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

Adaptive Video Encoder for Network Bandwidth Drops in Real-Time Communication

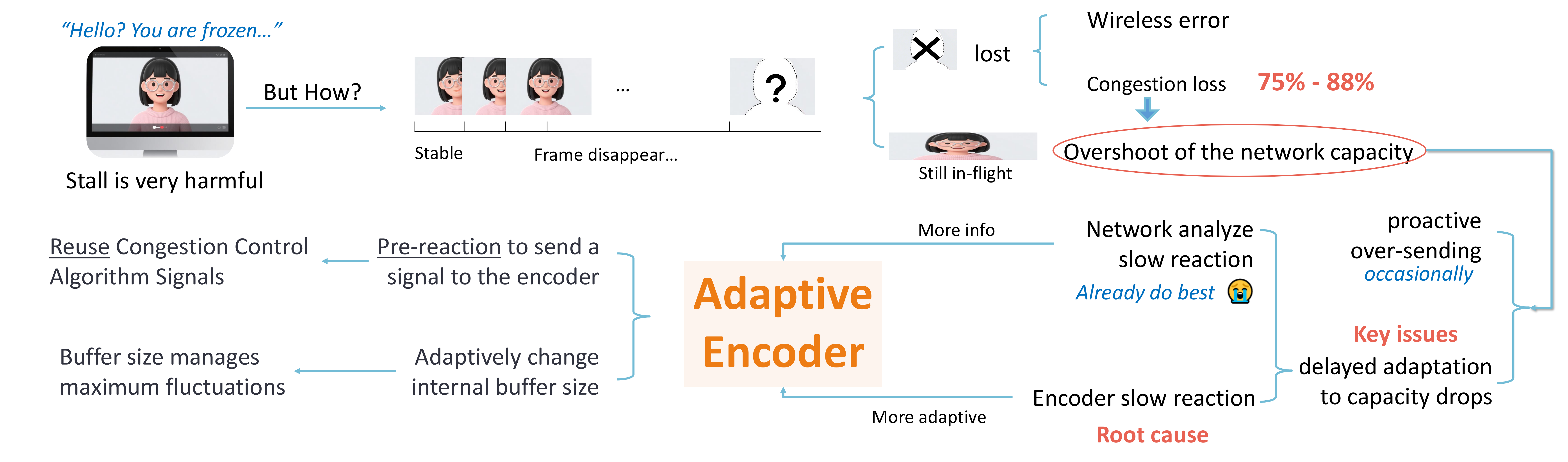
sigcomm2025
COIMBRA

Hua Meng, Xiangjie Huang, Zili Meng

HKUST

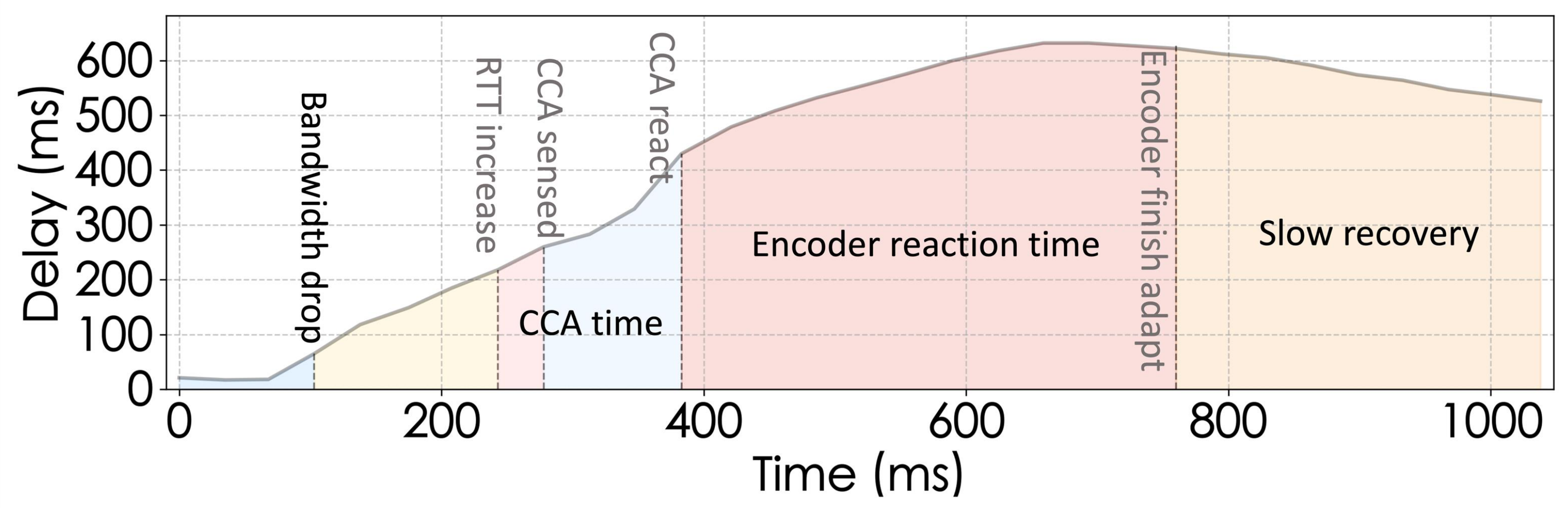
Project details: https://huameng15.github.io/

