JitBright: towards Low-Latency Mobile Cloud Rendering through Jitter Buffer Optimization

Yuankang Zhao, **Qinghua Wu**, Gerui Lv, Furong Yang, Jiuhai Zhang, Feng Peng, Yanmei Liu, Zhenyu Li, Ying Chen, Hongyu Guo, Gaogang Xie

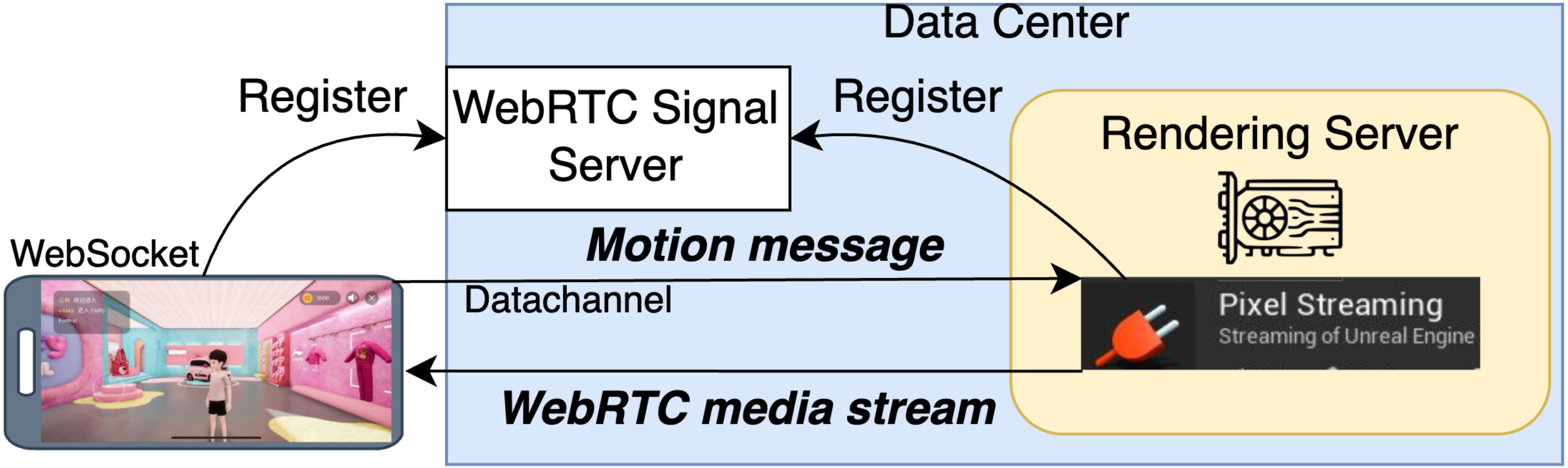
中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

中国科学院大学
University of Chinese Academy of Sciences

Alibaba

Background

WebRTC is used as the real-time cloud rendering system architecture, as it is widely supported by major web browsers and platforms

