NAME

tbf − Token Bucket Filter

SYNOPSIS

tc qdisc ... tbf rate rate **burst** bytes/cell **(latency** ms | **limit** bytes **)** [**mpu** bytes [**peakrate** rate **mtu** bytes/cell]]

burst is also known as buffer and maxburst. mtu is also known as minburst.

DESCRIPTION

The Token Bucket Filter is a classless queueing discipline available for traffic control with the **tc**(8) command.

TBF is a pure shaper and never schedules traffic. It is non-work-conserving and may throttle itself, although packets are available, to ensure that the configured rate is not exceeded. On all platforms except for Alpha, it is able to shape up to 1mbit/s of normal traffic with ideal minimal burstiness, sending out data exactly at the configured rates.

Much higher rates are possible but at the cost of losing the minimal burstiness. In that case, data is on average dequeued at the configured rate but may be sent much faster at millisecond timescales. Because of further queues living in network adaptors, this is often not a problem.

Kernels with a higher 'HZ' can achieve higher rates with perfect burstiness. On Alpha, HZ is ten times higher, leading to a 10mbit/s limit to perfection. These calculations hold for packets of on average 1000 bytes.

ALGORITHM

As the name implies, traffic is filtered based on the expenditure of **tokens.** Tokens roughly correspond to bytes, with the additional constraint that each packet consumes some tokens, no matter how small it is. This reflects the fact that even a zero-sized packet occupies the link for some time.

On creation, the TBF is stocked with tokens which correspond to the amount of traffic that can be burst in one go. Tokens arrive at a steady rate, until the bucket is full.

If no tokens are available, packets are queued, up to a configured limit. The TBF now calculates the token deficit, and throttles until the first packet in the queue can be sent.

If it is not acceptable to burst out packets at maximum speed, a peakrate can be configured to limit the speed at which the bucket empties. This peakrate is implemented as a second TBF with a very small bucket, so that it doesn't burst.

To achieve perfection, the second bucket may contain only a single packet, which leads to the earlier mentioned 1mbit/s limit.

This limit is caused by the fact that the kernel can only throttle for at minimum 1 'jiffy', which depends on HZ as 1/HZ. For perfect shaping, only a single packet can get sent per jiffy - for HZ=100, this means 100 packets of on average 1000 bytes each, which roughly corresponds to 1mbit/s.

PARAMETERS

See **tc**(8) for how to specify the units of these values.

limit or latency

Limit is the number of bytes that can be queued waiting for tokens to become available. You can also specify this the other way around by setting the latency parameter, which specifies the maximum amount of time a packet can sit in the TBF. The latter calculation takes into account the size of the bucket, the rate and possibly the peakrate (if set). These two parameters are mutually exclusive.

burst
: Also known as buffer or maxburst. Size of the bucket, in bytes. This is the maximum amount of bytes that tokens can be available for instantaneously. In general, larger shaping rates require a larger buffer. For 10mbit/s on Intel, you need at least 10kbyte buffer if you want to reach your configured rate!

  If your buffer is too small, packets may be dropped because more tokens arrive per timer tick than fit in your bucket. The minimum buffer size can be calculated by dividing the rate by HZ.

  Token usage calculations are performed using a table which by default has a resolution of 8 packets. This resolution can be changed by specifying the **cell** size with the burst. For example, to specify a 6000 byte buffer with a 16 byte cell size, set a burst of 6000/16. You will probably never have to set this. Must be an integral power of 2.

mpu
: A zero-sized packet does not use zero bandwidth. For ethernet, no packet uses less than 64 bytes. The Minimum Packet Unit determines the minimal token usage (specified in bytes) for a packet. Defaults to zero.

rate
: The speed knob. See remarks above about limits! See **tc**(8) for units.

Furthermore, if a peakrate is desired, the following parameters are available:

peakrate
: Maximum depletion rate of the bucket. Limited to 1mbit/s on Intel, 10mbit/s on Alpha. The peakrate does not need to be set, it is only necessary if perfect millisecond timescale shaping is required.

mtu/minburst
: Specifies the size of the peakrate bucket. For perfect accuracy, should be set to the MTU of the interface. If a peakrate is needed, but some burstiness is acceptable, this size can be raised. A 3000 byte minburst allows around 3mbit/s of peakrate, given 1000 byte packets.

  Like the regular burstsize you can also specify a **cell** size.

EXAMPLE & USAGE

To attach a TBF with a sustained maximum rate of 0.5mbit/s, a peakrate of 1.0mbit/s, a 5kilobyte buffer, with a pre-bucket queue size limit calculated so the TBF causes at most 70ms of latency, with perfect peakrate behaviour, issue:

```
# tc qdisc add dev eth0 root tbf rate 0.5mbit \
  burst 5kb latency 70ms peakrate 1mbit       \
  minburst 1540
```

SEE ALSO

tc(8)

AUTHOR

Alexey N. Kuznetsov, <kuznet@ms2.inr.ac.ru>. This manpage maintained by bert hubert <ahu@ds9a.nl>