Overview

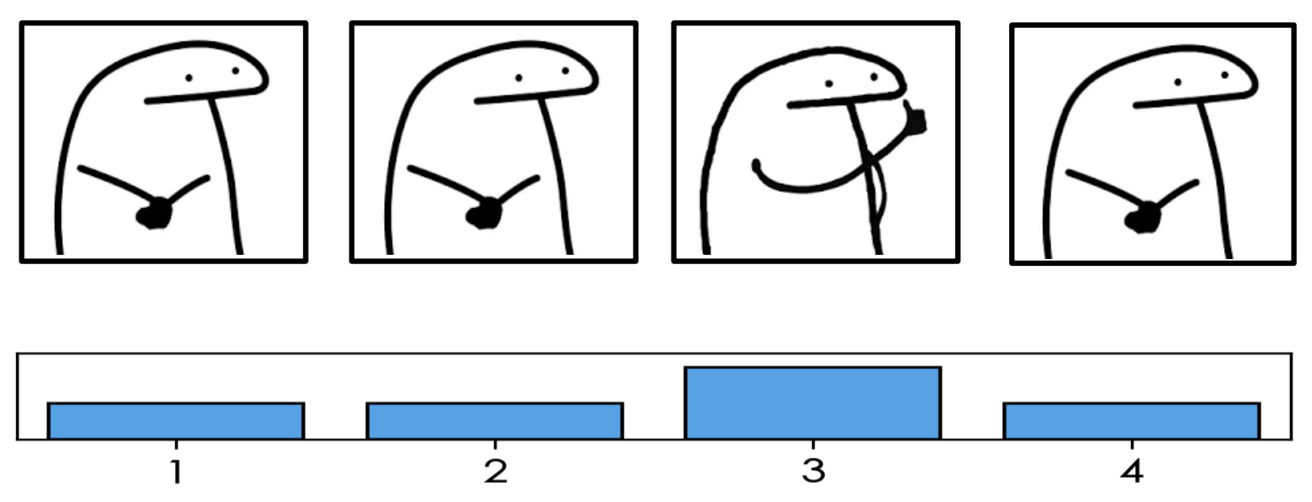


Motivation

Why current CCAs already do their best?



