

**NAME**

CoDel - Fair Queuing (FQ) with Controlled Delay (CoDel)

**SYNOPSIS**

```
tc qdisc ... fq_codel [ limit PACKETS ] [ flows NUMBER ] [ target TIME ] [ interval TIME ] [ quantum
BYTES ] [ ecn | noecn ]
```

**DESCRIPTION**

FQ\_Codel (Fair Queuing Controlled Delay) is queuing discipline that combines Fair Queuing with the CoDel AQM scheme. FQ\_Codel uses a stochastic model to classify incoming packets into different flows and is used to provide a fair share of the bandwidth to all the flows using the queue. Each such flow is managed by the CoDel queuing discipline. Reordering within a flow is avoided since Codel internally uses a FIFO queue.

**PARAMETERS****limit**

has the same semantics as **codell** and is the hard limit on the real queue size. When this limit is reached, incoming packets are dropped. Default is 10240 packets.

**flows**

is the number of flows into which the incoming packets are classified. Due to the stochastic nature of hashing, multiple flows may end up being hashed into the same slot. Newer flows have priority over older ones. This parameter can be set only at load time since memory has to be allocated for the hash table. Default value is 1024.

**target**

has the same semantics as **codell** and is the acceptable minimum standing/persistent queue delay. This minimum delay is identified by tracking the local minimum queue delay that packets experience. Default value is 5ms.

**interval**

has the same semantics as **codell** and is used to ensure that the measured minimum delay does not become too stale. The minimum delay must be experienced in the last epoch of length .B interval. It should be set on the order of the worst-case RTT through the bottleneck to give endpoints sufficient time to react. Default value is 100ms.

**quantum**

is the number of bytes used as 'deficit' in the fair queuing algorithm. Default is set to 1514 bytes which corresponds to the Ethernet MTU plus the hardware header length of 14 bytes.

**ecn | noecn**

has the same semantics as **codell** and can be used to mark packets instead of dropping them. If **ecn** has been enabled, **noecn** can be used to turn it off and vice-a-versa. Unlike **codell**, **ecn** is turned on by default.

**EXAMPLES**

```
#tc qdisc add dev eth0 root fq_codel
#tc -s qdisc show
qdisc fq_codel 8002: dev eth0 root refcnt 2 limit 10240p flows 1024 quantum 1514 target 5.0ms interval
100.0ms ecn Sent 428514 bytes 2269 pkt (dropped 0, overlimits 0 requeues 0) backlog 0b 0p requeues 0
maxpacket 256 drop_overlimit 0 new_flow_count 0 ecn_mark 0 new_flows_len 0 old_flows_len 0

#tc qdisc add dev eth0 root fq_codel limit 2000 target 3ms interval 40ms noecn
#tc -s qdisc show
qdisc fq_codel 8003: dev eth0 root refcnt 2 limit 2000p flows 1024 quantum 1514 target 3.0ms interval
40.0ms Sent 2588985006 bytes 1783629 pkt (dropped 0, overlimits 0 requeues 34869) backlog 0b 0p
requeues 34869 maxpacket 65226 drop_overlimit 0 new_flow_count 73 ecn_mark 0 new_flows_len 1
```

old\_flows\_len 3

**SEE ALSO**

[tc\(8\)](#), [tc-codel\(8\)](#), [tc-red\(8\)](#)

**AUTHORS**

FQ\_CoDel was implemented by Eric Dumazet. This manpage was written by Vijay Subramanian. Please report corrections to the Linux Networking mailing list <netdev@vger.kernel.org>.