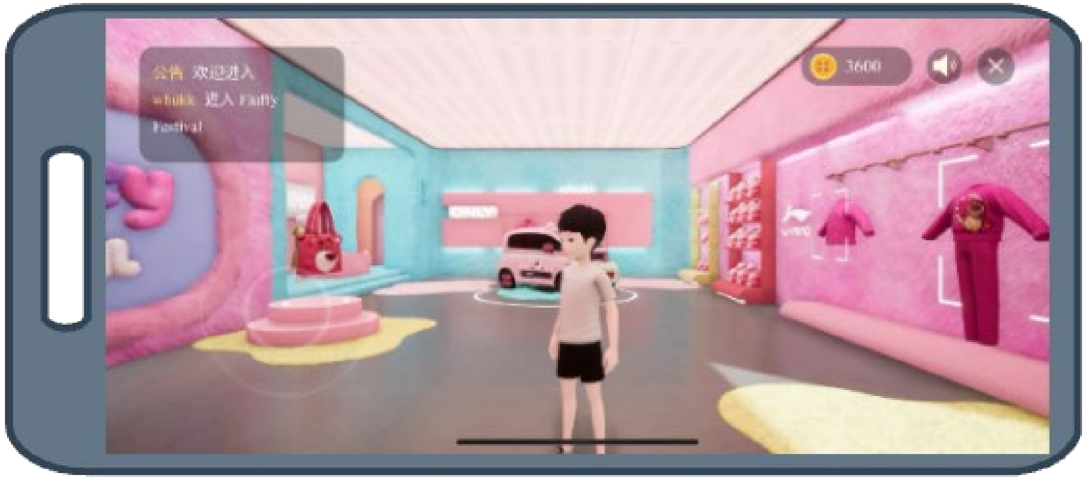


UNREAL ENGINE

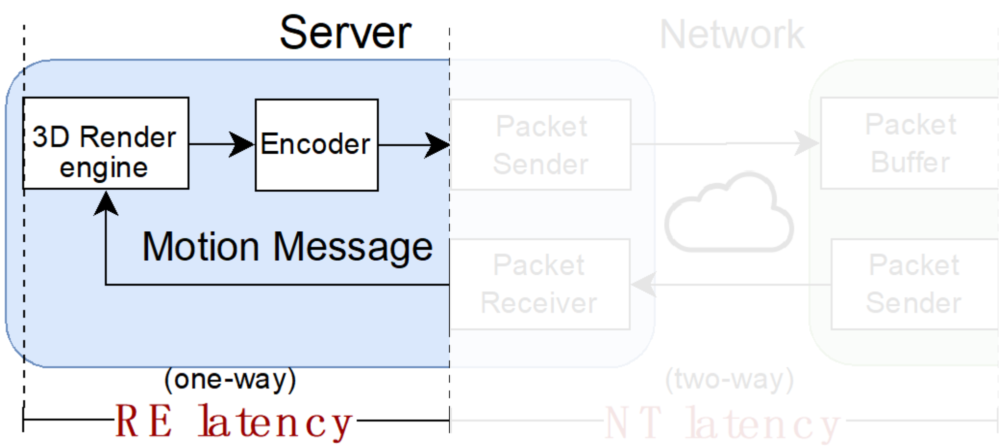
Background

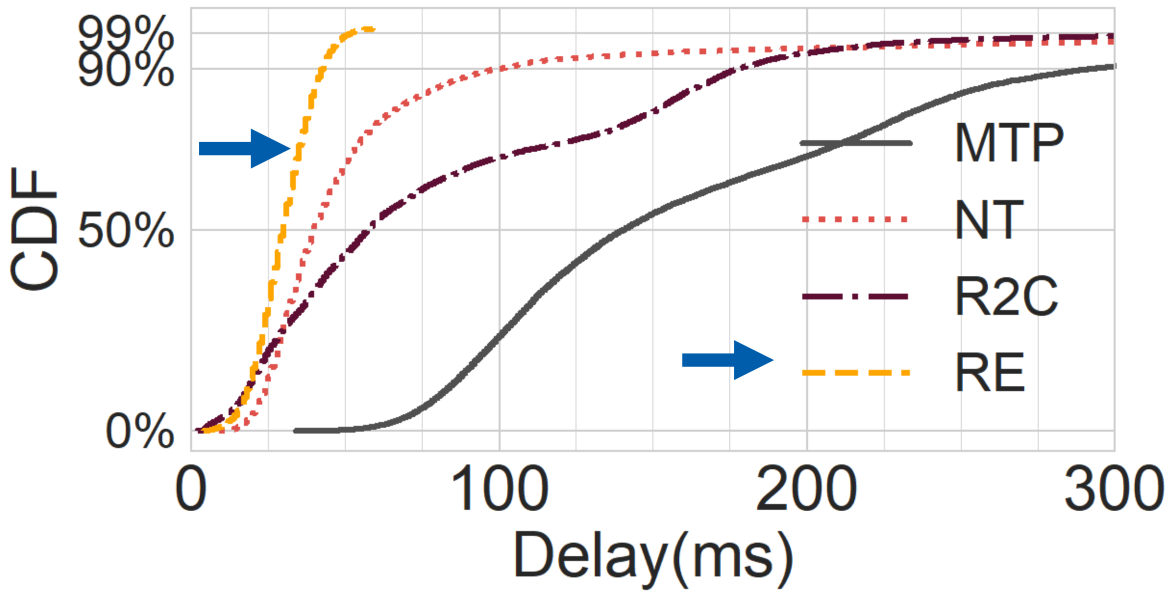
Latency and playback smoothness are critical

- Latency: Motion-to-photon latency
- Playback smoothness: frame stutter rate
    - i.e. Actual frame interval>2 original frame interval (2×16.6ms in 60fps)



