PAVE: Mitigating Non-Congestive Delay for Seamless Video Calls over NextG Mobile Networks

Goodsol Lee*, *Seyeon Kim, Juheon Yi, Junhong Min*

Sangtae Ha, Kyunghan Lee, Saewoong Bahk

SEOUL NATIONAL UNIVERSITY

KOREA UNIVERSITY 1905

University of Colorado Boulder

**Currently a Member of Technical Staff at Nokia Bell Labs (NJ, USA)*

Mobile Real-Time Communications (RTC)

Video Call

VR/AR

Telesurgery

Requirements for High QoE*: **Consistent low latency under deadlines**
(e.g., 1080p video Call: <150 ms end-to-end delay even at 99-th tail)

*QoE: Quality of Experience

Mobile Real-Time Communications (RTC)

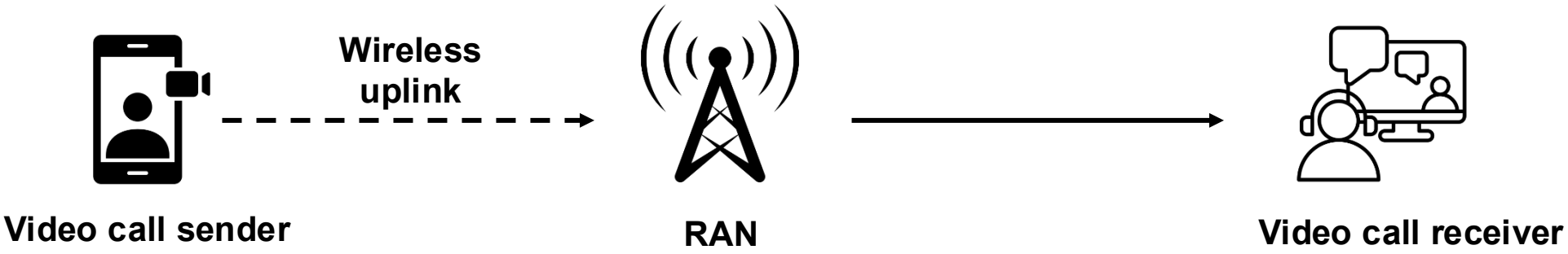
Video Call

VR/AR

Telesurgery

5G as the key technology to enable seamless mobile RTC
from its **high bandwidth & low latency + ubiquitous coverage**

Problem: Still High Tail Latency over 5G



WebRTC 1080p video call experiment

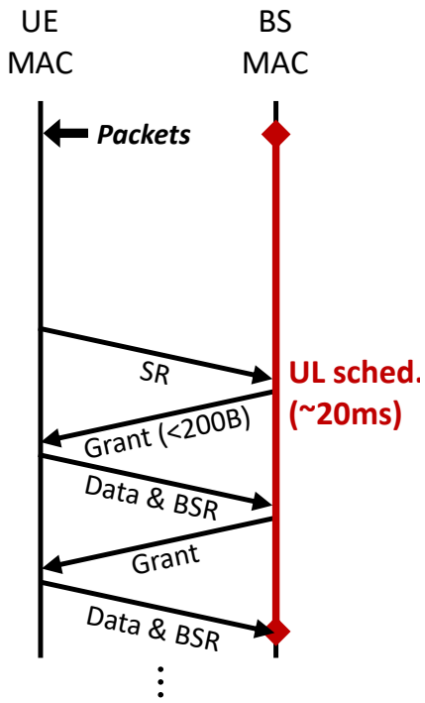
Experiment	Avg. 5G BW	Avg. Bitrate	P99/P99.9 frame delay	Stall rate (>150 ms)
Verizon (VZ)	**61Mbps**	3.6Mbps	**253/264ms**	5.3%
T-Mobile (TM)	**54Mbps**	3.7Mbps	**212/364ms**	4.7%

Average **5G bandwidth > 50 Mbps** while **app only requires < 5Mbps**

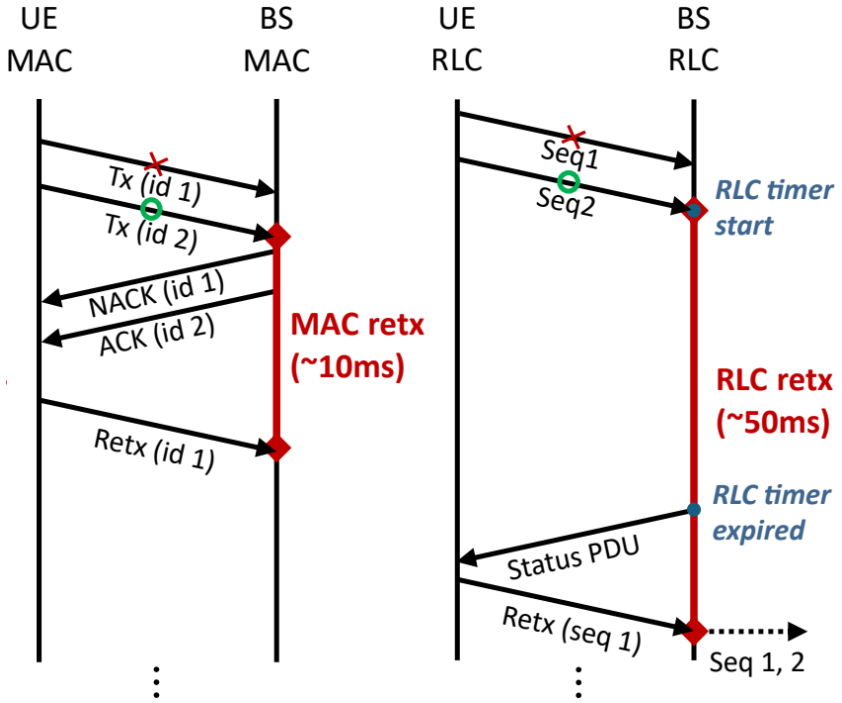
Where does latency come from?

