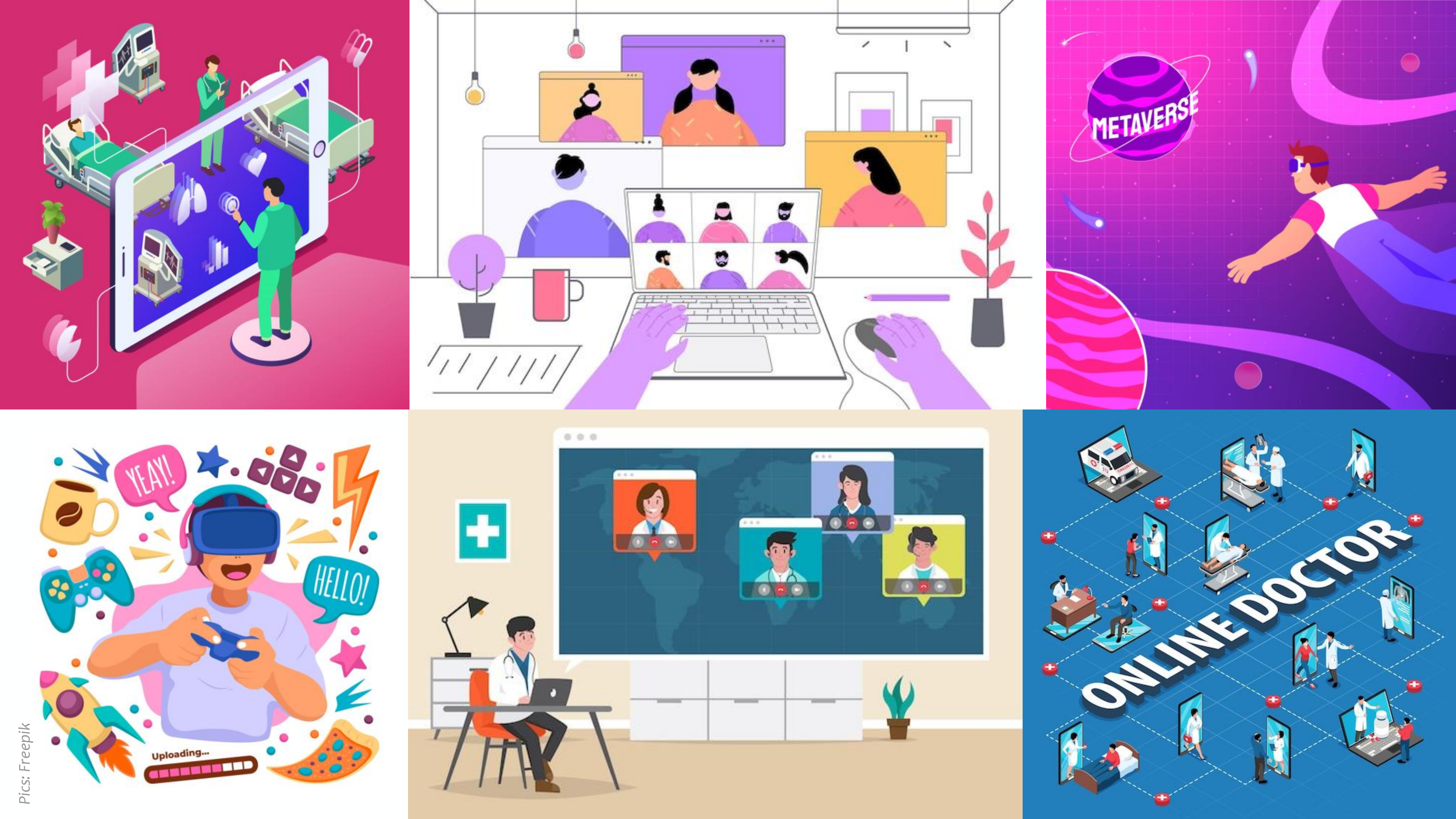
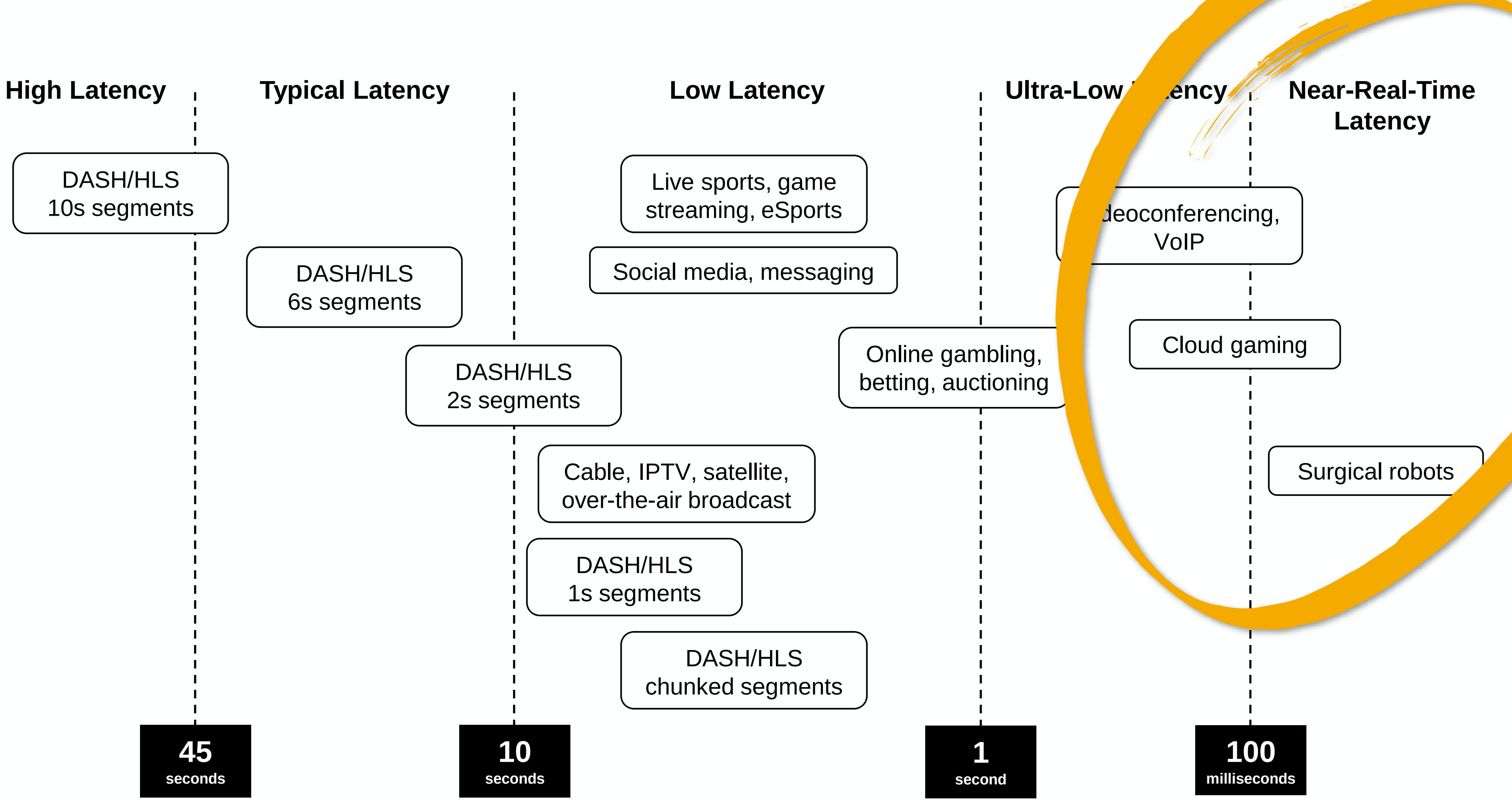
Bandwidth Prediction in Low-Latency Media Transport

