Validation and Evaluation of HyStart++ for Linux

Maryam Ataei Kachooei
Clive Thompson
Zimraan Ahmad
Joshua Solomon
Mark Claypool

15 July, 2026

Jae Chung
Feng Li
Benjamin Peters

TCP Slow Start

Goal:

- quickly discover available capacity.
- TCP congestion window approximately doubles for each RTT

Problem:

exponential growth can overshoot.

Need a signal to exit slow start

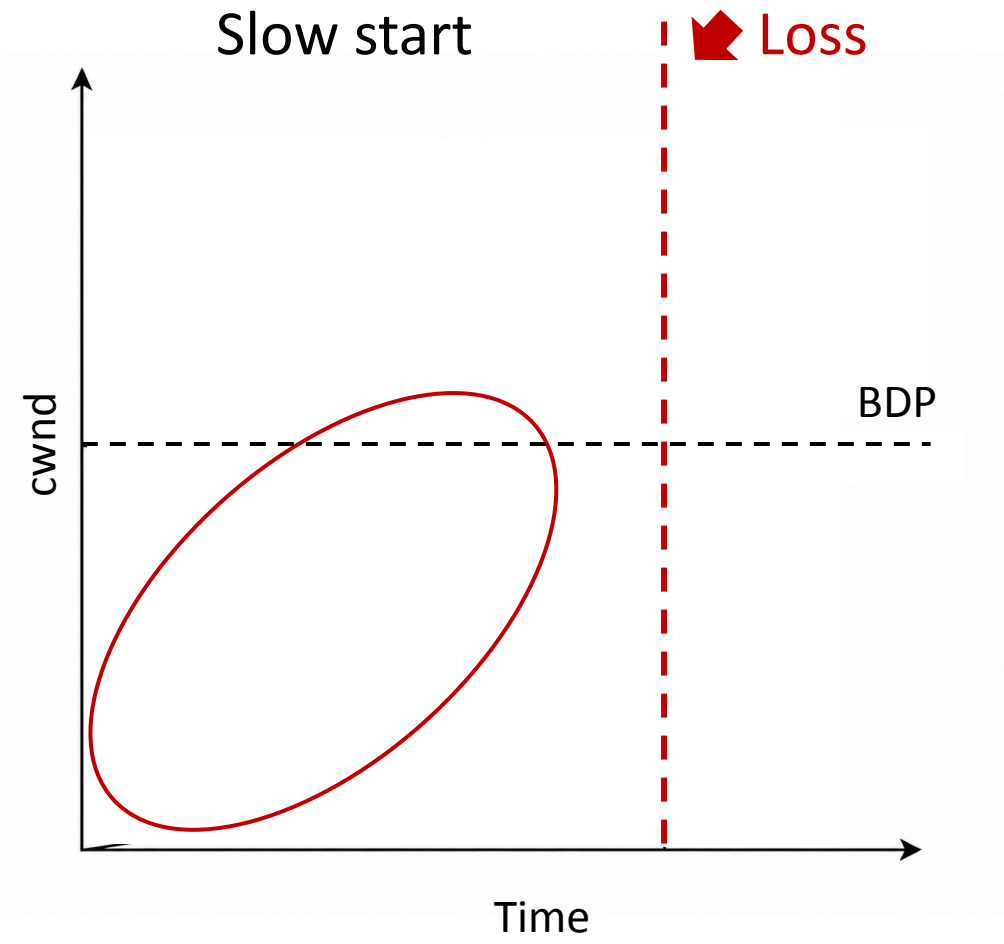
Traditional TCP → Loss → Too late

HyStart → RTT & Ack behavior → Linux default

Too Early →

Limitation

- RTT variation can look like congestion



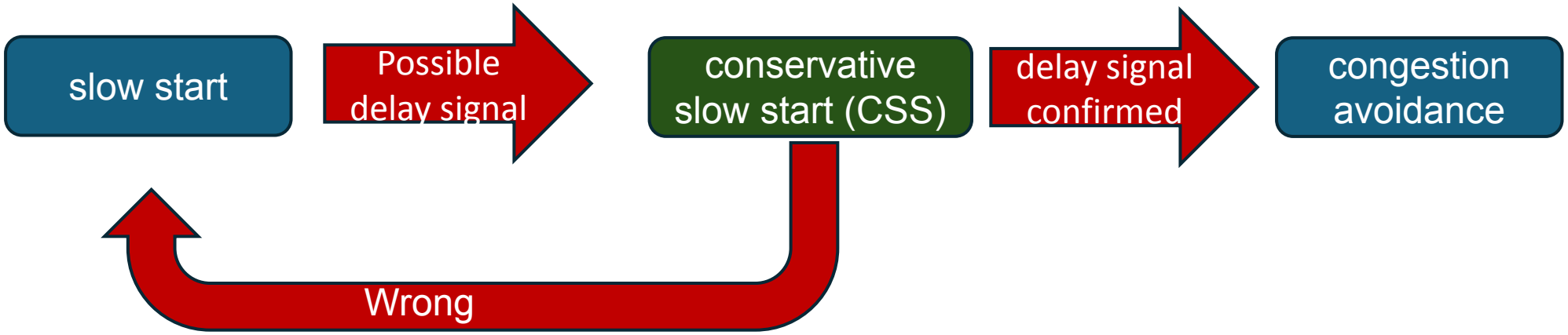
HyStart++

Conservative slow start (CSS): Growth is reduced, not stopped

Exit slow start | Enter congestion avoidance

HyStart

HyStart++



IETF RFC 9406, HyStart++: Modified Slow Start for TCP, May 2023.