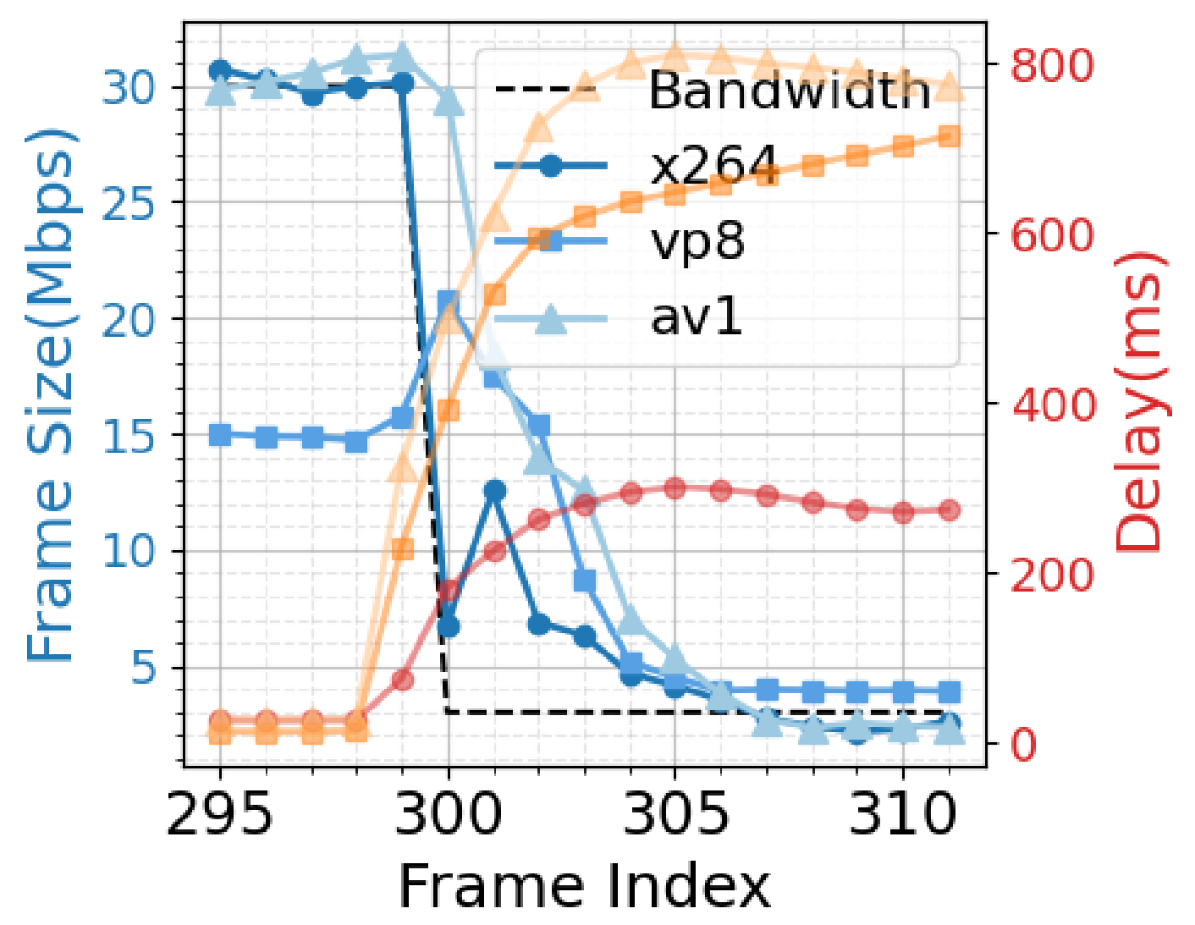
Why encoders react slow?



