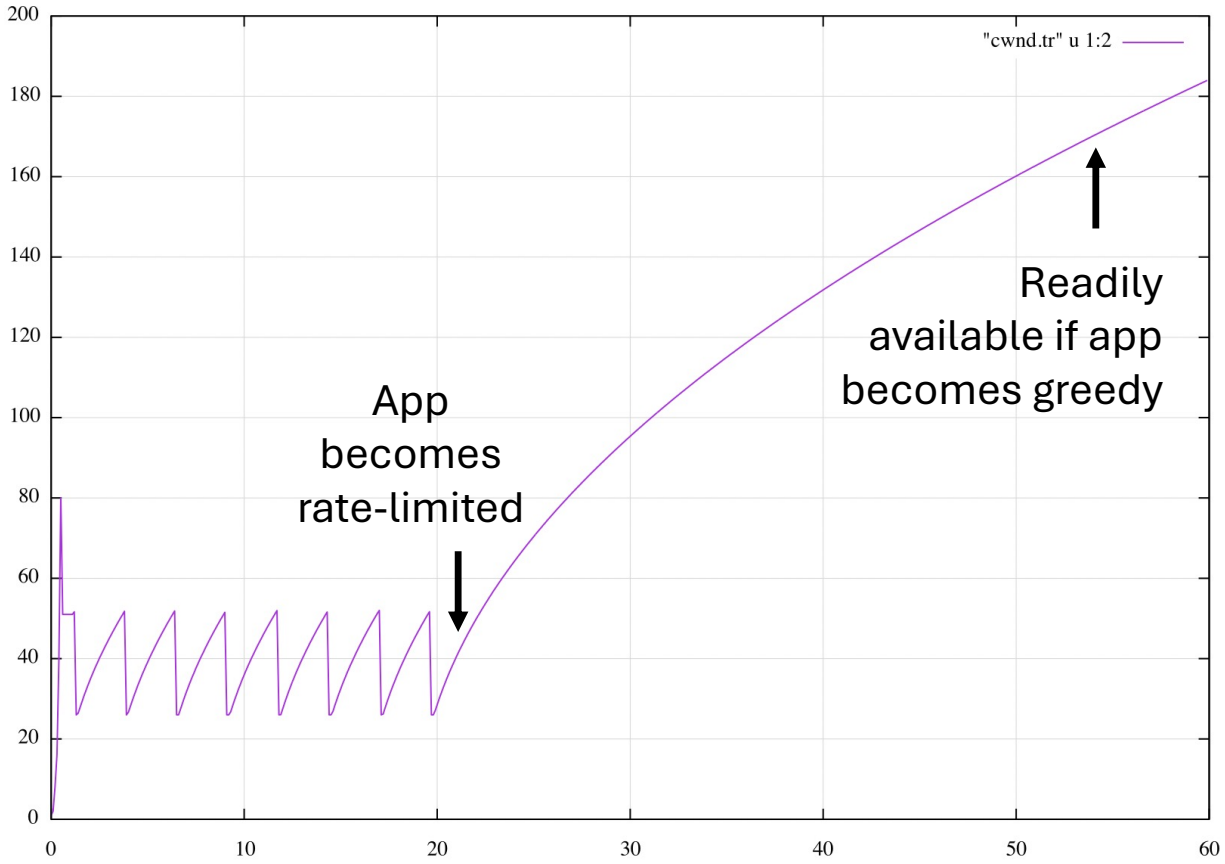
Increase of the Congestion Window When the Sender Is Rate-Limited

draft-welzl-ccwg-ratelimited-increase-02

Michael Welzl, Tom Henderson, Gorry Fairhurst

CCWG
IETF 120

Reminder of what this is



- **Current TCP specifications allow this**, when app is just not producing more data, or rwnd is limited!
- ns-3 & ns-2 try to follow the specs; and implement this
  - We hope no one else does…

=> Primary motivation:
fix this with a PS.

("clean up": RFCs a bit messy)

Specifying the increase (unchanged)

- Our proposed rules: senders (..)
  1. **MUST** include a limit to the growth of cwnd when FlightSize < cwnd.
  2. **SHOULD** limit the growth of cwnd when FlightSize < cwnd with inc(maxFS).

- maxFS: max. FlightSize so far, since last increase (or beginning)
- inc: the CC's increase if the sender were not rate-limited

- In slow start, Linux (kernel >= 3.16, 2024) implements our rule #2

What happened since -01?

- **New subsections on rate-based controls and pacing**
  (feedback from IETF-119)

- "The guiding principle behind the rules (..) applies to all congestion control algorithms: in the absence of a congestion indication, a sender should be allowed to increase its rate from the amount of data that it has transmitted during the previous RTT. This holds irrespective of whether the sender is rate-limited or not"

- Pacing typically operates on sub-RTT timescales, so nothing special needs to be done; our rules don't constrain it

What happened since -01? /2

- **Trimmed the list of RFCs to update** *(feedback from IETF-119)*

- TCP "Reno" congestion control: RFC 5681
- QUIC: RFC 9002
- SCTP: RFC 9260
- Cubic congestion control: RFC 9438

Thank you !