香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

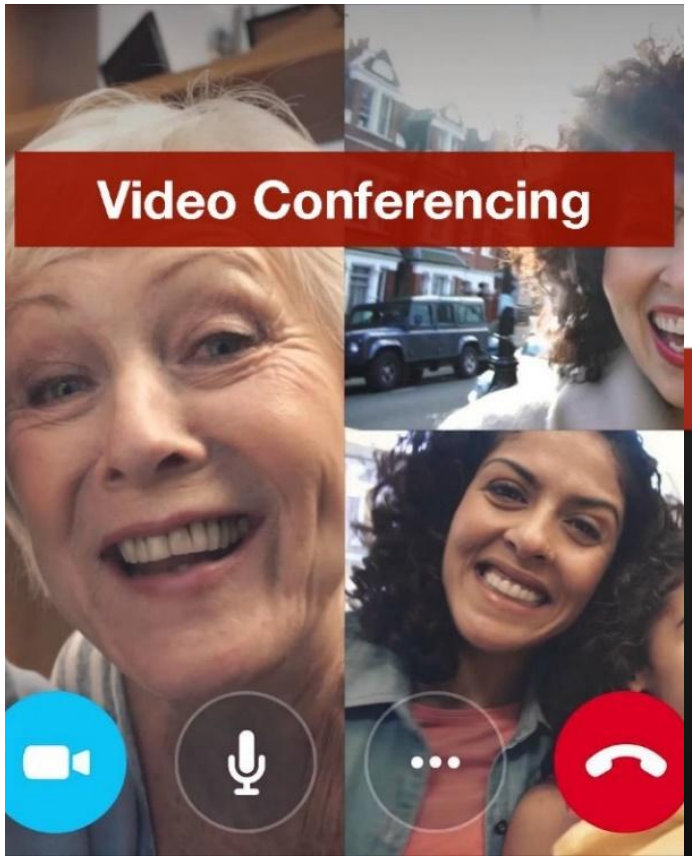
ACE: Sending Burstiness Control for High Quality Real-Time Communication

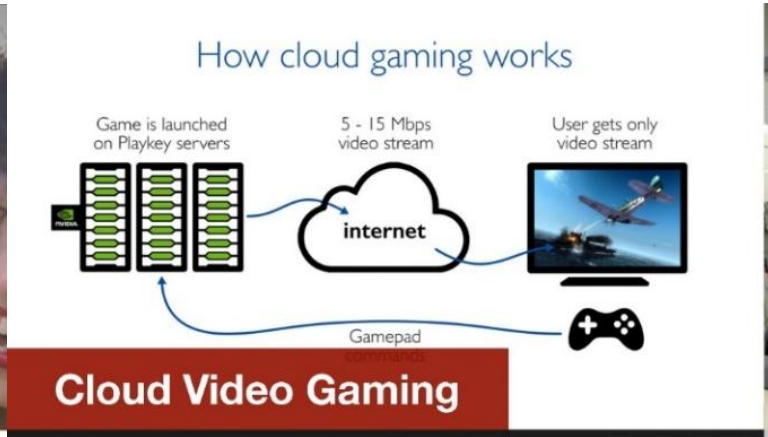
Xiangjie Huang, Jiayang Xu, Haiping Wang, Hebin Yu
Sandesh Dhawaskar Sathyanarayana, Shu Shi, **Zili Meng**

