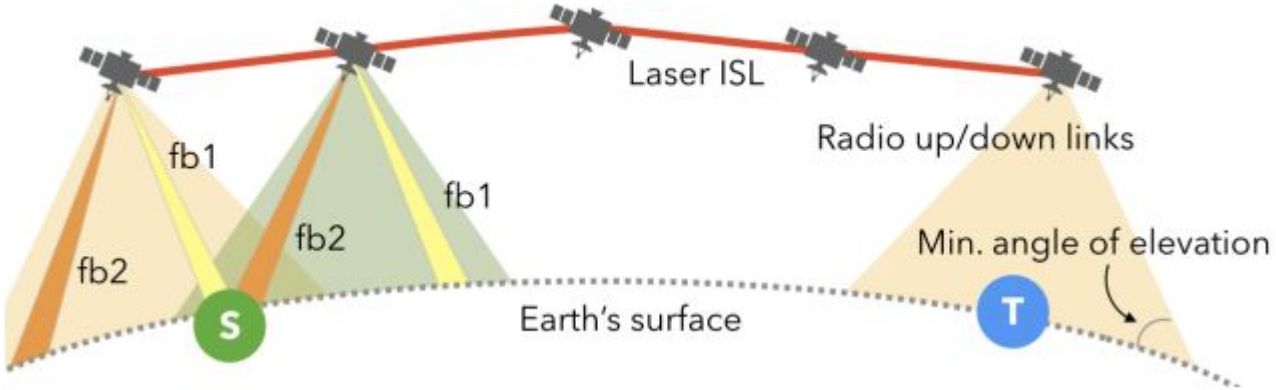
Congestion Control on LEO Satellite Networks using Hypatia

CS538 Final Presentation

By: Sairamasubash Muppalaneni, Eashan Gupta, and Milind Kumar Vaddiraju

Hypatia

- Tool to simulate satellite networking
- Based on NS3 for packet level simulations



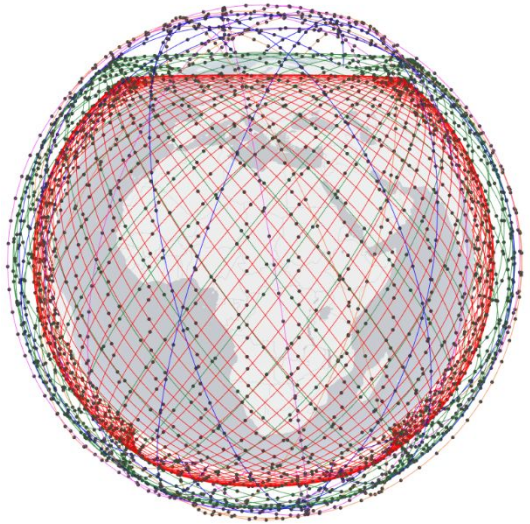


Figure from: Exploring the "Internet from space" with Hypatia, Kassing et al. ACM IMC'20

Overview

- Observe performance of CCA for competing flows
  - Hypatia does not analyse competing flows
  - Design oracle reliant loss based congestion control for greater fairness

- Hybla, COPA and BBR on Hypatia
  - Integrating BBR
  - Implementing and integrating Copa

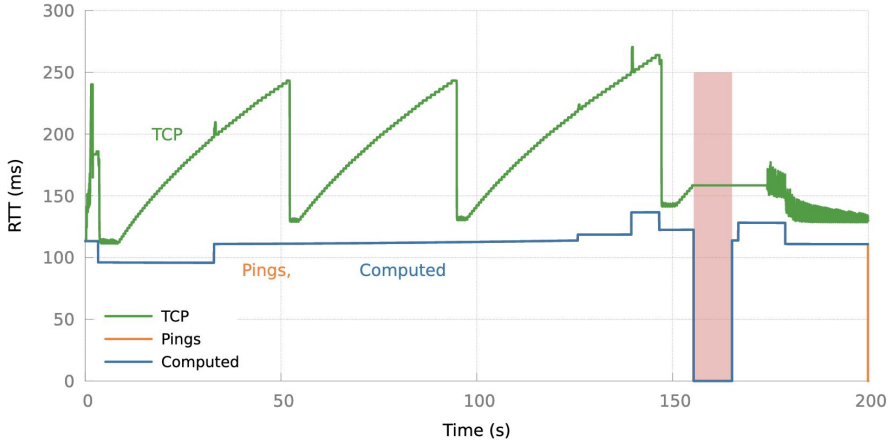
Features of LEO Satellite Network

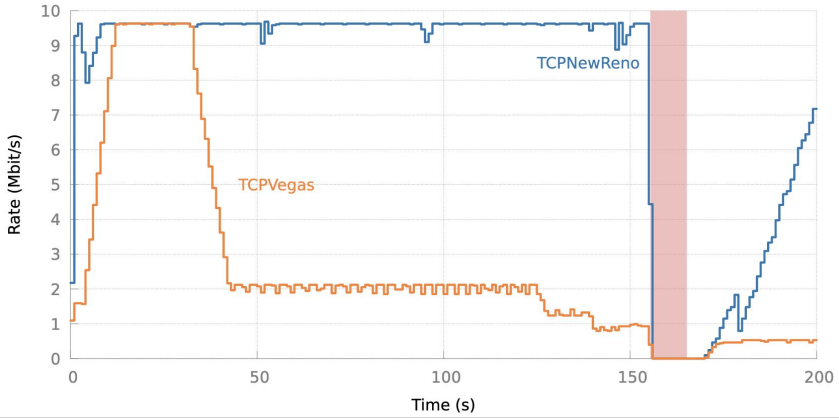
- Mobility: larger distances and velocities
- Core infrastructure itself is mobile
- LEO mobility is predictable
- Thousands of network switches (satellites) providing Tbps of connectivity

