Predictable Real-Time Video Latency Control with Frame-level Collaboration

Jiaxing Zhang, Qinghua Wu , Gerui Lv, Wenji Du, Qingyue Tan,
Wanghong Yang, Kai Lv, Yuankang Zhao, Yongmao Ren, Zhenyu Li, Gaogang Xie
zhangjiaxing1998@gmail.com

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

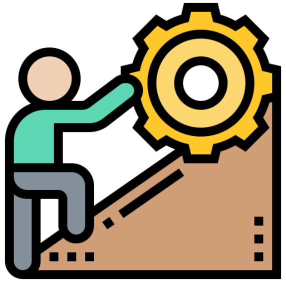
Emerging real-time video (RTV) services

Transitioning from **passive viewing** to immersive **interaction**.

Interactive Experience

Need

Low Latency (< 100 ms)



Challenging

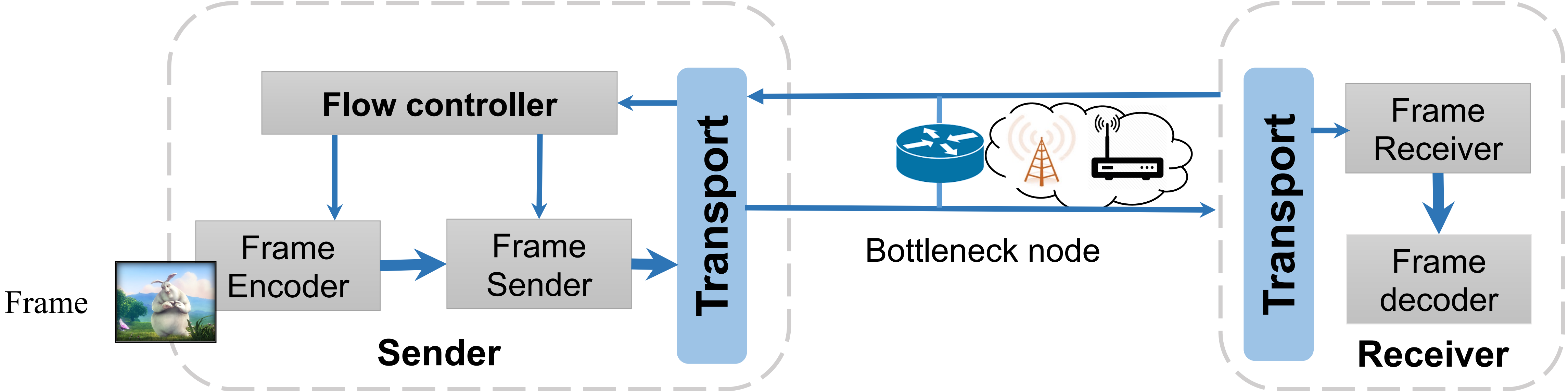
RTV experience over 10 million users

	Wired	Wi-Fi (5 G)	Wi-Fi (2.4 G)	LTE/5G
Frame (> 100ms)	1.28%	3.42%	4.9%	2.1%
User (5% > 100ms)	2.9%	5.7%	**29.1%**	**15.6%**

Proportion of frames experiencing delay > 100 ms and proportion of users with over 5% of frames delayed over 100 ms

High frame delay remains a persistent pain point for users.

RTV system overview

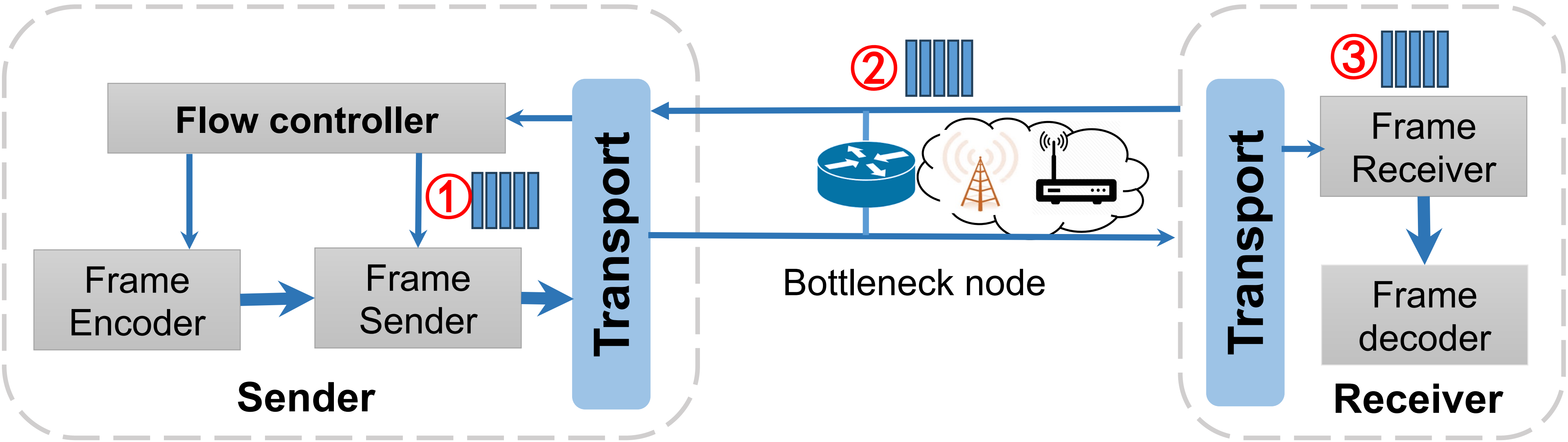


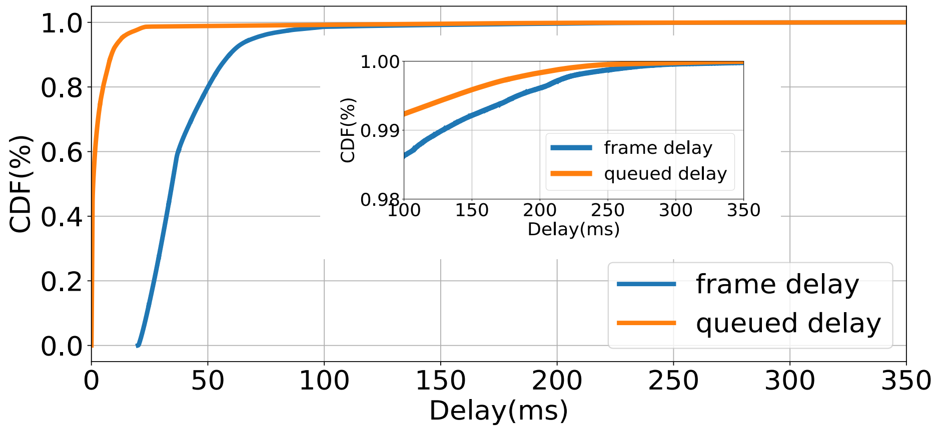
Input

Network Bandwidth > Bitrate → Low Latency

What factors contribute to the high frame delay?

