Fixing TCP Slow Start for Slow Fat Links

Maryam Ataei Kachooei

Pinhan Zhao

Mark Claypool

WPI

Feng Li

Jae Chung

Viasat

Introduction – Satellites

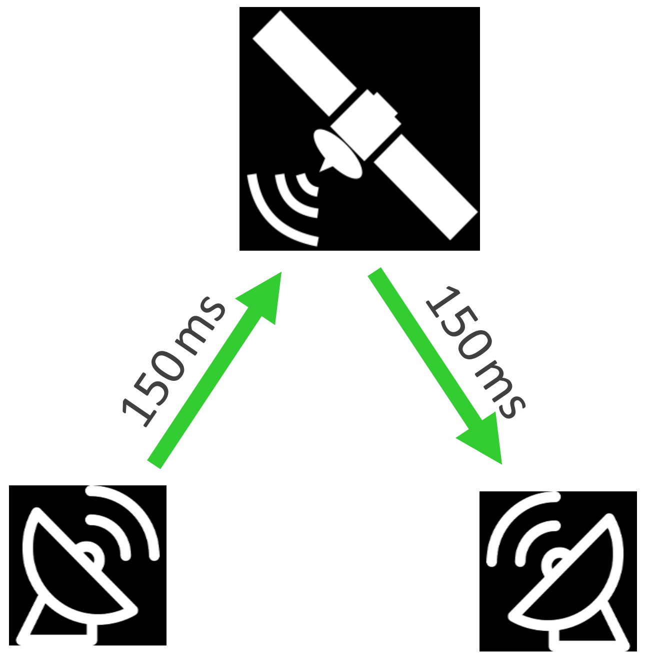
Geo Satellites provide global networking

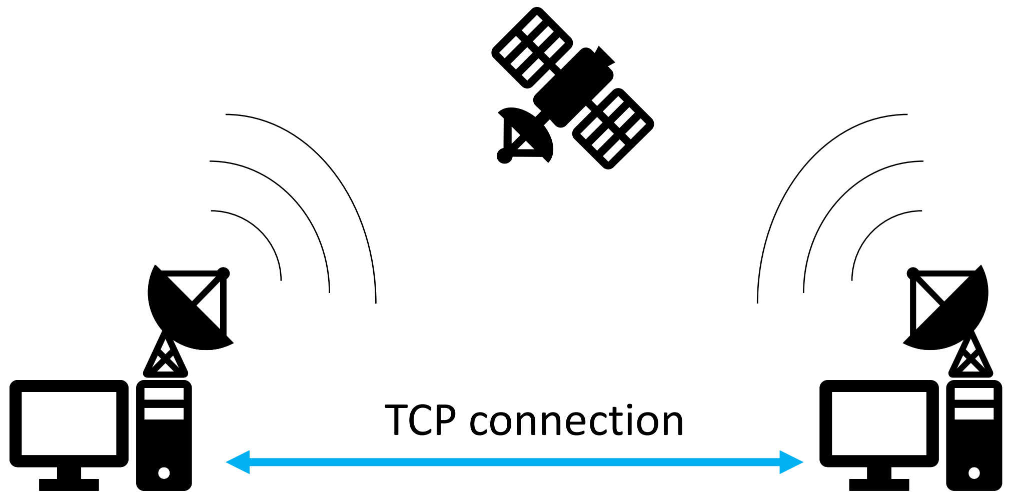
- Remote locations
- On airplanes
- During natural disasters
- Bitrates increasing (20x recently)

Challenge is latency

- About 600 ms round trip

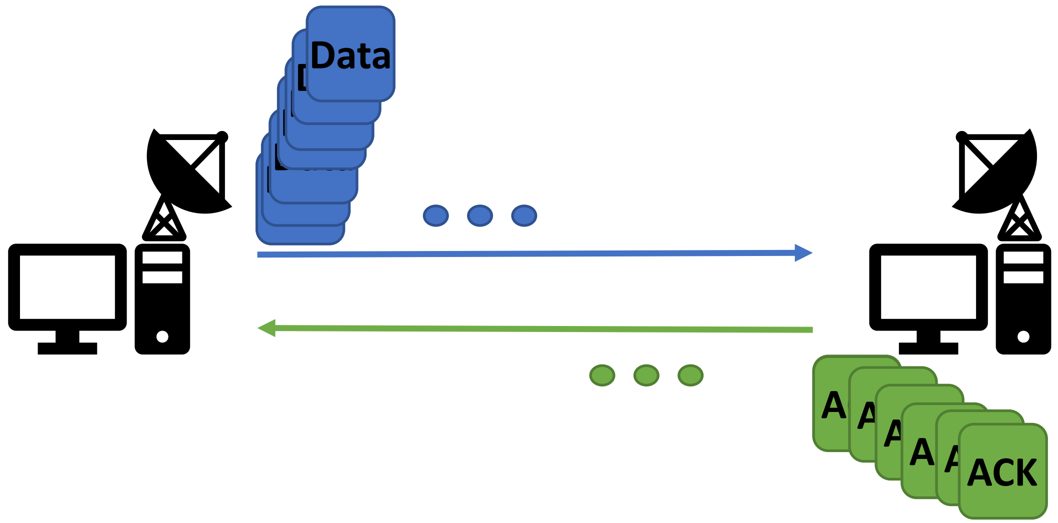
Latency impacts TCP bitrates

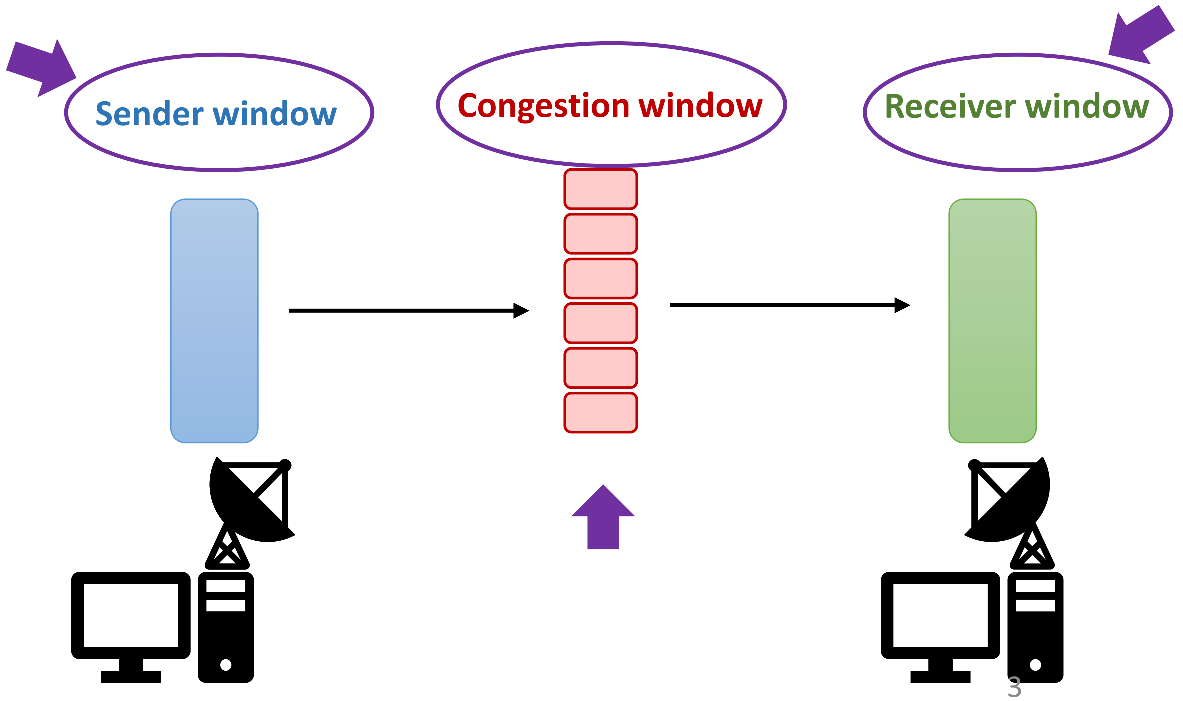




Introduction – TCP

- TCP sends one window of data each RTT
- Window starts small, doubles each RTT during slow start
- Window size limited by sender, receiver, congestion window
- TCP limits the size of the windows to the smallest of these three windows



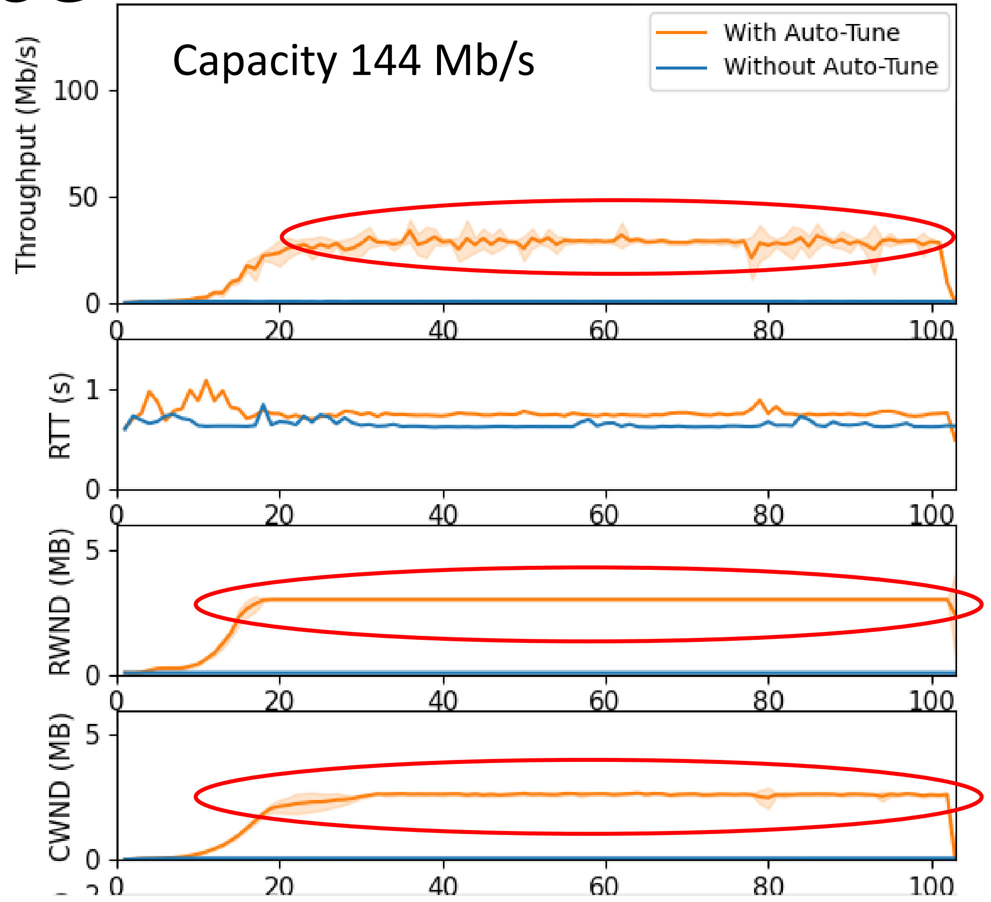


Buffer Sizes Limit Performance

Linux Defaults

- Auto-Tune enabled
- rmem = 4 KB, 128 KB, 6 MB
- wmem = 4 KB, 16 KB, 4 MB

Bitrate below link capacity, limited by Linux buffer setting



Benjamin Peters, Pinhan Zhao, Jae Won Chung, Mark Claypool. TCP HyStart Performance over a Satellite Network, In *Proceedings of 0x15 NetDev*, July 2021.