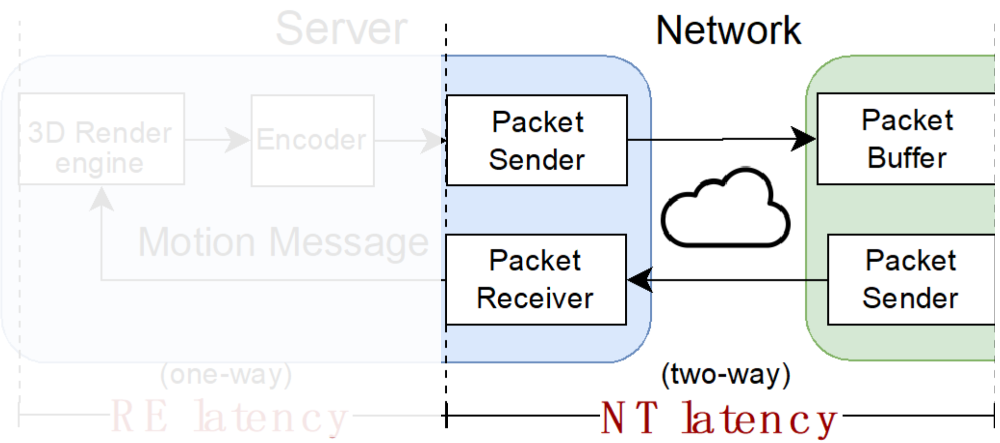
Latency Decomposition: RE Latency

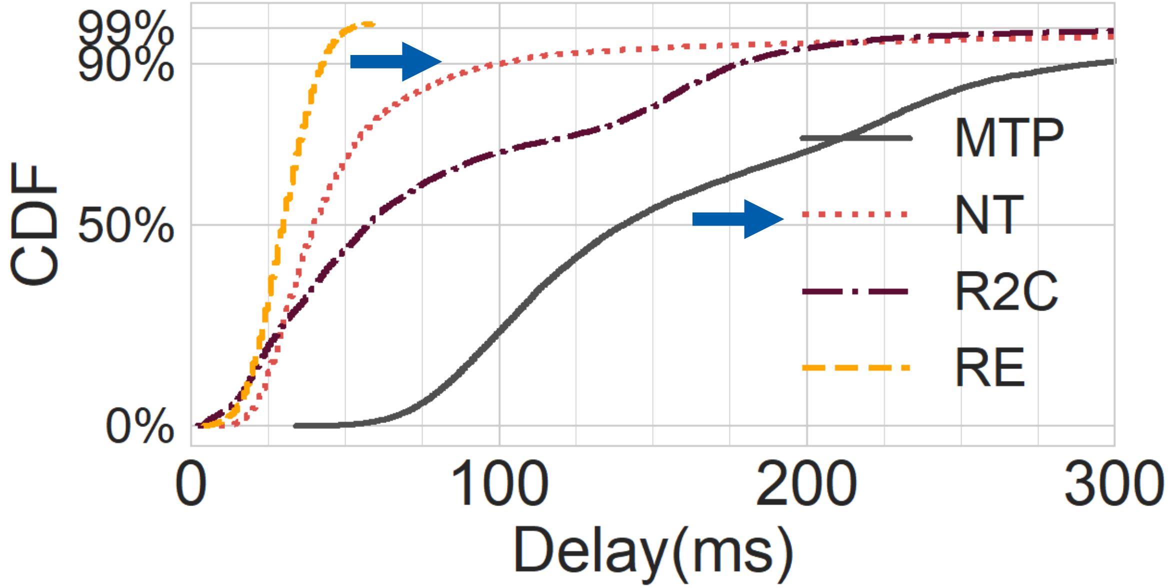




- RE latency: Rendering Enging latency (time for rendering and encoding a frame)