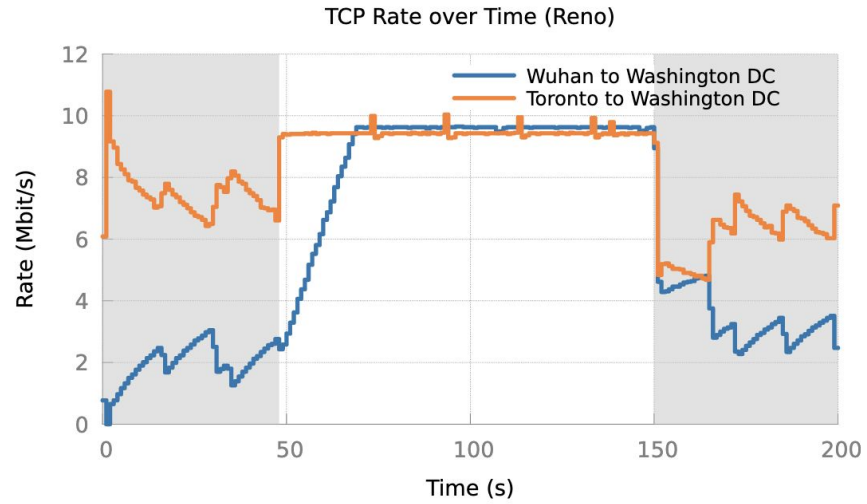
Problem 1: Improving Loss-Based CCA

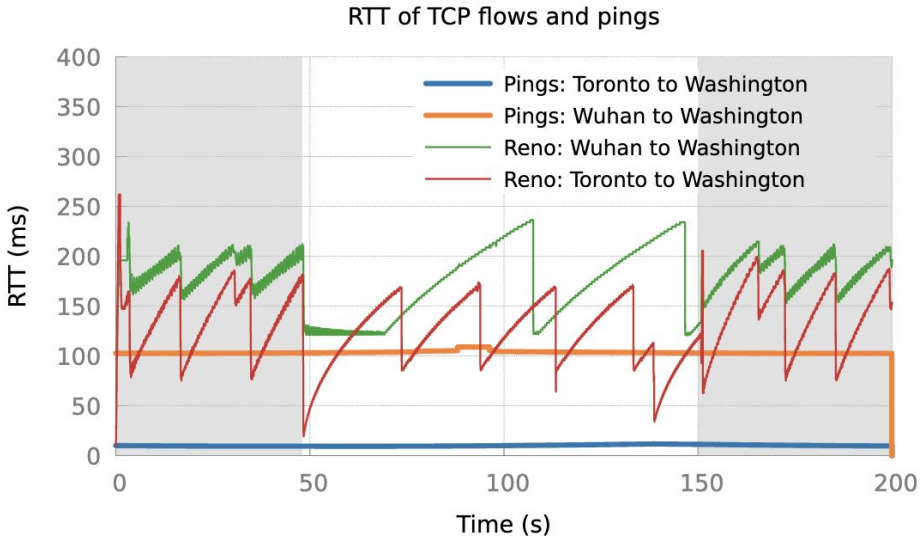
- Earlier results show Loss-based CCA work better

- Run competing flows
- Identify paths:
  - With a common bottleneck link
  - Visibly different RTTs





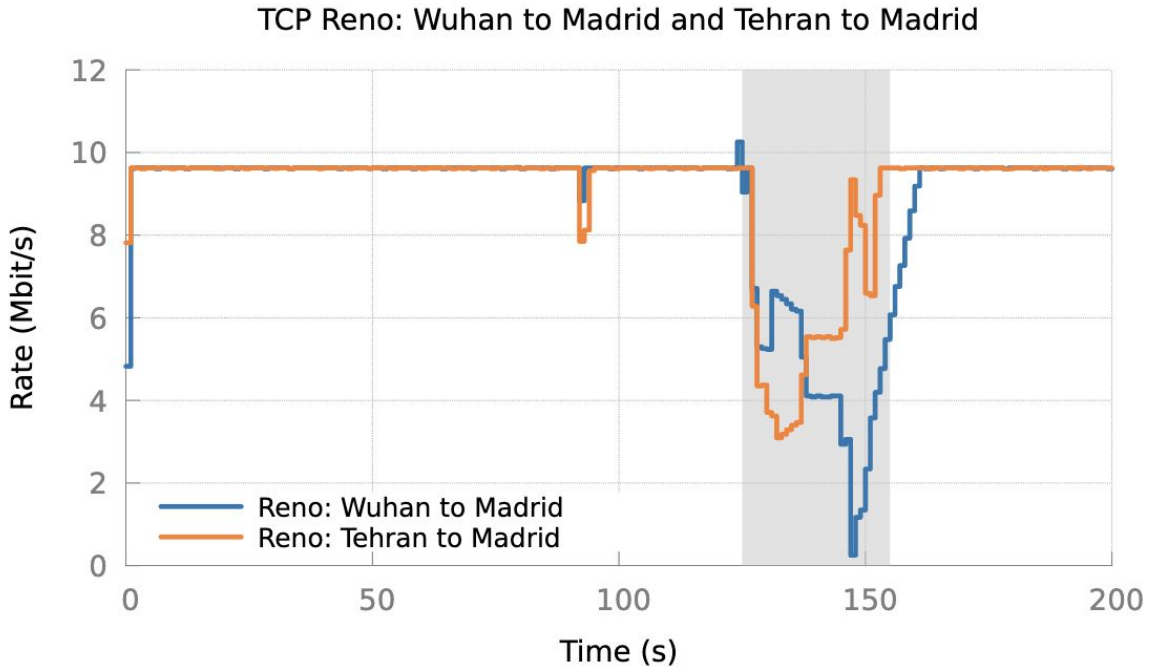




RTT(Wuhan-Washington) >> RTT(Toronto-Washington)