Root Cause: Non-Congestive Delay in 5G RAN

- Non-congestive delay
  - Time spent without sending traffic even though there is an available bandwidth

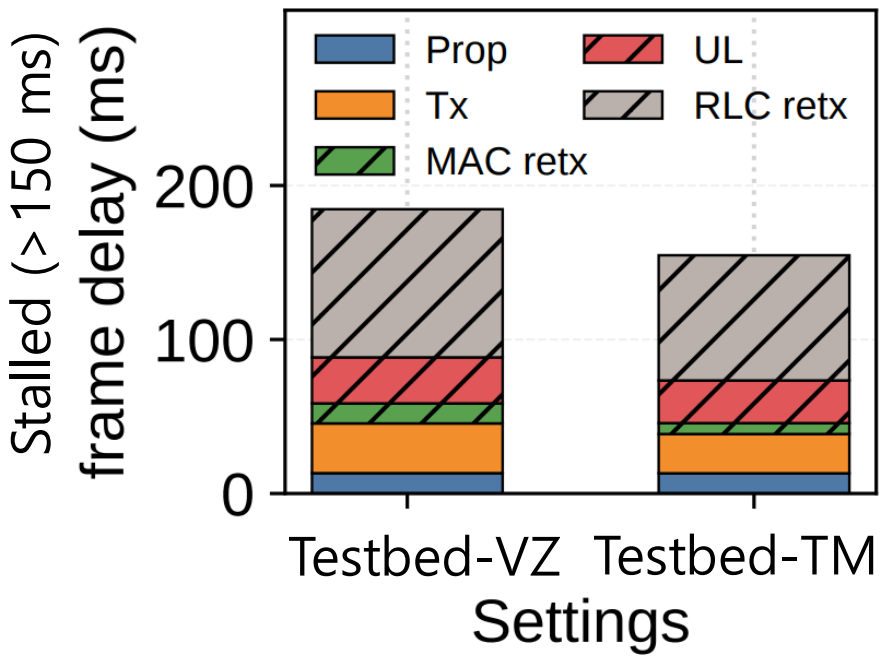


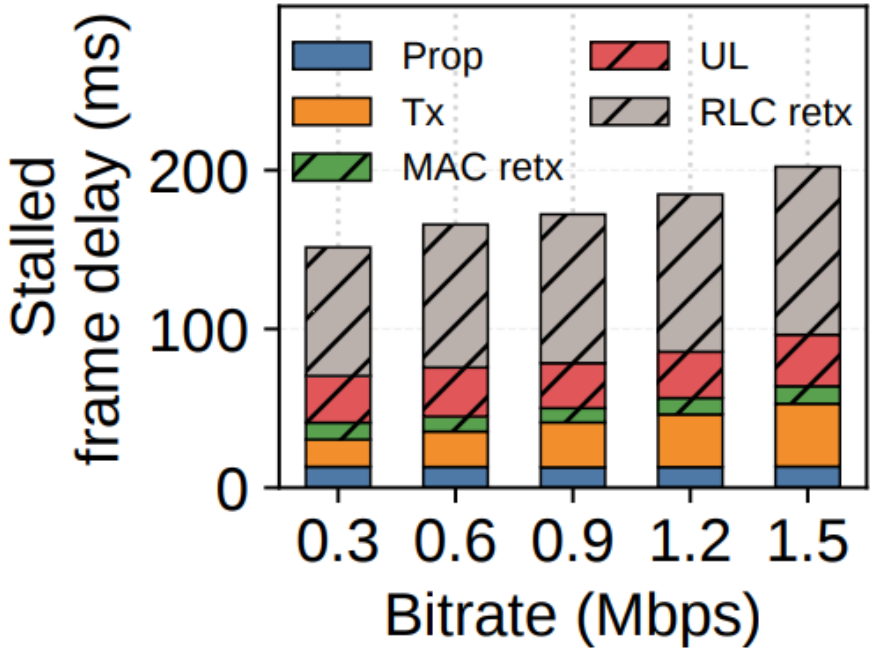
Notifying user equipment (UE) buffer status to the base station (BS)



Two-stage procedures to guarantee in-order delivery

Root Cause: Non-Congestive Delay in 5G RAN





High tail latency is due to **high non-congestive delay in 5G RAN**

Existing Solutions Mainly Tackle Congestive Delay

Congestion control for low-latency traffic
[NSDI'24] Pudica,
[INFOCOM'24] Exstream,
[SIGCOMM'22] Zhuge,
[SIGCOMM'20] PBE-CC,
[NSDI'18] Salsify
[NSDI'18] Copa,
[CoNEXT'18] ExLL

Transport layer
Adjust the sending rate to the bandwidth

Network slicing/Radio resource allocation for low-latency traffic
[NSDI'25] S-MEC,
[NSDI'24] Zipper
[MobiSys'24] ARMA,
[NSDI'23] RadioSaber,
[MobiCom'22] Tutti,
[CoNEXT'21] OnSlicing

Link layer
Adjust the bandwidth to the sending rate

Existing Solutions Mainly Tackle Congestive Delay

Congestion control for low-latency traffic
[NSDI'24] Pudica,
[INFOCOM'24] Exstream,

Network slicing/Radio resource allocation for low-latency traffic
[NSDI'25] S-MEC,

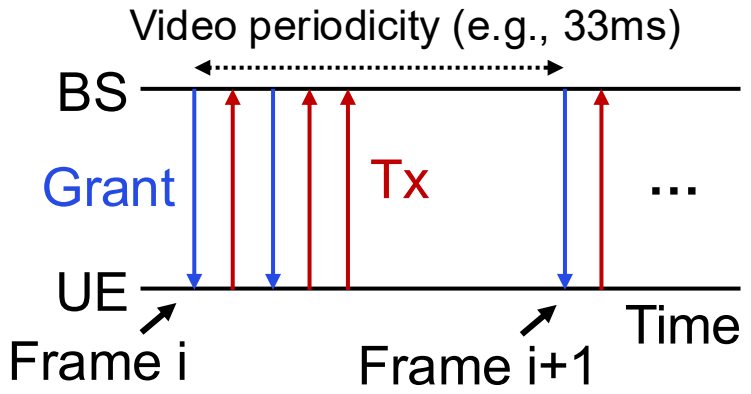
However, **non-congestive delay** can occur **even without congestion!**