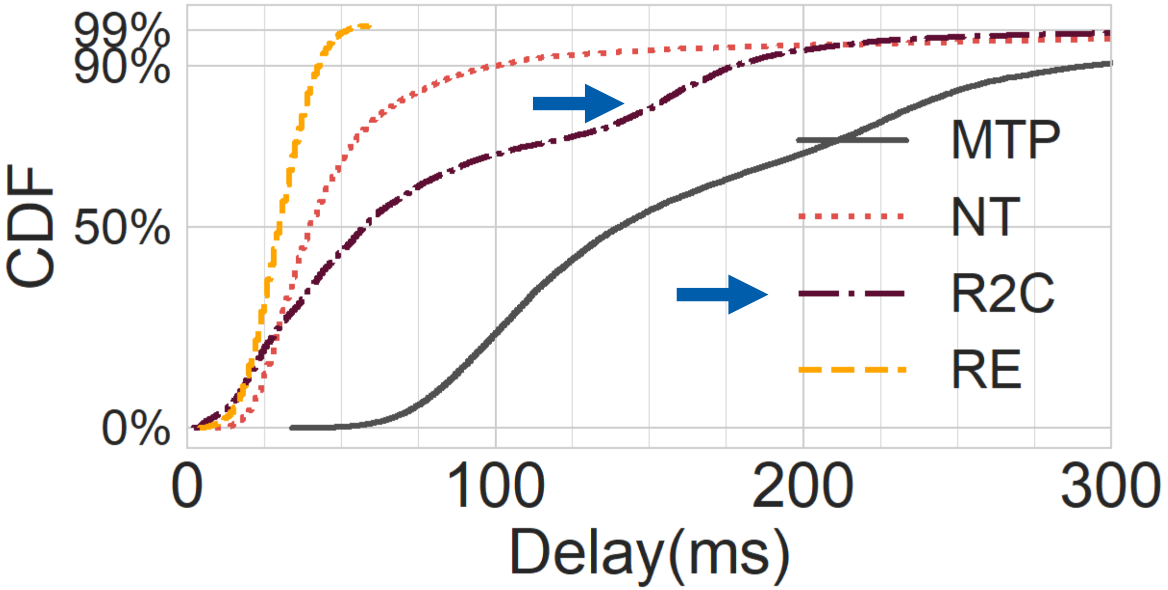
Latency Decomposition: NT Latency

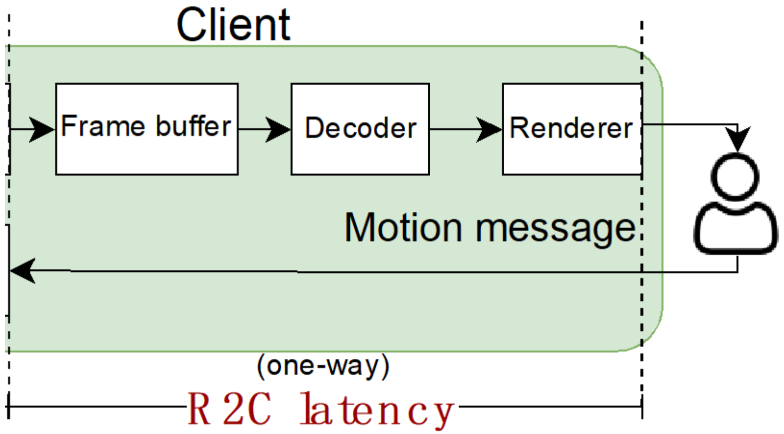




- RE latency: Rendering Enging latency (time for rendering and encoding a frame)
- NT latency: Network latency (two-way: the command upload + the frame download)

