

Linux Base Driver for Intel(R) Ethernet Network Connection

Intel Gigabit Linux driver. Copyright(c) 1999 - 2013 Intel Corporation.

Contents

- Identifying Your Adapter
- Command Line Parameters
- Speed and Duplex Configuration
- Additional Configurations
- Support

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/go/network/adapter/idguide.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://support.intel.com/support/go/network/adapter/home.htm>

Command Line Parameters

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/aplnots/ap450.htm>

AutoNeg

(Supported only on adapters with copper connections)

Valid Range: 0x01-0x0F, 0x20-0x2F

Default Value: 0x2F

This parameter is a bit-mask that specifies the speed and duplex settings advertised by the adapter. When this parameter is used, the Speed and Duplex parameters must not be specified.

NOTE:

Refer to the Speed and Duplex section of this readme for more information on the AutoNeg parameter.

Duplex

(Supported only on adapters with copper connections)

Valid Range: 0-2 (0=auto-negotiate, 1=half, 2=full)

Default Value: 0

This 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.

FlowControl

Valid Range: 0-3 (0=none, 1=Rx only, 2=Tx only, 3=Rx&Tx)

Default Value: Reads flow control settings from the EEPROM

This parameter controls the automatic generation(Tx) and response(Rx) to Ethernet PAUSE frames.

InterruptThrottleRate

(not supported on Intel(R) 82542, 82543 or 82544-based adapters)

Valid Range: 0,1,3,4,100-100000 (0=off, 1=dynamic, 3=dynamic conservative, 4=simplified balancing)

Default Value: 3

The driver can limit the amount of interrupts per second that the adapter will generate for incoming packets. It does this by writing a value to the adapter that is based on the maximum amount of interrupts that the adapter will generate per second.

Setting InterruptThrottleRate to a value greater or equal to 100 will program the adapter to send out a maximum of 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.

The default behaviour of the driver previously assumed a static InterruptThrottleRate value of 8000, providing a good fallback value for all traffic types, but lacking in small packet performance and latency. The hardware can handle many more small packets per second however, and for this reason an adaptive interrupt moderation algorithm was implemented.

Since 7.3.x, the driver has two adaptive modes (setting 1 or 3) in which it dynamically adjusts the InterruptThrottleRate value based on the traffic that it receives. After determining the type of incoming traffic in the last timeframe, it will adjust the InterruptThrottleRate to an appropriate value for that traffic.

The algorithm classifies the incoming traffic every interval into classes. Once the class is determined, the InterruptThrottleRate value is adjusted to suit that traffic type the best. There are three classes defined: "Bulk traffic", for large amounts of packets of normal size; "Low latency", for small amounts of traffic and/or a significant percentage of small packets; and "Lowest latency", for almost completely small packets or minimal traffic.

In dynamic conservative mode, the InterruptThrottleRate value is set to 4000 for traffic that falls in class "Bulk traffic". If traffic falls in the "Low latency" or "Lowest latency" class, the InterruptThrottleRate is increased stepwise to 20000. This default mode is suitable for most applications.

For situations where low latency is vital such as cluster or grid computing, the algorithm can reduce latency even more when InterruptThrottleRate is set to mode 1. In this mode, which operates the same as mode 3, the InterruptThrottleRate will be increased stepwise to 70000 for traffic in class "Lowest latency".

In simplified mode the interrupt rate is based on the ratio of TX and RX traffic. If the bytes per second rate is approximately equal, the interrupt rate will drop as low as 2000 interrupts per second. If the traffic is mostly transmit or mostly receive, the interrupt rate could be as high as 8000.

Setting InterruptThrottleRate to 0 turns off any interrupt moderation and may improve small packet latency, but is generally not suitable for bulk throughput traffic.

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(R) 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:

```
modprobe e1000 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

Valid Range:

- 48-256 for 82542 and 82543-based adapters
- 48-4096 for all other supported adapters

Default Value: 256

This value specifies the number of receive buffer descriptors allocated by the driver. Increasing this value allows the driver to buffer more incoming packets, at the expense of increased system memory utilization.

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. 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

Valid Range: 0-65535 (0=off)

Default Value: 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 0.

RxAbsIntDelay

(This parameter is supported only on 82540, 82545 and later adapters.)

Valid Range: 0-65535 (0=off)

Default Value: 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.

Speed

(This parameter is supported only on adapters with copper connections.)

Valid Settings: 0, 10, 100, 1000

Default Value: 0 (auto-negotiate at all supported speeds)

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 should also be set when Speed is set to either 10 or 100.

TxDescriptors

Valid Range:

- 48-256 for 82542 and 82543-based adapters

- 48-4096 for all other supported adapters

Default Value: 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.

NOTE:

Depending on the available system resources, the request for a higher number of transmit descriptors may be denied. In this case, use a lower number.

TxIntDelay

Valid Range: 0-65535 (0=off)

Default Value: 8

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

(This parameter is supported only on 82540, 82545 and later adapters.)

Valid Range: 0-65535 (0=off)

Default Value: 32

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.

XsumRX

(This parameter is NOT supported on the 82542-based adapter.)

Valid Range: 0-1

Default Value: 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.

Copybreak

Valid Range: 0-xxxxxxx (0=off)

Default Value: 256

Usage: modprobe e1000.ko copybreak=128

Driver copies all packets below or equaling this size to a fresh RX buffer before handing it up the stack.

This parameter is different than other parameters, in that 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/e1000/parameters/copybreak

SmartPowerDownEnable

Valid Range: 0-1

Default Value: 0 (disabled)

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

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 hexadecimal 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

Jumbo Frames

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
```

This setting is not saved across reboots. It can be made permanent if you add:

```
MTU=9000
```

to the file `/etc/sysconfig/network-scripts/ifcfg-eth<x>`. This example applies to the Red Hat distributions; other distributions may store this setting in a different location.

Notes:

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.

- The maximum MTU setting for Jumbo Frames is 16110. This value coincides with the maximum Jumbo Frames size of 16128.
- Using Jumbo frames at 10 or 100 Mbps is not supported and may result in poor performance or loss of link.
- Adapters based on the Intel(R) 82542 and 82573V/E controller do not support Jumbo Frames. These correspond to the following product names:

```
Intel(R) PRO/1000 Gigabit Server Adapter
Intel(R) PRO/1000 PM Network Connection
```

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 configured through the ethtool utility.

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 when shutting down or rebooting the system.

Support

For general information, go to the Intel support website at:

<http://support.intel.com>

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 the supported kernel with a supported adapter, email the specific information related to the issue to e1000-devel@lists.sf.net