

FreeBSD 14? TCP ?? ?????

📄 : [scheffenegger.pdf](#)

FreeBSD 14 中, TCP 的 RACK 功能被引入, 旨在解决在慢速网络或高延迟网络中, 传统 TCP 的拥塞控制算法 (如 Reno, Cubic) 可能导致的性能下降问题。RACK 是 "RACK - Acknowledgement" 的缩写, 它通过更智能地管理重传和拥塞控制, 提高了网络吞吐量和稳定性。在 FreeBSD 14 之前, 系统使用的是 BSD4.4 的 TCP 实现, 而 2018 年引入的 RACK 功能 (在 FreeBSD 13.0 中首次亮相) 显著提升了网络性能, 特别是在高延迟或丢包率较高的环境中。RACK 通过更精细地控制重传和拥塞, 减少了不必要的重传, 从而提高了网络效率。此外, RACK 还引入了新的拥塞控制算法, 如 NewReno 和 Cubic, 以进一步优化网络性能。这些改进使得 FreeBSD 14 在网络性能方面有了显著提升, 特别是在处理高延迟或丢包率较高的网络环境时。RACK 的引入是 FreeBSD 网络栈的一个重要里程碑, 为未来的网络应用提供了更好的支持。Michael Tuexen 和 Randall Stewart 是 RACK 功能的主要开发者, 他们的贡献极大地提升了 FreeBSD 的网络性能。

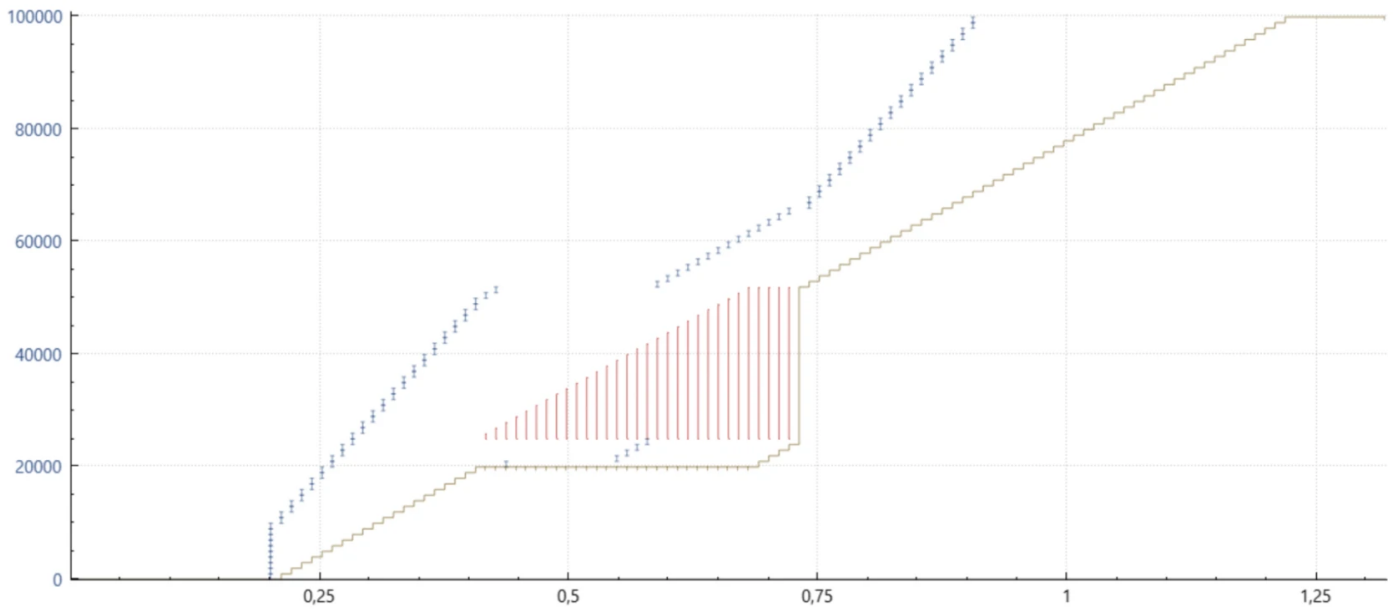
FreeBSD 14 的 RACK 功能在系统启动时会自动启用, 用户可以通过查看系统日志来确认 RACK 是否已成功加载。此外, 用户还可以通过配置系统参数来调整 RACK 的行为, 以满足特定的网络需求。RACK 的引入不仅提高了网络性能, 还增强了系统的稳定性和可靠性, 使其成为企业级网络环境的理想选择。

