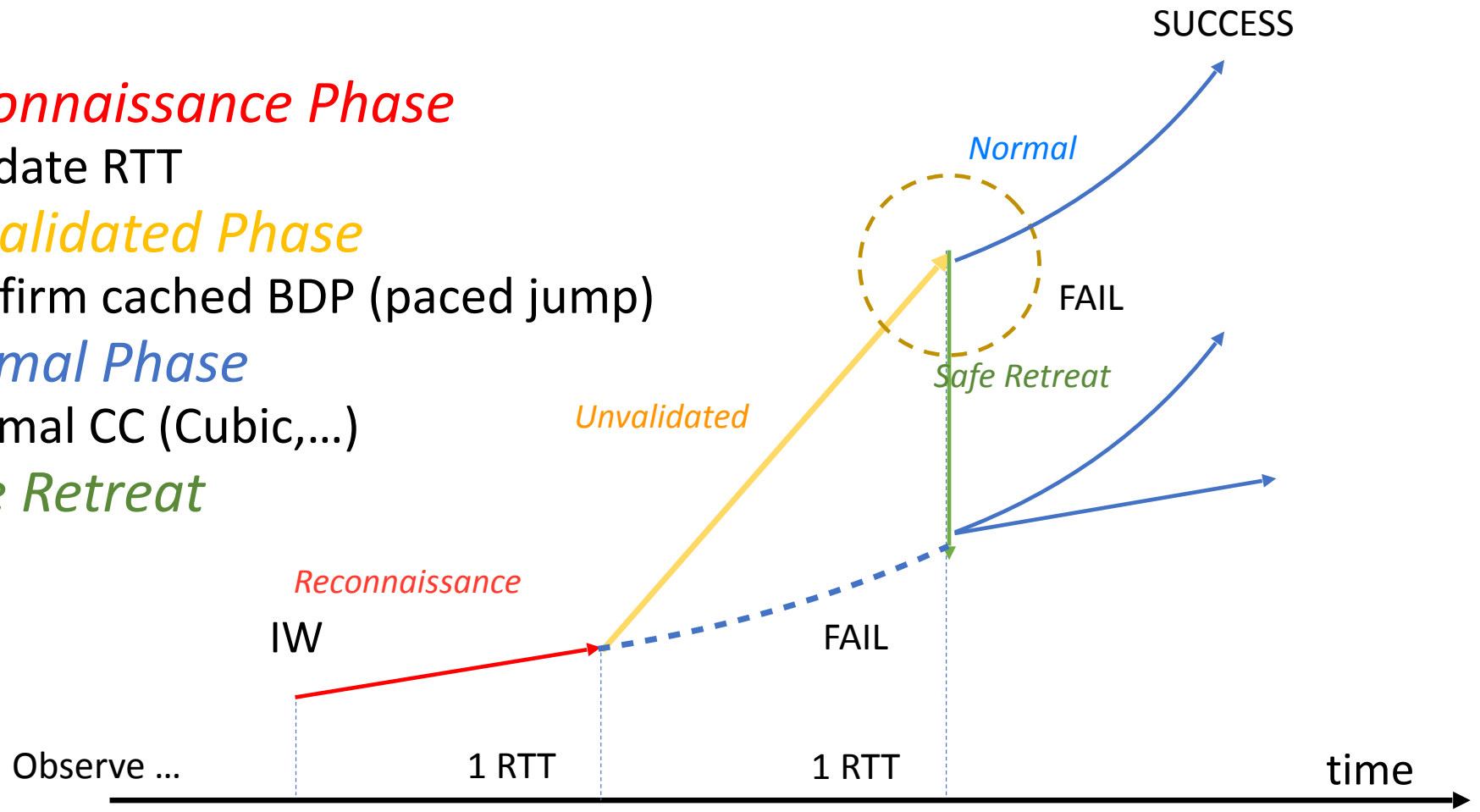
Careful Convergence of Congestion Control from Retained State

draft-ietf-tsvwg-careful-resume-01

Nicolas Kuhn , Stephan Emile , Gorry Fairhurst , Christian Huitema

Careful Resume, CC at startup

- *Reconnaissance Phase*
  Validate RTT
- *Unvalidated Phase*
  Confirm cached BDP (paced jump)
- *Normal Phase*
  Normal CC (Cubic,…)
- *Safe Retreat*