Paper Roadmap

1. RFC Flowchart
What HyStart++ is intended to do

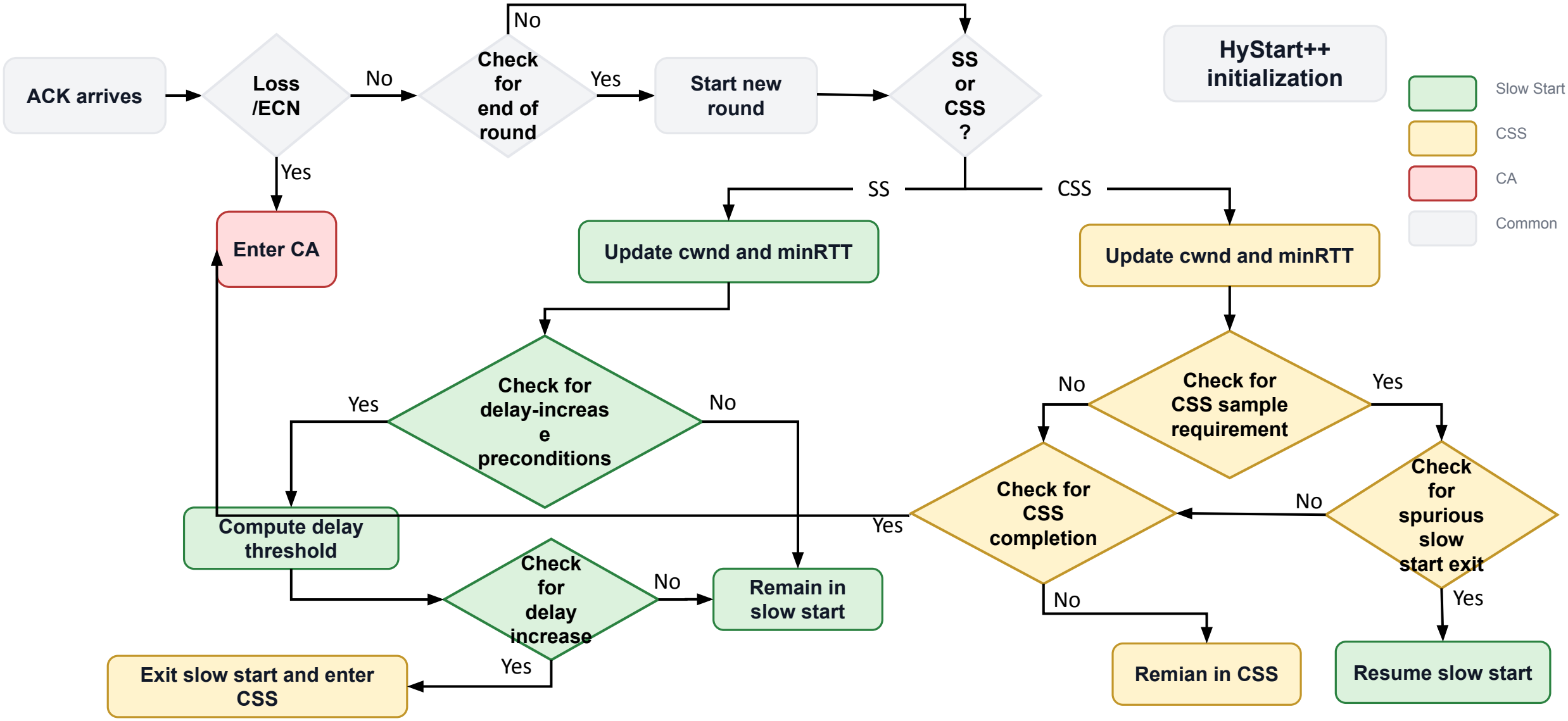
2. Linux Paths
Where that behavior appears

3. Validation Summary
Match

4. Evaluation
Controlled + GEO testbeds

5. Conclusion and future works

HyStart++ RFC (9406)



