Update

SEARCH – Better Slow Start for TCP and QUIC

Jae Chung
Feng Li
Maryam Ataei Kachooei
Mark Claypool

IETF CCWG
Montral, Canada
November 2025

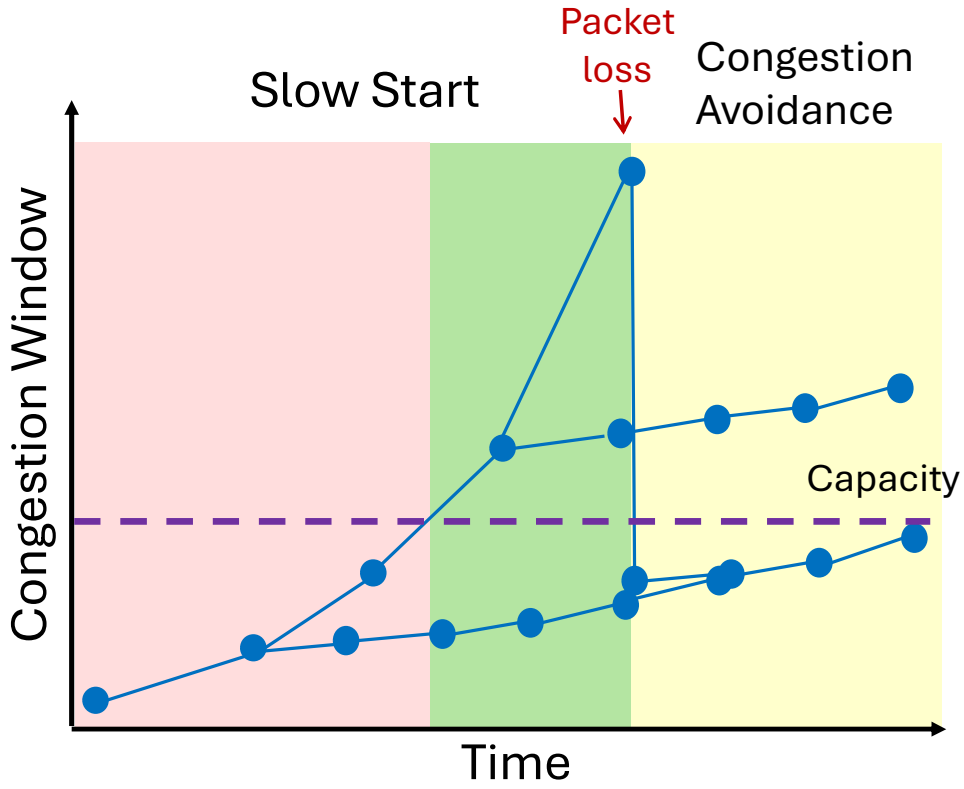
Viasat

WPI

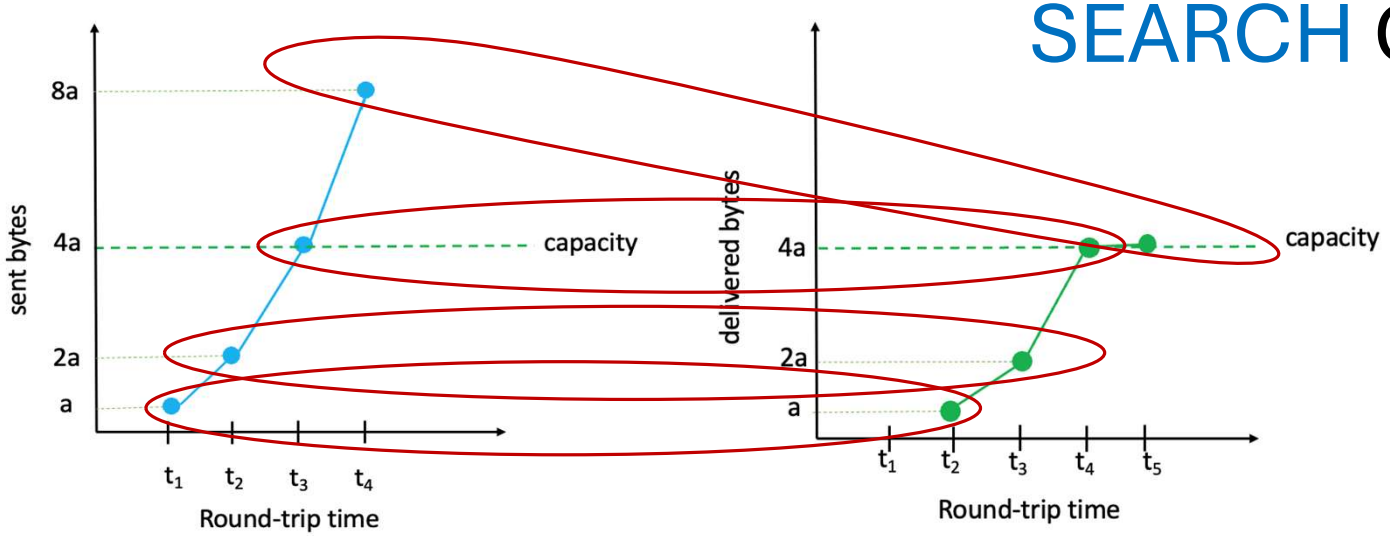
TCP Slow Start

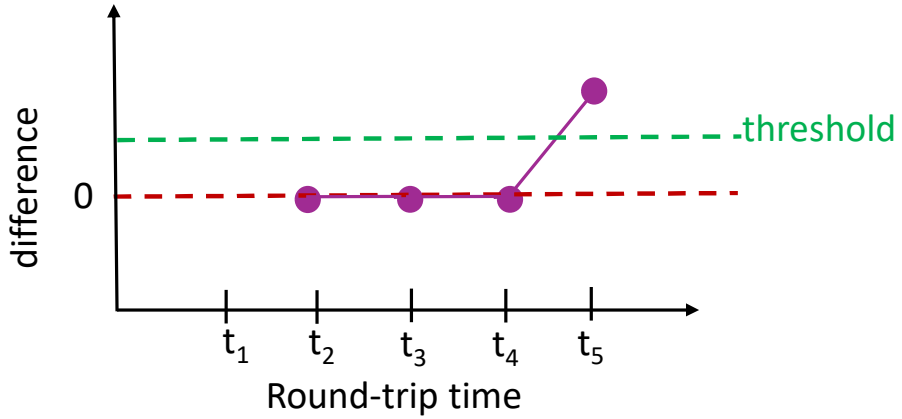
- Exponential growth in congestion window
- Packet loss ➡ Exit slow start: **Too late (After loss)**
- Idea: Exit after reaching capacity and before loss
- HyStart (Linux default)
- HyStart over wireless ➡ **Too early (Before capacity)**