ACM Mile-High Video, May 2023

Abdelhak Bentaleb, Mehmet N. Akcay, May Lim, Ali C. Begen and Roger Zimmermann



Pics: Freepik



High Latency

Typical Latency

Low Latency

Ultra-Low Latency

Near-Real-Time Latency

DASH/HLS 10s segments

DASH/HLS 6s segments

DASH/HLS 2s segments

Live sports, game streaming, eSports

Social media, messaging

Cable, IPTV, satellite, over-the-air broadcast

DASH/HLS 1s segments

DASH/HLS chunked segments

Online gambling, betting, auctioning

Videoconferencing, VoIP

Cloud gaming

Surgical robots

45 seconds

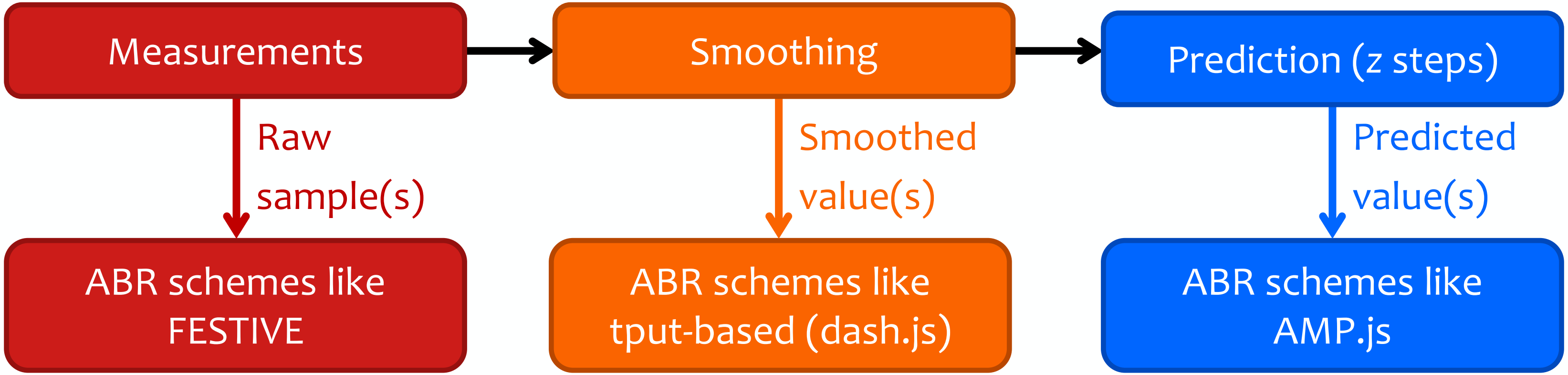
10 seconds

1 second

100 milliseconds

Reading: RFC 9317

Measurement, Smoothing and Prediction



Measurements

Smoothing

Prediction (z steps)

Raw sample(s)

Smoothed value(s)

Predicted value(s)

ABR schemes like FESTIVE

ABR schemes like tput-based (dash.js)

ABR schemes like AMP.js

The Challenge

Accurate bandwidth estimation/prediction

- Adapting the video bitrate dynamically

Superior overall QoE

- High quality

- Low latency (< 500 ms)

- Few (and short) stalls

The Considerations

Relation to congestion control (application-level performance vs. packet-level behavior)

