

**NAME**

**gve** - Ethernet driver for Google Virtual NIC (gVNIC)

**SYNOPSIS**

To compile this driver into the kernel, place the following lines in your kernel configuration file:

```
device gve
```

Alternatively, to load the driver as a module at boot time, place the following line in loader.conf(5):

```
if_gve_load="YES"
```

**DESCRIPTION**

gVNIC is a virtual network interface designed specifically for Google Compute Engine (GCE). It is required to support per-VM Tier-1 networking performance, and for using certain VM shapes on GCE.

**gve** is the driver for gVNIC. It supports the following features:

- ⌘ RX checksum offload
- ⌘ TX checksum offload
- ⌘ TCP Segmentation Offload (TSO)
- ⌘ Large Receive Offload (LRO) in software
- ⌘ Jumbo frames
- ⌘ Receive Side Scaling (RSS)

For more information on configuring this device, see ifconfig(8).

**HARDWARE**

**gve** binds to a single PCI device ID presented by gVNIC:

- ⌘ 0x1AE0:0x0042

**DIAGNOSTICS**

The following messages are recorded during driver initialization:

**Enabled MSIX with %d vectors**

**Configured device resources**

**Successfully attached %s**

**Deconfigured device resources**

These messages are seen if driver initialization fails. Global (across-queues) allocation failures:

**Failed to configure device resources: err=%d**  
**No compatible queue formats**  
**Failed to allocate ifnet struct**  
**Failed to allocate admin queue mem**  
**Failed to alloc DMA mem for DescribeDevice**  
**Failed to allocate QPL page**

irq and BAR allocation failures:

**Failed to acquire any msix vectors**  
**Tried to acquire %d msix vectors, got only %d**  
**Failed to setup irq %d for Tx queue %d**  
**Failed to setup irq %d for Rx queue %d**  
**Failed to allocate irq %d for mgmnt queue**  
**Failed to setup irq %d for mgmnt queue, err: %d**  
**Failed to allocate BAR0**  
**Failed to allocate BAR2**  
**Failed to allocate msix table**

Rx queue-specific allocation failures:

**No QPL left for rx ring %d**  
**Failed to alloc queue resources for rx ring %d**  
**Failed to alloc desc ring for rx ring %d**  
**Failed to alloc data ring for rx ring %d**

Tx queue-specific allocation failures:

**No QPL left for tx ring %d**  
**Failed to alloc queue resources for tx ring %d**  
**Failed to alloc desc ring for tx ring %d**  
**Failed to vmap fifo, qpl\_id = %d**

The following messages are recorded when the interface detach fails:

**Failed to deconfigure device resources: err=%d**

If bootverbose is on, the following messages are recorded when the interface is being brought up:

**Created %d rx queues**

**Created %d tx queues**

**MTU set to %d**

The following messages are recorded when the interface is being brought down:

**Destroyed %d rx queues**

**Destroyed %d tx queues**

These messages are seen if errors are encountered when bringing the interface up or down:

**Failed to destroy rxq %d, err: %d**

**Failed to destroy txq %d, err: %d**

**Failed to create rxq %d, err: %d**

**Failed to create txq %d, err: %d**

**Failed to set MTU to %d**

**Invalid new MTU setting. new mtu: %d max mtu: %d min mtu: %d**

**Cannot bring the iface up when detached**

**Reached max number of registered pages %lu > %lu**

**Failed to init lro for rx ring %d**

These messages are seen if any admin queue command fails:

**AQ command(%u): failed with status %d**

**AQ command(%u): unknown status code %d**

**AQ commands timed out, need to reset AQ**

**Unknown AQ command opcode %d**

These messages are recorded when the device is being reset due to an error:

**Scheduling reset task!**

**Waiting until admin queue is released.**

**Admin queue released**

If it was the NIC that requested the reset, this message is recorded:

**Device requested reset**

If the reset fails during the reinitialization phase, this message is recorded:

**Restore failed!**

These two messages correspond to the NIC alerting the driver to link state changes:

**Device link is up.**

**Device link is down.**

Apart from these messages, the driver exposes per-queue packet and error counters as sysctl nodes.

Global (across queues) counters can be read using netstat(8).

**LIMITATIONS**

**gve** does not support the transmission of VLAN-tagged packets. All VLAN-tagged traffic is dropped.

**SUPPORT**

Please email [gynic-drivers@google.com](mailto:gynic-drivers@google.com) with the specifics of the issue encountered.

**SEE ALSO**

ifconfig(8), netstat(8)

**HISTORY**

The **gve** device driver first appeared in FreeBSD 14.0.

**AUTHORS**

The **gve** driver was written by Google.