在 FreeBSD 13.0 中, 系统默认启用了 RACK 功能, 用户可以通过查看 `sys/netinet` 目录下的配置文件来了解 RACK 的默认配置。此外, 用户还可以通过修改配置文件来调整 RACK 的参数, 以优化网络性能。RACK 的引入是 FreeBSD 网络栈的一个重要里程碑, 为未来的网络应用提供了更好的支持。

Proportional Rate Reduction (?? ??? ??)

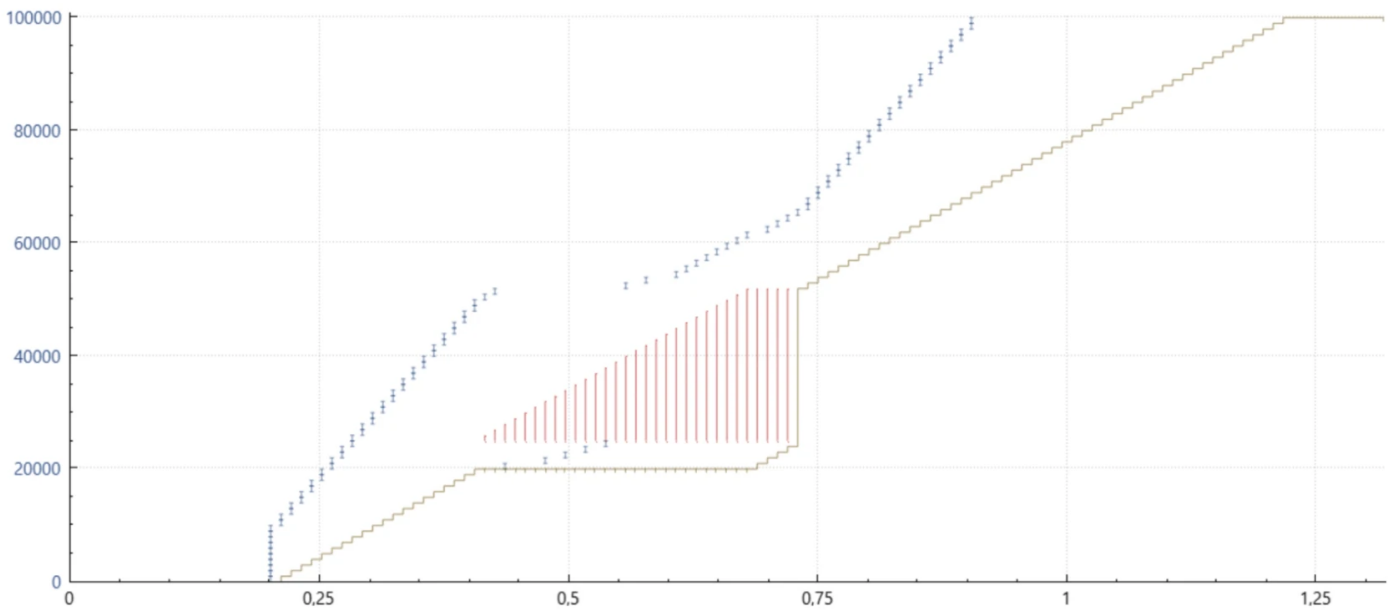
PRR - Proportional Rate Reduction (RFC6937) 是一种拥塞控制算法, 旨在通过按比例减少发送速率来缓解网络拥塞。PRR 通过动态调整发送速率, 确保网络在拥塞时能够保持稳定的性能。PRR 的引入是 FreeBSD 网络栈的一个重要里程碑, 为未来的网络应用提供了更好的支持。PRR 通过更智能地管理重传和拥塞, 提高了网络吞吐量和稳定性。在 FreeBSD 14 之前, 系统使用的是 BSD4.4 的 TCP 实现, 而 2018 年引入的 PRR 功能 (在 FreeBSD 13.0 中首次亮相) 显著提升了网络性能, 特别是在高延迟或丢包率较高的环境中。PRR 通过更精细地控制重传和拥塞, 减少了不必要的重传, 从而提高了网络效率。此外, PRR 还引入了新的拥塞控制算法, 如 NewReno 和 Cubic, 以进一步优化网络性能。这些改进使得 FreeBSD 14 在网络性能方面有了显著提升, 特别是在处理高延迟或丢包率较高的网络环境时。PRR 的引入是 FreeBSD 网络栈的一个重要里程碑, 为未来的网络应用提供了更好的支持。Michael Tuexen 和 Randall Stewart 是 PRR 功能的主要开发者, 他们的贡献极大地提升了 FreeBSD 的网络性能。

