Oboe: Auto-tuning Video ABR Algorithms to Network Conditions

Zahaib Akhtar★, **Yun Seong Nam★**, Ramesh Govindan, Sanjay Rao, Jessica Chen, Ethan Katz-Bassett, Bruno Ribeiro, Jibin Zhan, Hui Zhang

★: Co-primary authors

USC University of Southern California
PURDUE UNIVERSITY
COLUMBIA UNIVERSITY
University of Windsor
CONVIVA

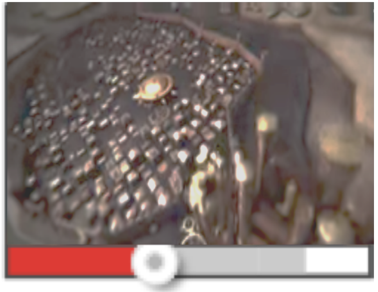
Internet Video Streaming Today

- Internet video is delivered over:
  - Heterogeneous networks: WiFi, wired, 3G/4G LTE
  - Highly varying or challenging network conditions

Internet Video Streaming Today

- Internet video is delivered over:
    - Heterogeneous networks: WiFi, wired, 3G/4G LTE
    - Highly varying or challenging network conditions

- Quality of experience (QoE) issues are common place



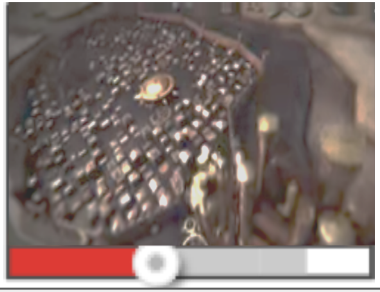
Low quality

Rebuffering

Internet Video Streaming Today

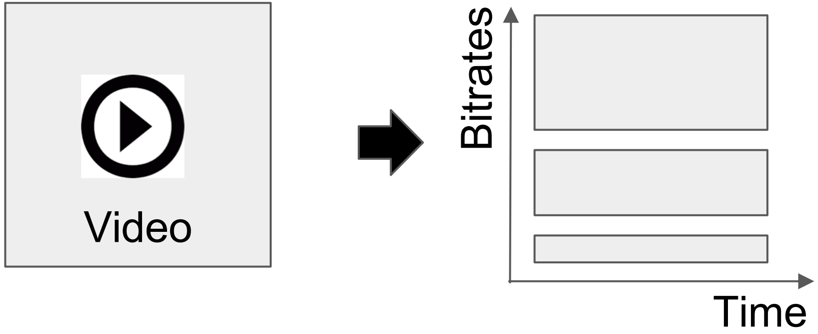
- Internet video is delivered over:
  - Heterogeneous networks: WiFi, wired, 3G/4G LTE
  - Highly varying or challenging network conditions

- Quality of experience (QoE) issues are common place



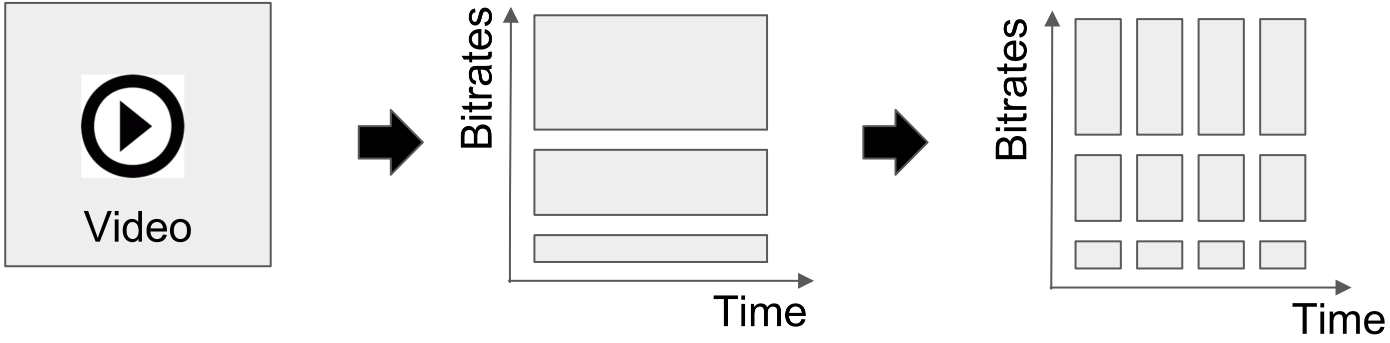
Low QoE adversely impacts user engagement and revenue

Background: Adaptive Bitrate Streaming



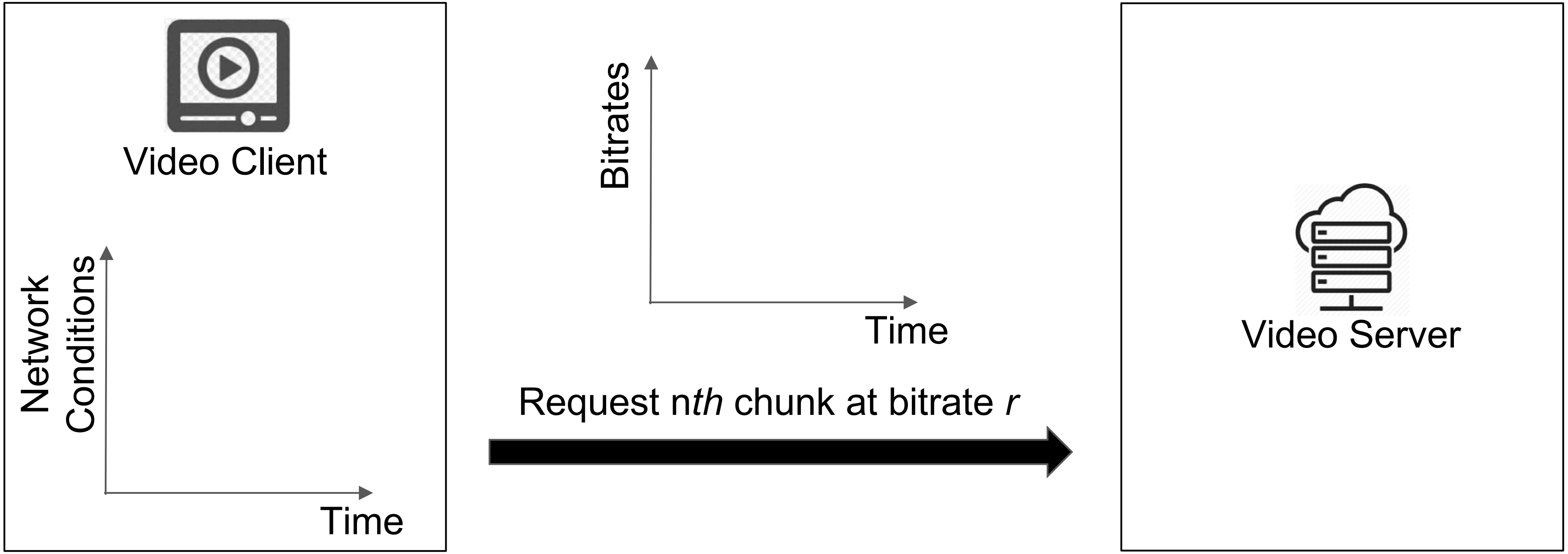
A video clip is encoded
with multiple qualities (bitrates)