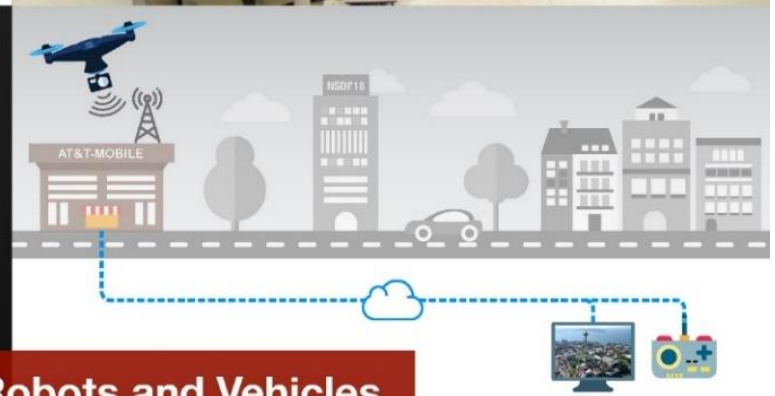
July 21, 2025
Accepted to SIGCOMM'25

Ultra-low-latency Video Streaming

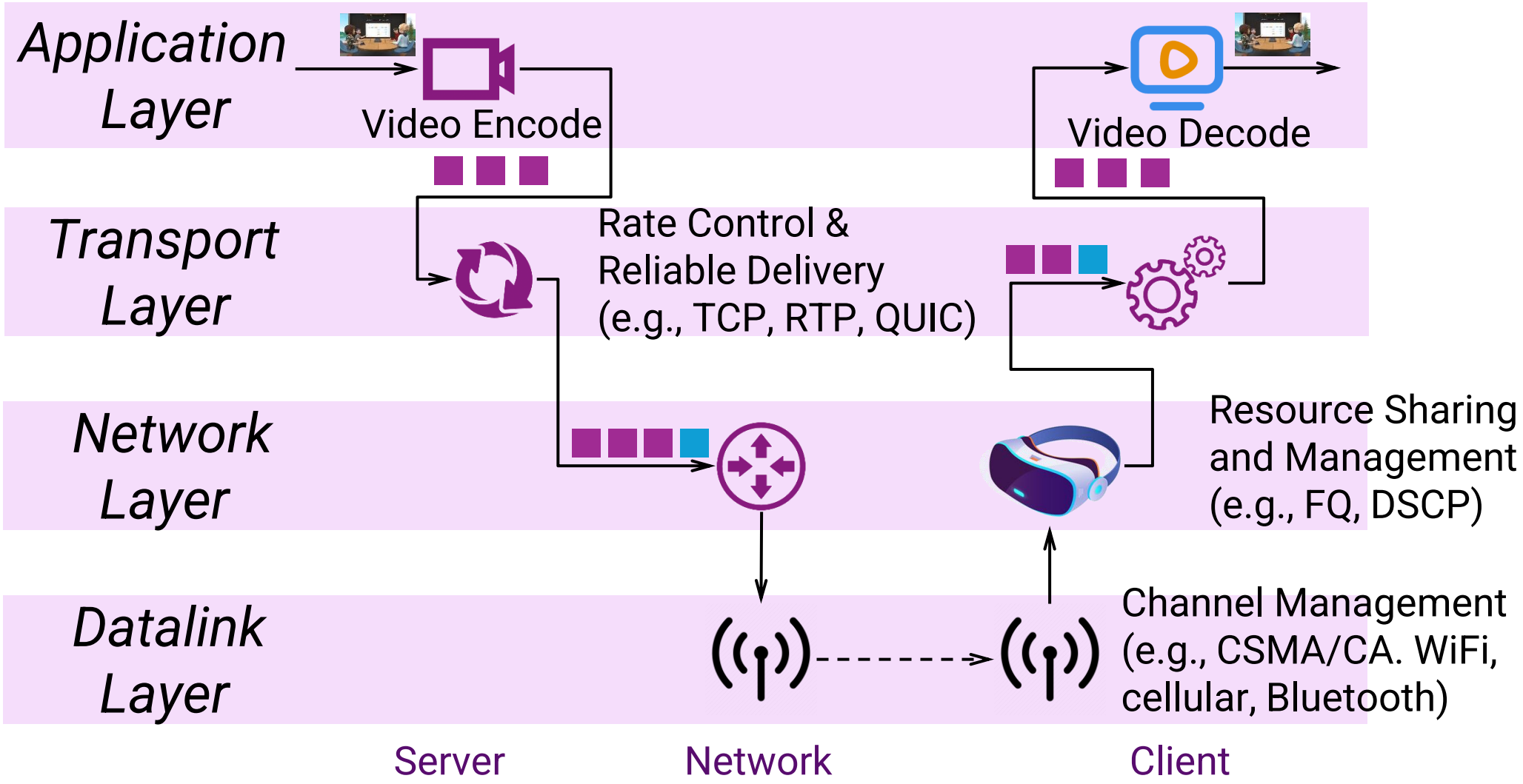
- One of the key uses on the Internet today.