Queues

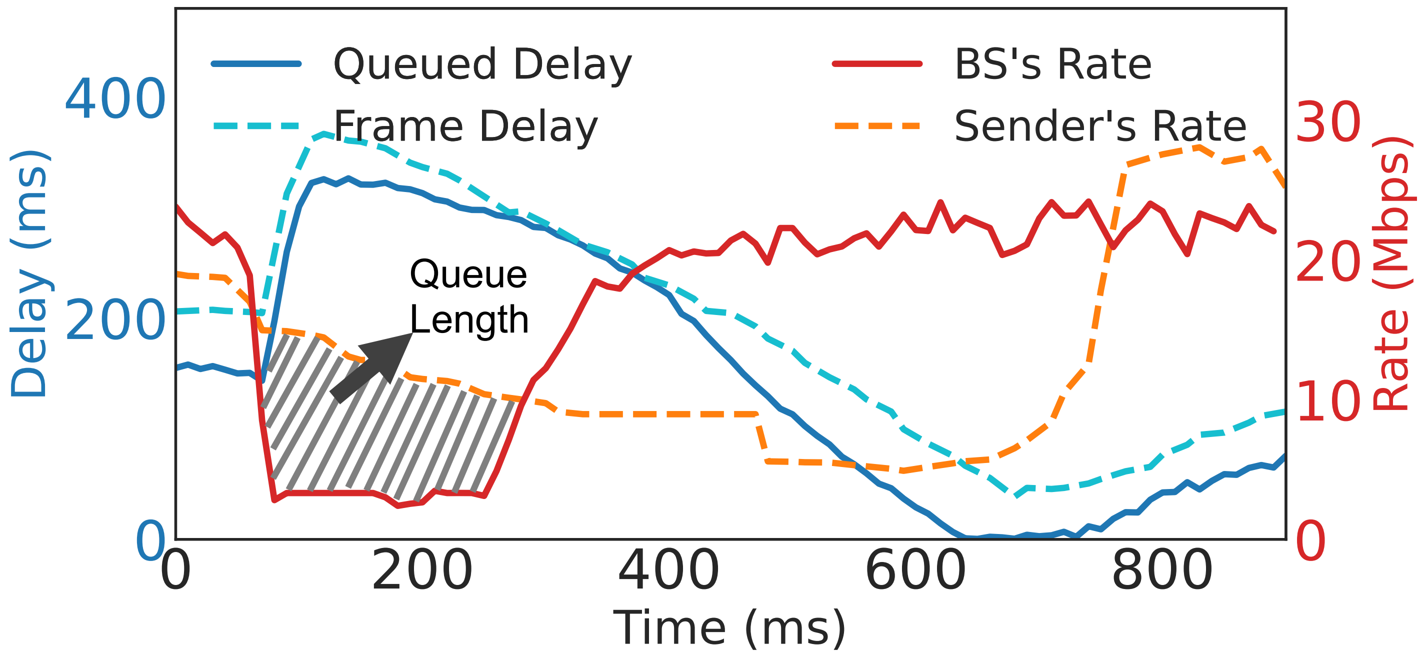




Queue 2 dominates high frame delay.

How queue 2 arises?

Sender overshooting: **Delayed** network bandwidth awareness (reactive)

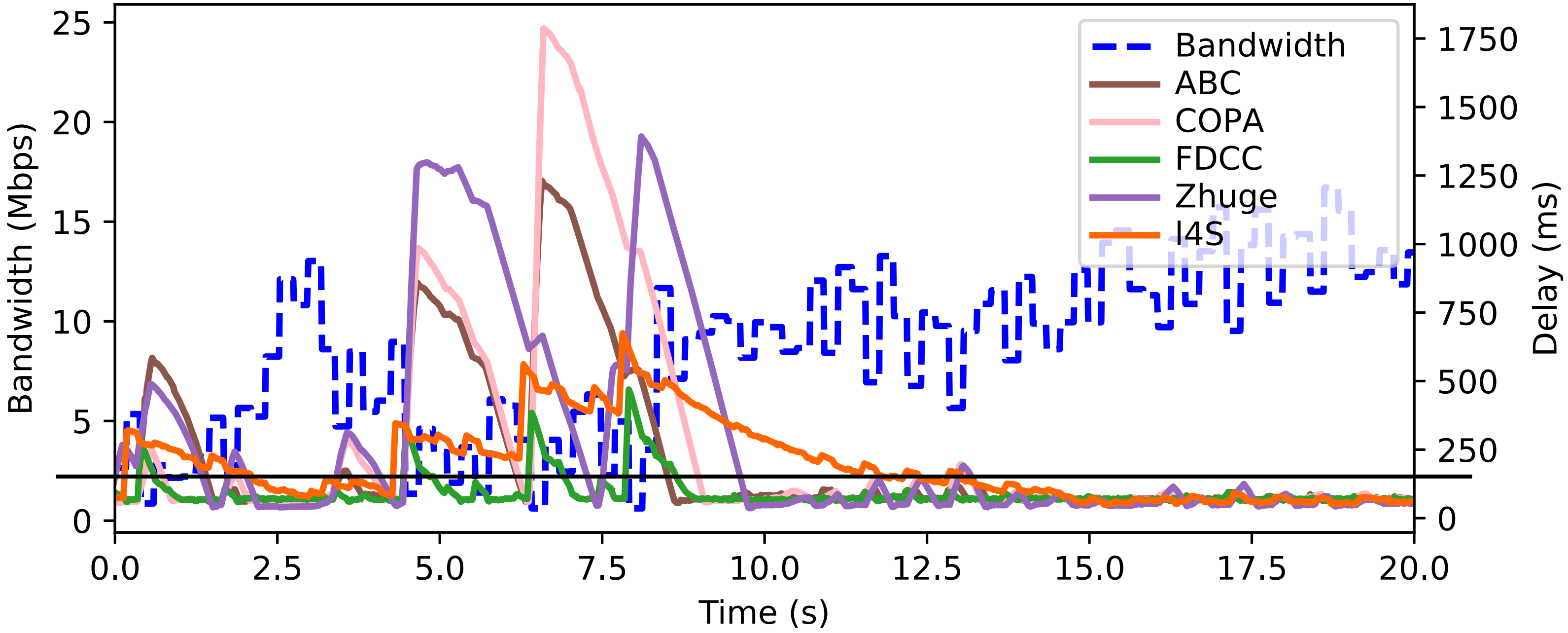


Existing Solutions： **“Faster”** bandwidth awareness
ABC(ECN), Zhuge, L4S (ECN)

Existing solutions attempt

How about **"fastest"**?

FDCC：Bottleneck nodes directly feedback the actual Network Rate. (fastest)