PRR 的引入不仅提高了网络性能, 还增强了系统的稳定性和可靠性, 使其成为企业级网络环境的理想选择。用户可以通过查看系统日志来确认 PRR 是否已成功加载。此外, 用户还可以通过配置系统参数来调整 PRR 的行为, 以满足特定的网络需求。PRR 的引入是 FreeBSD 网络栈的一个重要里程碑, 为未来的网络应用提供了更好的支持。



Cubic with SACK, but no PRR

Otrazuje działanie algorytmu Cubic z SACK, ale bez PRR. Wykres przedstawia zmiany w wielkości danych przesyłanych (niebieska linia) i odbieranych (czerwona linia) w czasie. Żółta linia pokazuje stan bufora odbiorcy. W tym przypadku, mimo SACK, brak PRR powoduje wystąpienie poważnego zastoju (widoczne jako gwałtowny wzrost i oscylacje czerwonej linii) w okolicach czasu 0,5-0,75, co skutkuje znaczącym spadkiem efektywności przesyłu.



Cubic with SACK (6675) and PRR

Otrazuje działanie algorytmu Cubic z SACK (6675) i PRR. Wykres przedstawia zmiany w wielkości danych przesyłanych (niebieska linia) i odbieranych (czerwona linia) w czasie. Żółta linia pokazuje stan bufora odbiorcy. Dzięki włączeniu PRR (kontrola przepływu nadmiarowego) udało się znacznie zredukować amplitudę zastoju (czerwona linia), co przyczyniło się do poprawy efektywności przesyłu.

Wykresy te porównują efektywność przesyłu w różnych konfiguracjach algorytmu Cubic. Widać, że PRR (kontrola przepływu nadmiarowego) jest kluczowym elementem, który pozwala uniknąć poważnych zastoju i utrzymać stabilną wydajność przesyłu.

ACK 0.7ms .

PRR () SACK , SACK ACK ECN PRR

SACK Handling

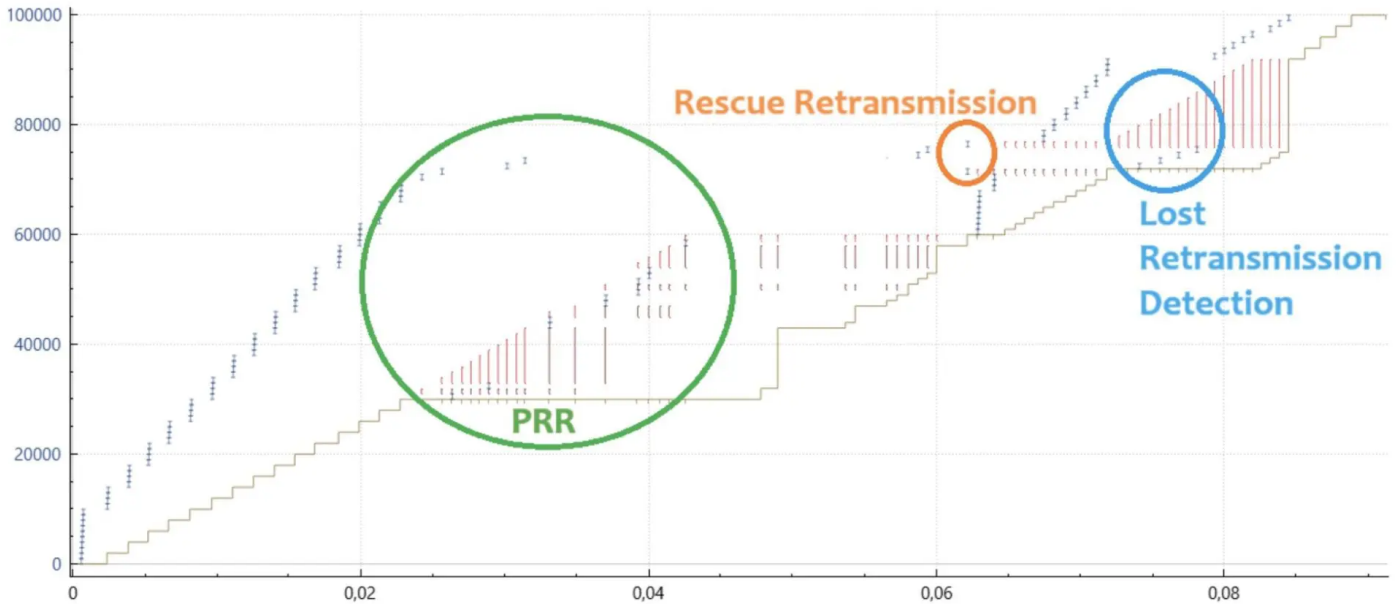
RFC6675 SACK .

