Intel igb Device Driver Version 5.8.5 Installation on RedHawk OS

Release Notes

November 2,2021

concurrent
REAL-TIME

1. Introduction:

本書は、RedHawk に移植済の Intel igb version 5.8.5 デバイスドライバ・リリースノートである。.

2. Requirements:

この rpm ファイルをインストールする OS は、RedHawk 32/64 bits を想定している。
igb ドライバーは、インテル® PRO/1000PCI-E*(82563 /6/7 , 82571 /2/3/4/7/8 ,または 82583)
ベースのギガビット・ネットワーク・アダプター と、すべての PCI および PCI-X サポートインテル®
ギガビット・ネットワーク・アダプター コントローラーをサポートする。

この版のオリジナルコードは、メーカーにより以下の製品をサポートしている。

インテル® 82580EB ギガビット・イーサネット・コントローラー
Intel® Ethernet Controller I350-BT2
Intel® Ethernet Controller I350-AM4
Intel® Ethernet Server Adapter I340-T4
Intel® Gigabit ET Dual Port Server Adapter
Intel® Ethernet Controller I210-IS
Intel® 82576EB Gigabit Ethernet Controller
Intel® 82576NS Gigabit Ethernet Controller
Intel® Gigabit ET2 Quad Port Server Adapter
Intel® Ethernet Server Adapter I210-T1
インテル® 82575EB ギガビット・イーサネット・コントローラー
Intel® Ethernet Server Adapter I350-T2
Intel® Ethernet Server Adapter I350-T4
Intel® Gigabit VT Quad Port Server Adapter
Intel® Gigabit ET Quad Port Server Adapter
Intel® Ethernet Server Adapter I350-F2
Intel® Ethernet Server Adapter I350-F4
Intel® Ethernet Controller I210-AT
Intel® Ethernet Controller I210-AS
Intel® Ethernet Controller I210-CS
Intel® Ethernet Server Adapter I340-F4
Intel® Gigabit EF Dual Port Server Adapter
Intel® Ethernet Controller I350-AM2
Intel® Ethernet Controller I210-IT
Intel® Ethernet Server Adapter I350-T2V2
Intel® Ethernet Server Adapter I350-T4V2

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、RedHawk6.0x では、
”/lib/modules/`uname -r`/updates/drivers/net/igb/” であり
それ以降の版では、
“/lib/modules`uname –r`/updates/drivers/net/ethernet/intel/igb/” である