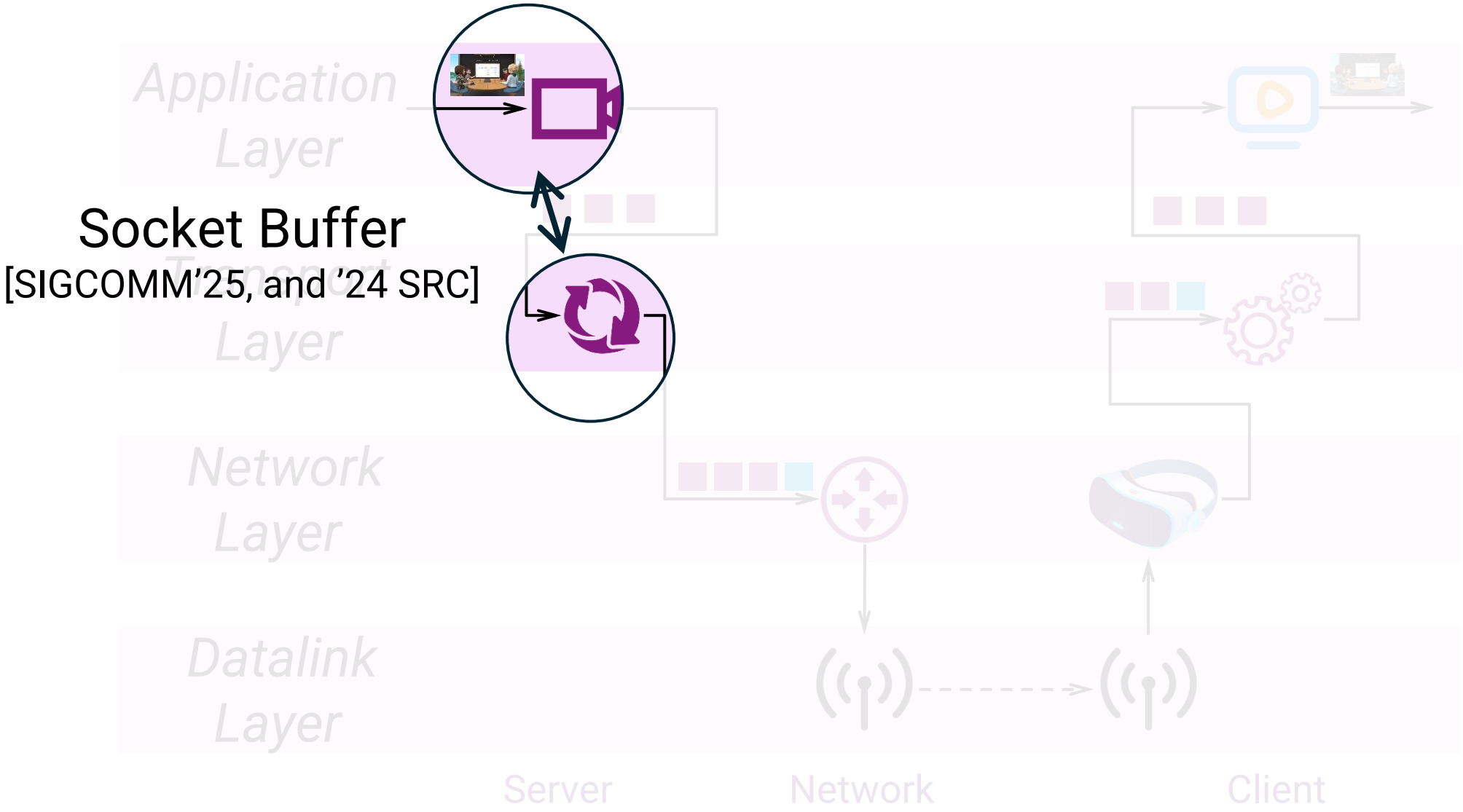




Internet Layering Architecture



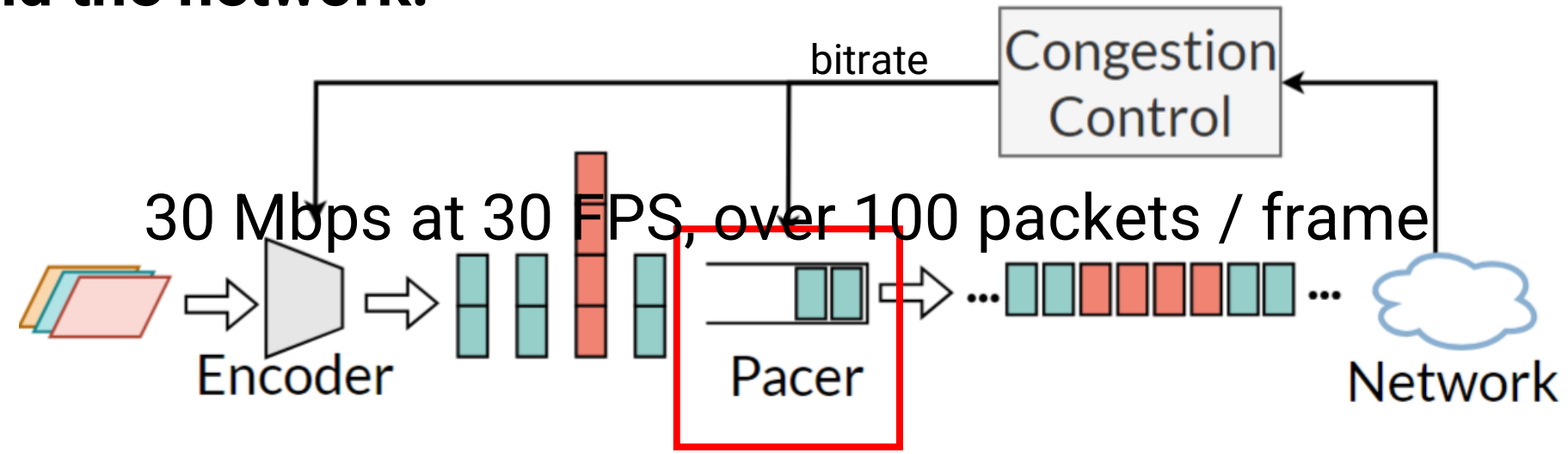
What have we done so far



Socket Buffer
[SIGCOMM'25, and '24 SRC]