Transport layer
Adjust the sending rate to the bandwidth

Link layer
Adjust the bandwidth to the sending rate

Our Approach: App Insights into 5G RAN

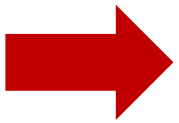
- Goal #1. Reduce uplink scheduling delay
- Naïve approach #1. Pre-allocate radio resources in every slots
  - Problem: Wasted radio resources
- Insights #1. Periodic frame transmission



Our Approach: App Insights into 5G RAN

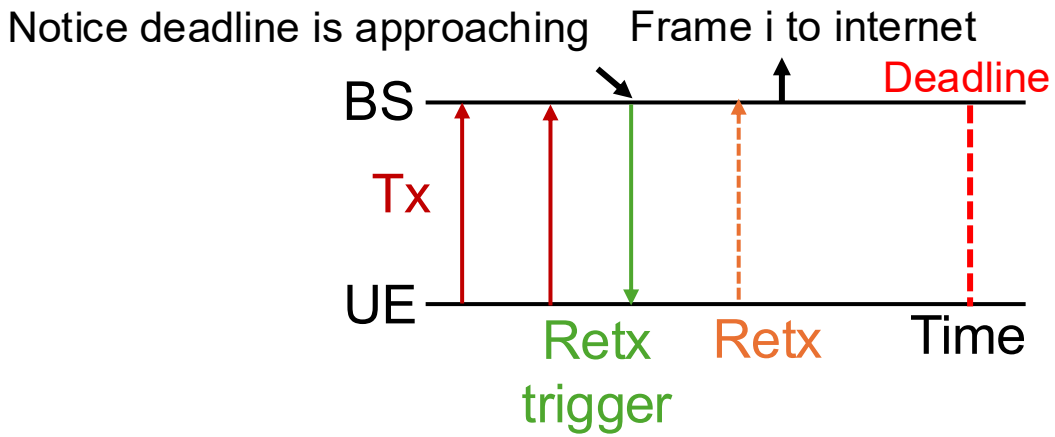
- Goal #2. Reduce retransmission delay
- Naïve approach #2. Reduce retransmission via conservative channel coding

Reducing channel coding's target error rate from **10% → 0.1%**



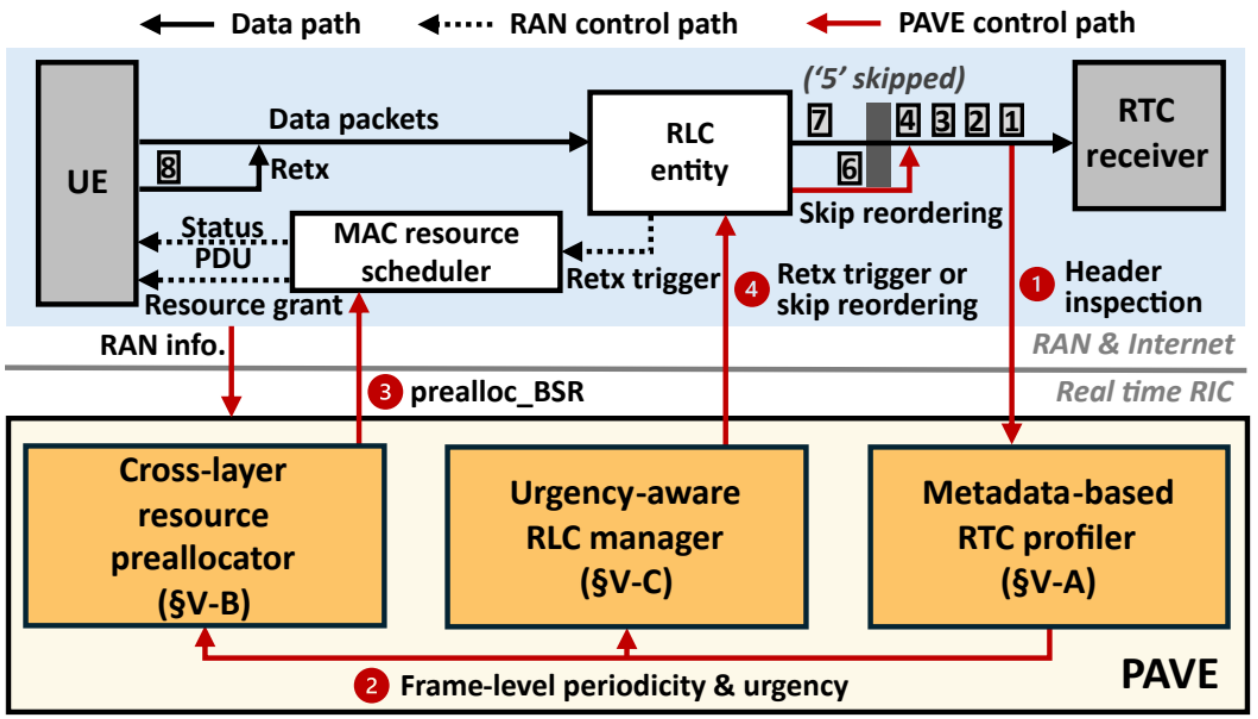
42% of throughput drop
24 Mbps → 14 Mbps

