NAME

cc_vegas - Vegas Congestion Control Algorithm

DESCRIPTION

The Vegas congestion control algorithm uses what the authors term the actual and expected transmission rates to determine whether there is congestion along the network path i.e.

> actual rate = (total data sent in a RTT) / RTT
>
> expected rate = cwnd / RTTmin
>
> diff = expected - actual

where RTT is the measured instantaneous round trip time and RTTmin is the smallest round trip time observed during the connection.

The algorithm aims to keep diff between two parameters alpha and beta, such that:

> alpha < diff < beta

If diff > beta, congestion is inferred and cwnd is decremented by one packet (or the maximum TCP segment size). If diff < alpha, then cwnd is incremented by one packet. Alpha and beta govern the amount of buffering along the path.

The implementation was done in a clean-room fashion, and is based on the paper referenced in the *SEE ALSO* section below.

IMPLEMENTATION NOTES

The time from the transmission of a marked packet until the receipt of an acknowledgement for that packet is measured once per RTT. This implementation does not implement Brakmo's and Peterson's original duplicate ACK policy since clock ticks in today's machines are not as coarse as they were (i.e. 500ms) when Vegas was originally designed. Note that modern TCP recovery processes such as fast retransmit and SACK are enabled by default in the TCP stack.

MIB Variables

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

alpha Query or set the Vegas alpha parameter as a number of buffers on the path. When setting alpha, the value must satisfy: 0 < alpha < beta. Default is 1.

beta Query or set the Vegas beta parameter as a number of buffers on the path. When setting beta, the value must satisfy: 0 < alpha < beta. Default is 3.

SEE ALSO

cc_cdg(4), cc_chd(4), cc_cubic(4), cc_dctcp(4), cc_hd(4), cc_htcp(4), cc_newreno(4), h_ertt(4), mod_cc(4), tcp(4), khelp(9), mod_cc(9)

L. S. Brakmo and L. L. Peterson, "TCP Vegas: end to end congestion avoidance on a global internet", *IEEE J. Sel. Areas Commun.*, 8, 13, 1465-1480, October 1995.

ACKNOWLEDGEMENTS

Development and testing of this software were made possible in part by grants from the FreeBSD Foundation and Cisco University Research Program Fund at Community Foundation Silicon Valley.

HISTORY

The **cc_vegas** congestion control module first appeared in FreeBSD 9.0.

The module was first released in 2010 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_vegas** congestion control module and this manual page were written by David Hayes <*david.hayes@ieee.org*>.