Goal: ➡ **At chokepoint: after capacity, before loss**

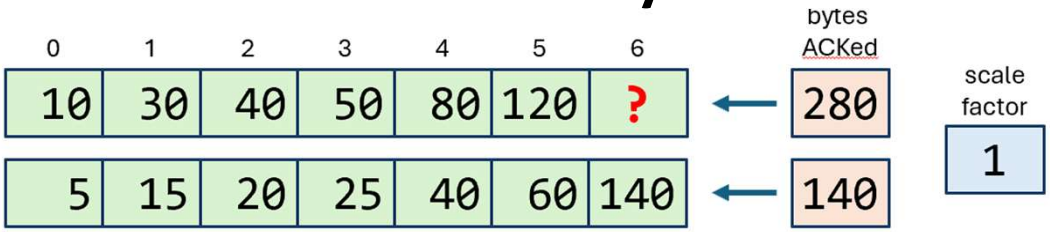


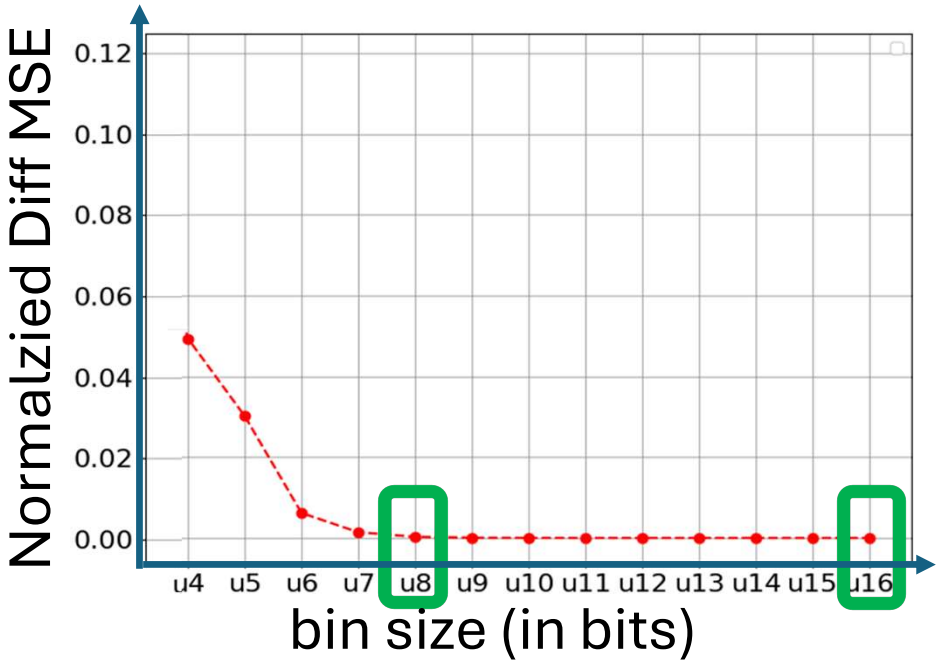
SEARCH Concept

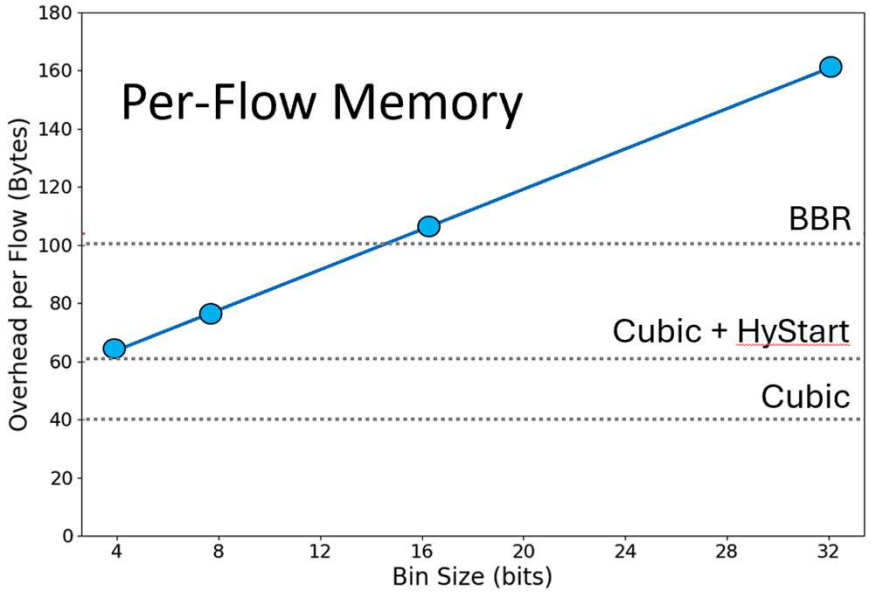




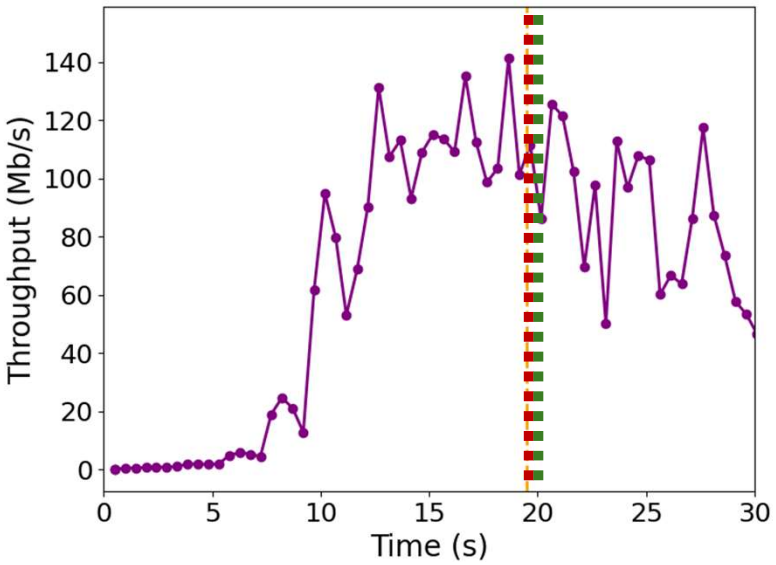
SEARCH Reduced Per-flow Memory

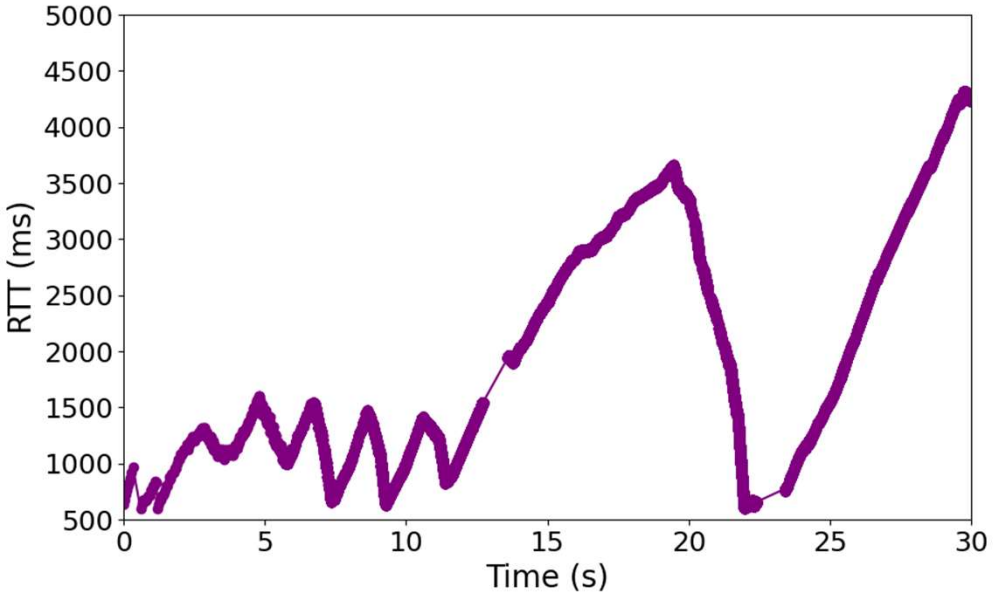