以下に各版の 64bit 版のインストール例を示す。
6.5 以降では、initramfs への組み込みは行われない。

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

# rpm -ivh igb-5.8.5-1.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:igb-5.8.5-1            ################################# [100%]
original pci.ids saved in /usr/local/share/igb
Original driver saved in /usr/local/src/igb-5.8.5
Installing igb-5.8.5 drivers succeeded!
```

なお、README などの、ファイルは、/usr/share/doc/igb-5.8.5 ディレクトリに、オリジナルソースコードは/usr/local/src/igb-5.8.5/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 igb-5.8.5-1.RedHawk-7.2.x86_64.rpm
/lib/modules/4.1.15-rt17-RedHawk-7.2-debug/updates/drivers/net/ethernet/intel/igb/igb.ko.new
/lib/modules/4.1.15-rt17-RedHawk-7.2-kdump/updates/drivers/net/ethernet/intel/igb/igb.ko.new
/lib/modules/4.1.15-rt17-RedHawk-7.2-trace/updates/drivers/net/ethernet/intel/igb/igb.ko.new
/lib/modules/4.1.15-rt17-RedHawk-7.2/updates/drivers/net/ethernet/intel/igb/igb.ko.new
/usr/local/src/igb-5.8.5/igb-5.8.5.src.tar.gz
/usr/share/doc/igb-5.8.5
/usr/share/doc/igb-5.8.5/COPYING
/usr/share/doc/igb-5.8.5/README
/usr/share/doc/igb-5.8.5/file.list
/usr/share/doc/igb-5.8.5/pci.updates
/usr/share/man/man7/igb.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 igb
filename:       /lib/modules/4.1.15-rt17-RedHawk-7.2-trace/updates/drivers/net/ethernet/intel/igb/igb.ko
version:        5.8.5
license:        GPL
description:    Intel(R) Gigabit Ethernet Linux Driver
author:         Intel Corporation, <e1000-devel@lists.sourceforge.net>
srcversion:     1111A5FF2EDDBEE10837147
alias:          pci:v00008086d000010D6sv*sd*bc*sc*i*
alias:          pci:v00008086d000010A9sv*sd*bc*sc*i*
alias:          pci:v00008086d000010A7sv*sd*bc*sc*i*
alias:          pci:v00008086d000010E8sv*sd*bc*sc*i*
alias:          pci:v00008086d00001526sv*sd*bc*sc*i*
alias:          pci:v00008086d0000150Dsv*sd*bc*sc*i*
alias:          pci:v00008086d000010E7sv*sd*bc*sc*i*
alias:          pci:v00008086d000010E6sv*sd*bc*sc*i*
alias:          pci:v00008086d00001518sv*sd*bc*sc*i*
alias:          pci:v00008086d0000150Asv*sd*bc*sc*i*
alias:          pci:v00008086d000010C9sv*sd*bc*sc*i*
alias:          pci:v00008086d00000440sv*sd*bc*sc*i*
alias:          pci:v00008086d0000043Csv*sd*bc*sc*i*
alias:          pci:v00008086d0000043Asv*sd*bc*sc*i*
alias:          pci:v00008086d00000438sv*sd*bc*sc*i*
alias:          pci:v00008086d00001516sv*sd*bc*sc*i*
alias:          pci:v00008086d00001511sv*sd*bc*sc*i*
alias:          pci:v00008086d00001510sv*sd*bc*sc*i*
alias:          pci:v00008086d00001527sv*sd*bc*sc*i*
alias:          pci:v00008086d0000150Fsv*sd*bc*sc*i*
alias:          pci:v00008086d0000150Esv*sd*bc*sc*i*
alias:          pci:v00008086d00001524sv*sd*bc*sc*i*
alias:          pci:v00008086d00001523sv*sd*bc*sc*i*
alias:          pci:v00008086d00001522sv*sd*bc*sc*i*
alias:          pci:v00008086d00001521sv*sd*bc*sc*i*
alias:          pci:v00008086d00001539sv*sd*bc*sc*i*
alias:          pci:v00008086d0000157Csv*sd*bc*sc*i*
```

```
alias:          pci:v00008086d0000157Bsv*sd*bc*sc*i*
alias:          pci:v00008086d00001538sv*sd*bc*sc*i*
alias:          pci:v00008086d00001537sv*sd*bc*sc*i*
alias:          pci:v00008086d00001536sv*sd*bc*sc*i*
alias:          pci:v00008086d00001533sv*sd*bc*sc*i*
alias:          pci:v00008086d00001F45sv*sd*bc*sc*i*
alias:          pci:v00008086d00001F41sv*sd*bc*sc*i*
alias:          pci:v00008086d00001F40sv*sd*bc*sc*i*
depends:        hwmon
vermagic:       4.1.15-rt17-RedHawk-7.2-trace SMP preempt mod_unload
parm:           InterruptThrottleRate:Maximum interrupts per second, per vector, (max 100000), default 3=adaptive (array of int)
parm:           IntMode:Change Interrupt Mode (0=Legacy, 1=MSI, 2=MSI-X), default 2 (array of int)
parm:           Node:set the starting node to allocate memory on, default -1 (array of int)
parm:           LLIPort:Low Latency Interrupt TCP Port (0-65535), default 0=off (array of int)
parm:           LLIPush:Low Latency Interrupt on TCP Push flag (0,1), default 0=off (array of int)
parm:           LLISize:Low Latency Interrupt on Packet Size (0-1500), default 0=off (array of int)
parm:           RSS:Number of Receive-Side Scaling Descriptor Queues (0-8), default 1, 0=number of cpus (array of int)
parm:           VMDQ:Number of Virtual Machine Device Queues: 0-1 = disable, 2-8 enable, default 0 (array of int)
parm:           max_vfs:Number of Virtual Functions: 0 = disable, 1-7 enable, default 0 (array of int)
parm:           MDD:Malicious Driver Detection (0/1), default 1 = enabled. Only available when max_vfs is greater than 0 (array
of int)
parm:           QueuePairs:Enable Tx/Rx queue pairs for interrupt handling (0,1), default 1=on (array of int)
parm:           EEE:Enable/disable on parts that support the feature (array of int)
parm:           DMAC:Disable or set latency for DMA Coalescing ((0=off, 1000-10000(msec), 250, 500 (usec)) (array of int)
parm:           LRO:Large Receive Offload (0,1), default 0=off (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/igb REDHAWKFLAVOR=`cat /proc/ccur/flavor` modules
# make –C `pwd` SUBDIRS=`pwd`/drivers/net/ethernet/intel/igb 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 "hwmon igb" -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。

```
Intel(R) Gigabit Ethernet Network Driver - version 5.8.5
Copyright (c) 2007-2021 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 igb-5.8.5-1
Uninstalling igb-5.8.5 drivers succeeded!
```

6. README