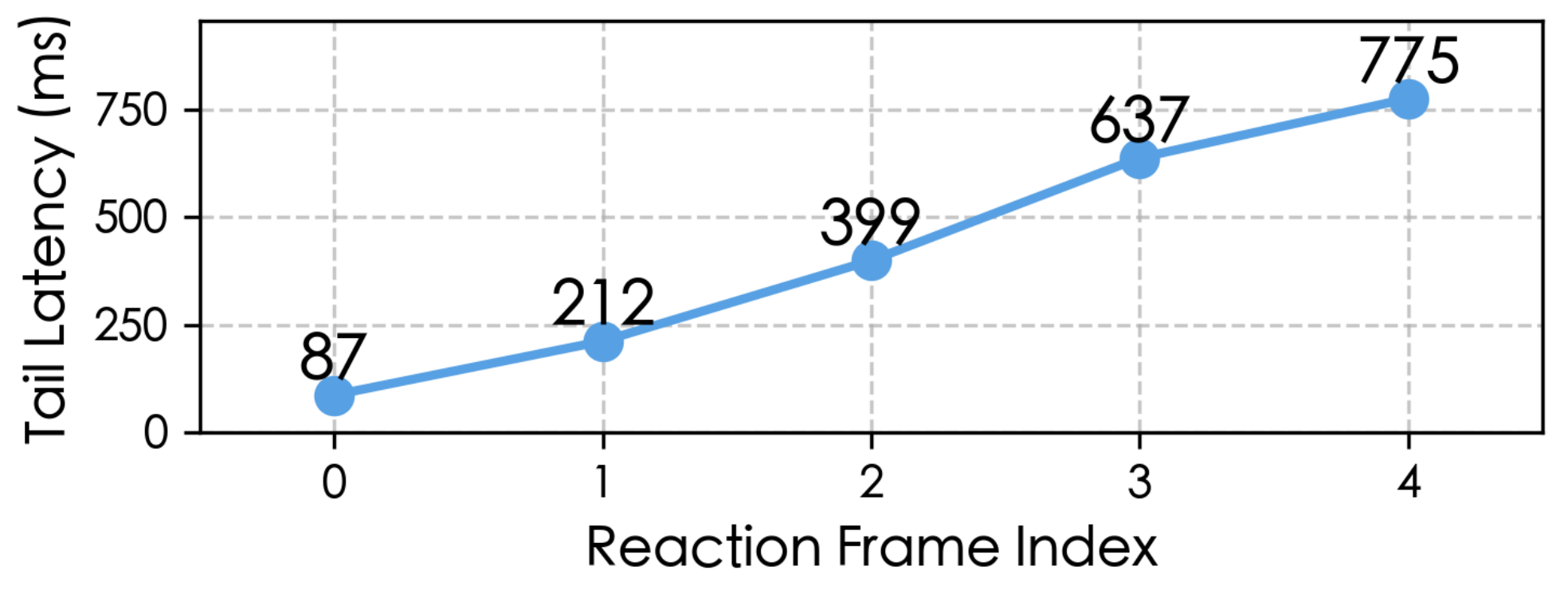
Fluctuated frame size → Buffer manage

↓

Buffer too large for bandwidth drop



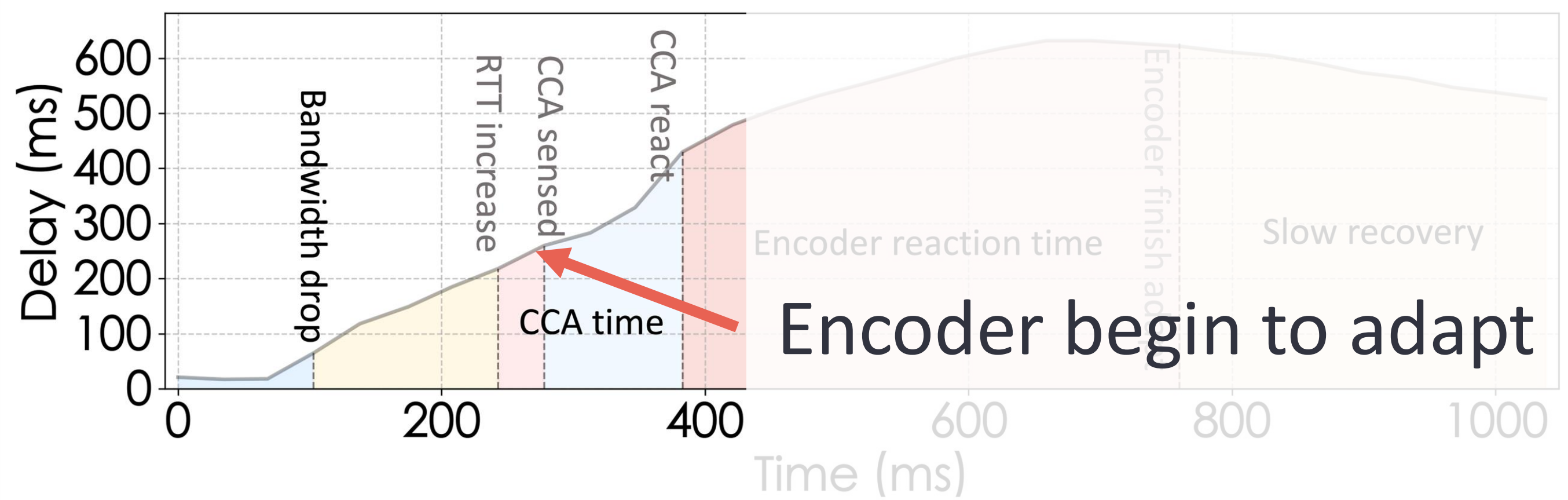
Why we need pre-reaction?