Motivation
Unnecessary Pacing Latency

- **We found a key missing part – a queue between the video codec and the network.**



Pacer: Smoothly sending the data => Designed to handle burstiness
(1) **Codec:** Frame by frame natural basis

(2) **Content:** Variation in frame sizes -- oversized frames can be much larger!

Motivation
Unnecessary Pacing Latency

Pacer: Smoothly sending the data => Designed to handle burstiness

(1) **Codec:** Frame by frame natural basis

30 Mbps at 30 FPS, over 100 packets / frame

(2) **Content:** Variation in frame sizes -- oversized frames can be much larger!

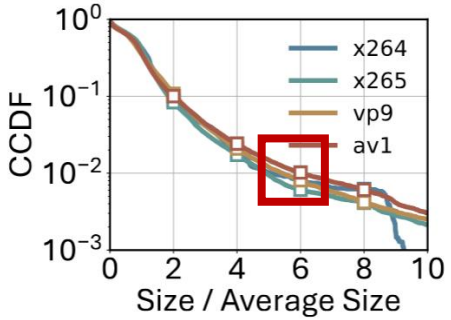


Figure 3: CCDF of encoded frame size

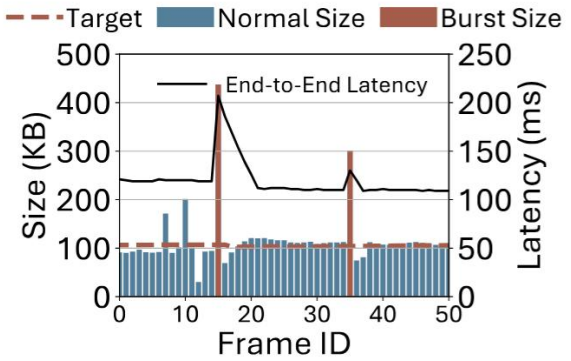
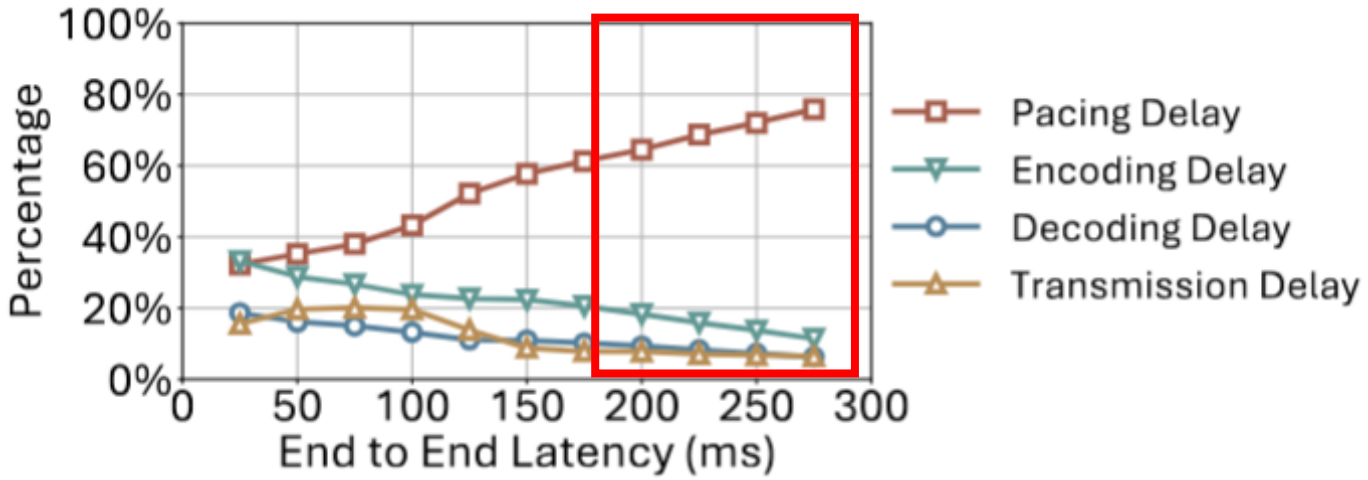


