TCP HyStart Performance over a Satellite Network

BENJAMIN PETERS, PINHAN ZHAO, JAE CHUNG, MARK CLAYPOOL

WPI

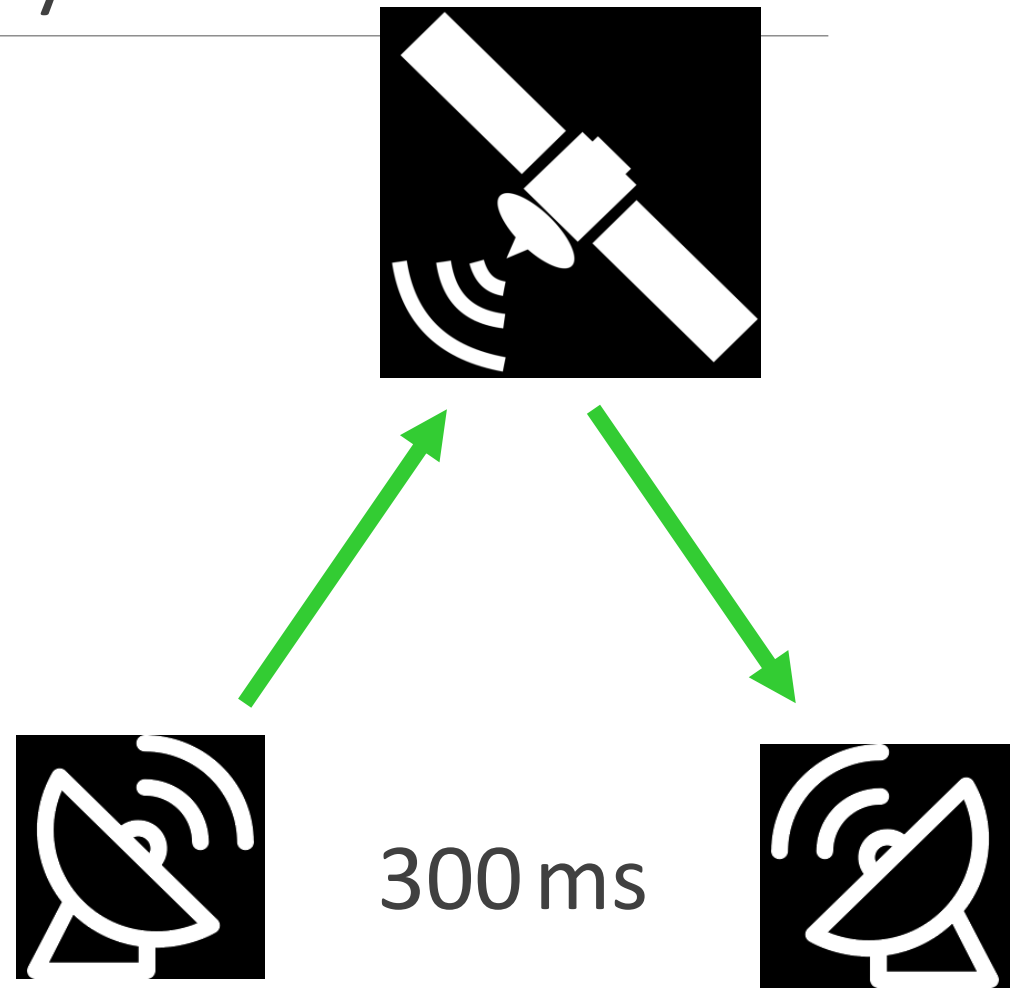
Viasat™

Satellites Coverage is Everywhere

Satellites provide global networking

- Remote locations
- On airplanes
- During emergencies
- Increased bandwidth (up to 150 Mb/s)

High latency (~300ms one-way)



Performance Issues Over Satellites

Sender and receiver buffer sizes can affect performance

HyStart can exit slow start before loss to avoid overshooting

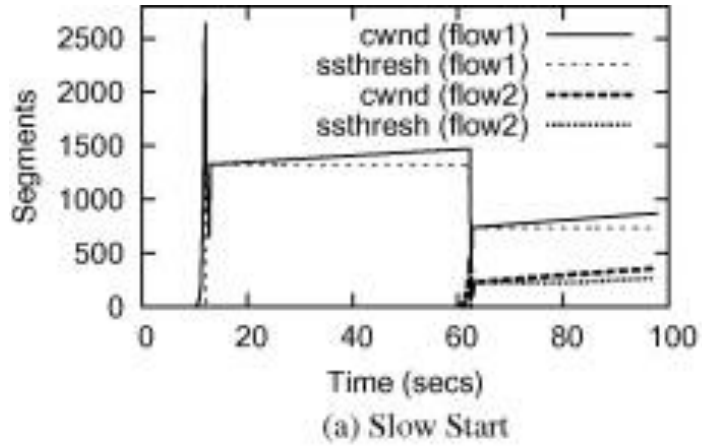
HyStart may exit slow start prematurely

Performance Enhancing Proxies (PEPs) are becoming increasingly ineffective

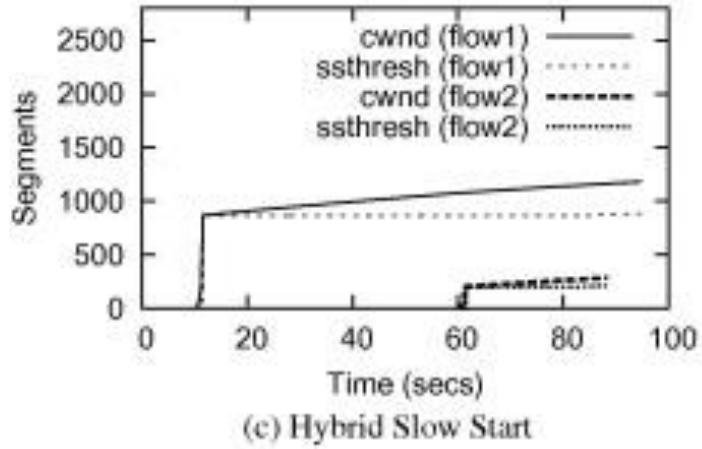
- VPNs
- QUIC

Few published studies over actual satellite networks

- Most use simulation or emulation



(a) Slow Start



(c) Hybrid Slow Start

Outline

Introduction (Done)