Background: Adaptive Bitrate Streaming

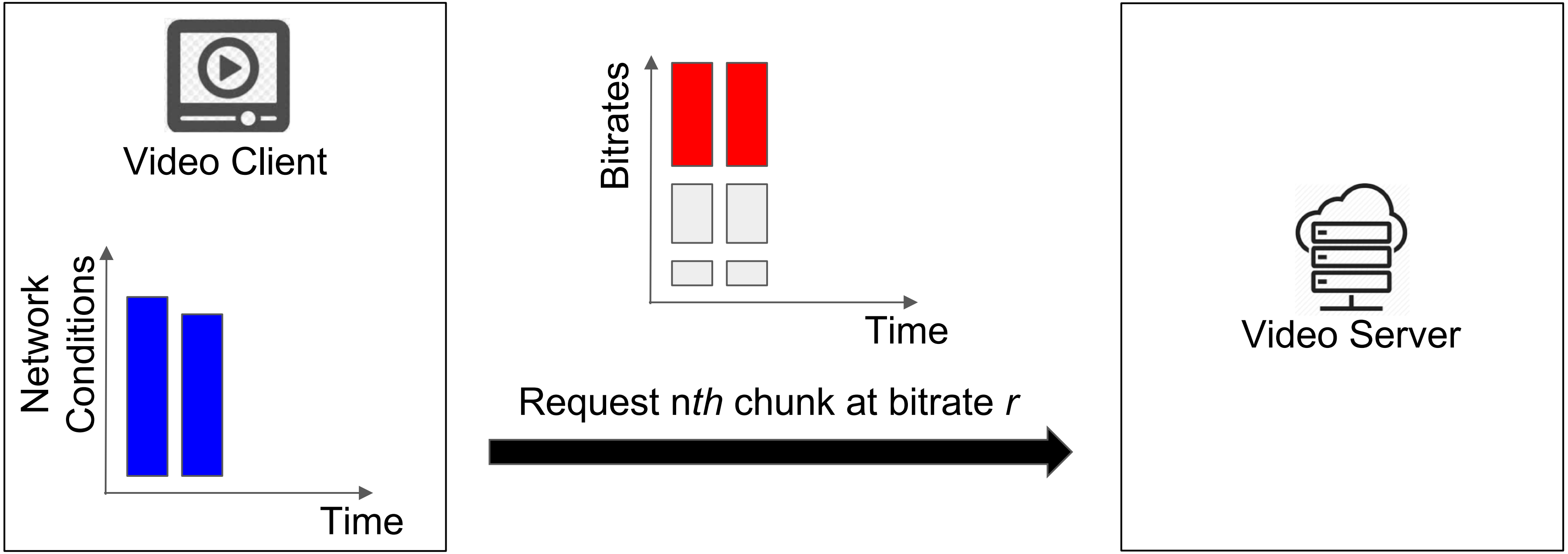


Each bitrate is split into chunks

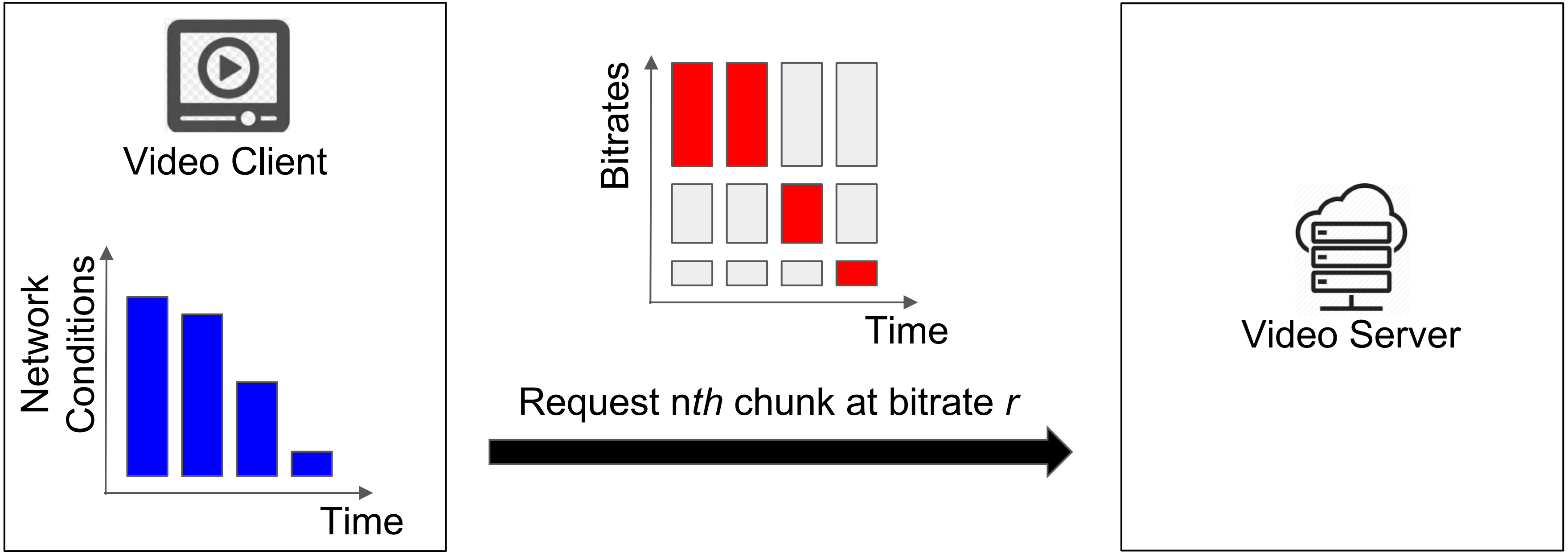
Background: Adaptive Bitrate Streaming



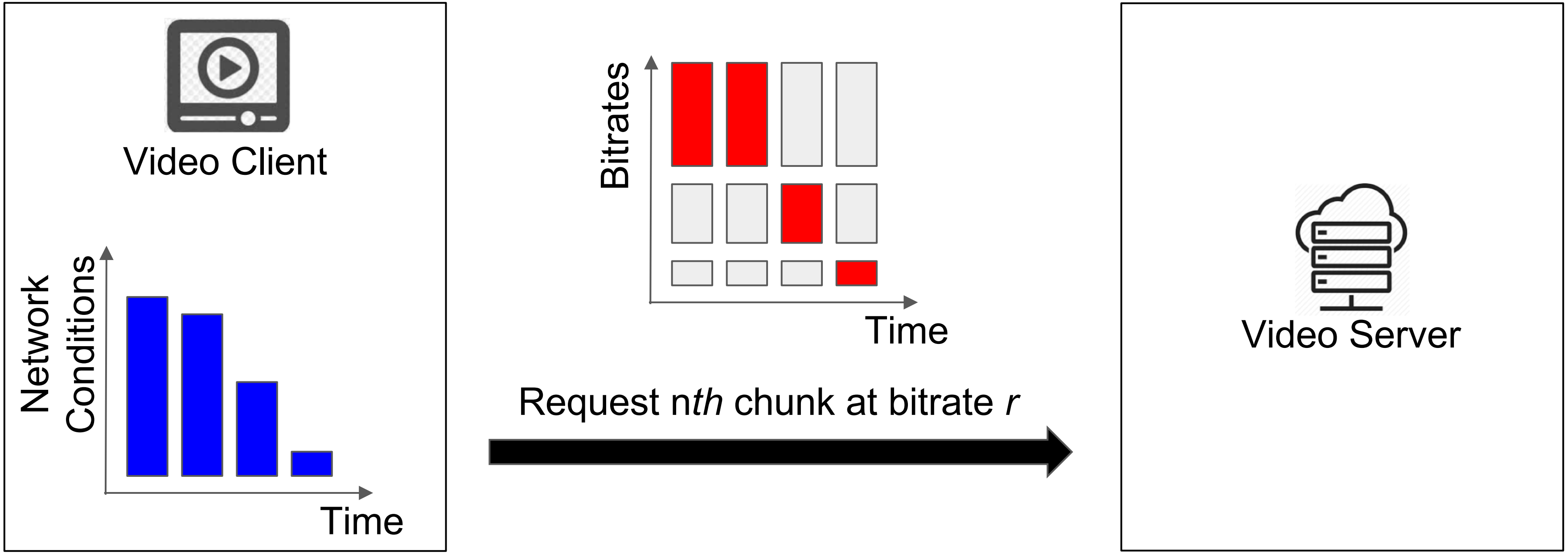
Background: Adaptive Bitrate Streaming



Background: Adaptive Bitrate Streaming

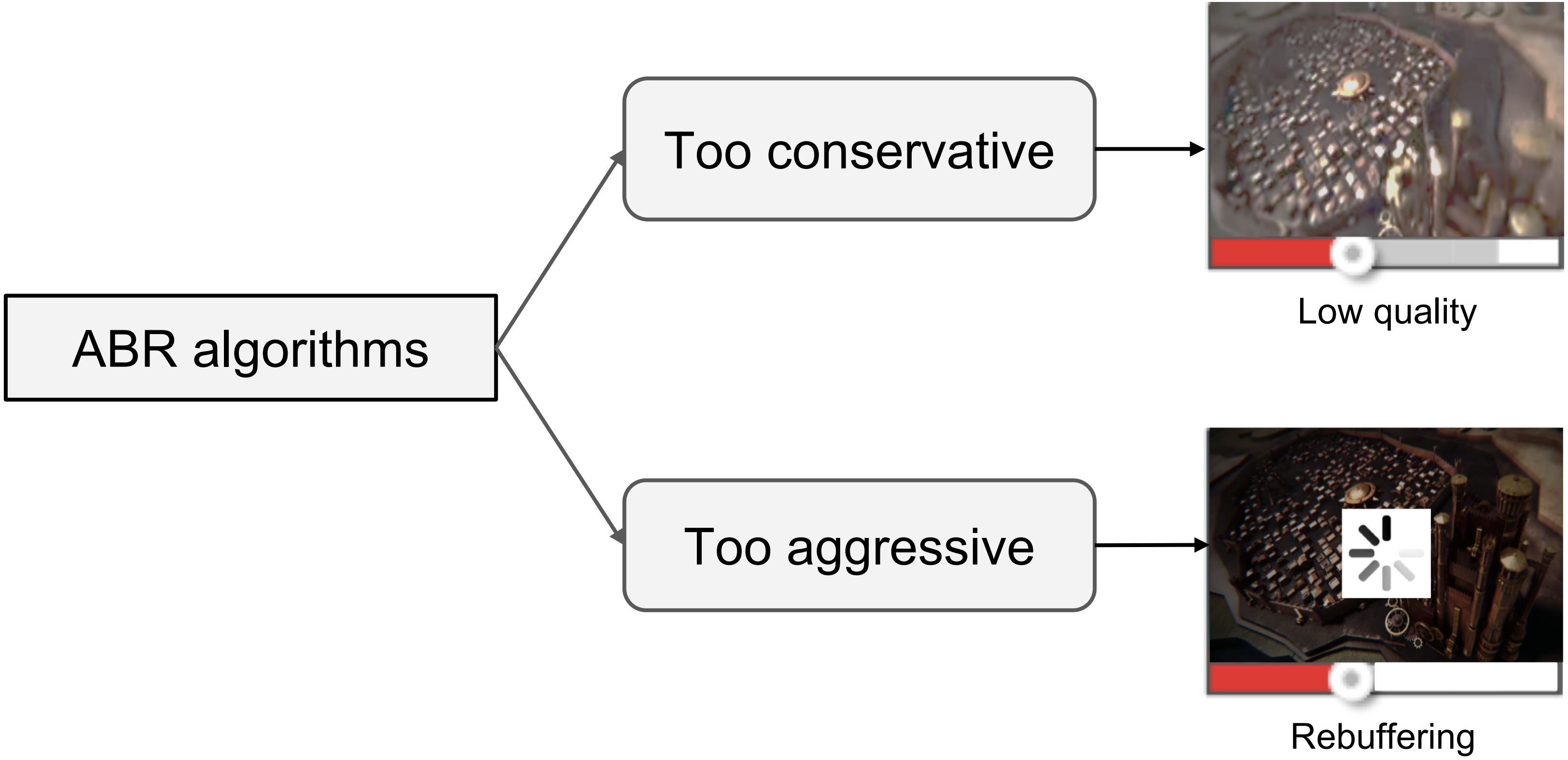


Background: Adaptive Bitrate Streaming

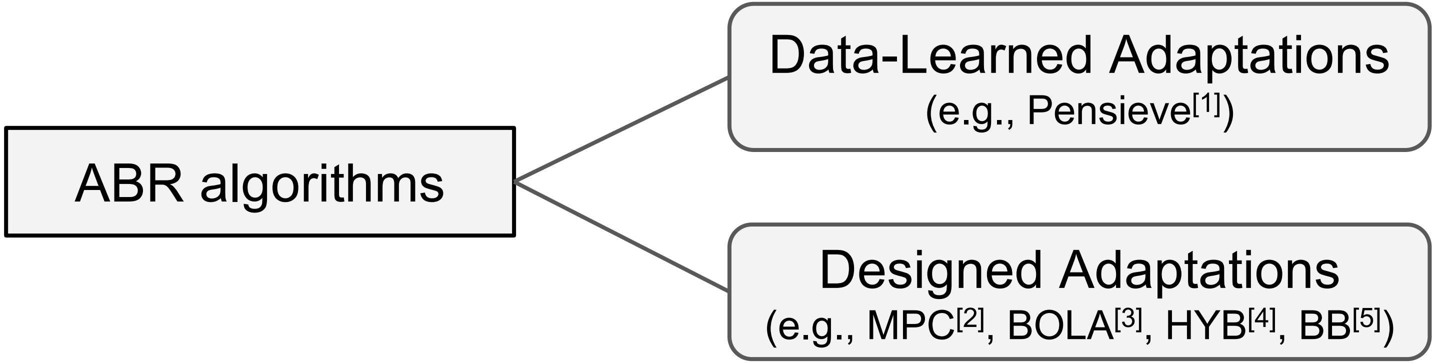


Adaptive Bitrate Algorithms(ABR)

