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

$ man tc-fq.8

FQ(8) Linux FQ(8)

NAME

FQ - Fair Queue traffic policing

SYNOPSIS

tc qdisc ... fq [limit PACKETS] [flow_limit PACKETS] [quantum BYTES] [initial_quantum BYTES] [maxrate RATE] [buckets NUMBER] [orphan_mask NUMBER] [pacing | nopacing] [ce_threshold TIME]

DESCRIPTION

FQ (Fair Queue) is a classless packet scheduler meant to be mostly used for locally generated traffic. It is designed to achieve per flow pac? ing. FQ does flow separation, and is able to respect pacing require? ments set by TCP stack. All packets belonging to a socket are consid? ered as a 'flow'. For non local packets (router workload), packet hash is used as fallback.

An application can specify a maximum pacing rate using the SO_MAX_PAC? ING_RATE setsockopt call. This packet scheduler adds delay between packets to respect rate limitation set on each socket. Note that after linux-4.20, linux adopted EDT (Earliest Departure Time) and TCP di? rectly sets the appropriate Departure Time for each skb.

Dequeueing happens in a round-robin fashion. A special FIFO queue is reserved for high priority packets (TC_PRIO_CONTROL priority), such packets are always dequeued first.

FQ is non-work-conserving.

TCP pacing is good for flows having idle times, as the congestion win? dow permits TCP stack to queue a possibly large number of packets. This removes the 'slow start after idle' choice, badly hitting large BDP flows and applications delivering chunks of data such as video streams.

PARAMETERS

limit

Hard limit on the real queue size. When this limit is reached, new packets are dropped. If the value is lowered, packets are dropped so that the new limit is met. Default is 10000 packets.

flow_limit

Hard limit on the maximum number of packets queued per flow. Default value is 100.

quantum

The credit per dequeue RR round, i.e. the amount of bytes a flow is al? lowed to dequeue at once. A larger value means a longer time period be? fore the next flow will be served. Default is 2 * interface MTU bytes.

initial_quantum

The initial sending rate credit, i.e. the amount of bytes a new flow is allowed to dequeue initially. This is specifically meant to allow us?ing IW10 without added delay. Default is 10 * interface MTU, i.e. 15140 for 'standard' ethernet.

maxrate

Maximum sending rate of a flow. Default is unlimited. Application specific setting via SO_MAX_PACING_RATE is ignored only if it is larger than this value.

buckets

The size of the hash table used for flow lookups. Each bucket is as?signed a red-black tree for efficient collision sorting. Default: 1024.

orphan_mask

For packets not owned by a socket, fq is able to mask a part of skb->hash and reduce number of buckets associated with the traffic. This is a DDOS prevention mechanism, and the default is 1023 (meaning no more than 1024 flows are allocated for these packets)

[no]pacing

Enable or disable flow pacing. Default is enabled.

ce_threshold

sets a threshold above which all packets are marked with ECN Congestion Experienced. This is useful for DCTCP-style congestion control algo?rithms that require marking at very shallow queueing thresholds.

EXAMPLES

#tc qdisc add dev eth0 root fq ce_threshold 4ms

#tc -s -d qdisc show dev eth0

qdisc fq 8001: dev eth0 root refcnt 2 limit 10000p flow_limit 100p

buckets 1024 orphan_mask 1023 quantum 3028b initial_quantum 15140b

low_rate_threshold 550Kbit refill_delay 40.0ms ce_threshold 4.0ms

Sent 72149092 bytes 48062 pkt (dropped 2176, overlimits 0 requeues 0)

backlog 1937920b 1280p requeues 0

flows 34 (inactive 17 throttled 0)

gc 0 highprio 0 throttled 0 ce_mark 47622 flows_plimit 2176

SEE ALSO

tc(8), socket(7)

AUTHORS

FQ was written by Eric Dumazet.