Background (Done)

Methodology (Up Next)

Buffer Settings Results

Hystart Results

Conclusion

Methodology

ViaSat link to represent a client with a “last mile” satellite connection

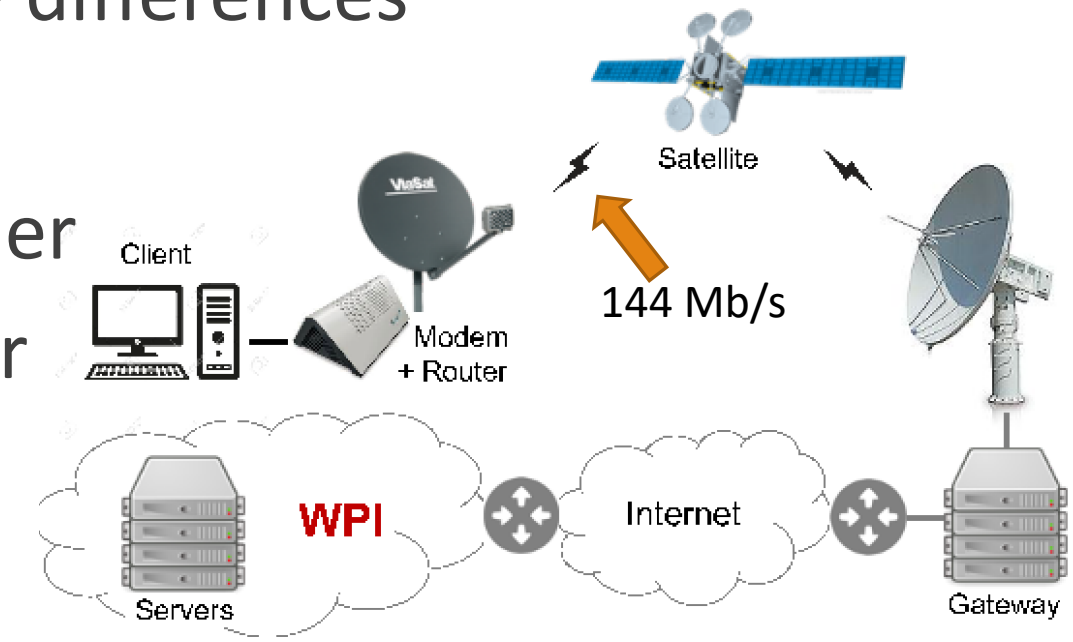
Tests were run serially to minimize differences