Background: Adaptive Bitrate Streaming



Background: ABR algorithms



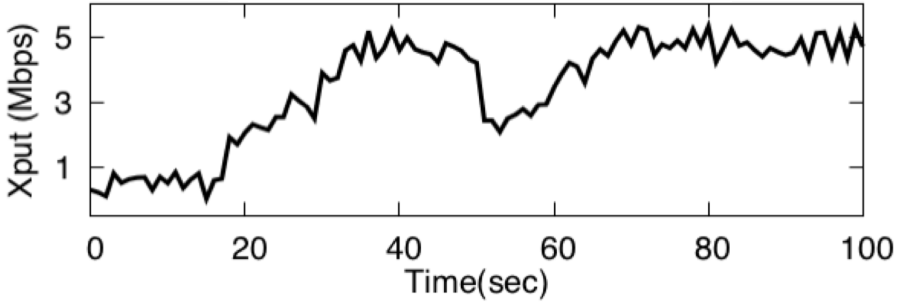
Performance of Designed Adaptation based ABRs critically depends on **configurable parameters**

[1] Hongzi Mao, et al., SIGCOMM, 2017.
[2] Xiaoqi Yin, et al., SIGCOMM, 2015.
[3] Kevin Spiteri, et al., INFOCOM, 2016.
[4] An ABR algorithm that's widely used in industry.
[5] Te-Yuan Huang, et al., SIGCOMM, 2014.

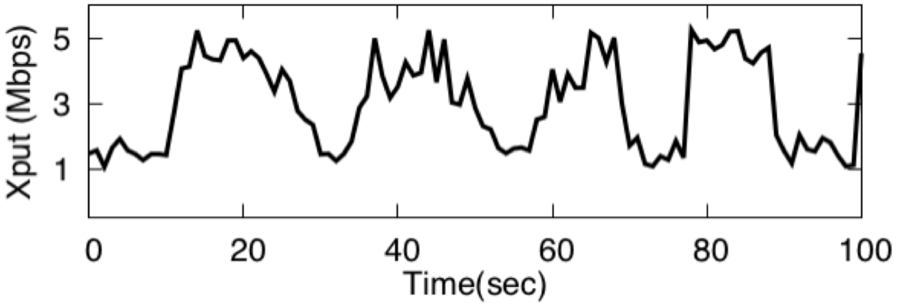
Parameters are sensitive to network conditions

