ETC: An Elastic Transmission Control Using End-to-End Available Bandwidth Perception

Feixue Han, Qing Li, Peng Zhang, Gareth Tyson,

Yong Jiang, Mingwei Xu, Yulong Lan, ZhiCheng Li

Tsinghua PCL HKUST (GZ) Tencent

The 2024 USENIX Annual Technical Conference, July, 2024

Targets of Congestion Control

- **High throughput**

  Quickly grabbing the spare bandwidth.

- **Low latency**

  Keep the queue at a low length.

- **Fairness**

  Max-min fairness.

- **Convergence speed**

  How quickly the flow rate becomes stable.

- **Practically**

  Acceptable overhead.

  Requirements on hardware (GPU, programmable switches).

Current CC Algorithms in WAN

01 Heuristics

Loss, Delay, ECN* based methods:

Cubic, Vegas, Copa, ...

02 Periodically Decision

Learning-based Algorithms:

PCC, Vivace, Indigo, ...

Limitations of Current CC signals

Loss and RTT -- Mainstream CC signals in WAN.

1. RTT == minimal until packets accumulating in the router buffers
2. Loss == router buffers are full

× They work only once the collective flows' rates exceed the link capacity.

Limitations of current CC

01

Lack of Fair Convergence

1. Convergence is considered only after the available bandwidth has been occupied.
2. Rate increase/decrease step size is not related to the available bandwidth.

02

Lack of Optimal Rate Change

1. Conservative step size at start-up.
2. Becoming more aggressive in the subsequent adjustments.

Example: slow-start, velocity parameter..

ETC Design Principles

01

Pre-Congestion Consensus

- Find a pre-congestion signal used by all senders (sharing a bottleneck) to reach a common estimate of the available bottleneck bandwidth.

02

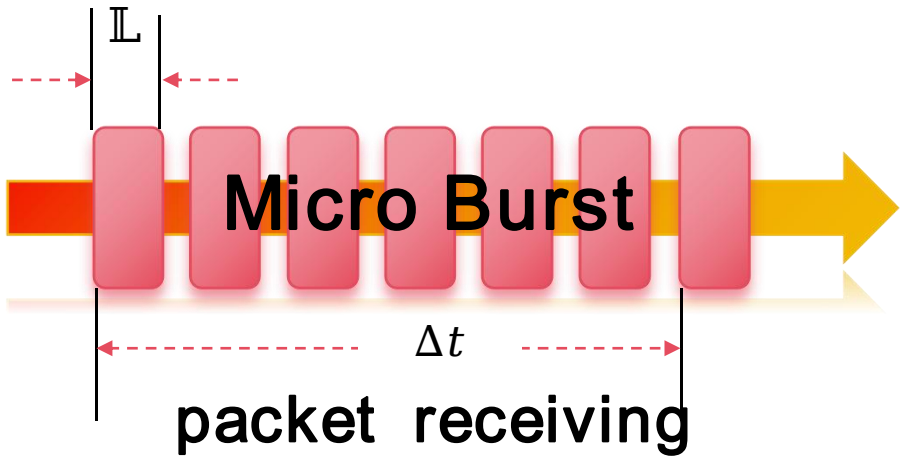
Dynamic Step Sizes

- Employ dynamic step sizes to achieve safe, fast, and fair rate adjustment.

The pre-congestion signal: pulling rate

Pulling rate

- The instantaneous receipt rate of a micro-burst.
- Micro-burst: N(N ≥ 2) consecutive packets.



ETC paces data in micro-bursts.

Flows that pass through the same last bottleneck should obtain approximately the same estimate of the available bandwidth.