Figure 4: Latency impact by Oversized frames

Motivation
Unnecessary Pacing Latency

This pacing latency can be a significant contirbutor to high delay and stalls

(1) Evaluation on WebRTC



(2) Online Data on Douyin Cloud Gaming

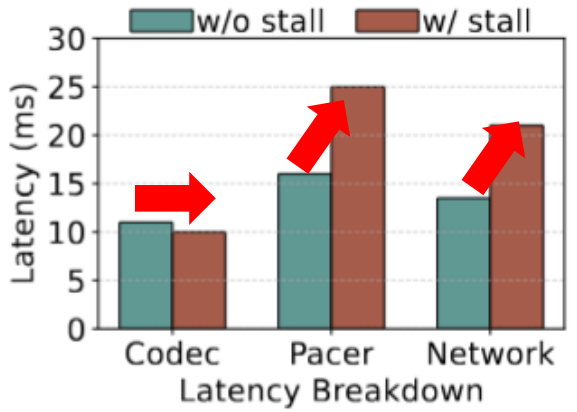
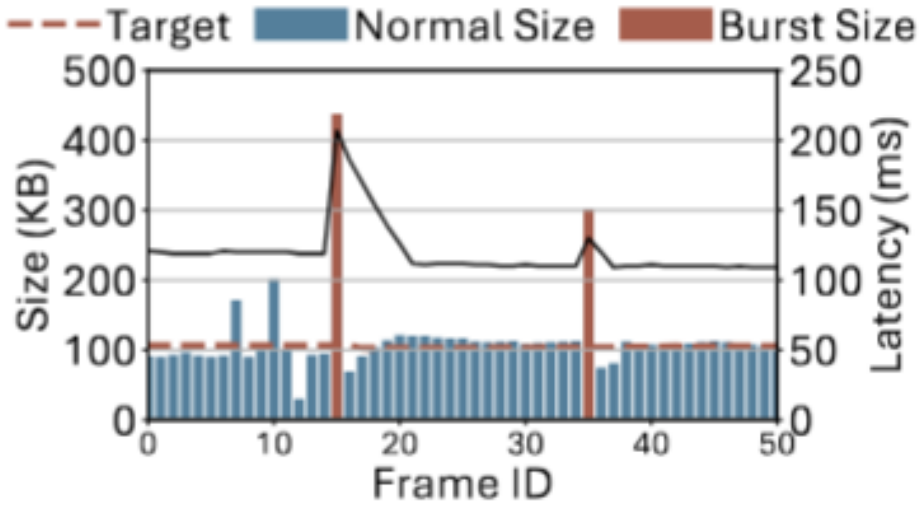


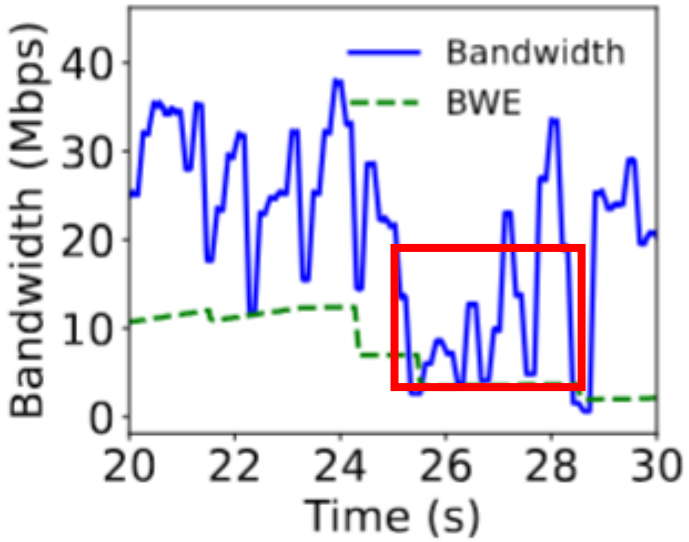
Figure 6: Latency breakdown

- **Pacing latency is dominating** when total delay > 200 ms.
- Other components contribute less than 1/3.
- From online data, Pacing is a contributor to video stall!

Motivation
Why pacer fails?

The burstiness of sending pattern is lack of management in short time scale.