Update since Adoption

- Changed to PS
- Aligned terms
- Some initial text on flow control interactions …

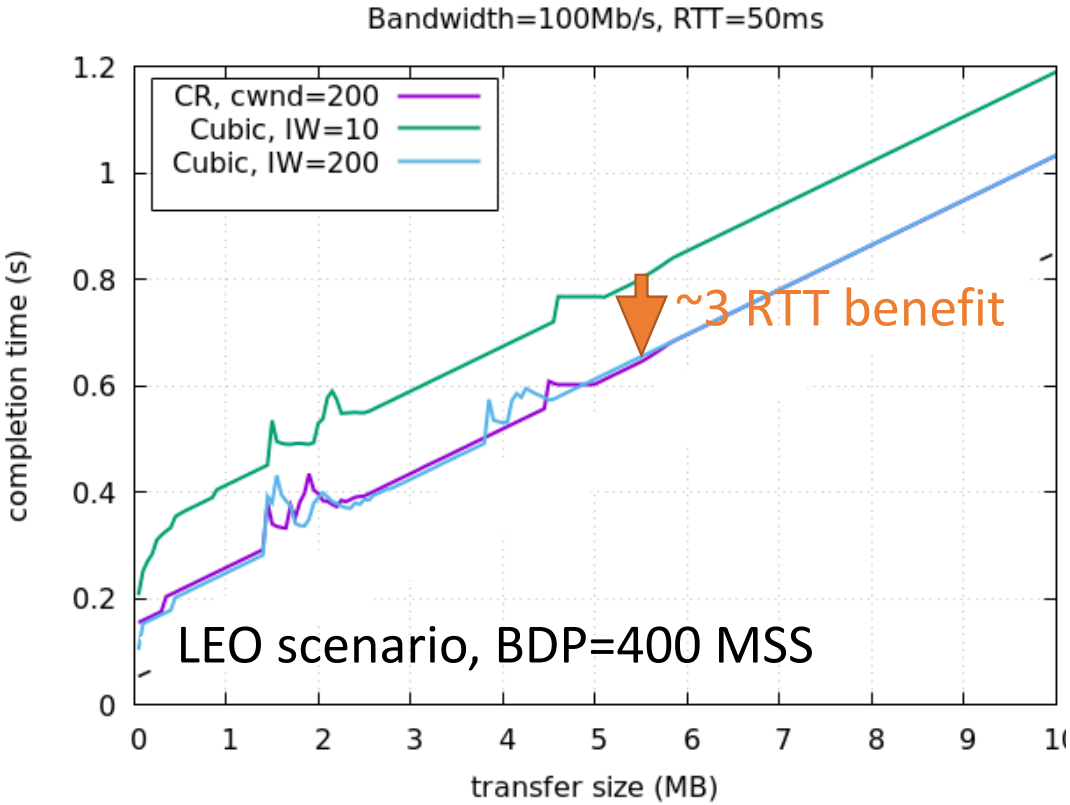
State of CR Implementation

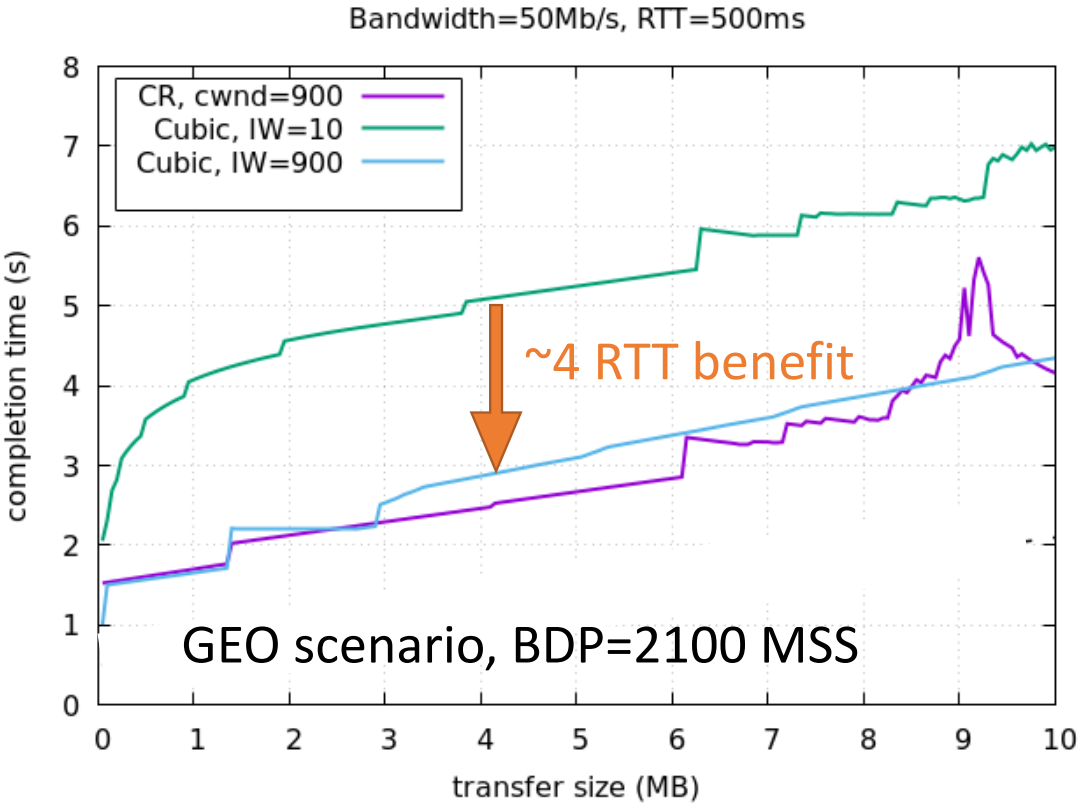
- **ns-3 model of Careful Resume for TCP**
  - Reconnaissance, Unvalidated, (Safe Retreat to be do evaluated)
  - TCP Cubic + Hystart++ modification
  - https://github.com/rsecchi/quicoptsat

- **Linux fork of Careful Resume**
  - Cubic Congestion Control
  - Uses FQ for pacing of unvalidated

- **Careful Resume for QUIC (next)**
  - BDP_Frame (to negotiate flow credit, etc.)

Simulated Performance over LEO/GEO satellite: IW=10, Jump=BDP/2, CR+Pacing, no competing traffic