Buffer Sizes Limit Performance

Link	Delay (ms)	Capacity (Mb/s)
GEO	600	150
LEO	30	100
5G	10	3000
4G	20	200

Receiver buffer (tcp_rmem)

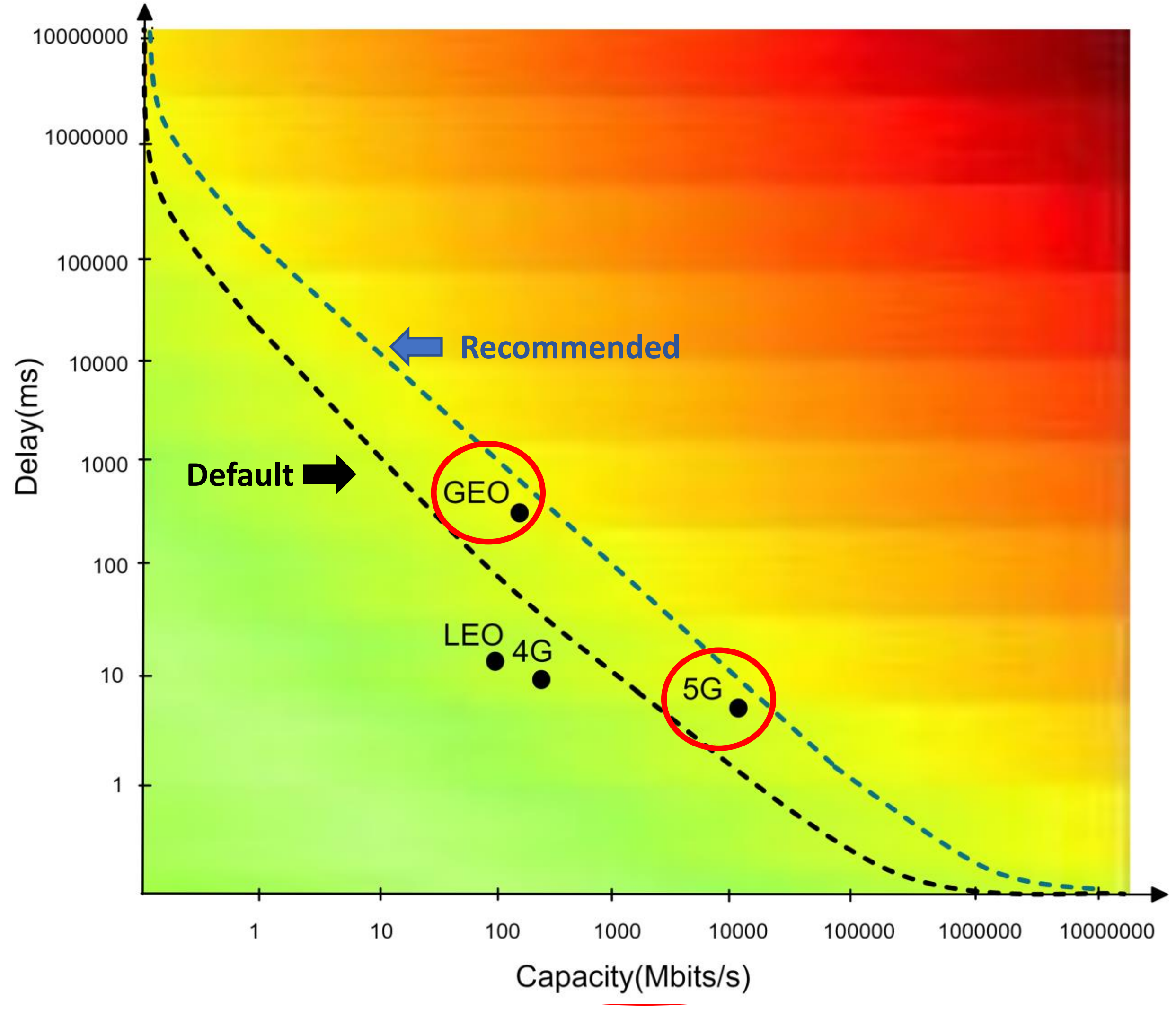
Default: `4096 131072` **6291456**

Recommended: `4096 131072` **26214400**

Sender buffer (tcp_wmem)

Default: `4096 16384` **4194304**

Recommended: `4096 16384` **26214400**

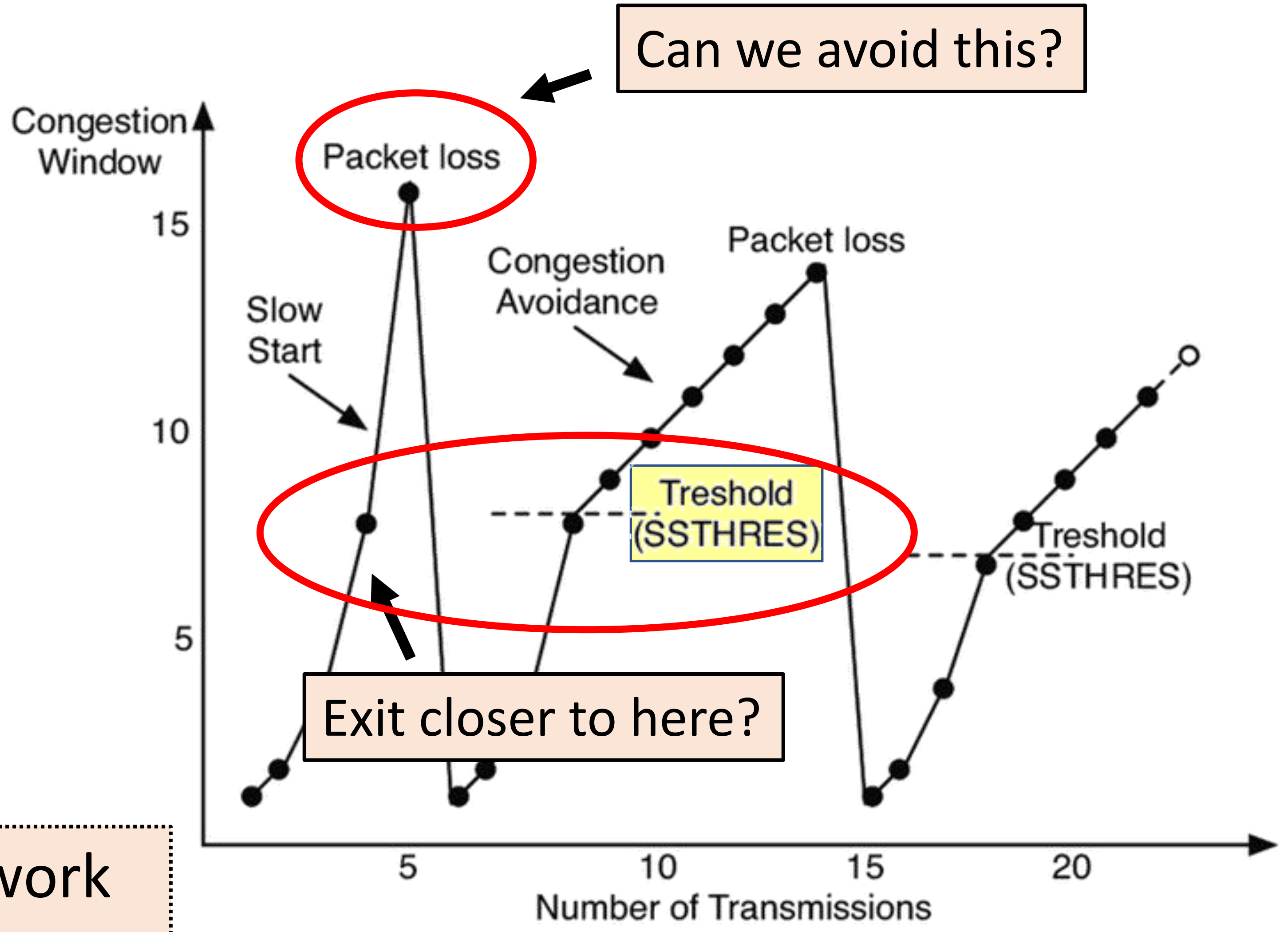


Benjamin Peters, Pinhan Zhao, Jae Won Chung, Mark Claypool. TCP HyStart Performance over a Satellite Network, In *Proceedings of 0x15 NetDev*, July 2021.

TCP Slow Start – Revisited

- Hystart is designed to **exit TCP slow** start *before* packet loss to avoid overshooting link throughput
- Hystart is on as Linux default
- When Hystart works well, it exits slow start before loss to avoid overshooting
- When Hystart does not work well, it exits slow start prematurely

So, how well does Hystart work for a Geo Satellite link?



Methodology

Viasat testbed

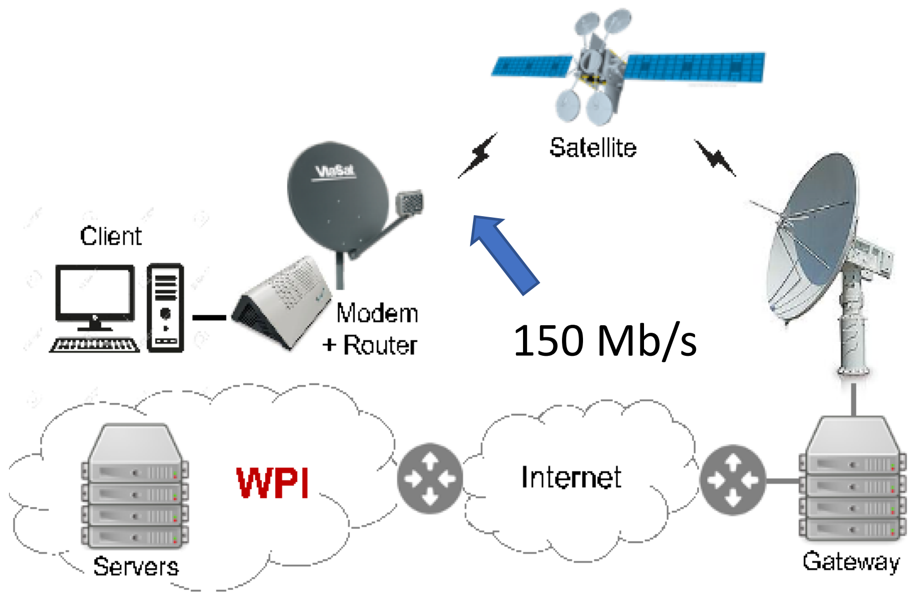
- High RTT
- Consistent capacity with single satellite
- Transient uplink scheduling may impact ACK timing

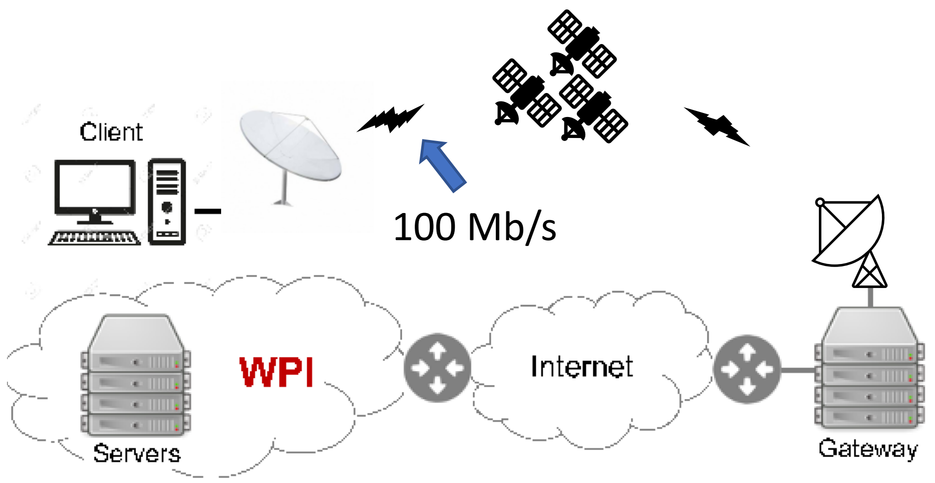
LEO testbed

- Variable Link Capacity
  - Sensitive to the weather
- Asymmetric Links
  - Downlink bitrate is higher than uplink bitrate
- Transient uplink scheduling and handover may impact ACK timing

