NAME

tcp_rack - TCP RACK-TLP Loss Detection Algorithm for TCP

SYNOPSIS

To load the TCP stack as a module at boot time, place the following line in loader.conf(5):

```
tcp_rack_load="YES"
```

To enable the TCP stack, place the following line in the sysctl.conf(5):

```
net.inet.tcp.functions_default=rack
```

DESCRIPTION

RACK-TLP uses per-segment transmit timestamps and selective acknowledgments (SACKs) and has two parts. Recent Acknowledgment (RACK) starts fast recovery quickly using time-based inferences derived from acknowledgment (ACK) feedback, and Tail Loss Probe (TLP) leverages RACK and sends a probe packet to trigger ACK feedback to avoid retransmission timeout (RTO) events.

Compared to the widely used duplicate acknowledgment (DupAck) threshold approach, RACK-TLP detects losses more efficiently when there are application-limited flights of data, lost retransmissions, or data packet reordering events.

It is intended to be an alternative to the DupAck threshold approach.

MIB Variables

The algorithm exposes the following scopes in the *net.inet.tcp.rack* branch of the sysctl(3) MIB:

net.inet.tcp.rack.misc
: Misc related controls

net.inet.tcp.rack.features
: Feature controls

net.inet.tcp.rack.measure
: Measure related controls

net.inet.tcp.rack.timers
: Timer related controls

net.inet.tcp.rack.tlp

TLP and Rack related Controls

net.inet.tcp.rack.timely
Rack Timely RTT Controls

net.inet.tcp.rack.hdwr_pacing
Pacing related Controls

net.inet.tcp.rack.pacing
Pacing related Controls

net.inet.tcp.rack.tp
Rack tracepoint facility

net.inet.tcp.rack.probertt
ProbeRTT related Controls

net.inet.tcp.rack.stats
Rack Counters

net.inet.tcp.rack.sack_attack
Rack Sack Attack Counters and Controls

Besides the variables within the above scopes the following variables are also exposed in the *net.inet.tcp.rack* branch:

net.inet.tcp.rack.clear
Clear counters

net.inet.tcp.rack.opts
RACK Option Stats

net.inet.tcp.rack.outsize
MSS send sizes

net.inet.tcp.rack.req_measure_cnt
If doing dynamic pacing, how many measurements must be in before we start pacing?

net.inet.tcp.rack.use_pacing
If set we use pacing, if clear we use only the original burst mitigation

net.inet.tcp.rack.rate_sample_method
What method should we use for rate sampling 0=high, 1=low

SEE ALSO

cc_chd(4), cc_cubic(4), cc_hd(4), cc_htcp(4), cc_newreno(4), cc_vegas(4), h_ertt(4), mod_cc(4), tcp(4), tcp_bbr(4), mod_cc(9)

Neal Cardwell, Yuchung Cheng, Nandita Dukkipati, and Priyaranjan Jha, *The RACK-TLP Loss Detection Algorithm for TCP*, February 2021, RFC 8985.

M. Allman, V. Paxson, and E. Blanton, *TCP Congestion Control*, September 2009, RFC 5681.

M. Mathis, Nandita Dukkipati, and Yuchung Cheng, *Proportional Rate Reduction for TCP*, May 2013, RFC 6937.

HISTORY

The **tcp_rack** congestion control module first appeared in FreeBSD 13.0.

AUTHORS

The **tcp_rack** congestion control module was written by Randall Stewart <*rrs@FreeBSD.org*> and sponsored by Netflix, Inc. This manual page was written by Gordon Bergling <*gbe@FreeBSD.org*>.