Video and audio quality

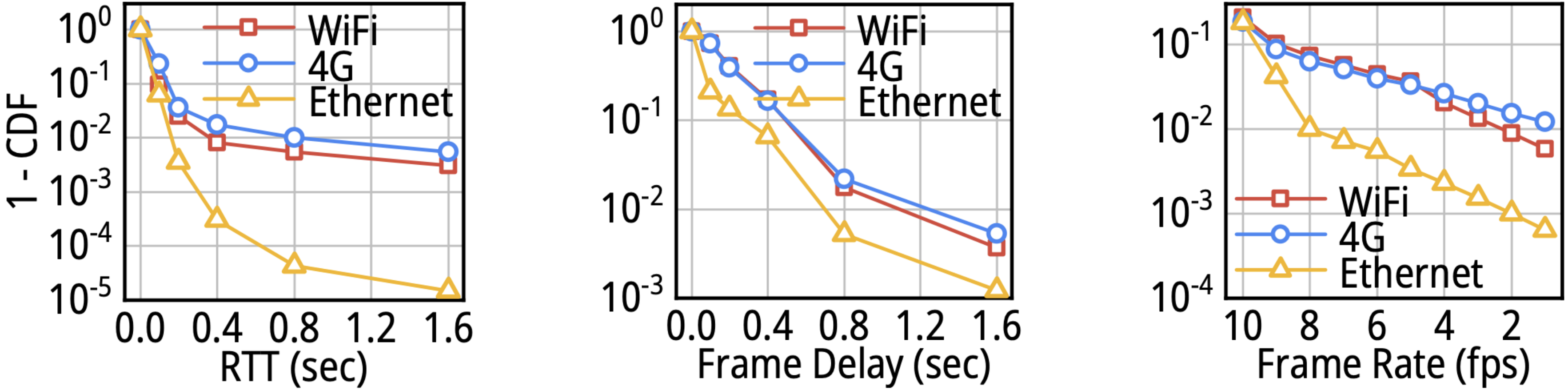
Frame loss and delay

Coarse-grained rate control

“Unpredictable” bandwidth fluctuations

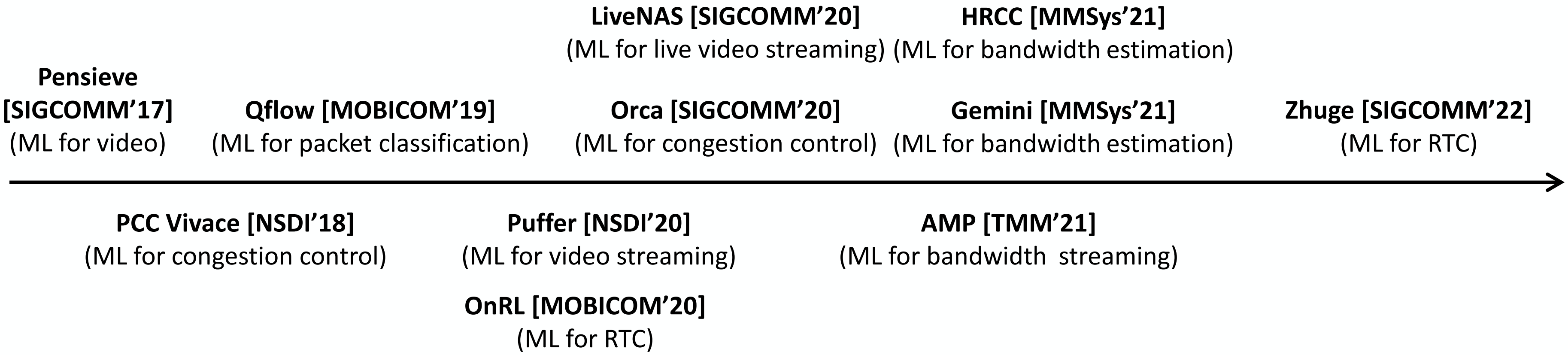
Motivation

Last-mile access network is the main culprit in RTC system performance



Source: Achieving consistent low latency for wireless real-time communications with the shortest control loop – ACM SIGCOMM'22

Learning-Based Solutions for Bandwidth Estimation and ABR



Pensieve [SIGCOMM'17]
(ML for video)

PCC Vivace [NSDI'18]
(ML for congestion control)

Qflow [MOBICOM'19]
(ML for packet classification)

Puffer [NSDI'20]
(ML for video streaming)

OnRL [MOBICOM'20]
(ML for RTC)

LiveNAS [SIGCOMM'20]
(ML for live video streaming)

Orca [SIGCOMM'20]
(ML for congestion control)

AMP [TMM'21]
(ML for bandwidth streaming)

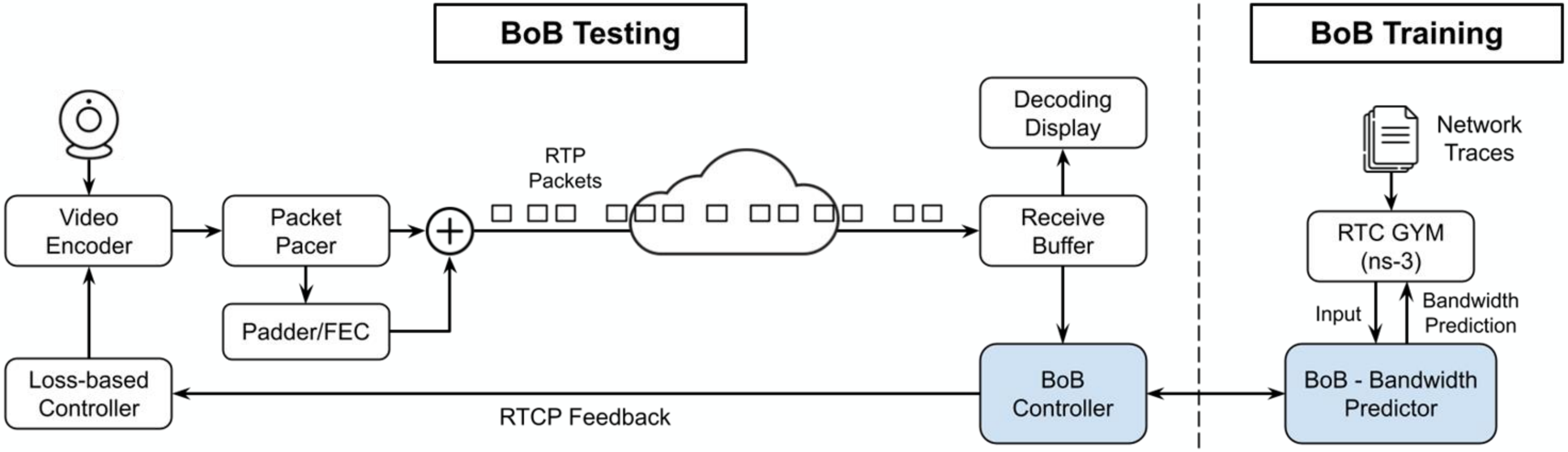
HRCC [MMSys'21]
(ML for bandwidth estimation)

Gemini [MMSys'21]
(ML for bandwidth estimation)