- Insights #2. Deadline-driven urgency to reduce retransmission delay



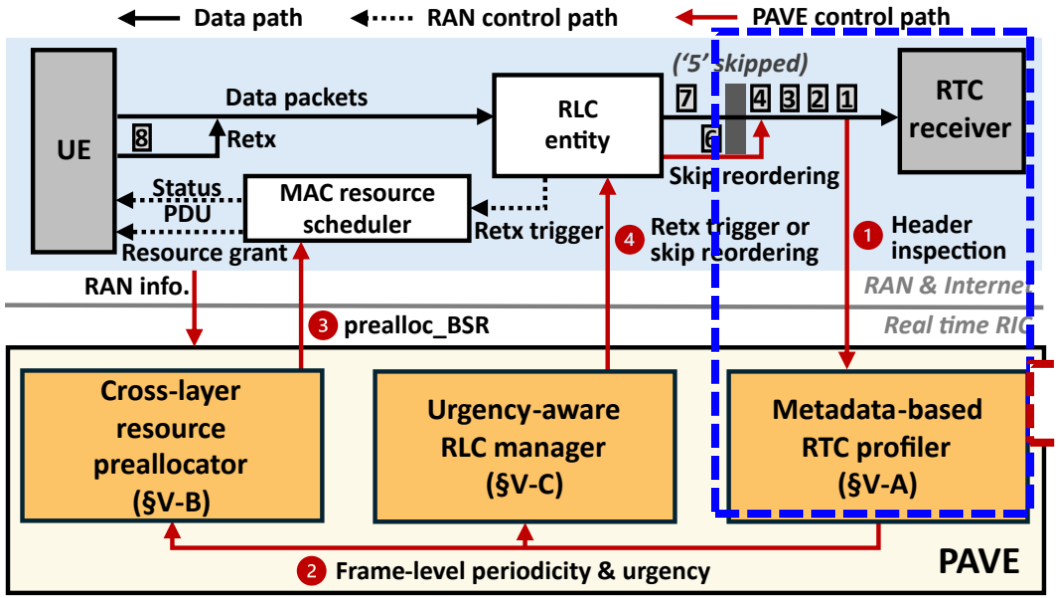
PAVE Overview

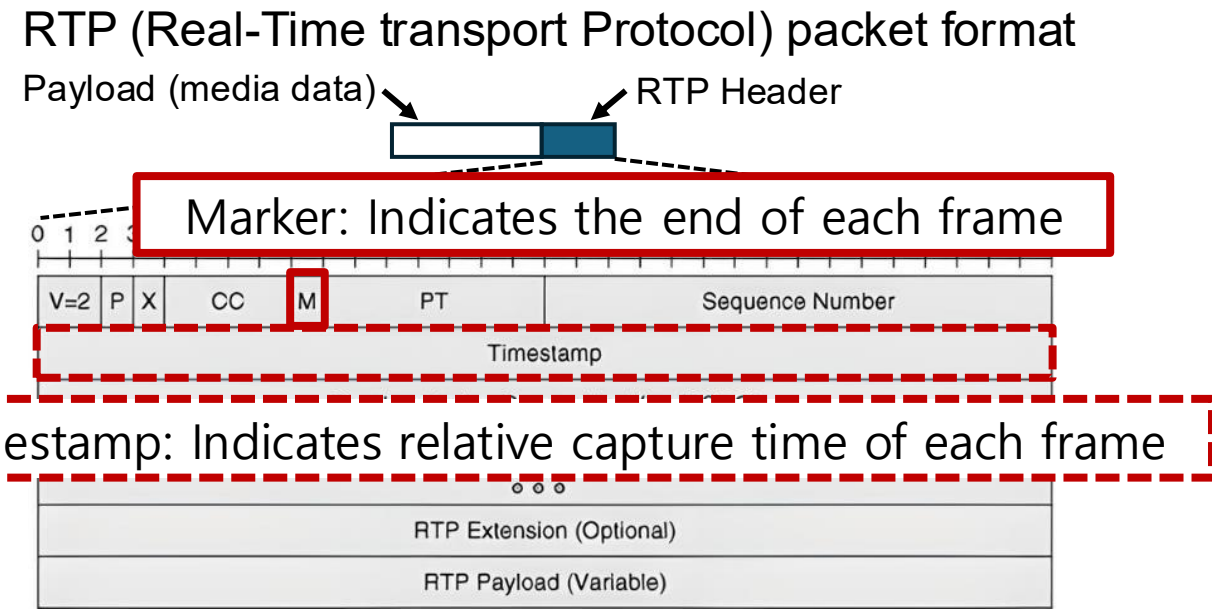
- Optimize RAN protocol for RTC applications in radio resource efficient way



C#1: Inferring RTC characteristics

- How to achieve **application layer frame periodicity & urgency** in the link-layer?
- S#1: Utilizing standard protocol metadata
  - Periodicity: Calculated by difference between timestamp of consecutive frame
  - Urgency: Calculated by frame delivery progress & SLA*