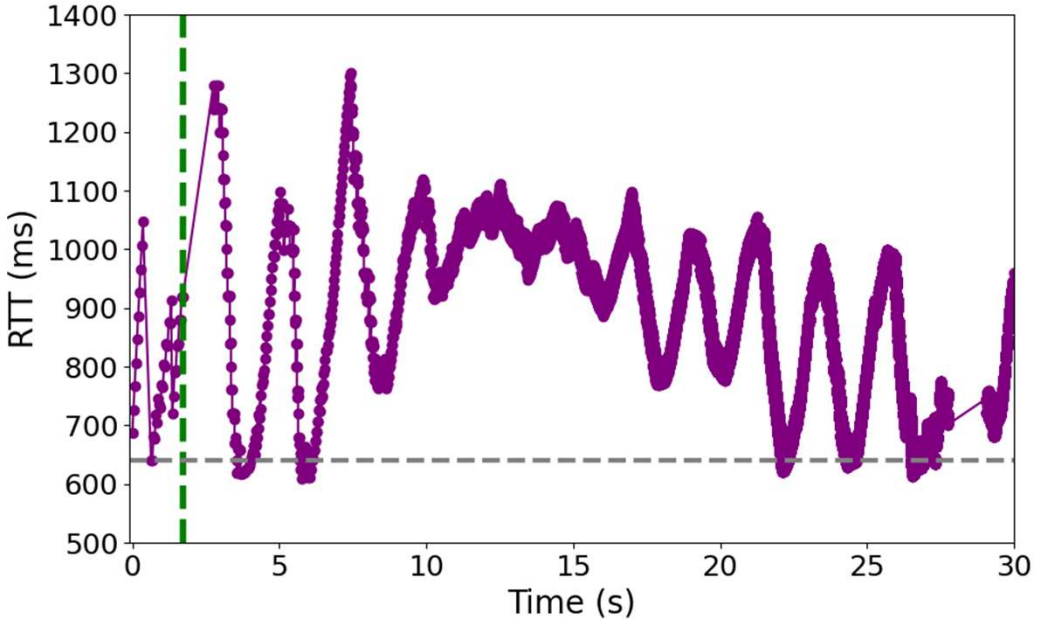
TCP Cubic (without HyStart)

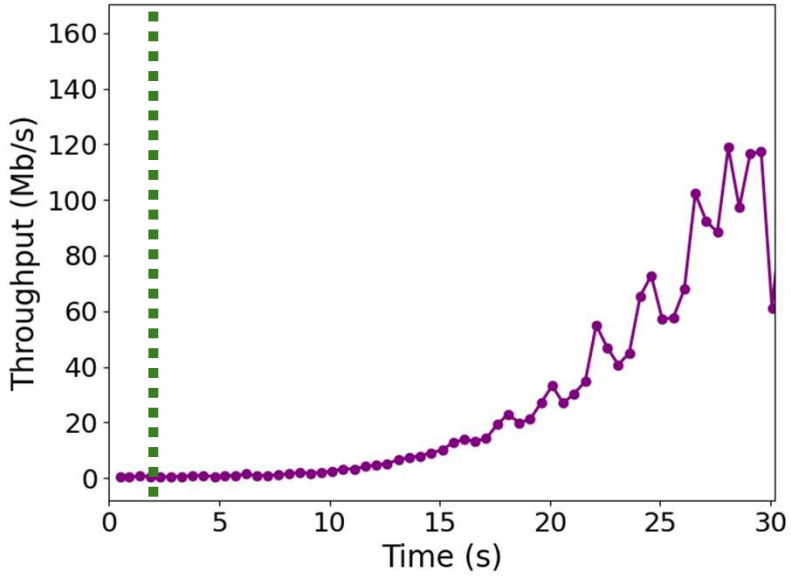




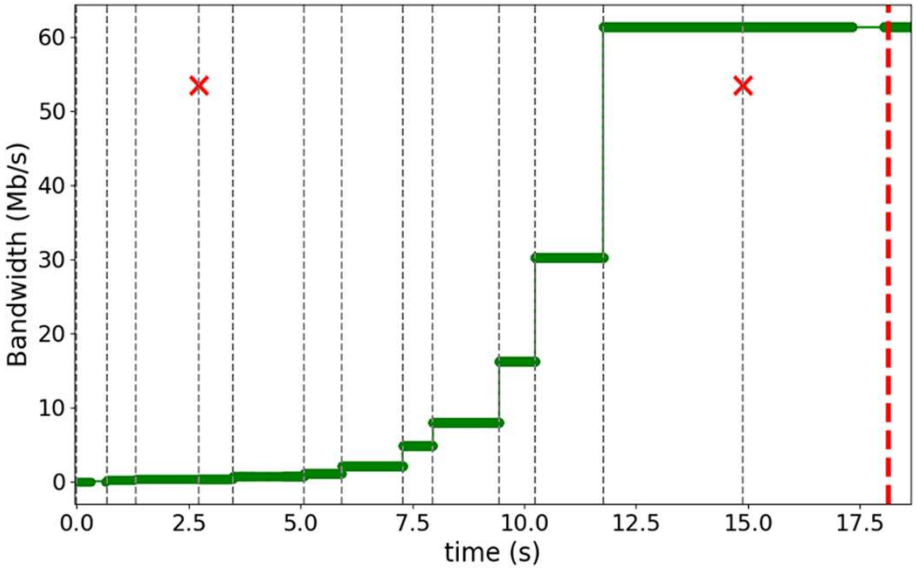
Delay-Based: HyStart, HyStart++

- RTT “spikes” → exit SS
- Sensitive to RTT variation → early exit, under utilization
  - Bad in high BDP networks