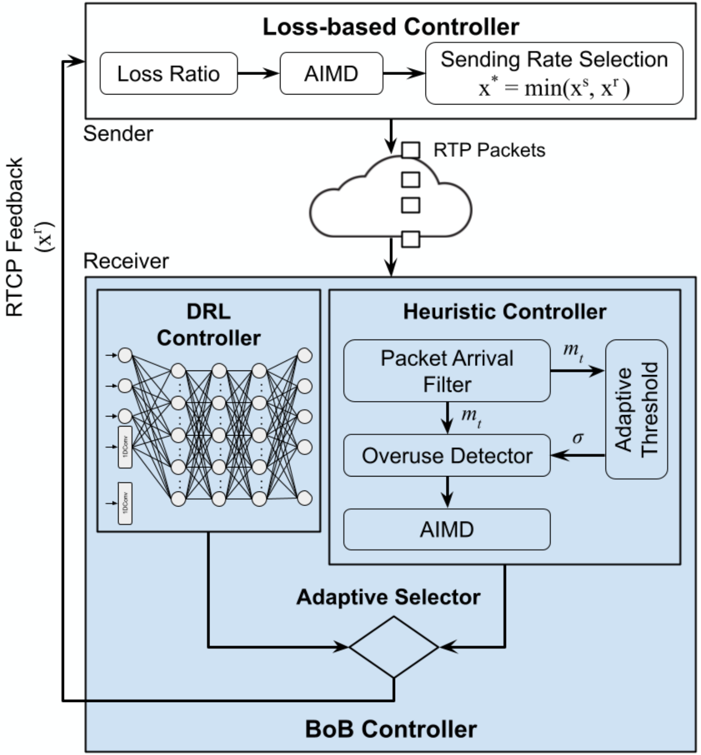
Zhuge [SIGCOMM'22]
(ML for RTC)

BoB (Bang-on-Bandwidth) Workflow

BoB combines a heuristic-based controller (inspired by the GCC algorithm) with a DRL controller



Receiver-Side BoB Controller



- BoB learning
  - BoB is added to AlphaRTC GYM simulator (ns-3 with WebRTC implementation)
  - BoB NN input: receiving rate, packet delay and loss and most recent bandwidth prediction samples
  - BoB NN output (every 200 ms): Predicted bandwidth
  - DRL architecture: actor-critic with PPO and Adam optimizer
- BoB testing
  - Sender: Loss-based controller
    - Compute target sending rate x*
  - Receiver: BoB controller
    - Compute bitrate (x^r) based on the BoB prediction

Heuristic Controller

- Key idea
  - Estimate the bandwidth based on aggregated per-packet RTP feedback
- Strength
  - Good convergence in some scenarios
  - Easy to implement
  - Low overhead
- Weakness
  - Fail to achieve consistent high performance in diverse network environments

DRL Controller

- Key idea
  - Predict the bandwidth based on NN model trained using various network conditions
- Strength
  - Strong adaptability to diverse network environments
- Weakness
  - Not easy to deploy
  - Sometimes it may not converge

BoB Adaptive Selector

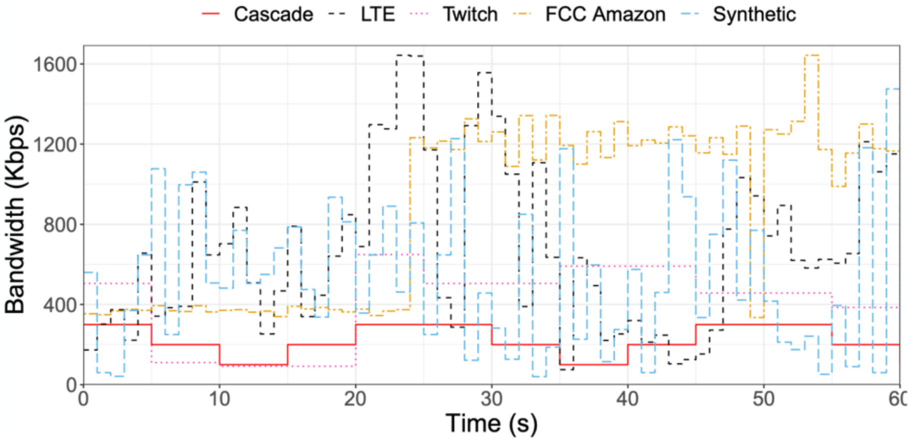
Decide when to switch between the heuristic and DRL controllers

- At the beginning of an RTC session
  - Use the heuristic controller
- During the RTC session
  - Compute the absolute difference (diff) between
    - bandwidth predicted by the heuristic controller
    - bandwidth predicted by the DRL controller
  - Compute the average predicted bandwidth (avg) value based on both controllers
  - If diff/avg is less than 30%
    - Use the DRL controller
  - Otherwise
    - Use the heuristic controller