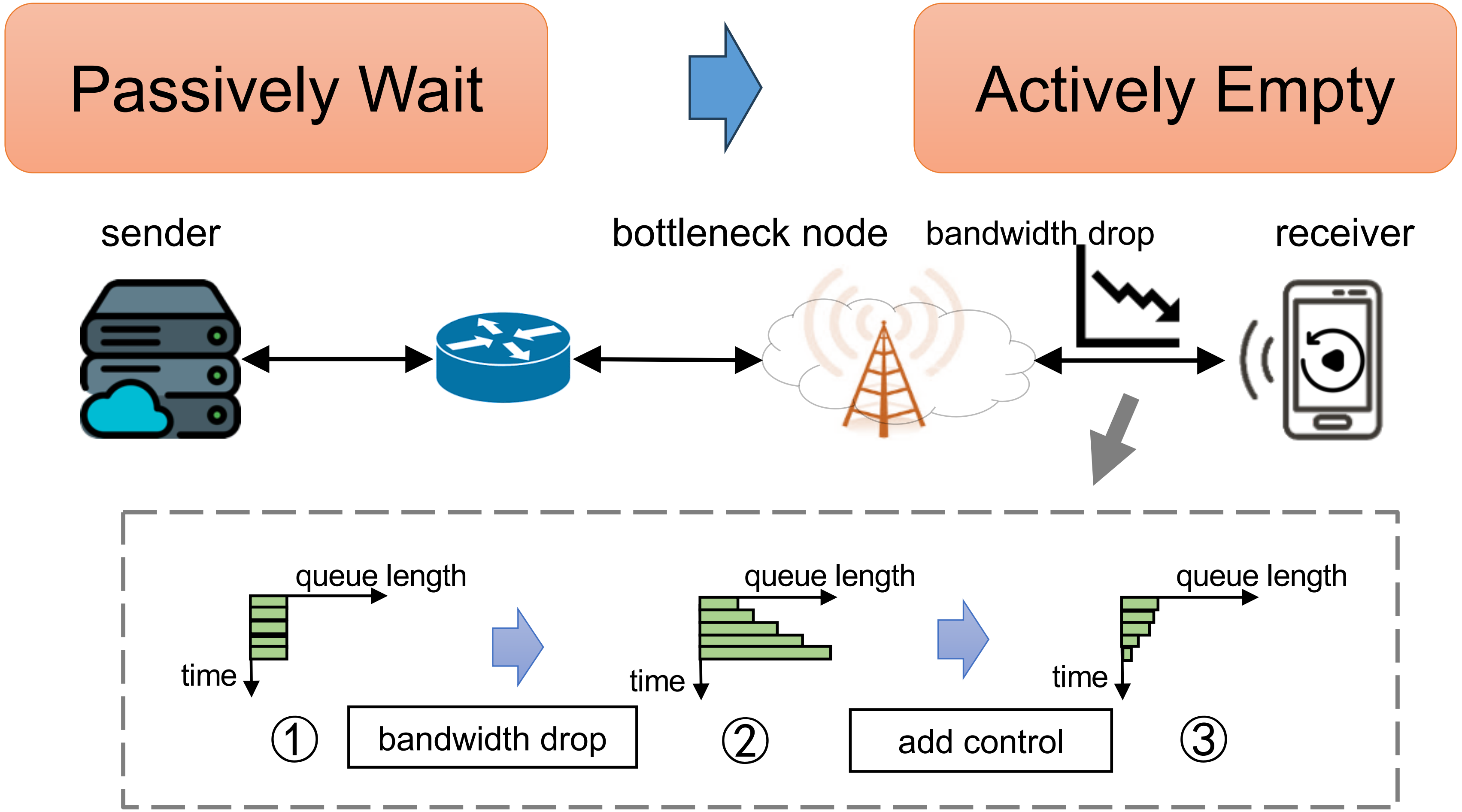


Even "fastest" is not enough 😭

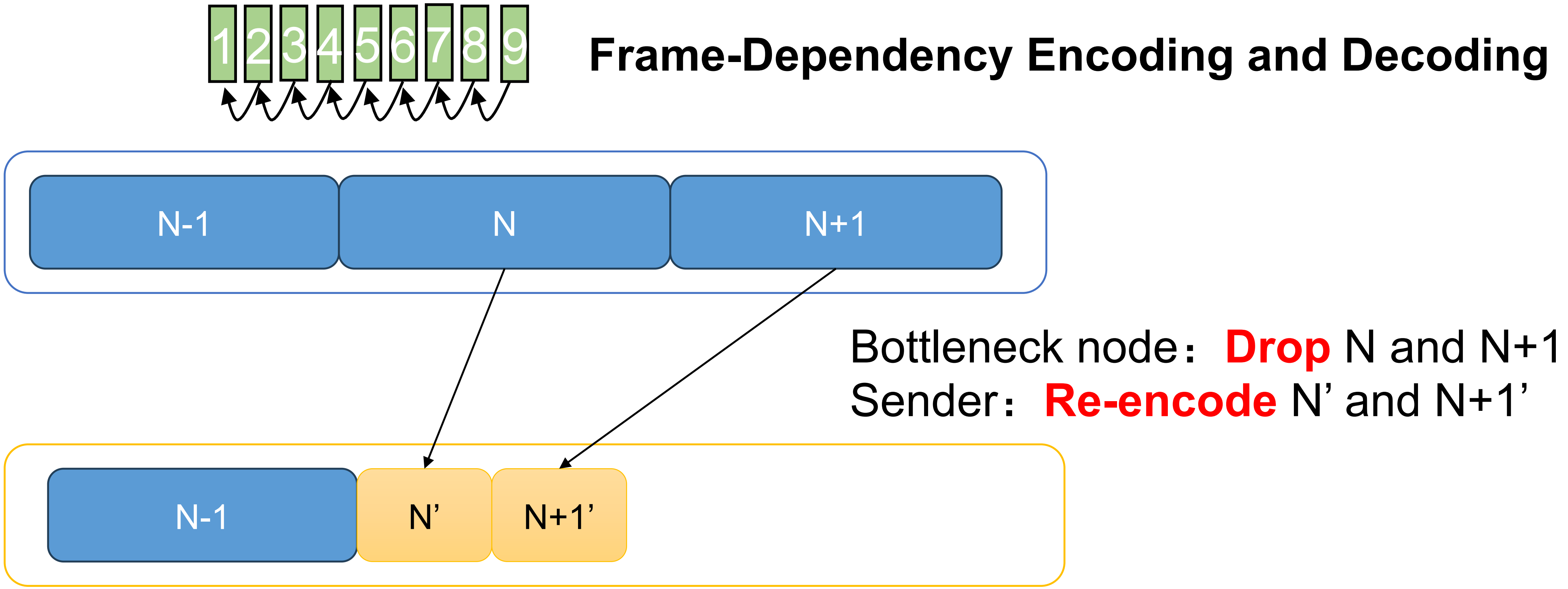
"Feedback" still need time, Queue 2 will also form.

Our insight: actively drop queue 2

The emerging IETF SCONE and MoQ standards provide opportunities for **controlling the bottleneck node**.



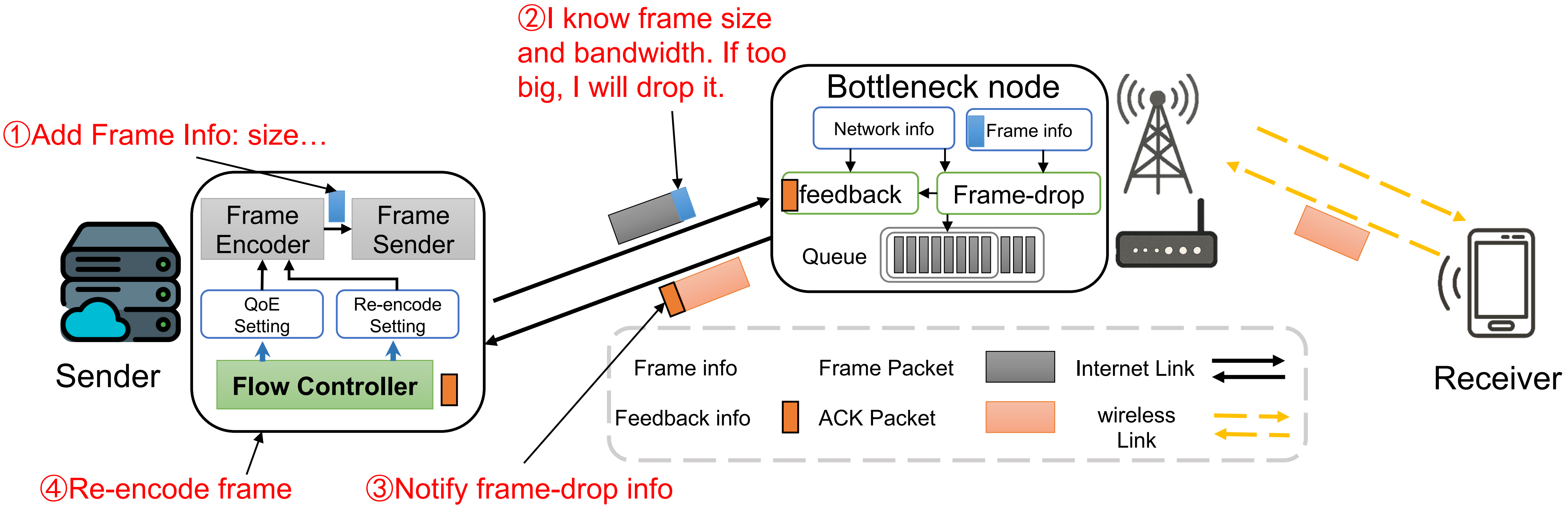
How to empty queue 2？



Dropping only frames N and +1 causes the sender to retransmit them, highlighting the need for **collaboration**.

