Breath: Adaptive Protection Boundary in FEC Encoding for Mobile Real Time Video Streaming

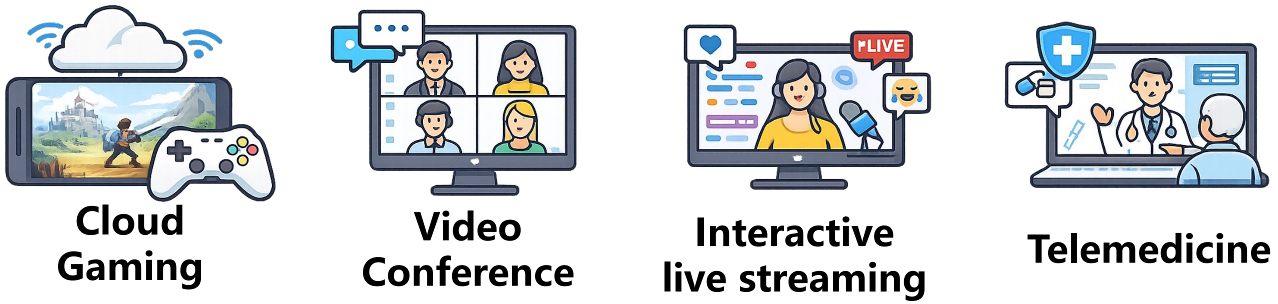
Shiyang Huang, Gerui Lv, Yuankang Zhao, Jiaxing Zhang, Qingyue Tan, Congkai An, Huanhuan Zhang, Xinyi Zhang, Qinghua Wu, Zhenyu Li

ICT, CAS, UCAS, BUPT, CNIC, CAS

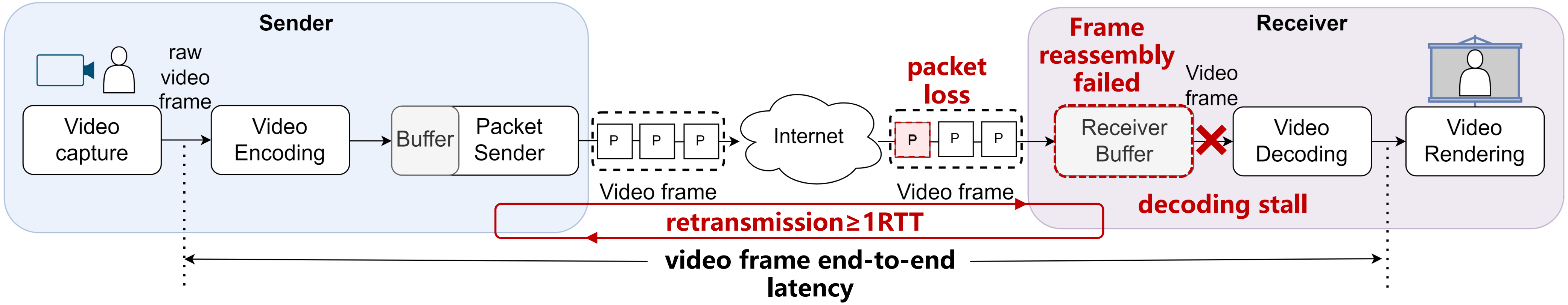
2026/7/6

Real Time Video Streaming

❖ Real-time video streaming (RTVS) applications are highly sensitive to interaction latency