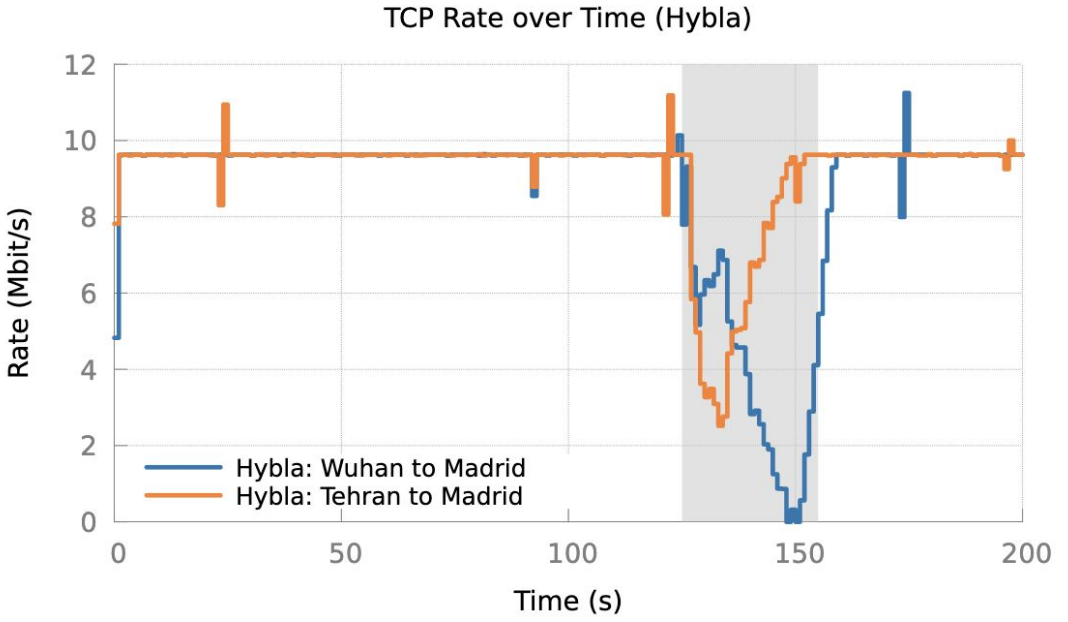
Lower latency flow **dominates**

In the worst case scenario...



Attempt 1: Hybla

- CCA for heterogeneous IP networks
- Tries to balance for RTT difference between competing flows
- Designed specially for Satellite and cellular networks



Solution: Geolocation-based Reno

- Source and destination are known
- From Hypatia simulations:

RTT(source-destination) $\propto$ Geodesic distance(source, destination)

During Congestion Avoidance Phase:

For every Ack,

$$CWND = CWND + \alpha * d(src, dst)/CWND$$

where,

$d(src, dst)$ = Geodesic distance between src and dst

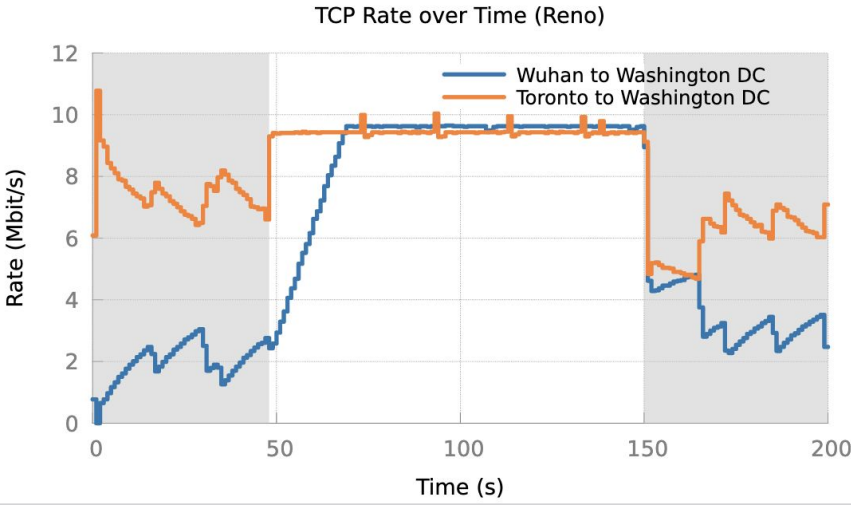
α = Normalizing Constant (based on minimum RTT)

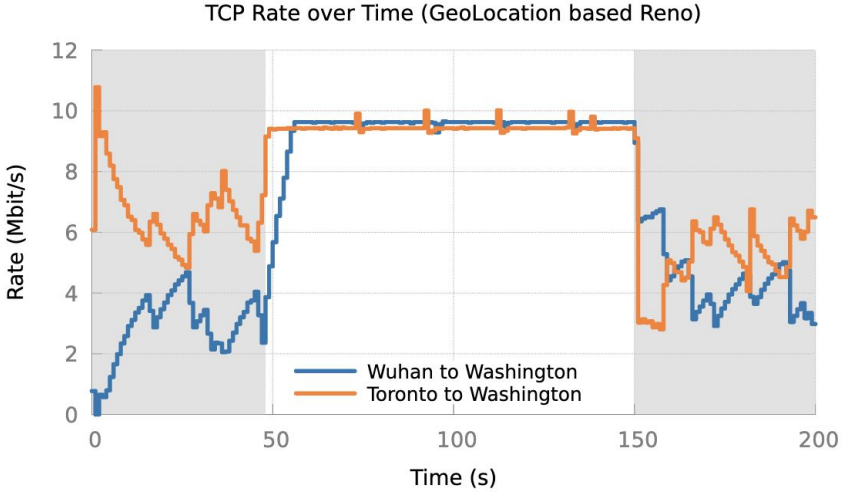
Experimental Results: Geolocation-based Reno