Parameters are sensitive to network conditions

Network Condition A

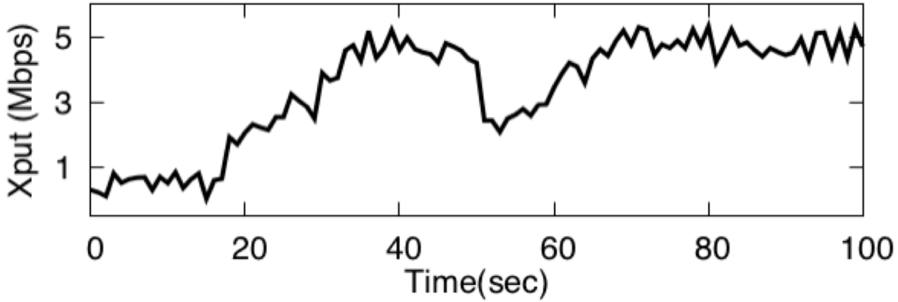


Network Condition B

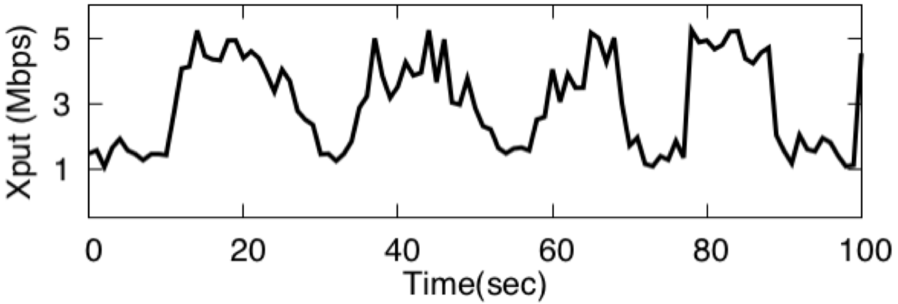


Parameters are sensitive to network conditions

Network Condition A

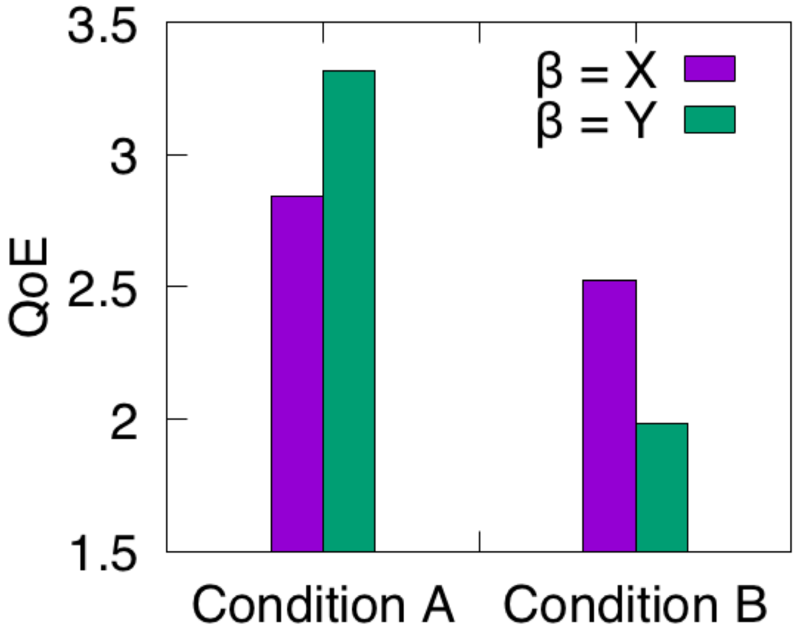


Network Condition B

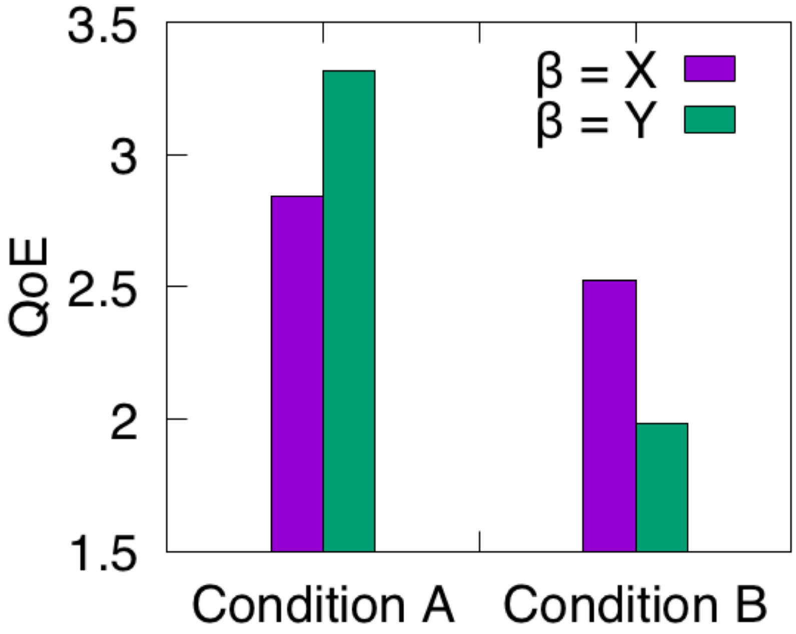


Widely deployed ABR algorithm with parameter β

Parameters are sensitive to network conditions



Parameters of ABRs must be set in a manner sensitive to network conditions



The problem with ABR algorithms

ABR algorithms	Parameter
MPC	Discount factor *d*
BOLA	Parameter γ
HYB	Safety margin β
BB	Reservoir *r*

ABR algorithms use **fixed parameter value** or **simple heuristic**

The problem with ABR algorithms

ABR algorithms	Parameter
MPC	Discount factor d
BOLA	Parameter γ
HYB	Safety margin β
BB	Reservoir r

ABR algorithms use
fixed parameter value
or **simple heuristic**

Do not perform well across all network conditions

Goal of our work

Design a system to make ABR algorithms work better over a wide range of network conditions

Key Challenges

How to model network conditions?

How to find the best parameter for a given condition?

How to adapt to changes in network conditions?

Contributions

How to model network conditions?

- Leverage **stationarity** of network connections

How to find the best parameter for a given condition?

- **Pre-compute** offline

How to adapt to changes in network conditions?

- **Detect change points online** and adjust parameters

Contributions

How to model network conditions?

- Leveraging **stationarity** of network connections

Our system, Oboe improves state-of-art ABRs (MPC, BOLA and HYB) upto 38% and outperforms Pensieve by 24%

- **Online change point detection** and adjusting ABR algorithms in online

Key Challenges

How to model network conditions?

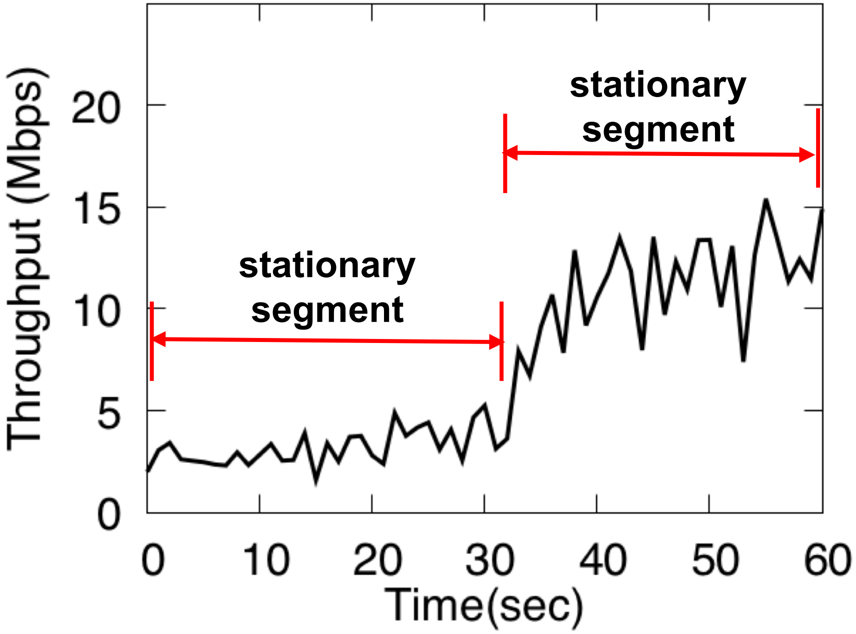
How to find the best parameter for a given condition?

- Oboe Offline Stage

How to adapt to changes in network conditions?