Bulk downloads: Hystart On, Hystart Off

Measurements taken at the sender (e.g., throughput)

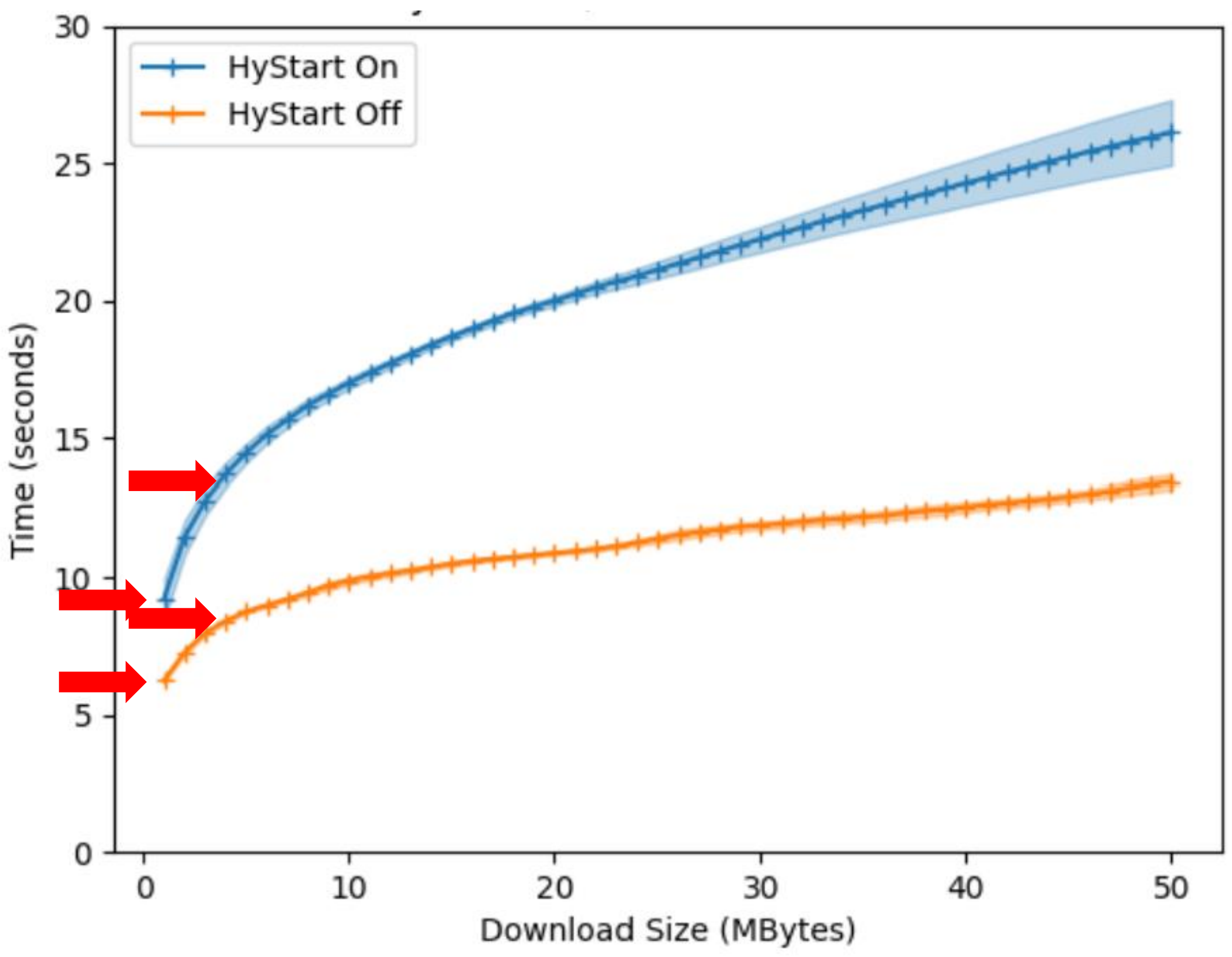




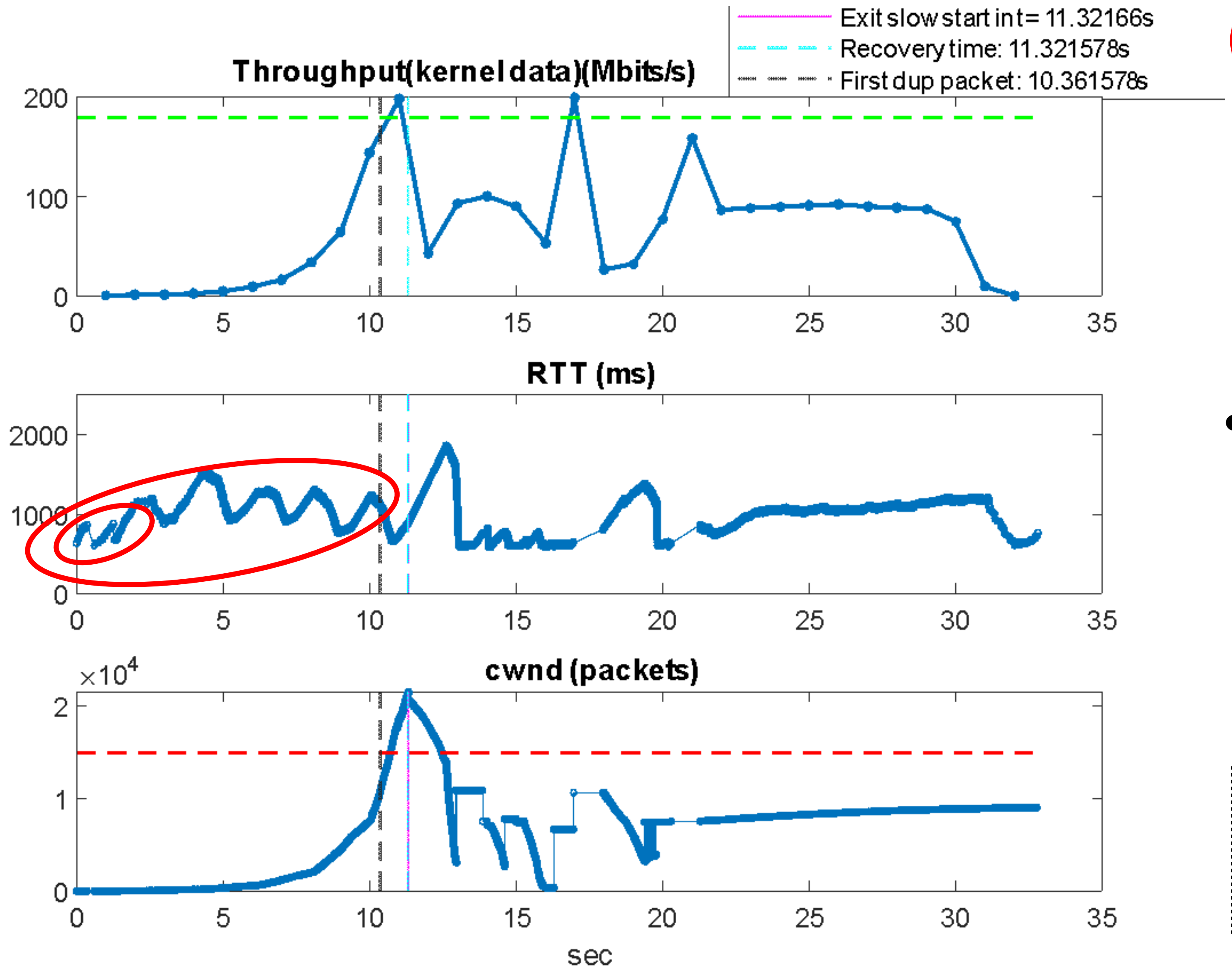
Download Times

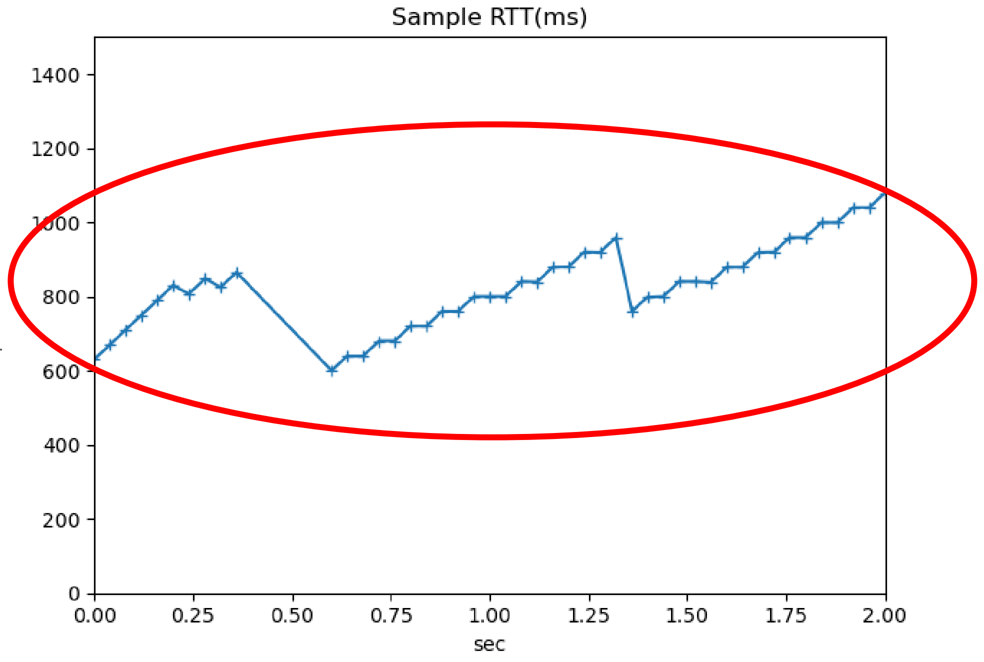
- 1 MB downloads take 50% longer with HyStart
- Average website (5 MB) takes 2x longer with HyStart

Why doesn't HyStart help for Geo Satellite link?