1Gbyte bulk downloads

5 or 10 runs were averaged together

Measurements taken at the sender

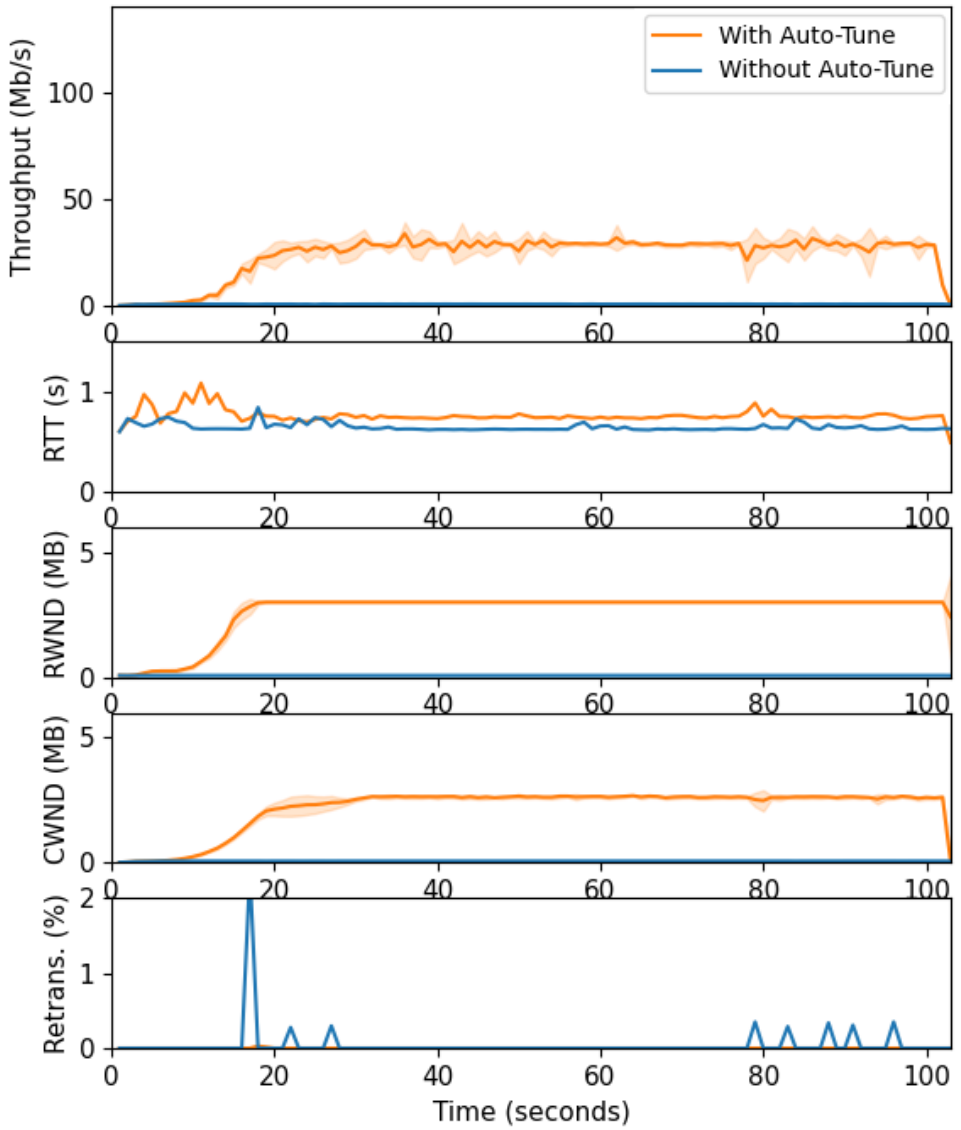
BDP: 18 MB/s * 600 ms = 10.8 MB



Default Settings

- Linux Defaults
  - HyStart enabled
  - Auto-Tune enabled
  - rmem = 4KiB, 128KiB, 6MiB
  - wmem = 4KiB, 16KiB, 4MiB

Message: Utilization below link capacity

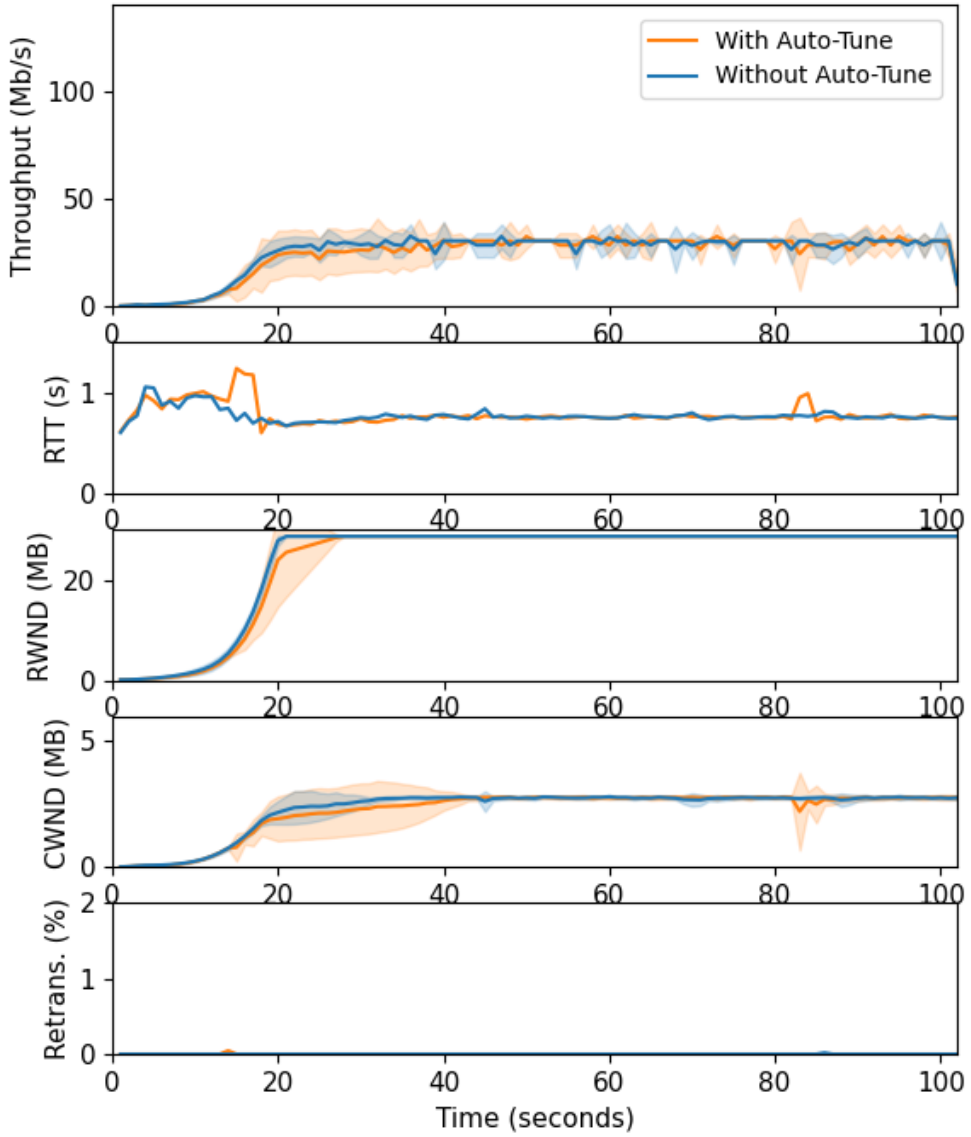


Increasing RMEM

- Run settings:
  - HyStart enabled
  - rmem = 60MB, 60MB, 60MB
  - wmem = 4KiB, 16KiB, 4MiB

Messages:

- Increasing receiver buffer means some improvement without auto-tune
- Still well below expected throughput
- Similar throughput is achieved with and without auto-tune