❖ Requirements on end-to-end video frame latency < 150 ms*

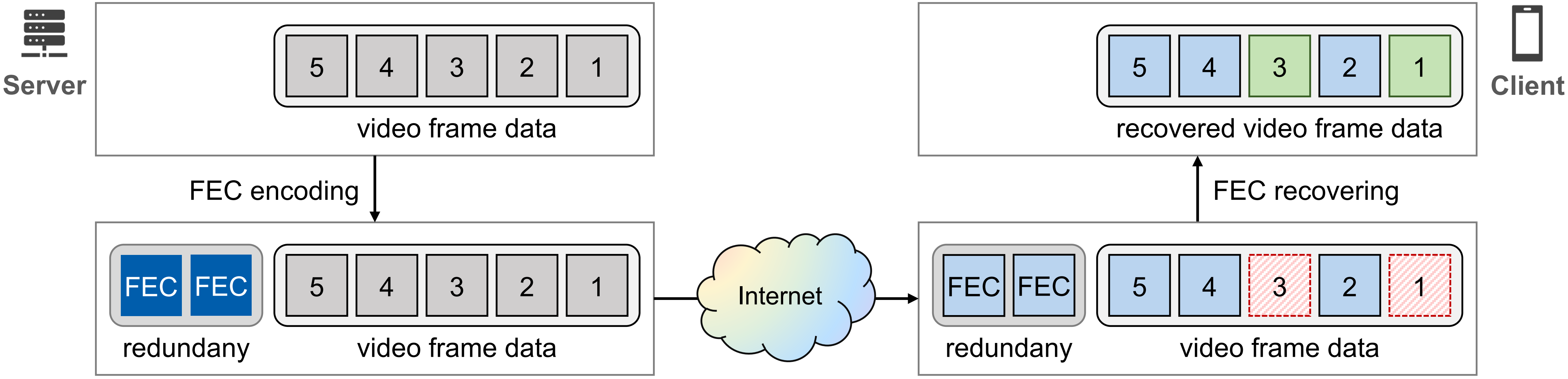


Real-time Video Streaming System

Packet loss significantly increases frame latency in RTVS systems

Why Forward Error Correction (FEC)?

- **FEC**: Recovering loss without additonal latency
  - Predicting per-frame loss rate → Adding redundancy for each frame independently
  - Require: Redundant packets ≧ **Lost packets**
  - Rule-driven: WebRTC; Learning-driven: Tooth [NSDI'25]



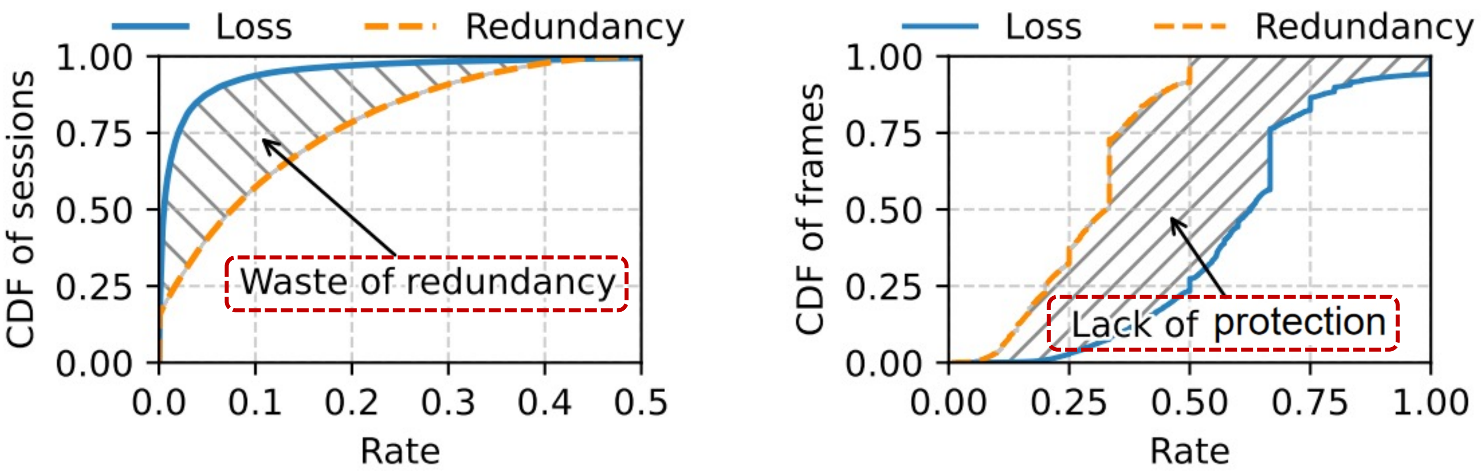
Send: Redundant packets × **2**

Receive： **Lost packets ≤ 2**

Tradeoff: redundancy vs avaliable bandwidth

Limitations of Existing FEC Schemes

- A large-scale measurement
  - A mobile RTVS system built on WebRTC
  - Include 65,742 sessions with loss, 100,000+ users and 1,500+ hours of video
- Paradox
  - **11.2%** redundancy vs **2.7%** loss
  - Still **27.6%** frames fail to be recovered

