```
# mount /dev/dvd /mnt
# cd /mnt

# rpm -ivh e1000e-2.16.11-1.RedHawk-7.2.x86_64.rpm
準備しています... ##### [100%]
更新中 / インストール中...
  1:e1000e-2.16.11-1 ##### [100%]
original pci.ids saved in /usr/local/share/e1000e
original driver saved in /usr/local/src/e1000e-3.8.4
Installing e1000e-1.6.42 drivers succeeded!
```

なお、README などの、ファイルは、/usr/share/doc/e1000e-2.16.11 ディレクトリに、バックアップソースコードは/usr/local/src/e1000e-3.8.4 ディレクトリにインストールされる。

以下に、rpm ファイルで、供給されるファイル例を示す。

```
# rpm -qpl e1000e-3.8.4-1.RedHawk-7.2.x86_64.rpm
/lib/modules/4.1.15-rt17-RedHawk-7.2-debug/updates/drivers/net/ethernet/intel/e1000e/e1000e.ko.new
/lib/modules/4.1.15-rt17-RedHawk-7.2-kdump/updates/drivers/net/ethernet/intel/e1000e/e1000e.ko.new
/lib/modules/4.1.15-rt17-RedHawk-7.2-trace/updates/drivers/net/ethernet/intel/e1000e/e1000e.ko.new
/lib/modules/4.1.15-rt17-RedHawk-7.2/updates/drivers/net/ethernet/intel/e1000e/e1000e.ko.new
/usr/local/src/e1000e-3.8.4/e1000e-3.8.4.src.tar.gz
/usr/share/doc/e1000e-3.8.4
/usr/share/doc/e1000e-3.8.4/COPYING
/usr/share/doc/e1000e-3.8.4/README
/usr/share/doc/e1000e-3.8.4/file.list
/usr/share/doc/e1000e-3.8.4/pci.updates
/usr/share/man/man7/e1000e.7.gz
```