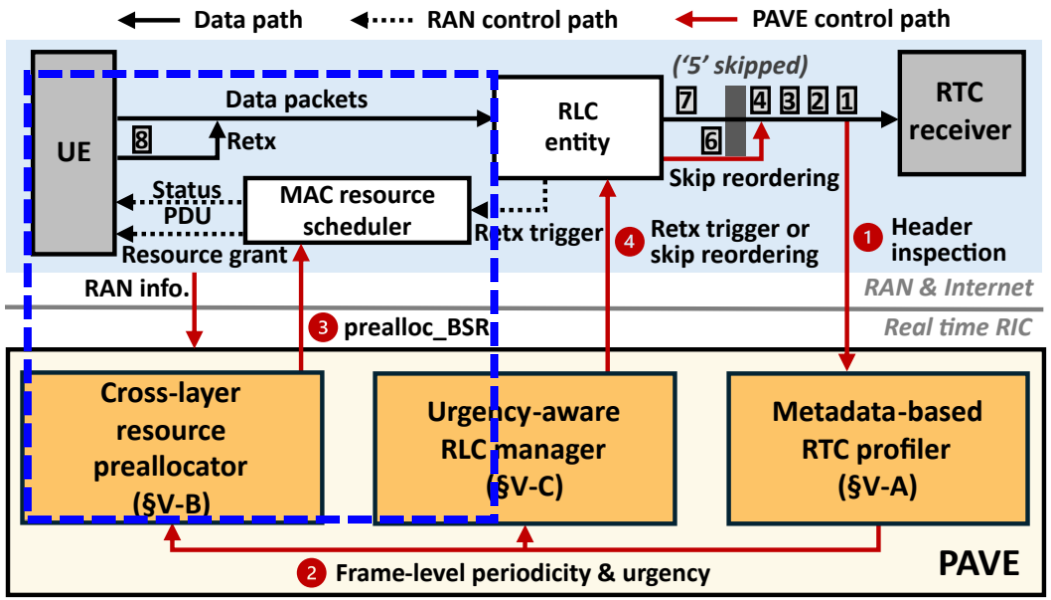


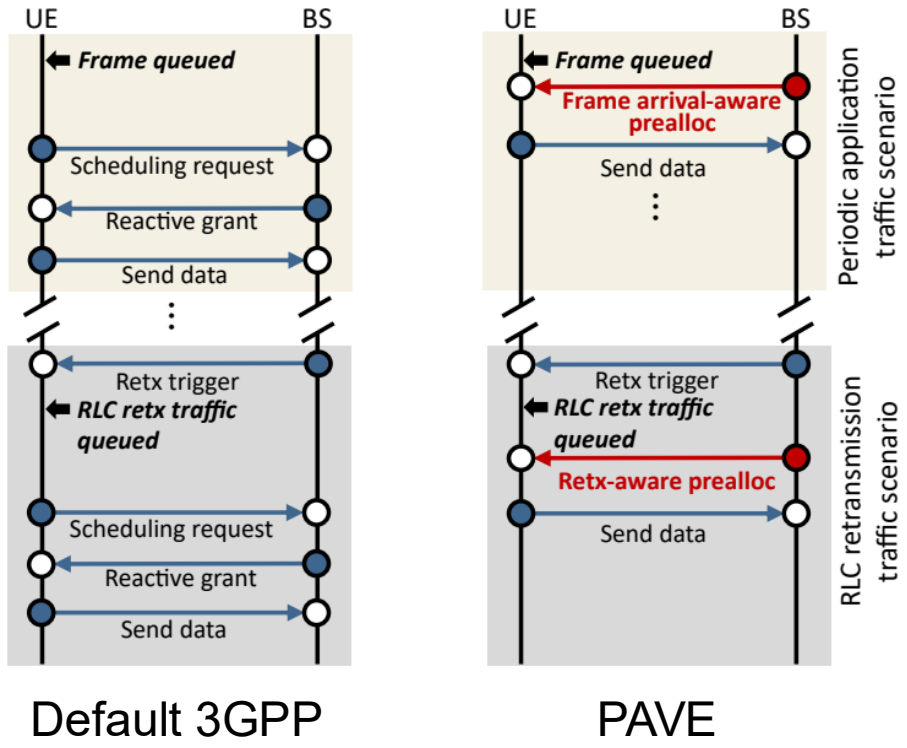


*Service Level Agreements

C#2: Reducing Delay without Efficiency Loss

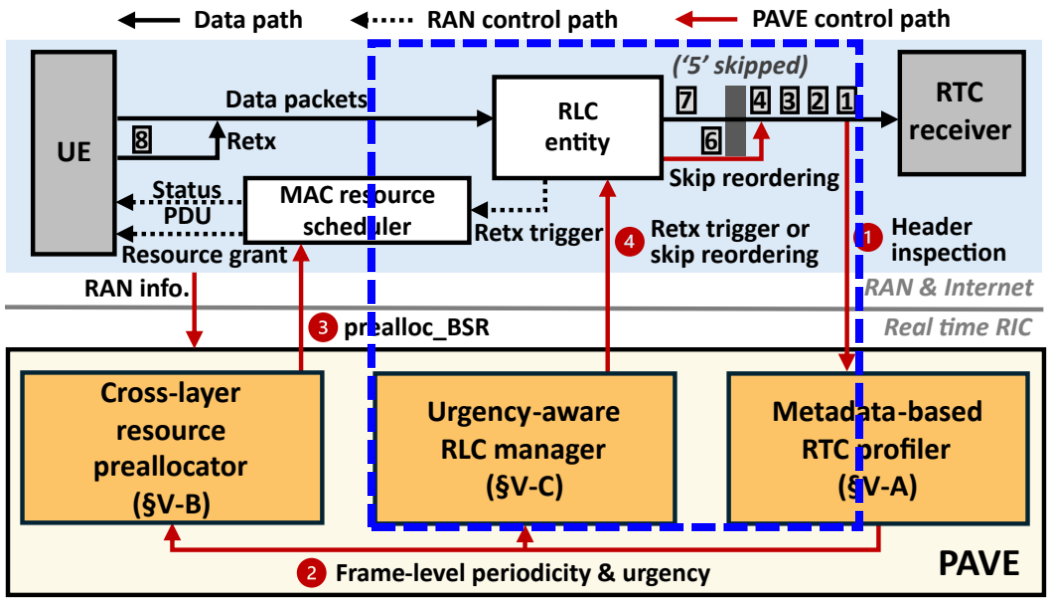
- How to optimize both **app QoE** and **radio resource efficiency**?
- S#2: Cross-layer info-assisted delay control
  - App traffic-induced resource preallocation
  - RLC retx-induced resource preallocation