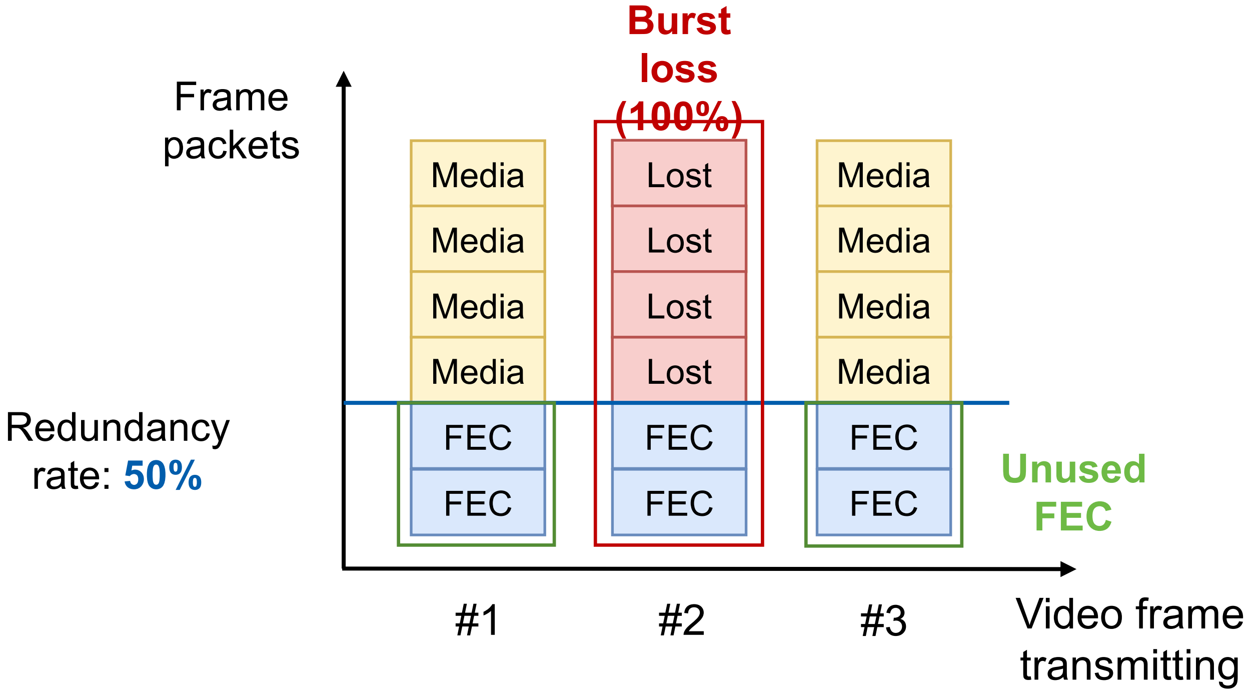


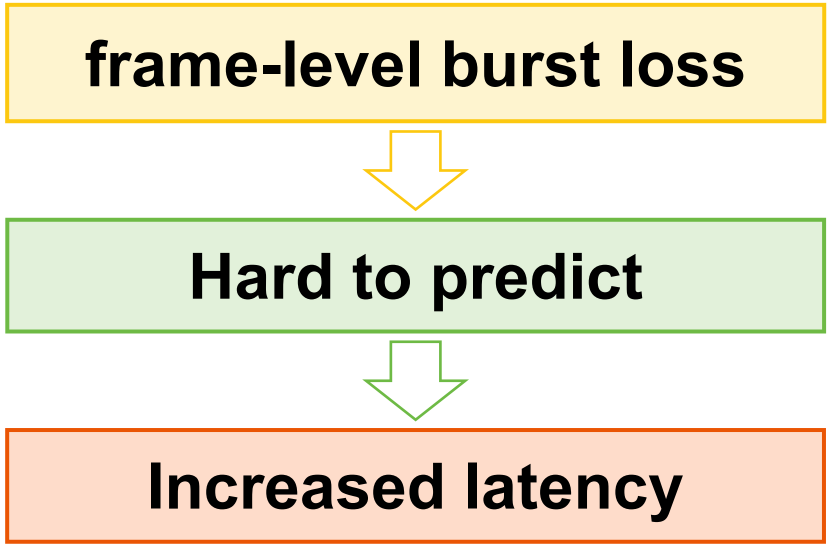
Sufficient session-level redundancy does not guarantee frame-level protection

Root Cause

❖ Unpredictable, high-intensity burst loss events