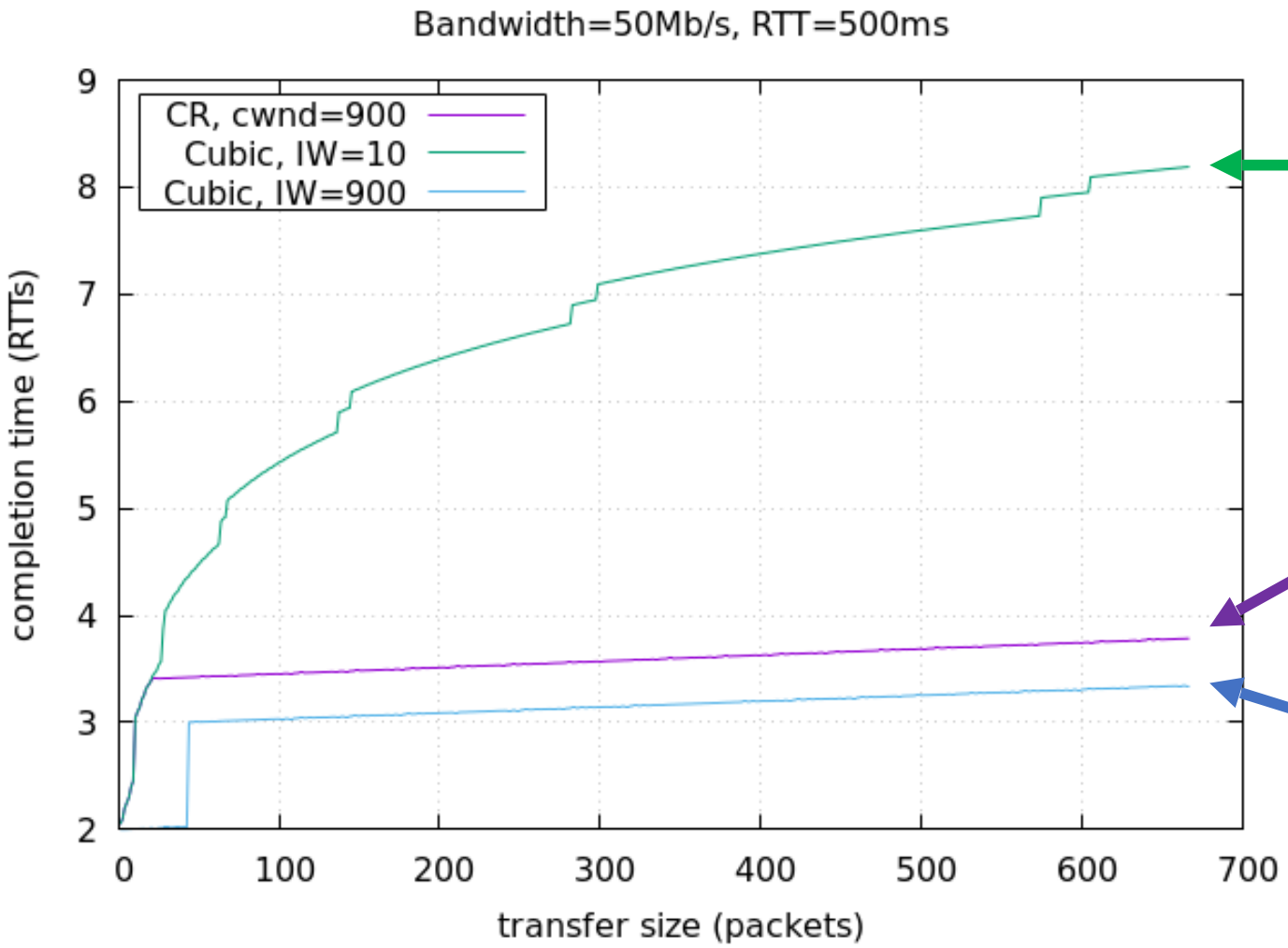




IETF 117, San Francisco, July 2023