RTT(Wuhan-Washington) >> RTT(Toronto-Washington)

Lower latency flow **dominates** in Reno **(71.8% fairness)**

Geolocation-based Reno adjusts for this latency difference and ensures fair share **(85.8% fairness)**



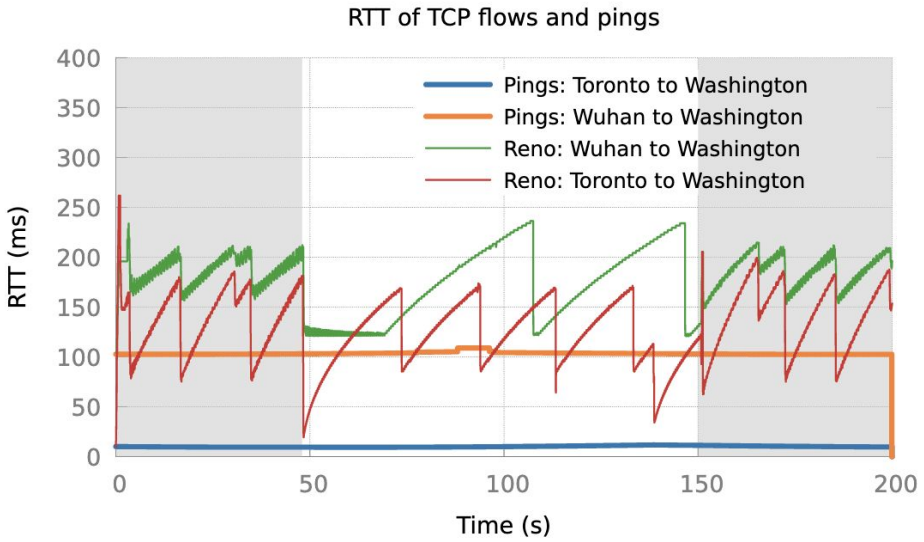


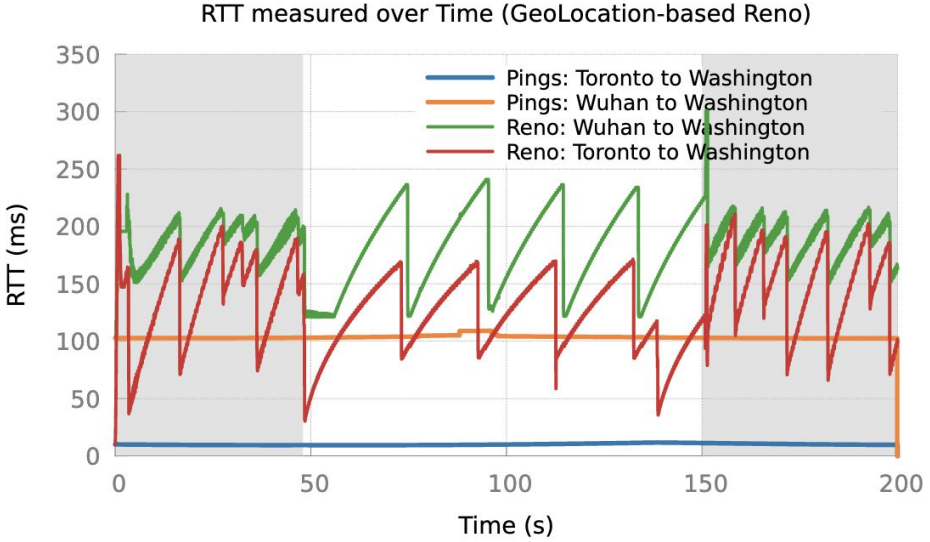
Experimental Results: Geolocation-based Reno

RTT(Wuhan-Washington) >> RTT(Toronto-Washington)