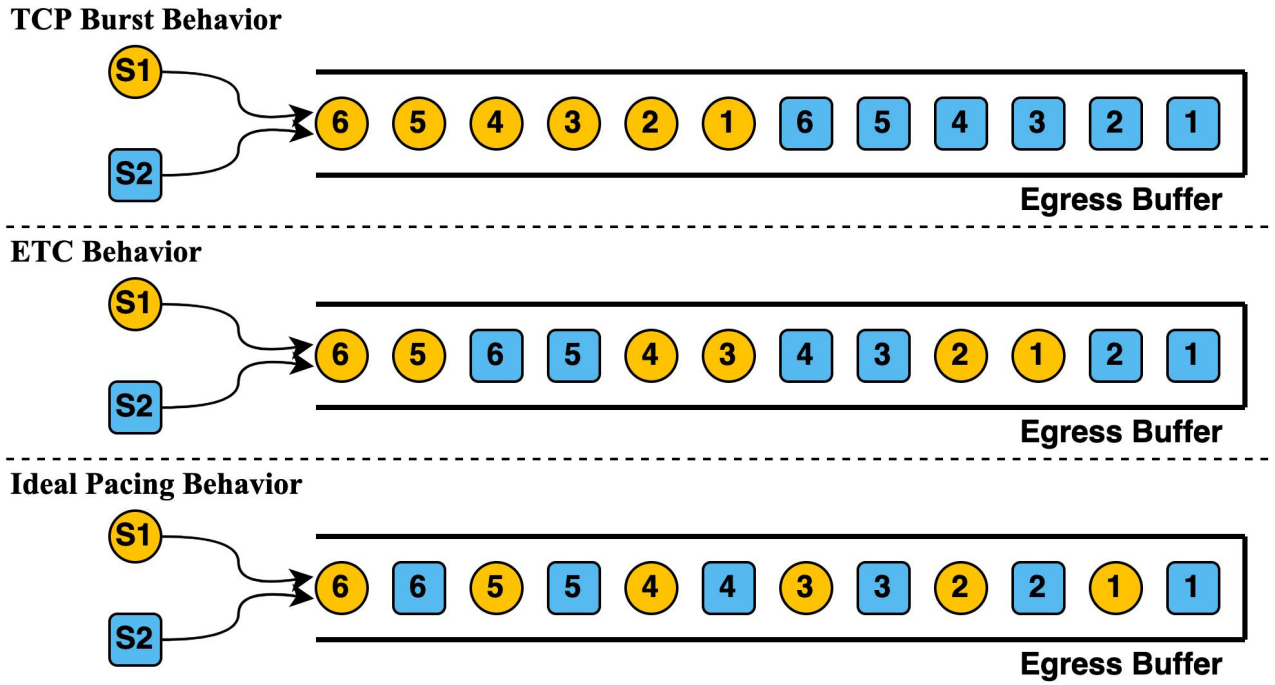
Pulling Rate $\in$ [ABW, Capacity]

1. The pulling rate is always **greater than or equal to** the available bandwidth.
2. The pulling rate is **less than or equal to** the link capacity.
3. The pulling rate is **positively correlated with** link capacity.

Focus more on consistency rather than measurement accuracy!

Transmission in micro-bursts

Pacing has long been shipped in Linux and data center networks.
Dependent on system timers!



Sending interval is much more precise (us order) compared with the operating system scheduler's (ms order).

Current high-precision pacing solutions

01 Hardware solutions

- Limited realizability

02 Software solutions

- High CPU overhead
  - Timer interrupt-based.
  - Gap packet-based.

ETC pacing with ACK-Clocking

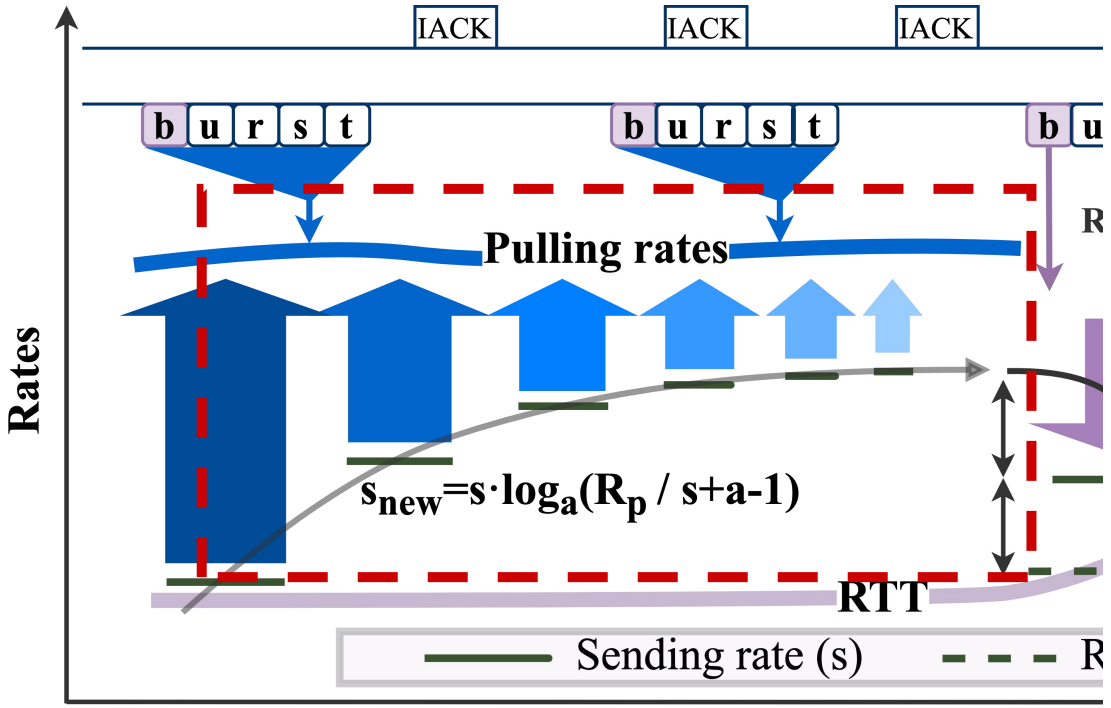
ACK Seq	Time(s)-12Mbps	ACK Seq	Time(s)-12Mbps	ACK Seq	Time(s)-100Mbps	ACK Seq	Time(s)-100Mbps
279	5.100818	283	5.105269	279	7.686917	283	7.687006
280	5.101901	284	5.106348	280	7.686937	284	7.687036
281	5.102996	285	5.107473	281	7.686959	285	7.687071
282	5.104066	286	5.108532	282	7.686980	286	7.687098

