Linux Driver for Intel(R) Ethernet Network Connection

Contents

Identifying Your Adapter

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

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

InterruptThrottleRate

Valid Range: 0,1,3,4,100-100000
Default Value: 3

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.

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.

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.

- 0: Off
  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: Dynamic mode
  This 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.
- 3: Dynamic Conservative mode (default)
  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.

- 4: Simplified Balancing mode
  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.
- 100-100000:
  Setting InterruptThrottleRate to a value greater or equal to 100 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)
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

Valid Range: 0-65535 (0=off)
Default Value: 8

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)
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

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. 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)
Default Value: 256

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

Default Value: 0 (disabled)

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

KumeranLockLoss

Valid Range: 0,1
Default Value: 1 (enabled)

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

IntMode

Valid Range: 0-2

Default Value: 0

Value	Interrupt Mode
0	Legacy
1	MSI
2	MSI-X

IntMode allows 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.

This command will show different values for each type of interrupt:

```
cat /proc/interrupts
```

CrcStripping

Valid Range: 0,1
Default Value: 1 (enabled)

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.

WriteProtectNVM

Valid Range: 0,1
Default Value: 1 (enabled)

If set to 1, configure the hardware to ignore all write/erase cycles to the GbE region in the ICHx NVM (in order to prevent accidental corruption of the NVM). This feature can be disabled by setting the parameter to 0 during initial driver load.

NOTE: The machine must be power cycled (full off/on) when enabling NVM writes via setting the parameter to zero. Once the NVM has been locked (via the parameter at 1 when the driver loads) it cannot be unlocked except via power cycle.

Debug

Valid Range: 0-16 (0=none,...,16=all)
Default Value: 0

This parameter adjusts the level of debug messages displayed in the system logs.

Additional Features and Configurations

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 <x> is the interface number:

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

Alternatively, you can use the ip command as follows:

```
ip link set mtu 9000 dev eth<x>
ip link set up dev eth<x>
```

This setting is not saved across reboots. The setting change can be made permanent by adding 'MTU=9000' to the file:

- For RHEL: /etc/sysconfig/network-scripts/ifcfg-eth<x>
- For SLES: /etc/sysconfig/network/<config_file>

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: 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

NOTE: Jumbo Frames cannot be configured on an 82579-based Network device if MACSec is enabled on the system.

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://www.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.

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.

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.

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 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

Support

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

https://www.intel.com/support/

or the Intel Wired Networking project hosted by Sourceforge at:

https://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.