- Oboe Online Stage

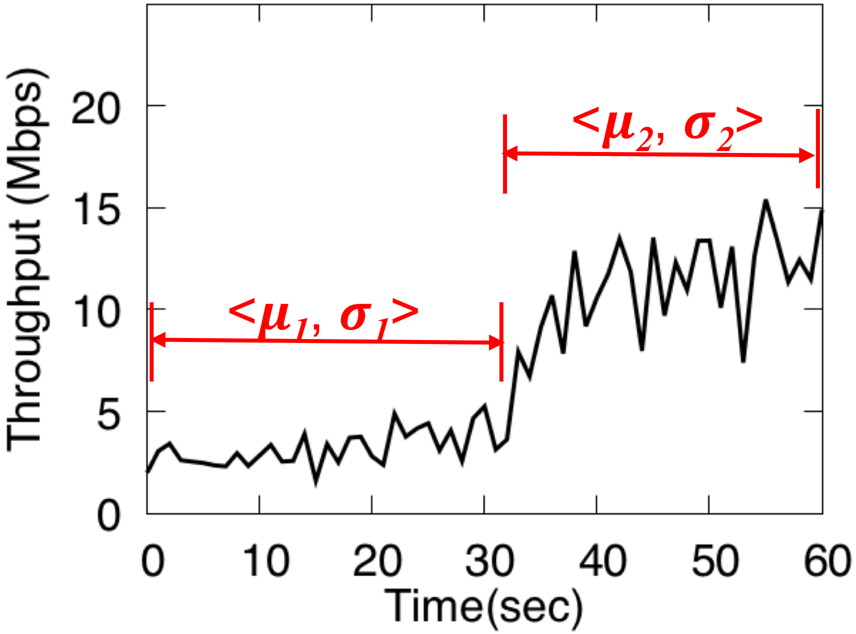
Modeling network conditions



TCP connection throughput can be modeled as a **piecewise stationary**[6-10] sequence of **network states**

[6] Hari Balakrishnan, et. al. SIGMETRICS, 1997
[7] James Jobin, et. al. INFOCOM, 2004
[8] Dong Lu, et. al. ICDCS, 2005
[9] Guillaume Urvoy-Keller. PAM, 2005.
[10] Yin Zhang, et al. IM, 2001

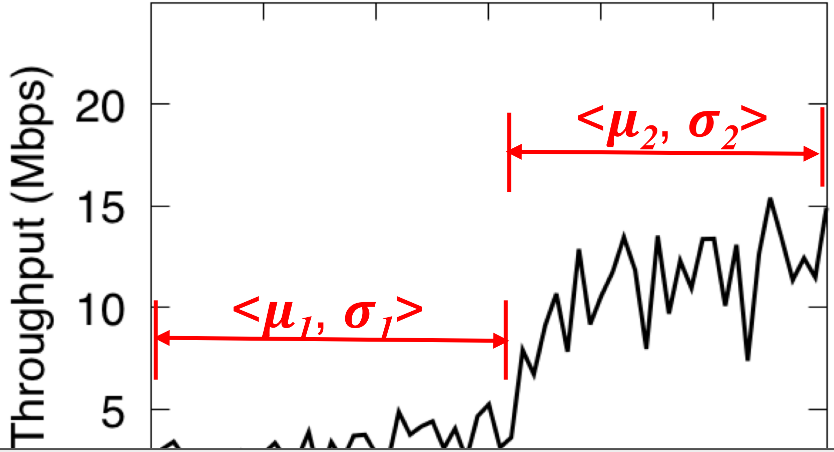
Modeling network conditions



Network state ***s*** = $\langle \mu_s, \sigma_s \rangle$

where μ_s is the mean and σ_s is the standard deviation of throughput

Modeling network conditions



Network state $\boldsymbol{s}$ = $<\mu_s, \sigma_s>$

where μ_s is the mean and σ_s is the standard deviation of throughput

Key idea : Use the best parameter for each network state

Key Challenges

How to model network conditions?

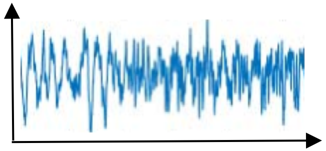
How to find the best parameter for each network state?

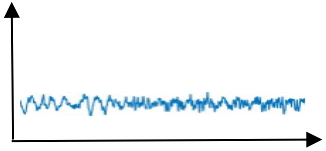
- Oboe Offline Stage

How to adapt to changes in network state?

- Oboe Online Stage

Finding the best parameter : Oboe Offline Step 1

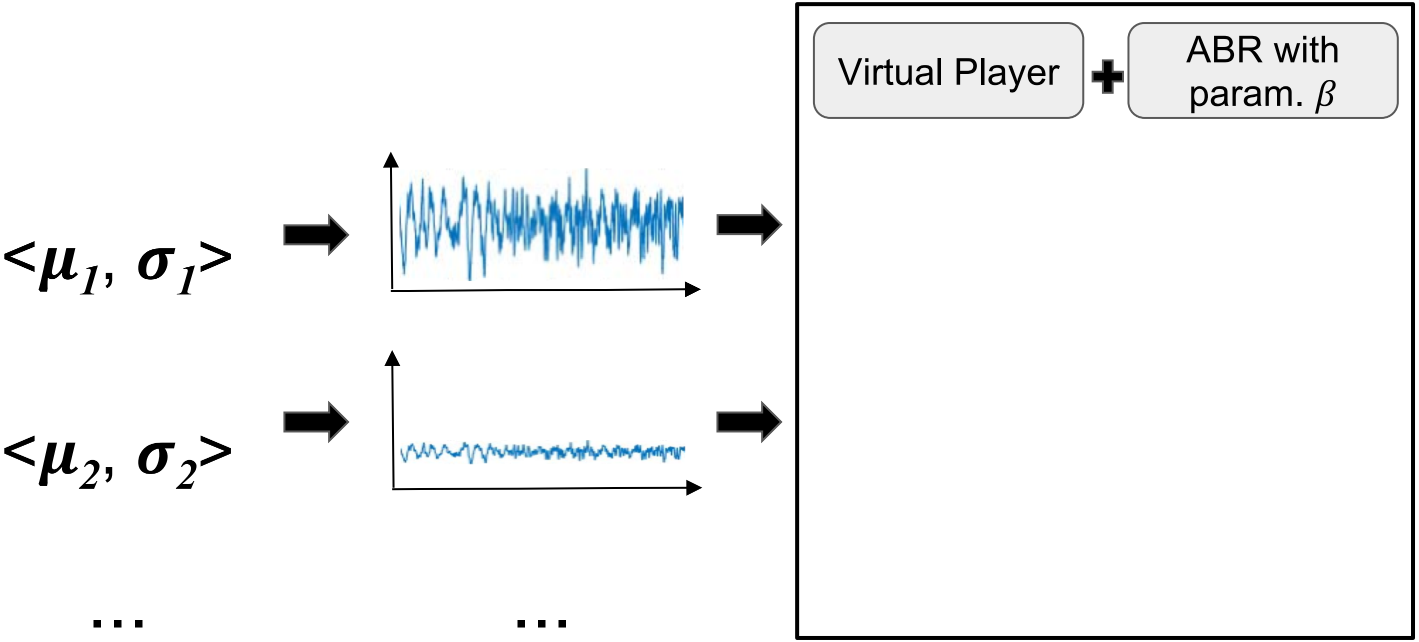
$<\mu_1, \sigma_1>$ ➡ 

$<\mu_2, \sigma_2>$ ➡ 

… …

Generate **synthetic stationary traces** for each network state

Finding the best parameter : Oboe Offline Step 2

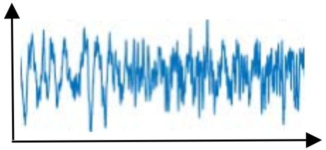


Explore **parameter space** for each state and get **QoE vectors**

Finding the best parameter : Oboe Offline Step 2

Virtual Player + ABR with param. β

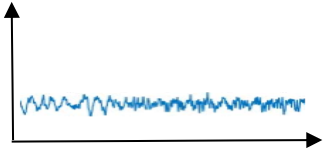
$\langle \mu_1, \sigma_1 \rangle$ ➡



➡

Param	β=0.1	β=0.2	β=0.3	…
QoE				…

$\langle \mu_2, \sigma_2 \rangle$ ➡



➡

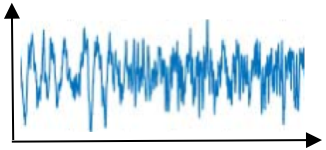
Param	β=0.1	β=0.2	β=0.3	…
QoE				…

… … …

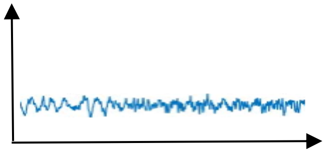
Explore **parameter space** for each state and get **QoE vectors**