1. ACKs'arrival interval is more fine-grained than the inherent ms-level timer!
2. ETC senders checks whether a micro-burst should be sent at every ACK arrival -- integrates the processing of data sending into the processing of ACKs.

Dynamic Step Sizes--Rate Acceleration

- Adjust based on the distance of the pulling rate and current sending rate.

$$distance = \frac{pulling\ rate}{sending\ rate}$$

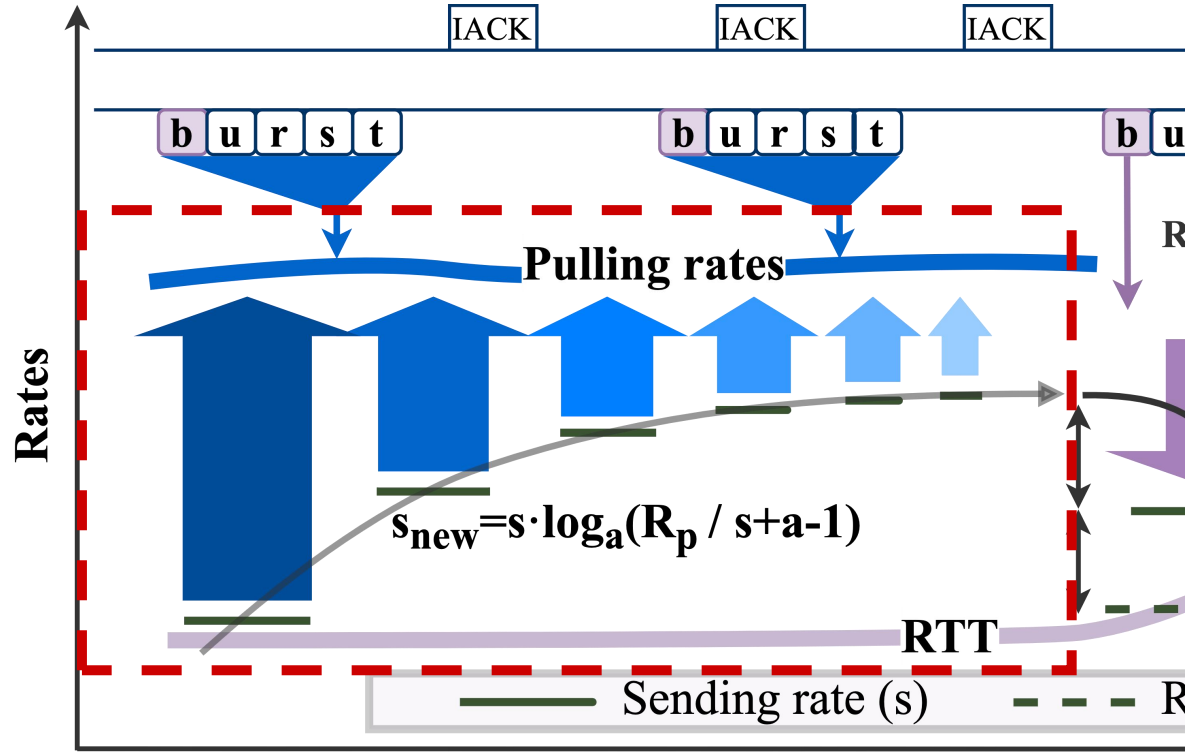


- Choose a convex function.

