Contents

[Network APIs in Linux Kernel 1](#_Toc168318116)

[1. netdev\_priv: - 1](#_Toc168318117)

[2. netif\_carrier\_off() 1](#_Toc168318118)

# 

# Network APIs in Linux Kernel

## netdev\_priv: -

/\*\*

\* netdev\_priv - access network device private data

\* @dev: network device

\*

\* Get network device private data

\*/

static inline void \*[netdev\_priv](https://elixir.bootlin.com/linux/v4.3/C/ident/netdev_priv)(const struct [net\_device](https://elixir.bootlin.com/linux/v4.3/C/ident/net_device) \*dev)

{

return (char \*)dev + [ALIGN](https://elixir.bootlin.com/linux/v4.3/C/ident/ALIGN)(sizeof(struct [net\_device](https://elixir.bootlin.com/linux/v4.3/C/ident/net_device)), [NETDEV\_ALIGN](https://elixir.bootlin.com/linux/v4.3/C/ident/NETDEV_ALIGN));

}

## netif\_carrier\_off()

File: [/](https://elixir.bootlin.com/linux/v5.18.18/source)[net](https://elixir.bootlin.com/linux/v5.18.18/source/net)/[sched](https://elixir.bootlin.com/linux/v5.18.18/source/net/sched)/[sch\_generic.c](https://elixir.bootlin.com/linux/v5.18.18/source/net/sched/sch_generic.c)

Function prototype: void [netif\_carrier\_off](https://elixir.bootlin.com/linux/latest/C/ident/netif_carrier_off)(struct [net\_device](https://elixir.bootlin.com/linux/latest/C/ident/net_device) \*dev);

/\*\*

\* netif\_carrier\_off - clear carrier

\* @dev: network device

\*

\* Device has detected loss of carrier.

\*/

void [netif\_carrier\_off](https://elixir.bootlin.com/linux/latest/C/ident/netif_carrier_off)(struct [net\_device](https://elixir.bootlin.com/linux/latest/C/ident/net_device) \*dev)

{

if (![test\_and\_set\_bit](https://elixir.bootlin.com/linux/latest/C/ident/test_and_set_bit)([\_\_LINK\_STATE\_NOCARRIER](https://elixir.bootlin.com/linux/latest/C/ident/__LINK_STATE_NOCARRIER), &dev->state)) {

if (dev->[reg\_state](https://elixir.bootlin.com/linux/latest/C/ident/reg_state) == [NETREG\_UNINITIALIZED](https://elixir.bootlin.com/linux/latest/C/ident/NETREG_UNINITIALIZED))

return;

[atomic\_inc](https://elixir.bootlin.com/linux/latest/C/ident/atomic_inc)(&dev->[carrier\_down\_count](https://elixir.bootlin.com/linux/latest/C/ident/carrier_down_count));

[linkwatch\_fire\_event](https://elixir.bootlin.com/linux/latest/C/ident/linkwatch_fire_event)(dev);

}

}

=> Kernel drivers have access to two flags that map to IFF\_LOWER\_UP and IFF\_DORMANT. These flags can be set from everywhere, even from interrupts.

\_\_LINK\_STATE\_NOCARRIER, maps to !IFF\_LOWER\_UP:

The driver uses netif\_carrier\_on() to clear and netif\_carrier\_off() to set this flag. On netif\_carrier\_off(), the scheduler stops sending packets. The name 'carrier' and the inversion are historical, think of it as lower layer.