## 4. Building driver on a currently running RedHawk kernel

インストール後、カーネルモジュールおよび、ソースコードは、新しい版に入れ替えられているため、実際に動作しているデバイスドライバを、以下のコマンドで確認できる。

```
# modinfo e1000e
filename:    /lib/modules/4.1.15-rt17-RedHawk-7.2-trace/updates/drivers/net/ethernet/intel/e1000e/e1000e.ko
version:     3.8.4-NAPI
license:     GPL
description: Intel(R) PRO/1000 Network Driver
author:      Intel Corporation, <linux.nics@intel.com>
srcversion:  E6A1C79D652D933F24B8587
alias:       pci:v00008086d00001A1Dsv*sd*bc*sc*i*
alias:       pci:v00008086d00001A1Csv*sd*bc*sc*i*
alias:       pci:v00008086d00001A1Fsv*sd*bc*sc*i*
alias:       pci:v00008086d00001A1Esv*sd*bc*sc*i*
alias:       pci:v00008086d000015F5sv*sd*bc*sc*i*
alias:       pci:v00008086d000015F4sv*sd*bc*sc*i*
alias:       pci:v00008086d000015FAsv*sd*bc*sc*i*
alias:       pci:v00008086d000015F9sv*sd*bc*sc*i*
alias:       pci:v00008086d000015FCsv*sd*bc*sc*i*
alias:       pci:v00008086d000015FBsv*sd*bc*sc*i*
alias:       pci:v00008086d0000D4Dsv*sd*bc*sc*i*
alias:       pci:v00008086d0000D4Csv*sd*bc*sc*i*
alias:       pci:v00008086d0000D4Fsv*sd*bc*sc*i*
alias:       pci:v00008086d0000D4Esv*sd*bc*sc*i*
alias:       pci:v00008086d000015E2sv*sd*bc*sc*i*
alias:       pci:v00008086d000015E1sv*sd*bc*sc*i*
alias:       pci:v00008086d000015E0sv*sd*bc*sc*i*
alias:       pci:v00008086d000015DFsv*sd*bc*sc*i*
alias:       pci:v00008086d000015BCsv*sd*bc*sc*i*
alias:       pci:v00008086d000015BBsv*sd*bc*sc*i*
alias:       pci:v00008086d000015BEsv*sd*bc*sc*i*
alias:       pci:v00008086d000015BDsv*sd*bc*sc*i*
alias:       pci:v00008086d0000D55sv*sd*bc*sc*i*
alias:       pci:v00008086d0000D53sv*sd*bc*sc*i*
alias:       pci:v00008086d000015D6sv*sd*bc*sc*i*
alias:       pci:v00008086d000015E3sv*sd*bc*sc*i*
alias:       pci:v00008086d000015D8sv*sd*bc*sc*i*
alias:       pci:v00008086d000015D7sv*sd*bc*sc*i*
alias:       pci:v00008086d000015B9sv*sd*bc*sc*i*
alias:       pci:v00008086d000015B8sv*sd*bc*sc*i*
alias:       pci:v00008086d000015B7sv*sd*bc*sc*i*
alias:       pci:v00008086d00001570sv*sd*bc*sc*i*
alias:       pci:v00008086d0000156Fsv*sd*bc*sc*i*
alias:       pci:v00008086d000015A3sv*sd*bc*sc*i*
alias:       pci:v00008086d000015A2sv*sd*bc*sc*i*
alias:       pci:v00008086d000015A1sv*sd*bc*sc*i*
alias:       pci:v00008086d000015A0sv*sd*bc*sc*i*
alias:       pci:v00008086d00001559sv*sd*bc*sc*i*
alias:       pci:v00008086d0000155Asv*sd*bc*sc*i*
alias:       pci:v00008086d0000153Bsv*sd*bc*sc*i*
alias:       pci:v00008086d0000153Asv*sd*bc*sc*i*
alias:       pci:v00008086d00001503sv*sd*bc*sc*i*
alias:       pci:v00008086d00001502sv*sd*bc*sc*i*
alias:       pci:v00008086d000010F0sv*sd*bc*sc*i*
alias:       pci:v00008086d000010EFsv*sd*bc*sc*i*
alias:       pci:v00008086d000010EBsv*sd*bc*sc*i*
alias:       pci:v00008086d000010EAsv*sd*bc*sc*i*
alias:       pci:v00008086d00001525sv*sd*bc*sc*i*
alias:       pci:v00008086d000010DFsv*sd*bc*sc*i*
alias:       pci:v00008086d000010DEsv*sd*bc*sc*i*
alias:       pci:v00008086d000010CEsv*sd*bc*sc*i*
alias:       pci:v00008086d000010CDsv*sd*bc*sc*i*
alias:       pci:v00008086d000010CCsv*sd*bc*sc*i*
alias:       pci:v00008086d000010CBsv*sd*bc*sc*i*
alias:       pci:v00008086d000010F5sv*sd*bc*sc*i*
alias:       pci:v00008086d000010BFsv*sd*bc*sc*i*
alias:       pci:v00008086d000010E5sv*sd*bc*sc*i*
alias:       pci:v00008086d0000294Csv*sd*bc*sc*i*
alias:       pci:v00008086d000010BDsv*sd*bc*sc*i*
alias:       pci:v00008086d000010C3sv*sd*bc*sc*i*
alias:       pci:v00008086d000010C2sv*sd*bc*sc*i*
alias:       pci:v00008086d000010C0sv*sd*bc*sc*i*
alias:       pci:v00008086d00001501sv*sd*bc*sc*i*
```

```

alias: pci:v00008086d00001049sv*sd*bc*sc*i*
alias: pci:v00008086d0000104Dsv*sd*bc*sc*i*
alias: pci:v00008086d0000104Bsv*sd*bc*sc*i*
alias: pci:v00008086d0000104Asv*sd*bc*sc*i*
alias: pci:v00008086d000010C4sv*sd*bc*sc*i*
alias: pci:v00008086d000010C5sv*sd*bc*sc*i*
alias: pci:v00008086d0000104Csv*sd*bc*sc*i*
alias: pci:v00008086d000010BBsv*sd*bc*sc*i*
alias: pci:v00008086d00001098sv*sd*bc*sc*i*
alias: pci:v00008086d000010BASv*sd*bc*sc*i*
alias: pci:v00008086d00001096sv*sd*bc*sc*i*
alias: pci:v00008086d0000150Csv*sd*bc*sc*i*
alias: pci:v00008086d000010F6sv*sd*bc*sc*i*
alias: pci:v00008086d000010D3sv*sd*bc*sc*i*
alias: pci:v00008086d0000109Asv*sd*bc*sc*i*
alias: pci:v00008086d0000108Csv*sd*bc*sc*i*
alias: pci:v00008086d0000108Bsv*sd*bc*sc*i*
alias: pci:v00008086d0000107Fsv*sd*bc*sc*i*
alias: pci:v00008086d0000107Esv*sd*bc*sc*i*
alias: pci:v00008086d0000107Dsv*sd*bc*sc*i*
alias: pci:v00008086d000010B9sv*sd*bc*sc*i*
alias: pci:v00008086d000010D5sv*sd*bc*sc*i*
alias: pci:v00008086d000010DAsv*sd*bc*sc*i*
alias: pci:v00008086d000010D9sv*sd*bc*sc*i*
alias: pci:v00008086d00001060sv*sd*bc*sc*i*
alias: pci:v00008086d000010A5sv*sd*bc*sc*i*
alias: pci:v00008086d000010BCsv*sd*bc*sc*i*
alias: pci:v00008086d000010A4sv*sd*bc*sc*i*
alias: pci:v00008086d0000105Fsv*sd*bc*sc*i*
alias: pci:v00008086d0000105Esv*sd*bc*sc*i*
depends:
vermagic: 4.1.15-rt17-RedHawk-7.2-trace SMP preempt mod_unload
parm: copybreak:Maximum size of packet that is copied to a new buffer on receive (uint)
parm: TxIntDelay:Transmit Interrupt Delay (array of int)
parm: TxAbsIntDelay:Transmit Absolute Interrupt Delay (array of int)
parm: RxIntDelay:Receive Interrupt Delay (array of int)
parm: RxAbsIntDelay:Receive Absolute Interrupt Delay (array of int)
parm: InterruptThrottleRate:Interrupt Throttling Rate (array of int)
parm: IntMode:Interrupt Mode (array of int)
parm: SmartPowerDownEnable:Enable PHY smart power down (array of int)
parm: KumeranLockLoss:Enable Kumeran lock loss workaround (array of int)
parm: CrcStripping:Enable CRC Stripping, disable if your BMC needs the CRC (array of int)
parm: EEE:Enable/disable on parts that support the feature (array of int)
parm: Node:[ROUTING] Node to allocate memory on, default -1 (array of int)
parm: debug:Debug level (0=none, ..., 16=all) (int)