Paper Roadmap

1. RFC Flowchart
What HyStart++ is intended to do

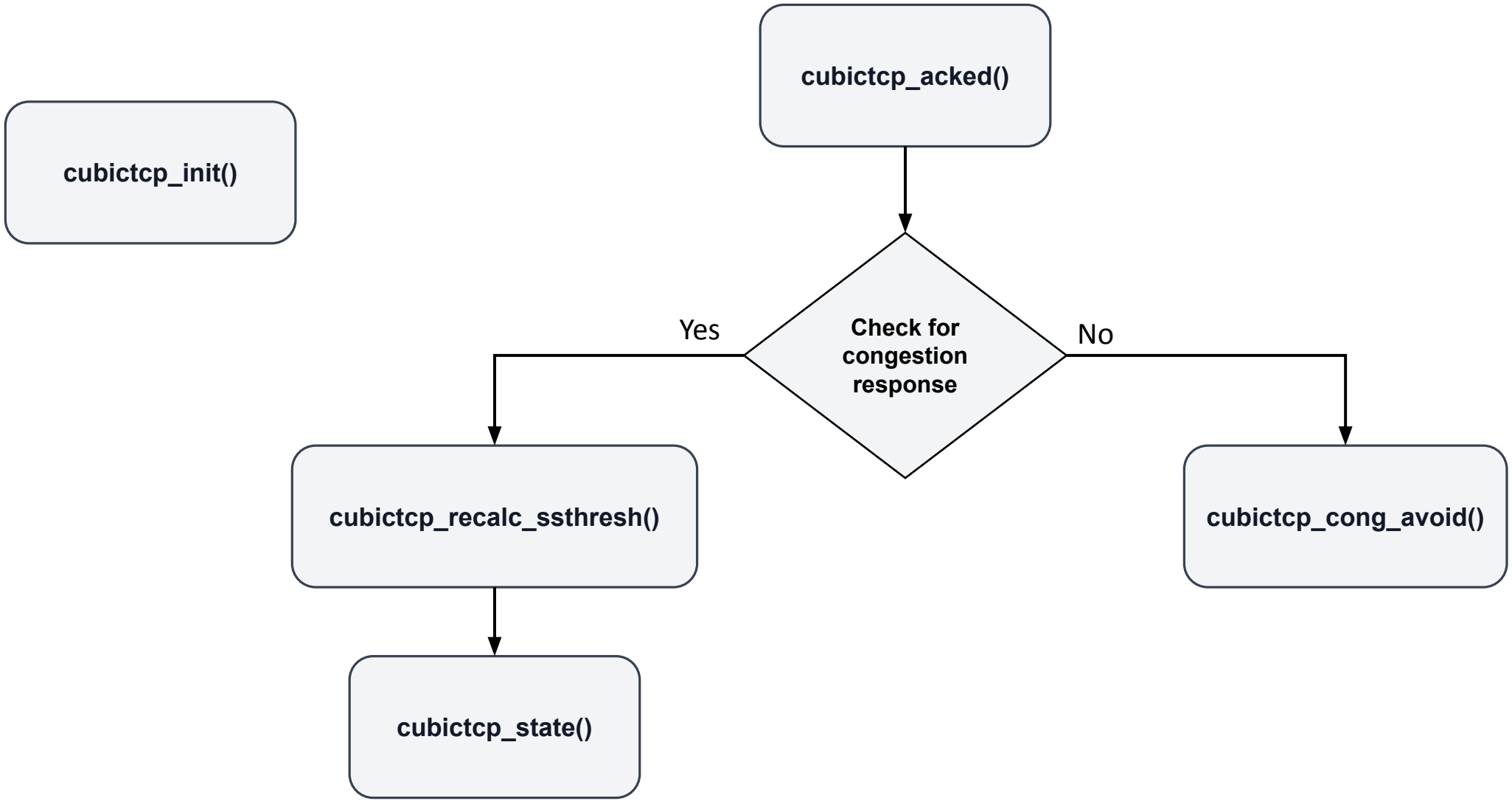
2. Linux Paths
Where that behavior appears