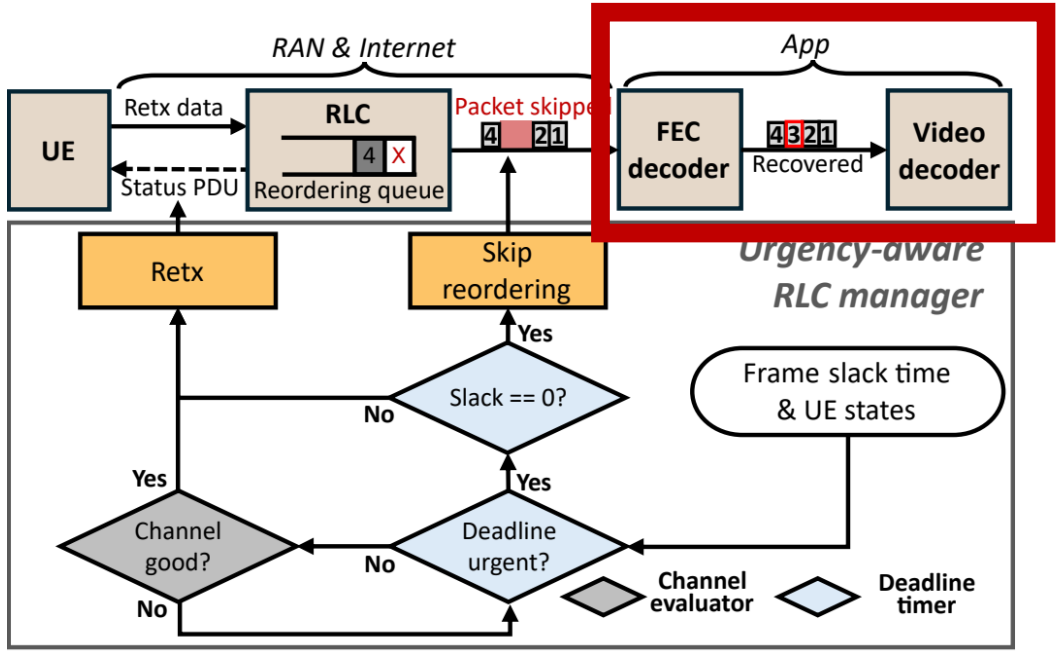




C#3: Protecting QoE over Channel Impairments

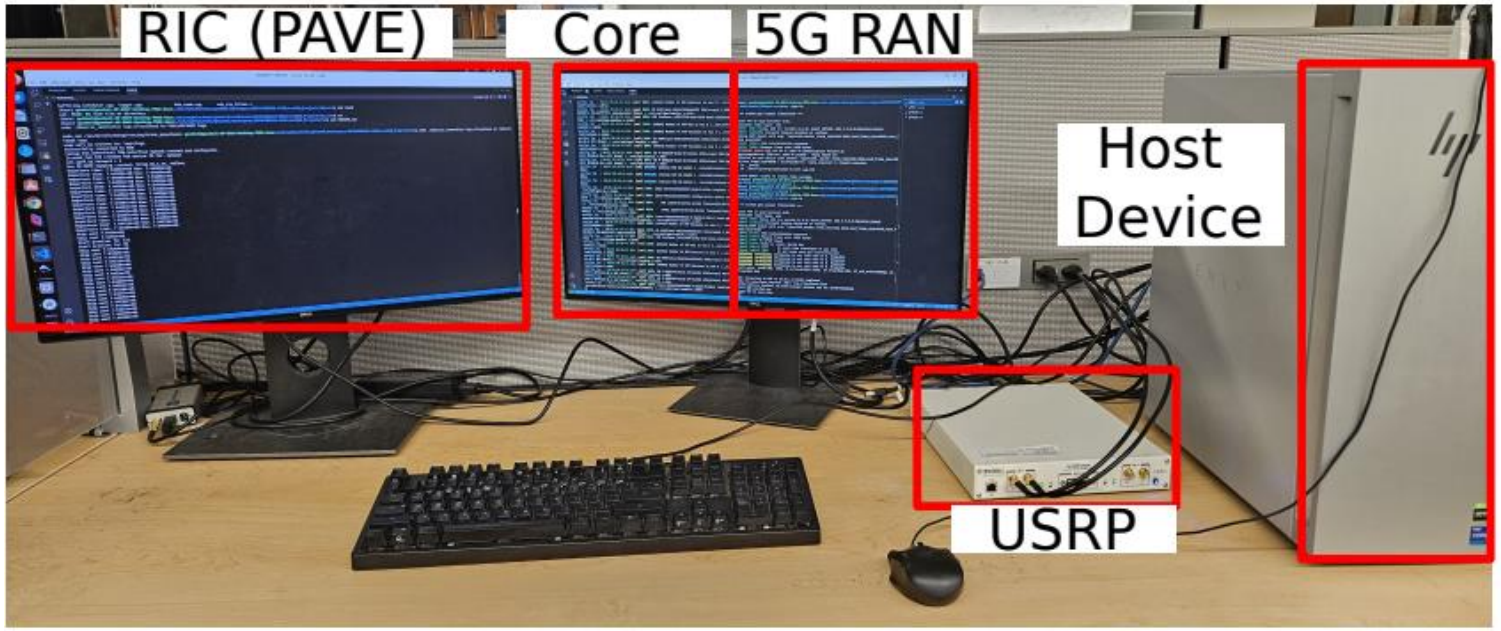
- How to **prevent bursty frame stalls** even when **channel impairment persists?**
- S#3: QoE-driven retransmission or frame skipping
  - Channel-aware retransmission
  - Skip reordering and send unordered packets when the deadline has been passed

