

# Interface statistics

## Overview

This document is a guide to Linux network interface statistics.

There are three main sources of interface statistics in Linux:

- standard interface statistics based on `c.type:'struct rtnl_link_stats64 <rtnl_link_stats64>';`

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\networking\[linux-master] [Documentation] [networking]statistics.rst, line 14); [backlink](#)  
Unknown interpreted text role "c.type".

- protocol-specific statistics; and
- driver-defined statistics available via ethtool.

## Standard interface statistics

There are multiple interfaces to reach the standard statistics. Most commonly used is the `ip` command from `iproute2`:

```
$ ip -s -s link show dev ens4ul1
6: ens4ul1: <BROADCAST,MULTICAST,UP,LOWER UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 48:2a:e3:4c:b1:d1 brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped overrun mcast
    74327665117 69016965 0        0        0        0
    RX errors: length  crc      frame  fifo  missed
                  0        0        0        0        0
    TX: bytes  packets  errors  dropped carrier collsns
    21405556176 44608960 0        0        0        0
    TX errors: aborted  fifo  window heartbeat transns
                  0        0        0        0        128
    altname enp58s0ul1
```

Note that `-s` has been specified twice to see all members of `c.type:'struct rtnl_link_stats64 <rtnl_link_stats64>'`. If `-s` is specified once the detailed errors won't be shown.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\networking\[linux-master] [Documentation] [networking]statistics.rst, line 38); [backlink](#)  
Unknown interpreted text role "c.type".

`ip` supports JSON formatting via the `-j` option.

## Protocol-specific statistics

Protocol-specific statistics are exposed via relevant interfaces, the same interfaces as are used to configure them.

### ethtool

Ethtool exposes common low-level statistics. All the standard statistics are expected to be maintained by the device, not the driver (as opposed to driver-defined stats described in the next section which mix software and hardware stats). For devices which contain unmanaged switches (e.g. legacy SR-IOV or multi-host NICs) the events counted may not pertain exclusively to the packets destined to the local host interface. In other words the events may be counted at the network port (MAC/PHY blocks) without separation for different host side (PCIe) devices. Such ambiguity must not be present when internal switch is managed by Linux (so called switchdev mode for NICs).

Standard ethtool statistics can be accessed via the interfaces used for configuration. For example ethtool interface used to configure pause frames can report corresponding hardware counters:

```
$ ethtool --include-statistics -a eth0
Pause parameters for eth0:
Autonegotiate:      on
RX:                  on
TX:                  on
Statistics:
  tx_pause_frames: 1
  rx_pause_frames: 1
```

General Ethernet statistics not associated with any particular functionality are exposed via `ethtool -S $ifc` by specifying the `--groups` parameter:

```
$ ethtool -S eth0 --groups eth-phy eth-mac eth-ctrl rmon
Stats for eth0:
eth-phy-SymbolErrorDuringCarrier: 0
eth-mac-FramesTransmittedOK: 1
eth-mac-FrameTooLongErrors: 1
```

```
eth-ctrl-MACControlFramesTransmitted: 1
eth-ctrl-MACControlFramesReceived: 0
eth-ctrl-UnsupportedOpCodesReceived: 1
rmon-etherStatsUndersizePkts: 1
rmon-etherStatsJabbers: 0
rmon-rx-etherStatsPkts64Octets: 1
rmon-rx-etherStatsPkts65to127Octets: 0
rmon-rx-etherStatsPkts128to255Octets: 0
rmon-tx-etherStatsPkts64Octets: 2
rmon-tx-etherStatsPkts65to127Octets: 3
rmon-tx-etherStatsPkts128to255Octets: 0
```

## Driver-defined statistics

Driver-defined ethtool statistics can be dumped using *ethtool -S \$ifc*, e.g.:

```
$ ethtool -S ens4ulul
NIC statistics:
tx_single_collisions: 0
tx_multi_collisions: 0
```

## uAPIs

### procfs

The historical */proc/net/dev* text interface gives access to the list of interfaces as well as their statistics.

Note that even though this interface is using `:c:type:'struct rtnl_link_stats64 <rtnl_link_stats64>'` internally it combines some of the fields.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\networking\linux-master [Documentation] [networking] statistics.rst, line 119); [backlink](#)  
Unknown interpreted text role "c:type".

