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-red.8' command

$ man tc-red.8

RED(8) Linux RED(8)

NAME

red - Random Early Detection

SYNOPSIS

tc qdisc ... red limit bytes [min bytes] [max bytes] avpkt bytes [burst packets] [ecn] [harddrop] [nodrop] [bandwidth rate] [probability chance] [adaptive] [qevent early_drop block index] [qevent mark block index]

DESCRIPTION

Random Early Detection is a classless qdisc which manages its queue size smartly. Regular queues simply drop packets from the tail when they are full, which may not be the optimal behaviour. RED also per? forms tail drop, but does so in a more gradual way.

Once the queue hits a certain average length, packets enqueued have a configurable chance of being marked (which may mean dropped). This chance increases linearly up to a point called the max average queue length, although the queue might get bigger.

This has a host of benefits over simple taildrop, while not being pro? cessor intensive. It prevents synchronous retransmits after a burst in traffic, which cause further retransmits, etc.

The goal is to have a small queue size, which is good for interactivity while not disturbing TCP/IP traffic with too many sudden drops after a burst of traffic.

Depending on if ECN is configured, marking either means dropping or purely marking a packet as overlimit.

ALGORITHM

The average queue size is used for determining the marking probability. This is calculated using an Exponential Weighted Moving Average, which can be more or less sensitive to bursts.

When the average queue size is below min bytes, no packet will ever be marked. When it exceeds min, the probability of doing so climbs lin? early up to probability, until the average queue size hits max bytes. Because probability is normally not set to 100%, the queue size might conceivably rise above max bytes, so the limit parameter is provided to set a hard maximum for the size of the queue.

PARAMETERS

min Average queue size at which marking becomes a possibility. De? faults to max /3

max At this average queue size, the marking probability is maximal. Should be at least twice min to prevent synchronous retransmits, higher for low min. Default to limit /4

probability

Maximum probability for marking, specified as a floating point number from 0.0 to 1.0. Suggested values are 0.01 or 0.02 (1 or 2%, respectively). Default : 0.02

limit Hard limit on the real (not average) queue size in bytes. Fur? ther packets are dropped. Should be set higher than max+burst. It is advised to set this a few times higher than max.

burst Used for determining how fast the average queue size is influ? enced by the real queue size. Larger values make the calculation more sluggish, allowing longer bursts of traffic before marking starts. Real life experiments support the following guideline: (min+min+max)/(3*avpkt).

avpkt Specified in bytes. Used with burst to determine the time con? stant for average queue size calculations. 1000 is a good value.

bandwidth

This rate is used for calculating the average queue size after some idle time. Should be set to the bandwidth of your inter? face. Does not mean that RED will shape for you! Optional. De? fault : 10Mbit

ecn As mentioned before, RED can either 'mark' or 'drop'. Explicit Congestion Notification allows RED to notify remote hosts that their rate exceeds the amount of bandwidth available. Non-ECN capable hosts can only be notified by dropping a packet. If this parameter is specified, packets which indicate that their hosts honor ECN will only be marked and not dropped, unless the queue size hits limit bytes. Recommended.

harddrop

If average flow queue size is above max bytes, this parameter forces a drop instead of ecn marking.

nodrop With this parameter, traffic that should be marked, but is not ECN-capable, is enqueued. Without the parameter it is early-dropped.

adaptive

(Added in linux-3.3) Sets RED in adaptive mode as described in http://icir.org/floyd/papers/adaptiveRed.pdf

Goal of Adaptive RED is to make 'probability' dynamic value between 1% and 50% to reach the target average queue :

(max - min) / 2

QEVENTS

See tc (8) for some general notes about qevents. The RED qdisc supports the following qevents:

early_drop

The associated block is executed when packets are early-dropped. This includes non-ECT packets in ECN mode.

mark The associated block is executed when packets are marked in ECN mode.

EXAMPLE

tc qdisc add dev eth0 parent 1:1 handle 10: red

limit 400000 min 30000 max 90000 avpkt 1000

burst 55 ecn adaptive bandwidth 10Mbit

SEE ALSO

tc(8), tc-choke(8)

SOURCES

- Floyd, S., and Jacobson, V., Random Early Detection gateways for Congestion Avoidance. http://www.aciri.org/floyd/pa? pers/red/red.html
- Some changes to the algorithm by Alexey N. Kuznetsov.
- Adaptive RED : http://icir.org/floyd/papers/adaptiveRed.pdf

AUTHORS

Alexey N. Kuznetsov, <kuznet@ms2.inr.ac.ru>, Alexey Makarenko <makar@phoenix.kharkov.ua>, J Hadi Salim <hadi@nortelnetworks.com>, Eric Dumazet <eric.dumazet@gmail.com>. This manpage maintained by bert hubert <ahu@ds9a.nl>

iproute2 13 December 2001 RED(8)