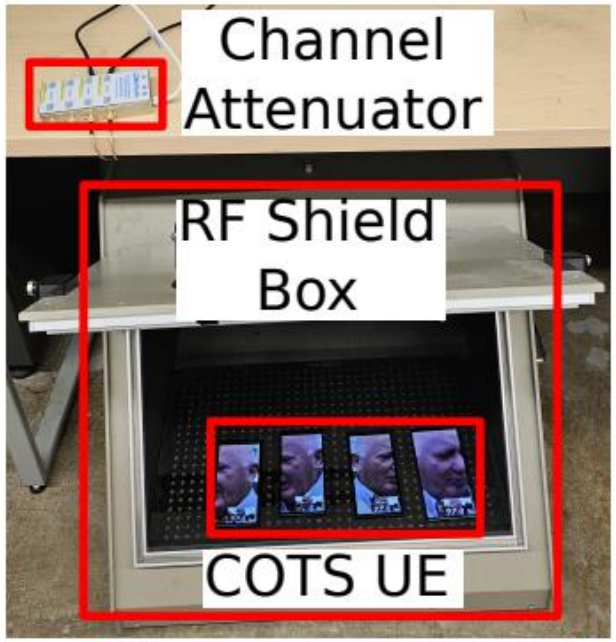




Implementation on Open-RAN Testbed

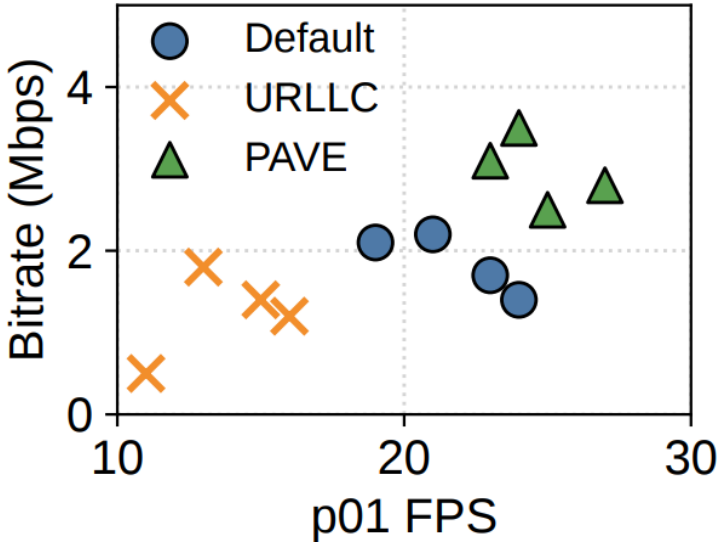
- Implemented on open-source radio suite srsRAN 5G and custom RIC based on EdgeRIC[1)]



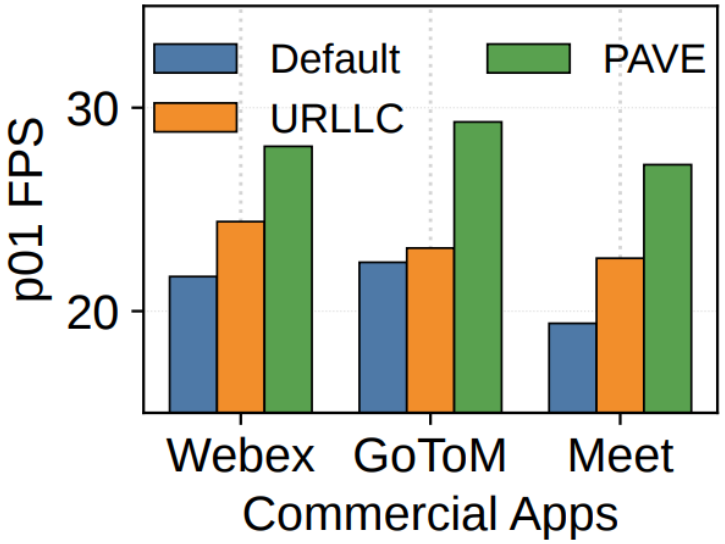


1) W. Ko et al., "EdgeRIC: Empowering Real-time Intelligent Optimization and Control in NextG Cellular Networks" USENIX NSDI 2024.

Evaluation: App Performance Enhancement



Multi-UE (WebRTC)

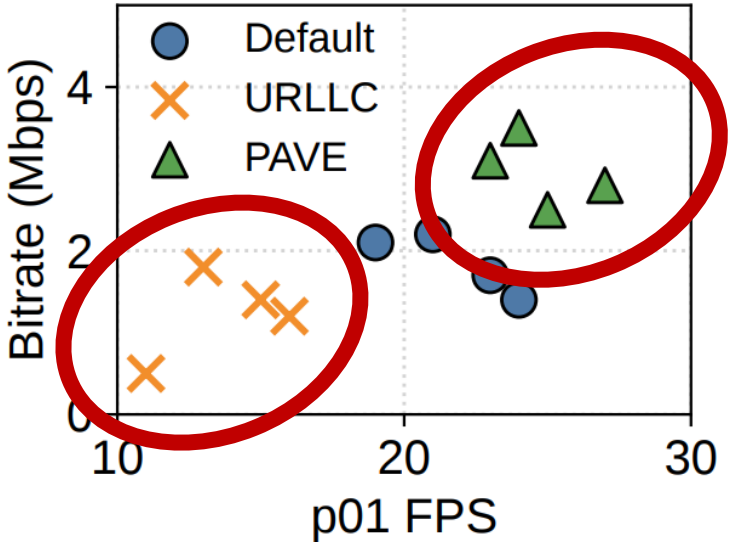


Commercial application (Single-UE)

* Walking mobility (SINR: 20.5±7.9 dB, Bandwidth: 23.5±6.8 Mbps), deadline: 150 ms

