

# Fq\_codel

## fq\_codel

In 2012, Eric Dumazet contributed the `fq_codel` queueing discipline to the Linux kernel. It is thus available in recent Linux versions (3.5 and later). It combines CoDel with a variant (Deficit Round Robin) of stochastic flow-based fair queueing, and some logic to the effect that new flows get priority over old ones. It also enables [ECN](#) by default, has low computational and memory (64 bytes per flow) overhead, and can be used even on 10Gb/s links with contemporary hardware.

## How to Use

For a simple device *your\_device*, `fq_codel` can be activated simply by saying (as root):

```
tc qdisc add dev your_device root fq_codel
```

Unfortunately some devices are not that simple, e.g. Wifi interfaces have more than one queue, and `fq_codel` would have to be attached to each of them. Dave Täht has written a [debloat](#) script that should work for any type of interface. Hopefully, future releases of GNU/Linux distributions will be enhanced to set it up automatically by default for all devices on which it makes sense. (Source: [Message by D. Taht to the LWN mailing list](#), June 2012.)

## Contributed Packages for Debian/Ubuntu and Arch Linux

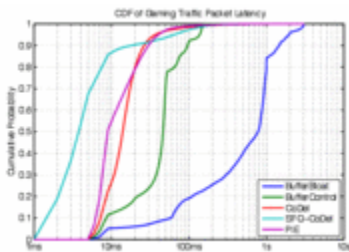
Toke Høiland-Jørgensen has provided packages of `iproute2`, `debloat`, `netperf` and `netperf-wrapper` with support for `fq_codel` for recent releases of Debian, Ubuntu and Arch GNU/Linux on <http://archive.tohjo.dk/>.

## Performance Evaluation Results

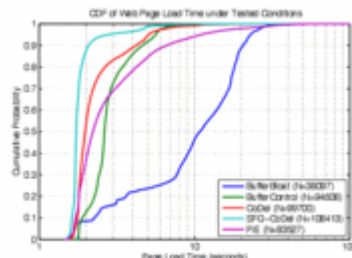
The following graphs illustrate the effect of `fq_codel` compared to conventional FIFO (tail-drop) queueing with large buffers, and also compared to other mechanisms such as PIE.

### Simulation Study for DOCSIS Cable Modems

These two CDF plots are from an IETF-86 presentation by Greg White (CableLabs) of NS-2 simulations of DOCSIS 3.0 cable modems with different queueing mechanisms under different simulated workloads.



CDF of Gaming Traffic Packet Latency,



CDF of Web Page

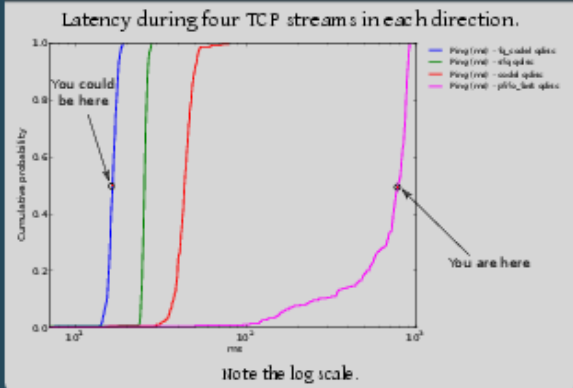
Load Time (courtesy Greg White)

## Measurements on Dedicated Lab Network

Toke Høiland-Jørgensen from the University of Roskilde performed comparative measurements of CoDel, `fq_codel`, traditional Linux SFQ and the default FIFO queueing on a dedicated network of workstations connected by Ethernet. Toke implemented a new "RRUL" (Realtime Response Under Load) test, which is implemented as a Python wrapper around the [netperf](#) benchmarking tool, and is distributed as part of the [netperf-wrapper](#) package. The results were also presented at IETF-86. The summary/spoiler slide again clearly shows the benefits of `fq_codel` over previous methods.

## Spoiler

### Effects of bufferbloat mitigation - RRUL test



(slides)

The data underlying this presentation is available on <http://akira.ruc.dk/~tohojo/bufferbloat/>.

## References

- [PATCH net-next] fq\_codel: Fair Queue Codel AQM, E. Dumazet, Message to the Linux netdev mailing list, May 2012
- Codel page on Bufferbloat Wiki
- The State of the Art in Bufferbloat Testing and Reduction on Linux, T. Høiland-Jørgensen, presentation to IRTF ICCRG at IETF-86, Orlando, FL, March 2013
- Simulation Study of AQM Performance in DOCSIS 3.0, G. White, presentation to IRTF ICCRG at IETF-86, Orlando, FL, March 2013
- draft-ietf-aqm-fq-codel-06, The FlowQueue-CoDel Packet Scheduler and Active Queue Management Algorithm, Toke Høiland-Jørgensen, Paul McKenney, Dave Täht, Jim Gettys, Eric Dumazet, March 2016

– Main.SimonLeinen - 2013-03-30 - 2016-03-20