
*Full credit is given to the above companies including the OS that this PDF file was generated!*

## *Red Hat Enterprise Linux Release 9.2 Manual Pages on 'tc-ets.8' command*

***$ man tc-ets.8***

TC(8) Linux TC(8)

NAME

ETS - Enhanced Transmission Selection scheduler

SYNOPSIS

tc qdisc ... ets [ bands number ] [ strict number ] [ quanta bytes

bytes bytes... ] [ priomap band band band... ]

tc class ... ets [ quantum bytes ]

DESCRIPTION

The Enhanced Transmission Selection scheduler is a classful queuing discipline that merges functionality of PRIO and DRR qdiscs in one scheduler. ETS makes it easy to configure a set of strict and band? width-sharing bands to implement the transmission selection described in 802.1Qaz.

On creation with 'tc qdisc add', a fixed number of bands is created. Each band is a class, although it is not possible to directly add and remove bands with 'tc class' commands. The number of bands to be cre? ated must instead be specified on the command line as the qdisc is added.

The minor number of classid to use when referring to a band is the band number increased by one. Thus band 0 will have classid of major:1, band 1 that of major:2, etc.

ETS bands are of two types: some number may be in strict mode, the re? maining ones are in bandwidth-sharing mode.

ALGORITHM

When dequeuing, strict bands are tried first, if there are any. Band 0 is tried first. If it did not deliver a packet, band 1 is tried next, and so on until one of the bands delivers a packet, or the strict bands are exhausted.

If no packet has been dequeued from any of the strict bands, if there are any bandwidth-sharing bands, the dequeuing proceeds according to the DRR algorithm. Each bandwidth-sharing band is assigned a deficit counter, initialized to quantum assigned by a quanta element. ETS main? tains an (internal) "active" list of bandwidth-sharing bands whose qdiscs are non-empty. This list is used for dequeuing. A packet is de? queued from the band at the head of the list if the packet size is smaller or equal to the deficit counter. If the counter is too small, it is increased by quantum and the scheduler moves on to the next band in the active list.

Only qdiscs that own their queue should be added below the bandwidth-sharing bands. Attaching to them non-work-conserving qdiscs like TBF does not make sense -- other qdiscs in the active list will be skipped until the dequeue operation succeeds. This limitation does not exist with the strict bands.

CLASSIFICATION

The ETS qdisc allows three ways to decide which band to enqueue a packet to:

- Packet priority can be directly set to a class handle, in which case that
  is the queue where the packet will be put. For example, band number 2 of
  a qdisc with handle of 11: will have classid 11:3. To mark a packet for
  queuing to this band, the packet priority should be set to 0x110003.
- A tc filter attached to the qdisc can put the packet to a band by us? ing
  the flowid keyword.

- As a last resort, the ETS qdisc consults its priomap (see below), which
maps packets to bands based on packet priority.

PARAMETERS

strict The number of bands that should be created in strict mode. If not given, this value is 0.

quanta Each bandwidth-sharing band needs to know its quantum, which is the amount of bytes a band is allowed to dequeue before the scheduler moves to the next bandwidth-sharing band. The quanta argument lists quanta for the individual bandwidth-sharing bands. The minimum value of each quantum is 1. If quanta is not given, the default is no bandwidth-sharing bands, but note that when specifying a large number of bands, the extra ones are in bandwidth-sharing mode by default.

bands Number of bands given explicitly. This value has to be at least large enough to cover the strict bands specified through the strict keyword and bandwidth-sharing bands specified in quanta. If a larger value is given, any extra bands are in bandwidth-sharing mode, and their quanta are deduced from the interface MTU. If no value is given, as many bands are created as neces? sary to cover all bands implied by the strict and quanta key? words.

priomap

The priomap maps the priority of a packet to a band. The argu? ment is a list of numbers. The first number indicates which band the packets with priority 0 should be put to, the second is for priority 1, and so on.

There can be up to 16 numbers in the list. If there are fewer, the default band that traffic with one of the unmentioned prior? ities goes to is the last one.

EXAMPLE & USAGE

Add a qdisc with 8 bandwidth-sharing bands, using the interface MTU as their quanta. Since all quanta are the same, this will lead to equal

distribution of bandwidth between the bands, each will get about 12.5%
of the link. The low 8 priorities go to individual bands in a reverse
1:1 fashion (such that the highest priority goes to the first band).

```
# tc qdisc add dev eth0 root handle 1: ets bands 8 priomap 7 6 5 4 3 2
1 0
# tc qdisc show dev eth0
qdisc ets 1: root refcnt 2 bands 8 quanta 1514 1514 1514 1514 1514 1514
1514 1514 priomap 7 6 5 4 3 2 1 0 7 7 7 7 7 7 7 7
```

Tweak the first band of the above qdisc to give it a quantum of 2650,
which will give it about 20% of the link (and about 11.5% to the re?
maining bands):

```
# tc class change dev eth0 classid 1:1 ets quantum 2650
# tc qdisc show dev eth0
qdisc ets 1: root refcnt 2 bands 8 quanta 2650 1514 1514 1514 1514 1514
1514 1514 priomap 7 6 5 4 3 2 1 0 7 7 7 7 7 7 7 7
```

Create a purely strict Qdisc with reverse 1:1 mapping between priori?
ties and bands:

```
# tc qdisc add dev eth0 root handle 1: ets strict 8 priomap 7 6 5 4 3 2
1 0
# tc qdisc sh dev eth0
qdisc ets 1: root refcnt 2 bands 8 strict 8 priomap 7 6 5 4 3 2 1 0 7 7
7 7 7 7 7 7
```

Add a Qdisc with 6 bands, 3 strict and 3 ETS with 35%-30%-25% weights:

```
# tc qdisc add dev eth0 root handle 1: ets strict 3 quanta 3500 3000
2500 priomap 0 1 1 1 2 3 4 5
# tc qdisc sh dev eth0
qdisc ets 1: root refcnt 2 bands 6 strict 3 quanta 3500 3000 2500 pri?
omap 0 1 1 1 2 3 4 5 5 5 5 5 5 5 5 5
```

Create a Qdisc such that traffic with priorities 2, 3 and 4 are
strictly prioritized over other traffic, and the rest goes into band?
width-sharing classes with equal weights:

```
# tc qdisc add dev eth0 root handle 1: ets bands 8 strict 3 priomap 3 4
0 1 2 5 6 7
```

# tc qdisc sh dev eth0

qdisc ets 1: root refcnt 2 bands 8 strict 3 quanta 1514 1514 1514 1514 1514 priomap 3 4 0 1 2 5 6 7 7 7 7 7 7 7 7 7

## SEE ALSO

tc(8), tc-prio(8), tc-drr(8)

## AUTHOR

Parts of both this manual page and the code itself are taken from PRIO and DRR qdiscs.

ETS qdisc itself was written by Petr Machata.