BDP smaller for LEO case

Careful Resume, Gain in RTTs vs transfer size

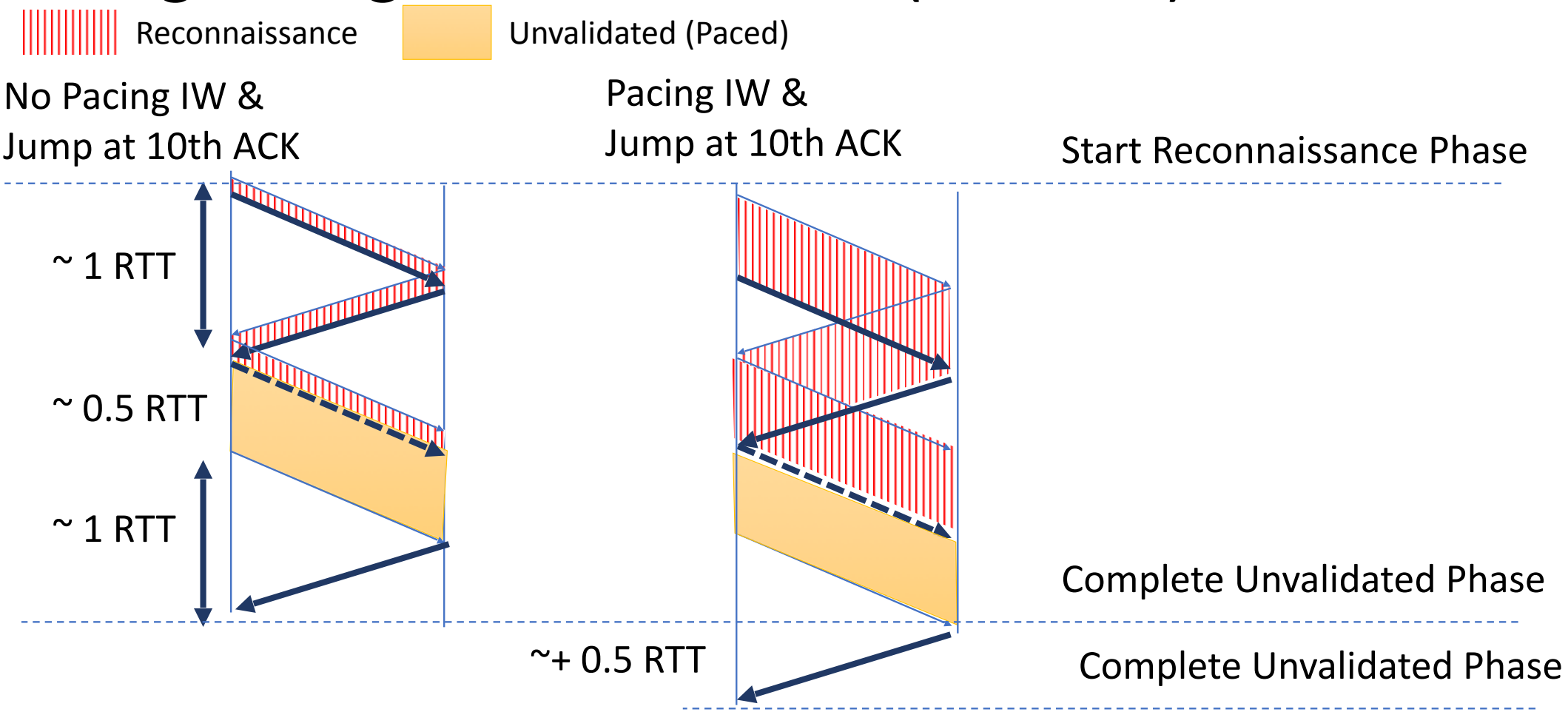