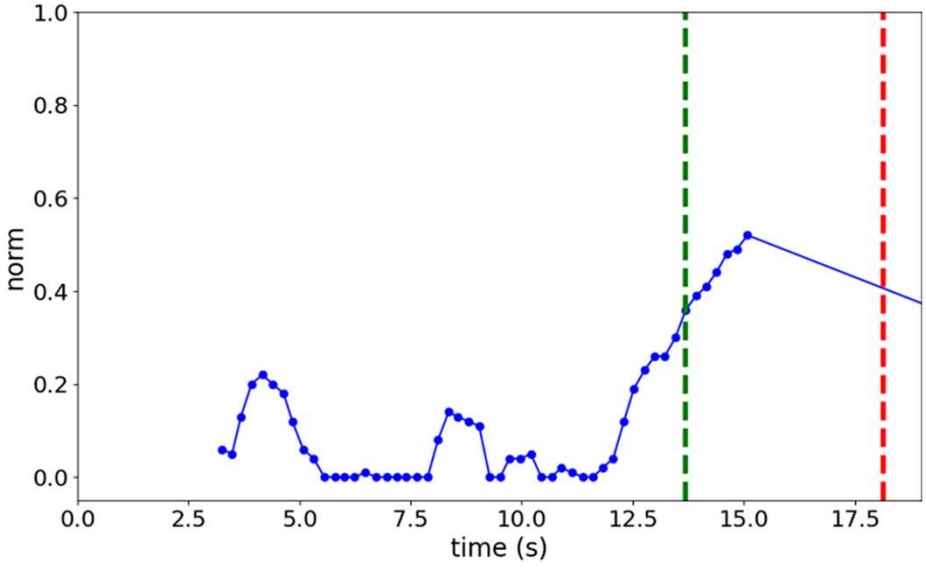




Delivered Bytes-Based: BBR, SEARCH



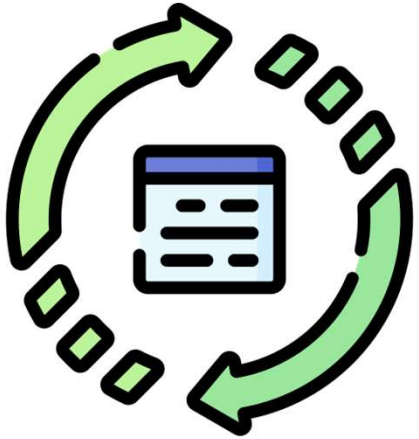
BBR: Uses current RTT, responds slowly after capacity



SEARCH: Uses initial RTT → resilient to post-capacity delay

Update

1. Empirical evidence of safety and intent to deploy
2. Compare **SEARCH** to HyStart++



Linux – HyStart++

Zimraan Ahmad
Joshua Solomon
Clive Thompson
WPI

Balasubramanian, P., Huang, Y., and M. Olson, "**HyStart++: Modified Slow Start for TCP**", *IETF RFC 9406*, DOI 10.17487/RFC9406, May 2023 https://www.rfc-editor.org/info/rfc9406.

M. Arghavani, H. Zhang, D. Eyers, and A. Arghavani, "**SUSS: Improving TCP Performance by Speeding Up Slow-Start**," in *Proceedings of the ACM SIGCOMM* Conference, Sydney, Australia, 2024 https://dl.acm.org/doi/abs/10.1145/3651890.3672234