Co-RTV design

- Bottleneck node： active **frame-drop**
- Sender: collaborative **re-encoding**

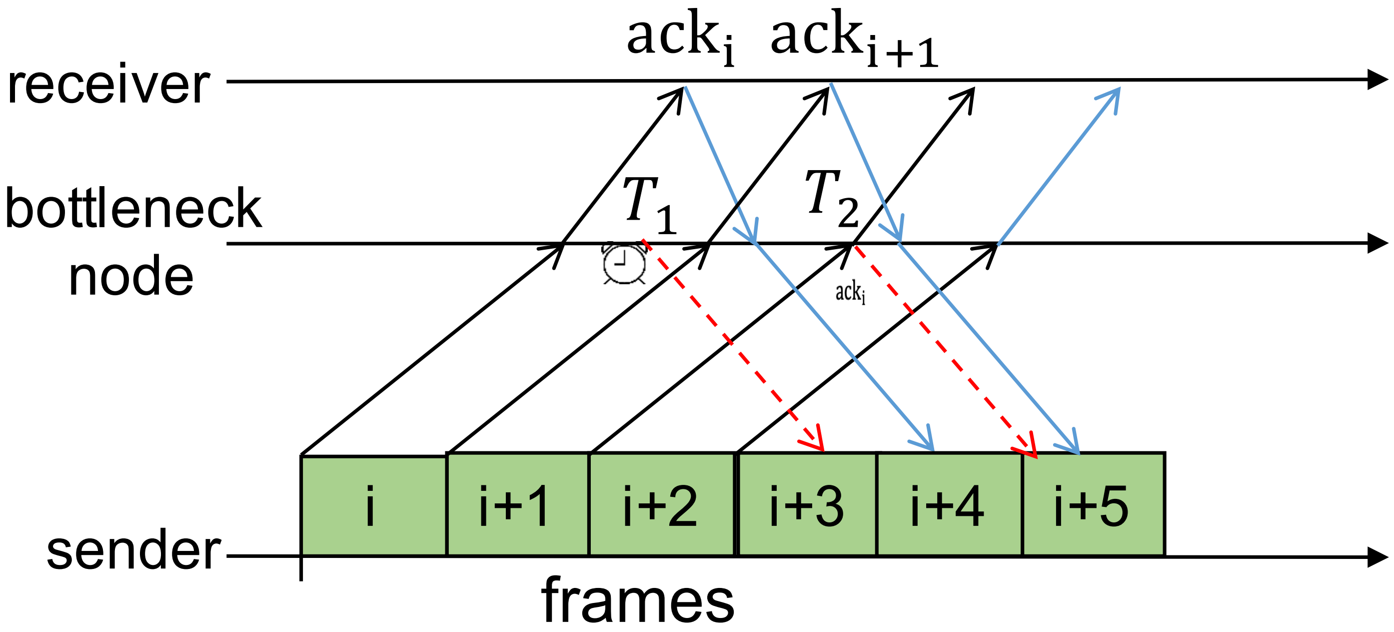


Co-RTV design

- Bottleneck node： active **frame-drop**
  - Drop timing: when to drop frames
  - Drop selection: which frames to drop
- Sender: collaborative **re-encoding**
  - re-encoding strategy (frame rate, latency and quality)

Drop timing: when to drop frames

Bandwidth begins to decrease at **T_1**. Frame i+2 arrives at **T_2**.



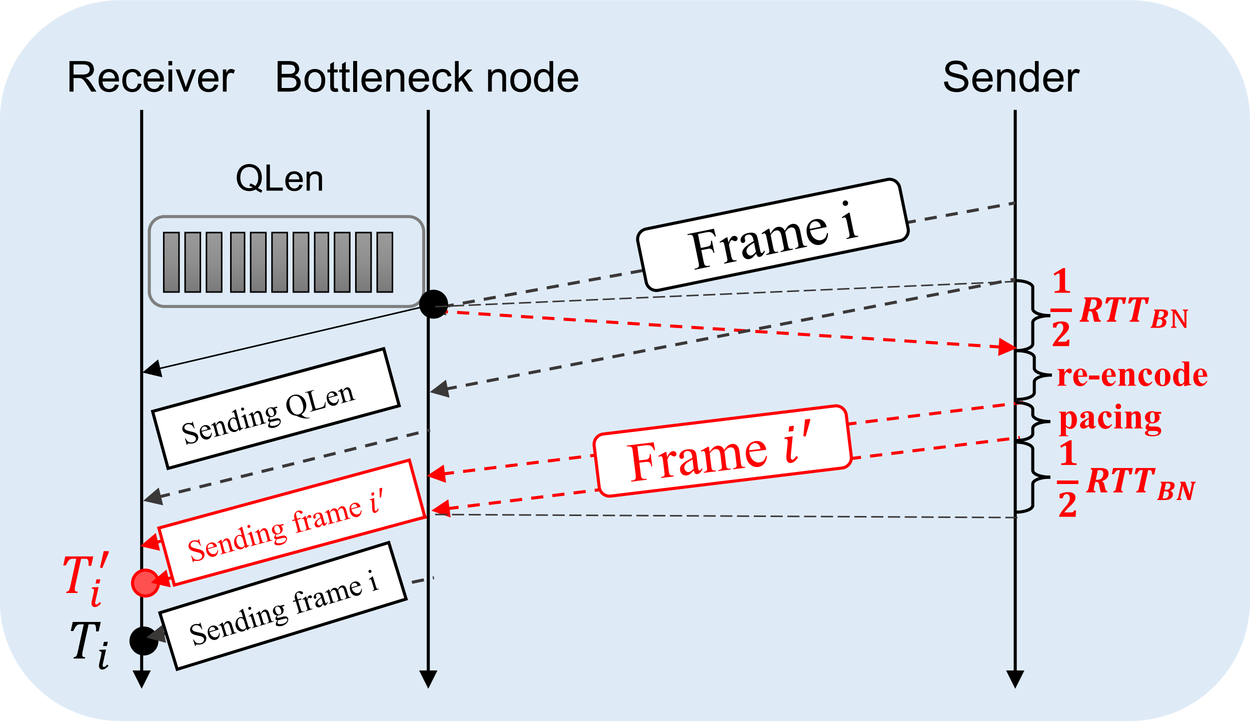
Drop frame i+2 at T_1 rather than T_2. More **timely**!

Receiving a re-encoded frame i+2 will continue forwarding.

