PRIO(8) Linux PRIO(8)

NAMEPRIO − Priority qdisc SYNOPSIS tc qdisc... dev dev (parent classid | root) [handle ma jor:] prio [bands bands] [pri- omap band,band,band...] [estimator interval timeconstant] DESCRIPTION The PRIO qdisc is a simple classful queueing discipline that contains an arbitrary number of

classes of differing priority. The classes are dequeued in numerical descending order of priority.
PRIO is a scheduler and never delays packets - it is a work-conserving qdisc, though the qdiscs
co ntained in the classes may not be.

Very useful for lowering latency when there is no need for slowing down traffic. ALGORITHM On creation with ’tc qdisc add’, a fixed number of bands is created. Each band is a class, although

is not possible to add classes with ’tc qdisc add’, the number of bands to be created must instead
be specified on the commandline attaching PRIO to its root.

When dequeueing, band 0 is tried first and only if it did not deliver a packet does PRIO try band

1, and so onwards. Maximum reliability packets should therefore go to band 0, minimum delay to

band 1 and the rest to band 2.

As the PRIO qdisc itself will have minor number 0, band 0 is actually major:1, band 1 is major:2,

etc. For major, substitute the major number assigned to the qdisc on ’tc qdisc add’ with the han- dle parameter. CLASSIFICATION Three methods are available to PRIO to determine in which band a packet will be enqueued.

From userspace A process with sufficient privileges can encode the destination class directly with SO PRI- ORITY, see tc(7). with a tc filter A tc filter attached to the root qdisc can point traffic directly to a class with the priomap Based on the packet priority, which in turn is derived from the Type of Service assigned
to the packet. Only the priomap is specific to this qdisc. QDISC PARAMETERS bands Number of bands. If changed from the default of 3, priomap must be updated as well. priomapThe priomap maps the priority of a packet to a class. The priority can either be set

directly from userspace, or be derived from the Type of Service of the packet.

Determines how packet priorities, as assigned by the kernel, map to bands. Mapping
occurs based on the TOS octet of the packet, which looks like this:

0 1 2 3 4 5 6 7
+---+---+---+---+---+---+---+---+

| | | | |PRECEDENCE | TOS |MBZ| iproute2 16 December 2001 1

PRIO(8) Linux PRIO(8) | | | |

+---+---+---+---+---+---+---+---+

The four TOS bits (the ’TOS field’) are defined as:

Binary Decimcal Meaning

1000 8 Minimize delay (md)
0100 4 Maximize throughput (mt)

0010 2 Maximize r eliability (mr)
0001 1 Minimize monetary cost (mmc)
0000 0 N ormal S ervi ce

As there is 1 bit to the right of these four bits, the actual value of the TOS field is double

the value of the TOS bits. Tcpdump -v -v shows you the value of the entire TOS field,
n ot just the four bits. It is the value you see in the first column of this table:

TOS Bits Means Linux Priority Band

0x0 0 Normal Service 0 Best Effort 1

0x2 1 Minimize Monetary Cost 1 Filler 2
0x4 2 Maximize Reliability 0 Best Effort 1

0x6 3 mmc+mr 0 Best Effort 1
0x8 4 Maximize Throughput 2 Bulk 2

0xa 5 mmc+mt 2 Bulk 2
0xc 6 mr+mt 2 Bulk 2
0xe 7 mmc+mr+mt 2 Bulk 2

0x10 8 Minimize Delay 6 Interactive 0
0x12 9 mmc+md 6 Interactive 0

0x14 10 mr+md 6 Interactive 0
0x16 11 mmc+mr+md 6 Interactive 0

0x18 12 mt+md 4 Int. Bulk 1
0x1a 13 mmc+mt+md 4 Int. Bulk 1

0x1c 14 mr+mt+md 4 Int. Bulk 1
0x1e 15 mmc+mr+mt+md 4 Int. Bulk 1

The second column contains the value of the relevant four TOS bits, followed by their

trans lated meaning. For example, 15 stands for a packet wanting Minimal Montetary
Cost, Maximum Reliability, Maximum Throughput AND Minimum Delay.

The fourth column lists the way the Linux kernel interprets the TOS bits, by showing to

which Priority they are mapped.
The last column shows the result of the default priomap. On the commandline, the

default priomap looks like this:

1, 2, 2, 2, 1, 2, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1

This means that priority 4, for example, gets mapped to band number 1. The priomap

also allows you to list higher priorities (> 7) which do not correspond to TOS mappings, but which are set by other means.
This table from RFC 1349 (read it for more details) explains how applications might very

well set their TOS bits: iproute2 16 December 2001 2

PRIO(8) Linux PRIO(8) TELNET 1000 (minimize delay)

FTP Control 1000 (minimize delay)

Data 0100 (maximize throughput) TFTP 1000 (minimize delay)
SMTPCommand phase 1000 (minimize delay)

DATA phase 0100 (maximize throughput) Domain Name Service UDP Query 1000 (minimize delay)

TCP Query 0000

Zone Transfer 0100 (maximize throughput) NNTP 0001 (minimize monetary cost)
ICMPErrors 0000
Requests 0000 (mostly)

Responses <same as request> (mostly) CLASSES PRIO classes cannot be configured further - they are automatically created when the PRIO qdisc
is attached. Each class however can contain yet a further qdisc. BUGSLarge amounts of traffic in the lower bands can cause starvation of higher bands. Can be pre-

ve nted by attaching a shaper (for example, tc-tbf(8) to these bands to make sure they cannot dominate the link. AUTHORS Alexey N. Kuznetsov, <kuznet@ms2.inr.ac.ru>, J Hadi Salim <hadi@cyberus.ca>. This manpage maintained by bert hubert <ahu@ds9a.nl>

iproute2 16 December 2001 3