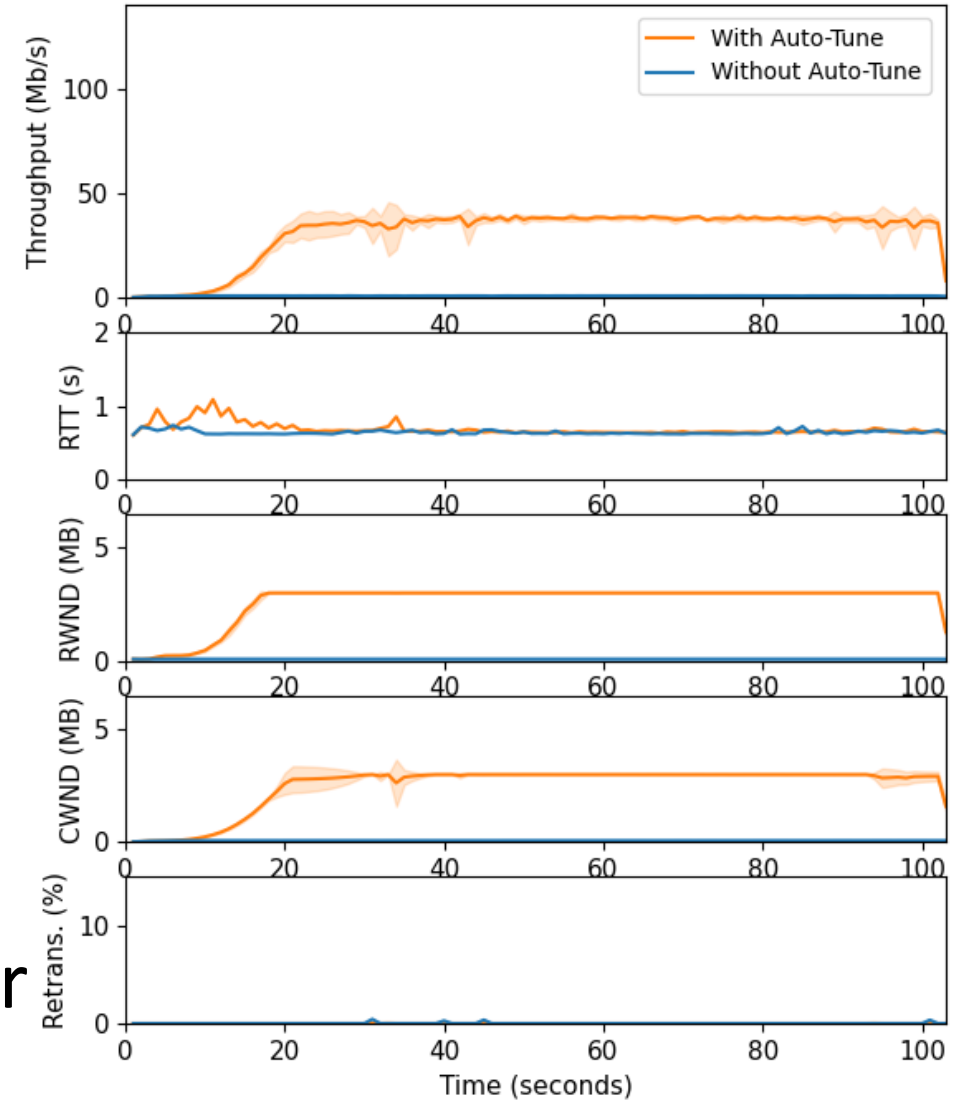


Increasing WMEM

- Run Settings:
  - HyStart enabled
  - rmem = 4KiB, 128KiB, 6MiB
  - wmem = 60MB, 60MB, 60MB

Messages:

- Increasing sender buffer improves performance for auto-tune streams
- Throughput is still limited by receiver buffer sizes