M. Arghavani, "**HyStartPP**" [Source Code]. Accessed: November 2, 2025 https://github.com/SUSSdeveloper/HyStartPP

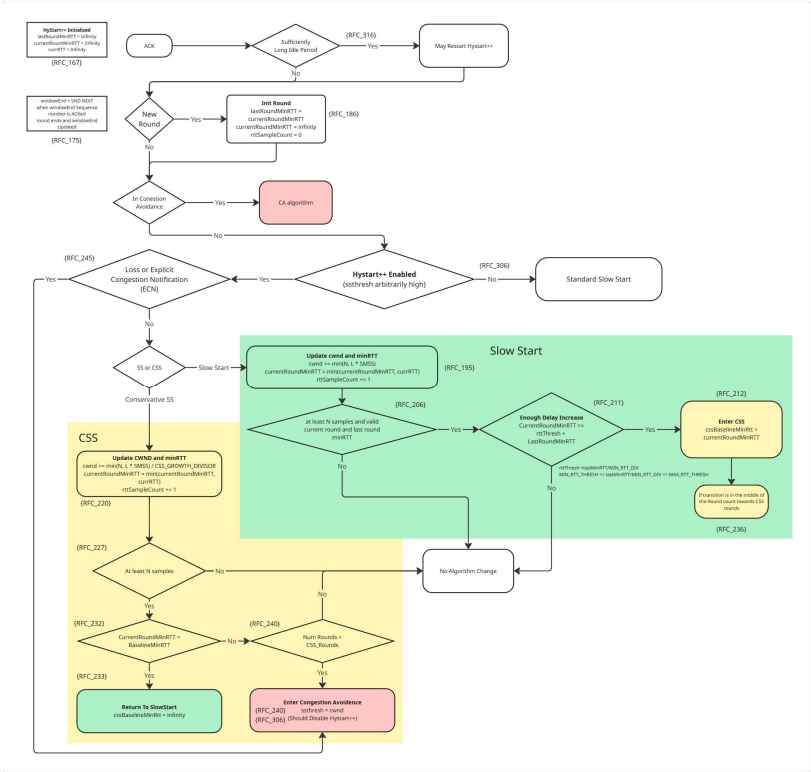
RFC
REQUEST FOR COMMENTS
9406 HyStart++

GitHub
HyStartPP

Linux – HyStart++

Zimraan Ahmad
Joshua Solomon
Clive Thompson
WPI

RFC
REQUEST FOR COMMENTS
9406 HyStart++

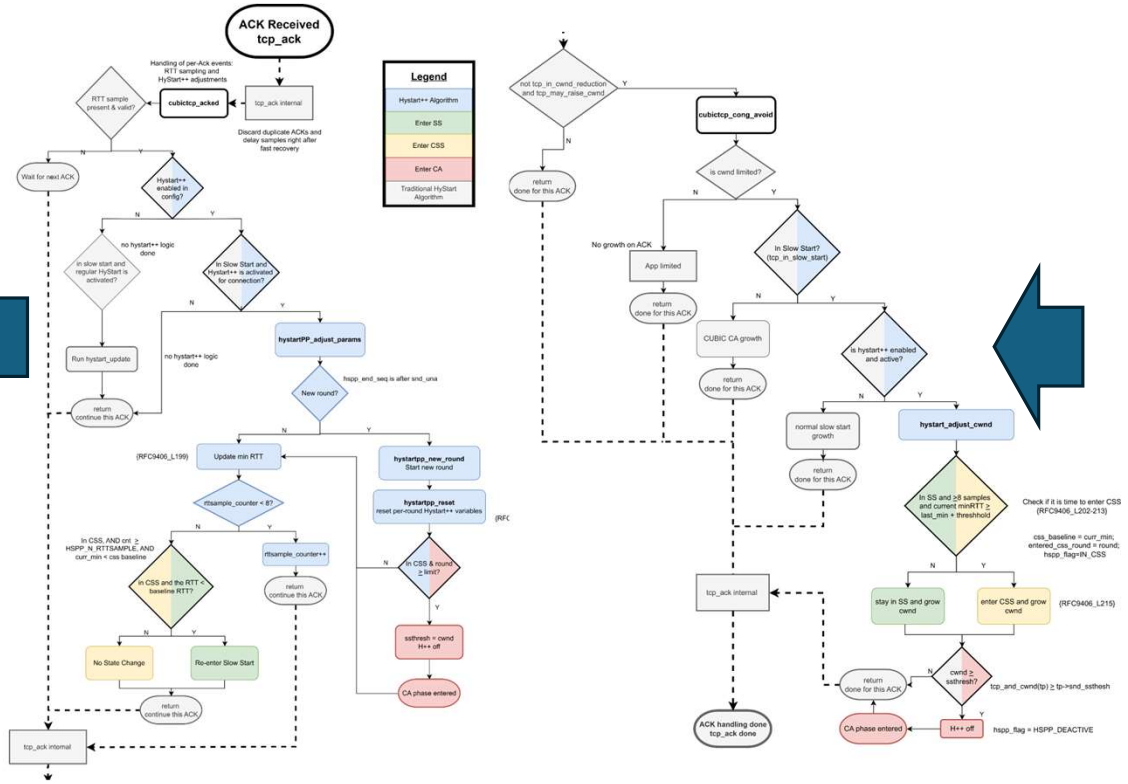


GitHub
HyStartPP

Linux – HyStart++

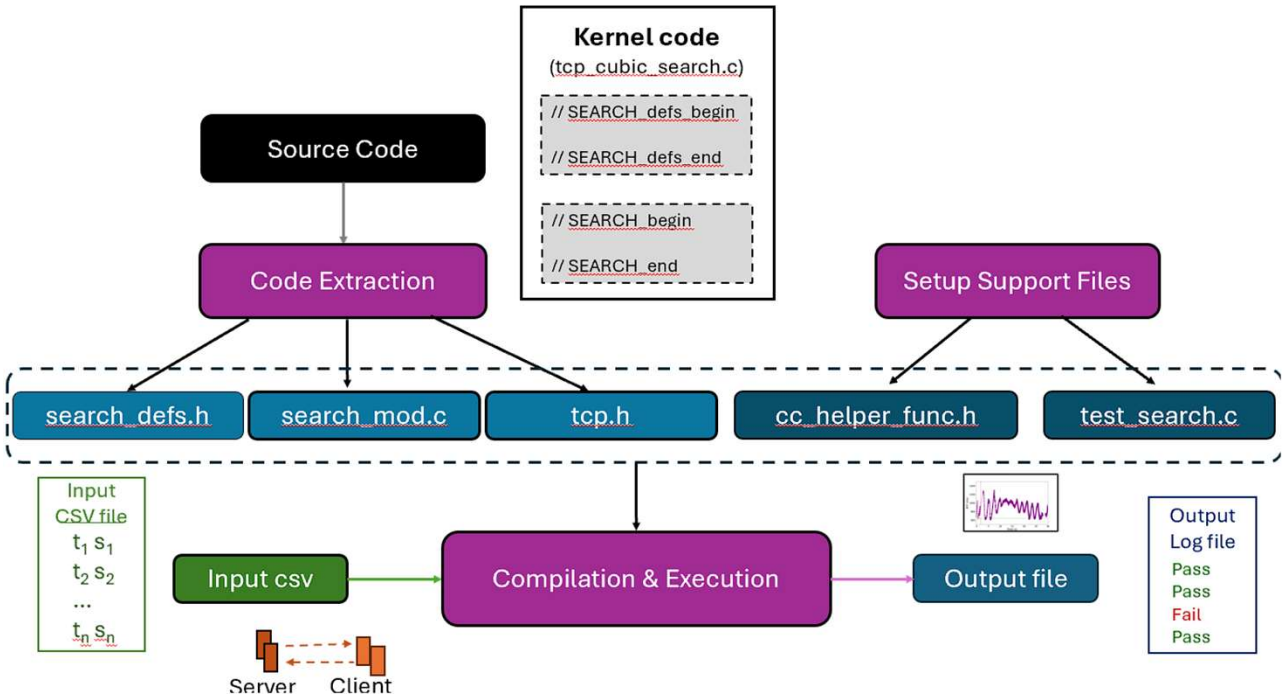
Zimraan Ahmad
Joshua Solomon
Clive Thompson
WPI

RFC
REQUEST FOR COMMENTS
9406 HyStart++



GitHub
HyStartPP

Linux – HyStart++: Verification



Verification:

HyStart++ coded properly?

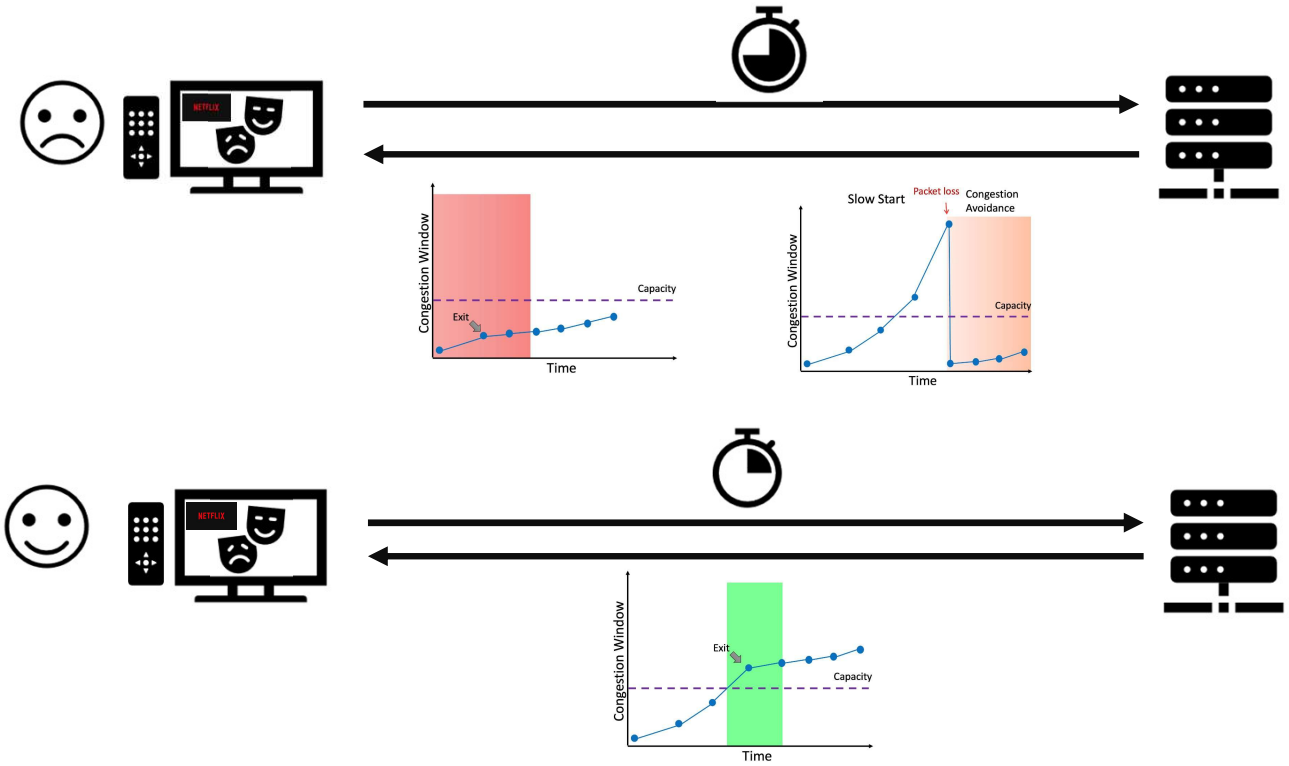
1. SS goes to CSS, CSS goes to CA
2. SS goes to CSS, CSS goes back to SS