Latency Decomposition: R2C Latency

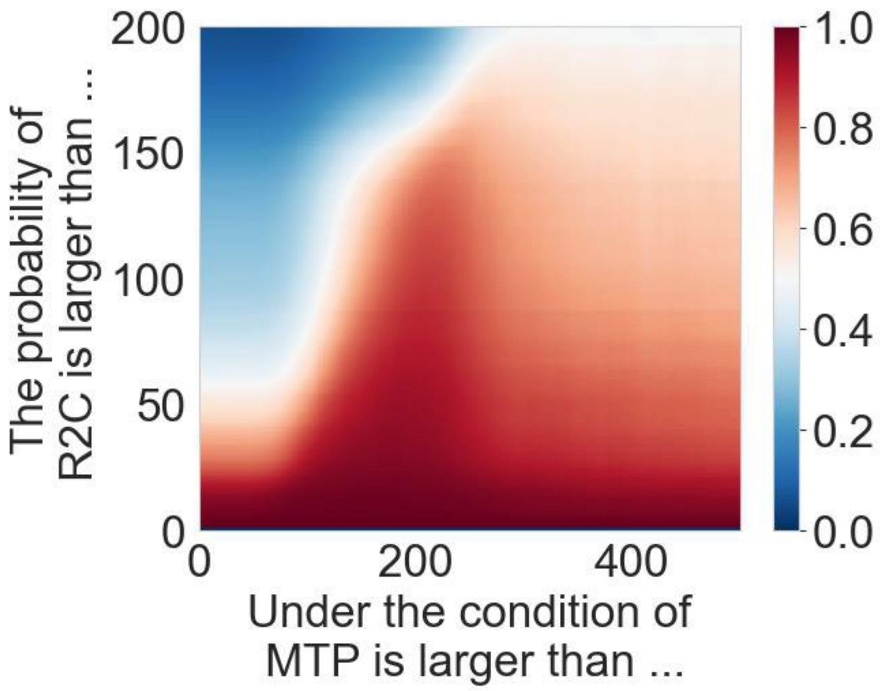


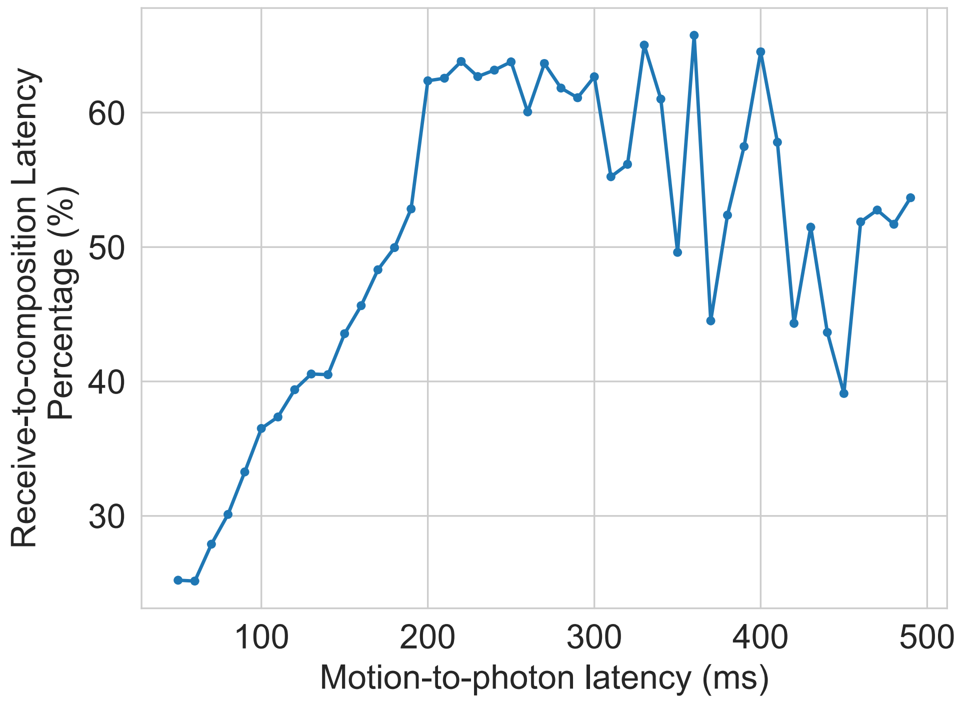


- RE latency: Rendering Enging latency (time for rendering and encoding a frame)
- NT latency: Network latency (two-way: the command upload + the frame download)
- R2C latency: Receive-to-composition latency

R2C Latency Dominates

- R2C latency accounts for the highest proportion of MTP latency in 57.2% cases.

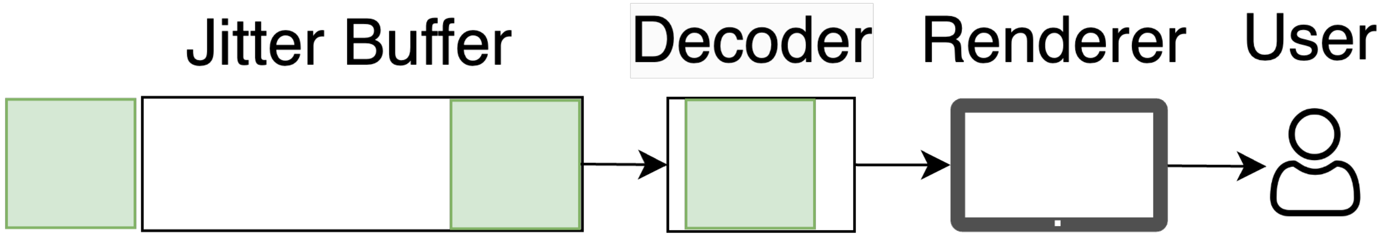




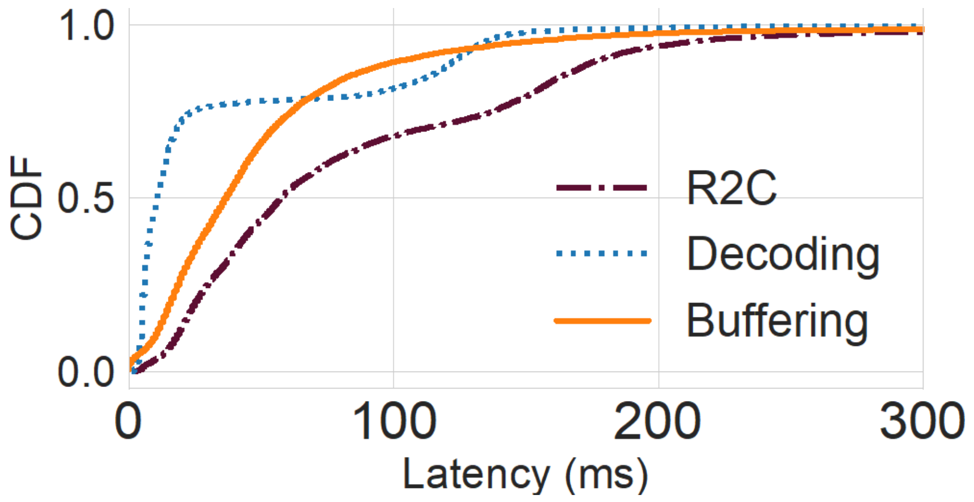
Unsatisfactory MTP latency is dominated by inflated R2C latency.