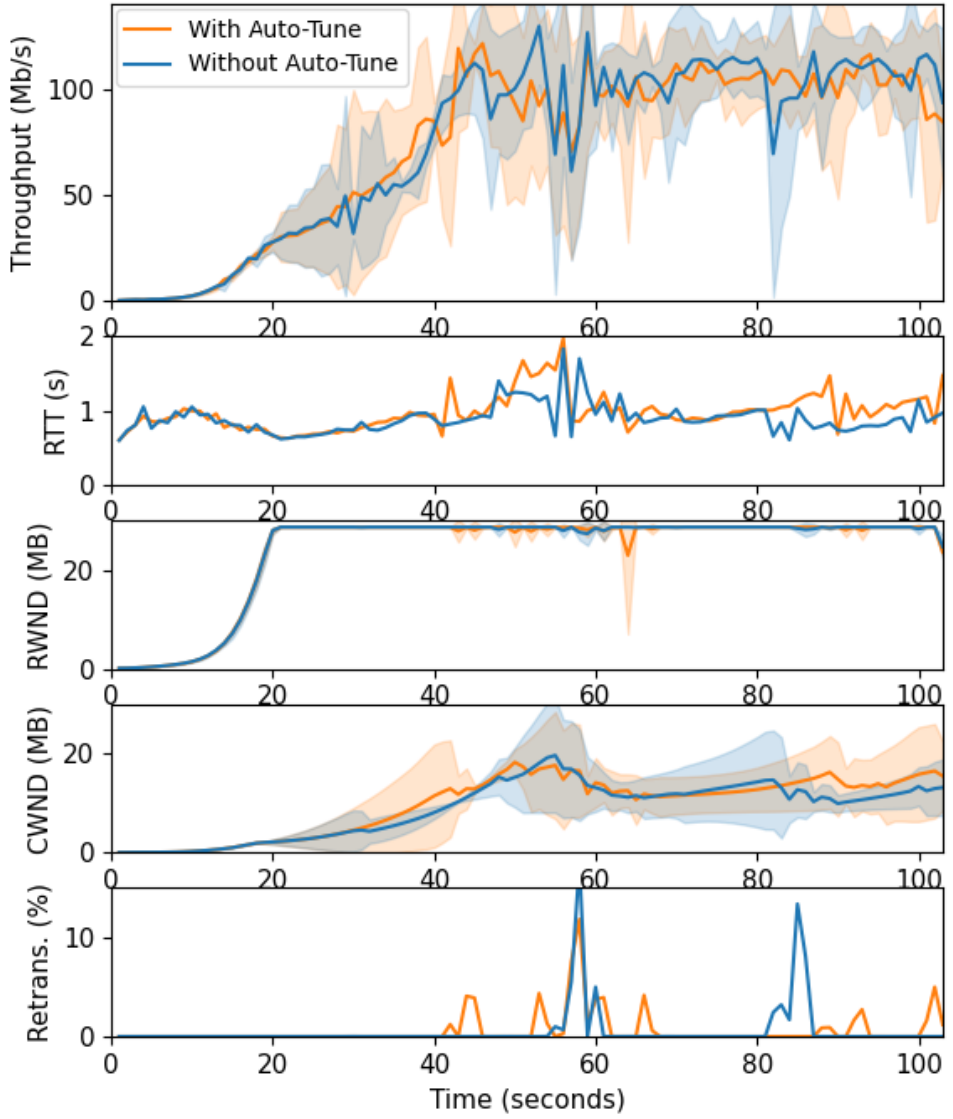


Increasing All Buffers

- Run Settings:
  - HyStart enabled
  - rmem = 60MB, 60MB, 60MB
  - wmem = 60MB, 60MB, 60MB

Messages:

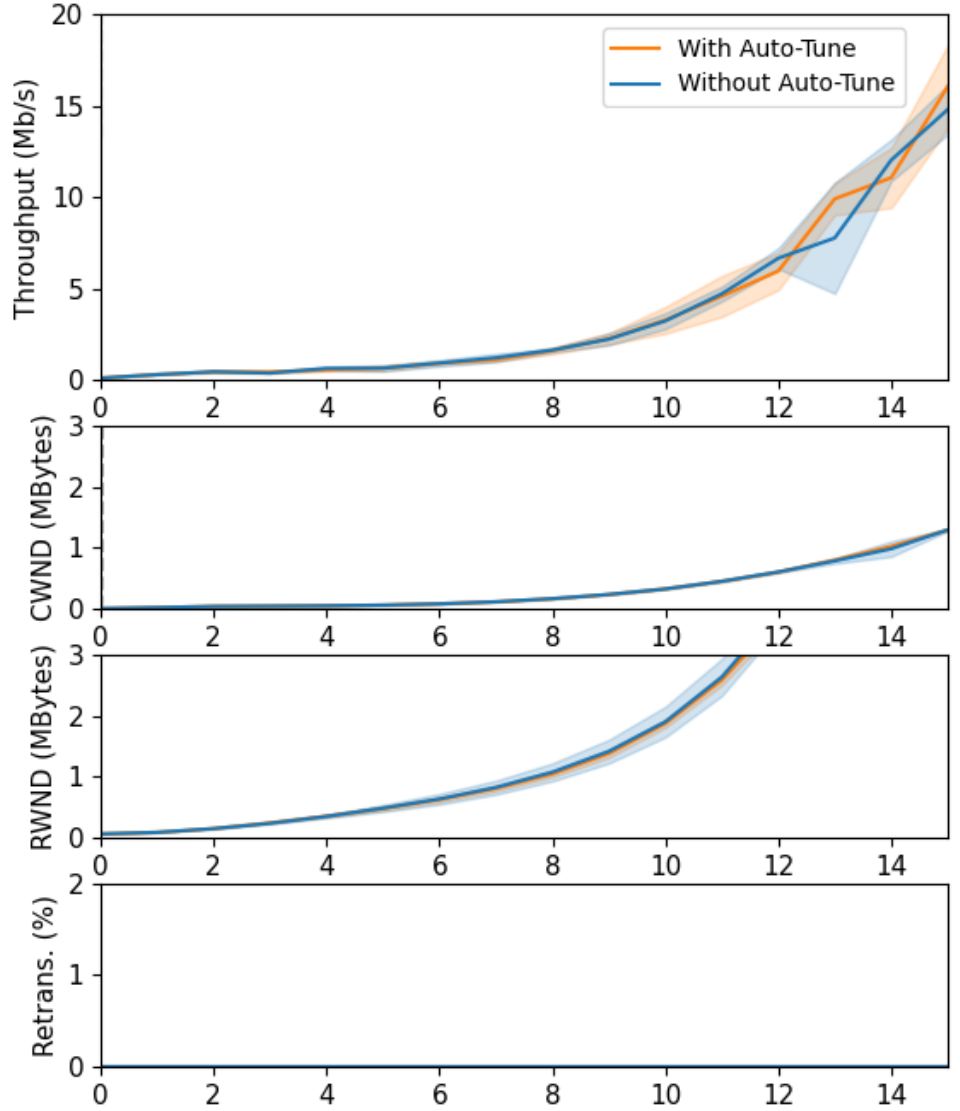
- Increasing all buffers results in full utilization
- Runs are visually similar with and without auto-tune