Slower reaction,
Higher latency

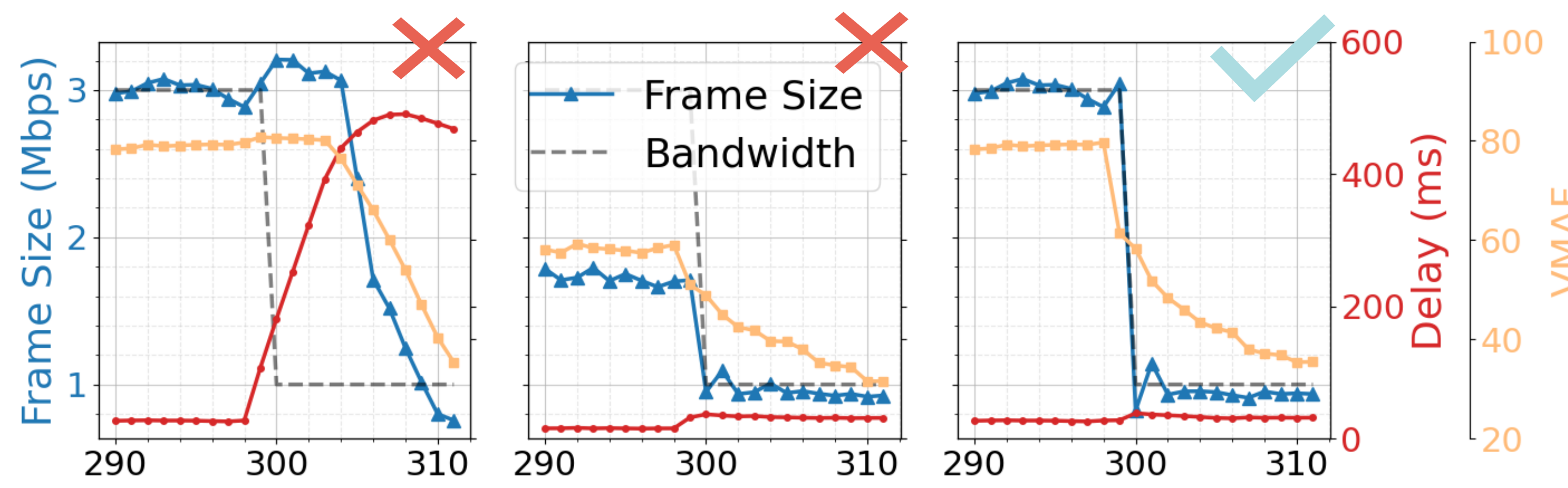
Design

Network side pre-reaction



Encoder begin to adapt