Drop selection: which frames to drop

- For frame i, calculating the **gain** of frame-drop： **latency and quality**

$$Gain_i = L_{di} - Q_{di}$$



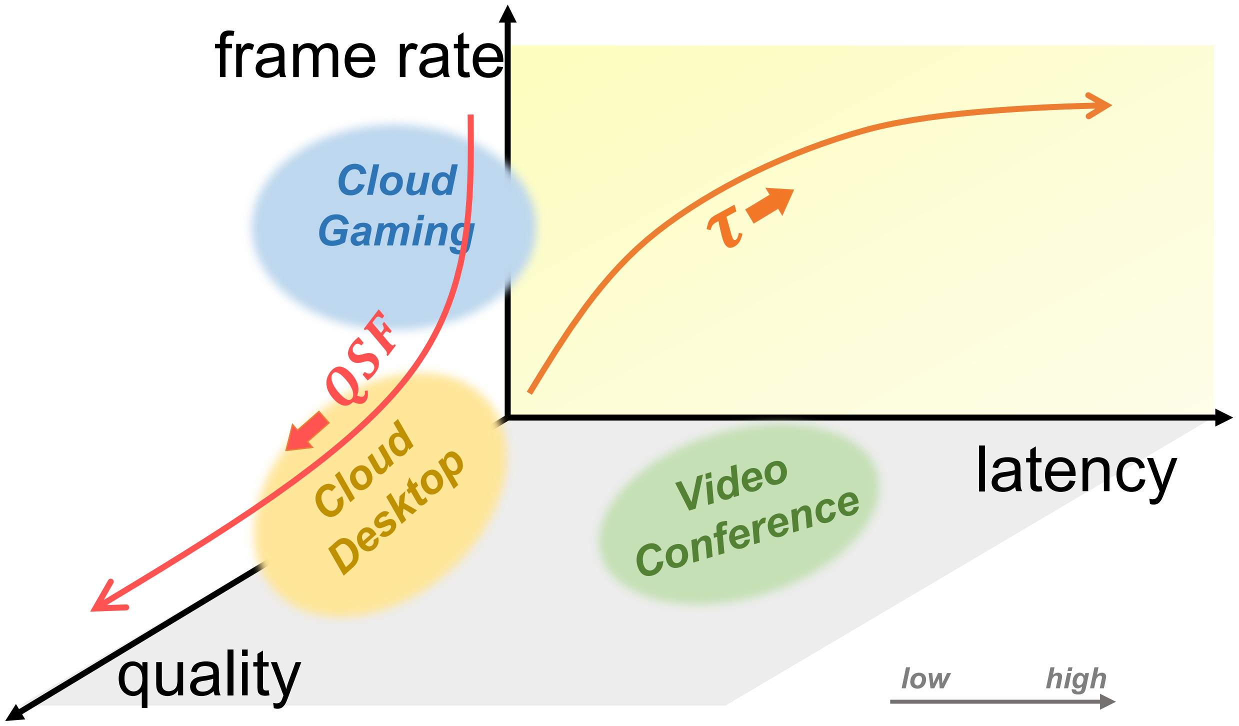
$$L_{di} = T_i - T_i'$$

$$Q_{di} = \alpha \frac{Bw_{old} - Bw}{Bw} \frac{1}{FR}$$

Further computational details are provided in the paper.

Re-encoding strategy

Sender re-encodes subsequent frames at a lower bitrate from feedback. How should it trade off **frame rate, latency and quality**?



Two Level

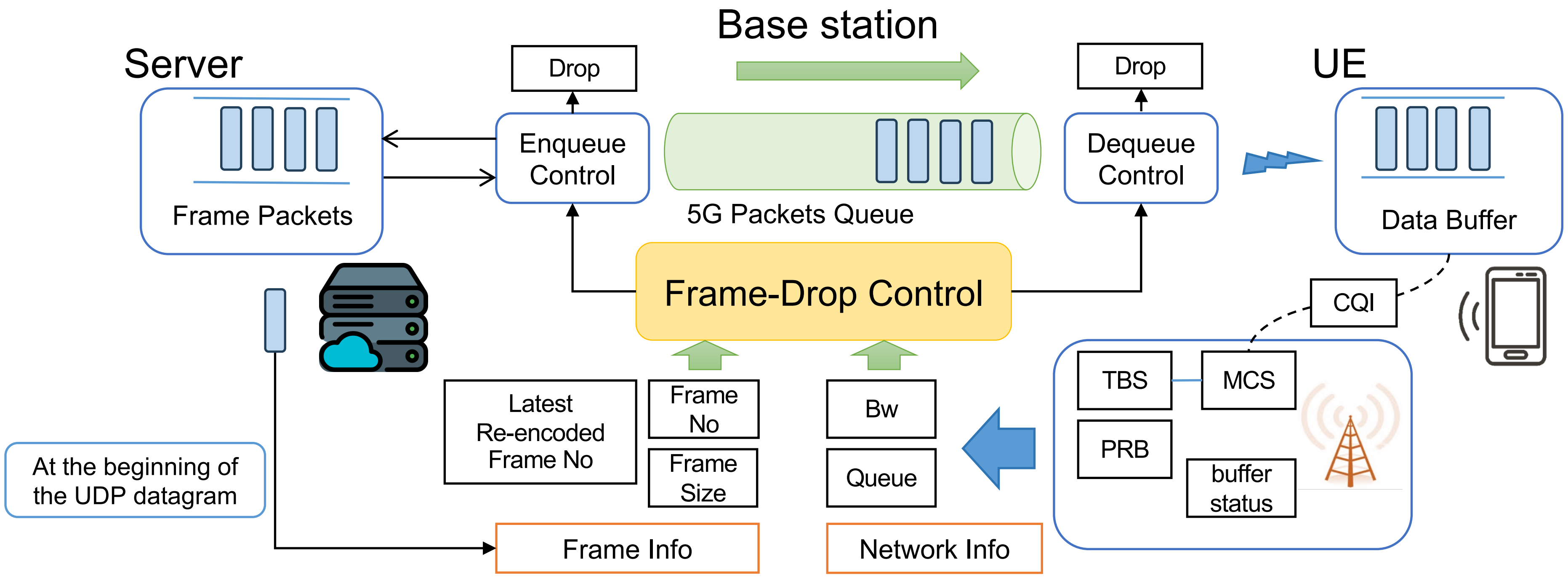
1. (frame rate x quality) and latency

$$R = SndRate - \frac{QLen}{\tau}$$

2. frame rate and quality

$$N_s = QSF \cdot (\tau \cdot FR + N_d)(1 - \frac{R}{SndRate})$$

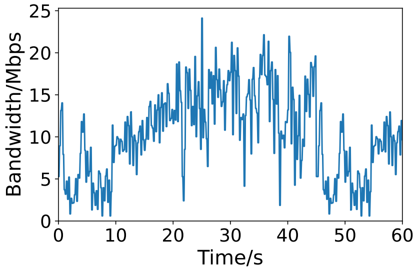
Co-RTV implementation in 5G



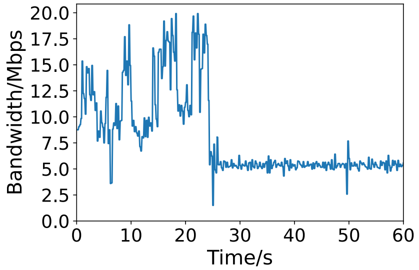
We use **QUIC** instead of **TCP**, as Co-RTV requires **semi-reliable** delivery.