TCP Round-trip Times





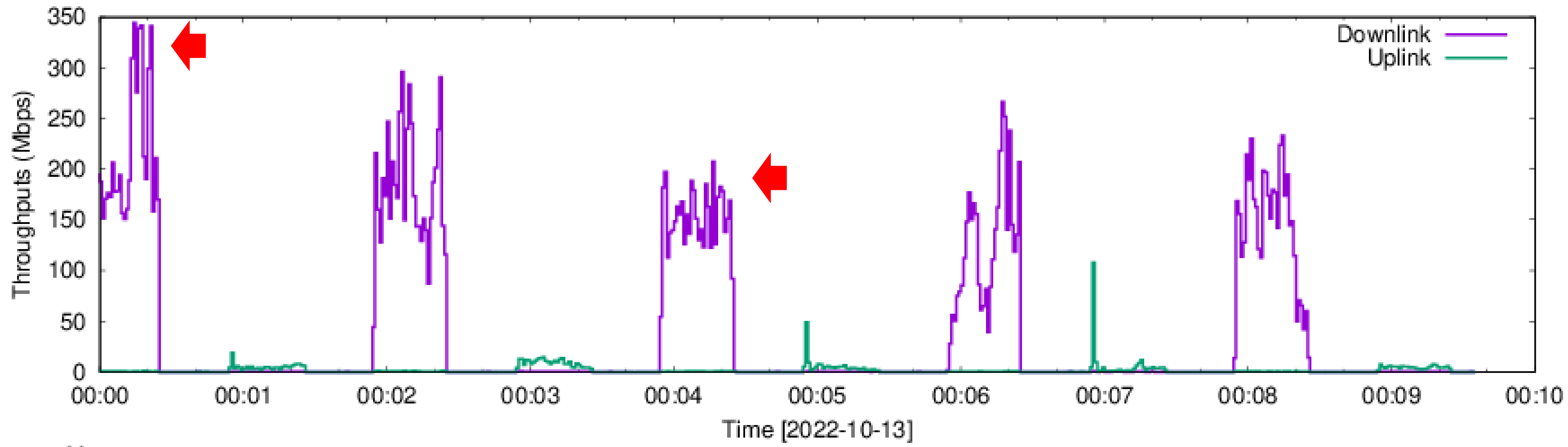
- RTT increase but download is *not* saturating link
  - TCP Acks need channel grant
  - Channel estimates adapt slower than TCP's doubling

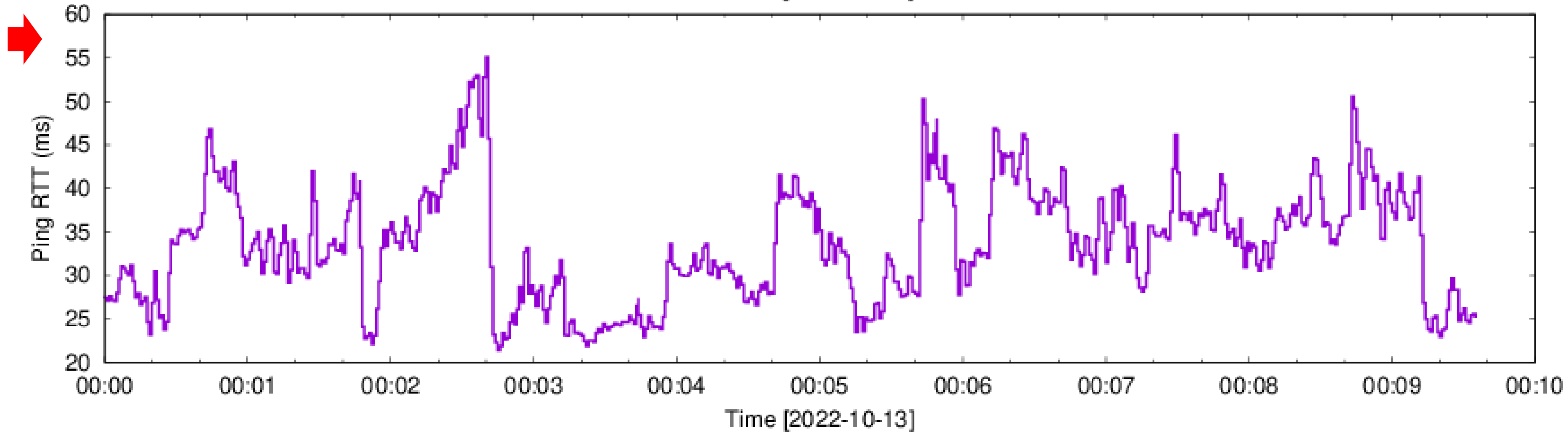
Finding exit condition by delay only difficult!

Example of download over LEO link

RTT varies a lot
Throughput varies a lot

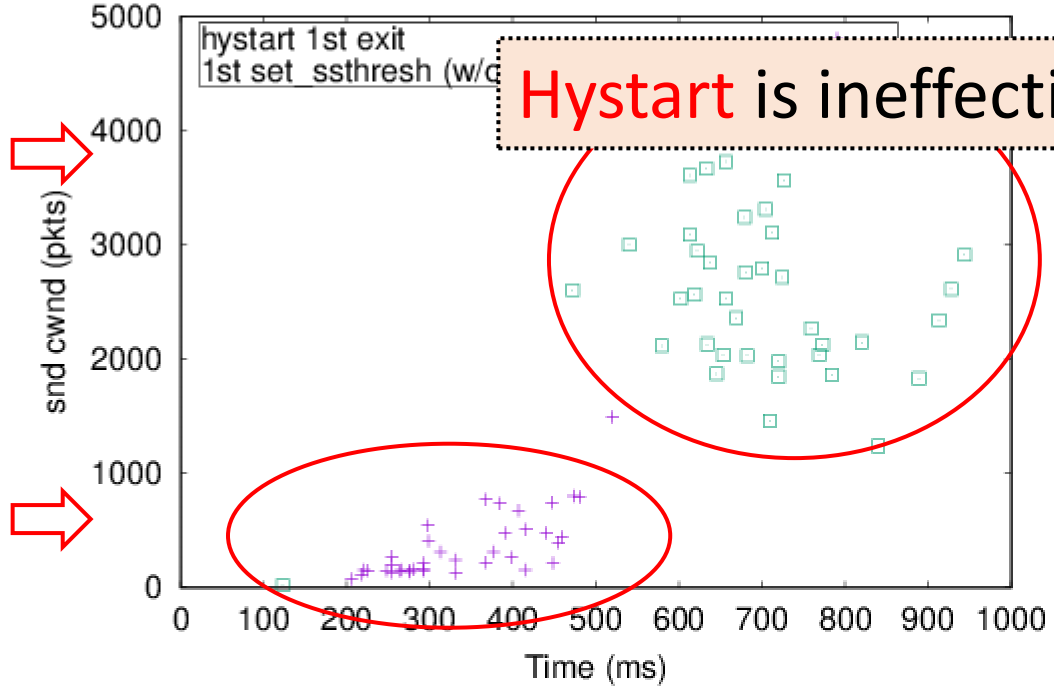
Delay still is a problem over LEO link!



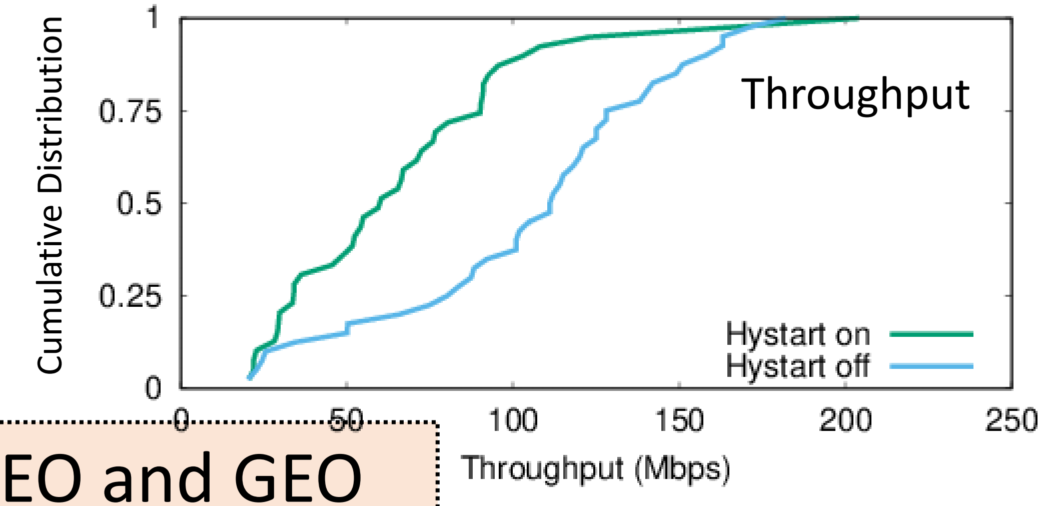


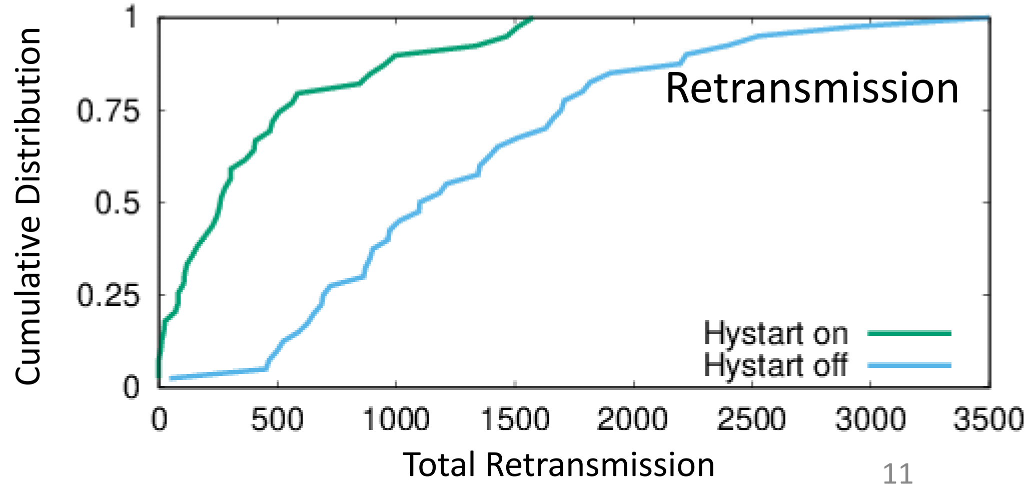
Hystart Exits too Early over LEO Link

- Hystart exits too early, when cwnd is still small.
- When Hystart off, the cwnd is much bigger when loss happens (first time call set_ssthresh()).



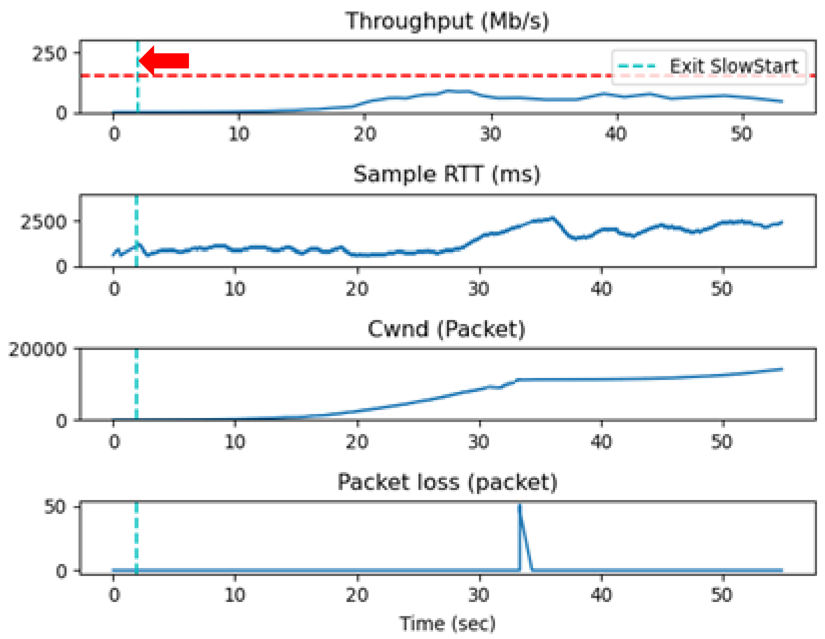
Hystart is ineffective for LEO and GEO





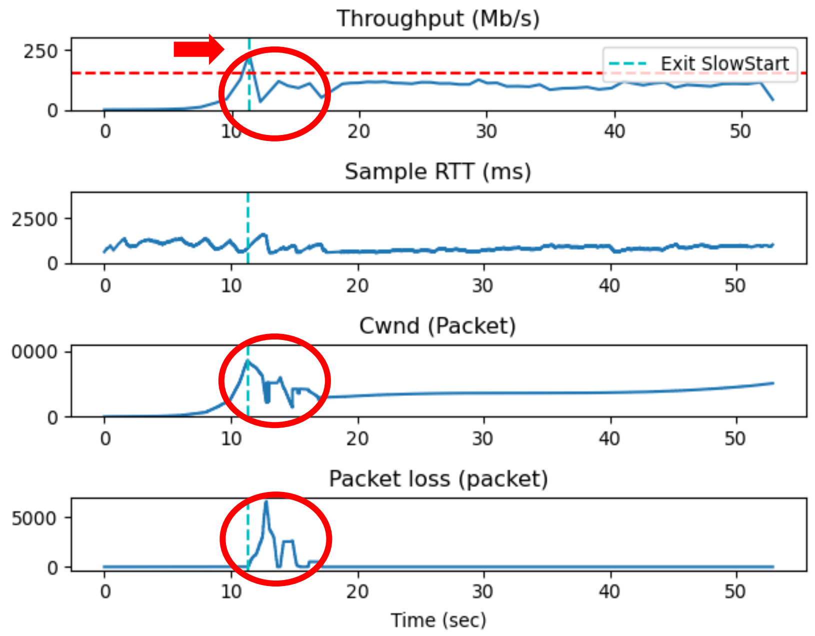
Results for different Exit point over GEO link