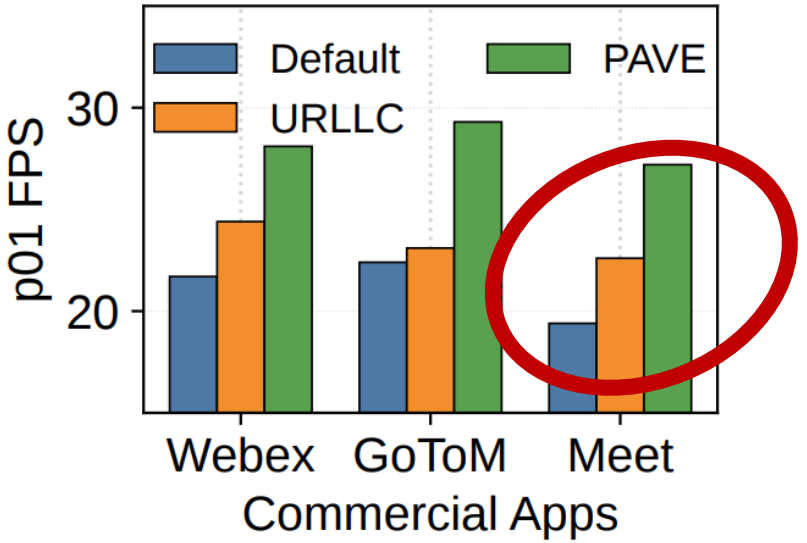
Evaluation: App Performance Enhancement

1.4X bitrate, 1.2X tail FPS



Multi-UE

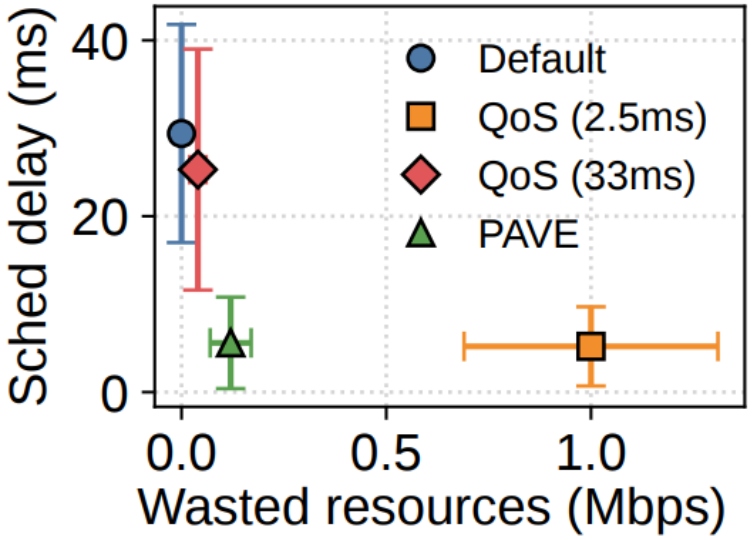
1.4X tail FPS



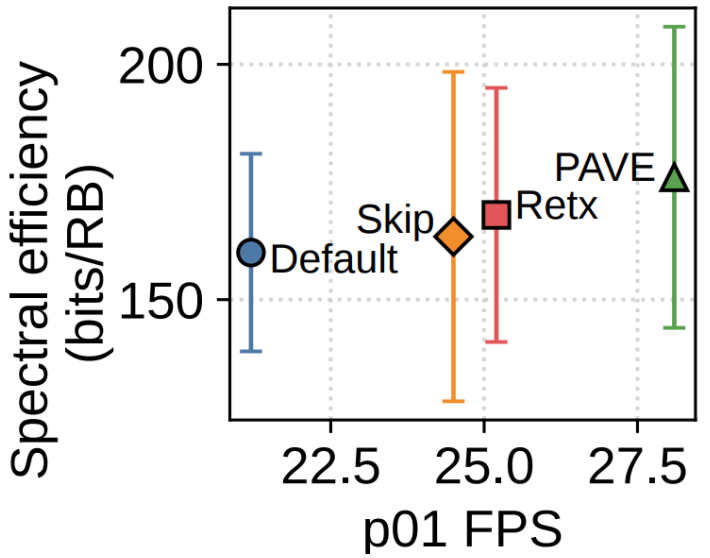
Commercial application (Single-UE)

* Walking mobility (SINR: 20.5±7.9 dB, Bandwidth: 23.5±6.8 Mbps), deadline: 150 ms

Evaluation: Radio Resource Efficiency



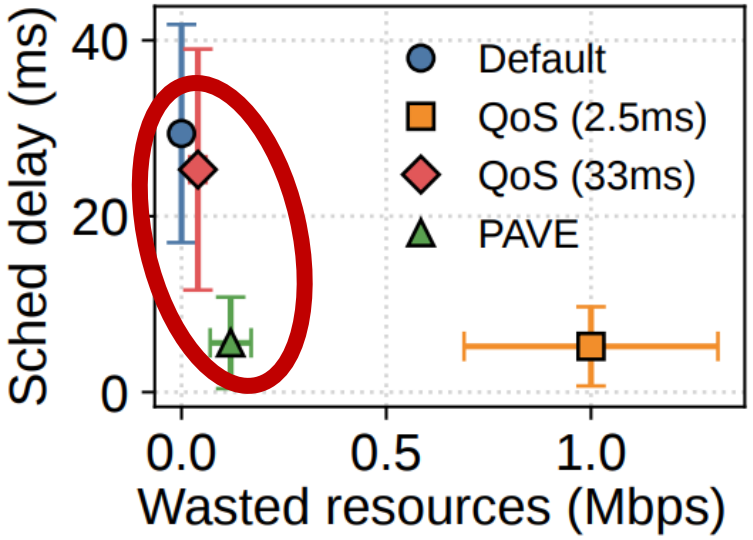
Scheduling delay of video frames



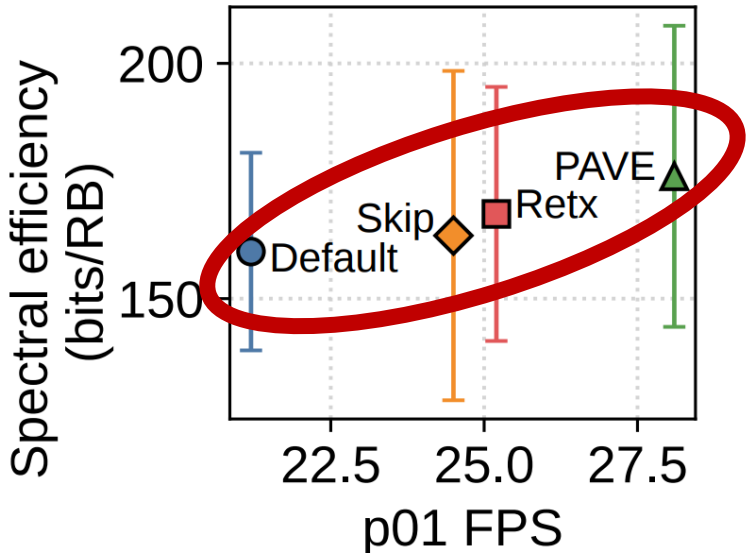
Spectral efficiency

Evaluation: Radio Resource Efficiency

Improved performance without compromising radio resource efficiency



Scheduling delay of video frames



Spectral efficiency

Conclusion & Takeaway

- Motivation
  - Mobile real-time video streaming suffers from **non-congestive RAN delay**
- Existing approach
  - Mostly focus on **congestive delay**
- Innovation
  - **PAVE** enables **seamless mobile video calls** with **app insights-driven RAN protocol design** that does not compromise radio resource efficiency with only RAN modifications
- Takeaway
  - **Non-congestive delay** can be the **major reason of delay** for RTC apps
  - **App-awareness is possible in RAN** with RTC header inspection

Thank you! Q&A

Email: goodsol.lee@nokia-bell-labs.com
Homepage: https://goodsollee.github.io/