Startup Performance

- First 15 seconds of previous graph
- Run Settings:
  - HyStart enabled
  - rmem = 60MB, 60MB, 60MB
  - wmem = 60MB, 60MB, 60MB

Message: Visually no difference between runs with and without auto-tune



Buffer Setting Recommendations

Auto-tune (tcp_moderate_rcvbuf)
- Default: **Enabled**
- Recommended: **Enabled**

Receiver buffer (tcp_rmem)
- Default: 4096 131072 **6291456**
- Recommended: 4096 131072 **26214400**

Sender buffer (tcp_wmem)
- Default: 4096 16384 **4194304**
- Recommended: 4096 16384 **26214400**

TCP only advertises up to half of the buffer size

Outline

Introduction (Done)

Background (Done)

Methodology (Done)

Buffer Settings Results (Done)

Hystart Results(Up Next)

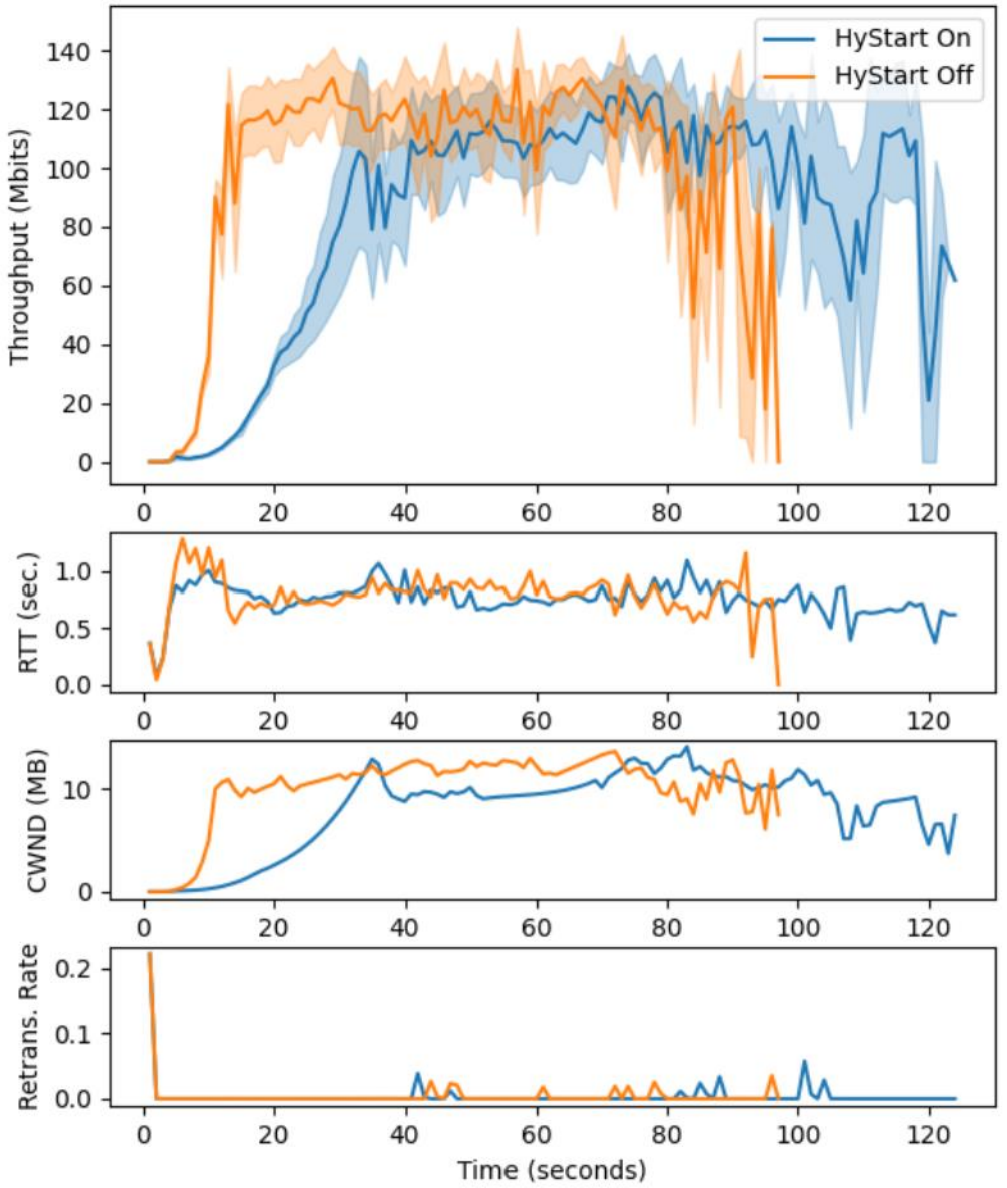
Conclusion

Comparing HyStart On/OFF

- Run Settings:
  - Auto-tune enabled
  - rmem = 60MB, 60MB, 60MB
  - wmem = 60MB, 60MB, 60MB
  - 1G download

Messages:

- HyStart off reaches steady state about 20 seconds earlier than HyStart on
- HyStart off finishes the download about 25 seconds faster than HyStart on