Hystart on



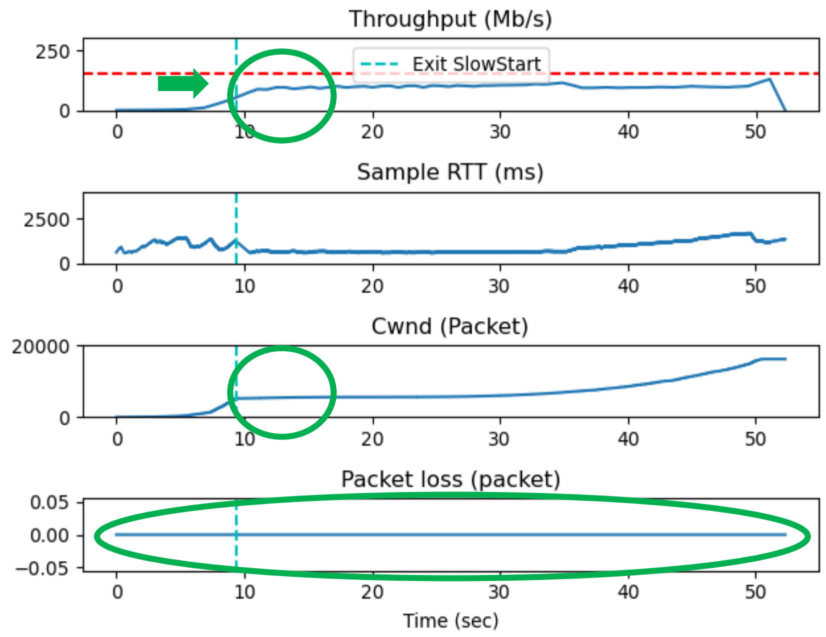
Exit too early

Hystart off



Exit too late

Exit in optimal point



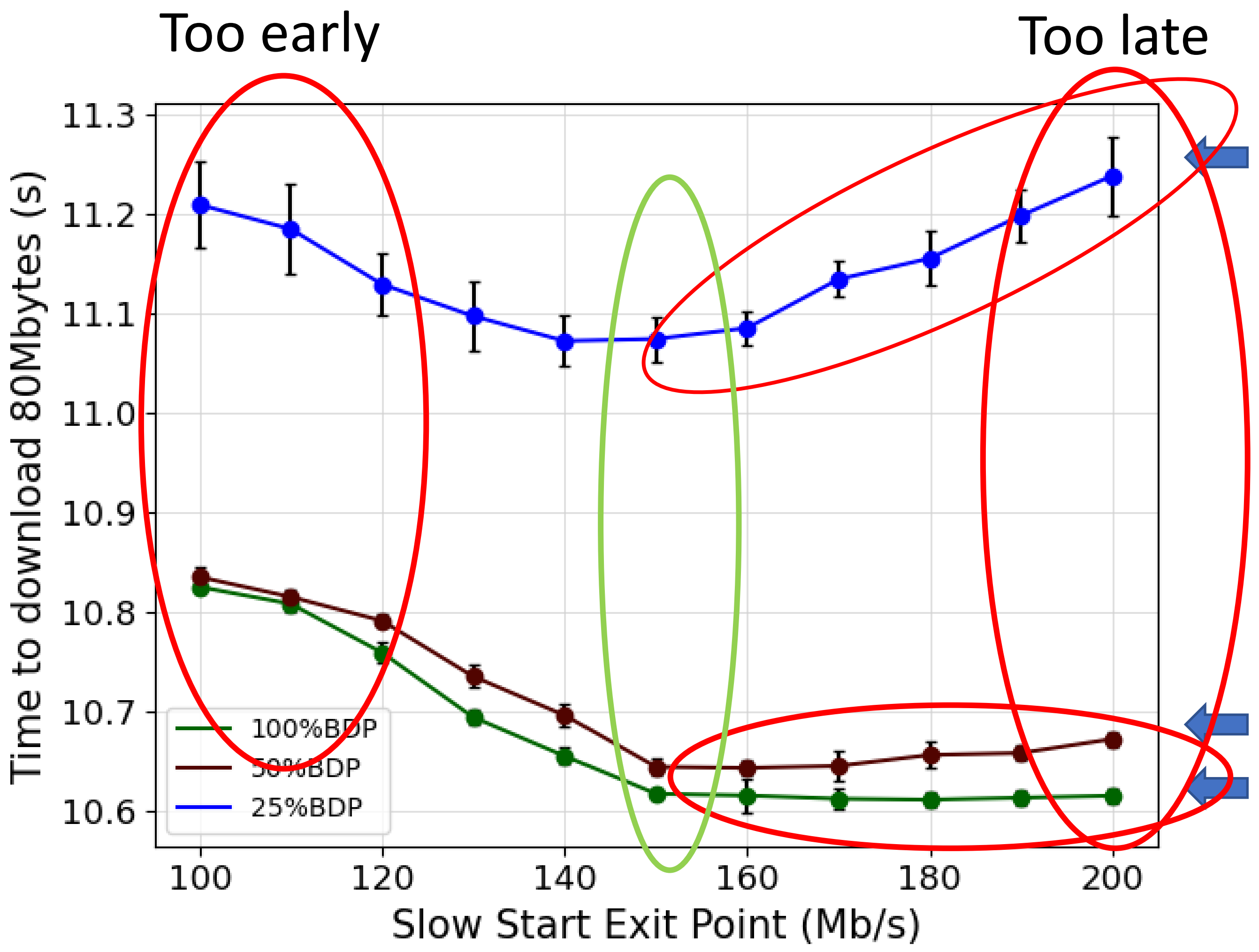
When should exit?

Need to find the optimal Exit point

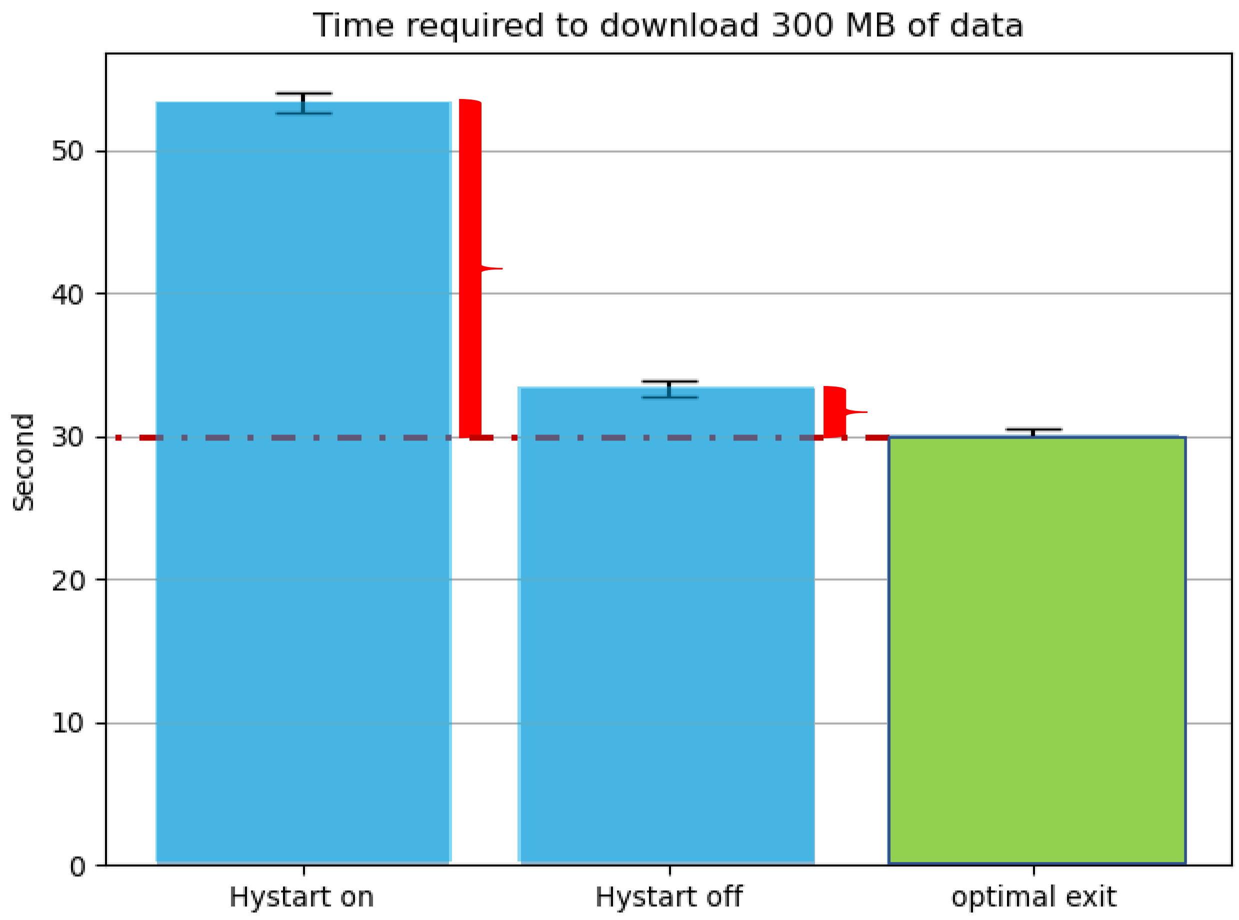
Optimal exit point with different queue sizes

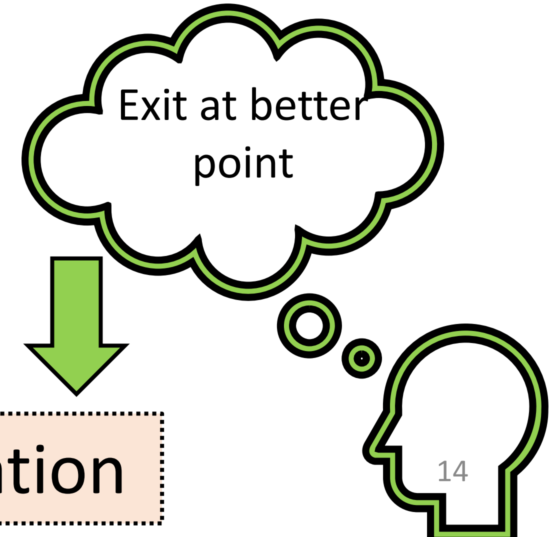
Size ratio (%)	Queue size (Mb)
100	11
50	5.5
25	2.75