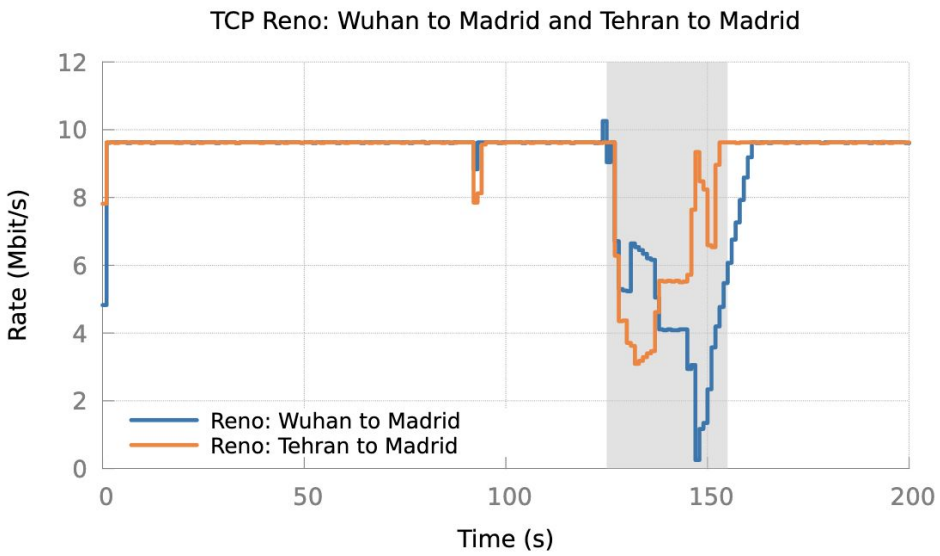
Lower latency flow is more reactive

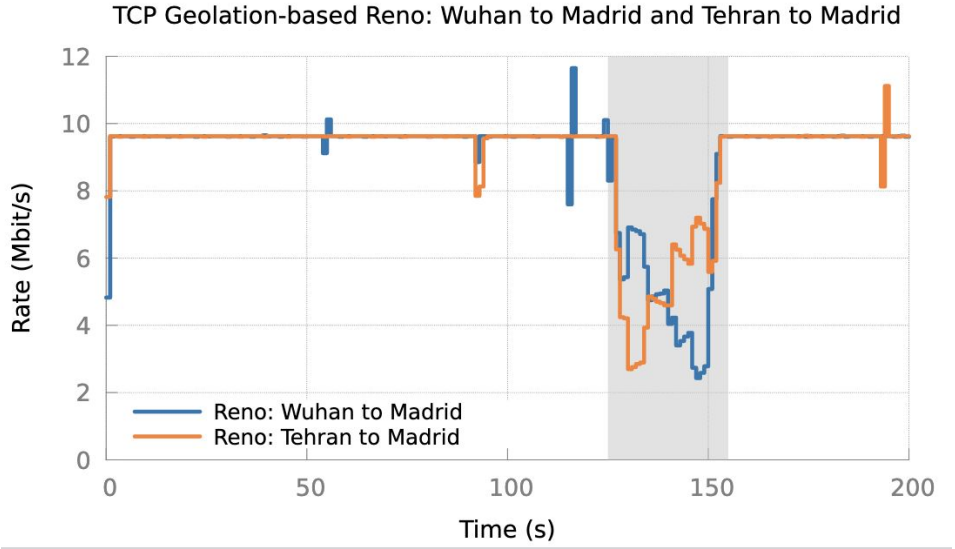
With Geolocation-based Reno, both react at a similar frequency





Another result:



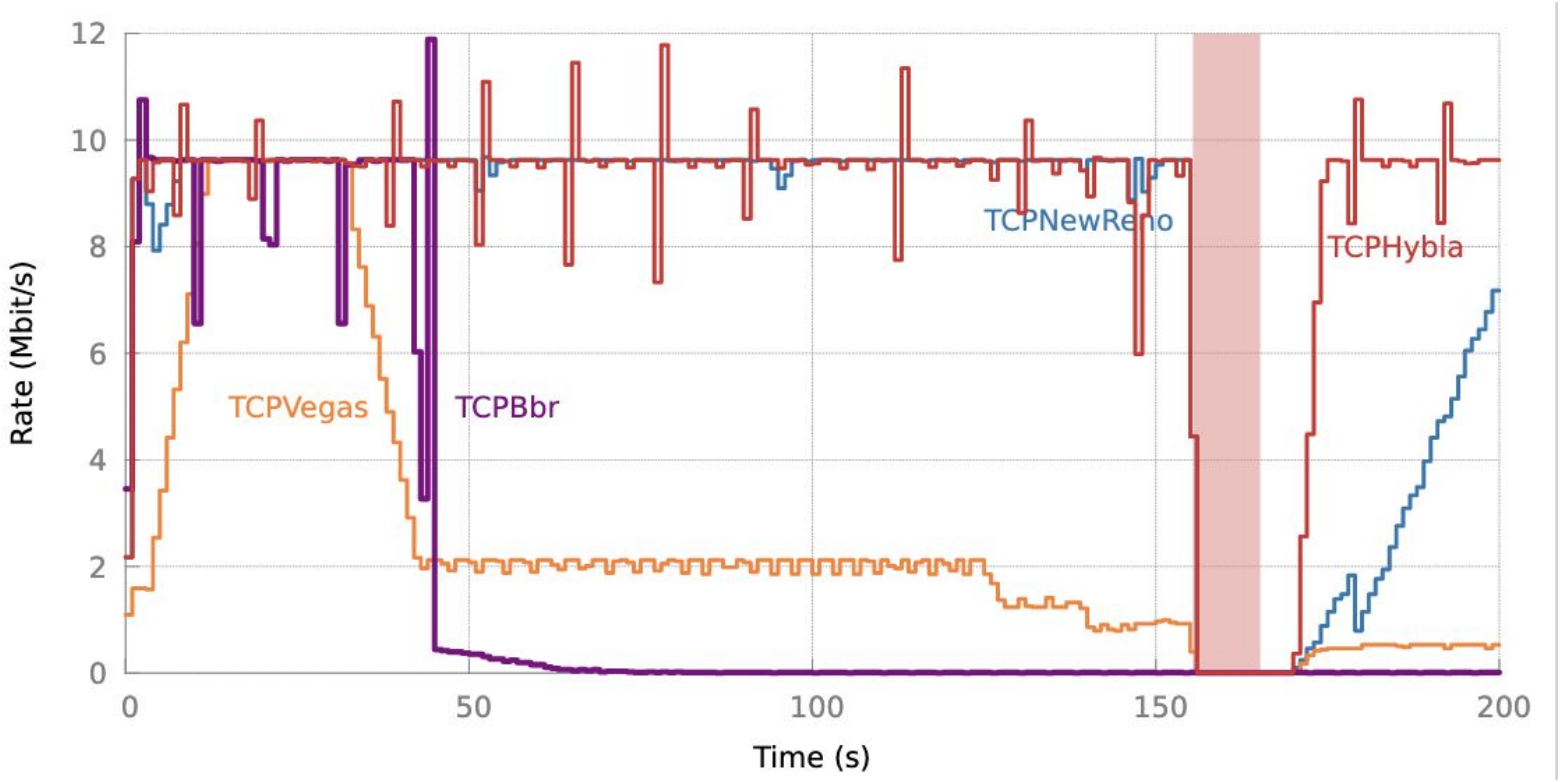


Metrics used to evaluate

- Fairness Ratio:
  - Ratio of bandwidth utilized by both flows
  - TCP Reno: **71.9%**
  - TCP Geolocation-based Reno: **85.8%**