R2C Latency Decomposition

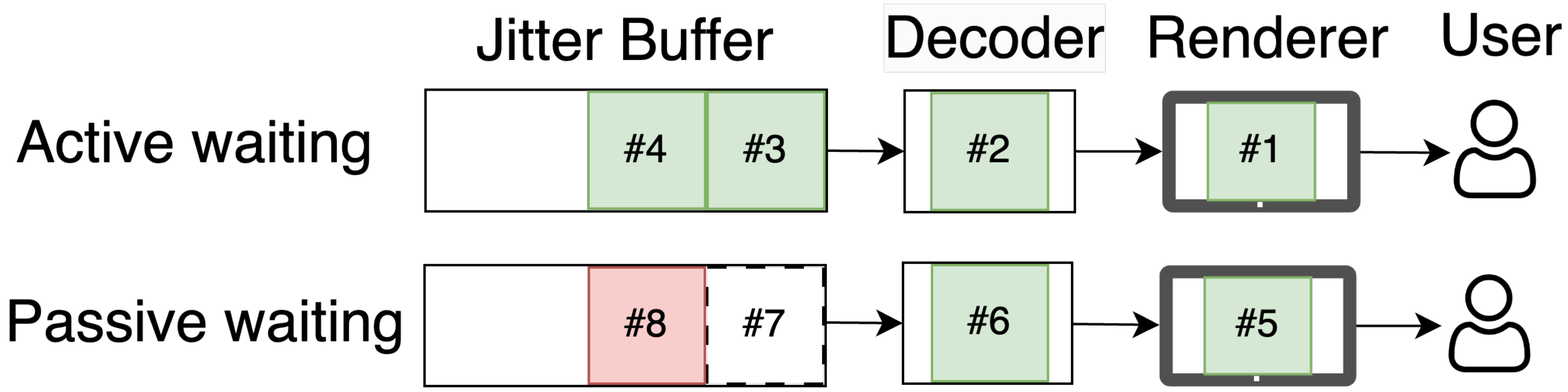
- ❖ R2C(Receive-to-Composition) Latency decomposition
  - R2C latency = Buffering + Decoding + Rendering
  - Buffering latency is the key factor impacting R2C latency in most (71.5%) cases.



The process of receiving to compositing a frame.



Behind the R2C Latency

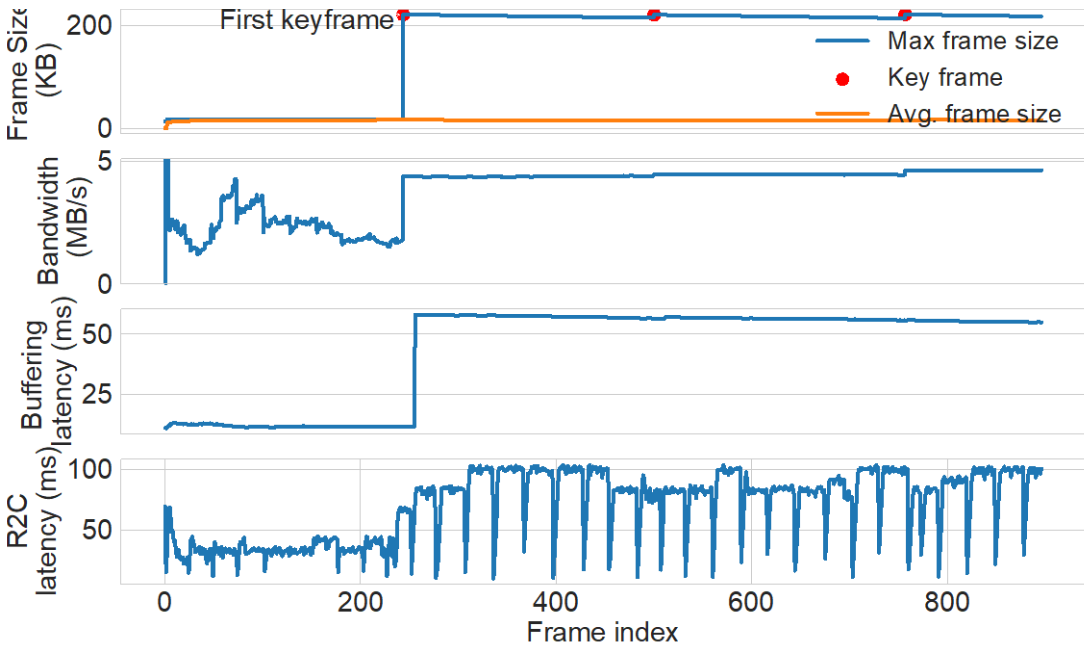


Understanding Buffering latency

- Active waiting: Frames **actively** waiting their scheduled decoding in the jitter buffer.
- Passive waiting: Frame **passively** waiting for their reference frame to arrive.

Observation

The default active waiting strategy optimizes for the worst case (receiving keyframe), while the keyframes are rare in cloud rendering scenarios.