IW=10

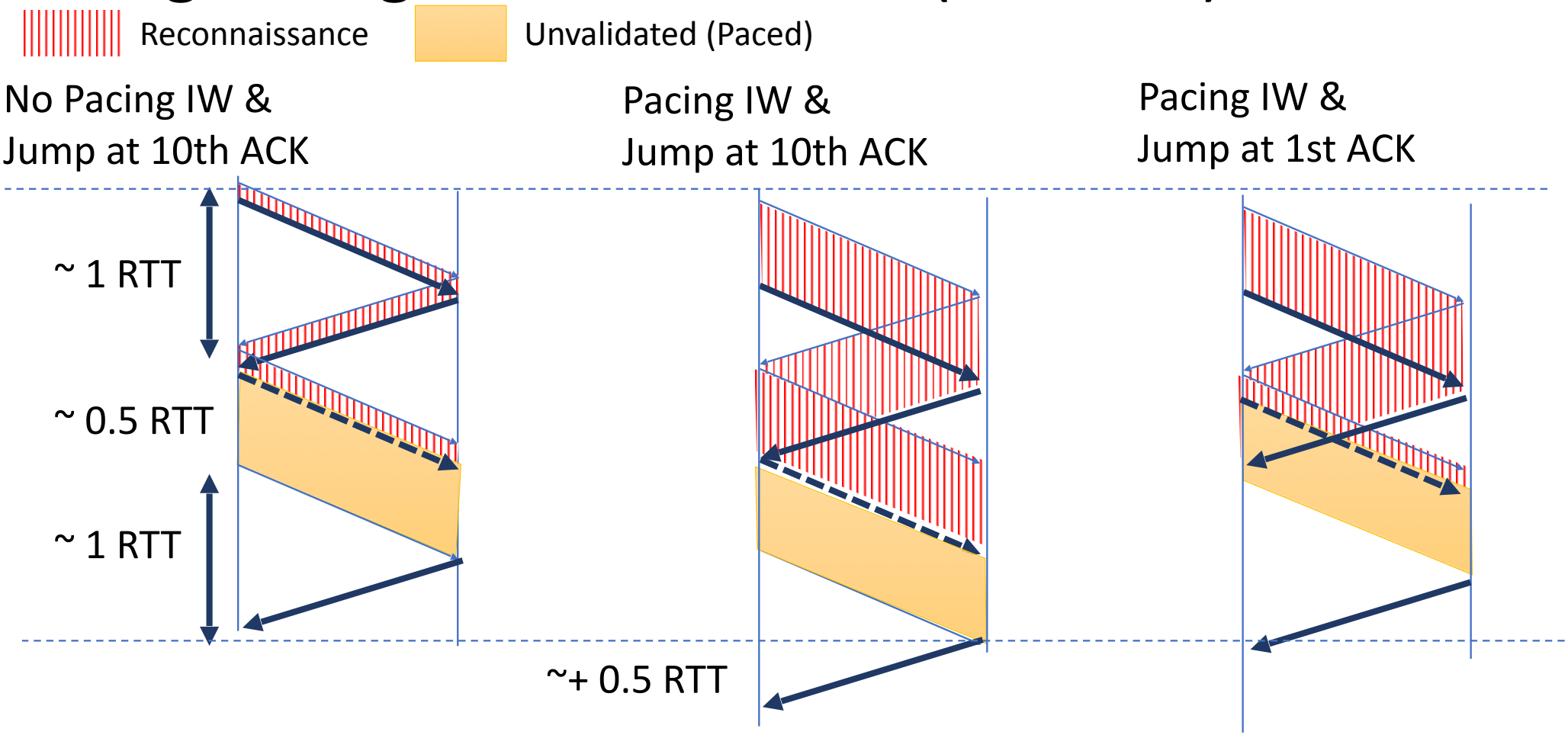
CR, cwnd=900, paced

IW=900, paced