- RTT between flows
  - Toronto to Washington: **11ms**
  - Wuhan to Washington: **105ms**

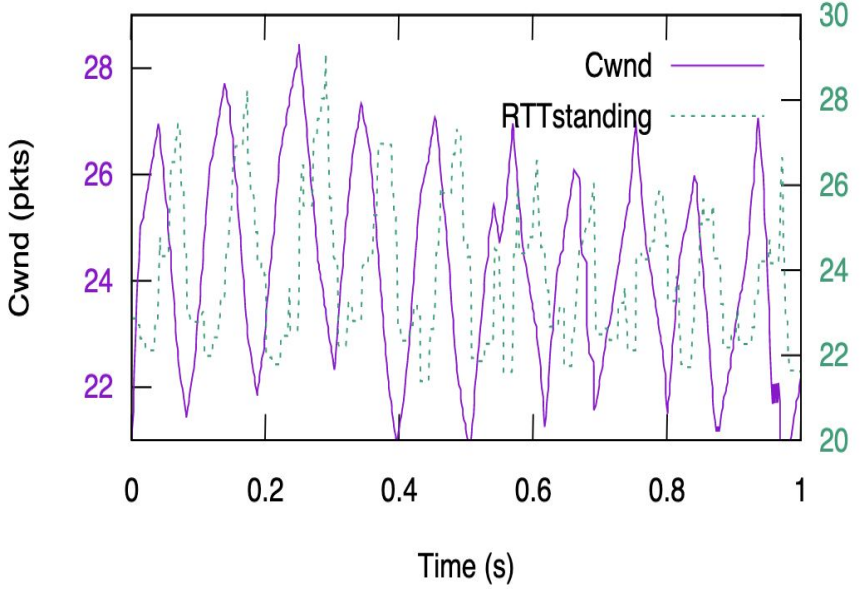
- Overall Throughput:
  - Throughput is unaffected comparing Reno and Geolocation-based Reno


Rate (Mbit/s)
Time (s)
0
2
4
6
8
10
12
0
50
100
150
200
TCPVegas
TCPBbr
TCPNewReno
TCPHybla

Problem 2: Best of loss and delay based CCA

- Buffer filling algorithms
  - Increase latency
- Delay based algorithms
  - Misinterpret latency rise as congestion
- Copa
  - Optimize f(latency, throughput)
  - High utilization
  - Low delay