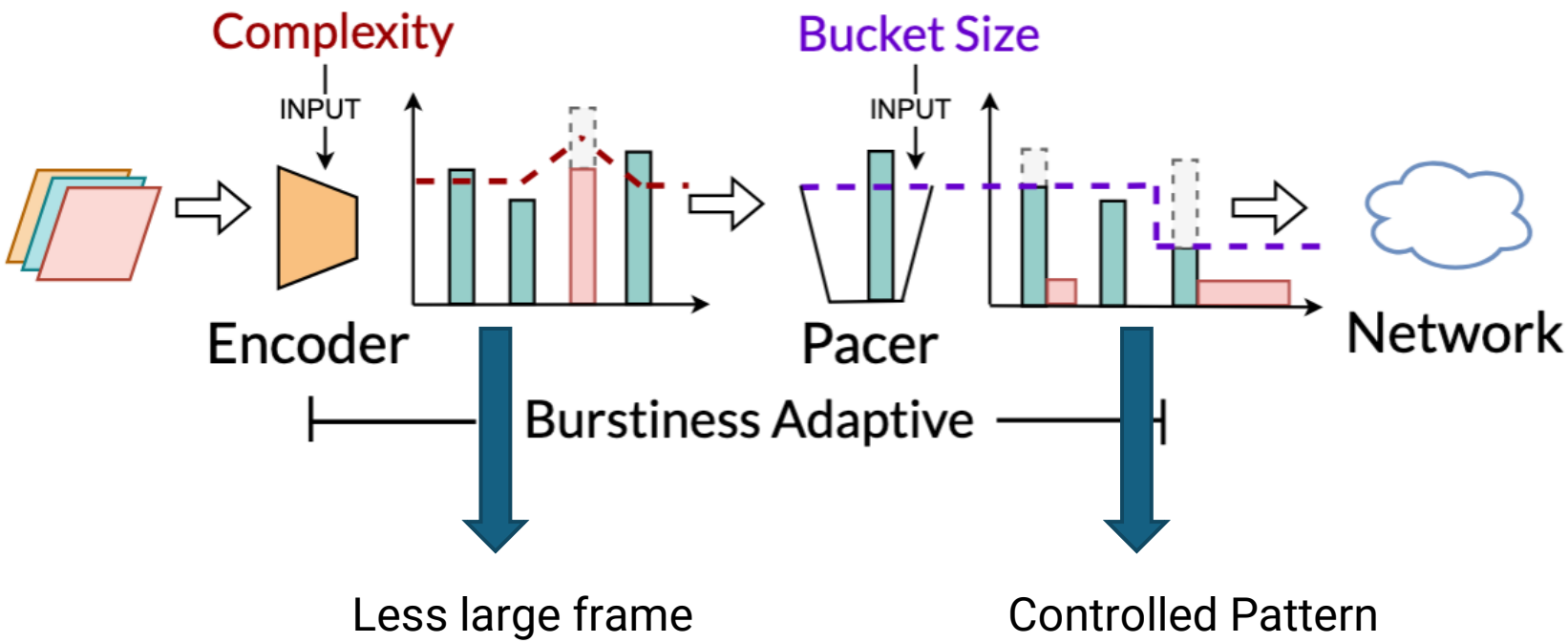


Enqueue: Oversized frames take longer to deliver

Dequeue: Bandwidth underestimation results in low pacing rate

Design ACE

Choices: Managing burstiness from both **enqueue** and **dequeue** perspectives.

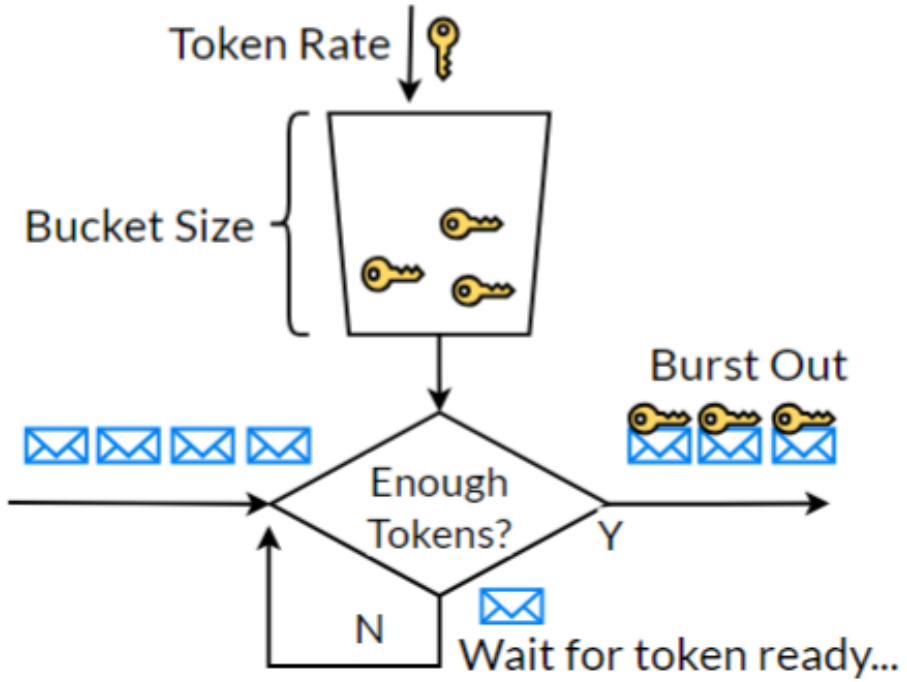


Design
Dequeue: Adaptive Token Bucket

Insight: Adaptively control the size of a token-based pacer.