Encoder side adaptation



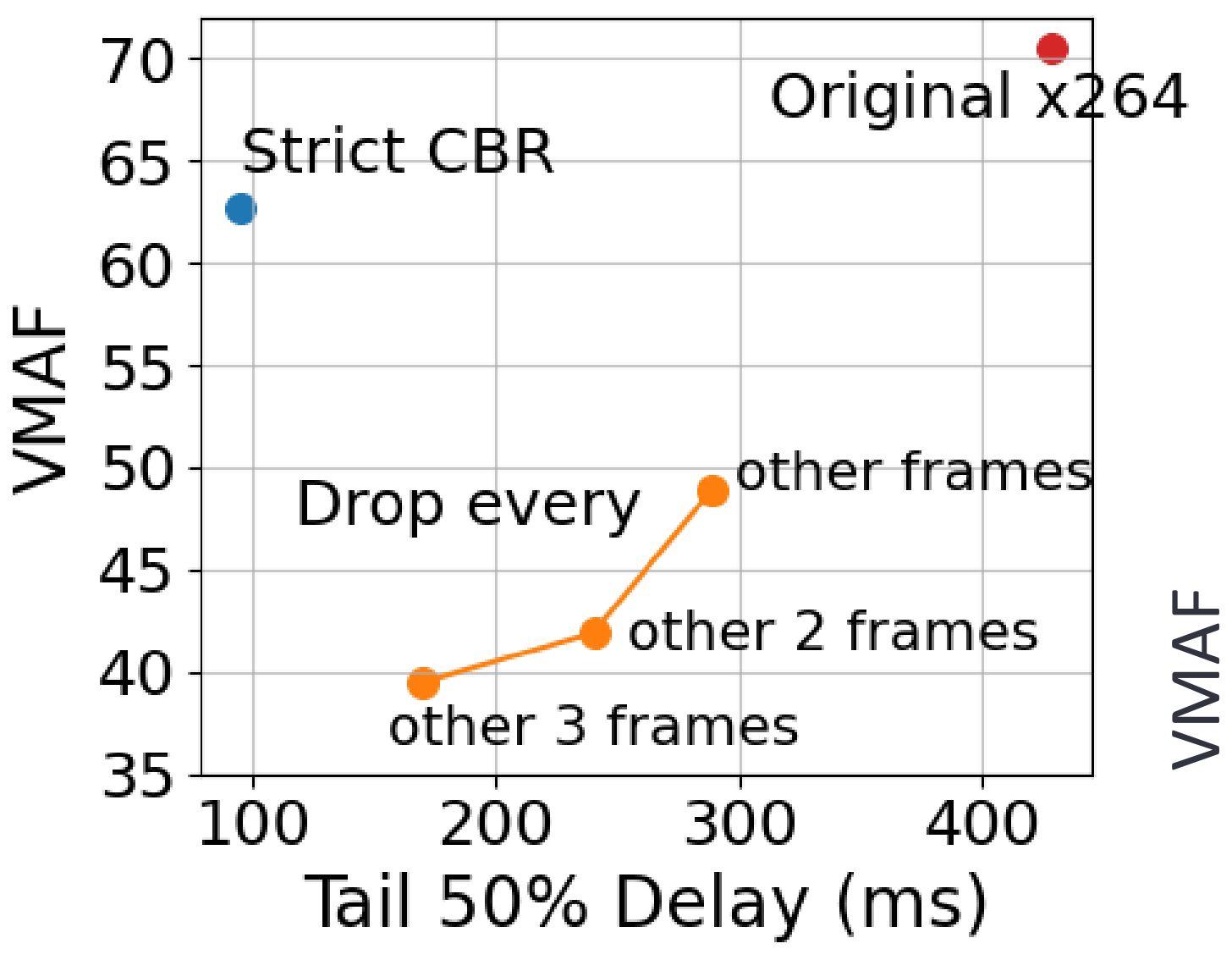
(a) 0.5x
Delay increase

(b) 0.04x
Quality loss

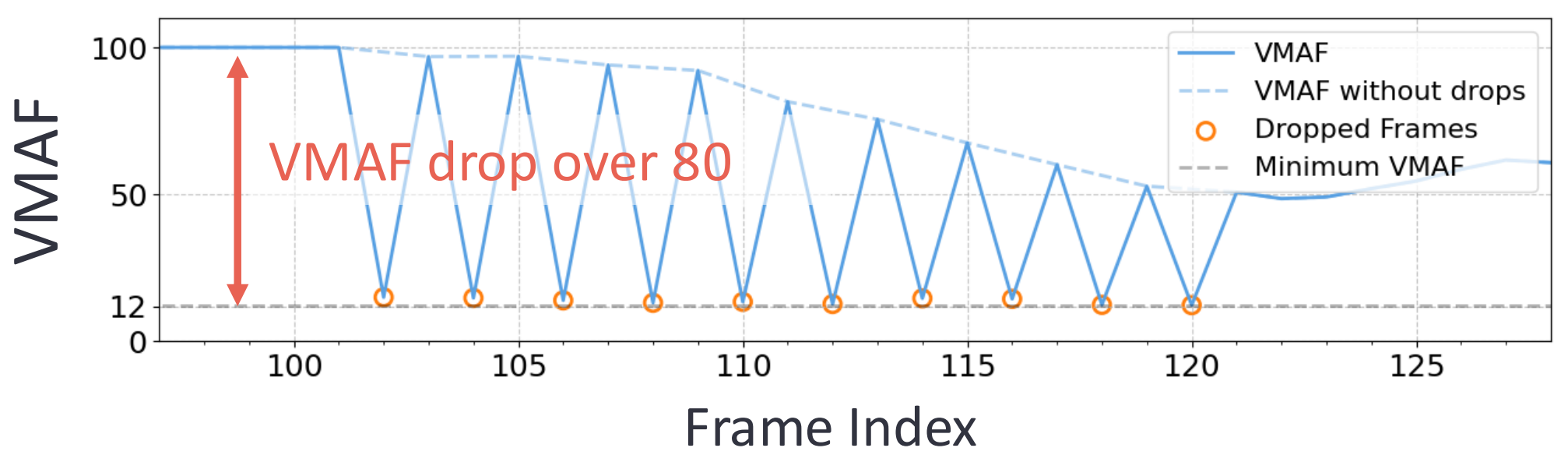