Evaluation setup

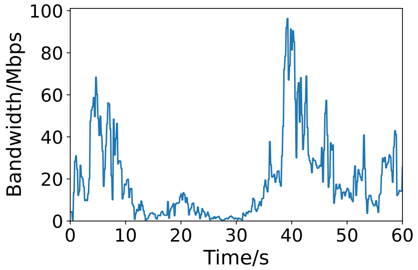
- Baselines: COPA, ABC, Zhuge, FDCC
- Metrics: Frame delay, Frame size
- Trace-driven emulation and 5G testbed evaluation



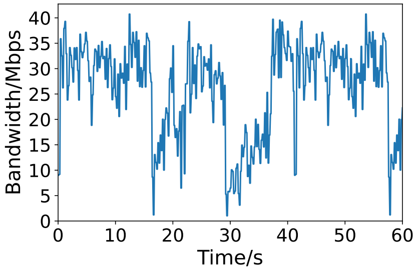
Mall



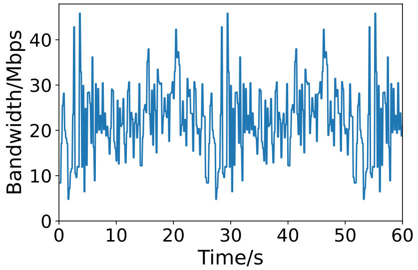
Office



Classroom

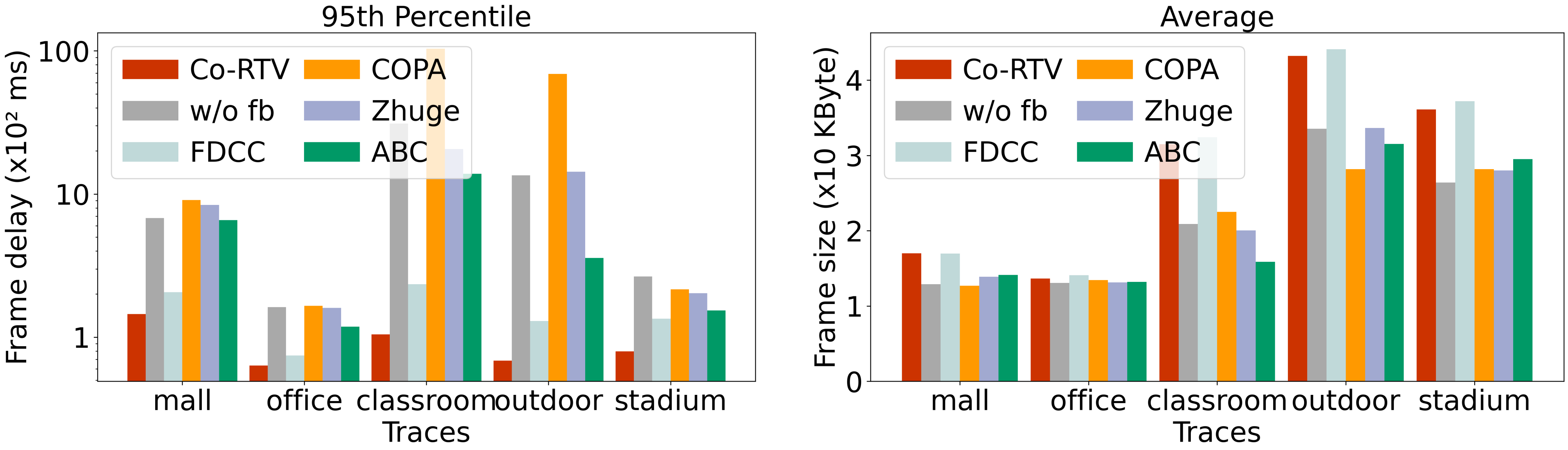


Outdoor



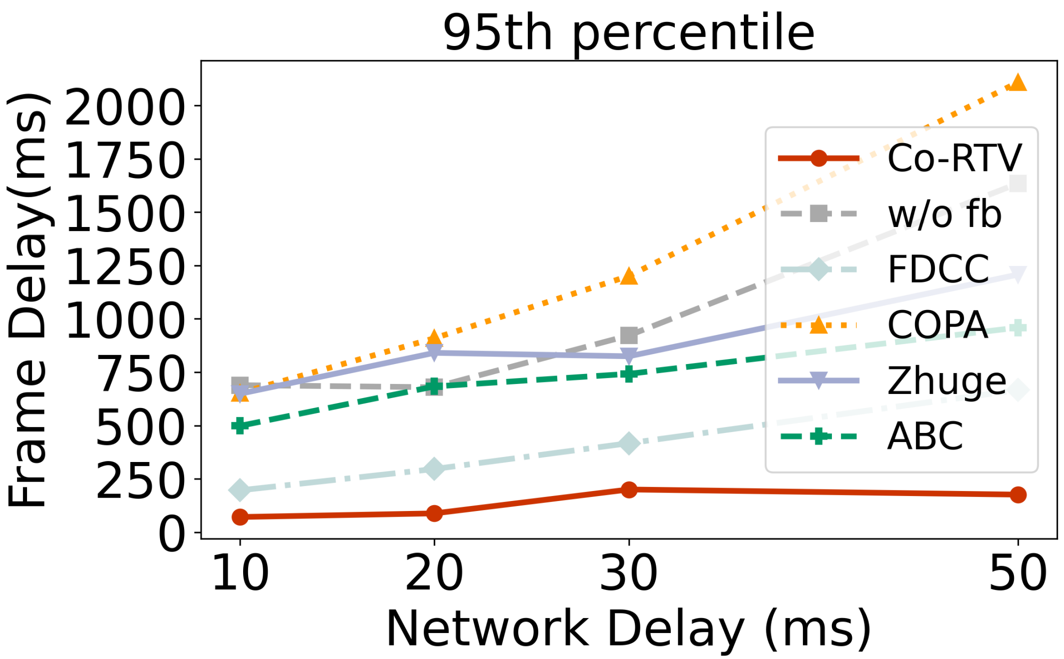
Stadium

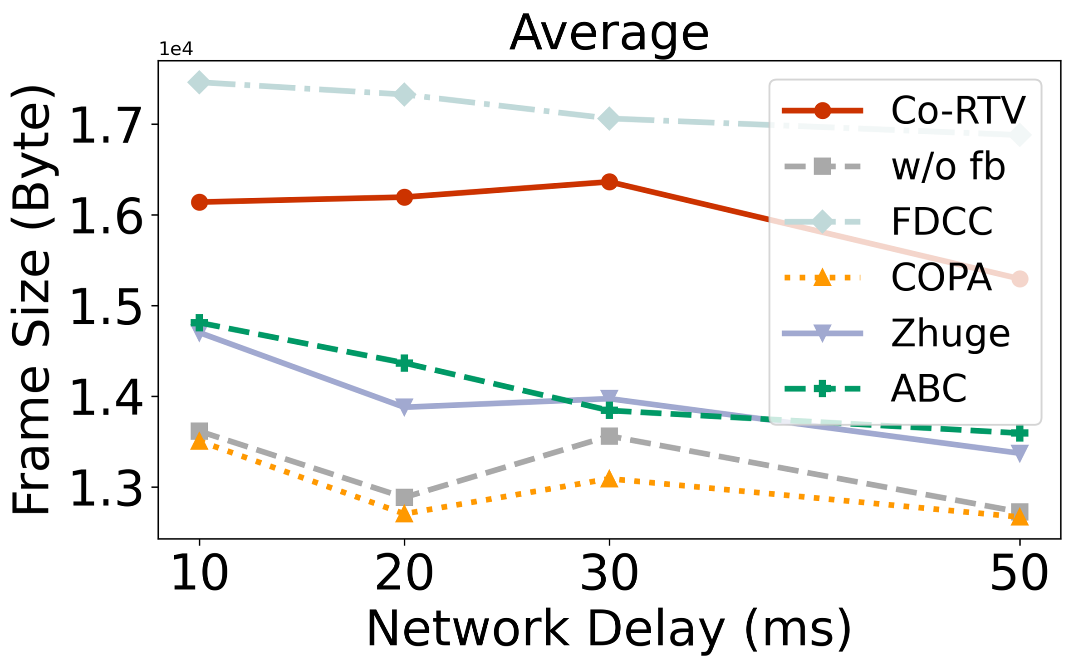
Overall performance



Co-RTV reduces 95th percentile frame delay by over 69% and increases average video frame size by 28.2% - 36.3%.