RACK .

SACK .

RACK , TCP (: RPC) /IO FreeBSD 14 net.inet.tcp.do_lrd FreeBSD 15 net.inet.tcp.sack.lrd

IP



RACK (RACK) DSACK(RFC2883) DSACK
 . DSACK
 (Linux)
 (RTT)

Logging and Debugging

trpt FreeBSD 14 (dtrace, siftr, bblog, ...).

RACK (https://github.com/Netflix/tcplog_dumper https://github.com/Netflix/read_bblog)

Cubic

TCP FreeBSD

HyStart++ TCP (ECN) RTT RTT (CSS)

ECN (Explicit Congestion Notification) 是 TCP 协议的一个扩展，用于在网络拥塞时通知发送方。

Accurate Explicit Congestion Notification

ECN 是 TCP 协议的一个扩展，用于在网络拥塞时通知发送方。它通过在 TCP 头部的 ECN 字段中设置标志来实现。RFC 2914 定义了 ECN 的基本操作。在 10 年后的今天，IETF 正在制定更精确的 ECN 规范，以解决现有实现中的问题。这些规范包括对 ECN 标志的重新定义，以及对拥塞控制算法的改进。此外，还提到了对 FreeBSD 和 Linux 内核的支持，以及 DCTCP 和 AccECN 等高级拥塞控制算法。

ECN 通过减少 RTT 来提高网络效率。它允许发送方在收到拥塞通知后立即调整发送速率，而不是等待超时。RFC 3168 定义了 ECN 的部署策略。AccECN 是一种改进的 ECN 实现，旨在提高拥塞控制的精度。DCTCP 是一种基于 ECN 的拥塞控制算法，旨在减少队列延迟。TCP Prague 是另一种改进的 ECN 实现，旨在提高拥塞控制的鲁棒性。L4S 是一种端到端的拥塞控制方案，旨在进一步减少延迟。

Authentication and Security

RACK 是 TCP 协议的一个扩展，用于提高连接的安全性。它通过引入 MD5 校验和来防止中间人攻击。RFC 3746 定义了 RACK 的部署策略。BGP 是互联网路由协议，用于在自治系统之间交换路由信息。RACK 可以与 BGP 结合使用，以提高路由的安全性。

RFC 7323 (RFC 1323) 是 TCP 协议的一个扩展，用于提高连接的可靠性。它通过引入窗口缩放和选择性确认来实现。net.inet.tcp.rfc1323 是 FreeBSD 内核中的一个配置选项，用于启用 RFC 1323。窗口缩放允许接收方接收更大的数据块，从而提高吞吐量。选择性确认允许接收方接收乱序的数据包，从而提高连接的效率。TCP 窗口缩放和选择性确认是 TCP 协议的重要组成部分。net.inet.tcp.tolerate_missing_ts 是 FreeBSD 内核中的一个配置选项，用于容忍缺失的序列号。

What's Next?

TCP 协议的下一个版本将包括对 RACK 和 RACK 的改进，以及对窗口缩放和选择性确认的进一步支持。