- 72.5% of FEC failed frames experience a frame loss rate higher than 50%
- 80% of severe loss events last for only 1 frame interval

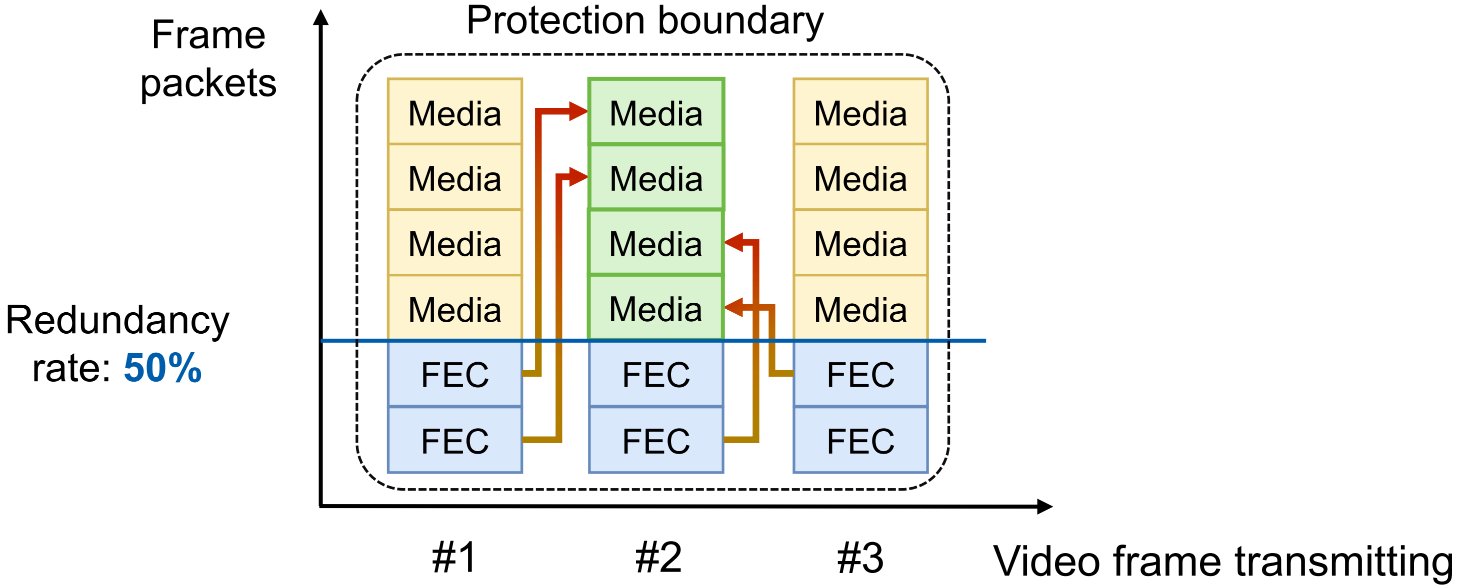




The unpredictability of frame-level burst loss makes per-frame protection inefficient