```
igb Linux* Base Driver for Intel(R) Ethernet Network Connection

===============================================================================

July 29, 2021

===============================================================================
```

Contents
========

Important Notes
===============

Configuring SR-IOV for improved network security

In a virtualized environment, on Intel(R) Ethernet Network Adapters that support SR-IOV, the virtual function (VF) may be subject to malicious behavior. Software-generated layer two frames, like IEEE 802.3x (link flow control), IEEE 802.1Qbb (priority based flow-control), and others of this type, are not expected and can throttle traffic between the host and the virtual switch, reducing performance. To resolve this issue, and to ensure isolation from unintended traffic streams, configure all SR-IOV enabled ports for VLAN tagging from the administrative interface on the PF. This configuration allows unexpected, and potentially malicious, frames to be dropped.

Overview
========
This driver supports kernel versions 2.6.30 or newer. However, some features may require a newer kernel version. The associated Virtual Function (VF) driver for this driver is igbvf.

Driver information can be obtained using ethtool, lspci, and ip. 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.

The igb driver supports IEEE 1588 time stamping for kernels 2.6.30 and newer.

The igb driver supports 2.5 Gbps operating speed on 2500BASE-KX only for I354-based network connections.

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 manually build the driver
----
1. Move the base driver tar file to the directory of your choice.
   For example, use '/home/username/igb' or '/usr/local/src/igb'.

2. Untar/unzip the archive, where <x.x.x> is the version number for the
   driver tar file:

   # tar zxf igb-<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 igb-<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/igb/igb.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 igb
   # modprobe igb [parameter=port1_value,port2_value]

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

   # rmmod igb; modprobe igb

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 igb driver.
   For Red Hat distributions:
            # dracut --force

   For Ubuntu:
            # update-initramfs -u


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 igb-<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, elfutilsl-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 build igb driver with DCA

If your kernel supports DCA, the driver will build by default with DCA enabled.

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 igb [<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 igb InterruptThrottleRate=16000,16000

In this case, there are two network ports supported by igb 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/applnots/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
3=dynamic conservative
<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 igb in a latency sensitive environment, this parameter should be considered.
<min_ITR>-<max_ITR> = 100-100000
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 is NOT supported by 82542, 82543, or 82544-based adapters.

LLI (Low Latency Interrupts)

LLI allows for immediate generation of an interrupt upon processing receive packets that match certain criteria as set by the parameters described below. LLI parameters are not enabled when Legacy interrupts are used. You must be using MSI or MSI-X (see cat /proc/interrupts) to successfully use LLI.

LLIPort

Valid Range: 0-65535
LLI is configured with the LLIPort command-line parameter, which specifies which TCP port should generate Low Latency Interrupts.

For example, using LLIPort=80 would cause the board to generate an immediate interrupt upon receipt of any packet sent to TCP port 80 on the local machine.
WARNING: Enabling LLI can result in an excessive number of interrupts/second that may cause problems with the system and in some cases may cause a kernel panic.

LLIPush

Valid Range: 0-1
LLIPush can be set to be enabled or disabled (default). It is most effective in an environment with many small transactions.
NOTE: Enabling LLIPush may allow a denial of service attack.

LLISize

Valid Range: 0-1500
LLISize causes an immediate interrupt if the board receives a packet smaller than the specified size.

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.

RSS

Valid Range: 0-8
0 = Assign up to the lesser value of the number of CPUs or the number of queues
X = Assign X queues, where X is less than or equal to the maximum number of queues (8 queues).

The maximum number of queues allowed are:
- I350-based adapters: 8 queues
- 82575-based adapters: 4 queues
- 82576-based and newer adapters: 8 queues
- I210-based adapters: 4 queues
- I211-based adapters: 2 queues

This parameter is also affected by the VMDq parameter in that it will limit the queues more. For example, if you set an 82575 device to VMDQ Mode 2, you will only be able to set 3 RSS queues. See the following table.

Model Number	VMDQ Mode 0	1	2	3+
82575	4	4	3	1
82576	8	2	2	2
82580	8	1	1	1

VMDQ

Valid Range: 0-4 on 82575-based adapters; 0-8 for 82576/82580-based adapters
Supports enabling VMDq pools as this is needed to support SR-IOV.
0 = Disabled
1 = Sets the netdev as pool 0
2+ = Add additional queues but they currently are not used
This parameter is forced to 1 or more if the max_vfs module parameter is used. In addition, the number of queues available for RSS is limited if this is set to 1 or greater.