Implementation and Evaluation

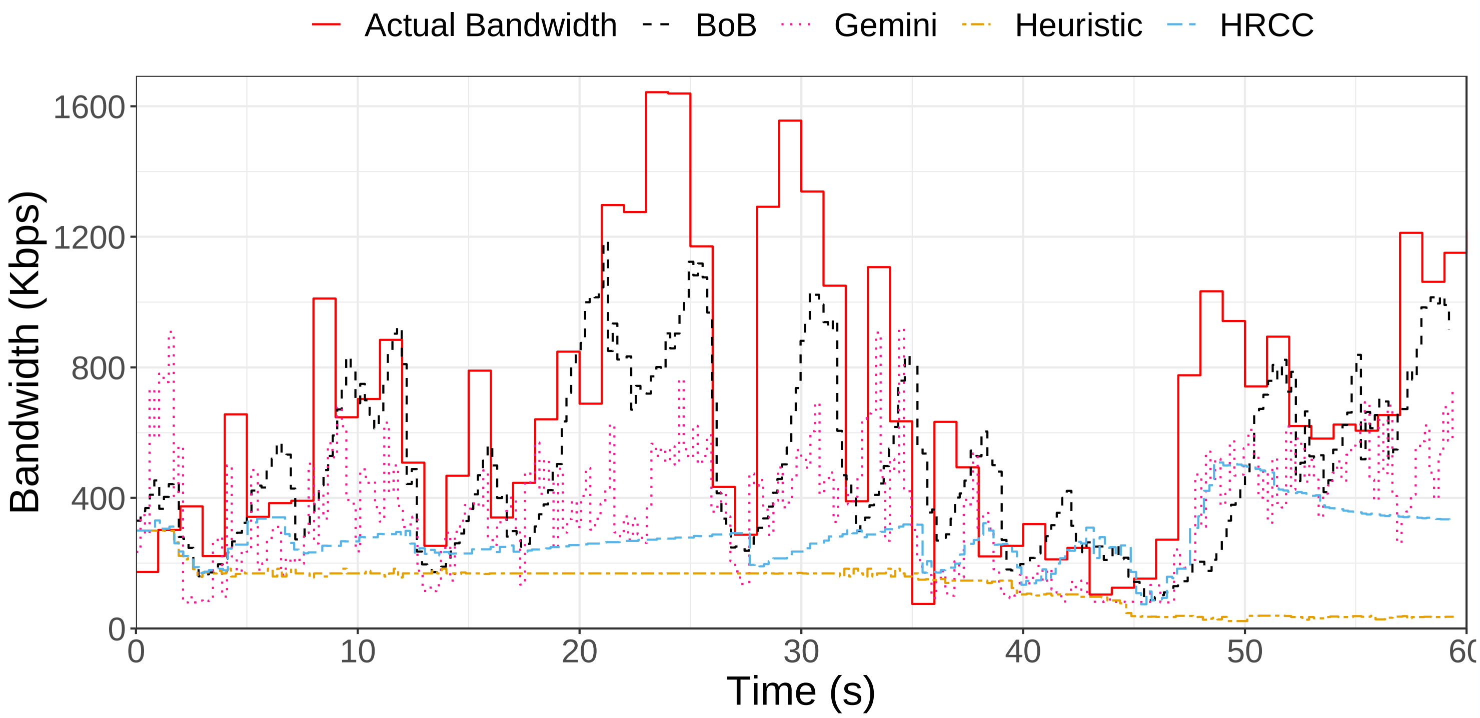
- Implemented in Microsoft's AlphaRTC
  - Code available on GitHub
- Network profiles:
  - Cascade
  - LTE
  - Twitch
  - FCC Amazon
  - Synthetic
- Video sample:
  - BBB with 24 fps
- Comparison: BoB vs. heuristic-based, Gemini and HRCC
  - Further details in the paper

- Evaluation metrics
  - sMAPE (symmetric mean abs. % error) and accuracy
  - Network score: a combination of delay score, loss score and receiving rate score
  - Video score: VMAF
  - Total Score: Network score + video score