- Adapting the bucket size to control bursts

- Packets without available tokens remain in the bucket, awaiting pacing,
Similar to pacer.



Design
Enqueue: Smoother Frame Sizes

Insight: Adapting the encoding complexity to enable a smoother stream.

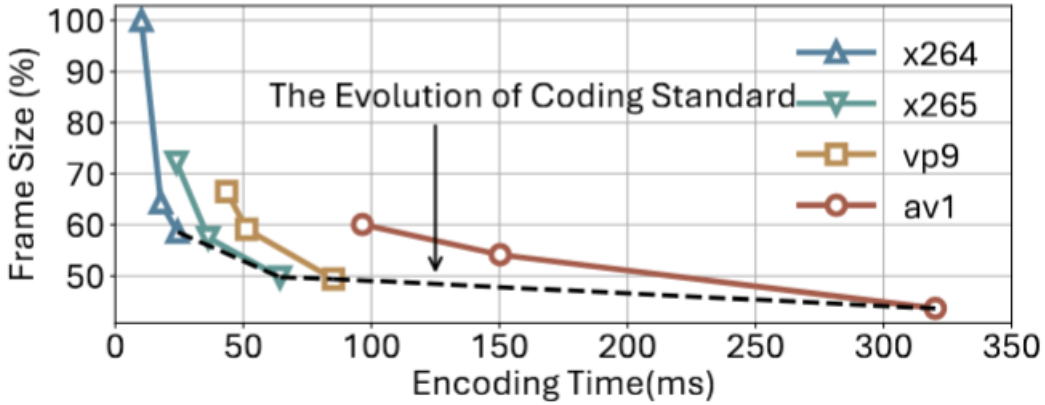
More advanced video coding technique

More complex coding

Higher coding delay

Smaller frame size

Lower pacing delay



Trade-offs are possible:

Sacrifice encoding time for transmission time during bad network conditions.

Design
Enqueue: Smoother Frame Sizes

Solution: Selecting the complexity level to minimize the total delay.

Gain function is defined to calculate the gain between the default comple
and the advanced complexity.

$$\text{Gain} = \Delta T_t - \Delta T_e = \frac{\Delta F}{\text{BWE}} - \Delta T_e$$

We can find the best complexity by maximize the gain function.

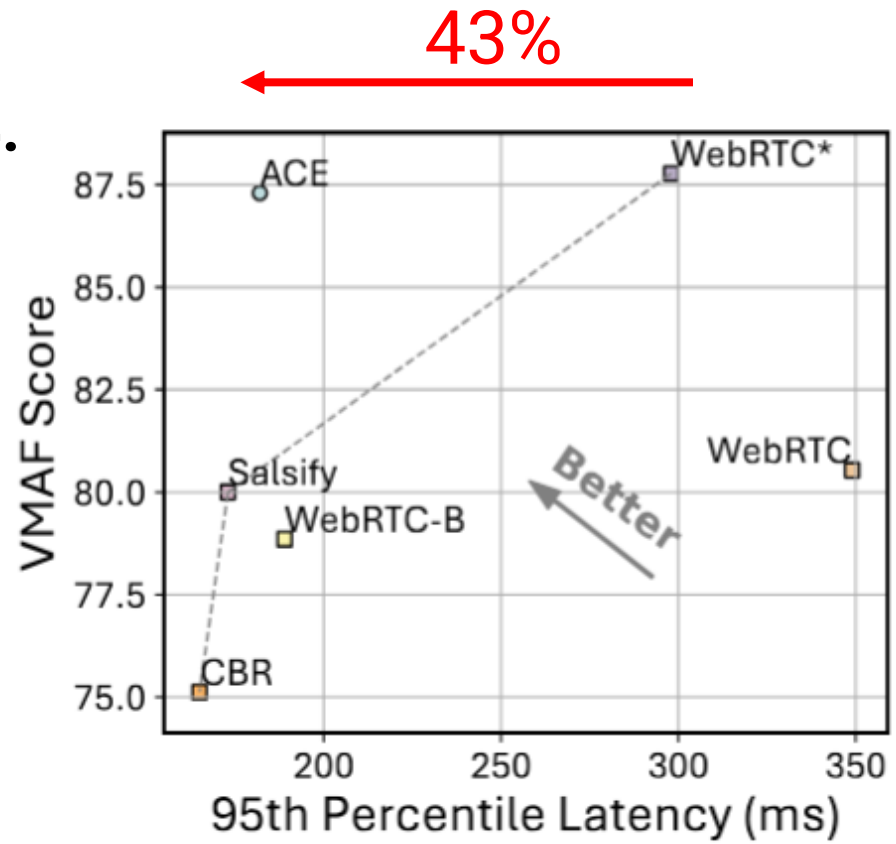
Challenge**: Get the predicted transmission time.**

We use the Sum of Absolute Transformed Differences (SATD), to estimate the size before encoding.