```

また、通常の下記マニュアル手順で、カーネルの再構築を行うことも可能である。

```

# cd /lib/modules/ `uname -r`/build
# ./ccur-config -c -n
# make -C `pwd` SUBDIRS=`pwd`/drivers/net/ethernet/intel/e1000e REDHAWKFLAVOR=`cat /proc/ccur/flavor` modules
# make -C `pwd` SUBDIRS=`pwd`/drivers/net/ethernet/intel/e1000e REDHAWKFLAVOR=`cat /proc/ccur/flavor` modules_install

```

但し、このマニュアル手順による方法では、オリジナルカーネルモジュールのファイルの上書きを行うため、本 **rpm** パッケージとモジュールをインストールする位置が異なる。このため、整合性に注意しなければならない。

なお、システムを再構築後、下記手順で **initramfs** カーネルモジュールを組み込むことができる。

```
# dracut --add-drivers " dca hwmon ixgbe" -f /boot/initramfs-`uname -r`.img `uname -r`
```

正常に組み込むことが出来ると、下記のメッセージがコンソールに表示される。

```

e1000e: Intel(R) PRO/1000 Network Driver - 3.8.4
e1000e: Copyright(c) 1999 - 2020 Intel Corporation.

```

## 5. Remove driver

**rpm** パッケージを削除するためには、以下のコマンドを使用する。  
インストール時に作成されたバックアップファイルを使用し、元の状態に戻される。

```

# rpm -e e1000e-3.8.4-1
Uninstalling e1000e-2.16.11-1 drivers succeeded!

```

## 6. README

e1000e Linux\* Base Driver for Intel(R) Network Connection

=====

May 15, 2019

=====

### Contents

=====

- Overview
- Identifying Your Adapter
- Building and Installation
- Command Line Parameters
- Additional Features and Configurations
- Speed and Duplex Configuration
- Known Issues
- Support
- License

### Overview

=====

This driver supports kernel versions 2.4.x, 2.6.x and later.

Driver information can be obtained using ethtool, lspci, and ifconfig. Instructions on updating ethtool can be found in the section Additional Configurations later in this document.