$$f(pulling,\ sending) = log_{\alpha}(distance + \alpha - 1)$$
$$s.t.\ f(pulling,\ pulling) = 1$$

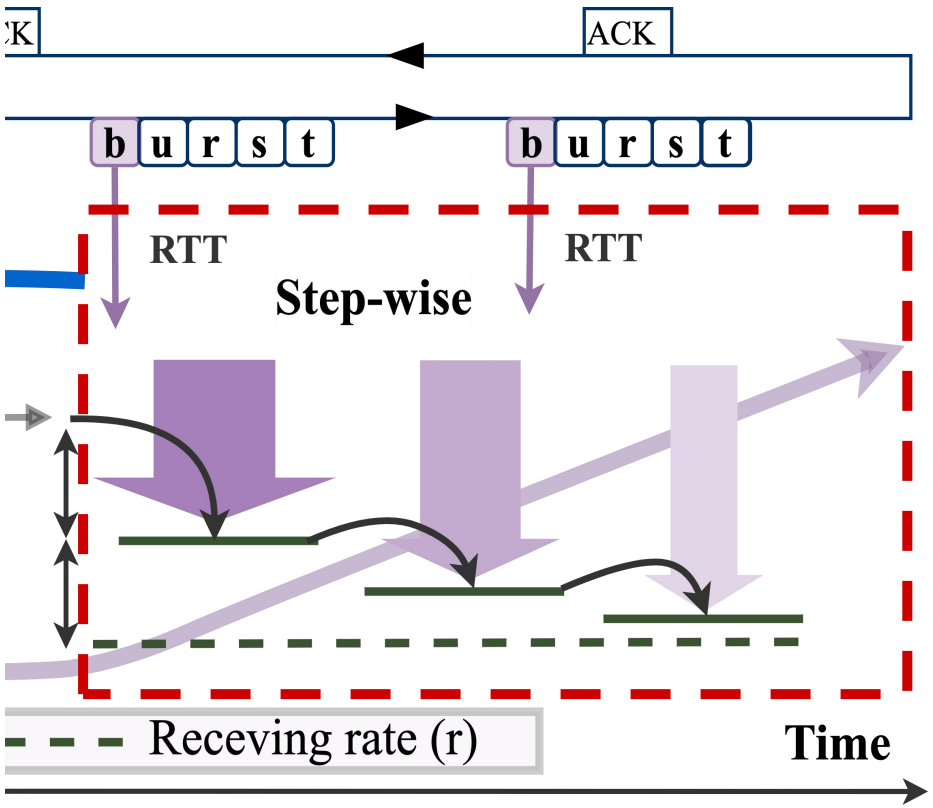
Dynamic Step Sizes--Rate Acceleration

1. The pulling rate caps the sending rate.
2. A higher rate with a more conservative step.
3. Flows move towards fairness after each adjustment.



Dynamic Step Sizes--Rate Acceleration

1. Rate decreases based on the receiving rate.
2. Step-wise manner avoids overreaction.
   - Considering that the measured receiving rate is usually low.
3. Flushes the accumulated bytes with a lower rate: η·s.
   - Try to keep minimum latency.