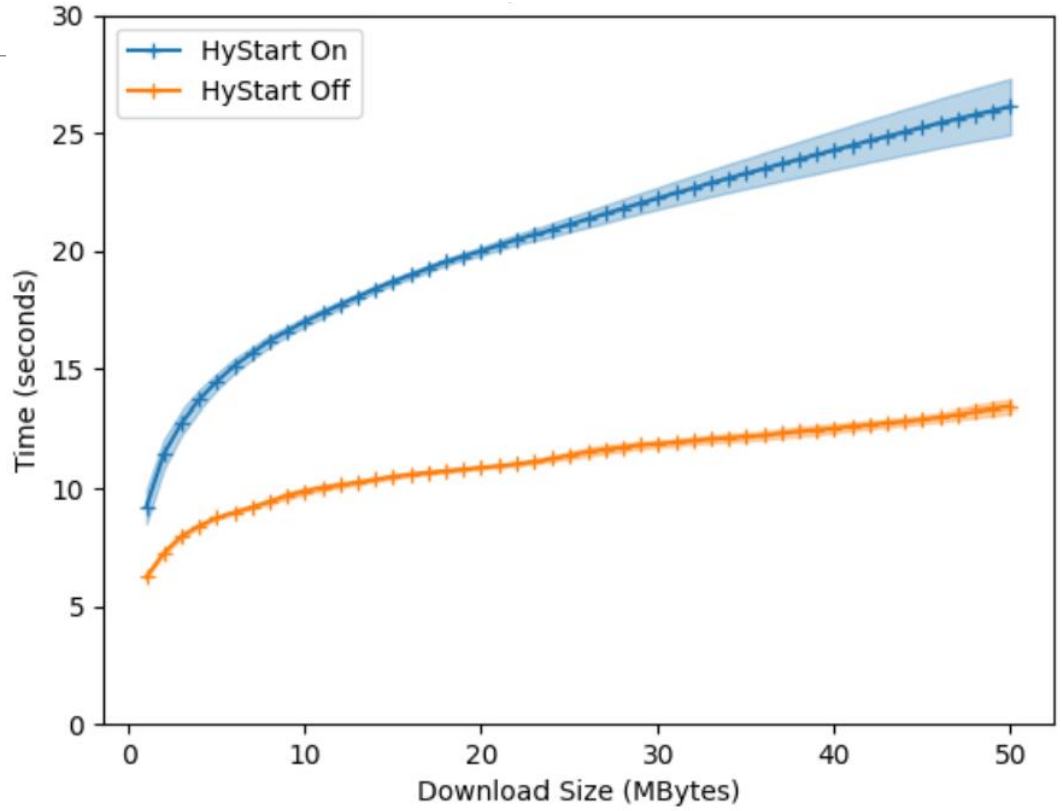


Download Times

- Run Settings:
  - rmem = 60MB, 60MB, 60MB
  - wmem = 60MB, 60MB, 60MB
  - Auto-tune enabled
- 1 MB downloads takes 50% longer with HyStart
- The average website (5MB) takes 2x longer with HyStart on

Messages:

- HyStart on makes small transfers take longer



Exiting HyStart

HyStart looks for 2 signals to exit slow start

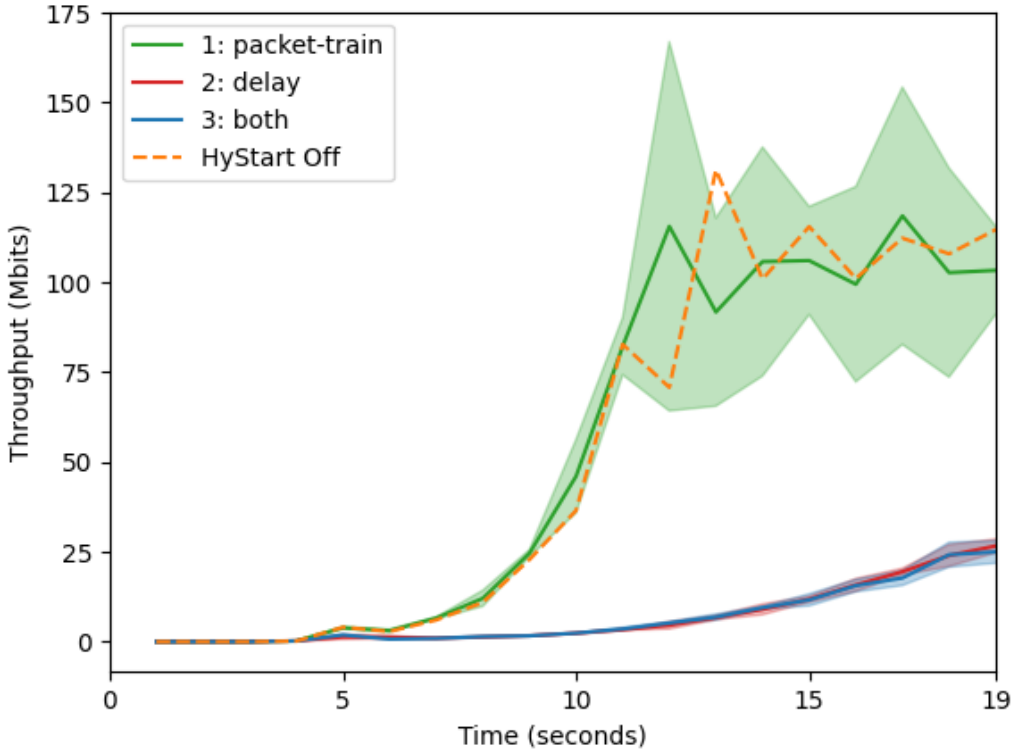
- ACK packet trains
- Increases in round-trip delay