(c) adaptive
Balance

Adaptive
buffer size

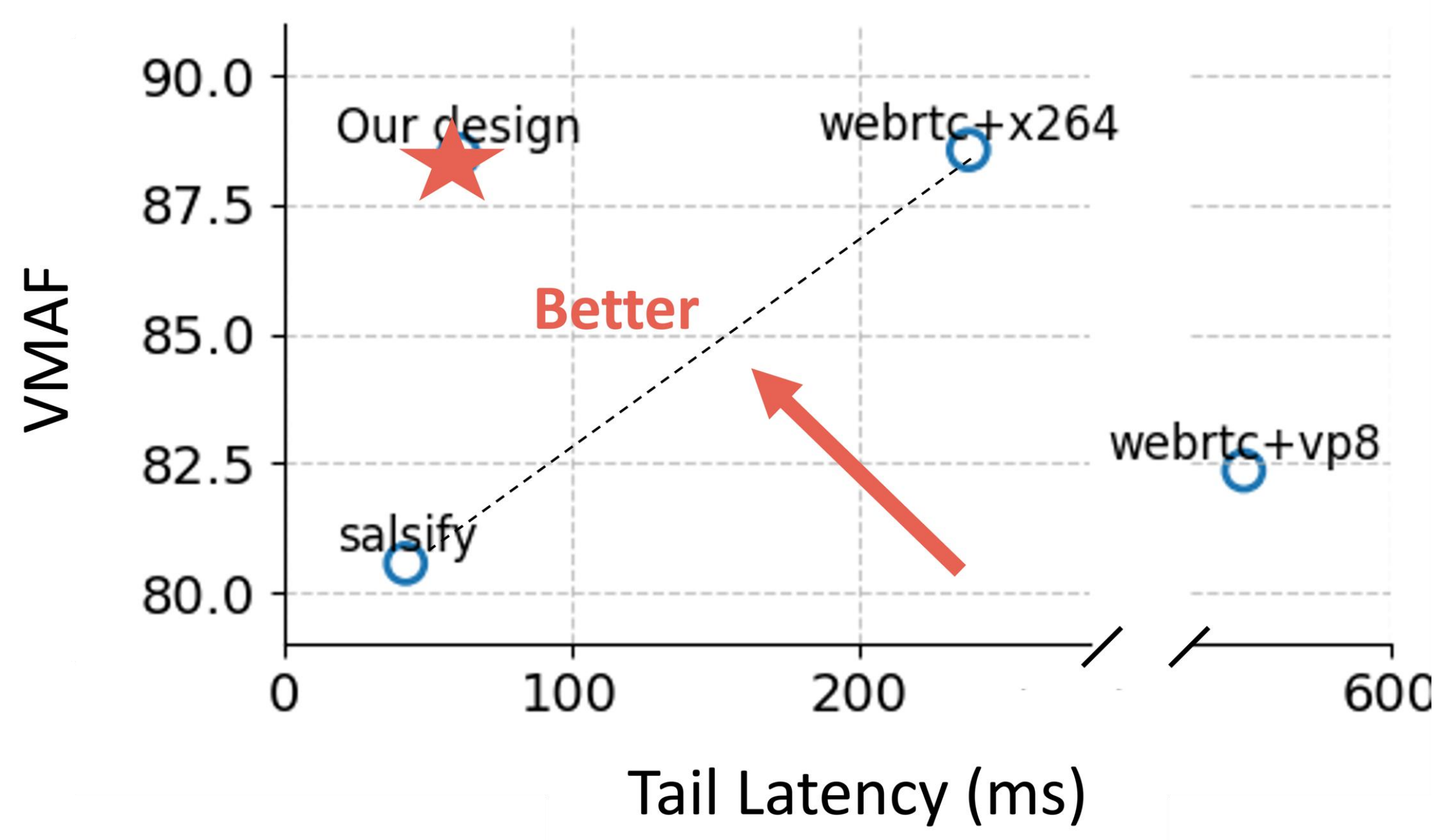
Why cannot directly drop frame?