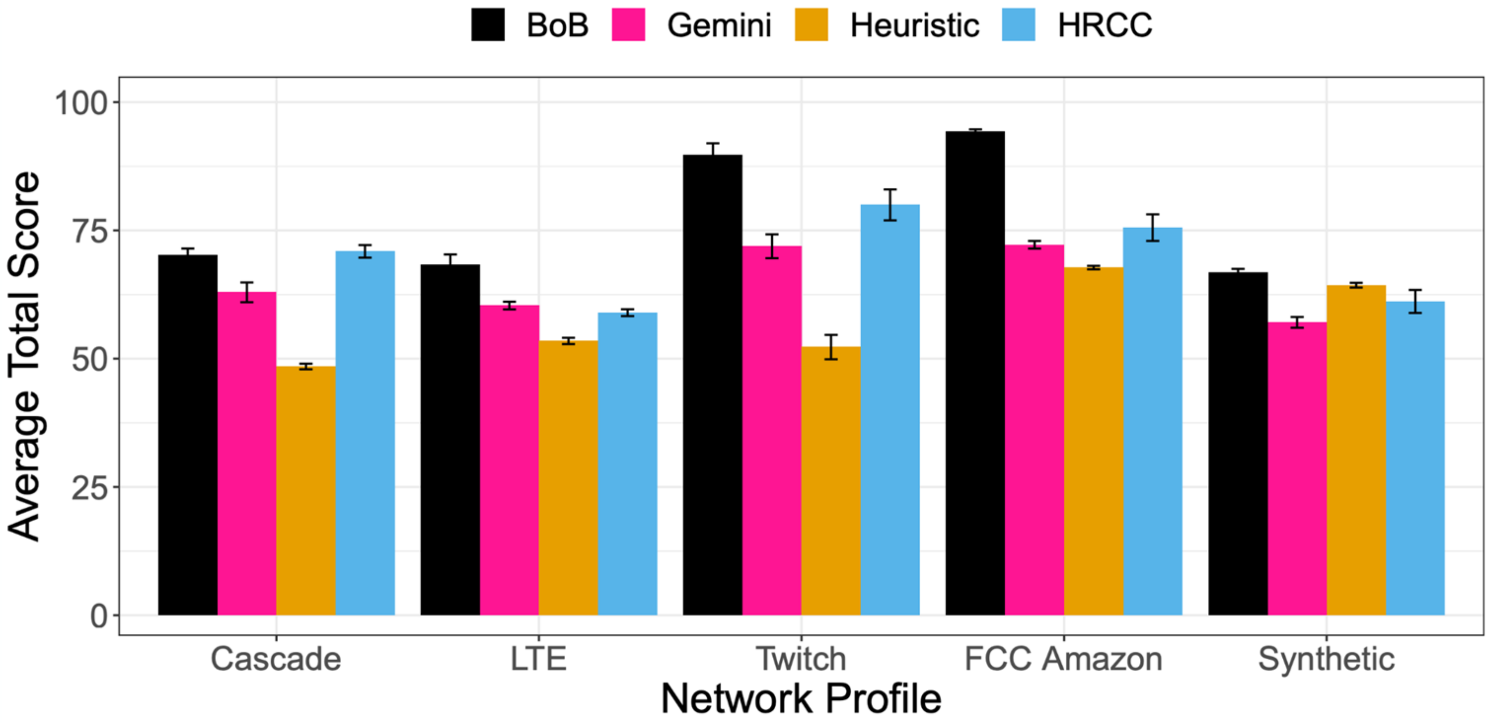
Actual and Predicted Bandwidth

LTE



Total Score

Average for different network profiles



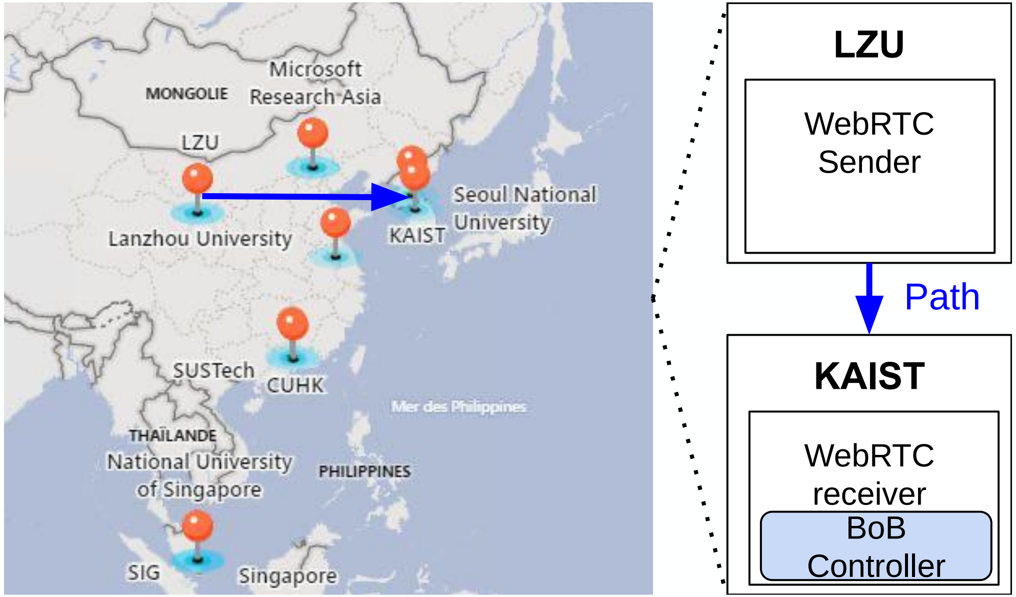
Internet-Based Setup

OpenNetLab (https://opennetlab.org/)

