

**NAME**

**cc\_cdg** - CDG Congestion Control Algorithm

**DESCRIPTION**

CAIA-Delay Gradient (CDG) is a hybrid congestion control algorithm which reacts to both packet loss and inferred queuing delay. It attempts to operate as a delay-based algorithm where possible, but utilises heuristics to detect loss-based TCP cross traffic and will compete effectively as required. CDG is therefore incrementally deployable and suitable for use on shared networks.

During delay-based operation, CDG uses a delay-gradient based probabilistic backoff mechanism, and will also try to infer non congestion related packet losses and avoid backing off when they occur.

During loss-based operation, CDG essentially reverts to `cc_newreno(4)`-like behaviour.

CDG switches to loss-based operation when it detects that a configurable number of consecutive delay-based backoffs have had no measurable effect. It periodically attempts to return to delay-based operation, but will keep switching back to loss-based operation as required.

**MIB Variables**

The algorithm exposes the following variables in the `net.inet.tcp.cc.cdg` branch of the `sysctl(3)` MIB:

<i>version</i>	Current algorithm/implementation version number.
<i>beta_delay</i>	Delay-based window decrease factor as a percentage (on delay-based backoff, $w = w * \text{beta\_delay} / 100$ ). Default is 70.
<i>beta_loss</i>	Loss-based window decrease factor as a percentage (on loss-based backoff, $w = w * \text{beta\_loss} / 100$ ). Default is 50.
<i>exp_backoff_scale</i>	Scaling parameter for the probabilistic exponential backoff. Default is 2.
<i>smoothing_factor</i>	Number of samples used for moving average smoothing (0 means no smoothing). Default is 8.
<i>loss_compete_consec_cong</i>	Number of consecutive delay-gradient based congestion episodes which will trigger loss-based CC compatibility. Default is 5.
<i>loss_compete_hold_backoff</i>	Number of consecutive delay-gradient based congestion episodes to hold the window backoff for loss-based CC compatibility. Default is 5.

*alpha\_inc* If non-zero, this enables an experimental mode where CDG's window increase factor (alpha) is increased by 1 MSS every *alpha\_inc* RTTs during congestion avoidance mode. (Setting *alpha\_inc* to 1 results in the most aggressive growth of the window increase factor over time. Use higher *alpha\_inc* values for slower growth.) Default is 0.

## SEE ALSO

`cc_chd(4)`, `cc_cubic(4)`, `cc_dctcp(4)`, `cc_hd(4)`, `cc_htcp(4)`, `cc_newreno(4)`, `cc_vegas(4)`, `h_ertt(4)`, `mod_cc(4)`, `tcp(4)`, `khhelp(9)`, `mod_cc(9)`

D. A. Hayes and G. Armitage, "Revisiting TCP Congestion Control using Delay Gradients", *Networking 2011 Proceedings, Part II*, 328-341, May 2011.

N. Khademi and G. Armitage, *Minimising RTT across homogeneous 802.11 WLANs with CAIA Delay-Gradient TCP (v0.1)*, CAIA Technical Report 121113A, <http://caia.swin.edu.au/reports/121113A/CAIA-TR-121113A.pdf>, November 2012.

## ACKNOWLEDGEMENTS

Development and testing of this software were made possible in part by grants from the FreeBSD Foundation and The Cisco University Research Program Fund, a corporate advised fund of Silicon Valley Community Foundation.

## HISTORY

The `cc_cdg` congestion control module first appeared in FreeBSD 9.2.

The module was first released in 2011 by David Hayes whilst working on the NewTCP research project at Swinburne University of Technology's Centre for Advanced Internet Architectures, Melbourne, Australia. More details are available at:

<http://caia.swin.edu.au/urp/newtcp/>

## AUTHORS

The `cc_cdg` congestion control module was written by David Hayes <[david.hayes@ieee.org](mailto:david.hayes@ieee.org)>. This manual page was written by Lawrence Stewart <[lstewart@FreeBSD.org](mailto:lstewart@FreeBSD.org)> and Grenville Armitage <[garmitage@swin.edu.au](mailto:garmitage@swin.edu.au)>.

## BUGS

The underlying algorithm and parameter values are still a work in progress and may not be optimal for some network scenarios.