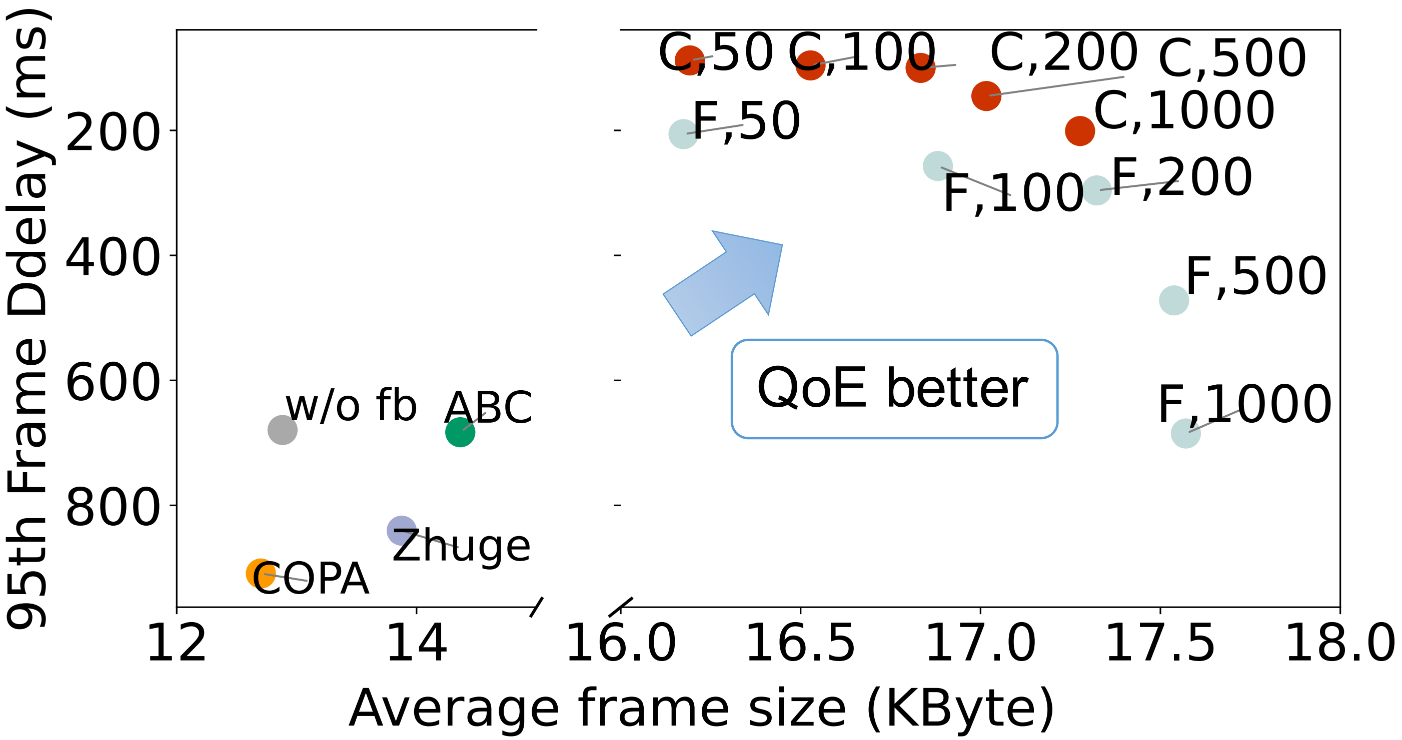
Performance under different network delays





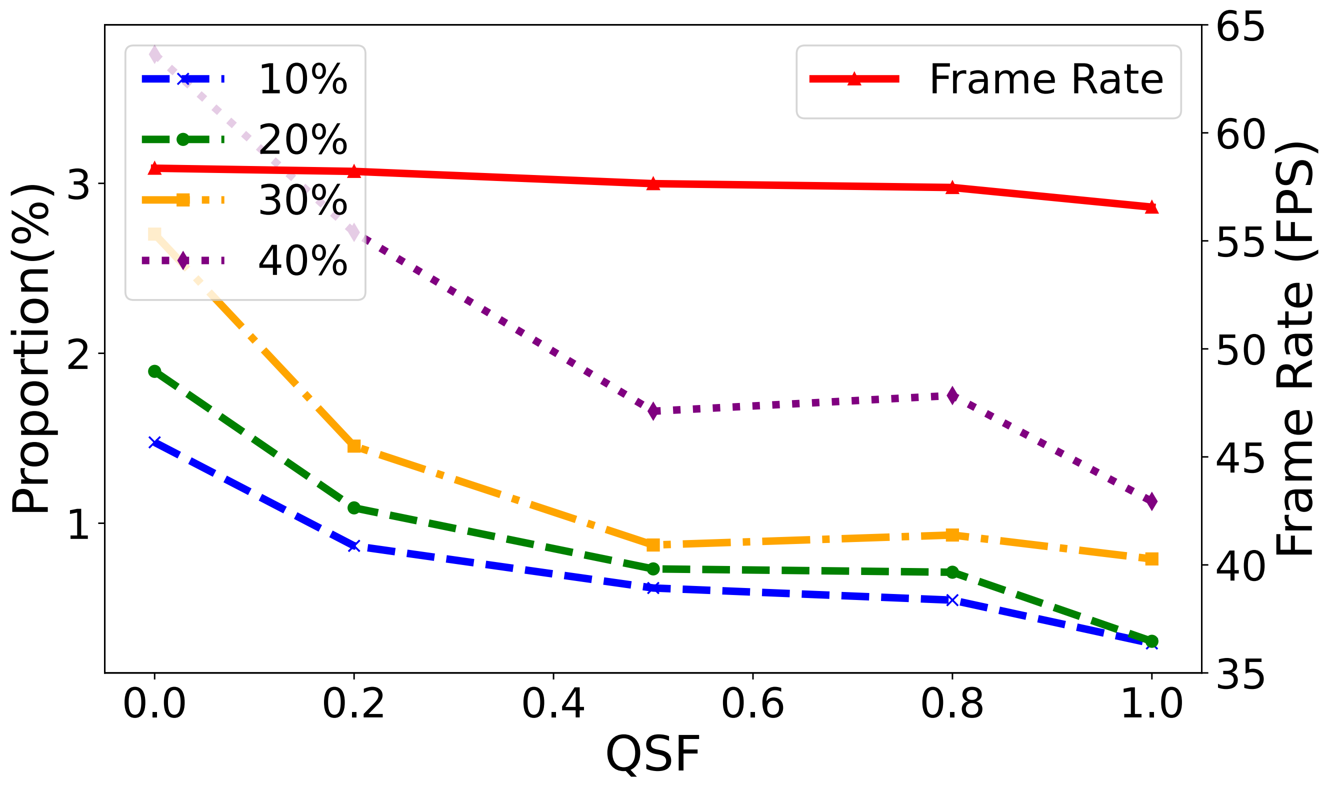
Co-RTV achieves better performance under different network delays.