	Network Profile	Sender Node	Receiver Node	Path Bandwidth	Avg. RTT
A	High Bandwidth	Lanzhou (Wired)	Seoul (Wired)	> 100 Mbps	30 ms
B	Medium Bandwidth	Beijing (Mobile)	Hong Kong (Wired)	2-3 Mbps	62 ms
C	Low Bandwidth	Beijing (Weak wireless)	Hong Kong (Wired)	< 1 Mbps	55 ms

Internet-Based Setup

OpenNetLab (https://opennetlab.org/)



LZU

WebRTC Sender

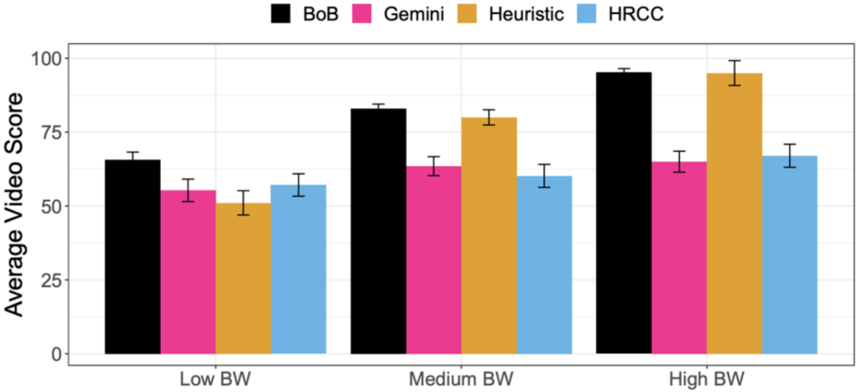
Path

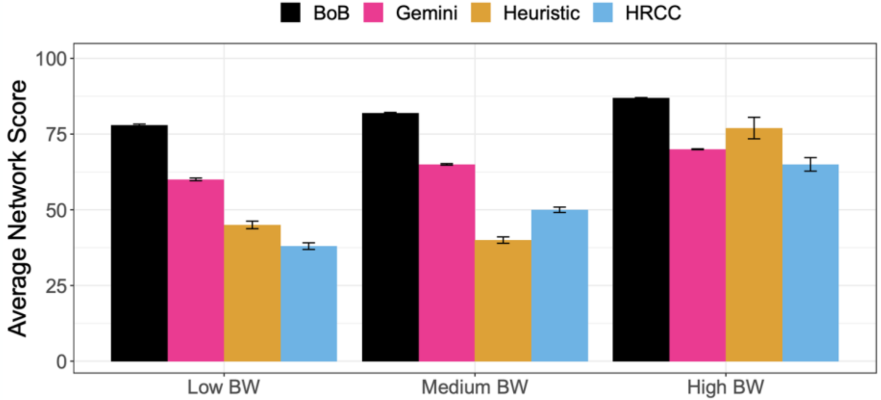
KAIST

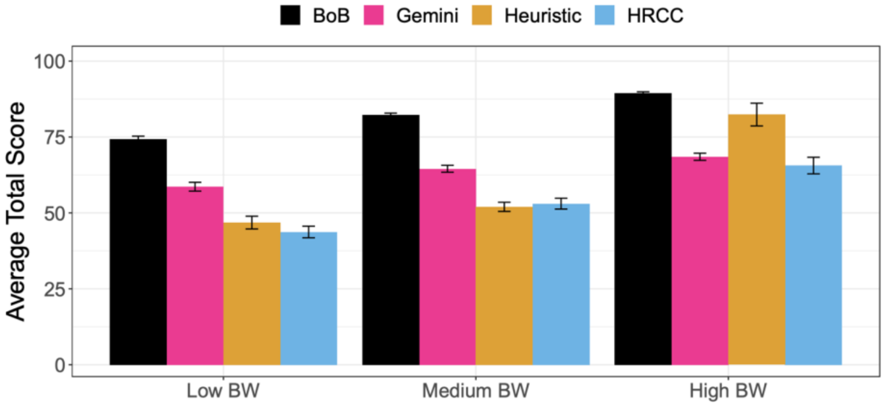
WebRTC receiver

BoB Controller

Video, Network and Total Scores







AMAAAS*

- Download and use the BoB code
  - GitHub: https://github.com/NUStreaming/BoB
  - Use and test it, open issues and report bugs
- Download the BoB paper (open access)
  - https://doi.org/10.1109/TMM.2022.3216456

BoB code

BoB paper

- Reach out to any of us for questions
  - abdelhak.bentaleb@concordia.ca
  - necmettin.akcay@ozu.edu.tr
  - maylim@comp.nus.edu.sg
  - ali.begen@ozyegin.edu.tr
  - rogerz@comp.nus.edu.sg

This deck (and many others) are posted at https://ali.begen.net

* AMAAAS: Ask me almost anything about streaming