Exiting slow start too early degrades performance

```
/* first detection parameter-ack-train detection */
if((s32)(now - ca->last_ack) <= hystart_ack_delta){
    ca->last_ack = now;
    if((s32)(now - ca->round_start) > ca->delay_min
        >> 4){
        ca->found |= HYSTART_ACK_TRAIN;
    }
}
#define HYSTART_MIN_SAMPLES  8
#define HYSTART_DELAY_MIN    (4U<<3)
#define HYSTART_DELAY_MAX    (16U<<3)
#define HYSTART_DELAY_THRESH(x)  clamp(x,
HYSTART_DELAY_MIN, HYSTART_DELAY_MAX)
if (ca->curr_rtt > ca->delay_min +
HYSTART_DELAY_THRESH(ca->delay_min >> 3)) {
        ca->found |= HYSTART_DELAY;
}
```

HyStart Exit Triggers

- Packet-train only performs similarly to HyStart off
  - Packet-train doesn't cause HyStart to exit early
- Delay only performs similarly to both
  - Delay is the reason HyStart exits slow start early

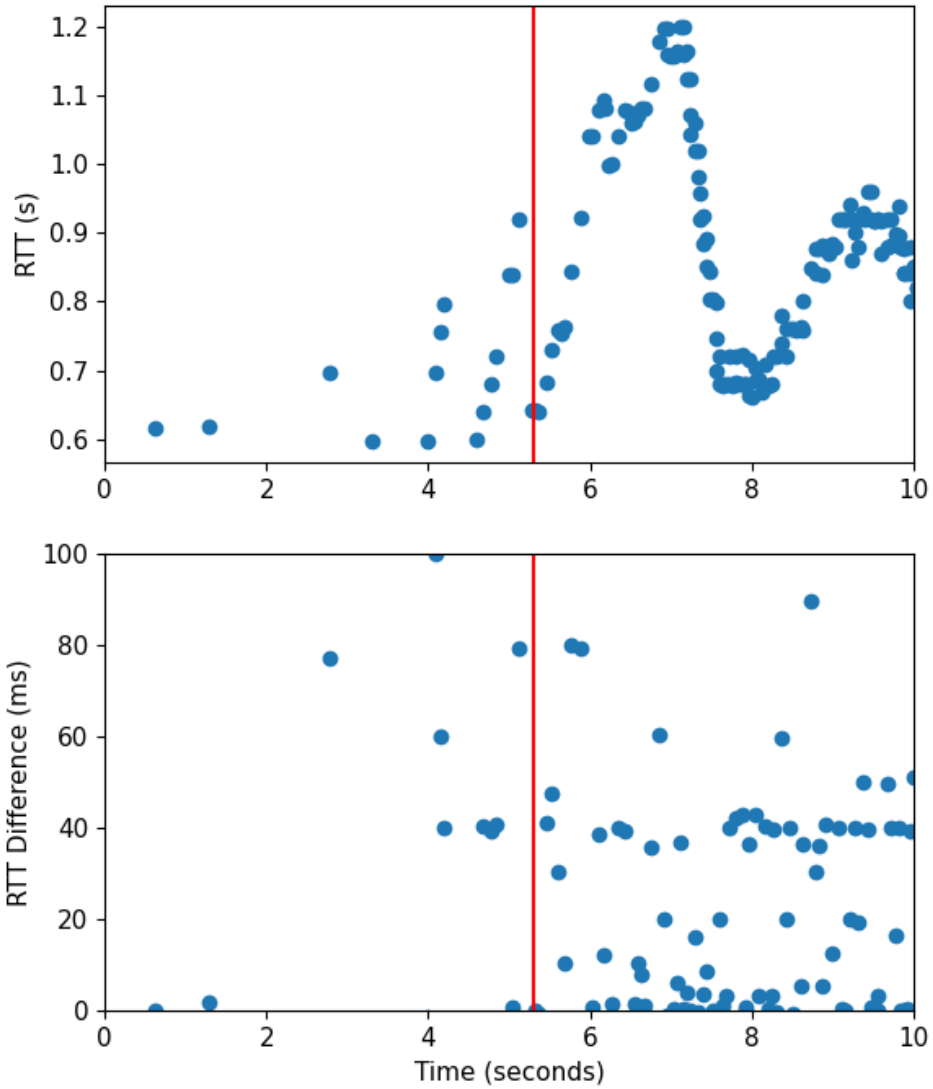


HyStart exit conditions

HyStart exits slow start after only 15-22 ACKs are received

HYSTART_MAX_DELAY is fixed at 16 ms (~3% of total RTT)

Delay changes regularly exceeded 16 ms causing HyStart to exit



HyStart Recommendations

HYSTART_DELAY_MAX should change based on network characteristics

- Tracking average and standard deviation statistics about RTT should be sufficient
- Testing is needed to determine an appropriate value

```
#define HYSTART_DELAY_THRESH(x) clamp(x, HYSTART_DELAY_MIN, UINT32_MAX)
ca->delay_min_avg = ca->delay_min_avg * (1 - EWA) + ca->delay_min * EWA
HYSTART_DELAY_THRESH(ca->delay_min_avg >> 3)
```

Conclusion

Satellites are used everywhere, and PEPs are becoming increasingly ineffective

Linux default buffer sizes are too small degrading performance over high BDP networks

Buffer auto-tuning is effective and does not limit exponential growth during slow start

HyStart can exit slow start prematurely on connections with more variable round-trip times

Future work

Improving HyStart RTT based exit

HyStart++

Other TCP settings

- Initial congestion window
- Fast convergence
- The β parameter for window factor decrease
- HyStart RTT_0

Competing flows over a satellite Internet link

- BBR
- Hybla
- QUIC

Thank You!

BENJAMIN PETERS, PINHAN ZHAO, JAE CHUNG, MARK CLAYPOOL

WPI

Viasat