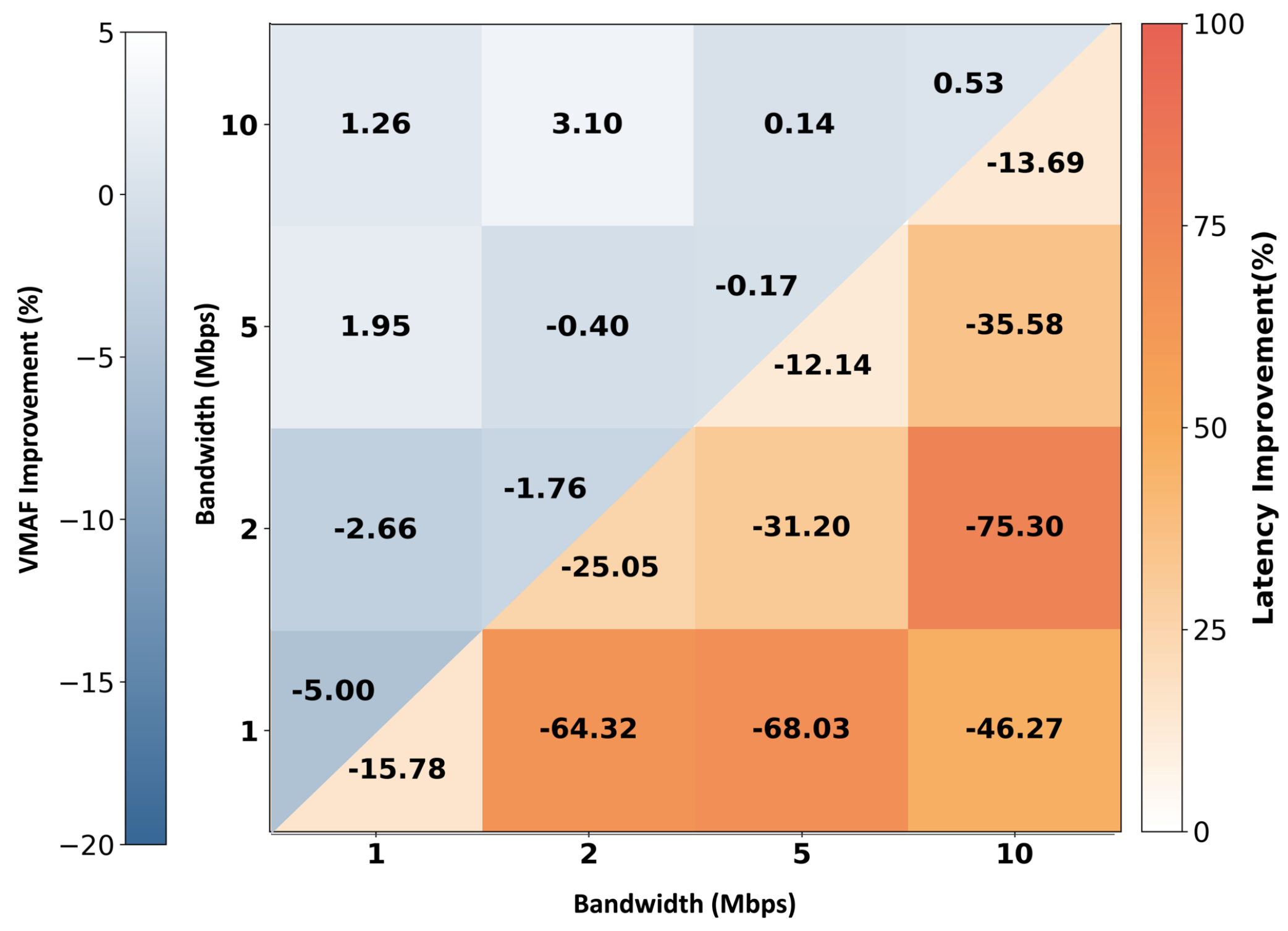
Counterintuitively, directly dropping frames can significantly *harm quality*, even more than strictly limiting each frame to not exceed the target bitrate.



Evaluation



Performance under real world Wi-Fi trace.



Bandwidth (Mbps)	1	2	5	10
10	1.26	3.10	0.14	0.53 -13.69
5	1.95	-0.40	-0.17 -12.14	-35.58
2	-2.66	-1.76 -25.05	-31.20	-75.30
1	-5.00 -15.78	-64.32	-68.03	-46.27

Latency reduction and VMAF loss compare for different bandwidth condition

Takeaways

- The root cause of stalls in RTC is the **slow response** of the **encoder** to bandwidth drops.
- Adaptively adjusting the encoder's internal parameters can help mitigate this issue.
- Reusing the Congestion Control Alsogithm signal can enable a proactive response to bandwidth drops.