Evaluation of ETC

01

Implementation

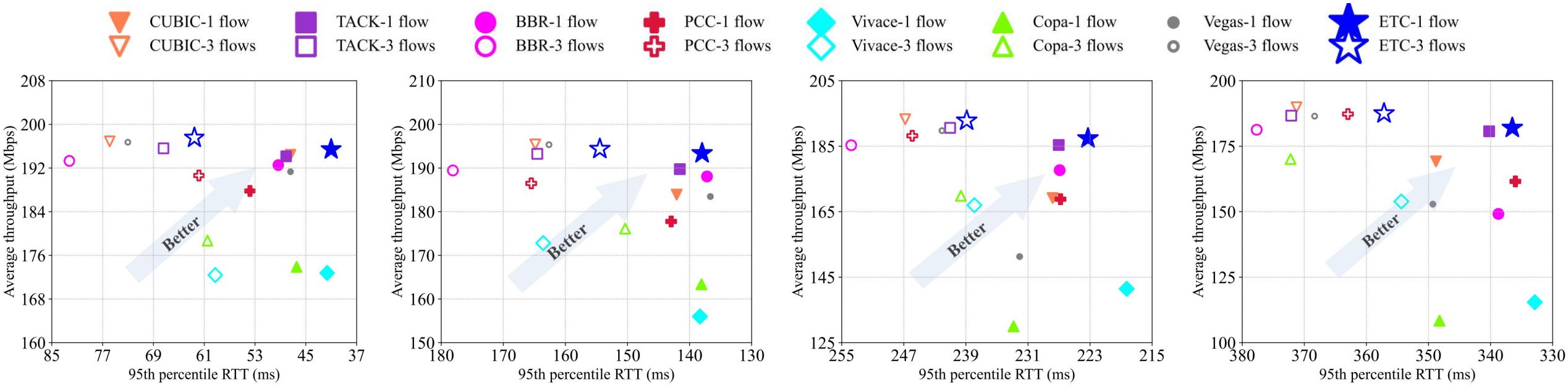
- A user-space tranport protocol with UDP as the substrate.
- Implemented in the form of SDK.
- Supports multiple platforms: Windows, macOS, Android, iOS, and OpenWrt.

02

Setup

- Comparing schemes: CUBIC/BBR/Vegas/PCC/PCC Vivace/Copa/TACK
- Platforms: Pantheon nodes deployed in cloud servers/local.

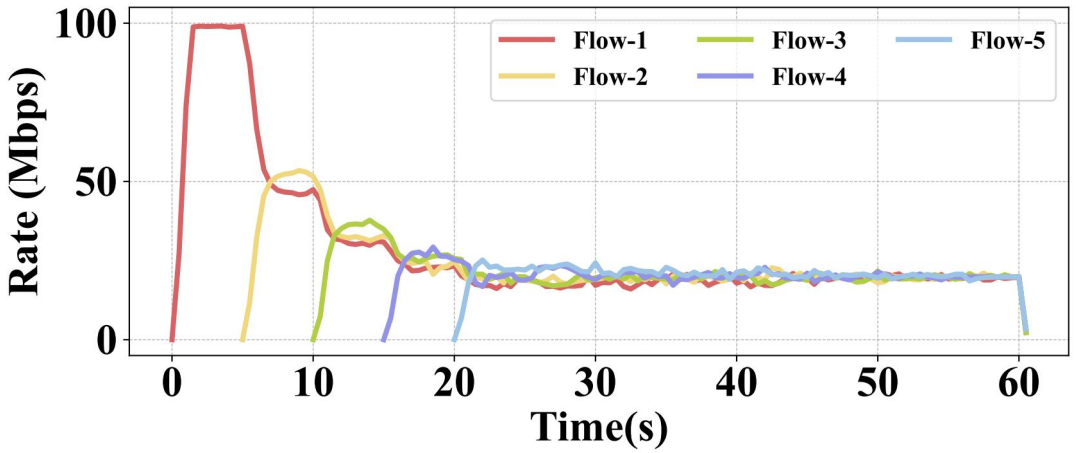
Throughput & Delay



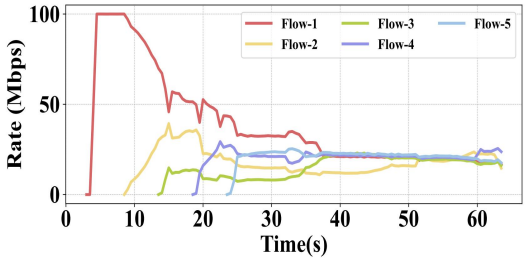
Beijing to Shenzhen | Beijing to Bombay | Beijing to Virginia | Beijing to Sao Paulo

1. 10% and 30% one-way-delay improvement in one-flow or three-flow scenarios compared with BBR.
2. 15% throughput asvantage than Copa/Vivace with almost the same one-way-delay.