Validation:

HyStart++ compared to SEARCH

Empirical Evidence of Safety & Intent to Deploy

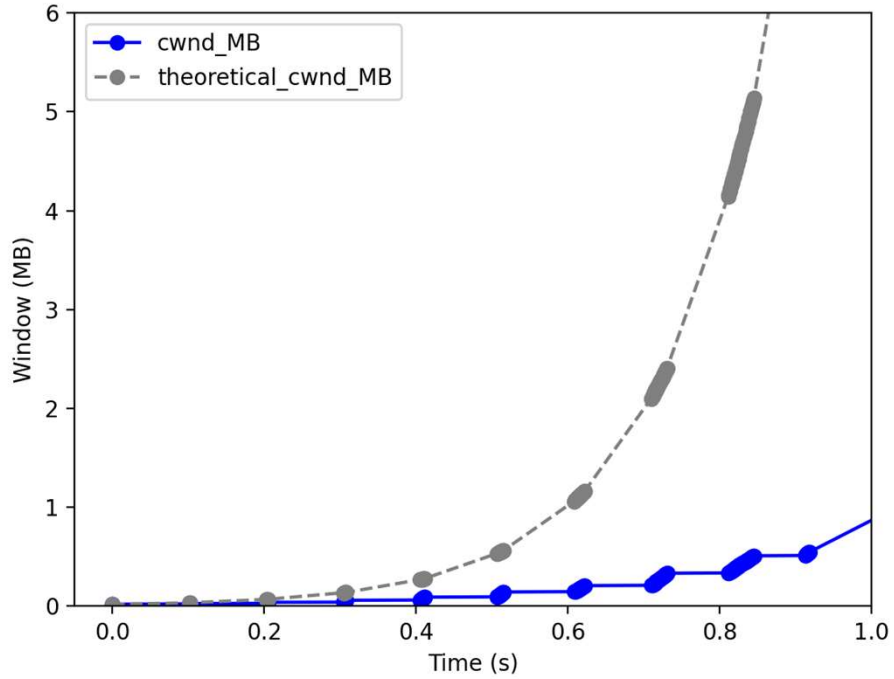
- Evaluate SEARCH in a production environment
- Measure delay for streaming video commencement
- Many cases – no difference, but tail of performance distribution

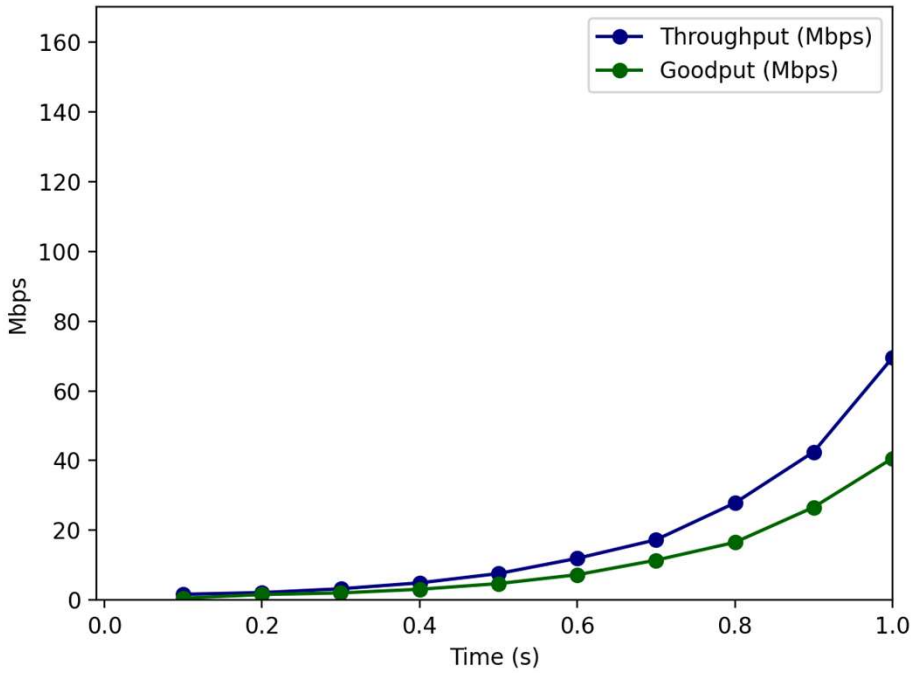


Rate: 80 Mb/s | Delay: 100 ms | Queue: 5x BDP

FreeBSD – Default

Parameter	Default
abc_l_var	2
sendspace	32768
sendbuf_inc	8192
sendbuf_max	2097152
maxsockbuf	2097152