Opportunity: Cross-frame Protection

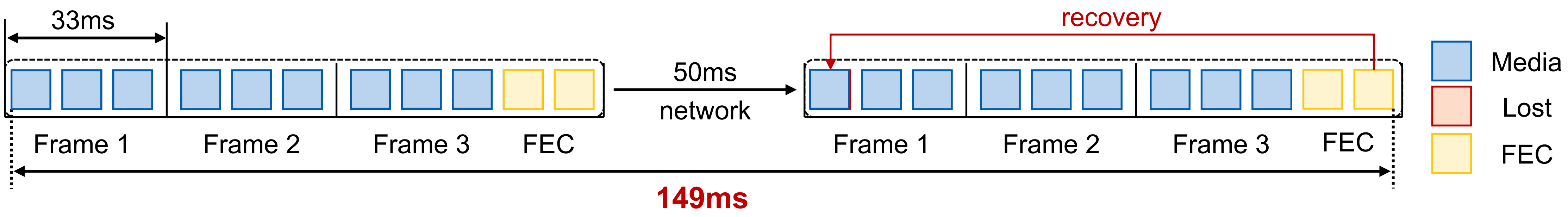
- **Insight**: Expanding the FEC protection boundary from **one frame** to **multiple frames** can reduce the impact of burst loss unpredictability.
  - Enabling redundancy sharing across frames
  - Smoothing out burst loss due to the Law of Large Numbers



Cross-frame protection is promising countering burst loss

Strawman Solution: Maximum Protection Boundary

- Grouping as many frames into one block as possible until the estimated recovery latency reaches the deadline



- Why strawman solution fails?
  - Improving recovery rate at the cost of recovery latency
  - **Latency** is the key performance metric in RTVS

Alg.	Avg. (ms)	P50 (ms)	P95 (ms)
WebRTC	85	74	145
Tooth	86	75	153
Mpb-WebRTC	93	81	148

Motivation: Tradeoff in recovery rate vs. recovery latency

Solution

❖ **Idea**: Adaptively adjusting protection boundary of FEC

Goal 1
Minimizing temporal and spatial FEC overhead

Challenge 1
Trade off in recovery rate, recovery latency and redundancy bandwidth cost

↓

Solution 1
QoE-driven FEC overhead model

Goal 2
Adapting to highly variable network status in real-time

Challenge 2
Ensuring per-frame FEC decision in strict frame-interval time limit

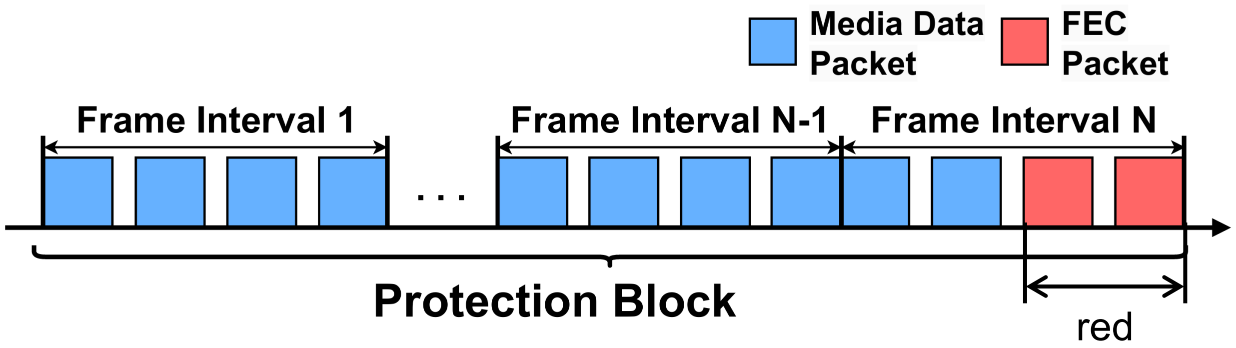
↓

Solution 2
Lightweight response-based optimization

Breath: Adaptive Protection Boundary FEC

FEC Overhead Model

- A protection block with **N** frames and **red** FEC packet:



- Block-level protection overhead:

Redundancy bandwidth cost

$$\mathbb{E}_{\text{overhead}} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{L_i}(N,\mathbf{red}) + \lambda \cdot \mathbb{R}(N,\mathbf{red}),$$

Recovery latency expectaion

$$\sum_{i=1}^{N}\mathbb{E}_{L_i}(N,\mathbf{red}) = \sum_{i=1}^{N}(P_{\text{recovery}}^{(i)} \cdot L_{\text{recovery}}^{(i)} + \omega \cdot P_{\text{fail}}^{(i)} \cdot L_{\text{fail}}^{(i)}), \quad \mathbb{R}(N,\mathbf{red}) = \frac{\mathbf{red}}{\sum_{i=1}^{N} d(i)}.$$

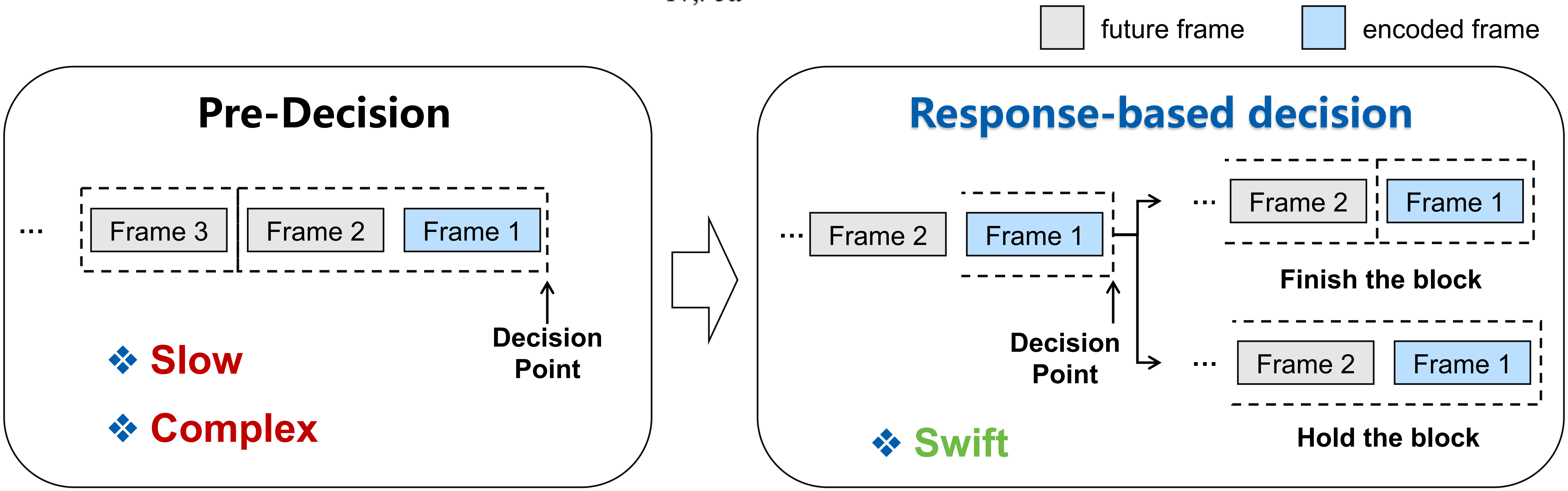
Total media packets

Modeling effect of protection boundary in FEC protection