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 drivers.

For questions related to hardware requirements, refer to the documentation supplied with your Intel adapter. All hardware requirements listed apply to use with Linux.

NOTE: The Intel(R) 82562v 10/100 Network Connection only provides 10/100 support.

### Upgrading

-----

If you currently have the e1000 driver installed and need to install e1000e, perform the following:

- If your version of e1000 is 7.6.15.5 or less, upgrade to e1000 version 8.x, using the instructions in the e1000 README.
- Install the e1000e driver using the instructions in the Building and Installation section below.
- Modify /etc/modprobe.conf to point your PCIe devices to use the new e1000e driver using alias <ethX> e1000e, or use your distribution's specific method for configuring network adapters like RedHat's setup/system-config-network or SuSE's yast2.

## Identifying Your Adapter

=====

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

## Building and Installation

=====

To build a binary RPM package of this driver

-----

Note: RPM functionality has only been tested in Red Hat distributions.

1. Run the following command, where <x.x.x> is the version number for the driver tar file.

```
# rpmbuild -tb e1000e-<x.x.x>.tar.gz
```

NOTE: 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 before building.

2. After building the RPM, the last few lines of the tool output contain the location of the RPM file that was built. Install the RPM with one of the following commands, where <RPM> is the location of the RPM file:

```
# rpm -Uvh <RPM>
or
# dnf/yum localinstall <RPM>
```

### NOTES:

- To compile the driver on some kernel/arch combinations, you may need to install a package with the development version of libelf (e.g. libelf-dev, libelf-devel, elfutils-libelf-devel).
- When compiling an out-of-tree driver, details will vary by distribution. However, you will usually need a kernel-devel RPM or some RPM that provides the kernel headers at a minimum. The RPM kernel-devel will usually fill in the link at /lib/modules/`uname -r`/build.

To manually build the driver

-----

1. Move the base driver tar file to the directory of your choice.  
For example, use '/home/username/e1000e' or '/usr/local/src/e1000e'.
2. Untar/unzip the archive, where <x.x.x> is the version number for the driver tar file:

```
# tar xzf e1000e-<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 e1000e-<x.x.x>/src/
```

4. Compile the driver module:

```
# make install
```

The binary will be installed as:

```
/lib/modules/<KERNEL VER>/updates/drivers/net/ethernet/intel/e1000e/e1000e.ko
```

The install location listed above is the default location. This may differ for various Linux distributions.

5. Load the module using the modprobe command.

To check the version of the driver and then load it:

```
# modinfo e1000e
# modprobe e1000e [parameter=port1_value,port2_value]
```

Alternately, make sure that any older e1000e drivers are removed from the kernel before loading the new module:

```
# rmmod e1000e; modprobe e1000e
```

6. Assign an IP address to the interface by entering the following, where <ethX> is the interface name that was shown in dmesg after modprobe:

```
# ip address add <IP_address>/<netmask bits> dev <ethX>
```

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>
```

Note: For certain distributions like (but not limited to) Red Hat Enterprise Linux 7 and Ubuntu, once the driver is installed, you may need to update the initrd/initramfs file to prevent the OS loading old versions of the e1000e driver. Use the dracut utility on Red Hat distributions:

```
# dracut --force
```

For Ubuntu:

```
# update-initramfs -u
```

## 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 command using this syntax:

```
# modprobe e1000e [<option>=<VAL1>,<VAL2>,...]
```

There needs to be a <VAL#> for each network port in the system supported by this driver. The values will be applied to each instance, in function order. For example:

```
# modprobe e1000e InterruptThrottleRate=16000,16000
```

In this case, there are two network ports supported by e1000e in the system. The default value for each parameter is generally the recommended setting, unless otherwise noted.

NOTE: For more information about the command line parameters, see the

application note at: <http://www.intel.com/design/network/aplnots/ap450.htm>.

NOTE: A descriptor describes a data buffer and attributes related to the data buffer. This information is accessed by the hardware.

#### InterruptThrottleRate

-----

Valid Range:

0=off

1=dynamic

4=simplified balancing

<min\_ITR>-<max\_ITR>

Interrupt Throttle Rate controls the number of interrupts each interrupt vector can generate per second. Increasing ITR lowers latency at the cost of increased CPU utilization, though it may help throughput in some circumstances.

0 = Setting InterruptThrottleRate to 0 turns off any interrupt moderation and may improve small packet latency. However, this is generally not suitable for bulk throughput traffic due to the increased CPU utilization of the higher interrupt rate.

1 = Setting InterruptThrottleRate to Dynamic mode attempts to moderate interrupts per vector while maintaining very low latency. This can sometimes cause extra CPU utilization. If planning on deploying e1000e in a latency sensitive environment, this parameter should be considered.

<min\_ITR>-<max\_ITR> =

Setting InterruptThrottleRate to a value greater or equal to <min\_ITR> will program the adapter to send at most that many interrupts per second, even if more packets have come in. This reduces interrupt load on the system and can lower CPU utilization under heavy load, but will increase latency as packets are not processed as quickly.

#### 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.

#### RxIntDelay

-----

Valid Range: 0-65535 (0=off)

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 0.

#### RxAbsIntDelay

-----

Valid Range: 0-65535 (0=off)

This value, in units of 1.024 microseconds, limits the delay in which a receive interrupt is generated. This value ensures that an interrupt is generated after the initial packet is received within the set amount of time, which is useful only if RxIntDelay is non-zero. Proper tuning, along with



RxIntDelay, may improve traffic throughput in specific network conditions.

#### TxIntDelay

-----

Valid Range: 0-65535 (0=off)

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

-----

Valid Range: 0-65535 (0=off)

This value, in units of 1.024 microseconds, limits the delay in which a transmit interrupt is generated. It is useful only if TxIntDelay is non-zero. It 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.

#### copybreak

-----

Valid Range: 0-xxxxxxx (0=off)

The driver copies all packets below or equaling this size to a fresh receive buffer before handing it up the stack.

This parameter differs from other parameters because it is a single (not 1,1,1 etc.) parameter applied to all driver instances and it is also available during runtime at /sys/module/e1000e/parameters/copybreak.

To use copybreak, type:

```
# modprobe e1000e.ko copybreak=128
```

#### SmartPowerDownEnable

-----

Valid Range: 0-1

Allows Phy to turn off in lower power states. The user can turn off this parameter in supported chipsets.

#### KumeranLockLoss

-----

Valid Range: 0-1

This workaround skips resetting the Phy at shutdown for the initial silicon releases of ICH8 systems.

#### IntMode

-----

Valid Range: 0-2 (0 = Legacy Int, 1 = MSI and 2 = MSI-X)

IntMode controls the allowed load time control over the type of interrupt registered for by the driver. MSI-X is required for multiple queue support, and some kernels and combinations of kernel .config options will force a lower level of interrupt support.

'cat /proc/interrupts' will show different values for each type of interrupt.

## CrcStripping

-----

Valid Range: 0-1

Strip the CRC from received packets before sending up the network stack. If you have a machine with a BMC enabled but cannot receive IPMI traffic after loading or enabling the driver, try disabling this feature.

## EEE (Energy Efficient Ethernet)

-----

Valid Range: 0-1

0 = Disables EEE

1 = Enables EEE

A link between two EEE-compliant devices will result in periodic bursts of data followed by periods where the link is in an idle state. This Low Power Idle (LPI) state is supported at 1 Gbps and 100 Mbps link speeds.

### NOTES:

- EEE support requires auto-negotiation.
- Both link partners must support EEE.
- EEE is not supported on all Intel(R) Ethernet Network devices or at all link speeds.

Example:

```
# ethtool --show-eee <ethX>
# ethtool --set-eee <ethX> [eee on|off]
```

## Node

----

Valid Range: 0-n

0 - n: where n is the number of the NUMA node that should be used to allocate memory for this adapter port.

-1: uses the driver default of allocating memory on whichever processor is running modprobe.

The Node parameter allows you to choose which NUMA node you want to have the adapter allocate memory from. All driver structures, in-memory queues, and receive buffers will be allocated on the node specified. This parameter is only useful when interrupt affinity is specified; otherwise, part of the interrupt time could run on a different core than where the memory is allocated causing slower memory access and impacting throughput, CPU, or both.

## Additional Features and Configurations

=====

### ethtool

-----

The driver utilizes the ethtool interface for driver configuration and diagnostics, as well as displaying statistical information. The latest ethtool version is required for this functionality. Download it at:  
<https://kernel.org/pub/software/network/ethtool/>

NOTE: When validating enable/disable tests on some parts (for example, 82578), it is necessary to add a few seconds between tests when working with ethtool.

## 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.

## IEEE 1588 Precision Time Protocol (PTP) Hardware Clock (PHC)

Precision Time Protocol (PTP) is used to synchronize clocks in a computer network. PTP support varies among Intel devices that support this driver. Use 'ethtool -T <ethX>' to get a definitive list of PTP capabilities supported by the device.

E1000E\_PTP is a compile time flag. The user can enable it at compile time to add support for PTP from the driver. The flag is used by editing the make file as follows when it is being compiled:

```
# make CFLAGS_EXTRA="-DE1000E_PTP" install
```

## 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 or /etc/modprobe.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 Base Driver is e1000e.

For example, if you install the e1000e driver for two adapters (eth0 and eth1) and want to set the interrupt mode to MSI-X and MSI, respectively, add the following to modules.conf or /etc/modprobe.conf:

```
alias eth0 e1000e
alias eth1 e1000e
options e1000e IntMode=2,1
```

## Jumbo Frames

Jumbo Frames support is enabled by changing the Maximum Transmission Unit (MTU) to a value larger than the default value of 1500.

Use the ifconfig command to increase the MTU size. For example, enter the following where <ethX> is the interface number:

```
# ifconfig <ethX> mtu 9000 up
```

Alternatively, you can use the ip command as follows:

```
# ip link set mtu 9000 dev <ethX>
# ip link set up dev <ethX>
```

This setting is not saved across reboots. The setting change can be made

permanent by adding 'MTU=9000' to the following file:  
/etc/sysconfig/network-scripts/ifcfg-<ethX> for RHEL  
or  
/etc/sysconfig/network/<config\_file> for SLES

NOTE: The maximum MTU setting for Jumbo Frames is 8996. This value coincides with the maximum Jumbo Frames size of 9018 bytes.

NOTE: Using Jumbo frames at 10 or 100 Mbps is not supported and may result in poor performance or loss of link.

NOTE: Packet loss may have a greater impact on throughput when you use jumbo frames. If you observe a drop in performance after enabling jumbo frames, enabling flow control may mitigate the issue.

NOTE: The following adapters limit Jumbo Frames sized packets to a maximum of 4088 bytes:

- Intel(R) 82578DM Gigabit Network Connection
- Intel(R) 82577LM Gigabit Network Connection
- The following adapters do not support Jumbo Frames:
  - Intel(R) PRO/1000 Gigabit Server Adapter
  - Intel(R) PRO/1000 PM Network Connection
  - Intel(R) 82562G 10/100 Network Connection
  - Intel(R) 82562G-2 10/100 Network Connection
  - Intel(R) 82562GT 10/100 Network Connection
  - Intel(R) 82562GT-2 10/100 Network Connection
  - Intel(R) 82562V 10/100 Network Connection
  - Intel(R) 82562V-2 10/100 Network Connection
  - Intel(R) 82566DC Gigabit Network Connection
  - Intel(R) 82566DC-2 Gigabit Network Connection
  - Intel(R) 82566DM Gigabit Network Connection
  - Intel(R) 82566MC Gigabit Network Connection
  - Intel(R) 82566MM Gigabit Network Connection
  - Intel(R) 82567V-3 Gigabit Network Connection
  - Intel(R) 82577LC Gigabit Network Connection
  - Intel(R) 82578DC Gigabit Network Connection
- Jumbo Frames cannot be configured on an 82579-based Network device if MACSec is enabled on the system.

## Speed and Duplex Configuration

-----  
In addressing speed and duplex configuration issues, you need to distinguish between copper-based adapters and fiber-based adapters.

In the default mode, an Intel(R) Ethernet Network Adapter using copper connections will attempt to auto-negotiate with its link partner to determine the best setting. If the adapter cannot establish link with the link partner using auto-negotiation, you may need to manually configure the adapter and link partner to identical settings to establish link and pass packets. This should only be needed when attempting to link with an older switch that does not support auto-negotiation or one that has been forced to a specific speed or duplex mode. Your link partner must match the setting you choose. 1 Gbps speeds and higher cannot be forced. Use the autonegotiation advertising setting to manually set devices for 1 Gbps and higher.

Speed, duplex, and autonegotiation advertising are configured through the ethtool\* utility. ethtool is included with all versions of Red Hat after Red Hat 7.2. For the latest version, download and install ethtool from the following website:

<https://kernel.org/pub/software/network/ethtool/>

To see the speed configurations your device supports, run the following:

```
# ethtool <ethX>
```

Caution: Only experienced network administrators should force speed and duplex or change autonegotiation advertising manually. The settings at the switch must always match the adapter settings. Adapter performance may suffer or your adapter may not operate if you configure the adapter differently from your switch.

An Intel(R) Ethernet Network Adapter using fiber-based connections, however, will not attempt to auto-negotiate with its link partner since those adapters operate only in full duplex and only at their native speed.

#### Wake on LAN (WoL) Support

-----

Some adapters do not support Wake on LAN (WoL). To determine if your adapter supports WoL, run the following command:

```
# ethtool <ethX>
```

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:  
<https://kernel.org/pub/software/network/ethtool/>.

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

WoL will be enabled on the system during the next shutdown or reboot. For this driver version, in order to enable WoL, the e1000e driver must be loaded prior to shutting down or suspending the system.

NOTE: Wake on LAN is only supported on port A for the following devices:

- Intel(R) PRO/1000 PT Dual Port Network Connection
- Intel(R) PRO/1000 PT Dual Port Server Connection
- Intel(R) PRO/1000 PT Dual Port Server Adapter
- Intel(R) PRO/1000 PF Dual Port Server Adapter
- Intel(R) PRO/1000 PT Quad Port Server Adapter
- Intel(R) Gigabit PT Quad Port Server ExpressModule

#### NAPI

----

This driver supports NAPI (Rx polling mode).

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

```
# make CFLAGS_EXTRA=-DE1000E_NO_NAPI install
```

For more information on NAPI, see

<https://www.linuxfoundation.org/collaborate/workgroups/networking/napi>

#### Known Issues/Troubleshooting

=====

#### Hardware Issues

-----

For known hardware and troubleshooting issues, either refer to the "Release Notes" in your User Guide, or for more detailed information, go to <http://www.intel.com>.

In the search box enter your devices controller ID followed by "spec update" (i.e., 82599 spec update). The specification update file has complete information on known hardware issues.

## Software Issues

NOTE: After installing the driver, if your Intel Ethernet Network Connection is not working, verify that you have installed the correct driver.

Intel(R) Active Management Technology 2.0, 2.1, 2.5 are not supported in conjunction with the linux driver.

## Detected Tx Unit Hang in Quad Port Adapters

In some cases ports 3 and 4 don't pass traffic and report 'Detected Tx Unit Hang' followed by 'NETDEV WATCHDOG: <ethX>: transmit timed out' errors. Ports 1 and 2 do not show any errors and will pass traffic.

This issue may be resolved by updating to the latest kernel and BIOS. You should use an OS that fully supports Message Signaled Interrupts (MSI) and make sure that MSI is enabled in your system's BIOS.

## Adapters with 4 Ports Behind a PCIe Bridge

Adapters that have 4 ports behind a PCIe bridge may be incompatible with some systems. The user should run the Linux firmware kit from <http://www.linuxfirmwarekit.org/> to test their BIOS, if they have interrupt or "missing interface" problems, especially with older kernels.

## 82573(V/L/E) TX Unit Hang Messages

Several adapters with the 82573 chipset display "TX unit hang" messages during normal operation with the e1000edriver. The issue appears both with TSO enabled and disabled and is caused by a power management function that is enabled in the EEPROM. Early releases of the chipsets to vendors had the EEPROM bit that enabled the feature. After the issue was discovered newer adapters were released with the feature disabled in the EEPROM.

If you encounter the problem in an adapter, and the chipset is an 82573-based one, you can verify that your adapter needs the fix by using ethtool:

```
# ethtool -e eth0
```

```
Offset  Values
```

```
-----  -----  
0x0000  00 12 34 56 fe dc 30 0d 46 f7 f4 00 ff ff ff ff  
0x0010  ff ff ff ff 6b 02 8c 10 d9 15 8c 10 86 80 de 83  
                                     ^^
```

The value at offset 0x001e (de) has bit 0 unset. This enables the problematic power saving feature. In this case, the EEPROM needs to read "df" at offset 0x001e.

A one-time EEPROM fix is available as a shell script. This script will verify that the adapter is applicable to the fix and if the fix is needed or not. If the fix is required, it applies the change to the EEPROM and updates the checksum. The user must reboot the system after applying the fix if changes were made to the EEPROM.

Example output of the script:

```
# bash fixEEP-82573-dspd.sh eth0
eth0: is a "82573E Gigabit Ethernet Controller"
This fixup is applicable to your hardware executing command:
# ethtool -E eth0 magic 0x109a8086 offset 0x1e value 0xdf
Change made. You *MUST* reboot your machine before changes take effect!
The script can be downloaded at
http://e1000.sourceforge.net/files/fixEEP-82573-dspd.sh.
```

### Dropped Receive Packets on Half-duplex 10/100 Networks

-----

If you have an Intel PCI Express adapter running at 10mbps or 100mbps, half-duplex, you may observe occasional dropped receive packets. There are no workarounds for this problem in this network configuration. The network must be updated to operate in full-duplex, and/or 1000mbps only.

### 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
```

### 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.

### Allocating Rx Buffers When Using Jumbo Frames

-----

Allocating Rx buffers when using Jumbo Frames on 2.6.x kernels may fail if the available memory is heavily fragmented. This issue may be seen with PCI-X adapters or with packet split disabled. This can be reduced or eliminated by changing the amount of available memory for receive buffer allocation, by

increasing `/proc/sys/vm/min_free_kbytes`.

#### 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 the following:

```
# 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 following line to the file `/etc/sysctl.conf`:

```
net.ipv4.conf.all.arp_filter = 1
```

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

#### Disable Rx Flow Control with ethtool

---

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

```
# ethtool -A <ethX> autoneg off rx off
```

#### Unplugging Network Cable While ethtool -p is Running

---

In kernel versions 2.5.50 and newer, unplugging the network cable while ethtool -p is running will cause the system to become unresponsive to keyboard commands, except for control-alt-delete. Restarting the system should resolve the issue.

#### Rx Page Allocation Errors

---

'Page allocation failure. order:0' errors may occur under stress with kernels 2.6.25 and newer. This is caused by the way the Linux kernel reports this stressed condition.

#### Network Throughput Degradation Observed with Onboard Video Versus Add-in Video Card on 82579LM Gigabit Network Connection When Used with Some Older Kernels

This issue can be worked around by specifying "pci=nommconf" in the kernel boot parameter or by using another kernel boot parameter "memmap=128M\$0x100000000", which marks 128 MB region at 4 GB as reserved so that the OS will not use these RAM pages.

This issue is fixed in kernel version 2.6.21, where the kernel dynamically detects the mmconfig size by looking at the number of buses that the mmconfig segment maps to.



This issue will not be seen on the 32-bit version of EL5. In that case, the kernel sees that RAM is located around the 256 MB window and avoids using the mmconfig space.

#### Activity LED Blinks Unexpectedly

-----  
If a system based on the 82577, 82578, or 82579 controller is connected to a hub, the Activity LED will blink for all network traffic present on the hub. Connecting the system to a switch or router will filter out most traffic not addressed to the local port.

#### Link may take longer than expected

-----  
With some Phy and switch combinations, link can take longer than expected. This can be an issue on Linux distributions that timeout when checking for link prior to acquiring a DHCP address; however there is usually a way to work around this (for example, set LINKDELAY in the interface configuration on RHEL).

#### Tx flow control is disabled by default on 82577 and 82578-based adapters

#### Possible performance degradation on certain 82566 and 82577 devices

-----  
Internal stress testing with jumbo frames shows the reliability on some 82566 and 82567 devices is improved in certain corner cases by disabling the Early Receive feature. Doing so can impact Tx performance. To reduce the impact, the packet buffer sizes and relevant flow control settings are modified accordingly.

#### Support

=====

For general information, go to the Intel support website at:  
<http://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](mailto:e1000-devel@lists.sf.net).

#### License

=====

This program is free software; you can redistribute it and/or modify it under the terms and conditions of the GNU General Public License, version 2, as published by the Free Software Foundation.

This program is distributed in the hope it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St - Fifth Floor, Boston, MA 02110-1301 USA.

The full GNU General Public License is included in this distribution in the file called "COPYING".

Copyright(c) 1999 - 2019 Intel Corporation.

#### Trademarks

=====

Intel is a trademark or registered trademark of Intel Corporation or its subsidiaries in the United States and/or other countries.

\* Other names and brands may be claimed as the property of others.