Finding the best parameter : Oboe Offline Step 2

$\langle \mu_1, \sigma_1 \rangle$



$\langle \mu_2, \sigma_2 \rangle$



…

Virtual Player + ABR with param. β

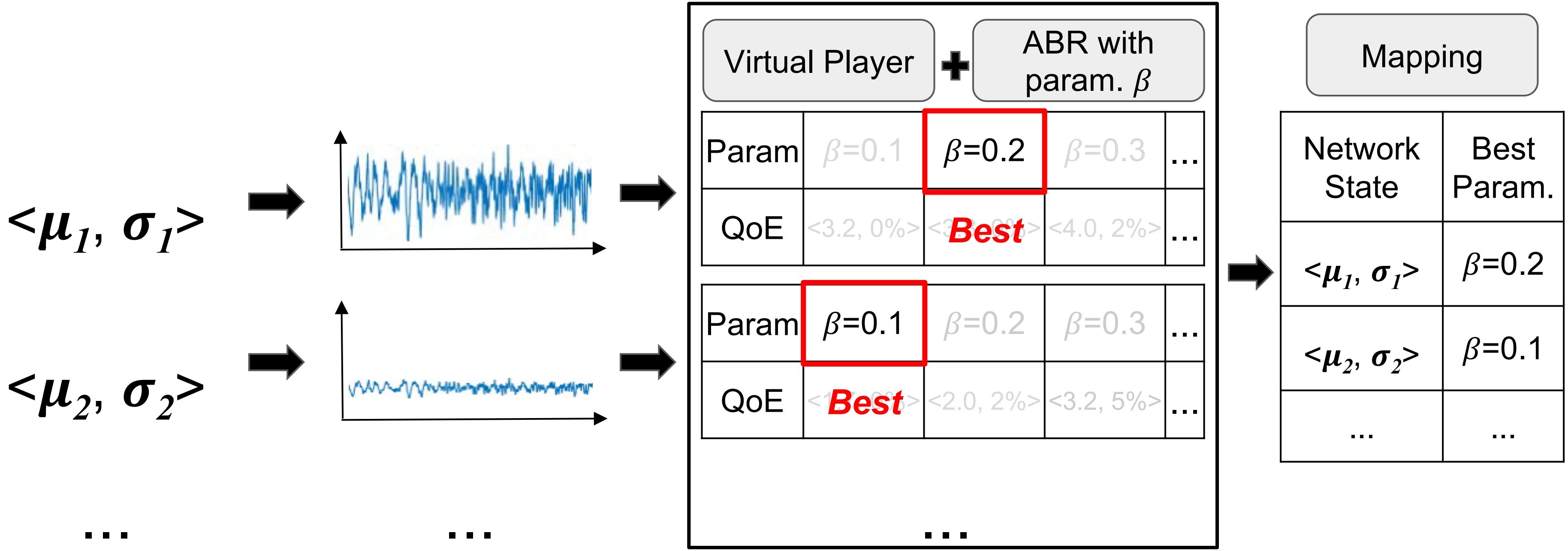
Param	β=0.1	β=0.2	β=0.3	…
QoE	<3.2, 0%>	<3.8, 0%>	<4.0, 2%>	…

Param	β=0.1	β=0.2	β=0.3	…
QoE	<1.7, 0%>	<2.0, 2%>	<3.2, 5%>	…

…

Explore **parameter space** for each state and get **QoE vectors**

Finding the best parameter : Oboe Offline Step 3



Find **the best parameter** by vector dominance for each state

Oboe Offline Stage: Design Questions

- **Use real or synthetic traces?**
- **How to quantize network state space?**
- How to reduce the cost of parameter space exploration?
- How to decouple ABR algorithms from Virtual Player?
- How to take publisher preferences into account?

Key Challenges

How to model network conditions?

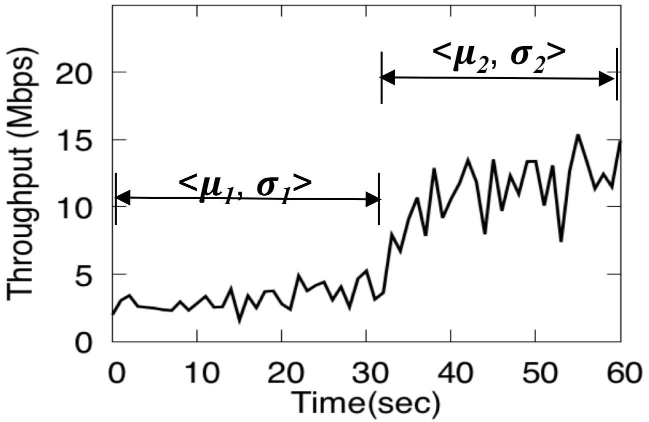
How to find the best parameter for each network state?

- Oboe Offline Stage

How to adapt to changes in network state?

- Oboe Online Stage

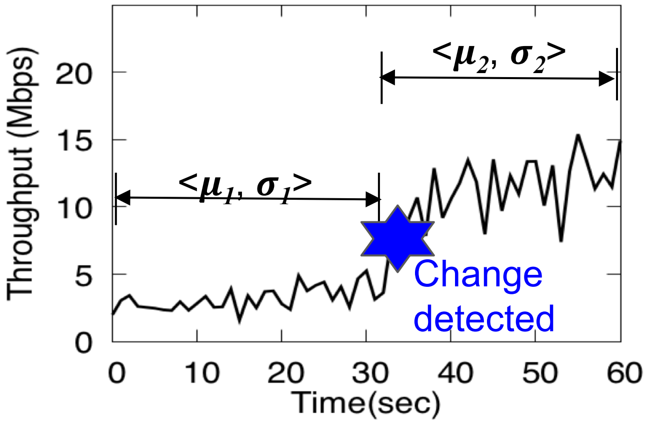
Adapting to network state changes



Online change point detection[11] algorithm identifies network throughput distribution changes in real time

[11] Ryan Prescott Adams and David JC MacKay. Bayesian Online Changepoint Detection. In arXiv:0710.3742v1, 2007

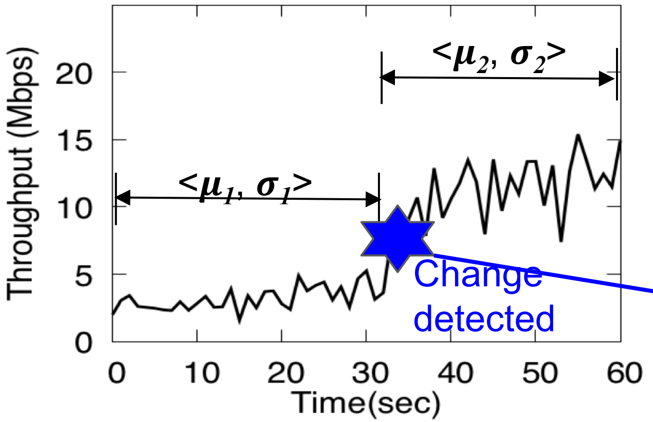
Adapting to network state changes : Online Step 1



Online change point detection[11] algorithm identifies network throughput distribution changes in real time

[11] Ryan Prescott Adams and David JC MacKay. Bayesian Online Changepoint Detection. In arXiv:0710.3742v1, 2007

Adapting to network state changes : Online Step 2

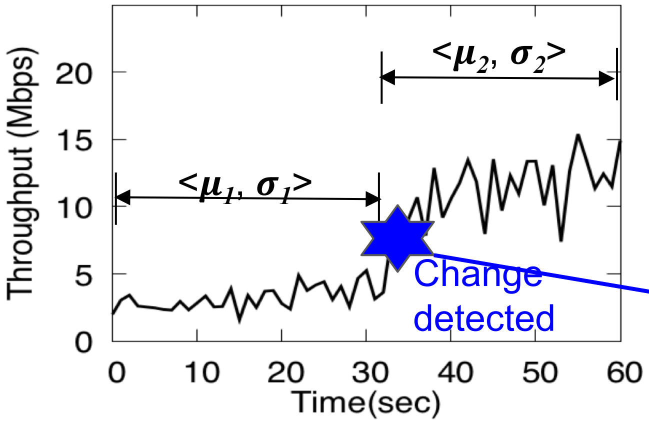


Mapping

Network State	Best Param.
$<\mu_1, \sigma_1>$	β=0.2
$<\mu_2, \sigma_2>$	β=0.1
...	...