3. Validation Summary
Match

4. Evaluation
Controlled + GEO testbeds

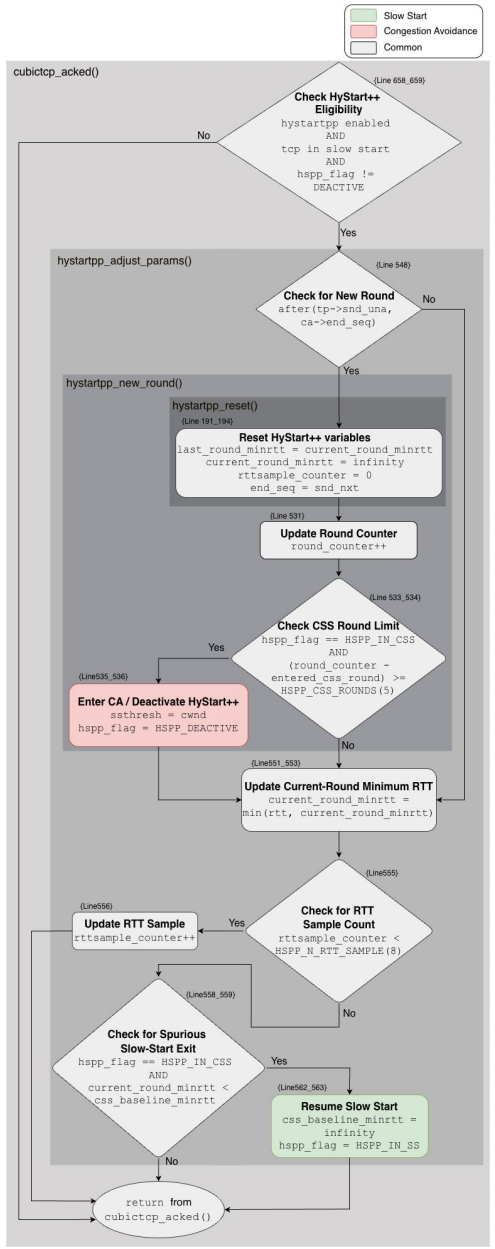
5. Conclusion and future works

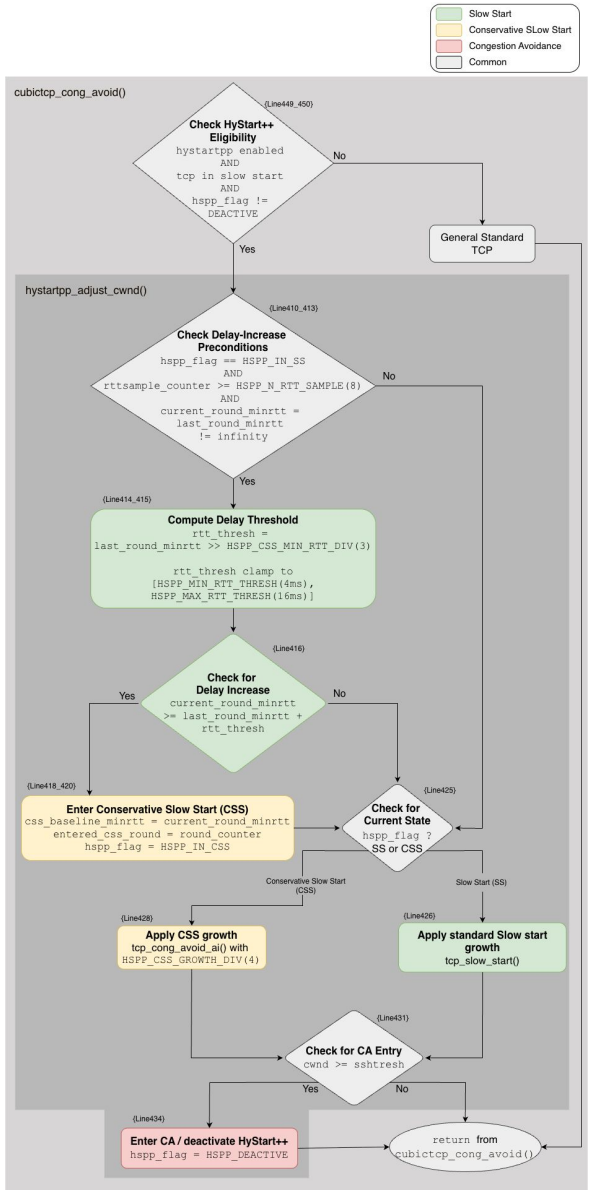
Linux Implementation

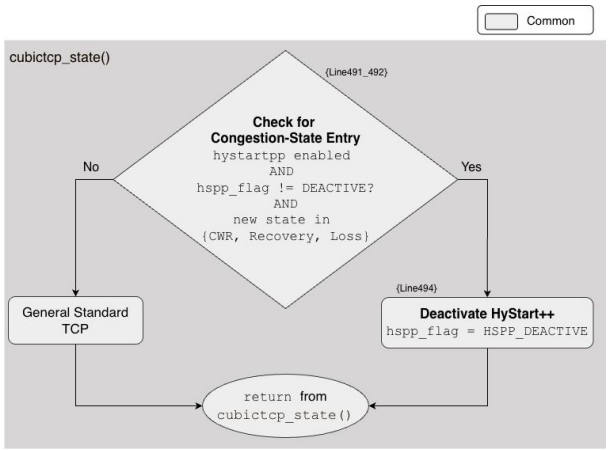


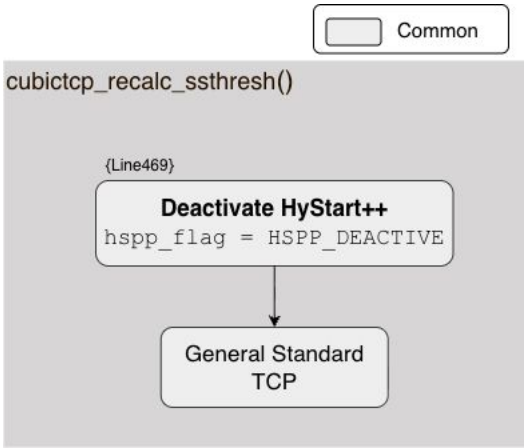
M. Arghavani, HyStartPP: Modified Linux CUBIC Implementation, GitHub, commit c0a089e, 2025.

Linux Implementation









RTT / Rounds

cubictcp_acked() maintains RTT samples, round tracking, rollback to slow start, and exit slow start by completion round in CSS.

cwnd growth

cubictcp_cong_avoid() applies standard slow start or CSS growth and detects CSS entry.

deactivation

state and ssthresh callbacks deactivate HyStart++ after congestion events.