NOTE: When either SR-IOV mode or VMDq mode is enabled, hardware VLAN filtering and VLAN tag stripping/insertion will remain enabled.

max_vfs

This parameter adds support for SR-IOV. It causes the driver to spawn up to max_vfs worth of virtual functions.
Valid Range: 0-7

If the value is greater than 0 it will also force the VMDq parameter to be 1 or more.

The parameters for the driver are referenced by position. Thus, if you have a dual port adapter, or more than one adapter in your system, and want N virtual functions per port, you must specify a number for each port with each parameter separated by a comma. For example:

modprobe igb max_vfs=4

This will spawn 4 VFs on the first port.

modprobe igb max_vfs=2,4

This will spawn 2 VFs on the first port and 4 VFs on the second port.

NOTE: Caution must be used in loading the driver with these parameters. Depending on your system configuration, number of slots, etc., it is impossible to predict in all cases where the positions would be on the command line.

NOTE: Neither the device nor the driver control how VFs are mapped into config space. Bus layout will vary by operating system. On operating systems that support it, you can check sysfs to find the mapping.

NOTE: When either SR-IOV mode or VMDq mode is enabled, hardware VLAN filtering and VLAN tag stripping/insertion will remain enabled. Please remove the old VLAN filter before the new VLAN filter is added. For example:

ip link set eth0 vf 0 vlan 100 // set vlan 100 for VF 0
ip link set eth0 vf 0 vlan 0 // Delete vlan 100
ip link set eth0 vf 0 vlan 200 // set a new vlan 200 for VF 0

QueuePairs

Valid Range: 0-1
If set to 0, when MSI-X is enabled, the Tx and Rx will attempt to occupy separate vectors.
This option can be overridden to 1 if there are not sufficient interrupts available. This can occur if any combination of RSS, VMDQ, and max_vfs results in more than 4 queues being used.

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.

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]

DMAC

Valid Range: 0, 1, 250, 500, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000
This parameter enables or disables DMA Coalescing feature. Values are in microseconds and set the internal DMA Coalescing internal timer.
DMA (Direct Memory Access) allows the network device to move packet data directly to the system's memory, reducing CPU utilization. However, the frequency and random intervals at which packets arrive do not allow the system to enter a lower power state. DMA Coalescing allows the adapter to collect packets before it initiates a DMA event. This may increase network latency but also increases the chances that the system will enter a lower power state. Turning on DMA Coalescing may save energy with kernel 2.6.32 and newer. DMA Coalescing must be enabled across all active ports in order to save platform power.

MDD (Malicious Driver Detection)

Valid Range: 0-1
0 = Disabled
1 = Enabled
This parameter is only relevant for I350 devices operating in SR-IOV mode. When this parameter is set, the driver detects malicious VF driver and disables its Tx/Rx queues until a VF driver reset occurs.

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/

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.

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 igb.

For example, if you install the igb 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 igb
  alias eth1 igb
  options igb 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 ip command to increase the MTU size. For example, enter the following where <ethX> is the interface number:

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

This setting is not saved across reboots.

Add 'MTU=9000' to the following file to make the setting change permanent:
/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 9216. This corresponds to the maximum jumbo frame size of 9234 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.

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.

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. If your Linux distribution does not include ethtool, download and install it 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 igb driver must be loaded prior to shutting down or suspending the system.

NOTES:
- Wake on LAN is only supported on port A of multi-port devices.
- Wake On LAN is not supported for the Intel(R) Gigabit VT Quad Port Server Adapter.

Multiqueue

In this mode, a separate MSI-X vector is allocated for each queue and one for "other" interrupts such as link status change and errors. All interrupts are throttled via interrupt moderation. Interrupt moderation must be used to avoid interrupt storms while the driver is processing one interrupt. The moderation value should be at least as large as the expected time for the driver to process an interrupt. Multiqueue is off by default.

REQUIREMENTS: MSI-X support is required for Multiqueue. If MSI-X is not found, the system will fallback to MSI or to Legacy interrupts. This driver supports multiqueue in kernel versions 2.6.24 and newer. This driver supports receive multiqueue on all kernels that support MSI-X.

NOTES:
- Do not use MSI-X with the 2.6.19 or 2.6.20 kernels.
- On some kernels a reboot is required to switch between single queue mode and multiqueue mode or vice-versa.

LRO

Valid Range: 0(off), 1(on)