Copa: replicating results



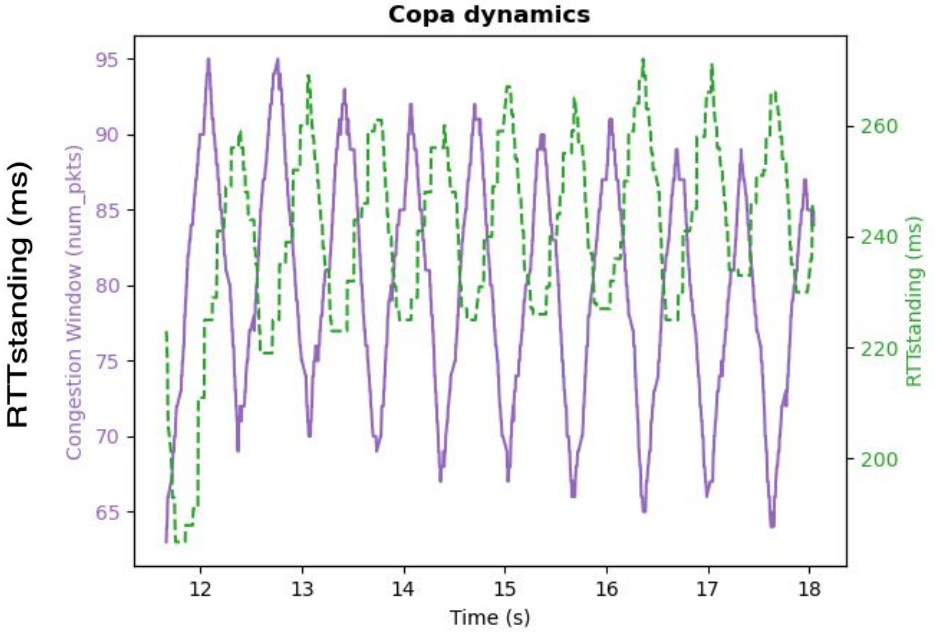
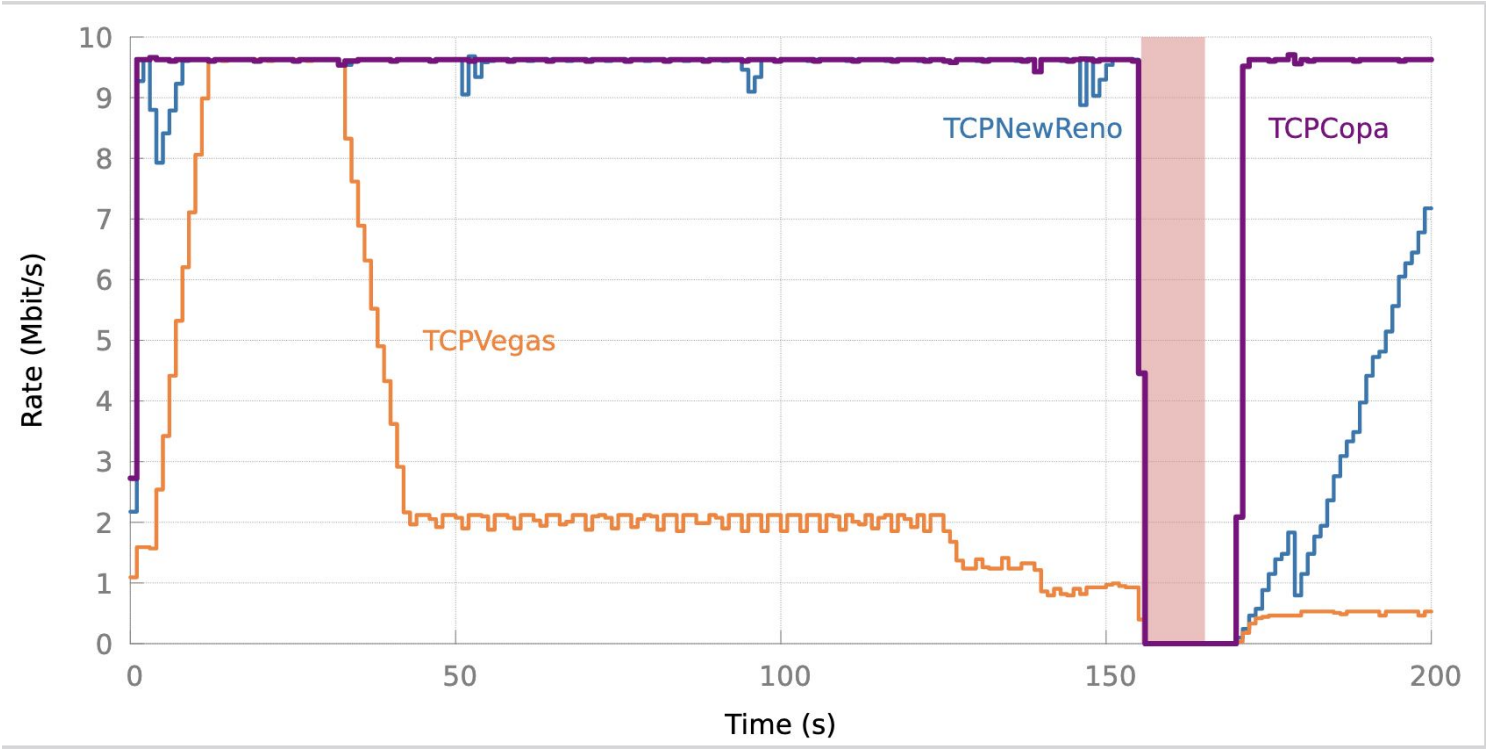
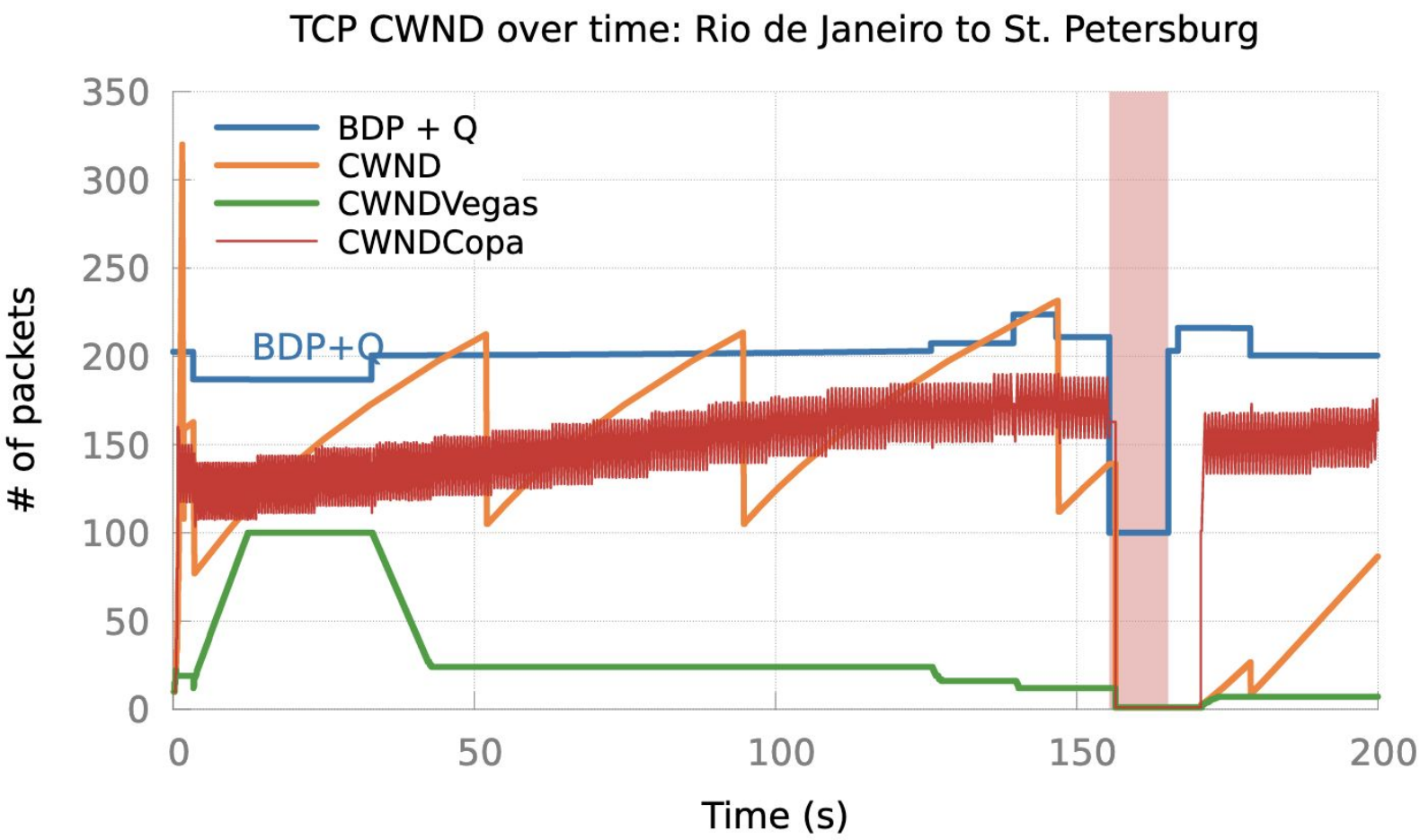
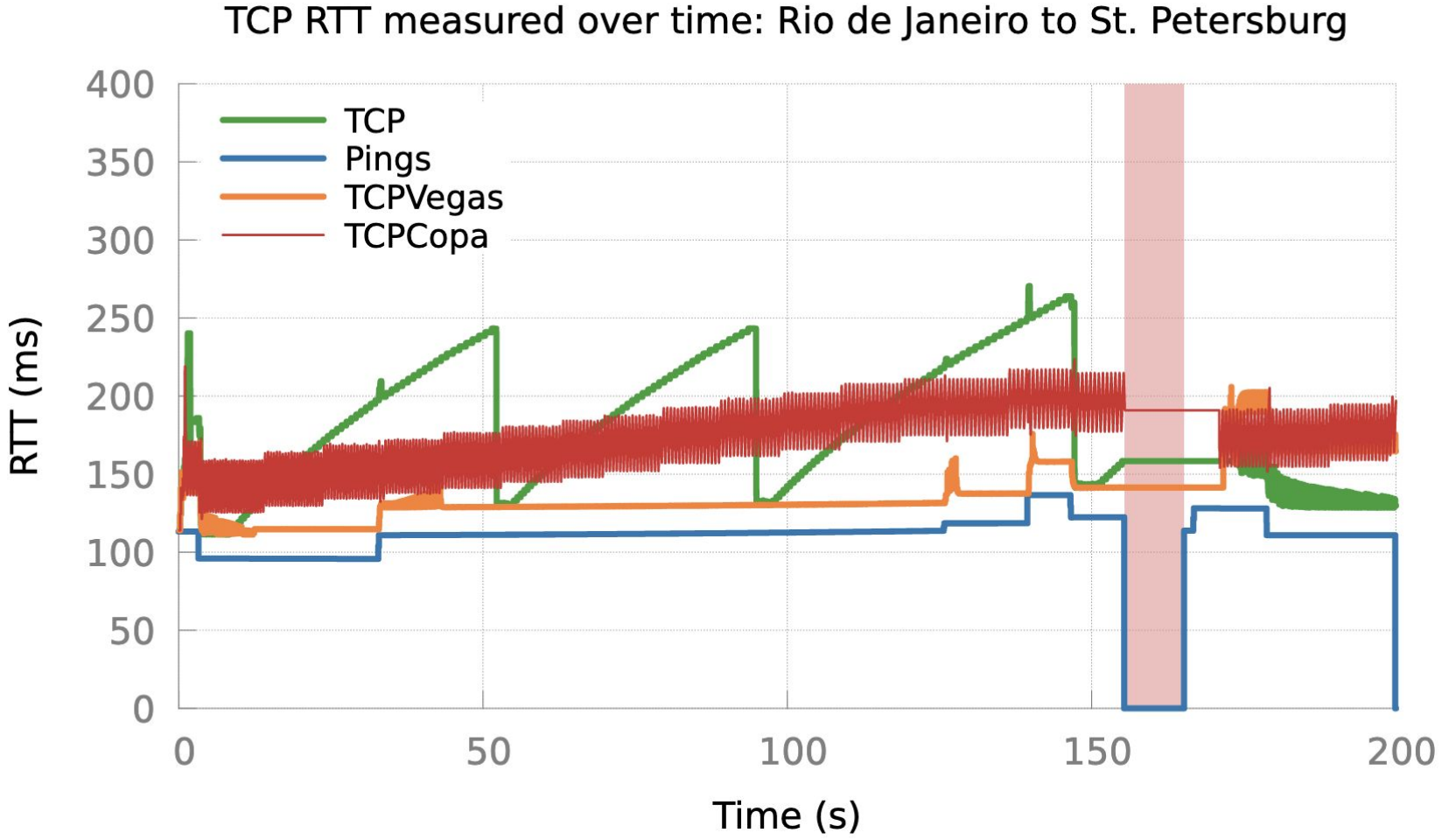


Figure from: Copa: Practical Delay-Based Congestion Control for the Internet, Arun et al. NSDI'18

Copa for LEO satellite networks




TCP CWND over time: Rio de Janeiro to St. Petersburg
BDP + Q
CWND
CWNDVegas
CWNDCopa
BDP+Q
# of packets
Time (s)
0
50
100
150
200
250
300
350


TCP RTT measured over time: Rio de Janeiro to St. Petersburg
TCP
Pings
TCPVegas
TCPCopa
RTT (ms)
Time (s)
400
350
300
250
200
150
100
50
0
0
50
100
150
200

Conclusion

- Loss-based CCA
  - Variant using simple Geodesic distance modification on Hypatia
- Copa and other CCAs on Hypatia

Unanswered questions

- How do different algorithms compete?
- How does Doppler effect affect congestion control?
- How can we leverage predictable path changes?

Thank you!