Small router queues make exit point decision even more important



Example of download over GEO link

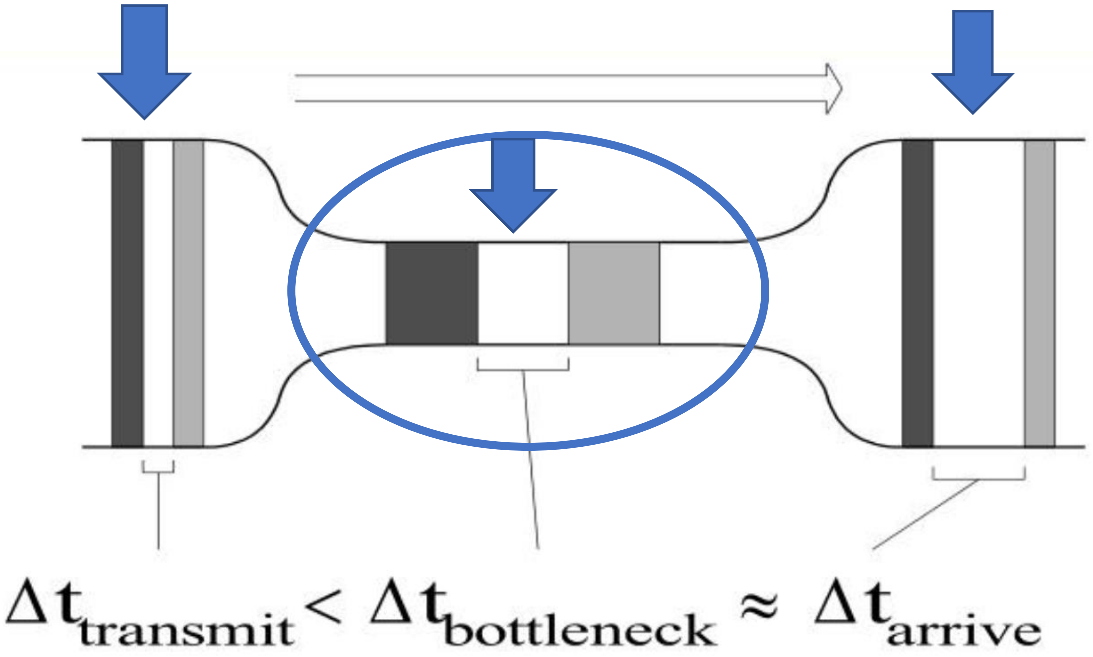




Deciding on exit point based on bandwidth estimation

Approach

Use packet-pairs to estimate bandwidth

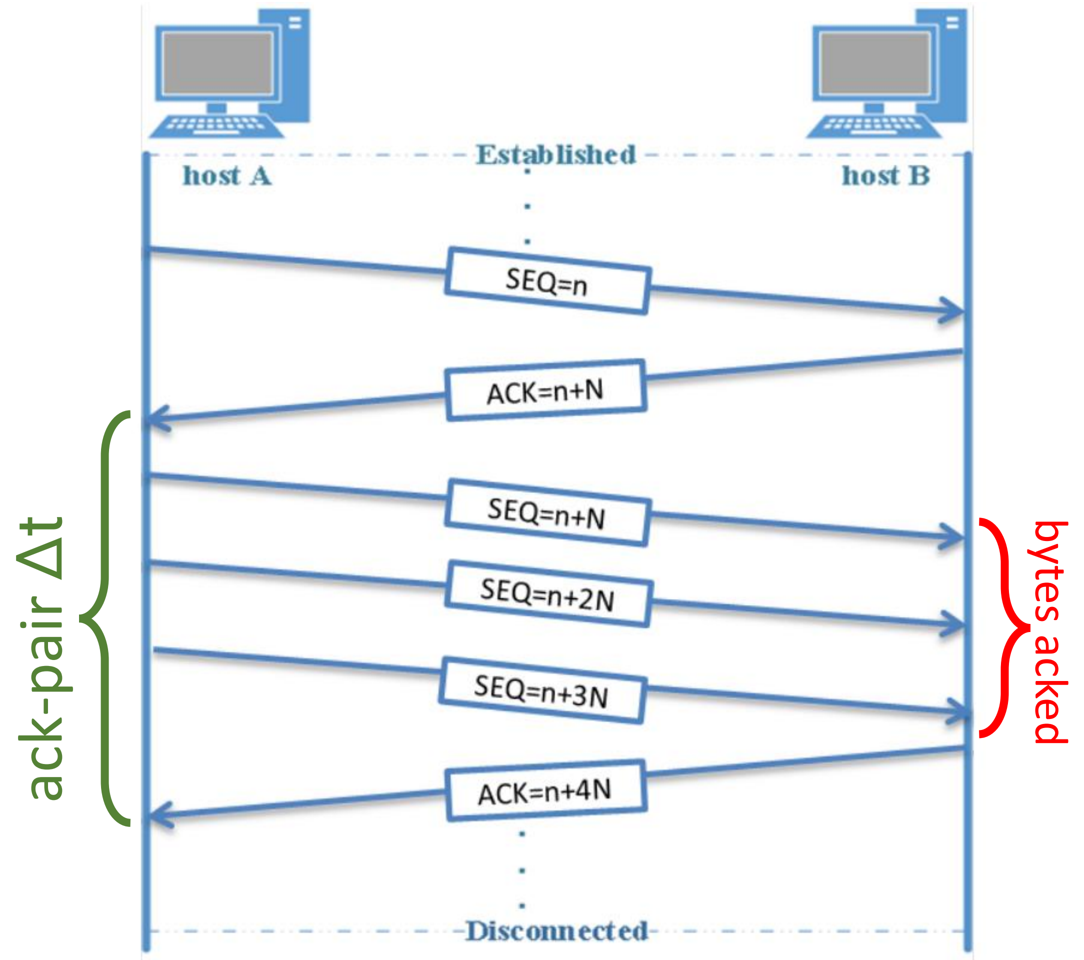


$\Delta t_{transmit} < \Delta t_{bottleneck} \approx \Delta t_{arrive}$

bandwidth = packet size / Δt

How to do at server?

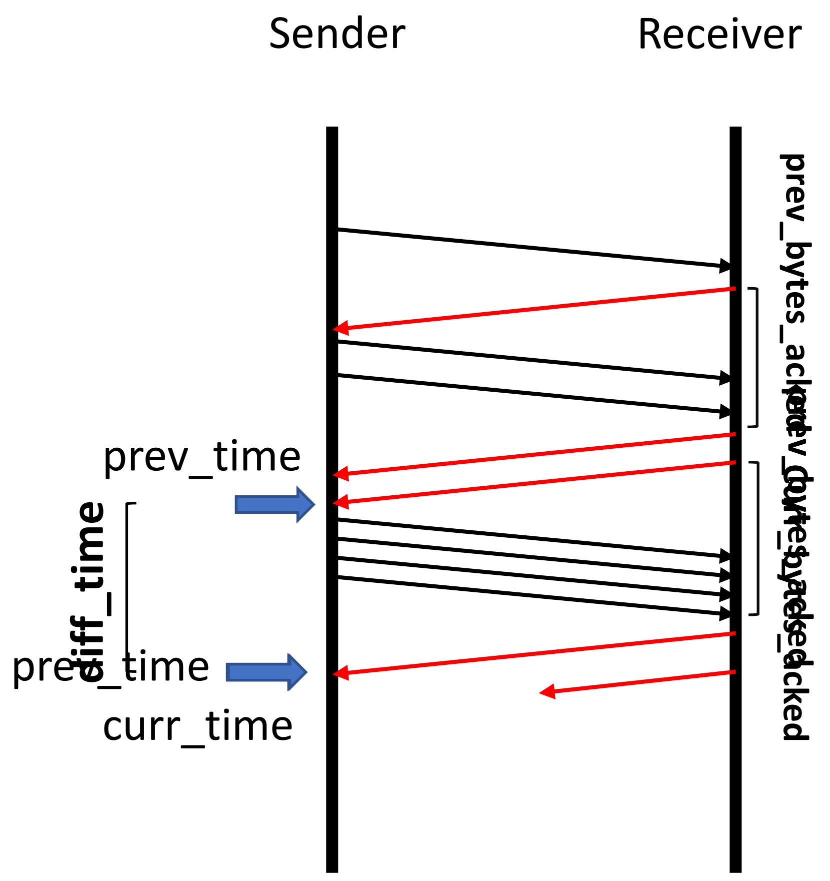
→ Use ack-pairs



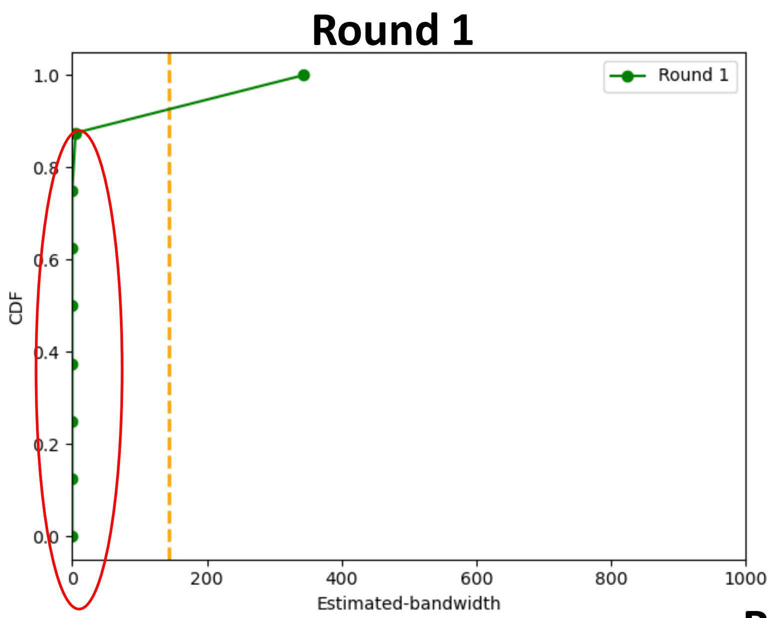
bandwidth = bytes acked / ack-pair Δt

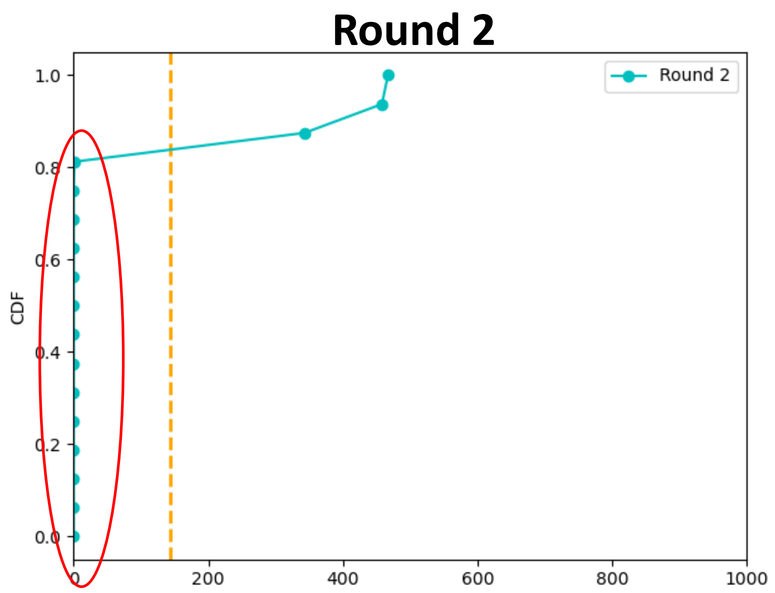
bictcp_acked_function()