Paper Roadmap

1. RFC Flowchart
What HyStart++ is intended to do

2. Linux Paths
Where that behavior appears

3. Validation Summary
Match

4. Evaluation
Controlled + GEO testbeds

5. Conclusion and future works

Functionality Match

RFC behavior RFC	RFC flowchart state	Linux implementation function
Initialize minimum RTT values and round boundary	**Initialize HyStart++ Variables**	cubictcp_init(), hystartpp_reset()
Start a new round and reset per-round measurements	**Check for End of Round; Start New Round**	cubictcp_acked(), hystartpp_adjust_params(), hystartpp_new_round(), hystartpp_reset()
Update current-round RTT measurements	**Update cwnd and minRTT** (RTT-update portion)	cubictcp_acked(), hystartpp_adjust_params()
Detect delay increase and enter CSS	**Check for Delay-Increase Preconditions; Compute Delay Threshold; Check for Delay Increase; Remain in Slow Start; Exit Slow Start and Enter CSS**	cubictcp_cong_avoid(), hystartpp_adjust_cwnd()
Apply slow start and CSS growth	**Check for Current State; Update cwnd and minRTT** (cwnd-update portion)	cubictcp_cong_avoid(), hystartpp_adjust_cwnd(), tcp_slow_start(), tcp_cong_avoid_ai()
Return from CSS after a spurious slow start exit	**Check for CSS Sample Requirement; Check for Spurious Slow-Start Exit; Resume Slow Start**	cubictcp_acked(), hystartpp_adjust_params()
Complete CSS and enter congestion avoidance	**Check for CSS Completion; Remain in CSS; Enter Congestion Avoidance**	cubictcp_acked(), hystartpp_adjust_params(), hystartpp_new_round()
Handle loss or ECN during slow start or CSS	**Check for Congestion Signal; Enter Congestion Avoidance**	cubictcp_state(), cubictcp_recalc_ssthresh()
Restrict HyStart++ to the initial slow start	**Enter Congestion Avoidance**	cubictcp_init(), hystartpp_new_round(), hystartpp_adjust_cwnd(), cubictcp_state(), cubictcp_recalc_ssthresh()

Functionality Match

RFC behavior RFC	Status	Comments
Initialize minimum RTT values and round boundary		
Start a new round and reset per-round measurements		
Update current-round RTT measurements		
Detect delay increase and enter CSS		
Apply slow start and CSS growth		
Return from CSS after a spurious slow start exit		
Complete CSS and enter congestion avoidance		
Handle loss or ECN during slow start or CSS		
Restrict HyStart++ to the initial slow start		

Paper Roadmap

1. RFC Flowchart
What HyStart++ is intended to do

2. Linux Paths
Where that behavior appears

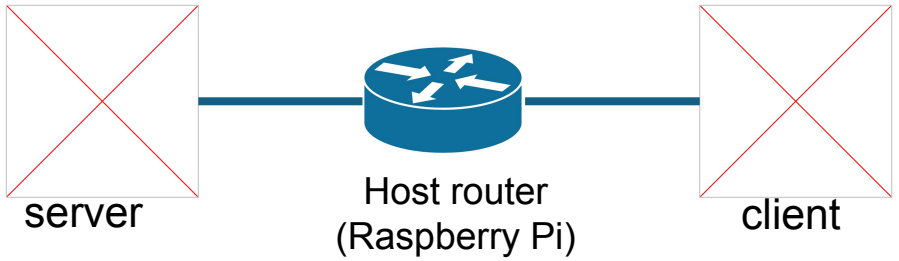
3. Validation Summary
Match

4. Evaluation
Controlled + GEO testbeds

5. Conclusion and future works

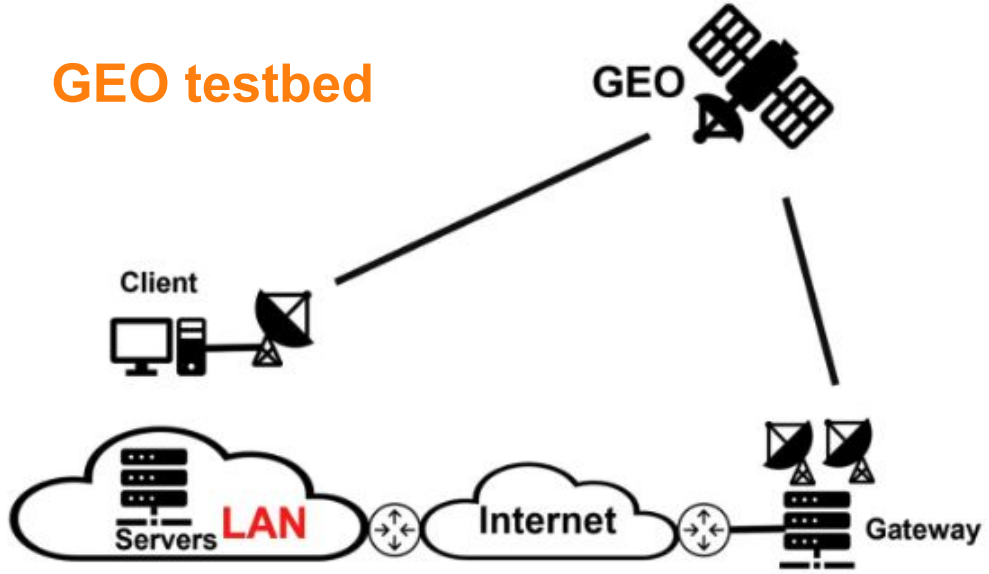
Evaluation Testbeds

Controlled testbed



Factor	Value
Algorithm	Traditional, HyStart, HyStart++
Rate	50, 100, 150 Mb/s
Base RTT	25, 100, 400 ms
Queue	1, 2, 4 X BDP
Jitter	RTT/4, RTT/2, RTT
Duration	10 seconds
Runs	30 per config; 2,430 per algorithm