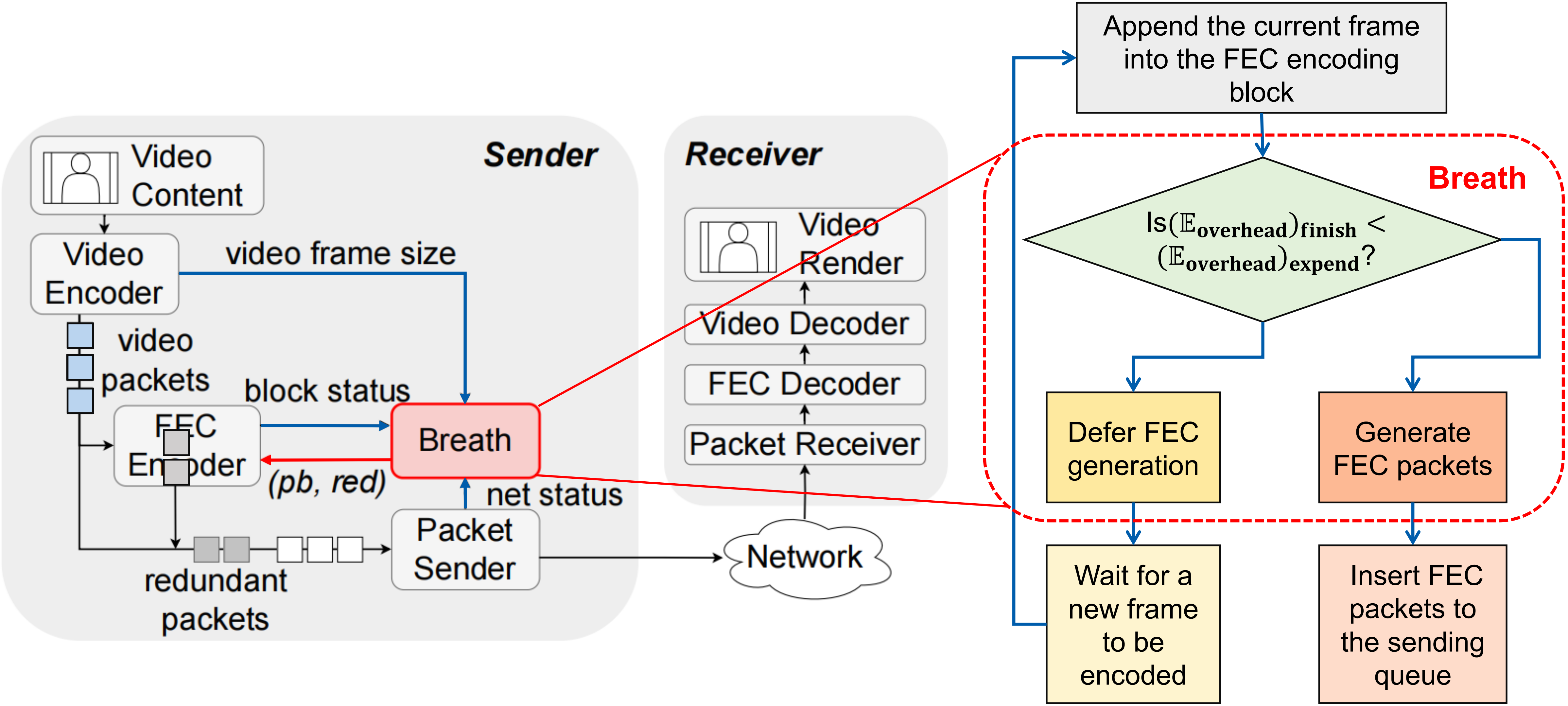
Lightweight Response-based Optimization

- Determining optimal protection boundary (N) and redundancy packets (red)

$$\min_{N,red} \mathbb{E}_{\text{overhead}}$$

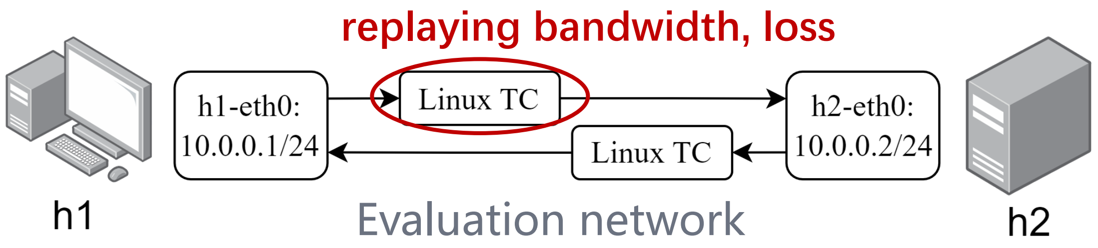


Breath Framework



Evaluation Setup

- ❖ Platform
  - Linux TC-based network emulation
  - Replay of 100+ real-world loss & bandwidth traces
- ❖ Baselines
  - Rule-driven scheme: WebRTC
  - Learning-driven scheme: Tooth [NSDI'25]
  - Strawman solution: Mpb-WebRTC
- ❖ QoE metrics
  - Visual quality: average bitrate
  - Video latency: 150ms deadline missing rate (DMR)、video frame latency