Find the best parameter from the mapping for a new state

Adapting to network state changes : Online Step 3



Mapping

Network State	Best Param.
$<\mu_1, \sigma_1>$	β=0.2
$<\mu_2, \sigma_2>$	β=0.1
...	...

ABR algorithm parameter

β=0.2 → β=0.1

Reconfigure ABR algorithm parameter

Oboe Online Design Questions

- **Why not a simple moving average?**
- How many throughput samples to detect changes?
- Are computational overheads acceptable in real time?

Evaluation methodology

Comparison

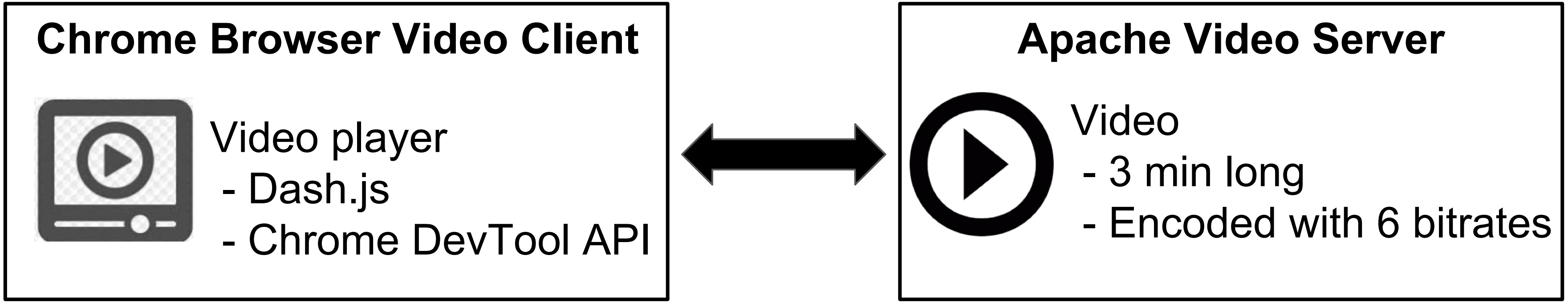
- Existing ABRs vs. **Existing ABRs + Oboe**

QoE Metrics

- Average Bitrate
- Rebuffering Ratio
- Bitrate change magnitude
- QoE-lin (Linear combination of three metrics)

Evaluation methodology

TestBed Setup



Dataset

- 600 throughput traces from real users
- Real users used a desktop or a mobile
- Upto 6 Mbps

Various Evaluations

Method	Detail
ABRs performance	**MPC**, BOLA, BB, HYB and **Pensieve**
Public datasets	HSDPA[11] and FCC[12]
Various settings	Live setting and different videos
Alternative predictors	Ideal predictors on MPC
Publisher specification	Different rebuffering tolerance
Pilot Deployment	Partial deployment on AWS

[11] Haakon Riiser, et. al. MMSys, 2013
[12] Federal Communications Commission. Raw Data 2016

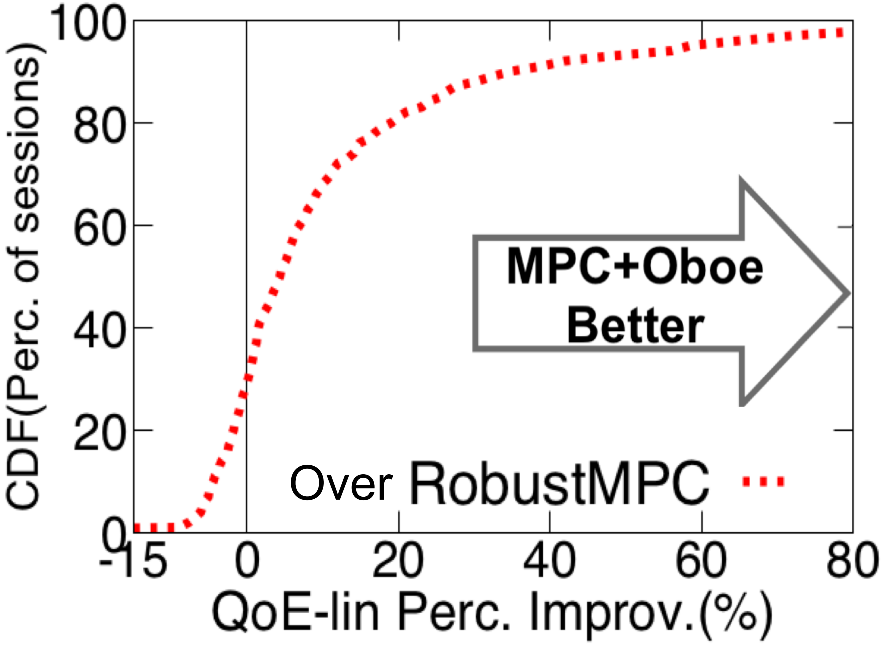
RobustMPC vs MPC+Oboe

Solves an optimization problem to choose bitrates using **predicted throughputs** and **player buffer occupancy**

Discount factor *d*

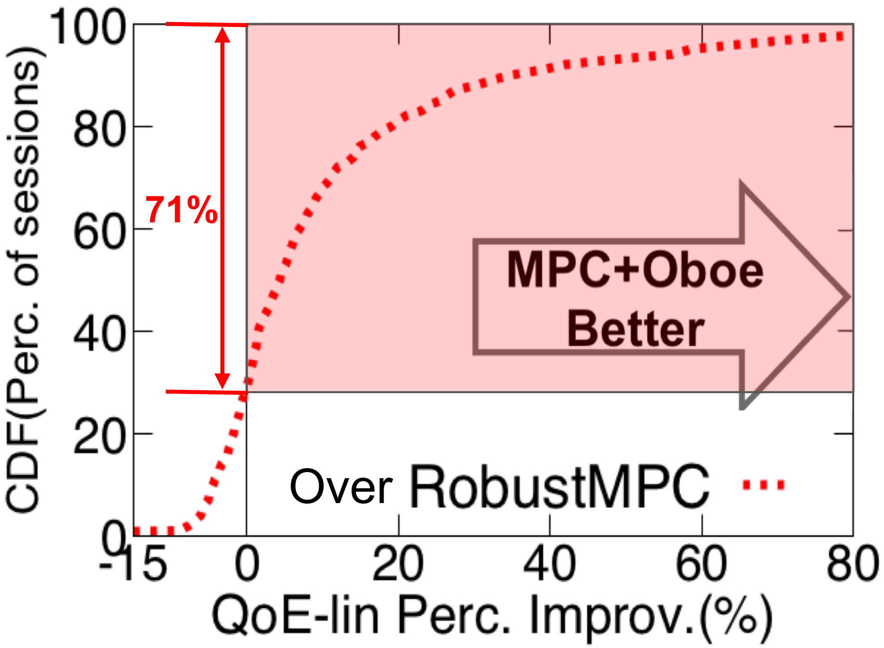
- Reduces a predicted throughput by ***d*** to compensate prediction errors
- Simple heuristic based on previous prediction errors