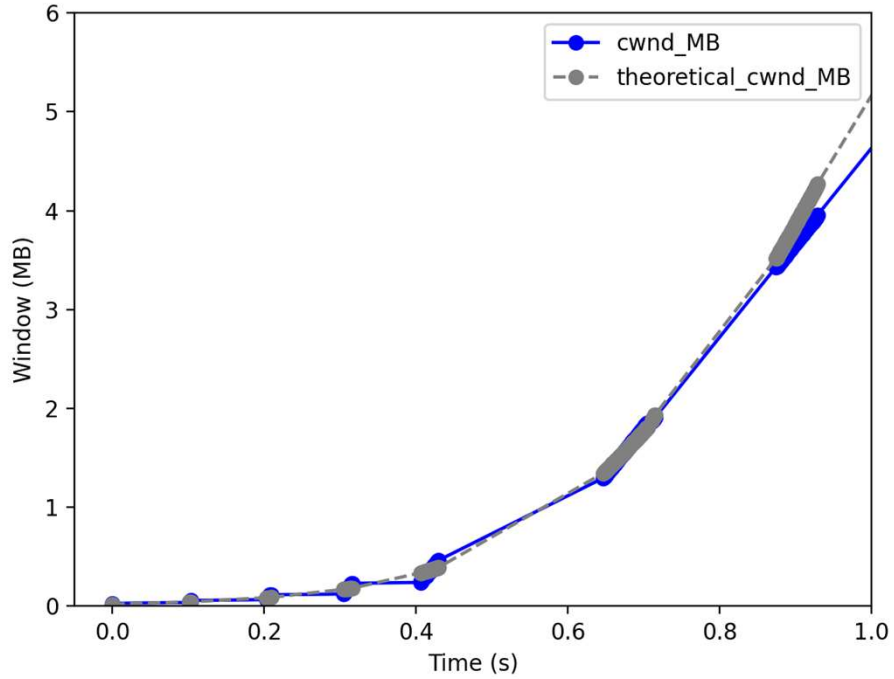


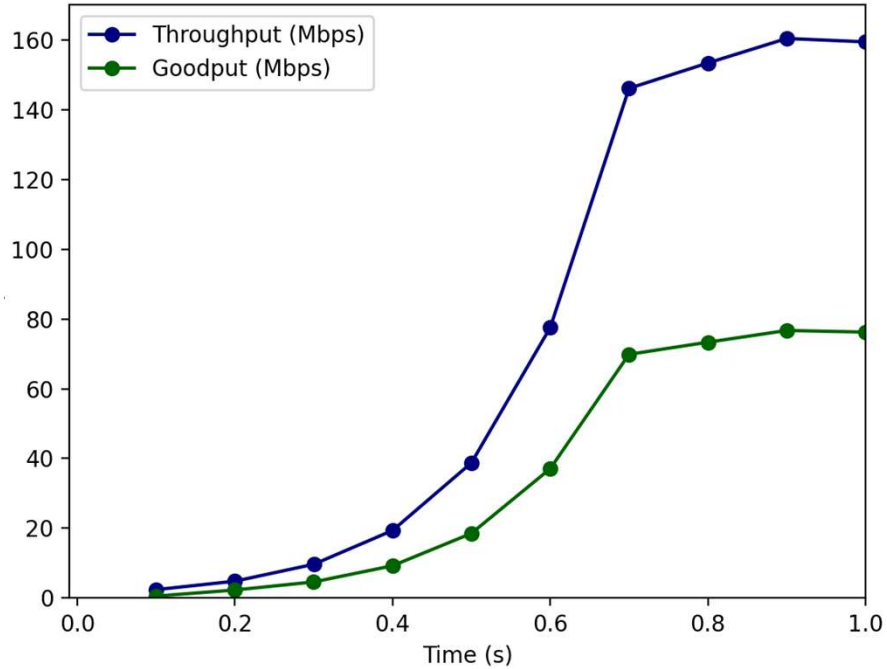


Rate: 80 Mb/s | Delay: 100 ms | Queue: 5x BDP

FreeBSD – Tuned

Parameter	Default	Set Value
abc_l_var	2	2
sendspace	32768	**1048576**
sendbuf_inc	8192	8192
sendbuf_max	2097152	**134217728**
maxsockbuf	2097152	**268435456**

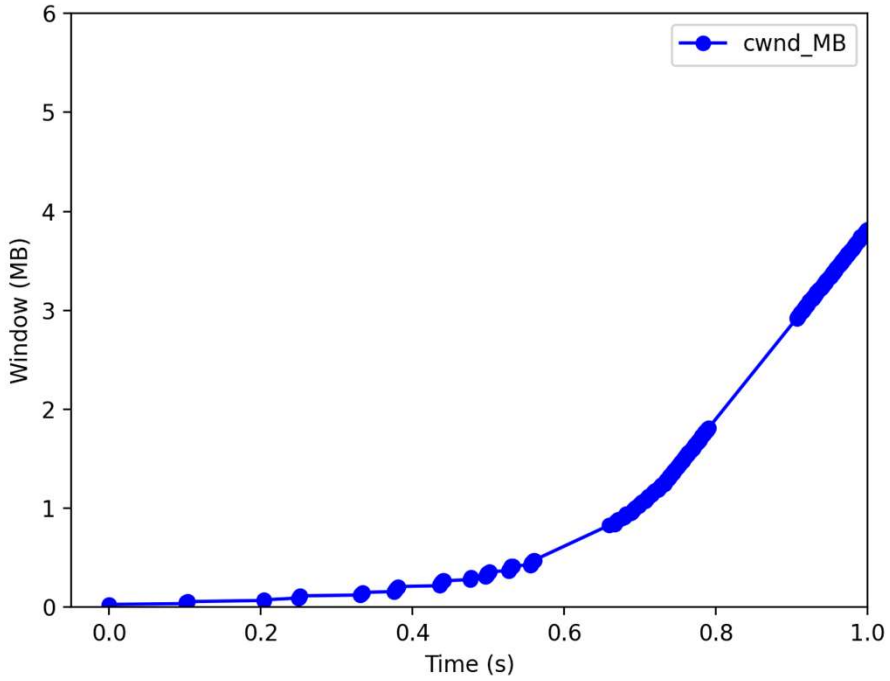


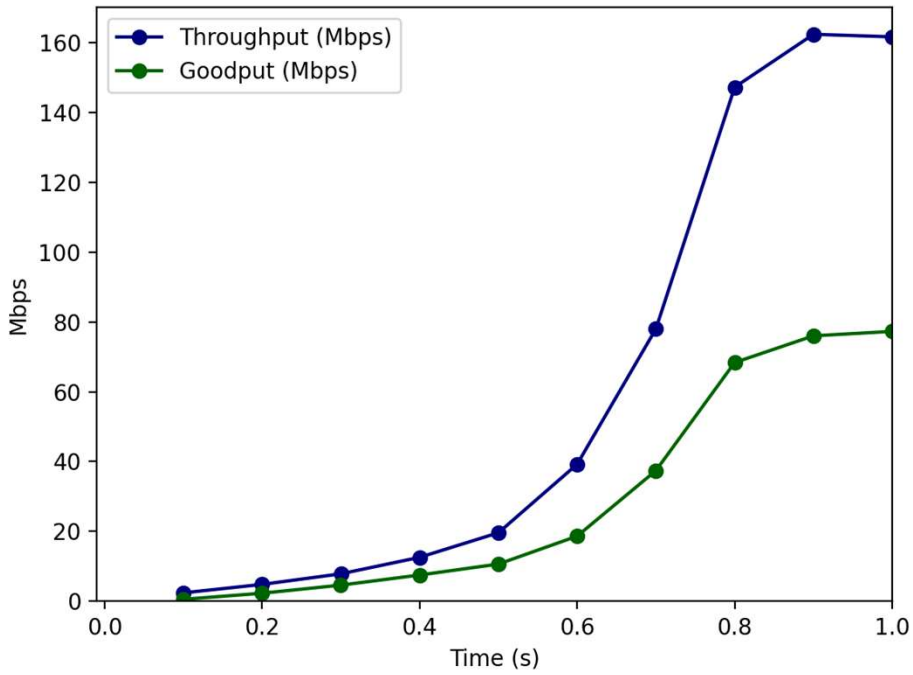


Rate: 80 Mb/s | Delay: 100 ms | Queue: 5x BDP

FreeBSD – Tuned, RACK

Parameter	Default	Set Value
abc_l_var	2	**200**
sendspace	32768	**1048576**
sendbuf_inc	8192	8192
sendbuf_max	2097152	**134217728**
maxsockbuf	2097152	**268435456**

