



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

Rocky Enterprise Linux 9.2 Manual Pages on command 'tc-drr.8'

\$ man tc-drr.8

TC(8) Linux

TC(8)

NAME

drr - deficit round robin scheduler

SYNOPSIS

```
tc qdisc ... add drr [ quantum bytes ]
```

DESCRIPTION

The Deficit Round Robin Scheduler is a classful queuing discipline as a more flexible replacement for Stochastic Fairness Queuing.

Unlike SFQ, there are no built-in queues -- you need to add classes and then set up filters to classify packets accordingly. This can be useful e.g. for using RED qdiscs with different settings for particular traffic. There is no default class -- if a packet cannot be classified, it is dropped.

ALGORITHM

Each class is assigned a deficit counter, initialized to quantum.

DRR maintains an (internal) "active" list of classes whose qdiscs are non-empty. This list is used for dequeuing. A packet is dequeued from the class 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 class in the active list.

PARAMETERS

quantum

Amount of bytes a flow is allowed to dequeue before the scheduler moves to the next class. Defaults to the MTU of the interface. The minimum value is 1.

EXAMPLE & USAGE

To attach to device eth0, using the interface MTU as its quantum:

```
# tc qdisc add dev eth0 handle 1 root drr
```

Adding two classes:

```
# tc class add dev eth0 parent 1: classid 1:1 drr
# tc class add dev eth0 parent 1: classid 1:2 drr
```

You also need to add at least one filter to classify packets.

```
# tc filter add dev eth0 protocol .. classid 1:1
```

Like SFQ, DRR is only useful when it owns the queue -- it is a pure scheduler and does not delay packets. Attaching non-work-conserving qdiscs like tbf to it does not make sense -- other qdiscs in the active

list will also become inactive until the dequeue operation succeeds.

Embed DRR within another qdisc like HTB or HFSC to ensure it owns the queue.

You can mimic SFQ behavior by assigning packets to the attached classes using the flow filter:

```
tc qdisc add dev .. drr
```

```
for i in .. 1024;do
```

```
    tc class add dev .. classid $handle:$i
```

```
    tc qdisc add dev .. fifo limit 16
```

```
done
```

```
tc filter add .. protocol ip .. $handle flow hash keys
```

```
src,dst,proto,proto-src,proto-dst divisor 1024 perturb 10
```

SOURCE

- o M. Shreedhar and George Varghese "Efficient Fair Queuing using Deficit Round Robin", Proc. SIGCOMM 95.

NOTES

This implementation does not drop packets from the longest queue on overrun, as limits are handled by the individual child qdiscs.

SEE ALSO

[tc\(8\)](#), [tc-htb\(8\)](#), [tc-sfq\(8\)](#)

AUTHOR

`sched_drr` was written by Patrick McHardy.