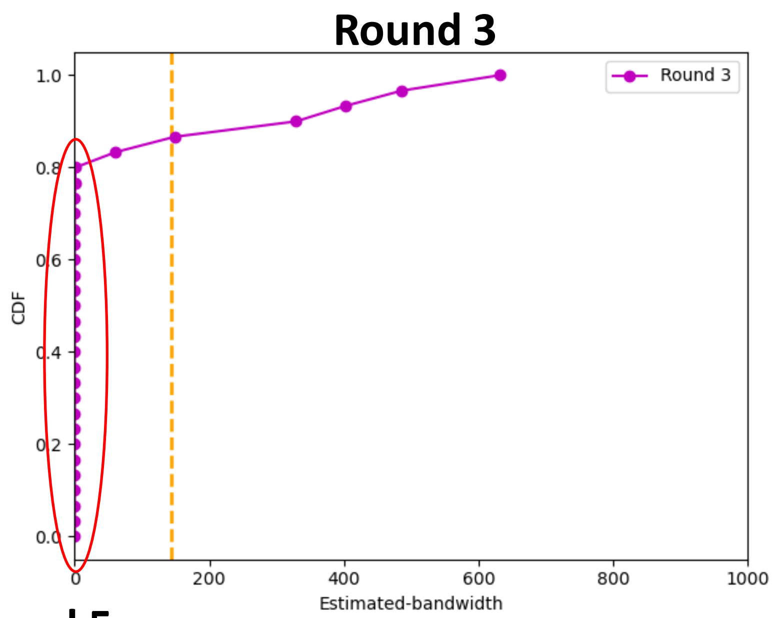
```
/*this function call for every ack*/

curr_time = now
diff_time = curr_time – prev_time

curr_bytes_acked = tp->bytes_acked
diff_bytes_acked = curr_bytes_acked – prev_bytes_acked

bandwidth_estimate =  diff_bytes_acked / diff_time

prev_bytes_acked = curr_bytes_acked
prev_time = curr_time
```

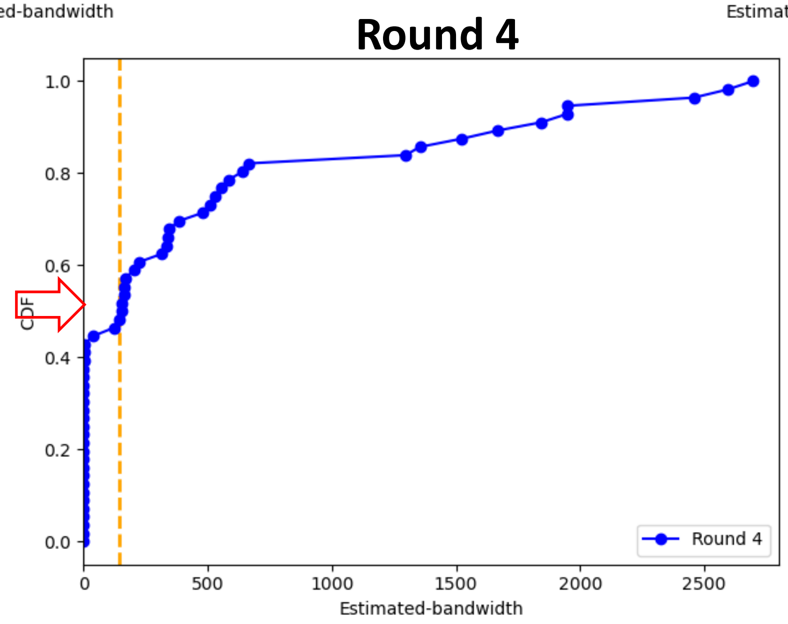


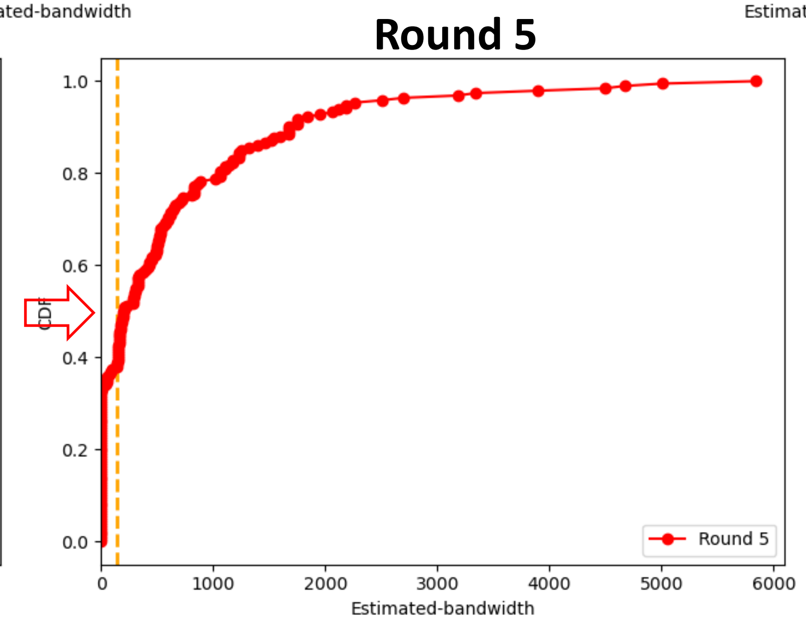
Bandwidth Estimates over Geo Sat Link







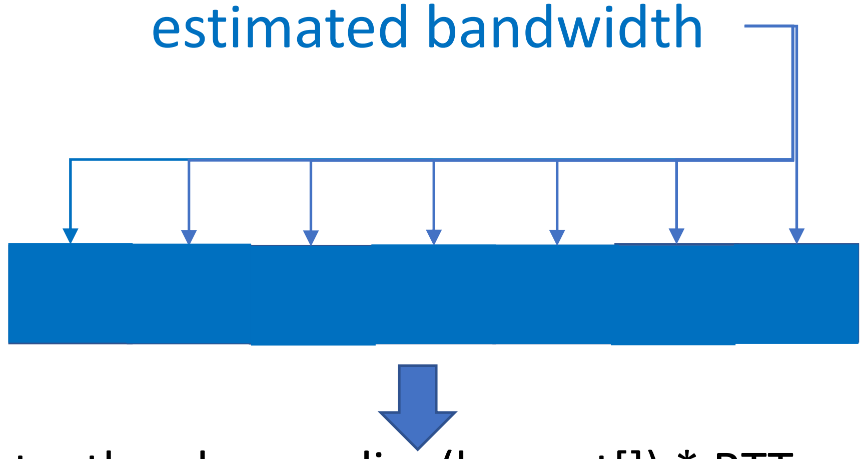




bictcp_acked_function()

```
insert bandwidth_estimate to bw_est [] array
when RTT round ends:
        median_est = median of bw_est[]
        if median_est > 0:
                ssthresh = median_est x RTT
        else
                clear bw_est []   //for the next RTT
```

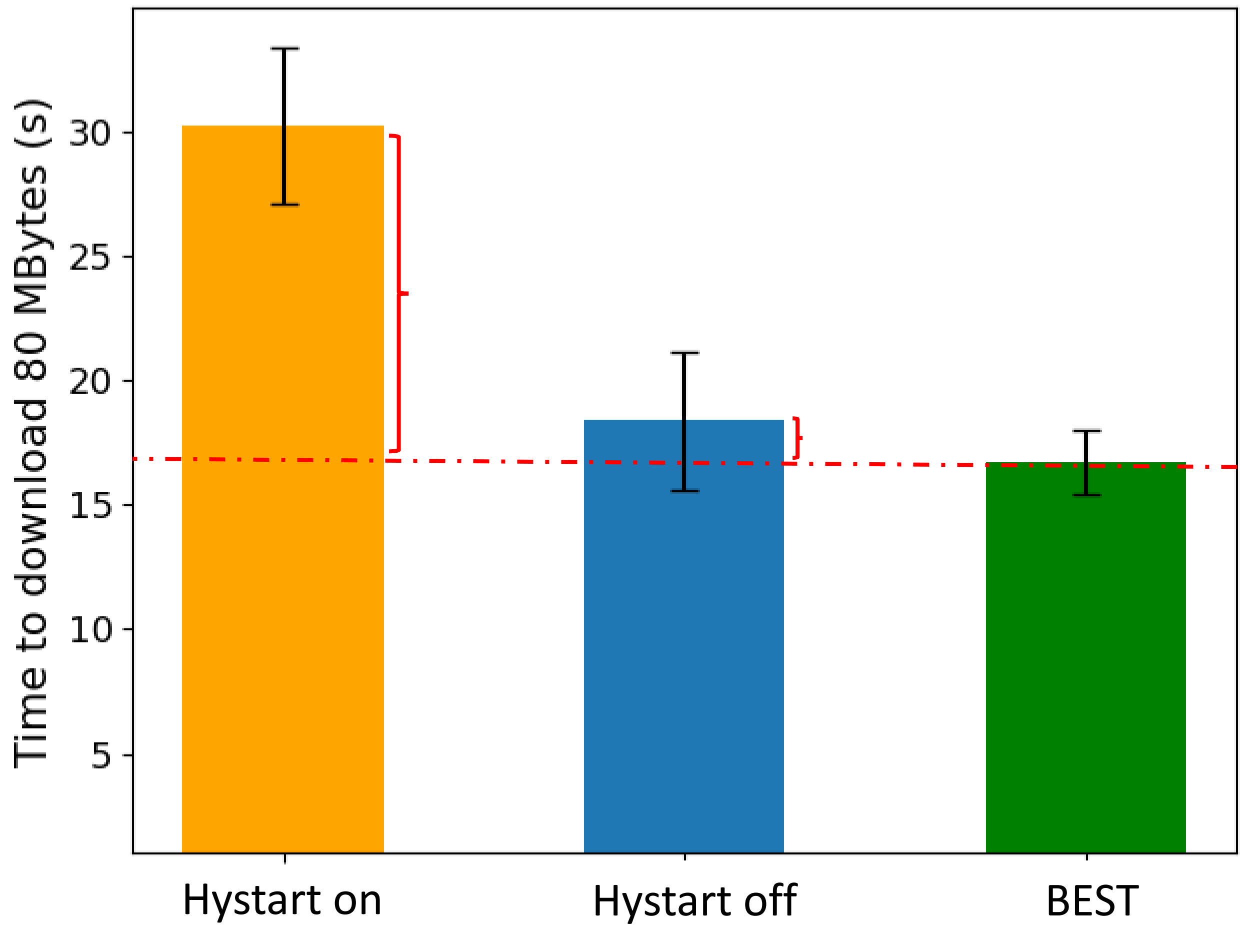
Bandwidth Estimated Slow StarT (BEST)