### sysfs

Each device directory in sysfs contains a *statistics* directory (e.g. */sys/class/net/lo/statistics/*) with files corresponding to members of `:c:type:'struct rtnl_link_stats64 <rtnl_link_stats64>'`.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\networking\linux-master [Documentation] [networking] statistics.rst, line 126); [backlink](#)  
Unknown interpreted text role "c:type".

This simple interface is convenient especially in constrained/embedded environments without access to tools. However, it's inefficient when reading multiple stats as it internally performs a full dump of `:c:type:'struct rtnl_link_stats64 <rtnl_link_stats64>'` and reports only the stat corresponding to the accessed file.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\networking\linux-master [Documentation] [networking] statistics.rst, line 130); [backlink](#)  
Unknown interpreted text role "c:type".

Sysfs files are documented in *Documentation/ABI/testing/sysfs-class-net-statistics*.

### netlink

*rtnetlink* (*NETLINK\_ROUTE*) is the preferred method of accessing `:c:type:'struct rtnl_link_stats64 <rtnl_link_stats64>'` stats.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\networking\linux-master [Documentation] [networking] statistics.rst, line 143); [backlink](#)  
Unknown interpreted text role "c:type".

Statistics are reported both in the responses to link information requests (*RTM\_GETLINK*) and statistic requests (*RTM\_GETSTATS*, when *IFLA\_STATS\_LINK\_64* bit is set in the *filter\_mask* of the request).

### ethtool

Ethtool IOCTL interface allows drivers to report implementation specific statistics. Historically it has also been used to report statistics for which other APIs did not exist, like per-device-queue statistics, or standard-based statistics (e.g. RFC 2863).

Statistics and their string identifiers are retrieved separately. Identifiers via *ETHTOOL\_GSTRINGS* with *string\_set* set to *ETH\_SS\_STATS*, and values via *ETHTOOL\_GSTATS*. User space should use *ETHTOOL\_GDRVINFO* to retrieve the number of

statistics (*n\_stats*).

## ethtool-netlink

Ethtool netlink is a replacement for the older IOCTL interface.

Protocol-related statistics can be requested in get commands by setting the *ETHTOOL\_FLAG\_STATS* flag in *ETHTOOL\_A\_HEADER\_FLAGS*. Currently statistics are supported in the following commands:

- *ETHTOOL\_MSG\_PAUSE\_GET*
- *ETHTOOL\_MSG\_FEC\_GET*

## debugfs

Some drivers expose extra statistics via *debugfs*.

## struct rtnl\_link\_stats64

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\networking\linux-master [Documentation] [networking] statistics.rst, line 183)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: include/uapi/linux/if_link.h
   :identifiers: rtnl_link_stats64
```

## Notes for driver authors

Drivers should report all statistics which have a matching member in `:c:type:'struct rtnl_link_stats64 <rtnl_link_stats64>'` exclusively via *ndo\_get\_stats64*. Reporting such standard stats via ethtool or debugfs will not be accepted.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\networking\linux-master [Documentation] [networking] statistics.rst, line 189); [backlink](#)

Unknown interpreted text role "c:type".

Drivers must ensure best possible compliance with `:c:type:'struct rtnl_link_stats64 <rtnl_link_stats64>'`. Please note for example that detailed error statistics must be added into the general *rx\_error* / *tx\_error* counters.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\networking\linux-master [Documentation] [networking] statistics.rst, line 194); [backlink](#)

Unknown interpreted text role "c:type".

The *ndo\_get\_stats64* callback can not sleep because of accesses via */proc/net/dev*. If driver may sleep when retrieving the statistics from the device it should do so periodically asynchronously and only return a recent copy from *ndo\_get\_stats64*. Ethtool interrupt coalescing interface allows setting the frequency of refreshing statistics, if needed.

Retrieving ethtool statistics is a multi-syscall process, drivers are advised to keep the number of statistics constant to avoid race conditions with user space trying to read them.

Statistics must persist across routine operations like bringing the interface down and up.

## Kernel-internal data structures

The following structures are internal to the kernel, their members are translated to netlink attributes when dumped. Drivers must not overwrite the statistics they don't report with 0.

- *ethtool\_pause\_stats()*
- *ethtool\_fec\_stats()*