RobustMPC **vs MPC+Oboe**



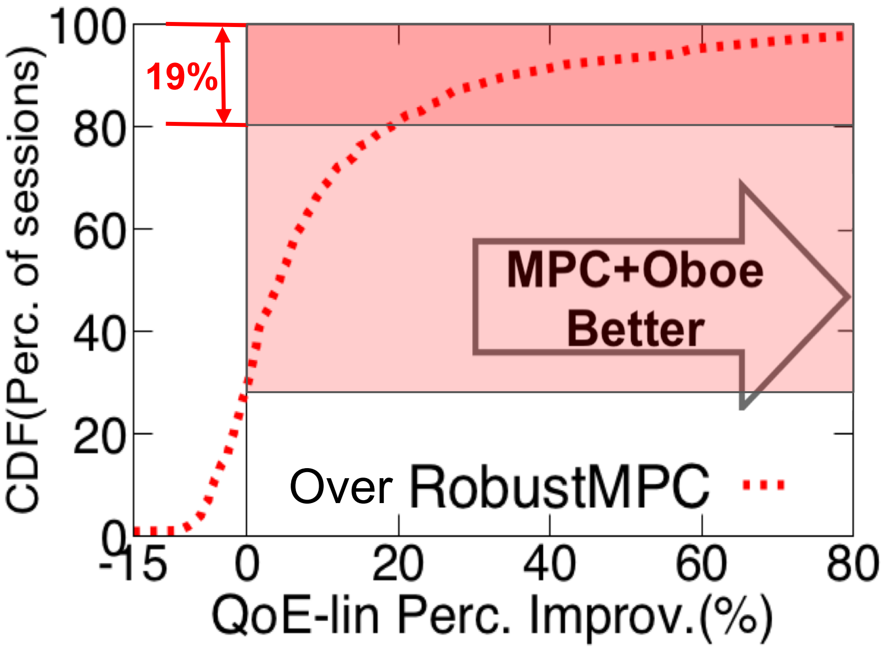
- Improves QoE-lin **for 71% of sessions**
- For **19%** of the sessions, more than **20%** benefit

RobustMPC **vs MPC+Oboe**



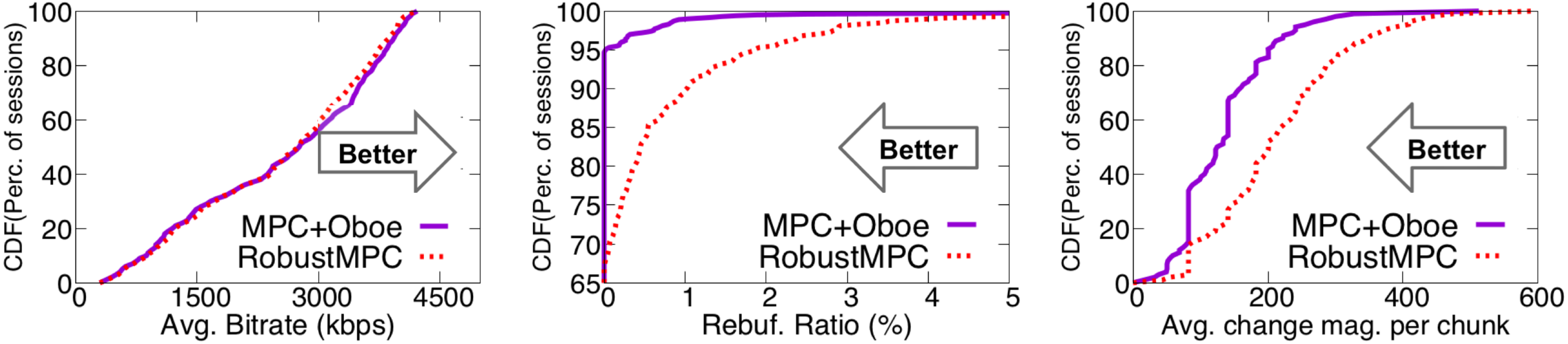
- Improves QoE-lin **for 71% of sessions**
- For **19%** of the sessions, more than **20%** benefit

RobustMPC **vs MPC+Oboe**



- Improves QoE-lin **for 71% of sessions**
- For **19%** of the sessions, more than **20%** benefit

Where does benefit come from over RobustMPC

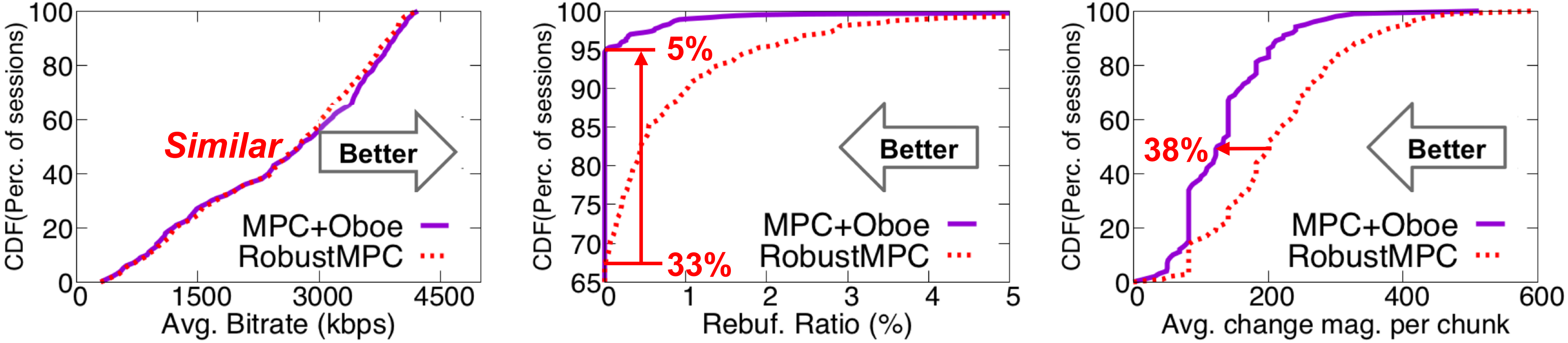


Similar average bitrates

Reduces the # of sessions with rebuffering **from 33% to 5%**

Improves the median per chunk change magnitude **by 38%**

Where does benefit come from over RobustMPC

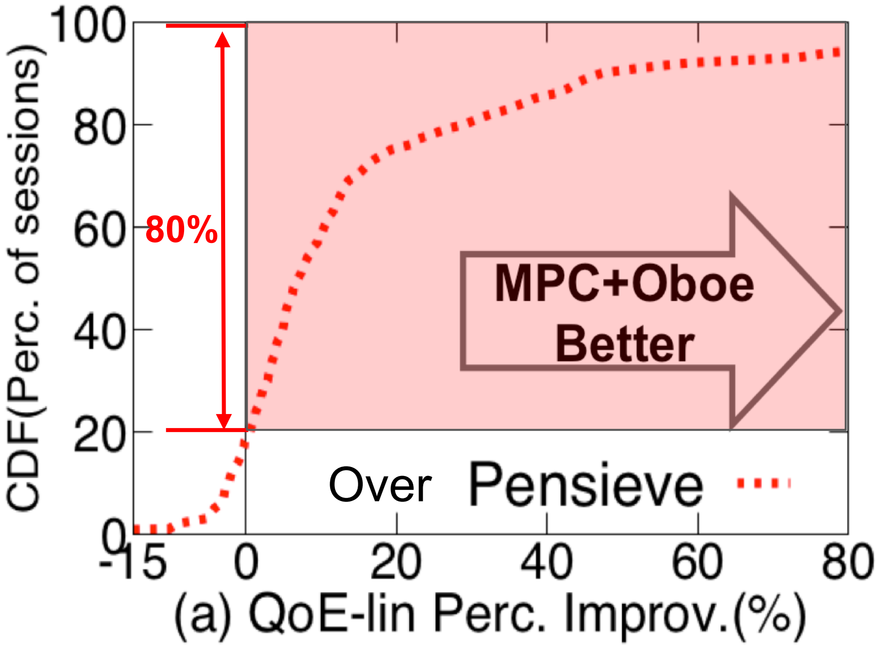


Similar average bitrates

Reduces the # of sessions with rebuffering **from 33% to 5%**

Improves the median per chunk change magnitude **by 38%**

MPC+Oboe vs Pensieve



- Improves QoE-lin **for 80% of sessions**
- **24%** better in average QoE-lin

Where does benefit come from over Pensieve?

- Hypothesis : Pensieve performs better when it is trained with a constrained throughput range

Model	Trained	Tested	Result
Pen-Specialized	0 - 3 Mbps	0 - 3 Mbps	Better
Original Pensieve	0 - 6 Mbps		

- Pensieve unable to specialize to network conditions
- **Oboe specializes parameters for every network state**

Summary

Oboe is a system to make ABR algorithms work better in a wide range of network conditions

- by auto-tuning parameters to current network state

Oboe is general

- Can be applied to many existing ABRs
- Improves existing ABRs upto 38% in QoE metrics
- Outperforms Pensieve by 24% in average QoE

Live demo tomorrow in the demo session!

https://github.com/USC-NSL/Oboe

Thanks!