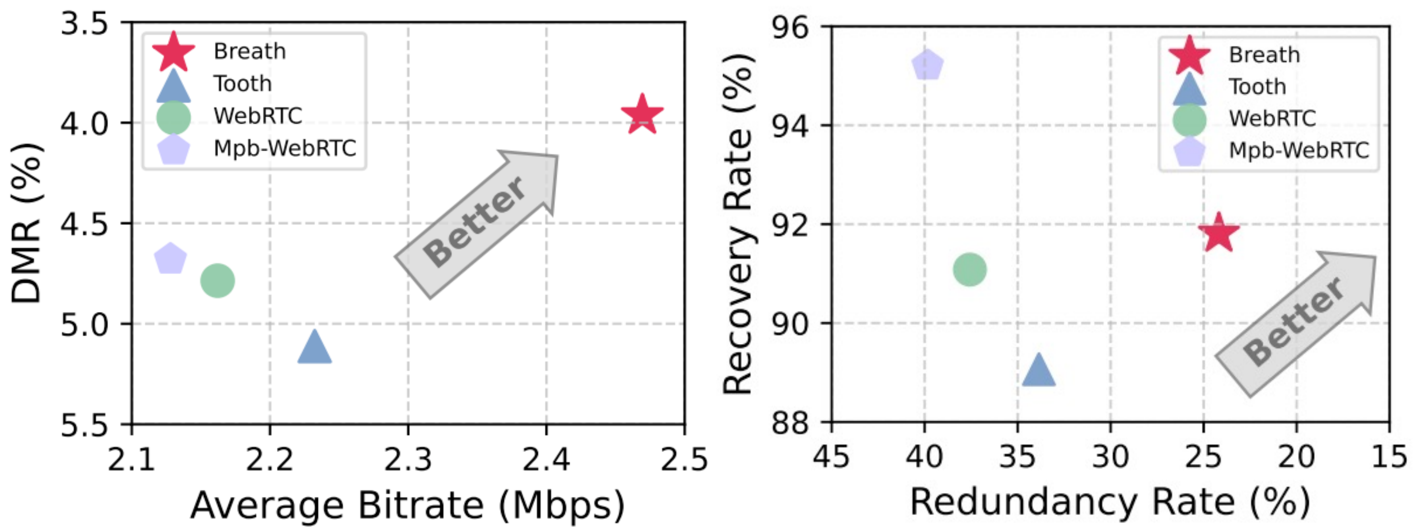


Evaluation network topology

Evaluation Results

❖ Breath significantly improves user QoE while reducing FEC overhead.

- Deadline missing rate 17.2%~22.5% ▼
- Video bitrate 10.6%~14.2% ▲
- P95 frame transmission latency 6.9%~11.8% ▼



Alg.	Avg. (ms)	P50 (ms)	P95 (ms)
WebRTC	85	74	145
Tooth	86	75	153
Mpb-WebRTC	93	81	148
Breath (Ours)	**87**	**76**	**135**

Breath pushes forward the Pareto frontier of FEC protection efficiency and overhead

Takeaway Messages

- **Observation:** Per-frame FEC schemes rely on per-frame loss rate estimation and fail in the burst loss conditions

- **Insight:** Cross-frame protection has potential in mitigating burst loss

- **Benefits: Breath** improves FEC protection effectiveness without additional redundancy cost, achieving 17-22% DMR reduction and 10-14% video bitrate improvement.

Thanks!

Q & A

For any further questions, please contact:

Shiyang Huang (ICT, CAS)

huangshiyang23s@ict.ac.cn