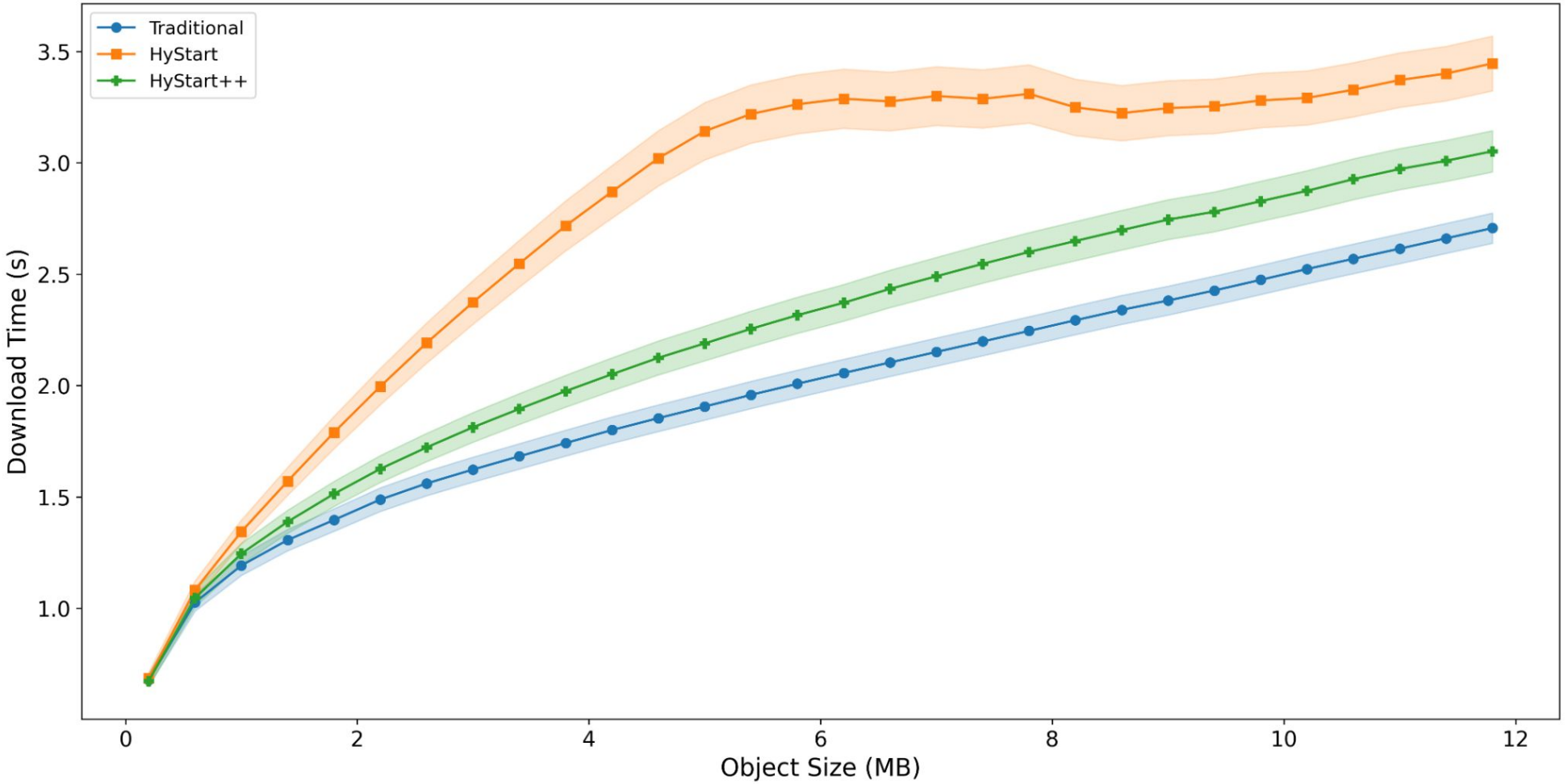
GEO testbed



Factor	Value
Algorithm	Traditional, HyStart, HyStart++
Duration	45 seconds
Schedule	24 hours
Runs	161 per algorithm