Evaluation
Emulation Results

Emulation Setup:
- Implementations on **WebRTC** and **libx264.**
- **Mahimahi** emulations
- Real world WIFI/ Cellular traces from Zhuge[SIGCOMM'22].

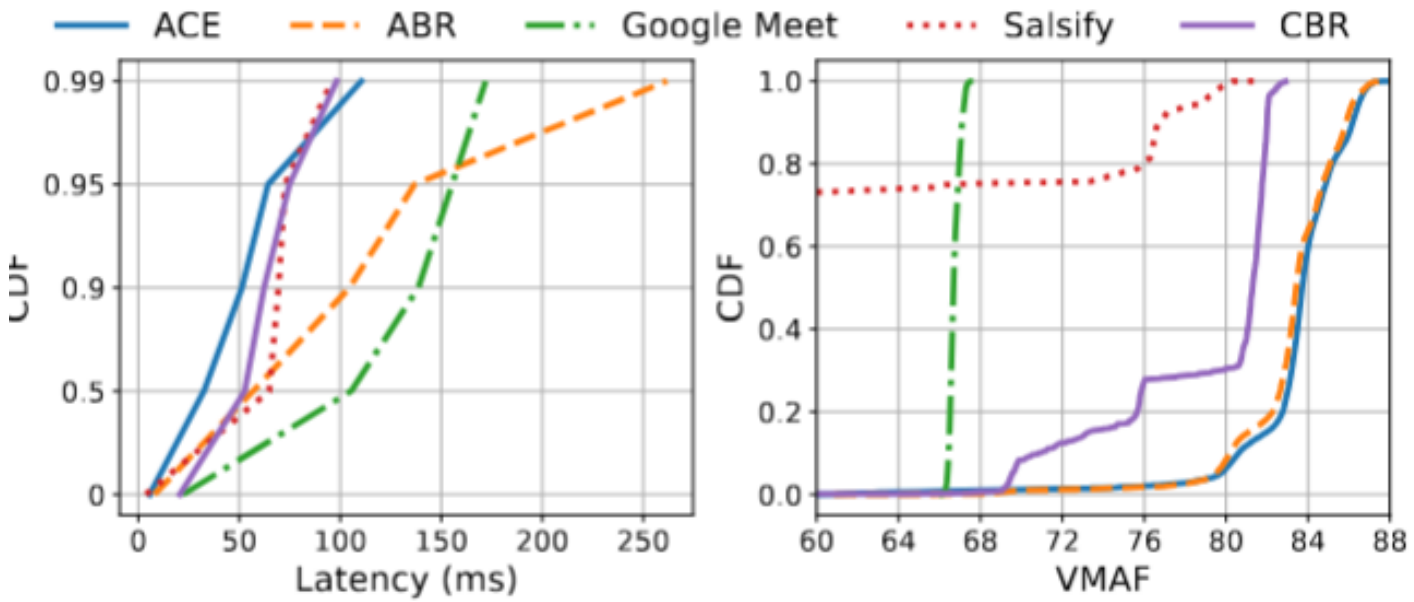
- Baselines:
  - Encoding approaches: CBR, VBR
  - RTC works: Salsify, WebRTC
- Metrics:
  - Vmaf Score for video quality
  - P95 latency
- More evaluations on loss rates, link utilizations, parameter sensitivities….



Evaluation
Real World Results

Experiment Setup:

- Environment: Eduroam wifi in campus
- **Sender**: Ubuntu Server, **Receiver**: Laptop connected to university wireless networ
- Baselines: Frontier in emulations + Google Meet



ACE outperforms

all the baselines in both latency and quality,

Evaluation
Real World Results

Production Application Test: Good Result in emulation!

Method	Stall Rate	Latency	Recv FPS
ACE-N	**2.89**	**136.97**	**56.80**
AlwaysPace	2.96	161.06	56.58
AlwaysBurst	13.37	323.34	53.79

Table 2: Performance on production application

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Summary

- We purposed ACE for burstiness management in RTC
- Choosing encoding complexity results in smoother bitrate.
- Token Bucket algorithm is provided to adaptively send bursts.
- Evaluation demonstrates satisfactory improvement in the end-to-end latency by up to 43% without quality degradation.

Thank you!
All credits go to my students

HKUST Spark Lab

https://zilimeng.com