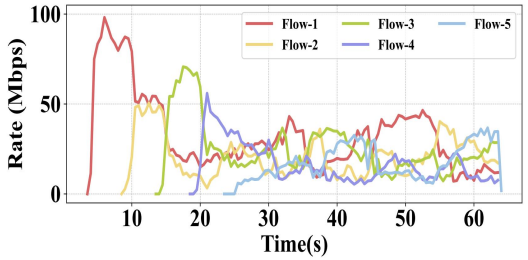
Fairness & Convergence



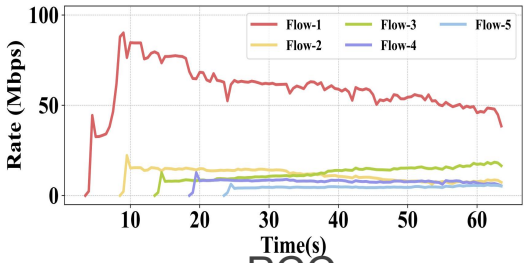
ETC: Fast and fair convergence!



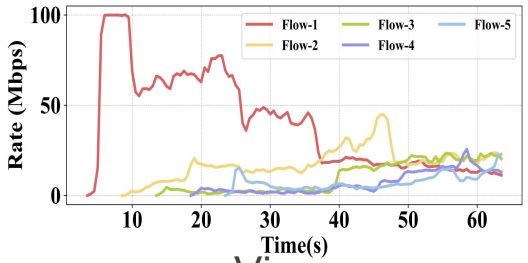
CUBIC



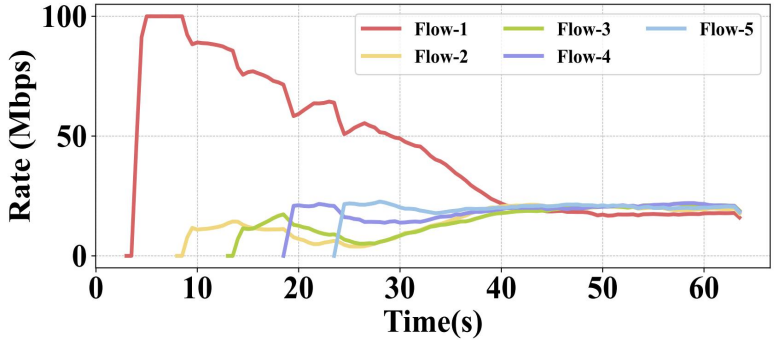
TACK



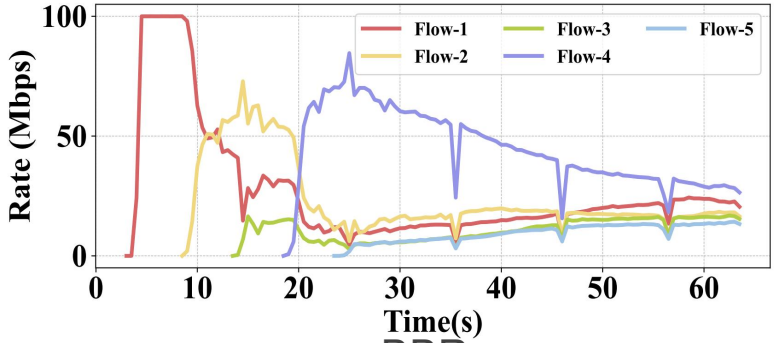
PCC



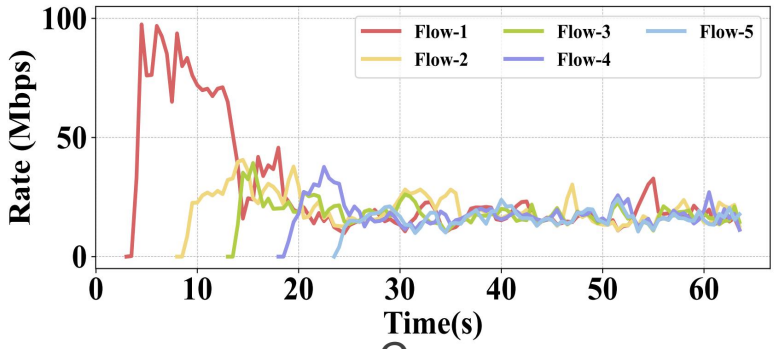
Vivace



vegas

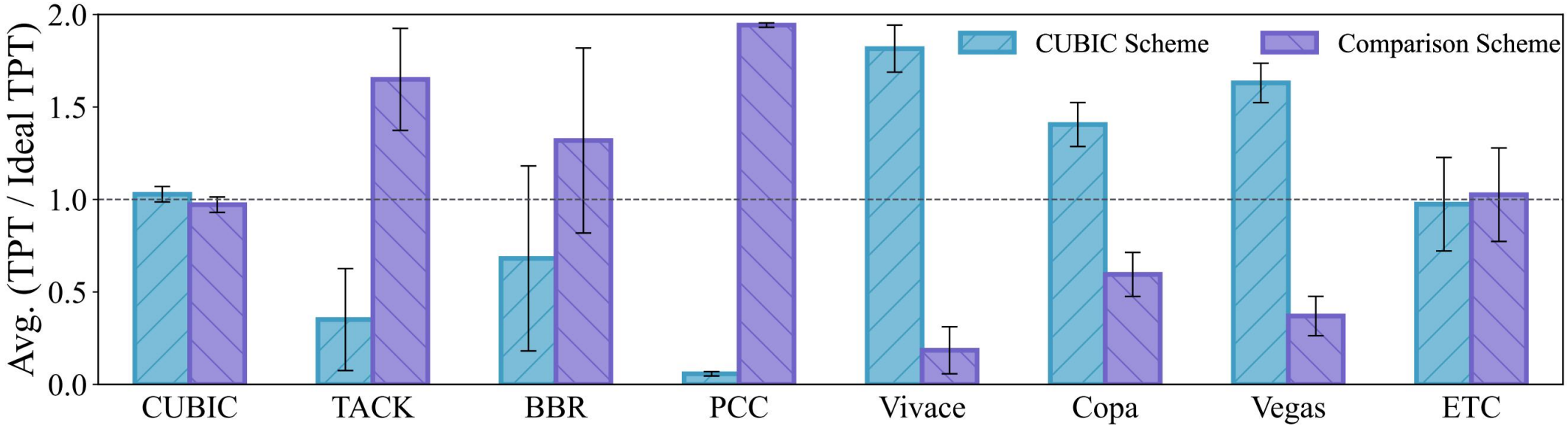


BBR



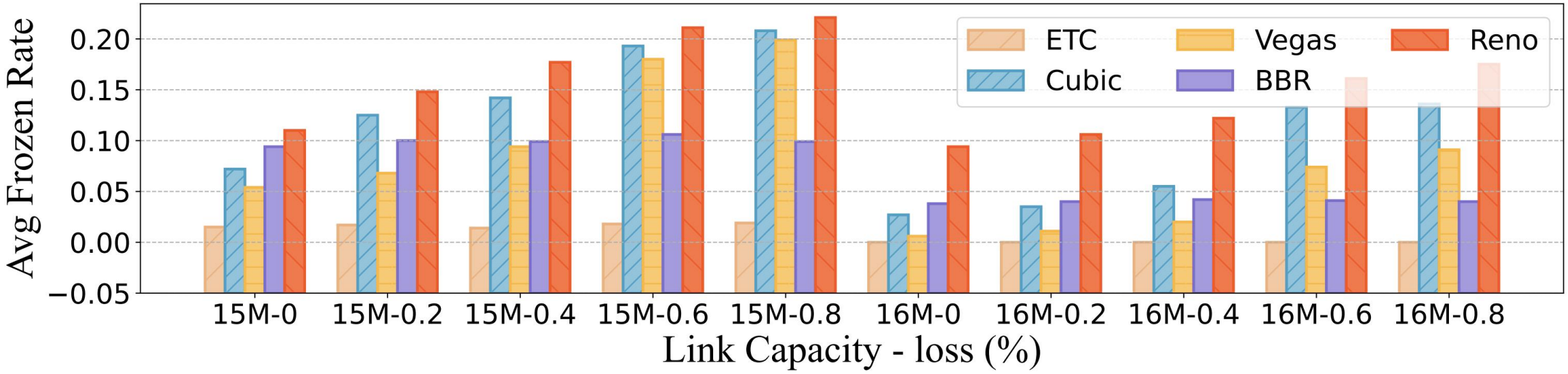
Copa

Coexistence with Loss-Oriented Scheme



ETC shows a good throughput without damaging CUBIC performance.

Video Transmission



1. Tested in real video application.
2. The bit rate of the video flow is fluctuating around 12Mbps.
   - ETC maintains a zero rebuffer rate in 16Mbps link, making more efficient use of the bandwidth.

Conclusion

01

A pre-congestion consensus signal

- ETC use pulling rate to guide the rate adjustment of flows.

02

Dynamic step sizes

- ETC moves slower as approaching the pulling rate.

Thanks for Listening!

Q & A