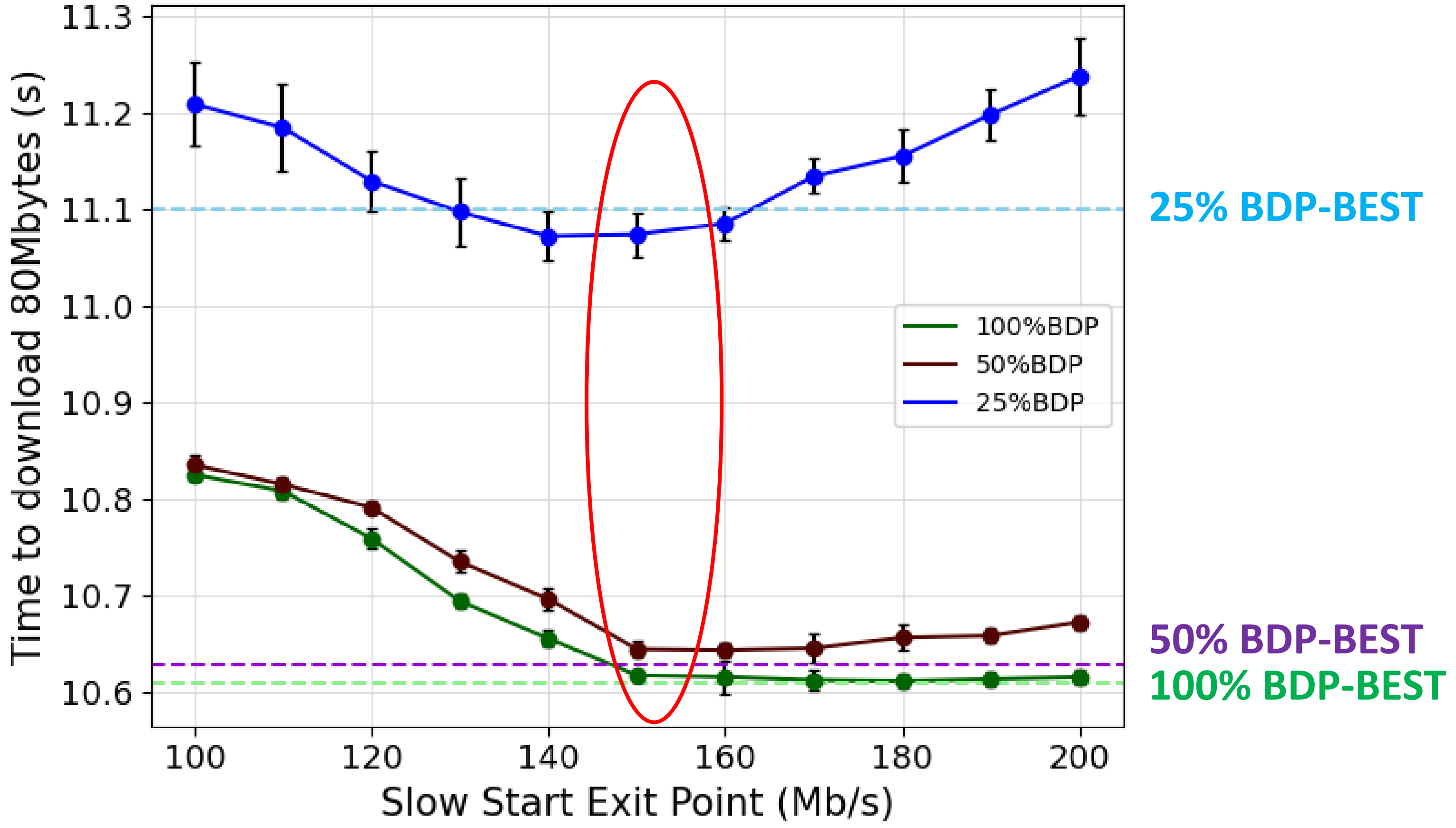
set ssthresh = median(bw_est[]) * RTT

Results over Geo link

- Hystart on : Exit prematurely
  - High download time
- Hystart off : Exit too late
  - Lower download time
- BEST : Exit at the better point
  - Lowest download time



Results over Geo link



BEST algorithm is near the optimal Exit point for different queue sizes

Conclusion

Slow fat links need larger sender and receiver buffer sizes in Linux than default

- GEO links and 5G links under utilized

Slow fat links challenging since getting right TCP window size critical

- Exit too early: under utilization, Exit too late: packet loss
- TCP Hystart exits too early for LEO link and GEO link

BEST uses packet pair bandwidth estimation to set ssthresh

- Performs better than Hystart over GEO link

GitHub link: https://github.com/maryam-ataei/tcp_bw

Ongoing Work

Accommodate LEO link characteristic

Parameters for current heuristic

- Number rounds, percentage of distribution

Median from limited space

- Memory optimization
- Running median vs. “true” median

Filtering out extremely low or extremely high estimations

- Identifying and removing extreme values before making an estimate
- May need to have special consideration for data centers

Evaluation in more networks

- More network + system configuration

Thank-you for your attention!

Fixing TCP Slow Start for Slow Fat Links

Maryam Ataei Kachooei
Pinhan Zhao
Mark Claypool

WPI

Feng Li
Jae Chung

Viasat

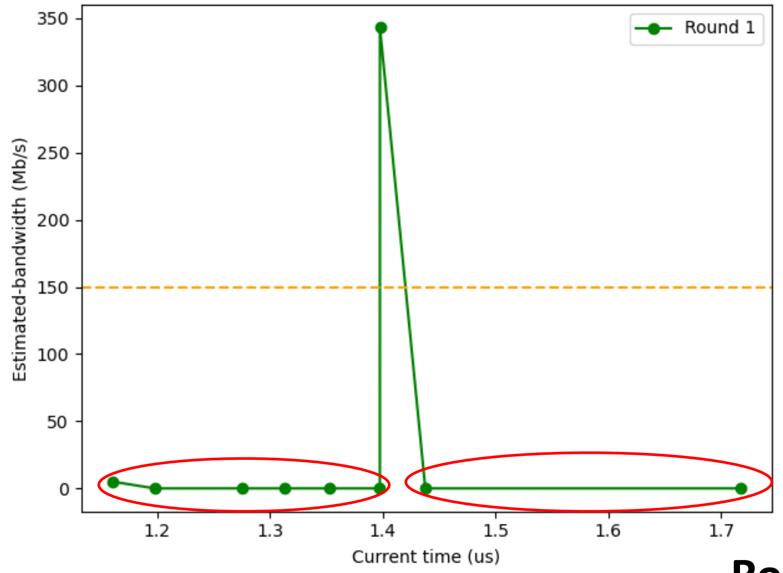
Appendix

Algorithm

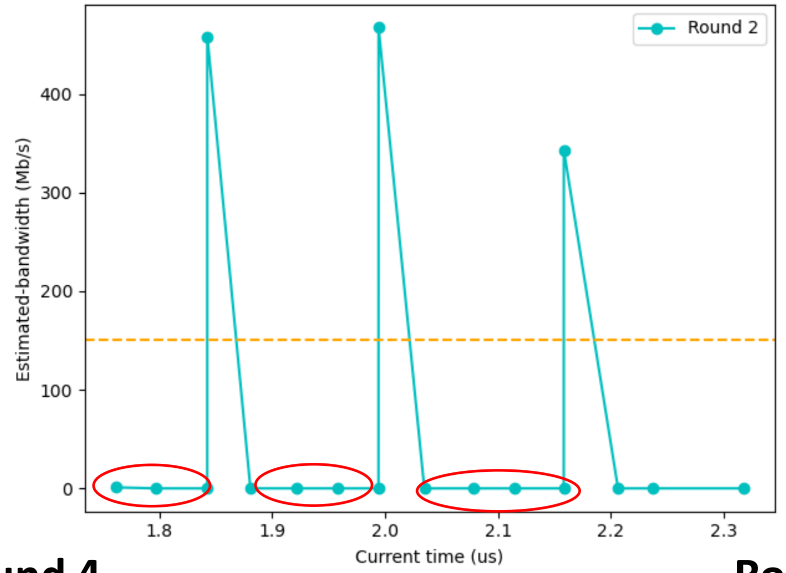
```
Every ack:
    bw_est = bytes_acked / $\Delta t$
    insert (bw_est, bw_est_array)

At end of an RTT round:
    bw_est = median (bw_est_array)
    if median is greater than 0
        set ssthresh = bw_est x RTT
    else
        clear bw_est_array
```

Bandwidth Estimates over Geo Sat Link

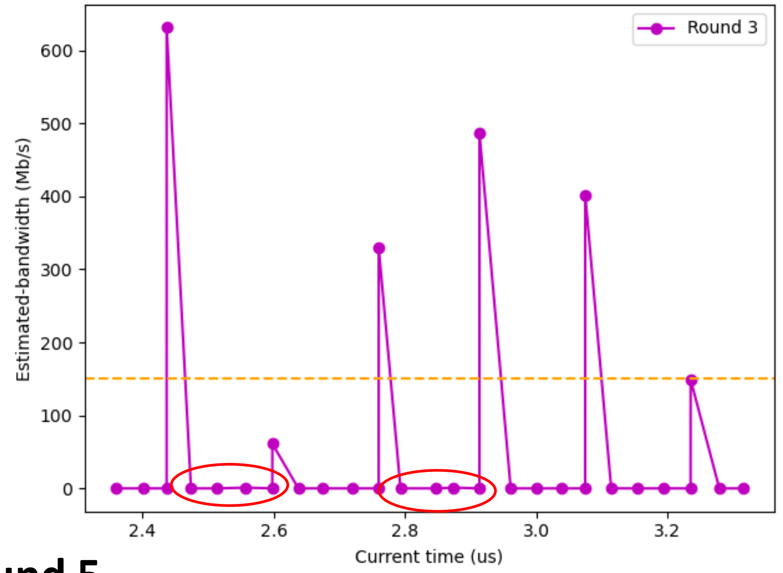
Round 1



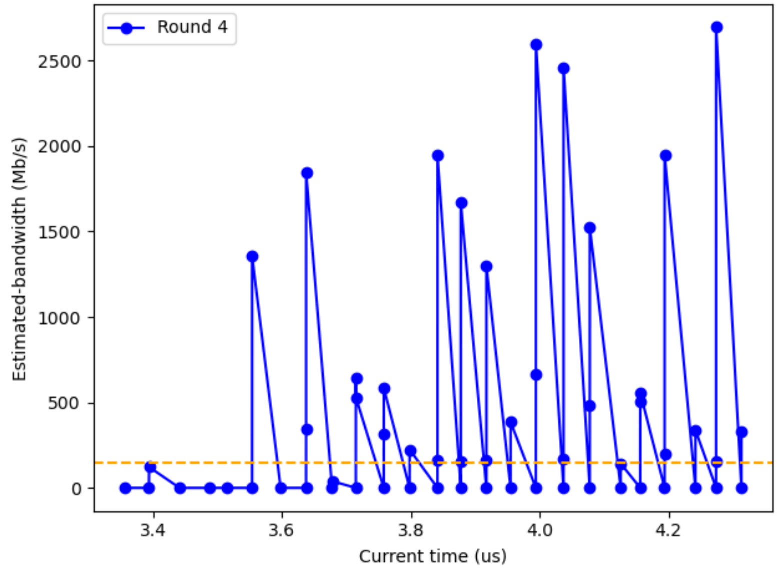
Round 2



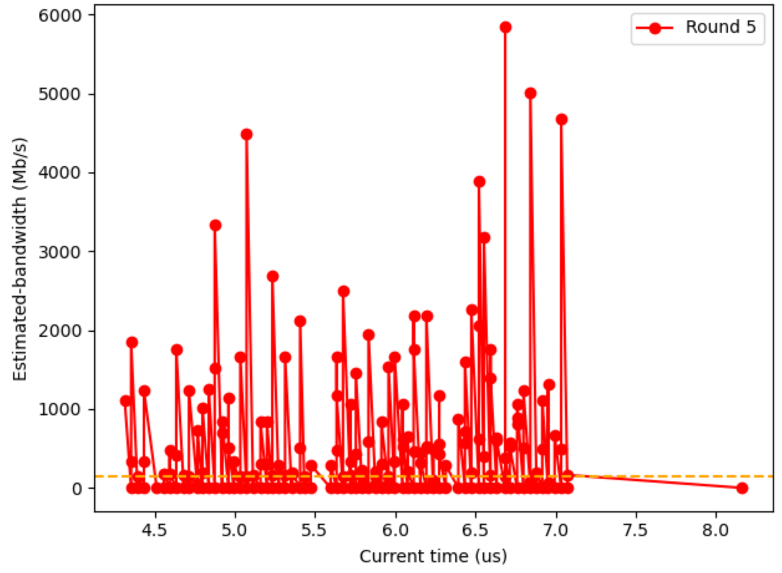
Round 3



Round 4



Round 5


Round 5
Estimated-bandwidth (Mb/s)
Current time (us)