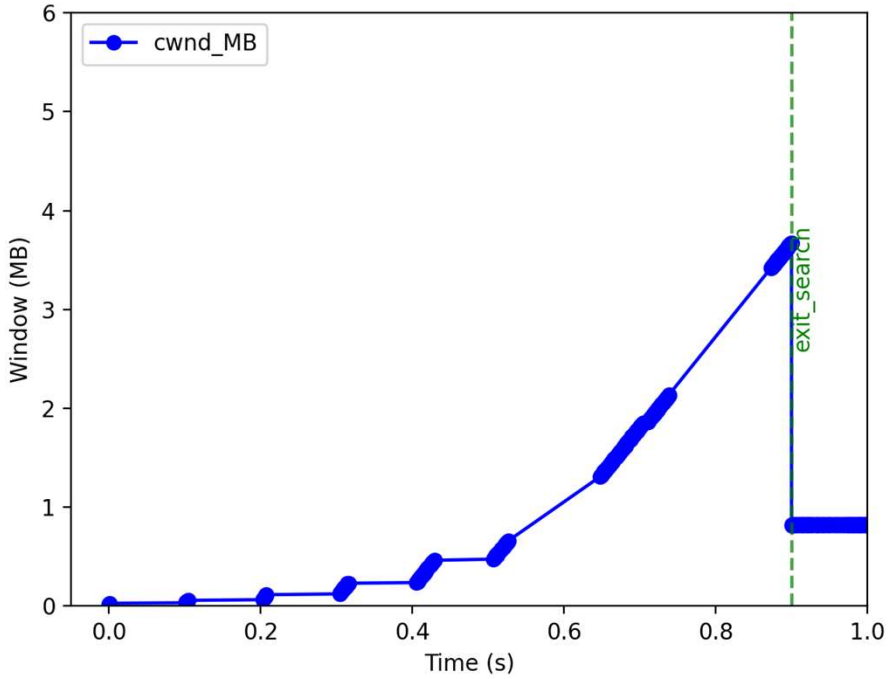


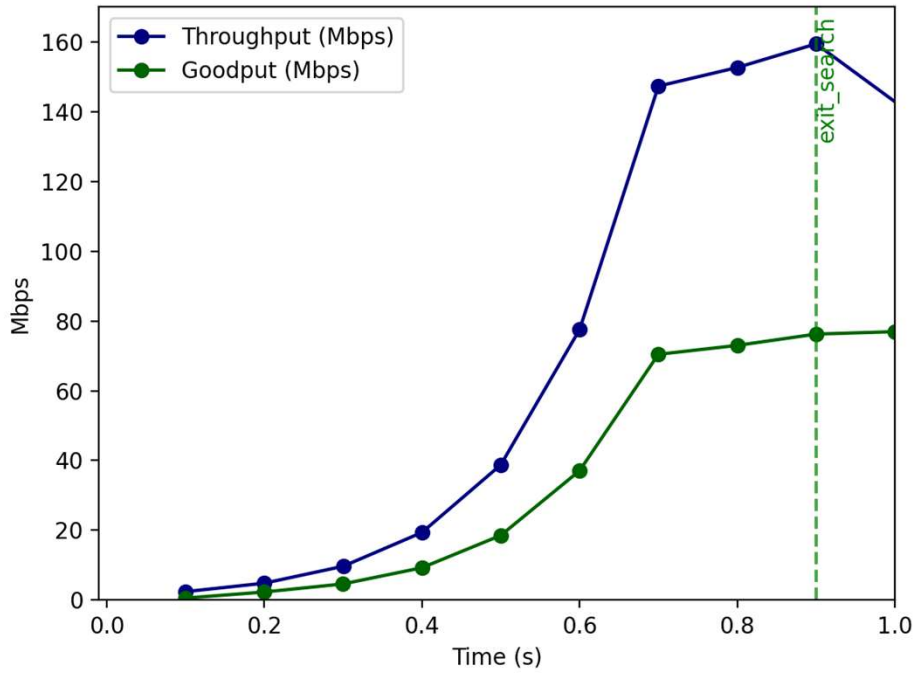


Rate: 80 Mb/s | Delay: 100 ms | Queue: 5x BDP

FreeBSD – Tuned, RACK – SEARCH

Parameter	Default	Set Value
abc_l_var	2	200
sendspace	32768	1048576
sendbuf_inc	8192	8192
sendbuf_max	2097152	134217728
maxsockbuf	2097152	268435456

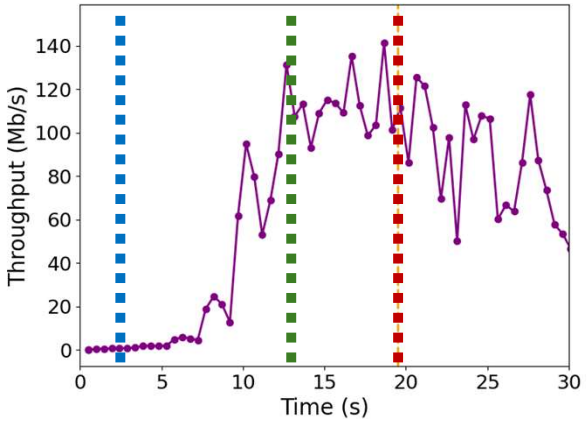




Next Steps

- Verify and validate Linux HyStart++ implementation
- Compare performance
  - **Traditional**, **HyStart++**, **SEARCH**
- Broad-based performance eval
  - Normal, edge cases
  - WiFi, 4g/5g, GEO, LEO, Wired
- Evidence of safety, intent to deploy
  - Tail of the performance distribution

Output
Log file
Pass
Pass
Fail
Pass



Thank-you for your attention!

Update

SEARCH – Better Slow Start for TCP and QUIC

Jae Chung
Feng Li
Maryam Ataei Kachooei
Mark Claypool

Viasat
WPI