Performance under different empty time



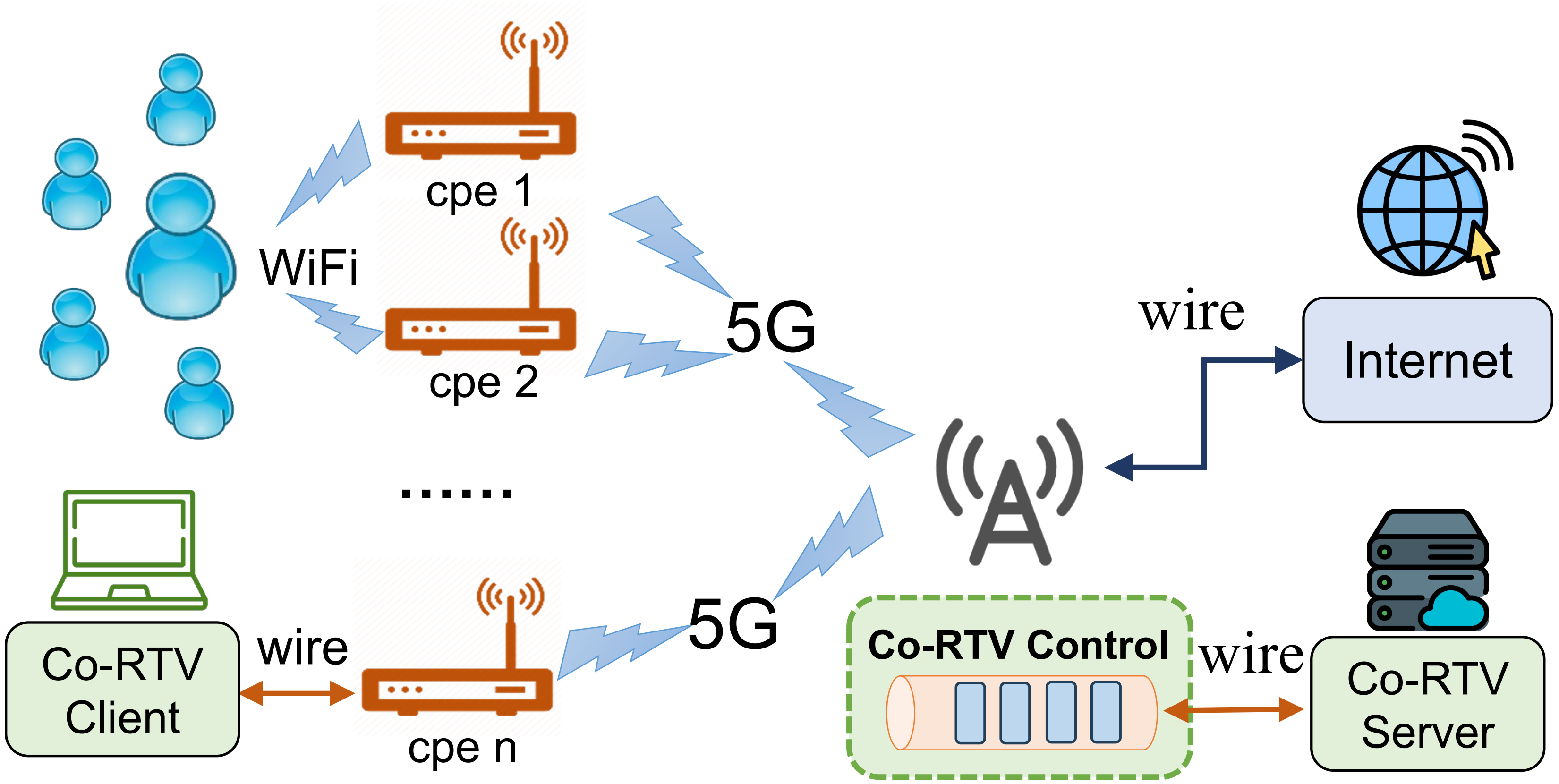
Co-RTV achieves better performance under different empty time.

Impact of QSF

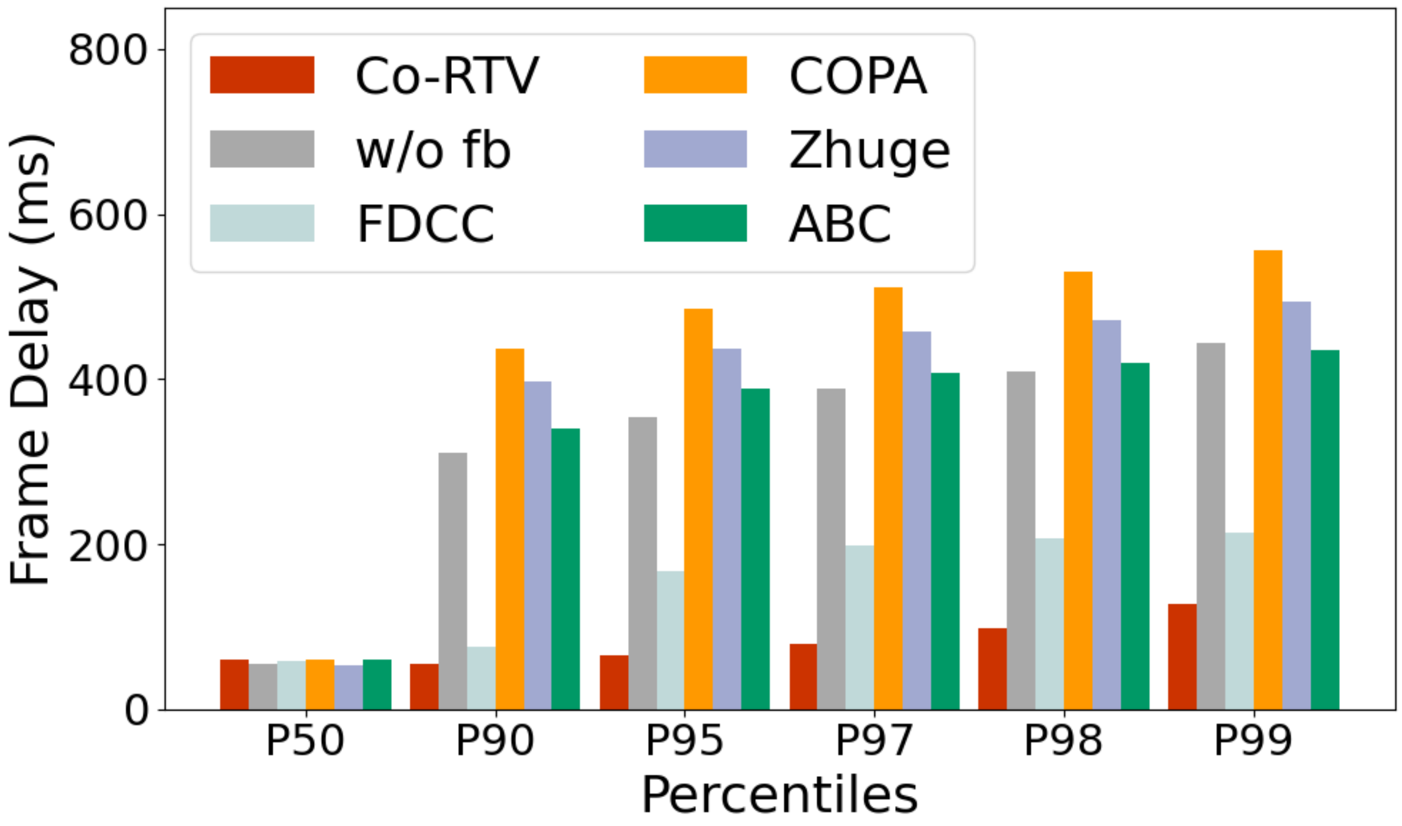


QSF effectively manages the trade-off between frame size and frame rate during bandwidth reduction.

5G Testbed Setting



5G Testbed Results



Consistent low frame delay in 5G testbed.

Summary

- **Motivation:**
  - Passive rate control falls short in handling dynamic bandwidth, leading to high queuing delays.
- **Solution: Co-RTV**
  - **Bottleneck Node: Active Frame Dropping** (Smart selection of *When* and *Which* to drop).
  - **Sender: Collaborative Re-encoding** to balance frame rate, latency and quality.
- **Performance:**
  - Reduces 95th percentile frame delay by over **69%**.
  - Improves video quality (frame size) by **30%** in real 5G testbed tests.

Thank You