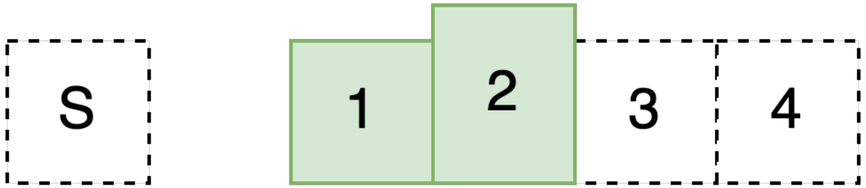
The default active waiting strategy $B \propto \dfrac{L_{max} - L_{avg}}{\hat{C}}$

JitBright: Reducing R2C latency

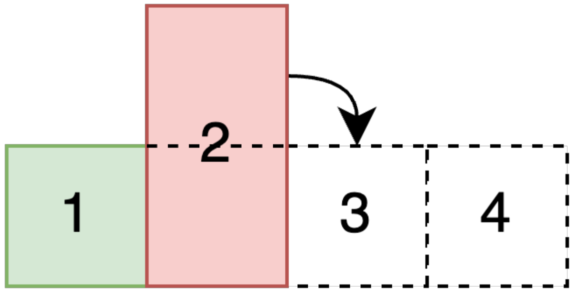
Optimizing the default active waiting strategy

Goal: balancing latency and smoothness

Example: A frame not satisfying the smoothness requirement event



S: Bandwidth in a time interval



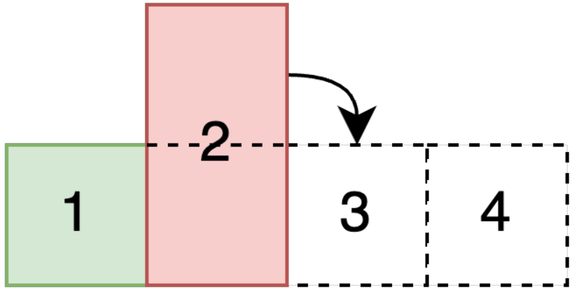
The extra part of frame#2 can be transmitted within 1 extra frame interval.

The extra part of frame#2 **cannot** be transmitted within 1 extra frame interval.

JitBright: Active Waiting Strategy

Optimizing the default active waiting strategy

- Goal: balancing latency and smoothness with an adaptive *gain*
- Idea: the *gain* should be proportional to the *probability* of a frame not satisfying the smoothness requirement.



$$B = gain * \frac{(L_{max} - L_{avg})}{\hat{C}}. \qquad gain \propto P(L - E(L) \geq S), \quad gain = Var(L)/S^2$$

$$P(L - E(L) \geq S) < P(|L - E(L)| \geq S) \leq Var(L)/S^2$$

JitBright: Passive Waiting Strategy

Reducing the passive waiting latency

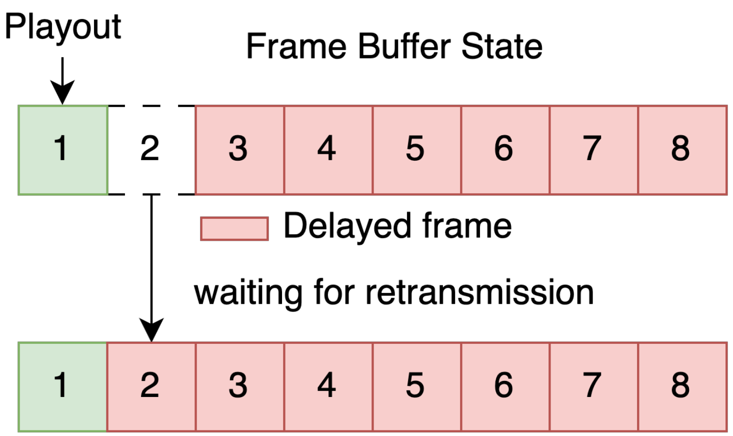
- Actively request a keyframe according to the cost functions.

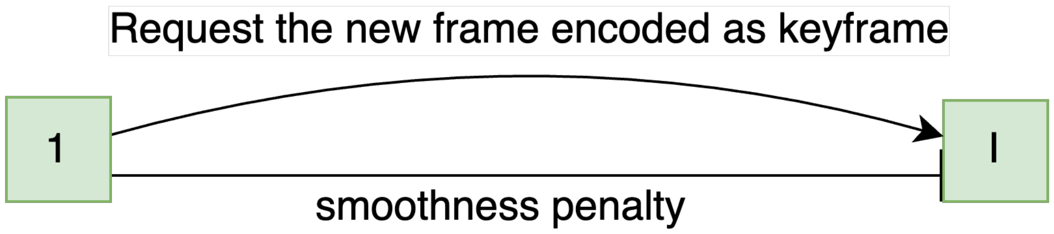
Cost Function of waiting for retransmission

