TC(8) Linux TC(8)

NAMEsfq − Stochastic Fairness Queueing SYNOPSIS tc qdisc... perturb seconds quantum bytes DESCRIPTION Stochastic Fairness Queueing is a classless queueing discipline available for traffic control with the

tc(8) command.

SFQ does not shape traffic but only schedules the transmission of packets, based on ’flows’. The
goal is to ensure fairness so that each flow is able to send data in turn, thus preventing any single
flow from drowning out the rest.

This may in fact have some effect in mitigating a Denial of Service attempt.

SFQ is work-conserving and therefore always delivers a packet if it has one available. ALGORITHM On enqueueing, each packet is assigned to a hash bucket, based on

(i) Source address
(ii) Destination address
(iii) Source port
If these are available. SFQ knows about ipv4 and ipv6 and also UDP, TCP and ESP. Packets
with other protocols are hashed based on the 32bits representation of their destination and the
socket they belong to. A flow corresponds mostly to a TCP/IP connection.

Each of these buckets should represent a unique flow. Because multiple flows may get hashed to
the same buc ket, the hashing algorithm is perturbed at configurable intervals so that the unfair-
ness lasts only for a short while. Perturbation may however cause some inadvertent packet
reor deri ng to occur.

When dequeuing, each hashbucket with data is queried in a round robin fashion.
The compile time maximum length of the SFQ is 128 packets, which can be spread over at most
128 buckets of 1024 available. In case of overflow, tail-drop is performed on the fullest bucket, thus

maintaining fairness. PARAMETERS perturb Interval in seconds for queue algorithm perturbation. Defaults to 0, which means that no

p er turbation occurs. Do not set too low for each perturbation may cause some packet

reordering. Advised value: 10 quantumAmount of bytes a flow is allowed to dequeue during a round of the round robin process.

Defaults to the MTU of the interface which is also the advised value and the minimum

value. EXAMPLE & USAGE To attach to device ppp0:

# tc qdisc add dev ppp0 root sfq perturb 10
Please note that SFQ, like all non-shaping (work-conserving) qdiscs, is only useful if it owns the

iproute2 8 December 2001 1

TC(8) Linux TC(8) queue. This is the case when the link speed equals the actually available bandwidth. This holds

for regular phone modems, ISDN connections and direct non-switched ethernet links.
Most often, cable modems and DSL devices do not fall into this category. The same holds for
when connected to a switch and trying to send data to a congested segment also connected to the
switch.
In this case, the effective queue does not reside within Linux and is therefore not available for
scheduling.

Embed SFQ in a classful qdisc to make sure it owns the queue. SOURCEo Paul E. McKenney "Stochastic Fairness Queuing ", IEEE INFOCOMM’90 Proceedings, San Francisco, 1990. o Paul E. McKenney "Stochastic Fairness Queuing ", "Interworking: Research and Experi- ence", v.2, 1991, p.113-131. o See also: M. Shreedhar and George Varghese "Efficient Fair Queuing using Deficit Round Robin", Proc. SIGCOMM 95. SEE ALSO tc(8) AUTHORAlexey N. Kuznetsov, <kuznet@ms2.inr.ac.ru>. This manpage maintained by bert hubert <ahu@ds9a.nl>

iproute2 8 December 2001 2