Large Receive Offload (LRO) is a technique for increasing inbound throughput of high-bandwidth network connections by reducing CPU overhead. It works by aggregating multiple incoming packets from a single stream into a larger buffer before they are passed higher up the networking stack, thus reducing the number of packets that have to be processed. LRO combines multiple Ethernet frames into a single receive in the stack, thereby potentially decreasing CPU utilization for receives.
NOTE: LRO requires 2.4.22 or later kernel version.
IGB_NO_LRO is a compile time flag. The user can enable it at compile time to add support for LRO from the driver. The flag is used by adding CFLAGS_EXTRA="-DIGB_NO_LRO" to the make file when it's being compiled.

make CFLAGS_EXTRA="-DIGB_NO_LRO" install
You can verify that the driver is using LRO by looking at these counters in ethtool:
- lro_aggregated - counts total packets that were combined
- lro_flushed - counts the number of packets flushed out of LRO
- lro_recycled - counts the number of buffers returned to the ring from recycling

NOTE: IPv6 and UDP are not supported by LRO.

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.

NOTE: PTP requires 3.0.0 or later kernel version that has PTP support enabled in the kernel and a user-space software daemon.

IGB_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="-DIGB_PTP" install

NOTE: The driver will fail to compile if your kernel does not support PTP.

You can verify that the driver is using PTP by looking at the system log to see whether a PHC was attempted to be registered or not. If you have a kernel and

version of ethtool with PTP support, you can check the PTP support in the driver by executing:

ethtool -T <ethX>

Configuring VLAN Tagging on SR-IOV Enabled Adapter Ports

To configure VLAN tagging for the ports on an SR-IOV enabled adapter, use the following command. The VLAN configuration should be done before the VF driver is loaded or the VM is booted. The VF is not aware of the VLAN tag being inserted on transmit and removed on received frames (sometimes called "port VLAN" mode).

ip link set dev <ethX> vf <id> vlan <vlan id>

For example, the following will configure PF eth0 and the first VF on VLAN 10:

ip link set dev eth0 vf 0 vlan 10

MAC and VLAN Anti-Spoofing Feature for VFs

When a malicious driver on a Virtual Function (VF) interface attempts to send a spoofed packet, it is dropped by the hardware and not transmitted.

An interrupt is sent to the PF driver notifying it of the spoof attempt. When a spoofed packet is detected, the PF driver will send the following message to the system log (displayed by the "dmesg" command):

  Spoof event(s) detected on VF(n), where n = the VF that attempted to do the spoofing

Setting MAC Address, VLAN, and Rate Limit Using IProute2 Tool

You can set a MAC address of a Virtual Function (VF), a default VLAN, and the rate limit using the IProute2 tool. Download the latest version of the IProute2 tool from Sourceforge if your version does not have all the features you require.

Known Issues/Troubleshooting

MAC address of Virtual Function changes unexpectedly

If a Virtual Function's MAC address is not assigned in the host, then the VF (virtual function) driver will use a random MAC address. This random MAC address may change each time the VF driver is reloaded. You can assign a static MAC address in the host machine. This static MAC address will survive a VF driver reload.

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.

Using the igb driver on 2.4 or older 2.6 based kernels

Due to limited support for PCI-Express in 2.4 kernels and older 2.6 kernels, the igb driver may run into interrupt related problems on some systems, such as no link or hang when bringing up the device.

We recommend the newer 2.6 based kernels, as these kernels correctly configure the PCI-Express configuration space of the adapter and all intervening bridges. If you are required to use a 2.4 kernel, use a 2.4 kernel newer than 2.4.30. For 2.6 kernels we recommend using the 2.6.21 kernel or newer.

Alternatively, on 2.6 kernels you may disable MSI support in the kernel by booting with the "pci=nomsi" option or permanently disable MSI support in your kernel by configuring your kernel with CONFIG_PCI_MSI unset.

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.

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.

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.

Do Not Use LRO When Routing Packets

Due to a known general compatibility issue with LRO and routing, do not use LRO
when routing packets.

Build Error with Asianux 3.0 - Redefinition of typedef 'irq_handler_t'

Some systems may experience build issues due to the redefinition of
irq_handler_t. To resolve this issue, build the driver (step 4 above) using the
command:

make CFLAGS_EXTRA=-DAX_RELEASE_CODE=1 install

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.

Unloading Physical Function (PF) Driver Causes System Reboots When VM is
Running and VF is Loaded on the VM.

Do not unload the PF driver (igb) while VFs are assigned to guests.

Host May Reboot after Removing PF when VF is Active in Guest

Using kernel versions earlier than 3.2, do not unload the PF driver with
active VFs. Doing this will cause your VFs to stop working until you reload
the PF driver and may cause a spontaneous reboot of your system.

Prior to unloading the PF driver, you must first ensure that all VFs are
no longer active. Do this by shutting down all VMs and unloading the VF driver.

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.

License
=======

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".

Trademarks
==========