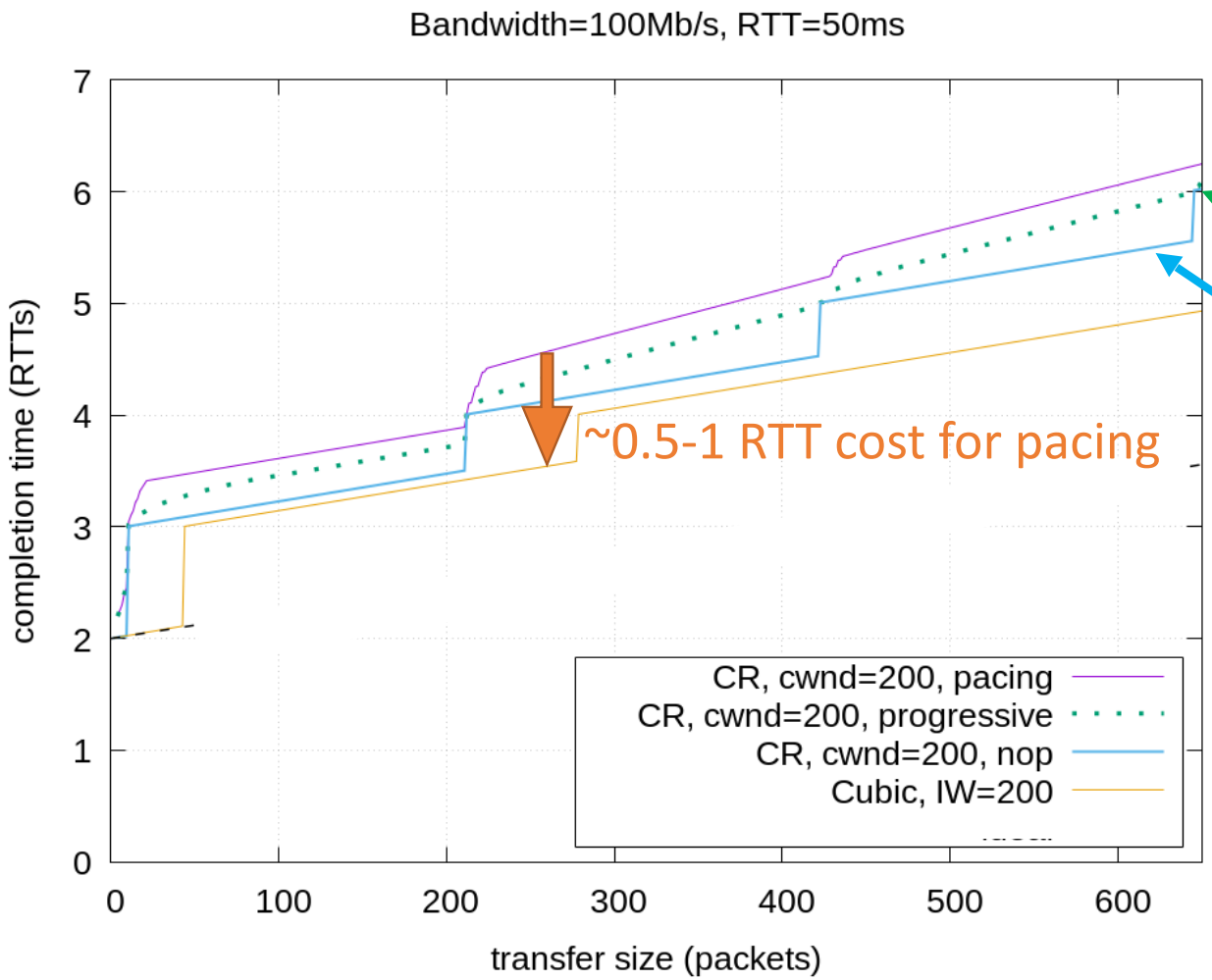
Pacing during Reconnaissance (first RTT)