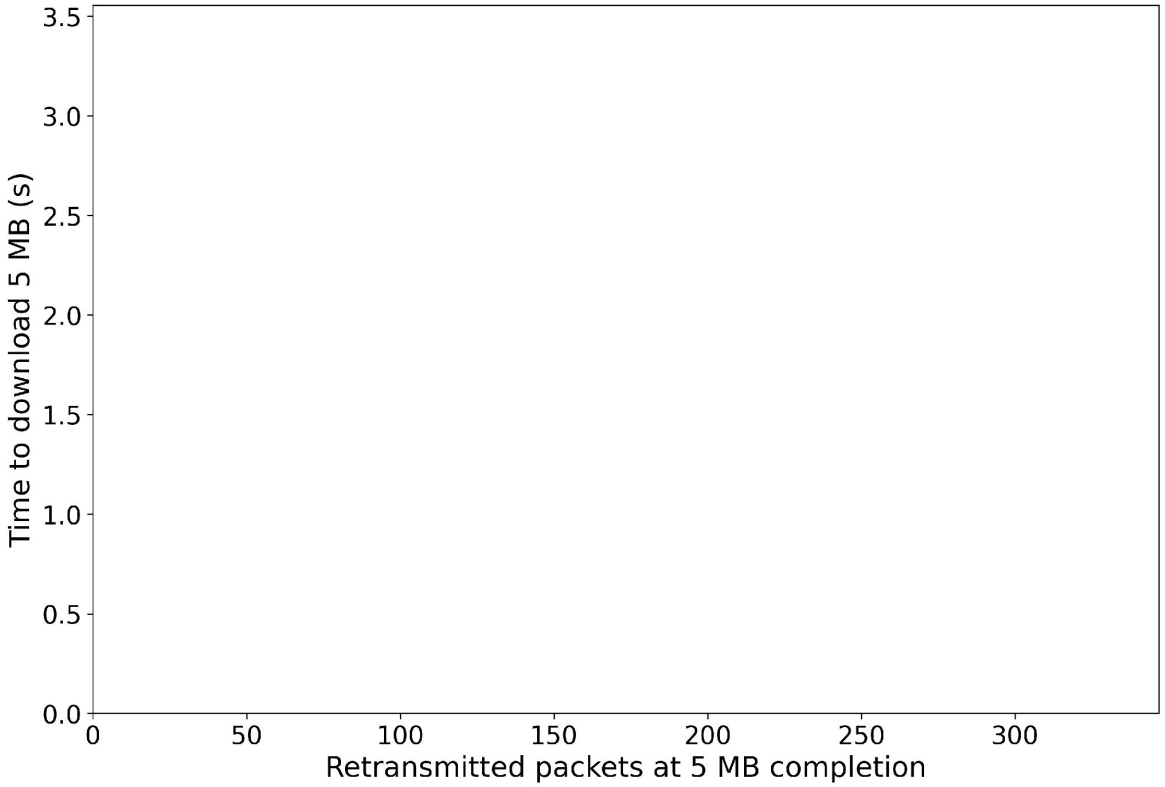
Evaluation Result: Download Time

Controlled dataset

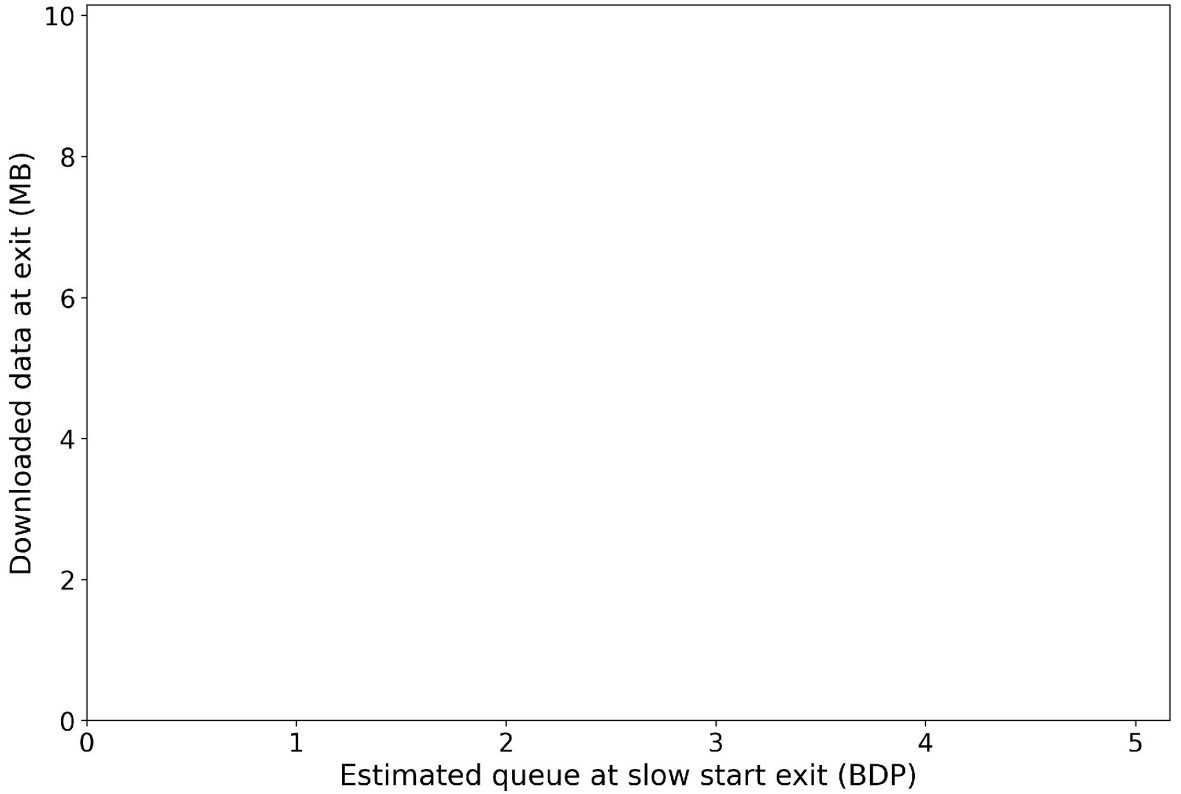


Evaluation Result

Controlled dataset



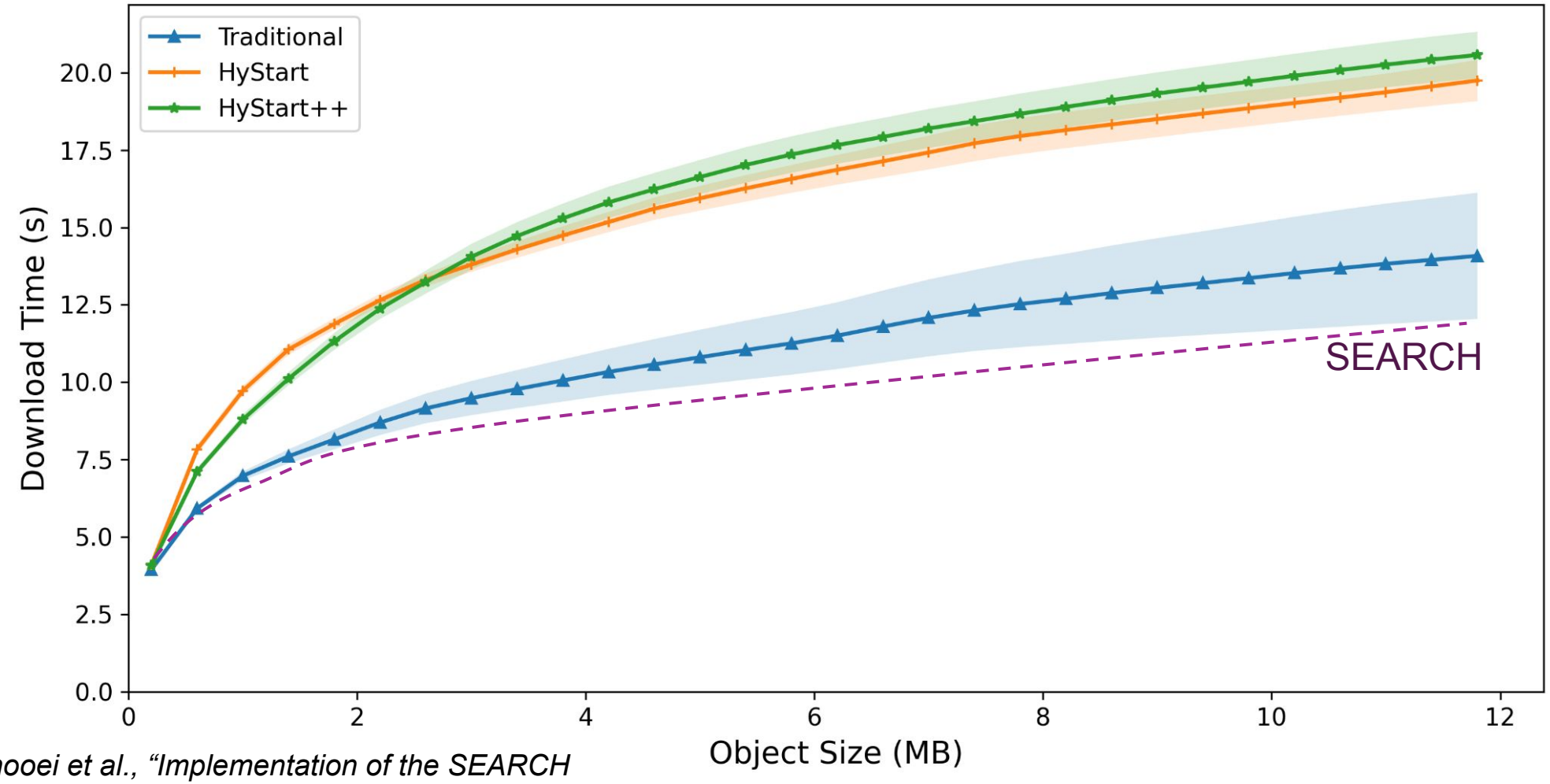
Mean time to download 5 MB versus retransmitted packets at 5 MB completion



Downloaded data and estimated queue occupancy when the slow start exit has happened

Evaluation Result: Download Time

GEO dataset



M. A. Kachooei et al., "Implementation of the SEARCH Slow Start Algorithm in the Linux Kernel," Proc. NetDev 0x18, 2024.

Paper Roadmap

1. RFC Flowchart
What HyStart++ is intended to do

2. Linux Paths
Where that behavior appears

3. Validation Summary
Match

4. Evaluation
Controlled + GEO testbeds

5. Conclusion and future works

Conclusion

- Linux HyStart++ faithfully implements the main RFC 9406 behavior:

  Slow Start → CSS → Congestion Avoidance

- CSS reduces premature HyStart exits by probing after the first delay signal.
- HyStart++ is a middle point:

  less conservative than HyStart, less aggressive than traditional slow start

- RTT variation still challenges delay-based exit on wireless/satellite paths.

Future Work

- Tune HyStart++ parameters for high-variation paths:

  CSS duration, CSS growth rate, delay threshold

- Explore delivery/rate-based signals for slow start exit:
  - RTT-confirmed signal
  - Standalone signal

Thank you for your attention!

Validation and Evaluation of HyStart++ for Linux

Maryam Ataei Kachooei
Clive Thompson
Zimraan Ahmad
Joshua Solomon
Mark Claypool

15 July, 2026

Jae Chung
Feng Li
Benjamin Peters