$$U_{Wait} = T_{RTT} + Q \times \bar{T}_{Dec},$$

Cost Function of requesting a keyframe

$$U_{Req} = T_{RTT} + L_{max}/\hat{C} + \bar{T}_{Dec} + \lambda \times Q,$$





Waiting for retransmission（ RTT ）+decoding all frames in the buffer

Transmitting a keyframe – penalty for smoothness

Evaluation - Experiement Setup

Environments

- Local testbed
- Online evaluation in the wild with O(10000) users

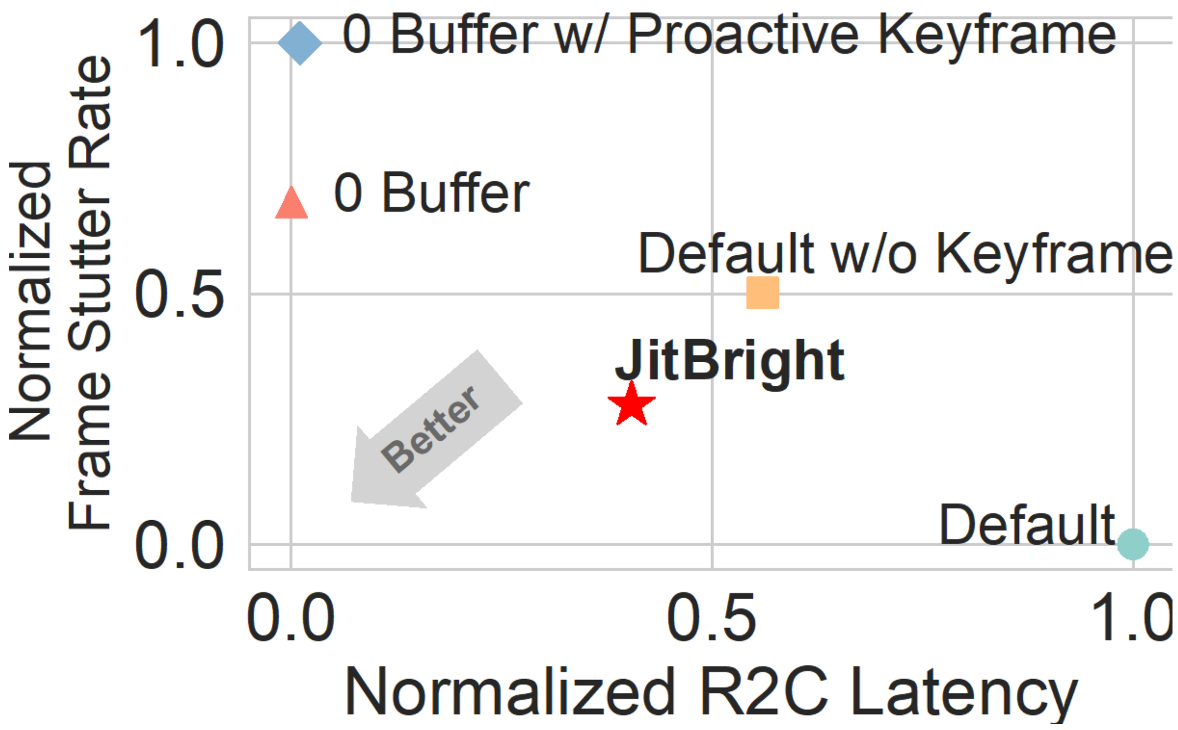
Baselines

- WebRTC Default: Include periodic (every 300 frames) keyframe
- Default without periodic keyframe
- 0-Buffer
- 0-Buffer with proactive keyframe request

Evaluation - R2C Latency and Smoothness

JitBright strikes an optimal balance between R2C latency and playout smoothness.

- stutter event: the actual frame interval > 2 original frame intervals



Evaluation - MTP Latency

The increased proportion for MTP latency satisfies the latency requirement (150ms in our case) in the online evaluation.

	WiFi	4G	5G
High	+15%	+27%	+9%
Mid	+13%	+16%	+20%
Low	+23%	+6%	+15%

Summary

- We identified that R2C delay is the primary factor influencing MTP latency in Mobile Cloud Rendering.
- We proposed JitBright to reduce the R2C latency by reducing active waiting and passive waiting latency.
  - JitBright increases the proportion of sessions meeting MTP latency requirements (i.e., <150 ms) by **15%-23%** in WiFi, **9%-20%** in 5G, and **6%-27%** in 4G networks.
- JitBright is lightweight and deployable in the WebRTC framework.

Thanks!

Q&A

中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY ,CHINESE ACADEMY OF SCIENCES

中国科学院大学
University of Chinese Academy of Sciences

Alibaba