Pacing during Reconnaissance (first RTT)



Completion Time and Pacing



CR, IW paced

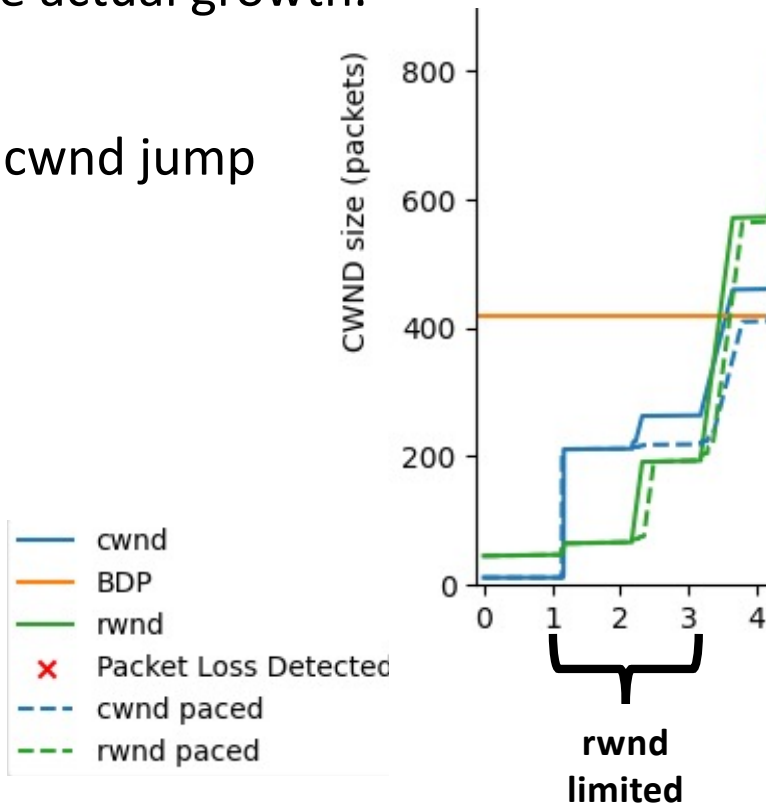
CR, IW paced & prog. growth

CR, jump at 1st ACK
or
CR, IW not paced

And there is flow-control also!

- rwnd/flow credit is set by the receiver
- Receiver typically autotunes rwnd and this restricts the actual growth.

- Receiver would not a-priori set rwnd/flow credit for a cwnd jump
- … see **draft-kuhn-quic-bdpframe-extension**



Planned Next Steps for IETF-118

- Results for QUIC

- Explore Safe Retreat Response
  - How to reset the cwnd after a loss in the Unvalidated Phase?
  - cwnd = IW, cwnd = 2*IW?
  - Specifying the Safe Retreat Phase!

- More WG “help” is always helpful!
  - Which path scenarios ought we consider?